1968

An investigation of the feasibility of applying linear progamming to assembly line balancing

Walter P. Mydlow
Lehigh University

AN INVESTIGATION OF THE FEASIBILITY OF APPLYING LINEAR PROGRAMMING TO ASSEMBLY LINE BALANCING

by

Walter P. Mydlow

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in Industrial Engineering

Lehigh University

1968

CERTIFICATE OF APPROVAL

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

May 17, 1968
Date

Wallace J. Richardson
Professor in Charge

G. E. Kane
Chairman of the Department of Industrial Engineering

ACKNOWLEDGEMENTS

The author wishes to thank Professor W. J. Richardson of the Industrial Engineering Department, Lehigh University, for his generous devotion of time and excellent guidance during the counseling of this thesis.

The assistance received from Mr. S. D. Hester of the Western Electric Company, Inc., is also appreciated. The author also extends thanks to the Western Electric Company, Inc., whose sponsorship has made this study possible.

TABLE OF CONTENTS

ABSTRACT

An investigation of the feasibility of applying linear programming to assembly line balancing problems is presented. Salveson and Bowman developed the mathematical models. Bowman presented two formulations, one of which was improved upon by White.

The models used in this investigation were Bowman's second and White's improvement of Bowman's first linear programming formulation. Testing the models required the use of integer linear algorithms. Of the many integer codes available, the Direct Search Zero-One Integer-Program and Integer-Program-Western Electric were selected to perform the computations.

Based on the experience reported the approach proved to work. However, the method is not practical since it requires excessive amounts of computer processing time.

I INTRODUCTION

The first progressive (production) assembly line as a modern manufacturing technique is generally credited to Henry Ford's well known application at his Highland Park Plant. In 1944 Muther (23) wrote

> "Line production is a method of manufacture of an arrangement of work areas where the material moves continuously and at a uniform rate through a sequence of balanced operations which permits of simultaneous performance throughout, the work progressing toward completion along reasonably direct path."

Inherent in line production as noted by Muther, is the concept of balanced operations, and he presented a systematic procedure to facilitate the balancing operation. Although the method of line production was introduced in 1913, the first published report describing an analytical method applied to assembly line balancing appeared in 1954. Since 1954 many analytical and heuristic techniques for assembly line balancing have been developed and published. These methods can be classified as either exhaustive analytical enumeration solutions, which provide optimal solutions or heuristic solutions, which provide a feasible but not necessarily an optimal solution.

Several of the techniques are considered to be computationally feasible when implemented by use of a computer, while two methods are characterized by their authors, Bowman (1), and Klien (15) as being of academic interest only.

The known exceptions to the above are:

1. Bowman proposed two separate integer linear programming models, one of which was improved upon by White (28) who

transformed Bowman's first formulation into a zero-one integer programming problem. Bowman's second formulation partially parallels the model developed by Manne (17) which is an integer linear program.

2. Klien gives an analytical method of balancing an assembly line when the order of operations is specified. His method involves determining the minimum balance delay for a series of cycle times. Klien makes the remark that his method will probably suffer from the same defect which plagues all mathematically-based methods of line balancing; the required enumeration of all feasible orderings will be impractical for large problems.

Bowman's second formulation appears superior to the first because it requires less constraint equations and fewer variables including special integer variables. Although White improved on Bowman's first formulation, he concluded that Bowman's second formulation seems superior to the first. There have been no published reports of any application of assembly line balancing using linear programming. Various writers have commented that sample problems and their computations may provide experience from which some generalization can be made. The objective of this study was to investigate the feasibility of assembly line balancing using the linear programming approach, and to provide the experience from which some conclusions can be made.

II NATURE OF THE LINE BALANCING PROBLEM

An assembly line in the sense used in this paper will agree generally with the description given by Muther for line production. The line is considered to consist of a series of work positions (stations) at each of which an operator performs a portion of the total assembly work on a particular product. The transfer of the partially assembled product between work stations is accomplished by means of a moving conveyor. The sequence in which the operations may be performed is usually subject to constraints imposed by the product design and the process technology.

A search of the literature on assembly line balancing indicates that the major problems of assembly line manufacture is that of balancing the work load among the various operators. The generally accepted definition of an optimum solution to this problem is the one attributed to Salveson (25): "To minimize the total amount of idle time or equivalently to minimize the number of operators to do a given amount of work for a given conveyor belt speed."

Definitions, associated with general line balancing are the following:

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

2. Station: A location on the assembly line where a given amount of work is performed by an operator, "station" and "operator" may sometimes be used interchangeably. The symbol K is used to

represent a station such that $1 \leq k \leq K$ where K = maximum number.

3. Total Work Content: The total amount of work, measured in time units, required to completely assemble one unit of product, and it is the same whether one or many operators are involved in the assembly.

4. Work Element: A rational division of total work content in an assembly process. Usually it involves the completion of a minor task and is represented by the symbol E_i where i is the element identification number and has the range $1 \leq i \leq N$. N is the total number of work elements.

5. Cycle Time: The amount of time a unit of product being assembled is normally available to an operator performing his assigned tasks. The cycle time is usually pre-determined by management in order to achieve a desired productive output within a given period of time. The cycle time for an assembly is

$$C = \frac{T}{q}$$

where q is the number of units of the product to be produced and T is the number of units of time in the period.

6. Station Time: The actual amount of work, in time units, assigned to a specific station on the assembly line. The total work content of any station cannot exceed the cycle time and will not be less than the maximum-length work element assigned to that station. The symbol S_k represents the total time assigned to the k^{th} station. The following relationship restricts station time

$$\text{MAX } E_i(k) \leq S_k \leq C$$

The S_k's will be equal for all stations only if $\sum_{i=1}^{N} E_i/K$ is an integer, and a perfect balance has been achieved.

7. Balance Delay: The amount of idle time for the entire assembly line due to unequal values of S_k. In a perfectly balanced assembly line where $S = S_k = C$, there exists no balance delay.

8. Procedure Diagram: A graphical description of any ordering in which work elements must be performed in achieving the total assembly of the product. The following diagram for a nine-element assembly process, is taken from Hoffman

