

Division of Research
School of Business Administration

October 1989

LOAD SMOOTHING IN ASSEMBLY LINES

Working Paper #619

Ram Rachamadugu
Brian Talbot
The University of Michigan

ABSTRACT

In this paper we address the problem of levelling the workload for a given number of workstations in an assembly line. Most heuristic and optimal procedures used in designing the assembly lines are greedy and tend to load earlier stations more than later stations. We characterize dominant solutions and these characterizations are incorporated into an iterative procedure. We report computational results. When used in conjunction with currently available assembly line procedures, our procedure provides significant leveling of workloads. Future research directions are indicated.

LOAD SMOOTHING IN ASSEMBLY LINES

1. Introduction

Assembly lines are widely used for high volume production of goods and services. Substantial amount of research effort has been devoted in the past to the efficient design and operation of the assembly systems. Most of the earlier research focused on the development of enumerative and heuristic methods for minimizing the number of workers required for a desired production rate or minimizing the cycle time for a given number of workstations. Recent advances in these methods (Johnson [1988]) are useful for finding optimal and/or near optimal solutions to practical size problems. However, in practice, many aspects are to be considered in the design of assembly lines such as zoning, task compatibility, safety, tool requirement overlaps, etc. (Johnson [1983]). While some of these considerations can easily be dealt with using prior methods, some need development of special procedures. We address one such issue here--smoothing the workload among the given workstations.

Workload smoothing has quite a few beneficial features associated with it. In a recent paper, Farber, Luss and Yu [1987] described the need for workload smoothing in printed circuit board assembly plants. In case of manual lines, the notion of equity warrants it. It is common practice to attempt to assign nearly identical workloads to each station in order to provide comparable work and hence slack time to each person. Uneven assignments are viewed as inherently unfair and usually call for some kind of compensatory action on the part of management such as differential pay. Further, Smunt and Perkins [1985] have shown that where line lengths are long and/or task times variance is low, it is best to allocate the tasks as evenly as possible among the workstations to maximize the output. These considerations motivate the need to develop workload smoothing procedures.

Our paper is organized as follows: in the next section, we define the problem, and develop criteria for measuring workload smoothness. In Section 3, we discuss prior research. In Section 4, we provide characterizations of dominant solutions for the problem

and develop procedures for solving the problem. In Section 5, we provide computational results. Finally, we discuss future research directions.

2. Problem Description

The line designer is faced with the problem of designing a paced assembly line to provide specified output rate of the product. The output rate defines the maximum cycle time for the line. The primary objective of the line designer is to use as few stations as possible. The secondary objective is to distribute the workload as evenly as possible among the workstations on the line. There are many alternative criteria which can be used as surrogates for the secondary objective. Some prominent ones are listed below (notation used in this paper is shown in Table 1).

(i) **Workload Range**

$W_{\max} - W_{\min}$, where W_{\max} and W_{\min} represent maximum and minimum station workloads respectively.

(ii) **Cumulative Difference**

Make a cumulative plot of the workloads (Figure 1). Also, plot the cumulative workload if it were evenly distributed. The area between the two curves is a measure of uneven workload distribution.

