



Integer programming models versus advanced planning business software for a multi-level mixed-model assembly line problem

Tobias Kreiter¹ · Ulrich Pferschy¹ 

Published online: 21 August 2019
© The Author(s) 2019

Abstract

Many manufactures are shifting from classical production environments with large batch sizes towards mixed-model assembly lines due to increasing product variations and highly individual customer requests. However, an assembly line should still be run with constant speed and cycle time. Clearly, the consecutive production of different models will cause a highly unbalanced temporal distribution of workload. This can be avoided by moving some assembly steps to pre-levels thus smoothing out the utilization of the main line. In the resulting multi-level assembly line the sequencing decision on the main line has to take into account the balancing of workload for all pre-levels. Otherwise, the modules or parts delivered from the pre-levels would cause congestion of the main line. One planning strategy aims at mixing the models on the main line to avoid blocks of identical units. In this contribution we compare two different realizations for this approach. On one hand we present a mixed-integer programming model (MIP), strengthen it by adding valid inequalities and enrich it with a number of relevant practical extensions. Also the actual objective of explicitly balancing pre-level workloads is considered. On the other hand, we illustrate how this strategy could be realized in an advanced planning system linked to an enterprise resource planning system, namely SAP APO. Finally, we perform a computational study to investigate the possibilities and limitations of MIP models and the realization in SAP APO. The experiments rely on a real-world production planning problem of a company producing engines and gearboxes.

Keywords Multi-level production system · Load balancing · Model-mix planning · Mixed-integer programming · SAP APO

This research was performed in cooperation with *scc EDV-Beratung AG*, Vienna, Austria, and partially financed by the Austrian Research Promotion Agency (FFG).

✉ Ulrich Pferschy
ulrich.pferschy@uni-graz.at

Extended author information available on the last page of the article

1 Introduction

Product variety increased drastically over the last decade in a wide range of industries through a higher degree of product customization and the renunciation of standardized products (Meyr 2004; Pil and Holweg 2004). As an example consider the well-known car manufacturer *AUDI* who produced only 4 different car models and derivatives in 1996, 14 in 2004, but 53 in the year 2014 (Wörner 2014). Especially industries with a significant share of labor costs tend to provide increasing degrees of customization. This is particularly true for manufacturers located in high-wage countries, who cannot compete with the unit-costs of manufacturers in low-wage countries for mass-produced items. But by employing their capabilities to provide higher customized products, they can make use of the higher willingness-to-pay of consumers and charge higher prices for those customized products.

Thus, manufacturers face the challenge of providing a growing variety of each product at a low cost. The traditional production schemes with large batch sizes, designed for cost-efficient mass production of a few standardized products, are transformed by decreasing lot sizes. The end point of this development will be mixed-model assembly lines, where different products can be jointly manufactured in intermixed product sequences on the same line, see e.g. Boysen et al. (2009b). In such a scheme lot size is decreased to one and consequently the production plan has to consider every single unit of each product individually.

While this development runs against the principle of economy of scale which led to a tremendous decrease in production cost in the past, it allows for a fulfillment of individual customer requests, instead of forcing the customer to pick from a small number of options. This makes sense in particular for products serving customers with advanced technical insights and an ability to express their individually chosen technical specifications, but also for products with a strong design aspect where customers like to choose individual patterns of shape, colour and other features. Customers are often willing to pay a considerable premium for these individually configured products, but of course the willingness of extra payment is not unlimited. Thus, it remains as a demanding challenge for production companies to switch their organization from large batch sizes to lot size one without incurring excessive extra cost.

It is the classical task of assembly line production planning to balance the workload assigned to each work station and thus allow a constant speed and cycle time of the assembly line. This allows the efficient assembling of products that are synchronously conveyed through all stations of the line. As long as products are produced in large or medium scale lot sizes, the workload is stable for the specific time frame of the lot and the change in workload in between the lots could be managed rather simply by changing the speed of the assembly line or by adapting the size of the workforce.

However, things become more difficult for small lot sizes. Different products, or even different variants of the same basic product, naturally differ in their production requirements and in their induced workload, see e.g. Öztürk et al. (2013). Substantial reductions in setup times and cost as well as the application of flexible workers and machinery are natural prerequisites for executing these smaller lot sizes as described in Miltenburg and Sinnamon (1989). The goal of an efficient, constant speed assembly line will be impossible to reach if a sequence of smaller batches, each consisting of

products with very different requirements, is to be produced on one line. Adapting workforce or line speed after each small batch are both infeasible in practice.

To overcome this obstacle to a flexible utilization of the assembly line and to reach a task allocation where the work intensity of all products lies within a limited range, manufacturers with a heterogeneous product portfolio can apply the following strategy: Firstly, for each product some assembly steps are outsourced to pre-levels to permit a more or less uniform remaining workload on the main line, while the variances in workload are shifted to the pre-level assembly stations. These can be organized in one or several levels. Secondly, products have to be scheduled on the main line in such a way that the implied schedule on the pre-levels smooth out and balance out differences in pre-level workload instead of causing bottlenecks as discussed in Pereira and Vilà (2015). Note that in modern lean manufacturing systems the modules or parts produced on pre-assembly lines should not pile up at the transition to the main line since usually only very small intermediate buffers exist (if any).

In particular, a thorough *mixing of different models* produced in sequence will be sought, since this is much more likely to balance out the pre-level work intensities. An alternative approach would vary the time gap between consecutive products as described in Tong et al. (2013). However, this point of view often runs into organizational or technical limitations and thus will not be followed in this paper.

Our contribution focuses on the short-term sequencing decisions, where all tasks are already allocated to a particular shift and assembly line. The related mid- and long-term decisions of how to allocate assembly tasks among different stations or production levels will be assumed to be given by a previous decision step, see e.g. Scholl and Becker (2006), Sivasankaran and Shahabudeen (2014), Dörmer et al. (2015) and the references therein.

The current research is inspired by the real-world case of a Central-European manufacturer who produces highly heterogeneous sets of engines and gearboxes, some of them in relatively small numbers, but individual configurations. In order to increase flexibility in production and allow rapid response to urgent customer demands or last-minute changes of customer orders, a mixed-model multi-level assembly line system was introduced. Mixed-model sequencing has received increasing attention in the last decade as illustrated by more than 200 citations of the seminal survey paper (Boysen et al. 2009b) from 2009. Very recent publications in the field of mixed-model literature focus i.e. on resequencing decisions in the very short term (Taube and Minner 2018) and on special constraints like on machine idle times (Abdul Nazar and Pillai 2018).

Nowadays, all major manufacturing companies employ Enterprise Resource Planning (ERP) software tools, usually embedded into a general-purpose software covering all aspects of the company. It is a widely voiced wish to run also sophisticated production planning tasks within the umbrella of such an IT system. Introducing tailor-made optimization routines into the IT architecture was described e.g. in Garcia-Sabater et al. (2012) for the same type of industry, but poses several disadvantages. These are licence costs, limited maintenance, high-degree of already existing IT know-how for ERP-software within the introducing organisation and lack of know-how about tailor-made solutions, high adaptation costs incurred by system changes, etc. Therefore, major software suppliers offer various types of advanced planning systems to

fulfill this demand. The software system SAP ECC¹ and its advanced planning system SAP APO² can be seen as an industry standard and were also used as ERP-software in the company of the underlying real-world application case to organize the production planning tasks.

The main contributions given in this paper can be summarized as follows:

1. We bring attention to the increasingly important aspect of assembling units of different models in one sequence of a production line (in contrary to classical batch production) and point out the difficulties of attaining a balanced utilization at all production levels, in particular under the consideration of pre-level steps, such as sub-assembly, kitting and intralogistics.
2. We represent the resulting mixed-model assembly line scheduling problem by means of a mixed-integer program (MIP) in Sect. 3 and strengthen it by adding valid inequalities. A number of relevant practical extensions, such as due dates, release dates and limited buffers at the end of the line are discussed and realized in the MIP model (Sect. 4).
3. We describe how these aspects could be achieved (directly or indirectly) in the environment offered by the industry standard ERP-system SAP and its planning system APO (Sect. 5). While plentiful MIP models for various complicated production scheduling tasks are given in the literature, in practice companies often prefer to have the production planning done within their existing ERP-system. Thus, we believe that it is an important and new step to consider how complex realistic constraints could be modelled by a generic, but highly configurable software system such as SAP APO.
4. We present computational experiments based on a real-world case study and compare the possibilities and limitations of modelling (Sect. 5) as well as the computational performance for the MIP models compared to the realizations in SAP APO (Sect. 6). This should provide valuable information for the deployment decisions in other mixed-model planning problems. We are not aware of any previously existing computational comparison between tailor-made MIP models and SAP implementations for mixed-model production scheduling problems with real-world data.

The bottom line of our results can be described as follows: SAP APO permits the computation of mixed-model production sequences with maximal sum of gaps between units of the same model within a very short computation time. Realizing the same objective with a MIP model requires considerably higher running times and must be seen as inferior to the SAP APO approach. However, it turns out that the actual goal of balancing workload over all pre-levels of a multi-level mixed-model assembly line is missed by a fairly large margin by this indirect modelling approach. On the other hand, implementing this balancing goal explicitly in a MIP model yields a computationally more tractable approach which delivers highly satisfactory results. Additional practical constraints occurring in a real-world application can be fully integrated in

¹ ERP Central Component. For more details see: www.sap.com/product/enterprise-management/erp.html.

² Advanced Planning and Optimization. For more details see: www.sap.com/product/scm/advanced-planning-optimization.html.

the MIP model, while SAP APO allows only a partial representation of these extra requirements.

2 Detailed problem description

2.1 Background of the case study

Producing on mixed-model assembly lines implies a major production paradigm shift from large and medium-scale batch production towards lot size one production. This leads to a situation where many different models of lot size one are assembled on the same production line. However, different models induce a different workload for the employed workers and the corresponding stations of the line. To guarantee a smooth workload at all levels of production even in such a mixed-model assembly line setting for heterogeneous products sophisticated production planning and sequencing is required. The major trigger for the production paradigm shift towards smaller lot sizes is the higher degree of product customization and the renunciation of standardized products (Meyr 2004). Product customization is a recent trend in many industries. Especially industries with a significant share of labor costs tend to provide increasing degrees of customization. This is particularly true for manufacturers located in high-wage countries, which cannot compete with the unit-costs of manufacturers in low-wage countries for standardized high-volume products. But by employing their capabilities to provide heavily customized products, they can make use of the higher willingness-to-pay of consumers and charge higher prices for those tailored products.

Producing many different products on one main assembly line means that products with a different level of induced workload have to be considered in planning. Such a balancing of an assembly line has been the daily staple of production engineers over many decades. As long as those products are produced in large or medium scale lot sizes, the workload is stable for the specific time frame of the lot and the change in workload between the lots could be managed by changing the speed of the assembly line or by adapting the size of the workforce. However, things become clearly more difficult for decreasing lot sizes. Dealing with a heterogeneous workload for mixed-model assembly lines, where each item is considered individually and the assembled product and therefore the induced workload could change after each position in the production sequence, is a much harder exercise. The large-scale strategies of adapting workforce after each product or changing the speed of the line after each product are both infeasible. There are basically two approaches to deal with that issue for intermixed production sequences: First, differing the time gap between the consecutive products, see Tong et al. (2013), or, second, producing in a sequence which smooths out the workload and avoids peaks in certain subsequences (Pereira and Vilà 2015). As the first approach is often not feasible due to organizational or technical limitations, our approach is focused on the second strategy.