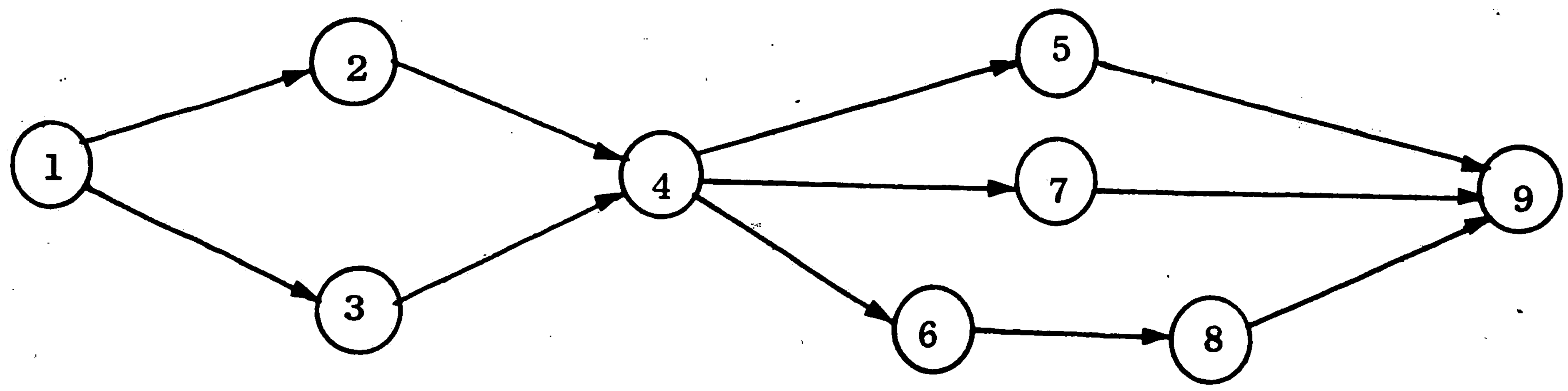


FIGURE 1

An element to the right of another element and connected to it by a line cannot be performed before that element.

9. Precedence Matrix: A precedence matrix is a square matrix which shows the precedence diagram to every other element in the precedence diagram. It has a row (i) and column (j) for each element, and each cell, represented by a row-column intersection can contain a (+1) for must precede, a (-1) for must follow, and a (0) for no relationship. The following matrix shows the precedence relationship of the preceding nine-element problem.

i/j	1	2	3	4	5	6	7	8	9
1	0	1	1	1	1	1	1	1	1
2	-1	0	1	1	1	1	1	1	1
3	-1	-1	0	1	1	1	1	1	1
4	-1	-1	-1	0	1	1	1	1	1
5	-1	-1	-1	-1	0	1	1	1	1
6	-1	-1	-1	-1	-1	0	1	1	1
7	-1	-1	-1	-1	-1	-1	0	1	1
8	-1	-1	-1	-1	-1	-1	-1	0	1
0	-1	-1	-1	-1	-1	-1	-1	-1	0

FIGURE 2

The precedence constraints impose restrictions on the line balancing problem. These restrictions are of three types:

1. Restriction on the order in which the piece parts or subassemblies may be assembled which are inherent in the design of the product.
2. Restrictions imposed by the position of fixed facilities on the assembly line.
3. Restrictions imposed by the need to assemble a part from the front or back or top or bottom - usually associated with the assembly of larger products such as the various major appliances.

Fortified with the definitions associated with assembly line balancing a formulation of the line balancing problem is the following: Given a cycle time c in units of time per unit output a set of tasks

E_i $i = 1,2,\ldots,n$ to be performed according to specified precedence relationships (the latter given by a precedence matrix) and given the processing time t_i of task E_i, it is desired to find the minimum number of groupings of these tasks into stations such that:

1. The total amount of idle time is minimized.
2. The precedence restrictions are not violated.
3. The total of the elemental times assigned to and one work station does not exceed the cycle time being considered.

This is not the only possible formulation of the line balancing problem. Here is another formulation: given the above information, minimize the cycle time c for a given number of stations K subject to the precedence requirements and to the condition that the total time at any station does not exceed c. Moodie (21) in his second phase and Tonge (26) attempt to do so. Other variations of the problem which have been considered less frequently include:

1. Variability of time values for the elemental operations.
2. Operator learning time and performance in relation to cycle time.

Killbridge and Webster (14) have investigated various aspects of industrial learning cost as related to cycle time: and Moodie (21) has proposed a model which considers the variability of elemental time values. Mansoor and Ben-Tuvia (20) analyzed two models for improving the efficiency of balanced lines by taking into account the variability of task times. Both systems aim at minimum labor costs per unit.

Assembly Line Balancing Techniques

One of the earliest attempts at solving the assembly line balancing problem as defined by Salveson was by Bryton (2), in his Master's thesis at Northwestern University entitled, "Balancing of a Continuous Production Line." Bryton starts with a feasible arrangement of the work elements for a given, fixed number of work stations. Briefly stated, the procedure is as follows:

1. Select the work stations having the maximum and minimum total work times.

2. Subject to any technological restrictions, select for interchange an element from each of the chosen work stations such that the difference of the element times is nearest to one-half the difference of the total work station times. Make the interchange.

   a. If no interchange will reduce the difference, select the next ranked minimum total work station time, and examine this relative to the maximum. Interchange as above.

   b. Repeat as above until all stations have been examined relative to the maximum.

3. Re-examine all total work station times after an interchange and again select a maximum and a minimum.

4. Interchange elements by Rule 2.

5. Continue smoothing the difference in station times until further interchanges yield no improvement.

Since this method assumes a fixed number of work stations, the line balancing problem is solved by minimizing the maximum work station time. Most of the subsequent methods will be based on the minimum number of work stations for a given cycle time. Salveson (25) proposed an approach to this problem in which the cycle time was fixed and the number of stations a variable. His definition of cycle time is a function of the production volume:

$$c = \frac{\text{Time in production period}}{\text{Production volume}}$$

He defines the minimum number of stations for an assembly line as the smallest positive integer out of n integers where n is greater than or equal to the sum of the elemental times (E_i) divided by the cycle time

$$K_{min} = \min_{n} \left\{ \underset{n}{\text{integers}} \;\middle|\; n \geq \frac{\sum E_i}{c} \right\}$$

The method suggests a linear programming type model encompassing all possible combinations of element assignments to various work stations. In this formulation, Salveson describes his suggested procedure generally as follows:

Let $B = (b_{ij})$ be a matrix of 0 or 1 and indicating 0 if an element is not assigned to j^{th} column (vector); and, 1 if it is. Each column vector is then a combination of certain specified elements and as such may be considered as a possible work station. Each such combination will have a characteristic idle time denoted by,

$$dj = c - \sum b_{ij} a_i \quad j = 1, 2, \ldots, n$$

where a_i indicates the units of time required to carry the i^{th}

elemental task.

The problem can then be stated as:

Minimize $\sum x_j d_j$

subject to:

1. $\sum x_j \; b_{ij} = 1, \; i = 1,2,\ldots,I$
2. $0 \leq x_j \leq 1, \quad j - 1,2,\ldots,n$
3. $0 \leq \sum b_{ij} \; a_{ij} \leq c, \; j = 1,2,\ldots,n$

The restrictions define a convex subset of an n-dimensional space, each point in which is specified by an n-tuple $(x_1, x_2,\ldots,x_n)$. The extremes of the subset are those n-tuples in which:

$$x = \begin{matrix} 0 \\ 1 \end{matrix} \qquad j = 1, 2,\ldots,n$$

It is then desired to find at least one extreme for which $\sum x_j d_j$ is minimized. However, Salveson points out that, "in order to obtain this convenient mathematical model, one must use a pre-enumerated matrix of feasible combinations which can be enormously large, although due to precedence and cycle time restrictions, the number often may be considerably reduced."

