AN ABSTRACT OF THE THESIS OF

Dodla Nageswara Rao for the Master of Science
(Name) (Degree)

in Industrial Engineering presented on March 16, 1971
(Major) (Date)

Title: SINGLE AND MIXED-MODEL ASSEMBLY LINE BALANCING METHODS FOR BOTH DETERMINISTIC AND NORMALLY DISTRIBUTED WORK ELEMENT TIMES

Abstract approved: Michael S. Inoue

Line balancing is concerned with the optimal assignment of work elements to individual operators in an assembly line of a mass producing system.

This paper summarizes the assembly line balancing terminology, the computational methods, and objective functions applicable to a wide variety of assembly lines. Single and mixed-model situations for both constant and variable work element times are examined.

A Back Tracking Method of Assembly Line Balancing (BALB) is developed and programmed in FORTRAN IV. BALB, as a manual procedure was able to find an optimal solution to problems that other existing methods such as Helgeson and Birnie's positional weight technique, could not yield. In general, BALB was also found to be simple and more efficient than the heuristic methods by Tonge, Hoffman, Mansoor and Arcus.

The computer program, *BALB, accepts data for both single and mixed-model ALB problems and considers both constant and variable work element times. This program uses production shift time as the criterion for balancing the mixed-model lines.

Numerical examples are used throughout the paper to illustrate the steps of various methods.

Single and Mixed-Model Assembly Line Balancing Methods for Both Deterministic and Normally Distributed Work Element Times

by

Dodla Nageswara Rao

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

June 1971

APPROVED:

Michael S. Inoue

Associate Professor of Industrial Engineering

in charge of major

James L. Riggs

Head of Department of Industrial Engineering

H. P. Hansen

Dean of Graduate School

Date thesis is presented March 16, 1971

Typed by Barbara Eby for Dodla Nageswara Rao

ACKNOWLEDGEMENT

I am greatly indebted to Dr. Michael S. Inoue for his most unforgettable academic guidance, advice and continuous moral support during my graduate work and the preparation of this thesis.

I would also like to express my sincere appreciation and thanks to Dr. James L. Riggs for his personal encouragement throughout my graduate work.

I am also highly grateful to Dr. Justus Seely of the Department of Statistics and Dr. B. Linn Soule of the School of Business for the help and valuable instruction during my graduate work.

Thanks are also due to Dr. Nick T. Thomopoulos of Illinois Institute of Technology at Chicago for his suggestions and encouragement in pursuing this investigation.

Finally I wish to express my appreciation to Mrs. Barbara Eby for her excellent job of typing the final copy and to Type Ink for helping to meet the deadlines.

This research has been partially financed by a grant from the Educational Co-ordinating Council of the State of Oregon.

DEDICATION

To my wife, Mani, without whose encouragement and patience this work would not have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table		Page

Table		Page
5-11	Summary of station assignments for both model 1 and model 2 with variability of work elements.	187
5-12	Data of the standard deviations for all 19 work elements for the 3 model example in Figure 5-5.	191
5-13	Data for the total work content and the variance of each element for 19 element example.	192
5-14	Summary of results with *BALB for 19 element example problem with variability at c = 419 and S. I. = 9.34.	192
5-15	Summary of results with Moodie's phase 1 of the HALB for 19 element example problem with variability at c = 419 and S. I. = 13.21.	192
5-16	Summary of results for the elements of sequencing.	196
5-17	Cyclical sequencing of units, for the mixed-model problem by fixed rate launching.	206
6-1	Present state of art of ALB models.	211

SINGLE AND MIXED-MODEL ASSEMBLY LINE BALANCING METHODS FOR BOTH DETERMINISTIC AND NORMALLY DISTRIBUTED WORK ELEMENT TIMES

I. INTRODUCTION

A. Assembly Line Balancing (ALB) Problem

Assembly line balancing is an important and challenging problem facing industrial engineers in today's mass production oriented society. A survey conducted in the United States by Lehman (1969) showed that the task of assembly line balancing had been assigned to an industrial engineer in more than one half of the companies responding to the survey. A basic characteristic of an assembly line is the movement of individual work piece from one work station to another by means of an assembly conveyor. The tasks required to complete the assembly of a product are divided among the operators so that a given worker performs the same set of operations to every work piece that passes him.

> Assembly line balancing is a constrained combinatorial optimization problem for which the constraints are in the form of a precedence network. The problem can be briefly described as the assignment of tasks in the assembly of a product to work stations in the line so to optimize a measure of efficiency (Roberts and Villa, 1970, p. 361).

B. Historical Review

According to Kilbridge and Wester (1961, p. 292),

> The first progressive assembly line was started at the Ford Highland Park plant in 1913, and Henry Ford is properly credited for its invention. He combined the long known principles of the division of labor, the fabrication of interchangeable parts, and the movement of product past fixed work stations into the concept of assembly as a continuous process.

Bryton (1954) by his master's thesis became the first person to treat the line balancing problem in an analytical manner. The first published analytical statement of the line balancing problem was by Salveson (1955), who gave a thorough description of the problem with respect to a practical assembly situation.

About 15 years of subsequent research by engineers and mathematicians on this line balancing problem has resulted in a variety of solution methods ranging from rigorous mathematical techniques to heuristic routines, offering optimum to near optimum solutions. A review of the existing literature revealed that almost every author who tackled the problem used a different notation. These diversified notations run against all efforts to unify the theory of operations research. An attempt has been made in this thesis to present a standard ALB model. It is recognized, however, that certain model formulations (e. g., mixed-model assembly lines) will require additional nomenclature which will be referred to in the later chapters of this thesis. VanGigch (1965) and Ignall (1965) presented standard formulations of the line balancing problem, but the former dealt with only five analytical approaches while the latter did not extend the formulation to mixed-models and variable work element times.

C. Recent Developments

Earlier assembly lines were used to assemble identical products. With the growing need to manufacture a variety of models to meet customer demands, many assembly lines have now been transformed to handle multi-model production schedules. When several models of the same general product are assembled on the same conveyor line, it is commonly called a mixed-model or multi-model assembly line.

Though much work has been carried out in the past on the methods of single model line balancing, very little has been reported in the literature with respect to mixed-model line balancing. Thomoponlos (1967) and Roberts and Villa (1970) considered mixed-model line balancing problem assuming deterministic elemental times in the assembly process. But in the real-world situations the elemental times of an assembly process are independent and identically distributed random variables approximated by the normal distribution. Practical studies by Hicks and Young (1962) and Walker (1958) confirmed this assumption. Based on this variability of work elemental times, attempts have been made (e.g., Freeman, David, 1968) to minimize the total product cost associated in an assembly rather than to minimize the direct labor cost, used in traditional approaches. Klein (1963) proposed a method of balancing an assembly line using feasible linear sequences. He concluded saying

> ... It is unfortunate that at present an answer to the question of practicability does not seem to be available, since there apparently is no formula or prescription on hand to determine the number of feasible orderings for a given problem (Klein, 1963, p. 281).

This conjecture expressed by Klein has now been solved by Okamura and Yamashina (1969) who succeeded first in developing an algorithm to identify all distinct feasible sequences and then finding an optimal sequence using Little _et_ _al_. (1963)'s Branch and Bound Algorithm.

There are a limited number of computer programs developed by major industries to suit their particular assembly situation, such as Target Job Line Balancing (TJLB) by Cnossen (1967) used in Ford Motor Company and commercial soft ware packages that are available to the participants of the Advanced Assembly Methods (AAM) program conducted by Illinois Institute of Technology Research Institute (IITRI, 1970).

D. Outline of the Thesis

In this thesis the definitions and terminology encountered in the existing literature on line-balancing problem is followed by a standard formulation of the problem. A Cause and Effect diagram has been developed in Chapter II. This diagram gives a pictorial representation of the controllable factors and the stated objectives for a practical ALB procedure.

Chapter III summarizes and illustrates the traditional analytical and heuristic approaches applicable to single-model manufacturing situations.

A Back Tracking Method of Assembly Line Balancing (BALB) has been developed in Chapter IV to balance either single or mixed model assembly lines for both deterministic and normally distributed work element times. This method, programmed in FORTRAN IV, attempts to find an optimal minimal station balance at the given cycle time (production shift time in the case of a mixed-model line). If an optimal solution is not obtained by this method at the given cycle time, near optimal solutions can be attained either by incrementing the cycle time by units of 1 or at an optional cycle time. The listing of the program appears in the Appendix.

In Chapter V extensions of mixed-model ALB procedures is made for variable work element times. A summary of the solution algorithms developed for sequencing the various models on a mixed-model assembly line is given at the end of Chapter V. The Table 1-1 at the end of this chapter gives the summary of the classifications for ALB models.

Table 1-1. Summary of classifications for ALB formulations.

Approaches	Assembly Line	Objective Function	Work Element Times	Procedures	Chapter Section
Traditional	Single Model	Minimize direct labor cost	Deterministic	Analytical Heuristic	III
		Minimize direct labor cost	For both deterministic and normally distributed	Heuristic	IV-A
Extensions	Single Model	Minimize total perturbation costs	Deterministic	Heuristic	IV-B
		Minimize total production cost	Variable	Search	IV-C
	Mixed Model	Minimize direct labor costs	Deterministic	Analytical & Heuristic	V-A
			Normally distributed	Analytical & Heuristic	V-B
		Minimize sequence delay costs	Deterministic	Search	V-C

II. ALB MODEL FORMULATION

The development of different formulations of the ALB problem has led to the conception of various terms and conventions by different authors. It is felt that the elucidation of all the definitions and terminology conceived by different authors is of much importance for a standard formulation of the problem.

It is also realized that the objective of either minimizing a cycle time for a given number of stations or minimizing the number of stations for a given cycle time has the same connotation of minimizing the total idle time or the direct labor cost per unit assembled.

In the following pages a summary of the ALB terminology precedes the development of a cause and effect diagram and the standard formulation of an ALB model to minimize the costs per unit assembled, either of direct labor or of total production in the assembly process.

A. Definitions and Terminology

The definitions and terms encountered in a practical ALB procedure are summarized below and are arranged in an order of relevance to the problem.

1. Assembly Process. The overall work that is to be accomplished in a line production.
2. Work Element. A rational division of the total work content

in an assembly process; an element is represented here by u_i, where i is the identification number with the range $1 \leq i \leq n$. The number n indicates the total number of work elements required to complete a product assembly. "Task" and "operation" are other terms sometimes used in place of "work element".

3. Job. A job can be defined as an aggregate of tasks or work elements. Some authors use the term "job" to mean "work element" (e. g., Cnossen, 1967).

4. Work Station. A location on the assembly line where a given amount of work is performed by an operator. Assembly line work stations are generally manned by one operator. However on short runs an operator may man more than one station, and on lines of large products (aircraft, autombile etc.) work stations are frequently manned by several operators.

The following four general types of stations are identified in actual practice. They assume a limited range of movement of the operators within their stations adopting the convention that the conveyor line moves from left to right (Thomopoulos, 1967, p. B-61).

a. closed station. In this station, the work must be performed within the limits of the station. This happens, for example, in pits or in paint booths where the assigned work cannot be accomplished outside the station. The symbol, [ ], will represent a closed station.

b. open station. In an open station the operator will be allowed to move in either direction outside his station up to some specified limits. The limits are necessary to prevent him from either moving more than the desirable distance away from his work station or from entering a closed station. The symbol, ( ), will represent an open station.

c. closed-to-the-right and open-to-the-left station. This station is a combination of the open and closed stations. The symbol, ( ], will represent this type of station.

d. closed-to-the-left and open-to-the-right station. This station is also a combination of open and closed stations. The symbol, [ ), will represent it.

5. Operator. An individual who does specific work assigned upon the units of a product, during a progressive assembly as they are conveyed through his work station.

6. Work Element Time. To each work element i will be associated a performance time t_i. The sum of the performance times of all the elements is the total work content in each product being assembled. The value of t_i can be either an integer or a positive fraction. Most of the authors treat t_i as a deterministic time.

[According to Riggs (1970, p. 322)] the operation times are far from constant in an actual production line. Performance by human operators is continually modulated by enthusiasm, health, and social conditions.

The research by Hicks and Young (1962) and by Walker (1959) show, moreover, that the time an operator takes in performing a task is an independent and identically distributed random variable and can be approximated by a normal distribution.

7. Work Station Time. The actual amount of work, usually in minutes assigned to a specific station on the assembly line is termed as work station time. If t_i is the unit time of the element i and T_k is the kth station time, then $t_i \leq T_k \leq c$ where c is the cycle time defined below.

8. Cycle Time. Cycle time is the amount of time the product spends at each work station on the line when the line is moving at a standard pace (100 percent efficiency). It is the amount of time elapsed between successive units as they move down the line at a standard pace. The feasible cycle time c with a particular line design (i. e., when a particular group of elements are being assigned to stations) will satisfy the inequality $c_L \leq c \leq c_H$ where $c_L = \max_i\{t_i\}$ represents the lowest feasible cycle time and $c_H = \max_k\{T_k\}$ represents the highest feasible cycle time while $T_k = \sum_{i \in J_k} t_i$, where J_k denotes the subsets of tasks aggregated at the kth station. For mixed-model scheduling the time that separates the launching of two consecutive units (either of

the same or of different models) on the main conveyor line is termed as production cycle time.

9. Station Idle Time. This is the amount of time an operator is idle due to the difference between the cycle time and his work station time. The symbol d_k denotes the idle time at the kth station.

10. Balance Delay. This is the amount of idle time on the line due to the imperfect divisibility of assembly work between stations. In practice those operators having shorter work assignments will not actually stand idle at the end of each cycle but will work continuously at a slower pace. The effect measured in terms of labor cost, however, is the same as if they were idle part of the time and working at a faster pace the rest of the time. The degree or the percent of imbalance, called "balance delay", is the ratio between the average idle time at the stations and the maximum operator time (cycle time); i. e. ,

$$d = \left(\frac{c - \bar{c}}{c}\right) \times 100$$

where d = percent balance delay

c = cycle time for a particular production line design

$\bar{c}$ = average station time

If an assembly is manned by k operators, then,

$$d = 100\left\{\frac{kc - k\bar{c}}{kc}\right\} = 100\left\{\frac{kc - \sum_{i=1}^{n} t_i}{kc}\right\}$$

Subject to the condition $(kc - \sum_{i=1}^{n} t_i) > 0$

where t_i, $(i = 1, 2, \cdots, n)$ is the i-th elemental time and $\sum_{i=1}^{n} t_i$, is the total work content time which is constant for a given assembly process.

For a given value of c and total work content time Σt_i, there exists a minimum number of operators, m^*, given by the bracket function $[\frac{\Sigma t_i}{n}]$ where $[x]$ = smallest integer $\geq x$. The maximum possible value of k is n, the total number of work elements under study. n_{min} (not necessarily equal to m^*) will indicate the minimal value of the number of operators for a given minimal balance delay which can be defined as

$$d_{min} = \frac{100(n_{min}c - \sum_{i=1}^{n} t_i)}{n_{min}c},$$

and this is a discrete function of cycle time c. The balance delay function tells what cycle time to select for a given distribution of work elements and a given number of operators.

11. Technical Division of Labor. The following four types of costs have been identified with respect to the division of

tasks (Kilbridge and Wester, 1966, p. B-257).

a. imbalance-of-work cost. This results from the imperfect divisibility of productive jobs. By nature the productive jobs are not perfectly divisible, but in extending the division of labor, these jobs must be subdivided into smaller tasks. These sub-tasks must be assigned to separate workers so that each worker has approximately the same amount of work to do in a given time. For any given job, the imbalance-of-work cost rises with the division of labor until it is technically impossible to divide the task further.

b. learning cost. This is the cost incurred by assembly workers in learning to perform their tasks at an acceptably fast pace. In this context "learning" implies "group learning" since all operators on the line must progress at the same rate. When model changes are frequent and employee turnover is high the learning costs may represent a considerable part of direct labor cost.

c. non-productive work cost. This cost is a derivative of division of labor. Handling of product from worker to worker, time spent in starting and stopping work on each unit of product and the increased communications and control necessitated by the interdependence of functions represent this cost.

d. wage cost of skill. This relates the division of labor to deskilling of work. As tasks become more specialized, range of skills required to perform each is narrowed, and the workers of general skill are no longer required.

In general, the imbalance of work and non-productive work tend to increase with the division of labor, while the cost of learning and wage cost of skill tend to decrease. The productivity, in a long run, however, may suffer due to job specialization in product assemblies.

12. Feasible sequence. A feasible sequence is one that may be performed in the indicated order without the prior completion of any other task. The generation of feasible sequences

is the backbone of the assembly line scheduling problem. If there are n tasks, they can be arranged in $n!$ distinct sequences. Because of precedence relations, only some of these $n!$ will be feasible. If there are r precedence relations among the n tasks (r arrows on the directed graph) then there are roughly $n!/2^r$ distinct feasible sequences (Ignall, 1965).

The approaches adopted by Held, et al. (1963) and Klein (1963) reflect the importance of the generation of the feasible sequences in the line balancing algorithms. Only Okamura and Yamashina (1969) have succeeded in finding the procedure for the generation of feasible linear sequences.

13. Feasible subset. A feasible subset is a subset of n tasks that can be executed in some order without the prior execution of any other tasks. We let J_k represent the subset of operations aggregated at the kth station.

14. Smoothness index. A number used by Moodie and Young (1965) to indicate the relative smoothness of a given assembly line balance. It consists of the square root of the sum of the squares of the time deviations, $\sqrt{\sum_{k=1}^{m} (d_k)^2}$, for each of the stations in the balance from the maximum station time. Ignall (1965, p. 252) points out that:

an objective of minimizing idle time is superior to the objective of minimizing "smoothness index" since smoothness does not have the cost interpretation that idle time has.

15. Sub-assembly line. Some independent assembly works can be performed either on the main assembly line or off the main line. In Figure 4-5, the off-line coil subassembly is enclosed in dotted lines. The sub-assembly lines can be parallel and very near to the main line or they can be situated completely in a different location and the semi-assembled product brought to the main line.

In the on-line sub-assemblies, the assembly work is usually done on the particular sub-assembly component by a main line operator before the component is assembled onto the product. For example, the sub-assembly of a carburetor is done on the main line intended for engine dress up in an automobile company (Figure 4-5).

16. Labor groups. An assembly line may consist of several labor groups when a work performed in one labor group cannot be performed in any other group. Each labor group is comprised of work stations with one or more operators manning each station. Each operator is assigned to only one labor group.

17. Production schedules. Production schedules are derived from the demand forecast. By knowing a daily production schedule of the units to be assembled on the conveyor line, the cycle time is arrived at by the equation,

$$c = 1/\text{production rate} = 1/(Q/T) = T/Q \quad \text{where}$$

$$T = \text{total productive time available/day, and}$$

$$Q = \text{number of units to be assembled in period } T.$$

This value of c is used in balancing a single model assembly line. A company manufacturing a variety of models of the same general product will have a schedule for each model to be assembled. If $j(j = 1, 2, 3, \cdots, J)$ represents the model to be produced on the assembly line and N_j, the number of units per model to be assembled in duration T, then the total number of units to be assembled in period T is equal to $N = \sum_j N_j$. The period T will be used as a basis of balancing the mixed-model line(s).

18. Conveyor belt speed. If v represents the conveyor belt speed then $v = \ell/c$, where ℓ is the fixed length separating two consecutive units on the conveyor and c is the cycle time.

19. Assembly line inefficiencies. Assembly stations are subjected to four kinds of inefficiencies (Thomopoulos, 1967).

With the assumption that the conveyor belt moves from left to right, the inefficiencies are defined as follows:

a. idleness. This can occur in all four types of stations mentioned earlier. It results when an operator is kept idle waiting for work to enter the upstream limit of his allowable work area.

b. work deficiency. This can occur in two types of stations, open and open-to-the-left stations. It results when products flow through a station so slowly that the operator is able to complete work on a product before the next product has entered his station, and must leave his station to the left to start assembly.

c. utility work. A utility work can occur in all four types of stations. It results when products flow through an operator's downstream limit of his work area faster than he can complete work on them. In this situation, one or more utility workers may be assigned to the station to assist the operator, so that the work on the product is complete, or else the unfinished work is completed in a touch-up station farther down the lines.

d. work congestion. This can occur in two types of stations, open and open-to-the-right stations. It results when products flow through a station faster than the operator can complete work on them, forcing him to move out of his station to the right.

20. Position restrictions. These consist of operator-product and operator-line relationships, and each element on the diagram must be coded (either by a letter, color or by a geometrical code) to describe the restrictions imposed on it. These are also known as Front and Back, Top and Bottom, Right or Left restrictions. In Figure 4-5, 4-7 R represents that the element must be done on the right side of the

conveyor, L represents that the element must be done on the left side of the conveyor while E stands for either side.

21. Fixed facility (or locational) restrictions. Fixed facilities imply machine tools, processes, testing facilities and indexing stations that are an integral part of the assembly line and form immovable stations. Such restrictions decrease the commutability of work elements and cause rigid ordering. These elements are identified on the precedence diagram by placing an asterisk (for example, elements 36 and 40 in Figure 4-7). Additional comments must be made on the data sheets about those elements. A schematic drawing of the fixed facility location and a process flow chart giving the details of this fixed facility usually accompany the precedence diagram.

22. Multi-option elements. Some elements that can be performed either on the sub-assembly or on the main line are called multi-option elements. These are shown "boxed" rather than circled to identify them as elements that appear twice on the diagram.

23. Closely related elements. Sometimes two or more elements are closely related, requiring that the performance of the first element of the group be immediately followed by the performance of the other elements. It would not be proper

to combine these tasks into one, since it is possible to have them performed by successive operations. Such a group is denoted by enclosing it with a solid line. For instance, due to safety reasons, the placement of the picture tube, element 14, must immediately be followed by certain fastening operations, elements 16 and 17 (Figure 2-1).

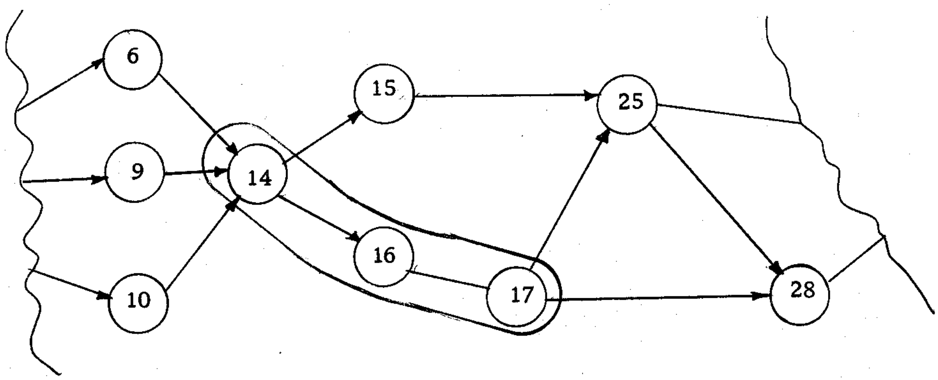


Figure 2-1. A part of the precedence diagram for work elements on television line (Prenting and Battaglin, 1964, p. 210).

24. Target jobs. A list of target jobs were defined by Cnossen (1967) to develop a heuristic method known as Target Job Line Balancing (TJLB). These consist of:

a. the last job in a job set where the job set is defined as one or more jobs related to a sub-assembly. Tasks 6, 16, 19 and 43 in Figure 4-7, correspond to this list.

b. the elements restricted by fixed facility. Tasks 36 and 40 in Figure 4-7, fall in this category.

25. Minimum perturbation (Cnossen, 1967). Using the Target Job list, heuristics were developed to minimize perturbation. Minimization of perturbation is defined as the attempt to prevent the relocation of relatively fixed facilities on the assembly line, and to preserve as much as possible the existing work assignments of the assembly operations. This is an important criterion for assembly lines where model changes occur frequently.

26. Transferability and permutability of work elements. Transferability is the property of elements which can be moved laterally from their stages (columns) to positions to their right without disturbing the precedence restrictions in a precedence diagram (Figure 3-11) constructed by Jackson (1956)'s method. Permutability is the property of the elements which can be moved among themselves in any work sequence without violating restrictions on precedence relations. Kilbridge and Wester (1961) exploited these two properties of work elements for developing a heuristic method of assembly line balancing.

27. Positional weight. This is a mere number obtained by adding together the time values for the specific work element and all work elements that must follow as defined in a precedence matrix. This is the criterion used in developing a

heuristic method by Helgeson and Birnie (1961). The criterion of three different types of positional weights; 1) linear positional weight, 2) logarithmic positional weight, and 3) square positional weight, has been used to develop a new balancing technique in this thesis (Chapter IV).

28. Trades and transfers. These are some of the heuristics used by Tonge (1960) and Moodie and Young (1965) to shift tasks between stations in an attempt to reduce idle time.

29. Chain and set (Tonge, 1960). A chain is a group of adjacent elements whose relative order is completely determined, each except the first having a single direct predecessor and each except the last having a single direct follower (e. g., set v_2 and element u_{15} in Figure 3-6a).

A set is a group of elements whose relative order is completely unspecified, all having the same direct predecessors and followers (e.g., elements u_{10}, u_{11}, u_{12} from the set v_2 in Figure 3-6a).

30. Bowl rule. According to Hillier and Boling (1966) the traditional rule stating that work should be distributed evenly among the work stations is no longer considered the best rule for all systems. They showed that for certain systems a balancing rule which they call the "Bowl Rule", is in fact an improvement over the equal balance. The bowl rule states

that the end stations should have more work in terms of the station times than the stations in the center of the conveyor line. The term "bowl" depicts the shape of a curve obtained with the station time on ordinate and the stations on abscissa.

31. <u>Buffer inventory</u>. Items located between work stations as a float to lessen the impact of blocking. Such buffers of inventory can exist between any or all stations (Freeman and Jucker, 1967, p. 361).

32. <u>Blocking</u>. A station is said to be blocked if work on the item is complete but the item cannot pass to the next station because the operator is busy and no available space for in-process inventory exists, or if the operator passes an item to the next station but cannot receive a piece since the preceding station is busy and no available inventory exists from which items can be drawn. This will occur in case of closed-to-the-left stations mentioned earlier.

33. <u>Paralleling (Freeman and Jueker, 1967)</u>. Paralleling implies duplicating the facilities (machine tools etc.). It may be possible to increase output or reduce the number of operations required on a line by paralleling certain work stations.

34. <u>Dynamic programming ALB model</u>. Using Bellman's dynamic programming method (1957) a mathematical solution procedure for line balancing problem was developed by Held <u>et al</u>. (1963). A rule conforming to the principle of optimality

is used. It can be stated as, that an optimal sequence must have the property that regardless of the route taken to enter a particular state, the remaining decisions must constitute an optimal sequence for leaving that state. To apply this principle the precedence diagram must have one starting node and one ending node.

35. Shortest route criterion. Gutjahr and Nemhauser (1964) and Reiter (1969) treated the line balancing problem as a finite and directed network with a source and sink to find the shortest route between the two nodes. The shortest route indicates the minimal number of arcs where each arc in the directed network represents the idle time when going from one state to the next. A state is defined as a collection of elements feasible for a station assignment without the cycle time constraint.

36. Inter-departure time. This is the time between successive units coming off the end of an assembly line. This is equal to the cycle time of the line for the deterministic elemental times.

B. Precedence Diagram

The basic convention adopted to represent an assembly process is the precedence graph. The data of the precedence graph when arranged in a matrix form is called a precedence matrix. The notations used for the construction of a precedence diagram and the related terms are summarized below.

1. Precedence graph. The tasks are represented in a graph along a path following the technological precedence restrictions. Precedence graph is also called by such other names as "precedence diagram", "directed graph", etc. If a commodity has been manufactured, a precedence graph can be constructed free from inconsistencies. Arrows are optional in the preparation of this diagram, but inclusion of arrows will aid in identifying the relation between two tasks. Whenever a task, a, must precede another task, b, the arrow is drawn from a to b and is read as "a precedes b". If two tasks are unordered with respect to each other, they are not connected by a direct line (Figure 2-2).

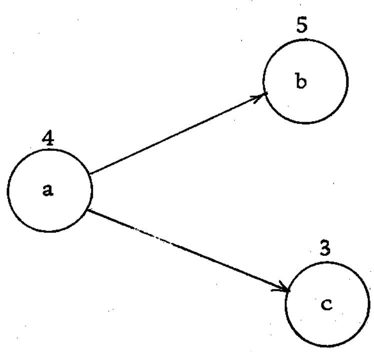


Figure 2-2. Illustration of precedence diagram and diagramming notation.

In Figure 2-2, note that a precedes b (apb) and a precedes c (apc) while b and c are unordered with respect to each other. The basic purpose of a precedence diagram is to convert the actual assembly line station into a diagrammatic representation that completely describes the work element for the purpose of balancing the line. PERT and CPM basically employ the similar diagram. The preparation of a precedence diagram in an actual industrial situation is thoroughly discussed by Prenting and Battaglin(1964, p. 201).

2. Diagramming notation. Every task or work element is represented by a circle. The numbers inside the circle identify the various elements of work and the numbers outside the circles (e.g., 4 above the circle a and 5 above b in Figure 2-2) refer to the corresponding time durations. The connection between the circles is made either by an arrow or a line to indicate the precedence relationship and the numbers are assigned in an ascending order from left to right as a path. This convention is commonly referred as techological ordering.

3. Precedence matrix. This is a square matrix of zeros and ones. The precedence diagram is represented in the form of a matrix. Let $P[p_{ij}]$ be a matrix of zeros and ones.

Then

$$p_{ij} = 1 \quad \text{if } ipj$$

$$p_{ij} = 0 \quad \text{if } i \text{ and } j \text{ are unordered}$$

$$p_{ij} = -1 \quad \text{if } jpi$$

4. Dual precedence matrices. To save storage space in a computer, Moodie and Young (1965) split an $n \times n$ square precedence matrix into two lists known as: 1) immediate predecessor matrix (IP - matrix) and 2) immediate follower matrix (IF - matrix). An IP-matrix is a list or an array containing the immediate preceding elements of each individual element while an IF-matrix is a list or an array containing the immediate following elements of each individual element. These two lists are referred to as dual precedence matrices.

C. Cause and Effect Diagram

The cause and effect diagram is an aid to visualize the parameters of a system. This is a basic representation of a thought process which precedes the design and synthesis of a system model. The value of visualization is vivid in most of the pictorial representations such as: Gantt charts, arrow networks, flow process charts and control charts.

> [According to Inoue and Riggs (1971)] imposing mathematical formulation of systems may challenge theorists but often dismay practitioners. While we have need for the sophisticated techniques of operations research, systems engineering, and statistical inference, we first need visual representations to help us identify and define system problems.

A cause and effect diagram (Figure 2-3) is presented showing the causes and effects involved in developing a practical solution method for balancing the assembly lines. The construction of this diagram is based on the steps given by Inoue and Riggs (1971). The problem under investigation, " a practical ALB procedure" is enclosed in the hexagonal symbol at the center. The main shafts to the left of the symbol represent principal causes and to the right represent main effects. Smaller arrows directed toward the major arrows relate control parameters to cause factors or detail the results of the basic effects.

D. Methodology

During the last sixteen years a number of models have been formulated to solve the Assembly Line Balancing (ALB) problem. Kilbridge and Wester (1962) and Ignall (1965) gave a good review of the ALB literature. A recent review of assembly line balancing algorithms made by Cauley (1968) includes an appendix for a bibliography on ALB compiled by Lewin (1967). Most of the ALB models assume deterministic elemental times and seek a solution that either

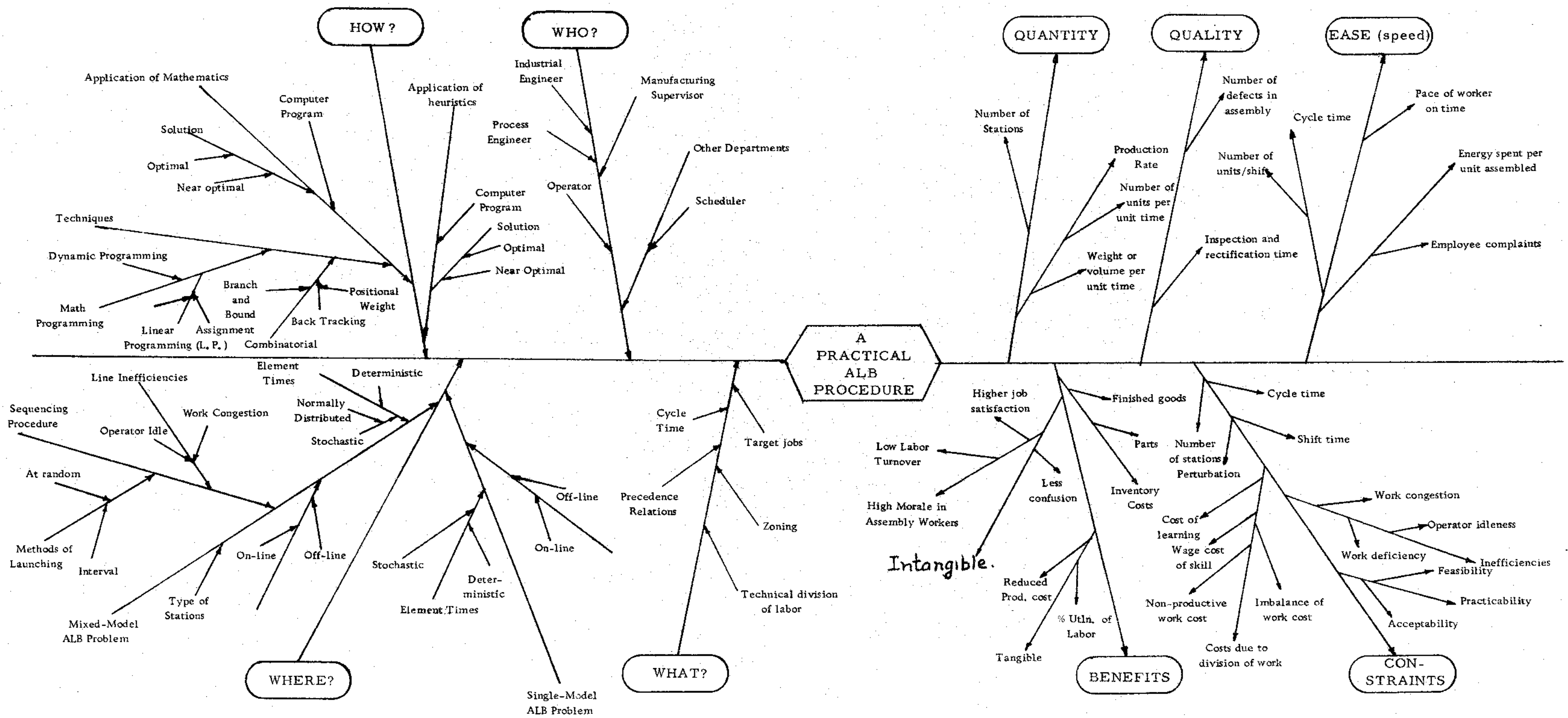


Figure 2-3. A cause and effect diagram for a practical ALB procedure.

minimizes the number of work stations or minimizes the total idle time. Minimizing a cycle time is more appropriate in assembly line situations where frequent changes in production schedules, product mix, and product design often make obsolete a given assembly line balance.

Both the classical formulations, either minimizing cycle time or the number of stations, aim at the same general objective of minimizing the balance delay and thereby attaining a minimum direct labor cost per unit assembled. Freeman and Jucker (1967) and Moodie (1968) remarked that traditional formulations to minimize the direct labor cost were inadequate for an efficient assembly line scheduling system and that a model to minimize the total production cost would be more appropriate in real-world situations. Two such standard formulations can be conceived:

1. Minimization of the direct labor cost per unit assembled.
2. Minimization of the total production cost per unit assembled.

Minimization of the direct labor cost. This can be either for a) deterministic elemental times or b) variable elemental times. Let us first examine the labor cost models in more detail.

a. Deterministic elemental time line balancing models. With the terminology explained earlier, the formulations can be interpreted mathematically as follows. Let

i represent the identification numbers for the indivisible work elements u_i to be assigned for $1 \leq i \leq n$.

t_i denote the deterministic work element time (an integer or a fraction) of the i-th element.

J_k be the subset containing all tasks at the k-th station.

T_k be the total time needed for the job at the k-th station, where $T_k = \sum_{i \in J_k} t_i$.

c denote the cycle time desired (time units per unit of output).

$\mathcal{C} = \{c\}$ a script c represents the set of all realizable cycle times.

Since the objective is to minimize labor cost, which is proportional to the idle time in the assembly, we can write the total idle time D as,

$$\left.\begin{aligned} D &= \sum_{k=1}^{m} (c - T_k) = mc - \sum_{k=1}^{m} \sum_{i \in J_k} (t_i) \\ &= mc - \sum_{i=1}^{m} t_i \end{aligned}\right\} \quad (2.1)$$

In the Equation (2.1), due to the deterministic nature of the elemental times, $\sum_{i=1}^{n} t_i = K =$ a constant . So minimization of D involves only the minimization of the product, mc, subject to:

$$c - T_k \geq 0 \quad \text{for each } k = 1, 2, \cdots, m \qquad (2.2)$$

If k is given, c can be minimized and vice versa. But a more general measure of balance for a particular design (a feasible set of grouping of work elements obeying the precedence relations in a precedence diagram) of production line and cycle time is given by a discrete delay function:

$$d = \frac{D}{mc} \times 100 \qquad (2.3)$$

> A necessary but not sufficient condition for perfect balance (or zero balance delay) is that $mc - \sum_{i=1}^{n} t_i = 0$, where m (the number of work stations) is an integer (Kilbridge and Wester, 1961, p. 80).

Equation (2.3) gives a more general measure of balance since it gives the percentage underutilization of the resources on the average and this figure can be used to compare the effective utilization of different assembly lines at various locations.

The two standard formulations, thus, consist of minimizing (2.1) and minimizing (2.3), both subject to the constraint (2.2) over the set $\mathcal{C} = \{c\}$. The range of c is given by

$$t_{max} = c_L \leq c \leq c_H = \sum_i t_i \qquad (2.4)$$

The lower bound of c $(c_L = t_{max})$ follows from the definition of the cycle time. The upper bound of c $(c_H = \Sigma t_i)$ is based on practical considerations. There is no sense in allowing the cycle time to increase beyond the total work content time. To defy the lower bound by allowing $t_{max} > c$ would necessitate the use of two or more lines or two or more operators as parallel stations for the single task.

b. Variable elemental time line balancing model. Moodie (1964) and Ramsing and Downing (1970) used variable time data to balance the assembly lines. The fact that variability is a factor in assembly line balancing was brought out by Buffa (1961) in a study of pacing effects in production lines. Hicks and Young (1962) reported a study which showed that the elemental times are actually random variables approximated by the normal distribution. Walker (1959) extended this hypothesis further to claim that these variables (work element times) are distributed normally, mutually independent and their covariance: $\sigma_{ij} = 0$ for $i \neq j$ (i and j are the identification numbers for variable elements).

Assuming the normality and independence, the variance of the sum of the elements which make up a station on an assembly line is equal to the sum of the individual variances of each element, i. e.,

$$V(T_k) = \sum_{i \in J_k} V(t_i) \quad .$$

The normal curve theory can be used to determine the probability that the time to complete the work assigned to a work station will exceed the cycle time.

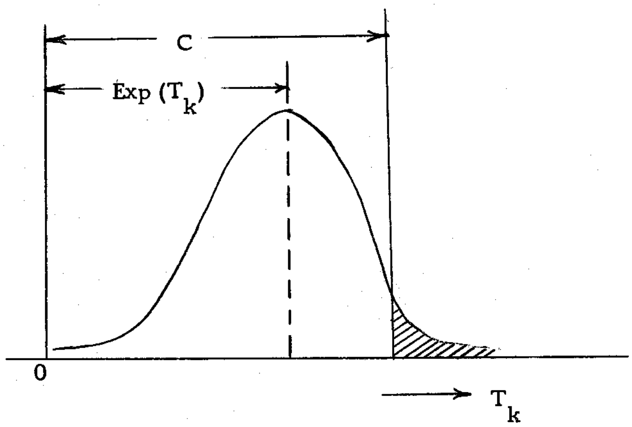


Figure 2-4. Station time with normal variation.

From Figure 2-4, it can be seen that during a certain portion of the time (the hatched area) a station time can exceed the cycle time. This causes certain line inefficiencies including work congestion which occurs when a station time exceeds the cycle time. Penalty for work congestion can be assigned by arriving at a cost rate for a particular probability of k stations to exceed the cycle time c . The way of arriving at a cost rate for this penalty is left entirely to the discretion of the management. When the times are assumed to be random variables, it has been shown by several authors (e.g., Buffa, 1961 and Freeman, M. C., 1964) that inventory between stations can indeed improve the output rate of the assembly line. This model suffers from the disadvantage of neglecting in-process inventory costs.

The balance delay criterion of deterministic times can be modified to suit the normally distributed random times as follows. In the deterministic case we have to minimize

$$d = \frac{(mc - \sum_{k=1}^{m} T_k)}{mc}$$

subject to

$$c - T_k \geq 0 .$$

Since T_k is a random variable, normally distributed with a mean $Exp(T_k)$, and a standard deviation $\sigma(T_k) = \sqrt{V(T_k)}$, we can define the individual station times with variability as,

$$T_k' = Exp(T_k) + Z\sqrt{V(T_k)}$$

where Z, the standard normal deviate, obtained from the statistical tables for a given value of probability of station times to exceed the cycle time c. If we allow the individual station times T_k to exceed the cycle time c, 5 percent of the time, the multiplier Z would be 1.645, while for 15 percent it is 1.035. Thus the modified objective function for the variable data is

$$\text{minimize} \quad d = [mc - \sum_{k=1}^{m} \{Exp(T_k) + Z\sqrt{V(T_k)}\}] \,|\, mc \qquad (2.5)$$

Here,

$Exp(T_k)$ is the mean value of the sum of element times that make up T_k, and

$V(T_k)$ is the variance of this sum of element times at k-th station.

If $(x_1, x_2, \cdots, x_i, \cdots, x_n)$ are the normally distributed work element times with means $(t_1, t_2, \cdots, t_i, \cdots, t_n)$ and variance $(v(t_1), v(t_2), \cdots, v(t_i), \cdots, v(t_n))$, then by the assumptions of normality and independence we have,

$$Exp(T_k) = Exp[\sum_{i \in J_k} x_i] = \sum_{i \in J_k} Exp(x_i) = \sum_{i \in J_k} t_i = \overline{T}_k$$

and

$$V(T_k) = \sum_{i \in J_k} V(t_i) .$$

If we let $\sigma(T_k)$ as the standard deviation of the k-th station time given by $\sqrt{V(T_k)}$ then substituting these values in the Equation (2.5) we will have:

$$\text{minimize } d = [mc - \sum_{k=1}^{m} \{\overline{T}_k + Z\sigma(T_k)\}]/mc \qquad (2.6)$$

By writing $T_k^1 = \overline{T}_k + Z\sigma(T_k)$, Equation (2.6) can be rewritten as

$$\text{minimize } d = [mc - \sum_{k=1}^{m} T_k^1]/mc \qquad (2.7)$$

Thus the line balancing solutions used for deterministic data can be converted to handle the variable data by redefining the T_k values. This general method holds good both for single and mixed-model situations with the two different interpretations of cycle time. In mixed-model lines the method of balancing a line can be either on the basis of a cycle time or a daily production time (Chapter V). Thus the total labor cost = $\{mc - \sum_k T_k^1\}$ + (penalty cost due to work congestion based on the value of Z).

