The Mixed and Multi Model Line Balancing Problem: A Comparison

J.I. van Zante-de Fokkert and A.G. de Kok*

Research Report TUE/BDK/LBS/95-07
June, 1995

* Graduate School of Industrial Engineering and Management Science
Eindhoven University of Technology
P.O. Box 513, Paviljoen F16
NL-5600 MB Eindhoven
The Netherlands
Phone: +31.40.472230
Fax: +31.40.464596

This paper should not be quoted or referred to without the prior written permission of the author
The Mixed and Multi Model Line Balancing Problem: A Comparison

Jannet van Zante-de Fokkert*, Ton de Kok*

Abstract

This paper gives a review of the literature on the mixed and the multi model line balancing problem. Compared with the single model line little attention has been paid to the mixed and the multi model line balancing problem. The mixed model line balancing problem can be subdivided into two related problems. The balancing problem deals with the assignment of work elements to work stations. The sequencing problem deals with the determination of the sequence in which the models are produced. It appears that the literature on the mixed model line balancing problem consists of two different approaches that transform the mixed model line into a single model line balancing problem. To accomplish this transformation, the first procedure defines a combined precedence diagram and the second approach uses average task processing times. An experiment was carried out in order to compare several heuristics that are based on the combined precedence diagram, on their performance. An optimisation method was added. The results indicate that the position of common work elements in the precedence diagram of the different models influences both the CPU time of a heuristic and the way in which the total work content of the single models is spread over the work stations. The results also show that good solutions with regard to the number of required stations go together with long CPU times. For a number of instances that took a long CPU time, we can however decrease these CPU times considerably (about factor 20) without deteriorating the performance of the methods by applying the heuristics to the reversed combined precedence diagram.

* Eindhoven University of Technology, Department of Operations Planning and Control, Paviljoen F14, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
1. Introduction

In general, flow lines consist of a number of work stations or operators that are linked together by a conveyor or some other material handling system. Each work station carries out some tasks (tasks) and the consecutive execution of these tasks completes the assembly of a product. Moreover, the sequence in which the work stations are passed through, is the same for every product.

Buxey, Slack, and Wild (1973), identified two variants of mass production flow lines. The first class of these two variants are the single model lines, which are dedicated to the production of one single model or product. The second class of lines are designed for the assembly of more than one model. This latter category can be subdivided into mixed model and multi model flow lines. On a multi model line the different models are produced in (large) batches. On mixed model lines however, the lot sizes equal one. This results in the simultaneous assembly of two or more different models on the same production line.

The line balancing problem is related to the design of a flow line. The assembly of a product can be divided into a number of tasks. The precedence relations between these tasks are placed in a precedence diagram. The line balancing problem deals with the assignment of these tasks to the successive work stations of the line, without violating the precedence relations between the tasks.

In their review articles, Bhattacharjee and Sahu (1986) and Ghosh and Gagnon (1989) gave the following classification of the literature on the line balancing problem:

- The single model line balancing problem:
  - deterministic task processing times
  - stochastic task processing times
- The mixed and multi model line balancing problem:
  - deterministic task processing times
  - stochastic task processing times

The mixed and multi model line balancing problem can be subdivided into two related problems (Macaskill (1972), Okamura and Yamashina (1979), and Wester and Kilbridge (1963)). Firstly the balancing problem that deals with the assignment of tasks to operators (we assume that a work station is manned by one operator). Secondly the sequencing problem that deals with the determination of the sequence in which the models are produced. For multi model lines the optimal batch sizes have to be determined as well.

Although much research has been done for the single model line balancing problem, only a few papers discuss the mixed model and multi model line balancing problem. Our literature survey discussed in Section 3 and 4, shows that most of the articles in this area date back to before 1980. We foresee a renewed interest in the mixed model and multi model balancing problem because of the evolution of the change-over problems in mass assembly. Increased application of software-controlled equipment results in change-overs that have more and more an organizational instead of a technical character.

This paper presents not only a review of the literature on the mixed model and multi model line balancing problem, but also compares different balancing heuristics on their performance. In Section 2 the single model line balancing problem is briefly described. Most of the given references in this section also dealt with the mixed model line balancing problem. Section 3 and 4 present methods for the solution of the mixed model and multi model balancing problem, respectively. Section 5 gives the design and the results of the experiment, that was
carried out in order to compare some of these methods mutually. Finally, some conclusions and research directions are given in Section 6.

2. The single model line balancing problem

To measure the quality of a solution to the single model line balancing problem we can use two different performance criteria (Arcus (1966), Buxey, Slack, and Wild (1973), Ghosh and Gagnon (1989), Helgeson and Birnie (1961), and Thomopoulos (1970)). These two performance criteria result in the following two problem formulations:

(P1) Minimize \( n \)
subject to
\[
\sum_j jx_{ij} \leq \sum_j jx_{pj} \text{ if task } i \text{ preceeds task } p \quad (1)
\]
\[
\sum_j x_{ij} = 1 \quad \forall i \quad (2)
\]
\[
\sum_i t_i x_{ij} \leq C \quad \forall j \quad (3)
\]
\[
x_{ij} \in \{0,1\} \quad \forall i,j \quad (4)
\]

(P2) Minimize \( C \)
subject to
\[
\sum_j jx_{ij} \leq \sum_j jx_{pj} \text{ if task } i \text{ preceeds task } p \quad (5)
\]
\[
\sum_j x_{ij} = 1 \quad \forall i \quad (6)
\]
\[
x_{ij} \in \{0,1\} \quad \forall i,j \quad (7)
\]
\[
j \leq n, \quad (8)
\]

where \( n \) is the number of work stations, \( t_i \) the processing time of task \( i \), and \( C \) the cycle time that is defined as the time between two successive products coming off the end of the line (Chase and Aquilano (1992), chapter 9) or, equivalently, the amount of time available to an operator to perform the tasks that are assigned to his work station. The decision variable \( x_{ij} \) equals 1 if task \( i \) is assigned to work station \( j \) and 0 otherwise. Constraints (1) and (5) represent the precedence relations and Constraints (2) and (6) warrant that every task is assigned to exactly one work station.

