Objectives for Order–Sequencing in Automobile Production

Christoph Engel and Jürgen Zimmermann
University of Karlsruhe

Alfons Steinhoff
IBM Informationssysteme GmbH

Abstract: The problem of sequencing units on a mixed–model assembly line can be viewed with several objectives in mind. This paper presents different optimization criteria and objectives for the order–sequencing problem. Former research has focused mainly on leveling procedures for model–sequencing and has emphasized material supply. In contrast, we provide a polynomial heuristic for order–sequencing by leveling the workload. In the context of automobile production we investigate different sequencing policies and introduce an extended heuristic for the case of color–batch–sequencing. For different types of objectives the performance of the heuristics presented is analyzed, taking known heuristics into consideration.

1 Introduction

Global competition forces enterprises, particularly in automobile industry, to increasing customer orientation and product diversification. With respect to the production system, this results in build–to–order production, out–sourcing of capacities, and the integration of pre–manufactured sub–systems.

Due to that development, there arise new requirements to material supply systems and sequencing procedures. In a *build–to–order production* the configuration of each product is determined by an individual selection of options corresponding to a customer order. In a *build–to–plan production* only a few model types are produced repeatedly. The variation in workload per order increases, if we consider build–to–order production of individual products instead of a build–to–plan production of a few model types.

In order to achieve a smooth workload distribution at the assembly stations, the extension of procedures for model–sequencing to the case of order–sequencing is necessary. A *model–sequence* is a production sequence where each unit represents a model type. In an *order–sequence* each unit of the sequence corresponds to a customer order and, therefore, is individual in its configuration. With respect to automobile production a *batch–sequence* may denote, e.g., a sequence where orders of a uniform color are combined to several color batches. For the basic concepts of assembly line sequencing we refer to [1, 12, 13, 23].

Increasing integration of sub–systems and build–to–order production result in an order–based component fabrication. In connection with Just–in–Time (JIT) production systems, this requires a sufficient look ahead of the order–sequence. In the case of order–sequencing, the underlying model demands, in general, are equal to one. However, many algorithms (approximately) solving the sequencing problem (cf. [10, 17, 24]) consider production rates of the underlying models by determining a model–sequence and, therefore, cannot be applied to the case of order–sequencing. Algorithms that consider production rates, which are not based on model types, can easily be adapted to the case of order–sequencing. In general, we can distinguish between algorithms for model– and order–sequencing or, with respect to the objective, between workload– and component–based approaches.

One of the first approaches considering the workload of models was presented by Thomopoulos [28], who treated the sequencing of assembly lines in combination with the balancing problem. Open and closed stations are described and four kinds of inefficiencies termed *idleness*, *deficiency*, *congestion*, and *utility work* are introduced. The unit to be scheduled next is determined by comparing penalty cost that are incurred by these inefficiencies. A drawback of this proceeding is the accumulation of models with high workload at the end of the sequence.

[28] motivates further research on workload based sequencing by Görke & Lentes [9] and Macaskill [16]. Macaskill alternately schedules a model with lowest penalty cost and a model with highest workload, which does not incur any utility work.

Dar–El [4] and Dar–El & Cother [5] studied the minimization of the line length by building model–sequences with minimal operator displacements from the left station border. These approaches are only useful in combination with the design and the balancing of assembly lines, since a change in the model–mix implies a change in the line length.

Okamura & Yamashina [21] proposed an improvement method for the minimization of the maximal operator displacement from the left station border, which is considered to be equivalent to the risk of stopping the conveyor. Tsai [29] introduced an optimal algorithm for minimizing the maximal operator displacement and the total utility work for the single station case. Finally, Sumichrast et al. [26, 27] transformed the algorithm of Monden [20] to an algorithm for workload leveling instead of leveling the usage of components.

Leveling the variation in component usage is the objective of a second category of algorithms. This idea has been introduced by Monden [20], who describes two scheduling algorithms used by Toyota. The first alternative, known as *Goal Chasing I*, consecutively schedules the model that incurs the minimal mean squared deviation between the expected accumulated component usage and the actual accumulated component usage. A simplified approach, termed *Goal Chasing II*, only takes the few critical parts into consideration and schedules the model that would, if not scheduled, incur the maximal deviation from the expected component usage. Miltenburg [17] adopted the idea of leveling the usage of components and suggested three improved scheduling heuristics. Miltenburg & Sinnamon [18, 19] generalized the previous approach to the case of multi–level production systems by leveling the production rates of the corresponding sub–assemblies.

Apart from the aforementioned priority–rule–based heuristics, various solution techniques for the sequencing problem on mixed–model assembly lines are discussed in the open literature.

Bard et al. [2] suggest a tabu search algorithm which seeks to minimize the total line length and to level the component usage by a multi–criteria objective. Branch and bound techniques have been used by Scholl [23] to minimize work overload, as well as by

Bolat [3], who additionally considered setup costs. Kim et al. [11] present a genetic algorithm for the minimization of the total line length and Rachamadugu & Yano [22] propose a Markov process approach to minimize work overload.

Moreover, Steiner & Yeomans [24, 25] investigated a graph–theoretic procedure to minimize the deviation between actual and expected production rate of models. McCormick et al. [15] devise a transformation to a network flow formulation, whereas Kubiak & Sethi [14] introduce a transformation to the well–known assignment problem. Decker [6] considered a transformation to the traveling salesman problem.

As mentioned in Decker [6] and Domschke et al. [7], the sequencing problem of mixed–model assembly lines shows a close relationship to the permutation flow shop problem including earliest and latest start times. Nevertheless, this problem class cannot easily be adapted to the problem of sequencing mixed–model assembly lines because minimizing the makespan, as the most common objective in permutation flow shop, is not of crucial interest in the context of mixed–model assembly line sequencing.

