ANALYTICAL TOOLS FOR ASSEMBLY LINE DESIGN AND SEQUENCING

Ram Rachamadugu
School of Business Administration

Candace Arai Yano
Department of Industrial and Operations Engineering

The University of Michigan
Ann Arbor, MI 48109

Technical Report 89-28

September 1989
Revised May 1990, May 1991, and December 1991

ANALYTICAL TOOLS FOR ASSEMBLY LINE DESIGN AND SEQUENCING

ABSTRACT

Many complex assembly lines, such as those in the automobile industry, have dozens or hundreds of stations that are affected by customer-selected options on the jobs being assembled. The various options often require significantly different amounts of processing time, and the role of assembly line sequencing in this context is to smooth out the flow of work to each station. However, most assembly line sequencing algorithms developed for such situations cannot consider so many stations or options effectively. In this paper, we develop an analytical method to compute a criticality index for each station, which can be used to determine which stations are most important to include in an assembly line sequencing algorithm. We report computational results using actual industry data which indicates that substantial improvements can be obtained by selecting stations based upon this criticality index.

ANALYTICAL TOOLS FOR ASSEMBLY LINE DESIGN AND SEQUENCING

I. Introduction

Assembly lines are an efficient means to manufacture high volume products such as consumer appliances. However, there are many instances in which many variations of the same basic product are produced on the same assembly line. When the number of such variations is small and the demand rate for each is stable, existing mixed model assembly line techniques can be used to design and sequence the assembly line. However, situations exist in practice where the number of variations is so large that each unit is almost unique, making it impossible to use mixed model procedures. The U.S. automotive industry is a good illustration. For example, the total number of option combinations on the Buick Century exceed 176 trillion (Treece 1989). Similar conditions exist at other automotive assembly plants (Weiner 1985).

In practice, the problem of producing such a large number of combinations of options is handled in two phases. In the first (design) phase, the assembly line is designed and balanced to accommodate the anticipated mix of options. Since there are no systematic techniques to balance assembly lines when the number of models is large, it is common practice to "balance" the line *on the average* (i.e., the average job will be completed within the allocated time), with the additional proviso that stations with higher processing time variability should have lower average utilization levels. The additional slack provides a cushion for jobs with long processing times. Sometimes average utilization levels are as low as 50 percent to accommodate processing time variations. Another common practice is to provide a larger work space at stations with high processing time variances. This permits the assembler to work on the job farther down the assembly line than he/she could otherwise.

In the second (operational) phase, jobs are sequenced on the assembly line so as to smooth out the flow of work as much as possible. This can be accomplished, for example, by alternating jobs with long and short processing times. However, the best sequence for a particular workstation may not be good for another workstation. We addressed the problem of determining a good sequence in an earlier paper (Yano and Rachamadugu 1991). In that paper, we assumed that the assembly line had already been designed and that critical workstations with respect to sequencing could be easily identified.

The primary motivation for the work to date on assembly line design and sequencing has been to maintain both high quality levels (by ensuring that the assembly operators have adequate time to complete their tasks) and high labor productivity simultaneously. Most of the earlier research on problems with many different models has been focused on determining the appropriate input sequence to the assembly line (e.g., Coffman, Hoffman and Weiner 1985, Okamura and Yamashina 1979, Parrello 1988, Bolat 1988, Miltenberg 1989, Yano and Bolat 1989, Yano and Rachamadugu 1991).

In this paper, we develop an analytical method that can be used in two different and complementary ways: (i) to identify important workstations with respect to sequencing, and (ii) to estimate the consequences of not attempting to smooth the work at stations that are not considered in the sequencing algorithm. To the best of our knowledge, the former topic has not been studied before. The latter has been investigated via simulation by Koether (1985).

Since we are developing one model that can be used for two purposes, it is important to explain the context of the application. A few good heuristic sequencing procedures now exist for assembly lines with many combinations of options. These sequencing procedures can easily incorporate several dozen different options, thus also an enormous number of different models, if the models can be characterized by the presence or absence of options. (Optimal procedures are practical only for very small problems, or those with only one or two options.) If only a few stations are considered in the

sequencing procedures, they will provide a fairly smooth flow of work to each of these stations. This, in turn, reduces the workspace requirements and the required staffing at these stations. However, the flow of work to the stations excluded from consideration will be unpredictable. If the objective function for the sequencing procedure considers the impact of too many stations, it is difficult for even a good sequencing procedure to smooth out the flow of work to *all* stations because of the tradeoffs implicit in the objective function. Thus, the stations lengths and staffing must be set to accommodate this moderate level of variability throughout the system. This raises the questions of the "best" number of stations, and which stations, out of a few hundred in a typical automotive assembly line, to include in a heuristic sequencing procedure. This set of stations will be retained for several months to a year until a change in the product mix warrants a modification. These are the primary issues that we address in this paper.

The remainder of this paper is organized as follows: in the next section, we provide a description of the problem and the underlying assumptions. In section 3, we show that certain portions of the system can be modeled as a Markov process, which permits us to express an important performance measure in closed form. In section 4, we show how the performance measure mentioned above can be approximated from a regression model in some special (but quite common) circumstances. The regression model provides a quick, yet accurate, means to identify critical work stations. Section 5 describes an empirical study of the impact of station selection rules on the overall cost of operating the system. In the last section, we discuss our conclusions and provide directions for future research.

