ELSEVIER

Int. J. Production Economics 51 (1997) 177–190

international journal of
production
economics

A heuristic for solving mixed-model line balancing problems with stochastic task durations and parallel stations

Patrick R. McMullen[a,*], Gregory V. Frazier[b]

[a] *College of Business, University of Maine, Orono, ME 04469-5723, USA*
[b] *Department of Information Systems and Management Sciences, College of Business Administration, The University of Texas at Arlington, Arlington, TX 76019-0377, USA*

Received 23 August 1995; accepted 17 January 1997

Abstract

This paper describes an approach for solving a mixed-model assembly line-balancing problem with stochastic task times when paralleling of tasks within work centers is permitted. The research modifies previous work and incorporates new and existing task selection rules for assigning tasks to work centers. The heuristic is applied to six different line-balancing problems for each presented rule. The resulting layouts are simulated and performance results are analyzed. Through simulation, the research demonstrates how the presented heuristic can be used to solve complex line-balancing problems using different strategies, and these strategies can be evaluated across many performance dimensions.

Keywords: Line balancing; Simulation; Heuristics; Mixed models; Paralleling

1. Introduction

Research on assembly line balancing was quite popular in the 1960s and 1970s. Researchers during these years developed many approaches for finding optimal and heuristic solutions to the simplified line balancing problem. Today, many researchers seem to consider line-balancing research as mostly unimportant or irrelevant. Presumably, this view is because the earlier line-balancing approaches made many simplifying assumptions, and in modern manufacturing environments the same assumptions may not be realistic.

The research presented here addresses the issue of line balancing in a setting that is becoming more common in modern manufacturing: a multiple product, high-volume assembly or production line, with mixed-model sequencing, stochastic task durations, and the occasional use of duplicate equipment. One place this setting can occur is in companies that have adopted JIT techniques and strive for short cycle times. Little research has been conducted to address this version of the line-balancing problem, even though this type of setting is increasingly encountered as companies try to increase their flexibility through mixed-model production, as well as increase their output through higher volume production. New approaches to help solve this problem could be very helpful to these companies.

* Corresponding author. Tel.: 207-581-1994; e-mail: patmc@maine.maine.edu.

PII S0925-5273(97)00048-0

Assembly line balancing is the practice of distributing work into work centers in order to achieve some goal. Type I line-balancing problems attempt to assign workers in such a way that the total number of workers required is minimized, given a specified cycle time. In this context, cycle time is the amount of time that elapses between the completion of two consecutive units of production. Type II problems attempt to assign a specified number of workers to cells in such a way that cycle time is minimized. Type I line-balancing problems are frequently encountered when a high production rate is required. Type II line-balancing problems are often encountered when there is a limitation of available working space. This research addresses the Type I assembly line-balancing problem with high production rates.

Most line-balancing research assumes that task durations are deterministic. Since there is always some variation when humans perform tasks, and since task time variation has a greater impact with shorter cycle times, this research addresses stochastic task times. Most line-balancing research also assumes that a single product is being produced. With the popularity of just-in-time (JIT) production control systems, mixed-model situations have become more common. Mixed-model scheduling better manages the flow of products through the system than does single-product batch scheduling. With the flow better regulated, the in-process inventory levels and flow times of mixed-model systems will be lower than with single-product batch systems. In general, the benefits of utilizing a JIT system should be greater when using mixed-model sequencing as compared to the batch sequencing of a single-product system (Ding and Cheng, 1993). Mixed-model sequencing is used in this research.

With high production rates, a situation can occur where the longest task time exceeds the specified cycle time. A common remedy is to create parallel work stations, where multiple workers perform an identical set of tasks. This also allows more flexibility in assigning tasks. Consider a single employee in a work center, as shown in Fig. 1. The employee performs four tasks (3, 5, 6, and 9). An assumption is made that one piece of equipment is required for each task. Fig. 2 shows an example where

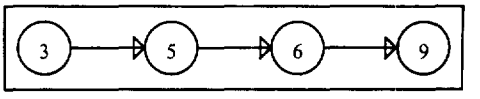

Fig. 1. Example of Non-Paralleling.

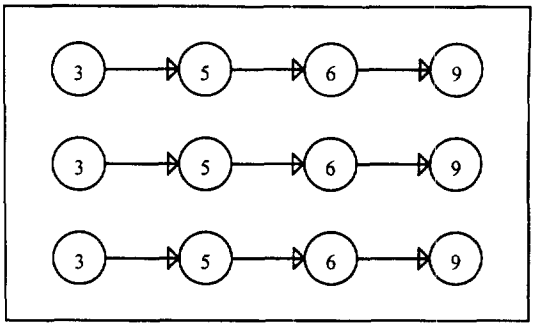

Fig. 2. Example of Paralleling.

paralleling is used. In this situation, three workers each perform tasks 3, 5, 6, and 9. Twelve pieces of equipment are then necessary (3 workers × 4 tasks/worker). While paralleling results in additional equipment cost, it allows for shorter cycle times. The heuristic presented in this paper permits task paralleling to occur.

To evaluate the performance of different task selection rules, the following output measures are considered: average work-in-process (WIP) inventory level, average flow time – the average amount of time a unit spends in the system, system throughput – the number of units through the system during a given time period, average unit labor cost, average system utilization, and percentage of units completed at each work center within the pre-specified cycle time – analogous to on-time completion. In addition to these six output measures, the ratio of pre-specified cycle time to actual cycle time attained by finished units was also considered – a measure of the layout's ability to achieve the desired cycle time.