In the following section, we describe the sequencing problem on mixed–model assembly lines. We discuss possible objectives, before we present a problem formulation. In the third section, we devise an algorithm for the assembly line order–sequencing (AOS) based on the leveling of workload. We illustrate the sequencing procedure by an example and present an extended assembly line batch sequencing algorithm (ABS) with respect to the requirements of automobile production. In the fourth section, we briefly describe experimental results concerning the performance of the AOS–algorithm and evaluate different policies of building sequences in automobile production. Finally, we give conclusions of this study, evaluate the operative usefulness of our algorithms, and give an outlook to further developments of the suggested approach.

2 Problem formulation

In this section we consider the effect of order–sequencing on the performance of mixed–model assembly lines. We discuss different criteria and objectives to evaluate the performance of an assembly line. Then, we briefly classify procedures for the assembly line sequencing problem described in the literature and give a motivation for the concept of workload leveling. Finally, we provide an integer programming formulation for the sequencing problem in question.

In the context of planning and running assembly lines we can distinguish between two main problem types [28]:

- Assembly line balancing
- Determination of an order–sequence

In what follows, we deal with the latter problem, the determination of a "good" order–sequence. In the open literature, the sequencing problem is either considered in a rather short–term context [23] or as part of the process of line balancing [16, 28].

We consider the sequencing problem in relation to the balancing problem for the following reason: the optimization of the line balance as well as the optimization of the order–sequence should improve the efficiency of an assembly line. In order to evaluate the performance of an assembly line in process, we suggest the following criteria:

- Utility work
- Labor utilization
- Component usage

Utility work denotes the work overload which cannot be performed by the regular operators of each station. We distinguish between the distribution and the maximum of utility work per station. The distribution of utility work to the stations and over the time determines the number and allocation of necessary "utility workers". Reducing the maximum utility work is considered to be equivalent to reducing the risk of stopping the conveyor [21]. A high labor utilization obviously corresponds to a high productivity of the assembly line. The labor utilization UT_l at station l

$(l = 1, \ldots, s)$ can be defined as

$$UT_l := \frac{T_{nl} - \sum_{k=1}^{n} U_{kl}}{\tau\, n\, w_l\, s_l}$$

where T_{nl} denotes the accumulated workload at station l for a sequence of length n. U_{kl} denotes the utility work incurred at station l by the unit in sequence position (stage) k. The labor capacity of station l can be computed by $\tau\, n\, w_l\, s_l$, where s_l denotes the number of units in station l. We assume, that station l has a fixed rate launch interval τ and that each unit in the station is processed by w_l operators. The labor utilization UT_l can be increased by reducing the labor capacity $\tau\, n\, w_l\, s_l$ or by decreasing the accumulated utility work $\sum_{k=1}^{n} U_{kl}$ of the sequence. Determining the labor capacity of each station l is part of the line balancing. The accumulated utility work $\sum_{k=1}^{n} U_{kl}$ depends on the sequence and the labor capacity. Thus, the utilization UT_l of each station can only be determined, if the line balance is known and a sequencing algorithm is available for a given order set. Therefore, sequencing does not only represent a short–term problem, but has to be considered in the context of line balancing, too. Obviously, the same sequencing procedure should be used to evaluate the line balance as to determine the order–sequence. This fact is not appropriately covered in recent research. Finally, with respect to material supply, a constant rate of component usage reduces the effort of material supply and leads to a smooth production in the underlying sub–assemblies.

We now introduce three objectives for the order–sequencing problem related to the evaluation criteria mentioned above. Leveling the deviation of the actual from the expected accumulated workload until stage k leads to uniform workload over the sequence. Due to that, the displacement of operators in their stations is reduced and utility work becomes less probable. Thus, we first consider the objective of leveling the workload in the different stations over the sequence. Let order i possess options $o \in O_i$. Then, the workload

$$t_{il} := \sum_{o \in O_i} p_{ol}$$

is required to assemble order i at station l, where p_{ol} denotes the workload caused by option o at station l. The order i assigned to sequence position k is denoted by unit i_k. The accumulated workload

$$T_{kl} := \sum_{q=1}^{k} t_{i_q l}$$

performed at station l until position (stage) k depends on the currently scheduled units $i_q\,(q = 1, \ldots, k)$. The average workload $\bar{t}_l$ performed at station l per unit is calculated by

$$\bar{t}_l := \frac{\sum_{i=1}^{n} t_{il}}{n},$$

such that the objective

$$\text{Min.} \sum_{k=1}^{n} \sum_{l=1}^{s} \left(k\bar{t}_l - T_{kl}\right)^2 \tag{2.1}$$

represents the total squared deviation of the accumulated workload T_{kl} from the expected accumulated workload $k\bar{t}_l$ of each station l and of each stage k.

The second objective to evaluate the quality of a sequence considers the minimization of total utility work. In order to determine the total utility work some details of the underlying assembly line are required. According to [16, 28] we assume a paced assembly line with a fixed rate launch interval τ and open stations with *upstream allowance time* t_l^u and *downstream allowance time* t_l^d. Concurrent work is assumed to be not allowed, which means that two options assigned to different stations cannot be assembled simultaneously. The length s_l of station l is indicated by the number of units assigned to the station at the same time. Furthermore, the number of operators w_l assigned to each unit at station l may differ from station to station. The arrival time a_{kl} and the departure time d_{kl} of the unit in position k at station l $(l = 1, \ldots, s)$ are given by

$$a_{kl} := \tau(k-1) + \tau \sum_{q=1}^{l-1} s_q$$

and

$$d_{kl} := a_{kl} + \tau s_l.$$

The earliest possible start time to process the unit of stage k at station l is equal to $a_{kl} - t_l^u$, i.e., the point in time when the unit of stage k reaches the upstream limit. The areas between station boundary and upstream limit as well as downstream limit are called *overlap areas* of the station. The station length enlarged by these overlap areas is termed working area. With respect to utility work, we assume that the latest possible finish time of the unit in position k at station l is equal to the time $d_{kl} + t_l^d$ when the unit in question reaches the downstream limit. The configuration of a station l including the working area as well as the corresponding overlap areas is depicted in Fig. 1.