By considering the problem as one in a discrete variable, rather than a continuous variable, Salveson then presents a procedure which substitutes combinatorial analysis for the matrix inversion routine of the linear programming formulation. The objective of minimizing idle time remains the same.

In this procedure, a matrix B is enumerated as before subject to the restrictions that

$$0 \geq b_{ij} \; a_i \geq c \qquad j = 1,\ldots,n$$

and such that the precedence relations are not violated. The associated idle time, d_j, for each combination is computed, where

$$d_j = c - \sum_i b_{ij} a_i \qquad j = 1,\ldots,n$$

An initial feasible solution is generated by selecting from matrix B combinations having the lowest value of d_j and not containing any element already selected. From this initial solution, a systematic procedure is given for repeated regrouping of the elements to find the combination giving the best balance.

Although Salveson's contribution is not considered practical for actual application, it has been of great academic usefulness in giving a clear statement of the problem and defining much of the line balancing nomenclature which is generally used today. The combinatorial magnitude of a line-balancing problem can be appreciated through study of Salveson's method of solution.

Jackson (11) developed an algorithm which optimally solves the line balancing problem as defined by Salveson; his method involves examination of all possible work station element combinations by exhaustive enumeration. As the solution progresses, there is a systematic elimination of the less valuable alternatives. The rules for the step by step enumeration, as taken from Jackson are the following:

Given a sequence X(1),..., X(n-1), where each X(i) is a set of elements, the "collection of next assignments after X(1),..., X(n-1)" is the collection of sets of elements as follows:

1. Delete from the precedence graph of the line-balancing problem all operations included in the sequence X(1), ..., X(n-1), and all precedence lines emanating from these elements.

2. List all sets X of elements on the graph of Step 1 such that:
   a. If a given element is an X, so is every element from which a precedence line leads to the given element.
   b. The sum of the performance times for the elements in X is not greater than the upper limit on the cycle time.
   c. No operation can be added to X without violating a or b.

3. (Can be omitted, but often at the cost of substantially enlarged enumeration.) Successively cross off the list of step 2 sets X for which there is another set Y on the list (still not crossed off), such that:
   a. There is just one element x in X which is not also in Y.
   b. There is some element y in Y, which is not in X, which has a performance time at least as great as that of x, and such that precedence lines can be followed from y direct to any element Z from which there is a direct precedence from x to z.

When no further sets can be crossed off by step 3, the computations are completed. In review, the method begins with step 1 and is completed when it is first found that the collection of next assignments after a sequence of sets of elements consists of the set of all elements not included in the sequence. From here the solution to the line balancing problem is obtained by assigning the sets of the sequence to successive stations and the remaining elements to the station immediately after that to which the last set in sequence is assigned. Jackson provides proofs that the method is exhaustive and will, therefore, find an optimal solution.

The main objection to Jackson's algorithm is the amount of computational effort needed to solve large problems; however, his method is the basis for the most widely used analytic system of line balancing used in industry today.

Subsequent to the publication of Jackson's procedure, a modification was suggested by Helgeson and Kwo (8) which, for equivalent total idle time, is designed to minimize the station to station variation in work assignment time. This later criterion is more psychological than economic but it might be important for good employee relations in actual applications.

Held, et. al. (6) presented a dynamic programming approach to assembly line balancing which they show as being computationally feasible for problems of limited size on existing computers. For large problems the algorithm has been supplemented by a successive approximations technique. For the approximation procedure, they indicate an experimental program for the IBM-7090 as having a two minute running time on a sample 111- element problem.

Hoffman (10) in a University of Wisconsin Ph. D. thesis presented an enumeration method for the assembly line balancing problem which uses a precedence matrix to generate all feasible assignments of elements which do not exceed a given cycle time to the assembly station. The matrix, as defined by Hoffman, contains a (1) for each ij entry where element Ei precedes element Ej; all other entries are equal to zero.

Hoffman calls the procedure a "successive maximum elemental time" technique. The search technique used in selecting an element for possible assignment to a work station is based on column sums developed from the precedence matrix which in effect gives a positional weighting to the elements. Assignments are made and cumulative station time compared to the cycle time in much the same way as the procedure suggested by Helgeson and Birnie (7). Hand calculation by this method can be extremely tedious, however, the method is suitable for computer application and Hoffman has developed a Fortran program for that purpose. The program has apparently been used successfully on CDC 1604.

Klien (15) gives an analytical method of line balancing when the order of operations is specified. His method involves determining the minimum balance delay for a series of cycle times, $\underline{C} \leq c \leq \overline{C}$, where $\underline{C}$ is the minimum cycle time ($C = \max Ei$), and $\overline{C}$ is the maximum cycle time acceptable to management. C is considered an integer. He lists the following three steps:

1. Enumerate the set of all feasible orderings of the n elementary operations.
2. Compute an optimal balance for each of these orders overall possible cycle times associated with the order.
3. Select the best balance from (2).

Klien gives a proof for the optimality of his method but does not give an enumeration technique. Computing an optimal balance for several cycle times in order of declining cycle time will result in balances with increasing number of stations, which may be unacceptable if a specific amount of work is to be done in a given amount of time. He makes the remark that his method will probably suffer from the same defect which plagues all mathematically-based methods of line balancing; the required enumeration of all feasible orderings will be impractical for large problems.

Gutjahr and Nemhauser (5) give an algorithm based on finding a shortest route in a finite directed network. They claim that this method represents a considerable improvement over the shortest route approach of Klein. They indicate that there is a close relationship between their algorithm and the dynamic programming approach of Held et. al., but claim a reduction in the necessary computation.

Mansoor (19) suggests an improvement to the algorithm developed by Gutjahr and Nemhauser. The improvement suggested by Mansoor applies to the second stage of the algorithm and generally results in a reduction in the number of computations leading to an optimal solution. Mansoor claims that the improved algorithm is likely to be more efficient than the original because in most instances an optimal solution is generated after considering only a fraction of the shortest route calculations.

One of the first of what can be called the heuristic methods of assembly line balancing was given by Helgeson and Birnie (7) and involves choosing elements for assignment to stations by a ranked positional weight technique. They have considered two formulations of the problem.

1. Minimize the number of work stations for a given cycle time.

2. Minimize the cycle time for a given number of work stations.

The first method is easier to solve, but rarely gives the best balances. If two or more balances are equivalent by either of the above formulations, the preferred balance is the one with the most even distribution of work across the stations.

The technique as applied to the first formulation is described. The first step is to form a precedence matrix of the form described in an earlier section where:

+1 denotes that one element must be performed before another element.

-1 denotes one element must not be performed before another element has been performed.

0 denotes "does not matter."

The next step is to calculate the positional weight for each work element and it is equal to its time plus the sum of the times of all elements which must follow it. The elements are assigned in order of declining positional weight, hence precedence restrictions are not violated.

Since the positional weights are listed in declining order, the associated ordering of elements represents a permutation of the elements in which the precedence restrictions are observed. The elements can be assigned to consecutive stations on the assembly line in that order. If, during the assignment of elements to stations, an assignment of less idle time can be obtained by assigning an element of a lower positional weight than the one indicated by the original permutation, this assignment must be made if precedence is not violated. Logical decision rules are given for the element assignment.