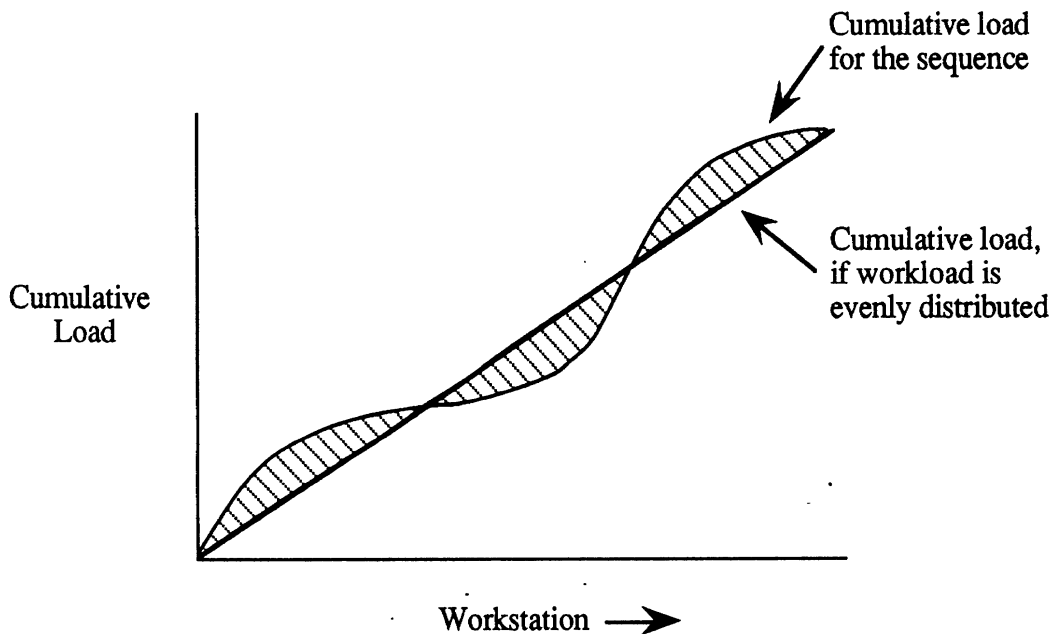


FIGURE 1

TABLE 1

Notation

D	: Demand rate for the product
$n(D)$: Number of stations in the line to achieve the desired output rate of D . This may be heuristic or optimal solution
n^*	: Minimum value of $n(D)$. $n(D) \geq n^*$
c	: Cycle time ($c = 1/D$)
N	: Number of tasks
i	: Task index $i = 1, .2, \dots, N$
t_i	: Processing time for task i
T	: Work content
	: $\sum_{i=1}^N t_i$
P_i	: Set of tasks immediately preceding task i in the precedence network
A_i	: Set of tasks immediately succeeding task i
x_{ij}	: 1 if task i is assigned to station j 0 otherwise
W_j	: Total work assigned to station j
\bar{W}	: Mean workload = $T/n(D)$
U_j	: Underload at station j
	: $\max(0, \bar{W} - W_j)$
O_j	: Overload at station j
	: $\max(0, W_j - \bar{W})$
AD_j	: Absolute deviation of the workload at station j from mean workload
	: $U_j + O_j$
MAD	: Mean absolute deviation of workloads
	: $(1/n(D)) \sum_{j=1}^{n(D)} AD_j$
$E(i)$: Index of the earliest station to which one (or more) of the immediate successors of task i has (have) been assigned
$L(i)$: Index of the latest station to which one (or more) of the immediate predecessors of task i has (have) been assigned
$S(i)$: Index of the station to which task i is currently assigned

(iii) Workload Variance (V)

$$(1/n(D)) \sum_{j=1}^{n(D)} \left(W_j - \frac{T}{n(D)} \right)^2$$

(iv) The sum of absolute deviations of workloads (MAD)

$$\sum_{j=1}^{n(D)} \left| W_j - \frac{T}{n(D)} \right|$$

Although all of these measures are potential candidates, they are clearly not identical. (i) measures only the extreme values of the workloads without regard to the distribution of workload among the workstations. For this reason, (ii) is likely to be better than (i). But the area between the curves depends on the sequence of workloads. For example, suppose that two neighboring workstations have been assigned precedent free tasks. Interchanging them may change the measure though neither improvement nor deterioration has taken place in the workload distribution. Criteria (iii) and (iv) overcome both these deficiencies. They are similar to each other, but workload variance penalizes deviations quadratically whereas absolute deviation applies a linear penalty. From a practical perspective, we can see no reason to choose one over the other. We chose (iv) because it seems reasonable from an equity point of view and it has certain desirable analytical properties.

2.1 Mathematical Programming Formulation

Using the sum of absolute deviations as the criterion, we can formulate the problem as a mixed integer programming problem. Problem formulation is given below:

$$\min \sum_{j=1}^{n^*} (U_j + O_j) \tag{1}$$

s.t.

$$\sum_{j=1}^{n^*} x_{ij} = 1 \quad \forall i = 1, 2, \dots, N \tag{2}$$

$$x_{kj} \leq x_{ij} \quad \forall k \in P_i, j = 1, \dots, n^* \quad (3)$$

$$\sum_{i=1}^{n^*} t_i x_{ij} \leq c \quad \forall j = 1, \dots, n^* \quad (4)$$

$$\sum_{i=1}^{n^*} t_i x_{ij} + U_j - O_j = \frac{T}{n^*} \quad \forall j = 1, \dots, n^* \quad (5)$$

$$x_{ij} = 0 \text{ or } 1$$

Constraint set (2) implies that each task has to be completed. Precedence relations are observed in (3). Constraint set (4) ensures that the cycle time restriction is not violated. Structural equations to measure relative underloading and overloading of the workstations are given by (5).

Though, in principle, the above problem can be solved using any standard mixed integer programming code, the number of binary variables and constraints are too large to solve the problem optimally for practical size problems. Hence the situation warrants alternate approaches to solving this problem. A categorization of possible approaches is shown in Figure 2.

3. Prior Research

Most prior research on assembly lines has focused attention on minimization of the number of workstations required to maintain a desired output rate or maximize the output rate for a given number of workstations. For a recent survey of exact procedures for solving these problems, the reader is referred to the study by Baybars [1986]. Talbot et al. [1986] provide a comparative evaluation of various heuristics for these problems. Almost all these procedures are single pass procedures. Arcus [1966] suggested a multi-pass biased search method. These studies do not take into consideration equitable allocation of tasks among workstations. Some results stated in this paper appear in a preliminary form in Rachamadugu and Talbot [1987].

