Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Assembly Line Balancing with Multiple Stations

G. M. Buxey,

To cite this article:
G. M. Buxey, (1974) Assembly Line Balancing with Multiple Stations. Management Science 20(6):1010-1021. http://dx.doi.org/10.1287/mnsc.20.6.1010

MANAGEMENT SCIENCE
Vol. 20, No. 6, February, 1974
Printed in U.S.A.

ASSEMBLY LINE BALANCING WITH MULTIPLE STATIONS*†

G. M. BUXEY

University of Bradford, England

A case is stated for extending the techniques of assembly line balancing to provide for the parallel operation of identical stations, where this leads to a reduction in idle time. The practical implications of operating with this type of system are discussed, both for the stations themselves and the line as a whole, with reference to various classifications of assembly line, and ways in which balancing can be made to fit into an overall strategy for production line design are touched upon. Two distinct types of computer program have been developed to enable multiple stations to become a recognised feature incorporated into 'heuristic' line balancing, rather than an appendage to be applied ad hoc by industrial engineers when current techniques have proven inadequate. One approach is based on a more sophisticated version of the 'positional weight' method while the other relies on the contrasting philosophy of the 'random generation' method, and a comparison is made of their relative success in solving two assembly line problems, and their potential from an industrial viewpoint.

1. Introduction

Since the assembly line balancing problem was first formulated in 1955, [16], much progress has been made in finding techniques of solution that are applicable in industry. Currently available methods, mainly taking a 'heuristic' approach, have been reviewed and compared by several authors, ([7], [15], [17]), and sophisticated computer programs are now used to balance lines assembling motor cars, washing machines, and other complex consumer products, with significant increases in efficiency reported. Despite this a survey by Lehman [13] revealed that some 40% of U.S. companies balanced their lines by trial and error and only 15% were using a computer-aided program.

Heavy, highly-engineered products are usually assembled on capital-intensive production lines with a high return for increased efficiency and a great emphasis on careful planning of production facilities. Under these circumstances computer-aided design has been economically attractive and this is reflected in the development of a line balance methodology most suited to this type of manufacture. In particular, these lines often have the workpiece riding on a powered conveyor past a single file of operators, so that to employ more than one operator on any equivalent work station and still ensure an even work flow presents difficulties. However, it is generally unnecessary, an efficient balance being obtainable while still dividing the work between a large number of operatives because individual work elements are very small compared to total job time. In contrast, manual flow lines are liable to fewer layout difficulties but may exhibit some cumbersome elements that make efficient task division difficult unless they may be shared between several operators, each processing only a fraction of items.

Lehman's survey probably indicates a correlation between the lack of general application of current techniques and their unsuitability for many unpaced lines, since

* Processed by Professor Samuel Eilon, Departmental Editor for Production Management; received November 1972, revised. April 1973. This paper was with the author 1 month for revision.

† This research project was supported by the Plessey Company Limited-Plessey Telecommunications Group.

the median line length is only ten stations, and most significantly the median number of operators is fifteen.

The purpose of this research was to develop a computerized line balance program that can use groups of workers in parallel, i.e. at equivalent work stations, at certain points along a sequential line where an improvement in production efficiency is expected. Since this adds another dimension to current methodology some attention is given to the conditions under which it might be considered suitable in the factory, and there is some discussion on the means of ensuring in advance that solutions will be within the bounds of usefulness. Finally, the success of the program(s) is illustrated and their special advantages for assembly line design briefly mentioned.

1.1 *Multiple Stations*

A brief discussion follows on the major advantages of allowing multiple, i.e. parallel, stations on a line, the spin-off benefits, and the disadvantages. A basic knowledge of line balancing will be assumed throughout.

(i) The primary reason for multiple stations is to improve balance efficiency. Each station that is duplicated has an effective cycle time of (cycle time $\times$ station multiple), thus a range of times is available and there is more likelihood of a good fit. This is particularly relevant when balance difficulty is encountered due to the larger work elements being of the same order as the cycle time [9].

The point has been well illustrated by Mariotti [14], in part as a repudiation of computer algorithms. However, even a cursory examination of manual techniques [8], [10], shows them to be very laborious unless confined to small problems and this is amplified by the complications of element multiplicity (the number of operators allowed to perform the work element), so that to be of any real use the concept of multiple stations must be amalgamated with a computerised line balance heuristic. This has been achieved by Heskiaoff [6] who reports sharply improved efficiency when allowing one or two operators/station during the application of the simple Positional Weight Technique [5].