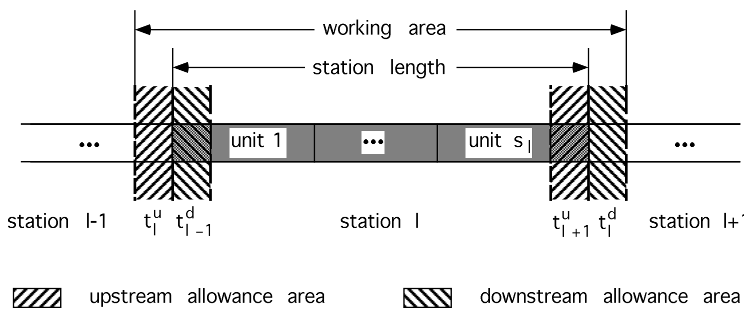

Figure 1: Configuration of station l

The part of the workload that cannot be performed within the working area is assumed to be utility work. Therefore, utility work occurs, iff the expected finish time $s_{kl} + t_{i_k l}/w_l$ exceeds the latest possible finish time $d_{kl} + t_l^d$ and is defined by

$$U_{kl} := \max\{0, s_{kl} + \frac{t_{i_k l}}{w_l} - (d_{kl} + t_l^d)\}.$$

Here, the start time s_{kl} of unit i_k at station l depends on the earliest possible start time $a_{kl} - t_l^u$ when unit i_k enters the working area, the finish time $f_{k,l-1}$ of unit i_k at station $l-1$ and the availability of operators at station l. In order to determine whether operators are available to process the unit in question, we assign the operators to the s_l units at station l following this policy: the $s_l w_l$ operators available at station l are divided into s_l teams of w_l operators, which are assigned to one unit each. Within the station preemption of workload is assumed to be allowed. Let i_k be the unit entering station l next. If the w_l operators of team s_l assigned to unit i_{k-s_l} have finished their workload, they continue with unit i_{k-s_l+1}. Simultaneously, team $s_l - 1$ shifts to unit i_{k-s_l+2}, and so on. Team 1 of station l is now without any unit and ready to process the unit i_k entering the station. In consequence, we consider the finish time $f_{k-s_l,l}$ of the unit i_{k-s_l} in order to determine the availability of operators to process the unit i_k that enters station l. Start time s_{kl} and finish time f_{kl} of unit i_k at station l can be calculated by

$$s_{kl} := \max\{a_{kl} - t_l^u, f_{k-s_l,l}, f_{k,l-1}\}$$

$$f_{kl} := \min\{s_{kl} + t_{i_k l}/w_l, d_{kl} + t_l^d\}$$

where we set $a_{11} := 0$, $f_{0l} := 0$ $(l = 1, \ldots, s)$, and $f_{k0} := 0$ $(k = -s_1 + 1, \ldots, n)$. The minimization of total utility work can be formulated as

$$\text{Min.} \sum_{k=1}^{n} \sum_{l=1}^{s} U_{kl}. \tag{2.2}$$

The start and finish times at station l are illustrated by Fig. 2, which depicts station l with a station length of $s_l = 2$ units. The horizontal bars represent the unit i_k of that stage k, given on the ordinate. The length of station l is represented by the gray area; dotted lines parallel to the station boundary mark the allowance limits of the stations as mentioned in the legend. The intersections of the boundary lines of station l and the bottom line of unit–bar i_k represent arrival time a_{kl} as well as departure time d_{kl} of unit i_k at station l; the start time of unit i_k at station l is determined by the finish time $f_{k-2,l}$ of unit i_{k-2} at station l. The handling of utility work is shown by unit i_{k-1}. That part of the workload exceeding the downstream limit $d_{k-1,l} + t_l^d$ is eliminated and is assumed to be performed by utility workers. The start time $s_{k+2,l}$ of unit i_{k+2} at station l is determined by the finish time $f_{k+2,l-1}$ of unit i_{k+2} at the previous station $l - 1$.

