AD 700 117

ON A MULTI-PRODUCT ASSEMBLY LINE BALANCING PROBLEM

Stephen D. Roberts, et al

Florida University
Gainesville, Florida

November 1969

ON A MULTI-PRODUCT ASSEMBLY LINE BALANCING PROBLEM

Technical Report No. 29

PROJECT THEMIS

ON A MULTI-PRODUCT ASSEMBLY
LINE BALANCING PROBLEM

Technical Report No. 29

SUBMITTED BY:

Carlos D. Villa
Research Associate

Stephen D. Roberts
Principal Investigator

APPROVED BY:

R. L. Patterson
Director, Project THEMIS

Department of Industrial and Systems Engineering
The University of Florida
Gainesville, Florida 32601

November 1969

This research was performed under Project THEMIS, ARO-D Contract No. DAH C04 68C0002.

ABSTRACT

A multi-product assembly line balancing problem is presented. The problem is formulated as a linear integer programming problem for which zero-one programming is applicable. A reformulation based on finding the shortest route in a finite directed network is developed and an example problem is solved.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Introduction

Assembly line balancing is a constrained combinatorial 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 as to optimize a measure of efficiency. Numerous algorithms have been developed for the single product assembly line balancing problem. Ignall (5), Mastor (7), and Cauley (3) give a summary and evaluation of some of these procedures.

A more common situation calls for more than one product to be assembled on the same line. This occurs, for instance, when the number of products exceeds the number of assembly lines and it is desired to keep every product in production rather than make batches intermittently for inventory. Kilbridge and Webster (6) propose a sequencing scheme for multi-product lines and Themopoulis (8), Arcus (1), and Bryton (2) allude to some procedures for balancing multi-product assembly lines. However, none explicitly treat the following problem.

A Multi-Product Assembly Line Balancing Problem

The multi-product assembly line balancing problem can be stated as follows:

Given N products, each with its own precedence constraints, assign the work elements to work stations so that:

a. Each work element is assigned to exactly one work station.

b. The number of stations is the same for all products.

c. The precedence constraints are satisfied.

d. The work content of any station for any given product does not exceed the cycle time, i.e.,

$$S_{ki} \leq C \qquad i=1,\ldots,N \quad k=1,\ldots,K \qquad \text{Eqn.[1]}$$

where:

S_{ki} is the work content of station k for product i

C is the cycle time

K is the number of stations in the line.

e. Total idle time is minimized. Total idle time is given by:

$$I = \sum_{i=1}^{N} \sum_{k=1}^{K} n_i (C - S_{ki}) \qquad \text{Eqn.[2]}$$

where:

n_i is the number of units of product i to be assembled.

It can very easily be established that minimizing idle time is equivalent to minimizing the number of stations in the line. From [2]

$$I = \sum_{i=1}^{N} \sum_{k=1}^{K} n_i (C - S_{ki}) = \sum_{i=1}^{N} \sum_{k=1}^{K} n_i C - \sum_{i=1}^{N} \sum_{k=1}^{K} n_i S_{ki} \qquad \text{Eqn.[3]}$$

let,

$$\sum_{i=1}^{N} \sum_{k=1}^{K} n_i S_{ki} = T \qquad \text{Eqn.[4]}$$

then T is the total work content of the assemblies, a constant; let,

$$\sum_{i=1}^{N} n_i = M \qquad \text{Eqn.[5]}$$

be the total number of assemblies, also a constant. Thus, we can write [3] as

$$I = KCM - T \qquad \text{Eqn.[6]}$$

Thus, for a fixed cycle time, minimizing I reduces to finding the minimum number of stations (or alternatively, for a fixed number of stations, minimizing I reduces to finding the minimum cycle time).