In Problem (P1) the number of work stations is minimised. In this situation the cycle time \( C \) is given. Constraints (3) indicate that all work stations must be able to perform the assigned tasks within this cycle time.

In Problem (P2) the cycle time is minimised. In this case the number of work stations is given. Constraint (8) defines the upper bound for the number of work stations.

In fact both problems are very similar. They aim at equalization of \( nC \), which is the time available to assemble a product, and \( \sum t_i \), which is the time required to produce a single product. Equation (9) gives the balance delay, \( d \), which is a numerical expression for the balance between the available and the required time (Chow (1990), Kilbridge and Wester (1961), and Thomopoulos (1968)).
Note that $nC - \sum t_j$ equals the idle time of the work stations.

Beside the precedence relations other constraints can be added to the balancing problem. The approximation algorithm of Arcus (1966), for example, dealt, among other things, with tasks that require two men or that require a fixed station, tasks with processing times larger than $C$, and time required to obtain a tool. Johnson (1983), gave a branch-and-bound algorithm for two modifications of the original balancing problem, the planning of imbalance between stations and the assignment of tasks to specific station types. These adjustments allowed him to handle several more special situations, including the treatment of task separation, where two tasks require different work stations.

Compared with the line balancing problem for deterministic task processing times, little attention has been paid to the stochastic line balancing problem. Vrat and Virani (1976) gave a heuristic for the latter problem, that has the objective to minimize the total assembly costs, consisting of the labour costs and incompletion costs. They assumed the task processing times to be normally distributed and defined the expected incompletion costs of task $k$ to be the multiplication of the costs of finishing the product, that has been built up to and including task $k-1$, off the line, and the probability of incompletion of this task. Incompletion costs occur when the tasks that are assigned to a work station require more operation time than the cycle time permits. The assignment of a particular task to a work station in this balancing method is based on a comparison of labour costs and expected incompletion costs. The procedure of Vrat and Virani can also handle parallel work stations for tasks with processing times that exceed $C$.

As Bhattacharjee and Sahu (1986) stated, the emphasis of the research in line balancing has shifted from optimisation methods towards approximation algorithms. Arcus (1966) tested the performance of different approximation rules, according to which the tasks are assigned to the work stations. The branch-and-bound algorithm of Johnson (1983) provided an optimal solution.

The techniques that were developed to solve the single model balancing problem, can also be applied to the mixed model line balancing problem, as follows from the following section.

3. The mixed model line balancing problem

The single model and the mixed model line balancing problem differ in the precedence constraints. On a mixed model line every model has its own precedence diagram, and a balance may not violate any of these orderings. For a single model line balancing problem, however, only one precedence diagram is given.

The papers that dealt with mixed model line balancing were almost all based on the same principle. In general the mixed model line balancing problem was transformed into a (less difficult) single model line balancing problem. Hence, techniques that were developed for the solution of the single model line balancing problem could also be used to balance a mixed model line. The methods that appeared in the literature for this transformation can be classified in two different approaches. The first one combines the precedence diagrams of the $m$ models into a single, so-called combined precedence diagram. The second method uses adjusted task processing times. In the following both approaches will be discussed more extensively. Section 3.3. gives some methods that are not based on a transformation of the mixed model into the single model line balancing problem.
3.1. The combined precedence diagram methods

Thomopoulos (1967, 1970) was the first researcher who used the combined precedence diagram for the balancing of a mixed model line. Macaskill (1972), gave a formal description of the combination of a number of single models into this diagram. We summarize this procedure in the following paragraph.

Assume that there are \( m \) different single models. The precedence diagram of model \( i \) can be represented by a directed graph \( (N_i, A_i) \), where \( N_i \) is the set of tasks of model \( i \) and \( A_i \) is the set of precedence relations between these tasks.

The combined precedence diagram that is composed of these \( m \) models, can be represented by a directed graph \( G=(N, A) \), where \( N = \bigcup_{i=1}^{m} N_i \) and \( A = \bigcup_{i=1}^{m} A_i \setminus \text{redundant arcs} \). An arc from node \( i \in N \) to node \( j \in N \) is called redundant if there exists another path from node \( i \) to node \( j \) in \( G \). The task processing time of a node \( i \in N \) is equal to the total time that is required for the processing of this task in a given model mix. This model mix defines the number of units of a model that has to be produced during a shift of duration \( T \). Figure 1 on page 5, illustrates the above described procedure. Note that the redundant arcs have been indicated by dotted arrows.

The balancing of the mixed model line based on the combined precedence diagram can be compared with the balancing of a single model line. Note however that in the former case the tasks are assigned to the work stations on basis of \( T \), instead of on basis of the cycle time \( C \) as in the single model line balancing problem.

As mentioned before, Thomopoulos (1967, 1970) was the first to use the combined precedence diagram for the balancing of a mixed model line. In his paper of 1967 he used the single model line balancing heuristic of Kilbridge and Wester (1961) to minimize the number of work stations. Figure 1 shows the columns in which they divided the precedence diagram. A task is assigned to column \( i \), if the maximum of the column numbers of his predecessors equals \( i-1 \). This column identification places a task in a column with the lowest index as possible. It is however possible that a task can be moved to a column more to the right in the precedence diagram without violating any precedence restrictions or column assignments of other tasks. The numbers of these columns are characteristic for each task.