2. Problem Description

We are concerned with the sequencing of a wide variety of different models of the same general product on an assembly line. We assume that the line is paced, i.e., that the line moves at a constant speed and one job (unit) is launched to the line at equal time intervals known as the *cycle time*, T. Without loss of generality, we assume that the line

moves from left to right. Each work station (or *window*) consists of a contiguous portion of the assembly line in which the operator is to perform the assigned tasks. We assume that at any point in time, a job can be worked on by an operator (or a coordinated team of operators) only within a single station. In other words, operators at two or more stations cannot work on a job simultaneously. This reduces the chance of operators interfering with one another. We refer to the point at which jobs enter the station as the origin of the station.

We assume that the specific set of tasks to be performed on a job at a given station depends on the combination of options on that job. This can be assumed without loss of generality since the options can be defined in such a way that any number of different models can be considered. At some stations, the same tasks are performed for all combinations of options, but at many stations there are two or more sets of tasks, where each set corresponds to a particular combination of options. Thus, the processing time depends upon the combination of options. Some options affect multiple stations because the tasks associated with the option are spread across several workstations. The options are not necessarily binary choices. For example, when a customer orders an automobile, he/she may be able to choose one of several different types of engines. The long-run average mix of options is assumed to be known (or estimated) and to remain stable (although not perfectly constant) over the horizon under consideration.

We also assume that the staffing level and the assignment of tasks to work stations is prespecified. However, our approach can be used to evaluate these decisions, as we will explain later.

The operator works on each job in sequence, following it along the assembly line as it moves to the right. Whenever the operator completes a job or must stop working on it because it has left the station, he/she returns upstream to start working on the following job. (See Figure 1 for an illustration.) We assume that the travel time to the next job is negligible in comparison to the processing times and therefore need not be considered in the

analysis. Moreover, in practice, operators perform some work while walking, so it is often difficult to distinguish travel time from processing time. If the required work on a job cannot be completed during the sojourn time of the job in the station, then unfinished work may be completed at a downstream repair station. If there is insufficient repair capacity (sometimes caused by the strategic decision to eliminate repairers so as to encourage "doing it right the first time"), the operator is forced to work more quickly than normal and this often results in quality problems.

Insert Figure 1 here

These problems can be reduced or eliminated by sequencing the jobs so that each operator receives a smooth flow of work which is within his/her capacity to complete. It is clear that the amount of incomplete work (or the amount of quality deterioration) depends on not only the sequence of jobs, but also the length of the window. A longer window increases the sojourn time of each job in the station and permits smoothing of the flow of work over a larger number of consecutive jobs. The primary metric that we use in evaluating the quality of the sequence is what we call *work overload.* This is defined as the amount of work (processing time) that would not be completed if the operator were to work at the normal rate and stay within the boundaries of his/her station. Work overload provides a measure of the extent to which a sequence deviates from providing a smooth flow of work.

We have shown elsewhere (Yano and Rachamadugu 1991), and it is intuitively clear that for any *given* sequence, work overload is minimized by a policy of working on each job as long as possible before starting work on the subsequent job. Thus, we will assume that the schedule of *when* each job is worked on is specified in this fashion.

The goal of this paper is to develop approaches that permit strategic scheduling decisions and certain types of assembly line design decisions to be made more quickly and effectively when the number of option combinations is large. These approaches are based

upon a descriptive stochastic model of the system. This model provides both a measure of how critical a station is with regard to its inclusion in the scheduling algorithm, as well as an estimate of the controllable costs (repair and/or quality costs) if the station is not included in the scheduling algorithm. Notation used in this paper is described in Table 1.

Insert Table 1 here

3. Markovian Model of the System

In this section, we assume that jobs are sequenced randomly. This approximates the situation when there is no attempt to sequence the jobs. We develop a descriptive Markovian model from which we can compute the expected amount of work overload at a given station for a random sequence. In this model, the state of the system at "time" t is the distance of the operator from the origin of the station (or equivalently, location of the operator within the station) when he/she starts work on the job in the t^{th} position in the sequence. Since the assembly line moves at a constant speed, there is an equivalent state definition which specifies how long the job has been in the station, rather than its location, when work begins on it. The latter state definition is easier to use and we will do so here. Since the jobs require different processing times, the state of the system is not observed at equal time intervals, but at points in time when work on a new job commences. With very little loss of accuracy, let us assume that there are a finite number of equally spaced states. (Since processing times are specified with limited precision, e.g., to the nearest one thousandth of a minute, for practical purposes, the representation can be made exact by making the number of states large enough. Even if such a detailed state definition is not employed, the loss of accuracy in any time measurement, including work overload, is at most equal to the smallest time increment.)

For any rule that dictates when (or where) work on each job must terminate, it is easy to determine the probability that the operator will be in state j at "time" t + 1 given that

he/she was in state i at "time" t. Under our assumptions, the operator must stay within the limits of the station, and therefore must terminate work on a job when it reaches this boundary, even if work on it is not complete. Since the sequence is random, each job will have processing time τ_k with probability p_k, k = 1,...K. (If several option combinations have identical processing times, for the purposes of this model we can combine them into a single option which occurs with a probability equal to the the *sum* of the individual probabilities.)