Although attaining high levels of the output performance measures is desirable, it is also important to have tolerable levels of inputs to implement these layouts. These inputs are the number of workers and the amount of equipment necessary to run the assembly line.

This research was undertaken to provide production management with a methodology to solve the assembly line-balancing problem with stochastic task durations, mixed-model sequencing, and paralleling of tasks. This methodology offers several alternative task-selection rules.

First, a mixed-model problem is reduced into a "composite" single-model problem. The methodology then examines the potential effects of placing the tasks into work centers according to the task-selection rule. A previous heuristic is modified and used with seven alternative task-selection rules. Of these rules, three currently exist, and the other four were developed for this research (McMullen, 1995). A simulation experiment is then used to compare the seven rules with three other "seat of the pants" rules in terms of performance measures. The following sections discuss relevant literature on this problem, present the line-balancing heuristic, describe the experimental design, and offer conclusions.

2. Literature review

Salveson (1995) was the first to empirically address the assembly line-balancing problem. He described the assembly line as a set of precedence relationships among tasks that must be adhered to. This means that a task cannot be assigned into a cell unless its immediate predecessors have already been assigned into cells. Salveson also recognized the line-balancing problem as a combinatorial one, realizing that a problem will have many feasible solutions. Bowman (1960) soon followed with an integer linear programming solution to the line-balancing problem. From the work of both Salveson and Bowman, however, it became quite clear that developing optimal solutions to this problem type were usually impractical due to the excessive number of decision variables and constraints which were required for even small problems.

Over time, efforts were made to solve the line-balancing problem by more computationally efficient means. Klein (1963) solved the line-balancing problem by using the logic behind the shortest route problem. With this approach, the problem was solved by finding the shortest "task distance" from the source node to the sink node of the precedence matrix. Patterson and Albracht (1975) solved the problem by using an integer programming search technique. This solution was more computationally efficient than some of the earlier ones because some unnecessary variables were eliminated from the formulation. Pinto (1975) and Pinto et al. (1975) developed efficient approaches to the line-balancing problem by utilizing branch and bound techniques. A significant contribution of their research was that they provided a solution to the line-balancing problem by permitting paralleling of tasks to occur.

As line balancing became a more popular research topic, several heuristics were contributed which provided solutions that were not necessarily optimal, but were usually adequate. These heuristic solutions were usually very computationally efficient. Helgeson and Birnie (1961) developed a heuristic known as the Ranked Positional Weight Technique (RPWT) which can provide near-optimal solutions. The heuristic works by placing tasks into cells according to which tasks have the most total task time succeeding the task of interest. Moodie and Young (1965) developed a heuristic which places tasks into cells according to which tasks have the longest task time. This heuristic, as well as the others, has the provision that a task cannot be placed into a cell unless all of its immediate predecessors have been assigned into cells. Arcus (1966) developed a heuristic known as COMSOAL, computer method of sequencing operations for assembly lines. COMSOAL works by first exploiting the fact that line-balancing problems are combinatorial ones. COMSOAL generates many different solutions to the same problem by using one of several different task selection rules which are Monte-Carlo simulation based. COMSOAL then selects the solution which has the lowest level of idle time.

All of the line-balancing techniques previously described treat task durations as deterministic. These solution techniques provide a strong foundation for line-balancing research, but they do not address the reality that most task times are stochastic. Because of this, several stochastic line-balancing techniques have been developed, most of which

use similar logic. As units are processed on an assembly line, units requiring more time than allowed (i.e., more than the cycle time) are either taken off-line and completed off-line at a later time, or the units-in-process have as many on-line tasks completed as possible, and the remaining required tasks are completed off-line at a later time. The processing that is done off-line results in an additional cost referred to as the incompletion cost. The usual objective of stochastic line-balancing techniques is to minimize the sum of regular processing cost and incompletion cost.

Kottas and Lau (1981) developed a technique to minimize this total cost by developing an initial balance using a deterministic approach, and then "swapping" tasks from cell to cell until total cost is minimized. Carter and Silverman (1984) and Silverman and Carter (1986) also developed a cost minimizing approach similar to that of Kottas and Lau, but the Carter and Silverman approach ensures that the probability of completing all tasks on time is above a certain specified threshold level. Vrat and Virani (1976) presented a modification of the Kottas and Lau methodology to solve line-balancing problems with the complexities of stochastic task durations, task durations greater than cycle times, and mixed models. The Vrat and Virani methodology assigns tasks into cells in an effort to minimize the incompletion cost of in-process units.

Driscoll and Abdel-Shafi (1985) presented an integrated line-balancing and simulation based evaluation technique to address the line-balancing problem having complications of stochastic task durations, mixed-model processing, task times greater than cycle time, and zoning requirements. The technique first performs a line balance using the RPWT, then performs simulations to assess the performance of the layout. This technique is intended to provide line-balancing solutions that are applicable to real-world situations.

Other techniques have also been developed to exploit the combinatorial nature of the line-balancing problem. Peterson (1993) developed a tabu search procedure to solve the simple line-balancing problem. With this procedure, an initial solution is adjusted according to tabu restrictions in attempt to improve the solution to a near-optimal condition. Suresh and Sahu (1994) developed a simulated annealing heuristic solution to the stochastic line-balancing problem. This technique uses the objective function developed by Moodie and Young – minimization of "lumpiness" across all cells. Similar to the tabu-search procedure, the simulated annealing procedure also starts with an initial solution and attempts to reach some optimal condition. Both techniques are dedicated to avoiding being "trapped" at local optima. Leu et al. (1994) also developed a technique to exploit the combinatorial nature of the line-balancing problem by way of genetic algorithms. These genetic algorithm (GA) solutions work by attaining an initial solution and improving upon them by performing mutations on the prior solutions, until a desirable solution is found.