(ii) Multiple stations enable the production rate to be greater than the limitation imposed by the longest work element. Output can then be increased relatively smoothly to meet demand, rather than by such devices as overtime until the orders level rises to the point where duplicate line facilities become economic. Caruso [3] describes a program with the capability of assigning two operators/station whenever an element time $>$ cycle time is encountered.

(iii) Incorporating multiple stations in a line may lead to a substantial reduction in 'system loss', i.e. idle time incurred due to differences in operator process times (the human factor). Van Beek [18] has shown the importance of the mismatch of the *mean* work rates of different operatives on *unpaced* lines, and the resultant superiority of short lines compared with long ones. Paralleled stations not only reduce the length of the series chain but also allow careful personnel selection to ensure a compatible throughput from the station as a whole. The intermittent losses due to individual operator *variability* should also decrease, as the significant factors in the computer simulation of Wild and Slack [19] show. Firstly, the number of inter-dependent stages is less, secondly, the coefficient of variance will decrease providing that the element times grouped at the parallel stations can be assumed independent [2], and lastly, the operator 'interference' will decrease due to alternative sources of supply and demand generated at multiple stations.

(iv) The main problems posed are in ensuring a regular work flow without excessive layout difficulties and transportation times. On unpaced lines we may be limited to a maximum of two operators/station (symmetrically arranged) unless handling times are negligible. Frequently this is overcome by holding large buffer stocks at each station and occasionally transferring them en masse. Fortunately, buffer stocks greatly increase line efficiency by reducing system loss, although they represent an increased inventory cost.

Fixed-item conveyor lines will generally be restricted to station multiples of two (one operator/side of track) as station length and thus non-productive travelling time increase in proportion to the multiple and there may be difficulties with power cables, hoses etc. Removable-item conveyor lines could well benefit from multiple stations although colour coding may be necessary to show the operator which item to take from the line.

1.2 *Multiplicity and Cycle Time*

Unlimited paralleling of stations ultimately leads to complete job enlargement. Historically this trend has been reversed, due to the economics involved, and obviously care needs to be taken in specifying the maximum element multiplicities allowed for the assembly of any one product. This refers to the viability of the duplicated stations themselves rather than their inclusion in the overall scheme. The work of Kilbridge and Wester [11], [12], based on the consumer electronics industry, is indicative in this respect. They show that an 'optimum' cycle time (range) exists arising from the counter-directionary cost functions associated with on the one hand balance delay and nonproductive labour, and on the other, learning. The curves indicate that providing the original cycle time has been chosen correctly, for paralleling non-productive labour should either decrease slightly or rise by an amount small enough to be more than compensated for by improved balancing. Thus, in general, this will not increase labour costs or significantly affect the validity of the precedence diagram. By contrast, learning costs accelerate steeply with cycle time, leading in the Kilbridge and Wester example to a total cost curve with a minimum that covers a cycle time range from one to two minutes. However, it is reasonable to suppose that learning costs for a particular assembly are more a function of task complexity rather than time directly [4], and that if we limit the number of work elements at a station to a pre-specified number then increasing cycle time will not alter learning costs so substantially. This will also serve to keep non-productive labour, due to tool changes, re-orientation of workpiece, etc. within the bounds reported by Kilbridge and Wester, i.e. the independence of work elements will be preserved. Similarly, congestion due to grouping of machines and fixtures is also kept to an acceptable level.

Both the heuristics described in this paper incorporate a limit on the number of work elements/station. This in itself imposes a restriction on the multiplicities that can usefully be employed, and also ensures the basic principle of division of labour by necessitating a minimum number of sequential stations. Thus multiple stations are used only to make the *longer* elements adaptable to the work cycle pattern and should not detract from the essential benefits of flow line production.

2. Basic Balance Programs

In the lime balancing research reported below two programming approaches have been followed, one attempts to obtain a good balance by optimizing each station in turn and provides a unique solution, whilst the other generates a number of random sequences and selects the best. Both methods are summarised, together with the special features necessary to adapt them to the concept of multiple stations, and two simple ways of producing alternative balances derived from the first method. The programs described below have been written in Fortran.

2.1 *Optimal Station Program*

Several schemes have been reported for balancing that apply an overall strategy, listing elements in a logical order and allocating to the manufacturing sequence accordingly, except where an element time fails to fit into the remaining station time. Notable examples are the Positional Weight Technique [5] and ordering based on 'number of succeeding elements'. These *heuristics* have proved extremely useful but since they only consider one possible sequence at any particular cycle time they are somewhat inflexible and are liable to encounter difficulties sometimes. It has been possible, without extensive computation, to retain their essential advantages whilst widening the search for an efficient solution.

