Friedrich-Schiller-Universität Jena
seit 1558

Jena Research Papers in Business and Economics

Sequencing Mixed-Model Assembly Lines to Minimize Part Inventory Cost

Nils Boysen, Malte Fliedner, Armin Scholl

03/2007

Jenaer Schriften zur Wirtschaftswissenschaft

Working and Discussion Paper Series
School of Economics and Business Administration
Friedrich-Schiller-University Jena

ISSN 1864-3108

Publisher:

Wirtschaftswissenschaftliche Fakultät
Friedrich-Schiller-Universität Jena
Carl-Zeiß-Str. 3, D-07743 Jena

www.wiwi.uni-jena.de

Editor:

Prof. Dr. Hans-Walter Lorenz
h.w.lorenz@wiwi.uni-jena.de

Prof. Dr. Armin Scholl
armin.scholl@wiwi.uni-jena.de

www.jbe.uni-jena.de

Sequencing Mixed-Model Assembly Lines to Minimize Part Inventory Cost

Nils Boysen[a,*], Malte Fliedner[a], Armin Scholl[b]

[a] *Universität Hamburg, Institut für Industrielles Management, Von-Melle-Park 5, D-20146 Hamburg, Germany,* {boysen,fliedner}@econ.uni-hamburg.de

[*] *Corresponding author, phone +49 40 42838-4640.*

[b] *Friedrich-Schiller-Universität Jena, Lehrstuhl für Betriebswirtschaftliche Entscheidungsanalyse, Carl-Zeiß-Straße 3, D-07743 Jena, Germany,* a.scholl@wiwi.uni-jena.de

Abstract
A mixed-model assembly line enables the joint production of different models of a common base product in intermixed model sequence (lot size one). Previous approaches for the short-term planning task of model sequencing either aim at minimizing work overload (mixed-model sequencing and car sequencing) or leveling part usages (level scheduling). However, at many manufacturers parts are consolidated by a third party logistics provider, who stocks Just-in-Time delivered parts in a consignment warehouse adjacent to the line. The manufacturer issues a complete cargo carrier (e.g. a euro-pallet) whenever his own intermediate storage of parts is depleted. Thus, the manufacturer aims at a model sequence which minimizes his own inventory costs. This paper formalizes this novel model sequencing problem and describes different heuristic and exact procedures. Furthermore, the solutions yielded by these approaches are compared to the traditional level scheduling.

Keywords: Mixed-model assembly line; Sequencing; Consignment stock; Dynamic Programming; Ant Colony Optimization;

1 Introduction

In a mixed-model assembly line the effort for setups is reduced sufficiently enough to be ignored, so that different models of a common base product can be produced in intermixed sequences (lot size one). This way, an efficient flow-line production becomes available in spite of a highly diversified product portfolio and modern production strategies like mass customization (Pine, 1993) and assembly-to-order (Mather, 1989) render possible. Nevertheless, the thorough planning of the actual production sequence is indispensable.

Due to its great practical relevance, research has aimed at supporting short-term model sequencing with suited optimization approaches since more than four decades. Numerous different approaches have been proposed, which mainly consider two general types of objectives (c.f. Bard et al., 1994):

Workload related objectives: The manufacture of varying models typically leads to variations in processing times at the stations. If several work intensive models follow each other at the same station, work overloads might occur, which need to be compensated, e.g., by additional utility workers. Thus, work overload-oriented approaches aim at model sequences, where work intensive models alternate with less laborious ones at each station (e.g. Parrello et al., 1986; Yano and Rachamadugu, 1991; Scholl et al., 1998).

Just-in-Time objectives: JIT-centric sequencing approaches focus on the deviating material requirements (e.g. Miltenburg, 1989; Kubiak and Sethi, 1991; Monden, 1998). Different models are composed of different materials and parts, so that the model sequence influences the progression of material demands over time. To facilitate the JIT-supply of required materials by preceding production stages, a steady demand rate of material over time is preferred, as otherwise the advantages of JIT are negated by large safety stocks which may become necessary to avoid stock-outs during demand peaks (e.g. Joo and Wilhelm, 1993).

None of the two objectives seems appropriate in the following situation which was inspired by a major German car manufacturer and is described in the following:

The customers are allowed to change the configuration of their ordered car until short before the actual start of production. It turned out that customers tend to exalt the interior equipment when the production date and, thus, an irreversible fixation of the product is approached. This way, the contribution margin per unit can be raised considerably such that this flexibility is highly desirable for the manufacturer. As a consequence, the line has to be configured and manned such that stations are rather over- than understaffed. Thus, work overload considerations are less relevant than optimizing the material supply.

Consequently, model sequencing has to regard the requirements of JIT part supply as practiced by wide parts of the automobile industry. Previous approaches (see above), which have their origins in the famous Toyota Production System (e.g. Monden, 1998), aim at model sequences in the final assembly which induce equally distributed part usages over time. This way, the pull mechanism provides a smooth and synchronized flow of all parts through preceding production stages closely coupled via Kanban-systems or feeder lines (e.g. Joo and Wilhelm, 1993).

However, today's trend of decreasing vertical integration diminishes the number of parts produced in-house. In our case, the car manufacturer produces only the cars' axles in-house (albeit by another manufacturer in a factory-in-factory setting). All remaining parts are produced by a range of suppliers and then consolidated by a third party logistics provider (3PL), who delivers them Just-in-Time to his own consignment stocks adjacent to the assembly line. The manufacturer issues a complete cargo carrier (e.g. a euro-pallet or special container) not until his own intermediate storage of a part directly in front of the respective station is depleted and the part is required again. Just then, parts pass over in the manufacturer's accountability and are charged to him by an online billing system. Thus, the model sequence and the induced material demands heavily influence the manufacturer's inventory, so that he aims at a model sequence which minimizes his own inventory costs caused by capital commitment. This paper formalizes this novel model sequencing problem and provides suited solution procedures.

Although our study is inspired by a single real-world case, the underlying problem setting is highly relevant in practice as the concept of consignment stock gains more and more attention throughout the whole automobile industry (see Valentini and Zavanella, 2003). However, consignment stock is obviously not limited to automobile production, but can potentially be employed in many fields of business, especially wherever the production process is organized as a mixed-model assembly line.

The remainder of the paper is structured as follows. Section 2 shortly summarizes existing research on model sequencing and consignment stocks, whereas Section 3 describes the novel problem in detail and provides a mathematical program. In Section 4 exact and heuristic solution procedures are presented, evaluated with respect to their performance and compared to the solutions of JIT-centric sequencing approaches in the computational study of Section 5. The insights and future research issues are discussed in Section 6.

2 Literature review

Since more than forty years (Wester and Kilbridge, 1964) research has tried to support model sequencing decisions with suited optimization approaches. These approaches can be divided into three general categories (see Boysen et al., 2007, for a general survey and problem classifications of pure and hybrid approaches):

- *Mixed-model sequencing:* This approach aims at avoiding/minimizing sequence-dependent work overload based on a detailed scheduling which explicitly takes into account operation times, worker movements, station borders and other operational characteristics of the line. Depending on the operational characteristics considered (e.g. open and closed stations, finite vs. infinite return velocities of workers) different optimization procedures are proposed (e.g. Macaskill, 1973; Thomopoulos, 1976; Bard et al., 1992; Bolat, 1997). Survey articles are provided by Yano and Bolat (1989) and Boysen et al. (2007).