Nkasu and Leung (1995) developed a COMSOAL-based solution to solve the stochastic line-balancing problem. This procedure works by first simulating some of the stochastic attributes of the line-balancing problem and then using a modified version of COMSOAL to generate several feasible problem solutions, and then the procedure selects the "best" of all of the possible solutions. This procedure is a sampling approach to solving the line-balancing problem.

3. Methodology

3.1. Modified incremental utilization heuristic

The line-balancing technique presented in this research is a modification of Gaither's (1996) incremental utilization heuristic. The original heuristic only addressed deterministic task durations, where the modified heuristic addresses stochastic task durations. The modified heuristic also uses seven alternative different task-selection rules for assignment of tasks into cells (work centers), whereas Gaither's heuristic simply selects the task with the lowest task number of all unassigned tasks which are candidates for assignment. The modified heuristic uses composite task durations to address the mixed-model production requirement, where the original heuristic can only solve the single-model problem. The main attribute of the modified

heuristic that exists in Gaither's heuristic is the fact that tasks are placed into a work-center only if the utilization of the cell increases. Otherwise, the cell is closed and a new cell is opened and is ready for assignment of tasks.

Both the Gaither heuristic and the modified heuristic permit paralleling of tasks – allowing multiple workers to exist within cells. The motivation for paralleling is to increase utilization in cells as much as possible. There are occasions when having multiple workers in cells will result in higher utilizations as compared to having a single worker in a cell. Paralleling also permits the existence of having cycle times which are shorter than the actual task times. When the actual task duration is in excess of the cycle time, paralleling reduces the average value of the task duration proportional to the number of workers in the cell.

Before the methodology is introduced, the following general assumptions are made regarding the assembly line-balancing problem addressed here:

- A task cannot be assigned to a cell until all of its immediate predecessors have been assigned to cells.
- The number of parallel work stations is unrestricted.
- All workers on the assembly line possess the same level of skill.
- All tasks are independent of each other.
- Changeover times between products are negligible.
- Any needed equipment is readily available.
- All workers and all units of necessary equipment cost the same.

3.2. Description of heuristic

For convenience, a listing of all variables involved in the heuristic follows:

C cycle time,
cv the specified coefficient of variation,
D_T the total demand for all products,
d_h the demand for product h,
hp desired hourly output,
ILR integer labor requirement for cell j,
LR_e expected labor requirement (based upon expected task durations) for cell j,
m_e total cell processing minutes based upon expected task durations for cell j,
p_j probability of completing all tasks on time within cell j,
p_s probability of completing all tasks on time in all cells,
pm productive minutes per hour,
$(t_i)_h$ the mean task duration of task i for product h,
t_i the weighted mean (composite) task duration for task i,
U_j utilization of cell j,
U_s utilization of entire system,
W_a actual number of workers required,
W_m theoretical minimum number of workers required,
w_h the weight of product h,
Y upper bound of task completion time,
$(\hat{\sigma}_i)_h$ the estimated standard deviation of task i duration for product h,
$\hat{\sigma}_i$ the composite estimated standard deviation of task i duration,
s_j estimated standard deviation of labor requirement in cell j.

As another matter of convenience, Fig. 3 shows a flowchart which graphically describes the modified heuristic. Details of the heuristic follow Fig. 3.

3.2.1. Step 1: Attain composite task durations

The first step of this heuristic is to reduce the stochastic task durations of the mixed-model production demand into composite stochastic task durations. The variable d_h is the demand for product h, and the total demand, D_T, for all of the q different products demanded is given by

$$D_T = \sum_{h=1}^{q} d_h. \tag{1}$$

The weight of product h, w_h, is calculated as follows:

$$w_h = \frac{d_h}{D_T}. \tag{2}$$

With weights determined, proportions of product-mix for each product are known. This information is then used to compute weighted average, or composite, task durations for each individual task.

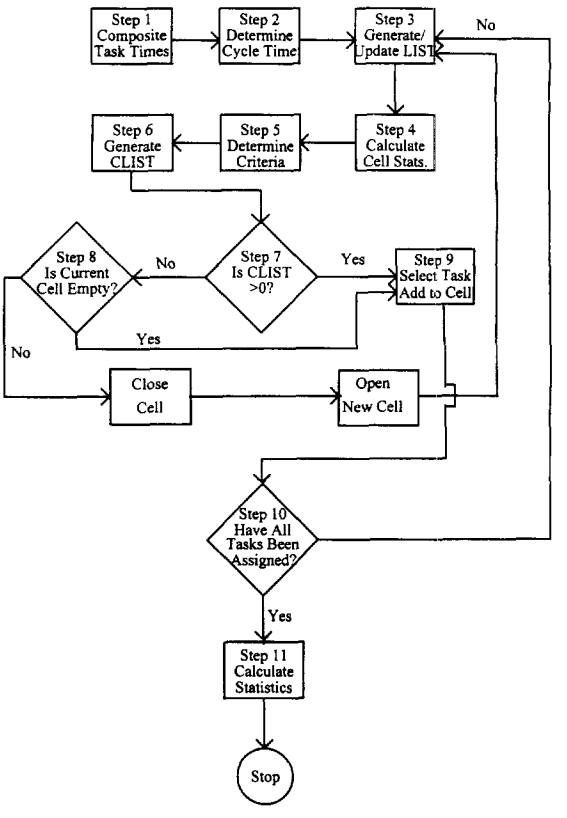

Fig. 3. Flowchart of Modified Incremental Utilization Heuristic.