If $i + \tau_k \leq L$, where L is the amount of time the job spends in the station, the job can be completed before it leaves the station, and it will be completed when job t has been in the station for a duration $i + \tau_k$. We assume that $L \geq T$. Otherwise, more than one operator could work on a job simultaneously, which violates our assumptions. In most practical applications, such situations are strongly discouraged to avoid interference between operators. (If a team of operators shares the work in a coordinated fashion within a single station, they can be treated as a single faster operator.) Since jobs are launched at intervals of T, job t+1 has been in the station for a duration of $(i + \tau_k - T)^+$. Thus, the transition from i to $(i + \tau_k - T)^+$ occurs with probability p_k if $i + \tau_k \leq L$. On the other hand, if $i + \tau_k > L$, then work on job t terminates when it has been in the station for a duration L. At that time job t+1 has been in the station for a duration L - T. Consequently, a transition from i to L - T occurs if $i + \tau_k > L$, which occurs with probability $\sum_{k \mid \tau_k > L-i} p_k$.

Observe that the transition probabilities depend only upon i (i.e., the state at "time" t), and consequently the system is Markovian. The general structure of the transition matrix for the case of two sets of tasks at a station is given in Figure 2. In this example, it is assumed without loss of generality that $\tau_1 > T > \tau_2$.

Insert Figure 2 here

With such a transition matrix, we can determine the long-run (steady state) probabilities of being in each of the states, π_i for i = 0,...L, using standard techniques (which involves solving a system of linear equations). Even if the number of states is large, this poses no difficulty because many available software packages for mainframe computers can solve systems of linear equations with thousands of variables. It is also useful to point out that some of the states are unreachable. This result is obvious from the description of the state transitions above. The system jumps from state i to state $(i + \tau_k - T)^+$ if $i + \tau_k \leq L$, k = 1,...K, or to state L - T otherwise. The maximum value of $(i + \tau_k - T)^+$ under the stated condition is L - T. Thus, only states in [0, L - T] are reachable and other states need not be considered in the analysis.

We can determine the expected work overload associated with each state in much the same way as the transition probabilities were developed. Conceptually, the expected overload associated with state i is the expected amount of incomplete work on a job which is started when it has been in the station for a duration i. If option combination k occurs and $i + \tau_k > L$, the amount of work overload is $i + \tau_k - L$. If $i + \tau_k \leq L$, there is no work overload. Thus, the expected overload associated with state i, W_i, is

$$\Sigma_k (i + \tau_k - L)^+ p_k, \tag{1}$$

and the expected work overload per job at this station is

$$\Sigma_i \pi_i W_i. \tag{2}$$

By summing these values over all stations, we can obtain the expected overload per job for the entire system.

What is the value of this model? It provides information on the average quality of the sequence *if we do not make a conscious effort to smooth the flow of work.* (Of course, the quality of any particular randomly generated sequence might deviate substantially from this average, but we are interested in long-run averages, not a few specific instances.) At the other extreme, suppose that we were able to construct a perfect sequence with the

minimum possible work overload per job at each individual station. For each station this minimum value is easily computed as the difference between the average work content per job and the total labor time available per job (usually the cycle time, T), or zero if this difference is negative. This minimum value represents the amount of uncontrollable work overload. The difference between the expected work overload per job in a random sequence and the minimum possible overload per job is an upper bound on the expected benefit that can be obtained from a logical sequencing procedure. If we compute these differences for each station individually, we have an estimate of the expected benefit obtainable by including that station in a sequencing algorithm. (This assumes that the inclusion of one station helps to smooth the work only at that station.) We propose this upper bound as a metric by which we can rank the relative criticality of each station with regard to the sequencing algorithm, and test this basic premise later in the paper (see section 6). For simplicity, the difference between the expected work overload in a random sequence and the minimum possible work overload at station s is referred to as the criticality index for station s, or C_s. In a well-designed line, we would expect the minimum possible work overload to be zero, but in some instances the value is positive due to difficulties balancing the line, engineering changes that occur after the line is designed, or shifts in the mix of options.

It is worthwhile to point out that this model also provides the *distribution* of the work overload for an arbitrary job. The probability that the work overload is equal to t can be expressed as:

$$\sum_i \pi_i \, p_{j(i)}, \tag{3}$$

where j(i) is the value of j for which $i+\tau_j-L=t$. The expression in (3) is simply the sum of the probabilities of all combinations of states and processing times for which the work overload is exactly equal to t. These values are readily determined once the π_i values have been determined.

Since the expected work overload values from the Markovian model are easy to compute on a mainframe computer, the model can be used to evaluate the impact of various factors such as the mix of option combinations and their associated processing times, the length of the station, staffing levels, and task assignments. In the next section, we illustrate some of these points with an example in which there are only two option combinations at each station. Although this may appear to be rather specific, many such situations occur in practice. For example, a customer may only be able to choose between the presence or absence of a particular option, or between the standard or deluxe versions of a particular feature.