2.2 Multi-level assembly lines

Synchronous conveyance of all products through the stations of the main assembly line with constant speed and cycle time is a common and efficient setting in industry. This requires a task allocation at the main assembly line where the work intensity of all products lies within a limited range. As pointed out in the Introduction, for a heterogeneous portfolio the work intensity of all products does not necessarily lie within such a limited range. Therefore, manufacturers have to outsource variances in work intensity to pre-levels to guarantee an acceptable situation at the main assembly line. Frequently used types of pre-levels include in particular among others:

- Sub-Assembly: Pre-assembling of modules that are used at the main assembly line. For complex modules even the sub-assembly steps could be divided into different levels resembling a herringbone diagram. Modules could be pre-assembled for different stations of the main line in parallel.
- Kitting Areas: Small parts (e.g. screws, flat washers, angle brackets) are sorted just-in-sequence into relatively small bins before being finally delivered to the assembly line (Boysen and Emde 2014). From these bins, assembly workers pick the parts required by the current workpiece. The comparatively small bins ease the part retrieval process for the assembly workers and mostly eliminated inventory at the line where space is usually notoriously scarce, see Bozer and McGinnis (1992).
- Warehouse activities like commissioning and all intralogistical activities that are needed to guarantee the just-in-sequence material flow of components to the respective point of use, i.e. station of the main-assembly line or sub-assembly point. For details about intralogistical activities and challenges we refer to the literature where these topics have been broadly covered, see e.g. Bode and Preuss (2005) and Gudehus and Kotzab (2012).

Figure 1 gives an example of the material flows in between the different pre-levels and from pre-levels to the main assembly line. This includes all major flows explained above: Components can flow directly to the respective stations at the main assembly line or can go to sub-assembly stations, where modules are pre-assembled. These modules again flow to the main assembly line to be used for the final assembly steps there. Small parts are sorted just-in-sequence into small bins in the kitting area. The kitted bins get attached to the empty body of the relevant workpiece unit and both flow through all stations in the same order with constant speed and cycle time.

Figure 1 illustrates a situation without (major) intermediate buffers, a situation that is observable in modern lean manufacturing and assembling systems. It can be seen that the sequence of the main assembly line directly affects the sequence and therefore the temporal distribution of workload at all pre-levels. In such a situation smoothing pre-level workload can only be attained by sequencing the main assembly line.

2.3 Planning objective

As pointed out in the Introduction, the main goal of the mixed-model planning task is a smooth balancing of the heterogeneous products sequenced on the assembly line.

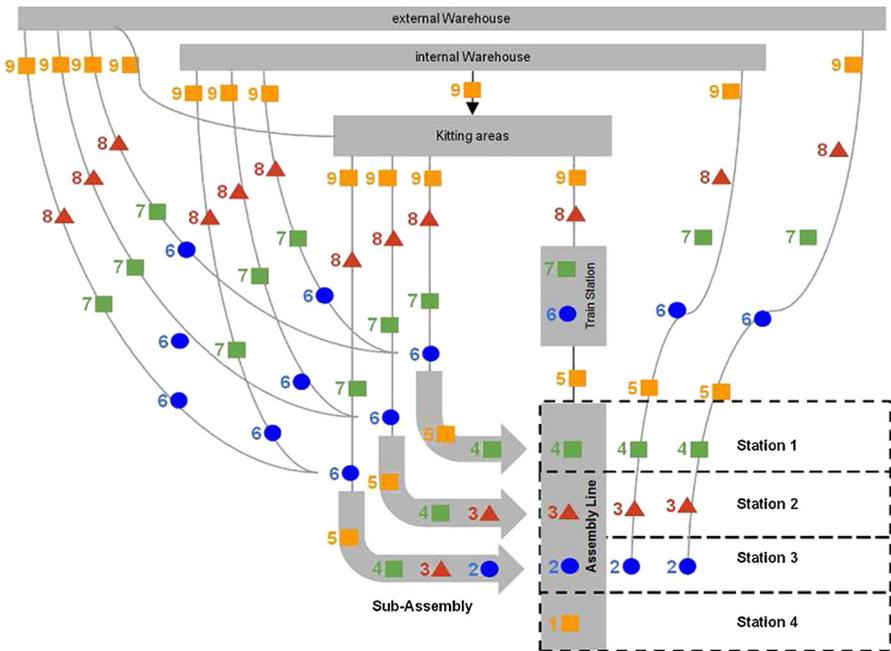


Fig. 1 Pre-levels of an assembly lines

Different workload intensities and different requirements from the pre-levels (see above) should be accommodated without causing major fluctuations in workload or even bottlenecks at any point of the production chain. There are a number of possible objective functions describing the level of fulfillment of this rather general goal for a given production plan. One basic strategy supported by advanced planning systems is the creation of a *maximum mixed* sequence (SAP 2017a), i.e. a sequence where the sum of distances (= time gap) between two identical products is as large as possible. In our setting the launching interval and the speed of the assembly line are constant over the entire sequencing horizon. Hence, the time gap can be measured as the number of sequence positions between two units of the same model.

This objective function aims at achieving an equal distribution of each model throughout the sequence and thus invokes indirectly a balanced utilization of pre-level resources. We followed this kind of *maximum mix* approach because it was considered favorably by the production company of our real-world case and because it is included as an option in the SAP APO system to support real-world mixed-model assembly lines [see SAP (2017a)]. Thus, a meaningful comparison can be performed between the outcome of the general purpose IT system and our tailor-made MIP model.

A different objective function would take the workload (resp. processing times) of the pre-levels explicitly into account. Considering processing times to minimize work overload is very common in mixed-model sequencing of assembly lines [e.g. Scholl et al. (1998) or Yano and Rachamadugu (1991)]. We extend this idea to multi-level production systems. Such an approach can be used to model more explicitly the actual

goal of guaranteeing the smooth execution of a sequence of heterogeneous orders. The idea itself is inspired by the famous Miltenburg (1989) paper that refers to it as “Just-In-Time usage problem”. A MIP representation of our approach is presented in Sect. 4.4. However, the advanced planning system SAP APO does not offer the implementation of such an approach. For comparison, we can evaluate this objective function for the solutions derived for the maximum mix approach.

2.4 Definitions and assumptions

In this paper, we focus on the short-term planning problem in mixed-model multi-level assembly lines and deal with the sequencing decision for a defined horizon (e.g. shift/day). The main assumptions and definitions are:

- There are M different models produced on the same main assembly line;
- the demand for each model m is given by d_m , $m = 1, \dots, M$, and d_m is known with certainty throughout the sequencing horizon;
- The total number of units to be assembled on the main line is $\sum_{m=1}^M d_m$ and equals the total number of available sequence positions T . Therefore, demand fulfillment is possible;
- the pre-level workload w_m (sum of all processing times required for a unit of model m over all pre-level operations) is given and depends on the model m ;
- the main assembly line is paced with constant speed;
- no intermediate buffers exist, therefore the sequence of the main assembly line directly implies the sequence at all pre-levels;
- the size of the workforce is stable over the sequencing horizon;
- no preemptions are allowed;
- all models have to start at the very first station of the main assembly line and flow in the same sequence over all stations (flow shop configuration).

3 MIP models

In this section we describe various modelling approaches for the given mixed-model planning task based on classical Mixed Integer Linear Programming (MIP) techniques. Recall that the objective function of maximizing distances between identical products serves as a proxy for representing balanced workloads to allow comparison with SAP APO. Explicit workload balancing including pre-levels will be considered in Sect. 4.4.

3.1 Maximum mix MIP model

A MIP formulation realizing the maximum mix planning logic available also in the advanced planning system can be stated as follows: The solution can be represented by a sequence of T positions, in which the binary variables x_{tm} ($t = 1 \dots T$, $m = 1 \dots M$) take value 1 iff a unit of model m is sequenced in position t of the sequence and 0 otherwise. Next, binary variables y_{tum} ($t = 1 \dots T$, $u = t + 1 \dots T$, $m = 1 \dots M$) indicate whether the distance between sequence positions t and u is relevant

for measuring the maximum mix or not. Therefore, y_{tum} takes a value of 1 if in positions t and u ($t < u$) units of the same model m are scheduled and 0 otherwise. The resulting basic MIP model can be written as follows:

$$\max \sum_m \sum_t \sum_{u>t} y_{tum} \cdot (u - t) \tag{1}$$

$$\text{s.t. } \sum_t x_{tm} = d_m \quad \forall m, \tag{2}$$

$$\sum_m x_{tm} = 1 \quad \forall t, \tag{3}$$

$$\frac{1}{2}(x_{tm} + x_{um}) \geq y_{tum} \quad \forall m, t, u > t, \tag{4}$$

$$x_{tm}, y_{tum} \in \{0, 1\} \quad \forall m, t, u > t. \tag{5}$$

The objective function (1) maximizes the mix in terms of number of sequence positions that are in between each pair of two single units of the same model m . Equation (2) ensure that the demand of each model m is satisfied. Note that by assumption there exists a feasible solution fulfilling the equality. Constraints (3) guarantee that exactly one product is assigned to each position in the sequence. Constraint set (4) enforces that only those pairs of sequence positions are considered as relevant, where units of model m are actually sequenced in the positions t and u . Finally, (5) are the usual binarity constraints.

3.2 Strengthening the basic model

In order to improve the solution behaviour of the MIP solver, additional inequalities were developed to strengthen the LP relaxation of the basic maximum mix MIP model (1)–(5). This means that each such additional inequality is fulfilled by every feasible integer solution, but there are solutions of the LP relaxation of the original model which violate the new inequality. Thus, upper bounds generated by LP relaxations for subproblems during a branch-and-bound process may be improved, i.e. reduced. For the performance of an ILP-solver this may yield better integral solutions within a given running time bound, which is our main goal, but may also imply shorter running times for finding an optimal solution. Since it is impossible to follow the sophisticated details of modern ILP-solvers it is hard to predict the effect of any such inequality. Note that the bounding procedures of ILP-solvers employ complex fine-tuned methods invoking some types of strengthening of their own. We will analyze in Sect. 6.2.2 which of the new inequalities actually brought an improvement in solution behaviour and which did not. It is likely that this effect is also solver dependent, but a computational comparison study of ILP-solvers is beyond the scope of this paper.

At first the following strengthening inequalities were introduced.

$$\sum_{u>t} y_{tum} \leq x_{tm} \quad \forall m, t \tag{6}$$

$$\sum_{t < u} y_{tum} \leq x_{um} \quad \forall m, u \quad (7)$$

$$\frac{3}{2}(x_{tm} + x_{um}) - 2 \leq y_{tum} \quad \forall m, t, u > t \quad (8)$$

$$y_{tum} + y_{usm} \leq y_{tsm} + 1 \quad \forall m, t, u > t, s > u \quad (9)$$

$$\sum_m y_{tum} \leq 1 \quad \forall t, u \quad (10)$$

Inequalities (6) and (7) strengthen the connection between x - and y -variables. The conjunction between x_{tm} , x_{um} and y_{tum} is represented in the original model only by an upper bound on y_{tum} in (4) while the lower bound will be implicitly enforced by the maximization criterion. However, we can explicitly force y_{tum} to 1, whenever units of model m are actually sequenced in positions t and u by (8). Furthermore, it is easy to see that variables y_{tum} have to obey a triangle condition: If the distances between positions t and u as well as u and s are relevant for a certain model m , then also the distances between t and s will be relevant for m . This is represented by inequalities (9). Unfortunately, the huge number of these inequalities (cubic in the number of positions) renders their generation impossible in practice (see Sect. 6.2.2). Inequality (10) represents the property that the distance between each pair of sequence positions t and u can only be relevant for one model m .