The variable $(t_i)_h$ represents the mean task duration of task i for product h. The variable t_i represents the weighted mean (composite) task duration for task i, and is estimated as follows:

$$t_i = \sum_{h=1}^{q} w_h (t_i)_h. \tag{3}$$

The estimates of composite task durations have now been attained. These estimates of composite task durations were used in conjunction with a specified coefficient of variation, denoted by cv, to obtain standard deviations of composite task durations. The estimated standard deviation then of task i for product h is symbolically represented by $(\hat{\sigma}_i)_h$, and is calculated by the following formula:

$$(\hat{\sigma}_i)_h = (t_i)_h \cdot \mathrm{cv}. \tag{4}$$

The estimate of the standard deviation for composite task i, denoted by $\hat{\sigma}_i$, is calculated as follows:

$$\hat{\sigma}_i = \sqrt{\left(\frac{1}{D_T - 1}\right)\left[\sum_{h=1}^{q} d_h((t_i)_h - t_i) + \sum_{h=1}^{q} (d_h - 1)(\hat{\sigma}_i)_h^2\right]}. \tag{5}$$

When the total annual demand is large, however, the following simpler formula can be used to estimate the composite standard deviation (which was used for this research):

$$\hat{\sigma}_i = \sqrt{\sum_{h=1}^{q} [(\hat{\sigma}_i)_h^2 + ((t_i)_h - t_i)^2] w_h}. \tag{6}$$

At this point, the mixed model has been reduced to a single-product model with composite task time t_i and composite standard deviation $\hat{\sigma}_i$.

3.2.2. Step 2: Compute cycle time and minimum crew

Cycle time, C, is determined by dividing productive minutes per hour, pm, by desired hourly output, hp:

$$C = \frac{\mathrm{pm}}{\mathrm{hp}}. \tag{7}$$

Cycle time, C, is measured in minutes per unit. The theoretical minimum number of workers required, W_m is determined as follows:

$$W_m = \frac{1}{C} \sum_{i=1}^{n} t_i. \tag{8}$$

At this point, the actual line balancing commences.

3.2.3. Step 3: Generate/update LIST

A list of all tasks that are ready for immediate assignment into cells is constructed. The tasks placed on this list are tasks which have all of their immediate predecessors already assigned into cells. This listing of eligible tasks is referred to as LIST. The number of expected minutes of cell time, m_e, is initialized to zero when opening a new cell. The cell utilization, U_j, is also initialized to zero.

3.2.4. Step 4: Calculate cell statistics

For each task in LIST, an updated version of m_e is calculated, as if task i were to be added to the

current cell. The updating of m_e is done by using the following formula:

$$m_e = m_e + t_i. \tag{9}$$

The expected labor requirement, LR_e, is calculated as if task i is added to the current cell. The following formula provides the basis for these calculations:

$$LR_e = \frac{m_e}{C}. \tag{10}$$

The integer value for the expected labor requirement, ILR, is rounded upward if the calculation for LR_e yields a non-integer value. The value of ILR represents the number of workers that could be placed into the cell, and must reflect "whole" workers. The integer labor requirement, ILR, represents this integer value of LR_e:

$$ILR = \begin{cases} \text{int}(LR_e) + 1 & \text{if } LR_e \text{ is non-integer,} \quad (11) \\ LR_e & \text{if } LR_e \text{ is integer.} \quad (12) \end{cases}$$

The cell utilization, U_j, is obtained by dividing the expected labor requirement by the integer labor requirement,

$$U_j = \frac{LR_e}{ILR}. \tag{13}$$

Cell utilization is a measure of how efficiently employees are used and will never exceed 100%.

When tasks are added to the current cell, there is a probability that the labor required will exceed the labor available due to the stochastic nature of the task durations. When this happens, there will not be enough time to complete the tasks in the cell within the desired time frame. In this context, the cycle time, C, is the driving force behind this desired time frame. When an excessive amount of time is required to perform a certain task, inventory build-up will occur, and the desired level of production will not be met. To address this concern, the probability of completing all tasks within a cell on time is approximated – again tasks are assumed independent. This approximation is derived by first computing a combined standard deviation estimate of the total expected labor requirement within cell j:

$$s_j = \frac{1}{C}\sqrt{\sum_{i=1,\, i\in j}^{n} \hat{\sigma}_i^2}. \tag{14}$$

In this equation, the composite standard deviation is calculated by taking the square root of the sum of the variances divided by the cycle time of all tasks in cell j. The rationale for dividing by C is to convert time into workers, as done in Eq. (8). It is desired to approximate the probability that the expected labor requirement will exceed the labor available. Fig. 4 displays the probability distribution of actual labor requirement. The mean value of this distribution is the expected labor requirement, LR_e. Also shown is an example level of labor

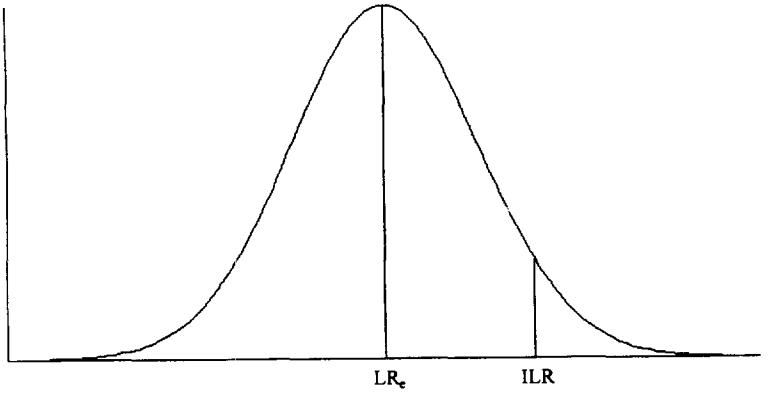

Fig. 4. Probability distribution of labor requirement with respect to labor available.

available, ILR, which would be determined by the heuristic.