The line balancing method of Kilbridge and Wester is based on the assignment of columns to a work station. If the work content of a column \( i \) exceeds the (remaining) available time of a work station, then all possible combinations of tasks in this column are investigated in order to find a perfect assignment. A perfect assignment of tasks to a work station has a work content that equals \( C \) (or \( T \) in case of a mixed model line). If a perfect assignment can not be found, all tasks in column \( i \) up to and including \( i \) are investigated on their ability to move to a column \( \geq i \). The task with the smallest task processing time is moved to column \( i \) \((i+1)\) if its column number is smaller than \( i \) (equals \( i \)). This sequence of combination and movement is repeated until a perfect assignment is found or until no feasible movements are left. In the latter case the assignment with the highest work content is selected. The assigned tasks are deleted from the precedence diagram, the column numbers are adjusted, and the procedure is repeated for the following work station.
It should be noted that this heuristic does not need to take into account the precedence relations explicitly because of the assignments of columns. Tasks in the same column are exchangeable without violating the precedence constraints. Furthermore this method can have an extremely long CPU time because of the investigation of all combinations of tasks in a column.

The mixed model line balancing method of Macaskill (1972) was also based on the combined precedence diagram. However, he used the single model line balancing heuristic of Helgeson and Birnie (1961). In this method T is given and for each of the tasks a ranked positional weight (RPW) is defined, which equals the sum of the processing time of the specific task and the processing times of its (in)direct successors. During the procedure of the assignment of tasks to successive work stations, an available list is kept of tasks of which the predecessors are assigned already. A selection is made of tasks of this availability list that have a task processing time that is lower than or equal to the remaining available time of the work station.
If there are no such tasks then the current work station is closed and the following one is opened. Otherwise, the task with the highest RPW from this selection is assigned to the current station. This procedure is repeated until the availability list is empty.

There are several advantages and disadvantages of the combined precedence diagram method. An advantage is that every repetition of a task is carried out by the same operator. This results in a minimum of learning costs. A disadvantage of the balancing on shift basis is that another model mix can lead to another balance, which creates a lot of confusion on the shop floor.

Another negative effect is that the above described procedure can only be applied if the resulting combined precedence diagram is acyclic. This shortcoming results in severe restrictions on the precedence diagrams of the m different models. Ahmadi and Wurgaft (1994) give two solutions for the problem that arises if the precedence diagrams can not be combined. The first solution divides the set of models into smaller subsets, which can be combined. The second solution repeats a number of tasks in the combined precedence diagram in order to remove the cycles. It should be noted that this latter solution does not guarantee that every repetition of a task is carried out by the same operator. We did not use any of those two methods for the generation of the combined precedence diagrams, discussed in Section 5.1.

A third disadvantage of the method is the unequal distribution of the total work contents of single models among the work stations, measured by $\Delta$, which was introduced by Thomopoulos (1970). Here $\Delta$ was defined as:

$$\Delta = \sum_{i=1}^{n_{\text{min}}} \sum_{j=1}^{m} |P_j - P_{ij}|,$$

where $m$ is the number of different models, $P_j$ is the average work content of all units of model $j$ per work station ($P_j = \text{total work content of model } j / n_{\text{min}}$), and $P_{ij}$ is the total work content of model $j$ that is assigned to work station $i$.

The consequences of this uneven sharing become obvious when we consider a sequence in which models that require a relatively large processing time at the same work station, are scheduled after each other. If the sequence in which the different models are fed to the line alternates models with a large work content at a particular station and models with a small work content, then a part of the difficulties that are introduced by this unequal balancing of single models can be neutralised. An even distribution of the work contents of single models among the work stations can reduce the complexity of the sequencing part of the mixed model line balancing problem considerably (Dar-EI and Cucuy (1977)). Thomopoulos (1970) gave a heuristic that pursues this objective. In his approach the minimum number of work stations $n_{\text{min}}$ was given. With the help of $n_{\text{min}}$ a lower bound and an upper bound of $T$ are determined. Given the assignment of tasks to preceding work stations, the heuristic investigates a predefined number of feasible assignments of tasks to a particular work station. An assignment is feasible if none of the precedence restrictions is violated and if the total work content does not exceed the upper bound of $T$. If during the investigation a feasible assignment is found with a work content that exceeds the lower bound of $T$, then the value of $\Delta$ is determined.

The investigation procedure continues until the predefined number of feasible assignments is achieved, or until an assignment with $\Delta=0$ is found. If there are no feasible assignments in the former case, that have a work content that exceeds the lower bound of $T$, then the assignment with the highest work content is chosen. Otherwise, the assignment with the minimum $\Delta$ is selected among the feasible assignments with a work content that exceeds the lower bound of $T$. 

6
Berger, Bourjolly, and Laporte (1992) described a branch-and-bound algorithm for the multi product line balancing problem, that is also based on a combined precedence diagram. This branch-and-bound method is an adjustment of the algorithm suggested by Hackman, Magazine, and Wee (1989). Multi products were defined as products that have linear precedence diagrams. This means that the products have only one possible order in which the tasks can be assigned to the successive work stations. Moreover, if two products have k tasks in common, then these tasks are the first k tasks required for the production of both products. This latter restriction results in a combined precedence diagram that is a forest.