- *Car sequencing:* To avoid the significant effort of data collection which accompanies mixed-model sequencing, car sequencing attempts to minimize sequence-dependent work overloads in an implicit manner. This is achieved by formulating a set of sequencing rules of type $H_o : N_o$, which postulate that among N_o subsequent sequence positions at most H_o occurrences of a certain option o are allowed. Minimizing the number or extent of rule violations is assumed to likewise minimizing work overload. Solution procedures originally stem from the field of constraint programming (e.g. Parrello et al., 1986; Dincbas et al., 1988) and just recently traditional optimization approaches gained wider attention (Gravel et al., 2005; Gagné et al., 2006; Fliedner and Boysen, 2006). A recent survey on car sequencing can be found in Solnon et al. (2006).

- *Level scheduling:* This modeling approach focuses on the material supply by trying to achieve a smooth part usage over time in order to facilitate Just-in-Time (JIT) delivery of all parts. Respective level scheduling procedures are provided, for instance, by Monden (1998), Miltenburg and Goldstein (1991) or Bautista et al. (1996). Assuming that all models require (almost) the same number and mix of parts (Miltenburg, 1989), it is sufficient to level the model occurrences over time (e.g. Kubiak and Sethi, 1991; Inman and Bulfin, 1991; Steiner and Yeomans, 1993). Literature reviews are provided by Kubiak (1993), Dhamala and Kubiak (2005) as well as Boysen et al. (2006).

In the automobile industry, especially level scheduling and car sequencing are known to be applied for model sequencing. For instance, Toyota employs two simple heuristic start procedures for leveling the part usages, which are known as Goal Chasing methods (see Monden, 1998). A very similar procedure is applied at South Korean car manufacturer Hyundai (see Duplaga et al., 1996). In contrast to that, French car manufacturer Renault employs a car sequencing procedure, which is extended by some special operational characteristics like paint-shop batching and hard- vs. soft-constraints (see Gagné et al., 2006; Solnon et al., 2006).

However, none of these approaches seems appropriate for adequately modeling and solving the sequencing problem when the line is supplied via consignment stock. With respect to the consignment stock concept, previous research exclusively deals with the problem of sizing minimum and maximum stock levels to be guaranteed by the supplier. Different analytical approaches to determine both variables and to conciliate conflicting interests of the supplier and OEM with respect to stock sizes are, for instance, presented by Corbett (2001) as well as Braglia and Zavanella (2003). Although the consignment stock concept gains increasing interest, especially in the automobile industry (Valentini and Zavanella, 2003), up to now, there exist no papers which deal with its operational consequences on scheduling decisions.

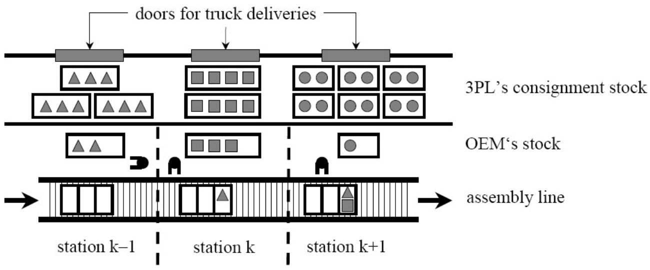

Figure 1: Schematic representation of the assembly line

3 Detailed Problem description

The part supply of the OEM's final assembly line is executed by a 3PL, who stocks all required parts (delivered from different part manufacturers) in an intermediate warehouse near the location of the OEM's plant. Here, parts are consolidated with respect to the model sequence as planned by the OEM. For instance, variants of a part for a unique product option (e.g. seats of different cloth) are sorted and arranged on cargo carriers (e.g. euro-pallet, special container or box) in the sequence they are (presumably) utilized (Just-in-Sequence part delivery). Together with unique parts (zero/one options) without variants (e.g. air-conditioning yes/no), which don't need to be sorted, they are delivered by trucks to the location of the OEM's assembly line and stocked in a consignment warehouse near the line, which is in the property of the 3PL. Directly between the consignment stock and the assembly line (typically only separated by a painted line on the floor) lies the OEM's stock of parts to be mounted onto the workpieces flowing down the line. Whenever the OEM's stock is depleted and the part is required again, a new cargo carrier is issued from the consignment warehouse (as the sequence is known in advance, the points in time when a cargo carrier is required are known as well). Financial settlement is arranged via automated self-billing, so that an invoice is generated and payments are triggered automatically, whenever a new cargo carrier is issued (e.g. Mulligan, 1998). The replenishment of the consignment stock is handled by the 3PL, who is responsible to ensure that no stock-outs occur. The schematic layout at the assembly line is depicted in Figure 1.

The OEM's management highly favors production sequences which utilize cargo carriers such that they can be depleted as soon as possible due to the following two reasons:

- As parts can only be issued in complete cargo carriers and already single parts can be costly, in-process inventory and, thus, capital commitment, is to be reduced as much as possible. Because the model sequence defines the points in time when parts are required, it directly determines the schedule of cargo carrier deliveries and the stock levels of the parts at the line. Even if it seems to be negligible that a part container might be delivered a few cycles (some minutes) too early or that a part contained in a half-full container is not required during some future cycles, the total increase of inventory cost is very considerable due to the large numbers of different parts (in our case, several thousands), stations (some hundreds) and cycles (some hundreds per shift).

- Usually, the space in front of the assembly line is very scarce whereas the number of parts is very large due to the high product variety offered to the customers. Thus, it is necessary to reduce the number of (active) cargo carriers standing alongside the line as much as possible. Currently unneeded cargo carriers serve as obstacles that can impede the production process considerably (see Klampfl et al., 2006). As cargo carriers are immediately disposed of once they are emptied and the next ones are issued not before the respective part is needed again, this frees valuable maneuvering space.

The progression of the OEM's part inventory depending on the model sequence shall be clarified by an example with the data given in Table 1. Three models $m = 1, 2, 3$ with demands d_m are produced on the line. The coefficients b_{pm} specify the number of units of two parts $p = 1, 2$ required to produce one unit of m. The parts are supplied in containers each of which contains G_p units. For storing a unit of part p during a cycle, an inventory holding (capital commitment) cost of c_p has to be paid. It is further supposed that OEM's stock is completely empty at the beginning of the shift ($S_1 = S_2 = 0$). The used notation is summarized in Table 3.

		m				
	b_{pm}	1	2	3	G_p	c_p
p	1	1	1	0	2	1
	2	1	0	1	3	1
	d_m	2	1	2		

Table 1: Example data

Table 2 displays two alternative model sequences π along with the resulting number of units l_{pt} of part p stored in each cycle $t = 1, \ldots, 5$. Whereas solution A results to an total inventory cost of 10, solution B merely amounts to a total cost of 7.

t	1	2	3	4	5	t	1	2	3	4	5
π	3	2	3	1	1	π	2	1	3	3	1
$b_{1\pi_t}$	0	1	0	1	1	$b_{1\pi_t}$	1	1	0	0	1
$b_{2\pi_t}$	1	0	1	1	1	$b_{2\pi_t}$	0	1	1	1	1
l_{1t}	0	1	1	0	1	l_{1t}	1	0	0	0	1
l_{2t}	2	2	1	0	2	l_{2t}	0	2	1	0	2
	solution A						solution B				

Table 2: Impact of the model sequence on inventory

To formalize this new Part Inventory based Mixed-Model Sequencing Problem (PIMSP), the following additional assumptions are introduced:

- We restrict our research to binary demand coefficients b_{pm} for parts, i.e., each model $m \in M$ contains one unit of part $p \in P$ ($b_{pm} = 1$) or none ($b_{pm} = 0$). This assumption is not restricting whenever each model that contains part p requires the same number q of units of p. Even if $q > 1$, the binary approach is sufficient, because we may build bundles each of which contains q units of p. The capacity G_p of the carrier is divided by q (G_p should be an integer multiple of q) and the cost factor c_p, multiplied by q to relate these parameters to the bundles instead of single units. Due to this transformation the above restriction to binary requirement coefficients is fulfilled in many real-world problem settings. In particular, this is usually the case in automobile and related industries (see Monden, 1998). Furthermore, this assumption has no impact on the model formulation but simplifies the bound computation in Section 4.2.