The probability of the actual labor requirement being less than the labor available implies the probability of completing all tasks in the cell on time (within cycle time). This probability is graphically represented by the area to the left of the vertical line at ILR in Fig. 4. To calculate this probability, the data must be normalized:

$$Y = \frac{\mathrm{ILR} - \mathrm{LR_e}}{s_j}. \tag{15}$$

The probability of completing all tasks in the cell on time will be represented by p_j. The value of p_j is calculated by integrating the normalized probability distribution function as follows:

$$p_j = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{Y} e^{-Z^2/2}\, \mathrm{d}Z. \tag{16}$$

Numerical integration with the trapezoid rule was used with 250 intervals to determine the value of p_j.

3.2.5. Step 5: Determine task acceptance criteria

A task will not be added to the current cell unless the cell utilization increases as a result of adding the task (along with any necessary workers) to the cell. Additionally, a task will not be added to the current cell unless the probability of all tasks within the cell on time (p_j) is above a specified lower threshold level.

3.2.6. Step 6: Generate CLIST

From LIST, all tasks need to be examined to determine if they meet the necessary criteria to be added to the current cell – utilization of the cell as the result of adding the task to the current cell must increase, and the p_j of adding the task to the current cell must be above the specified lower threshold level. All tasks from LIST that meet these criteria will be placed on CLIST.

3.2.7. Step 7: Is CLIST empty?

If there are tasks on CLIST, proceed to Step 9 so that a task can be selected to be added to the current cell. Otherwise proceed to Step 8 to determine the proper action.

3.2.8. Step 8: Is current cell empty?

If the current cell is empty, a task must be selected and added to the cell, so proceed to Step 9. If the current cell is not empty, the cell must be closed out, a new one opened, and LIST must be updated. When a cell is closed out, the utilization of the cell is the utilization as the result of the last task added. Similarly, the p_j of the cell closed out is the p_j as the result of the last task that was added. When a new cell is opened, all relevant statistics are reinitialized. After closing out the cell and opening a new one, proceed to Step 3.

3.2.9. Step 9: Select task

At this point, a task needs to be selected to add to the current cell. One of seven alternative task selection rules will be used to choose the task. For each rule, the list of tasks eligible for selection will either be CLIST or LIST, depending upon the number of tasks appearing on CLIST.

3.2.9.1. Task selection rule 1. This task selection rule examines the resulting cell utilization, U_j, for each of the tasks appearing on the list. Rule 1 selects the task that would result in the highest cell utilization, based on the expected task time. The reasoning behind this rule is so that the user can attempt to maximize labor utilization.

3.2.9.2. Task selection rule 2. The second task selection rule is an application of COMSOAL (Arcus, 1966). For this selection rule, a pseudo-random number is drawn, and is used to determine which task on the list to assign to the current cell. The probability of selection for each task appearing on the appropriate list is directly proportional to the task's expected duration. This rule is a Monte-Carlo simulation-based selection procedure.

3.2.9.3. Task selection rule 3. This selection rule uses the logic introduced by Moodie and Young (1965). The task having the longest simulated task duration is selected from the list to enter the current cell. Since longer tasks are more difficult to fit into partially loaded cells, this rule attempts to fit these tasks into cells as early as possible.

3.2.9.4. Task selection rule 4. With this rule, the task having the shortest simulated task time is selected to enter the current cell. This rule is related to Task Selection Rule 3, but it attempts to place as many tasks into cells as early as possible.

3.2.9.5. Task selection rule 5. This rule is similar to Rule 1, but Rule 5 selects the task on the list which provides the lowest cell utilization. The reasoning for this rule is that if tasks are placed into cells based on lower utilization, more tasks can be placed into cells earlier.

3.2.9.6. Task selection rule 6. The task on the list that would result in the highest value of p_j is selected to enter the current cell with this rule. Rule 6 is an attempt to increase the probability of tasks being completed within the cycle time (Carter and Silverman, 1984; Silverman and Carter, 1986)

3.2.9.7. Task selection rule 7. This rule selects the task on the list with the highest product of cell utilization (U_j) and the resulting value (p_j), hence, $\max(U_j \cdot p_j)$. The reasoning for this rule is to attempt to select a task which will provide a high cell utilization as well as supply a high probability of completing the tasks in the cell within the cycle time. This selection rule is an attempt to "have the best of both worlds" in terms of cell utilization and probability of on-time completion.

3.2.10. Step 10: Determine if all tasks have been assigned

If all tasks have been assigned, proceed to Step 11 to calculate all statistics pertaining to the line balance. If all tasks have not been assigned, proceed to Step 3 and update LIST.