The algorithm uses depth-first search. The nodes in level i of the search tree correspond to a maximal assignment of tasks to the i\textsuperscript{th} work station. An assignment is maximal if it can not be extended with another task without exceeding the available, predefined T. In each node a lower bound and an upper bound for the resulting solution can be determined. We refer to Berger et al. for an accurate description of the computation of the lower bound. The upper bound is determined by the so called IUFF6-heuristic (Hackman, Magazine, and Wee (1989)). This heuristic is similar to the above described ranked positional weight (RPW) method of Helgeson and Birnie (1961), but uses the processing time of the specific task instead of the RPW as selection criterion.

Berger et al. used their branch and bound algorithm as a heuristic, by the application of a partial node generation (they choose six nodes with the least idle time out of the generated maximal assignments). Furthermore, they compared their method with the basic algorithm of Hackman et al. and an optimal one. Although the definition of multi products results in a special case of the mixed model line balancing problem (the combined precedence diagram is a forest), we can apply this method to the general mixed model line balancing problem.

A common characteristic of the above described heuristics of Thomopoulos (1967, 1970) and Macaskill (1972) is the extension of a partial solution. The work stations are filled successively and none of these methods returns to an earlier made assignment for further research. The branch-and-bound algorithm of Berger et al. (1992) on the other hand, may return to an earlier made assignment. All these heuristics can also be used to provide an initial solution for local search algorithms, that investigate a number of complete solutions.

A heuristic for the so-called "machine allocation and staging problem", that is also based on the combined precedence diagram, was given by Ahmadi and Wurgaft (1994). They assumed that a station consists of a number of identical machines. The given heuristic firstly determines the number of machines that is assigned to a station (the machine allocation problem) and secondly assigns the tasks to the machines (the staging problem). The assignment of tasks to machines is restricted by the precedence relations and a maximum number of tasks per machine. If this maximum number is infinite and if the number of machines equals the number of stations, then the machine allocation and staging problem reduces to the mixed model line balancing problem.

They distinguished between three different situations. The first situation concerns linear combined precedence diagrams, which can be solved with a network method in combination with binary search on the number of machines. The second situation concerns the non-linear combined precedence diagrams, for which an approximation algorithm was given, that combines the above mentioned optimisation algorithm and a heuristic that forms a linear graph of the diagram. The last situation concerns unconnected diagrams, without any precedence relations. The heuristic that is supposed for these diagrams, is based on a division between the allocation and the staging problem.
As for the papers on single model line balancing described in Section 2, the objective of the main part of the literature on mixed model line balancing has been to minimize the idle time of the work stations. Note that this corresponds to the minimization of the labour costs. Chakravarty and Shhtub (1985) considered besides these labour costs also the set-up costs and the inventory costs in their balancing approach. They assumed that the models are produced in batches and that a batch is transported to the following station in its entirety. Further they allowed the batch sizes to vary between stations by placing buffers between two adjacent work stations.

3.2. The adjusted task processing times methods

The second method that is used to transform the mixed model line balancing problem into a single model line balancing problem, determines average task processing times for the tasks that are required by more than one model (Arcus (1966), Johnson (1983), Milas (1990), and Vrat and Virani (1976)). The average task processing time of task \( i \), \( \bar{t}_i \), can be calculated with the following formula:

\[
\bar{t}_i = \sum_j f_j t_{ij},
\]

where \( t_{ij} \) equals the processing time of task \( i \) in model \( j \) and \( f_j \) is the probability that the model in production is model \( j \) or, equivalently, \( f_j \) is the relative frequency of model \( j \). Vrat and Virani (1976), who dealt with stochastic task processing times that follow a normal distribution (see also Section 2), used a similar equation for the average standard deviation of the processing time of task \( i \).

The above mentioned authors suggested balancing procedures that can tackle the single model balancing problem created by the above adjustment of the task processing times in the mixed model line balancing problem.

Like the combined precedence diagram methods, this transformation technique also has some advantages and disadvantages. An advantage is that the balancing procedure is based on \( C \) instead of the combined precedence diagram methods, that balance on shift basis. There is however no method that determines the sequence in which the models are produced. Another shortcoming of the adjusted task processing times methods is that they do not take the eventually different precedence diagrams of the models into account. This makes the methods only useful for the balancing of models, that are all derived from the same model, by increasing or decreasing the task processing times of this underlying model.

3.3. Alternative methods

Roberts and Villa (1970) proposed a quite different method that can not be placed in one of the two categories that are discussed above in Section 3.1. and Section 3.2. This method solves the mixed model line balancing problem optimally by using a 0-1 linear integer programming formulation of the balancing problem. The total idle time is minimised and the cycle time and the number of work stations are assumed to be the same for all models. Because this 0-1 formulation requires a large number of variables, it can only be used to solve small problems. For large balancing problems, the authors suggested a network approach. The nodes in this directed network represent possible assignments to the first station for all models. These assignments may violate the cycle time constraints (Constraints (3) of Problem (P1)). If a node is an extension of another node, which means that it represents an assignment
to the first station that consists of the same and some more tasks for all models, and if the tasks in this extension can be carried out within C, then the nodes are connected. The length of the arc equals the total idle time. A shortest route in this network from the node that represents no assignment to the node that represents full assignment, corresponds to a balance that minimizes the total idle time.

Another problem that is related to mixed model lines is the assignment of the m models to a given number of different mixed model lines. Each line has a maximum production capacity that may not be exceeded. Lehman (1969) gave a heuristic for this problem where the minimization of the total assembly costs was the objective. This costs included the balance delay, the learning of operators, and the sequence delay, which takes the amount of time that an operator works outside his station to complete the job into account. Ahmadi, Dasu, and Tang (1992) gave three heuristics for the 'dynamic line allocation problem'. Given the number of products to be assembled, these heuristics allocate the available mixed model lines to the different products, in order to minimize the change-over costs and waiting costs per period.

