Jena Research Papers in Business and Economics

Scheduling Just-in-Time Part Supply for Mixed-Model Assembly Lines

Nils Boysen, Stefan Bock

01/2010

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.jbe.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

Scheduling Just-in-Time Part Supply for Mixed-Model Assembly Lines

Nils Boysen[a], Stefan Bock[b]

[a] *Friedrich-Schiller-Universität Jena, Lehrstuhl für Operations Management, Carl-Zeiß-Straße 3, D-07743 Jena, Germany,* nils.boysen@uni-jena.de

[b] *Bergische Universität Wuppertal, Lehrstuhl für Wirtschaftsinformatik und Operations Research, Gaußstraße 20, D-42097 Wuppertal, Germany,* sbock@winfor.de

Abstract
With increasing cost competition and product variety, providing an efficient Just-in-Time (JIT) supply has become one of the greatest challenges in the use of mixed-model assembly line production systems. In the present paper, therefore, we propose a new approach for scheduling JIT part supply from a central storage center. Usually, materials are stored in boxes that are allotted to the consumptive stations of the line by forklift. For such a real-world problem, a new model, a complexity proof as well as different exact and heuristic solution procedures are provided. Furthermore, at the interface between logistics and assembly operations, strategic management implications are obtained. Specifically, basing on the new approach, it is the first time a statistical analysis is being made as to whether widespread Level Scheduling policies, which are well-known from the Toyota Production System, indeed facilitate material supply. Note that in literature it is frequently claimed that this causality exists.

Keywords: Scheduling; Mixed-model assembly line; Just-in-Time; In-house Logistics

1 Introduction

An increasing necessity for close cooperation with suppliers and ongoing proliferation of product variety are two recent trends that are increasingly shifting focus to an efficient JIT part supply. For instance, in the automotive industry, production programs often comprise billions of different models (see Boysen et al., 2009a). Note that these complex model programs have to be assembled on a single mixed-model line. As a result of this, installing and maintaining an efficient JIT part supply gets more and more challenging in modern mixed-model assembly line production systems.

Therefore, the present paper proposes a new approach for efficiently handling JIT supply for mixed-model assembly lines. This approach is motivated from the real-world storage processes involved in the production of luxury cars by a major car manufacturer in Germany. To be more

specific, the approach seeks to find JIT-conform sequences of material boxes that are delivered from a central storage center.

In this specific setting the final assembly process, which has to be supplied with parts, is located on the second floor of a factory building. The central receiving store is located on the ground floor. Here, parts from multiple suppliers and preceding in-house production stages are received and temporarily stored. Both levels are connected by a hoist system, which is to be filled with transport boxes (in the ground floor) containing parts. (Note that the parts in the transport boxes are stored in standardized skeleton containers or in part-specific boxes e.g. a box for door sets). Once they have arrived on the second floor, the transport boxes are picked up by a forklift from the hoist gate and delivered to the stations of the final assembly line, where the respective parts are assembled into end-products. The line is subdivided into multiple u-shaped segments consisting of multiple stations. At the stations, each u-segment is assigned a forklift and a separate gate to the hoist system. Once a forklift has delivered a box to a station, it returns to the hoist gate and immediately fetches the next box waiting in queue. A schematic representation of part supply is depicted in Figure 1.

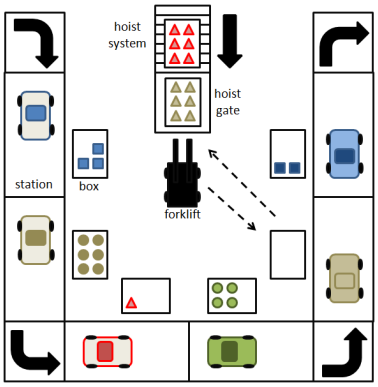

Figure 1: Schematic representation of problem setting

In our problem setting, part distribution by forklift is a bottleneck. Therefore, a suitable sequence of given material boxes has to be found. Note that each material box may contain a varying number of parts fed into the hoist system. The distances and delivery times taken by the forklift between gate and stations are known with certainty. Moreover, supplying parts by forklift takes much longer than replenishing boxes via the hoist system. Therefore, the sequence of boxes fed into the hoist system solely determines the number of parts available per station and production cycle. On the one hand, by all means, a material shortage should be avoided, otherwise there is a threat of having a line stoppage leading to hundreds of assembly workers being idle. On the other hand, however, the space at the stations of the line is notoriously scarce (see Klampfl et al., 2006; Boysen et al., 2009b). Thus, we aim at a box sequence, which minimizes the weighted maximum of part inventory near the line over all stations and production cycles. However, in order to feasibly execute the production process defined by the model sequence at

the final assembly line, resulting part demands have to be satisfied.

Clearly, although the work in this paper is inspired by a specific real-world case, our findings are easily generalizable to numerous other applications. Specifically, this problem setting becomes relevant whenever the widespread *pallet to work station*-policy is applied (e.g., see Battini et al., 2009). In this scenario pallets or boxes for parts are successively delivered by a vehicle (e.g., a forklift or tow train) from some central storage area to a consuming production system under time-varying demand. Such a storage area might be the queue of a hoist system but can also be a (centralized or decentralized) warehouse or an inbound area for truck deliveries. Note that our problem could even be more general and can also be applied if point-to-point deliveries of a single vehicle are to be scheduled in a one-warehouse-multiple-retailer setting with deterministic and time-varying demand. However, in our description we will focus on our real-world case of supplying parts to a mixed-model assembly line.