An Illustrative Example

Suppose that L = 15 time units, T = 6, $\tau_1 = 9$, and $\tau_2 = 4$. Let p_1 (= 1 - p_2) denote the fraction of jobs that require option combination 1. For this case, the expected work overload is given by

$$\frac{p_1^{10} - 8p_1^{9} + 22p_1^{8} - 14p_1^{7} - 15p_1^{6} + 11p_1^{5} + 6p_1^{4}}{8p_1^{5} - 24p_1^{4} + 24p_1^{3} - 8p_1^{2} + 1}$$

This expression was obtained by using the symbolic manipulation program REDUCE from the Rand Corporation.

Suppose that $p_1 = 0.36$. Steady state probabilities are given by

$\pi_0 = 0.24937$
$\pi_1 = 0.09077$
$\pi_2 = 0.04950$
$\pi_3 = 0.14182$
$\pi_4 = 0.07735$
$\pi_5 = 0.08133$
$\pi_6 = 0.06980$
$\pi_7 = 0.09924$
$\pi_8 = 0.02928$
$\pi_9 = 0.11155$

Using the above values, expected work overload per job for a random sequence is 0.1773. The minimum achievable work overload per job is given by $\{(1 - p_1)\tau_2 + p_1\tau_1 - T\}^+$, which is the same as the work overload per job as L --> ∞. (As the station becomes infinitely long, all available labor time can be used productively as long as there is work to be done.) For the above illustration, as $L \rightarrow \infty$, the expected work overload approaches zero, as would be true for any station which is balanced *on the average*. Thus, for this station, the criticality index is 0.1773, which is fairly small relative to the cycle time of 6 time units. Consequently, it may not be worthwhile to include this station in the sequencing algorithm. If this station is not included in the algorithm, the sequence that it receives is effectively random, unless other stations that are affected by some or all of the same options are included in the algorithm. Moreover, even if these other stations are included, it is not clear that doing so will help to smooth out the flow of work at the station in question because the option combinations affect the processing times at the various stations in different ways.

As mentioned earlier, one way to smooth the flow of work is to increase the length of the station. For the example above, we computed the expected work overload for various station lengths to illustrate this effect, as shown in Table 2. In this instance, increasing the station length from 15 to 25 units decreases the expected work overload by nearly 65 percent.

Insert Table 2 here

The Markov chain model provides exact values of the expected work overload for a random sequence. In many applications, however, exact values may not be needed to identify the five or ten most important stations. In the next section, we develop a regression model which is a prototype for others that can be developed for applications in which option choices are binary.

4. Regression Model

In this section we present a regression model which we constructed so that it could be used in a variety of applications. The model pertains to the case of two combinations of options at a station which, as we mentioned earlier, is common in practice. For simplicity, in this section, p denotes the fraction of jobs with the first combination of options. Using all combinations of parameters listed in Table 3 that give an average utilization of approximately 60 to 90% and with $\tau_2 < L$, we computed expected work overload values. We then developed a regression model which can serve as a "quick and dirty" estimator of expected work overload. We intentionally tried to keep the model as simple as possible without sacrificing predictive ability. After some trial and error, we found that it was easier to obtain a good fit for $\sqrt{EWO}$ rather than EWO, where EWO is the expected work overload. We first obtained a regression model using a second-order polynomial with all cross-product terms. We then added third-order terms which resulted in a substantially better fit. For the data in Table 3, we obtained the following regression model:

$$
\begin{aligned}
EWO = \quad & (0.02743 - 0.12394L - 0.13454\tau_1 + 0.21061\tau_2 - 3.96628p \\
& + 0.00223L^2 + 0.00500\tau_1^2 - 0.00780\tau_2^2 + 2.70767p^2 \\
& - 0.000016L^3 - 0.00011\tau_2^3 - 1.14251p^3 + 0.00406L\tau_1 \\
& + 0.00015L\tau_2 + 0.02942Lp + 0.00103\tau_1\tau_2 - 0.21278\tau_1 p \\
& + 0.26172\tau_2 p - 0.00007L\tau_1\tau_2 + 0.00029L\tau_1 p - 0.000084L\tau_2 p \\
& + 0.02866\tau_1\tau_2 p - 0.000033L\tau_1\tau_2 p)^2
\end{aligned}
$$

for which the coefficient of determination (R^2) is 0.9975. Additional details on the development and evaluation of the regression model appear in the Appendix.

Insert Table 3 here

To test the accuracy of the model for other combinations of parameters, we randomly generated 100 parameter sets within and near the ranges of the parameters in Table 3. For all sets of parameters with actual expected overload values of 0.2 or greater (i.e., greater than 2% of the cycle time), the errors were less than 10%, with the average and standard deviation of the errors for this category being 2.76% and 2.4%, respectively. We observed that the magnitude of the relative errors increased as the actual expected overload values declined. Thus, for cases with very small values of the actual expected overload (i.e., less than .01), the relative errors were quite large, although the absolute errors were small. However, even in these cases, the predicted values were sufficiently small to preclude selection of the stations for consideration in a sequencing algorithm.

For cases in which the actual expected overload was between 0.1 and 0.2 (1% to 2% of the cycle time), we observed that the largest relative errors occurred when one or more of the parameters was outside the range of the set of parameters on which the regression model is based. Thus, if it is deemed necessary to include even stations with such small expected work overload values in a sequencing procedure (i.e., if an *extremely* smooth schedule is necessary), it is important to ensure that an appropriate set of parameters be chosen for the regression model.