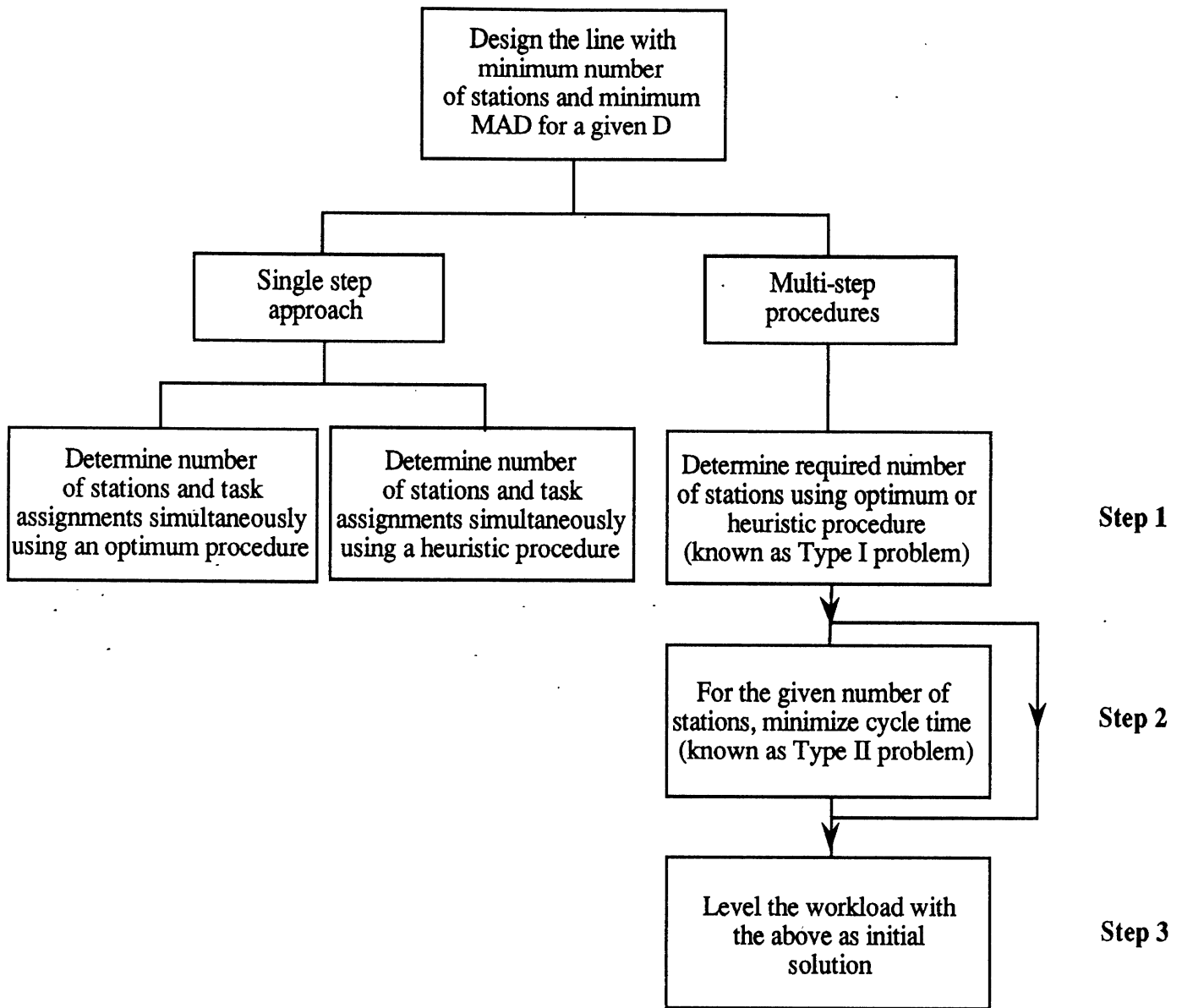


FIGURE 2

Alternate Approaches to Solving the Dual Criteria Problem

There are many heuristic procedures available for solving assembly line balancing problems (see Talbot et al. [1986]). Almost all of them are greedy in the sense that they assign as much work as possible to the early stations so that the number of stations is reduced. This results in more idle time appearing in later stations. However, an exception to this statement is a modification to the Hoffmann procedure [1963] suggested by Gehrlein and Patterson [1978]. Original Hoffmann's procedure for Type I problems is as follows: from the available tasks, a subset is selected such that the current workstation is loaded as much as

possible. This is done through complete enumeration. After a station has received its best assignment of tasks, the procedure is repeated until all tasks are assigned. It can be noticed that such a procedure tends to concentrate idle time either at the first few stations or the last few stations depending upon whether a forward or reverse network problem is solved. Gehrlein and Patterson [1978] noted that in Hoffmann's procedure enumeration of all maximal feasible combinations at each station can be computationally costly. So they modified Hoffmann's procedure as follows: curtail the search for the set of tasks to be assigned to a workstation when the idle time at the station falls within a given value

$$\text{Idle time} \leq \theta \left\{ \text{cycle time} - \frac{\text{total work content}}{\text{theoretical minimum number of stations}} \right\}$$

where θ is a parameter. Gehrlein and Patterson [1978] found that this modification not only reduced the computational time, but also smoothed the idle time present among workstations for certain classes of line balancing problems. Prior studies by Gehrlein and Patterson [1978] and Talbot et al. [1986] suggest use of θ value in the range [0, 2.0]. Since this heuristic has the desirable property of workload smoothing, it is appropriate to investigate this procedure.

Yet another approach to workload smoothing is to use a multi-step approach (Figure 2). In the first step, required number of stations can be determined by using either an optimum or heuristic procedure. In the next step, a Type II problem is solved to minimize the cycle time for the number of workstations determined in the previous step. This is intuitively appealing since minimizing the cycle time for a given number of stations would force more work to be assigned to later stations. However, procedures designed for Type II problems solve the problem as a series of Type I problems. Hence this can still result in underloading of later stations. Thus, whether or not solution to a Type I problem has been resolved as a Type II problem, need exists for developing a procedure for smoothing workload among a given number of workstations. Such a procedure can be used as a post-processor on the following solutions:

- i) Solution of a Type I problem (heuristic or optimum).
- ii) Solution from Type I problem (heuristic or optimum) solved as a Type II problem (heuristic or optimum).

In the next section we describe characterizations of dominant solutions for workload smoothing for a given number of stations.

4. Characterizations of Dominant Solutions

Given any heuristic or optimal solution to a Type I or Type II problem, it is generally necessary to reassign tasks among the workstations to smooth out workloads across the stations.

Proposition 1: Task i can be transferred from station j to station k ($k > j$) only if

$$E(i) \geq k \text{ and } W_k + t_i < c \quad \forall i$$

Proposition 2: Task i can be transferred from station k to j only if

$$L(i) \leq j \text{ and } W_j + t_i < c \quad \forall i$$

Proposition 3: If the work content of a pair of stations is either above or below \bar{W} , reassignment of tasks among these two stations will not improve MAD.

It is clear that Propositions 1 and 2 follow directly from precedence feasibility and cycle time considerations. Proposition 3 can easily be verified and omitted here for the sake of brevity. It follows directly from the property of absolute deviations.

Proposition 4: Consider any two stations j and k . MAD can be improved by shifting task i from station j to station k if

$$W_j - t_i > W_k, \quad W_j > \bar{W}, W_k < \bar{W}$$

Proof: By Proposition 3, it is clear that to reduce MAD we can transfer tasks from a station with higher than mean workload content to a station with lower than mean

workload content. Suppose we transfer task i from station j to k . Further, $W_j - t_i > W_k$. We have to consider four cases. They are shown in Table 2.