- The OEM issues a cargo carrier for part p as soon as a model copy requires a unit of p and his own inventory, i.e., the current carrier for p, is already empty. Notice that, due to the sequence being known in advance, it is not difficult to match this point in time exactly. If a model copy in cycle t requires a part p and a cargo carrier with G_p units of part p is issued from consignment stock, it is immediately accessible at the beginning of cycle t, so that $G_p - 1$ units remain in the part inventory in cycle t.

- We only consider parts which are issued in part-specific cargo carriers of size $G_p > 1$, which turns out to be the vast majority of parts in real-world applications. Parts which are made available unit-by-unit directly from consignment stock have no impact on the OEM's inventory and are, thus, omitted.

- The consignment stock of the 3PL is supposed to be no bottleneck and always ready for deliveries.
- As usual or even necessary in practice due to organizational considerations and space restrictions, each cargo carrier is exclusively assigned to one station where the respective part is required. Due to this unique assignment, it is not necessary to model stations explicitly, as each station has a constant offset compared to the beginning of the line. This offset only depends on line speed and the stations' positions. It has no impact on the units to be stored during the planning horizon and, thus, no influence on inventory cost.
- At some stations, different variants of a basic part p (e.g. differently colored seats, electrical or manual sunroof) are assembled. Due to their large variety and/or value, these parts must be provided Just-in-Sequence by the 3PL. Then, the carrier is dimensioned to accommodate G_p units whatever variants are contained in the sequence. In this usual case, the variants need not be distinguished and can be unified to a single part.

Under these assumptions, PIMSP can be formalized with the notation listed in Table 3.

P	set of parts (index p)
T	number of production cycles (index t)
M	set of models (index m)
b_{pm}	binary demand coefficient: 1, part p is needed for the production of a model m; 0, otherwise
d_m	demand of model m
c_p	inventory holding cost for storing a unit of part p during a cycle
G_p	capacity of the cargo carrier for part p
S_p	quantity of part p initially stored in OEM's stock
y_{pt}	integer variable: number of cargo carriers for part p taken from consignment stock up to cycle t
x_{mt}	integer variable: number of scheduled copies of model m up to cycle t
l_{pt}	continuous variable: quantity of part p stored during cycle t in the OEM's stock

Table 3: Notation

The quantity l_{pt} stored for part p during a cycle t can be calculated by determining the total number y_{pt} of issued cargo carriers of part p up to cycle t and the cumulative part usage, which in turn depends on the total number of scheduled model copies x_{mt} over all types m assigned up to cycle t. This leads to the following mathematical program with objective function (1) and constraints (2)-(8):

$$
\begin{aligned}
&\text{(PIMSP) Minimize } C(X,Y,L) = && \textstyle\sum_{p\in P}(c_p \cdot \sum_{t=1}^{T} l_{pt}) && (1)\\
&\sum_{m\in M} x_{mt} = t && \forall\, t = 1,\dots,T && (2)\\
&x_{mT} = d_m && \forall\, m \in M && (3)\\
&\sum_{m\in M} x_{mt}\cdot b_{pm} + l_{pt} = y_{pt}\cdot G_p + S_p && \forall\, p \in P;\; t = 1,\dots,T && (4)\\
&0 \le x_{mt} - x_{mt-1} \le 1 && \forall\, m \in M;\; t = 2,\dots,T && (5)\\
&0 \le y_{pt} - y_{pt-1} \le 1 && \forall\, p \in P;\; t = 2,\dots,T && (6)\\
&y_{pt} \in \mathbb{N}^0;\; l_{pt} \ge 0 && \forall\, p \in P;\; t = 1,\dots,T && (7)\\
&x_{mt} \in \mathbb{N}^0 && \forall\, m \in M;\; t = 1,\dots,T && (8)
\end{aligned}
$$

The objective function (1) minimizes the total cost of inventory summing up the quantities l_{pt} of all parts p stored in all cycles t each of which is weighted with the part-specific inventory holding cost factor c_p. Constraints (2) and (5) ensure that in each cycle t exactly one model copy is produced, whereas equalities (3) enforce that the demand d_m of model m is met during the planning horizon, i.e., d_m cycles are assigned to model m. The balance equations (4) define the quantity l_{pt} stored per part p and cycle t as the difference between the overall number of issued units (number of issued carriers y_{pt} times carrier size G_p) plus initial stock S_p and the cumulative consumption of the part through previously scheduled model copies. Constraints (6) enforce the integer variables y_{pt} to monotonically increase over time.

4 Algorithms

In the following, we present suited algorithms for this model sequencing problem. First, a Dynamic Programming approach is presented, which is then enhanced by a lower bounding procedure to a so called "Bounded Dynamic Programming" procedure. Then, a simple heuristic start procedure is provided, which is an adoption of the famous Goal Chasing methods applied to level scheduling. Finally, a meta-heuristic Ant Colony approach is described.

4.1 Dynamic Programming approach

The Dynamic Programming (DP) approach is based on an acyclic digraph $G = (V, E, r)$ with a node set V divided into $T + 1$ *stages*, a set E of arcs connecting nodes of adjacent stages and a node weighting function $r : V \to \mathbb{R}$ (see Bautista et al., 1996, for a similar approach to level scheduling). Each sequence position t is represented by a stage which contains a subset $V_t \subset V$ of nodes representing *states* of the production system in cycle t. Additionally, a start level 0 is introduced. Each index $i \in V_t$ identifies a state (t, i) defined by the vector $\mathbf{X}_{ti}$ of cumulated quantities X_{tim} of all models $m \in M$ produced up to cycle t. It is sufficient to store the cumulated quantities instead of the partial sequence up to cycle t, because the inventory cost of cycle $t+1$ only depends on the given initial inventories S_p, the cumulated production quantities X_{tim} and the model produced in $t + 1$.

The following conditions define all *feasible* states to be represented as nodes of the graph:

$$
\begin{aligned}
&\sum_{m\in M} X_{tim} = t && \forall\, t = 0,\dots,T;\; i \in V_t && (9)\\
&0 \le X_{tim} \le d_m && \forall\, m \in M;\; t = 0,\dots,T;\; i \in V_t && (10)
\end{aligned}
$$

Obviously, the node set V_0 contains only a single node (initial state $(0, 1)$) corresponding to the vector $\mathbf{X}_{01} = [0, 0, \dots, 0]$. Similarly, the node set V_T contains a single node (final state $(T, 1)$)

with $\mathbf{X}_{T1} = [d_1, d_2, \ldots, d_{|M|}]$. The remaining stages have a variable number of nodes depending on the number of different model vectors $\mathbf{X}_{ti}$ possible.