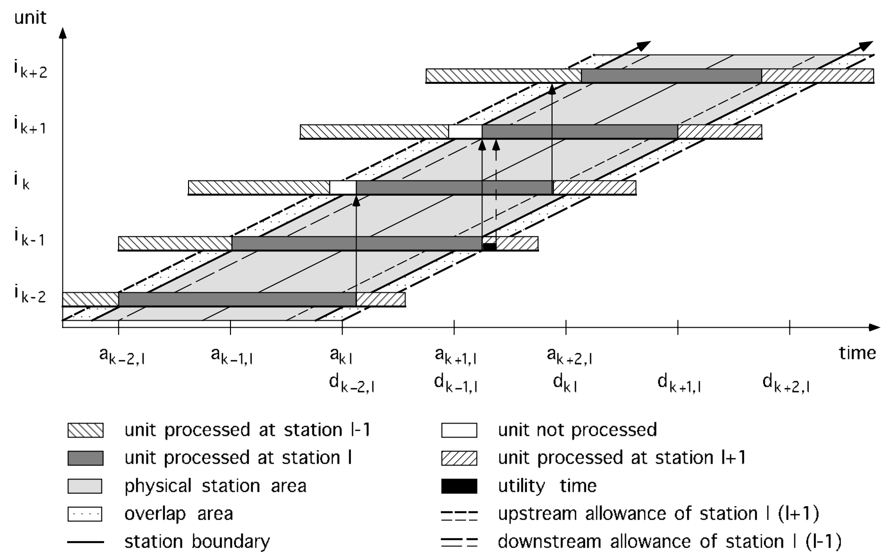

Figure 2: Start times of orders at station l

The third criteria to evaluate a sequence is a smooth component usage. In the case of automobile production a product option consists of several components, whereas each component belongs to exactly one product option. A uniform distribution of options over the sequence is considered to be equivalent to a uniform usage of components. Different options show different frequencies in the order set. Thus, a measure to make different options comparable with respect to their distribution over the sequence has to be defined. Therefore, we calculate the variation coefficient $\bar{s}_o/\bar{x}_o$ of the distances $k_2 - k_1$ between two consecutive orders i_{k_1} and i_{k_2} with option o. Mean distance $\bar{x}_o$ and standard deviation $\bar{s}_o$ can be calculated by

$$\bar{x}_o := \frac{1}{n_o} \sum_{j_o=2}^{n_o} k_{oj_o} - k_{o,j_o-1}$$

and

$$\bar{s}_o := \sqrt{\frac{1}{n_o - 1} \sum_{j_o=2}^{n_o} \left(k_{oj_o} - k_{o,j_o-1} - \bar{x}_o\right)^2}$$

respectively, where n_o denotes the number of orders with option o in the order set and k_{oj_o} the position of the j_o–th order with option o. Given $\bar{x}_o$ and $\bar{s}_o$, we can formulate the third objective, i.e. the minimization of the mean variation coefficient of all N_o options as

$$\text{Min.}\ \frac{1}{N_o}\sum_{o=1}^{N_o}\frac{\bar{s}_o}{\bar{x}_o}\ . \tag{2.3}$$

In this paper the main emphasis is on the leveling of workload as done by Sumichrast [26], whereas most sequencing algorithms known from literature consider one of the following objectives to determine a "good" order–sequence:

- Minimization of work overload [23, 30]
- Leveling occurrence of models types [17]
- Leveling component usage [20]

As mentioned above, the minimization of total work overload is equivalent to the minimization of total utility work. However, the minimization of total work overload does not necessarily lead to an even distribution of workload, which is desired for reasons of ergonomics and continuity of work, as well as for quality reasons. In order to avoid this drawback, the leveling of workload has to be considered for each station.

The real objective of leveling the occurrence of models over the sequence or leveling the usage of components is to achieve an even material supply. Implicitly, these approaches intend to level the workload and to minimize the variation in displacement of the operators. In the case of build–to–order production, leveling of models is impossible, since the model demand is equal to one.

In practical applications the determination of an order–sequence is often done by leveling the usage of components. Hereby, components are classified with respect to their influence on material supply and the variation in workload on the line. We distinguish between:

- basic components
- optional components
- order–dependent components

Basic components are required for each unit in an identical manner and lead to a constant workload and a constant effort of material supply at the stations. Optional components are used only in some units, depending on the order configuration, and cause additional workload and supplementary effort of material handling. Components required for each unit in an order–dependent configuration differ in workload, but do not cause an additional effort of material handling. Thus, only optional components need to be considered, if the focus is on the additional effort of material handling.

Different components lead to different workloads at the stations, such that the leveling of components does not necessarily lead to a uniform workload. Furthermore, different components, that incur workloads at the same station, may result in a high utility work at that station. Thus, high utility work is possible, even if each component shows a uniform distribution over the sequence. Due to the relationship between the usage of a component and the workload p_{ol} at a station l, we expect

1. an even distribution of components causing intensive workload
2. an even distribution of a modest total utility work over the time horizon

in leveling the workload.

The above considerations indicate that leveling the workload is a reasonable objective to order–sequencing on mixed–model assembly lines. Using (2.1) we present the workload leveling problem (WLP) for order–sequencing on a mixed model assembly line. The objective of WLP is to minimize the mean squared deviation of the actual accumulated workload from the expected accumulated workload of each stage at each station. The decision variable x_{ik} indicates whether an order i has already been scheduled at position $k_i \leq k$ of the sequence and is given by

$$x_{ik} = \begin{cases} 1, & \text{if order } i \text{ is scheduled in position } k_i \leq k \\ 0, & \text{otherwise} \end{cases}$$

Thus the WLP can be formulated as

$$\text{Min.} \sum_{k=1}^{n} \sum_{l=1}^{s} (k\,\bar{t}_l - x_{i_k k} t_{i_k l})^2$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_{ik} = k \qquad (k = 1, \ldots, n) \qquad (2.4)$$

$$0 \le x_{i,k-1} \le x_{ik} \le 1 \qquad (i = 1, \ldots, n; k = 2, \ldots, n) \qquad (2.5)$$

$$x_{ik} \in \{0, 1\} \qquad (i = 1, \ldots, n; k = 1, \ldots, n) \qquad (2.6)$$

Restriction (2.4) and (2.6) ensure that there are exactly k units scheduled until stage k and restriction (2.5) guarantees that each order i is only scheduled once.

3 Sequencing algorithm

In this section we introduce an algorithm for the order–sequencing problem WLP on mixed–model assembly lines. We describe the basic ideas of the approach and give a formal representation of the basic workload leveling algorithm. Then, we illustrate the application of the algorithm by an example. Next, we discuss problems that arise in the practical application of sequencing algorithms in automobile production. With regard to that case, we suggest an extended workload leveling algorithm computing a batch–sequence.