The remainder of the paper is organized as follows. Section 2 reviews related literature. Section 3 provides a detailed problem description. This is followed by sections 4 and 5, which introduce an exact Dynamic Programming procedure and a Simulated Annealing approach, respectively. The computational study is presented in Section 6. In addition to evaluating the solution performance of algorithms, this section also investigates the interface between in-house part logistics and the sequencing problem at the assembly line. The latter determines the sequence of product models launched down the assembly line. The famous Toyota Production System promotes a sequencing procedure labeled "Level Scheduling", which aims at evenly spreading part supply (induced by the model sequence) over the planning horizon. Since demand peaks are significantly reduced, it is frequently conjectured that material supply is alleviated. Our aim is to test whether leveled production sequences in final assembly are indeed helpful in avoiding excessive inventory in our part supply setting. Finally, Section 7 presents a summary and conclusions of our findings.

2 Literature Review

Existing literature on mixed-model assembly lines mainly deals either with the strategic assembly line balancing problem or the operational sequencing problem, which determines the succession of models launched down the line. For recent review papers on both topics see Boysen et al. (2007) and Boysen et al. (2009c), respectively. In relation to our hoist scheduling problem, it is assumed that both problems have already been solved. Thus, the layout of the line is already determined, so that it is easy to accurately define which part is required at what station. Moreover, the short-term model sequence is assumed to be given, so that exact demands for parts per station and production cycle are known with certainty.

Literature on organizing the part supply of mixed-model assembly lines is extremely scarce. For instance, Klampfl et al. (2006) investigate the location of material boxes within a station's restricted space. The organization of part supply from a consignment stock organized by a third party logistics provider is considered by Boysen et al. (2008). Other part feeding strategies are,

e.g., investigated by Baek et al., (1997) as well as Choi and Lee (2002). However, part supply of mixed-model assembly lines still remains a widely unexplored field of research.

If movements of material boxes from hoist system to the assembly line stations are interpreted as jobs and the forklift as a processor, our problem is closely related to traditional machine scheduling problems with given due dates (for an overview see Józefowska, 2007). In this field, single-machine scheduling problems (without allowing idle times of the processor and with job dependent due dates) which aim at minimizing weighted earliness and tardiness, are investigated by Abdul-Razaq and Potts (1988), Ow and Morton (1989), Li (1997) as well as Liaw (1999). However, the major difference between single-machine scheduling problems and our problem setting is that material boxes per station arrive from suppliers and preceding in-house production stages filled with diverging numbers of parts. Thus, unique due dates per material box can not be calculated upfront as the point in time a station runs out of parts depends on the number of parts delivered in preceding boxes. Moreover, late jobs are not allowed in our setting as tardiness costs for a line stoppage are prohibitive. Therefore, since our hoist scheduling problem has not yet been considered in the literature, it is described in detail in the following section.

3 Detailed Problem Description

In our problem setting, different boxes containing parts to be consumed by a mixed-model final assembly line are to be fed into a hoist system, which conveys boxes from the central receiving store close to a specific line segment. At the output gate of the hoist system a forklift fetches these boxes one after another (in the given sequence) and delivers them to the respective stations of the line. Upon returning from a station, the forklift directly loads the next box waiting in the queue of the hoist system and the course of events repeats itself until the end of the planning horizon is reached. Note that in our real-world case the planning horizon comprises exactly one shift, which typically lasts 8 hours.

Once a material box is delivered to a station, assembly workers remove parts from the box and successively assemble them into the workpiece. The sequence of production is typically fixed three to four days before production starts, whereas determining the box sequence is an operational problem solved the same day. Consequently, in what follows, demands for parts per station and production cycle are exactly defined. Note that a material shortage at a station would be extremely costly due to a necessary line stoppage or off-line repairs. Therefore, an efficient part supply has to guarantee that no station may run out of parts. Consequently, when determining the box sequence, material shortages are to be avoided. On the other hand, inventory at the stations violates the JIT-principle. Typically, the space at the stations is extremely scarce, so that excessive inventory obstructs assembly operations. For instance, workers may cover considerably longer distances in order to collect required materials. Thus, we aim at minimizing the maximum inventory over all stations and production cycles, where each part owns different weights, e.g., to represent different part dimensions or values. Note that the min-max objective

restricts the amount of weighted inventory in every production cycle while preventing inventory becoming extraordinarily high in single cycles, as might occur in the min-sum case. As such the obstruction of assembly operations by such peaks of inventory is disproportionately high, in our case, the min-max objective seems better suited.

In addition to this general problem setting, the following premises are introduced:

- No forklift idle time is allowed between two consecutive jobs. Note that by allowing idle time, more degrees of freedom with regard to the objective function would exist. Specifically, by delaying certain deliveries, inventory near the line could be reduced. However, such a policy would require access to an information system for a forklift driver, which exactly defines his/her idle time between two jobs. In order to avoid respective investment costs, a no-wait policy is applied.

- Processing times of box deliveries by forklift to and from a station as well as maneuvering times for picking up and dropping boxes are static and deterministic.

- Time units are normalized to the equidistant length of a production cycle, which is not a very restrictive premise as typical cycle times are fairly short, e.g., between 60 and 90 seconds in the automobile industry.