The produced quantities of all models up to cycle t in a state (t,i) directly determine the cumulative demands D_{tip} for all parts p:

$$D_{tip} = \sum_{m \in M} X_{tim} \cdot b_{pm} \qquad \forall\, p \in P \tag{11}$$

The inventories I_{tip} of the parts $p \in P$ during a cycle t in state (t,i) are easily derived by (12), because they are either units from initial stock S_p not consumed by cumulated demand D_{tip} or residual units out of newly issued cargo carriers of size G_p. The special case $I_{tip} = 0$ arises when the carrier has been emptied at the beginning of t or was already empty and no unit of p has been required in cycle t.

$$I_{tip} = \begin{cases} S_p - D_{tip}, & \text{if } S_p \geq D_{tip} \\ 0, & \text{else if } (D_{tip} - S_p) \bmod G_p = 0 \\ G_p - (D_{tip} - S_p) \bmod G_p, & \text{otherwise} \end{cases} \qquad \forall\, p \in P \tag{12}$$

Because the state (t,i) directly determines the quantities stored for each part $p \in P$, the corresponding node can be assigned a unique node weight r_{ti} equal to the inventory holding cost during cycle t as follows:

$$r_{ti} = \sum_{p \in P} c_p \cdot I_{tip} \qquad \forall\, t = 0, \ldots, T;\; i \in V_t \tag{13}$$

Two nodes (t,i) and $(t+1,j)$ of two consecutive stages t and $t+1$ are connected by an arc if the associated vectors $\mathbf{X}_{ti}$ and $\mathbf{X}_{t+1j}$ differ only in one element, i.e., a copy of exactly one model is additionally produced in cycle $t+1$. This is true if $X_{tim} \leq X_{t+1jm}$ holds for all $m \in M$, because both states are feasible according to (9) and (10). The overall arc set is defined as follows:

$$E = \{((t,i),(t+1,j)) \mid t = 0, \ldots, T-1; i \in V_t; j \in V_{t+1} \text{ and } X_{tim} \leq X_{t+1jm}\ \forall m \in M\} \tag{14}$$

With this graph on hand, the optimal solution of our model sequencing problem reduces to finding the shortest path from the unique source node at level 0 to the unique sink node at level T, where the length of the path is given by the sum of weights of the nodes contained. The length of the shortest path is equal to the minimal total inventory cost. The corresponding model sequence π can be deduced by considering each arc $((t,i),(t+1,j))$ with $t = 0, ..., T-1$ on the shortest path SP. The model to be assigned at sequence position $t+1$ is the only one for which $X_{t+1jm} - X_{tim} = 1$ holds.

The resulting graph along with a (bold-faced) shortest path for our example is depicted in Figure 2. This path corresponds to the optimal model sequence $\pi = \{3,3,1,2,1\}$ with minimal total inventory cost $C = 7$. A second optimal sequence is $\pi = \{2,1,3,3,1\}$.

Instead of constructing the complete graph before computing the shortest path, the more efficient DP approach consists of determining the shortest path from the initial state to each node stage-by-stage ($t = 0, ..., T-1$). In order to do so, only two stages of the graph have to be stored simultaneously, because the shortest path to a node $(t+1,j)$ in stage $t+1$ is composed of a shortest path to a node (t,i) in stage t (already determined and stored) and the connecting arc $((t,i),(t+1,j))$. Among all such paths to $(t+1,j)$ one with minimal sum of node weights (length of path to (t,i) plus $r_{t+1,j}$) is to be selected. The length-minimizing node (t,i) is stored as the predecessor in the shortest path to $(t+1,j)$ together with the length of this path. After reaching the final state $(T,1)$ in stage T, the optimal path can be retrieved in backward direction stage-by-stage using the stored predecessor nodes.

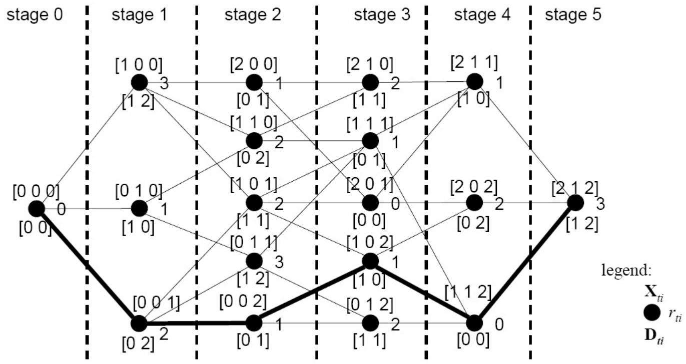

Figure 2: Example graph of the Dynamic Programming procedure

Remark: This graph structure and the DP approach (with modified node weights) can be applied to any model sequencing problem whose contribution of the sequencing decision at a level t only depends on the production quantities of models up to level $t-1$ irrespective of their exact order. This is given for level scheduling (see Bautista et al., 2006), but not for mixed-model sequencing and car sequencing.

4.2 Bounded Dynamic Programming

Although, the number of nodes to be generated is considerably reduced compared to a direct assignment of individual model copies to sequence positions (e.g. in a model-oriented branching scheme, see Fliedner and Boysen, 2006 as well as Drexl et al., 2006 for car sequencing) it will be too large for problem instances with plenty production cycles T and models $|M|$. Thus, to further reduce the number of nodes we employ the idea of Bounded Dynamic Programming (BDP) (e.g. Morin and Marsten, 1976; Marsten and Morin, 1978; Carraway and Schmidt, 1991; Bautista et al., 1996).

BDP extends the DP approach explained above by additionally computing a lower bound $LB(t,i)$ on the path lengths from a node (t,i), considered while constructing the graph stage-by-stage, to the sink node $(T,1)$. Furthermore, a global upper bound is determined upfront by some heuristic procedure(s). Let $FI(t,i)$ be the already fixed inventory cost (= minimum path length from the source node to (t,i)). Whenever $LB(t,i)+FI(t,i) \geq UB$, the node (t,i) can be fathomed as it can not be part of a solution with a better objective value than the incumbent solution.

For the lower bound computation, the remaining problem (cycles $t+1$ to T) in state (t,i) is decomposed in $|P|$ subproblems by cutting off the model coherency of parts. For each separate part the minimum inventory cost induced by the remaining demand $R_{tip} = \sum_{m \in M} d_m \cdot b_{pm} - D_{tip}$ and the current inventory I_{tip} is determined.

The optimal solution for each subproblem can be readily determined by considering the following consumption pattern: A cargo carrier once taken from consignment stock is to be emptied a soon as possible. Thus, a carrier of size G_p results to the least possible inventory cost, if G_p models, which require part p, directly follow each other starting from cycle t when the carrier arrives at the line. It also follows that the current leftover inventory $I_{tip} > 0$ is to be consumed as soon as possible in direct succession. Moreover, a cargo carrier, which can not be completely emptied because the remaining demand R_{tip} is not sufficient, is to be issued as late as possible.