Minimization of the total production cost per unit assembled. Due to the variability of work station times, the desired cycle time cannot be maintained throughout the production. The time between successive items coming off the end of the line is a random variable.

In fact, one can view this as the interdeparture time from a series queue. Unfortunately, a queueing theory provides little insight into the behaviour of this random variable (except for very special distributions)(Freeman, 1968, p. 231).

In process inventory in between the stations avoids the blocking with an added cost of inventory. In the determinisitc case the interdeparture time is identical to the maximum station time, which is equal to the cycle time. Minimization of the total cost per unit assembled subject to the precedure relations complicates the line balancing model due to the consideration of various costs. Mansoor and Ben-Tuvia (1966) gave a procedure to find the best cycle tune, for n station, perfectly balanced line with variable work station times but this optimum c, they seek minimizes only labor costs and hence is inadequate. Freeman (1968) outlined a solution procedure with an objective function of the form,

$$\begin{aligned}\text{Total cost/unit} &= \text{Labor cost/unit} + \text{Inventory cost/unit} \\ &\quad + \text{facility cost/unit} + \text{Penalty cost/unit} .\end{aligned}$$

Detailed description of this model appears in Chapter IV.

The standard ALB model formulated in this chapter would help to realize that the problem of balancing an assembly line is directly related to the problem of minimizing the production costs per unit assembled on a conveyor in any mass producing industry.

III. TRADITIONAL SINGLE-MODEL ALB-MINIMIZATION OF DIRECT LABOR COST-DETERMINISTIC WORK ELEMENTAL TIMES

> Line-balancing problems have received a great deal of attention, perhaps more than the prevalence of assembly lines warrants. Some techniques yield exact solutions for the given assumptions. Others are designed to yield approximate solutions based on practical considerations (Riggs, 1970, p. 320).

The traditional research on line balancing, however, had been focused on developing solution procedures to minimize the total direct cost associated with the total idle time along the line in the assembly of a single-model or product. Though some authors (e.g.,Arcus, 1966) had briefly mentioned the variability of the work element times and the handling of several models of the same general product on a single conveyor line, none of the traditional approaches gave a detailed description of a solution method.

A summary of both the analytical and heuristic procedures developed in the past and suitable for a single-model problem is made in this chapter with illustrated examples. This summary will give an insight into the various techniques developed earlier and aid in the creation of new and better procedures suitable for both single and mixed-model assembly lines.

Analytical Procedures

The mathematical interpretation of a practical ALB problem is to minimize a function measuring the idle time which is defined over a permutation of a group of work elements, the members of which are subject to technologically determined precedence relations on their permissible linear sequences.

Linear programming and dynamic programming are two major techniques applied to this problem. Shortest route criterion was also used by some authors to minimize the idle time for a given cycle time in an ALB problem.

The following analytical models are presented with examples and are arranged in a chronological order of their publications.

A. Linear Programming (L. P.) Model (Salveson, 1955).

The first published analytical statement of the line balancing problem was by Salveson who made use of deterministic time.

> We assume determinism in the production rate function (work standard) by using company experienced data on allowances for normal stochastic perturbation in production (Salveson, 1955, p. 18).

Our objective is to minimize total idle time $f(t)$ in all the stations on the line. Using the notation defined in Chapter II, we have

$$\min\ f(t) = \sum_{k=1}^{m} \{c - \sum_{i \epsilon J_k} t_i(k)\}$$

$$= \sum_{k=1}^{m} \{c - \sum_{i \epsilon J_k} x_{ik} t_i\} \qquad (3.1)$$

subject to the following conditions:

a. if any $p_{ij} = 1$ (where $P = [p_{ij}]$ is a precedence matrix of zeros and ones) and $i \epsilon J_k$, then $j \epsilon (J_k \text{ or } J_k^*)$ where k precedes k^* and the J_k are numbered in accordance with the same rules (left to right) for numbering the task i.

b. $\sum_{k=1}^{m} x_{ik} = 1$, $i = 1, 2, 3, \cdots, n$

$$x_{ik} = \begin{cases} 1 & \text{if task } i \text{ is assigned to the k-th station} \\ 0 & \text{if task } i \text{ is not assigned to the k-th station} \end{cases}$$

c. $\sum_{i \epsilon J_k} x_{ik} t_i \leq c$ for all $k = 1, 2, 3, \cdots, m$

However, minimizing cycle time for a given number of stations, k, implies selecting x_{ik} of the tasks i for the k stations so as to minimize the maximum total work time (cycle time) assigned to any station, ie., minimize

$$\max \left\{ T_k = \sum_{i=1}^{n} x_{ik} t_i , \quad k = 1, 2, \cdots, m \right\} \tag{3. 2}$$

subject to the restrictions (a) and (b) of Equation (3. 1).

1. <u>Linear programming formulation to minimize total idle time.</u> The following changes in the notation will allow us to formulate an L. P. problem from an ALB problem. Let $B = b_{ij}$ be a candidate matrix of zeros and ones, indicating

the cell conditions, 0 if an element i is not assigned to the j-th column (vector) and 1 if it is. Each column is then a combination of certain specified elements and forms a "candidate" for becoming a station. Each such combination will have a characteristic delay or idle time denoted by d_j:

$$d_j = c - \sum_i b_{ij} t_i , \quad j = 1, 2, \cdots, m .$$

The problem then becomes:

$$\text{minimize} \sum x_j d_j \tag{3.3}$$

subject to the following constraints.

a. $\sum_{j=1}^{m} x_j b_{ij} = 1; \; i = 1, 2, \cdots, n$

b. $0 \leq x_j \leq 1 \; ; \; j = 1, 2, \cdots, m$

c. $0 \leq \sum_i b_{ij} t_i \leq c \; ; \; j = 1, 2, \cdots, m$

The above restrictions define a convex subset of an m dimensional space, each point in which is specified by an m-tuple, $(x_1, x_2, \cdots, x_m)$. The extrema of that convex sub-subset are those m-tuples in which:

i) $x_j = \{ {0, \text{ or} \atop 1}$ for $j = 1, 2, 3, \cdots, m$

ii) for which the restrictions in Equation (3.3) hold

iii) for every i there is exactly one $x_j b_{ij} = 1$, all other $x_j b_{ij} = 0$ for that i.

A pre-enumerated matrix of feasible combinations is needed for the above L. P. model and this matrix can be extremely large and computationally unfeasible. For example in a 9 element problem we may have $\sum_{r=1}^{9} {}_9C_r$ combinations = 531, while the number of feasible combinations may be reduced due to precedence and cycle time constraints.

By considering x_j in the above L. P. model as a discrete variable a combinational approach is adopted with the following steps.

2. <u>Steps for the combinational algorithm.</u>

<u>Step 1.</u> Prepare a candidate matrix [B]. The candidate matrix will contain all the possible combinations of elements subject to the cycle time and precedence requirements.

<u>Step 2.</u> Select a feasible solution. From the candidate matrix select column $d_j (d_j = c - \sum_i b_{ij} t_i)$ with minimum idle time. Enter this column d_j into a solution matrix [S]. This should be continued until each row has one and only one entry. The solution matrix now contains an initial feasible

solution to the ALB problem (ie. $\Sigma d_j < c$).

Step 3. Determining the optimum: compute the total delay time (Σd_j) for the feasible solution and carry out the following checks.

a. If the $\Sigma d_j < c$, then it is the minimal station balance.

b. If the $\Sigma d_j \geqslant c$, then alter the feasible solution by removing certain candidates and entering another combination of candidates such that all elements are covered by the new combinations of candidates. Compute the difference in idle time between the rejected candidates and the recently entered candidates. Let

$$Z = \Sigma d_j \text{ out} - \Sigma d_j \text{ in}$$

If $Z \leq 0$, then the present solution is a better one,

If $Z > 0$, then the proposed solution is a better solution,

If $Z > 0$, proceed to Step 4,

Step 4. To change the solution:

a. Select the combinations with the maximum positive valued Z (or any arbitrarily selected tie).

b. Enter that combination of columns into the solution matrix [S] and delete from S all columns which have a non-zero entry in any row in which the selected column has a non-zero entry. This will yield a new solution matrix S.

c. Return to Step 3 and Step 4 in cycles until the criterion in Step 3(a) or 3(b) is satisfied, (until all possible combinations have been tried, or until one considers that the solution he has is satisfactory).

Example:

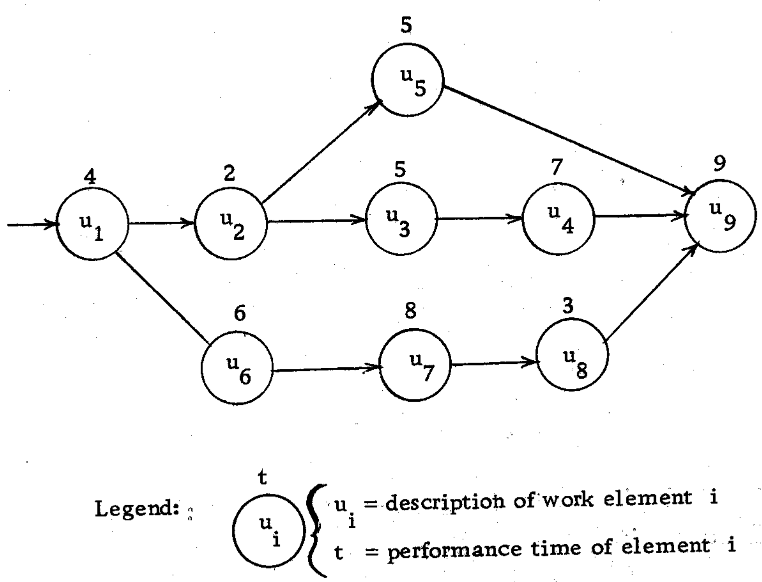


Figure 3-1. Precedence diagram of 9 work elements to illustrate Salveson's model.

Solution: Let the given cycle time = 10 minutes and we have to minimize the number of work stations subject to the cycle time constraint and the precedence relations shown in Figure 3-1. Note all

$$t_i \leq c = 10$$

$$\sum_{i=1}^{9} t_i = 48$$

$$\begin{aligned} k \text{ feasible} &= \{(u_i) \mid t_i > (\tfrac{c}{2})\} + \{(u_i) \mid t_i = (\tfrac{c}{2})\} \\ &= \{u_4, u_6, u_7, u_9\} + \{u_3, u_5\} \\ &= 4 + 2 = 6 \end{aligned}$$

and

$$\begin{aligned} m^* &= \text{minimum number of stations} \\ &= [\tfrac{48}{10}] = [4.8] = 5 \ . \end{aligned}$$

The application of the several steps involved in the combinatorial method is illustrated below to arrive at the minimal station balance.

Step 1. Generation of candidate matrix [B] as shown in Table 3-1.

Table 3-1. Candidate matrix generated from Figure 3-1.

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
d_j	6	8	5	3	5	4	2	7	2	4	0	3	3	0	0	5	0
i																	
1	1									1	1						
2		1								1		1	1			1	1
3			1									1		1			
4				1											1		
5					1								1	1			
6						1					1						
7							1										1
8								1							1	1	
9									1								

Step 2. From the candidate matrix [B] a solution matrix [S] is listed in Table 3-2 with minimum d_j values for candidates selected.

Table 3-2. Solution matrix obtained from Table 3-1.

j	11	14	15	17	9
d_j	0	0	0	0	2
i=1	1				
2				1	
3		1			
4			1		
5		1			
6	1				
7				1	
8			1		
9					1

In the solution matrix [S], we have $\Sigma d_j = 2 < c = 10$; hence we arrive at the optimal (minimal station) balance for the given value of cycle time. However, to illustrate the other steps in the method, we can start with another feasible solution in Table 3-3, where $\Sigma d_j = 22 > c$.

Table 3-3. Alternate feasible solution with $\Sigma d_j = 22$.

j	11	14	7	9	4	8	2
d_j	0	0	2	2	3	7	8
i=1	1						
2							1
3		1					
4					1		
5		1					
6	1						
7			1				
8						1	
9				1			

Step 3. Determining the optimum.

Step 3a. Does not apply here.

Step 3b. Alter the feasible solution by removing certain candidates and entering another combination of candidates as shown in Table 3-4.

Table 3-4. Details of entering and departing candidates to arrive at the alteration #1.

Departing			Entering		
i	j	d_j	i	j	d_j
2	2	8	2, 7	17	0
7	7	2			
Σd_j out=10			Σd_j in=0		

From Table 3-4, $Z = \Sigma d_j \text{ out} - \Sigma d_j \text{ in} = 10$.

Step 4. To change the solution.

Step 4a. Does not apply here.

Step 4b. Modify the alternate feasible solution with the alteration #1. The modified solution appears in Table 3-5.

Table 3-5. Summary of modified feasible solution with alteration #1.

j	11	14	9	4	8	17
i	1, 6	3, 5	9	4	8	2, 7
d_j	0	0	2	3	7	0

From Table 3-5, $\Sigma d_j = 12 > c$, implying we have to proceed further to obtain a minimal station balance.

Step 4c. By returning to Steps 3 and 4 we have to find alteration #2 as shown in Table 3-6.

Table 3-6. Details of entering and departing candidates to arrive at the alteration #2.

Departing			Entering		
i	j	d_j	i	j	d_j
4	4	3	4, 8	15	0
8	8	7			
Σd_j out=10			Σd_j in=0		

From Table 3-6, Z = 10 - 0 = 10. At this stage, the modified feasible solution will have 5 stations with candidates 9, 11, 14, 15, 17 and $\Sigma d_j = 2 < c$, which tallies with the earlier result. Thus it took two transformations to arrive at the optimal station balance.

B. Dynamic Programming Algoithm (Jackson, 1956).

An exhaustive and enumerative algorithm based on dynamic programming technique is given by Jackson. This algorithm, if carried to completion, finds an optimal solution. The algorithm enumerates feasible first-station assignments of elements, then for each first-station assignment, the feasible second-station assignments, and so forth, until all the feasible station assignments have been listed. Optimal station assignment is selected from this list. Jackson presents a convenient method of drawing the precedence diagram to suit his algorithm.

The following steps explain the diagramming method. It starts with Step 1 and ends after the completion of the first step in which all operations have appeared on the diagram.

Step 1: In column 1, on the left side of a page, list all operations which need not follow any operation.

Step n(a), $n \geq 2$: In column n, to the right of column (n-1), list all operations not already on the diagram, which need not follow any operation which is not already on the diagram.

Step n(b): Draw all arrows from operations in column (n-1) to operations in column n which must follow them. Repeat this procedure, replacing column (n-1) by column (n-2), (n-3), ···, 1, successively; except that no arrow is drawn from one operation to another if it is possible to follow arrows already drawn from the first operation to the second (Jackson, 1956, p. 263).

The enumeration method given by Jackson can be divided into two routines, namely, the MAIN ROUTINE and the SUBROUTINE as described below.

SUBROUTINE: This is a set of rules for enumerating certain possible assignments to the first station. It's use is further extended to enumerate several possible "next" assignments when a certain sequence of sets is considered for assignment to the first few stations.

Given a sequence $\{X(1), \cdots, X(n-1)$, where each X(i) is a set of operations, the

"collection of next assignments after $\{X(1), \cdots, X(n-1)\}$

is the collection of sets of operations obtained as follows:

Rule 1. Delete from the diagrammatic representation of the line balancing problem all operations in the sequence $\{X(1), \cdots, X(n-1)\}$ and all arrows emanating from these operations.

Rule 2. List all sets X of operations on the diagram, obtained by Rule 1 such that:

2(a). If a given operation is in X, so is every operation from which an arrow leads to the given operation.

2(b). The sum of the performance times for the operations in X is not greater than the upper limit on the cycle time.

2(c). No operation can be added to X without violating 2(a) or 2(b).

Rule 3. (can be omitted, but often at the cost of a substantially enlarged enumerations.) Successively cross off the list of Rule 2 sets X for which there is another set Y on the list (still not crossed off), such that:

3(a). There is just one operation x in X which is not also in Y.

3(b). There is some operation y in Y, which is not in X, which has performance time at least as great as that of x, and such that arrows can be followed from y to any operation z for which there is an arrow from x to z (Jackson, 1956, p. 265-266).

The subroutine ends when there are no more sets that can be crossed off by Rule 3.

MAIN ROUTINE: The following steps illustrate this procedure.

Step 1(a). Construct the collection of next assignments after $\{\phi\}$, where ϕ is the empty set. Here Rules 2 and 3 of the SUBROUTINE are used to arrive at all the possible assignments to station 1.

Step 1(b). Write List 1, the list of sequences $\{X(1)\}$ of a single set of operations with $X(1)$ in the collection obtained in Step 1(a).

Step n(a), $n > 2$. For each sequence $\{X(1), \cdots, X(n-1)\}$ of sets of operation on list $(n-1)$, construct the collection of next assignments after $\{X(1), \cdots, X(n-1)\}$.

Step n(b). Write list n(b), the list of sequences $\{X(1), \cdots, X(n-1), X(n)\}$, with $\{X(1), \cdots, X(n-1)\}$ on list $(n-1)$ and $X(n)$ in the collection of next assignments after $\{X(1), \cdots, X(n-1)\}$.

Step n(c). Obtain list n from list n(b), by successively crossing off sequences $\{Y(1), \cdots, Y(n)\}$ on the list (still not crossed off); such that each operation included in any $X(i)$ is included in some $Y(j)$ (there may be operations included in some $Y(i)$ which are not in any $X(i)$) (Jackson, 1956, p. 266-267).

Step n is completed and thus the main routine ends when no more sequences can be crossed off by step n(c).

Mathematical Justification: Let

A = finite set of all required operations.

m = number of stations.

$t(a)$ = performance time for element a.

T = upper limit on the cycle time

apb imply a precedes b.

If we have a set A, partially ordered by relation p with a positive valued function t on A, and number $T \geq \max\{t(a)\}$ over $a \in A$; the ALB problem can be expressed as

minimize m, over the partitions $A(1), A(2), \cdots, A(m)$ of A

such that,

1. If apb , $a \epsilon A(k)$ and $b \epsilon A(\ell)$, then $k \leq \ell$

2. $\sum_{a \epsilon A(\ell)} t(a) \leq T$, for $\ell = 1, 2, \cdots, m$

The set of partitions or stations satisfying (1) and (2) depends upon A, p, and the function t/T . Thus an ALB problem can be represented by $(A, p, t/T)$. The corresponding minimum value of m will be denoted by

$$m(A, p, t/T)$$

LEMMA 1. A necessary and sufficient conditon that there be a partition $A(1), A(2), \cdots, A(m)$ of A , satisfying (1) and (2) and with $A(1) = B$, where B is a subet of A , is that

3. If apb and $b \epsilon B$ then $a \epsilon B$

4. $\sum_{b \epsilon B} t(b) \leq T$

(Jackson, 1956, p. 271).

If B is any such subset, then the least value of m for any such partition will be equal to $1 + m(A - B, p, t/T)$. Thus it would be possible to choose a subset B , satisfying the conditions in (3) and (4) so as to minimize the function $m(A-B, p, t/T)$. This method resembles a Dynamic Programming approach.

To minimize the function $m(A-B, p, t/T)$ the SUBROUTINE discussed earlier was used and its usage was mathematically justified by the following theorem.

THEOREM. Consider two ALB problems (A, p, t/T) and (B, ϕ, u/U). Suppose there is a single valued mapping k: A $\rightarrow$ B such that for a and a' in A, and $b \epsilon B$ we have,

$$apa' \Longrightarrow k(a)qk(a') \text{ and}$$

$$\sum_{k(a)=b} \frac{t(a)}{T} \leq \frac{u(b)}{U}$$

Then

$$m(A, p, t/T) \leq m(B, q, u/U)$$

Proof: If B(1), B(2), $\cdots$, B(m) is a partition of B which satisfies (1) and (2) above (with A, p, t, and T replaced by B, q, u, and U respectively) then a partition of A satisfying (1) and (2) is obtained by assigning to $A(\ell)$ precisely those $a \epsilon A$ for which $k(a) \epsilon B(\ell)$ (Jackson, 1956, p. 271).

Example:

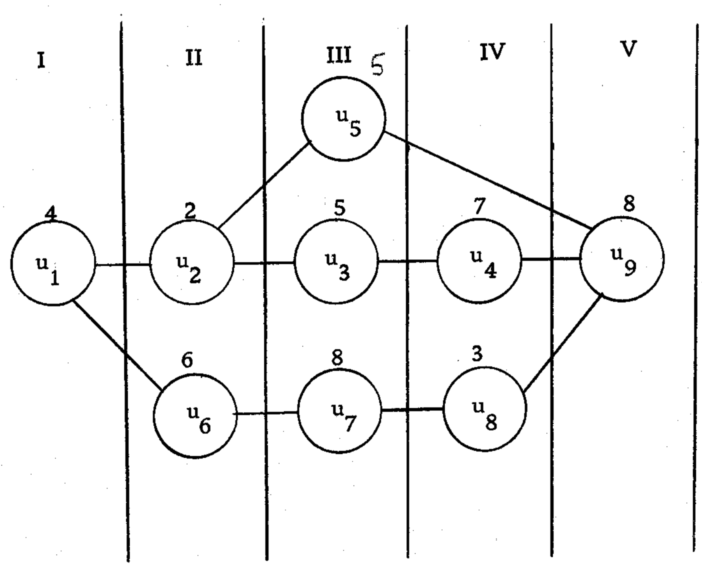


Figure 3-2. Precedence diagram of 9 work elements to illustrate Jackson's enumeration method.

Solution:

Figure 3-2 is constructed using Jackson's method of constructing precedence diagram. The example is solved using the steps given earlier. The sets in curly brackets shown below indicate that they are ordered sets and are obtained by Rule 2 of the SUBROUTINE. The sets in square brackets are those which form two set sequences which are eliminated by Step 2(c).

<u>Step 1 a, b:</u> $u_1 u_2, u_1 u_6$

<u>Step 2:</u> $\{u_1 u_2\}$: $u_3 u_5, u_6$

$\{u_1 u_6\}$: $u_2 u_5, u_2 u_7, u_2 u_3$

<u>Step 3:</u> $\{u_1 u_2, u_3 u_5\}$: $u_4 [u_6]$

$\{u_1 u_2, u_6\}$: $[u_3 u_5], [u_7]$

$\{u_1 u_6, u_2 u_7\}$: $u_3 u_5, u_3 u_8, u_5 u_8$

$\{u_1 u_6, u_2 u_5\}$: $[u_3], [u_7]$

<u>Step 4:</u> $\{u_1 u_2, u_3 u_5, u_4\}$: $[u_6], [u_7]$

$\{u_1 u_6, u_2 u_7, u_3 u_5\}$: $u_4 u_8$

$\{u_1 u_6, u_2 u_7, u_3 u_8\}$: $[u_5], [u_4]$

$\{u_1 u_6, u_2 u_7, u_3 u_5\}$: $u_4 u_8$

$$\{u_1u_6,\ u_2u_7,\ u_3u_8\}:\ [u_5],\ [u_4]$$

$$\{u_1u_6,\ u_2u_7,\ u_5u_8\}:\ [u_3],\ [u_4]$$

Step 5: This step is shown in Table 3-7.

Table 3-7. Summary of station assignments at the end of Jackson's enumeration method.

u_i	u_1u_6	u_2u_7	u_3u_5	u_4u_8	u_9
$\sum_{i \in J_k} ti$	10	10	10	10	8
d_k	0	0	0	0	2
k	1	2	3	4	5

$\Sigma d_i = 2 < 10$ implying an optimal solution for the given cycle time of $c = 10$ minutes.

C. Integer Linear Programming Model (Bowman, 1960 and White, 1961).

Two different linear-programming formulations (known as first linear program and second linear program) were developed to the ALB problem. The first of the two formulations by Bowman (1960) was reformed by White (1961) who adopted Kronecker delta functions.

The two different formulations of Bowman and the reformation by White are illustrated below using the precedence diagram, in Figure 3-3.

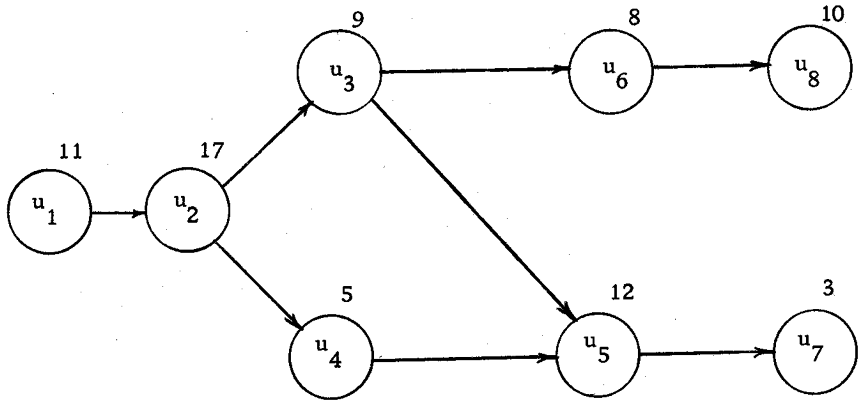


Figure 3-3. Precedence diagram of 8 work elements to illustrate Bowman's L. P. Formulations (source Bowman, 1960, p. 385).

Referring to Figure 3-3, let

t_i = time required to perform job $i = 1, 2, \cdots, 8$

c = cycle time = 20 time units

A_i = number of time units devoted to operation u_i of station A

For convenience, Bowman considers 7 work stations as an upper limit to this problem. The seven possible stations are identified by the letters A, B, C, D, E, F and G.

First Linear Program

a. To assure that none of the stations are overloaded, the following set of constraints (3.4) is developed, i. e.

$$\begin{aligned} A_1 + A_2 + A_3 + \cdots + A_8 &\leq 20 \\ B_1 + B_2 + B_3 + \cdots + B_8 &\leq 20 \\ \vdots \qquad\qquad &\quad \vdots \\ G_1 + G_2 + G_3 + \cdots + G_8 &\leq 20 \end{aligned} \tag{3.4}$$

b. To make sure that each operation is performed, the set of constraints (3.5) is developed.

$$\begin{aligned} A_1 + B_1 + C_1 + \cdots + G_1 &= 11 \\ A_2 + B_2 + C_2 + \cdots + G_2 &= 17 \\ \vdots \qquad\qquad &\quad \vdots \\ A_8 + B_8 + C_8 + \cdots + G_8 &= 10 \end{aligned} \tag{3.5}$$

c. The following set of constraints (3.6) includes integer variables of the form A_iI which must take the values zero or one.

$$\begin{aligned} &\frac{1}{11} A_1 + A_1I = 1; \frac{1}{11} B_1 + B_1I = 1; \cdots \frac{1}{11} G_1 + G_1I = 1; \\ &\frac{1}{17} A_2 + A_2I = 1; \frac{1}{17} B_2 + B_2I = 1; \cdots \frac{1}{17} G_2 + G_2I = 1; \\ &\quad\vdots \qquad\qquad\qquad \vdots \qquad\qquad\qquad \vdots \\ &\frac{1}{10} A_8 + A_8I = 1; \frac{1}{10} B_2 + B_2I = 1; \cdots \frac{1}{10} G_8 + G_8I = 1\,. \end{aligned} \tag{3.6}$$

The basic purpose of the constraint set (3.6) is to assure that the operations are not split between stations, in other words, that they are assigned to only one station. For instance, the constraint on A_1 insists that it takes the value of either eleven or zero, considering that $A_1 I$ must take the value of either zero or one.

d. The following set of constraints (3.7) assures proper ordering.

$$\frac{1}{17}A_2 \leq \frac{1}{11}A_1;\ \frac{1}{17}B_2 \leq \frac{1}{11}A_1 + \frac{1}{11}B_1;$$

$$\frac{1}{17}C_2 \leq \frac{1}{11}A_1 + \frac{1}{11}B_1 + \frac{1}{11}C_1;\ \cdots; \tag{3.7}$$

$$\frac{1}{10}G_8 \leq \frac{1}{8}A_6 + \frac{1}{8}B_6 + \frac{1}{8}C_6 + \frac{1}{8}D_6 + \frac{1}{8}E_6 + \frac{1}{8}F_6 + \frac{1}{8}G_6;\ \cdots;$$

$$\frac{1}{3}G_7 \leq \frac{1}{12}A_5 + \frac{1}{12}B_5 + \frac{1}{12}C_5 + \frac{1}{12}D_5 + \frac{1}{12}E_5 + \frac{1}{12}F_5 + \frac{1}{12}G_5$$

The first constraint for instance assures operation 1 precedes operation 2 on the line.

The objective function of this linear program is

$$\text{minimize } Z = 1(E_7 + E_8) + 14(F_7 + F_8) + 196(G_7 + G_8)$$

The purpose of this objective function is to make later stations exceedingly costly, pushing the operations as far forward as is physically possible. Stations A through D must certainly be used and need assume no cost. Only operations with no succeeding operations in an ordering need positive costs in the objective function, i. e., they may be the last on the line. The nature of cost explosion, 1, 14, 196 is to make one unit of later assignment more costly than the sum of all preceding station assignments [14 > 3 + 10, 196 > 3(14) + 10(14) + 10 + 3] (Bowman, 1960, p. 387).

Reformation of the First Linear Program Formulation by White

Consider the fact that each of the variables A_i is discrete; for instance, A_1 is either eleven or zero, depending upon whether we do operation u_1, at station A or not. Using Kronecker deltas the 3rd set of constraints of Bowman's First Linear Program can be eliminated along with his "special integer variables".

Let

$$\delta_{Ai} = 1 \text{ if we do operation } i \text{ at station } A \text{ and}$$

$$\delta_{Ai} = 0 \text{ otherwise.}$$

Then $11\,\delta_{A1}$ is the number of time units devoted to operation 1 at station A. Using this notation Bowman's first set of constraints becomes

$$\begin{aligned} 11\,\delta_{A1} + 17\,\delta_{A2} + 9\,\delta_{A3} + 5\,\delta_{A4} + \cdots + 10\,\delta_{A8} &\leq 20 \\ 11\,\delta_{B1} + 17\,\delta_{B2} + 9\,\delta_{B3} + 5\,\delta_{B4} + \cdots + 10\,\delta_{B8} &\leq 20 \\ \vdots \qquad\qquad \vdots \qquad\qquad & \vdots \\ 11\,\delta_{G1} + 17\,\delta_{G2} + 9\,\delta_{G3} + 5\,\delta_{G4} + \cdots + 10\,\delta_{G8} &\leq 20 \end{aligned} \tag{3.8}$$

The second set of constraints can be modified by dividing out the common factor of the time required to do each operation.

We get

$$\left.\begin{array}{l} \delta_{A1} + \delta_{B1} + \delta_{C1} + \cdots + \delta_{G1} = 1 \\ \delta_{A2} + \delta_{B2} + \delta_{C2} + \cdots + \delta_{G2} = 1 \\ \quad\vdots \qquad\qquad\qquad\qquad \vdots \\ \delta_{A8} + \delta_{B8} + \delta_{C8} + \cdots + \delta_{G8} = 1 \end{array}\right\} \qquad (3.9)$$

This set of constraints (3.9) not only assures that each operation is performed but also guarantees that the operations are not split between stations. Further, it assumes that each variable must be either zero or one in any integer solution, thus eliminating Bowman's 3rd set of constraints. Bowman's last set of constraints remains same as given earlier. The objective function is refined as to minimize,

$$Z = 1(3\delta_{E7} + 10\delta_{E8}) + 14(3\delta_{F7} + 10\delta_{F8}) + 196(3\delta_{G7} + 10\delta_{G8})$$

where the cost coefficients 1, 14, 196 imply the same logic stated earlier. The advantage of White's approach is that the number of constraints and variables to be handled will be reduced.

> For example, this approach reduces Bowman's problem from 135 constraint equations (or inequalities) with 112 variables, not counting slacks, to a problem with 71 constraint equations (or inequalities) with 56 variables, not counting slacks and also we have a zero-one integer programming problem which can be solved using Gomory (1958)'s "cutting plane approach" (White, 1961, p. 276).

Bowman's Second Linear Programming Formulation

Referring to the precedence diagram of Figure 3-3, one of the feasible sequences is given by

$$u_1 \rightarrow u_2 \rightarrow u_3 \rightarrow u_4 \rightarrow u_5 \rightarrow u_6 \rightarrow u_7 \rightarrow u_8$$

Let x_i be the time at which the element u_i will be started.

a. To preserve ordering the constraint set (3.10) is developed.

$$x_2 \geq x_1 + t_1, \; x_3 \geq x_2 + t_2, \; \cdots, \; x_8 \geq x_7 + t_7 \qquad (3.10)$$

where $t_1, t_2, t_3, \cdots, t_7$ are the execution times of each job in the sequence given.

b. To avoid interference, i. e., one work element must be completed before the next one starts, constraint set (3.11) is formed.

$$\left.\begin{array}{l}
(x_1 - x_2) + w_\alpha [\Sigma t_i + t_2] \geq t_2 \\
(x_2 - x_1) + (1 - w_\alpha [\Sigma t_i + t_1] \geq t_1 \\
(x_2 - x_3) + w_\beta [\Sigma t_i + t_3] \geq t_3 \\
(x_3 - x_2) + (1 - w_\beta)[\Sigma t_i + t_2] \geq t_2 \\
\quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
(x_7 - x_8) + w_\gamma [\Sigma t_i + t_8] \geq t_8 \\
(x_8 - x_7) + (1 - w_\gamma [\Sigma t_i + t_7] \geq t_7 \quad \text{and}
\end{array}\right\} \qquad (3.11)$$

$w_\alpha \geq 1, w_\beta \geq 1, \cdots, w_\gamma \geq 1$

The constraint set (3.11) implied that the variables w_α, w_β, $\ldots$, w_γ will take integer values of 0 or 1.

c. To assure that no station is overloaded and that the work elements be wholly assigned to one station, the constraint set (3.12) is written:

$$
\begin{aligned}
x_1 + t_1 &\leq cV_\alpha + c;\ x_1 \geq cV_\alpha \\
x_2 + t_2 &\leq cV_\beta + c;\ x_2 \geq cV_\beta \\
&\vdots \\
x_8 + t_8 &\leq cV_\gamma + c;\ x_8 \geq cV_\gamma
\end{aligned}
\tag{3.12}
$$

V_α, V_β, $\ldots$, V_γ will take integer values from zero to m-1 where m is the number of stations to which jobs will be assigned and c is the cycle time.

Constraints (3.12) insure that each job will be assigned to stations according to times as shown in Table 3-8.

Table 3-8. Time allocations for assembly line stations.

Station	A	B	C	D	E
Time	0-c	c-2c	2c-3c	3c-4c	4c-5c

For instance, when $V_\alpha = 0$, the work element u_1 will be performed within the first station.

d. The following constraint set (3.13) assures that X_T, the total completion time for the last work element in the sequence will be at least as large as the starting time for the last jobs in the sequence plus their execution times.

$$\begin{aligned} X_T &\geq x_8 + t_8 \\ X_T &\geq x_7 + t_7 \end{aligned} \tag{3.13}$$

e. The objective function now becomes

$$\text{minimize } Z = X_T$$

This second linear program for the same problem uses 33 constraint equations (inequalities) with 8 variables plus 15 special integer variables, plus the variable X_T which equals 24 variables not counting slacks. This is a considerable improvement over the first linear program which requires 135 inequalities and 112 variables. Due to the excessive amount of computation involved, no example problem has been worked out here for illustration.

D. Dynamic Programming Model (Held, et al., 1963).

A generalized dynamic programming formulation of the ALB problem was made by Held et al. (1935). This model is formulated with the objective of minimizing the number of work stations for a given cycle time. The problem of minimizing idle time as per

Equation (2.2) is equivalent to minimizing the number of work stations for a given c.

An ALB problem is characterized by the set of jobs $\{u_1, u_2, \cdots, u_n\}$, to be assigned, their execution times, $\{t_1, t_2, t_3, \cdots, t_n\}$, the cycle time c, and partial ordering p expressing the precedence relations among the jobs.

A subset of tasks $S = \{u_{i_1}, u_{i_2}, u_{i_3}, \cdots, u_{i_{n(S)}}\}$ is feasible if $u_j \epsilon S$ and $u_i p u_j$ imply that $u_i \epsilon S$. Similarly a sequence $\sigma = (u_{i_1}, u_{i_2}, u_{i_3}, \cdots, u_{i_n})$ is feasible if for $1 \leq q \leq n(\sigma)$, $\{u_{i_1}, u_{i_2}, \cdots, u_{i_q}\}$ is a feasible set. Associated with each feasible sequence σ there is a particular assignment of tasks to work stations known as induced assignments, which satisfies the cycle time constraint. The problem is stated as

> .... minimizing the number of work stations to accomplish the feasible sequences, by assigning as many jobs as possible to the first work station, as many as possible from the beginning of the remaining sub-sequence to the second sub-station and so on, subject to $c - T_k \geq 0$ (Held et al., 1964, p. 444).

However, the Jackson's (1956) algorithm and the dynamic programming model of Held et al. are principally the same except for the generalization made by Held et al. The same example given under Jackson's dynamic programming would suffice to explain this model.

E. Assignment Model (Klein, 1963)

The ALB problem with a fixed value of cycle time c can be shown equivalent to an assignment type problem using the same terminology described earlier, which we shall repeat for easy reference. We have

$$\text{minimize} \quad D = \sum_{k=1}^{m} (c - T_k)$$

subject to the constraints

$$c - T_k \geq 0 \tag{3.14}$$

and

$$t_{max} \leq c \leq \sum_{i=1}^{n} t_i$$

Reformulate the problem as below:

Define an $n \times n$ matrix with entries a_{ij}, where for $i \leq j$

$$a_{ij} = c - \sum_{k=i}^{k=j} t_k \quad \text{if } c - \sum_{k=i}^{k=j} t_k \geq 0$$

$$= M \text{ (abritrarily large) if } c - \sum_{k=i}^{k=j} t_k < 0$$

and for $i > j$

$$a_{i,i-1} = 0 \qquad (i = 2, 3, \cdots, n)$$

$$= M \qquad \text{otherwise} \tag{3.15}$$

Now the problem is to minimize the linear form

$$\left.\begin{aligned}
&S(x_{11}, \cdots, x_{nn}) = \sum_{j=1}^{n} \sum_{i=1}^{n} x_{ij} a_{ij} \\
&\text{Subject to the constraints} \\
&x_{ij} = 0, 1 \qquad (i, j = 1, 2, \cdots, n) \\
&\sum_{i=1}^{n} x_{ij} = 1 \qquad (j = 1, 2, 3, \cdots, n) \\
&\sum_{j=1}^{n} x_{ij} = 1 \qquad (i = 1, 2, 3, \cdots, n)
\end{aligned}\right\} \tag{3.16}$$

This reverts to an assignment problem, which is another formulation of the linear programming problem where variables have to take integer value forms. It is therefore related to the formulation proposed by Bowman.

The elements of the cost matrix can be interpreted as follows. If

$$i \leq j \quad \text{and} \quad c - \sum_{k=i}^{k=j} t_k \geq 0$$

then a_{ij} is the idle associated with a station to which successive operations i through j have been assigned. If

$$c - \sum_{k=i}^{k=j} t_k < 0 ,$$

the station time associated with the assignment exceeds the cycle time; since such an assignment is infeasible, a_{ij} is set equal to M to force the associated variable

x_{ij} to be equal to zero. When $i > j$, the a_{ij}'s have no physical interpretation. If a variable $x_{i, i-1}$ (associated with element $a_{i, i-1} = 0$) is equal to one, it is interpreted as a dummy variable and ignored. The remaining elements (set equal to M) are included to "fill out" the matrix; the associated variables will always be zero in a solution to the problem (Klein, 1963, p. 278).

The steps for the algorithm proposed by Klein are as follows:

<u>Step 1</u>. Generate all possible orderings for the given work elements subject to the precedence constraints.

<u>Step 2</u>. For a range of possible cycle times, within the limitations imposed from below the largest work element (t_{max}), generate the $n \times n$ matrix of values a_{ij} according to the constraint set (3. 15).

<u>Step 3</u>. Find the optimal assignment by the regular assignment method or by inspection of the a_{ij} matrix.

The key to the algorithm is the Step 1, i. e., to generate feasible sequences from a precedence diagram. Klein remarked that apparently there is no formula or prescription on hand to determine the number of feasible sequences for a given problem. However, the research by Okamura and Yamashina (1969) gives a promising answer to the question of generating feasible linear sequences.

Example:

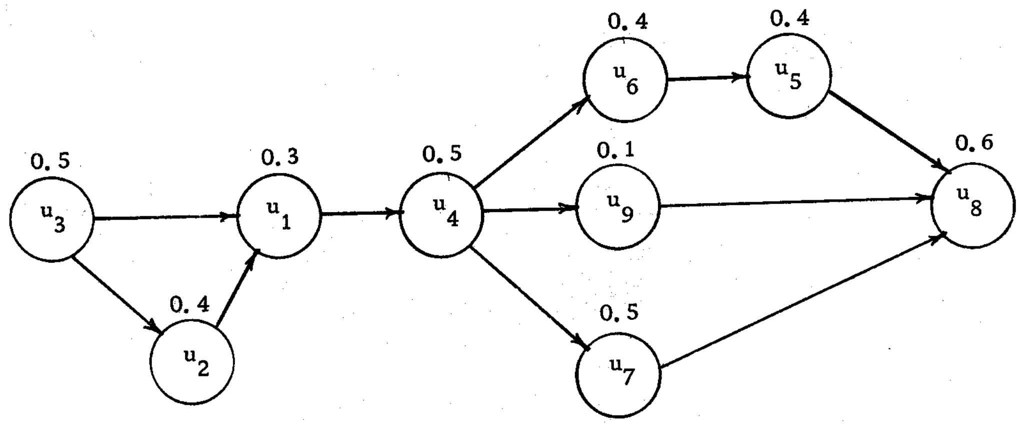


Figure 3-4. Precedence diagram to illustrate Klein's assignment model.

Solution:

Step 1. Since it is not possible to illustrate for all feasible sequences, let us consider one feasible sequence, ($u_3 \rightarrow u_2 \rightarrow u_1 \rightarrow u_4 \rightarrow u_9 \rightarrow u_6 \rightarrow u_5 \rightarrow u_7 \rightarrow u_8$) in Figure 3-4.