- Each box contains a predetermined number of units of a single item. In order to avoid costly line stoppages or off-line repairs, a feasible schedule has to guarantee that at every station the material demand is satisfied in every time period, i.e., that the total number of items delivered is larger or equal to the cumulative demand.

- For simplicity reasons, it is assumed, that each station assembles only a single part. Consequently, each part can be unambiguously assigned to a single station. Clearly, extensions towards the supply of multiple parts per station could be easily integrated into our problem definition and solutions procedures.

- Typically, multiple boxes containing the same part type are to be fed into the hoist system during the planning horizon. All of these boxes may contain a varying number of parts.

- After the arrival of the forklift at a station, parts are assumed to be available for assembly just after a given static maneuvering time (labeled "drop time") has elapsed.

- W.l.o.g. safety stock must not explicitly be considered, but can simply be added to the first demand period.

- Empty material boxes are collected by separate tugger trains, which cycle through multiple u-segments and return empty boxes. Thus, box removals are not considered within our problem. Note that, however, returns by forklift could be easily integrated into our problem by simply adding station specific return times.

Based on these assumptions, our hoist scheduling problem can be formally defined as follows:

M	set of stations (index m)
T	number of production cycles (index $t = 1, \ldots, T$)
s	index of sequence positions in the hoist system (with $s = 1, \ldots, \lvert B \rvert$)
B	set of boxes containing parts (index p)
$B_m \subseteq B$	subset of boxes that have to be delivered to station m
$\zeta_p \in M$	station where box p has to be delivered to
$D_{m,t}$	cumulated demand for parts at station m up to cycle t
p_m	delivery time for forklift from hoist to station m
p^l	total time span that elapse for the forklift to pick up and drop a box, i.e., the total time span for executing both operations
a_p	number of parts that are contained in box p
w_m	weighted priority of parts at station m
A_m	initial inventory of parts at station m
π	solution vector that defines the sequence of boxes fed into the hoist
$\pi(s)$	box on sequence position s within solution vector π
t_p^π	point in time where box p becomes available at its station ζ_p. Clearly, this time assignment depends on the definition of the schedule π
$\Gamma_{m,t}$	set of boxes that are delivered up to cycle t at station m
$L_{m,t}$	inventory at station m stored near the line in cycle t

Table 1: Notations

A solution is determined by a vector π: $B \longrightarrow B$, which stores the sequence of boxes fed into the hoist system. Consequently, the point in time $t^\pi_{\pi(s)}$ where a box $\pi(s)$, which is scheduled at position s, becomes available at the line, can be calculated as follows:

$$t^\pi_{\pi(s)} = p_{\zeta_{\pi(s)}} + \sum_{s'=1}^{s-1} 2 \cdot p_{\zeta_{\pi(s')}} + s \cdot p^l \quad \forall\, s = 1, \ldots, |B|. \tag{1}$$

Clearly, for delivering box $\pi(s)$ at sequence position s, the forklift has to transport this box from the hoist to the respective station. This requires $p_{\pi(s)}$ time units. Additionally, all preceding boxes must be processed by the forklift. These operations require the transport time from hoist to station and back. Finally, s pick and drop operations need to be considered. Furthermore, we denote $\Gamma_{m,t}$ as the set of boxes that are delivered on time up to cycle t at station m. Specifically, it holds that:

$$\Gamma_{m,t} = \left\{ p \in B_m | t^\pi_p \leq t \right\} \quad \forall\, m \in M,\, t = 1, \ldots, T. \tag{2}$$

With these abbreviations, resulting inventory levels $L_{m,t}$ at station m in cycle t are calculated by:

$$L_{m,t} = A_m + \sum_{p \in \Gamma_{m,t}} a_p - D_{m,t} \quad \forall\, m \in M,\, t = 1, \ldots, T. \tag{3}$$

In cycle t resulting inventory $L_{m,t}$ at station m amounts to initial inventory A_m (first term)

plus the sum of parts that are delivered on time in boxes (second term) minus the respective cumulated part demand $D_{m,t}$ of the assembly production process at station m. By applying these equations and the notations summarized in Table 1, the hoist scheduling problem (HSP) can be formulated as a mathematical optimization model:

Model formulation: Within HSP a permutation vector π: $B \longrightarrow B$ has to be determined that defines the sequence of boxes $p \in B$ fed into the hoist system. The solution vector is bound to the following objective function (4) and feasibility constraints (1) to (3), and (5):

$$\text{(HSP) minimize } Z(\pi) = \max_{m \in M;\, t=1..T} \{w_m \cdot L_{m,t}\} \tag{4}$$

$$\text{subject to (1), (2), and (3) as well as to}$$

$$L_{m,t} \geq 0 \quad \forall\, m \in M,\, t = 1, \ldots, T \tag{5}$$

Objective function (4) pursues the minimization of the maximum inventory level in all stations over all cycles t. Weighting factors w_m are applied to assess each station m individually, so that, e.g., diverging part dimensions or values can be adequately represented. Furthermore, constraint (5) ensures that no material shortages can occur.

Example: Consider four boxes to be delivered to two stations ($M = \{1, 2\}$). Boxes 1 and 2 contain $a_1 = 2$ and $a_2 = 3$ parts for station $m = 1$, respectively. The third (with $a_3 = 2$) and fourth box (with $a_4 = 3$) are dedicated to station $m = 2$. Processing times are $p_1 = p_2 = 1$ while drop and pickup times are of negligible size, i.e., it holds that $p^l = 0$. Moreover, since all stations are equally weighted, it holds that $w_1 = w_2 = 1$. More data is given in Table 2.