The second formulation of the problem is actually a continuation of the first. The line is balanced for a given cycle time as above. A second problem is then solved in like manner using a cycle time which is one time increment smaller than the limiting station time. Successive iterations in the same manner will lead to a minimum cycle time for a given number of work stations.

Mansoor (18) suggests an "improvement on the ranked positional weight technique" of Helgeson and Birnie. He states that the method described by Helgeson and Birnie does not always lead to an optimal solution, although, with slight modifications to the approach, an optimal solution can be obtained. The problem that Mansoor considers is to minimize the operating cycle time for a given number of work stations or, more generally to determine the most efficient operating conditions required to balance a given assembly line.

The theoretical cycle time is given by:

$$C_1 = \frac{T}{K}$$

$$= p + \frac{S}{K}$$

where

T = total work content of the assembly in units of time

K = number of work stations

and S is the remainder term.

If $S \neq 0$, C_1 must be rounded upward to the next integer and a perfect balance is not possible; i.e., there will be a certain amount of "slack."

Mansoor makes use of this slack time to evaluate each total station assignment as it is made. No combination is permitted which has unassigned time greater than the total remaining slack units. If no solution is found, the cycle time is increased by one unit and the procedure repeated.

Starting with a value of C_1 for the particular K under consideration, the method is in other details essentially the same as that proposed by Helgeson and Birnie. However, it appears that Mansoor's modification will find assignment combinations that might not have been found by the original method. Apparently, no attempt has been made to program the method for a computer.

A heuristic method which slightly predates the Helgeson-Birnie method is that proposed by Kilbridge and Webster (13). Their method, which they claim to be practical for paper and pencil balancing of assembly lines of realistic size utilizes the permutability of elements within the precedence diagram (without violating precedence restrictions), along with human judgement to obtain good feasible, if not optimal, balances. Where other methods rely on manipulation of the precedence matrix, this one emphasizes analysis based on the precedence diagram. The Kildridge-Webster method slightly modifies the precedence diagram by aligning all elements which have no ordering relationships among themselves, under specific columns. Thus, the precedence diagram would be presented in the form illustrated by Figure 3 if its solution were to be derived by the Kildridge-Webster method.

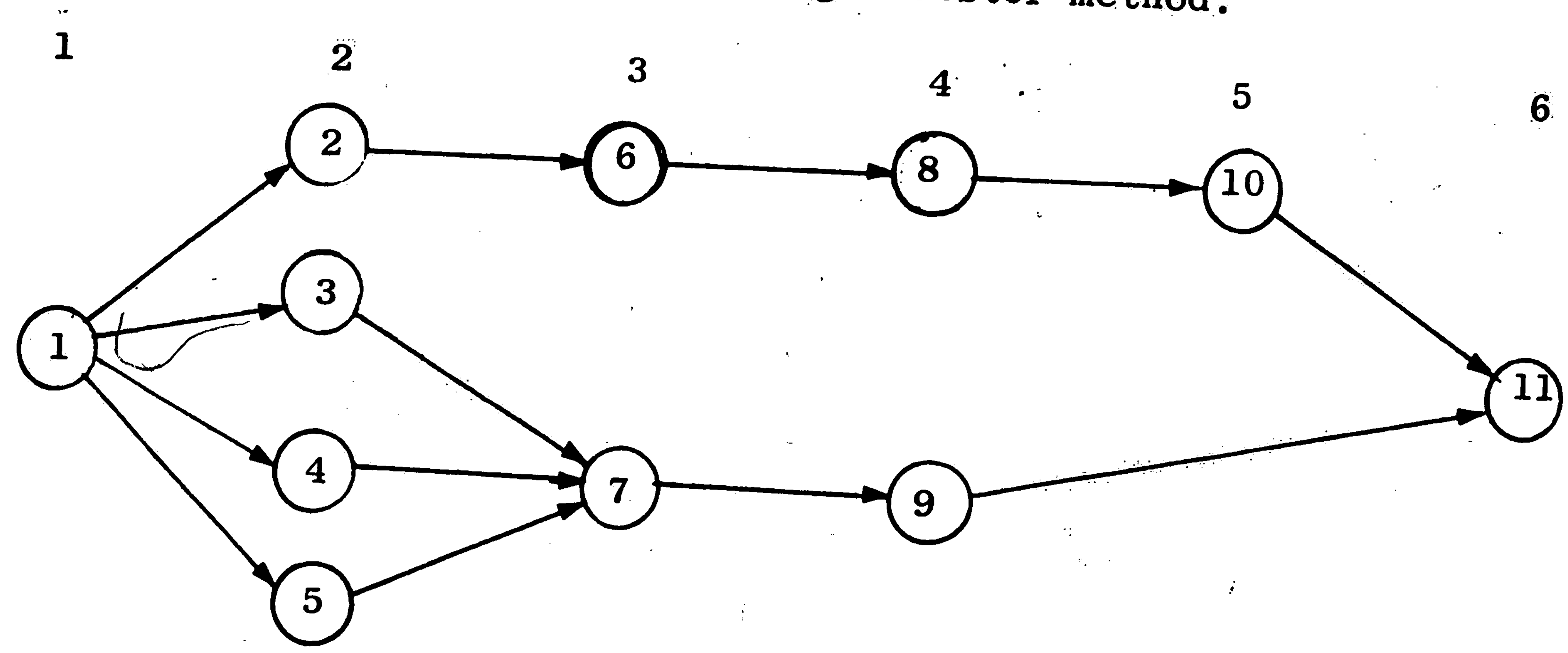


FIGURE 3

The characteristics of the above diagram can now be observed. The columns of the diagram shown in Figure 3 contain elements which are mutually independent and which can be permuted, within a given column. In addition some elements can be moved laterally from left to right without violating precedence. It is these properties of permutability within columns and lateral transferability that are used by Kilbridge and Webster to obtain a good balance.

The procedure is to set up a tabular representation of the precedence diagram as shown in Figure 4.

Column Number	Element Number	Columns Task Can Move To	Element Time	Sum	Cumulative Sum
1	1		6	6	6
2	2		2		
	3	Can go to column 3	5		
	4	If 7 and 8 move to 4 and 5	7		
	5		1	15	21
3	6		2		
	7	Can go to column 4 if 9 moves to 5			
4	8		6		
	9	Can go to column 5	5	11	37
5	10		5	5	42
6	11		4	4	46

Figure 4

Tonge (26) presented a "Heuristic Line Balancing Procedure". In the introduction he states, "Because this approach does not guarantee an optimal solution, the ultimate measure of a heuristic program is whether it provides better solutions more quickly and/or less expensively than other methods".

Tonge's procedure consists of three phases:

Phase I - repeated simplification of the initial problem by grouping adjacent elemental tasks into compound tasks.

Phase II - solution of the simpler problems through created by assigning the compound tasks to work stations. The compound tasks are broken up only when assignment is otherwise not possible.