The positional weight method has been chosen as a base for the computerized program since it takes into account both element duration and positional 'importance'. In effect this concept seeks to allot those elements that free from precedence constraints the greatest proportion of total job time, i.e. it tries to assure the widest possible choice at each stage of selection whilst favouring the allocation of larger elements first. Precedence data may be input in two forms, a full matrix where a code number indicates the precedence relation of the corresponding column and row elements or where just the immediate predecessors of each element are listed. In the former case a subroutine compiles positional weights and re-shuffles elements into ranked order, whilst for the latter the elements must be pre-sorted. For larger problems any inconvenience in using the latter alternative is fully justified by the great savings in computer storage, computer running time and data preparation. The program works by a series of logically identical stages, one stage for each element that may be assigned to a station. Thus the maximum number of elements per station must be estimated, according to the principles of §1.2, and the program expanded correspondingly. At any point in the computation the subset of elements that are under consideration as the best grouping for the current station are represented by indices equivalent to their placings in the positional weight table. Each stage tests elements for availability by asking if they are still free and whether (immediate) predecessors have been allocated to prior stations (by consulting an assignment matrix) or are included in the subset itself, and also calculates if the remaining station idle time is great enough to accommodate them. If an element is suitable it is included in an expanded subset and the program moves on to the next stage, and in any case the search continues from the next element in the table. Initially elements are placed in a station in a manner identical to the positional weight method, but all possible combinations are examined by sweeping through the table like a clock, one complete search by the i th stage causing a change of 1 in the index (or element) of the (i–1) th stage, the first 'best fit' being eventually selected, unless an exact fit is found, in which case it is immediately assigned and the search halted. After a station allocation is completed an assignment matrix is updated and the process repeated for the next station. The positional weight ordering is invaluable as no element can ever make a predecessor become available, and there is no need to 'backtrack' when passing through the stages—a great saving of unnecessary computation. Since the first element etc. has the slowest rate of incrementation from the initial setting, positional weight order is preserved to the greatest extent compatible with an optimal station, and if the heuristic is accurate little extra calculation is involved. All stations are therefore optimized, subject to others being pre-assigned, although this does not necessarily guarantee the best sequence.

2.2 *Random Generation*

The basic technique is identical to that of Arcus [1], except that a limit is imposed on the number of elements that can be at any one station. A specific number of sequences is produced at 'random' by employing a random number generator to select elements from a 'fit' list, which is continuously updated as the sequence progresses and more elements are allocated. The 'fit' list comprises all elements that are currently free from precedence constraints (these form an 'availability' list) and whose duration is not greater than the remaining unused station time. The selection process is in fact weighted in proportion to the positional weights of the elements, as this has been shown to increase efficiency. A further improvement is made by aborting sequences when their accumulated idle time becomes greater than the best result to date. As before, precedence data can be supplied in 'immediate predecessor' notation if the positional weights are known, but as the program operates by random selection no particular ordering is required.

2.3 *Multiple Stations*

Both programs have been extended to enable the use of multiple stations. The maximum multiplicity allowable for every element must be defined; this must be high enough at least to make the proposed cycle time possible. The line will then be balanced with each station being given a multiple value, but in no circumstances may it be greater than any of the individual values nominated for those elements comprising the station.

In the 'optimal station' program instead of calculating whether an element will fit into the remaining station time directly one must allow for all values of station multiple up to the lowest maximum in the subset under consideration. If a fit is obtained, reiterating the calculation with an increment in station multiple can only increase idle time, thus the program proceeds as before, either assigning the subset or moving on to the next stage, and unduly high station multiples are avoided.

For the 'random generation' program the station multiple must be known in advance for a 'fit' list to be extracted from the 'availability' list. It would be possible to compile a list using associated provisional multiples, but this would be highly undesirable as it would squeeze elements into the 'fit' list by using high multiples, although after selection station idle time might well *increase* due to a greater station multiple becoming necessary. This problem is circumvented by choosing a fixed multiple at random within the range of minimum to maximum for the first element allocated to every station. One further refinement is added; if the last station comprises a single element, the minimum multiple is always nominated.

2.4 *Reverse Balancing*

It is quite easy to use an inverse positional weight table or 'immediate successors' notation and balance the line from the last station back to the first. This can give improved answers in certain cases with the 'optimal station' approach, but has no relevance for 'random generation'.