Step 2. Generate the a_{ij} matrix as shown in Table 3-9 subject to the constraint (3.15) for a given cycle time, c = 0.9 time units. In the Table 3-9, note

$$a_{11} = a_{u_3, u_3} = c - t_3 = 0.9 - 0.5 = 0.4$$

$a_{23} = a_{u_2, u_1}$ which implies the idle time associated with a station if the operations u_2 through u_1 in the specified feasible ordering are assigned to that station =

Table 3-9. nxn matrix showing a_{ij} entries for a specified feasible sequence u_3-u_2-u_1-u_4-u_9-u_6-u_5-u_7-u_8 and a cycle time $c = 0.9$ time units.

$j \rightarrow$		1	2	3	4	5	6	7	8	9
$i \downarrow$	u_i $i=j$	u_3	u_2	u_1	u_4	u_9	u_6	u_5	u_7	u_8
1	u_3	a_{11} 0.3	a_{12} 0	a_{13} M	a_{14} M	a_{15} M	a_{16} M	a_{17} M	a_{18} M	a_{19} M
2	u_2	a_{21} M	a_{22} 0.5	a_{23} 0.2	a_{24} M	a_{25} M	a_{26} M	a_{27} M	a_{28} M	a_{29} M
3	u_1	a_{31} M	a_{32} M	a_{33} 0.4	a_{34} 0.1	a_{35} 0	a_{36} M	a_{37} M	a_{38} M	a_{39} M
4	u_4	a_{41} M	a_{42} M	a_{43} M	a_{44} 0.4	a_{45} 0.3	a_{46} 0	a_{47} M	a_{48} M	a_{49} M
5	u_9	a_{51} M	a_{52} M	a_{53} M	a_{54} M	a_{55} 0.8	a_{56} 0.4	a_{57} 0	a_{58} M	a_{59} M
6	u_6	a_{61} M	a_{62} M	a_{63} M	a_{64} M	a_{65} M	a_{66} 0.5	a_{67} 0.1	a_{68} M	a_{69} M
7	u_5	a_{71} M	a_{72} M	a_{73} M	a_{74} M	a_{75} M	a_{76} M	a_{77} 0.5	a_{78} 0	a_{79} M
8	u_7	a_{81} M	a_{82} M	a_{83} M	a_{84} M	a_{85} M	a_{86} M	a_{87} M	a_{88} 0.4	a_{89} M
9	u_8	a_{91} M	a_{92} M	a_{93} M	a_{94} M	a_{95} M	a_{96} M	a_{97} M	a_{98} M	a_{99} 0.3

$c - (t_2 + t_1) = 0.9 - (0.4 + 0.3) = 0.2.$ Similarly

$a_{34} = a_{u_1, u_4} = c - (t_1 + t_4) = 0.9 - (0.3 + 0.5) = 0.1$

$a_{35} = a_{u_1, u_9} = c - (t_1 + t_4 + t_9) = 0.9 - (0.3 + 0.5 + 0.1) = 0$

$a_{36} = a_{u_1, u_6} = c - (t_1 + t_4 + t_9 + t_6)$

$= 0.9 - (0.3 + 0.5 + 0.1 + 0.4) = -0.4$, hence a_{36} is assigned a big value M. Continue this for other entries in the matrix (a_{ij}). However, for all i, $(i = 1, 2, \cdots, n)$, the cells $a_{i,i-1}$ are assigned a value equal to M instead of zero, as required in the constraint set (3.15). This is to avoid confusion in selecting a minimum feasible cell from the matrix.

Step 3. To get the optimal solution (minimum number of stations for the given value of $c = 0.9$), select the minimum element in row 1, which is $a_{12} = 0$, to imply operations u_3 through u_2 are assigned to the station 1. Cancel these two elements u_3 and u_2 from the sequence specified. Also cancel the corresponding rows and columns in the matrix, i.e., rows 1 and 2 and columns 1 and 2. Then we will have the remaining sequence $(u_1 \rightarrow u_4 \rightarrow u_9 \rightarrow u_6 \rightarrow u_5 \rightarrow u_7 \rightarrow u_8)$. Here u_1 corresponds to row 3 in the matrix. So the minimum element in the row 3 of the reduced matrix will be

$a_{35} = 0$, indicating an assignment of operations u_1 through u_9 (i. e., u_1, u_4, u_8) to the station 2. Now cancelling rows 3, 4, 5 and columns 3, 4, 5 we have another reduced matrix, with the remaining feasible sequence ($u_6 \rightarrow u_5 \rightarrow u_7 \rightarrow u_8$). The minimum element in the row 6 is $a_{67} = 0.1$, thus combining u_6 and u_5 as the station 3. By reducing the matrix further and doing the similar operations we find that $a_{88} = 0.4$ and $a_{99} = 0.3$ thus implying station 4 consists of element u_7 only, while u_8 alone fills the station 5. Table 3-10 summarizes the grouping of the tasks into stations.

Table 3-10. Summary of the assignment of tasks to stations by Klein's assignment model.

Work Station	Station Time	Station Idle Time	Tasks
1	0.90	0.0	u_3, u_2
2	0.90	0.0	u_1, u_4, u_9
3	0.80	0.1	u_6, u_5
4	0.50	0.4	u_7
5	0.60	0.3	u_8

An alternate method of arriving at a minimal station balance from a specified feasible sequence can be described in the following steps.

Let the order under consideration be $(1, 2, 3, \cdots, n)$, with a fixed value of c, to minimize the number of stations.

Step 1. Assign the first S operations to the first station where S satisfies the relation,

$$\sum_{i=1}^{i=S} t_i \leq c < \sum_{i=1}^{i=S+1} t_i$$

Step 2. Continue (a) with successive groups of operations until all have been assigned. The number of stations will be minimum at the end of step 2.

Thus if we can enumerate all feasible sequences for the given problem, we can find the minimal station balance for each sequence and then select that station balance which has the least idle time.

F. Shortest Route Model (Gutjahr and Nemhauser, 1964)

Klein (1963) indicated how a shortest route problem can be formulated as an ALB problem. However, a more thorough presentation was made by Gutjahr and Nemhauser. The shortest route problem is a special case of general network flow theory.

The steps for the shortest route algorithm are as follows:

<u>Step 1.</u> Generate all feasible ordered sets called states. The state S_i corresponds to node i, $i = 0, 1, 2, \cdots, r$.

> A state is a collection of work elements that can be processed without prior completion of any other work elements and in any order that satisfies the procedure relations (Gutjahr and Nemhauser, 1964, p. 309).

Note: S_o = empty set and S_r = a set containing all the elements of the precedence diagram. Each state number S_i is assigned a number equal to the sum of the processing times of the individual elements in the state, i. e.,

$$t(S_0) = 0$$

and

$$t(S_i) = \sum_{x \in S_i} t(\overset{x}{\cancel{s}}) \; ; \quad i = 1, 2, \cdots, r$$

x represents the work elements contained in the state S_i. Here the generation of states is independent of the cycle time.

<u>Step 2.</u> Given a cycle time c (one value in the whole range to be considered), construct a network using the states generated in step 1 as nodes, with state S_i corresponding to the node i. In the network there is a directed arc (ij) from nodes i to j and only if $S_i \subset S_j$ and

$t(S_j - S_i) \leq c$. Each directed arc (ij) is assigned a distance $c - \{t(S_j) - t(S_i)\}$, which represents the total delay time for the corresponding assignment. The network construction starts with the state 0, and generating all arcs from it. There is an arc from 0 to i if $t(S_i) \leq c$. Thus the first set of arcs from it spans all nodes from which the state time is less than or equal to the cycle time. The nodes reached with this step are called first nodes.

Step 3. From every first state or node construct all arcs ij. There is an arc from node i to node j, if $S_i \subset S_j$ and $t(S_j) - t(S_i) \leq c$.

Step 4. Repeat Step 3 until the last node r is reached for the first time. Thus the construction of arcs is completed. The minimum number of arcs required to span the nodes 0 to r corresponds to the minimum number of work stations to balance the line.

Step 5. To identify the states on the shortest route, determine the states which are spanned by the arcs (0, i), (i, j), ···, (r-2, r-1), (r-1, r).

In brief the algorithm consists of

a. generating all feasible states

b. constructing the network N with the states generated in (a) as nodes.

c. finding shortest route through the network based on an algorithm given by Dantzig (1960).

Example: Consider the same example used by Bowman (1960), Figure 3-3.

Solution:

Table 3-11 summarizes all the states generated from Figure 3-3, while Table 3-12 shows the calculation of shortest route for a given value of c = 20. Figure 3-5 is drawn from Tables 3-11 and 3-12. Referring to Figure 3-5 and starting at node 0 (state number 0), node 1 can be reached with one arc. Then from node 1, node 2 can be reached with 2 arcs from node 0 . This procedure is carried out until node 15 (the destination node consisting of all work elements) is reached for the first time. It is reached for the first time from node 10 with a path containing 5 arcs. Thus the elements in $S_{15} - S_{10}$ (4, 5, 7) constitute an optimal 5th station assignment. Similarly since node 10 is reached from node 3, the elements in $S_{10} - S_3$ (6, 8) constitute the 4th station optimal assignment. The other assignments are obtained in this way and are given in Table 3-13.

Table 3-11. Generation of states for the precedence diagram in Figure 3-3.

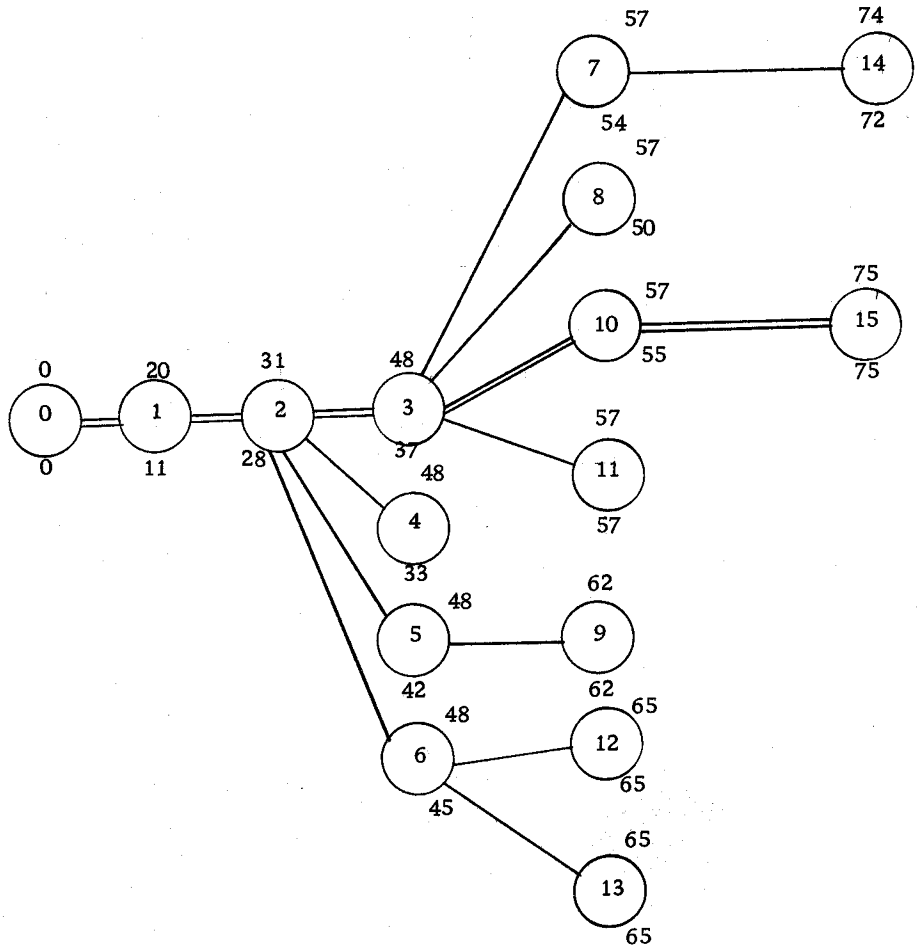
Stage	Marked Element Numbers	State Number	State Element Numbers	State Time	Unmarked Element Numbers
0		0	Empty set	0	1
1	1	1	1	11	2
2	2	2	1, 2	28	3, 4
3	3, 4	3	1, 2, 3	37	6
		4	1, 2, 4	33	5*
		5	1, 2, 3, 4	42	5, 6
4	5, 6	6	1, 2, 3, 6	45	8
		7	1, 2, 3, 4, 5	54	7
		8	1, 2, 3, 4, 6	50	8
		9	1, 2, 3, 4, 5, 6	62	7, 8
5	7, 8	10	1, 2, 3, 6, 8	55	-
		11	1, 2, 3, 4, 5, 7	57	-
		12	1, 2, 3, 4, 6, 8	60	-
		13	1, 2, 3, 4, 5, 6, 7	65	-
		14	1, 2, 3, 4, 5, 6, 8	72	-
		15	1, 2, 3, 4, 5, 6, 7, 8	75	-

* Task 5 can not be done unless task 3 is done.

Table 3-12. Shortest route calculation.

Minimum Total Number of Arcs to Nodes (I)	Node Number (II)	Node Numbers in the Next Stage to Which Arcs are Spread from the Node Number in Column II. (III)
0	0	0
1	1	2
2	2	3, 4, 5, 6
3	3	7, 8, 10, 11
	4	-
	5	9
	6	12, 13
4	7	14
	8	-
	9	-
	10	15
	11	-
	12	-
	13	-

In Table 3-5, if we start at node 15 in column III, we reach node 0 via nodes 10, 3, 2, 1 in the shortest way. So this indicates the shortest path ($0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 10 \rightarrow 15$) shown in double line in Figure 3-5.



Legend: shortest route

b

a

c

b = maximum state time allowed to reach that state [$t(S_j) + c$]

a = node or state number

c = state times of node a = sum of times of tasks included in the node a = $t(S_i)$

Figure 3-5. Shortest route through network according to Gutjahr-Nemhauser's algorithm.

Table 3-13. Results of station assignments by Gutjahr-Nemhauser Algorithm.

Arc Number	Nodes Spanned by Arc	Work Elements Spanned by Arc $(S_j - S_i)$	Station Time	Station Idle Time
1	S_1, S_0	u_1	11	9
2	S_2, S_1	u_2	17	3
3	S_3, S_2	u_3	9	11
4	S_{10}, S_3	u_6, u_8	18	2
5	S_{15}, S_{10}	u_4, u_5, u_7	20	0

Heuristic Approaches

The computational difficulty of using an analytical approach, even for a smaller size problem, had lead to the development of heuristic procedures to solve the combinatorial ALB problem. Heuristic procedures involve simple rules based on intuition and judgement. These rules are easily programmable on a digital computer. Also, additional constraints such as positional and zoning restrictions can be added easily. The acronym HALB is used to indicate heuristic method of balancing assembly lines.

The first HALB was developed by Tonge (1960). Later several authors developed heuristic methods based on different criteria. Mastor (1966) made a comparative study of the ALB techniques and suggested that COMSOAL developed by Arcus (1966) was an effective method to solve large size problems. However, COMSOAL does not guarantee an optimum solution and, moreover, employs numerous heuristics and weighting rules.

The existing heuristic methods are illustrated in the following pages in the unified terminology (Chapter II) with example. The models are arranged in a chronological order of their publication.

A. Aggregation of Groups of Elements (Tonge, 1960)

The heuristic method developed by Tonge involves three phases and employs various heuristics based on grouping of work elements. The different phases are summarized as follows:

Phase 1. This phase constructs a hierarchy of increasingly simpler line balancing problems, by aggregating groups of elements into a single compound element. Each of these compound elements is in itself a member of this same class of line balancing problems. This is because each compound element is composed of elements requiring a given elemental time and among whom precedence relations exist. To accomplish phase 1 of the solution method, three types of compound elements are defined. They are:

1. Chain. A group of adjacent elements whose relative order is completely determined, and each except the first having a single direct-predecessor and each except

the last having a single direct follower, can be replaced by a compound element called a "chain".

2. Set. A group of elements whose relative order is completely unspecified, all having the same direct predecessors and followers, can be replaced by a single compound element called a "set".

3. Z. A Z is a group of four elements with two front elements having common predecessors and the other two back elements having common followers. The single direct follower of one front element is one of the back elements; the two direct followers of the front element are the back elements. The back elements have no predecessors (Tonge, 1960, p. 27-28).

Figure 3-6 shows the three types of compound elements defined with respect to the 21 element problem in Figure 3-11. Given an assembly of elements with a single front element, the procedure attempts to create a "Chain". Given an assemblage of elements with several front elements, the procedure attempts to create a "Set" if possible or a "Z" otherwise. Phase 1 is to be carried out only once for a given problem.

Phase 2. This phase solves a simple line balancing problem (with a small number of compound elements) by (1) assigning groups of available workmen (work stations) to elements and then (2) taking as subproblems those compound elements (simple problems in themselves)

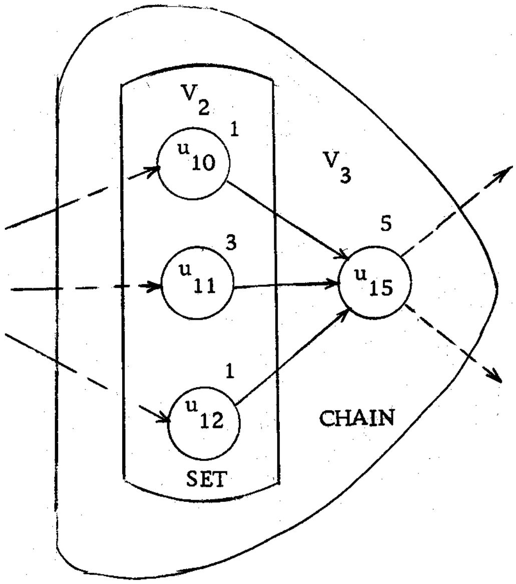


Figure 3-6a. Illustration of set and chain.

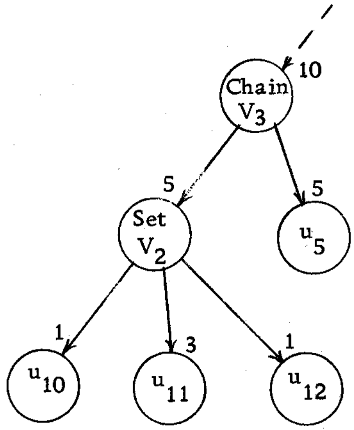


Figure 3-6b. Tree diagram representation of set and chain relations.

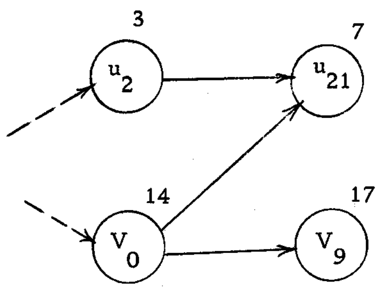


Figure 3-6c. Illustration of Z-aggregation.

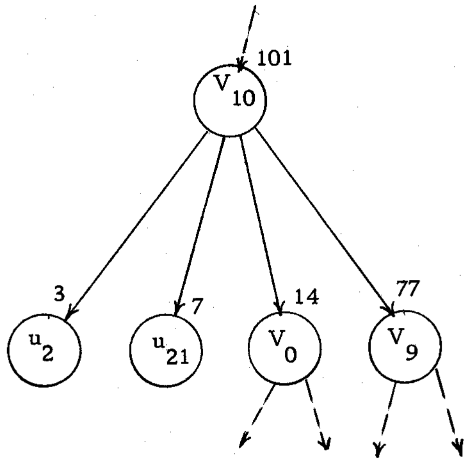


Figure 3-6d. Tree diagram representation of Z-aggregation.

Figure 3-6. Illustration of set, chain and Z-aggregations.

which have been assigned more than one man. The inputs to phase 2 are the problem hierarchy as developed in phase 1, a cycle time determined by the production rate, and a "percent usable time" supplied as a guide to set and accept the potential work stations. Phase 2 uses five regrouping heuristics, namely, (a) Direct Transfer, (b) Trading, (c) Sequential Grouping. (d) Complete Grouping and (e) Exhaustive Grouping. These are thoroughly discussed by Tonge (1960, 1961).

Phase 3. This phase attempts to even the distribution of work among the work stations by repeatedly reducing the time requirement of the largest work station. The output of the second phase and a cycle time would be the inputs to this phase. This phase minimizes the cycle time directly by using the five regrouping heuristics used by phase 2.

Example:

Consider the 11 element problem in Figure 3-7. The tree diagram in Figure 3-8 is constructed using phase 1. Note that the groups that make up a chain must be performed from left to right. This does not apply to the groups that make up a set. For exampel C_2 is made

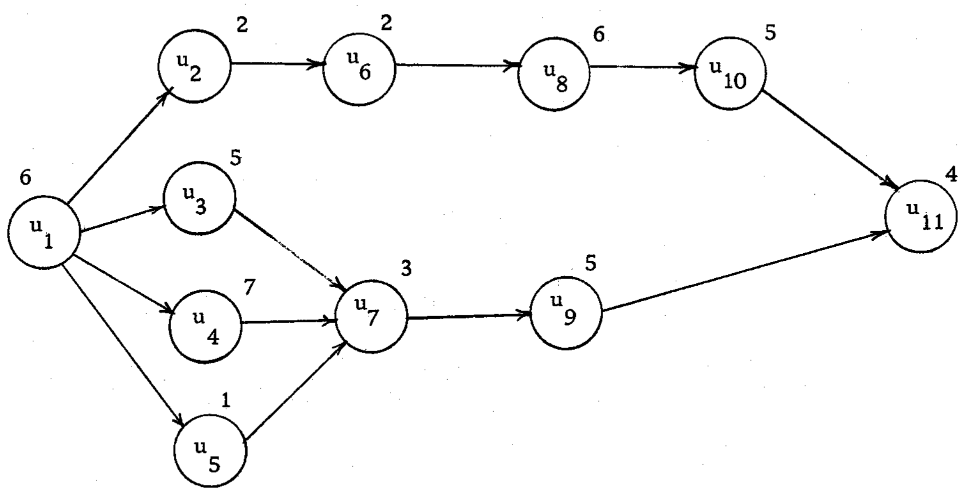


Figure 3-7. Precedence diagram for 11 elements to illustrate Tonge (1960)'s heuristic method.

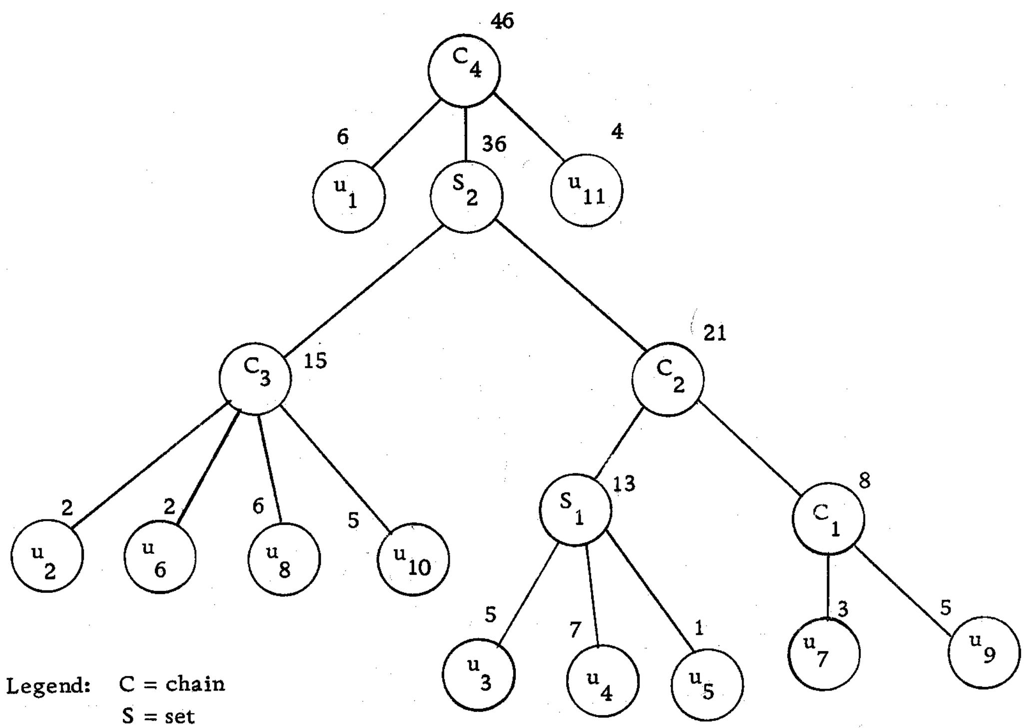


Figure 3-8. A tree diagram of compound elements for 11 element problem.

up of two groups S_1 and C_1 (Refer Figure 3-8). This indicates that S_1 precedes C_1. The set S_2 is made up of C_3 and C_2 but this does not imply that C_3 precedes C_2. C_4 is made up of u_1, S_2 and u_{11}. Thus u_1 precedes S_2 and S_2 in turn precedes u_{11}. This precedence relation applies to u_7 and u_8 for C_1 and u_2, u_6, u_8 and u_{10} for C_3 as well.

Now phase 2 is to be applied. For this we need a cycle time. Let $c = 10$ minutes and $\Sigma t_i = 46$. m^* (integer) $= [\frac{46}{10}] = 5$. Starting at the top of the tree diagram of Figure 3-8, we note that C_4 is the topmost group and it is a chain. So to observe the precedence relation elements must be picked from left. u_1 requires 6 minutes and there are 4 left. S_2 is too big, so its components C_2 and C_3 are examined. They are also too big. Since S_2 is a set, C_3 is arbitrarily tried. It is found that u_2 and u_6 fill up the first station. Hence, the set $\{u_1, u_2, u_6\}$ makes up the first station.

Now the attack is from the rear, and the tasks are assigned to the last station. Working from back of C_4, u_{11} is inserted first. Again the components of S_2 are examined. This time C_2 is tried first. Since C_2 is a chain, and because we are assigning to the last station C_1 must be tried first to comply with the precedence relations. Working from the back of C_1, u_9 is assigned leaving 1 free minute in the station. The element u_7 will not fit due to precedence violations. Going back up the tree S_1 can not be tried since C_1 is not completely assigned. But C_3 can be tried since S_2 is a set. Working backwards from C_3, u_{10} is too big and hence we stop the search. At this point the last station consists of u_9 and u_{11} (Figure 3-9).

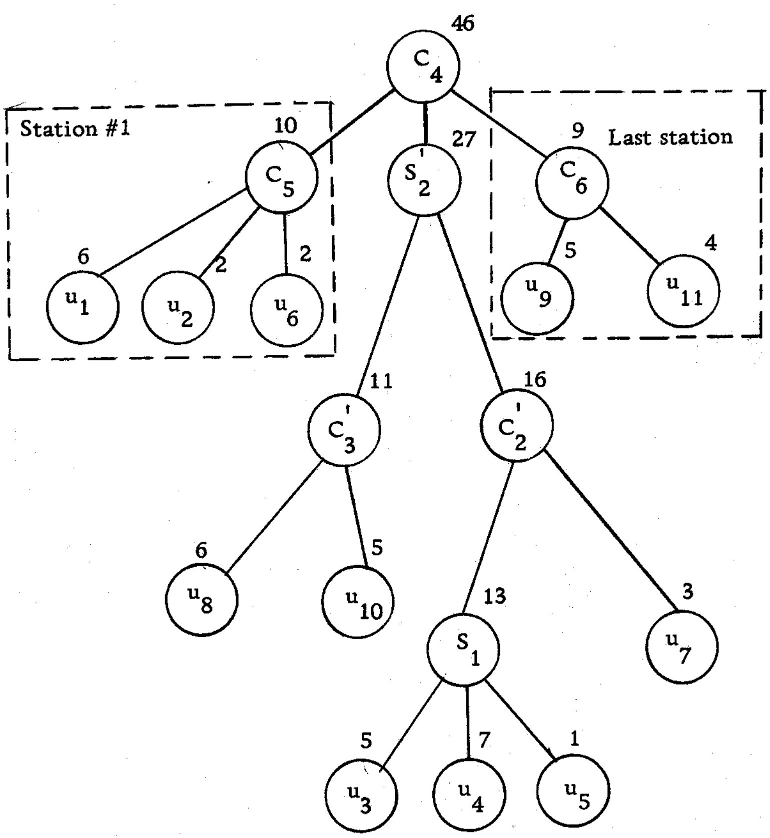


Figure 3-9. Tree diagram after partial application of phase 2 of Tonge's heuristic method.

Now an attempt is made to assign the tasks in S_2' to the three remaining stations. We start at the top of the modified diagram (Figure 3-9). Since S_2' is at the top, choose C_3' arbitrarily. The element u_8 requires 6 minutes, so there are 4 minutes left out. Work element u_{10} is too big while either u_3 or u_4 also do not fit for u_8. However, the set $\{u_8, u_5\}$ can be accommodated at station 2 leaving 3 minutes of slack time. Now the assignment of all but one station can be attained by proceeding from the rear of S_2', i.e., u_7 is selected. The set S_1 is too big, while u_4 fits leaving zero idle time. So station 4 will contain the set of elements $\{u_{10}, u_3\}$. The tree at the end of phase 2 (Figure 3-10) shows that the assignments are complete at this stage.

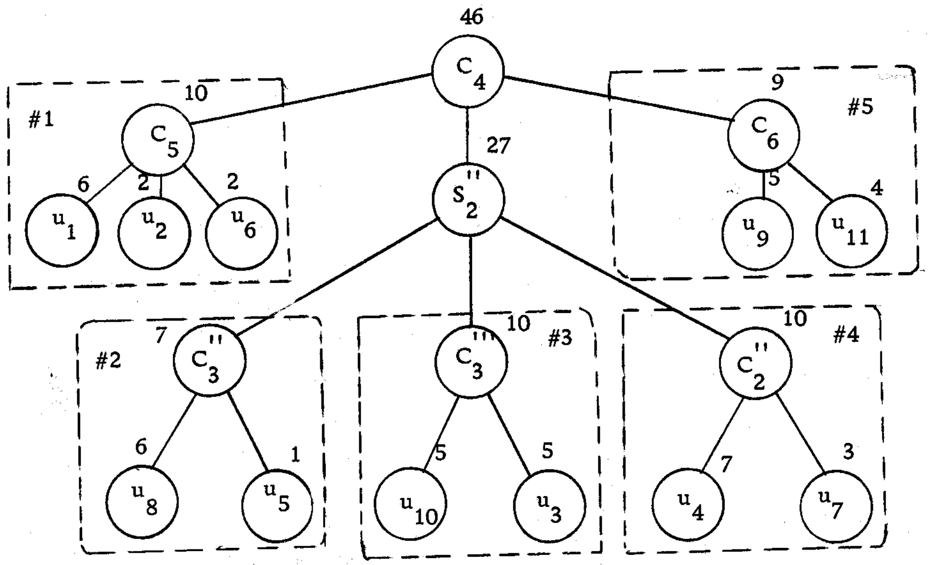


Figure 3-10. Tree diagram at the end of phase 2 of Tonge's procedure.

B. Transferability and Permutability of Work Elements (Kilbridge and Wester, 1961)

A heuristic method of ALB was developed by Kilbridge and Wester using the method of Jackson (1956) in constructing a precedence diagram. This method can be extended to most practical problems by adding zoning and positional restrictions.

The basic heuristics used in this method are based on

1. The permutability of elements within each column. This is the indifference of the order in which tasks in the same column are performed.
2. The lateral transferability of work elements. The elements can be moved between columns of precedence diagram.

The objective is to minimize the % balance delay, i. e.,

$$\text{minimize} \qquad d = \frac{(nc - \sum t_i)}{nc} \times 100$$

Example:

The 21-element problem shown in Figure 3-11 is solved by this approach for a cycle time $c = 20$ minutes. The precedence diagram in Figure 3-11 is constructed using the method of Jackson (1956). Summary of the diagram appears in Table 3-14. Column C of the Table 3-14 indicates the transferability of the elements.

Solution:

Since $c = 20$, and we have 21 as the cumulative time in column III of the Table 3-14, we have to attempt the transferability heuristic for the element u_2. By shifting u_2 to column III after u_4 the modified grouping of the elements is shown in Table 3-14a where entries up to column V are shown and other entries which did not change from the earlier Table 3-14, are not repeated.

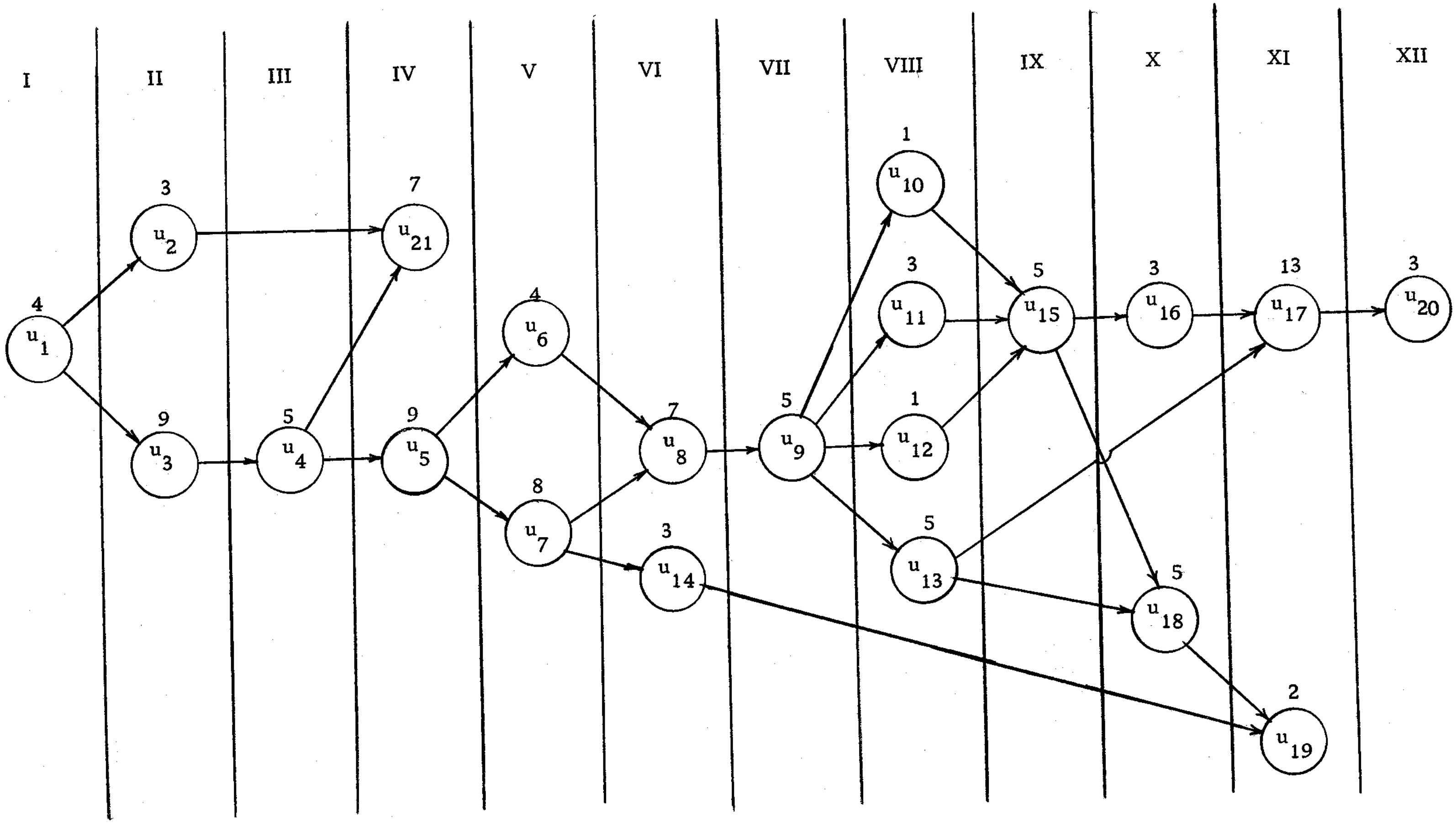


Figure 3-11. Precedence diagram of 21 elements constructed as per Jackson's method to illustrate Kilbridge and Wester's HALB.

Table 3-14. Tabular representation of work elements for 21 element problem to illustrate Kilbridge and Wester's HALB.

Column Number of Diagram (A)	Element (B)	Remarks (C)	Element Time Duration (D)	Sum of Time durations in Each Column (E)	Cumulative Time Units (F)
I	u_1		4	4	4
II	$u_2(w. u_{21})$	→ III to XI	3		
	u_3		9	12	16
III	u_4		5	5	21
IV	u_{21}	→ V to XII	7		
	u_5		9	16	37
V	u_6		4		
	u_7		8	12	49
VI	u_8		7		
	$u_{14}(w. u_9)$	→ VII to XI	3	10	59
VII	u_9		5	5	64
VIII	u_{10}		1		
	u_{11}		3		
	u_{12}		1		
	$u_{13}(w. u_{18}, u_{19})$	→ to X	5	10	74
IX	u_{15}		5	5	79
X	u_{16}		3		
	$u_{18}(w. u_{19})$	→ to XI	5	8	87
XI	u_{17}		13		
	u_{19}	→ to XII	2	15	102
XII	u_{20}		3	3	105

$(w. u_i)$ indicates with the element u_i.

Table 3-14a. Modified grouping of work elements obtained from Table 3-14 for 21 element problem.

(A)	(B)	(C)	(D)	(E)	(F)
I	u_1		4		
II	u_3		9		
III	u_4		5	18	18
	u_2(w. u_{21})	→ III to XI	3	3	21
IV	u_{21}	→ V to XII	7		
	u_5		9	16	37
V	u_6		4		
	u_7		8	12	49

(The remaining entries of Table 3-14 remain same.)

Now referring to column F of Table 3-14a, it is evident that the cumulative sum of 38 will occur within the column V. We cannot transfer u_2 to column V. Since u_2 is associated with u_{21} for a transfer. So u_2, u_{21} and u_5 must be placed under the station 2, resulting in 1 unit of idle time. Similar heuristics will result in another Table 3-14b.

From Table 3-14b, station 5 has only 13 time units resulting in a maximum delay of 7 time units, while stations 2 and 3 have 19 time units each with a delay of 1 time unit at each station. So to smoothen out the station assignments, we can transfer u_{14} to column X of the diagram so that station 4 will have 15 time units as its work station

Table 3-14b. Revised table obtained by Kilbridge and Wester's HALB for 21 element problem for a cycle time c = 20 units of time.

(A)	(B)	(C)	(D)	(E)	(F)	
I	u_1		4			Station #1
II	u_3		9			Station #1
III	u_4		5	18	18	Station #1
	u_2		3			Station #2
IV	u_{21}	→	7			Station #2
	u_5		9	19	37	Station #2
V	u_6		4			Station #3
	u_7		8			Station #3
VI	u_8		7	19	56	Station #3
	u_{14}	→	3			Station #4
VII	u_9		5			Station #4
VIII	u_{10}		1			Station #4
	u_{11}		3			Station #4
	u_{12}		1			Station #4
	u_{13}	→	5	18	74	Station #4
IX	u_{15}		5			Station #5
X	u_{16}		3			Station #5
	u_{18}	→	5	13	87	Station #5
XI	u_{17}		17			Station #6
	u_{19}	→	2			Station #6
XII	u_{20}		3	18	105	Station #6

time while the station 5 will have 16 units. With this small modification the station assignments are shown in Table 3-15.

Table 3-15. Final solution of task assignments to work stations by Kilbridge and Wester's HALB for a a cycle time c = 20 minutes.

Station Number	1	2	3	4	5	6
Tasks	u_1, u_3, u_4	u_2, u_{21}, u_5	u_6, u_7, u_8	u_9, u_{10}, u_{11} u_{12}, u_{13}	u_{14}, u_{15} u_{16}, u_{18}	u_{17}, u_{19} u_{20}
Station Time	18	19	19	15	16	18

Since $c = 20$ and $\Sigma t_i = 105$, minimum number of stations $(m^*) = [\frac{105}{20}] = 6$, which is the same as the result obtained in Table 3-15.

<u>Remarks:</u>

If this solution is implemented, the cycle time will be 19 times units (in the steady state) instead of the given value of $c = 20$. The balance delay is 12.5% at $c = 20$ and 3% at $c = 19$. The cycle time of 20 will reduce the balance delay and increase the production rate in case of increased demand schedule.

This heuristic approach is suitable for flexible production schedules since it is easy to arrive at different station balances by merely shifting the elements in the columns of the diagram.

C. Ranked positional Weight Technique (Helgeson and Birnie, 1961)

This is a heuristic approach suitable for computer applications. This method does not offer optimal solutions and is effective only when the problem is formulated to minimize the number of work stations for a given cycle time. An alternate formulation with minimum cycle time for a given number of work stations can be obtained by balancing the assembly line over a wide range of cycle times. The positional weight calculations are based on either a precedence diagram or a precedence matrix.

Table 3-9 denotes the precedence matrix for the 9 element problem shown in Figure 3-1. In this matrix entries are shown only above the diagonal, where presence of 1 indicates "must precede" and 0 an "unordered" relationship.

Table 3-16. Diagonal precedence matrix of 9 work elements problem shown in Figure 3-1 to illustrate positional weight HALB.

Element Time	Element → ↓	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9
4	u_1		1	1	1	1	1	1	1	1
2	u_2			1	1	1	0	0	0	1
5	u_3				1	0	0	0	0	1
7	u_4					0	0	0	0	1
5	u_5						0	0	0	1
6	u_6							1	1	1
8	u_7								1	1
3	u_8									1
8	u_9									

Positional Weight Calculations. This is done by adding together the time values for the specific work element and all work elements that must follow it as defined in the precedence matrix. This positional weight is labelled as "linear positional weight" (Linpow), e. g., Linpow of $u_2 = (t_2 + t_3 + t_4 + t_5 + t_9) = 2 + 5 + 7 + 5 + 8 = 27$. This calculation can be easily done by a computer using the precedence matrix in Table 3-16. The listing of a computer program written in FORTRAN-IV to calculate positional weights appears in the Appendix. First, the positional weights are calculated and listed in the unsorted Table 3-17. Then work elements are sorted and listed in their descending order of positional weights (Table 3-18).

Table 3-17. Unsorted positional weight list.

Work Element	Positional Weight	Immediate Precedence
u_1	48	-
u_2	27	u_1
u_3	20	u_1
u_4	15	u_2
u_5	13	u_6
u_6	25	u_3
u_7	19	u_2
u_8	11	u_7
u_9	8	u_4, u_5, u_8

Table 3-18. Sorted positional weight list.

Work Element	Positional Weight	Immediate Precedence
u_1	48	-
u_2	27	u_1
u_3	25	u_1
u_4	20	u_2
u_5	19	u_6
u_6	15	u_3
u_7	13	u_2
u_8	11	u_7
u_9	8	u_4, u_5, u_8

The following rules denote the heuristic method of assigning of work elements to work stations assuming a given cycle time for the ALB problem.

Rule 1. Select the work element with the highest positional weight and assign it to the first work station [all $t_i \leq c$].

Rule 2. Calculate the unassigned time for the work station by calculating the cumulative times of all work elements assigned to the station and subtract this sum from the cycle time.