The above described problem is also related to multi model lines. The following section discusses the line balancing for multi model lines.

4. The multi model line balancing problem

As already mentioned in Section 1, the mixed model and multi model line balancing problem differ in the lot sizes. On a mixed model line these lot sizes equal one. As a result of the change-over costs and the change-over times on a multi model line, however, these lot sizes are greater than one. The former lines assume (implicitly) that both these change-over costs and change-over times are negligible. Multi model lines, instead,

As Buxey, Slack, and Wild, (1973) stated, the objective of the multi model line balancing problem is to minimize the production costs. These production costs also include the change-over costs that are related to the set-up of the line for the production of another model. Note that this aim differs from the objective of the mixed model line balancing problem, that takes (in most cases) only the idle time of the work station into account.

Buxey et al. (1973) suggested two different methods to balance a multi model line. The first one is used if the lot sizes are large and balances every single model separately. The second approach balances the line for the model that has the largest frequency $f_j$. This basic balance can then be adjusted for the other models.

Chakravarty and Shtub (1985) proposed the procedure that is described in Section 3.1. This method assumes buffer capacity between two adjacent work stations and supposes that the models are produced in batches, which are transported to the following station in their entirety. Moreover, the approach assumes the models to be produced in cycles, which means that between the completion of two batches of the same model, all the other models are produced exactly once.

Thomopoulos (1970) showed that his mixed model line balancing method that gives an even distribution of the work contents of single models among work stations, can also be used for batch production. However, the balancing procedure that is based on the combined precedence diagram, remains the same, and makes no difference between the mixed model and the multi model line balancing problem. Furthermore, both the change-over costs and the change-over times were not taken into account.
The following section describes an experiment to test the combined precedence diagram methods of Section 3.1.

5. A numerical comparison

An experiment was done in order to compare the heuristics of Thomopoulos (1967, 1970), Macaskill (1972), and Berger, Bourjolly, and Laporte (1992), on their performance with regard to the balance of a mixed model line. The performance of the heuristics was tested against an optimisation method, which can be derived from the branch-and-bound algorithm of Berger et al., by allowing a complete node generation, instead of the partial node generation (maximum of six nodes) that Berger et al. use. All proposed balancing methods, with the exception of the second heuristic of Thomopoulos, take a T as given and try to minimize the number of work stations, n (see Problem P1). The second method of Thomopoulos however, can be easily adjusted to this situation by assuming both the upper bound and the lower bound of T to be equal to T itself. The performance of the heuristics was measured by the number of work stations they found, as compared to the optimal number of work stations.

Furthermore, the algorithms were compared with regard to $\Delta$ (Thomopoulos (1970), Formula (10)), the overall balance of the line. Section 5.1. and 5.2. give the design and the results of the experiment, respectively.

5.1. The design of the experiment

The algorithms were tested on several combined precedence diagrams (instances). Each combined precedence diagram was composed of four different models. The models were generated randomly and differed with respect to the following characteristics:

1. The number of tasks. This number is equal for all four models of a combined precedence diagram and can be varied at three levels: 16, 20, or 24 tasks per model.

2. The position of common tasks. We assumed that the combined precedence diagrams are composed of models that share 75% of their tasks. These so-called common tasks have equal task processing times for each model. The remaining 25% of the tasks in a model are specific. We can place these common tasks at three different locations (levels) in the precedence diagrams of the models:
   - at the beginning of the precedence diagrams.
   - scattered over the precedence diagrams.
   - at the end of the precedence diagrams.

3. The number of columns. This number is equal for all four models of a combined precedence diagram. A choice between 4 or 6 columns per precedence diagram can be made, i.e. two levels.

The above mentioned characteristics define the structure of the combined precedence diagrams. However, the task processing times can also be varied. These task processing times in a combined precedence diagram depend on the processing time of tasks in the underlying models and on the demand per model. In order to vary the task processing times in the combined precedence diagrams, it is sufficient to vary only one of these characteristics. We
assumed the processing times of tasks in single models to be constant, which means that corresponding tasks have equal processing times in each model and in each instance. Therefore, the demand per model is the fourth characteristic that can be varied in the instances:

4. Demand per model. We chose the following three levels for the demand per model:
   - Demand per model is the same for all models. Demand pattern: 1-1-1-1.
   - Demand per model is the same for the first three models, demand of the fourth model deviates. Demand pattern: 1-1-1-5.
   - Demand per model varies. Demand pattern: 5-5-1-2.

With respect to the performance of the five algorithms, there is another characteristic that plays an important role. Each method requires a given shift duration, T, as input. This T has to be greater than or equal to the maximum task processing time in the combined precedence diagram and could influence the difficulty of the problem to be solved.

5. The shift duration T. Given a minimal number of work stations (n_{min}), we defined a minimum T, T_{min} ([TPT/n_{min}] if TPT/(n_{min}-1) not integer
   T_{min} = \begin{cases} 
   \lfloor TPT/(n_{min}-1) \rfloor & \text{if } TPT/(n_{min}-1) \text{ not integer} \\
   TPT/(n_{min}-1) - 1 & \text{else} 
   \end{cases}

   Furthermore, we chose a T_{half} which equals T_{min} + [(T_{max} - T_{min})/2].