2.5 *Reduced Multiple Stations*

Where the use of multiple stations is considered a serious handicap the 'optimal station' program may be modified to accept from those element groups giving smallest idle time the one requiring the lowest multiple, i.e. a secondary criteria is applied to all possible 'best' subsets. Although this may rely less on the use of parallel stages there is a greater tendency to depart from positional weight ordering, perhaps at times leading to inferior results.

3. Special Restrictions

For most factory situations a precedence diagram cannot express all the implicatons of setting up an assembly line. However, no general program can be expected to cover all eventualities, but rather should be capable of adaptation to cater for individual line peculiarities. Some of the more common restrictions have been added to the programs, so that only minor changes are necessary to include/exclude them, and redundant working is avoided. Since any extra rigidity is liable to worsen the balances obtainable and also to increase the computation, restrictions should be used sparingly.

3.1 *Spatial Zoning*

Zoning constraints occur frequently on lines assembling large products, where operations requiring a positional change of worker would cause unduly high non-productive labour if performed at the same station, e.g. motor car assembly with work on opposite sides of the track in assembly of left-hand and right-hand parts. Each element is given a code number to represent the associated position of the operator, zero meaning that there is no restriction. An extra test is then inserted in the programs between those tests for availability (precedence) and fit (time) to ensure the spatial compatibility of all elements assigned to one station.

3.2 *Negative Zoning*

Negative zoning is used when it is technologically undesirable for two or more work elements to be performed by the same operator, or when tasks should be separated by at least one 'buffer' station. Each element involved in a negative restriction is specified along with those elements that it may not be grouped with and a code number indicating the degree of separation. The procedure is then similar to spatial zoning, the programs testing whether an element is restricted by any other member of the subset, but in addition, it must sometimes test for interference from the prior station.

3.3 *Positive Zoning*

When it is necessary to assign two work elements to the same *or* adjacent stations positive zoning is used. In almost every case it seems likely that they will be required *next* to each other, and one must be made the *sole* immediate predecessor of the other on the precedence diagram to ensure that by rearrangement of elements at a station this is always possible. It should perhaps be noted that all tasks that *must* be grouped at the *same* station should be combined on the precedence diagram and precedence arrows redrawn accordingly. Since positively zoned elements are assumed to be consecutive on the precedence chart it is only necessary on completion of a station to assign immediately to the next station any element that is so required, providing it has not already been designated (to the same station). A most important change arises in the 'optimal station' method because the immediate allocation of a pre-specified element to the first position in a station breaks positional weight convention in the search process. The second stage must therefore return to the start of the element table instead of advancing to the next element, if all feasible combinations are to be examined. For both positive and negative zoning it is sufficient to specify just one restriction for a pair of elements that have a precedence relationship, since only the selection of the successor is subject to zoning.

3.4 *Complementary Elements*

Some jobs need two operators working in unison. This 'complementary' allocation is achieved simply by applying both positive and negative zoning (coded so that allocation to the same station only is barred) to the two elements concerned.

All or any combinations of the above zoning constraints may be used but there is a possibility of conflicting constraints bringing a sequence to a halt or giving errors if excessive zoning is present. With the 'optimal station' technique a unique sequence is generated and programs may need to be re-run with extra negative zoning to avoid these confrontations. Under such circumstances the 'random' generation method may show clear advantages and it is also more suited to the programming alterations, one test for any zoning restriction before compiling the 'fit' list being all that is necessary, whereas for the 'optimal station' scheme this must be repeated at every stage.

4. Application and Results

The success of the programs was first tested by solving the small problem [14] reproduced in precedence diagram form in Figure 1, as this was originally used to expose the shortcomings of computerized balance programs. Cycle times in the range 3 to 12 were tried as this ensures the need for multiple stations but always requires at least four operators, and by limiting five elements to a station a task division of three distinct stations is assured. Only those cycle times with a theoretical minimum balance loss of $<5\%$ were investigated and results for the Optimal Station program are tabulated in Figure 2. For each cycle time four variations of the Optimal Station program have been tried, and they have all been repeated (where applicable) using equal multiplicity allowances for each element from 1 up to 4, making a total of $4 \times 4 = 16$ possible solutions. In the column under each cycle time all the *different* balances obtained that use the minimum possible number of operators are numbered B1, B2 etc., while when 1 extra operator is needed a U is shown. N.B. U does not necessarily indicate a failure for the program as this refers to the 'theoretical' minimum where precedence and time constraints are not accounted for. When different program variations and/or different multiplicity allowances at the *same* cycle time result in identical 'optimum' sequences the table is left blank, while a dash indicates that the corresponding parameters are not applicable for this example.