The algorithm to be proposed is an iterative greedy heuristic. The two characteristic features are the leveling of workload and the determination of an order–sequence. Recent research in the field of sequencing on mixed–model assembly lines emphasizes the leveling of the production rate of outputs and the determination of model–sequences [13]. A workload based sequencing algorithm was proposed by Sumichrast [26], who used, as the rate to level, the accumulated workload T_{nl} of the order set at station l divided by the total workload $\sum_{l=1}^{s} T_{nl}$ of the order set at all stations. In contrast to Sumichrast, we consider the average workload $\bar{t}_l$ at station l over the sequence as the expected workload of station l at each stage. Thus, the accumulated workload expected to be performed until stage k at station l is given by $k\,\bar{t}_l$. The algorithm consecutively schedules order $i_1, \ldots, i_n$ where at each stage k the eligible order i^* with minimal priority value v_{i^*k} is scheduled. The priority value v_{ik} of each eligible order $i \in E$ at stage k is given by the minimal squared deviation of the expected accumulated workload $k\bar{t}_l$ from the actual accumulated workload $T_{k-1,l} + t_{il}$. If there is more than one eligible order i^* with minimal priority value v_{i^*k}, the order with smallest index is chosen. A representation in pseudo code of the algorithm, approximately solving the WLP with time complexity $O(n^2 s)$, is given in the following:

Algorithm [AOS]

Step 1: Initialization

$k := 1$

$E := \{1, ..., n\}$

For all $l \in \{1, \ldots, s\}$: $T_{kl} := 0$ and $\bar{t}_l := \frac{\sum_{i=1}^{n} t_{il}}{n}$

Step 2: Sequencing the orders

While $E \neq \emptyset$ Do

For all $i \in E$: $v_{ik} := \sum_{l=1}^{s} (k\bar{t}_l - T_{k-1,l} - t_{il})^2$

$i_k := \min\{i \in E | v_{ik} = \min_{j \in E} v_{jk}\}$

$E := E \setminus \{i_k\}$

For all $l \in \{1, \ldots, s\}$: $T_{kl} := T_{k-1,l} + t_{i_k l}$

$k := k + 1$

End (While)

In order to illustrate the proceeding of the AOS–algorithm, we consider an example with five stations and six orders. Again, each order consists of several options and each option may incur workload at several stations. Therefore, the set of options O_i determines the workload t_{il} of order i at station l. Table 1 shows the workloads t_{il} of orders $i = 1, \ldots, 6$ at stations $l_1, \ldots, l_5$. These quantities lead to a constant average workload of $\bar{t}_l = 2.7$ time units at each station l, depicted in the bottom line of Table 1. Assuming a fixed–rate launch interval τ of three minutes, we obtain an expected labor utilization of 90% at each station l.

order i \ station l	1	2	3	4	5
1	1.4	4.2	1.3	4.3	1.7
2	1.4	1.8	1.3	3.1	1.7
3	5.2	2.4	4.9	3.1	4.5
4	3.4	1.8	3.7	3.1	3.3
5	1.4	4.2	1.3	1.9	1.7
6	3.4	1.8	3.7	0.7	3.3
$\bar{t}_l$	2.7	2.7	2.7	2.7	2.7

Table 1: Allocation of workload t_{il}

Applying the AOS–algorithm, we compute the priority values v_{i1} of each order i. The priority values v_{i1} $(i = 1, \ldots, 6)$ are shown in the second row of Table 2.

stage k \order i	1	2	3	4	5	6
1	9.46	5.62	14.58	2.82	7.54	6.66
2	5.04	4.56	28.32		1.20	13.2
3	14.66	8.90	8.90			6.26
4	4.80	8.16	22.96			
5		14.58	5.62			
6		0				

Table 2: Priority value v_{ik} of orders

For example, with $T_{0l} := 0$ for each station l, the priority value v_{11} of order $i = 1$ at the stage $k = 1$ can be calculated by

$$(2.7 - 1.4)^2 + (2.7 - 4.2)^2 + (2.7 - 1.3)^2 + (2.7 - 4.3)^2 + (2.7 - 1.7)^2 = 9.46$$

Since order $i = 4$ has the minimal priority value v_{i^*1}, we set $i_1 = 4$. At every further stage k, the priority values v_{ik} of units, that are not yet scheduled, are calculated. For example, the priority value v_{12} of order $i = 1$ at stage $k = 2$ can be computed by

$$(5.4 - 3.4 - 1.4)^2 + (5.4 - 1.8 - 4.2)^2 + (5.4 - 3.7 - 1.3)^2 + (5.4 - 3.1 - 4.3)^2 + (5.4 - 3.3 - 1.7)^2 = 5.04$$

Continuing with the AOS–algorithm, we obtain the order–sequence $(4, 5, 6, 1, 3, 2)$ with an objective function value equal to 20.7, whereas an optimal sequence is $(4, 1, 6, 5, 3, 2)$ with an objective function value equal to 18.78.

With regard to automobile production, an algorithm for the determination of an order–sequence has to consider, additionally, the structure of the production system and the constraints of material supply. Before we investigate sequencing policies in context of automobile production, we give a short overview of the organization of the production system considered.

In automobile production the three sub–systems *body shop*, *paint shop* and *assembly shop* are distinguished. Each sub–system has different production and scheduling restrictions. Body shop as well as assembly shop are organized as a mixed–model–system, whereas the paint shop is typically a multi–model line. With respect to sequencing, the "models" are defined by specific shop–related options, which differ from shop to shop. For instance, in the body shop the number of doors and the sunroof may determine the model type of an order, whereas in the paint shop models are defined by the color, in general. The basic idea of the new approach is to define models in the assembly shop with respect to the options of an order. Since it is unlikely that two cars possess the same set of options, a model demand equal to one has to be assumed in the assembly shop. Since in different shops different options are considered to define model types, the model types differ from shop to shop. Therefore, an optimal sequence for the paint shop in general does not correspond to an optimal sequence for the other shops.