Rule 3. Select the work element with the next highest positional weight and attempt to assign it to the work station after making the following checks.

a. Check the list of already assigned work elements. If the "immediate precedent" work element has been assigned precedence will not be violated; proceed to rule 3-b, otherwise proceed to rule 4.

b. Compare the work elemental time with the unassigned time. If the work element time is less than the work station unassigned time, assign the work element and recalculate the unassigned time. If the work unit time is greater than the unassigned time proceed to rule 4.

Rule 4. Continue to select, check and assign if possible until one of the two conditions has been met.

a. All work elements have been assigned

b. No unassigned work element remains that can satisfy both the precedence requirements and the "less than the unassigned time" requirement.

Rule 5. Assign the unassigned work element with the highest positional weight to the second work station and proceed through the preceding rules in the same manner.

Rule 6. Continue assigning work elements to work stations until all work elements have been assigned. At that time a solution to the balancing problem will have been found (Helgeson and Birnie, 1961, p. 396).

Example:

Consider the problem of 9 work elements given in the precedence matrix of Table 3-16. For a given cycle time of c = 10 minutes, the solution is obtained in the following Tables 3-19 through 3-19e. The column (7) shows whether an element is accepted or rejected. A rejection may be for two reasons: (1) the cumulative station time is greater than the given cycle time (this is denoted by reject > c in the remarks), and (2) the particular element violates precedence relations (denoted by p in the remarks). A square around an entry in column (6) denotes that further assignment stopped at that element and another work station must be started. The sum of the times in all the enclosed squares give total delay in all the stations.

Table 3-19. Assignment to work station 1 by positional weight method.

Element (1)	Positional Weight (2)	Immediate Precedence (3)	Element Time (4)	Cumulative Station Time (5)	Unassigned Station Time (6)	Remarks (7)
u_1	48	--	4	4	6	assign
u_2	27	u_1	2	6	$\boxed{4}$	assign
u_6	25	u_1	6	12		reject > c
u_3	20	u_2	5	11		reject > c
u_7	19	u_6				reject, p
u_4	15	u_3				reject, p
u_5	13	u_2	5			reject > c
u_8	11	u_7				reject, p
u_9	8	u_5, u_4, u_8				reject, p

Table 3-19a. Assignments to work station 2.

(1)	(2)	(3)	(4)	(5)	(6)	(7)
u_6	25	u_1	6	6	$\boxed{4}$	assign
u_3	20	u_2	5	11		reject > c
u_4	15	u_5	7			reject, p
u_5	13	u_2	5	11		reject > c
u_8	11	u_7	3			reject, p
u_9	8	u_8, u_4, u_5	-			reject, p

Table 3-19b. Assignments to work station 3.

(1)	(2)	(3)	(4)	(5)	(6)	(7)
u_3	20	u_2	5			assign
u_7	19	u_6	8	13		reject > c
u_4	15	u_3	7	13		reject > c
u_5	13	u_2	5	10	$\boxed{0}$	assign

Table 3-19c. Assignments to work station 4.

(1)	(2)	(3)	(4)	(5)	(6)	(7)
u_7	19	u_6	8	8	2	assign
u_4	15	u_3	7	15		reject > c
u_8	11	u_7				reject, p
u_9	8	u_8, u_4, u_5				reject, p

Table 3-19d. Assignments to work station 5.

(1)	(2)	(3)	(4)	(5)	(6)	(7)
u_4	15	u_6	7	7	3	assign
u_8	11	u_7	3	10	$\boxed{0}$	assign

Table 3-19e. Assignments to work station 6.

(1)	(2)	(3)	(4)	(5)	(6)	(7)
u_9	8	u_7, u_4, u_5	8	8	$\boxed{2}$	assign

The algorithm is ended at this stage since all work elements have been assigned.

The summary of the assignments to work stations is in Table 3-20.

Table 3-20. Summary of assignments to work stations by Helgeson and Birnie's positional weight HALB at a cycle time $c = 10$.

Station Number (k)	1	2	3	4	5	6
Tasks	u_1, u_2	u_6	u_3, u_5	u_7	u_4, u_8	u_9
Station Time	6	6	10	8	10	8
Station Idle Time	4	4	0	2	0	2

Since $\sum_{k=1}^{6} d_k = 12 > 10$ implies the result is not optimal (not minimal station) balance. The balance delay for this result is 20%. However this approach is suitable for computer applications to arrive at alternative solutions.

An alternative set of solutions, not necessarily an improvement over the original set of solutions, can be obtained by the "inverse positional weight" technique. The calculations for the inverse positional weights are made by summing the work element time with the elemental times of all work elements indicating a 1 in the column of the specified work element rather than in the row of the precedence matrix.

Using Table 3-16, the inverse positional weights are calculated and the unsorted list is shown in Table 3-21, while 3-22 is the sorted list in descending order of inverse positional weights.

Table 3-21. Unsorted inverse positional weight list.

Work Element	Inverse Positional Weight	Immediate Precedence
u_9	48	--
u_8	21	u_9
u_7	18	u_9
u_6	10	u_7
u_5	11	u_9
u_4	18	u_9
u_3	11	u_4
u_2	6	u_3, u_5
u_1	4	u_2, u_6

Table 3-22. Sorted inverse positional weight list.

Work Element	Inverse Positional Weight	Immediate Precedence
u_9	48	--
u_8	21	u_9
u_7	18	u_8
u_4	18	u_9
u_5	11	u_9
u_3	11	u_4
u_6	10	u_7
u_2	6	u_3, u_5
u_1	4	u_2, u_6

Following the same rules as described earlier, an alternative grouping of elements is obtained in Table 3-23.

Table 3-23. Summary of an alternative assignment to work stations by inverse positional weight method.

Station Number (k)	6	5	4	3	2	1
Tasks	u_9	u_8, u_4	u_7	u_5, u_3	u_6, u_2	u_1
Station Time	8	10	8	10	8	4
Station Idle Time (d_k)	2	0	2	0	2	6

Balance delay = 20% (no improvement)

Total delay = 12 > c , i. e., not optimal.

Thus, the inverse positional weight method also does not guarantee an optimal solution.

D. Random Generation of Feasible Sequences (Arcus, 1966)

A Computer Method of Sequencing Operations for Assembly Lines (COMSOAL) was developed by Arcus. It was based on the idea of generating a large number of feasible sequences by Random Generator. Out of those one sequence is chosen which gives the fewest stations, where each work station is loaded with tasks in the order of the sequence. For example, a feasible sequence of tasks in the problem given in Figure 3-3, is $u_1 - u_2 - u_4 - u_3 - u_5 - u_6 - u_8 - u_7$. The least number of work stations required for this sequence for a given cycle time of c = 20 minutes is given and tasks being assigned to stations as shown in Table 3-24.

Table 3-24. Summary of assignments of tasks to work stations.

Station Number	1	2	3	4	5
Elements Assigned	u_1	u_2	u_4, u_3	u_5, u_6	u_8, u_7
Station Time	11	17	14	20	13

Determination of Sample Size for the Number of Feasible Sequences to be Generated

Let

N = number of feasible sequences (including replications) generated.

r = proportion of the universe of feasible sequences which constitutes optimal sequences.

Then the probability that the first sequence generated is optimal is r that that it will not is $(1-r)$. Thus the probability that none is optimal is $(1-r)^N$ and probability that the optimal number is greater than or equals $P = 1 - (1-r)^N$. From this $1 - P = (1-r)^N$.

Taking logarithms of both sides:

$$\text{Log}\,(1-P) = N \cdot \text{Log}\,(1-r)$$

$$N = \text{Log}\,(1-P)/\text{Log}\,(1-r)$$

For example, if $r = 0.001$ about 4600 sequences (including replications) would be required to be 99 percent certain that at least one is optimal. The problems encountered in the determination of N is in identifying r and generating of feasible sequences randomly and economically. Arcus claimed that the only economical generator was a progressive selector of tasks from among those which have no unassigned preceding tasks. Assuming that no means of determining r (or its process equivalent) is known, other than by the impractical enumeration and evaluation of sequences, Arcus hypothesized that an economic sample (say 1000 sequences), in the vast majority of lines would contain at least one optimal sequence.

Steps for the Generation of Feasible Sequences

The following steps are illustrated with the example in Figure 3-12 for a cycle time of $c = 10$ minutes.

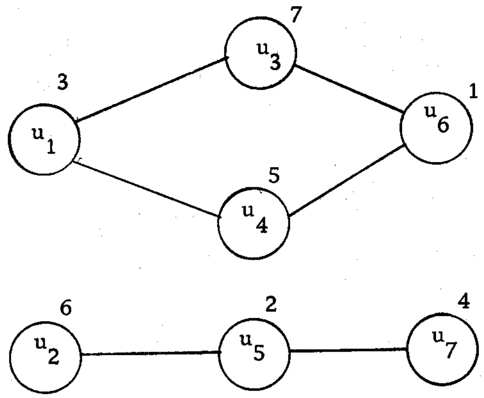


Figure 3-12. Precedence diagram to illustrate Arcus's method of generating feasible sequences.

Step 1. Represent each task and its immediate followers in the precedence diagram either by a matrix or by a single list, plus a list of row entry points. Call this list the Initial List.

Initial List:

Task	u_1	u_2	u_3	u_4	u_5	u_6	u_7
Immediate Followers	u_3, u_4	u_5	u_6	u_6	u_7	NIL	NIL

Step 2. From one scan of the Initial List, place in another List A for each task on the line, the total number of tasks which immediately precede it in the precedence diagram.

List A:

Task	u_1	u_2	u_3	u_4	u_5	u_6	u_7
Total Number of Immediate Precedents	0	0	1	1	1	2	1

Step 3a. Referring to List A, place in a new List B all tasks with 0 against them in List A.

List B: Task

u_1

u_2

Step 3b. Transfer to a third list, List C, those tasks in List B which have elemental times no greater than the available time.

List C: Task

u_1

u_2

Step 4. Select a task from List C randomly and assign it to the first station. Calculate the slack units available. For example, let us select the task (say) u_2.

Step 5. Eliminate the selected task from List B and move all tasks below the selected task up one position in List B.

List B: (updated) Task

u_1

<u>Step 6</u>. Scan the row of the selected task in the Initial List. Note the tasks immediately following the selected task and deduct 1 from the number associated with each in List A.

<u>List A:</u>

Task	u_1	u_2	u_3	u_4	u_5	u_6	u_7
Total Number of Preceding Tasks	0	0	1	1	0	2	1

<u>Step 7</u>. Add to List B those tasks which (a) immediately follow the selected task and (b) presently have a zero against them in List A.

<u>List B:</u>

Task
u_1
u_5

Repeat steps 3- b through 7 till all tasks have been assigned.

In this procedure, the evaluation is progressive, i. e., the available time is diminished as each task is generated and a task too large for remaining time becomes the first at the next station. As a sequence is completed, the number of stations obtained in that sequence is compared with that of the previous best sequence. If there is an improvement, a new sequence is stored and the old is discarded; thus there are never more than two sequences in store at any time (Arcus, 1966, p. 264).

The steps 1 through 7 described above proceed by assigning probability to all the tasks that could come next. But this procedure may not be successful for certain problems (for example for Kilbridge and Wester (1960)'s 45 element problem. So to obtain an almost optimal sequence, a weighting scheme with some additional rules is given. The summary of the additional rules is as follows.

Rule 1. Weight tasks that fit in proportion to the standard performance time. One effect of this rule is to prefer large tasks early at each station and in the entire sequence.

Rule 2. Weight tasks that fit in proportion to the number of immediately following tasks plus 1. This leads to tighter packing. By adding 1, it is ensured that a task without followers will not be excluded.

Rule 3. Weight tasks that fit in proportion to the number of immediately following tasks (which subsequently become available and fit, plus the number of other tasks in the fit list (which subsequently fit) plus 1. This rule favours small tasks unlike Rule 1.

Rule 4. Weight tasks that fit by $1/X$, where X is equal to the total number of tasks to be performed minus the number of all tasks which precede or follow (immediately and subsequently) the task being considered.

Rule 5. Weight tasks that fit by $1/X'$ where X' is equal to the total number of unassigned tasks minus 1 and minus the number of all the tasks which follow the task being considered.

Rule 6. Weight tasks that fit by the total number of all following tasks plus 1.

Rule 7. Weight tasks that fit by the times of the task and of all the following tasks.

Rule 8. Weight tasks that fit by the total number of following tasks plus 1, divided by the number of levels, which those following tasks occupy plus 1.

Rule 9. Weight tasks that fit by the product of the weights computed by Rules 1, 5, 6, 7 and 8 (Arcus, 1966, p. 265-267).

For very large complex problems COMSOAL and heuristic method of line balancing by Kilbridge and Wester are probably the most effective (Buffa, 1968, p. 261).

E. Precedence Matrix Manipulation (Hoffman, 1963)

By simple operations on a precedence matrix of zeros and ones, optimal balances can be arrived by Hoffman's precedence matrix method. This method can handle lines up to 99 tasks and has balanced 19 to 76 lines in 3 to 10 minutes on CDC-1604 computer. The difference between this technique and that of Helgeson and Birnier (1961), is that the latter sums the times for all succeeding elements while the former selects the elements on the basis of the total number of succeeding elements. Generation of feasible sequences in a precedence matrix is achieved by simple matrix operations considered by Hoffman (1959). If the immediate precedence matrix is called P, then letting $S = P + P^2 + P^3 + \cdots + P^n$, and as n tends to infinity, we have $\lim_{n \to \infty} S = \frac{P}{(I-P)}$, where I denotes the identity matrix.

The elements of S are the number of paths from the task in the row to the task in the column. Manipulating the matrix S, all feasible sequences can be obtained. Hoffman (1963) proposes successive maximum elemental time method to the ALB problem and claims that backward balancing usually does lead to optimal solutions if forward balancing does not for a specified feasible sequence.

Steps for the Algorithm

To illustrate the various steps proposed by Hoffman, an augumented matrix has to be prepared first.

From the precedence diagram a precedence matrix, representing the immediate precedence relations, is made. Each column of the matrix is summed and these sums form another row adjoined to the bottom row of the matrix. A Code Number, say K_1, is given to this new row of the augmented matrix. Next the diagonal of the matrix is labelled with any arbitrary value (D). The first code number, K_1, consists of n integers, n being the number of elements to be scheduled, at least one of which is zero. The elements heading the columns in which there are zeros in K_1 are candidates for the first position in the list of feasible permutations and only those elements can be candidates. The following steps describe the Hoffman's procedure.

Step 1. Search left to right in the Code number for a zero.

Step 2. Select the element which heads the column in which the zero is located.

Step 3. Subtract the element's time from the cycle time remaining.

Step 4. If the result of step 3 is positive go to step 5.

Step 4a. If the result of step 3 is negative go to step 6.

Step 5. Subtract from the Code number the row corresponding to the element selected and use this result as a new Code number.

Step 6. Go to step 1 and start search one element to the right of the one just selected and repeat steps 1 to 6 until all the columns have been examined, then go to step 7.

Step 7. Subtract the remaining cycle time (the slack time) of the present combination from the slack time of the previous combination generated (if this is the first, then subtract from the cycle time).

Step 8. If the result of step 7 is zero or negative go to step 9.

Step 8a. If the result of step 7 is positive, then this set of elements just generated becomes the new combination for this station. Go to step 10.

Step 9. Go back one Code number and go back to step 1, starting one element to the right of the element which had 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 8 is the one having the maximum elemental time for the station.

Step 10. 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.

Step 11. Repeat the previous steps until all the elements have been assigned (Hoffman, 1963, p. 553-554).

Example:

Table 3-25 illustrates the solution procedure for a 9 element problem, represented in a matrix form at the top of the table. It has 9! or 362,880 permutations of 9 elements. Hoffman's method reduced

Table 3-25. Generation of feasible combinations for station assignments by Hoffman's procedure.

i \ j	1	2	3	4	5	6	7	8	9
1	D	1	1						
2		D		1					
3			D	1					
4				D	1	1	1		
5					D			1	
6						D			1
7							D		1
8								D	1
9									D

	1	2	3	4	5	6	7	8	9	Element i selected	Elemental time (t_i)	Remaining station time before selection	Remaining station time after selection	Elements contained in combination after selection	Steps applied
K_1	[0	1	1	2	1	1	1	1	3]	1	5	10	5	1	1, 2, 3
	D	1	1	0	0	0	0	0	0						
K_2	-D	[0	0	2	1	1	1	1	3]	2	3	5	2	1, 2	4, 5, 6
	0	D	0	1	0	0	0	0	0						
K_3	-D	-D	[0	1	1	1	1	1	3]	3	4	2	(-)	1, 2	4, 5, 6
K_3	-D	-D	0	1	1	1	1	1	3						6
K_2	-D	0	[0	2	1	1	1	1	3]	3	4	5	1	1, 3	7, 8, 9
	0	0	D	1	0	0	0	0	0						
K_3'	-D	0	-D	[1	1	1	1	1	3]				1	1, 3	4, 5, 6
K_2	-D	0	0	[2	1	1	1	1	3]			5		1	9
K_1	0	[1	1	2	1	1	1	1	3]			10		0	9*
K_3'	-D	[0	-D	1	1	1	1	1	3]	2	3	10	7	2	1, 2, 3
	0	D	0	1	0	0	0	0	0						
K_4	-D	-D	[-D	0	1	1	1	1	3]	4	5	7	2	2, 4	4, 5, 6
	0	0	0	D	1	1	1	0	0						
K_5	-D	-D	-D	[-D	0	0	0	1	3]	5	4	2	(-)	2, 4	4, 5, 6
K_5	-D	-D	-D	-D	0	[0	0	1	3]	6	5	2	(-)	2, 4	6
K_5	-D	-D	-D	-D	0	0	[0	1	3]	7	1	2	1	2, 4, 7	6
	0	0	0	0	0	0	D	0	1						
K_6	-D	-D	-D	-D	0	0	-D	1	2			1		2, 4, 7	6
K_5	-D	-D	-D	-D	0	0	0	1	3			1		2, 4, 7	9
K_4	-D	-D	-D	0	1	1	1	1	3			1		2, 4, 7	9**
K_3	-D	0	-D	1	1	1	1	1	3						

and so on

* Assignment to station 1 is complete.

[ ] = range of search

** Assignment to station 2 is complete.

this to 24 permissible combinations within the 11 precedence restrictions. Of these 6 are distinct and optimal combinations. The procedure is illustrated only for the first 2 station assignments. Similar steps when repeated gives a final result in Table 3-26. The total delay in the line is 3 minutes less than $c = 10$. Thus, we arrived at a minimal station balance with 4 stations.

Table 3-26. Summary of station assignments by Hoffman's procedure.

Station Number	1	2	3	4
Element	1, 3	2, 4, 7	5, 6	8, 9
Station Time	9	9	9	10
Delay at Each Station	1	1	1	0

F. Improvement on the Ranked Positional Weight Technique (Mansoor, 1964).

The additional rules and extensions, given by Mansoor (1964a, 1964b) improved the ranked positional weight technique originally suggested by Helgeson and Birnie (1961). Mansoor approached the problem with the objective of minimizing cycle time for a given number of work stations. This improved technique though offers optimal solutions involves various heuristics and takes larger computational time even for a smaller size problem.

Analysis of the Problem

Using the same terminology as used in Chapter II, the maximum possible number of work stations is limited by the maximum work element time (t_{max}) in the assembly process. i. e.,

$$k_{max} = \frac{T}{t_{max}} = \text{a whole number rounded down to the lower integer.}$$

e. g. If $T = 185$ and $t_{max} = 45$, then $k_{max} = \frac{185}{45} = 4.1 = 4$ (rounded down to the lower integer). However, k_{max} can theoretically be equal to n, the number of tasks in the precedence diagram. Assuming the cycle time as an integer, the smallest is one unit (e. g. 1 unit = 0.01 minute). The theoretical minimum cycle time for m work stations is given by

$$C_1 = T/m = p + (R/m)$$

where R is the remainder term and $mp + R = T$.

If $R \neq 0$, C_1 must be rounded upwards to the next integer. Perfect balance may be possible if $R = 0$, while it is impossible when $R \neq 0$. Thus at the first attempt in the procedure we use a cycle time, C_1, which has $(m-R)$ slack units if $R \neq 0$ while zero slack units if $R = 0$.

At the second attempt a new cycle time is given by

$$C_2 = C_1 + 1$$

which has $(2m-R)$ slack units if $R \neq 0$ and m slack units if $R = 0$. At the rth attempt, we use a cycle time

$$C_r = C_1 + (r-1)$$

which has $(r \cdot m - R)$ slack units if $R \neq 0$ and $(r-1)m$ slack units if $R = 0$. For example, when $T = 48$ and if it were possible to have 5 work stations, then the theoretical minimum cycle time $= C_1 = 48/5 = 9 + 3/5 = 10$ (rounded upwards) with 2 slack units. At the second attempt, $C_2 = C_1 + 1 = 10 + 1 = 11$ with 7 slack units.

Assignment Rules

The set of rules given by Helgeson and Birnie (1961) are reproduced here for easy reference with the additional instructions (Rules 1, 6 and 7) and the extensions (Rules 9, 10 and 11) given by Mansoor in his improved method. Minor modification was also suggested by Mansoor to Rule 5.

Rule 1. Begin by selecting the lowest cycle time corresponding to each of the number of work stations possible. Record the slack units available.

Rule 2. Select the work unit with the highest positional weight and assign it to the first work station.

Rule 3. Calculate the unassigned time for the work station by calculating the cumulative time of all work units assigned to the station and subtracting this sum from the cycle time.

Rule 4. Select the work unit with the next highest positional weight and attempt to assign it to the work station after making the following checks.

a. Check the list of already assigned work units. If the "immediate precedent" has not been assigned proceed to Rule 5.

b. Compare the work unit time with the unassigned time. If the work unit is less than the work station unassigned time, assign the work unit and recalculate the unassigned time. If the work unit is greater than the unassigned time, proceed to Rule 5.

Rule 5. Continue to select, check and assign if possible until one of the two conditions are met.

a. A combination is obtained when the remaining unassigned time is less than, or equal to the slack units available (proceed to Rule 8).

b. No unassigned work unit remains that can satisfy both the "precedence" and the "unassigned time requirements (proceed to Rule 6).

Rule 6. Cancel each assigned work unit in turn, starting with the one having the lowest positional weight (i. e., the last one assigned and eventually work back) and go through Rules 4 and 5 until either,

a. A combination is obtained where the remaining unassigned time is less than or equals the slack units available (proceed to Rule 8).

b. All combinations possible have unassigned times in excess of the slack units available so that no solution is possible (proceed to Rule 7).

Rule 7. Select a cycle time having one more suit and go through Rule 2 onward (Mansoor, 1964a, p. 76-77).

The following rules are the extensions made in the year by Mansoor (1964b).

Rule 8. Assign the unassigned work unit with the highest positional weight to the second work station, and proceed through the preceding rules in the same manner except when 6-b applies (proceed to Rule 9).

Rule 9. Rebalance the first work station with a new combination of work units (apply Rule 6-a) and make the following checks.

a. If rebalances are possible, renew attempt at balancing the second work station (Rule 8) and if successful, proceed to Rule 11; if unsuccessful, proceed to Rule 10.

b. If rebalances are not possible, proceed to Rule 10.

Rule 10. Select a cycle time having one more unit (i. e., Rule 7) and go through Rule 2 onwards, starting from the first work station. Repeat this procedure until both stations are balanced at a minimum cycle time (proceed to Rule 11).

Rule 11. Continue assigning work units to work stations until all work units have been assigned (extend Rules 9 and 10 where appropriate, to cover three or more stations). At that time a solution to the problem will have been found (Mansoor, 1964b, p. 323).

Example:

Consider the problem given in the prcedence diagram of Figure 3-1. We have

$$\sum t_i = 48 \text{ minutes and } t_{max} = 8 \text{ minutes.}$$

Hence,

$$8 \leq \text{cycle time} \leq 48$$

The maximum number of theoretical work stations,

$$k_{max} = \frac{\sum t_i}{t_{max}} = \frac{48}{8} = 6 .$$

The solution to the example is illustrated in Table 3-20 for the minimum cycle time of $c = 8$ and at the maximum number of work stations, $k_{max} = 6$. The terms "assign" and "reject" in column (9) and the "square around the unassigned time" in column (8) carry the same meaning as indicated in ranked positional weight technique of Helgeson and Birnie as explained earlier.

Table 3-27. Illustration of improvement over positional weight technique at $c = 8$ and $k = 6$.

Cycle Time (1)	Work Station (2)	Slack Units Available (3)	Work Unit (4)	Immediate Precedence (5)	Element Time (6)	Cumulative Station Time (7)	Unassigned Time (8)	Remarks (9)
8	1	0						
			u_1	-	4	4	4	assign
			u_2	u_1	2	6	$\boxed{2}$	assign
			u_6	u_1	6			reject > c
			u_3	u_2	5			reject >c
			u_7	u_6				reject, p
			u_4	u_3				reject, p
			u_5	u_2	5			reject >c
			u_8	u_7				reject, p
			u_9	u_4, u_5, u_8				reject p & goto Rule 6b

Note that all possible combinations have unassigned times in excess of slack units available (units) and do not result in a possible solution. We proceed to Rule 7.

Table 3-27 continued.

Cycle Time (1)	Work Station (2)	Slack Units Available (3)	Work Unit (4)	Immediate Precedence (5)	Element Time (6)	Cumulative Station Time (7)	Unassigned Time (8)	Remarks (9)
9	1	6	u_1	-	4	4	5	assign
(8+1)			u_2	u_1	2	6	$\boxed{3}$	assign
			u_6	u_1	6			reject > c
			u_3	u_2	5			reject > c
			u_7	u_6				reject, p
			u_4	u_3				reject, p
			u_5	u_2	5			reject >c
			u_8	u_7				reject, p
			u_9	u_4, u_5, u_8				reject, p & goto Rule 8
	3	0	u_3	u_2	5	5	$\boxed{4}$	assign
			u_7	u_6	8			reject, p
			u_4	u_3	7			reject >c
			u_5	u_2	5			reject >c
			u_7	u_6				reject, p
			u_4	u_3				reject, p
			u_5	u_2				reject >c
			u_8	u_7				reject, p
			u_9	u_4, u_5, u_8				reject, p & goto Rule 7
10	1	12	u_1	-	4	4	8	assign
(9+1)			u_2	u_1	2	6	$\boxed{4}$	assign
			u_6	u_2	6			reject >c
			u_3	u_6	5			reject > c

Table 3-27 continued.

Cycle Time (1)	Work Station (2)	Slack Units Available (3)	Work Unit (4)	Immediate Precedence (5)	Element Time (6)	Cumulative Station Time (7)	Unassigned Time (8)	Remarks (9)
10 (9+1) (continued)			u_8	u_7				reject, p
			u_9	u_4, u_5, u_8				reject, p & go to Rule 8
	2	3	u_6	u_1	6	6	$\boxed{4}$	assign

The rest of the tasks have to be rejected either to cycle time constraint or precedence violation; thus, leading us to Rule 8.

Cycle Time (1)	Work Station (2)	Slack Units Available (3)	Work Unit (4)	Immediate Precedence (5)	Element Time (6)	Cumulative Station Time (7)	Unassigned Time (8)	Remarks (9)
	3	4	u_3	u_2	5	5	5	assign
			u_7	u_6	8			reject $>c$
			u_4	u_3	7			reject $>c$
			u_5	u_2	5	10	$\boxed{0}$	assign & go to Rule 8
	4	4	u_7	u_6	8	8	$\boxed{2}$	assign

The rest of the tasks violate the cycle time or precedence restrictions. We go to Rule 8.

Cycle Time (1)	Work Station (2)	Slack Units Available (3)	Work Unit (4)	Immediate Precedence (5)	Element Time (6)	Cumulative Station Time (7)	Unassigned Time (8)	Remarks (9)
	5	2	u_4	u_3	7	7	3	assign
			u_8	u_7	3	10	$\boxed{0}$	assign & go to Rule 8
	6	2	u_9	u_4, u_5, u_8	8	8	$\boxed{2}$	assign & terminate

Thus the total slack units (12) are distributed in the six stations and this is a feasible minimum cycle time (c = 10) schedule for a maximum of six stations as summarized in Table 3-28.

Table 3-28. Summary of minimum cycle time ($c = 10$) schedule for a maximum number of work stations ($k = 6$).

Station Number	1	2	3	4	5	6
Tasks	u_1, u_2	u_6	u_3, u_5	u_7	u_4, u_8	u_9
Station Time	6	6	10	8	10	8
Station Idle Time	4	4	0	2	0	2

Balance delay at $c = 10$ and $k = 6$ is found to be 20%. Since the balance delay is very high, reject this schedule and attempt for a better schedule. We can now try for the next possible number of stations (i. e., one unit below maximum selected) which is equal to $c = 5$. Table 3-29 summarizes the results of the algorithm when the number of stations (m) $= 5$ and $c = [\frac{48}{5}] = 10$.

Table 3-29. Summary of minimum cycle time ($c = 10$) schedule for a maximum number of work stations equals 5.

Station Number	1	2	3	4	5
Tasks	u_1, u_6	u_2, u_7	u_3, u_5	u_4, u_8	u_9
Station Time	10	10	10	10	8
Station Idle Time	0	0	0	0	2

From Table 3-29, balance delay is 4%.

We can end the algorithm here or we can even attempt for another solution when $m = 4$ and then opening up on the cycle time i. e., when $m = 4$, $c_1 = \frac{48}{4} = 12$. Table 3-30 summarizes the results obtained at $m = 4$ and $c = 13$.[1]

Table 3-30. Summary of minimum cycle time schedule for a maximum number of work stations.

Station Number	1	2	3	4
Tasks	u_1, u_2, u_6	u_3, u_7	u_4, u_5	u_8, u_9
Station Time	12	13	12	11
Station Idle Time	1	0	1	2

From Table 3-30 balance delay $= \dfrac{\sum_{k=1}^{4} d_k \times 100}{4 \times 100} = \dfrac{4 \times 100}{4 \times 100} = 8.33\%$.

Thus the optimum result (minimum balance delay) is at $m = 5$ and $c = 10$ for which balance delay is 4%.

Various traditional methods described in this chapter are applied only to single-model assembly lines where performance times are assumed to be deterministic.

A large amount of research and computational efforts have been directed toward the solution of the ALB problem "perhaps more than the prevalence of assembly line warrants" (Riggs, 1970). Recent extensions of the single-deterministic models to variable elemental times and mixed-model problem are more significant in their contribution to practical assembly line problems. These extensions to be

---

[1] $c = c_1 + 1 = 12 + 1 = 13.$

treated as a system with the global optimization objective shall be described in the following chapters.

IV. SINGLE MODEL ALB EXTENSIONS

The traditional methods employed for balancing single-model assembly lines had the objective of minimizing only the direct labor cost and assumed deterministic work element times. However, in an actual assembly process, the work element times are seldom deterministic. An ALB procedure to consider this variability will be of practical value. In addition the inclusion of parallel stations and consideration of in-process inventory between the work stations will necessitate the formulation of a new model that has the objective of minimizing total production cost.

> The earlier proposed ALB solution methods either analytical or the heuristic have merits but none has the advantage of simplicity (Kilbridge and Wester, 1961, p. 292).

The method developed by Kilbridge and Wester (1961) may be simple but it lacks an algorithm. The only method developed for a single-model ALB, considering the variability of work elements was a heuristic model by Moodie (1964) consisting of two phases and many cumbersome heuristics, mostly in phase 2.

A systematic and simple ALB procedure based on three different positional weight criteria of a work element has been developed and programmed in FORTRAN IV. The procedure described in this chapter has both the advantages of being simple and algorithmic. It is easily adaptable as a manual method or for a computer application.

Also included in this chapter is the summary of the recent attempts to minimize either the total production cost or the perturbation in ALB problems.

A. Single-Model - Minimization of Direct Labor Cost for Both Deterministic and Normally Distributed Elemental Times - Heuristic Approaches

The research by Hicks and Young (1962) and by Walker (1959) has found that the work element times are actually independent and identically distributed normal variates. Using these results Moodie (1964, 1965) extended the single model ALB problem to normally distributed work element times. In the following sections Moodie's heuristic method and the Back Tracking Method of Assembly Line Balancing (BALB) developed in this thesis are explained with examples.

Successive Minimum Elemental Times and Interchangeability of Work Elements (Moodie and Young, 1965)

This is a two phase heuristic procedure used to balance an assembly line with either deterministic or normally distributed work element times. Smoothness Index (S. I.) is the criterion used to estimate the degree of balance among the station times:

$$S.\,I. = \sqrt{\sum_{k=1}^{m} (T_{max} - T_k)^2} \;;$$

where T_{max} = maximum station time

T_k = total time in the k-th station.

A perfect balance will have a S. I. of zero. Moodie's two relation matrices; viz., IP-matrix and IF-matrix are to be prepared before the application of the steps referred in this method.

Steps for the phase-1 of the procedure (station minimization): Phase 1 systematically selects the successive maximum element times available for assignment. This phase is programmed in FORTRAN IV and is used to compare the results of this method and BALB as explained later.

Step 1. By scanning IP-matrix all the zero rows are noted. The elemental times corresponding to each zero row are arranged in a decreasing order of magnitude if more than one element exist with zero rows. The element with the highest performance time is selected and assigned to the first station. Ties are resolved arbitrarily. The heuristics involved in this step are partially carried out by the ZEROFIND and DEORSEQ in the program listed in the Appendix.

Step 2. Note the element numbers in the row of the IF-matrix which corresponds to the assigned element and go to the rows of IP-matrix indicated by these numbers. Replace the assigned element's identification number with a zero.

This step is carried out by the subroutine ZIGZAG in the program.

Step 3. The steps 1 and 2 are repeated and the elements are assigned to the stations satisfying the restriction $t_{nax} \leq T_k \leq c$. When the IP-matrix contains all zeros the problem has been solved. These checks are made by the subroutine CHEK in the program.

Steps 1 through 3 usually will result in a minimal station balance satisfying the conditions $\sum_{k=1}^{m} (c - T_k) < c$ and $m^* = [\frac{\Sigma t_i}{c}]$. However, if this minimal station result is not possible for certain operating conditions, incrementing the cycle time will aid in obtaining the minimal station balance. Another important observation is that often the balance of phase 1, although requiring the minimum possible number of stations, will have unequal amounts of station times alloted to each of the stations and a higher smoothness index number. In such cases phase 2 of the heuristic method attempts to distribute this idle time equally to all stations through the mechanism of trades and transfers of elements as allowed by precedence restrictions between the stations.

Steps for phase 2 of the procedure (smoothening of station assignments and/or balance delay minimization). Phase 2 involves numerous heuristics to trade or transfer some of the elements from an overcrowded station to an under-utilized station. Interchangeability of work elements is attempted wherever possible.

Step 1. Determine both the largest and smallest station from the balance of phase 1.

Step 2. Call one-half the difference between these two values GOAL.

$$[\text{i. e.}, \ (T_{max} - T_{min})/(2) = \text{GOAL}]$$

Step 3. Determine all single elements in T_{max} which are less than twice the value of GOAL and will not violate precedence restrictions if transferred to T_{min}.

Step 4. Determine all possible trades of single elements from T_{max} for single elements from T_{min} such that the reduction in T_{max} and subsequent gain in T_{min} will be less than 2xGOAL.

Step 5. Carry out the trade or transfer indicated by the candidate with the smallest absolute difference between itself and GOAL [This step is carried out after ranking the available candidates in the order of their closeness to GOAL].

Step 6. If no trade or transfer is possible, between the largest and smallest stations, attempt trades and transfers between the ranked stations in the following order. One with n (n-th ranked station has greatest amount of idle time), n-1, ⋯ , 3, 2; 2 with n, n-1, ⋯ , 4, 3, ; 3 with n, n-1, ⋯ , 5, 4; 4 with n, n-1, ⋯ , 5, and so forth until the last comparison is between the station ranked n-1 and the station ranked n.

Step 7. If a trade or transfer is still not possible, drop the restrictions imposed by the value of GOAL and attempt, via the first six steps to get a trade or transfer which will not increase the value of any station beyond that of the original cycle time (Moodie and Young, 1965, p. 25).

Example:

Consider the problem of 9 work elements shown in Figure 4-1 where Σt_i is equal to 26 and the given value of $c = 10$ minutes.

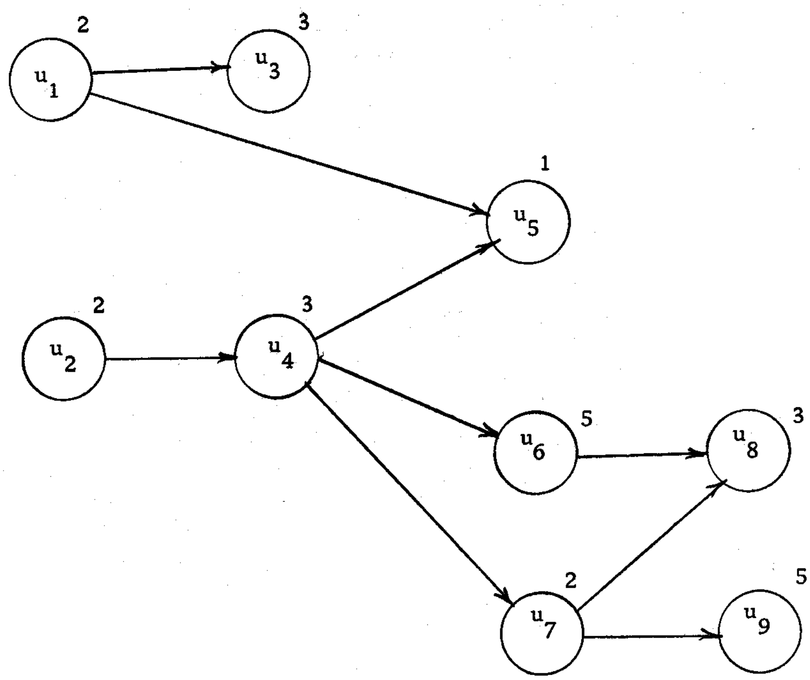


Figure 4-1. Precedence diagram of 9 work elements to illustrate Moodie's HALB.

Solution:

From the Figure 4.1, the IP-matrix and the IF-matrix are prepared and the data of the elemental times and their variances are included for each element as shown in Table 4-1. The Table 4-2 summarizes the results after the application of phase 1 procedure for a cycle time of 10 minutes and for deterministic elemental times. Since c = 10, the minimum number of stations $m^* = 3$ and this tallies with the result in Table 4-2.

Table 4-1. Precedence matrices and data of elemental times and variances to illustrate Moodie's HALB for a 9 element problem.

IP-Matrix				Elemental		IF-Matrix			
Element Number	Precedents			Time (t_i)	Variance (V_i)	Element Number	Followers		
1	0	0	0	2	0.5	1	3	5	0
2	0	0	0	2	0.8	2	4	0	0
3	1	0	0	3	1.1	3	0	0	0
4	2	0	0	3	1.2	4	5	6	7
5	1	4	0	1	0.2	5	0	0	0
6	4	0	0	5	1.8	6	8	0	0
7	4	0	0	2	0.2	7	8	9	0
8	6	7	0	3	1.0	8	0	0	0
9	7	0	0	5	1.5	9	0	0	0

Table 4-2. Summary of results of phase 1 procedure of Moodie's HALB for 9 work element problem without variance and for $c = 10$.

Station Number (k)	Element Number (i)	Element Time (t_i)	Cumulative Station Time (T_k)	Station Idle Time ($c - T_k$)
1	2	2	2	8
1	4	3	5	5
1	2	5	10	0^*
2	7	2	2	8
2	9	5	7	3
2	8	3	10	0^*
4	1	2	2	8
4	3	3	5	5
4	5	1	6	4^*

* denotes the station idle time at the end of grouping of tasks

The station assignments in Table 4-2 show that station 1 and 2 have a delay time of zero minutes and station 3 has a delay time of 4 minutes. The balance delay is equal to $(\sum_{k=1}^{m} d_k \times 100)/mc = \frac{400}{30} =$ 13.3% and S. I. is equal to $\sqrt{16.0} = 4.0$. Thus phase 1 gave a result with a minimal station balance but not necessarily the minimum balance delay or the lowest smoothness index. To decrease the balance delay and improve the smoothness of station assignments phase 2 is applied. Three applications of phase 2 decision rules reduced the smoothness index to 2.82. The trades and transfers consisted of the following:

i) Trade u_8 from station 2 for u_1 from station 3

ii) Trade u_8 from station 3 for u_9 from station 2

iii) Transfer u_5 from station 3 to station 2.

The alternate station balance with an improved S. I. of 2.82 after the application of phase 2 is shown in Table 3-4.

Table 4-3. Summary of results of phase 2 procedure of Moodie's HALB for 9 work element problem without variance and for c = 10.

Station Number	Elements Grouped	Station Time	Station Idle Time
1	u_2, u_4, u_6	10	0
2	u_1, u_7, u_5, u_8	8	2
3	u_3, u_9	8	2

Consideration of variability of work element times: Recalling Equation (2-7) of the variable elemental time ALB model we have,

$$\text{minimize } d = (mc - \sum_{k=1}^{m} T_k')/mc$$

where

$$T_k' = \overline{T}_k + Z\,\sigma(T_k)$$

and Z is the given standard normal deviate.

For T_k values to exceed c approximately 15% of the time Z = 1 (from table of areas under normal curve). When Z = 1, it implies that we can assume with 85% confidence, that all work assigned to an assembly line will be completed within the cycle time provided. The results of phase 1 solution of the 9 element problem with variance included are shown in Table 4-4, for a value of $Z = 1$.

Table 4-4. Summary of results of phase 1 procedure of Moodie's HALB for 9 work element problem with variance for $c = 10$ and $Z = 1$.

Station Number (k)	Element Number (i)	Element Time (t_i)	Variance (V_i)	Cumulative Station Time T_k	Station Idle Time ($c-T_k$)
1	2	2.0	0.8	2.9	7.1
1	4	3.0	1.2	6.4	3.6
1	1	2.0	0.5	8.6	1.4
1	5	1.0	0.2	9.6	0.4*
2	6	5.0	1.8	6.3	3.7
2	3	3.0	1.1	9.7	0.3*
3	7	2.0	0.2	2.4	7.6
3	9	5.0	1.5	8.3	1.7*
4	8	3.0	1.0	4.0	6.0*

* Indicates station idle time at the end of grouping of tasks

From Table 4-4, it can be observed that the same problem shown in Figure 4-1 requires 4 stations when the variability of the elemental times is introduced as compared to the original 3 station result under deterministic elemental time assumption. The balance delay with the inclusion of variance = (8.4 x 100)/(4 x 10) = 21%, and S. I. 6.25, based on the given value of C = 10. In Table 4-4, since $T_{max} = T_2 = 9.7$, balance delay in steady state condition is given by,

$$d = \{(m \times T_{max}) - \sum_{k=1}^{m} T_k\} \times 100/(m \times T_{max})$$

$$= \{(4 \times 9.7) - 31.6\} \times 100/(4 \times 9.7) = 18.55\% .$$

and

$$S.I. = (9.7-9.6)^2 + (9.7-9.7)^2 + (9.7-8.3)^2 + (9.7-40.)^2$$

$$= 5.87$$

Two applications of phase 2 decision rules reduced the smoothness index to 1.66. The heuristics consisted of the following:

i) Transfer u_3 from station 2 to station 3

ii) Transfer u_6 from station 1 to station 2

The results of phase 2 are summarized in Table 4-5.