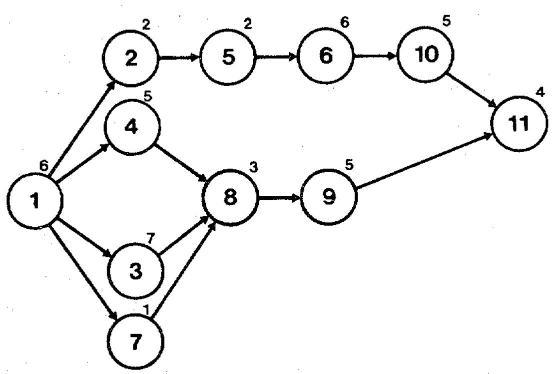

FIGURE 1. Precedence Diagram of Example 1.

BALANCE PARAMETERS		CYCLE TIME				
5 Elements/Station Limit		3	4	6	8	12
Max.Multiplicity 1,	Forward	–	–	–	U	U
	Reduced Forward	–	–	–	–	–
	Reverse	–	–	–	U	B1
	Reduced Reverse	–	–	–	–	–
Max.Multiplicity 2,	Forward	–	U	B1	B1	B2
	Reduced Forward	–	U	U	B2	
	Reverse	–	U	B2	B3	B3
	Reduced Reverse	–	U			
Max.Multiplicity 3,	Forward	B1	B1	B3		
	Reduced Forward	B2	B2	B4		
	Reverse	B3	B3	B5	B4	
	Reduced Reverse		B4		B5	
Max.Multiplicity 4,	Forward		B5			
	Reduced Forward		B6			
	Reverse		B7			
	Reduced Reverse		B8			
Theoretical Minimum Balance Loss%		4·17	4·17	4·17	4·17	4·17

FIGURE 2. Optimal Station Method Balances for Example 1.

For every cycle time it is possible to obtain several 'perfect' balances even when well below the longest work element limit, provided a high enough station multiple is acceptable. In fact the only answers to require extra personnel are those where the minimum possible element multiplicity has been used as the upper limit. Comparison with the results of [14] shows that it is in fact never necessary to use a multiple as high as 4, as given for cycle times 3 and 6, and the sequence given for cycle time 8 is equivalent to Reduced Forward, Multiplicity 2. The solutions tend to be degenerate and find their own level of station duplication, often well below the limits permitted, 4 being used only for a cycle time of 4 for example.

For the Random Generation program a balance was tried for cycle time 12, element multiplicity 1, and optimality obtained after one sequence only.

A more extensive example is taken from [17, p. 86] incorporating both 'positive' and 'spatial' zoning; it is redrawn in Figure 3 with those elements that *must* be located at the test facility combined and the resultant implied precedence arrows shown dashed (N.B. A zoning restriction has been inadvertently transferred from element 14 to 10; elements are renumbered according to positional weight). Even number cycle times in the range 20 to 54 have been used, where the theoretical minimum balance loss is $<5\%$ and the solution is therefore not trivial. A limit of 5, 7 and 9 elements/station has been used, so that *at least* six operators total and 6, 5 or 4 respectively in sequence are needed, thus the solutions given will always represent a progressive line situation.

Results for the two programs are given in Figures 4 and 6, only unique sequences being shown for the sake of clarity within each section of constant elements/station restriction. The highest value used for a multiple station and the number of sequences generated before an 'optimal' solution was found are also indicated in Figure 6; since only 'nondegenerate' answers are shown for the Optimal Station program at least one station has a multiple equal to the limiting value. No balance is worse than one operator above 'minimum' and most are optimum, whilst even with no multiple station a marked improvement over the simple positional weight algorithm is found (Figure 5).

With the *Optimal Station* program several 'perfect' balances are obtainable at each cycle time over the whole range provided the maximum element multiplicities *and*