Consider a solution $\pi = (3, 1, 4, 2)$ of boxes fed into the hoist system. Due to predetermined processing times, these boxes arrive at their respective stations in cycle 1 (station 2), 3 (station 1), 5 (station 2) and 7 (station 1), respectively. The objective value amounts to $Z(\pi) = 3$ resulting from the derived inventory levels L_{mt} that are listed in Table 3.

t	1	2	3	4	5	6	7	8	A_m
$D_{1,t}$	0	1	1	2	3	3	5	6	1
$D_{2,t}$	1	2	2	4	4	5	6	7	2

Table 2: Example data

Proposition: HSP is NP-hard in the strong sense.

Proof: See appendix.

t	0	1	2	3	4	5	6	7	8
$L_{1,t}$	1	1	0	2	1	0	0	1	0
$L_{2,t}$	2	3	2	2	0	3	2	1	0

Table 3: Inventory $L_{m,t}$ for box sequence $\pi = (3, 1, 4, 2)$

4 Bounded Dynamic Programming

In this section, we propose a Dynamic Programming (DP) procedure for solving HSP to optimality. Subsequently, in order to obtain a more efficient enumeration process, this approach is extended by upper and lower bounds to a Bounded Dynamic Programming procedure.

Basically, our DP-approach is an extension of the well-known DP-procedure for sequencing problems proposed by Held and Karp (1962). Thus, the decision process is carried out in $|B|$ stages, where each stage $s = 1, \ldots, |B|$ represents a current sequence position.

Any stage s comprises a set of states, where each state represents a possible subset $B^s \subseteq B$ of boxes scheduled up to the current sequence position s, i.e., $|B^s| = s$. In order to find an optimal schedule, we introduce function $h^*(B^s)$ as the minimal objective function value of a schedule for boxes of set $B^s \subseteq B$, i.e., the min-max weighted inventory. This function can be calculated by making use of the following recursive formula:

$$h^*(B^s) = \min_{p \in B^s} \left\{ \max \left\{ h^*(B^s \setminus \{p\}),\, f(B^s, p) \right\} \right\}. \tag{6}$$

This calculation is initialized by setting $h^*(\emptyset) = 0$. Thus, (partial) objective values $h^*(B^s)$ are calculated by choosing the minimum over all boxes $p \in B^s$ to be scheduled at the last position s. Consequently, partial objective function values of the respective predecessor state containing boxes $B^s \setminus \{p\}$ (first term of max-function) are compared with the current resulting contribution $f(B^s, p)$ to the overall objective function that occurs by scheduling box p at the last position s.

Obviously, a maximum inventory level potentially augmenting the overall objective value can only occur in a period of a forklift delivery, because all other periods merely consume inventory and step-wise reduce stock between two delivery periods. Moreover, at a specific delivery period a maximum can only occur for the station that is currently supplied with parts. Thus, in order to identify the maximum contribution of box p, we consider the respective delivery period $t(B^s, p)$, in which box p arrives at station ζ_p:

$$t(B^s, p) = p_{\zeta_p} + \sum_{\tau \in B^s \setminus \{p\}} 2 \cdot p_{\zeta_\tau} + |B^s| \cdot p^l. \tag{7}$$

For this specific point in time and corresponding station ζ_p the maximum contribution $f(B^s, p)$ of box p scheduled at the sequence position s and antecedent subset $B_s \setminus \{p\}$ is determined as follows:

$$f(B^s, p) = w_{\zeta_p} \cdot \left(A_{\zeta_p} + \sum_{\tau \in B^s \cap B_{\zeta_p}} a_\tau - D_{\zeta_p, t(B^s, \zeta_p)} \right). \tag{8}$$

Furthermore, infeasible states, which result in material shortages in final assembly, need to be excluded from the decision process. Obviously, a minimum inventory level (and, thus, a potential material shortage) can only occur in the production cycle right before a forklift delivery. Thus, for each state it is checked whether there is enough inventory before a successive forklift can visit the respective station again or before the final cycle T is reached. This period $t(B^s, m)$ that is right before the next possible forklift arrival at station m can be calculated by:

$$t(B^s, m) = \min \left\{ p_m + \sum_{p \in B^s} 2 \cdot p_{\zeta_p} + (|B^s| + 1) \cdot p^l - 1,\, T \right\}. \tag{9}$$

Consequently, for each subset $B^s \subseteq B$ of boxes resulting inventory level $L(B^s, m)$ can be calculated at each station m by making use of the formula:

$$L(B^s, m) = A_m + \sum_{p \in B^s \cap B_m} a_p - D_{m, t(B^s, m)}. \tag{10}$$

Clearly, if $\min_{m \in M}\{L(B^s, m)\} < 0$ holds, the considered state is infeasible and is therefore excluded from the succeeding enumeration process.

Finally, when the final stage $|B|$ is reached and an optimal solution value $h^*(B)$ is determined, a simple backward recursion can be applied in order to determine the optimal sequence of material boxes. Hence, in a worst case scenario, where no infeasible state can be excluded, there are altogether $2^{|B|}$ states to be evaluated, so that the computational worst case time complexity of the algorithm amounts to $O(2^{|B|})$ (see Held and Karp, 1962).