3.2.11. Step 11: Calculate line balance statistics

At this point, the line balance is complete, and statistics on the performance of the line balance are calculated. The first measurement to calculate is the utilization of the overall line, U_s. This is the ratio of the minimum number of workers used, W_m, as determined in Eq. (8), to the actual number of workers used. The actual number of workers, W_a, is the sum of workers across all m cells. Hence,

$$W_a = \sum_{j=1}^{m} \mathrm{ILR}_j. \tag{17}$$

The overall system utilization is

$$U_s = \frac{W_m}{W_a}. \tag{18}$$

The probability of on-time completion for the overall system, p_s, is the product of all p_j's (Carter and Silverman, 1984; Silverman and Carter, 1986). Recall that the p_j of each cell is the p_j as a result of placing the final task in each cell. The overall system probability of on-time completion is computed as follows:

$$p_s = \prod_{j=1}^{m} p_j. \tag{19}$$

4. Design of experiment

An experiment was designed to evaluate the performance of the methodology. Comparisons were based on output performance measures of: average WIP inventory level, average system flow time, units of throughput, average unit labor cost, percentage of units being completed within cycle time, average system utilization, and percent of desired cycle time attained. These output performance measures were obtained via computer simulation of a production line.

Along with the seven presented rules, three additional, more informal rules were used to generate assembly line layouts. These ten rules provided ten assembly line layouts. There were also six different problems used for this experiment: 21, 25, 29, 40, 45, and 74 task problems. Each rule with each problem resulted in a total of 60 layouts. Each of these layouts was simulated with SLAMSYSTEM v.4.6 and FORTRAN v.5.1 user-written inserts so that output performance measures could be attained. Each simulated production run is replicated 25 times for a total of 1500 records in the database (6 problems × 10 rules × 25 simulation runs). Each simulation was run until steady-state conditions were attained, then statistical arrays were cleared

Table 1
Description of six different line-balancing problems

Tasks	Different products	Product-mix weights
21	1	$w_1 = 1$
25	1	$w_1 = 1$
29	2	$w_1 = 2, w_2 = 1$
40	3	$w_1 = 3, w_2 = 2, w_3 = 1$
45	2	$w_1 = 2, w_2 = 1$
74	4	$w_1 = 4, w_2 = 2, w_3 = 1, w_4 = 1$

and the run was continued until stopping criteria were met, at which point, values of output performance measures were collected. Table 1 provides information about the six different problems specific to their mixed-model nature.

For each of the six problems, the cycle time was specified to be 10 min per unit. The durations for the tasks were created via a random number generator where, for each individual task, there was a 75% probability the task duration would be uniformly distributed between 2 and 10 min, and a 25% probability the task duration would be uniformly distributed between 10 and 15 min. The precedence diagram for the 21 task problem is from Tonge (19xx), the precedence diagram for the 29 task problem is from Buxey (1974), and the precedence diagram for the 45 task problem is from Thomopoulos, (1967). The 25, 40, and 74 task problems were arbitrarily generated for this research.

4.1. Informal rules

As previously mentioned, there are three informal rules which were compared to the seven other rules. These three rules are less sophisticated than the others, but provide feasible solutions nonetheless. The first of these informal rules (referred to as Rule 8) is a lexicographic task selection rule – Gaither's (1996) incremental utilization heuristic without modifications. This rule simply selects the first task which appears on the list of tasks which have met criteria for being assigned into a work center. It is appropriate to note that the incremental utilization policy for the work centers still applies – a task will not be added to a work center unless the utilization of the work center increases as a result of adding the task. The second informal rule (referred to as Rule 9) ignores the incremental utilization policy of the work centers and places all tasks into a single "mega" work center. This rule minimizes the number of workers, but simultaneously maximizes the amount of paralleling necessary. The third informal rule (referred to as Rule 10) places each task in its own unique work center. This rule minimizes the amount of paralleling necessary, but simultaneously maximizes the number of workers.

4.2. Research questions

In order to evaluate whether the first seven rules provided desirable layouts, the following research questions were addressed:

1. Do the task selection rules have an overall multivariate effect on the output performance measures?
2. If so, which output performance measures are most affected by the task selection rules? Additionally, which task selection rules generally provide the best layouts?
3. Does the number of work centers affect the output performance measures? If so, do the task selection rules influence the number of work centers?

5. Results

Before addressing the research questions, it should be noted that some of the output performance measures were highly correlated. Average WIP inventory level and average system flow time were correlated ($\rho = 0.9999$) as were average WIP level and average unit labor cost ($\rho = 0.9042$). As a result, average system flow time and average unit labor cost were omitted from the analysis and were explained by average WIP level.

The first research question was addressed by performing a MANOVA. As suggested by this analysis, the task selection rules did have an overall multivariate effect on the output performance

Table 2
Discriminant coefficients and univariate statistics for output measures

Output measure	Func. 1	Func. 2	Func. 3	Func. 4	Univariate F	$p <$
WIP	− 0.8840	3.3646	0.8966	3.1436	5.62	0.0001
Throughput	0.6072	− 1.5933	− 0.0170	− 1.8302	1.80	0.0640
PCCT[a]	− 0.4557	1.1941	− 0.7303	− 0.7750	172.03	0.0001
Utilization	− 6.9533	0.7723	0.3590	− 0.6713	178.74	0.0001
PCTA[b]	4.5004	1.9889	2.0270	2.6080	22.34	0.0001

[a] Percent of units completed within cycle time.
[b] Percentage of desired cycle time attained.