The following inequalities impose strengthened bounds on the maximum possible contribution to the objective function (1) of each model m .

$$\sum_t y_{tum} \leq (d_m - 1) \cdot x_{um} \quad \forall m, u \geq d_m \quad (11)$$

$$\sum_t y_{tum} \leq (u - 1) \cdot x_{um} \quad \forall m, u < d_m \quad (12)$$

$$\sum_u y_{tum} \leq (d_m - 1) \cdot x_{tm} \quad \forall m, t \leq N - d_m + 1 \quad (13)$$

$$\sum_u y_{tum} \leq (N - t) \cdot x_{tm} \quad \forall m, t > N - d_m + 1 \quad (14)$$

$$\sum_t \sum_u y_{tum} \leq \frac{d_m \cdot (d_m - 1)}{2} \quad \forall m \quad (15)$$

Inequalities (11) and (12) are based on a simple counting argument for every positions u . No more than $d_m - 1$ units of the same model m can be sequenced in earlier positions (there exist just d_m in total). For the first $d_m - 1$ positions the limit can even be strengthened, as for each position u at most $u - 1$ predecessors of the same model exist. Vice versa for each position t at most $d_m - 1$ succeeding units of the same model m exist (13) and also that limit can be strengthened for sequence positions close to the last position (14). Finally, for each model m we can sum up all contributions to y_{tum} over all positions t and u . Since we know from (2) that exactly d_m positions are assigned to model m , the k -th occurrence of model m must have exactly $d_m - k$ successors. Therefore, we can bound the total sum of pairwise relations by

$\sum_{i=1}^{d_m-1} (d_m - i) = \frac{d_m \cdot (d_m - 1)}{2}$ which is given by (15), where also an equality could be used.

4 Model extensions

In the following we will elaborate on a number of extensions and variations of the basic MIP model stated in Sect. 3.1.

4.1 Minimum distance

The objective function (1) maximizes the total number of sequence positions between each pair of two identical models. While this leads to a significant mix of models along the sequencing horizon in general, it does not rule out the occurrence of subsequences containing clusters of identical models. Such a behavior could lead to temporal overload of work in the pre-levels. It can be avoided by taking the *minimum* distance between pairs of two single units of the same model also into consideration.

For illustration consider the following simple example where two units need to be assembled of each model *A* and *B*. Model *A* induces a low workload whereas model *B* requires a relatively high workload. The two sequences *ABAB* and *ABBA* are both optimal (among others) with objective function value 4 w.r.t. (1). However, *ABBA* contains the subsequence *BB* which unnecessarily causes a peak in workload. If the minimum distance would be considered as well, *ABAB* would be preferred to *ABBA*.

It can be expected that the risk for suffering this type of problem decreases with an increasing number of models. Nevertheless, including the minimum distance is a valuable extension of the maximum mix MIP model. Basically, there are three options for doing so:

- Add an additional constraint.
- Add the minimum distance to the objective function as part of a linear combination.
- Lexicographic objective function: Among all optimal solutions, i.e. maximum mixed sequences, choose one with maximal minimum distance.

The implementation of minimum distance as an *additional constraint* requires an additional input parameter z^{\min} (minimum distance) and the following two restrictions:

$$y_{tum} \cdot (u - t) \geq z^{\min} + (y_{tum} - 1) \cdot T \quad \forall m, t, u > t, \tag{16}$$

$$x_{tm} + x_{um} - 1 \leq y_{tum} \quad \forall m, t, u > t. \tag{17}$$

The left-hand side of constraint (16) corresponds to the distance between pairs of units of the same model *m* sequenced in positions *t* and *u*, which has to be at least as large as the minimum distance z^{\min} on the right-hand side (active only if $y_{tum} = 1$). As constraint (16) counteracts the tendency of the objective function (1) to set y_{tum} to 1 whenever (4) allows to do so, constraint (17) is added to guarantee correct values for variables y_{tum} .

Defining a minimum distance as additional constraint raises the threat of infeasibility (recalling that the planning horizon is exactly equal to the total demand) and

the required input of a “reasonable” value for z^{\min} may be difficult to find. Therefore, it might be preferred to include the minimum distance into the objective function. In this case, z^{\min} represents a variable which is added in (1) with an appropriately chosen weight factor λ , while constraints (16) and (17) still ensure the correctness of z^{\min} .

$$\max \sum_m \sum_t \sum_{u>t} y_{tum} \cdot (u - t) + \lambda \cdot z^{\min} \tag{18}$$

The third option of considering the minimum distance is a *lexicographic* approach, i.e. among all optimal sequences w.r.t. (1), a sequence with maximal minimum distance is chosen. Formally, this can be done easily by solving the maximum mix problem (1)–(5) in a first step yielding an objective function value z^S for the sum criterion. In a second step the problem is solved again with the additional constraint (20) and the maximization of z^{\min} as objective:

$$\begin{aligned} \max \quad & z^{\min} & (19) \\ \sum_m \sum_t \sum_{u>t} y_{tum} \geq & z^S & (20) \end{aligned}$$

4.2 Due dates and release dates

Due dates and release dates [see e.g. Pinedo (2015)] are a valuable extension for our planning model. As the launching interval and the speed of the assembly line are constant over the entire sequencing horizon a constant cycle time is given. Therefore each sequence position t represents a certain point in time independently from the actual production sequence. Moreover, each due date or release date refers by definition to a certain model m . The model assignment in combination with the translation between sequence positions and time allows the integration of due dates and release dates into our MIP model.

Let r_{tm} resp. e_{tm} be the total number of units of model m with release date t resp. due date t . Clearly, there must be $\sum_{t \in T} r_{tm} = \sum_{t \in T} e_{tm} = d_m$ for all $m = 1, \dots, M$. Then constraint sets for respecting release dates and due dates can be written as follows:

$$\sum_{\tau \leq t} x_{\tau m} \leq d_m - \sum_{\tau > t} r_{\tau m} \quad \forall m, t \tag{21}$$

$$\sum_{\tau \leq t} x_{\tau m} \geq \sum_{\tau \leq t} e_{\tau m} \quad \forall m, t \tag{22}$$

(21) states that at any point of time t the total number of units of each model m assembled until this position cannot exceed the number of units with release date $\leq \tau$. On the other hand, constraints (22) ensure that until each position t the number of assembled units of each model is at least as large as the number of units due until that position.

Naturally, too tight input data for these dates may incur infeasibilities. For release dates infeasibilities could be avoided by decoupling the sequence length T from the

total demand and allowing $T > \sum_m d_m$. Then, one could relax (3) to $\sum_m x_{tm} \leq 1$. Infeasibilities caused by due dates could be handled by relaxing hard constraints (22), but including a violation of due dates into the objective function, analogous to the consideration of the minimum distance described above. This corresponds very well to the usual situation encountered in practice where violations of due dates are a common issue. There are diverse opportunities of respecting due dates in the objective function, e.g. minimizing maximum or total tardiness, total lateness or number of late jobs. We refer to the literature where these topics have been broadly covered [see e.g. Pinedo (2015)].

4.3 Limited pallet space

While mixed-model assembly can help to reduce stock quantities and therefore reduce space requirements for finished products, it tends to increase space requirements in close vicinity to the line due to the large number of different parts and pre-assembled components which have to be held available. But space is usually a scarce resource on the shop floor, especially for growing and evolving companies with given, natural limits to expansion plans. In particular, space is required for commissioning, dispatching, etc. at the end of the line. We will consider the very natural case, where the number of pallet positions at the end of the main assembly line is bounded by a parameter pp .

All units produced of a certain model are collected on one pallet for further shipping. We assume a 1:1 relation between models and pallets. Now, a unit of model m can only be assembled if a pallet assigned to model m is available at the end of the line. The capacity of a pallet naturally depends on the assigned model m and will be denoted by p_m . Once a pallet is placed at the end of the line, it will stay there until it is completely filled. Then it is transported onward by forklift and replaced by an empty pallet of any type. Note that the problem becomes infeasible if more than pp models have a demand d_m which is not an integer multiple of p_m .

It could be argued that the general goal of mixed-model assembly lines runs completely contrary to the requirement posed by such a limited number of pallet positions, namely that only a limited number of different models can be produced within a longer subsequence. The actual reducing influence depends on the pallet capacities p_m (higher capacities \Rightarrow lower mixing possibilities) and on the number of pallet positions pp (more positions \Rightarrow higher mixing possibilities). In practice, mixed-model assembly lines also call for more flexible logistics systems and are not easily compatible with intralogistic systems originating in an era of large-size batch processing. However, in a transition phase of a production system it will be unavoidable to take limitations such as space, which cannot be easily overcome, explicitly into account.

In the following we describe two related, but different MIP models for representing the requirements of limited pallet space.

4.3.1 Option A

The first variant takes for each model the total number of all units produced and stacked so far explicitly into account. Binary variables o_{tm} take value 1 iff at time t (= sequence

position)³ a pallet assigned to model m is available at the end of the assembly line. Integer variables n_{tm} and q_{tm} are auxiliary variables for accounting purposes, where n_{tm} represents the number of completely filled (and therefore already replaced) pallets of model m until sequence position t and q_{tm} equals the number of units of model m already placed on the respective pallet available at the end of the main assembly line at time t . Consequently, the available leftover capacity of the pallet at time t is given by $p_m - q_{tm}$.

$$\sum_m o_{tm} \leq pp \quad \forall t, \quad (23)$$

$$q_{tm} \leq o_{tm} \cdot (p_m - 1) \quad \forall t, m, \quad (24)$$

$$\sum_{\tau \leq t} x_{\tau m} = p_m \cdot n_{tm} + q_{tm} \quad \forall t, m, \quad (25)$$

$$o_{tm} \in \{0, 1\}, \quad n_{tm}, q_{tm} \in \mathbb{Z}^+ \quad \forall t, m. \quad (26)$$

(23) bounds the number of pallets currently available. Constraints (24) bound the capacity of each pallet assigned to some model m . Note that q_{tm} always remains smaller than p_m because a completely filled pallet will be removed immediately (thus increasing n_{tm}) and is no longer available for packing. The relation between produced and packed units at any point in time is kept by (25). Finally, (26) limits the ranges of the new variable sets.

4.3.2 Option B

In a second variant we avoid the explicit counting of all units produced so far. Instead, we only track the points in time where a pallet is filled up completely and must be exchanged. Therefore, binary variables f_{tm} ($t = 1 \dots T$, $m = 1 \dots M$) are introduced and take a value of 1, iff at sequence position t a pallet assigned to model m receives its very last item (i.e. is filled to capacity), and 0 otherwise. Variables q_{tm} are used as in Option A. The following constraints are introduced in addition to (23), (24) and (26):

$$q_{t-1,m} + x_{tm} - f_{tm} \cdot p_m = q_{tm} \quad \forall t > 1, m, \quad (27)$$

$$q_{1m} = x_{1m} \quad \forall m, \quad (28)$$

$$f_{tm} \in \{0, 1\} \quad \forall t, m. \quad (29)$$

Constraints (27) and (28) ensure that q_{tm} contains the appropriate value: Through (23) and (24) q_{tm} will always attain values as small as possible, i.e. bounded by (27), as long as $f_{tm} = 0$. Only if $f_{tm} = 1$ and a pallet of the respective model becomes full, the value of q_{tm} can “be switched back” to 0. (28) guarantees that this can happen only if the required number of units was actually produced. Rather surprisingly, it turned

³ Recall that the launching interval and the speed of the assembly line are constant over the entire sequencing horizon a constant cycle time is given. Therefore each sequence position t represents a certain point in time independently from the actual production sequence.

out in our experiments that Option A performed better in the test runs. Details are given in Sect. 6.2.4.