Table 4-5. Summary of results of phase 2 procedure of Moodie's HALB with variance for $c = T_{max}$.

Station Number	Elements Grouped	Station Time	Station Idle Time
1	u_2, u_4, u_1	8.600	0.00
2	u_5, u_6	7.44	1.16
3	u_7, u_9	8.30	0.30
4	u_3, u_8	7.45	1.15

From Table 4-5, balance delay in steady state is given by

$$d = (2.61 \times 100)/(4 \times 8.6) = 7.5\%$$

and

$$S.I. = 1.66 .$$

This shows that there is about 350% improvement in the smoothness index after the application of phase 2 to the earlier result in Table 4-4. However, application of BALB with Linpows (Section B) will result in the same solution shown in Table 4-5 in one iteration and at a lesser time.

Back Tracing Method of Assembly Line Balancing (BALB)

Based on the different positional weights of a work element in a precedence diagram, a new method of balancing the assembly lines is developed in this thesis and has been programmed in FORTRAN IV. This method gave optimal solutions to most of the example problems considered by various authors. This method is more compact, more systematic and less time consuming than the earlier heuristic approaches. The computer program, *BALB, can accept data for both single and mixed-model problems for either with or without variability consideration.

This approach can be broadly classified under the Branch and Bound technique. However, since there is no lower bound, the term "back tracking" or its equivalent, the term "branch and exclude" (used by Lawler and Wood (1966)), would be more appropriate to describe this approach.

Reeve and Thomas (1967) have developed a heuristic branch and bound procedure for ALB problem but the criterion of a positional weight is not used in their method.

For *BALB, the positional weights were calculated in the basis of three criteria. They were (1) Linear positional weights (Linpow), (2) Logarithmic positional weights(Logpows), and (3) Square positional weights (Squarepows). By using the standard notation from Chapter II, the calculations for the various positional weights can be mathematically expressed as follows:

$$\text{Linpow of an element number } i = t_i + \sum_{x \in F_i} t_x$$

$$\text{Logpow of an element number } i = \log_2(t_i) + \log_2\{\sum_{x \in F_i} t_x\}$$

$$\text{Squarepow of an element number } i = t_i + \sqrt{\sum_{x \in F_i} t_x^2}$$

where F_i denotes all the elements following the element number i. For example, considering the Figure 4-2, the values of different positional weights are shown in the Table 4-6. A computer program, *ALLPOW, in FORTRAN-IV has been written to calculate these positional weights after inputing the data of the followers of an element and their corresponding times. The listing of the program is given in the Appendix.

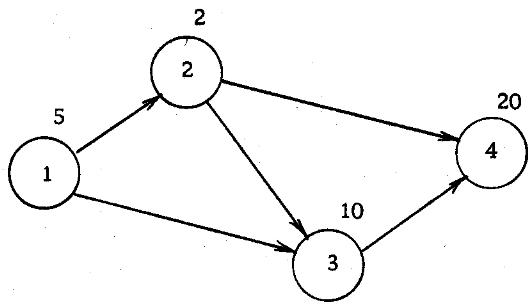


Figure 4-2. Precedence diagram of the example problem to illustrate the calculation of the positional weights.

Table 4-6. Positional weights for the example problem.

Task	Linpows	Logpows (bits)	Squarepows
1	37	7. 323	27. 45
2	32	5. 908	24. 361
3	30	7. 645	30. 000
4	20	4. 323	20. 000

It is clear from the Table 4-6 that the ordering of the tasks can change based on the criterion, i. e., in Linpows task 1 is ranked the highest while in Logpows and Squarepows, task 3 is ranked the highest. These three were studied to find the most effective *BALB procedure for finding optimal or near optimal solutions to the ALB problem. The steps for the *BALB are as follows:

<u>Step 1</u>. For a given cycle time c, calculate m^*, the minimum feasible number of work stations:

$$m^* = \frac{\sum_{i=1}^{n} t_i}{c} .$$

Knowing c and m^*, find the slack units available, i. e., s.u. = $m^* c - \sum_{i=1}^{n} t_i$.

<u>Step 2</u>. Consider one of the three criteria developed for positional weights. List the tasks available for assignment, and arrange them in a decreasing order of their positional weights. When the tasks are represented in a tree diagram, each node corresponds to a task and tasks are rearranged in the descending order of positional weights from right to left (Figure 4-4). Ties are broken arbitrarily.

<u>Step 3</u>. Select the task with the highest positional weight and branch out by successively selecting the task with the next positional weight in the list of available tasks. Ties are again broken arbitrarily. The branching procedure is carried out till either

a. the assignment to the first work station is complete subject to the cycle time constraint, or

b. the station time exceeds the given cycle time.

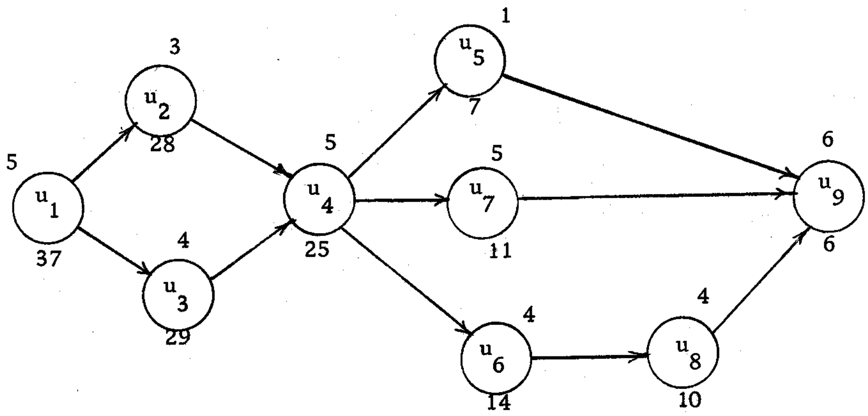
Step 4. When step 3-a occurs, the idle time of the assigned station is to be subtracted from the available slack units and the revised slack units should be noted. Then repeat the steps 2 to 3 till the assignments to all the m^* stations are complete with the slack units being distributed among all the stations. When step 3-b occurs, exclude this branch and back track to the node ranked next to the starting node in the earlier branch. Step 2 to 3 are repeated from that node.

Step 5. If the assignments to the m^* stations are not complete even after the application of steps 2 to 4, the cycle time can be incremented arbitrarily by one or more units and steps 1 to 4 are repeated till the minimum station balance is obtained.

These steps are illustrated in the following example.

Example:

To demonstrate its value, *BALB is applied to a 9 and 21 work element single model ALB problems. These are originally considered by Moodie (1964, 1965) and represented by Figures 3-11 and 4-3, respectively. The solution of these two problems are illustrated first by phase 1 and phase 2 of Moodie's HALB and then compared with solutions obtained by *BALB method using Linpows. The examples are illustrated only for deterministic times and can easily be extended to variable element times.



Legend:

a
u_i
b

u_i = work element i

a = element time (t_i) of element number i

b = linear positional weight of element i

Figure 4-3. Precedence diagram for the 9 work element problem to compare Moodie's HALB with *BALB.

Table 4-7. Moodie's dual procedure matrices and details of element times and Linpows.

Element Number	IP-Matrix			Elemental Time	Elemental Linpow	Element Number	IF-Matrix		
1	0	0	0	5	37	1	2	3	0
2	1	0	0	3	28	2	4	0	0
3	1	0	0	4	29	3	4	0	0
4	2	3	0	5	25	4	5	7	6
5	4	0	0	1	7	5	9	0	0
6	4	0	0	4	14	6	8	0	0
7	4	0	0	5	11	7	9	0	0
8	6	0	0	4	10	8	9	0	0
9	5	7	8	6	6	9	0	0	0

Application of phase 1 heuristics of Moodie (Section A-1 of this chapter) to the 9 element problem results in a four station balance in Table 4-7 for a given cycle time $c = 13$. Since $m^* = [(\Sigma t_i/c)] = [37/3] = 3$, the cycle time is incremented by 1 to 14 and phase 1 is repeated. Now a three station balance is obtained with a balance delay of 8.93% and a smoothness index of 3.16. These two iterations are summarized in Table 4-8.

Table 4-8. Summary of results of phase 1 of Moodie's HALB for the 9 work element problem without element variability at c = 13 .

Cycle Time	Station Number (k)	Element Number (i)	Cumulative Station Time (T_k)	Station Delay Time (d_k)
c=13	1	1, 3, 2	12	1
	2	4, 7, 5	11	2
	3	6, 8	8	5
	4	9	6	7
c=14	1	1, 3, 2	12	2
	2	4, 7, 6	14	0
	3	8, 5, 9	11	3

Table 4-9. Summary of results of phase 2 of Moodie's HALB for the 9 work element problem.

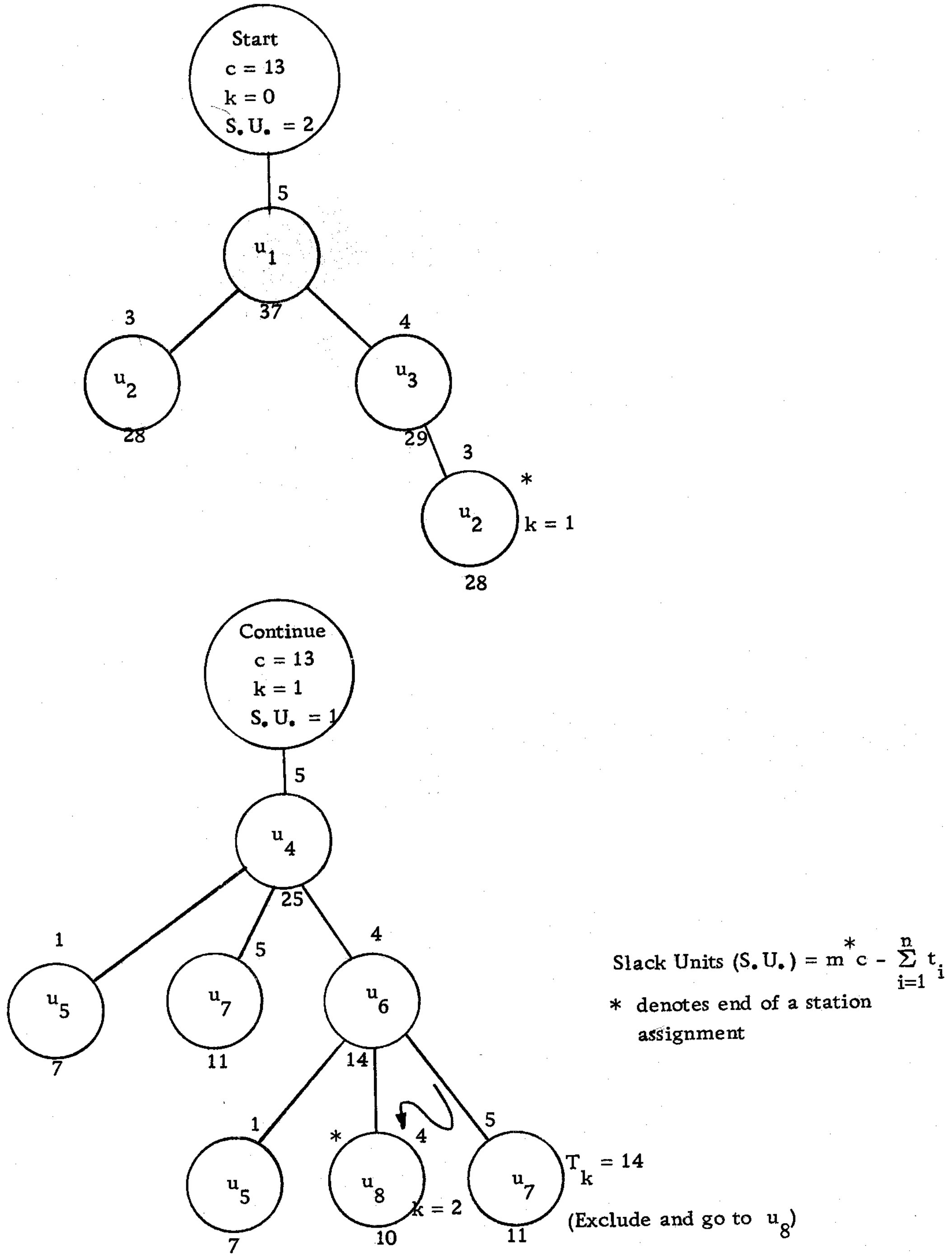
(k)	(i)	(T_k)	(d_k)
1	1, 3, 2	12	1
2	4, 8, 6	13	0
3	7, 5, 9	12	1

In order to reduce the balance delay and improve the smoothness among the stations, we apply phase 2 of Moodie's HALB to the results in Table 4-8. Since the maximum station time here is $T_2 = 14$ and the minimum is $T_3 = 11$, step 2 of Moodie's phase 2 yields GOAL = $(T_2 - T_3)/2 = 1.50$). A trade or transfer should be sought which increases the value of T_3 by a value as close to 1.50 as possible and thereby reducing T_2 by a like amount. Applying steps 3, 4, and 5 of phase 2, it can be seen that the best candidate consists of a trade of u_7 from T_2 for u_8 from T_3. Carrying out this trade reduces the total idle time to a minimum of 2 minutes as illustrated in Table 4-9. The balance delay is 5.13% and S. I. = 1.41, which indicates an improvement of over 200% on S. I.

The same solution as obtained by Moodie by his phase 1 and phase 2 heuristics is arrived at by *BALB more quickly and systematically without incrementing the cycle time, in Figure 4-4.

The effective use of *BALB when compared to Moodie's HALB can be demonstrated by working out the 21 element problem in Figure 3-11. Table 4-10 illustrates the two phases of Moodie's method applied to this problem and shows the improvement in the smoothness index with each trade or transfer during the application of phase 2. The same solution obtained by Moodie's heuristic method in 9 iterations can be obtained in one iteration by the application of *BALB (with Linpows) using the optimum cycle time of c = 21 minutes (Figure 4-5). The optimum cycle time is obtained from the given c = 25, as follows,

$$m^* = \frac{\sum_{i=1}^{n} t_i}{c} = [\frac{105}{5}] = 5 ;$$



Assign the rest of the tasks to the last station.

Figure 4-4. Illustration of BALB with Linpows for 9 element problem.

Table 4-10. Illustration of Moodie's HALB for 21 element problem.

Iteration Number	Station (k)		Station Time T_k	Remarks
1	1	1, 3, 4, 2	21	at c = 25
	2	5, 7, 21	24	phase 1:
	3	6, 8, 9, 13, 11, 10	23	S. I. =11. 26
	4	14, 12, 15, 18, 16, 19	19	
	5	17, 20	16	
2	1	1, 3, 4, 2	21	phase 2 starts
	2	5, 7, 21	24	max T_k= 24 = c
	3	6, 8, 9, 10, 13	22	S. I. =9. 0
	4	14, 12, 15, 18, 16, 19, 11	22	
	5	17, 20	16	
3	1	1, 3, 4, 2	21	
	2	5, 7	17	max T_k = 23 = c
	3	6, 8, 9, 10, 13	22	S. I. =6. 5
	4	14, 12, 15, 18, 16, 19, 11	22	
	5	17, 20, 21	23	
4	1	1, 3, 4, 2	21	
	2	5, 7, 14	20	max T_k = 23 = c
	3	6, 8, 9, 10, 13	22	S. I. =5. 5
	4	11, 12, 15, 16, 18, 19	19	
	5	17, 20, 21	23	
5	1	1, 3, 4, 2	21	
	2	5, 7, 14	20	max T_k = 23 = c
	3	6, 8, 9, 13	21	S. I. =5. 1
	4	11, 12, 15, 16, 18, 19, 10	20	
	5	17, 20, 21	23	

Table 4-10 continued.

Iteration Number	Station (k)	Element (i)	Station Time T_k	Remarks
6	1	1, 3, 4, 2	21	max T_k= 24 = c
	2	5, 7, 14	20	S. I. =7. 7
	3	6, 8, 9, 13	21	
	4	11, 12, 15, 21, 18, 19, 10	24	
	5	17, 20, 16	19	
7	1	1, 3, 4, 2	21	max T_k = 22 = c
	2	5, 7, 14	20	S. I. =2. 6
	3	6, 8, 9, 13	21	
	4	10, 11, 12, 15, 18, 21	22	
	5	17, 20, 16, 19	21	
8	1	1, 3, 4, 2	21	
	2	5, 7, 6	21	max T_k = 22 = c
	3	14, 8, 9, 13	20	S. I. =2. 6
	4	10, 11, 12, 15, 18, 21	22	
	5	17, 20, 16, 19	21	
9	1	1, 3, 4, 2	21	max T_k = 21 = c
	2	5, 7, 6	21	S. I. =0
	3	14, 8, 9, 13, 10	21	optimal solution is
	4	11, 12, 15, 18, 21	21	attained
	5	17, 20, 16, 19	21	after transferring 10 from T_4 to T_3

then the optimum cycle time

$$c^* = \frac{\sum_{i=1}^{n} t_i}{m^*} = [\frac{105}{5}] = 21 .$$

Since the elemental times are commonly given in integer values, the value of c^* is approximated to the least integer greater than or equal to the value within the bracket function.

The example problems used by other authors, i. e., the 8 elemental problem of Bowman (1960, p. 385), the 11 elemental problem of Jackson (1956, p. 264) and Ignall (1965, p. 244), the 18 elemental problem of Buffa (1961b, p. 421) and the 17 elemental problem of Thomopoulos (1969, p. 348) and the 19 elemental problem of Thomopoulos (1970, p. 599), have been worked out by *BALB using Linpows. It is found that in almost all the above cases with the sole exception of 17 work element problem (Figure 4-6) of Thomopoulos, the BALB approach yielded at least as good a balance as has been obtained by other methods (Chapter III) in less time and in two cases (problem in Figure 3-1 and Figure 4-1) better line balances as compared to other heuristic methods (e. g., Helgeson and Birnie's positional weight technique). For the mixed model problem[1] (Figure 4-6), *BALB yielded only a near-optimal solution at a cycle time $c = 464$, resulting

---

[1] For this problem cycle time equals the average work load per operator and we need 3 operators.

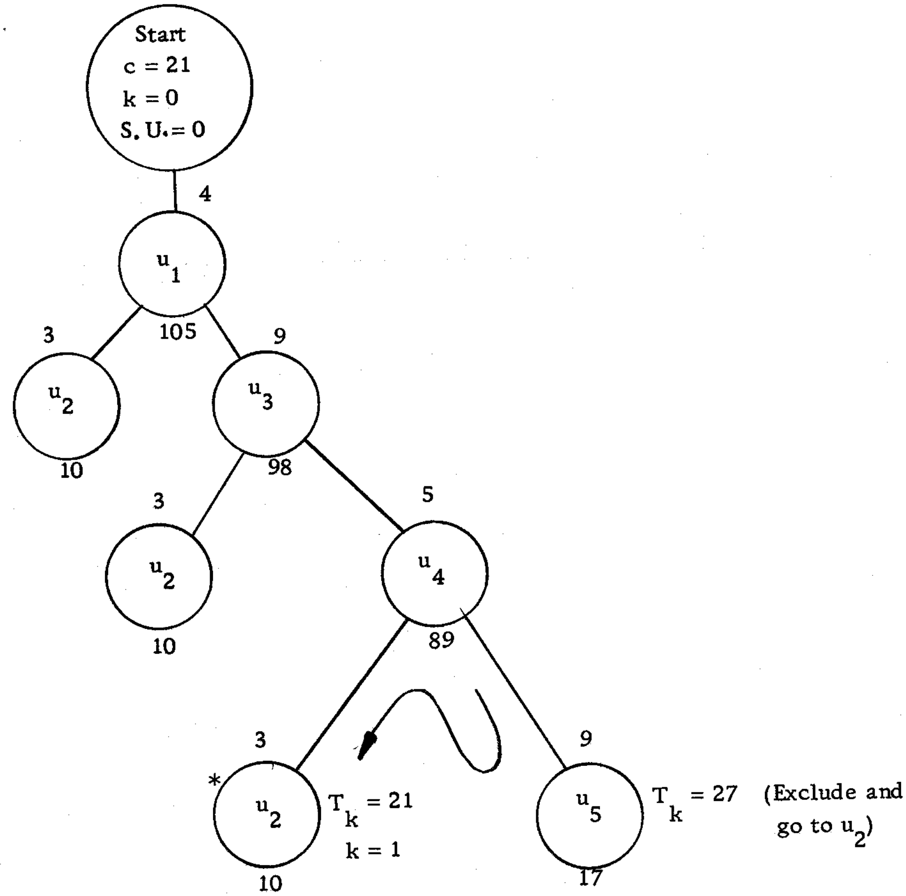


Figure 4-5. Illustration of BALB with Linpows for 21 element problem.

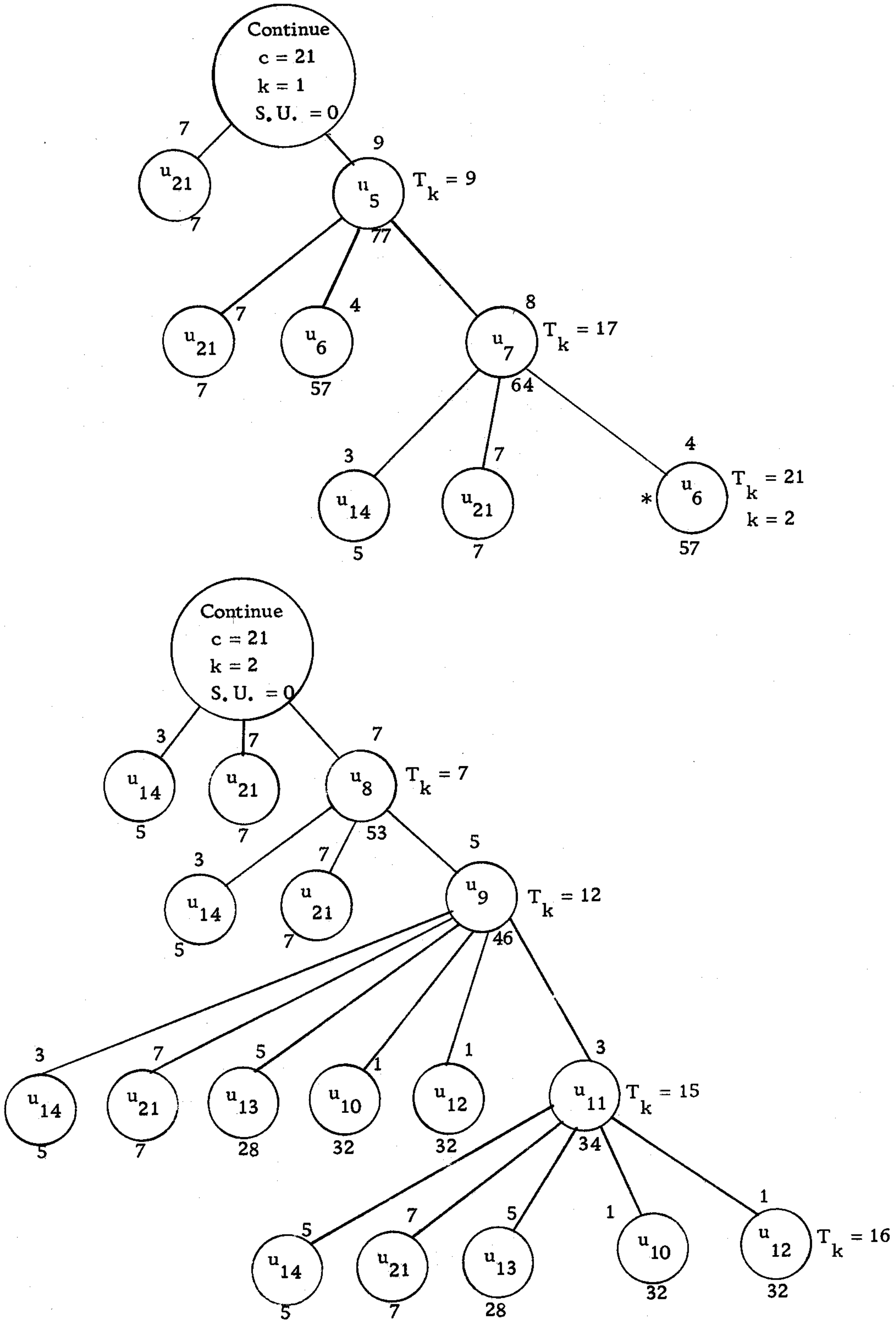
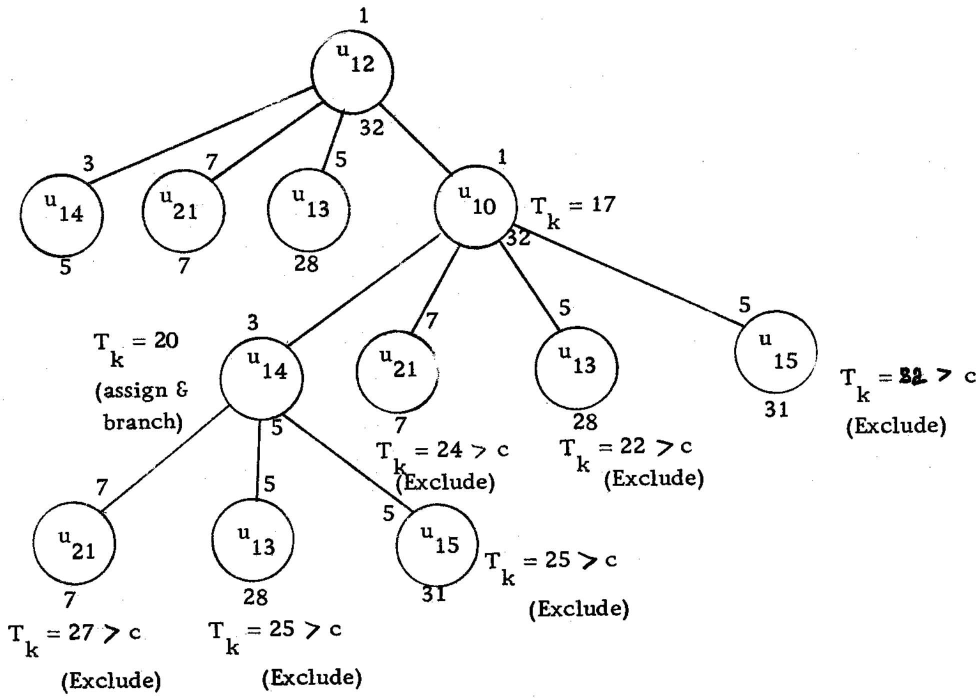


Figure 4-5 continued.



Since no more tasks are available for branching, back track (or bounding occurs here) to the main nodes and branch out from u_{21} as follows:

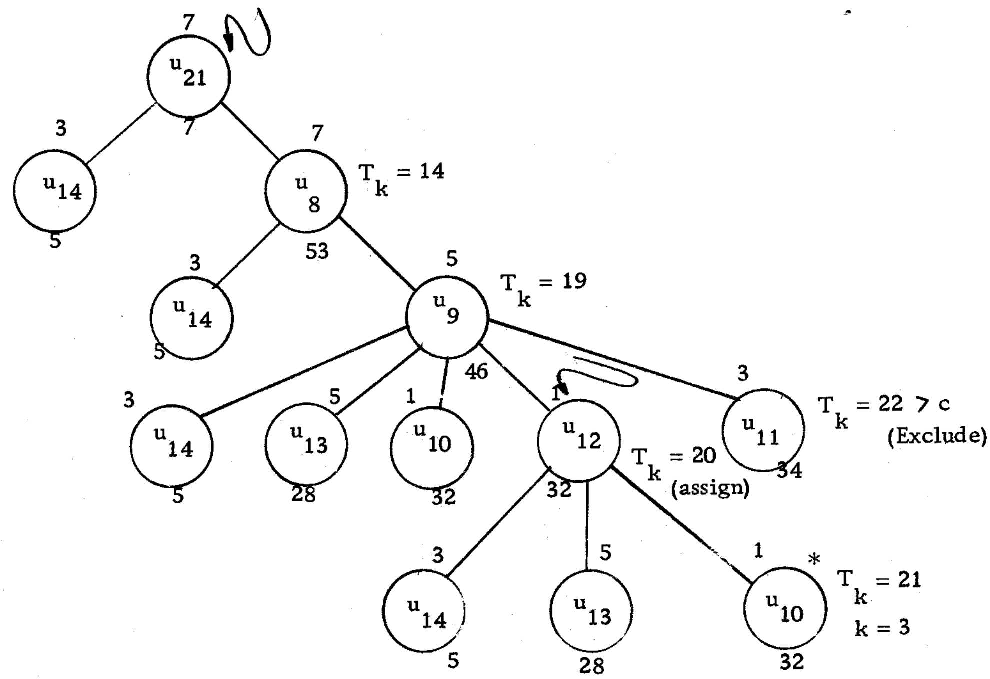
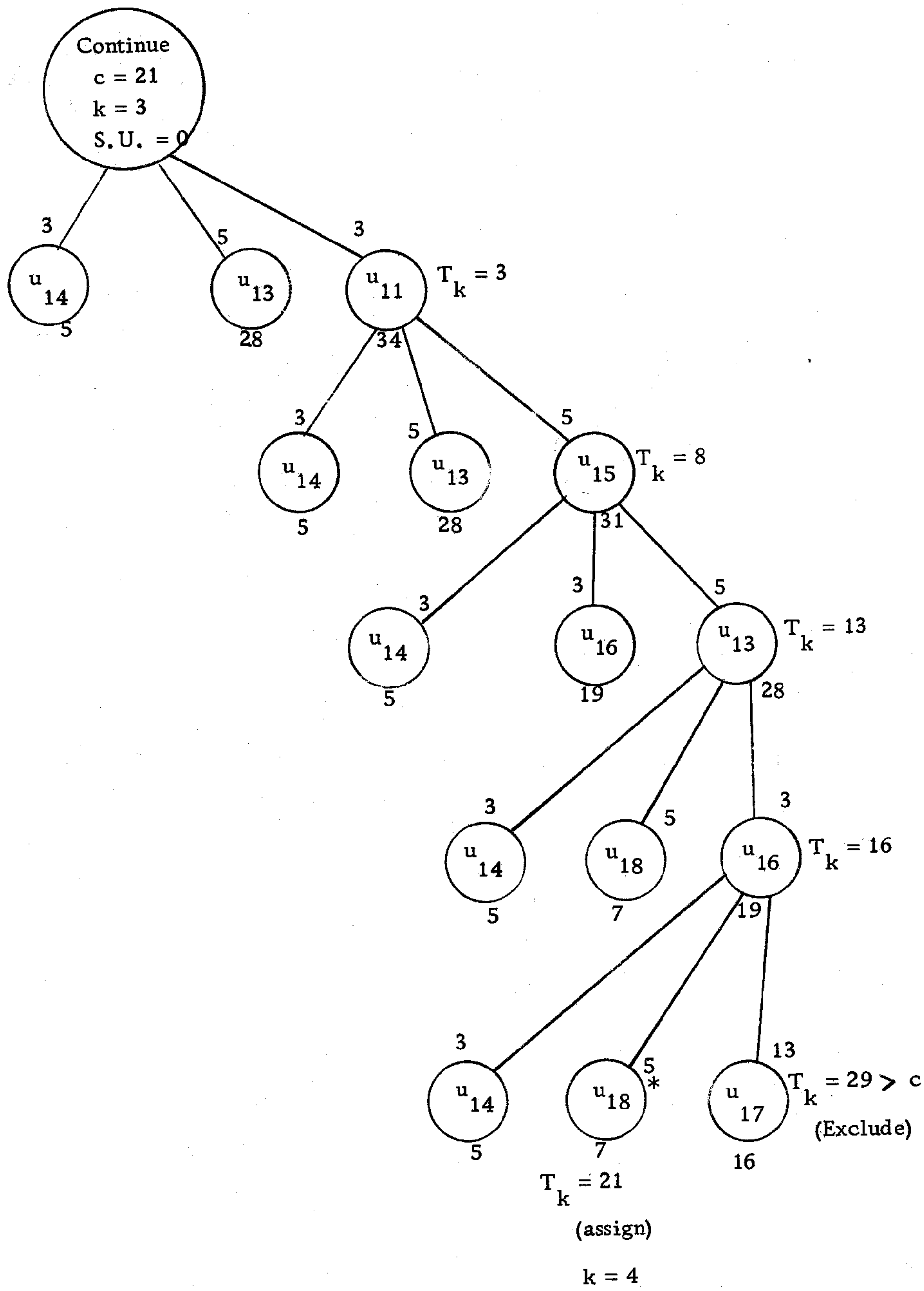
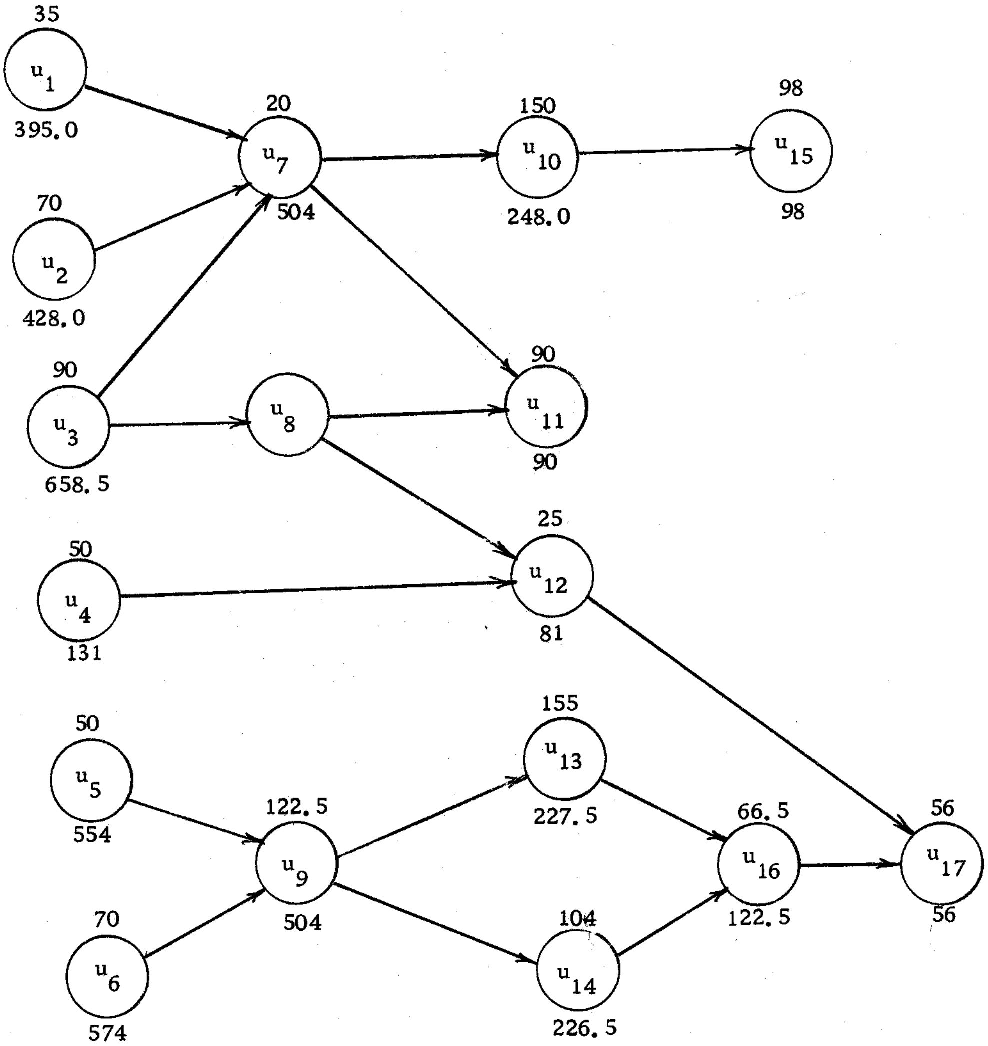


Figure 4-5 continued.



Assign the rest of the tasks to the last station.

Figure 4-5 end.



Legend:

a
u_i
b

u_i = Label for work element i

a = Lime of element i

b = Linear positional weight of element i

Figure 4-6. Combined precedence diagram for 17 work elements for a mixed-model line balancing problem (Source: Thomopoulos, 1968, p. 348).

in a 3 station balance, with a S. I. of 7. 399. All positional weight criteria were equally effective (refer Appendix for computer output using Linpows). The heuristic method of Kilbridge and Wester (1961) gives an optimal solution for this problem at a cycle time of c = 461. 5 (maximum station time)resulting in a 3 station balance with a S. I. of 2. 236. However, phase 1 procedure of Moodie (1965)'s HALB fails to give a 3 station balance till c has been incremented by units of one to 473 (starting from 460). The Moodie's 13 iterations were reduced by *BALB to only 5 iterations, with c incremented from 460 to 464 by units of one.

In general, all three criteria were found to be equally effective and superior to other known methods. However, the application of back tracking method (*BALB) with Logpows or Squarepows to the problem in Figure 3-1 resulted in a five station optimal balance without bounding to the main nodes with a CPU time of 14. 488 seconds (includes compilation time) while it failed to give optimal solution with Linpows (refer Appendix for computer output).

The ease of applying *BALB seems to warrant its being preferred over many other traditional ALB methods, which consist of enormous and cumbersome heuristics.

B. Minimization of Perturbation Costs with Deterministic Elemental Times (Cnossen, 1967)

The direct labor cost model of ALB is not applicable to many assembly lines where the cost of reallocating and reassigning of tasks on production lines outweighs the operating cost reduction through either minimizing the number of stations or the idle time in the line. Minimum perturbation ALB model attempts to minimize the relocation of relatively fixed facilities on the assembly line and to maximize the utilization of the existing skills of the assembly operators.

Cnossen (1967) developed a heuristic method, known as the Target Job Line Balancing (TJLB) techniques to minimize the perturbation costs. Minimum perturbation is an important constraint, for example, in automotive assembly line balancing because it is desirable to minimize both the operator learning effect of a completely new assignment and the costly rearrangement of fixed facilities and stocks. The perturbation costs are incurred for each of the rebalances which frequently occur during the change of models on the assembly line.

Summary of Target Job Line Balancing: The TJLB heuristic method requires three basic sets of information as input. They are:

i) Precedence relations between the jobs in the assembly line,

ii) Job time data for each job, and

iii) A list of the target jobs selected by the user.

The target job list as shown in Figure 4-7 is a unique input to provide

Target Jobs: 6, 16, 19, 36, 41, 40

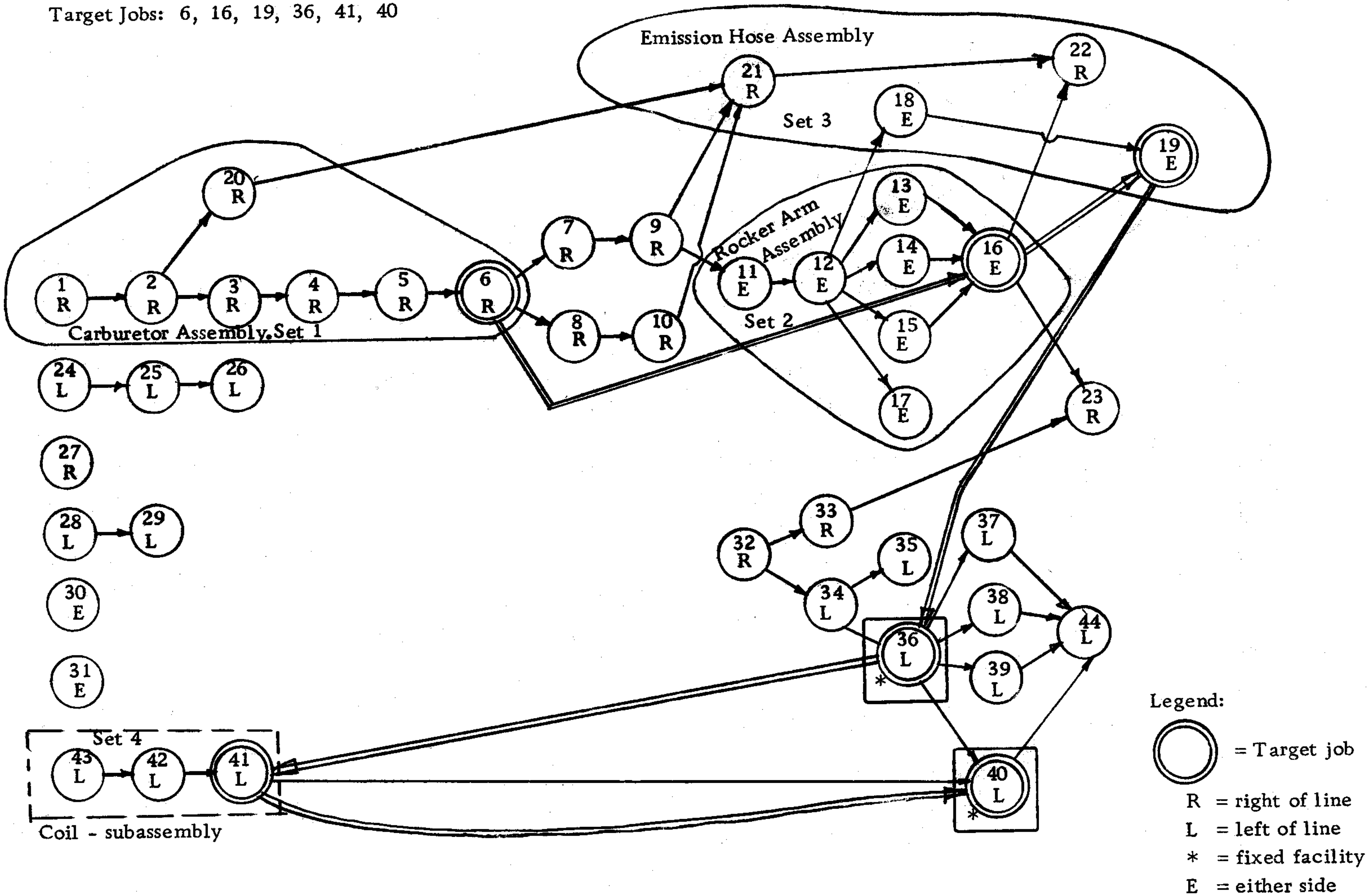


Figure 4-7. Precedence diagram of an engine dress up area in an automobile company (Source: Cnossen, 1967).

the user with a considerable amount of control over the order of the assembly assignment sequence. The target jobs are selected on the basis of the following criteria:

i) The last job in each job set, for example, jobs 6, 16, 19 and 41 in Figure 4-7.

ii) Jobs which require fixed tools, immovable facilities or stock supplies, for example, jobs 36 and 40 in Figure 4-7.