In order to derive a lower bound Z_{tip} on the total inventory of part p during the cycles $t+1$ to T for a node (t,i) the following two cases can be distinguished:

- $I_{tip} > R_{tip}$: In this case, the remaining demand R_{tip} can be completely satisfied from the current inventory I_{tip}. Thus, no further cargo carriers are needed for part p until the end of

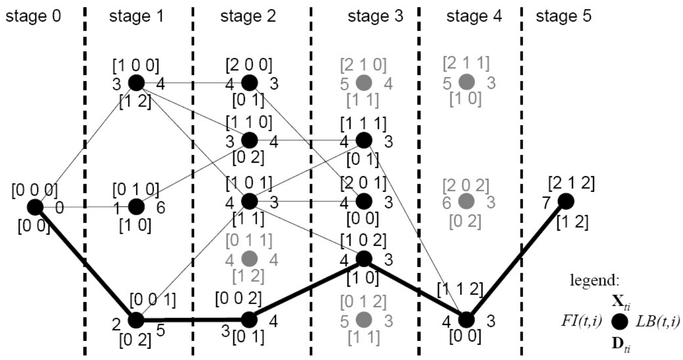

Figure 3: Example graph for BDP

the planning horizon. The available units are consumed in the following R_{tip} cycles thereby reducing the inventory by one unit in each cycle (first term). In the remaining periods the leftover quantity $I_{tip} - R_{tip}$ has to be stored (second term).

$$Z_{tip} = \sum_{\tau=1}^{R_{tip}} (I_{tip} - \tau) + (T - t - R_{tip}) \cdot (I_{tip} - R_{tip}) \tag{15}$$

- $I_{tip} \leq R_{tip}$: In the other case, the current inventory I_{tip} is not sufficient to meet the remaining demand R_{tip} and further cargo carriers are to be issued:

$$Z_{tip} = \sum_{\tau=1}^{I_{tip}-1} \tau + \left\lfloor \frac{R_{tip} - I_{tip}}{G_p} \right\rfloor \cdot \sum_{\tau=1}^{G_p-1} \tau + \sum_{\tau=1}^{(R_{tip}-I_{tip}) \bmod G_p} (G_p - \tau) \tag{16}$$

  Here, Z_{tip} results from actual inventory I_{tip} (first term) and $\left\lfloor \frac{R_{tip}-I_{tip}}{G_p} \right\rfloor$ cargo carriers (second term), which are all consumed as fast as possible. The final carrier, which is not completely emptied by material demand, is issued as late as possible (third term).

With these formulas on hand, the lower bound for the minimum path length from the current node (t, i) to the sink node amounts to the sum of the minimum inventory cost over all parts:

$$LB(t, i) = \sum_{p \in P} c_p \cdot Z_{tip} \qquad \forall\, (t, i) \in V \tag{17}$$

Example: Consider the node $(3, 1)$ in Figure 3. For part 1, $I_{311} = 1$ units are stored and the remaining demand is $R_{311} = 0$. Due to $I_{311} > R_{311}$ equation (15) is applied: $Z_{311} = 0 + (5 - 3 - 0) \cdot (1 - 0) = 2$. For the second part with $I_{312} = 1$ and $R_{312} = 2$, the second case holds: $Z_{312} = 0 + 0 \cdot (2 + 1) + (3 - 1) = 2$. Thus, the lower bound amounts to $LB(3, 1) = 4$. Assume that an upper bound of $UB = 9$ has been calculated by some heuristic procedure prior to generating the graph. In this case, node $(3, 1)$ can be fathomed, because the sum of lower bound $LB(3, 1) = 4$ and already fixed inventory cost on the shortest path to node $(3, 1)$ of $FI(3, 1) = 5$ is not lower than the upper bound. The overall reduction in the number of nodes resulting from the incorporation of lower bounds is depicted in Figure 3 for our example and an upper bound of $UB = 8$. Fathomed nodes are colored light grey.

4.3 Heuristic start procedure

Very famous construction procedures for model sequencing are the so called Goal Chasing methods, which are employed at car manufacturer Toyota to level the part usages over time (see

Monden, 1998). Goal Chasing method 1 is a very simple myopic heuristic procedure to derive a first start solution. The method simply fills the solution vector π from left to right by fixing that model with remaining demand at the current position, which increases the objective value the least (as Goal Chasing methods are applied to a minimization problem).

To derive a Goal Chasing method (GC) for our problem we just have to account for the modified objective function. At each decision point t only the set of possible alternatives POS_t is relevant, which includes all models m whose demand is not satisfied by preceding sequencing decisions:

$$POS_t = \{m \in M \mid x_{mt} < d_m\} \tag{18}$$

Then, for each model $m \in POS_t$ a priority value $f(t,m)$ has to be determined, in analogy to the calculation of node weights in the BDP. $D_p(t,m) = \sum_{\tau=1}^{t-1} b_{p\pi_\tau} + b_{pm}$ denotes the cumulative demand for parts units of type p provided that model $m \in POS_t$ is assigned to the current decision point t:

$$f(t,m) = \sum_{p \in P} c_p \cdot \begin{cases} (S_p - D_p(t,m)), & \text{if } S_p \geq D_p(t,m) \\ 0, & \text{else if } (D_p(t,m) - S_p) \bmod G_p = 0 \\ (G_p - (D_p(t,m) - S_p)) \bmod G_p, & \text{otherwise} \end{cases} \tag{19}$$

Finally, with these priority values on hand, a greedy choice assigns the best model available to the sequencing position t:

$$\pi_t = \operatorname{argmin}_{m \in POS_t}\{f(t,m)\} \tag{20}$$

Then, t is incremented and choices are repeated until model vector π is completely filled. In the DP-graph, GC equals a follow-up of that respective arc which leads to a connected node with the least node weight r_{ti}, in each stage. For our example, GC constructs the solution vector $\pi = \{2, 1, 3, 3, 1\}$ with total cost $C = 7$. Thus, the solution is optimal in this simple case.

4.4 Ant Colony approach

As a compromise between generating all paths through the graph (DP approach) and inspecting just a single one (GC), a meta-heuristic can be used in order to guide the search to a subset of promising paths. In the following, such an Ant Colony approach (e.g. see Dorigo et al., 1999) is introduced. In an Ant Colony approach, solutions are constructed repetitively by software agents (artificial ants), which typically base their decisions on some local heuristic measure and the collected experiences of all former ants, aggregated in a so called pheromone matrix. The search process of an individual ant resembles the GC method, such that at each sequence position t a single task is chosen out of the set POS_t of possible alternatives (models with remaining demand). An ant's sequence π is hence filled from left to right. However, the choices of an ant are not deterministic, but stochastic according to a weighted probability scheme which is repetitively calculated at each decision point (sequencing position).

The probability $P(t,m)$ that a copy of model m is assigned to position t is then determined on the basis of its priority value $f(t,m)$, which is already used in the GC method, and the intensity of the pheromone τ_{mt} with respect to its alternatives:

$$P(t,m) = \frac{{\tau_{mt}}^{\alpha} \cdot \left(\frac{1}{1+f(t,m)}\right)^{\beta}}{\sum_{k \in POS_t} {\tau_{kt}}^{\alpha} \cdot \left(\frac{1}{1+f(t,k)}\right)^{\beta}} \qquad \forall\, t = 1, \ldots, T;\ m \in POS_t \tag{21}$$

The priority value $f(t,m)$ has to be incremented by some positive constant c (here $c = 1$ is chosen) as at certain decision points models are possible, which do not evoke additional inventory costs ($f(t,m) = 0$). Parameters α and β control the relative importance of the pheromone versus the priority values. Because of experiences with other sequencing problems reported in the literature, these parameters are set to $\alpha = 1$ and $\beta = 2$ (see Stützle and Dorigo, 1999).

After assigning all tasks to sequence positions, the objective value $C(\pi)$ of a sequence π is easily calculated by summing up the priority values of chosen models: $C(\pi) = \sum_{t=1}^{T} f(t, \pi_t)$ like it is done in the GC method. In any iteration, several ants generate and evaluate solutions in the fashion described above. The ant with the best objective value $C(\pi)$ is then selected for updating the pheromone trail. The pheromone value $\tau_{mt}(k)$ in iteration k is calculated as follows:[1]

$$\tau_{mt}(k) = \tau_{mt}(k-1) \cdot (1-\rho) + \rho \cdot \begin{cases} \frac{1}{C(\pi)}, \text{ if } \pi_t = m \\ 0, \text{ otherwise} \end{cases} \quad \forall\, t = 1, \ldots, T;\, m \in M \qquad (22)$$

The formula incorporates two mechanisms for guiding the search. The older pheromone is constantly reduced (evaporation) which strengthens the influence of more recent solutions and new pheromone is assigned to all task-position assignments, which are part of the solution, in proportion to the respective objective value. The parameter ρ controls the relative importance of these two components. In the current implementation ρ is set to 0.5 and 40 ants are employed to construct solutions in any iteration. After 200 iterations the algorithm terminates and the best solution found is returned.