TABLE 2

Case 1	Case 2	Case 3	Case 4
$W_j > \bar{W}$, $W_k < \bar{W}$	$W_j > \bar{W}$, $W_k > \bar{W}$	$W_j < \bar{W}$, $W_k > \bar{W}$	$W_j < \bar{W}$, $W_k < \bar{W}$

Note that prior to the transfer of task i from station j to k , $W_j > \bar{W}$ and $W_k < \bar{W}$. It can easily be seen that MAD decreases in case 1. Case 2 is shown in Figure 3.

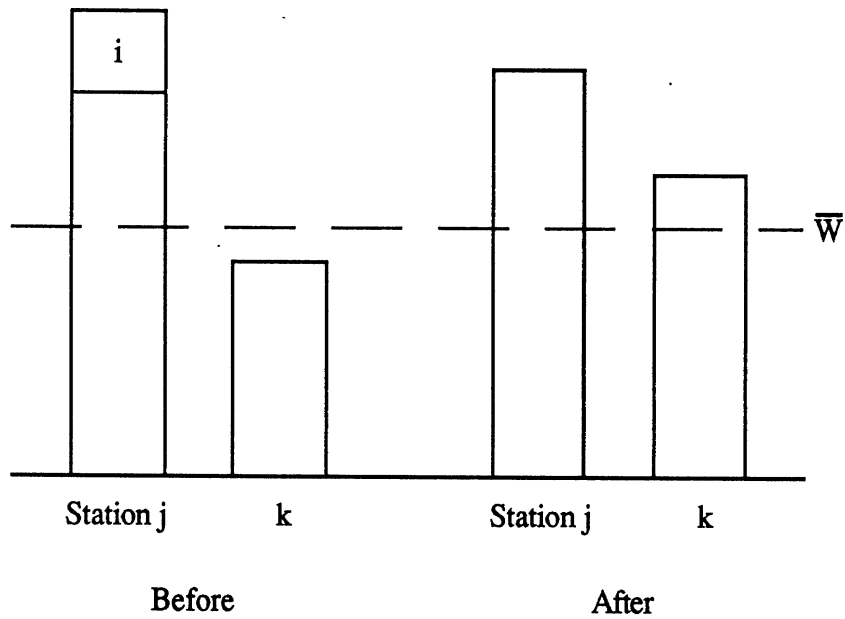


FIGURE 3

$$AD_j + AD_k \text{ before transfer} = (W_j - \bar{W}) + (\bar{W} - W_k) = W_j - W_k \quad \text{————— (1)}$$

$$\begin{aligned} AD_j + AD_k \text{ after transfer} &= (W_j - t_i - \bar{W}) + (W_k + t_i - \bar{W}) \\ &= W_j + W_k - 2\bar{W} \quad \text{————— (2)} \end{aligned}$$

It can easily be seen that $(1) - (2) \geq 0$. Thus the transfer results in an improvement in MAD. Case 3 is shown in Figure 4.

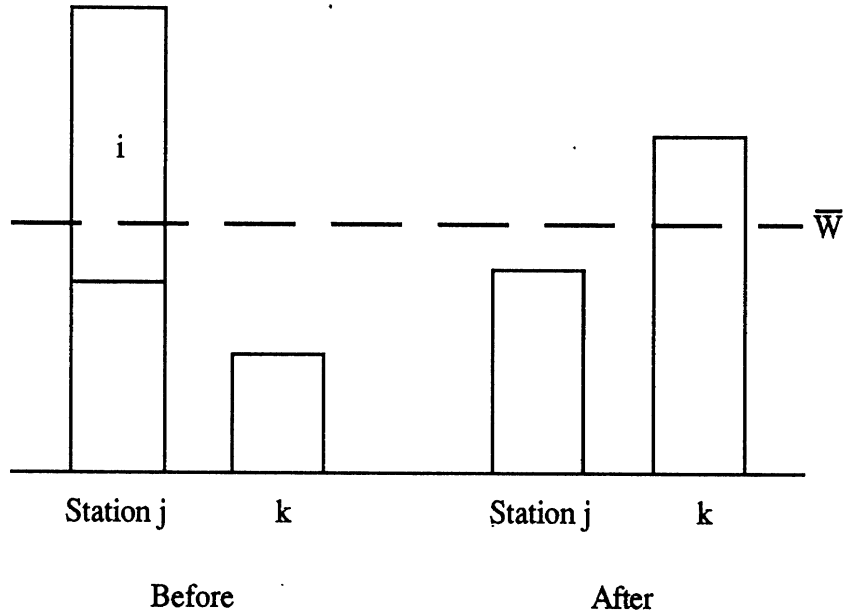


FIGURE 4

$$AD_j + AD_k \text{ before transfer} = (W_j - \bar{W}) + (\bar{W} - W_k) = W_j - W_k \quad \text{_____} \quad (3)$$

$$\begin{aligned} AD_j + AD_k \text{ after transfer} &= \bar{W} - (W_j - t_i) + (W_k + t_i - \bar{W}) \\ &= W_k - W_j + 2t_i \quad \text{_____} \quad (4) \end{aligned}$$

It is obvious that $(3) - (4) \geq 0$ and hence results in an improvement. Case 4 is shown in Figure 5.

$$AD_j + AD_k \text{ before transfer} = W_j - \bar{W} + \bar{W} - W_k = W_j - W_k \quad \text{_____} \quad (5)$$

$$\begin{aligned} AD_j + AD_k \text{ after transfer} &= \bar{W} - (W_j - t_j) + \bar{W} - (W_k + t_i) \\ &= 2\bar{W} + W_j - W_k \quad \text{_____} \quad (6) \end{aligned}$$