Phase III - smoothing the resulting balance by transferring tasks among work stations until the distribution of assigned time is as even as possible.

In order to form aggregations of elements, Tonge develops the concepts of a "chain" and a "set". 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, can be replaced by a single compound element called a "chain". 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".

The concept is illustrated below

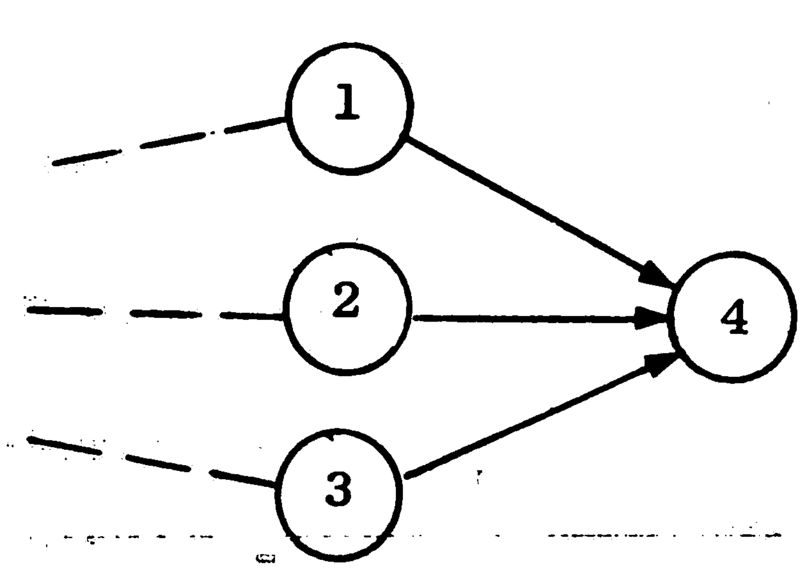


FIGURE 5

In Figure 5 the elements 1, 2 and 3 can be replaced by a set 'S2'. 'S2' and element 4 can be replaced by a chain 'C3'. The time requirement for a compound element is the sum of the times of its components.

To further facilitate the combining of elements, Tonge introduces the 'Z'. This is a group of four elements with the two front elements having common predecessors and the 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 other front elements are the back elements. The back elements have no other direct predecessors. See Figure 6

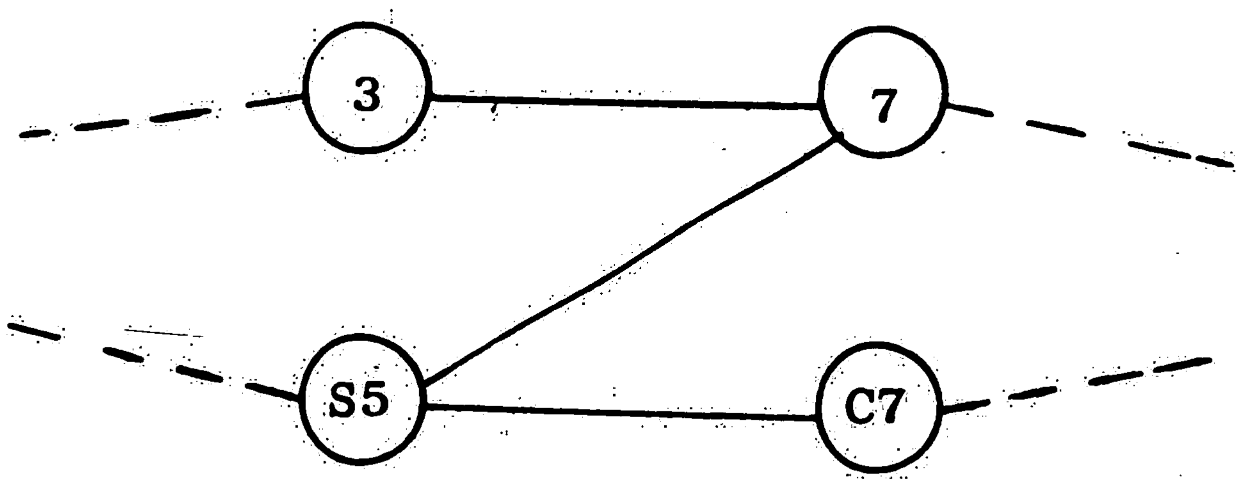


FIGURE 6

Moodie and Young (22) have developed a two-phase heuristic procedure for balancing lines. In the first phase a preliminary balance is obtained using the "largest candidate rule": Construct work stations sequentially by, at each stage, selecting from those tasks that are feasible and will fit the current station, the one with the largest performance time.

In the second phase as in Tonge's Phase III, heuristics are used to shift tasks between stations in attempting to reduce idle time. The heuristics prescribe a series of trades of single elements between stations, with each trade reducing, or at least not increasing, cycle time.

Moodie and Young's procedure allows task performance times to be variable. They assume that the times for n tasks are independent, normally distributed random variables with known means and variances. The criterion for a station being filled up becomes the probability that the sum of the performance times of the tasks in that station be greater than cycle time is less than (say) 2%. The second phase now also attempts to equalize the variances between stations.

Because of the growing availability of computers to most manufacturing concerns and the extreme rapidity with which they can perform mathematical computations or perform logical decisions in comparison to manual methods, much of the model development has been directed toward computer utilization. Exhaustive algorithms are not suitable for anything but a computer, because of the large number of computations necessary. The heuristic balancing techniques hold promise for manual as well as for computer use.

The initial industrial application of line balancing methods was the implementation of Jackson's algorithm, modified with zoning constraints, by Burgeson and Dawm (4) generally known as the IBM-Westinghouse Program for Computerized Line Balancing. Other companies, especially the appliance industry, have made use of the IBM-Westinghouse Program for their manufacturing and have made various improvements to increase the utility of the basic idea. The heuristic line balancing method is used in several industries, and its application in television assembly has been reported.

III MATHEMATICAL MODELS INVESTIGATED

This investigation was concerned with the application of the linear programming approach to the assembly line balancing problem. The mathematical linear programming models have been developed by Bowman, Salveson, and White. The models chosen for this study were White's modification which transformed Bowman's first linear programming approach into a zero-one integer programming problem and Bowman's second formulation, in which the solution is dependent upon integer solutions to linear programming models. Integer linear programming assures a task's being assigned to one and only one station and provides an optimal solution.