5 Computational study

The approaches presented in this paper explore a new area in model sequencing, hence, no established test-bed is available. Therefore, we first elaborate on the instances that are used in our computational study. Second, numerical results on the performance of algorithms are presented and, finally, solutions obtained by our approach are compared to traditional level scheduling.

5.1 Instance generation

In our computational study, we distinguish between three classes of test instances: small, medium and big instances. The small instances are designed such that our BDP approach can still solve all test instances to optimality (in acceptable time). Medium sized instances are used to explore the limit of problem size up to which the BDP approach is able to solve instances to optimality. Finally, big instances shall represent problem instances of a size relevant in real world settings. Furthermore, the parameters listed in Table 4 are used to generate the demand coefficients for parts b_{pm}, model demands d_m, sizes of cargo carriers G_p and weighting factors c_p which define a PIMSP-instance.

Within each test case, these parameters a combined in a full-factorial design, so that $3 \cdot 486 = 1458$ different PIMSP-instances were obtained. On the basis of a given set of parameters each single instance is generated as follows:

- *Demand coefficient matrix:* For each individual demand coefficient b_{pm} a continuous random number rnd out of the interval $[0, 1]$ is drawn and compared to the probability $PROB$ of a model containing the respective part, so that coefficients can be fixed with regard to the following formula:

$$b_{pm} = \begin{cases} 1, \text{ if } rnd \leq PROB \\ 0, \text{ otherwise} \end{cases} \quad \forall\, m \in M;\, p \in P \qquad (23)$$

- *Demand for models:* At first, each model demand d_m is initialized to one unit. Then, demands of randomly drawn (equally distributed) models are increased by one unit, until the overall model demand ($\sum_{m \in M} d_m$) equals the given number of production cycles T.

[1] The pheromone matrix has to be initialized with starting values $\tau_{mt}(0) = \frac{1}{C(\pi^{start})}$, where π^{start} represents a first feasible solution, which is randomly determined.

symbol	description	values small	medium	big
T	number of production cycles	10, 15, 20	25, 30, 35	100, 200, 300
$\|M\|$	number of models	5, 7, 9		15, 30, 45
$\|P\|$	number of parts	5, 7, 9		10, 15, 20
–	equal ($c_p = 1$) versus diverging weighting factors ($c_p \in [1, 5]$)			
MSC	maximum size of cargo carrier (in part units)		3, 5, 7	
$PROB$	probability of a model m containing part p		0.3, 0.5, 0.7	

Table 4: Parameters for instance generation

measure	BDP	LB	GC	ANT
number of optimal solutions (objective values)	486	100	89	323
average relative deviation from optimum in %	0	8.15	13.03	1.03
maximum relative deviation from optimum in %	0	45.0	123.08	11.2
average absolute deviation from optimum	0	12.4	17.3	1.9
maximum absolute deviation from optimum	0	136	246	44
average CPU-seconds	1.47	<0.1	<0.1	0.53

Table 5: Results aggregated over all small instances

- *Size of cargo carrier:* The number of units G_p of parts p to be stocked on a cargo carrier are randomly drawn by an equally distributed integer random number out of the interval $[2, MSC]$.
- *Weighting factor:* In instances, which only consist of equally weighted parts all weighting factors c_p are fixed to 1, whereas in the other instances weights are randomly drawn by an equally distributed integer random number out of the interval $[1, 5]$.

All generated instances can be downloaded from the internet (www.assembly-line-balancing.de).

5.2 Performance of algorithms

The methods described above have been implemented in Visual Basic.NET (Visual Studio 2003) and run on a Pentium IV, 1800 MHz PC, with 512 MB of memory. The aggregated results over all small test instances are reported in Table 5 for the four procedures: BDP = Bounded Dynamic Programming, LB = Lower Bound procedure, GC = Goal Chasing method, and ANT = Ant Colony approach. As the initial upper bound for the BDP the objective value obtained by GC is used.

As it shows, our simple lower bound procedure performs satisfactorily as it equals the optimal objective value C^* in 100 instances and produces an average relative deviation of 8.15% (note that the single deviations are computed by $\frac{C^*-LB}{C^*}$). Among the heuristic procedures ANT clearly outperforms GC with respect to the solution quality (based on relative deviations $\frac{UB-C^*}{C^*}$ with UB being the objective function value of the heuristic considered).

More detailed results of the algorithmic performance depending on the parameters of instance generation are presented in Figure 4. We restrict our analysis to the small test instances as it is the only class, where all instances are solved to optimality. This analysis suggests the following conclusions with respect to the dependency between performance (measured in average relative deviation from optimum = ⊘ rel. dev.) and parameters:

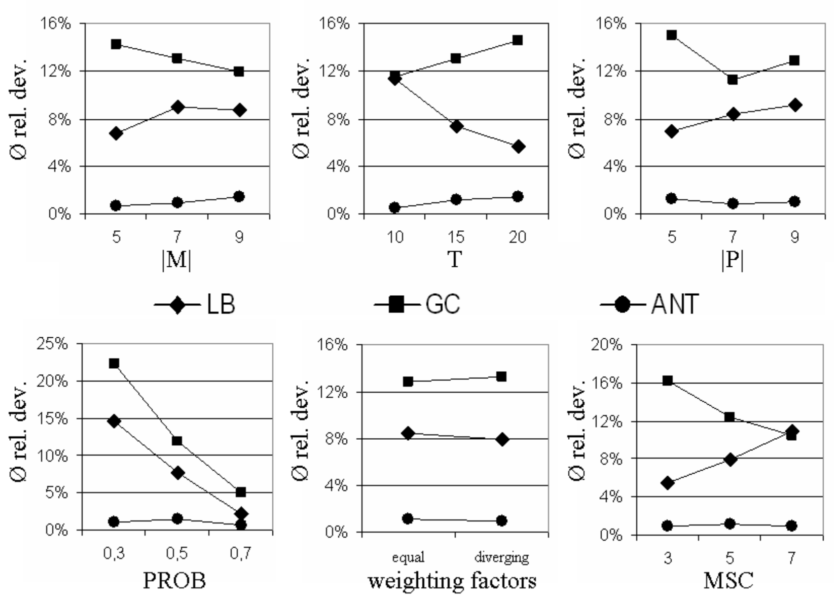

Figure 4: Algorithmic performance per parameter of instance generation (small instances)

- The ANT approach turns out to be comparatively robust as the algorithmic performance shows only minor deviations in any parameter constellation.
- The number of models $|M|$ and parts $|P|$ as well as the homogeneity of the weighting factors have only minor influence on the performances.
- An increase in $PROB$ yields a general increase in performance, as different models are likely to be more similar with regard to part consumption and thus different model sequences lead to comparable objective values.
- The higher MSC, the higher is the penalty value if parts are not immediately retrieved from stock at a given point in time. Consequently, the greedy approach of GC performs better in tendency, as models, which cause high inventory costs, receive a higher penalty and can be clearly distinguished from less costly models. In contrast to that the LB values are likely to deviate more, whenever ideal consumption patterns cannot be realized.
- Interestingly, an increase in T leads to the opposite performance progression. The more assignment slots become available, the closer part consumption can be guided towards the ideal pattern presumed by LB. The results of GC gradually decrease, the higher the combinatorial complexity.