Table 3
Means and standard deviations of performance measures by rule

Rule	WIP	Throughput	PCCT[a]	Utilization	PCTA[b]
1	104.6 (77.5)	310.4 (120.5)	0.732 (.032)	0.749 (.076)	0.835 (.077)
2	90.4 (53.8)	325.4 (148.1)	0.665 (.044)	0.802 (.054)	0.858 (.061)
3	116.1 (98.4)	297.6 (98.3)	0.756 (.036)	0.741 (.096)	0.814 (.104)
4	84.5 (52.1)	330.4 (149.3)	0.708 (.076)	0.817 (.051)	0.872 (.055)
5	95.9 (66.5)	319.5 (153.7)	0.740 (.063)	0.784 (.138)	0.840 (.143)
6	76.6 (48.5)	338.2 (153.5)	0.746 (.049)	0.822 (.055)	0.892 (.054)
7	73.0 (43.6)	342.0 (157.5)	0.761 (.053)	0.826 (.052)	0.900 (.046)
8	86.3 (60.7)	328.3 (139.2)	0.747 (.045)	0.809 (.058)	0.873 (.062)
9	86.2 (137.9)	320.4 (46.8)	0.797 (.114)	0.906 (.155)	0.921 (.176)
10	68.7 (53.8)	347.0 (151.3)	0.909 (.034)	0.547 (.032)	0.920 (.067)

[a] Percent of units completed within cycle time.
[b] Percentage of desired cycle time attained.

measures – Wilk's $\lambda = 0.0168$, with an associated F-statistic of 220.6166 and a $p < 0.0001$.

Since it appeared that the different task-selection rules did have an effect on the performance of the layouts, it was appropriate to determine which output measures were most affected by the rules. Discriminant analysis and univariate ANOVA were used to address the second research question. Four discriminant functions were found to be statistically significant at the $\alpha = 0.05$ level. Table 2 lists the discriminant function coefficients, and the Univariate F's and their associated p-values for each of the output performance measures.

Table 2 shows that the percent of units completed on time and system utilization are most affected by the task-selection rules. Average WIP level and percentage of desired cycle time attained are also affected by the task selection rules, whereas the system throughput is not.

Table 3 shows the means and standard deviations (in parentheses) for the performance measures by task-selection rule. From inspection of Table 3, it is clear that some of the task-selection rules performed well with respect to the performance measures, while others did not. For instance, Rule 1 did not perform well – its performance measures were dominated by most of the other rules. Also, Rule 3 did not perform well. On the other hand, rules 5, 6, and 7 performed relatively well – their output performance measures provided generally favorable results. As far as the three "informal" rules are concerned, Rule 8 provided average performance relative to the other rules. Rules 9 and 10 provided generally strong performances

Table 4
Crew and equipment requirement by rule

Rule	Crew requirement	Equipment requirement
1	32.83 (15.67)	79.17 (25.71)
2	31.50 (14.83)	112.83 (63.89)
3	32.33 (15.28)	102.33 (50.83)
3	32.33 (15.28)	102.33 (50.83)
4	31.33 (14.51)	132.33 (83.19)
5	31.33 (14.40)	180.33 (106.7)
6	31.50 (14.42)	176.83 (131.76)
7	31.83 (14.74)	174.67 (139.14)
8	31.67 (14.46)	122.00 (59.70)
9	29.83 (13.60)	1402.67 (1326.18)
10	49.17 (23.38)	49.17 (23.38)

with the exception of system utilization for Rule 10. Rule 10 yielded a very low utilization because each task was in its own unique work center, which minimized the work required in the work centers, and thereby minimized overall system utilization.

While Table 3 shows details of the output performance measures for each of the task-selection rules, Table 4 shows details of the inputs required for each of the task selection rules – specifically labor requirement and equipment requirement.

While Table 3 shows Rules 9 and 10 to be relatively strong performers in terms of output performance measures, Table 4 shows them to be quite expensive in terms of resource requirements. While Rule 9 was quite attractive in terms of requiring the fewest workers, it required a very large amount of equipment due to the amount of paralleling necessary. Rule 10 is most attractive in terms of needing the least amount of equipment, but it requires the most workers because each task comprises a single work center, diminishing the utilization of the entire layout. It is interesting to note that the crew requirement from rule to rule does not vary a great deal when compared to the variation of equipment requirement from rule to rule. In other words, equipment requirement was more sensitive to task-selection rules ($F = 135.58$, $p < 0.0001$), where crew requirement was not as sensitive to task-selection rule ($F = 19.05$, $p < 0.0001$). From a resource usage perspective, Rule 1 appears to offer the best balance between crew and equipment requirements.

The second research question seeks information about which task-selection rules generally provide the best layouts. While it is clear some rules perform well while others do not, consideration should also be given to the relationship between the number of work centers and the output performance measures. The ten task-selection rules did result in differing amounts of work centers ($F = 245.17$, $p < 0.0001$), but it is worthwhile to explore how the output measures were influenced by the number of work centers. To address this third research question, linear regression was used. The number of work centers was used as the lone independent variable, while the output measures were used as the response variables. Table 5 show the regression statistics for each of the output measures.

While the linear regression models show a strong relationship between the number of work centers and the output measure (despite some low R^2's), it is important to note that more work centers seemed to increase average WIP level, throughput and on-time completion, while decreasing average system utilization and percentage of desired cycle time

Table 5
Regression statistics for output measures in response to number of work centers

Output measure	β	t-statistic	R^2	F	$p <$
WIP	2.2746	15.52	0.1386	240.95	0.0001
Throughput	4.3552	16.66	0.1563	277.57	0.0001
PCCT[a]	0.0018	10.80	0.0722	116.65	0.0001
Utilization	− 0.0074	− 42.30	0.5443	1789.19	0.0001
PCTA[b]	− 0.0021	− 10.69	0.0709	114.31	0.0001

[a] Percent of units completed within cycle time.
[b] Percentage of desired cycle time attained.

attained. In other words, more work centers were desirable for throughput and on-time completion, but were undesirable for the other three output measures.