Example (cont.): The resulting DP-graph for the aforementioned example data is depicted in Figure 2. Optimal solution value amounts to a weighted maximum inventory of $h^*(B) = 3$. One of five optimal solutions is $\pi = (3, 1, 4, 2)$ (see Table 3).

Although the number of states to be generated is considerably reduced compared to a direct assignment of individual boxes to sequence positions (e.g., in a box-oriented branching scheme), it may be too large for problem instances with plenty boxes $|B|$. Thus, in order 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).

BDP extends the DP-approach that is introduced above by additionally computing a lower bound $LB(B^s)$ on the objective function value of the schedule of the remaining boxes $B \setminus B^s$ for any state. Furthermore, a global upper bound UB is determined upfront by some heuristic procedure (see Section 5). Let $h^*(B^s)$ be the minimal maximum weighted inventory level of

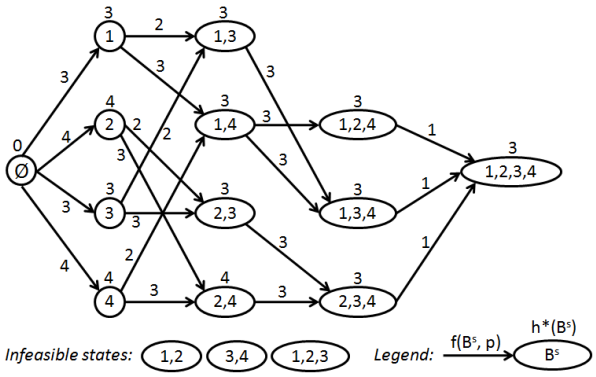

Figure 2: Resulting DP-graph for the example

already scheduled boxes B^s in a currently considered state s, i.e., the optimal objective function value for scheduling this set of boxes. Whenever it holds $\max\{LB(B^s), h^*(B^s)\} \geq UB$, this state can be fathomed as it cannot be part of a complete solution with a better objective function value than the incumbent best solution.

For the lower bound computation of the remaining problem (vacant sequence positions $s+1$ to $|B|$) the following simple consideration is applied: Recall that a potential weighted maximum inventory can only occur in a delivery period. In such a period, at least the number of parts a_p that are currently delivered will be in inventory reduced by the number of parts demanded for assembly in the respective production cycle. Thus, in a best case scenario (in sense of a minimum inventory level) the maximum number of parts to be delivered to station m in remaining boxes $(B_m \setminus B^s)$ is scheduled at the one of remaining cycles (from $t(B^s, m)$ to T) that causes the largest demand. The weighted maximum over all stations of this difference amounts to lower bound $LB(B^s)$:

$$LB(B^s) = \max_{m \in M} \left\{ w_m \cdot \left(\max_{p \in B_m \setminus B^s} \{a_p\} - \max_{\tau = t(B^s, m), \ldots, T} \{D_{m,\tau} - D_{m,\tau-1}\} \right) \right\} \tag{11}$$

Example (cont.): The resulting BDP-graph for a given upper bound of $UB = 4$ is depicted in Figure 3. Note that fathomed states are colored grey.

5 A Simulated Annealing approach

Since the problem HSP is NP-hard in the strong sense, optimal solutions can be generated for problem instances with a limited problem size only. Specifically, as within our DP- and BDP-approaches the number of states to be examined rises exponentially with the number of boxes $|B|$, problem instances of real-world size are not solvable to optimality. Hence, a Simulated Annealing (SA) approach is presented in the following. SA is a stochastic meta-heuristic that is able to overcome local optima. Specifically, it is based on the acceptance of a modified neighboring solution on a probabilistic scheme inspired by thermal processes for obtaining low-energy states

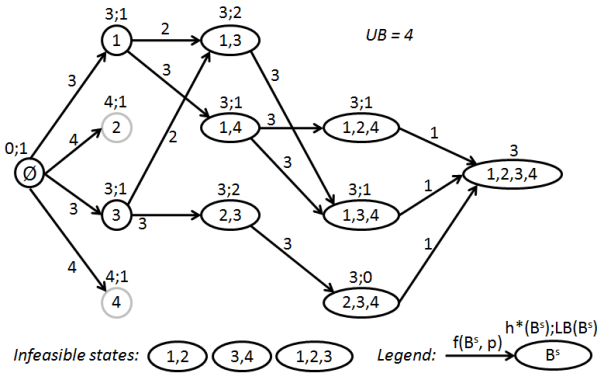

Figure 3: BDP-graph for the example

in heat baths (e.g. Kirkpatrick et al., 1983; Aarts et al., 1997).

Preliminary tests with simple priority rule-based start heuristics predominantly lead to infeasible solutions. In these unsuccessful start heuristics the box sequence is simply filled from left to right while choices are guided by a priority value. Thus, a meta-heuristic seems much more promising in order to guide the search into feasible regions of the solution space. Although other meta-heuristics such as tabu search are possible, we opted for SA as it is a quite simple yet powerful approach that has successfully been applied to many real-world automotive problems, e.g., at the French car manufacturer Renault (Solnon et al., 2008).