In the following, the question is investigated, up to which problem size the BDP approach can be reasonably applied to obtain optimal solutions. To this end, Table 6 summarizes the results for medium sized test instances. Here, the number of instances solved to optimality within a restricted time frame of 300 CPU-seconds is reported for the parameters: number of cycles T and models $|M|$. Additionally, the number of instances is indicated for which the optimal solution is proven by the equality of the initial lower bound (LB) and the initial upper bound (UB) of the heuristic GC method. By comparing both indicators, it can be deduced how often optimal solutions actually originate from the BDP approach and how often the optimum was proven beforehand without the BDP approach being started. As expected, the number of optimal solutions found by BDP decreases with increasing number of cycles T and models $|M|$. With $T = 35$ cycles and $|M| = 9$ different models none of 54 instances was actually solved by the BDP within the given time frame, so that an upper limit on instance size is probably reached.

$\|M\|$	T 25	30	35	Total
5	54/21	54/28	54/19	162/68
7	54/16	54/19	23/21	131/56
9	35/24	25/24	19/19	79/67
Total	143/61	133/71	96/59	372/191

legend: # optimal solutions/ # instances with initial LB=UB

Table 6: Number of medium instances solved by BDP

measure	GC	ANT
average relative deviation from LB in %	26.68/22.72/24.99	16.28/9.37/11.24
maximum relative deviation from LB in %	148.74/120.31/176.19	69.43/70.37/81.67
average CPU-seconds	<0.1/<0.1/<0.1	40.92/1.09/0.53

legend: big/medium/small instances

Table 7: Performance of heuristic procedures compared to LB

Finally, the suitability of the heuristic approaches for problem instances of real-world size are to be investigated. As optimal solutions could not be obtained, Table 7 compares the objective values with the lower bound LB. These results highlight the superiority of ANT over GC with respect to the solution quality. When comparing these results with those of small and medium sized instances, it can be stated that deviations remain in the same order of magnitude. Thus, deviations from the optimal solution values should also remain in the same range, so that the suitability especially of the ANT approach for problems in real-world size seems to be given. This statement is not restricted by the fact that ANT requires much more computation time since the problem should be important enough to spend some minutes of computation time.

5.3 Comparison to level scheduling

This section addresses the question of whether existing part-oriented approaches already lead to satisfying solutions for our problem setting or whether the newly developed procedures are better suited.

Previous research on level scheduling aims at a leveling of part usages over time (see Section 2). For this purpose each part receives a target demand rate r_p per production cycle, actual part usages are to be adjusted on:

$$r_p = \frac{\sum_{m\in M} d_m \cdot b_{pm}}{T} \qquad \forall\, p \in P \tag{24}$$

With respect to these demand rates a mathematical program, which Kubiak (1993) labels as the *ORV* (output [of preceding stages] rate variation problem), can be formulated as follows (Joo and Wilhelm, 1993; Monden, 1998; Bautista et al., 1996):

$$\text{Minimize } Z^{ORV}(X) = \sum_{t=1}^{T} \sum_{p\in P} c_p \cdot \left(\sum_{m\in M} x_{mt} \cdot b_{pm} - t \cdot r_p \right)^2 \tag{25}$$

subject to (2), (3), (5) and (8)

The objective function (25) penalizes squared deviations of actual from ideal cumulative part demands weighted by the part-specific factor c_p and aggregated over all cycles t and parts p.

In practical applications, where products may consist of thousands of different parts, the resulting problem instances of the *ORV* are barely solvable to optimality. Accordingly, the

measure	ORV	PRV	RND
number of optimal solutions	0	0	0
average relative deviation from optimum in %	89.31	84.37	80.34
maximum relative deviation from optimum in %	553.85	546.15	654.62
average absolute deviation from optimum	115.06	109.50	104.99
maximum absolute deviation from optimum	643	644	772

Table 8: Comparison to level scheduling approaches (small instances)

ORV is approximated by a simplified approach, which levels the production rates of the models over time. The objective of these model-oriented level scheduling problems, which Kubiak (1993) labels as PRV (product rate variation problem), is to approach the ideal production rate $r_m = d_m/T$ for each model $m \in M$ as close as possible. Thus, the objective (25) is replaced by the new objective function (26) (e.g. Miltenburg, 1989):

$$\text{Minimize } Z^{PRV}(X) = \sum_{t=1}^{T} \sum_{m \in M} (x_{mt} - t \cdot r_m)^2 \tag{26}$$

subject to (2), (3), (5) and (8)

To answer the research question stated above, a computational study is carried out in which the results obtained by level scheduling (ORV and PRV) are compared to those of our new model. For this purpose, the small instances of Section 5.1 are solved to optimality with respect to the models ORV and PRV. The ORV is solved by the algorithm of Bautista et al. (1996), whereas the PRV is solved by the assignment procedure of Kubiak and Sethi (1991). The model sequences obtained by these two approaches are then evaluated by the objective function (1) of the new model sequencing approach.

The results of this experiment are listed in Table 7. In addition to the average relative deviations of the two level scheduling approaches (ORV and PRV) from the optimal total cost value, the average deviations of, in each case, 10 randomly generated model sequences (RND) are reported. The results make clear, that level scheduling is not at all appropriate for solving PIMSP-type problems, as the performance even drops behind that of a random sequence. This result stems from the fact, that a leveling implies a spreading of part usages over time, which directly opposes to the accumulation of part usages required in the real-world situations considered by PIMSP.

6 Conclusions

This paper on hand presents a new problem setting for sequencing mixed-model assembly lines, where the OEM's final assembly line is supplied from a nearby consignment stock and the OEM aims at minimizing his own inventory directly in front of the stations. A mathematical program is provided to formalize this problem and a lower bounding procedure along with a Bounded Dynamic Programming algorithm, a simple heuristic start procedure and a meta-heuristic is presented. The computational study evaluates the applicability of these approaches and underlines the inapplicability of previous material-centric advancement of level scheduling. Nonetheless, there remain some future research issues:

- Although the authors consider the NP-hardness of the problem to be very likely, the complexity of the problem remains an open question.

- Non zero/one demand coefficients for parts ($b_{mp} > 1$) are to be investigated for cases where models require different numbers of the same part (e.g. printed circuit boards, see Cakir and Inman, 1993). In this case, the mathematical program, the adapted Goal Chasing method, and the Ant Colony approach can be applied as described, whereas the bound computation is to be modified.

- Due to high capital investments a mixed-model assembly line is often highly utilized, so that workers from consecutive shifts take over work on the fly and production runs continuously over time. In this case, the continuous sequencing problem has to be broken down in virtually separated problems. This raises the question of whether the concatenation of separately optimal sequences leads to an overall optimal solution or loses optimality. This so called cyclic scheduling is, for instance, an active field of research in level scheduling (see Kubiak, 2003; Brauner and Crama, 2004) but remains an open field for our new problem.
- In practice, the space available for storing parts is usually very scarce. This is partly considered in the PIMSP-approach by delivering cargo carriers to the line not before they are required. However, space restrictions might be even harder. Then, additional constraints restricting the required space per cycle to the amount available should be added. This will raise the additional question whether or not a feasible solution exists thereby complicating to find feasible solutions with heuristic methods. The model can easily be extended to consider space restrictions.