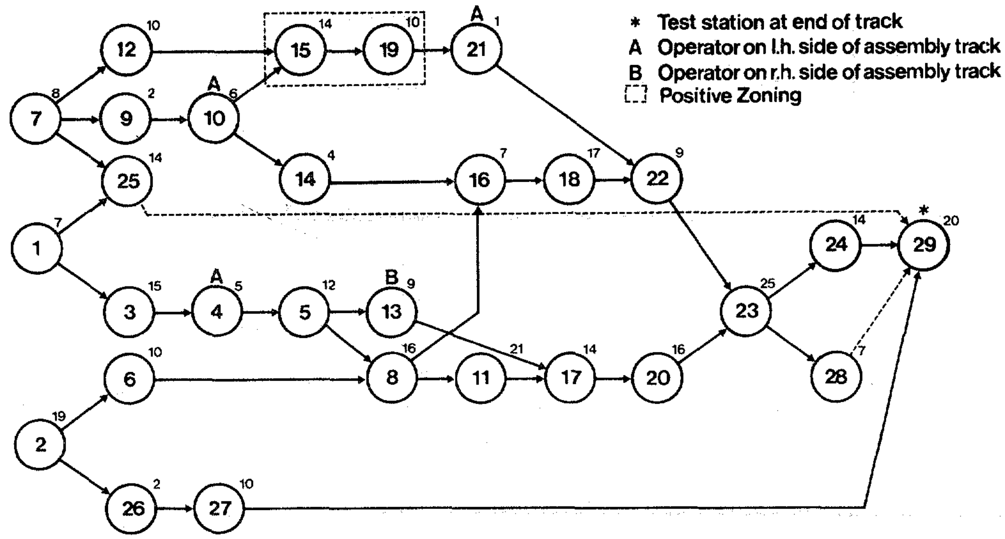

FIGURE 3. Precedence Diagram of Example 2.

BALANCE PARAMETERS		CYCLE TIME										
5 Elements/Station Limit		20	22	24	26	28	30	34	36	42	48	54
Max.Multiplicity 1,	Forward	–	–	–	U	U	U	B1	U	B1	U	U
Max.Multiplicity 2,	Forward	B1	U	B1	B1	B1	B1	B2	U		U	
Max.Multiplicity 3,	Forward	B2	B1	B2								
7 Elements/Station Limit												
Max.Multiplicity 1,	Forward	–	–	–	U	U	U	B1	U	B1	U	U
	Reduced Forward	–	–	–	–	–	–	–	–	–	–	–
	Reverse	–	–	–	U	U	U	B3	U	B2	B1	U
	Reduced Reverse	–	–	–	–	–	–	–	–	–	–	–
Max.Multiplicity 2,	Forward	B1	U	B1	B2	B2	B2	B4	U	B3	B2	U
	Reduced Forward	B3	U	B3	B3	B3	B3	B5	U	B4		
	Reverse	B4	B2	B4	B4	B4	B4	B6	U	B5	B3	U
	Reduced Reverse	B5		B5	B5		B5	B7				
Max.Multiplicity 3,	Forward	B6	B3	B6		B5						
	Reduced Forward		B4	B7				B8	B1			
	Reverse	B7	B5	B8	B6	B6	B6	B9	B2			
	Reduced Reverse	B8	B6		B7	B7	B7	B10	B3			
Max.Multiplicity 4,	Forward	B9	B7									
	Reduced Forward											
	Reverse	B10	B8	B9		B8						
	Reduced Reverse	B11	B9		B8							
9 Elements/Station Limit*												
Max.Multiplicity 1,	Forward										U	U
	Reverse										B1	U
Max.Multiplicity 2,	Forward										B4	B1
	Reverse										B5	U
Theoretical Minimum Balance Loss %		4·71	1·82	3·57	4·14	3·57	1·82	4·71	0·0	3·57	3·57	0·0

*Only cycle times 48 and 54 balanced with this criteria.

FIGURE 4. Optimal Station Method Balances for Example 2.

CYCLE TIME	26	28	30	34	36	42	48	54
Forward Balance	2	1	1	1	1	1	0	1
Reverse Balance	3	2	2	2	3	1	1	1
Min.Operators	13	12	11	10	9	8	7	6

FIGURE 5. Excess Operators Used—Ranked Positional. Weight Method, Example 2.

CYCLE TIME		20	22	24	26	28	30	34	36	42	48	54
Max.Multiplicity	1	–	–	–	U_1^1	$B1_1^{323}$	U_1^2	$B1_1^3$	U_1^1	$B1_1^4$	$B1_1^7$	U_1^1
	2	$B1_2^{442}$	U_2^{12}	$B1_2^{264}$	$B1_2^4$	$B2_2^2$	$B1_2^{302}$	$B2_2^1$	U_2^1	$B2_2^{61}$	$B2_2^5$	U_2^5
	3	$B2_3^{426}$	U_3^{135}	$B2_3^{96}$	$B2_2^{21}$	$B3_2^{158}$	$B2_3^{432}$	$B3_2^{26}$	U_3^{171}	$B3_2^{147}$	$B3_2^{145}$	U_2^{145}
	4	$B3_4^{373}$	$B1_4^{30}$	$B3_4^{474}$	$B3_4^{131}$	$B4_4^{321}$	U_3^{107}	$B4_3^6$	U_3^{356}	$B4_2^{969}$	$B4_2^{961}$	U_2^{961}