When target jobs have been selected for the entire assembly process, they are placed in the target job list in the desired sequence of their overall performance. The arrows joining the elements 6, 16, 19, 36, 41 and 40 in Figure 4-7 indicate this sequence.

Target Job Line Balancing Assignment Routines: The TJLB heuristics use two assignment routines. The first assignment routine generates the operator's basic assignment by systematically assigning the target jobs and their predecessors. This is called The Basic Assignment Routine as shown in Figure 4-8. When the first routine cannot assign a job to an operator because of balancing restrictions, a second assignment routine called The Close-Out Routine as shown in Figure 4-9, is used to complete the operator's assignment. This technique was implemented in Ford Motor Company in the year 1967 (Cnossen, 1967).

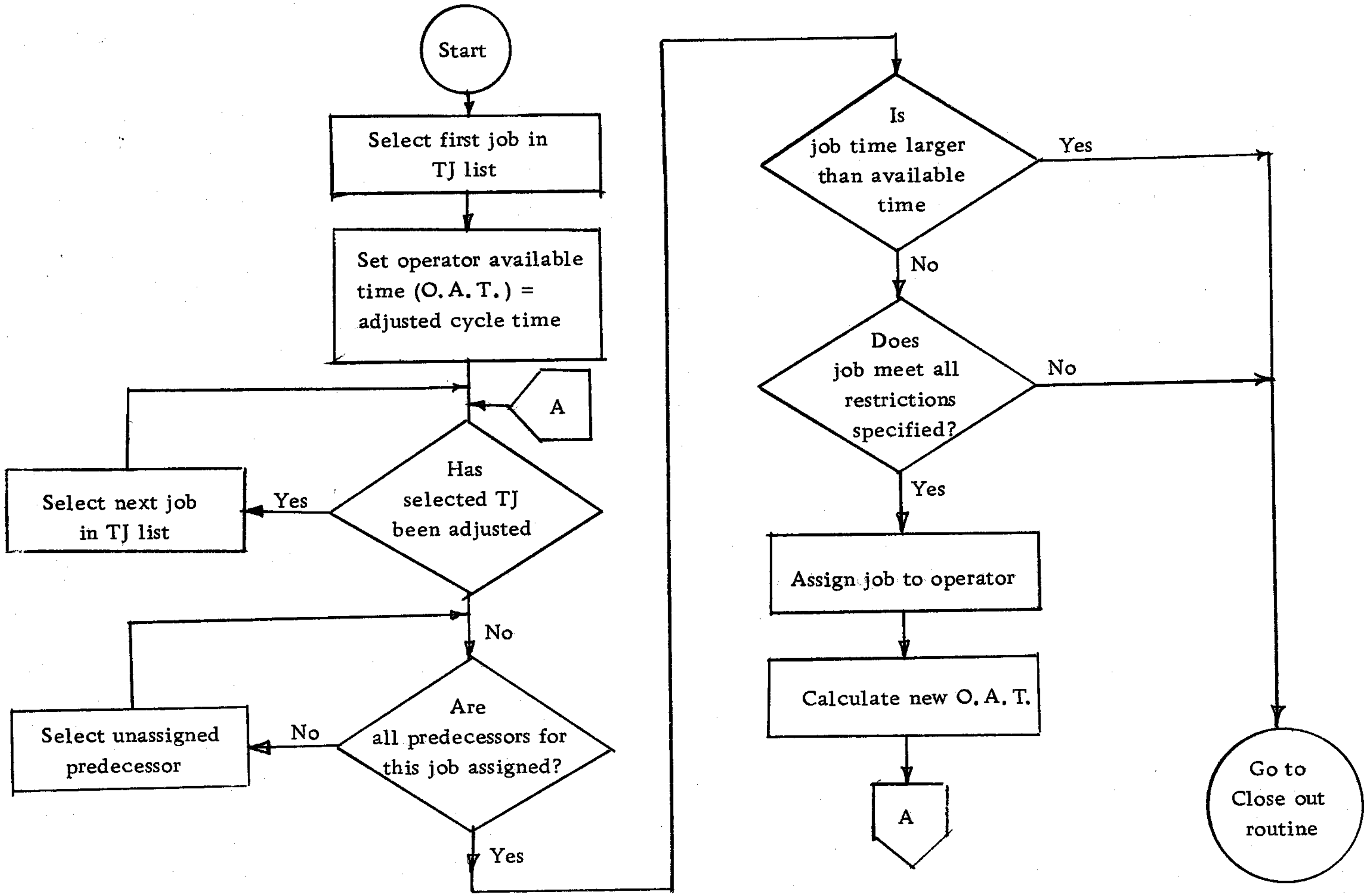


Figure 4-8. Flow chart of basic assignment routine illustrating target job line balancing (Source: Cnossen, 1967).

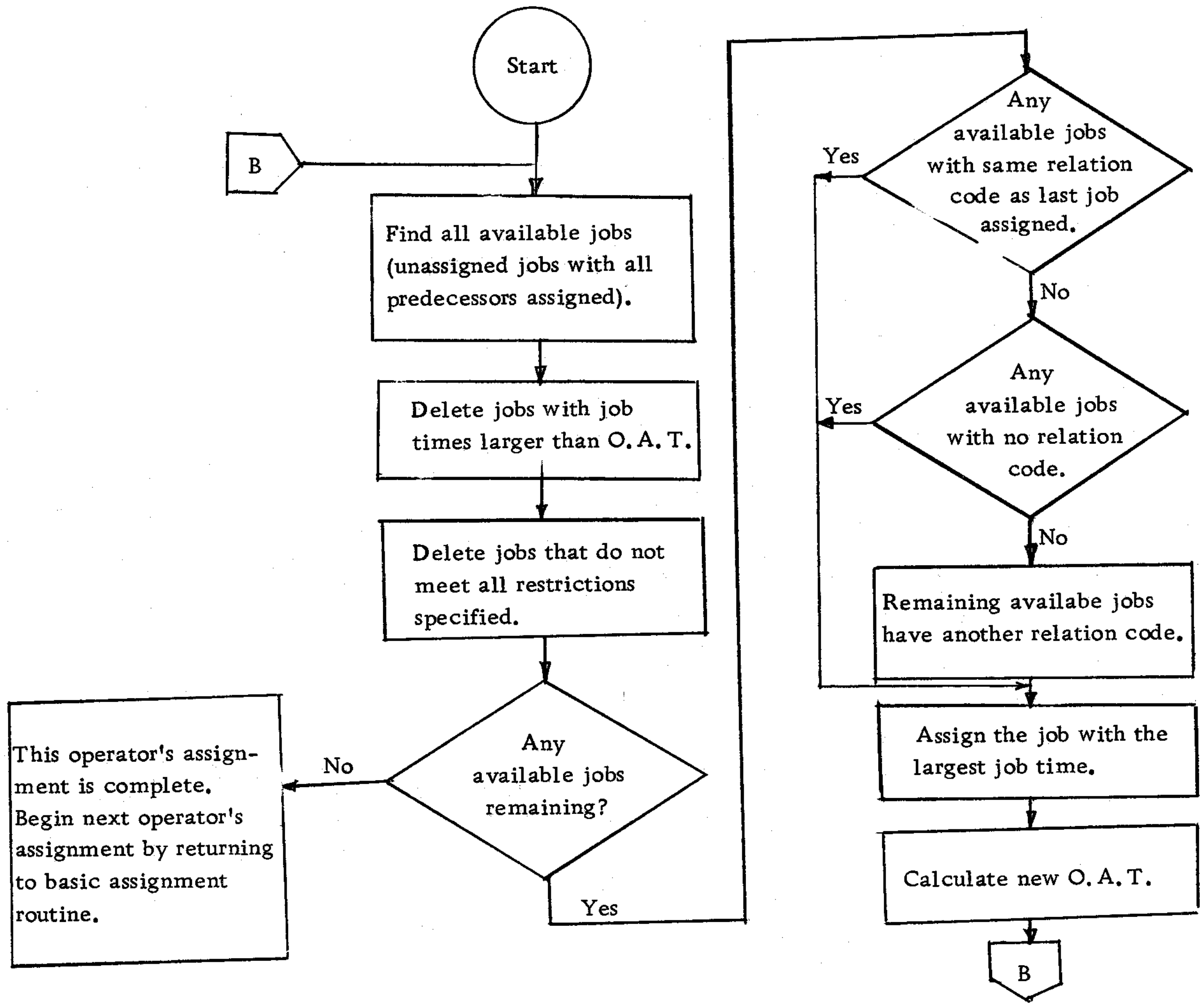


Figure 4.9. Flow chart of close-out routine (Source: Cnossen, 1967).

C. Single-Model Minimization of Total Production Cost with Random Work Element Times (Freeman, D. R., 1968)

> Although searching for solution techniques for the deterministic problem is worthwhile, a broader and more general problem exists and is receiving for too little attention (Freeman, D. R., 1968, p. 231).

If the work element time is considered a random variable, the work station time consisting of several work elements must also be taken as a random variable. In such a case, the time variable between successive items coming off the end of the conveyor line will also be random. This can then be viewed as the stochastic interdeparture time from a series queue. In a deterministic case, this interdeparture time is equal to the maximum station time on the line. Several studies, e. g., Hillier and Boling, 1966, Barten Kenneth, 1962, Buffa, E. S., 1961, Freeman, M. C., 1964, indicate that inventory between stations can indeed improve the output rate from the assembly line where the elemental times are random variables.

General Model: Here the objective is to minimize total production cost per unit subject to the following constraints:

i) Each work element is assigned to at least one work station

ii) The precedence constraints are satisfied

iii) The average output rate attained is at least as great as the desired output rate.

Let

Y = time between successive items coming off the line, a random variable

$\overline{Y}$ = expected value of Y

Q = quantity to be produced in period T

c = desired cycle time = T/Q

The objective function assumed by Freeman was of the form

Total cost/unit = Labor cost/unit + Inventory cost/unit + Facility cost/unit + Penalty cost/unit

Labor Cost (LC): This cost is a function of the number of stations in a balance, the mean output rate and the assignment of elements to stations (if elements requiring special skills demand higher rates, the assignment of these to stations may necessitate different labor rates at different stations).

Inventory Cost (IC): Allowing inventory build up between stations to "buffer" the line results in costs by increasing average inventory in process and additional costs of storage.

Facility Cost (FC): This reflects the penalty associated with duplicating facilities to permit parallel operations in the line.

Penalty Cost (PC): [Due to the random nature of the interdeparture time (in variable elements case), the desired production rate may not be achieved. Underproduction leads to overtime costs, while over-production results in inventory costs. Thus a penalty is assumed to penalize the system for failure to meet the desired output rate] (Freeman, D. R., 1968, p. 232).

The solution of the above model under very general conditions was discussed by Freeman (1967) in his Ph. D. dessertation. A general total cost equation without allowing parallel stations and with identical labor rates at each station is given by:

$$\text{Total cost (TC)} = LC + IC + PC \ .$$

Assuming the conveyor line is balanced with m stations, the total cost equation reduces to:

$$TC = mC_L\overline{Y} + C_f\overline{Y}\sum_{i=2}^{m} f_i + C_p(c - \overline{Y})^2 \ ;$$

where

C_L = labor cost in dollars per unit of time

C_p = penalty cost for failure to meet the desired output rate (in dollars)

C_f = cost of providing space for a unit of float in dollars per unit of time

f_i = provision for float before station i ; for $i = 2, 3, \cdots, m$ (station 1 is assumed to be preceded by an infinite bank).

In the total cost equation the term $(c - \overline{Y})^2$ denotes the assumption of a quadratic loss for the penalty term. The mean output interval $\overline{Y}$ is a function of the station times T_k , $k = 1, 2, 3, \cdots, m$, the inventory allowance between stations $f_2, f_3, \cdots, f_m$, and the nature of the parameters of the density function on T_k .

The behavior of $\overline{Y}$ was studied by Freeman (1968) using simulation (in ALGOL) for perfectly balanced stations each having times normally distributed with parameters μ and σ. By varying σ (μ/σ holding μ constant) one can study the effects of increased variability of station times. However for general purposes the interdeparture time $\overline{Y}$ was assumed as the sum of productive time plus delay caused by blocking effects, i.e.,

$$\overline{Y} = \mu(1 + \Delta) = \mu + \mu\Delta$$

where Δ is the fractional delay. Δ would be equal to zero in the deterministic case where all station times are assumed equal to μ. A theoretical expression based on the ratio μ/σ to calculate the values of Δ was arrived by Freeman (1968, p. 233). The determination of $\overline{Y}$ allows us to arrive at the total cost (TC) using this formulation. Freeman (1968) contends that more work need to be done in the total cost ALB model with the major focus on improving the predictive expression of $\overline{Y}$ and the output rate $1/\overline{Y}$ and on emperical studies to devise a generalized objective function.

V. MIXED-MODEL ALB EXTENSIONS

The need for a mixed-model assembly line occurs in a plant having several basic models of the same general product. Traditionally each model could be produced intermittently in batches and kept in finished goods inventory. Instead the company may decide to mix the product models on the same conveyor line. When several models are produced on the same conveyor line, it is commonly called a mixed-model, a mixed-product or a multi-model assembly line. Automobile industry, where the models differ, for example, by color, the number of doors, the wheel base, or the type of engine, is a classic example of this type of production. This situation can also be observed in television, home appliance and farm equipment industries.

The advantages of mixed-model assembly are numerous:

i) it provides a continuous flow of each model,

ii) it reduces finished goods inventory

iii) it eliminates line changeover, and

iv) it provides greater flexibility in production.

However, mixed-model lines do present some serious problems such as sequencing of models and scheduling of parts for various sub-assemblies. Sahgal (1970) pointed out that the merits of mixed-model lines are derived at the cost of efficiency and it is not uncommon to find mixed-model lines where productivity is only 70% of capacity.

According to Wester and Kilbridge (1964), the efficient utilization of mixed-model lines requires the optimum solution of two separate but related problems. They are:

i) the division of work between the operators, and

ii) the sequencing of models.

Line balancing is concerned with the former problem, which involves apportioning the elements of assembly work for each model among the operators on the line as evenly and compactly as possible. The latter problem involves in determining the optimum ordering in the flow of products to minimize the total inefficiency costs.

Grant (1962) was reported to be the first person to propose a computerized solution to the mixed-model ALB problem. The fundamental concepts and the related problems of mixed-model lines were presented by Thomopoulos (1966, 1967, 1969, 1970) and by Lehman (1969).

A description of the mixed-model balancing and sequencing procedures and the extensions of mixed-model line balancing for variable work elements is made in the following pages.

A. Minimization of Direct Labor Costs with Deterministic Work Element Times

The traditional approaches considered the determinism in the performance times of an operator in minimizing direct labor costs.

Two criteria namely cycle time and production shift time were used in balancing the mixed-model lines.

Cycle Time Basis

The first mathematical formulation for a mixed-model ALB problem was presented by Roberts and Villa (1970). They developed two models namely, integer programming model and shortest route model.

Problem Statement. Given J models each with its own precedence constraints, assign the work elements to work stations so that:

i) each work element is assigned to exactly one work station

ii) the number of stations is the same for all models

iii) the technological restrictions are satisfied

iv) the work content of any station for any given model does not exceed the cycle time, i.e.,

$$T_{kj} \leq c \qquad \begin{array}{l} j = 1, 2, \cdots, J \\ k = 1, 2, \cdots, m \end{array} \tag{5.1}$$

where T_{kj} is the work content of station k for model j and c is the cycle time

v) total idle time in all the stations is minimized, i.e.,

$$\text{minimize } D = \sum_{j=1}^{J} \sum_{k=1}^{m} N_j(c - T_{kj}) \tag{5.2}$$

where N_j is the number of units of model j to be assembled.

a. Integer (zero-one) programming model: Let

$$X_{kij} = \begin{cases} 1 & \text{if task } i \text{ is assigned to station } k \\ 0 & \text{otherwise} \end{cases}$$

t_{ij} = the time to perform the task i of model j

I_j = set of tasks of model j

For a given work content of a model and a given value of the cycle time, there is a minimum number of stations which is absolutely required for the assembly of that model. Let m_j^* denote the minimum number of stations for model j, where

$$m_j^* = [(\sum_{i \in I_j} t_{ij})/c]$$

where $[x]$ implies the smallest integer $\geq x$. Assume for the present that the value of c and the work content of the models are such that

$$m_j^* = m^*, \qquad j = 1, 2, \cdots, J \qquad (5.3)$$

The objective is to minimize the number of stations. Since it is known that the absolute minimum number of stations is m^*, the problem can be viewed as minimizing the work content of the stations that are allowed above the number m^*.

This must be done in such a way that, if the work content of a station k_1 $(k_1 > m^*)$ is zero, then, the work content of a station $k > k_1$ must also be zero and thus only k_1 stations will be needed. This implies that we should assign as much of the work content as possible to the first m^* stations and as little as possible to the stations after m^*. Hence the Equation (5. 2) can be modified as:

$$\text{minimize } D' = \sum_{k=m^*+1}^{M} W_k \sum_{j=1}^{J} N_j \sum_{i=1}^{I_j} X_{kij} t_{ij} \qquad (5.\ 4)$$

The variable W_k makes an assignment to a station k_2 less favorable than the same assignment to station k_1 when $k_1 < k_2$. This requires W_k to be an increasing sequence of positive number (1, 2, 4, 6, 16, etc) satisfying the relation,

$$W_k = 2^{k - (m^*+1)} \; ; \; k = m^*+1, \cdots, M \; .$$

where M = maximum possible number of work stations = $\max_j \{I_j\}$. The solution of (5. 4) is constrained by the restrictions mentioned earlier in this section. The mathematical representation of these restrictions are discussed in detail by Roberts and Villa (1970).

However, the large number of variables and constraints makes this formulation a difficult undertaking even for problems of modest size.

b. Shortest route model: The shortest route model of Gutjahr and Nemhauser (1964) for single model ALB problem can be extended to mixed-model ALB problem with considerable increase in computations. For the mixed-model case the shortest route model consists of developing a finite directed network similar to Figure 3-5 in which the arcs represent stations in the assembly line and the nodes (states) correspond to the feasible groupings of elements from each product. These groupsing are considered as possible station assignments. The arc lengths in the network denote the idle times within the stations. The optimization procedure consists of selecting the shortest path in the network from the starting node to the ending node in the network (this is equivalent to finding the minimum number of arcs).

Generation of States or Nodes. A state consists of an ordered set of work elements which form feasible (consistent with precedence requirements) first station assignments (without regard to cycle time). All states for each product, except for the empty set ϕ , would be generated as though that product were the only one to be assembled. This results in a set S_j of states for each product j . The set of all states for the multiproduct problem is then obtained from the Cartesian product of all S_j , i. e.,

$$S = \{S_1 \otimes S_2 \otimes S_3 \otimes \cdots S_j, \phi\} . \tag{5.5}$$

An element in S , say T_1 , is given by

$$T_1 = \{S_{11}, S_{21}, \cdots, S_{J1}\} , \tag{5.6}$$

where S_{j1} is an element of S_j for $j = 1, \cdots, J$.

If positional restrictions exist the states violating these restrictions are discarded. (The states generated become the nodes of the network.) The nodes are constructed starting from the empty state 0 and ended in the destination node r which consists of all elements for all models.

Construction of Directed Arcs. There is a directed arc from node (state) u to v if and only if

$$S_j u \subset S_j v \quad , \quad j = 1, 2, \cdots, J \tag{5.7}$$

and

$$\sum_{i \in \{S_j v - S_j u)} t_{ij} \leq c \;. \quad j = 1, 2, \cdots, J \tag{5.8}$$

Equations (5. 7) and (5. 8) allow us to make a network for a mixed-model case and it is sufficient to find the shortest path from 0 to r to arrive at the minimal station balance. However, it should be noted that when $J = 1$, the original Gutjahr and Nemhauser algorithm is obtained.

Example:

To illustrate the shortest route formulation consider Figure 5-1 showing the precedence diagrams for two models. Here we shall illustrate how the Gutjahr and Nemhauser's algorithm explained in Chapter III can be used separately for the individual models to compare the results with the solution obtained by Roberts and Villa when the models are considered together.

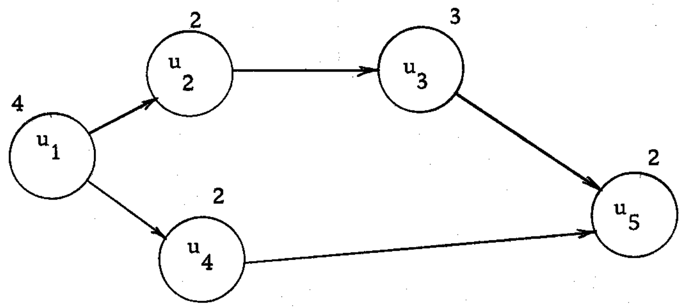


Figure 5-1a. Precedence diagram for model 1.

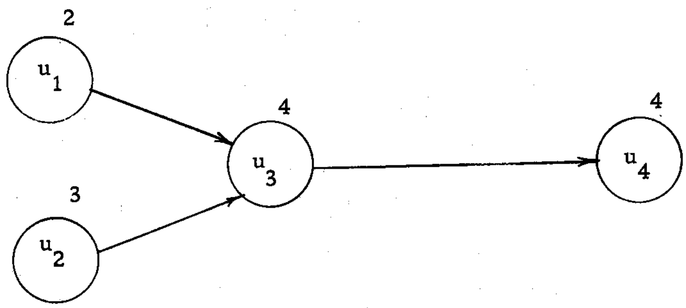


Figure 5-1b. Precedence diagram for model 2.

Figure 5-1. Precedence diagrams for model 1 and model 2 to illustrate shortest route model for mixed-model case.

Recalling the rules explained in Chapter III, the states (nodes) generated for the model 1 are shown in Table 5-1.

Table 5-1. Generation of states for model 1.

Stage	Marked Elements	State Number	State Elements	State Time	Unmarked Elements
0	NIL	0	ϕ	0	1
1	1	1	1	4	2, 4
2	2, 4	2	1, 2	6	3
		3	1, 4	6	
		4	1, 2, 4	8	
3	3	5	1, 2, 3	9	5
		6*	1, 4, 3		
		6	1, 2, 4, 3	11	
4	5	7*	1, 2, 3, 5		
		7	1, 2, 4, 3, 5	13	NIL

* Denotes rejection of a state due to precedence violation.

Assuming a cycle time $c = 5$, a network is constructed for model 1 as in Figure 5-2. From the Figure 5-2, there are two shortest routes (0-1-4-7 and 0-1-5-7) from the destination node 7 back to the origination node 0. Both solutions are equally good when the objective is to minimize idle time for the model 1. The summary of station assignments for both routes is in Table 5-2.

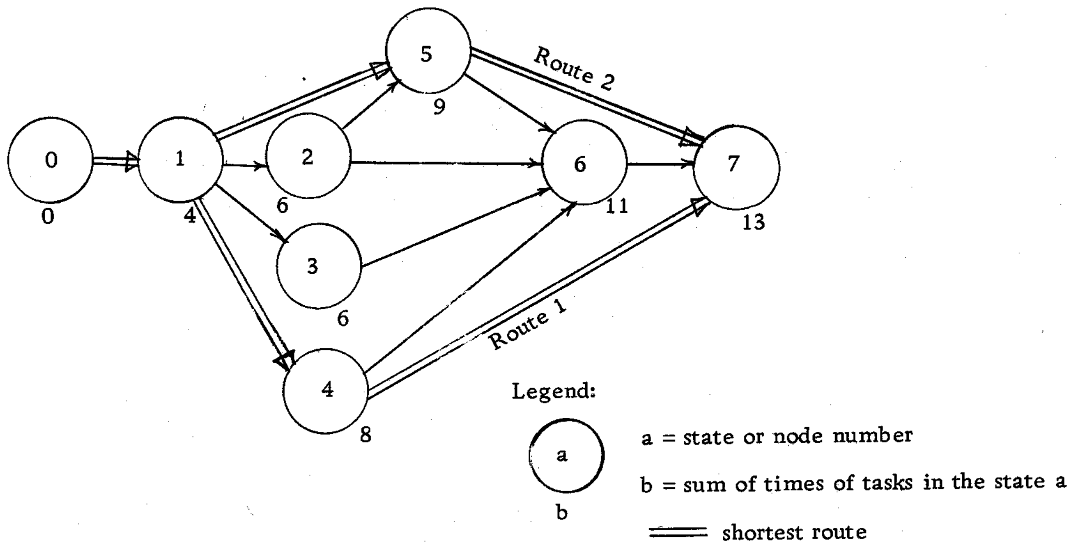


Figure 5-2. Network representation of nodes for model 1.

Table 5-2. Summary of station assignments for the two shortest routes shown in Figure 5-2 for model 1 at c = 5.

Number	Nodes Spanned by Arc		Work Elements Spanned by Arc		Station Time		Station Idle Time	
	Route 1	Route 2	Route 1	Route 2	Route 1	Route 2	Route 1	Route 2
1	0, 1	0, 1	u_1	u_1	4	4	1	1
2	1, 4	1, 5	u_2, u_4	u_2, u_3	4	5	1	0
3	4, 7	5, 7	u_3, u_5	u_4, u_5	5	4	0	1

Considering the model 2 and using similar procedure we will have Table 5-3 for the generation of the states. Figure 5-3 denotes the network.

Table 5-3. Generation of states for model 2.

Stage	Marked Elements	State Number	State Elements	State Time	Unmarked Elements
0	NIL	0	ϕ	0	1
1	1, 2	1	1	2	3
		2	2	3	
		3	1, 2	5	
2	3	4*	1, 3		4
		4*	2, 3		
		4	1, 2, 3	9	
3	4	5	1, 2, 3, 4	13	NIL

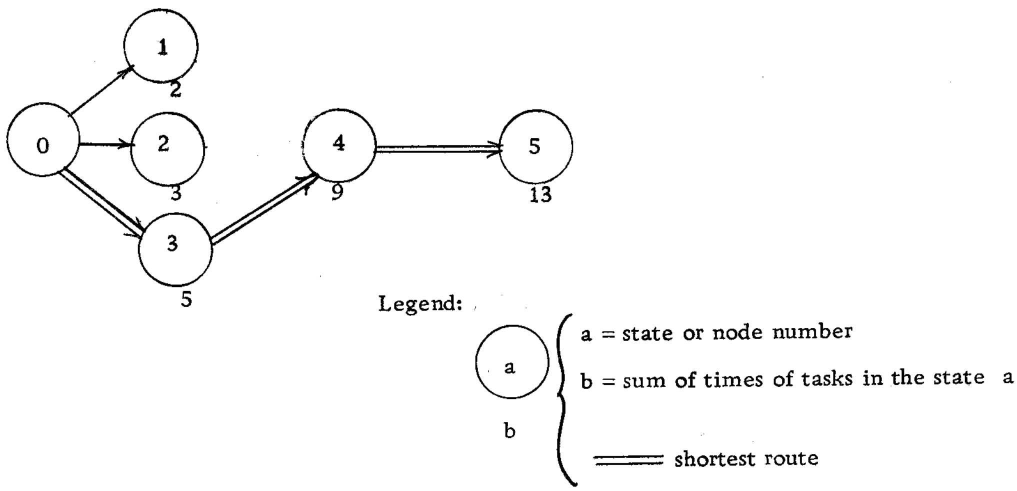


Figure 5-3. Network representation for model 2.

The elements to be assigned to each station are arrived at by using the same rules as explained earlier. A summary is shown in Table 5-4.

Table 5-4. Summary of station assignments for the shortest route shown in Figure 5-3, for model 2 at $c = 5$.

Arc Number	Nodes Spanned By Arc	Work Elements Spanned by Arc	Station Time	Station Idle Time
1	0, 3	u_1, u_2	5	0
2	3, 4	u_3	4	1
3	4, 5	u_4	4	1

Now consider the example with the shortest route formulation for mixed models in the Equations (5.5) through (5.8). Referring to the precedence diagram of model 1 we have:

$$t_{11} = 4, \ t_{21} = 2, \ t_{31} = 3, \ t_{41} = 2, \ t_{51} = 2,$$

and for model 2,

$$t_{12} = 2, \ t_{22} = 3, \ t_{32} = 4, \ t_{42} = 4.$$

The set of states for model 1, are given by

$$S_2 = \{(1), (1, 2), (1, 4), (1, 2, 3), (1, 2, 4), (1, 2, 3, 4), (1, 2, 3, 4, 5)\}$$

and for model 2,

$$S_2 = \{(1), (2), (1, 2), (1, 2, 3), (1, 2, 3, 4)\}.$$

From Equation (5.5)

$$S = \{[(1), (1)], [(1), (2)], [(1), (1, 2)], [(1, 2)], [(1), (1, 2, 3)],$$

$$[(1), (1, 2, 3, 4)], \cdots, [(1, 2, 3), (1), \cdots, [(1, 2, 3, 4, 5), (1, 2, 3, 4],$$

$$\phi\}.$$

Table 5-5 gives the complete list of all the states generated. A network is constructed from Table 5-5 and is shown in Figure 5-4.

Table 5-5. Generation of states for models 1 and 2.

Node	State Element Numbers Model 1	State Element Numbers Model 2	State Time Model 1	State Time Model 2
(1)	(2)	(3)	(4)	(5)
0	ϕ	ϕ	0	0
1	(1)	(1)	4	2
2	(1)	(2)	4	3
3	(1)	(1, 2)	4	5
4	(1)	(1, 2, 3)	4	9
5	(1)	(1, 2, 3, 4)	4	13
6	(1, 2)	(1)	6	2
7	(1, 2)	(2)	6	3
8	(1, 2)	(1, 2)	6	5
9	(1, 2)	(1, 2, 3)	6	9
10	(1, 2)	(1, 2, 3, 4)	6	13
11	(1, 4)	(1)	6	2
12	(1, 4)	(2)	6	3
13	(1, 4)	(1, 2)	6	5
14	(1, 4)	(1, 2, 3)	6	9
15	(1, 4)	(1, 2, 3, 4)	6	13
16	(1, 2, 3)	(1)	9	2
17	(1, 2, 3)	(2)	9	3
18	(1, 2, 3)	(1, 2)	9	5
19	(1, 2, 3)	(1, 2, 3)	9	9
20	(1, 2, 3)	(1, 2, 3, 4)	9	13
21	(1, 2, 4)	(1)	8	2
22	(1, 2, 4)	(2)	8	3
23	(1, 2, 4)	(1, 2)	8	5
24	(1, 2, 4)	(1, 2, 3)	8	9
25	(1, 2, 4)	(1, 2, 3, 4)	8	13
26	(1, 2, 3, 4)	(1)	11	2
27	(1, 2, 3, 4)	(2)	11	3
28	(1, 2, 3, 4)	(1, 2)	11	5
29	(1, 2, 3, 4)	(1, 2, 3)	11	9
30	(1, 2, 3, 4)	(1, 2, 3, 4)	11	13
31	(1, 2, 3, 4, 5)	(1)	13	2
32	(1, 2, 3, 4, 5)	(2)	13	3
33	(1, 2, 3, 4, 5)	(1, 2)	13	5
34	(1, 2, 3, 4, 5)	(1, 2, 3)	13	9
35 = r	(1, 2, 3, 4, 5)	(1, 2, 3, 4)	13	13

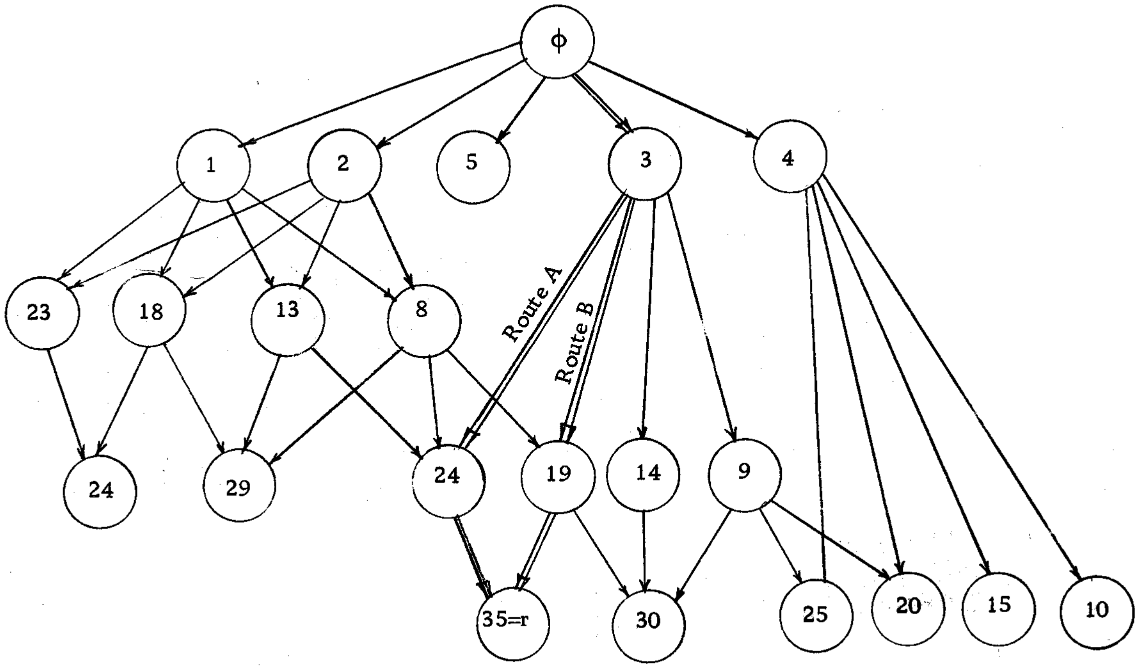


Figure 5-4. Network representation for models 1 and 2.

From the Figure 5-4, it can be seen that there are two shortest routes[2], i. e., (ϕ -3-19-35) and (ϕ -3-24-35) and this agrees with the result obtained when the models were considered independently. The summary of station assignments for the two best routes is shown in Table 5-6.

Table 5-6 shows that either route A or route B can give the same total idle time of 2 for each model in the line. Both are equally good for selecting the station assignments when the objective is to

---

[2] Roberts and Villa (1970) did not direct their attention to the route (ϕ -3-24-35) in their solution.

Table 5-6. Summary of station assignments for both the shortest routes of models 1 and 2 at c = 5.

Station	Model 1 Tasks				Model 2 Tasks	
	Route A	Station Time	Route B	Station Time	Route A&B	Station Time
1	u_1	4	u_1	4	u_1, u_2	5
2	u_2, u_4	4	u_2, u_3	5	u_3	4
3	u_3, u_5	5	u_4, u_5	4	u_4	4

minimize the total idle time in the line. However, the solution offered by route (ϕ-3-19-35) will force the operators 1 and 2 to be trained for the same task 2, while in the alternative solution from the route (ϕ-3-24-35), the task 2 must be performed by the operators 1 and 2 and the tasks 3 and 4 by the operators 2 and 3. Thus the route (ϕ-3-19-35) may be superior to the other route (ϕ-3-24-35) to cut the costs of learning and parts stocking.[3]

Production Shift Time Basis

A practical method of balancing a mixed-model line was first suggested by Thomopoulos (1966 and 1967). He considered the total schedule for a production shift and assigned work elements to operators on a shift basis rather than on a cycle time basis. This approach

---

[3] This logic ~~of ours~~ for the selection of route (ϕ-3-19-35) was agreed by Roberts and Villa (1970) in their reply dated February 19, 1971, to our letter of January 26, 1971.

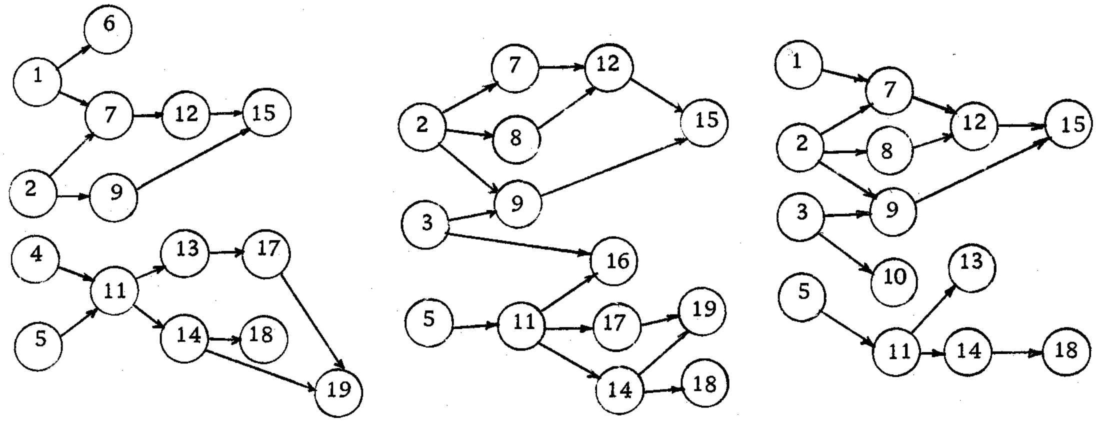
eliminated certain possible additional costs which arose in the mathematical formulation by Roberts and Villa (1970) for balancing a mixed-model line on a cycle time basis.

> Since an assembler is trained to perform a job which requires some skill, it is desirable, if not imperative, to assign jobs of a specified class to one operator or at most a small group of operators (Thomopoulos, 1967, p. B-64).

For example, the installation of steering wheel in any automobile should be assigned to one or, at most, a few operators sufficiently familiar with this type of job. When the elements are assigned to a station on a shift basis, the single-model ALB procedures are adaptable to mixed-model lines. This will assure that only one operator will perform the same tasks on all units of all models thereby reducing the costs of learning,parts stocking and confusion.

Example:

Consider the precedence diagrams for three different models A, B and C in Figure 5-5. If these three models are to be assembled on the same assembly line with a daily schedule of say 110, 60 and 50 units of A, B and C respectively, then we have a problem of assigning the total work content of all the scheduled units to a minimum number of operators subject to the condition that no operator will be assigned a work content more than the available production time in a day. The solution to this type of problem can be arrived at with the following steps.



Model A (110 units) Model B (60 units) Model C (50 units)

Figure 5-5. Precedence diagrams for 3 models.

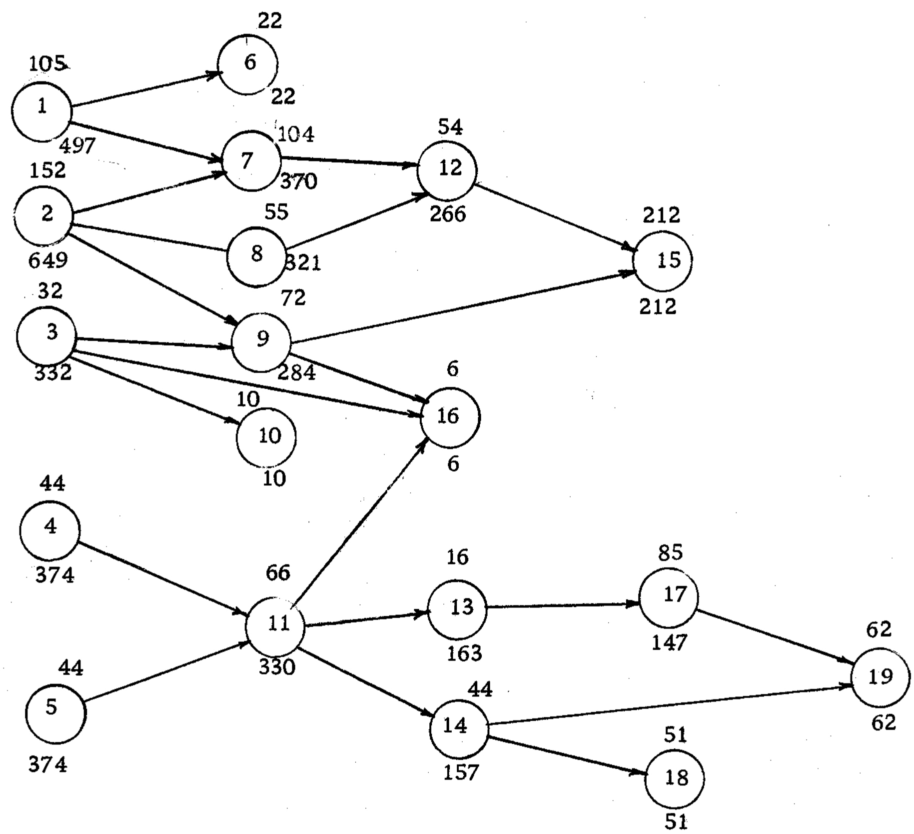


Figure 5-6. Combined precedence diagram for all the 3 models A, B and C .

<u>Step 1.</u> Construct a combined precedence diagram (Figure 5-6 for the example problem).

<u>Step 2.</u> Prepare a table showing all the elemental times for each model (Column II of Table 5-7) and the work content for each task per model (Column III of Table 5-7).

<u>Step 3.</u> Calculate the total work content for each task in a shift (Column IV of Table 5-7).

<u>Step 4.</u> Calculate the average work load per operator and specify an upper and a lower limit. These limits are left to the discretion of the management (usually an industrial engineer). For the example problem the total work content to complete 220 units of all models is 1, 236 minutes. Let the productive work time of an operator be 450 minutes excluding 30 minutes of personal allowance from an eight hour shift time. Then we require a minimum of $[\frac{1236}{450}] = 3$ operators. Thus the average work load per operator will be $[\frac{1236}{3}] = 412$ exactly. Let us specify the upper limit of this as say 420 and the lower limit as 412. Now our problem is to assign the total work content to each of the 3 operators such that their station times will lie between 412 and 420. If we can assign each operator exactly 412 minutes of work satisfying the precedence restrictions and assuring

Table 5-7. Summary of work element times for the models A, B and C to produce quantities of 110, 60 and 50 respectively.

I Element Number	II Element Times Per Model			III Work Content Time Per Model/Shift			IV Work Content of All Models/Shift Element
	Model A	Model B	Model C	Model A	Model B	Model C	
1	0.5	0.0	1.0	55	0	50	105
2	0.4	0.8	1.2	44	48	60	152
3	0.0	0.2	0.4	0	12	20	32
4	0.4	0.0	0.0	44	0	0	44
5	0.2	0.2	0.2	22	12	10	44
6	0.2	0.0	0.0	22	0	0	22
7	0.4	0.5	0.6	44	30	30	104
8	0.0	0.5	0.5	0	30	25	55
9	0.4	0.3	0.2	44	18	10	72
10	0.0	0.0	0.2	0	0	10	10
11	0.3	0.3	0.3	33	18	15	66
12	0.1	0.3	0.5	11	18	25	54
13	0.1	0.0	0.1	11	0	5	16
14	0.2	0.2	0.2	22	12	10	44
15	0.7	1.0	1.5	77	60	75	212
16	0.0	0.1	0.0	0	6	0	6
17	0.5	0.5	0.0	55	30	0	85
18	0.3	0.3	0.0	33	18	0	51
19	0.4	0.3	0.0	44	18	0	62
TOTALS	5.1	5.5	6.9	561	330	345	1236

that only one operator will perform the same tasks on all models, then we would have obtained an ideal optimal solution to the problem.