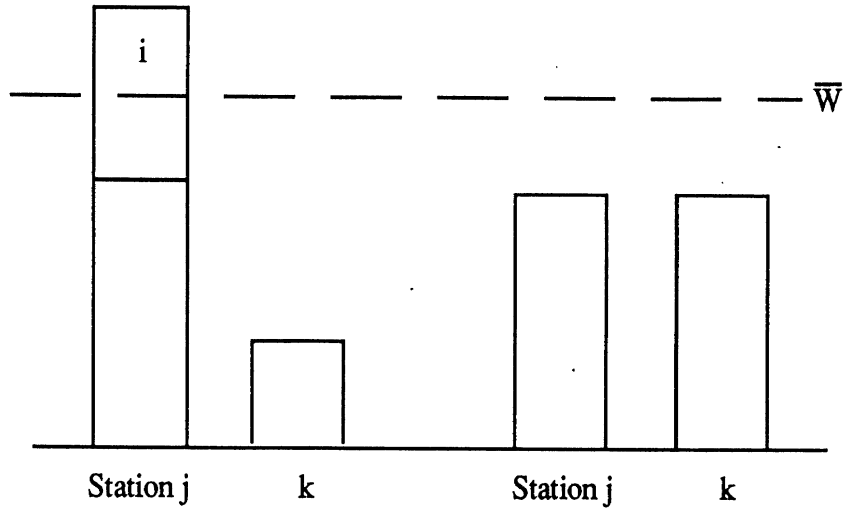


FIGURE 5

Comparing (5) and (6), it can easily be verified that $(5) - (6) \geq 0$. Hence, in all cases, there is an improvement in MAD by transferring tasks from station j to k. ■

While attempting to transfer tasks from one station to another, precedence feasibility can easily be verified by using Propositions 1 and 2. If Propositions 3 and 4 are used in selecting the tasks and stations, there is no need to verify cycle time feasibility. If the original assignment was cycle time feasible, the revised assignment is also cycle time feasible.

So far, we investigated only situations where MAD could be improved by transferring a task from one station to another. Next, we describe a condition in which simultaneous exchange of one task from a station with a set of tasks from another station can improve MAD.

Proposition 5: Suppose we wish to transfer task i from station j to k $(W_j > \bar{W} > W_k)$. At the same time, we transfer a set of tasks from k to j . Call this set s . This interchange improves MAD if the following condition holds good

$$(W_k - W_j + t_i) \leq \sum_{q \in s} t_q \leq t_i$$

Proof: Note that the contribution to MAD by any one station is a convex function of the workload at that station. So, contribution to MAD from station k is a convex function of the work content of s. Similarly, contribution to MAD from station j is also a convex function of the work content of s. Since the sum of any two convex functions is convex, total contribution to MAD from stations j and k is a convex function of the work content of s. We now show that outside the bounds stated in this proposition, contribution to MAD from stations j and k are higher than prior to interchange. Due to convexity property, this implies that within the bounds, contribution to MAD cannot be worse than before the interchange.

Case 1: Suppose $\sum_{q \in s} t_q > t_i$. This situation is shown in Figure 7.
 contribution to MAD before interchange

$$= (W_j - \bar{W}) + (\bar{W} - W_k) = W_j - W_k$$

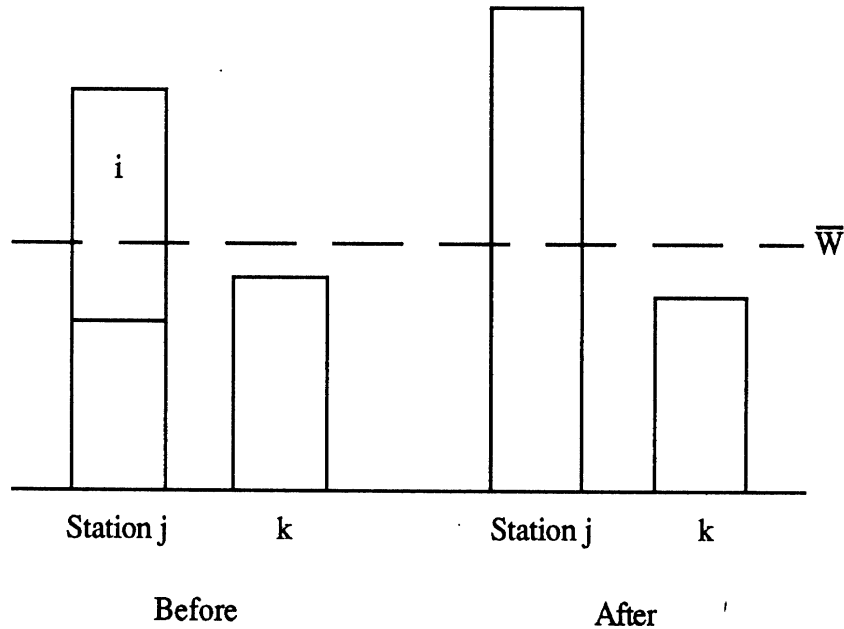


FIGURE 6

contribution to MAD after interchange

$$= \left(W_j + \sum_{q \in s} t_q - t_i - \bar{W} \right) + \left(\bar{W} - \left(W_k + t_i - \sum_{q \in s} t_q \right) \right)$$

$$= W_j - W_k + 2 \left(\sum_{q \in s} t_q - t_i \right)$$

Since $\sum_{q \in s} t_q \leq t_i$, we are worse off than prior to the interchange.