Alternatively to modelling limited pallet positions explicitly in the MIP model also a heuristic approach could be employed. The following clustering-based heuristic was used in our real-world case study. Before solving the maximum mix MIP model a clustering pre-step is applied. Within that pre-step the demand d_m over the whole planning horizon for each model is clustered into packets of the respective pallet size p_m . Consequently, the adapted demand quantity in the maximum mix MIP model is derived by dividing d_m by p_m and rounding up the result, if necessary. In the second step the maximum mix MIP model is solved with adapted demands in Eq. (2) given by $\lceil \frac{d_m}{p_m} \rceil$. In a third step the resulting sequence is used as a blue print for scheduling the original products, i.e., the clustering is dissolved again. The last cluster is reduced to match the actual demand size, if rounding up was necessary in the first step.

As a post-processing step, the units of a dissolved cluster, which were assigned to the same pallet, can be mixed locally with the next $pp - 1$ succeeding clusters. This helps a lot smooth out to some extent the workload peaks resulting from the clustering step while the pallet restrictions are still met.

4.4 Balancing pre-level workloads

The maximum mix model introduced in Sect. 3.1, but also its extension by a minimum distance aspect in Sect. 4.1, do not address directly the goal of the planning task as laid out in Sect. 2.3. Both aim only at the maximization of the distance between sequence positions of different units of the same model. This works out fine if the number of different models is large and the associated workloads are uniformly and randomly distributed. However, it is easy to imagine examples where a perfectly mixed sequence leads to unnecessary congestion of pre-level workloads. To avoid these shortcomings of the maximum mix approach we introduce a new objective function taking the specific workload of each model at the pre-levels explicitly into account. As the demand d_m is given for each model m and also the workload w_m is known upfront, the total pre-level workload $\sum_{m=1}^M d_m \cdot w_m$ within the entire sequencing horizon is given. Note that the technical planning process strives to distribute the total pre-level effort for a single model in a balanced way between the different pre-levels. Thus, we can restrict our attention to the total pre-level workload for each model since considering explicitly every individual pre-level step would give only minor improvements. Based on this total workload a theoretically ideal *target workload rate* r can be derived. The rate r is determined by smoothing the total workload evenly over the entire planning horizon and is therefore calculated as follows:

$$r = \frac{\sum_{m=1}^M d_m \cdot w_m}{T} \quad (30)$$

The target rate approach goes along with the concept of level scheduling [see e.g. Boysen et al. (2009b), Pereira and Vilà (2015), Kubiak (2009), Ng and Mak (1994) and the references therein], which received wide attention in the literature as part of the famous “Toyota Production System” [for details see e.g. the survey (Dhamala and

Kubiak 2005)] where target rates are employed for smoothing part supply in mixed-model assembly lines.

The target workload rate r does not directly imply an objective function. However, it makes sense to consider the deviation from the given target rate of the workload accrued for a certain production subsequence. For a subsequence of length ℓ , the target rate implies an ideal target workload of $r \cdot \ell$ which should be compared to the actually arising workload of the subsequence. Hence, we define the actual workload required for the production of models in a subsequence of length ℓ starting from sequence position t as

$$a_{t\ell} = \sum_{m=1}^M \sum_{t \leq \tau < t+\ell} w_m \cdot x_{\tau m}. \quad (31)$$

Note that if the entire sequence is considered ($\ell = T$), then actual workload equals target workload, i.e. $a_{1T} = r \cdot T$. For subsequences of length $\ell < T$ deviations can and most likely will occur. The deviation between target and actual workload of a subsequence of length ℓ starting from sequence position t is given as the absolute value of their difference:

$$\delta_{t\ell} = |a_{t\ell} - r \cdot \ell| = \left| \sum_{m=1}^M \sum_{t \leq \tau < t+\ell} w_m \cdot x_{\tau m} - \frac{\sum_{m=1}^M d_m \cdot w_m}{T} \cdot \ell \right| \quad (32)$$

Considering the minimization of the sum of these deviations as an objective functions leads to the following MIP formulation where L is a set of predefined subsequence lengths to be discussed below.

$$\min \sum_t \sum_{\ell \in L} \delta_{t\ell} \quad (33)$$

$$\sum_t x_{tm} = d_m \quad \forall m, \quad (34)$$

$$\sum_m x_{tm} = 1 \quad \forall t, \quad (35)$$

$$\sum_m \sum_{t \leq \tau < t+\ell} w_m \cdot x_{\tau m} - r \cdot \ell \leq \delta_{t\ell} \quad \forall t \leq T - \ell, \ell \in L, \quad (36)$$

$$r \cdot \ell - \sum_m \sum_{t \leq \tau < t+\ell} w_m \cdot x_{\tau m} \leq \delta_{t\ell} \quad \forall t \leq T - \ell, \ell \in L, \quad (37)$$

$$x_{tm} \in \{0, 1\}, \delta_{t\ell} \in \mathbb{R}. \quad (38)$$

The objective function (33) minimizes the deviations from target workload for all subsequences with length in L . Note that also quadratic deviations from the target rate are sometimes considered in the literature, see e.g. Pereira and Vilà (2015). The assignment constraints (34) and (35) are identical to (2) and (3). Constraints (36) and (37) together with the objective function ensure that $\delta_{t\ell}$ conforms to its definition (32), but we avoid the non-linear expression of (32).

Specifying relevant sets of subsequence lengths L should be based on the specific production and assembling environment. As sliding windows of length ℓ are considered, there are $T - \ell + 1$ positions for the window to be taken into account. Noting that a_{1T} is equal to $r \cdot T$ and consequently $\delta_{1T} = 0$, it does not make sense to choose a length ℓ close to T (entire sequence). Moreover, in that case only a small number of sliding windows would be considered. On the other hand, short subsequences seem to convey much better the idea of avoiding sudden bottlenecks or peaks in workload. However, going down to $\ell = 1$ is useless again as all δ_{t1} are independent from the actual sequence. These theoretical observations fit with the practical experience at the real-world problem which inspired our research. Practitioners cared mostly about the workload induced by the next 3–7 units to be assembled.

The presented MIP could be enhanced by all the extensions described in Sect. 4. Furthermore, the concept of target rate of workload could easily be applied for situations where workload intensity of the models differ significantly between the different pre-levels of the multi-level assembly line. Consider a situation where a model induces a high workload for pre-assembling but a low workload in the kitting area, while other models are work intensive in both. This could be captured by using model (m) and level (k) dependent workload values w_{mk} for calculating level dependent target workloads r_k and minimize level dependent deviations $\delta_{t\ell k}$. The same is true if differences between stations are relevant and could be implemented by analogous logic.

5 Modelling in SAP

SAP⁴ is the world's leading supplier of ERP (enterprise resource planning) software (Statista 2014). In most modern companies all major business processes are handled with help of ERP systems, which therefore represent the “backbone of most companies' information systems landscape” (cf. Kurbel 2013). Consequently, SAP ERP systems contain a multitude of master and transaction data and therefore embody rich data sources for real-world optimization applications. However, it is also possible to go beyond the approach of just using SAP ERP systems as data source for optimization since planning tasks can be carried out even within the SAP landscape of a company by using the advanced planning system SAP APO (Advanced Planning and Optimization).

On the positive side, the usage of SAP APO as an optimization device eliminates the necessity of building interfaces for data exchange, it possibly reduces the need for graphical output representations and it guarantees compatibility also after future software updates. It does not confront the IT-department of the company with additional third-party software providers and avoids additional training time for employees. On the negative side, there is a natural limit what a general purpose optimization device can achieve for individual, nonstandard problem settings with possibly complex side-constraints. Moreover, carrying the burden of a “jack-of-all-trades” may slow down computational performance considerably. It is one of the goals of this paper to shed more light on this often discussed but rarely investigated aspect.

⁴ SAP SE, see www.sap.com. SAP is a registered trademark.

SAP APO provides tools including heuristics and optimization methods for different planning purposes (e.g. demand planning, production planning and detailed scheduling, supply network planning) on different levels and for different temporal extensions. We refer to Ertel (2014) for more details about its range of functionalities. There are also some contributions in the academic literature on SAP APO implementations (Kurbel 2013; Bothe and Nissen 2013; Mabert et al. 2015; Günther et al. 2006). We refer especially to Stadtler (2011), which discusses real-world proportional lot-sizing and scheduling problems, and the comprehensive textbook of Kallrath and Maindl (2006), which provides deeper insights in supply network optimization and gives various detailed real-world application examples. However, to the best of our knowledge the usage of SAP APO for sequencing mixed-model assembly lines was never considered in the literature before.

Mixed-model planning is provided by the module SAP APO MMP (Model Mix Planning). MMP represents a “component for takt-based, flow manufacturing for configurable products with a high volume of orders”, see SAP (2017a). It can be executed to plan individual lines or a complete line network of serial or also parallel lines. In this paper we focus on MMP’s functionalities as a “sequence optimizer”, however, for the sake of completeness, it is pointed out that SAP APO Model Mix Planning provides various heuristics for medium- and long-term planning of assembly lines. The “sequence optimizer” tool is employed in companies for sequencing orders of lot size 1 on their real-world mixed-model assembly lines and can take a number of customer requirements into account.

5.1 Restrictions in SAP APO MMP

The main principle of SAP APO MMP is an objective function minimizing the sum of constraint violations. This means that all requirements are generally seen as “soft” constraints or *restrictions*, although some of them could also be specified as “hard” constraints. The “sequence optimizer” searches for a feasible (w.r.t. the hard constraints) production sequence with smallest restriction violation cost of the soft constraints. Classifying too many constraints as “hard” can easily lead to infeasibility. The strategy of minimizing restriction violations is in line with the common concept in car sequencing where production orderings are sought not violating given sequencing rules [see e.g. Gagné et al. (2006) or the survey (Boysen et al. 2009b)]. The following predefined restriction types are available for modelling.

- quantity restriction
- spacing restriction
- block restriction
- k -in- m restriction
- position restriction
- equal distribution

Every single restriction has to be defined in accordance with one of these given restriction types and must be linked to a certain *characteristic* value defining which specific subset of products is encompassed by the specific restriction, e.g. color = “blue”,

product group= “premium engines” or product number = “1234”. The six classes of restrictions can be used to obtain the following constraints, see SAP (2017a).

Quantity restrictions bound the occurrence of a characteristic to either a minimum or a maximum quantity per day or shift. Consequently, that type of restriction will only make sense, if the sequencing horizon includes several days or shifts. Spacing restrictions limit the minimum distance of the occurrence of a characteristic in the planned sequence of orders. Block restrictions bound the occurrence of a characteristic to a certain minimum (maximum) number of consecutive products. e.g., Assigning the minimum (maximum) number 5 to the characteristic color = “black” implies that at least (at most) five black products must be assembled consecutively. Setting equal minimum and maximum block restrictions permits the prescription of exact block sizes.

k-in-*m* restrictions limit a characteristic to a maximum quantity within a certain interval, such as a maximum of 7 products out of 20 products having a certain characteristic. Position restrictions limit the occurrence of a certain characteristic to a certain position or interval within the sequence. Finally, equal distribution restrictions aim to evenly distribute the occurrence of a certain characteristic over the sequencing horizon. e.g., If 100 units need to be produced within the sequencing horizon and 25 of them have the characteristic color = “red”, then the equal distribution restriction will assign red units to every fourth sequence position.

Every restriction is assigned a validity period and a weight. The weight of a restriction represents its importance and therefore determines the penalty cost for any violations. Basically, MMP always tries to fulfill all restrictions. However, the more restrictions, orders and characteristics are involved, the less likely it becomes to reach this goal. Restrictions with a weight of 0 are interpreted as hard constraints which must be unconditionally fulfilled.⁵ Furthermore, restrictions can also be defined as soft constraint by assigning a weight in the range of 1–9, where 1 gives the highest priority and consequently the highest violation costs and 9 gives the lowest priority and penalty costs for violations. Independently of the actual weight of the restriction, the penalty costs for violations increase quadratically with the magnitude of the violation.