Moreover, the new approach questions the onesided orientation on level scheduling in the field part-centric model sequencing. Parts can be delivered to the line in very different ways. The approach of leveling part usages to enable a smooth flow of parts through a pull-steered supply chain seems especially suited for closely coupled in-house production stages linked, e.g., by a Kanban-system or feeder line. Other parts are delivered in discrete intervals from distant locations and necessitate other peculiarities to be accounted for. With further part-centric approaches on hand combined approaches, which cover individual requirements of diverging parts, could be valuably applied to real-world assembly lines.

References

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

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

[3] Bautista, J., Companys, R., Corominas, A., 1996. Heuristics and exact algorithms for solving the Monden problem, European Journal of Operational Research 88, 101–131.

[4] Bolat, A., 1997. Stochastic procedures for scheduling minimum job sets on mixed model assembly lines, Journal of Operational Research Society 48, 490–501.

[5] Boysen, N., Fliedner, M., Scholl, A., 2007. Sequencing mixed-model assembly lines: Survey, Classification and Model Critique, Jenaer Schriften zur Wirtschaftswissenschaft 2/2007.

[6] Boysen, N., Fliedner, M., Scholl, A., 2006. Level-Scheduling bei Variantenfließfertigung: Literaturüberblick, Klassifikation und Modellkritik, Jenaer Schriften zur Wirtschaftswissenschaft 26/2006.

[7] Braglia, M., Zavanella, L., 2003. Modelling an industrial strategy for inventory management in supply chains: The “Consignment Stock” case, International Journal of Production Research 41, 3793–3808.

[8] Brauner, N., Crama, Y., 2004. The maximum deviation just-in-time scheduling problem, Discrete Applied Mathematics 134, 25–50.

[9] Cakir, A., Inman, R.R., 1993. Modified goal chasing for products with non-zero/one bills of material, International Journal of Production Research 31, 107–115.

[10] Carraway, R.L., Schmidt, R.L., 1991. An improved discrete dynamic programming algorithm for allocating resources among interdependent projects, Management Science 37, 1195–1200.

[11] Corbett, C.J., 2001. Stochastic inventory systems in a supply chain with asymmetric information: Cycle stocks, safety stocks, and consignment stock, Operations Research 49, 487–500.

[12] Dhamala, T.N., Kubiak, W., 2005. A brief survey of just-in-time sequencing for mixed-model systems, International Journal of Operations Research 2, 38–47.

[13] Dincbas, M., Simonis, H., Hentenryck, P. v., 1988. Solving the Car-Sequencing Problem in Contraint Logic Programming, 8th European Conference on Artificial Intelligence, London (1988), 290–295.

[14] Dorigo, M., Di Caro, G., Gambardella L. M., 1999. Ant algorithms for discrete optimization, Artifical Life 5, 137-172.

[15] Drexl, A., Kimms, A., Matthießen, L., 2006. Algorithms for the car sequencing and the level scheduling problem, Journal of Scheduling 9, 153–176.

[16] Duplaga, E.A., Hahn, C.K., Hur, D., 1996. Mixed-model assembly line sequencing at Hyundai Motor Company, Production and Inventory Management Journal 37, 20–26.

[17] Fliedner, M., Boysen, N., 2006. Solving the car sequencing problem via branch and bound, European Journal of Operational Research (to appear).

[18] Gagné, C., Gravel, M., Price, W.L., 2006. Solving real car sequencing problems with ant colony optimization, European Journal of Operational Research 174, 1427–1448.

[19] Gravel, M., Gagné, C., Price, W.L., 2005. Review and comparison of the three methods for the solution of the car sequencing problem, Journal of the Operational Research Society 56, 1287–1295.

[20] Inman, R.R., Bulfin, R.L., 1991. Sequencing JIT mixed model assembly lines, Management Science 37, 901–904.

[21] Joo, S.-H., Wilhelm, W.E., 1993. A review of quantitative approaches in just-in-time manufacturing, Production Planning & Control 4, 207–222.

[22] Klampfl, E., Gusikhin, O., Rossi, G., 2006. Optimization of workcell layouts in a mixed-model assebly line environment, International Journal of Flexible Manufacturing Systems 17, 277–299.

[23] Kubiak, W., 1993. Minimizing variation of production rates in just-in-time systems: A survey, European Journal of Operational Research 66, 259–271.

[24] Kubiak, W., 2003. Cyclic just-in-time sequences are optimal, Journal of Global Optimization 27, 333–347.

[25] Kubiak, W., Sethi, S.P., 1991. A note on "level schedules for mixed-model assembly lines in just-in-time production systems", Management Science 37, 121–122.

[26] Macaskill, J.L.C., 1973. Computer simulation for mixed-model production lines, Management Science 20, 341–348.

[27] Marsten, R.E., Morin, T.L., 1978. A hybrid approach to discrete mathematical programming, Mathematical Programming 14, 21–40.

[28] Mather, H., 1989. Competitive manufacturing, Prentice Hall, Englewood Cliffs, NJ.

[29] Miltenburg, J., 1989. Level Schedules for mixed-model assembly lines in just-in-time production systems, Management Science 35, 192–207.

[30] Miltenburg, J., Goldstein, T., 1991. Developing production schedules which balance part usage and smooth production loads for just-in-time production systems, Naval Research Logistics 38, 893–910.

[31] Monden, Y., 1998. Toyota Production System: an integrated approach to just-in-time, 3rd edition, Norcross, 1998.

[32] Morin, T.L., Marsten, R.E., 1976. Branch-and-bound strategies for dynamic programming, Operations Research 24, 611–627.

[33] Mulligan, R.M., 1998. EDI in forign trade. Case studies in utilisation, International Journal of Physical Distribution & Logistics Management 28, 794–804.

[34] Parrello, B.D., 1988. CAR WARS: The (almost) birth of an expert system, AI Expert 3(1), 60–64.

[35] Parrello, B., Kabat, W., Wos, L., 1986. Job-shop scheduling using automated reasoning: A case study of the car-sequencing problem, Journal of Automated Reasoning 2, 1–42.

[36] Pine, B.J. (1993): Mass customization: The new frontier in business competition, Boston.

[37] Scholl, A., Klein, R., Domschke, W., 1998. Pattern based vocabulary building for effectively sequencing mixed-model assembly lines, Journal of Heuristics 4, 359–381.

[38] Solnon, C., Cung, V.D., Nguyen, A., Artigues, C., 2006. The car sequencing problem: Overview of the state-of-the art methods and industrial case-study of the ROADEF'2005 challenge problem, European Journal of Operational Research (to appear).

[39] Steiner, G., Yeomans, J.S., 1993. Level schedules for mixed-model, just-in-time processes, Management Science 39, 728–735.

[40] Stützle, T., Dorigo, M. 1999. ACO algorithms for the quadratic assignment problem. In: Corne, D., Dorigo, M., Glover, F. (Eds.). New Ideas in Optimization, McGraw-Hill, 33-50.

[41] Thomopoulos, N.T., 1967. Line balancing-sequencing for mixed-model assembly, Management Science 14, 59-75.

[42] Valentini, G., Zavanella, L., 2003. The consignment stock of inventories: Industrial case and performance analysis, International Journal of Production Economics 81–82, 215–224.

[43] Wester, L., Kilbridge, M., 1964. The assembly line model-mix sequencing problem, Third international conference on Operation Research, Oslo 1963, (1964), 247–260.

[44] Yano, C.A., Bolat, A., 1989. Survey, development, and application of algorithms for sequencing paced assembly lines, Journal of Manufacturing and Operations Management 2, 172–198.

[45] Yano, C.A., Rachamadugu, R., 1991. Sequencing to minimize work overload in assembly lines with product options, Management Science 37, 572–586.