In conclusion, it appears that it is possible to develop regression models of expected work overload with very high predictive accuracy for situations in which the option choices are binary. Since job and processing time characteristics vary widely, it may be necessary to develop application-specific regression models. Although the initial development of such models may be time-consuming and normally will require the use of a mainframe computer, the resulting regression models can be implemented for interactive use on a personal computer, or even a hand-held calculator. This will facilitate the transfer of this

technology into practice. We now turn to an empirical investigation of the value of our proposed criticality index.

5. Selection of Critical Stations: An Empirical Study

We obtained data on option combinations and related processing times from an assembly facility of a major U.S. automobile manufacturer. This particular facility produces a product line with a high degree of option complexity, and as a consequence, faces very difficult sequencing problems. Management at the facility would like to smooth the flow of work related to over forty options which together affect several hundred work stations. Because of the obvious difficulty of identifying and analyzing hundreds of stations, we decided to ignore those with low processing time variability. After tedious analysis, we identified approximately 80 stations which appeared to be the most important to analyze. A majority of these stations were affected by multiple options.

For each of these 80 stations, we used the Markov chain model described in section 3 to compute the expected work overload for a random sequence. We also computed the amount of unavoidable work overload, which turned out to be zero in most cases, principally because of low average utilization. There were, however, a few stations that were overloaded, even on the average. (There is some benefit from including such stations in the sequencing procedure in order to eliminate any unnecessary work overload.) On the basis of this analysis, we retained only 25 stations with the largest expected work overloads because the remaining stations had expected work overload values that were not of practical significance (much less than 1% of the cycle time). We then ranked the stations in decreasing order of their criticality indices.

In order to test the usefulness of the proposed criticality index, we developed two different sequences for each of ten days, with approximately 1000 orders on each day. The sequences were determined using a heuristic procedure (see Yano and Rachamadugu 1991) which has the objective of minimizing total work overload at a set of stations. The first

sequence was determined using the five stations with the largest criticality indices, and the second sequence was determined using the five stations ranked 21 through 25 from our set of 25 important stations. For each sequence, we computed the total work overload at all 25 stations.

Scaled results (to preserve confidentiality) appear in Tables 4 and 5. Table 4 presents aggregate results by data set, while Table 5 presents average results by station. Note that the total expected work overload for a random sequence is 133,319. Thus, interestingly, using the five lowest-ranked stations (out of the top 25) gives consistently worse results than using a random sequencing procedure for the ten data sets. (See the first column in Table 4.) This highlights the need to choose stations carefully. Choosing the five lowest ranked stations rather than the top five stations reduces the aggregate work overload *for the 5 lowest ranked stations* by approximately 12%, but increases the work overload for the other 20 (more critical) stations by over 108%.

The third column in Table 4 shows the reduction in work overload obtained by choosing the five stations with the highest criticality indices rather than sequencing randomly. The average reduction is 38%, representing a substantial improvement. Thus, it appears that the proposed criticality index is a useful measure of the relative importance of a station with regard to sequencing. In this data set, several options affected multiple stations in a similar way (e.g., the presence of the option resulted in longer processing times). Consequently, selecting the five highest ranked stations actually helped to smooth the workload at several other stations (see Table 5).

Insert Tables 4 and 5 here

Of course, it may be desirable to include more or fewer stations, or perhaps a different subset of stations, in the sequencing algorithm. Also, interactions among stations, such as those described above, should be considered in selecting stations. The

criticality index enables us to reduce the number of stations that need to be considered, thereby allowing a more careful investigation of good alternatives.

6. Conclusions

In this paper, we developed a Markov chain model to determine the expected work overload (i.e., incomplete work) at a workstation facing a random sequence of jobs. This model forms the basis for the proposed criticality index to identify the most important workstations for sequencing purposes. The criticality index is the difference between the expected work overload in a random sequence and the amount of unavoidable work overload. Using actual industrial data, we show that significant reductions in work overload can be achieved by using this criticality index as the basis for identifying candidate stations to be considered in the sequencing algorithm.

The model can also be used to assess the impact of the station length, staffing levels, and task assignments on expected work overload when the station faces an effectively random sequence. These are areas which have been largely unexplored and for which practitioners have had to rely upon judgement and experience to make decisions. Practitioners find these problems extremely difficult, however, and consequently tend to build both labor and capital equipment slack into the system to reduce their perceived risk of incomplete jobs. Models developed in this paper provide formal and systematic means to evaluate these tradeoffs more accurately.

7. Acknowledgements

We would like to acknowledge the excellent contributions of Suk-Chul Rim in analyzing the data and running the computer programs for our experiments, and Gerald Bruce in the development of the regression model. We are grateful for the financial support of a major U.S. automobile manufacturer through a contract to the University of Michigan. We also appreciate the helpful comments of the two anonymous referees.

References

Bolat, A. (1988), "Generalized Mixed Model Assembly Line Sequencing Problem," unpublished Ph.D. Dissertation, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, Michigan.

Burns, L.D. and C. Daganzo (1987), "Assembly Line Job Sequencing Principles," ***International Journal of Production Research***, Vol. 25, No. 1, pp. 71-99.