Step 5. Apply one of the earlier described procedures of single-model ALB problem using the elemental times calculated in Step 3 above, and a cycle time c, where $412 \leq c \leq 420$. In the example problem the times in Column IV of Table 5-7 are considered, as if they were the elemental times of a single-model problem whose precedence diagram would be equivalent to the combined precedence diagram of all the 3 models in Figure 5-6. The BALB with Linpows is used to balance this mixed-model line to obtain a minimal station balance at $c = 416$ with a S. I. = 8.602 (Table 5-8)

Table 5-8. Summary of the results of BALB for 19 element example problem with deterministic work elements at $c = 416$ and S. I. = 8.602.

Station Number k	Tasks (i)	Station Time T_k	Station Delay d_k
1	2, 1, 5, 4, 3, 6, 10	409	7
2	7, 11, 8, 9, 12, 13, 14	411	5
3	15, 17, 19, 18, 16	416	0

However, Moodie's phase 1 procedure would give a minimal station balance with S. I. = 9.486 at $c = 416$ after 5 iterations starting at

$c = 412$. It is not possible to obtain a minimal station balance at an average work load of 412, 413, 414, and 415 due to the nature of work element times. This fact is revealed by the failure of BALB to obtain a 3 station balance at these values.

B. Minimization of Direct Labor Costs with Variable Work Element Times

The earlier approaches of mixed-model balancing were limited to deterministic work element times though in the actual assembly process the elemental times are independent and identically distributed normal variates. Extensions of mixed-model line balancing for variable work elements is illustrated in this section.

Cycle Time Basis

When the mixed-model line is balanced on the basis of a cycle time, our objective will be to minimize the total idle time in all the stations subject to the precedence and cycle time constraints. When the work element times are deterministic, the objective will be to minimize the delay function,

$$D = \sum_{j=1}^{J} \sum_{k=1}^{m} N_j (c - T_{kj})$$

subject to the condition $T_{kj} \leq c$; $j = 1, 2, \cdots, J$
$k = 1, 2, \cdots, m$

where N_j is the number of units of model j and T_{kj} is the work content of the station k for model j obtained by adding the deterministic performance times of all work elements grouped at the station k for the model j .

When we remove the assumptions of deterministic work element times and consider them as normal variates (under the same assumptions made earlier to illustrate Moodie's (1965) heuristic method) our objective function will be read as,

$$\text{minimize } D = \sum_{j=1}^{J} \sum_{k=1}^{m} N_j \{c - (T_{kj} + Z \sqrt{V(T_{kj})})\}$$

where $V(T_{kj})$ = variance of the random variable T_{kj} .

$$T_{kj} = \sum_{i \in I_j} t_{ij} + \sqrt{\sum_{i \in I_j} V(t_{ij})} \text{ ; } j = 1, 2, \cdots, m$$

Z = standard normal deviate obtained from the standard normal tables

For a given value of probability of station times to exceed the cycle time c (e. g., if we allow the individual values of T_{kj} to exceed the cycle time c 15% of the time, the value of Z would be 1.035).

t_{ij} = performance time of i-th element on j-th model

$V(t_{ij})$ = variance of t_{ij}

I_j = set of work elements contained in the j-th model

When we extend these concepts to the integer programming and shortest route model of section A in this chapter, the Equations (5.4) and (5.8) will be modified as follows:

$$\text{minimize } D' = \sum_{k=m_j^*+1}^{M} W_k \sum_{j=1}^{J} N_j \sum_{i\epsilon I_j} X_{kij} \{t_{ij} + \sqrt{V(t_{ij})}\} \tag{5.4'}$$

and

$$\sum_{i\epsilon\{\ \}} t_{ij} + Z \sqrt{(\sum_{i\epsilon\{\ \}} V(t_{ij})} \leq c\ ;\ j = 1, 2, \cdots, J \tag{5.8'}$$

and the symbol { } in the Equation (5.8)' denotes the set of all tasks contained in the state $\{S_{jv} - S_{ju}\}$ and Z is the standard normal deviate for the given value of probability of station times to exceed the cycle time.

Example:

Consider the example in Figure 5-1 by specifying some variance values for the models as shown in Table 5-9 and Table 5-10.

Table 5-9. Data of elemental times and variance values for model 1 to illustrate shortest route method.

Element Number	1	2	3	4	5
Element Time	4	2	3	2	2
Variance	1.00	0.25	0.50	0.30	0.30

Table 5-10. Data of elemental times and variance values for model 2 to illustrate shortest route method.

Element Number	1	2	3	4
Element Time	2	3	4	4
Variance	0.4	0.6	1.0	1.0

The generation of states in Table 5-5 and the Equations (5.7) and (5.8)' lead to the following results in Table 5-11. The results are obtained when Z = 1 (assuming that 15% of the time the station values might exceed the cycle time) with the shortest route (0-2-23-29-35) being spanned by 4 acrs.

Table 5-11. Summary of station assignments for both models 1 and model 2 with variability of work elements.

	Model 1			Model 2		
Number	i	T_k	d_k	i	T_k	d_k
1	1	5.00	0.00	2	3.77	1.23
2	2, 4	4.74	0.26	1	2.63	2.37
3	3	3.70	1.30	3	5.00	0.00
4	5	2.54	2.46	4	5.00	0.00

Comparisons of results in Table 5-6 and Table 5-11 will indicate that the consideration of variability does change the results. In Table 5-6 we obtained a 3 station balance while Table 5-7 indicates a 4 station balance.

Production Shift Time Basis

When the daily production shift time is used as the basis for a mixed-model ALB, certain assumptions must be made before it can be converted into an equivalent single-model ALB problem.

Let us assume that a same kind of task when performed by the same operator on different models will result in performance times which are independent and identically distributed normal variates. One might say that the results of earlier studies by Hicks and Young (1962) could be extended to defend this assumption. However, this might not be entirely justified since their research did not consider the elemental times of the same kind of task by the same operator on different products. The above assumption of normality for the distribution of mean times taken on the same task on different models can be supported by the central limit theorem.

Let $x_1, x_2, \cdots, x_n$ be a random sample of size n drawn from any population with a mean μ and variance σ^2. Then the distribution of $\frac{(\bar{x} - \mu)}{\sigma / \sqrt{n}} = \frac{(\bar{x} - \mu)}{\sigma_{\bar{x}}}$ approaches the standard normal as n increases. The theorem implies that the distribution of $\bar{x}$ is approximately normal with mean μ and variance σ^2/n for moderate or large n (Guenther, 1965, p. 81).

Also the assumption of independence between the variates $x_1, x_2, \cdots, x_n$ has been used successfully in the past by Walker (1959). In MTM and other predetermined work measurement, time data have usually assumed independence. The success of these usages in industry seem to imply that x_{ij} (the performance time of the i-th task on the j-th model) can be considered as an independent and identically distributed normal variate with a mean t_{ij} (the expected time measured by stop watch study for the i-th task on the j-th model) and a standard deviation σ_{ij}. We can thus use the additive property of normal variates to arrive at the mean work content of all models per shift for any particular element (i. e., the equivalent of the entries in Column IV of Table 5-7 for deterministic work elements). The general theorem of additivity for normal variates is stated as follows:

> If x_i $(i = 1, 2, \cdots, n)$ are independent normal variates with mean m_i and variance σ_i^2, the variate $\Sigma c_i x_i$ is a normal variate with mean $\Sigma c_i m_i$ and variance $\Sigma c_i^2 \sigma_i^2$, where c_i's are constants (Kapur and Saxena, 1961, p. 169).

The variability consideration can now be included in the procedure of mixed-model line balancing. Let

N_j = number of units to be assembled for the j-th model

x_{ij} = performance time of i-th element on j-th model, a normal variate with mean t_{ij} and variance σ_{ij}^2

t_{ij} = $\overline{x}_{ij}$ = mean of x_{ij} (note $t_{ij} = x_{ij}$ for deterministic case

σ_{ij} = standard deviation of the random variable x_{ij}

T_i = work content of all models per shift for the i-th element

Then in a deterministic case

$$T_i = \sum_{j=1}^{J} N_j t_{ij} \quad \text{for } i = 1, 2, 3, \cdots, n$$

and in a variable case the value T_i will be a normal variate with a mean $\overline{T}_i = \sum_j N_j \overline{x}_{ij} = \sum_j N_j t_{ij}$ = equivalent of $\overline{x}_i$ for a single-model; and variance $V(T_i) = \sum_j N_j^2 \sigma_{ij}^2$ = equivalent of $V(x_i)$ for a single-model. This will reduce the problem into an equivalent single-model problem with variable work element times. Hence we can apply BALB (described in Chapter IV) to solve this mixed-model problem by specifying the average work load per operator in place of cycle time. Table 5-12 shows the data of the standard deviations assumed in the example problem in Figure 5-5. Table 5-13 summarizes the dual precedence lists, values of $\overline{T}_i$, $V(T_i)$, and Linpows as an input data for BALB. Application of BALB results in a 3 station balance at a cycle time of 419 minutes.[4] The smoothness index was 9.34 as shown in Table 5-14.

---

[4] This is the average work load per operator.

The same problem was solved by Moodie's phase 1 method. It also gave a 3 station balance (Table 5-15) but with a smoothness index of 13, 21 (to the second decimal) which was higher than what was obtained by BALB. However, it should be realized now that consideration of variability had changed the results of the deterministic elemental problem by increasing the average load on the operator from a value of 416 to 419. Correspondingly, S. I. increased from 8. 60 to 9. 34. These changes may be of practical interest in arriving at the deviation of the shift time allowed in the assembly process and the criteria of minimizing the direct labor costs.

Table 5-12. Data of the standard deviations of all 19 work elements for the 3 model example in Figure 5-5.

Element Number (i)	1	2	3	4	5	6	7	8
Model A (σ_{iA})	0.01	0.008	0	0.007	0.006	0.005	0.008	0
Model B (σ_{iB})	0	0.015	0.005	0	0.006	0	0.010	0.01
Model C (σ_{iC})	0.02	0.030	0.008	0	0.007	0	0.012	0.01

Table 5-12 continued.

Element Number (i)	9	10	11	12	13	14	15	16
Model A (σ_{iA})	0.006	0	0.004	0*	0.001	0.005	0.015	0
Model B (σ_{iB})	0.004	0	0.004	0.004	0	0.005	0.018	0.001
Model C (σ_{iC})	0.003	0.005	0.003	0.010	0.001	0.005	0.022	0

Table 5-12 continued.

Element Number (i)	17	18	19
Model A (σ_{iA})	0.012	0.004	0.005
Model B (σ_{iB})	0.010	0.004	0.003
Model C (σ_{iC})	0	0	0

* Assume that the element 12 is done by a machine for the Model A.

Table 5-13. Data for the total work content and the variance of each element for 19 element example.

Element Number	IP-List			Mean Work Content $(\overline{T}_i)$	Variance $V(T_i)$	Linpows	IF-List		
1	0	0	0	105.0	2.2	497.0	6	7	0
2	0	0	0	152.0	3.8	649.0	7	8	9
3	0	0	0	32.0	.2	332.0	9	10	16
4	0	0	0	44.0	.6	374.0	11	0	0
5	0	0	0	44.0	.7	374.0	11	0	0
6	1	0	0	22.0	.3	22.0	0	0	0
7	1	2	0	104.0	1.5	371.5	12	0	0
8	2	0	0	55.0	.6	321.0	12	0	0
9	3	2	0	72.0	.5	284.0	15	0	0
10	3	0	0	10.0	.1	10.0	0	0	0
11	4	5	0	66.0	.3	330.3	13	14	16
12	7	8	0	54.0	.3	266.0	15	0	0
13	11	0	0	16.0	0	163.0	17	0	0
14	11	0	0	44.0	.5	157.0	19	18	0
15	9	12	0	212.0	5.1	212.0	0	0	0
16	3	11	0	6.0	0	6.0	0	0	0
17	13	0	0	85.0	2.1	147.0	19	0	0
18	14	0	0	51.0	.2	51.0	0	0	0
19	14	17	0	62.0	.3	62.0	0	0	0

Table 5-14. Summary of results with *BALB for 19 element example problem with variability at c = 419 and S.I. = 9.34.

Station Number	Tasks	Station Time	Station Delay
1	2, 1, 5, 4, 3, 6, 10	411.81	7.19
2	7, 11, 8, 9, 12, 13, 14	412.92	6.08
3	15, 17, 19, 18, 16	418.77	0.23

Table 5-15. Summary of results with Moodie's phase 1 of the HALB for 19 element example problem with variability at c = 419 and S.I. = 13.21.

Station Number	Tasks	Station Time	Station Delay
1	2, 1, 7, 8	418.85	0.15
2	12, 5, 4, 11, 14, 18, 3, 9, 10	418.84	0.16
3	6, 13, 17, 19, 16, 15	405.79	13.21

C. Minimization of Sequence Delay Costs

After balancing a mixed-model line, the next problem would be to find in what sequence to launch the various models on the conveyor line. The different models require different amounts of assembly work causing an uneven distribution of work load among the individual operators. This will result in various assembly line inefficiencies such as work congestion, work deficiency, operator idleness or utility work time. By assigning a penalty in cents per unit time associated with each inefficiency, it is possible to compute the total cost of inefficiencies resulting from scheduling a unit of a given model in the sequence. These penalty costs are commonly referred to as sequence delay costs and our objective is to minimize these costs by an optimal ordering of the flow of models.

Analysis of the Sequencing Problem

Using the notation of Section B of this chapter the minimum number of operators, m^*, required to produce the given number of units in time period T is given by:

$$m^* = \left\lceil \{\sum_j N_j \sum t_{ij}\}/T \right\rceil \qquad (5.9)$$

where the bracket function $[x]$ equals to the least integer $\geq x$.

Then the average load per operator is

$$T_a = \frac{\sum_j (N_j \sum_i t_{ij})}{m^*} \tag{5.10}$$

If on the other hand, the N_j's are not given, but the ratios of different models $f_1:f_2:f_3: \cdots$, are known, the maximum production of each model in time period T with m^* operators is

$$\max N_j = \frac{m^* \times T \times f_j}{\sum_j (f_j \sum_i t_{ij})} \; ; \; j = 1, 2, \cdots, J \tag{5.11}$$

We assume that the assembly work of each model can be evenly divided among the m^* operators so that each operator works on a given model for the same amount of time. This time deviation is commonly termed as "model cycle time", designated by c_j and found by the equation,

$$c_j = \frac{\sum_{i=1}^{n} t_{ij}}{m^*} \; ; \; j = 1, 2, \cdots, J$$

The model with the maximum total amount of assembly work will also have the maximum model cycle time for a given m^*. The maximum model cycle time will be denoted by θ, where

$$\theta = \max_j \{c_j\}$$

The weighted average of the model cycle time is denoted by β, where

$$\beta = \frac{\sum_{j=1}^{J} f_j c_j}{\sum_{j=1}^{J} f_j} \qquad (5.12)$$

To illustrate the above concepts consider the example in Figure 5-5 where the numbers of units required per shift (say 450 minutes) for the three modes A, B and C are:

$$N_A = 110\ ,\ N_B = 60\ ,\ N_C = 50$$

Thus, the corresponding ratios are $f_A : f_B : f_C = 11:6:5$. The amounts of assembly work, or work content times (in minutes) per unit of each model (Table 5-6) are $\sum_i t_i A = 5.10$, $\sum_i t_i B = 5.5$ and $\sum_i t_i C = 6.90$. From Equation (5.9) the minimum number of operators m^* required to satisfy the given production schedule in a 450 minutes working day = $\left[\frac{110 \times 5.1 + 60 \times 5.5 + 50 \times 6.9}{450}\right] = [2.74] = 3$. Thus, the three operators will suffice to perform the shift's work. The average load per operator from Equation (5.10) is

$$T_a = \frac{1236}{3} = 412 \text{ minutes} .$$

With the given production ratios 11:6:5 , the maximum production of each model actually attainable with three operators in a 450 minute work shift is,

$$\max N_j = \frac{450.0 \times 3 \times f_j}{\sum_j (f_j \sum_i t_{ij})}, \quad j = A, B \text{ and } C$$

Thus,

$$\max N_A = \frac{(450 \times 3 \times 11)}{11 \times 5.1 + 6 \times 5.5 + 5 \times 6.9)} \simeq 1120$$

$$\max N_B = \frac{450 \times 3 \times 6}{123.6} \simeq 66$$

$$\max N_C = \frac{450 \times 3 \times 5}{123.6} \simeq 55 .$$

The model cycle times are given by $c_A = \frac{5.1}{3} \sim 1.7$; $c_B = \frac{5.5}{3} \sim 1.83$; and $c_C = \frac{6.9}{3} \sim 2.3$. From Equation (5.12), $\beta = (11 \times 1.7 + 6 \times 1.83 + 5 \times 2.3)/(22) = 1.8718$. The maximum model cycle time $= \theta = \max \{1.7, 1.83, 2.3\} = 2.3$. The summary of these results is shown in Table 5-16.

Table 5-16. Summary of results for the elements of sequencing.

Model	Number of Units Required Per Shift	Work Content Time/Unit	Work Content Time/Shift	Model Cycle Time
A	110	5.1	561.0	1.70
B	60	5.5	330.0	1.83
C	50	6.9	345.0	2.30

Sequencing Criteria

Wester and Kilbridge (1964) used the criteria of idle time of an operator and work congestion at a station to develop a solution procedure for the sequencing. Later Thomopoulos (1966) generalized these solutions based on one day's schedule and inefficiencies like idle time, work congestion, work deficiency and utility work. Sahgal (1970) extended the general sequencing procedure of Thomopoulos by considering a weekly production schedule and including the shortage cost as an additional inefficiency cost, which might appear due to demand fluctuations and lack of finished goods inventory. Lehman (1969) discussed the use of sequence delay, balance delay and the operator learning as criteria in a sequencing procedure.

Sequencing Solutions

The solution methods based on the interval of launching units on a conveyor line was developed by Wester and Kilbridge. These two procedures on variable and fixed rate launching are summarized below.

Variable rate launching: In this procedure the time between the launching of successive units is considered to be proportional to the total work content time of the units. This can be achieved by spacing the units by a time interval equal to the model cycle time of the leading unit. If a unit of model A is to be followed by a unit of model B, then the elapsed time between launchings of the two units is the model cycle time of A.

Assumptions:

i) The assembly work on each model can be evenly divided among the m^* operators.

ii) The conveyor moves at a constant speed given by

$$v = \ell/\theta \quad \text{where} \quad \ell = \text{maximum length of the conveyor}$$
$$\theta = \text{maximum model cycle time}$$

iii) The work stations are non-overlapping, i.e., the operator $k+1$ can not start work on a given unit before operator k $(k = 1, 2, \cdots, m^*-1)$ finishes his work on it.

Figure 5-7 illustrates the variable rate launching system, where c denotes the model cycle time of the unit i launched while the subscripts of c denote the order of launching of the units. From Figure 5-7 and with the assumption that the stations are non-overlapping the following relations must be satisfied:

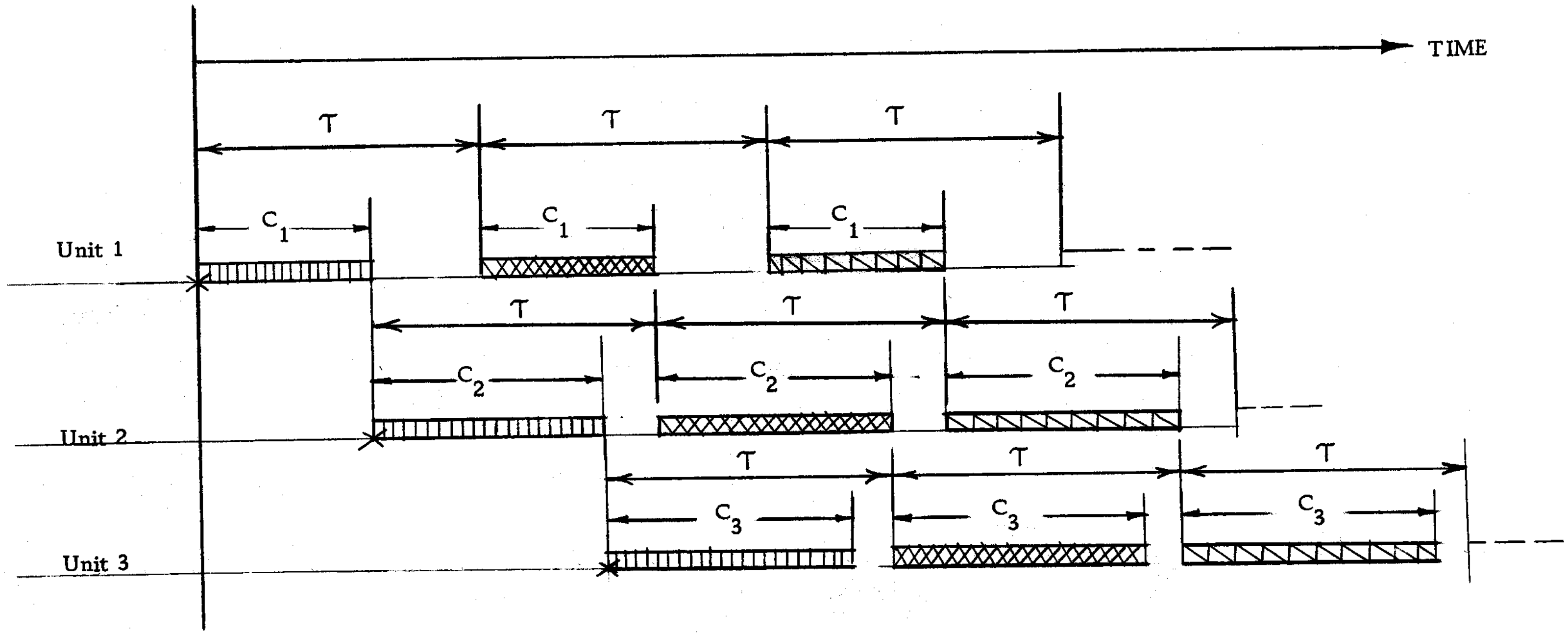
$$\tau \geq c_i \; ; \; i = 1, 2, \cdots, \sum_j N_j$$

Since

$$\{c_j\} \subset \{c_i\}$$

we have

$$\tau \geq c_j \; ; \; j = 1, 2, \cdots, J$$



Legend:

X Launching times

Work performed by operator 1

Work performed by operator 2

Work performed by operator 3

τ Operator cycle time, the maximum c_j, or the time separating the start of work on a given unit by two consecutive operators on the line.

c_i $(i = 1, 2, \cdots, N)$ is the model cycle time of unit i (the subscript of c denotes the order of launching the units

Figure 5-7. Variable rate launching.

Hence

$$\tau \geq \max_j \{c_j\}$$

If $\tau > \max c_j$, operator idle time will result. Therefore $\tau_{opt} = \max_j \{c_j\} = \theta$.

Line utilization increases with the overlapping of stations in this type of launching and this is commonly observed in automotive industry. The disadvantage in this type of system is that the scheduling of multi-model components and integration with other production lines will be difficult.

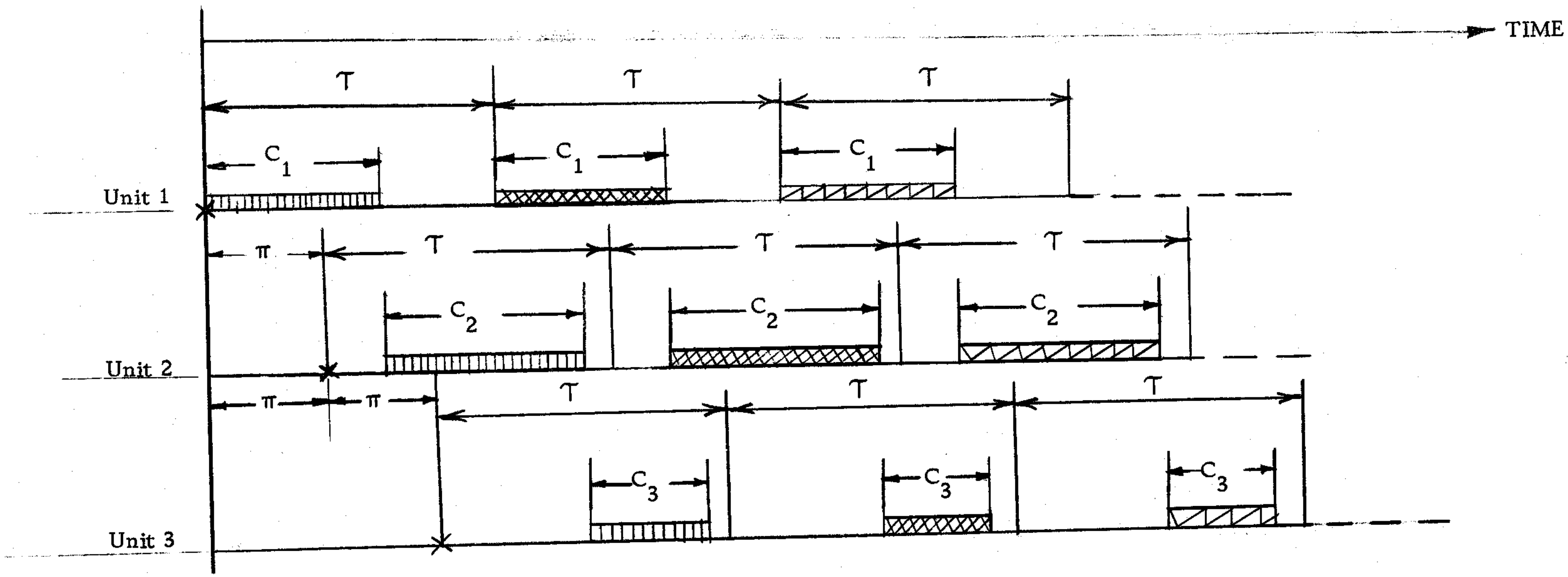
Fixed-rate launching: In this procedure, the consecutive units, regardless of the model, are launched at regular time intervals. To minimize the work congestion and operator idleness we have to choose the optimum fixed interval for lauching the units. Referring to Figure 5-8, the following results are obtained by Wester and Kilbridge (1964). The operator cycle time $\tau = \max_j \{c_j\}$. Also from lines 1 and 2 corresponding to units 1 and 2 in Figure 5-8, we have

$$\pi \leq c_1$$

and from lines 1, 2 and 3 corresponding to units 1, 2 and 3, we have

$$2\pi \leq c_1 + c_2$$

Thus by induction



Legend:

X Launching times

Work done by operator 1

Work done by operator 2

Work done by operator 3

T Operator cycle time, the maximum c_j, or the time separating the start of work on a given unit by two consecutive operators on the line.

c_i ($i = 1, 2, \cdots, N$) is the model cycle time of unit i.

π The production cycle time; the fixed time between two consecutive launchings.

Figure 5-8. Fixed rate launching.

$$N\pi \leq (c_1 + c_2 + c_3 + \cdots + c_N) \quad \text{where} \quad N = \sum_j N_j .$$

Rewriting the above equation

$$\pi \leq \frac{\sum_i c_i}{N} = \frac{\sum_j N_j c_j}{\sum_j N_j} ,$$

since for each c_j there are N_j number of c_i's. The above inequalities can not be satisfied simultaneously unless the c_i $(i = 1, 2, \cdots, N)$ are properly chosen. The proper selection will decide the optimum sequencing with respect to operator idle time.

From the relation

$$\pi \leq \frac{\sum_j N_j c_j}{\sum_j N_j}$$

if the strict inequality holds, then the units will appear on the line prematurely and will cause congestion at various stations. Hence π must equal $(\sum_j N_j c_j)/(\sum_j N_j)$.

The optimum conditions for fixed-rate launching are therefore

$$\pi_{opt} = \beta = \frac{\sum_j N_j c_j}{\sum_j N_j}$$

and

$$\tau_{opt} = \theta = \max_j \{c_j\}$$

Since π, the time between any two consecutive launchings, is constant and determines the production rate, it is called the production cycle time. Since $\tau \geq \max_j \{c_j\} = \max_j \{c_i\}$ and $\pi \leq \frac{\Sigma c_i}{i}$ for $i = 1, 2, \cdots, N$, we can conclude that

$$\pi \leq \tau \text{ since } \frac{\sum c_i}{i} \leq \max \{c_i\} \text{ for } i = 1, 2, \cdots, N$$

The equality holds good if all model cycle times are equal. This is the case when a single model is produced on the line. The single model problem is a special, but trivial, case of mixed-model situation. Thus to prevent operator idle time, we need

$$i\pi \leq \sum_{h=1}^{i} c_h \text{, where } i = 1, 2, \cdots, N \tag{5.13}$$

Our next criterion is to prevent work congestion in the line. This can be achieved subject to the following conditions:

$$\tau \geq c_1$$

$$\tau + \pi \geq c_1 + c_2$$

$$\tau + 2\pi \geq c_1 + c_2 + c_3$$

By induction,

$$\tau + (m-1)\pi \geq c_1 + c_2 + \cdots + c_N$$

or

$$\tau + (i-1)\pi \geq \sum_{h=1}^{i} c_h, \quad (i = 1, 2, \cdots, N) \tag{5.14}$$

The reason for these conditions is apparent from the Figure 5-8 and a mathematical justification can be found in the Ph. D. dessertation of Thomopoulos (1966).

The Equations (5.13) and (5.14) above must be satisfied to minimize the sequence delay costs, i.e.,

$$0 \leq \sum_{h=1}^{i} c_h - i\pi \leq \tau - \pi \tag{5.15}$$

> In some cases, depending on the distribution of the model cycle times, the above inequality can be satisfied. But in many other cases this is not possible. Since avoiding operator idle time is our primary objective, we shall choose the c_h to satisfy (5.13) (Wester and Kilbridge, 1964, p. 254).

However, to minimize the work congestion, choose c_h such that the difference $\sum_{h=1}^{i} c_h - i\pi$ is as small as possible at each step i of the procedure. At certain steps it happens that the operator is forced out of his work station in order to complete his assignment (for example see units 6, 11 and 16 of Table 5-17. Fixed rate launching provides a uniform rate of production, is more adaptable to practical situations, and is more suitable for computer application.

Example:

In Table 5-16, we have the set $\{N_j\} = \{110, 60, 50\}$ and the greatest common divisor is 10. Then the quotient set $\{f_j\} = \{11, 6, 5\}$. Hence the sequencing procedure can be simplified by applying the earlier mentioned rules strictly to 11 A's, 6 B's and 5 C's (blocking the unrequired models if necessary), and repeating the resulting sequence for 10 times to achieve the total scheduled production. Table 5-17 summarizes the sequencing.

Solution:

Table 5-17 summarizes the sequencing of units obtained by this procedure.

The mixed-model line balancing and sequencing, the two most recent extensions of the single-model assembly lines, have been discussed in this chapter. It must be pointed out that none of the researchers above including the present author claim to have found the optimum result for either balancing or sequencing problems.

However, the line balancing solutions discussed in this chapter serve as the beginning stage for solving the mixed-model assembly

Table 5-17. Cyclical sequencing of units, for the mixed-model problem by fixed rate launching.

Unit (i)	Multiples of Production Cycle Time (i)	Model (j)	Model Cycle Time (c_i)	$\sum_{h=1}^{i} c_h$	$\sum_{h=1}^{i} c_h - i\pi$
1	1.8718	C	2.30	2.30	0.4282
2	3.7436	A	1.70	4.00	0.2564
3	5.6154	A	1.70	5.70	0.0846
4	7.4872	B	1.83	7.53	0.0428
5	9.3590	B	1.83	9.36	0.0010
6	11.2308	C	2.30	11.66	0.4292*
7	13.1026	A	1.70	13.36	0.2574
8	14.9744	A	1.70	15.06	0.0856
9	16.8462	B	1.83	16.89	0.0438
10	18.7180	B	1.83	18.72	0.0020
11	20.5898	C	2.30	21.02	0.4302*
12	22.4616	A	1.70	22.72	0.2584
13	24.3334	A	1.70	24.42	0.0866
14	26.2052	B	1.83	26.25	0.0448
15	28.0770	B	1.83	28.08	0.0030
16	29.9488	C	2.30	30.38	0.4312*
17	31.8206	A	1.70	32.08	0.2594
18	33.6924	A	1.70	33.78	0.0876
19	35.5642	B	1.83	35.48	0.0458
20	37.4360	B	1.83	37.18	0.0040
21	39.3078	C	2.30	38.88	0.4322*
22	41.1796	A	1.70	41.18	0.2614

$\pi = 1.8718$, $\theta = T = 2.3$, $T - \pi = 0.4282$

* = For these units the operator will be forced out of his work station to complete his work.

line problem. We hope future research will focus on some of the related problems such as the integration of inter-related subassembly lines and the scheduling of the production of the multi-product components.

VI. DISCUSSIONS AND CONCLUSIONS

The present state of art of ALB models is summarized in Table 6-1. It includes our own contributions of a heuristic method, back tracking method of assembly line balancing (Chapter IV, Section A), a FORTRAN-IV computer program, *BALB (Appendix) for the heuristic method, and extensions of mixed-model ALB for variable work element times (Chapter V-Section B). Though we have not yet completed the testing of the efficiency of the program, evidences are accumulating that its performance is in par, if not far superior, to that of existing programs. It is expected to be further improved by eliminating phase 1 and separately writing the program for backtracking only. The research on mixed-model ALB has only begun recently. The development of *BALB and the extensions of mixed-model problem indicate two of the several directions for future research. In Chapter V Section C, only the deterministic mixed-model sequencing problems have been discussed. Future research may also extend this to include the variability of work element times and incorporate the learning effect of the operators.

A casual glance at Table 6-1 shows that an almost equal amount of effort seem to have been devoted to the development of analytical and heuristic solution methods for a single-model ALB problem. Because of the computational difficulties associated with the combinatorial nature of the ALB problem, heuristic methods yielding near optimal solutions seem, at least for the present, to outweigh analytical methods in both the practicability and availability of computer programs.

Since the establishment of feasible linear sequences is an important aspect of an ALB problem, the method developed by Okamura and Yamashina (1969) for establishing linear sequences is expected to become a very helpful tool in extending further research in ALB problems.

In these days of rising inflation, the increase of productivity is a very pressing engineering challenge. One solution is to apply systems thinking to production and globally optimize assembly line productions. The mere balancing of production lines may no longer produce adequate increase in productivity. Rather, some of more fundamental questions need be answered. First, why do we need a production conveyor line when there is an alternative of job-shop type production? Second, how susceptible is the cycle time to the discrepancies between the forecasted and actual demands? Third, how does an assembly line interact with the operation of other manufacturing centers in an industry?

There have been a variety of techniques developed to solve the traditional ALB problems. Sarcastically, we may say that there have been "too many ways of cooking the same receipe." The true, and more fundamental, problems may be closely related to, but untouched by, the existing ALB models. To this skepticism, we can only answer that the continued research in related fields is the only approach known today to find a new avenue into a large gain in productivity research. Better and more efficient methods could slowly develop, or suddenly lead to an entirely new view point that may answer some of the fundamental ALB questions.

This thesis is hoped to serve not only as a practical guide to solving immediate ALB problems by offering a digest of techniques currently available but also to act as a milestone and a small guide-post summarizing where we have been and pointing the new directions we can now pursue.

Table 6-1. Present state of art of ALB models.

Approach	Assembly Line Model	Objective Function	Work Element Times	ALB PROCEDURES			Remarks
				Analytical	Computer Application	Heuristic	
				Linear Programming	Programmed (1955)		Involves large computa-
	Mixe	Minimize Dir	Normally Distributed			Extensions of back tracking algorithm on shift time basis (Rao, 1971)	*BALB is used here by inputting the average work content of an operator in a shift as a cycle time
				Extension of variability of work elements to Roberts & Villa models (Rao, 1971)			
		Minimize Sequence Delay Cost	Deterministic		Programmed; no information available	Fixed and variable rate lauching	The software is available at IITRI, Chicago, Ill.

BIBLIOGRAPHY

Arcus, Albert L. 1966. COMSOAL - A computer method of sequencing operations for assembly lines. The International Journal of Production Research 4:259-277.

Barten, Kenneth. 1962. A queueing simulator for determining optimum inventory levels in a sequential process. The Journal of Industrial Engineering 13:245-252.

Bowman, Edward H. 1960. Assembly line balancing by linear programming. Operations Research 8:385-389.

Buffa, Elwood S. 1961a. Modern production management. New York, John Wiley and Sons. 636 p.

______________ 1961b. Pacing effects in production lines. Journal of Industrial Engineering 12:383-386.

______________ 1968. Production inventory systems: planning and control. Homewood, Illinois, Richard D. Irwin. 457 p.

Cauley, John M. 1968. A review of assembly line balancing algorithms. In: Proceedings of the 19th Annual Conference of the American Institute of Industrial Engineers, Tampa, Florida, 1968, New York. p. 223-229.

Cnossen, J. L. 1967. A heuristic approach to assembly line balancing. In: Thirty-Second National Meeting of Operations Research Society of America (ORSA), Chicago, November 1-3, 20 p. (Reprint)

Dantzig, George B. 1951. Maximization of a linear function of variables subject to linear inequalities. In: Activity Analysis of Production and Allocation, ed. by T. C. Koopmans. New York, John Wiley and Sons. (Cited in Bowman, 1960, p. 386).

______________ 1960. On the shortest route through a network. Management Science 6:187-190.

Freeman, Michael C. 1964. The effects of breakdowns and interstorage on production line capacity. The Journal of Industrial Engineering 15:194-200.

Freeman, David R. and J. V. Jucker. 1967. The line balancing problem. The Journal of Industrial Engineering 18:361-364.

Freeman, David R. 1968. A general line balancing model. In: Proceedings of the 19th Annual Conference of the American Institute of Industrial Engineers, Tampa, Florida, 1968, New York. p. 230-235.

Gomory, R. E. 1958. An algorithm for integer solutions to linear programs. Princeton. (IBM Mathematics Research Project, Technical Report No. 1) (Cited in White, 1961, p. 276).

Grant, S. 1962. Computer line balancing of mixed products. Evansville, Indiana, Whirlpool Corporation, Evansville Division, Industrial Engineering Department. (Cited in Sahgal, 1970, p. 21).

Guenther, William C. 1965. Concepts of mathematical statistics, New York, McGraw-Hill. 353 p.

Gutjahr, Allan L. and G. L. Nemhauser. 1964. An algorithm for the line balancing problem. Management Science 2:308-315.

Held, Michael, Richard M. Karp and Richard Shareshian. 1963. Assembly line balancing - dynamic programming with precedence constraints. Operations Research 2:442-459.

Helgeson, W. B. and D. P. Birnie. 1961. Assembly line balancing using the ranked positional weight technique. Journal of Industrial Engineering 12:394-398.

Hoffman, Thomas R. 1959. Generation of permutations and combinations. Madison. (University of Wisconsin. Engineering Experiment Station. Report No. 13). (Cited in Hoffman, 1963, p. 556).

________________ 1963. Assembly line balancing with a precedence matrix. Management Science 9:551-563.

Hu, T. C. 1961. Parallel sequencing and assembly line problems. Operations Research 6:841-848.

Ignall, Edward J. 1965. A review of assembly line balancing. Journal of Industrial Engineering 16:244-254.

Inoue, Michael S. and J. L. Riggs. 1971. Cause and effect diagram: An aid to system identification. Industrial Engineering Journal. In press.

IIT Research Institute. 1970. How to cut manufacturing costs: An introduction to advanced manufacturing methods program. Chicago, 15 p.

Jackson, James R. 1956. A computing procedure for a line balancing problem. Management Science 2:261-271.

Kapur, J. N. and H. C. Saxena. 1961. Mathematical statistics. 2d ed. Delhi, India, S. Chand and Company. 455 p.

Kilbridge, Maurice D. and Leon Wester. 1961. A heuristic method of assembly line balancing. The Journal of Industrial Engineering 12:292-298.

__________ 1961. The balance delay problem. Management Science 8:69-84.

__________ 1962. A review of analytical systems of line balancing. Operations Research 10:626-638.

__________ 1966. An economic model for the division of labor. Management Science 12:255-269.

Klein, Morton, 1963. On assembly line balancing. Operations Research 11:292-298.

Lawler, E. L. and D. E. Wood. 1966. Branch and bound methods: A survey. Operations Research 14:699-719.

Lewin, D. E. 1967. Assembly line balancing and related topics. Master's thesis, Cambridge, Massachusetts Institute of Technology, Sloan School of Management. (Cited in Cauley, 1968, p. 227).

Lehman, Melvin. 1969. On criteria of assigning models to assembly lines. The International Journal of Production Research 7:270-285.

__________ 1969. What's going on in product assembly. Industrial Engineering Journal, April. p 41-46.

Little, John D. C., et al. 1963. An algorithm for the travelling salesman problem. Operations Research 11:972-989.

Mansoor, E. M. 1964a. Assembly line balancing: An improvement on the ranked positional weight technique. The Journal of Industrial Engineering 15:73-74.

Mansoor, E. M. 1964b. Author's comments. The Journal of Industrial Engineering 15:323.

Mansoor, E. M. and S. B. Tuvia. 1966. Optimizing balanced assembly lines. The Journal of Industrial Engineering 17:126-131.

Mastor, A. A. 1966. An experimental investigation and comparitive evaluation of production line balancing techniques. Ph. D. thesis Los Angeles, University of California. 175 numb. leaves.

Moodie, C. L. 1964. A heuristic method of assembly line balancing for assumptions of constant or variable work element times. Ph.D. thesis. Lafayette, Ind., Purdue University, 199 numb. leaves.

______________ 1968. New directions in assembly line balancing. In: Proceedings of 19th Annual Conference of the American Institute of Industrial Engineers, Tampa, Florida, 1968, p. 236-242.

Moodie, C. L. and H. H. Young. 1965. A heuristic method of assembly line balancing for assumptions of constant or variable work element times. The Journal of Industrial Engineering 16:23-29.

Okamura, Kenjira and Hajime Yamashina. 1969. Establishment of linear sequences. Tokyo. Kyoto University. Engineering Faculty Memoirs. 31:307-331. (Reprint).

Prenting, Theodore O. and Robert M. Battaglin. 1964. The precedence diagram: A tool for analysis in assembly line balancing. The Journal of Industrial Engineering 15:208-213.

Ramsing, Kenneth and Donald Downing. 1970. Assembly line balancing with variable element times. Industrial Engineering Journal, January, p 41-43.

Reeve, Richard N. and W. H. Thomas. 1967. A heuristic branch and bound technique for line balancing. In: Thirty-Second National Meeting of the Operations Research Society of America (ORSA), Chicago, November 1-3, 18 p. (Reprint)

Reiter, Raymond. 1969. On assembly line balancing problems. Operations Research 17:685-700.