In fact, the above described design of experiment is a factorial design of experiment with five factors, four of which has three level and one factor (number of columns per model) has two levels. There are no replicates. Because we are only interested in the main effects of this design, a design of resolution IV (Montgomery (1991)) will be sufficient. This means that we can suffice with only a part of the full experiment (54 instances instead of 2·3·3·3·3=162 instances).

An instance corresponds to a combination of levels. Instances were randomly generated. The procedure to generate an instance can be described as follows:

Step 1: Generate a precedence diagram for model 1, that satisfies the restrictions with relation to the number of tasks, the position of the common elements, and the number of columns. This precedence diagram is constructed by the choice of a random number of common and specific tasks for each column. Next, for each task in column i (i>1), one or two predecessors are randomly chosen from the tasks that are assigned to column i-1.

Step 2: Generate a precedence diagram for model j, j=2,3,4, in the same way as described in step 1. Verify if this model is consistent, with respect to the absence of cycles in the combined precedence diagram of model 1 up to and including model j. Generate another model j if it is not consistent. Else, generate a precedence diagram for model
j+1. After 3000 iterations without finding a consistent precedence diagram for model j, a return is made to step 1.

It appeared that the CPU time to generate an instance is mainly determined by the location of the common tasks (see 2). Especially instances with 20 and 24 tasks per model, 6 columns per model, and the common tasks scattered over the precedence diagrams of the models, were very hard to generate within reasonable time. Therefore, we adjusted the above procedure for the situation with 20 tasks by comparing each fourth model not only with the present set of model 1, 2, and 3, but also with the sets of three models that are generated before, and that have not been completed with a consistent model 4 yet. The procedure for the situation with 24 tasks has been adjusted in a similar way. In this procedure, the sets of model 1 and 2 are kept as well, until a consistent model 3 has been found.

5.2. The results of the experiment

The experiment was carried out at an IBM PS/2, model P70, 386. The results are shown in Appendix 1. The instances for which the CPU time of the optimisation method is missing, were solved with help of the reversed precedence diagram method that is explained in Section 5.2.3. An analysis of variance (ANOVA) for the variables n, Δ, and CPU time of each heuristic, was carried out. Section 5.2.1. describes the results. We also compared the five approaches on their performance with regard to n, Δ, and CPU time. The results are discussed in Section 5.2.2.

5.2.1. The impact of the instance structure

An analysis of variance (ANOVA) for the variables n, Δ, and CPU time of each heuristic, was carried out. In this analysis the n of each heuristic was decreased with the optimal number of work stations. Table 1 on page 13 presents the results.

It can be seen that none of the factors has a significant (α=0.05) influence on the number of stations, except for the heuristic of Thomopoulos (1967). The significant effect of the number of columns on the n of this heuristic can be explained from the fact that this method searches all possible combinations of tasks within a column (Section 3.1.). This number of searches increases with the number of tasks in a column and so does the number of assignments of tasks to a work station, which results in smaller idle time.

It can also be seen from Table 1 that both the number of tasks and the demand pattern has a significant effect on Δ. This is a logical result of the definition of Δ (Formula (10)), which increases with the number of units required and with the number of stations (in our experiment equivalent with the number of tasks). If we repeat the analysis for Δ/total demand instead of Δ, then only the optimisation procedure and the heuristic of Berger et al. remain sensitive for the imbalance of the demand between the models. More interesting is the significant effect of the position of the common tasks on Δ. Although none of the heuristics is especially designed for the optimisation of Δ, it appears that instances with the common tasks scattered over the precedence diagram tend to have a higher Δ then instances of the other two categories.

Besides the expected increasing influence of the number of tasks on the CPU time, also the position of the common tasks has a significant effect on this variable. Instances with the common tasks at the end of the precedence diagram take the longest CPU time (about factor
four larger than the CPU times for instances with the common tasks scattered over the precedence diagram). This can be explained by the position of the specific tasks in these instances. Here, the specific tasks are in the beginning of the precedence diagram, resulting in a lot of possible assignments to the first stations. For the same reason instances with the common tasks scattered over the precedence diagram take a longer CPU time than instances with the common tasks at the beginning (about factor 60).

Table 1: The effects of the factors number of tasks, position of common tasks, T, demand pattern, and number of columns on the variables Δ, number of stations, and required CPU time of each heuristic

<table>
<thead>
<tr>
<th>factor</th>
<th>Thomopoulos '67</th>
<th>Thomopoulos '70</th>
<th>Macaskill</th>
<th>Berger et al.</th>
<th>Optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of tasks</td>
<td>n</td>
<td>Δ</td>
<td>cpu</td>
<td>n</td>
<td>Δ</td>
</tr>
<tr>
<td>position of common tasks</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>T</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>demand pattern</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>number of columns</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1) An 'x' means 'has a significant effect on'.

5.2.2. The performance of the heuristics

We also compared the five approaches on their performance with regard to n, Δ, and CPU time. This comparison was based on the results in Appendix 1. The results can be found in Figure 2.

<table>
<thead>
<tr>
<th></th>
<th>Optimal</th>
<th>Berger et al.</th>
<th>Macaskill</th>
<th>Thomopoulos '70</th>
<th>Thomopoulos '67</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>0.00</td>
<td>0.04</td>
<td>0.15</td>
<td>0.22</td>
<td>0.35</td>
</tr>
</tbody>
</table>

Figure 2a: Comparison of the heuristics with regard to the n decreased with optimal number of stations. Underlined heuristics are not significantly different.
Figure 2b: Comparison of the heuristics with regard to $\Delta$. Underlined heuristics are not significantly different.