Coffman, P. E. Jr., S. E. Hoffman, and S. A. Weiner (1985), "An O.R. View of Assembly Plant Modeling," paper presented at the TIMS/ORSA Conference in Boston, MA.

Koether, R. (1985), "Improving Productivity in Model-Mix Assembly," in **Toward the Factory of the Future: Proceedings of the Eighth International Conference on Production Research**, edited by H. J. Bullinger and H. J. Warnecke, Springer-Verlag, New York, pp. 761-766.

Miltenburg, J. (1989), "Level Schedules for Mixed-Model Assembly Lines in Just-In-Time Production Systems," ***Management Science***, Vol. 35, No. 2, pp. 192-207.

Neter, J., W. Wasserman and M.H. Kutner (1985), **Applied Linear Statistical Models: Regression, Analysis of Variance, and Experimental Design**, R.D. Irwin, Inc., Homewood, IL.

Okamura, K. and H. Yamashima (1979), "A Heuristic Algorithm for the Assembly Line Model Mix Sequencing Problem to Minimize the Risk of Stopping the Conveyor, ***International Journal of Production Research***, Vol. 17, No. 3, pp. 233-247.

Parrello, B. (1988), "Car Wars: The (Almost) Birth of an Expert System," ***AI Expert***, pp. 60-64.

Treece, J.B. (1989), "GM's Bumpy Ride on the Long Road Back," ***Business Week***, February 13, 1989, pp. 74-78.

Weiner, S. (1985), "Perspectives on Automotive Manufacturing," in **The Management of Productivity and Technology in Manufacturing**, edited by Paul R. Kleindorfer, Plenum Press, New York, pp. 57-71.

Yano, C.A. and A. Bolat (1989), "Survey, Development and Application of Algorithms for Sequencing Paced Assembly Lines," ***Journal of Manufacturing and Operations Management***, Vol. 2, No. 3, pp. 172-198.

Yano, C. A. and R. V. Rachamadugu (1991), "Sequencing to Minimize Work Overload in Assembly Lines with Product Options," ***Management Science***, Vol. 37, No. 5., pp. 572-586.

TABLE 1

Notation

L	=	sojourn time of a job in the station
K	=	number of option combinations
τ_k	=	processing time for a job with option combination k, k = 1,...,K
p_k	=	probability that option combination k occurs, k = 1,...,K
T	=	cycle time.
P	=	transition matrix
C_s	=	criticality index for station s
EWO(L)	=	expected work overload for a station of length L
π_i	=	steady state probability that the system is in state i
W_i	=	expected work overload in state i
x^+	=	max (0, x)

TABLE 2

Effect of Station Length on Expected Work Overload

Station Length	Expected Work Overload
15	0.1773
16	0.1566
17	0.1395
18	0.1245
19	0.1118
20	0.1001
21	0.0913
22	0.0828
23	0.0754
24	0.0688
25	0.0629
∞	0.0

TABLE 3

Parameter Values for Regression Model

Parameter	Values
T	10
L	16, 20, 23, 25, 30, 33, 35, 40, 43, 45, 50
τ_1	1, 2, 3, 4, 5, 6, 7, 8
τ_2	11, 13, 15, 17, 19, 21, 23, 25, 27
p	0.01, 0.04, 0.09, 0.16, 0.25, 0.36, 0.49, 0.64

Data Set #	Total Work Overload When Stations with 5 Smallest C_S Values are Used for Scheduling (column i)	Total Work Overload When Stations with 5 Largest C_S Values are Used for Scheduling (column ii)	improvement = $\frac{\text{column ii - EWO}}{\text{EWO}}$ (in percent)
1	146366	78016	41.48
2	147075	82619	38.03
3	152464	83055	37.70
4	166372	120067	9.94
5	150912	81183	39.11
6	141449	78960	40.77
7	143851	79153	40.63
8	146891	77015	42.23
9	146986	73444	44.91
10	147171	77232	42.07

EWO: Expected work overload for a random sequence =133319

Mean = 37.69

Total Work Overload Results by Data Set

Table 4

Station #	Average Work Overload Using 5 Least Critical Work Stations	Average Work Overload Using 5 Most Critical Work Stations	Expected Work Overload for a Random Sequence
1	10270	4346	8806
2	12313	2616	9662
3	6183	4840	6192
4	10515	5595	10157
5	8539	1128	5783
6	6962	4176	5696
7	2662	3208	4909
8	5628	3033	4889
9	4069	266	4393
10	4383	1815	4209
11	8023	3193	5404
12	4136	2936	4112
13	2833	3352	3771
14	8640	3933	6172
15	7837	1428	3558
16	401	112	3460
17	1579	2100	2517
18	5003	202	2372
19	13745	9391	11295
20	7333	5092	5861
21	79	1758	1837
22	2	571	544
23	6456	6152	6289
24	24	499	97
25	11339	11334	11334
	148954	83076	133319