Our SA-approach operates on a vector π with element $\pi(s)$ (with $s = 1, \ldots, |B|$) determining the box that is currently assigned to the respective sequence position s in the hoist system. As the neighborhood-function we apply a simple swapping move. Specifically, two boxes at randomly chosen sequence positions are interchanged. The initial solution vector is randomly filled with boxes. For a given sequence vector π the corresponding objective function value $Z(\pi)$ can be easily determined according to equations (1) to (4).

If this objective function value is directly applied for the acceptance decision of a neighboring solution, the procedure might yield infeasible solutions. As simple repair mechanisms are not available, we apply the idea of penalty costs. It penalizes solutions showing material shortages proportionally to the degree of violation. By integrating a penalty value PV, we derive a modified objective function value $Z^{SA}(\pi)$ with additional potential penalty costs, which is applied to guide the acceptance of neighboring solutions:

$$Z^{SA}(\pi) = Z(\pi) + PV \cdot \left(\sum_{m \in M} \sum_{t=1}^{T} \max \{-L_{m,t},\, 0\} \right) \tag{12}$$

Clearly, if a material shortage occurs within a current solution it holds that $L_{m,t} \leq 0$ for at least one station m and production cycle t. Consequently, based on the calculation of $L_{m,t}$ defined in restriction (3), this modified objective function weights each part unit that is not supplied in time with penalty costs PV.

A proper determination of the penalty value PV turns out to be a very critical task with regard to the overall performance of the SA. It should be avoided that the SA gets stuck in less promising regions of the solutions space. Such a region may be, for instance, characterized by feasible solutions with poor objective function values or infeasible solutions with further solutions around that violate even more constraints. In a negative scenario this region cannot be left by the enumeration process, due to only marginal acceptance probabilities of neighboring infeasible solutions.

Thus, dynamically changing penalty values in terms of a diversification-intensification strategy have been applied (e.g., see Boysen et al., 2009b). For this purpose, the penalty value PV is initialized with $PV^{start} = Z(\pi)^{start} \cdot 10$, where $Z(\pi)^{start}$ is the objective function value of the first randomly generated solution (be it feasible or not). Subsequently, with each improvement of the modified objective value $Z^{SA}(\pi)$ the intensiveness of the search towards a local optimum is augmented by raising the penalty value by applying $PV := PV \cdot 1.2$.

Otherwise, if within the last 30 swap moves (in direct succession) no improvement of $Z^{SA}(\pi)$ is obtained, the penalty value PV is decreased by applying $PV := PV \cdot 0.5$. By reducing the impact of the penalty value, the computation of infeasible constellations becomes more likely. Consequently, the searching process may be directed into other regions and is therefore diversified.

The decision about whether a neighboring solution π' obtained by a swap move is accepted is decided according to traditional probability schemes (cf. Aarts et al., 1997).

$$Prob(\pi' \text{ replacing } \pi) = \begin{cases} 1, & \text{if } Z^{SA}(\pi') \leq Z^{SA}(\pi) \\ exp\left(\frac{Z^{SA}(\pi) - Z^{SA}(\pi')}{C}\right), & \text{otherwise} \end{cases} \tag{13}$$

If accepted, current solution π is replaced by π' as the starting point for further local search moves.

Our SA is steered by a simple static cooling schedule (see Kirkpatrick et al., 1983). The initial value for control parameter C is calculated by $PV^{start} \cdot 10$. Subsequently, this value C is continuously decreased in the course of the procedure by multiplying it with factor 0.995 in each iteration. If penalty value PV falls below $\frac{PV^{start}}{10}$, the procedure is restarted with a new random sequence and a re-initialized control parameter C. A total of 100,000 neighboring solutions are evaluated by our SA-approach and the feasible solution (if obtained) with the minimum objective function value $Z(\pi)$ is returned. In our computational study, we will only report results for the values of control parameters described above. Note that preliminary studies have indicated that this parameter constellation outperforms other settings and has obtained promising results.

6 Computational study

In what follows, efficiency and management impacts of our problem model and solution approaches are analyzed. Since no established test-bed is available for a comprehensive computational study, we firstly focus on the generation of useful test instances. Based on these problem

instances, algorithmic performance of our solution procedures is subsequently analyzed in detail.

Additionally, we focus on possible management impacts of our approach. Specifically, we investigate the interface between part supply and production planning. By making use of our approach, we analyze in detail whether leveled production sequences indeed ease an efficient material supply. Note that in literature, it is often claimed that this causality exists.

6.1 Instance generation

In our computational study, we distinguish between two classes of test instances: case A instances, which are small enough to be solved to optimality, and case B instances, which represent instances of real-world size and, thus, need to be solved heuristically. In order to generate instances of HSP, the following input parameters that are listed in Table 4 are applied.

		values	
symbol	description	case A	case B
$\|M\|$	number of parts (=stations)	4, 5, 6	10, 15, 20
$\|B\|$	number of boxes	10, 12, 14, 16, 18, 20	30, 40, 50, 60, 70, 80
$PROB$	probability of a demand event per station m and cycle t	0.3, 0.5, 0.7	
p^l	drop time	0	
w_m	weight of part m	1	

Table 4: Parameters for instance generation

Within each test case, the parameters are combined in a full-factorial design while instance generation is repeated 10 times. Consequently, $3 \cdot 6 \cdot 3 \cdot 10 = 540$ different instances of case A and case B were obtained, respectively. On the basis of a resulting predetermined set of parameters, each single instance is generated as follows:

- At first, all $|B|$ boxes are randomly assigned to stations, where each station is assured of receiving at least one box. Furthermore, a random sequence π of boxes is determined. In order to ensure that each instance is feasibly solvable (according to demand constraints), remaining input data is generated in such a manner that sequence π is a feasible solution. This is done by the following steps.