FIGURE 6. Random Generation Method for Example 2.

number of elements/station are high enough. Since the number of elements required at a station will depend upon average element size and effective cycle time (cycle time × station multiple), it has been possible to prejudice the form of the solution in many cases without affecting its accuracy, and increasing production beyond the limit of longest element time has also proved no barrier. Using high element multiplicities is only profitable when the cycle time is small enough and the elements/station great enough, otherwise the solutions are 'degenerate' as we have seen, and great care is not needed in allocating multiplicities. For instance the results show that at cycle time 42, all multiplicities 4, and maximum 7 elements/station, the same solutions would be found as when using multiplicities of 2.

The *Random Generation* program was run for 1000 sequences maximum and a 7 element/station limit. It has provided optimum balances for all cycles except those with a theoretical minimum of zero, and still requires one extra operator for cycle time 54 when 9 elements/station is permitted, so that it is a little less satisfactory than the Optimal Station method. All 'unbalanced' places in the table were re-run with a different set of random numbers but only once (cycle time 22, multiplicity 3) did a 'perfect' balance ensue. On average the program inefficiency (number of sequences generated before optimum is produced) rises with increasing multiplicity due to many more options becoming available, and at the higher end of the cycle time range high multiples become redundant and lead to wastage of computing time. An essential difference between the two methods is that the Optimal Station program treats paralleling in a logical manner, and in order to make Random Generation more attractive the selection process for station multiples should not be completely random but be weighted to reflect such factors as the average element size, cycle time, and maximum number of elements/station.

All programs were run on an ICL 1909 computer and balances obtained for a whole series of cycle times during one run. The Optimal Station program solves the 11 element problem in approximately two seconds on average, the 29 element in seven seconds for multiplicities of 1 increasing to twenty seconds for 4, and a 42 element problem of mixed 1 and 2 multiplicities in twenty seconds. The reduced version approximately doubles these times, which are quite acceptable and indicate that much larger lines can be tackled using this particular computer. The Random Generation program takes about ten minutes to compute 1000 sequences for Example 2, so that it is much slower on average and would require a faster computer in order to process larger lines. It is capable of course of giving many alternative balances for any cycle time during one run, without altering the program or the input data, but it is not possible to cover the same number of cycle times.

A sample of the solutions recorded is given below with the station multiples in brackets, all from Example 2:

1. Optimal Station, Cycle Time 36, Multiplicities 3, 7 elements/station, Reduced Forward Program.
(1) 1, 2, 6; (1) 3, 4, 7, 9, 10 (LH); (1) 5, 12, 15; (1) 19, 25, 26, 27; (1) 8, 13, 14, 16 (RH); (1) 11, 17, 21 (LH); (3) 18, 20, 22, 23, 24, 28, 29. Idle Time 0%.

2. Optimal Station, Cycle Time 54, Multiplicities 2, 9 elements/station, Forward Program.
(1) 1, 2, 3, 4, 7 (LH); (2) 5, 6, 8, 9, 11, 12, 13, 17, 25 (RH); (1) 10, 14, 15, 16, 19, 21, 26, 27 (LH); (2) 18, 20, 22, 23, 24, 28, 29. Idle Time 0%.

3. Optimal Station, Cycle Time 26, Multiplicities 2, 7 elements/station, Reduced Forward Program.
(1) 1, 2; (1) 6, 7, 9, 10 (LH); (1) 3, 4, 14, 26 (LH); (1) 5, 25; (1) 8, 12; (1) 13, 16, 27 (RH); (2) 11, 15, 18; (1) 19, 17, 21 (LH); (1) 20, 22; (1) 23; (1) 24, 28; (1) 29. Idle Time 4.14%.
This sequence has only one double station and this has no spatial restrictions, which should make layout easier, especially for a conveyor belt system.

4. Optimal Station, Cycle Time 26, Multiplicities 3, 7 elements/station, Reverse Program.
(3) 29, 28, 27, 26, 24, 23; (3) 25, 22, 21, 20, 19, 17, 15 (LH); (1) 18, 13 (RH); (3) 16, 14, 11, 8, 6, 2; (1) 12, 10, 9, 7 (LH); (2) 5, 4, 3, 1 (LH). Idle Time 4.14%.
This sequence has only six distinct operations and may be suitable on an unpaced line if several groups of stations is found to be preferable to a very long line.