The constraints that will be imposed on the models are as follows:

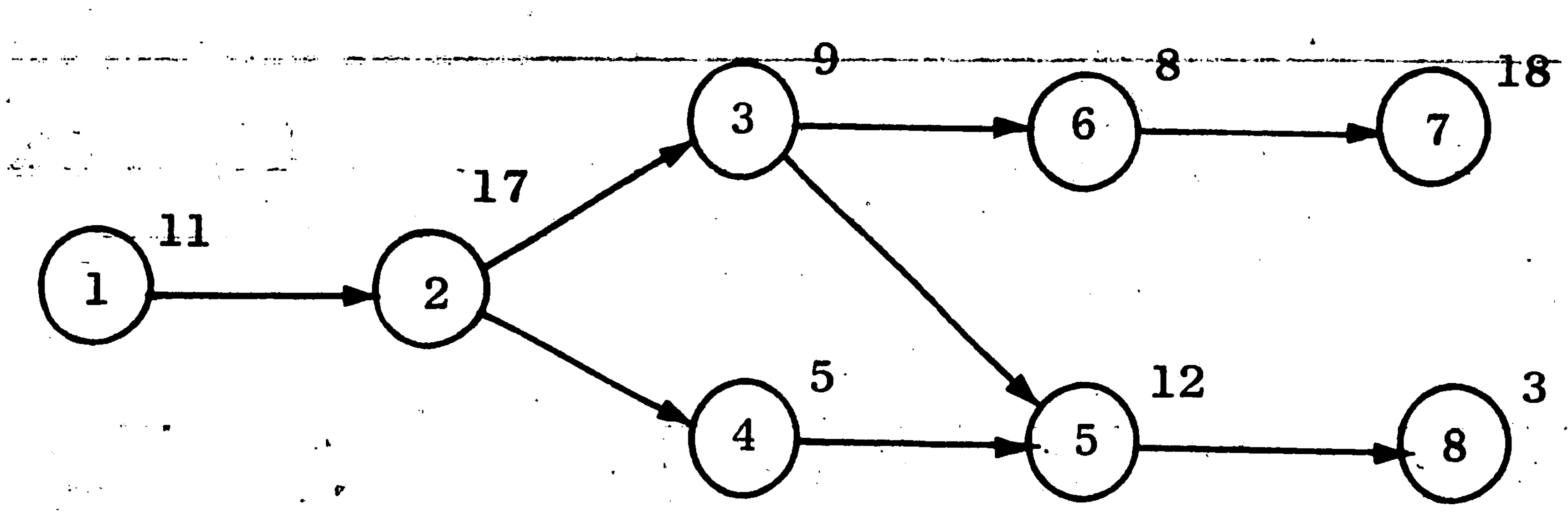
1. Constant work element times (deterministic).
2. Multiple stations.

   Note: Balancing an assembly line to a single station is considered to be the trivial case.

3. Precedence constraints or "must do".

The objective function is to minimize idle time for a given cycle time. Idle time as used here is the amount of idle time for the entire assembly line due to unequal values of station time, S_k.

Consider the following partial ordering constraint schematic:



Bowman's first linear programming formulation is as follows:

The sum of the task time is 75 minutes = $\sum_{j=1}^{8} t_j$ where

t_j = time to complete j^{th} task $j = 1,\ldots,8$. Consider 3 units per hour.

Number of station: $K_{min} = \frac{\sum t_j}{c}$, K must be an integer greater than 1.

Cycle time: c = 20 minutes.

$$K_{min} = \frac{75}{20} \cong 4 \text{ stations}$$

Consider seven stations: $K = 1,2,\ldots,7$.

Define X_{ij} as the amount of time devoted to the j^{th} operation at the i^{th} station

$$i = k = 1, 2,\ldots,7 \qquad j = 1, 2,\ldots,8$$

(1) $\sum_{j=1}^{8} X_{ij} \leq c$ true for all i

The set of constraints (1) assures that none of the stations are overloaded.

(2) $\sum_{i=1}^{7} X_{ij} = t_j$ true for all j

where t_j = time to complete j^{th} task.

This set of constraints (2) assures that each is performed and performed only once.

Define $X_{ij}I$ as an integer variable which must take the values of zero or one.

(3) $1/t_j X_{ij} + X_{ij} I = 1$

for $i = 1, 2,\ldots,7$; $j = 1, 2,\ldots,8$

The set of constraints (3) assures that the operations are not split between stations, and are assigned to only one station.

$$1/17\ X_{12} \leq \frac{1}{11}\ X_{11}$$

(4) $1/17\ X_{22} \leq \frac{1}{11}\ (X_{11} + X_{12})$ ...

Constraint set (4) assures proper ordering; that is the precedence constraints are satisfied.

The objective function:

(5) $\text{Min } Z = 1(X_{57} + X_{58}) + 14(X_{67} + X_{68}) + 196(X_{77} + X_{78})$

makes later stations exceedingly costly, pushing operations as far forward as physically possible. Station 1 through 4 must certainly be used and need assume no cost. The nature of the cost explosion 1, 14, 196 is to make one unit of a later assignment more costly than the sum of the costs of preceding station assignments.

The first linear program for the sample problem uses 135 constraint equations with 56 variables and 56 special integer variables. Slack variables are not counted.

White transformed Bowman's first linear program to a zero-one linear program and thus eliminated Bowman's third set of constraint equations and special integer variables. White's formulation is as follows:

Using the same problem, define $X_{ij} = 1$ if we do operation j at the i^{th} station; $X_{ij} = 0$ otherwise.

(1) $\sum_{j=1}^{8} t_j X_{ij} \leq C$ true for all i

This assures that none of the stations is overloaded.

(2) $\sum_{i=1}^{7} X_{ij} = 1$ true for all j

This assures that each operation is performed.

$X_{12} \leq X_{11}$

(3) $X_{22} \leq X_{11} + X_{21}$

$X_{32} \leq X_{11} + X_{21} + X_{31}$

.
.
.

This assures proper ordering; that is that precedence constraints are satisfied.

The objective function is:

$\min Z = (t_7X_{57} + t_8X_{58}) + 14(t_7X_{67} + t_8X_{68}) + 196(t_7X_{77} + t_8X_{78})$

This linear programming formulation reduced Bowman's first approach from 135 to 71 constraint equations and from 112 variables to 56 variables, not counting slacks. The 56 special integer variables X_{ij}I were eliminated.

Bowman's second formulation which partially parallels the model developed by Manne is described as follows:

Using the same problem define X_i as meaning the clock time when operation i is started. The number of stations are as before and are shown in the figure below.

Station	1	2	3	4	5	6	7
Clock Time	1-20	21-40	41-60	61-80	81-100	101-120	121-140

The stations need not be included explicitly in the formulation.

$$X_1 + 11 \leq X_2;\ X_2 + 17 \leq X_3;\ X_2 + 17 \leq X_4$$

(1)

$$\ldots \quad ;\ X_6 + 8 \leq X_8;\ X_5 + 12 \leq X_7$$

The constraint set (1) assures proper ordering; that is, operation 1 precedes operation 2 as 1's starting clock time is at least 11 minutes processing time before the starting time of operation 2, etc.

To assure that operations do not use the same clock time (non-interference) a new integer valued variable I_{jk} is defined. Operations 3 and 4 are used as potential interfering examples.

(2) $$I_{34} \leq 1$$

(3) $$(140 + 5)I_{34} + (X_3 - X_4) \geq 5$$

(4) $$(140 + 9)(1 - I_{34}) + (X_4 - X_3) \geq 9$$

Bowman treats the constraints (2), (3) and (4) as a single constraint. There are 6 other pairs that have to be considered in the above manner. They are the operation pairs (4,6), (5,6), (6,7), (7,8), (5,8) and (4,8). Constraint (2) assures a value of 1 or 0, and this value determines the precedence of the operations pairs involved. The operation pairs that are included in constraints (2), (3), and (4) are those pairs that are not constrained in the ordering requirement, constraint set (1), which excludes interference.