An Integer Programming Model

It will now be shown that this problem can be formulated as a zero-one programming problem. The following notation will be used:

$$X_{ki}^{j} = \begin{cases} 1 & \text{if task } j \text{ of product } i \text{ is assigned to station } k \\ 0 & \text{otherwise} \end{cases}$$

t_i^j is the time to perform task j of product i

J_i number of tasks in product i

The work content of station k for product i, S_{ki}, can be written as

$$S_{ki} = \sum_{j=1}^{J_i} t_i^j X_{ki}^j \qquad \text{Eqn.[7]}$$

Thus, the objective function, i.e., idle time can be written as

$$I = \sum_{i=1}^{N} \sum_{k=1}^{K} n_i \left(C - \sum_{j=1}^{J_i} t_i^j X_{ki}^j\right) \qquad \text{Eqn.[8]}$$

and the constraints may now be expressed as:

a. Each task is assigned to exactly one work station:

$$\sum_{k=1}^{K} X_{ki}^{j} = 1 \qquad j=1,\dots,J_i \quad i=1,\dots,N \qquad 9]$$

b. The precedence constraints: If for task j_1 and j_2 of product i, j_1 must precede j_2, written $j_1 \leftarrow j_2$, then

$$\sum_{k=1}^{K} k\, X_{ki}^{j_1} \leq \sum_{k=1}^{K} k\, X_{ki}^{j_2} \qquad i=1,\ldots,N \qquad \text{Eqn.[10]}$$

for all $j_1 \leftarrow j_2$

c. The work content of the stations does not exceed the cycle time:

$$\sum_{j=1}^{J_i} t_i^j\, X_{ki}^j \leq C \qquad k=1,\ldots,K \quad i=1,\ldots,N \qquad \text{Eqn.[11]}$$

d. $$X_{ki}^j \in \{0,1\} \qquad j=1,\ldots,J_i \quad i=1,\ldots,N \quad k=1,\ldots,K \qquad \text{Eqn.[12]}$$

Two problems immediately arise if one attempts to minimize [8] subject to [9], [10], [11], and [12]. First of all the number K is a decision variable, and it is used in [8]-[12] as a known constant. The second difficulty concerns the constraint that the number of stations be the same for all products, which is ignored in the formulation.

In order to resolve these problems several concepts must be introduced. Given the work content of a product and a value of the cycle time there is a minimum number of stations which are absolutely required for the assembly of that product. For product i, this minimum number of stations will be denoted by K_i^*. It can be seen that

$$K_i^* = \left\lceil \frac{\sum_{j=1}^{J_i} t_i^j}{C} \right\rceil \qquad \text{Eqn.[13]}$$

where:

[x] is the bracket function which yields the smallest integer greater than or equal to x.

Assume that the value of C and the work content of the products are such that

$$K_i^* = K \qquad i=1,\ldots,N \qquad \text{Eqn. [14]}$$

Therefore, since it is impossible to balance the line with less than K^* stations, to minimize the number of stations it is sufficient to minimize the work content of the stations after K^* in such a way that if the work content of a station k_1 ($k_1 > K^*$) is zero then the work content of all stations $k > k_1$ must also be zero and thus only $k_1 - 1$ stations are needed.

The total work content of station k is given by

$$\sum_{i=1}^{N} n_i \sum_{j=1}^{J_i} X_{ki}^j \; t_i^j \qquad \text{Eqn. [15]}$$

Thus, the objective function can be written as

$$I' = \sum_{k=K^*+1}^{K'} W_k \sum_{i=1}^{N} n_i \sum_{j=1}^{J_i} X_{ki}^j \; t_i^j \qquad \text{Eqn. [16]}$$

where:

W_k is a weight assigned to station k. This weight is set to insure that if the work content of station k_1 is zero, i.e.,

$$\sum_{i=1}^{N} n_i \sum_{j=1}^{J_i} X_{k_1 i}^j \; t_i^j = 0 \qquad \text{Eqn. [17]}$$

then,

$$\sum_{i=1}^{N} n_i \sum_{j=1}^{J_i} X_{ki}^{j} \; t_i^{j} = 0 \qquad \text{Eqn.[18]}$$

for all $k > k_1$.