6. Conclusions

A heuristic was presented to perform assembly line balancing for mixed models exhibiting stochastic task durations when paralleling of tasks within work centers is permitted. Using this heuristic, seven task-selection rules were presented to provide the user with alternative strategies. Based on a simulation experiment with six different problems, the performance of the layouts from the seven task-selection rules was compared with that of three additional, simple task-selection rules. Generally, the experiment showed that task-selection rules 4, 5, 6, 7, 9, and 10 performed well in terms of output measures while Rules 1 and 3 did not. Though Rules 9 and 10 performed well, the "expense" associated with their implementation must be addressed – Rule 9 required a large amount of equipment and Rule 10 required substantially more workers than the other rules. Rules 2 and 8 seemed to provide "average" performance relative to the other rules in terms of the output measures.

While this research does provide light on which task-selection rules seem to perform better, the primary goal was to provide a methodology that addresses the mixed-model problem with stochastic task durations and paralleling of tasks. The task-selection rules and the experimental design were intended to provide the user with options pertaining to the assembly line design decision.

Each of the presented task-selection rules, or strategies, has two general attributes: output performance measures (outputs) and resource requirements of workers and equipment (inputs). Since both inputs and outputs are involved in the assembly line-balancing decision, data envelopment analysis (DEA) could be used to determine which task-selection rules are more efficient regarding the relationship between the inputs and outputs. This is an opportunity for future research.

References

Arcus, A.L., 1966. COMSOAL: a computer method of sequencing operations for assembly lines. Int. J. Prod. Res. 4(4), 259–277.

Bowman, E.H., 1960. Assembly line balancing by linear programming. Oper. Res. 8(3), 386–389.

Buxey, G.M., 1974. Assembly line balancing with multiple stations. Mgmt. Sci. 20(6), 1010–1021.

Carter, J.C., Silverman, F.N., 1984. A cost-effective approach to stochastic line balancing with off-line repairs. J. Oper. Mgmt. 4(2), 145–157.

Ding, F., Cheng, L., 1993. An effective mixed-model assembly line sequencing heuristic for just-in-time production systems. J. Oper. Mgmt. 11(1), 45–50.

Driscoll, J., Abdel-Shafi, A.A.A., 1985. A simulation approach to evaluating assembly line balancing solutions. Int. J. Prod. Res. 23(5), 975–985.

Gaither N., 1996. Production and Operations Management, 7th ed. Duxbury Press, Boston, MA.

Ghosh, S., Gagnon, R.J., 1989. A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. Int. J. Prod. Res. 27(4), 637–670.

Helgeson, W.B., Birnie, D.P., 1961. Assembly line balancing using the ranked positional weight technique. J. Ind. Eng. 12(6), 394–398.

Ignall, E.J., 1965. A review of assembly line balancing techniques. J. Ind. Eng. 16(4), 244–254.

Kilbridge, M.D., Wester, L., 1962. A review of analytical systems of line balancing. Oper. Res. 10(5), 626–638.

Klein, M., 1963. On assembly line balancing. Oper. Res. 11(2), 274–281.

Kottas, J.F., Lau, H.S., 1981. A stochastic line balancing procedure. Int. J. Prod. Res. 19(2), 177–193.

Leu, Y., Matheson, L.A., Rees, L.P., 1994. Assembly line balancing using genetic algorithms. Decision Sci. 25(4), 581–606.

McMullen, P.R., 1995. A simulation approach to solving the type I assembly line balancing problem for mixed models with stochastic task durations. Doctoral Dissertation, University of Oregon.

Moodie, C.L., Young, H.H., 1965. A heuristic method of assembly line balancing for assumptions of constant or variable work element times. J. Ind. Eng. 16(1), 23–29.

Nkasu, M.M., Leung, K.H., 1995. A stochastic approach to assembly line balancing. Int. J. Prod. Res. 33(4), 975–991.

Patterson, J.H., Albracht, J.J., 1975. Assembly line balancing zero–one programming with fibonnaci search. Oper. Res. 23 166–172.

Peterson, C., 1993. A tabu search procedure for the simple assembly line balancing problem. Proceedings of the Decision Sciences Institute Conference, Washington, DC. pp. 1502–1504.

Pinto, P.A., 1975. Assembly line balancing with paralleling. Doctoral Dissertation, University of North Carolina at Chapel Hill.

Pinto, P.A., Dannenbring, D.A., Khumawala, B.M., 1975. A branch and bound algorithm for assembly line balancing with paralleling. Int. J. Prod. Res. 13(2), 183–196.

Salveson, M.E., 1955. The assembly line balancing problem. J. Ind. Eng. 6(3), 18–25.

Silverman, F.N., Carter, J.C., 1986. A cost-based methodology for stochastic line balancing with intermittent line stoppages. Mgmt. Sci. 32(4), 455–463.

Suresh, G., Sahu, S., 1994. Stochastic assembly line balancing using simulated annealing. Int. J. Prod. Res. 32(8), 1801–1810.

Thomopoulos, N.T., 1967. Line balancing sequencing for mixed-model assembly. Mgmt. Sci. 14(2), 59–75.

Tonge, F.M., 19xx. Assembly line balancing using probabilistic combinations of heuristics. Mgmt. Sci. 11(7), 727–735.

Vrat, P., Virani, A., 1976. A cost model for optimal mix of balanced stochastic assembly line and the modular assembly system for a customer oriented production system. Int. J. Prod. Res. 14(4), 445–463.