Case 2: Suppose $\sum_{q \in s} t_q \leq W_k - (W_j - t_i)$. This situation is shown in Figure 7.

contribution to MAD from station j and k after interchange

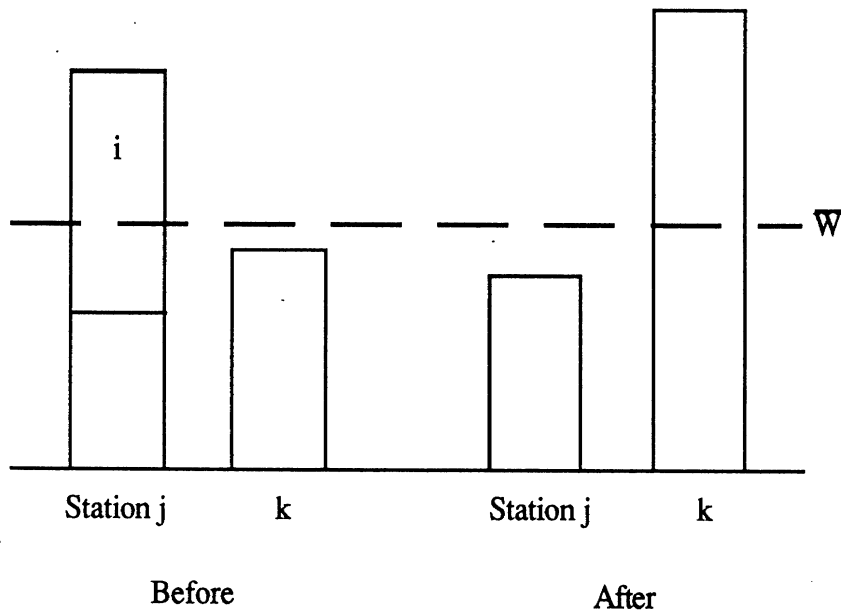


FIGURE 7

$$\begin{aligned}
 &= \left(\bar{W} - \left[W_j - t_i + \sum_{q \in s} t_q \right] \right) + \left(W_k + t_i - \sum_{q \in s} t_q - \bar{W} \right) \\
 &= 2 \left(t_i - \sum_{q \in s} t_q \right) + W_k - W_j \\
 &= W_j - W_k + 2 \left(t_i - \sum_{q \in s} t_q - W_j + W_k \right) \\
 &= W_j - W_k + 2 \left(W_k + t_i - W_j - \sum_{q \in s} t_q \right)
 \end{aligned}$$

Since $\sum_{q \in s} t_q \leq W_k - (W_j - t_i)$ (by definition for this case), above expression can be rewritten as

$$\geq W_j - W_k.$$

Hence within the bounds, contribution of stations i and k to MAD after interchange can be no more than prior to interchange. This completes the proof. ■

It may be noted that in an interchange using Proposition 5, it is not necessary to verify cycle time feasibility. If the original assignment was cycle time feasible, reassignment using Proposition 5 is also cycle time feasible. However, precedence feasibility needs to be verified. In our implementation, we restricted our choice to tasks to be included in s to those which did not have immediate successors assigned to earlier than j (if $k < j$) or to those which did not have immediate predecessors assigned to later than j (if $j < k$). Further tasks at k were inspected to SPT order and qualifying tasks were accumulated and whenever Proposition 5 was satisfied, further search for inclusion of tasks in S was terminated. It is possible to devise better search methods for determining s at the cost of larger computational time.

Using the characterizations described in Propositions 1-5, we developed an iterative heuristic procedure which can be used with heuristic and/or optimal solutions to Type I and Type II problems. The procedure is described below.

Step 0: $l \leftarrow n(D)$

Step 1: If $l = 0$ go to step 4

If $W_l > \bar{W}$ go to step 3

Step 2a: If available, choose the largest task i such that

$$\left\{ W_{S(i)} \geq \bar{W} \right\} \text{ and } \left\{ W_{S(i)} - t_i > W_l \right\}$$

and

either $\{E(i) \geq l \quad \text{if } l > S(i)\}$ or $\{L(i) \leq l \quad \text{if } l < S(i)\}$

Transfer i from $S(i)$ to l and update S , E , L and W . Go to step 3. Otherwise, go to step 2b.

Step 2b: If available, choose largest task i such that

$$W_{s(i)} - t_i < W_l.$$

Interchange tasks between $s(i)$ and l by invoking Proposition 5, subject to precedence feasibility. Update s , E , L , W .

Step 3: $l \leftarrow l - 1$

Go to step 1

Step 4: Stop.

Steps 0 through 4 are repeated until there is no further improvement in the MAD value.

It can easily be seen that step 2a in the above heuristic invokes Propositions 1-4 and step 2b invokes Proposition 5.

5. Computational Results

In order to validate our procedure, we tested it on solutions for sixty-four problems from the literature (Talbot et al. [1986]). Our procedure was coded in FORTRANVS (IBM version of Fortran 77) and implemented on IBM 3090-600 at The University of Michigan, Ann Arbor, Michigan. The code optimizer was invoked at level 3. These are Type I problems for which optimal number of stations are known. Beneficial effects of using our procedure on the results of Type I problems are shown in Table 2. Use of our procedure resulted, on average, in 33.65% reduction in MAD. If we exclude the cases where initial MAD was zero (MAD value cannot be less than 0 and hence the solution is also optimal for MAD) reduction in MAD due to the use our procedure is 37.13%.