Work Overload Results by Station

Table 5

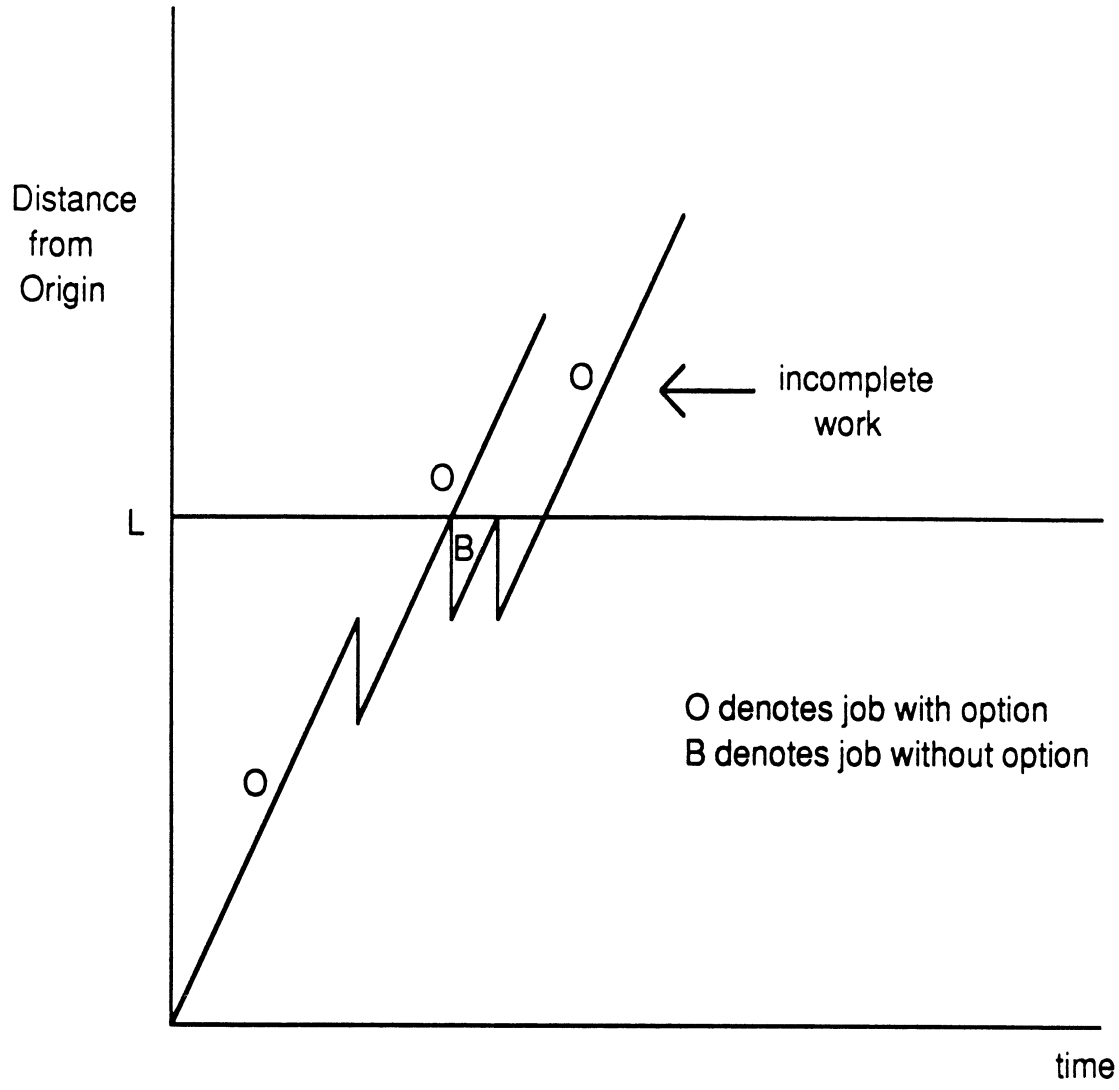

FIGURE 1

Operator Movement Diagram for Sequence (O,O,B,O) with $\tau_1 = 1$, $\tau_2 = 4$, and L = 5