In what follows, we investigate the problem "how to provide the assembly shop with a good order–sequence". Here, the procedure of order–sequencing for the assembly shop depends on:

- the sequencing policy
- the quality of the painting process with respect to sequencing
- the performance of the sorting buffer providing the assembly shop

Sequencing policies differ in the point in time at which the sequence is determined, the location in the production system where the sequence is built physically, the orders eligible for each position of the sequence, and other technological constraints. The quality of the paint process with respect to the sequencing problem depends on the probability of rework and the length of rework cycles. The order–sequence can be changed in *sorting buffers* between the shops. The performance of a sorting buffer depends on its size, the type of accessing stored units and the velocity of providing an expected unit. We now investigate three sequencing policies with respect to the resulting order–sequences in the assembly shop:

1. resorting a batch–sequence disturbed in the painting process
2. scheduling a buffer–sequence on the basis of the units available in the sorting buffer
3. scheduling an "optimal" sequence in the assembly shop, assuming that each unit can be provided by the sorting buffer in time

Applying the first policy, we suppose that the sequences in the paint shop and in the assembly shop are identical. In determining the batch–sequence, we have to consider the size of color batches in the paint shop as well as the workload of orders at the assembly stations. The advantage of this policy is a long–term look ahead of the order–sequence to assemble. In the paint shop the sequence of units is disturbed by rework cycles. The ability to resort all units disturbed in the paint process depends on the performance of the sorting buffer. The consideration of the color batch restriction, generally, results in a reduced quality of the sequence with respect to the leveling of workload.

Providing a *buffer–sequence* according to the second policy results in a short look ahead of the orders entering the assembly shop next. That is unfavorable with regard to material supply.

The third policy entails a long look ahead of the sequence in the assembly shop. However, the sequence of units leaving the paint shop is not deterministic due to the possibility of rework in the paint shop. Therefore, this policy is of more theoretical significance but it can be used as a reference for the quality of the sequences according to the policies 1 and 2, respectively.

In order to determine the batch–sequence of policy 1, we propose an extended workload leveling algorithm termed as assembly line batch sequencing algorithm (ABS). For the determination of order–sequences according to policy 2 and 3 we use the AOS–algorithm. Here, we can apply the AOS–algorithm for policy 2, if we consider the orders available in the sorting buffer to be the set of eligible orders at each stage.

The basic idea of the ABS–algorithm is to determine a sequence of batches and then to schedule orders within each batch with respect to the leveling of workload. In doing so, we first choose the color c^* of the orders of the next batch. Color c^* is the color with the maximum positive deviation of the actual from the expected amount of scheduled orders with color c. If there is more than one color c^* with that property, the color with smallest index is chosen. The actual size of the color batch AB is initially set to the minimum of batch size B and the amount of eligible orders with color c^*. Then, we schedule AB orders of the current color c^* according to the AOS–algorithm where c_i denotes the color of oder i. This procedure is repeated until all orders are scheduled.

A formal representation of the ABS–algorithm with time complexity $O(n^2 s)$ is given as follows:

Algorithm [ABS]

Step 1: Initialization

$k := 1$; $AB := 0$

$E := \{1, ..., n\}$

For all $c \in \{1, \ldots, N_c\}$ Do

$E_c := \{i | c_i = c\}$

$AC_c := \frac{|E_c|}{n}$

$S_c := 0$

For all $l \in \{1, \ldots, s\}$: $T_{0l} := 0$ and $\bar{t}_l := \sum_{i=1}^{n} t_{il}/n$

Step 2: Sequencing the orders

While $E \neq \emptyset$

If $AB = 0$ Then Do

$$\begin{aligned} c^* &:= \min\{c | E_c \neq \emptyset \wedge (k\, AC_c - S_c) \\ &= \max_{\gamma \in \{1,\ldots,N_c\}} (kAC_\gamma - S_\gamma)\} \end{aligned}$$

$AB := \min(B, |E_{c^*}|)$

For all $i \in E_{c^*}$: $v_{ik} := \sum_{l=1}^{s} (k\bar{t}_l - T_{k-1,l} - t_{il})^2$

$i_k := \min\{i \in E_{c^*} | v_{ik} = \min_{j \in E_{c^*}} v_{jk}\}$

$E := E \setminus \{i_k\}$

$E_{c^*} := E_{c^*} \setminus \{i_k\}$

$S_{c^*} := S_{c^*} + 1$

For all $l \in \{1, \ldots, s\}$: $T_{kl} := T_{k-1,l} + t_{i_k l}$

$k := k + 1$; $AB := AB - 1$

End (While)

In general, the ABS–algorithm can be used if the workload t_{il} of each order i at each station l is given and if a model type can be assigned to each order.

4 Experimental performance analysis

We briefly report on an experimental analysis of the algorithms introduced in Section 3. First, details of the underlying assembly line are presented. Then, the AOS–algorithm is compared with the two workload leveling heuristics of [13] and [26]. Finally, an evaluation of the sequencing policies proposed in the previous section is given and further experiments are briefly discussed. For a detailed view, we refer to Engel [8].

The experimental analysis was part of a recent research, initiated by IBM Informationssysteme GmbH, Germany, concerning production planning of mixed–model assembly lines in automobile production. The described sequencing policies have been used for the evaluation of mixed–model assembly lines in automobile production. Sequences are evaluated by the leveling of workload WL, the total amount of utility work U and the leveling of options OL according to the objectives (2.1), (2.2) and (2.3), respectively.