TABLE 2

Results of Using Workload Leveling Procedure
on Optimal Solutions for Type I Problems

Problem	Cycle Time	MAD before the use of leveling procedure	MAD after the use of leveling procedure	Percent reduction in MAD
Merten's 7-Element	6	0.8889	0.8889	0.00
Merten's 7-Element	7	0.6400	0.6400	0.00
Merten's 7-Element	8	1.0400	0.6400	38.46
Merten's 7-Element	10	0.4444	0.4444	0.00
Merten's 7-Element	15	0.5000	0.5000	0.00
Merten's 7-Element	18	2.5000	0.5000	80.00
Bowman's Problem	20	1.6000	1.6000	0.00
Jaeschke's 9-Element	6	0.8750	0.8750	0.00
Jaeschke's 9-Element	7	0.8980	0.6939	22.72
Jaeschke's 9-Element	8	1.2222	0.8889	27.27
Jaeschke's 9-Element	10	0.3750	0.3750	0.00
Jaeschke's 9-Element	18	4.2222	3.1111	26.31
Jackson's 11-Element	7	1.0000	1.0000	0.00
Jackson's 11-Element	9	1.4444	1.0000	30.76
Jackson's 11-Element	10	0.9600	0.6400	33.33
Jackson's 11-Element	13	1.5000	1.0000	33.33
Jackson's 11-Element	14	1.5000	1.5000	0.00
Jackson's 11-Element	21	7.5556	0.8889	88.23
Dar-El's Problem	48	1.7500	0.8750	50.00
Dar-El's Problem	62	0.4444	0.4444	0.00
Dar-El's Problem	94	0.5000	0.5000	0.00
Mitchell's 21-Element	14	0.6563	0.4375	33.33
Mitchell's 21-Element	15	2.0938	1.1563	44.77
Mitchell's 21-Element	21	0.0	0.0	0.00
Mitchell's 21-Element	26	3.6000	0.4000	88.88
Mitchell's 21-Element	35	0.0	0.0	0.00
Mitchell's 21-Element	39	2.6667	2.6667	0.00
Heskiaoff's 28-Element	138	14.0000	6.0000	57.14
Heskiaoff's 28-Element	205	0.3200	0.3200	0.00
Heskiaoff's 28-Element	216	12.2400	2.4800	79.73
Heskiaoff's 28-Element	256	0.0	0.0	0.00
Heskiaoff's 28-Element	324	92.0000	0.5000	99.45
Heskiaoff's 28-Element	342	0.4444	0.4444	0.00
Sawyer's 30 Element	25	1.9796	1.0408	47.42
Sawyer's 30 Element	27	2.1183	0.8639	59.21
Sawyer's 30 Element	30	2.6667	1.333	50.00
Sawyer's 30 Element	36	4.2400	2.2000	48.11
Sawyer's 30 Element	41	0.6250	0.6250	0.00
Sawyer's 30 Element	54	9.8776	7.3061	26.03
Sawyer's 30 Element	75	11.8400	3.7600	68.24
Kilbridge & Wester	57	0.8400	0.4800	42.85
Kilbridge & Wester	79	0.2449	0.2449	0.00
Kilbridge & Wester	92	0.0	0.0	0.00
Kilbridge & Wester	110	29.0000	25.3333	12.64

(Table 2 continued on next page)

TABLE 2 (continued)

Problem	Cycle Time	MAD before the use of leveling procedure	MAD after the use of leveling procedure	Percent reduction in MAD
Kilbridge & Wester	138	0.0	0.0	0.000
Kilbridge & Wester	184	0.0	0.0	0.00
Tonge's 70-Element	176	7.4694	5.6735	24.04
Tonge's 70-Element	364	18.0000	3.0000	83.33
Tonge's 70-Element	410	27.3333	11.3333	58.53
Tonge's 70-Element	468	44.4375	12.2500	72.43
Tonge's 70-Element	527	32.1225	1.6326	94.91
Arcus' 83-Element	5048	298.3438	242.1797	18.82
Arcus' 83-Element	5853	467.7566	426.8994	8.73
Arcus' 83-Element	6842	576.6406	309.1262	46.39
Arcus' 83-Element	7571	740.5298	393.6860	46.84
Arcus' 83-Element	8412	1018.4805	489.5601	51.93
Arcus' 83-Element	8898	377.0125	353.4570	32.77
Arcus' 83-Element	10816	1906.1875	1005.5938	47.27
Arcus' 111-Element	5755	162.4941	111.4570	31.41
Arcus' 111-Element	8847	733.2222	375.5554	48.78
Arcus' 111-Element	10027	919.9922	155.4297	83.10
Arcus' 111-Element	10743	1103.1487	461.4673	58.16
Arcus' 111-Element	11378	1129.6841	358.2756	68.28
Arcus' 111-Element	17067	502.8887	46.4444	90.76

Computational times in all cases were less than 100 milliseconds. Reduction in MAD in case of large size problems such as Tonge and Arcus problems can be substantial.

Next, we tested our procedure on the solutions of modified version of Hoffmann heuristic procedure for Type I problems as proposed by Gehrlein and Patterson [1978]. Computational experiments by Talbot et al. [1986] have shown that Hoffmann heuristic [1963] had provided better results than competing heuristics for Type I problems. Modified version of this heuristic is the only heuristic procedure that addresses the issue of smoothing workload while simultaneously attempting to minimize the use of workstations. Also, Gehrlein and Patterson [1978] reported that the use of θ factor (Section 2) spread the idle time evenly among workstations. Hence this procedure seemed a relevant benchmark against which procedure needs to be compared. We obtained solutions for the 64 literature problems using modified Hoffmann's procedure for various θ values. MAD for these

solutions was compared with the MAD obtained after using our procedure on these solutions. Summary of these results is given in Table 3.

TABLE 3

θ Value	0	1	2
Average % reduction in MAD resulting from the use of our heuristic	44.50	43.57%	43.26%

These results reported in Table 3 were somewhat counterintuitive. Larger θ values spread out idle time more evenly among the workstations and hence further reductions in MAD should be difficult to come by. However, even in these cases, reductions in MAD could be achieved with the use of our heuristic procedure. This is due to two factors. Firstly, using large θ values may distribute the workload more evenly, but in some cases they resulted in the use of more number of workstations than when $\theta \neq 0$. This observation is consistent with the computational studies done by Talbot et al. [1986]. This resulted in opportunities for further leveling the workload. Secondly, when $\theta = 0$ Hoffmann's procedure tends to pack earlier stations more densely than other heuristic or optimal procedures. It can easily be verified that the workload assigned to the first station by Hoffmann procedure for a Type I problem is no less than the workload assigned to the first station by any heuristic or optimal procedure (the statement holds good for the last station if reverse network is used). Though this is very useful in minimizing the number of workstations (thus explaining why it does better than other heuristic procedures [Talbot et al. 1986]), this also leads to greater opportunities for smoothing workloads.