- In order to emulate realistic processing times occurring in an u-line-segment, where first and last station are closest to the hoist gate, processing times are generated by making use of the following set of formulas: $p_m = p_{|M|+1-m} = m + 1$, $\forall m = 1, \ldots, \lceil \frac{|M|}{2} \rceil$.

  Using these processing times, the availability time t_{π_s} of each box can be determined according to restriction (1), where the number of cycles T is set to the return time of the forklift after its final delivery.

- The demand for stations is then determined by a random choice:

$$D_{m,t} = D_{m,t-1} + \begin{cases} 1, \text{ if } rnd \leq PROB \\ 0, \text{ otherwise} \end{cases} \qquad \forall\, m \in M;\, t = 1, \ldots, T, \tag{14}$$

with rnd being an equally distributed random number that is drawn out of interval $[0; 1]$.

- Finally, for a given position s in the box sequence π, with $\pi_s = p$, the number of items a_p per box is determined. In order to generate a problem instance, where the generated schedule π is feasible, suitable delivery times p_{ζ_p} are defined accordingly. Specifically, the content of box π_s is determined by aggregating the demand from the arrival time of this box p up to the succeeding box that is delivered to station ζ_p. An exception is given by the forklift's last arrival, where demand is aggregated up to last cycle T. Additionally, initial inventory A_m of each station m is set such that enough parts are available up to the first arrival of the forklift.

6.2 Algorithmic performance

The algorithms were coded in C# 2008 and all tests have been conducted on a 2.1 GHz x86 Personal Computer with 2 GB of memory.

First, the solution performance of the exact solution procedures DP (dynamic programming) and BDP (bounded dynamic programming) is investigated by solving all instances of test case A. The left-hand side of Figure 4 shows an exponential increase of runtime in the number of boxes $|B|$ for both procedures. However, due to the applied bounding rule, the increase of BDP is less distinct. Note that a timeout after 300 CPU-seconds was applied for each instance, which explains the descending increase at the last data point with $|B| = 20$.

In this largest parameter constellation with 20 boxes, DP cannot solve a single instance (out of 90) within the given maximum time limit, whereas BDP still finds 43 (47%) optimal solutions. The coherence between the size of instances (with regard to $|B|$) and the number of instances solved to optimality is depicted at the right-hand side of Figure 4. The figure reveals that the upper limit of boxes up to which DP and BDP can be reasonably applied ranges between 18 and 20.

The aggregated results for case A instances of heuristic SA (simulated annealing) and lower bound LB (see equation (11)) are listed in Table 5. SA seems well suited for finding near optimal solutions. It determines a feasible solution for all 540 instances and solves 480 instances (99.37%) out of 483, for which the optimal solution values were found by BDP, to optimality in merely 0.178 CPU-seconds on average. Our fast but simple lower bound shows a considerable deviation from optimum. Only 52 optimal objective values are found and average relative deviation (measured by $\frac{Z(DP)-Z(LB)}{Z(DP)}$ amounts to 19.75%.

Finally, case B instances representing problems of real-world size are investigated. Here, optimal solution values cannot be gained anymore, so that merely SA and LB are compared. In only three out of 540 instances SA was not able to find a feasible solution. For 55 problem

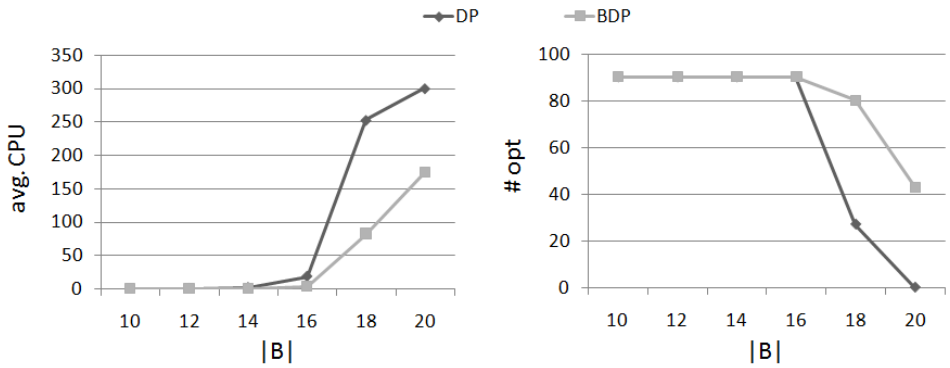

Figure 4: Performance of exact solution procedures for case A instances

measure	SA	LB
number of optimal solutions found	480	52
optimal solutions (in %)	99.37	10.77
average absolute deviation	0.006	3.573
maximum absolute deviation	1	14
average relative deviation (in %)	0.02	19.75
maximum relative deviation (in %)	4	67
average CPU-seconds	0.178	< 0.01
maximum CPU-seconds	0.312	< 0.01

Table 5: Algorithmic performance of SA and LB for case A instances

instances optimality of a found solution could be proven by showing $Z(SA) = Z(LB)$. The average relative deviation between SA and LB (measured by $\frac{Z(SA)-Z(LB)}{Z(LB)}$) for case B instances amounts to 33.9%. Compared to 32.4% within case A instances, this gap remains nearly constant. As in case A this gap was predominately caused by LB (see Table 5), it can be conjectured that SA finds near optimal solutions for the large instances, too. Together with a short runtime, which increases linearly in the number of boxes $|B|$ (see Figure 5), our SA-algorithm seems to be a promising approach for efficiently solving real-world instances of HSP.