The purpose of the W_k is to make an assignment to a station k_2 less favorable than the same assignment to station k_1 when $k_1 < k_2$. Thus, the W_k must be an increasing sequence of positive numbers. If we let

$$W_k = 2^{k-(K^* + 1)} \qquad k = K^* + 1 ,\ldots,K' \qquad \text{Eqn.[19]}$$

this will satisfy the above requirement.

K' is a number large enough to insure that no assignment of work elements to work stations will require more than K' stations. An obvious choice for K' is

$$K' = \max_i \{ J_i \} \qquad \text{Eqn.[20]}$$

It remains to insure that the number of stations is the same for all products. If the work content of station k for product i_1 is zero then the work content of this station for all other products must also be zero. Now, if the work content of station k for product i_1 is zero then

$$\sum_{j=1}^{J_{i_1}} X_{ki_1}^{j} = 0 \qquad \text{Eqn.[21]}$$

and therefore we must have

$$\sum_{j=1}^{J_i} X_{ki}^{j} = 0 \qquad \text{for all } i \qquad \text{Eqn.[22]}$$

Let M be a large number, and consider the following inequalities

$$\sum_{j=1}^{J_{i_1}} X^j_{ki_1} \leq M \sum_{j=1}^{J_{i_2}} X^j_{ki_2} \qquad \text{Eqn.[23]}$$

$$\sum_{j=1}^{J_{i_2}} X^j_{ki_2} \leq M \sum_{j=1}^{J_{i_1}} X^j_{ki_1} \qquad \text{Eqn.[24]}$$

If either $\sum_{j=1}^{J_{i_1}} X^j_{ki_1}$ or $\sum_{j=1}^{J_{i_2}} X^j_{ki_2}$ is zero the only value of the other that satisfies [23] and [24] is zero. On the other hand, if one of them is different from zero then the other can take any value different from zero. One obvious choice for M is

$$M = \max_i \{ J_i \} \qquad \text{Eqn.[25]}$$

Therefore if for every $k > K^*$ and for every pair of products $i_1=1, i_2=2, \ldots, N$, we require that inequalities [23] and [24] be satisfied we are requiring that the number of stations be the same for all products.

Thus, summarizing the above results, the multi-product assembly line balancing problem can be stated as follows

$$\text{minimize} \quad I' = \sum_{k=K^*+1}^{K^1} W_k \sum_{i=1}^{N} n_i \sum_{j=1}^{J_i} X^j_{ki} \, t^j_i \qquad \text{Eqn.[26]}$$

where $W_k = 2^{k-(K^*+1)}$ and $K^1 = \max_i \{ J_i \}$

subject to:

a. Each task is assigned exactly once

$$\sum_{k=1}^{K^1} X_{ki}^{j} = 1 \qquad j=1,\ldots,J_i \qquad i=1,\ldots,N \qquad \text{Eqn.[27]}$$

b. The number of stations is the same for all products:

$$\sum_{j=1}^{J_{i_1}} X_{ki_1}^{j} \le M \sum_{j=1}^{J_{i_2}} X_{ki_2}^{j} \qquad \text{Eqn.[28]}$$

$$\sum_{j=1}^{J_{i_2}} X_{ki_2}^{j} \le M \sum_{j=1}^{J_{i_1}} X_{ki_1}^{j} \qquad \text{Eqn.[29]}$$

$k = K^* + 1,\ldots,K^1$

$i_1 = 1$

$i_2 = 2,\ldots,N$

$M = \max_i \{J_i\}$

c. The precedence constraints: If $j_1 \leftarrow j_2$ in product i

$$\sum_{k=1}^{K^1} k \, X_{ki}^{j} \le \sum_{k=1}^{K^1} k \, X_{ki}^{j_2} \qquad i=1,\ldots,N \qquad \text{Eqn.[30]}$$

d. The work content of the stations does not exceed the cycle time

$$\sum_{j=1}^{J_i} t_i^j X_{ki}^{j} \le C \qquad k=1,\ldots,K \qquad i=1,\ldots,N \qquad \text{Eqn.[31]}$$

e. $X_{ki}^{j} \in \{0,1\}$ $\quad j=1,\ldots,J_i$

$$i=1,\ldots,N \qquad k=1,\ldots,K' \qquad \text{Eqn.[32]}$$

The assumption that $K_i^* = K^*$ $i=1,\ldots,N$ was necessary only to simplify the development. When [14] does not hold then

$$K^* = \max_i \{K_i^*\} \qquad \text{Eqn.[33]}$$

and additional constraints must be added to insure that the work content of stations K_i^* to K^* is not zero for those i such that $K_i^* < K^*$.