<table>
<thead>
<tr>
<th></th>
<th>Thomopoulos '67</th>
<th>Berger et al.</th>
<th>Optimal</th>
<th>Macaskill</th>
<th>Thomopoulos '70</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>182.23</td>
<td>188.71</td>
<td>189.63</td>
<td>192.59</td>
<td>218.48</td>
</tr>
</tbody>
</table>

Figure 2c: Comparison of the heuristics with regard to the CPU time (sec). Underlined heuristics are not significantly different.

<table>
<thead>
<tr>
<th></th>
<th>Macaskill '67</th>
<th>Thomopoulos '70</th>
<th>Thomopoulos '70</th>
<th>Berger et al.</th>
<th>Optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>4.92</td>
<td>151.89</td>
<td>373.76</td>
<td>1152.18</td>
<td>&gt;&gt;1152.18</td>
</tr>
</tbody>
</table>

Figure 2a shows that in general both the method of Berger et al. and the optimal procedure performs better than the other heuristics with regard to $n$. These results however can only be realised with long CPU times, as can be seen from Figure 2c. In some cases the optimal method even took days to find the solution. The heuristic of Macaskill gives acceptable solutions within only a few seconds.

The significantly higher values for $\Delta$ that the method of Thomopoulos (1970) gives, as follows from Figure 2c, are just accidentally. If we adjust the method of Thomopoulos (1970) by assuming both the upper bound and the lower bound of $T$ to be equal to $T$, then this heuristic does not take $\Delta$ into account anymore (see Section 3.1.).

5.2.3. The reversed precedence diagram methods

As we mentioned in the beginning of this section, some instances were optimally solved with the reversed precedence diagram method. We designed this method in particular for the instances with the common tasks at the end of the precedence diagram. We noticed that these diagrams took the longest CPU time, compared with the other instances. The relatively large number of tasks in the first column declares this observation.

This number of tasks can be decreased considerably by reversing the combined precedence diagram, which means that all precedence relations are turned. If the heuristics are applied to the reversed instead of the original precedence diagram, then the work stations are filled in a reversed order. The obtained solution, however, is also feasible for the original problem. Hence, we applied the heuristics on the reversed diagram of the above mentioned instances, in order to investigate whether similar solutions could be obtained, with regard to the number of stations, within less CPU time. Appendix 2 shows the results.

Note that both the heuristic of Berger et al. and the optimization algorithm is based on the observation that only maximal assignments of tasks to stations need to be investigated, because any other assignment can not result in a solution with less work stations (Johnson (1983)). For the definition of maximal assignments, we refer to Section 3.1.
Lemma 1 shows that, if the reversed and the original problem were solved with an optimisation method that is based on these maximal assignments, then both solutions have the same optimal number of work stations. Define for the original and reversed problem, respectively, $S$ and $S_{rev}$ as the sets of feasible solutions based on maximal assignments, $x \in S$ and $x_{rev} \in S_{rev}$ as feasible solutions, $f(x)$ and $f(x_{rev})$ as the number of work stations of a feasible solution, and $x^*$ and $x_{rev}^*$ as the optimal solutions ($f(x^*) \leq f(x) \forall x \in S$; $f(x_{rev}^*) \leq f(x_{rev}) \forall x_{rev} \in S_{rev}$).

**Lemma 1**

$f(x^*) = f(x_{rev}^*)$

**Proof**

First, define $SP$ as the set of all feasible solutions for the original problem. This set contains exactly the same solutions as the set of all feasible solutions for the reversed problem does. Note that both $S$ and $S_{rev}$ is included in $SP$. The observation that any other than a maximal assignment can not result in a solution with less work stations, and the definition of an optimal solution indicate that $f(x^*) \leq f(x) \forall x \in SP$ and that $f(x_{rev}^*) \leq f(x_{rev}) \forall x_{rev} \in SP$, which completes the proof.

We compared $n$ and the CPU time of the heuristics for the reversed problem (Appendix 2), with those for the original problem. The results are shown in Table 2.

**Table 2:** The comparison of $n$ and the CPU time of the heuristics for the reversed problem with those for the original problem.

<table>
<thead>
<tr>
<th></th>
<th>Thomopoulos '67</th>
<th>Thomopoulos '70</th>
<th>Macaskill</th>
<th>Berger et al.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU time</td>
<td>$x^{1)}$</td>
<td>$x$</td>
<td>$x$</td>
<td></td>
</tr>
</tbody>
</table>

1) An 'x' means 'significant difference between reversed and original problem'.

Table 2 shows that, except for the heuristic of Thomopoulos (1967), application of the heuristics on the reversed precedence diagram, results in similar solutions, with regard to $n$, within less CPU time. In fact, the CPU time that is required, is decreased with a factor 20 for the heuristic of Thomopoulos (1970) and the branch-and-bound method of Berger et al. The CPU time for the heuristic of Macaskill (1972) is almost halved. Note, that the effect of the position of the common tasks in the precedence diagrams, on the CPU time of the method of Thomopoulos (1967) is not significant (Table 1). Hence, it is not remarkable that the reversing of the combined precedence diagram has no impact on the required CPU time. Current research is directed at further improvement of these CPU times, by defining the column with the least number of tasks in it, as the beginning of the precedence diagram.
6. Concluding remarks and research directions

This article gives a review of the literature on mixed/multi model line balancing. Compared with the single model line balancing problem only little attention has been paid to the balancing of mixed model lines. Even less research has been done in the area of the balancing of multi model lines. The mixed/multi model line balancing problem can be subdivided into two related problems. The balancing part has to solve the assignment of tasks to work stations. The sequencing part deals with the determination of the sequence in which the models are produced.