5. Random Generation, Cycle Time 26, Multiplicities 4, 7 elements/station.
(2) 2, 1, 3, 7, 9; (2) 4, 5, 12, 10, 14, 15 (LH); (2) 19, 21, 6, 8, 25 (LH); (3) 11, 13, 17, 20, 26, 16 (RH); (4) 18, 22, 27, 23, 24, 28, 29. Idle Time 4.14%.
This sequence can be run at cycle time 25.5 as this is the greatest station time, with a corresponding balance loss of only 2.3%. Without multiple stations it would be fruitless to even select a cycle time for thirteen operators with a minimum loss of $<4.14\%$, as we are limited to integer values only.

6. Optimal Station, Cycle Time 41, Multiplicities 1, 7 elements/station, Reverse Program.
29, 28, 24, 23, 20; 25, 22, 21, 18 (LH); 27, 19, 17, 16; 15, 26, 14, 11; 13, 8, 5 (RH); 12, 10, 9, 7, 6, 4 (LH); 3, 2, 1. Idle Time 1.22%.
This cycle time was chosen to match the worked example in (17), and an equally good balance found.

References

1. Arcus, Albert L., "COMSOAL—A Computer Method of Sequencing Operations for Assembly Lines", *The International Journal of Production Research*, Vol. 4, No. 4, (1966), pp. 259–277.
2. Brady, W., and Drury, C. G., "The Dependence of the Coefficient of Variation of a Work-Time Distribution on the Number of Elements in the Work Task", *The International Journal of Production Research*, Vol. 7, No. 4, (1969), pp. 311–315.
3. Caruso, Franklyn R., "Assembly Line Balancing for Improved Profits", *Automation*, Vol. 12, (January 1965), pp. 48–52.
4. Chaffin, D. B., and Hancock, W. M., "Factors in Manual Skills Training", *M.T.M. Research Studies Report*, 114, M.T.M. Association, (1966).

5. Helgeson, W. B., and Birnie, D. P., "Assembly Line Balancing Using the Ranked Positional Weight Technique", *The Journal of Industrial Engineering*, Vol. 12, No. 6, (November-December 1961), pp. 394–398.
6. Heskiaoff, Heskia, "An Heuristic Method for Balancing Assembly Lines", *Western Electric Engineer*, (October 1968), pp. 9–16.
7. Ignall, Edward J., "A Review of Assembly Line Balancing", *The Journal of Industrial Engineering*, Vol. 16, No. 4, (July-August 1965), pp. 244–254.
8. Jackson, James R., "A Computing Procedure for a Line Balancing Problem", *Management Science*, Vol. 2, No. 3, (April 1956), pp. 261–271.
9. Kilbridge, Maurice D., and Wester, Leon, "The Balance Delay Problem", *Management Science*, Vol. 8, No. 1, (October 1961), pp. 69–84.
10. ——, "A Heuristic Method of Assembly Line Balancing", *The Journal of Industrial Engineering*, Vol. 12, No. 4, (July-August 1961), pp. 292–298.
11. ——, "Non-productive Work as a Factor in the Economic Division of Labour", *The Journal of Industrial Engineering*, Vol. 12, No. 3, (May-June 1961), pp. 155–159.
12. ——, "An Economic Model for the Division of Labour", *Management Science*, Vol. 12, No. 6, (February 1966), pp. B 255–B 269.
13. Lehman, Melvin, "What's Going On In Product Assembly", *Industrial Engineering* (April 1969), pp. 41–45.
14. Mariotti, John, "Four Approaches to Manual Assembly Line Balancing", *Industrial Engineering* (June 1970), pp. 35–40.
15. Mastor, Anthony A., "An Experimental Investigation and Comparative Evaluation of Production Line Balancing Techniques", *Management Science*, Vol. 16, No. 11, (July 1970), pp. 728–746.
16. Salveson, M. E., "The Assembly Line Balancing Problem", *The Journal of Industrial Engineering*, Vol. 6, No. 3, (May-June 1955), pp. 18–25.
17. Sawyer, J. H. F., "*Line Balancing*", Machinery Publishing Co., Brighton, (1970).
18. Van Beek, H. G., "The Influence of Assembly Line Organization on Output, Quality and Morale", *Occupational Psychology*, (1964), Vol. 38, pp. 161–172.
19. Wild, Ray, and Slack, N. D., "The Operating Characteristics of 'Single' and 'Double' Non-mechanical Flow Line Systems", *International Journal of Production Research*, Vol. 11, No. 2, (1973), pp. 139–145.