In addition to these classes of restrictions, SAP APO MMP can also take customer required dates for the orders into consideration. These are specific completion dates, where both early and late deliveries incur a violation, i.e. a generalization of due dates. However, the advanced planning system does not support the creation of any other types of constraints. This means that not all production situations can be represented in MMP and thus its capabilities are clearly not comparable to the flexibility of tailor-made MIP models.

5.2 Solution methods in SAP APO MMP

For the determination of the production sequence SAP APO MMP provides three different optimization procedures. The first method is the so-called *Slotting Heuristic*, a rather simple heuristic, which only aims at generating sequences that respect all hard

⁵ Equal distribution restrictions cannot be defined as hard constraints: <https://help.sap.com/saphelpem70/helpdata/en/29/3734414b75de2be1000000a1550b0/content.htm>.

constraints, but ignore all soft constraints. If no feasible sequence exists, demands are reduced artificially to avoid hard constraint violations. The second solution method is called *Prioritized Equal Distribution*. It is a heuristic method based on sorting by priorities which evenly distributes orders over the planning horizon if their characteristics are subject to restrictions. More details can be found at SAP (2017a). Note that due dates are not taken into account by this method.

The third and most sophisticated solution method is a built-in *Genetic Algorithm*, see e.g. Goldberg (2006) for a general introduction. The Genetic Algorithm aims at finding a sequence with minimal penalty costs for violation of restrictions. It is based on keeping a population of solutions sequences and rearranging subsequences for members of the population in different ways. The fitness value of a solution is given by its total penalty costs over all restriction violations [see SAP (2017b)]. Unfortunately, the Genetic Algorithm used in SAP APO MMP remains more or less a black box. Crucial features like the structure of the chromosomes, the crossover operator or the mutation rate can neither be adapted nor displayed. The only customizable parameter is the termination criterion, which can be specified either as runtime limit (in seconds) or by the number of iterations. The *Slotting Heuristic* is integrated in the Genetic Algorithm to create an initial sequence for the orders in a first step.

5.3 Modelling the considered problem in SAP APO MMP

In the following we present how the problem captured by the MIP model and its extensions as discussed in Sect. 3 can be transformed into a SAP APO MMP realization. The maximum mix MIP model presented in Sect. 3.1 could be modelled in SAP APO MMP by simply setting up M equal distribution restrictions, one for each model m , and each restriction with characteristic value set to model m . The weights of the restrictions should all be identical and can be set to any value between 1 and 9, recalling the fact that equal distribution restrictions must not be hard.

5.3.1 Distance and date extensions

The minimum distance extension (Sect. 4.1) could be covered by introducing spacing restrictions with minimum space requirements. They again have to be defined by using the characteristic value model as the distance (MMP: “space”) which is calculated between units of the same model. The first MIP option discussed in Sect. 4.1 “add an additional constraint” can be implemented by defining the spacing restrictions as hard (weight = 0). The second option “include the minimum distance into the objective function as part of a linear combination” could be implemented by assigning any weight between 1 and 9 to the spacing restrictions, whereby the relation between the weight factors for the equal distribution and spacing restrictions fulfill the same function as the weight factor λ in the MIP model. However, this SAP APO MMP modelling approach does not represent exactly the same logic as the MIP model. We recall the fact that in the MIP model variable z^{\min} (representing the smallest distance of any pair of positions assigned to the same model) was maximized. SAP APO MMP, however, minimizes the violation costs for all spacing restriction violations

and does not focus on the smallest distance. Nevertheless, properly applied the SAP APO MMP logic could be used at least for an approximation. We can make use of the quadratic increase of violation costs with the magnitude of the violation. Defining the minimum spacing requirement in the soft restriction high enough (i.e. $> M$), the high absolute difference between violation costs of weak violations (big distance) and strong violations (small distance) will lead to a focus on the strongest violations and therefore the smallest distances.

The third option presented in Sect. 4.1, namely the lexicographic consideration of the minimum distance, can also be only approximated in SAP APO MMP. Imposing a far stronger penalization of equal distribution restrictions than for spacing restrictions (e.g. weight 1 for all equal distribution restrictions and weight 9 for spacing restrictions) will do so.

Due dates (Sect. 4.2) can be respected in SAP APO MMP, but not all solution methods support them.⁶ No additional restrictions need to be defined, only the parameter (customer required dates) for penalizing late delivery in the so-called processing profile of the solution method need to be set and again weights have to be assigned. Also for customer required dates only soft weights are supported by SAP APO MMP,⁷ which is different to the approach presented in the MIP, but eliminates the threat of infeasibilities. Release dates are not supported for sequencing in SAP APO MMP. As already mentioned in Sect. 5.1, if required, not only a later but also a delivery earlier than the customer's preferred date can be taken into account by the Genetic Algorithm again by setting the respective parameter in the processing profile.

5.3.2 Modelling limited pallet space

Limited pallet space as described in Sect. 4.3 is supported by standard tools in SAP APO MMP only for the very special case that all model demands d_m are integer multiples of their respective pallet size p_m . In this case block restrictions with a characteristic value set to model m and maximum as well as minimum block size set to p_m will lead to a result where blocks of p_m units are sequenced one after the other if the block restrictions are stronger weighted than the equal distribution ones. Hence, even though planning is performed on lot size 1 level, the resulting sequence contains lots greater than 1. Consequently, model mix is more limited⁸ with such a solution than for the MIP approach. However, this approach becomes impractical if d_m is not a multiple of p_m . Consider for example a situation with demand $d_A = 27$ and $p_A = 10$: Hard block restriction would not give any solution due to infeasibility. Therefore block restrictions have to be defined as soft constraint. Because of the quadratic increase of violation costs, three lots of size 9 are preferred (w.r.t. violation costs) over a solution with two lots containing 10 units and one lot containing 7 units. However, this scheme leads to many almost filled pallets which is clearly unsatisfactory from a business perspective.

⁶ Genetic Algorithm does, but Prioritized Equal Distribution does not.

⁷ The Slotting Heuristic takes only hard constraints into account and will not respect customer required dates.

⁸ Equal distribution restrictions still aim at the maximization of the distance between two blocks of the same model.

In the following we present a more sophisticated way of implementing the limited pallet space requirement: As pointed out in Sect. 5.1 SAP APO MMP allows only a limited pre-defined set of different restriction types which limits its flexibility compared to tailor-made MIP models. However, some degree of flexibility can be gained by using so-called BAIs (Business Ad-Ins). BAIs are pre-defined access points, which can be used to enhance SAP standard coding. We used BAI-methods to communicate to the Genetic Algorithm that in situations as in the previous example, lots with distribution $10 - 10 - 7$ should be preferred over $9 - 9 - 9$. The idea follows the spirit of the heuristic approach discussed in Sect. 4.3. All occurring block restrictions with any soft weight ($1 - 9$) are set up in standard SAP APO MMP.

- We split the sequence positions $t = 1 \dots T$ into two partitions at position T_H : $(1, \dots, T_H)$ and $(T_H + 1, \dots, T)$.
- The standard block restriction is duplicated.
- The weights of the duplicated restrictions are changed to 0 (hard).
- We assign the (duplicated) hard block restrictions to sequence positions $(1, \dots, T_H)$ and the (original) soft block restrictions to the other partition $(T_H + 1, \dots, T)$. Therefore, in the first part of the sequence only lots of size p_m are allowed, in the second part of the sequence all others.
- Consequently, T_H is calculated by $\sum_m \left\lfloor \frac{d_m}{p_m} \right\rfloor \cdot p_m$ to induce the Genetic Algorithm to sequence as many completely filled pallets (p_m units on pallet) as possible.
- For the exceptional case that one model accounts for more than half of the completely filled pallets, T_H must be reduced, as two consecutive lots of size p_m in the hard part of the sequence are interpreted as one lot of size $2p_m$ and therefore induce a violation of the hard block constraint.
- As a post-processing step, the identical models of blocks assigned to the same set of pallets could be locally mixed by invoking the final part of the heuristic described at the end of Sect. 4.3 by using a second BAI-method.

Finally, it must be pointed out that there is no way in SAP APO MMP to implement the more advanced setting described in Sect. 4.4, where the actual workload of pre-levels is explicitly taken into account and which therefore corresponds most accurately to the actual business goals.

6 Computational tests for a real-world application

6.1 Real-world application

Together with *scc EDV-Beratung AG*, an Austrian IT-consultancy company specialized in introducing, customizing and developing SAP applications, we carried out a real-world project with an industrial partner, a Central European producer of engines, gear boxes and other machinery parts supplied mainly to vehicle manufacturers. The project involved a production paradigm shift from large- and medium-scale batch production to lot size one mixed-model assembly lines. This shift was triggered by the need for a higher degree of product customization and higher flexibility in production planning.

In the new setting numerous units of different models are produced and scheduled individually on the same main assembly line. However, each of those different models induces a different workload for the employed workers. To be able to maintain a constant speed and cycle time on the main assembly line, some production steps are outsourced to pre-levels, namely pre-assembly stations and kitting areas, to eliminate workload variances on the main line. As workforce size cannot be adapted within one shift the temporal distribution of workload should be smoothed out even at those pre-levels. Due to scarce space in the area of production (almost) no intermediate buffers between the pre-levels and the main assembly line exist. Therefore, a sophisticated production planning and sequencing mechanism is sought to guarantee a smooth workload at all levels.

The production process is organized as follows: Each single engine or gear box is conveyed by an automatic guided vehicle (AGV) through all stations of the main assembly line. The use of AGVs allows an easy reconfiguration of the main assembly line from a mid- and long-term point of view. For short-term planning it is given that all units pass through all stations in the same order as well as with constant speed and cycle time. Attached to the AGVs (as a separate logistical AGV) are the kitting parts sorted into small bins, one for each station. This requires the just-in-sequence commissioning of all required small parts in a pre-level step. At the beginning of the main line, the AGV contains only the empty body of the work piece unit. In every station of the main assembly line components or modules pre-assembled in sub-assembly work stations are added to the work piece unit. Intralogistical processes ensure the just-in-sequence supply of components to the work stations of the main and sub-assembly lines. The entire material flow corresponds roughly to the illustration given in Fig. 1.

6.2 Computational results

Based on data from the real-world application described in Sect. 6.1 we conducted computational experiments. The main goal of the experiments was the comparison of the computational performance of the MIP models and the realization in SAP APO by using real-world data for sequencing decisions from the industrial partner. Due dates and release dates are considered by the company in the medium-term planning decisions. Hence, they are no longer considered in the short-term sequencing decision and no real-world data on due dates and release dates was available. Therefore, we excluded this type of extension from our tests.

The MIP models were implemented in PuLP (<https://pypi.python.org/pypi/PuLP/1.4.9>) under Python (www.python.org), an open source modelling software, and solved by the exact solution framework of Gurobi 6.5.2. (www.gurobi.com). Gurobi provides a large set of parameters to control the exact solution approach and solver behavior (e.g. presolve options, branch directions, sifting methods). As our work was not focused on fine-tuning solver parameters, we applied the Gurobi default settings for MIP models. The underlying LP relaxations were solved within Gurobi by the dual simplex method. All tests were performed on a standard personal computer equipped with an Intel Core i5-3210M processor with 2.5 GHz and 4 GB memory.