Locational constraints, (e.g., that all welding must be performed at the same station) can also be incorporated. For example, if task j_1 of product i_1 and task j_2 of product i_2 must be performed at the same station then we require that

$$X_{ki_1}^{j_1} = X_{ki_2}^{j_2} \qquad \text{for } k=1,\ldots,K' \qquad \text{Eqn.[34]}$$

consequently, locational constraints should improve the solution efficiency.

The above formulation can then be solved by a zero-one programming algorithm for the assignment of tasks to stations which result in the minimum number of stations. The large number of variables and constraints would make this, however, a difficult undertaking even for problems of modest size.

The main result from the above formulation is theoretical. It shows that the multi-product assembly line balancing problem can be formulated as an optimization problem. Further study of this formulation might suggest a special structure which can then be solved by a specialized algorithm.

A Network Model

Gutjahr and Nemhauser (4) give an algorithm for solving a single product line balancing problem as a shortest route problem. The algorithm can be extended to the multi-product problem but only at the expense of a considerable increase in computations and memory requirement.

The process consists of developing a finite directed network for which the arcs represent stations in the assembly line and the nodes correspond to possible first station assignments. The arc lengths are the idle times within the stations. Thus the optimization procedure is equivalent to finding the shortest path in the network or, by Equation [6], the minimum number of arcs. To construct the network for the multi-product case, use the following to develop the nodes and arcs.

A) 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, except for the empty state $\emptyset$, for each product are generated as if that product were the only one to be assembled. This results in a set, S_i, of states for each product i. The set of all states for the multi-product problem is then obtained from the Cartesian product of all S_i, i.e.,

$$S = \{S_1 \bullet S_2 \bullet S_3 \bullet \ldots S_N, \emptyset\} \qquad \text{Eqn.[35]}$$

A typical element is S, say T_1 is given by

$$T_1 = \{S_{11}, S_{21}, \ldots, S_{N1}\} \qquad \text{Eqn.[36]}$$

where S_{i1} is an element of S_i for $i=1,\ldots,N$. Note that if locational restrictions exist, then those states which violate these conditions may be discarded immediately. The states developed form the nodes in the network. The nodes are constructed from the empty state, node o , forming the states or nodes according to the above procedure. The final node in the network will consist of all elements for all products and will be denoted r.

B) Construction of directed arcs: There is a directed arc from node or state T_1 to T_m if and only if

$$S_{i1} \subset S_{im} \qquad i=1,\ldots,N \qquad \text{Eqn.[37]}$$

and

$$\sum_{j \in S_{im} - S_{i1}} t_i^j \leq C \qquad i=1,\ldots,N \qquad \text{Eqn.[38]}$$

Thus, the network is formed according to the previous instructions and it is sufficient to find the shortest path from 0 to r containing the minimum number of arcs. With these extensions the algorithm of Gutjahr and Nemhauser (4) may be used to solve a multi-product problem.

An Example Problem

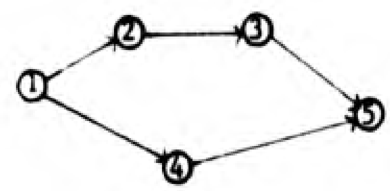
An example problem will now be given which shows the solution of a two product problem using the generalized Gutjahr-Nemhauser algorithm. Consider the following products with the corresponding precedence graphs:

Product 1:

$t_1^1 = 4$ $t_1^4 = 2$

$t_1^2 = 2$ $t_1^5 = 2$

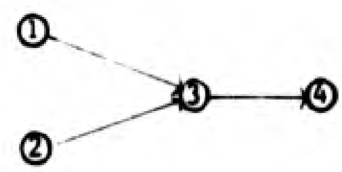
$t_1^3 = 3$



Product 2:

$t_2^1 = 2$ $t_2^3 = 4$

$t_2^2 = 3$ $t_2^4 = 4$



Let the cycle time, C, be equal to 5. The solution begins by forming,

$$S_1 = \{ (1), (1,2), (1,4), (1,2,3), (1,2,4), (1,2,3,4), (1,2,3,4,5)\}$$

$$S_2 = \{ (1), (2), (1,2), (1,2,3), (1,2,3,4)\}$$

and thus

$$S = \{ [(1), (1)], [(1), (2)], [(1),(1,2)], [(1), (1,2,3)], [(1),(1,2,3,4)],$$
$$\ldots, [(1,2,3), (1)], \ldots, [(1,2,3,4,5), (1,2,3,4)], \emptyset \}$$

A complete list of all states is given in Table 1 along with the state time.

The network generated until node r = [(1,2,3,4,5) (1,2,3,4)] is reached for the first time is given in Figure 1. The numbers in Figure 1 correspond to the ones given in Table 1. The optimal balance is then

Product 1	Task	Station
	1	1
	2, 3	2
	4, 5	3

Table 1: State Generation

Node	State Elements Product 1	State Elements Product 2	State Time Product 1	State Time Product 2
0	Ø		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

Table 1: (Continued)

Node	State Elements Product 1	State Elements Product 2	State Time Product 1	State Time Product 2
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	(1,2,3,4,5)	(1,2,3,4)	13	13