Define I_i to take any positive integer value (0, 1,...,6)

$$\text{(5)}\quad \begin{array}{ll} X_1 + 11 \le 20I_1 + 20; & X_1 \ge 20I_1 \\ X_2 + 17 \le 20I_2 + 20 & X_2 \ge 20I_2 \\ \vdots & \vdots \\ X_8 + 10 \le 20I_8 + 20 & X_8 \ge 20I_8 \end{array}$$

Constraint set (5) assures that the stations are not overloaded and that the operations be wholly assigned within a station, and simultaneously satisfy both of these requirements. Each station covers only a 20 minute assignment.

Define τ to be the total time, including time required to complete one unit.

(6) $X_7 + 3 \le \tau$, $X_8 + 10 \le \tau$

Constraint set (6) is made up of all the operations with no followers in a required ordering and permits the objective function to be simply

$$\min \quad Z = \tau$$

This formulation uses 33 constraints equations with 8 variables plus 15 special integer variables plus the variable τ which equals 24 variables not counting slacks.

IV SOLUTION PROCEDURES

To obtain an optimal solution to the assembly line balancing problem using Bowman or White's linear programming formulation required the use of an operating integer programming routine. From the operating integer codes that are available the two routines that were selected were the Lemke-Spielberg (30) "Direct Search Zero-One Integer Program," DZIP1, and IPWE (Integer Program-Western Electric). Both routines are written in Fortran.

IPWE and DZIP1 code are intended to solve minimization problems of the linear programming types; however, the DZIP1 routine is subject to the additional constraint that all the variables must take on values 0 or 1 only.

The IPWE code is based upon IPSC (Integer Program-Sandia Corporation) which is the work of Woolsey and Trauth (29). The IPSC was modified by Weingartner (27).

The basic differences from the IPSC are that IPWE includes the learning technique and stores only the non-zero elements of the tableau, thus allowing larger size problems to be attempted. The problem size that IPWE can operate on is determined by the following relationship:

$$d(r + c)(c) \leq 100,000$$

where

r is the number of rows

c is the number of columns, and

d is the ratio of non-zero elements to all elements.

The IPWE program has three pivot selection rules which allow for the selection of any single rule, any combination of the rules or the learning method, a heuristic decision rule, which chooses a combination of the three pivot selection rules through the use of a single control card. These features increased the basic program's speed.

The output of the program includes the optimal value of the objective function, the values of the decision variables, the number of iterations required to solve the problem, the time required to solve the problem and the number of iterations that each pivot selection rule was used during the solution of the problem.

The Lemke-Spielberg DZIP1 computer program is similar to the IPWE routine in the respect that the user has the option of altering strategy and direction of search by means of input parameters. The output provides a sub-optimal feasible solution, the values of the objective function, the values of the decision variables, and the number of iterations required to obtain the sub-optimal feasible solution. At optimality the output includes the iteration number at which the optimal feasible solution was ascertained, the values of the decision variables, and the number of times the preferred strategy was selected.

The authors claim that the algorithm used for this program is considerably superior to that originally proposed by E. Balas (31). Also, computer running times have been 67% to 75% shorter than the Balas scheme.

The DZIP1 program is currently designed for problems up to M = 50 or N = 150 and can be changed for use with problems of greater size M by N by changing the arguments of the dimension statements from 50 to M, from 150 to N and from 151 to N + 1.

V RESULTS OF THE INVESTIGATION

All the computations for this investigation were performed using the IPWE and Lemke-Spielberg codes on the IBM-360/50 system.

The input data for White's modification of Bowman's first linear programming model was prepared according to the specifications given in (27) and (30). A seven work station balance was attempted using IPWE. Two computer runs were made taking 10 minutes and 60 minutes each. An optimal solution was not attained in either run. During the 10 minute computational effort 962 iterations were made while the 60 minute run made 9733 iterations.

Similarly two runs of 10 minutes and 60 minutes duration were made using the Lemke-Spielberg code. An optimal solution was not obtained with either the 10 minute run requiring 963 iterations or the 60 minute run requiring 5782 iterations.

Bowman's second linear programming model for a seven work station balance was investigated using the IPWE code and yielded an optimal solution. The processing time was 71.14 minutes and required 8076 iterations to reach optimality. The solution obtained was verified by using Moodie's and Young's heuristic method of assembly line balancing.

White's method was investigated for a three and then a four work station balance. The three work station model yielded an optimal solution with a processing time of 17.5 minutes and required 2383 iterations to ascertain the optimal solution. The four work station model had a processing time of 60 minutes requiring 8923 iterations.

The above results were obtained using the IPWE code.

The results for the three work station model that were obtained using the Lemke-Spielberg code provided a sub-optimal feasible solution at iteration number 173 and an optimal solution at iteration 1105 with a processing time of 13.2 minutes. The four work station balance model did not yield an optimal or sub-optimal solution and was processed for 60 minutes going through 11959 iterations.

The balances for three work stations were compared and the optimal values were identical.

Since Bowman's second linear programming model yielded an optimal solution for seven work stations, no attempt was made to test the three and four work station models.

VI CONCLUSIONS

The objective was to investigate the feasibility of assembly line balancing using the linear programming approach. Two linear programming formulations have been considered:

(1) White's zero-one integer modification of Bowman's first linear programming formulation.

(2) Bowman's second linear programming formulation.

The results obtained indicate the following conclusions:

1. It is feasible to apply linear programming techniques to assembly line balancing problems. This was indicated by the results in that optimal solutions were obtained. However, using integer linear programming codes has not lead to uniformly good results.

2. White's zero-one integer program when compared to Bowman's second formulation was easier to formulate while the latter was more compact. Difficulties can be encountered when formulating the non-interference constraints of Bowman's second model, but once the constraints have been written down the remainder of the formulation becomes virtually automatic.

3. Bowman's second model is superior to White's zero-one formulation because of the lesser number of constraint equations (or inequalities) and variables involved as given in Section III. Therefore, Bowman's second model can handle problems of larger size (more work elements) than White's.

The size of the problem will be limited by computer storage capacity.

4. The assembly line balancing problem as a linear programming problem demands integer-valued solutions. This is because activities and resources such as work elements and people are indivisible. The problem can be solved as an ordinary linear programming problem, and attempts made to round the answers obtained to give integer valued solutions. However, simple examples have shown that scientific rounding procedures do not give feasible results. Hence, integer linear programming algorithms are absolutely necessary.

5. The integer linear programming algorithms are exhaustive mathematical algorithms which successively consider all combinations of elements and thus, obtain an optimum balance for any productive output desired. The processing time can be very long, favoring the heuristic-type algorithm which incorporates logical decision rules to reduce the combinatorial content of the problem. This latter type method sacrifices assurance of optimality for the economy, in that fewer iterations are required to achieve an acceptable balance.