6.2.1 Comparison of model mix

In the first computational experiment we compared the solution performance of the maximum mix MIP described in Sect. 3.1 using Gurobi with the performance of SAP APO MMP. In the latter, M equal distribution restrictions were established for each model m as described in Sect. 5.3. Solutions in SAP APO MMP were derived by the built-in (black box) Genetic Algorithm.

All tests were performed with real-world instances from our industrial partner. The instances vary concerning the number of sequence positions T and the number of different models M to be scheduled. Furthermore, for each combination of T and M , we tested one low and one high variance instance. The variance of an instance refers to the demands d_m for each single model, whereby no variance is given if all model demands d_m equal exactly $\frac{T}{M}$. We define low (high) variance instances as instances where all model demands d_m deviate at most 20% (100%) from $\frac{T}{M}$. The instances contain between 80 and 150 units to be assembled out of 4–15 different engine and gearbox models. Unfortunately, we could obtain from the company only 12 real-world instances of relevant size.

We solved each real-world instance applying three different time limits to study the possible improvement of the solution quality over time. The time limits for the Gurobi solver were set to 1 min, 10 min and 1 h. We present the objective function values of the best feasible solution found until the respective time limit in Table 1. For the SAP APO MMP built-in Genetic Algorithm we also set three different time limits as termination criterion and show the values of objective function (1) of the best solution found in Table 1. However, as pre-tests showed that hardly any improvement steps occur at later times, the time limits were set only to 30 s, 1 min and 10 min. This argument is validated by the results of the experiment, where the SAP APO MMP managed to improve the solution value in the time window between 1 and 10 min only for 3 instances and the largest such improvement was below 0.0001%. For half of the instances the best solution after 30 s was equally good as the best solution after 10 min.

The main insight from the experiment is the fact that SAP APO MMP finds quite good sequences very fast, compared to the MIP model. However, a longer running time does not help to further improve these solutions. Only for 3 out of 12 instances the maximum mix MIP received solutions within 1 h better than SAP APO MMP within 30 s. All of these 3 instances have high variances in the model demands. Gurobi could not solve any of these MIP instances to proven optimality within 1 h. However, for all MIP model instances Gurobi found feasible solutions within 1 h and none of them were more than 4% below the best SAP APO MMP solution. Within the time limit of 10 min (1 min) Gurobi delivered feasible solutions for 92% (33%) of all instances and outperformed SAP APO MMP for 17% (8%) of the instances. Compared to the MIP model it seems that SAP APO performs worse for test instances with high variance. The exceptions from this observation (instances 11 and 12) are due to the performance of the MIP solver, which slows down with the high number of time periods.

The moderate deviation between the solutions derived by the MIP model and by SAP APO MMP suggests that these solution values are rather close to the optimal

Table 1 Comparison of maximum mix MIP and SAP APO MMP

| Instance | Instance | <i>T</i> | <i>M</i> | Maximum mix MIP | | | SAP APO MMP | | |
|---------------|----------|----------|----------|-----------------|---------|---------|-------------|---------|---------|
| | | | | 1 min | 10 min | 60 min | 30 s | 1 min | 10 min |
| Low variance | 1 | 80 | 4 | 20,408 | 20,479 | 20,479 | 21,374 | 21,374 | 21,374 |
| | 2 | 120 | 8 | 9070 | 9761 | 10,219 | 10,656 | 10,658 | 10,658 |
| | 3 | 120 | 6 | - | 46,387 | 46,584 | 48,218 | 48,218 | 48,222 |
| | 4 | 150 | 10 | - | 26,967 | 27,567 | 28,889 | 28,889 | 28,889 |
| | 5 | 150 | 5 | - | 110,645 | 111,710 | 114,160 | 114,160 | 114,164 |
| | 6 | 80 | 15 | - | - | 36,363 | 37,936 | 37,936 | 37,936 |
| High variance | 7 | 80 | 4 | 29,771 | 30,627 | 30,627 | 29,590 | 29,590 | 29,590 |
| | 8 | 120 | 8 | 12,837 | 13,455 | 13,468 | 13,663 | 13,663 | 13,663 |
| | 9 | 120 | 6 | - | 54,039 | 59,061 | 58,625 | 58,629 | 58,629 |
| | 10 | 150 | 10 | - | 43,278 | 43,278 | 42,753 | 42,753 | 42,753 |
| | 11 | 150 | 5 | - | 169,637 | 169,898 | 171,379 | 171,389 | 171,389 |
| | 12 | 80 | 15 | - | 42,047 | 47,657 | 48,502 | 48,525 | 48,527 |

Values of objective function according to (1) for best solution found within timelimit

values. However, we cannot certify this assumption by computational results since the gaps to the upper bounds reported by the MIP solver are really huge (see Sect. 6.2.2). On the other hand, it would be very surprising if the two completely different solution algorithms end up with solution values in the same range for every instance while the optimum is larger by orders of magnitude.

6.2.2 Strengthening the maximum mix MIP with additional inequalities

In order to test the impact of adding the valid inequalities presented in Sect. 3.2 on the solution behaviour of the Gurobi MIP solver, we ran various tests with the provided instances. It turned out that the impact on the upper bounds was significant, while the best objective function values found within a certain time limit varied only moderately. To display the behaviour on both bounds and object function values, we report the percentage gaps between the best solution and the lowest upper bounds found.⁹

The triangular conditions represented by restrictions (9) produce a huge number of constraints, namely cubic in the number of positions for each model. It turned out that even generating all these constraints in Python PuLP and reading them into Gurobi takes several hours. Thus, we did not pursue these constraints any further.

The remaining constraints can be partitioned into five logical groups: (6)–(7), (8), (10), (11)–(14), and (15). Tables 2 and 3 show the optimality gaps of the computational experiments on the same instances that were used in Sect. 6.2.1.

First of all, the simple constraint (10) turned out to help the solver in finding feasible solutions faster, but yielded just a minor positive effect on bound improvement, if any. On the other hand, adding also (15) reduces the bounds significantly with gaps decreasing between 104 and 1129 percentage points. This is quite impressive given that only a very small quantity of $m \in [4, 15]$ constraints are added to the MIP model. Therefore, (10) and (15) were kept in the model for all further tests.

The other three groups of constraints (6)–(7), (11)–(14) and (8) were added separately to the model to study their individual influence. Finally, all constraints were added together. The results depicted in Tables 2 and 3 show that constraints (11)–(14) delivered better improvements than the two other groups of constraints for 29 out of the 36 instance-runtime combinations. Moreover, for short running times the results with these additional restrictions are even better than the results for using all additional inequalities.

The best results after 60 min for most of the instances were obtained by using all additional inequalities. However, for the three largest instances with low variance and for the largest instance with high variance adding just the constraints (10), (11)–(14), (15) led to smaller gaps. A closer analysis of the solution progress showed that the LP relaxation at each node becomes very time consuming for these larger instances, if all inequalities are taken into account. Therefore the larger number of visited nodes seems to be the reason for the smaller gaps of the reduced variant. In general, the magnitude of the gaps after 60 min runtime reported by the MIP solver are still really huge.

⁹ A value of 100 means that the upper bound is twice as large as the best feasible solution.

Table 2 Comparison of gaps for the maximum mix MIP with valid inequalities

| Instance | 1 | | | 2 | | | 3 | | | 4 | | | 5 | | | 6 | | |
|--------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|-----|-----|-----|-----|-----|------|------|------|
| | 1 | 10 | 60 | 1 | 10 | 60 | 1 | 10 | 60 | 1 | 10 | 60 | 1 | 10 | 60 | 1 | 10 | 60 |
| <i>Additional constraints</i> | | | | | | | | | | | | | | | | | | |
| None | 298 | 294 | 294 | 732 | 702 | 694 | - | 501 | 493 | - | 942 | 904 | - | 394 | 386 | - | - | 1424 |
| (10) | 300 | 290 | 290 | 768 | 732 | 698 | 530 | 522 | 500 | 1018 | 956 | 908 | 418 | 397 | 393 | 1742 | 1520 | 1459 |
| (10), (15) | 103 | 100 | 98 | 128 | 127 | 121 | 122 | 122 | 121 | 131 | 130 | 130 | 113 | 111 | 110 | 185 | 141 | 141 |
| (10), (15), (6), (7) | 101 | 98 | 93 | 167 | 132 | 123 | 122 | 121 | 119 | 148 | 148 | 132 | 117 | 110 | 108 | 188 | 188 | 154 |
| (10), (15), (8) | 99 | 80 | 78 | 126 | 120 | 117 | 134 | 119 | 115 | 139 | 133 | 131 | 113 | 112 | 106 | 182 | 141 | 141 |
| (10), (15), (11)-(14) | 54 | 52 | 52 | 74 | 55 | 52 | 89 | 57 | 56 | 144 | 57 | 57 | 112 | 63 | 62 | 185 | 151 | 67 |
| (10), (15), (6), (7), (8), (11)-(14) | 97 | 49 | 48 | 147 | 66 | 51 | 128 | 127 | 56 | 150 | 144 | 82 | 114 | 113 | 82 | 147 | 147 | 117 |

Values in %. Low variance instances

Table 3 Comparison of gaps for the maximum Mix MIP with valid inequalities

| Instance | 7 | | | 8 | | | 9 | | | 10 | | | 11 | | | 12 | | |
|--------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|
| | 1 | 10 | 60 | 1 | 10 | 60 | 1 | 10 | 60 | 1 | 10 | 60 | 1 | 10 | 60 | 1 | 10 | 60 |
| Runtime in minutes | 171 | 166 | 165 | 525 | 523 | 506 | - | 382 | 364 | - | 557 | 527 | - | 223 | 220 | - | 1042 | 1042 |
| Additional constraints | 176 | 165 | 164 | 522 | 514 | 495 | 404 | 384 | 378 | 640 | 547 | 547 | 236 | 236 | 219 | 1307 | 1149 | 1056 |
| None | 63 | 62 | 60 | 110 | 107 | 105 | 104 | 101 | 100 | 120 | 106 | 106 | 90 | 83 | 82 | 178 | 125 | 125 |
| (10), (15) | 63 | 60 | 53 | 110 | 106 | 106 | 122 | 103 | 101 | 138 | 107 | 105 | 96 | 82 | 81 | 182 | 182 | 182 |
| (10), (15), (6), (7) | 60 | 42 | 34 | 111 | 93 | 89 | 253 | 102 | 87 | - | 108 | 91 | 91 | 81 | 75 | - | 125 | 125 |
| (10), (15), (8) | 48 | 45 | 45 | 92 | 71 | 61 | 114 | 67 | 66 | 137 | 93 | 73 | 94 | 69 | 65 | 178 | 161 | 78 |
| (10), (15), (11)-(14) | 70 | 35 | 32 | 130 | 64 | 57 | 117 | 111 | 57 | 123 | 117 | 65 | 90 | 88 | 59 | 154 | 154 | 127 |
| (10), (15), (6), (7), (8), (11)-(14) | | | | | | | | | | | | | | | | | | |

Values in %. High variance instances

6.2.3 Comparison of pre-level balancing

In Sect. 4.4 we gave a MIP model for the actual business objective of explicitly balancing pre-level workloads, which could not be modelled directly in SAP APO MMP. Our second experiment examines whether the indirect balancing effort of favoring an equal distribution of each model throughout the sequence as conducted in SAP APO MMP (and also modelled in our maximum mix MIP) actually works out in practice. To do so we extended the real-world instances used in the experiments of Sect. 6.2.1 by pre-level workloads w_m given by the practitioners. The workloads range from 0 to 20. As these workloads w_m enter neither SAP APO MMP nor the maximum mix MIP explicitly, the sequences computed in Sect. 6.2.1 are not affected and we only calculated their objective function values according to (33). The results are presented in Table 4 and compared to the best solutions found by Gurobi for the MIP model for explicitly balancing pre-level workloads as described in Sect. 4.4.