With respect to the characteristics of automobile production, we consider a paced assembly line with $s = 30$ stations, where the station lengths s_l of three specific stations are equal to 6, 3, and 4 units. The remaining stations obtain a station length of $s_l = 2$ units. $w_l = 2$ operators are allocated to each unit at each station l $(l = 1, \ldots, s)$. The fixed launch rate τ is given by three minutes. Upstream allowance time t_l^u and downstream allowance time t_l^d of each station are uniformly set to 50% of the fixed launch rate τ. Frequencies of $N_o = 20$ options and workload p_{ol} incurred by option o in station l are given similar to those used in practical applications.

The AOS–algorithm can be applied to the case of order–sequencing as well as to the case of model–sequencing. For the case of order–sequencing, we generated 100 sets containing 100 orders where each order possesses an individual configuration of options. The set of orders was generated by a random procedure, so that the set of orders contains the fixed frequencies N_o for each option o. The order–sequence determined by the AOS–algorithm was compared with a random sequence and with the sequence determined by the *Time Spread* heuristic devised in [26].

For the case of model–sequencing, we generated 10 sets of 100 orders, where 5 or 10 model types are distinguished and all orders of a model type possess the same configuration of options. The model–sequence determined by AOS was compared with a random sequence and, additionally, with an algorithm described by Kubiak [13]. Table 3 and 4 show the mean objectives WL, U, and OL over all sequences of the test set.

Considering Table 3, we see that the AOS–algorithm is markedly superior to the Time Spread heuristic with respect to all objectives. The poor performance of the Time Spread heuristic can be explained by its scheduling criteria. The Time Spread heuristic prefers orders that have a small total workload and show a very even distribution of workload over the stations. With regard to objective (2.1) of workload leveling WL this leads to a high deviation of the accumulated workload from the expected workload for stages in the middle of the sequence. Since the Time Spread–heuristic, obviously, does not lead to a leveling of workload, it seems to be inadequate for the case of order–sequencing.

sequence\objective	WL	U	OL
Random	300147.19	443.19	0.74
Time Spread	5667728.76	551.09	0.80
AOS	15846.04	263.31	0.54

Table 3: Comparison between Random,Time Spread, and AOS

Considering the case of model–sequencing the AOS–algorithm outperforms the algorithm of Kubiak with respect to the objectives WL and U, whereas the latter one shows a better performance in OL (see Table 4). Kubiak determines the average workload for each station and for each model. At each stage the model is scheduled, which obtains the maximum deviation between expected and accumulated workload at the previous stage.

In order to evaluate the three sequencing policies for the assembly shop, mentioned in Section 3, we generated 10 sets containing 300 different orders. A specific color is assigned to each order. The number of colors available was given by $N_c = 10$. With respect to sequencing policy 1, we generate a *batch–sequence* with

sequence\objective	WL	U	OL
Random	274355.18	641.41	0.71
Kubiak	42836.45	586.07	0.30
AOS	32766.75	486.77	0.33

Table 4: Comparison between Random, Kubiak, and AOS

color batch size $B = 5$ by applying the ABS–algorithm.

The determined batch–sequence is assumed to be the input sequence of the paint shop. With no rework cycles in the paint shop, this sequence theoretically passes to the assembly shop. In general, the input sequence of the paint shop is disturbed by rework cycles. Therefore, the batch–sequence was randomly disturbed with a disturb factor of 10%, which means that on the average 90% of the units leave the paint shop without running through a rework cycle. The length of the rework cycle was uniformly chosen between 10 and 60 units per delay. The disturbed batch–sequence is called *disturbed sequence*. Since a long look ahead of the assembly sequence is desired, we seek to resort the disturbed sequence to the original batch–sequence. The ability to resort the batch–sequence depends on size and accessibility of the sorting buffer. In the considered problems, a buffer size of 40 units and random buffer access are assumed. The result of resorting the disturbed sequence is denoted by *resorted sequence*.

Sequencing policy 2 provides a sequence on the set of units actually leaving the paint shop. Considering the units of the disturbed sequence that are in the sorting buffer, a *buffer–sequence* can be computed with the AOS–algorithm. Finally, the AOS–sequence is generated on basis of the total order set. The AOS–sequence is only of theoretical significance because, in general, this sequence can only be provided to the assembly shop, if the buffer is of size n. Table 5 shows the objectives of the different sequences. Loosely speaking, the batch–sequence is not as good as the buffer–sequence or the AOS–sequence, because the consideration of batches reduces the set of eligible jobs at each stage. Due to a relative small buffer size the resorted sequence is generally not as good as the batch–sequence. Obviously, the AOS–sequence shows the best performance, but with respect to practical applications the buffer sequence outperforms each available sequence.

policy\objective	WL	U	OL
Batch–Sequence	114501.06	934.19	0.69
Disturbed Sequence	267942.29	1018.87	0.72
Resorted Sequence	129065.68	959.59	0.70
Buffer–Sequence	66324.83	732.38	0.60
AOS–Sequence	56247.26	697.56	0.58

Table 5: Sequencing policies

Further tests to determine the performance of AOS with regard to variations in the frequency of options, length of the sequence, size of the color batch or length of the allowance limits have been done. For further details we refer to Engel [8].

5 Conclusions

In this paper we discussed the performance analysis of mixed–model assembly lines with given line balance. We presented three objectives for the sequencing problem. We motivated a workload leveling approach and introduced an integer programming formulation for the workload leveling problem WLP. We devised two polynomial heuristics AOS and ABS with time complexity $O(n^2 s)$ for the WLP. Thereby, AOS provides an order–sequence and ABS computes batch–sequences of orders. We proposed three sequencing policies for practical applications in automobile production. In an experimental performance analysis we compared the AOS–algorithm with two workload leveling algorithms for order– and model–sequencing, respectively. The AOS–algorithm outperforms the heuristic of [26] for the problem of order–sequencing as well as the heuristic of [13] for the problem of model–sequencing. Finally, we evaluated the proposed sequencing–policies of automobile production.