Riggs, James L. 1970. Production systems: Planning, analysis and control. New York, John Wiley and Sons. 604 p.

Roberts, Stephen D. and Carlos D. Villa. 1970. A multiproduct assembly line - balancing problem. American Institute of Industrial Engineers (AIIE), Transactions 2:361-364.

Sahgal, Vijendra K. 1970. A mixed-model sequencing algorithm. Master's thesis. Mississippi, Mississippi State College, Department of Industrial Engineering. 113 numb. leaves.

Salveson, M. E. 1955. The assembly line balancing problem. The Journal of Industrial Engineering. 6(3) p. 18-25.

Thomopoulos, Nick T. 1966. A sequencing procedure for multi-model assembly lines. Ph. D. thesis. Chicago, Illinois Institute of Technology, Department of Industrial Engineering. 174 numb. leaves.

__________ 1967. Line balancing sequencing for mixed-model assembly. Management Science 14(2):B59-B75.

__________ 1968. Some analytical approaches to assembly line problems. The Production Engineer, July, p. 345-351.

__________ 1969. The mixed-model learning curve. American Institute of Industrial Engineers (AIIE), Transactions. 1:127-132.

__________ 1970. Mixed-model line balancing with smoothed station assignments. Management Science 16:593-603.

Tonge, Fred M. 1960. Summary of a heuristic line balancing procedure. Management Science 7:21-39.

__________ 1961. A heuristic program for assembly line balancing. Englewood Cliffs, New Jersey, Prentice-Hall. 115 p.

Van Gigch, J. P. 1965. The assembly line balancing problem. The case for a general optimizing theory: A review. Berkley, University of California, Graduate School of Business Administration. 26 numb. leaves.

Wester, Leon and M. D. Kilbridge. 1962. Heuristic line balancing - A case. The Journal of Industrial Engineering 13:139-149.

Wester, Leon and M. D. Kilbridge. 1964. The assembly line model-mix sequencing problem. In: Proceedings of the 3rd International Conference on Operations Research, Paris, 1963. Paris, Dunod. p 240-260. (Reprint)

White, William W. 1961. Comments on a paper by Bowman. Operations Research 9:274-276.

APPENDIX

APPENDIX

DETAILS OF *BALB

The program, *BALB, consists of two phases each being programmed separately but using common subroutines. The phase 1 is based on Moodie's successive maximum elemental times and is used to compare the results obtained by phase 2, the back tracking method. The phase 1 tries to find minimal station balance for the given cycle time and if this is not possible the cycle time is incremented in units of 1 till the total delay in all stations of the assembly line is less than or equal to the newly arrived cycle time. Phase 2 attempts to find optimal or near optimal solutions based on positional weights. An option is provided either to run phase 1 and phase 2 or phase 2 only. The inputs to both the phases include the maximum number of rows (M), the maximum number of columns (N) in the IP and IF matrices and the cycle time.

The positional weights calculations are made separately by the program, *ALLPOW listed later in this appendix. The precedence relation matrices are stored in a separate data file which are read by the main program using the CALL EQUIP statement. For mixed-model problems production shift time is inputed as an equivalent cycle time into the program. The data can be either of constant or variable elemental times. The general outline of the two phases and description of the variables in the program is given in the following pages.

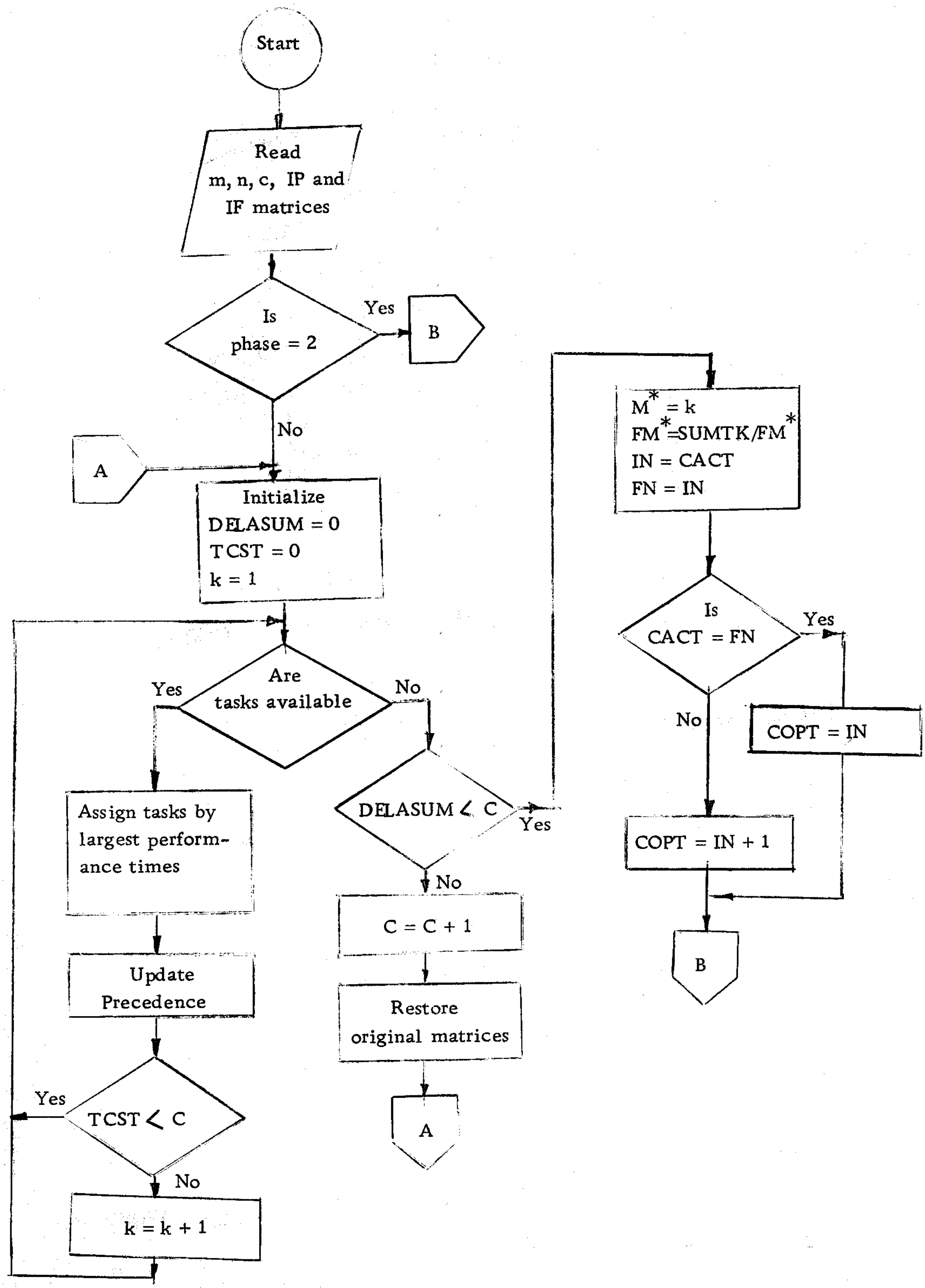


Figure A-1. Flow chart for phase 1 of *BALB.

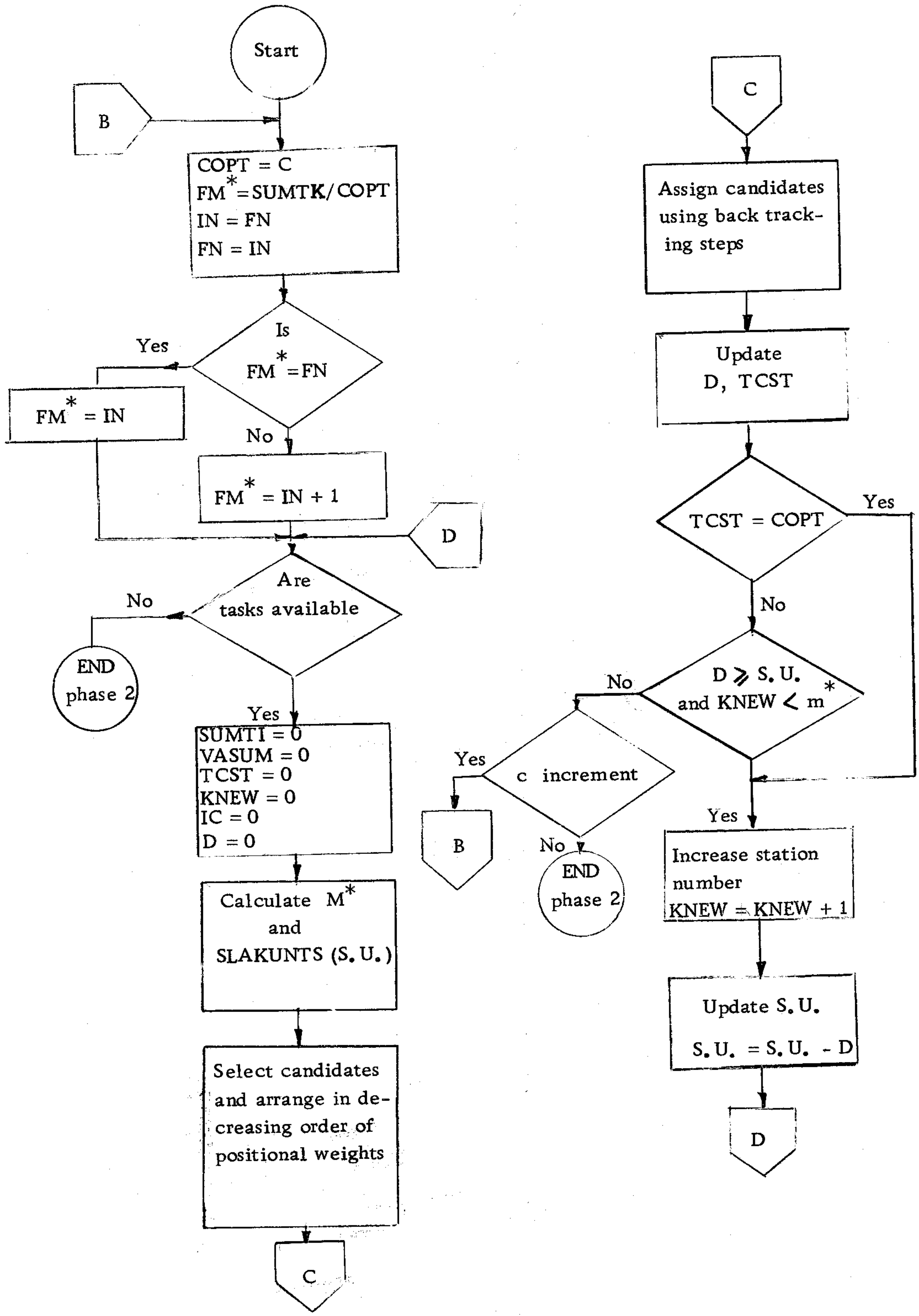


Figure A-2. Flowchart for phase 2 of *BALB.

LEGEND FOR *BALB

Variable	Description
IR	Array for rows of IP and IF matrices
E	Array for elemental times of each element i
IP	IP-matrix
MF	IF-matrix
IPO	IP-matrix copied into IPO
MFO	MF-matrix copied into MFO
ICP	IP-matrix copied into ICP
MAP	IP-matrix copied at the main branch into MAP
CST	Array of cumulative station times
ISR	Rows corresponding to a cumulative station time (CST) when arranged in decreasing order
IRR	Rows which have all zeros in IP-matrix
EE	Array of elemental times for the element numbers of a zero row
PW	Array of positional weights
PWZ	Array of positional weights for the feasible tasks
VAR	Array of variances of i
VARZ	Variances of feasible tasks
SUMTI	Sum of performance times; Σt_i
VASUM	Sum of variances; ΣV_i
TCST	Sum of SUMTI and VASUM
DELASUM	Sum of station delays
DOTI	Decreasing order times
DOVA	Decreasing ordered variances
STIME	Station Time
IC	Running index for main branch nodes
IS	Maximum number of nodes available at main branch
KD	Running index in DOCST subroutine
K	Station number
SLAKUNTS	Slack units in a station
MSTAR	Minimum number of stations (m^*)
CACT	Actual cycle time
COPT	Optimum cycle time
KNEW	Index for station increment
NBK	Index for the nodes assigned in branching
NKB	Indes for the nodes rejected in branching
NBLOK	Array of nodes assigned
NKBLOK	Array of nodes rejected
NBU	Running index in the subbranches

Variable	Description
ESUM	Total work content without variance
SUMTK	Total work content in the assembly process with variance
NBU2	A variable used as a check in main branch nodes
ID	An index to identify the first element assigned
DOCST	Decreasing order cumulative station times
KD	An index in the DOCST subroutine
NU	Element with highest station time
KU	An index used in ZIGZAG subroutine
D	Station delay time
C	Cycle time
M	Number of rows in IP and IF matrices
N	Number of columns in IP and IF matrices
X	A variable equals SUMTI or E(NU)
Y	A variable equals VASUM or VAR(NU)

```
JFIN,*BALB

JLIST
       PROGRAN BALB
       DIMENSION IR(100),E(100),IP(100,10),MF(100,10)
       DIMENSION MAP(100,10)
       DIMENSION IPO(100,10),MFO(100,10)
       DIMENSION ISR(20),ES(20)
       DIMENSION IRR(10),EE(20)
       DIMENSION VAR(100),VARZ(10),CST(20)
       DIMENSION ICP(100,10),NBLOK(20),PW(100),PWZ(10)
       DIMENSION NKBLOK(20)
       DIMENSION IBSR(10),EBS(10)
       PRINT 9999
 9999  FORMAT(' DO YOU WANT PHASE-1&2 OR PHASE-2'
      1' ONLY,PHASE-1&2=12,PHASE-2 ONLY=2')
       PHASE=TTYIN(2H?=)
       IF(PHASE.NE.2)GOTO 9994
       PRINT 9995
       GOTO 9997
 9995  FORMAT(' SOLUTION BY *BALB  ')
 9994  PRINT 1007
 1007  FORMAT(' PHASE-1 AND PHASE-2 SOLUTION OF ALB PROBLEM')
 9997  M=TTYIN(4HROWS,4HUM= )
       N= TTYIN(4HCOLS,4HUM= )
       C=TTYIN(4HCYCL,4HETME,2H= )
       IF(PHASE.EQ.2)COPT=C
C  READ IPAND MF MATRICES
       REWIND 37
C  READ DATA FILES.
       DO100I=1,M
  100  READ(37,99)(IR(I),(IP(I,J),J=1,N),E(I),VAR(I),PW(I),
      1(MF(I,J),J=1,N))
   99  FORMAT(I3,3I3,2F5.1,F6.1,3I3)
       VARSUM=0.
       ESUM=0.
       DO1008I=1,M
       VARSUM=VARSUM+VAR(I)
 1008  ESUM=ESUM+E(I)
       SUMTK=ESUM+SQRT(VARSUM)
       CALL COPY(M,N,IP,IPO,MF,MFO)
C  PRINT OUT THE PROBLEM
       PRINT 1
    1  FORMAT(' E(I)**P-MATRIX****T(I)**VAR(I)**PW(I)**F-MATRIX**
')
       DO102I=1,M
  102  WRITE(61,101)(IR(I),(IP(I,J),J=1,N),E(I),VAR(I),PW(I),
      1(MF(I,J),J=1,N))
  101  FORMAT(I4,3I3,3F8.1,3I3)
       IF(PHASE.EQ.2)GOTO9998
 1006  DELASUM=0.
       SQSUM=0.
       SUMTI=0.
       ID=1
       VASUM=0.
       D=0.
       TCST=0.
       K=1
 1000  CALL ZEROFIND(IP,MF,IR,IRR,E,EE,M,N,NR,VAR,VARZ,PW,PWZ)
       IF(NR-1)1001,1002,1003
 1002  SUMTI=SUMTI+EE(NR)
       VASUM=VASUM+VARZ(NR)
       TCST=SUMTI+SQRT(VASUM)
       IF(TCST.LE.C)GO TO 91
C  INCREASE STATION #
       K=K+1
       SQSUM=SQSUM+D**2
       SUMTI=EE(NR)
       VASUM=VARZ(NR)
       TCST=SUMTI+SQRT(VASUM)
C  NOTE STATION DELAY
       DELASUM=DELASUM+D
   91  NU=IRR(NR)
       IF(ID.GT.1)GOTO 1009
       NFNU=NU
       ID=ID+1
C  BLOCK NU ROW
 1009  DO4J=1,N
    4  IP(NU,J)=9999
       D=C-TCST
       X=EE(NR)
       Y=VARZ(NR)
       WRITE(61,95)K,NU,X,Y,TCST,D
   95  FORMAT(2I5,4F10.2)
       GOTO 1004
 1003  CALL DOCST(IRR,EE,NR,IS,ISR,ES,VARZ,SUMTI,VASUM)
       CALL CHEK(D,IS,ISR,ES,NU,K,TCST,C,DELASUM,SUMTI,VASUM,E,
      1VAR,ID,NFNU,SQSUM)
C  BLOCK NU ROW
       DO5J=1,N
    5  IP(NU,J)=9999
 1004  CALL ZIGZAG(NU,IP,MF,N)
       GOTO 1000
C  CHECK STATION DELAY TIME VS. CYCLE TIME
 1001  DELASUM=DELASUM+D
       SQSUM=SQSUM+D**2
       SI=SQRT(SQSUM)
C  PRINT OUT SMOOTHNESS INDEX
       WRITE(61,9993)SI
 9993  FORMAT(' SMOOTHNESS INDEX IS',F10.2)
       IF(DELASUM.LT.C)GOTO 1005
       C=C+1
       WRITE(61,52)C
   52  FORMAT(' ALTERNATE SOLUTION WHEN C=',F5.)
       CALL COPY(M,N,IPO,IP,MFO,MF)
       GOTO 1006
 1005  MSTAR=K
```

```
      CONTINUE
      FMSTAR=MSTAR
      CACT=SUMTK/FMSTAR
      IN=CACT
      FN=IN
      IF((CACT-FN).EQ.0.)GOTO 35
      COPT=IN+1
      GOTO 50
   35 COPT=IN
   50 SUMTI=0.
      VASUM=0.
      TCST=0.
      SLAKUNTS=FMSTAR*COPT-SUMTK
      KNEW=1
      IC=0
      MSTAR=FMSTAR
      WRITE(61,49)COPT,SLAKUNTS,MSTAR
   49 FORMAT(' CYCLETIME=',F6.1,' SLACK UNITS=',F6.1,' WHEN MSTA
R=',I4)
      CALL COPY(M,N,IPO,IP,MFO,MF)
   37 FORMAT(2I5,4F10.2)
   27 CALL ZEROFIND(IP,MF,IR,IRR,E,EE,M,N,NR,VAR,VARZ,PW,PWZ)
      IF((NR-1))38,39,40
   39 SUMTI=SUMTI+EE(NR)
      VASUM=VASUM+VARZ(NR)
      TCST=SUMTI+SQRT(VASUM)
      D=COPT-TCST
      NU=IRR(NR)
      IF(D.LT.0.)GOTO 32
      IF(D.GT.SLAKUNTS)GOTO 41
      WRITE(61,37)KNEW,NU,EE(NR),VARZ(NR),TCST,D
      SLAKUNTS=SLAKUNTS-D
      GOTO43
   41 WRITE(61,37)KNEW,NU,EE(NR),VARZ(NR),TCST,D
C  BLOCK NU ROW FOR ASSINMENT
   43 DO29J=1,N
   29 IP(NU,J)=9999
      CALL ZIGZAG(NU,IP,MF,N)
      GOTO 27
C  CALLING SUBROUTINE FOR MAIN BRANCH
   40 CALL DEORPWZ(IRR,NR,IS,ISR,ES,PWZ)
      CALL COPY(M,N,IP,MAP,MF,MF)
      IC=1
   28 NU=ISR(IC)
      NBK=0
      NKB=0
   53 SUMTI=SUMTI+E(NU)
      VASUM=VASUM+VAR(NU)
      TCST=SUMTI+SQRT(VASUM)
      D=COPT-TCST
      IF(D.LT.0.)GOTO 44
      IF(D.GT.SLAKUNTS)GOTO45
C CHECK IF NEXT MAIN BRANCH AVAILABLE T(I).LE.D
      IF(IC.EQ.IS)GOTO72
      NU2=ISR(IC+1)
      IF(E(NU2).GT.D)GOTO72
      GOTO45
   72 WRITE(61,37)KNEW,NU,E(NU),VAR(NU),TCST,D
      KNEW=KNEW+1
      SUMTI=0.
      VASUM=0.
      TCST=0.
      SLAKUNTS=SLAKUNTS-D
      DO2J=1,N
    2 IP(NU,J)=9999
      CALL ZIGZAG(NU,IP,MF,N)
C UNBLOK ASSIGNED BRANCH REJECTED ROWS
      DO68I=1,NKB
      DO68J=1,N
      NRJR=NKBLOK(I)
   68 IP(NRJR,J)=ICP(NRJR,J)
      GOTO27
C  PRESERVE EARLIER VALUES
   44 SUMTI=SUMTI-E(NU)
      VASUM=VASUM-VAR(NU)
      IC=IC+1
      IF(IC.GT.IS)GOTO70
      DO66I=1,NBK
      DO66J=1,N
      NL=NBLOK(I)
   66 IP(NL,J)=ICP(NL,J)
      CALL COPY(M,N,MAP,IP,MF,MF)
      GOTO 28
   45 WRITE(61,37)KNEW,NU,E(NU),VAR(NU),TCST,D
      NBK=NBK+1
      NBLOK(NBK)=NU
      DO3J=1,N
      ICP(NU,J)=IP(NU,J)
    3 IP(NU,J)=9999
C  ASSIGN NU ROW
      CALL ZIGZAG(NU,IP,MF,N)
      CALL ZEROFIND(IP,MF,IR,IRR,E,EE,M,N,NR,VAR,VARZ,PW,PWZ)
      IF(NR-1)64,65,46
   64 SUMTI=0.
      VASUM=0.
      TCST=0.
      IC=IC+1
C  UNBLOCK THE BLOCKED ROWS
      DO48I=1,NBK
      DO48J=1,N
      NL=NBLOK(I)
   48 IP(NL,J)=ICP(NL,J)
      CALL COPY(M,N,MAP,IP,MF,MF)
      GOTO28
   65 NU=IRR(NR)
      GOTO53
C  CALL SUBROUTINE FOR SUB-BRANCHING
```

```
   46 CALL DEORPWZ(IRR,NR,IBS,IBSR,EBS,PWZ)
      DO32I=1,IBS
      NBU=IBSR(I)
      NU=ISR(IC)
      SUMTI=SUMTI+E(NBU)
      VASUM=VASUM+VAR(NBU)
      TCST=SUMTI+SQRT(VASUM)
      D=COPT-TCST
      IF(D.LT.0.)GOTO63
      NU=NBU
      IF(D.GT.SLAKUNTS)GOTO45
C  CHECK IF NEXT AVAILABLE ELEMENTAL TIME.LE.D AT THIS STAGE
      IF(I.EQ.IBS)GOTO72
      NBU2=IBSR(I+1)
      IF(NBU2.GT.D)GOTO72
      GOTO45
C  PRESERVE EARLIER VALUES
   63 SUMTI=SUMTI-E(NBU)
      VASUM=VASUM-VAR(NBU)
      NBK=NBK+1
      NBLOK(NBK)=NBU
C  NOTE THE ROWS FOR UNBLOCKING AFTER ASSINGNMENT
      NKB=NKB+1
      NKBLOK(NKB)=NBU
C  BLOCK NU ROW FOR SUB BRANCHES
      DO34J=1,N
      ICP(NBU,J)=IP(NBU,J)
   34 IP(NBU,J)=9999
   32 CONTINUE
      IF(IC.EQ.0)GOTO70
      IC=IC+1
      IF(IC.GT.IS)GOTO70
      DO67I=1,NBK
      DO67J=1,N
      NL=NBLOK(I)
   67 IP(NL,J)=ICP(NL,J)
      SUMTI=0.
      VASUM=0.
      TCST=0.
      CALL COPY(M,N,MAP,IP,MF,MF)
      GOTO 28
   70 PRINT 71
   71 FORMAT(' PHASE-2 CAN NOT BE APPLIED TO THIS RESULT')
      PRINT 54
   54 FORMAT(' DO YOU WANT TO TRY AN OPTIONAL CYCLE TIME FOR',
     1' PHASE-2?, YES=1,NO=0')
      OPTL=TTYIN(2H?=)
      IF(OPTL.EQ.0.)GOTO38
      COPT=TTYIN(4HCPH2,2H= )
 9998 FMSTAR=SUMTK/COPT
      IN=FMSTAR
      FN=IN
      IF((FMSTAR-FN).EQ.0.)GOTO56
      FMSTAR=IN+1
      GOTO50
   56 FMSTAR=IN
      GOTO50
   38 END
C
      SUBROUTINE COPY(M,N,IP,IPO,MF,MFO)
      DIMENSION IP(100,10),IPO(100,10),MF(100,10),MFO(100,10)
      DIMENSION MAP(100,10)
      DO1I=1,M
      DO1J=1,N
      IPO(I,J)=IP(I,J)
    1 MFO(I,J)=MF(I,J)
      RETURN
      END
C
      SUBROUTINE ZEROFIND(IP,MF,IR,IRR,E,EE,M,N,NR,VAR,VARZ,PW,
     1PWZ)
      DIMENSION IP(100,10),MF(100,10),IR(100),E(100),IRR(5),EE(2
     10)
      DIMENSION VAR(100),VARZ(10),PW(100),PWZ(10)
      NR=0
      DO1I=1,M
      DO2J=1,N
      IF(IP(I,J)-0)1,2,1
    2 CONTINUE
      NR=NR+1
      IRR(NR)=IR(I)
      EE(NR)=E(I)
      VARZ(NR)=VAR(I)
      PWZ(NR)=PW(I)
    1 CONTINUE
      RETURN
      END
C
      SUBROUTINE ZIGZAG(NU,IP,MF,N)
      DIMENSION IP(100,10),MF(100,10)
      DO104J=1,N
      IF(MF(NU,J)-0)106,104,106
  106 KU=MF(NU,J)
      DO105JN=1,N
      IF(IP(KU,JN).EQ.NU)IP(KU,JN)=0
  105 CONTINUE
  104 CONTINUE
      RETURN
      END
      SUBROUTINE DOCST(IRR,EE,NR,IS,ISR,ES,VARZ,SUMTI,VASUM)
      DIMENSION IRR(10),EE(100),ISR(20),ES(20),CST(20),VARZ(10)
      IS=0
      BIG=0.
      KD=0
C  CONVERT ELEMENT TIMES TO STATION TIMES
      DO107I=1,NR
      DOTI=SUMTI+EE(I)
```

```
      DOVA=VASUM+VARZ(I)
      STIME=DOTI+SQRT(DOVA)
  107 CST(I)=STIME
C  ARRANGE CUMULATIVE STATION TIME IN DECREASE ORDER
   15 DO12I=1,NR
      IF(BIG.GT.CST(I))GOTO12
      IB=IRR(I)
      IZ=I
      BIG=CST(I)
   12 CONTINUE
      IS=IS+1
      ISR(IS)=IB
      ES(IS)=BIG
      CST(IZ)=0.
      BIG=0.
      KD=KD+1
      IF(KD.LT.NR)GOTO15
      RETURN
      END
C
      SUBROUTINE CHEK(D,IS,ISR,ES,NU,K,TCST,C,DELASUM,SUMTI,
     1VASUM,E,VAR,ID,NFNU,SQSUM)
      DIMENSION ISR(20),ES(20),VAR(100),E(100)
      DO94I=1,IS
      IF(ES(I).LE.C)GOTO 93
   94 CONTINUE
C  INCREASE STATION #
      K=K+1
      SQSUM=SQSUM+D**2
C  NOTE STATION DELAY
      DELASUM=DELASUM+D
C  ASSIGN BIG CANDIDATE TO NEW STATION
      NU=ISR(1)
      IF(ID.GT.1)GOTO 1010
      NFNU=NU
      ID=ID+1
C REVISE  SUMTI
 1010 SUMTI=E(NU)
      VASUM=VAR(NU)
      TCST=SUMTI+SQRT(VASUM)
      X=SUMTI

      Y=VASUM
      GOTO 92
   93 NU=ISR(I)
      IF(ID.GT.1)GOTO95
      NFNU=NU
      ID=ID+1
C  REVISE SUMTI & VASUM
   95 SUMTI=SUMTI+E(NU)
      VASUM=VASUM+VAR(NU)
      TCST=ES(I)
      X=E(NU)
      Y=VAR(NU)
   92 D=C-TCST
      WRITE(61,96)(K,NU,X,Y,TCST,D)
   96 FORMAT(2I5,4F10.2)
      RETURN
      END
C
      SUBROUTINE DEORPWZ(IRR,NR,IS,ISR,ES,PWZ)
      DIMENSION IRR(10),ISR(20),ES(20),PWZ(20)
      DIMENSION IBSR(10),EBS(10)
      IS=0
      KD=0
      BIG=0.
C  ARRANGE POSITIONAL WEIGHTS IN DECREASING ORDER
   15 DO12I=1,NR
      IF(BIG.GT.PWZ(I))GOTO12
      IB=IRR(I)
      IZ=I
      BIG=PWZ(I)
   12 CONTINUE
      IS=IS+1
      ISR(IS)=IB
      ES(IS)=BIG
      PWZ(IZ)=0.
      BIG=0.
      KD=KD+1
      IF(KD.LT.NR)GOTO 15
      RETURN
      END
]
```

[redacted]

```
MARCH 6, 1971  4:13 PM TERMINAL 041

#LOAD,*BALB

#EQUIP,37=*DMIX1
#LOAD,*BALB
RUN
RUN




DO YOU WANT PHASE-1&2 OR PHASE-2 ONLY,PHASE-1&2=12,PHASE-2 ONLY=2
?=  2
SOLUTION BY *BALB
ROWSUM= 17
COLSUM= 3
CYCLETME=    464
E(I)**P-MATRIX****T(I)**VAR(I)**PW(I)**F-MATRIX**
  1  0  0  0     35.0         0   393.0  7  0  0
  2  0  0  0     70.0         0   428.0  7  0  0
  3  0  0  0     90.0         0   658.5  7  8  0
  4  0  0  0     50.0         0   131.0 12  0  0
  5  0  0  0     50.0         0   554.0  9  0  0
  6  0  0  0     70.0         0   574.0  9  0  0
  7  1  2  3     20.0         0   358.0 10 11  0
  8  3  0  0    129.5         0   300.5 11 12  0
  9  5  6  0    122.5         0   504.0 13 14  0
 10  7  0  0    150.0         0   248.0 15  0  0
 11  7  8  0     90.0         0    90.0  0  0  0
 12  4  8  0     25.0         0    81.0 17  0  0
 13  9  0  0    155.0         0   277.5 16  0  0
 14  9  0  0    104.0         0   226.5 16  0  0
 15 10  0  0     98.0         0    98.0  0  0  0
 16 13 14  0     66.5         0   122.5 17  0  0
 17 12 16  0     56.0         0    56.0  0  0  0
CYCLETIME= 464.0 SLACK UNITS=  10.5 WHEN MSTAR=    3
    1     3     90.00          0     90.00     374.00
    1     6     70.00          0    160.00     304.00
    1     5     50.00          0    210.00     254.00
    1     9    122.50          0    332.50     131.50
    1     2     70.00          0    402.50      61.50
    1     1     35.00          0    437.50      26.50
    1     7     20.00          0    457.50       6.50
    2     8    129.50          0    129.50     334.50
    2    13    155.00          0    284.50     179.50
    2    10    150.00          0    434.50      29.50
    2    13    155.00          0    155.00     309.00
    2     8    129.50          0    284.50     179.50
    2    10    150.00          0    434.50      29.50
    2    10    150.00          0    150.00     314.00
    2     8    129.50          0    279.50     184.50
    2    13    155.00          0    434.50      29.50
    2    14    104.00          0    104.00     360.00
    2     8    129.50          0    233.50     230.50
    2    13    155.00          0    388.50      75.50
    2     4     50.00          0    438.50      25.50
    2    12     25.00          0    463.50        .50
    3    10    150.00          0    150.00     314.00
    3    16     66.50          0    216.50     247.50
    3    15     98.00          0    314.50     149.50
    3    11     90.00          0    404.50      59.50
    3    17     56.00          0    460.50       3.50




END OF FORTRAN EXECUTION

#TIME
TIME 6.003 SECONDS    MFBLKS 0  CFBLKS 0
#LOGOFF
TIME 6.117 SECONDS  MFBLKS 0  COST $0.76
```

```
[redacted]
MARCH 6, 1971  4:26 PM TERMINAL Ø41

#EQUIP,37=*DDTAL
#LOAD,*BALB
RUN
RUN

DO YOU WANT PHASE-1&2 OR PHASE-2 ONLY,PHASE-1&2=12,PHASE-2 ONLY=2
?=  12
PHASE-1 AND PHASE-2 SOLUTION OF ALB PROBLEM
ROWSUM= 9
COLSUM= 3
CYCLETME=    1Ø
E(I)**P-MATRIX****T(I)**VAR(I)**PW(I)**F-MATRIX**
  1  Ø  Ø  Ø      4.Ø        Ø      7.4  2  6  Ø
  2  1  Ø  Ø      2.Ø        Ø      5.6  3  5  Ø
  3  2  Ø  Ø      5.Ø        Ø      6.2  4  Ø  Ø
  4  3  Ø  Ø      7.Ø        Ø      5.8  9  Ø  Ø
  5  2  Ø  Ø      5.Ø        Ø      5.3  9  Ø  Ø
  6  1  Ø  Ø      6.Ø        Ø      6.8  7  Ø  Ø
  7  6  Ø  Ø      8.Ø        Ø      6.4  8  Ø  Ø
  8  7  Ø  Ø      3.Ø        Ø      4.5  9  Ø  Ø
  9  4  5  Ø      8.Ø        Ø      3.Ø  Ø  Ø  Ø
   1     1      4.ØØ         Ø       4.ØØ        6.ØØ
   1     6      6.ØØ         Ø      1Ø.ØØ           Ø
   2     7      8.ØØ         Ø       8.ØØ        2.ØØ
   2     2      2.ØØ         Ø      1Ø.ØØ           Ø
   3     5      5.ØØ         Ø       5.ØØ        5.ØØ
   3     3      5.ØØ         Ø      1Ø.ØØ           Ø
   4     4      7.ØØ         Ø       7.ØØ        3.ØØ
   4     8      3.ØØ         Ø      1Ø.ØØ           Ø
   5     9      8.ØØ         Ø       8.ØØ        2.ØØ
SMOOTHNESS INDEX IS       2.ØØ
CYCLETIME=   1Ø.Ø SLACK UNITS=    2.Ø WHEN MSTAR=    5
   1     1      4.ØØ         Ø       4.ØØ        6.ØØ
   1     6      6.ØØ         Ø      1Ø.ØØ           Ø
   2     7      8.ØØ         Ø       8.ØØ        2.ØØ
   2     2      2.ØØ         Ø      1Ø.ØØ           Ø
   3     3      5.ØØ         Ø       5.ØØ        5.ØØ
   3     5      5.ØØ         Ø      1Ø.ØØ           Ø
   4     4      7.ØØ         Ø       7.ØØ        3.ØØ
   4     8      3.ØØ         Ø      1Ø.ØØ           Ø
   5     9      8.ØØ         Ø       8.ØØ        2.ØØ

END OF FORTRAN EXECUTION

#LOGOFF
TIME 4.922 SECONDS   MFBLKS Ø   COST $Ø.58

[redacted]
MARCH 6, 1971  4:32 PM TERMINAL Ø41

#EQUIP,37=*DDTA1
#LOAD,*BALB
RUN
RUN

DO YOU WANT PHASE-1&2 OR PHASE-2 ONLY,PHASE-1&2=12,PHASE-2 ONLY=2
?=  2
SOLUTION BY *BALB
ROWSUM= 9
COLSUM= 3
CYCLETME=    1Ø
E(I)**P-MATRIX****T(I)**VAR(I)**PW(I)**F-MATRIX**
  1  Ø  Ø  Ø      4.Ø        Ø     48.Ø  2  6  Ø
  2  1  Ø  Ø      2.Ø        Ø     27.Ø  3  5  Ø
  3  2  Ø  Ø      5.Ø        Ø     2Ø.Ø  4  Ø  Ø
  4  3  Ø  Ø      7.Ø        Ø     15.Ø  9  Ø  Ø
  5  2  Ø  Ø      5.Ø        Ø     13.Ø  9  Ø  Ø
  6  1  Ø  Ø      6.Ø        Ø     25.Ø  7  Ø  Ø
  7  6  Ø  Ø      8.Ø        Ø     19.Ø  8  Ø  Ø
  8  7  Ø  Ø      3.Ø        Ø     11.Ø  9  Ø  Ø
  9  4  5  Ø      8.Ø        Ø      8.Ø  Ø  Ø  Ø
CYCLETIME=   1Ø.Ø SLACK UNITS=    2.Ø WHEN MSTAR=    5
   1     1      4.ØØ         Ø       4.ØØ        6.ØØ
   1     2      2.ØØ         Ø       6.ØØ        4.ØØ
   1     6      6.ØØ         Ø       6.ØØ        4.ØØ
   1     2      2.ØØ         Ø       8.ØØ        2.ØØ
   2     3      5.ØØ         Ø       5.ØØ        5.ØØ
   2     5      5.ØØ         Ø      1Ø.ØØ           Ø
   3     7      8.ØØ         Ø       8.ØØ        2.ØØ
   3     4      7.ØØ         Ø       7.ØØ        3.ØØ
PHASE-2 CAN NOT BE APPLIED TO THIS RESULT
DO YOU WANT TO TRY AN OPTIONAL CYCLE TIME FOR PHASE-2?, YES=1,NO=Ø
?=  Ø

END OF FORTRAN EXECUTION

#LOGOFF
TIME 4.4Ø8 SECONDS   MFBLKS Ø   COST $Ø.48
```

```
[redacted]
MARCH 6, 1971  3:17 PM TERMINAL 041

#EDIT

]FIN,*ALLPOW

]LIST
C  CALUCULATION OF 3 TYPES OF POSITIONAL WEIGHTS
      PROGRAM ALLPOW
      DIMENSION IFA(100,100)
      DIMENSION T(100)
C  INPUT MAX. # OF TASKS AND TIME OF EACH TASK
      N=TTYIN(4HLIMT,2H= )
      DO100I=1,N
      WRITE(61,101)I
      T(I)=TTYIN(4HTI= )
  101 FORMAT(' ROW',I4)
C  INPUT FOLLOWERS OF EACH TASK(ROW)
C   TYPE ZERO TO END INPUT DATA OF A TASK
  102 MF=TTYIN(4HIF= )
      IF(MF.EQ.0)GOTO100
      IFA(I,MF)=1
      GOTO102
  100 CONTINUE
C  ERROR CHECK FOR INPUT DATA
  109 PRINT 105
  105 FORMAT(' ANY CHANGES IN DATA? YES=1,NO=0')
      IERROR=TTYIN(2H?=)
      IF(IERROR.EQ.0)GOTO106
      I=TTYIN(4HROW=)
      DO108J=I,N
  108 IFA(I,J)=0
      T(I)=TTYIN(4HTI= )
  110 MF=TTYIN(4HIF= )
      IF(MF.EQ.0)GOTO109
      IFA(I,MF)=1
      GOTO110
C   PRINT OUT TASK TIMES AND DIAGONAL PRECEDENCE MATRIX
  106 DO104I=1,N
  104 WRITE(61,103)(T(I),(IFA(I,J),J=1,N))
C LINPOWS
      PRINT 90
   90 FORMAT(' VALUES OF LIN POWS')
      DO91I=1,N
      TSUM=T(I)
      DO92J=1,N
      IF(IFA(I,J).LE.0)GOTO92
      TSUM=TSUM+T(J)
   92 CONTINUE
      WRITE(61,93)I,TSUM
   91 CONTINUE
C LOG POWS
      PRINT 80
   80 FORMAT(' VALUES OF LOG POWS')
      DO81I=1,N
      TSUM2=0.
      TSUM1=0.
      DO82J=1,N
      IF(IFA(I,J).LE.0)GOTO82
      TSUM1=TSUM1+T(J)
   82 CONTINUE
      IF(TSUM1.EQ.0)GOTO83
      TSUM2=1.443*(ALOG(TSUM1))
   83 TSUM=ALOG(T(I))*(1.443)+(TSUM2)
      WRITE(61,94)I,TSUM
   81 CONTINUE
C  SQUARE POWS
      PRINT 70
   70 FORMAT(' VALUES OF SQUARE POWS')
      DO71I=1,N
      TSUM3=0.
      DO72J=1,N
      IF(IFA(I,J).LE.0)GOTO72
      TSUM3=TSUM3+T(J)**2
   72 CONTINUE
      TSUM=T(I)+SQRT(TSUM3)
      WRITE(61,93)I,TSUM
   71 CONTINUE
  103 FORMAT(F6.1,100I2)
   93 FORMAT(' PWT OF TASK',I3,' IS',F10.3)
   94 FORMAT(' PWT(IN BITS) OF TASK',I3,' IS',F10.3)
      END

]
```

```
[redacted]
MARCH 6, 1971  3:24 PM TERMINAL Ø41

#FORTRAN,I=*ALLPOW,R

 NO ERRORS FOR ALLPOW
RUN



LIMT=    4
ROW    1
TI= 5
IF= 2
IF= 3
IF= 4
IF= Ø
ROW    2
TI= 2
IF= 4
IF= 3
IF= Ø
ROW    3
TI= 1Ø
IF= 4
IF= Ø
ROW    4
TI= 2Ø
IF= Ø
ANY CHANGES IN DATA? YES=1,NO=Ø
?=  Ø
  5.Ø Ø 1 1 1
  2.Ø Ø Ø 1 1
 1Ø.Ø Ø Ø Ø 1
 2Ø.Ø Ø Ø Ø Ø
VALUES OF LIN POWS
PWT OF TASK  1 IS     37.ØØØ
PWT OF TASK  2 IS     32.ØØØ
PWT OF TASK  3 IS     3Ø.ØØØ
PWT OF TASK  4 IS     2Ø.ØØØ
VALUES OF LOG POWS
PWT(IN BITS) OF TASK  1 IS      7.323
PWT(IN BITS) OF TASK  2 IS      5.9Ø8
PWT(IN BITS) OF TASK  3 IS      7.645
PWT(IN BITS) OF TASK  4 IS      4.323
VALUES OF SQUARE POWS
PWT OF TASK  1 IS     27.45Ø
PWT OF TASK  2 IS     24.361
PWT OF TASK  3 IS     3Ø.ØØØ
PWT OF TASK  4 IS     2Ø.ØØØ



END OF FORTRAN EXECUTION

#LOGOFF
TIME 4.2Ø3 SECONDS  MFBLKS 4  COST $Ø.42
```