The experiment showed that Gurobi was (in contrast to the maximum mix MIP) able to solve some instances of the MIP model for explicitly balancing pre-level workloads to proven optimality and to find feasible solutions for all instances within the first minute. Also the improvements of the best solution found, if the time limit is increased, tend to be higher than in the first experiment. Furthermore a significant gap between direct (pre-level workload MIP) and indirect (maximum mix MIP and SAP APO MMP) balancing of pre-level workload is blatantly obvious in Table 4.

In Fig. 2 we illustrate this gap, i.e. the effect on the temporal distribution of workload on the pre-levels for the three different approaches. The figure shows the pre-level workload induced by the actual unit assembled and its 4 succeeding units ($= a_{t5}$) for the best solutions found for instance 12. The target workload rate r of that instance is 6.72 and therefore the target workload for subsequences of length 5 adds up to 33.6. While the best solutions of maximum mix MIP and SAP APO MMP show significant deviations from the target workload, the pre-level workload MIP is able to balance the workload smoothly. Note that this is the instance with the largest sum of deviations in the optimum, hence pre-level workload could be even better balanced for all other instances.

Similar to the first experiment SAP APO MMP outperforms the maximum mix MIP for most, but not for all instances. Note also that solutions with the same objective function value according to (1) in Table 1 can well have different objective function values according to (33) in Table 4, see e.g. instance 6. Moreover, solutions with better function values w.r.t. (1) may imply even worse solutions w.r.t. (33). This occurs i.e. if sequences are better mixed (larger total distance to all other units of the same model), but units of different models with relatively high pre-level workload are scheduled in succession. That effect supports the argument for using direct balancing (pre-level workload MIP) instead of indirect consideration (maximum mix MIP and SAP APO MMP) of the actual planning goal. All values were calculated with $L = \{5\}$ (i.e. considering the currently assembled unit and its four successors) as this conforms best with the goals stated by the practitioners.

The significant gap between direct and indirect balancing of pre-level workload (see also Fig. 2) represent significant cost saving potentials. Due to the higher volatility of pre-level workload, additional leasing workers need to be temporarily employed

Table 4 Comparison of maximum Mix MIP, SAP APO MMP and the MIP for explicitly balancing pre-level workloads. Values give the objective function according to (33). Values marked with * indicate solutions of proven optimality

| Instance | | T | | M | | Maximum mix MIP | | | SAP APO MMP | | | Pre-level workload MIP | | | |
|---------------|----------|-----|----|--------|---------|-----------------|--------|--------|-------------|--------|--------|------------------------|--------|--------|--------|
| Type | Instance | T | M | 1 min | 10 min | 60 min | 30 s | 1 min | 10 min | 1 min | 10 min | 60 min | 1 min | 10 min | 60 min |
| Low variance | 1 | 80 | 4 | 461.13 | 458.13 | 458.13 | 262.56 | 262.56 | 259.43 | 40.52 | 40.39* | 40.39* | 40.52 | 40.39* | 40.39* |
| | 2 | 120 | 8 | 400.38 | 447.38 | 362.38 | 321.63 | 343.13 | 379.13 | 41.78 | 39.29 | 25.29 | 41.78 | 39.29 | 25.29 |
| | 3 | 120 | 6 | - | 808.88 | 657.63 | 335.63 | 342.88 | 371.38 | 61.50 | 41.50 | 36.00 | 61.50 | 41.50 | 36.00 |
| | 4 | 150 | 10 | - | 705.63 | 683.63 | 498.04 | 481.38 | 571.13 | 82.44 | 70.88 | 67.88 | 82.44 | 70.88 | 67.88 |
| | 5 | 150 | 5 | - | 835.17 | 719.50 | 293.17 | 275.50 | 278.17 | 42.67 | 33.67 | 33.00 | 42.67 | 33.67 | 33.00 |
| | 6 | 150 | 15 | - | - | 884.40 | 934.00 | 890.40 | 848.20 | 114.20 | 95.20 | 89.80 | 114.20 | 95.20 | 89.80 |
| High variance | 7 | 80 | 4 | 545.00 | 685.50 | 685.50 | 282.94 | 282.94 | 282.94 | 70.60* | 70.60* | 70.60* | 70.60* | 70.60* | 70.60* |
| | 8 | 120 | 8 | 352.80 | 298.81 | 324.69 | 334.44 | 326.06 | 332.25 | 36.49 | 25.58 | 24.41 | 36.49 | 25.58 | 24.41 |
| | 9 | 120 | 6 | - | 594.13 | 588.00 | 420.13 | 437.63 | 437.63 | 47.00 | 43.26 | 31.26 | 47.00 | 43.26 | 31.26 |
| | 10 | 150 | 10 | - | 344.88 | 344.88 | 475.88 | 472.38 | 451.63 | 61.29 | 54.88 | 31.00 | 61.29 | 54.88 | 31.00 |
| | 11 | 150 | 5 | - | 416.97 | 418.97 | 273.83 | 270.97 | 275.83 | 82.00 | 82.00 | 82.00 | 82.00 | 82.00 | 82.00 |
| | 12 | 150 | 15 | - | 1150.83 | 1098.50 | 889.03 | 951.83 | 911.77 | 118.72 | 108.00 | 103.83 | 118.72 | 108.00 | 103.83 |

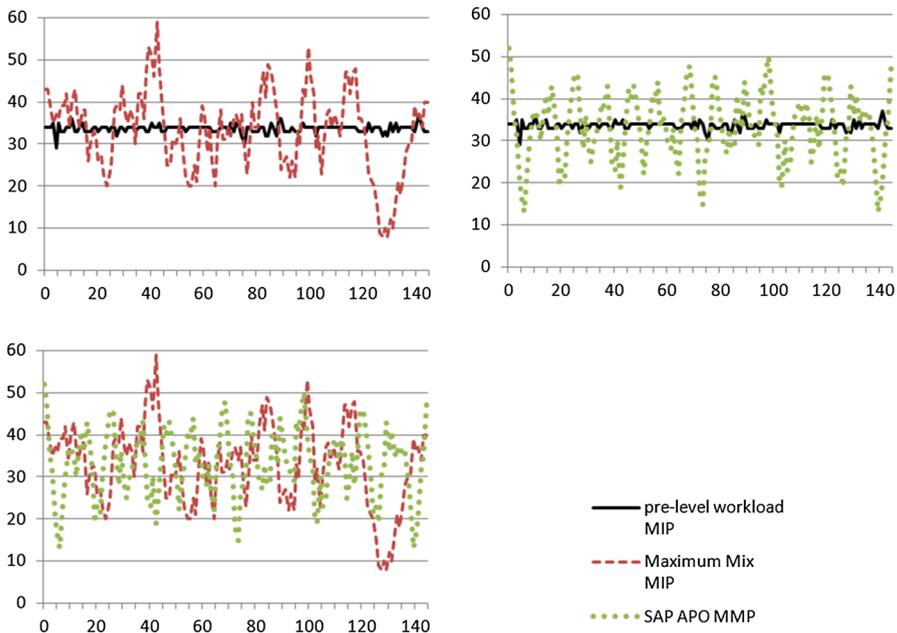


Fig. 2 Comparison of pre-level workload MIP versus Maximum Mix MIP (top left), pre-level workload MIP versus SAP APO MMP (top right) and SAP APO MMP vs. Maximum Mix MIP (bottom left). The x-axis shows sequence position t and the y-axis the actual pre-level workload for the actual and the 4 successor units ($= a_{t5}$) is shown for the best solutions of all approaches for instance 12. Target workload is given at 33.6

at sub-assembly stations in order to avoid downtime at the main line. The expected induced costs of additional workers in sub-assembly for indirectly balanced sequences compared to directly balanced was estimated with 1.5 additional full-time equivalents (FTE). These expected costs will scale up, if the mixed model approach will be rolled out to other production lines and plants.

6.2.4 Comparison of modelling options for limited pallet space

In Sect. 4.3 we presented two options to incorporate limited pallet spaces at the end of the main assembly line into the MIP model. While *Option A* relies on counting all units produced up to the respective sequence position, *Option B* uses binary variables indicating the time when the capacity of a pallet is reached. Both options were implemented in the approach balancing specific pre-level workload introduced in Sect. 4.4 as this model turned out to be superior to the maximum mix model as discussed in Sect. 6.2.1.

The instances of these test sets can be grouped into three types: simple, medium and hard instances. Five instances of each type were solved. Simple instances contain three different models ($M = 3$) and sequence positions $T = \sum_{m=1}^M d_m$ ranging from 28 to 34. Medium instances contain four different models ($M = 4$) and 35–45 sequence positions, while hard instances were defined with $M = 6$ and more than 45 sequence

positions. For all types of instances the capacity of a pallet p_m is an integer value between 5 and 10 and depends on the assigned model m . Each instance was solved for all relevant numbers of pallet positions pp , i.e. for all $pp = 1, \dots, M$. Note that these instances are too small to constitute relevant input for the general comparisons conducted in the previous Sects. 6.2.1 and 6.2.3.

The results of the experiment are shown in Table 5. It turned out that in all cases either all five instances could be solved to optimality within 1 h, or none of them. In the former case, the table gives the median solution times (over the five instances of each type) in CPU-seconds, since variations over these five instances did not exhibit any noteworthy patterns. In the latter case, the table gives the median relative gaps between the best solution found and the lower bound reported by Gurobi after 1 h of computation time.

The experiment shows that Option A outperforms Option B for the most relevant scenarios. However, Option B seems to be superior for the special case that only one pallet position is available ($pp = 1$), which in fact leads to a situation where batches of size p_m have to be produced. Also for situations where space is hardly restricted (pp close to M), Option B shows good results. Nevertheless, according to practitioners all relevant scenarios lie in between those two extremes and thus Option A should be preferred. Independently from the option it becomes clear that the practically relevant scenarios are harder to solve than the extreme situations. Note that $pp = M$ is in fact not a constraint and therefore the problem reduces to the balancing the specific pre-level workload approach without any pallet space restrictions. Therefore, we also compared the results of this experiment with the pure MIP model without any pallet space restrictions and (not surprisingly) found that the latter could be solved consuming 22–44% (resp. 13–59%) less running time compared to Option B (resp. Option A).

Despite the fact that the SAP APO MMP approach to model limited pallet spaces by means of block restrictions presented in Sect. 5.3.2 provides feasible solutions for limited pallet space, it displays less flexibility than the two MIP options compared in this section. More precisely, the approach does not adapt to the number of pallet positions pp , but in principle calculates as if only one pallet position were available ($pp = 1$).

It is obvious that pallet space restrictions limit the possibility to mix the sequence as well as to smooth the pre-level workload. Lower number of pallet positions pp induce stronger limitations. Stronger limitations lead to higher volatility of pre-level workload. As already described at the end of Sect. 6.2.3, additional leasing workers need to be temporarily employed at sub-assembly stations in order to avoid stopping the main line, if volatility of pre-level workload increases. Therefore the logistical necessities have relevant impact on the assembling costs. Investment in additional pallet positions decrease the assembly costs, but as long as $pp < M$ just the MIP formulation but not SAP APO MMP is able to gather the assembly cost saving potentials. Changing intralogistical processes after the main line to avoid pallet restrictions at all, would be the preferred option to solve that issue.