Important areas of further research are leveling the variation in workload at the stations over the sequence as well as resource constraints of options over time, which is important for practical applications.

References

[1] Bard, J.F., Dar–El, E.M., and Shtub, A. (1992), An analytic framework for sequencing mixed model assembly lines, *International Journal of Production Research*, Vol. 30, pp. 35–48

[2] Bard, J.F., Shtub, A., and Joshi, S.B. (1994), Sequencing mixed–model assembly lines to level parts usage and minimize line length, *International Journal of Production Research*, Vol. 32, pp. 2431–2454

[3] Bolat, A. (1994), Sequencing jobs on an automobile assembly line: objectives and procedures, *International Journal of Production Research*, Vol. 32, pp. 1219–1236

[4] Dar–El, E.M. (1978), Mixed–model assembly line sequencing problems, *OMEGA*, Vol. 6, pp. 313–323

[5] Dar–El, E.M. and Cother, R.F. (1975), Assembly lines sequencing for model mix, *International Journal of Production Research*, Vol. 13, pp. 463–477

[6] Decker, M. (1993), *Variantenfließfertigung*, Schriften zur quantitativen Betriebswirtschaftslehre, Vol. 7, Physica, Heidelberg

[7] Domschke, W., Scholl, A., and Voß S. (1993), *Produktionsplanung – Ablauforganisatorische Aspekte*, Springer, Berlin

[8] Engel, C. (1997), Belastungsnivellierung in der Variantenfließfertigung, *Diploma Thesis*, Institut für Wirtschaftstheorie und Operations Research, University of Karlsruhe

[9] Görke, M. and Lentes, H.–P. (1981), Modellfolgebestimmung bei gemischter Produktfertigung, *wt – Zeitschrift für industrielle Fertigung*, Vol. 71, pp. 153–160

[10] Inman, R.R. and Bulfin, R.L. (1992), Quick and dirty sequencing for mixed–model multi–level JIT systems, *International Journal of Production Research*, Vol. 30, pp. 2011–2018

[11] Kim, Y.K, Hyun, C.J. and Kim, Y. (1996), Sequencing in mixed–model assembly lines: A genetic algorithm approach, *Computers and Operations Research*, Vol. 23, pp. 1131–1145

[12] Köther, R. (1986), Verfahren zur Verringerung von Modell–Mix–Verlusten in Fließmontagen, *IPA–IAO Forschung und Praxis*, Vol. 93, Springer, Berlin

[13] Kubiak, W. (1993), Minimizing variation of production rates in just–in–time systems: A survey, *European Journal of Operational Research*, Vol. 66, pp. 259–271

[14] Kubiak, W. and Sethi, S. (1991), A note on level schedules for mixed–model assembly lines in just–in–time production systems, *Management Science*, Vol. 37, pp. 121–122

[15] McCormick, S.T., Pinedo, M.L., Shenker, S., and Wolf, B. (1989), Sequencing in an assembly line with blocking to minimize cycle time, *Operations Research*, Vol. 37, pp. 925–935

[16] Macaskill, J.L. (1973), Computer simulation for mixed–model production lines, *Management Science*, Vol. 20, pp. 341–348

[17] Miltenburg, J. (1989), Level schedules for mixed–model assembly lines in just–in–time production systems, *Management Science*, Vol. 35, pp. 192–207

[18] Miltenburg, J. and Sinnamon, G. (1989), Scheduling mixed–model multi–level just–in–time production systems, *International Journal of Production Research*, Vol. 27, pp. 1487–1509

[19] Miltenburg, J. and Sinnamon, G. (1992), Algorithms for scheduling multi–level just–in–time production systems, *IIE Transactions*, Vol. 24, pp. 121–130

[20] Monden, Y. (1983), *Toyota Production System*, Industrial Engineering and Management Press, Atlanta

[21] Okamura, K. and Yamashina, H. (1979), A heuristic algorithm for the assembly line model–mix sequencing problem to minimize the risk of stopping the conveyor, *International Journal of Production Research*, Vol. 17, pp. 233–247

[22] Rachamadugu, R. and Yano, C.A. (1994), Analytical tool for assembly line design and sequencing, *IIE Transactions*, Vol. 26, pp. 2–11

[23] Scholl, A. (1995), *Balancing and Sequencing of Assembly Lines*, Physica, Heidelberg

[24] Steiner, G. and Yeomans, S. (1993), Level schedules for mixed–model just–in–time processes, *Management Science*, Vol. 39, pp. 728–735

[25] Steiner, G. and Yeomans, S. (1996), Optimal level schedules in mixed–model, multi–level JIT assembly systems with pegging, *European Journal of Operational Research*, Vol. 95, pp. 38–52

[26] Sumichrast, R.T., Russell, R.S., and Taylor, B.W. (1992), A comparative analysis of sequencing procedures for mixed–model assembly lines in a just–in–time production system, *International Journal of Production Research*, Vol. 30, pp. 199–214

[27] Sumichrast, R.T. and Clayton, E.R. (1996), Evaluating sequences for paced, mixed–model assembly lines with JIT component fabrication, *International Journal of Production Research*, Vol. 34, pp. 3125–3143

[28] Thomopoulos, N.T. (1967), Line balancing–sequencing for mixed–model assembly, *Management Science*, Vol. 14, pp. 59–75

[29] Tsai, L.H. (1995), Mixed–model sequencing to minimize utility work and the risk of conveyor stoppage, *Management Science*, Vol. 41, pp. 485–495

[30] Yano, C.A. and Rachamadugu, R. (1991), Sequencing to minimize work overload in assembly lines with product options, *Management Science*, Vol. 37, pp. 572–586