It is shown that the literature on the mixed model line balancing part consists of two different approaches. Both methods transform the mixed model line balancing problem to a single model line balancing problem. To accomplish this transformation, the first procedure defines a combined precedence diagram and the second approach makes use of average task processing times. An experiment has been done in order to compare several heuristics that are based on the combined precedence diagram, on their performance with regard to the balancing of a mixed model line. The performance of the heuristics was tested against an optimisation method. It appeared among other things that the position of common tasks has a significant effect on the required CPU time and on the unequal distribution of the total work contents of single models among the work stations, \( \Delta \). The results also show that good solutions with regard to the number of required stations go together with long CPU times. For the instances with the common tasks at the end of the precedence diagram, that took a long CPU time, we can, however, decrease these CPU times considerably (factor 20 for some heuristics), without deteriorating the performance of the methods, by applying the heuristics to the reversed combined precedence diagram.

The objective of the main part of the literature on mixed model line balancing has been to minimize the idle time of the work stations. Other criteria, such as the length of the line or the risk of stopping the conveyor that transports jobs between work stations (Dar-El and Cother (1975), Dar-El and Cucuy (1977), and Okamura and Yamashina (1979)), are examined in the sequencing part of the balancing problem. We propose to consider these criteria earlier, in the balancing procedure, to reduce both the complexity of the sequencing procedure and the 'costs' associated with the obtained sequence. Note that Thomopoulos (1970) already considered this aspect in his mixed model balancing procedure that gives a more equal distribution of the total work contents of single models among the work stations.

The results in Appendix I indicate that the imbalance of this total work content of single models, can be significantly high in relation to the duration of the shift \( T \). It should be noted that in practice, mixed model lines, i.e. lines that produce products in very small lot sized, prefer lot sizes that equal one. It is required that both the number of work stations, \( n \), and the cycle time, \( C \), are more or less equal for all models. This imposes severe constraints on the assignments of the activities to the work stations. The imbalance might be avoided by having the activities that cause this imbalance, executed at separate feeding lines with a different cycle time. This approach is frequently applied in practice. The identification of such activities and the design of such feeding and main lines will be subject of further research.

To quote Buxey, Slack, and Wild (1973): "The main change-over (between the production of different models on a multi model line) costs are connected with reallocation of stock (parts) and equipment to work stations and learning curves of operatives in new jobs. To reduce these, the number of stations and location of equipment should be constant whenever possible, and tasks common to more than one model should always be performed by the same operator." A balancing procedure for multi model lines has to be developed which takes these
change-over costs into account.

Change-overs in mass assembly have more and more an organizational character instead of a technical character. This is caused by the increased application of software-controlled equipment. Change-over times are determined by the complexity of material flows through and into the production process. Furthermore the secondary effect of regaining work speed for manual workers after change-over cannot be neglected anymore by the reduction of the production batch sizes. The trend towards more diversity caused by a customer-oriented attitude (Womack, Jones, and Roos (1991)) and the reduction of product life cycles shows that the problem sketched above gets more and more important. Researchers should be motivated by this evolution to shift their attention from the single model line balancing problem to the more realistic multi/mixed model balancing problem.

References


Milas, G.H. (1990), 'Assembly Line Balancing...Let's Remove the Mystery', *Industrial Engineering*, May '90, 31-36.


Thomopoulos, N.T. (1968), 'Some Analytical Approaches to Assembly Line Problems', *The Production Engineer*, July, 345-351.


<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column 1</td>
<td>Column 2</td>
</tr>
<tr>
<td>Column 3</td>
<td>Column 4</td>
</tr>
<tr>
<td>Column 5</td>
<td>Column 6</td>
</tr>
<tr>
<td>Column 7</td>
<td>Column 8</td>
</tr>
<tr>
<td>Column 9</td>
<td>Column 10</td>
</tr>
</tbody>
</table>

**Notes:**
- The table contains various columns with numerical values.
- The values range from small to large.
- The table is organized in a structured manner with clear headings for each column.
- The table includes a variety of numerical data, likely for analysis or comparison.
<table>
<thead>
<tr>
<th>Position of Sensors</th>
<th>Demand Pattern</th>
<th>Number of Sensors</th>
<th>Number of Simulations</th>
<th>Original Simulation Mean</th>
<th>Original Simulation Standard Deviation</th>
<th>Reversal Simulation Mean</th>
<th>Reversal Simulation Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>7.00</td>
<td>7.00</td>
<td>2.97</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>7.00</td>
<td>8.00</td>
<td>0.28</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>6.00</td>
<td>6.00</td>
<td>2.42</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>7.00</td>
<td>8.00</td>
<td>0.28</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>6.00</td>
<td>6.00</td>
<td>2.41</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6.00</td>
<td>6.00</td>
<td>0.33</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>10.00</td>
<td>9.00</td>
<td>0.62</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>9.00</td>
<td>9.00</td>
<td>3.51</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>8.00</td>
<td>9.00</td>
<td>5.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>8.00</td>
<td>9.00</td>
<td>32.58</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>8.00</td>
<td>9.00</td>
<td>28.39</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>12.00</td>
<td>11.00</td>
<td>487.30</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>12.00</td>
<td>11.00</td>
<td>1918.27</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>11.00</td>
<td>10.00</td>
<td>482.36</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>11.00</td>
<td>11.00</td>
<td>1940.90</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>11.00</td>
<td>10.00</td>
<td>482.46</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>11.00</td>
<td>11.00</td>
<td>1918.93</td>
</tr>
</tbody>
</table>