We next tested the performance of our workload leveling procedure with Type II problems for which optimal solutions were known. These problems constitute tough benchmark since we would expect greater leveling of workloads with minimum cycle time

for a given number of stations. However, Type II problems are generally solved as a series of Type I problems. Since all optimum and/or heuristic procedures (with the known exception of modified Hoffmann procedure) for type I are greedy (in the sense that they load each station with as much work as possible subject precedence and chosen cycle time), it is conceivable that scope exists for workload leveling even in optimal solutions for Type II problems.

We used 23 test problems for which optimal cycle times were known (Wee and Magazine [1981]). In 5 cases, MAD for initial solutions was zero and hence optimal. In 7 other cases also, initial solution was optimal (optimality in these cases was verified by inspection--for example, if stations have work content equal to optimal cycle time or (optimal cycle time - 1), then the solution is optimal for MAD also). Results of using our procedure on the other 11 problems are shown in Table 4. On average, MAD improved by 22.36% in these cases. As mentioned earlier in our discussion, improvements resulting

TABLE 4
Solutions for Type II Problems for Which Minimal MAD
Assignments are Unknown

Problem	Number of Stations	Known Optimal Cycle Time	MAD Before the Use of Our Procedure	MAD After the Use of Our Procedure	% Reduction in MAD
Merten's Problem	5	7	0.6400	0.6400	0.00
	3	10	1.7500	1.2500	28.57
Jaeschke's Problem	7	7	0.8980	0.6939	22.72
Jackson's Problem	5	10	0.9600	0.6400	33.33
	4	12	0.5000	0.5000	0.00
	3	16	0.8889	0.4444	50.00
Mitchell's Problem	8	14	0.6563	0.4375	33.33
Sawyer's Problem	13	26	0.8876	0.8757	1.34
	8	41	0.7500	0.5000	33.33
Tonge's Problem	11	320	0.9917	0.9917	0.00
Arcus' Problem	9	16723	11.7778	6.6667	43.79

from the use of our procedure are less in the case of Type II problems than Type I problems.

6. Conclusion

In this paper we investigated an important problem in assembly line design--leveling of workloads across workstations. We characterized dominant solutions and incorporated these characterizations into a heuristic procedure. When used in conjunction with optimal or heuristic solutions for Type I and II problems, our procedure yielded better allocation of work across stations. Computational times for using our procedure are insignificant and in no case exceeded 100 milliseconds (on IBM 3090-600 system). Thus the procedure is computationally inexpensive. Effectiveness of our procedure was established through testing on solutions for Type I problems, modified Hoffmann procedure solutions and solutions for Type II problems. It yielded improvements in all categories, though improvements in case of Type II solutions were less than in other cases. Our procedure can also be used to accelerate convergence to optimality in Type II problems. We are currently investigating this aspect.

REFERENCES

- Arcus, A. L. (1966), "COMSOAL - A Computer Method of Sequencing Operations for Assembly Lines," International Journal of Production Research, Vol. 4, p. 259-277.
- Baybars, I. (1985), "A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem," Management Science, Vol. 32, p. 909-932.
- Farber, M., H. Luss, and C. S. Yu (1987), "Assembly Line Design Tools: Line Balancing and Line Layout," Proceedings of the Second International Conference on Production Systems, Paris, France, p. 209-220.
- Gehrlein, W. V., and J. H. Patterson (1978), "Balancing Single Model Assembly Lines: Comments on a Paper by E. M. Dar-El (Mansoor)," AIIE Transactions, Vol. 10, No. 1, p. 109-112.
- Hackman, S. T., M. J. Magazine, and T. S. Wee (1988), "Fast, Effective Algorithms for Simple Assembly Line Balancing Problems," Working Paper, Department of Management Sciences, University of Waterloo, Waterloo, Ontario N2L 3G1.
- Helgeson, W. B., and Birnie, D. P. (1961), "Assembly Line Balancing Using Ranked Positional Weight Technique," Journal of Industrial Engineering, Vol. 12, No. 6, p. 394-398.
- Hoffmann, T. R. (1963), "Assembly Line Balancing with a Precedence Matrix," Management Science, Vol. 9, No. 4, p. 551-562.
- Johnson, R. V. (1983), "A Branch and Bound Algorithm for Assembly Line Balancing Problems with Formulation Irregularities," Management Science, Vol. 29, p. 1309-1324.
- Johnson, R. V. (1988), "Optimally Balancing Large Assembly Lines with FABLE," Management Science, Vol. 34, No. 2, p. 240-253.
- Rachamadugu, R. V., and F. B. Talbot, "Assembly Line Balancing with Dual Criteria," Proceedings of the International Conference on Production Research, Cincinnati, Ohio, 1987.
- Smunt, T. L., and W. C. Perkins (1985), "Stochastic Unpaced Line Design: Review and Further Experimental Results," Journal of Operations Management, Vol. 5, No. 3, p. 351-373.
- Talbot, F. B., J. H. Patterson, and W. V. Gehrlein (1986), "A Comparative Evaluation of Heuristic Line Balancing Techniques," Management Science, Vol. 32, No. 4, p. 430-454.
- Wee, T. S., and M. J. Magazine (1981), "An Efficient Branch and Bound Algorithm for an Assembly Line Balancing Problem - Part II: Maximize the Production Rate," Working Paper No. 151, Department of Management Sciences, University of Waterloo, Waterloo, Ontario, Canada.