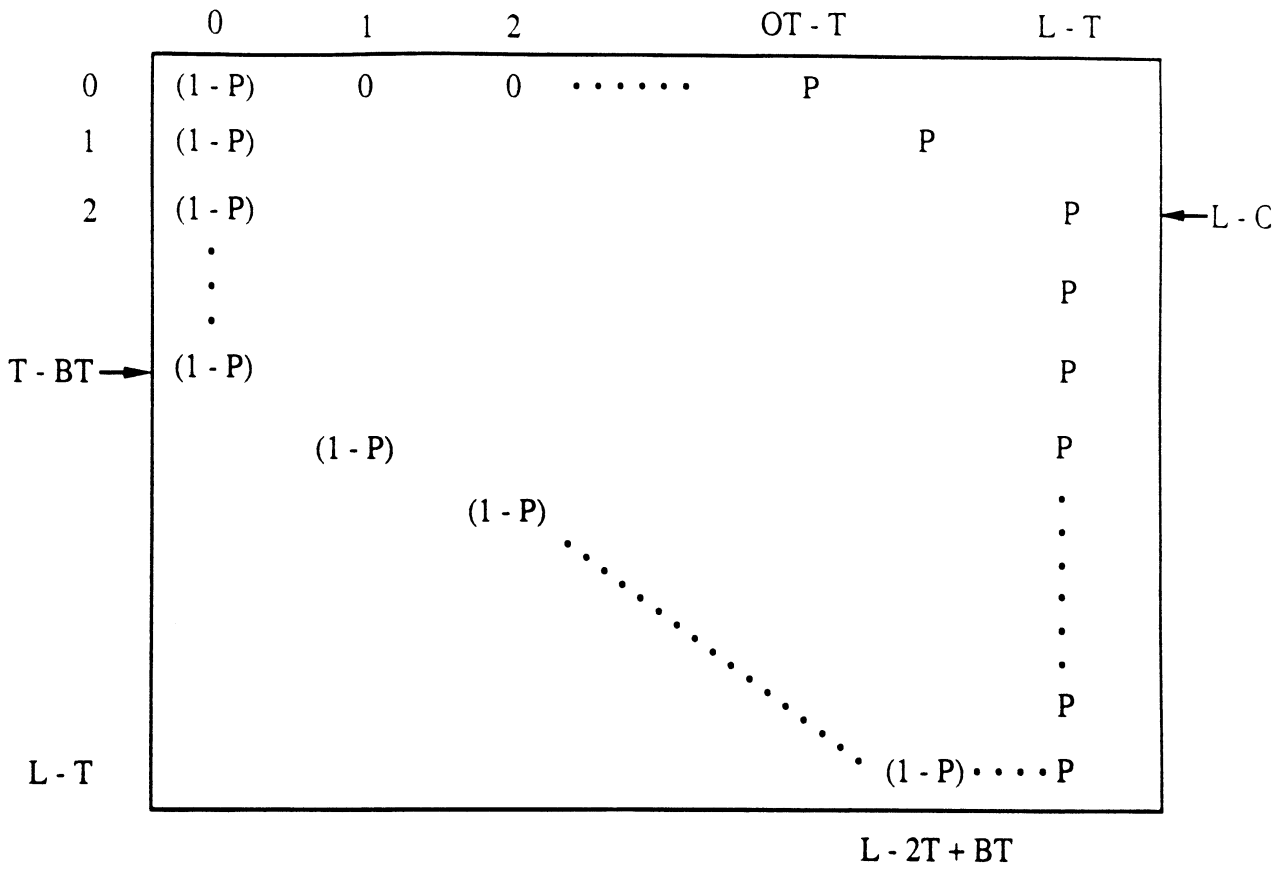

FIGURE 2

Markov Transition Matrix

APPENDIX

DEVELOPMENT AND EVALUATION OF THE REGRESSION MODEL

Our original goal in developing an expression for the steady state probabilities was to find a general closed-form expression which was a function of the parameters. Although we observed some interesting patterns, we were not able to develop a general expression. In particular, it was not clear exactly how the values of L, τ_1 and τ_2 affected the formulas. However, in a large number of examples, we observed that the probabilities could be expressed in exact form as a ratio of polynomial functions of p_1 (or p_2). This observation led us to investigate polynomial expressions as approximations.

A second-order polynomial plus all cross-product terms yielded a variety of systematic errors, most of which were reduced considerably by the introduction of third-order polynomial terms. Because this resulted in a pool of 23 independent variables, we used stepwise regression in an attempt to eliminate variables from the regression model. Using an F-to-enter of 4.000 and an F-to-remove of 3.996, we removed only one independent variable, τ_1^3. For the remaining 22 independent variables, the respective coefficients were found to be statistically significant at the $\alpha = .0001$ level (or smaller).

Several transformations of the data were made in order to improve computational and predictive accuracy and to reduce multicollinearity. First, we transformed L, τ_1, τ_2, and p by normalizing them with respect to their respective means and standard deviations. Thus, we let $L' = (L - \mu_L)/\sigma_L$ where μ_L is the mean of L in the data and σ_L is the standard deviation. τ_1, τ_2, and p were transformed in a similar fashion. The transformed variables are of the same order of magnitude, which reduces rounding errors, and this particular transformation also helps to reduce multicollinearity (Neter et al., p. 378). In addition to the above transformations, we also transformed the expected work overload by using its square root rather than the original value. This transformation reduces the range of these

values and resulted in smaller prediction errors. Partly as a consequence of these transformations, the resulting model showed no significant bias in the residuals. However, the errors for large work overload values were generally larger than those for small work overload values.

We investigated the predictive value of the regression model by examining the percentage error of the estimate relative to the actual value and by analyzing the ranks of the predicted and actual values. Our reason for performing the latter analysis is that one of the primary uses of the model is to identify those stations with the largest work overload so as to incorporate them into the scheduling algorithm. Thus, even moderate errors in the estimate of work overload would not be serious if it were still possible to identify the most critical stations.

The first error analysis indicated that over 70% of the fitted values were within 5% of the corresponding actual values and approximately 93% were within 10%. The larger percentage errors tended to be associated with very small expected work overload values, for which predictive accuracy is not critical, since it is unlikely that there would be any need for the corresponding station to be included in the scheduling procedure.

A comparison of the ranks of the fitted and actual values showed a strong correlation between the ranks, with an R^2 of .996. The absolute differences between the respective ranks is quite small for large expected work overload values, for which the rankings are most critical.

The results of these analyses indicate that the regression model provides an excellent predictor of expected work overload in an absolute sense, and a good predictor of the relative rank of various stations with respect to expected work overload.