6. Large integer programs are needed to formulate rather small assembly line balancing problems. The linear programming formulation, although mathematically correct, is awkward and unnatural when formulated as an integer program.

Such formulations lead to very large problems in numbers of constraints and variables.

7. Although the approach proved to work within the ranges selected, the technique is not practical because it required excessive amounts of computer processing time to obtain a solution.

VII RECOMMENDATIONS FOR FURTHER STUDY

Because Bowman's second programming formulation leads to a smaller problem in terms of number of constraints and variables than White's, future study should be concentrated on Bowman's second linear programming formulation.

In further study along these lines, one of the most important avenues to be explored concerns the possibility of reducing the number of unknowns I_{jk}. The unknowns are only involved in connection with work element interference conditions, and many of these restrictions will inevitably turn out to be redundant in any particular problem.

Future study should concentrate on deriving methods which more fully take into account the special structure of assembly line balancing problem, and which are capable of finding quickly an integer solution.

Since only two of the many linear integer algorithms that are available were used, further investigation of Bowman's linear programming approach should be made using the integer codes not used in this experiment.

BIBLIOGRAPHY

1. Bowman, E.H., "Assembly Line Balancing by Linear Programming," Operations Research, Vol. 8, No. 3, (May-June 1960), pp.385-9.

2. Bryton, B., "Balancing of a Continuous Production Line," Unpublished M.S. Thesis, Northwestern University, Evanston, Illinois, June 1954.

3. Brennecke, D.H., "An Investigation of the Role of Element Time Variability in the Assembly Line Balancing Problem, "Unpublished Ph.D. Dissertation, Indiana University, 1962.

4. Burgeson, J.W., T.E. Daum, Production Line Balancing, File No. 10.3.002, International Business Machines Corp., Akron, Ohio (1958).

5. Gutjahr, A.L., G.L. Nemhauser, "An Algorithm for the Line Balancing Problem," Management Science, Vol. 11 (1964),pp. 308-15.

6. Held, M., R.M. Karp, R. Shareshian, "Assembly Line Balancing with Precedence Constraints," Operations Research, Vol. 11 (1963), pp. 442-59.

7. Helgeson, W.B., D.P. Birnie, "Assembly Line Balancing Using the Ranked Positional Weight Technique," Journal of Industrial Engineering, Vol. 12, No. 6, (Nov.-Dec. 1961) pp. 394-8.

8. Helgeson, W.B., T.T. Kwo, Letter to the Editor, Management Science, Vol. 3, No. 1, (October 1956), p. 115.

9. Hoffman, T., "Assembly Line Balancing with a Precedence Matrix," Management Science, Vol. 9, No. 4, (July 1963), pp. 551-562.

10. Hoffman, T., Generation of Permutations and Combinations, Engineering Experiment Station Report No. 13, University of Wisconsin, (1959).

11. Jackson, J.R., "A Computing Procedure for a Line Balancing Problem," Management Science, Vol. 2, No. 3, (April 1956), pp. 261-271.

12. Kilbridge, M., "A Model for Industrial Learning Costs," Management Science, Vol. 8, No. 4, (July 1962), pp. 516-527.

13. Kilbridge, M. and L. Wester, "A Heuristic Method of Assembly Line Balancing," Journal of Industrial Engineering, Vol. 12, No. 4, (July-August 1961).

14. Kilbridge, M. and L. Webster, "An Economic Model for the Division of Labor," Management Science Vol. 12, No. 6, Feb. 1966, pp. B-255-B-269.

15. Klein, M., "On Assembly Line Balancing," Operations Research, Vol. 11 (1963), pp. 274-81.

16. Louden, H.M., "Evaluation of Two Methods of Assembly Line Balancing for Various Levels of Constraint," Unpublished M.S. Thesis, Lehigh University, Bethlehem, Pennsylvania, May 1965.

17. Manne, A.S., "On the job-Shop Scheduling Problem," Operations Research, Vol. 8, No. 2, March-April 1960, pp. 219-223.

18. Mansoor, E.M., "Assembly Line Balancing - An Improvement in the Ranked Positional Weight Technique," Journal of Industrial Engineering, (March-April 1964), pp. 73-77.

19. Mansoor, E.M., "Improvement on Gutjahr and Nemhauser's Algorithm for the Line Balancing Problem," Management Science, Vol. 14, No. 3, November 1967, pp. 250-254.

20. Mansoor, E.M., and S. Ben-Tuvia, "Optimizing Balanced Assembly Lines," Journal of Industrial Engineering, Vol. 17, No. 3, March 1966, pp. 126-131.

21. Moodie, C.L., "A Heuristic Method of Assembly Line Balancing for Assumptions of Constant or Variable Work Element Times," Unpublished Ph.D. Dissertation, Purdue University, 1964.

22. Moodie, C.L., and Young, H.H., "A Heuristic Method of Assembly Line Balancing for Assumptions of Constant or Variable Work Element Times," Journal of Industrial Engineering, Vol. 16 (1965), pp. 23-29.

23. Muther, R., Production Line Techniques, New York, McGraw-Hill, 1944.

24. Prenting, T.O., R.M. Battaglin, "The Precedence Diagram: A Tool for Analysis in Line Balancing," Journal of Industrial Engineering, Vol. 15, No. 4 (July-August 1964), pp. 208-213.

25. Salveson, M.E., "The Assembly Line Balancing Problem," Transactions of the ASME, Vol. 77, No. 6, (August 1955), pp. 939-947.

26. Tonge, F.M., "A Heuristic Program of Assembly Line Balancing," New York, Prentice Hall, 1961.

27. Weingartner, B.A., Research Engineer Western Electric Company Incorporated, Private Communication.

28. White, W.W., Letters to the Editor, Operations Research, Vol. 9, No. 2, (March-April 1961), pp. 274-276.

29. Woolsey, R.E., and C.A. Trauth, IPSC A Machine Independent Integer Linear Program, Sandia Corporation Monograph SC-RR-66-433, July, 1966.

30. Lemke, C.E., and K. Spielberg, "Direct Search Zero-One and Mixed Integer Programming," Report #3, IBM Data Processing Division, New York Scientific Center, June 1966.

31. Balas, E., "An Additive Algorithm for Solving Linear Programs with Zero-One Variables," Operations Research, Vol. 13, No. 4, (July-August 1965), pp. 517-545.

VITA

Personal History

Name	Walter P. Mydlow
Date of Birth	February 29, 1924
Place of Birth	Chicago, Illinois
Parents	Mary B. and Stanley M. Mydlowski
Wife	Mary Kate Mydlow
Children	Michael W., Patricia K., and Mark J.

Educational Background

University of Tennessee	Attended 1955-1957
University of Memphis State	Attended 1957-1961
Christian Brothers College Bachelor of Science in Electrical Engineering	Graduated 1963
Lehigh University Candidate for Master of Science in Industrial Engineering	1966-1968

Professional Experience

Western Electric Company, Inc. Winston-Salem, North Carolina	1963-1966
Western Electric Company, Inc. Princeton, New Jersey	1966-1968