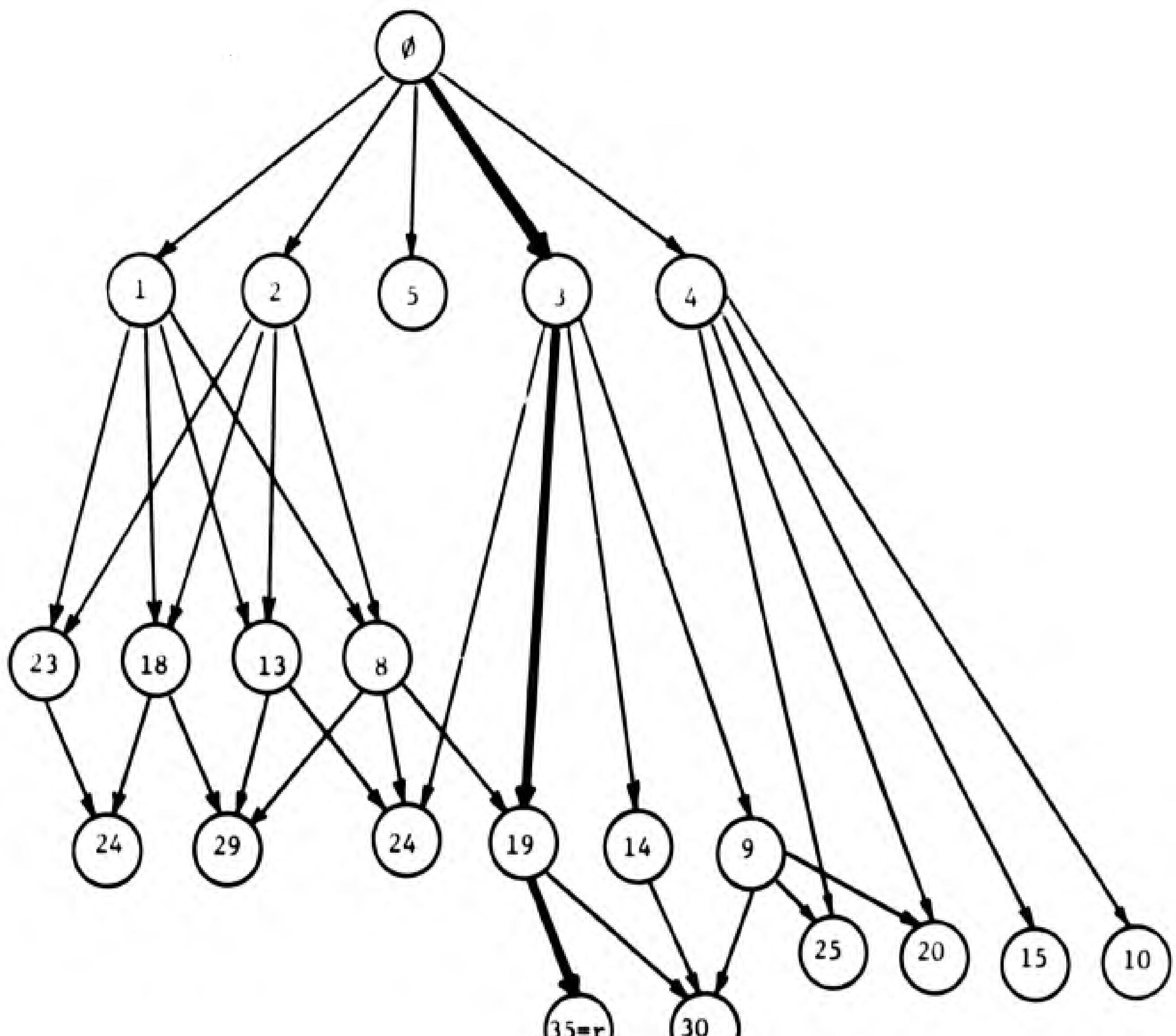


Figure 1: Resulting Network

Product 2	Task	Station
	1 }	1
	2 }	
	3	2
	4	3

Total idle time is 2 for each unit of product 1 and 2 for each unit of product 2 assembled.

BIBLIOGRAPHY

1. Arcus, A.L., "COMSOAL: A Computer Method of Sequencing Operations for Assembly Lines, "International Journal of Production Research, Volume 4, No. 4, 1966.

2. Bryton, B., Balancing of a Continuous Production Line, MS Thesis, Northwestern University, Department of Mechanical Engineering, 1954.

3. Cauley, J.M., A Review of Assembly Line Balancing Algorithms, AIIE 19th Annual Conference, Proceedings, 1968

4. Gutjahr, A.L. and Nemhauser, G.L. "An Algorithm for the Line Balancing Problem," Management Science, Volume II, No. 2, 1964.

5. Ignall, E.J., "A Review of Assembly Line Balancing", Journal of Industrial Engineering, Volume XVI, No. 4, 1965.

6. Kilbridge, M., and Webster, L., "The Assembly Line Model-Mix Sequencing Problem", Proceedings of the Third International Conference on Operations Research, Oslo, 1963.

7. Mastor, A.A., "An Experimental Investigation and Comparative Evaluation of Production Line Balancing Techniques", unpublished Ph.D. Dissertation, UCLA, 1966.

8. Thomopoulas, N.T., "Line Balancing-Sequencing for Mixed-Model Assembly", Management Science, Volume 4, 1967.

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author): University of Florida	2a. REPORT SECURITY CLASSIFICATION: Unclassified
	2b. GROUP: Unclassified

3. REPORT TITLE

"On a Multi-Product Assembly Line Balancing Problem"

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (Last name, first name, initial)

Stephen D. Roberts and Carlos D. Villa

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
November 1969	17	8

8a. CONTRACT OR GRANT NO. DAHC04 68C 0002	9a. ORIGINATOR'S REPORT NUMBER(S): Technical Report No. 29
b. PROJECT NO. 2T0 14501B81B	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
c.	
d.	

10. AVAILABILITY/LIMITATION NOTICES

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	ARO-Durham

13. ABSTRACT

A multi-product assembly line balancing problem is presented. The problem is formulated as a linear integer programming problem for which zero-one programming is applicable. A reformulation based on finding the shortest route in a finite directed network is developed and an example problem is solved.

DD FORM 1 JAN 64 1473

Security Classification