The results of our case study illustrate very well the trade-off between model mix and intralogistics, in particular the handling of the production output on different pallets. On one hand, if model mix is restricted by limited pallet space, additional workers are required on the pre-assembly lines. On the other hand, a well-balanced

Table 5 Comparison of Options A and B for limited pallet positions. Median running times (over five instances) are given in CPU-seconds

| Instance Type | Option | Pallet positions pp | | | | | |
|-----------------|--------|-----------------------|-----------|-----------|----------|---------|-------|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| Simple: $M = 3$ | A | 0.25 | 3.45 | 1.17 | - | - | - |
| | B | 0.13 | 5.03 | 1.78 | - | - | - |
| Medium: $M = 4$ | A | 18.19 | 667.43 | 115.01 | 10.38 | - | - |
| | B | 0.43 | 1029.36 | 292.87 | 10.43 | - | - |
| Hard: $M = 6$ | A | 651.78 | 30.6% gap | 24.1% gap | 660.69 | 2024.91 | 53.01 |
| | B | 0.44 | 59.5% gap | 70.2% gap | 7.6% gap | 1485.61 | 38.51 |

mixed production line obviously outputs a sequence of heterogeneous products which have to be processed in an appropriate way by intralogistics. Thus, a seemingly weakly connected company section imposes a critical bottleneck on the design of a mixed-model production line. Therefore, management has to assure that input and output of a mixed model production line are embedded in a suitable way into the overall workflow of the company.

7 Conclusions

This paper is devoted to the increasingly important aspect of assembling units of different models in one sequence of a production line in contrary to classical batch production. This allows a more flexible utilization of production resources to accommodate more varied and customized production orders posted by customers. We point out the difficulties of attaining a balanced utilization at all production levels for these mixed-model production systems, in particular under the consideration of pre-level steps, such as sub-assembly, kitting and intralogistics.

One planning strategy to indirectly achieve balanced utilization is offered in the planning environment APO of the industry standard ERP-system SAP. It aims at mixing the models on the main line to avoid blocks of identical units. We modelled this approach in SAP APO and also represented it by means of a mixed-integer program. However, both of these realizations of a maximum mix model do not directly address the underlying business goal of levelling the workload. Instead they just aim at maximizing the distance between the sequence positions of units of the same type, which in fact could lead to a situation where a model requiring the highest workload is followed e.g. by the model with the second highest workload. To overcome this shortfall we also introduced a MIP model to directly address the actual objective of balanced utilization including the workload of all pre-levels.

Furthermore, various extensions of the basic MIP model are presented in our work including: minimum distance, due dates and release dates, as well as the logistical constraint of having a limited number of pallets at the end of the line, where each pallet is assigned exclusively to the units of one model.

7.1 Real-world experience

A real-world production line was set up for mixed-model assembly usually running one shift per day. From the beginning planning was done by SAP APO MMP as described in Sect. 5.3. As can be expected from a newly installed production paradigm, the production system was also off-line for certain time periods as configurations were changed and fine-tuned. Nevertheless, more than 35,000 units (engines and gear boxes) were produced on this showcase production line in 1 year during a production time of ca. 100,000 min, i.e. roughly one unit every 3 min. A single shift (duration: 8 h or less) usually produced between 60 and 165 units, which matches the data used in our test instances. In total 18 different models were produced, but at most 15 in one shift, which is represented by the selection of $m \in [4, 15]$ in our benchmark instances.

After a certain tuning process, the mixed-model production plans derived from SAP APO MMP were perceived as highly satisfactory by the plant management, including the treatment of the limited pallet spaces as described in Sect. 5.3.2. Clearly, the embedding of the planning tool inside the existing ERP software SAP is a major convenience for the company. Unfortunately, these two aspects make the real-world implementation of a more advanced MIP-based optimization as described in Sects. 4.3 and 4.4 less likely even though management knows that assembly costs (FTE in production) could be decreased by using MIP-based optimization generated sequences (for details see Sects. 6.2.3 and 6.2.4).

In the presented case study the convenience of a solution inside the existing ERP software was crucial for the management decision to use SAP APO MMP and not to pursue a more advanced MIP-based optimization any further. The main argument against a solution outside the existing ERP software was the fact that internal IT consultants already have deep knowledge in SAP systems but no knowledge and experience with Gurobi and Python. Experience from other projects showed that quality of in-house support for the key users provided by internal IT personnel was a crucial success factor for innovative IT projects, especially for roll-outs of projects to other sites and production lines. Therefore, management preferred the more risk-averse architecture. License fee considerations played a rather minor role in this decision.

7.2 Computational experience

Computational experiments based on data from a real-world case study showed that the SAP APO realization, which is driven by a built-in (black-box) Genetic Algorithm, delivers good solutions for the maximum mix objective in a very short time, while the corresponding MIP model fails to reach an optimal solution within 1 h of computation time. This seems to be due to the low quality of the upper bounds, which still leave large gaps even after significant improvements gained from adding strengthening valid inequalities.

However, the solutions obtained by the Genetic Algorithm induce rather high variations in workload on the pre-levels. In contrast, the more refined MIP model explicitly addressing the balancing of pre-level workloads yields highly satisfactory solutions in much lower running time. Boysen et al. (2009a) criticized popular level scheduling approaches, namely product rate variation (PRV) and output rate variation (ORV), and questioned the relevance in sequencing problems in the practical sense (even though they concede its theoretical value). One of their main arguments was that literature neglects other important real world objectives like avoiding work overloads at the stations. This exactly fits with the results of our real world problem, where we saw that focusing on mixing the sequence not necessarily lead to a smooth workload over time, but focusing on the underlying workload does.

Acknowledgements Open access funding provided by University of Graz. We would like to thank *sec EDV-Beratung AG, Vienna*, and its industrial partner for the fruitful collaboration and for providing all the practical insights.

Funding This research was partially supported by the Austrian Research Promotion Agency (FFG) (Grant No. 853811) and by the Field of Excellence “COLIBRI” at the University of Graz.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Abdul Nazar KP, Pillai VM (2018) Mixed-model sequencing problem under capacity and machine idle time constraints in JIT production systems. *Comput Ind Eng* 66:226–236
- Bode W, Preuss RW (2005) Comprehensive introduction to intralogistics. STILL Academy, Wirtschaftsverlag, Suhl
- Bothe M, Nissen V (2013) SAP APO® in der Praxis: Erfahrungen mit dem Supply Chain Management-Werkzeug nutzen. Springer, Berlin
- Boysen N, Emde S (2014) Scheduling the part supply of mixed-model assembly lines in line-integrated supermarkets. *Eur J Oper Res* 239(3):820–829
- Boysen N, Fließner M, Scholl A (2009a) The product rate variation problem and its relevance in real world mixed-model assembly lines. *Eur J Oper Res* 197(2):818–824
- Boysen N, Fließner M, Scholl A (2009b) Sequencing mixed-model assembly lines: survey, classification and model critique. *Eur J Oper Res* 192(2):349–373
- Bozer YA, McGinnis LF (1992) Kitting versus line stocking: a conceptual framework and a descriptive model. *Int J Prod Econ* 28(1):1–19
- Dhamala TN, Kubiak W (2005) A brief survey of just-in-time sequencing for mixed-model systems. *Int J Oper Res* 2(2):38–47
- Dörmer J, Günther H-O, Gujjula R (2015) Master production scheduling and sequencing at mixed-model assembly lines in the automotive industry. *Flex Serv Manuf J* 27(1):1–29
- Ertel S (2014) SCM und seine Systeme: advanced planning. Igel Verlag RWS, Paderborn (in German)
- Gagné C, Gravel M, Price WL (2006) Solving real car sequencing problems with ant colony optimization. *Eur J Oper Res* 174(3):1427–1448
- García-Sabater JP, Maheut J, García-Sabater JJ (2012) A two-stage sequential planning scheme for integrated operations planning and scheduling system using MILP: the case of an engine assembler. *Flex Serv Manuf J* 24(2):171–209
- Goldberg DE (2006) Genetic algorithms. Pearson, London
- Gudehus T, Kotzab H (2012) Comprehensive logistics, 2nd edn. Springer, Berlin
- Günther H-O, Grunow M, Neuhaus U (2006) Realizing block planning concepts in make-and-pack production using MILP modelling and SAP APO. *Int J Prod Res* 44(18–19):3711–3726
- Kallrath J, Maindl TI (2006) Real optimization with SAP®. APO, Springer, Berlin
- Kubiak W (2009) Minimizing variation of production rates in just-in-time systems: a survey. *Eur J Oper Res* 66(3):259–271
- Kurbel KE (2013) Enterprise resource planning and supply chain management. Springer, Berlin
- Mabert VA, Soni A, Campbell G (2015) A case study of Pratt and Whitney aircraft’s commercial spares planning. *Commun IIMA* 6(3):2
- Meyr H (2004) Supply chain planning in the German automotive industry. *OR Spectr* 26(4):447–470
- Miltenburg J (1989) Level schedules for mixed-model assembly lines in just-in-time production systems. *Manag Sci* 35(2):192–207
- Miltenburg J, Sinnamón G (1989) Level schedules for mixed-model assembly lines in just-in-time production systems. *Int J Prod Res* 27(9):1487–1509
- Ng WC, Mak KL (1994) Sequencing of multi-level mixed-model assembly systems for just-in-time manufacturing. *Eng Optim* 23(1):57–70
- Öztürk C, Tunali S, Hnich B, Örnek MA (2013) Balancing and scheduling of flexible mixed model assembly lines. *Constraints* 18(3):434–469
- Pereira J, Vilà M (2015) An exact algorithm for the mixed-model level scheduling problem. *Int J Prod Res* 53(19):5809–5825

- Pil FK, Holweg M (2004) Linking product variety to order-fulfillment strategies. *Interfaces* 34(5):394–403
- Pinedo M (2015) *Scheduling*. Springer, Berlin
- SAP (2017a) Model mix planning. https://help.sap.com/saphelp_em70/helpdata/en/64/0ecd385e032c64e1000000a11405a/frameset.htm
- SAP (2017b) Genetic algorithm (single-lines). https://help.sap.com/saphelp_em70/helpdata/en/a2/83e666f9ca11d3b7620000e82d81c3/content.htm
- Scholl A, Becker C (2006) State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *Eur J Oper Res* 168(3):666–693
- Scholl A, Klein R, Domschke W (1998) Pattern based vocabulary building for effectively sequencing mixed-model assembly lines. *J Heuristics* 4(4):359–381
- Sivasankaran P, Shahabudeen P (2014) Literature review of assembly line balancing problems. *Int J Adv Manuf Technol* 73(9–12):1665–1694
- Stadtler H (2011) Multi-level single machine lot-sizing and scheduling with zero lead times. *Eur J Oper Res* 209(3):241–252
- Statista (2014) Market share of leading vendors for Enterprise-Resource-Planning-Software (ERP) in 2013. <https://de.statista.com/statistik/daten/studie/262342/umfrage/marktanteile-der-anbieter-von-erp-software-weltweit>
- Taube F, Minner S (2018) Resequencing mixed-model assembly lines with restoration to customer orders. *Omega* 78:99–111
- Tong K, Xu K, Zheng Y (2013) Sequencing mixed-model flexible assembly lines with variable launching intervals. *J Shanghai Jiaotong Univ (Sci)* 18:460–467
- Wörner D (2014) *Sequencing in car production based on material oriented balancing strategies*, vol 106. Audi Dissertations. Cuvillier Verlag, Göttingen (in German)
- Yano CA, Rachamadugu R (1991) Sequencing to minimize work overload in assembly lines with product options. *J Heuristics* 37(5):572–586

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Tobias Kreiter¹ · Ulrich Pferschy¹ 

Tobias Kreiter
tobias.kreiter@uni-graz.at

¹ Department of Statistics and Operations Research, University of Graz, Universitaetsstrasse 15, 8010 Graz, Austria