6.3 Interdependency between part supply and production sequencing

By making use of our new approach, we are able to analyze interesting possible interdependencies between part supply and production sequencing. Due to its application within the well-known Toyota Production System, Level Scheduling has received widespread attention in practice (see, e.g., Duplaga et al., 1996; Monden, 1998) as well as in research (for surveys see, Kubiak, 1993; Dahmala and Kubiak, 2005; Boysen et al., 2009c).

Basically, Level Scheduling pursues the finding of production sequences at the final assembly line such that resulting material demand at each station is smoothed over all cycles. Clearly, as is frequently claimed in literature (e.g., Monden, 1998; Boysen et al., 2009c), this objective

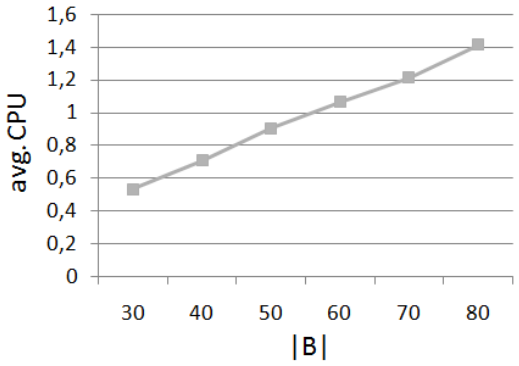

Figure 5: Performance of SA for case B instances

seems to be appropriate to facilitate an efficient JIT supply of material with small safety stocks.

However, because of a considerable recent reduction in vertical integration, assembly lines are increasingly supplied in batches. Therefore, adopting to Level Scheduling policies may restrict flexibility in assembly line sequencing considerably. On the other hand, for strategic management, a more detailed analysis should be made as to whether a smoothed model sequence may facilitate an efficient material supply from a warehouse. Our new approach allows us to analyze in detail for the first time whether there is a correlation between Level Scheduling and inventory-efficient material supply. Specifically, we measure the correlation between classic Level Scheduling goals and objective values for the new HSP in our specific supply setting.

In this analysis the well-known objective function for the Output Rate Variation (ORV) problem is applied (Kubiak, 1993). The target consumption rate r_m per station m, which is approximated by the current average part demand per cycle, is defined as:

$$r_m = \frac{D_{m,T}}{T} \; \forall m \in M. \tag{15}$$

Here, $D_{m,T}$ represents the total part demand at station m to be assembled over all T production cycles. This leads to an objective function:

$$ORV_1(\pi) = \sum_{t=1}^{T} \sum_{m \in M} \left(D_{m,t} - t \cdot r_m\right)^2, \tag{16}$$

with $D_{m,t}$ defining the cumulative part demand at station m up to cycle t. Based on this definition, objective function $ORV_1(\pi)$ compares actual demands at each station with an ideal part consumption per cycle, that is defined by parameter r_m.

Depending on the aggregation and deviation function, different alternatives to the sum-squared case that is defined by equation (16) have been presented in ORV-literature (see Boysen et al., 2009c). Therefore, we will also take another classic objective function – the max-abs case – into consideration:

$$ORV_2(\pi) = \max_{t=1..T;\, m \in M} |D_{m,t} - t \cdot r_m| \,. \tag{17}$$

With these functions (16) and (17) any demand vector D of a HSP-instances can be evaluated according to its ORV-efficiency and can then be compared to the respective optimal objective value attained for HSP. For such a comparison the ten test instances of the parameter constellation marked in bold in Table 4 are applied.

In any case, demand vector D is randomly shuffled by swapping demand events within a station, so that 100 instances are derived, which only differ in the distribution of demand per station over time. Note that instances that turned out to be infeasible according the demand constraint of HSP are removed, which occurred in less than 3% of demand vectors.

Based on these guidelines, ten test cases (overall containing 1000 instances) are derived, for which Pearson's product-moment correlation between Z and ORV_1 as well as between Z and ORV_2, respectively, is determined.

On an average of all the ten test cases, the correlation between objective values of HSP and ORV amount to merely 0.065 (ORV_1) and 0.057 (ORV_2), respectively. Even the maximum correlation is only 0.194 (ORV_1) and 0.185 (ORV_2). Thus, the conclusion can be drawn that no correlation exists and leveling material demand according to the ORV does not considerably facilitate part supply in our problem setting. These results can be mainly ascribed to the following two reasons:

- The ORV aims at a leveling over all production cycles, which seems particularly reasonable if parts are produced in direct vicinity of the final assembly line and are steadily "pulled" up the final assembly. These prerequisites are especially fulfilled if parts are produced on a feeder line (see Boysen et al., 2009c). In our setting, however, parts are delivered batch-wise in boxes, so that a leveling in every cycle seems of minor relevance. Instead, the aggregated part consumption, cumulated over all production cycles between two delivery events, needs to be leveled, whereas the detailed demand pattern between two delivery events is irrelevant from a JIT point of view.

- Moreover, ORV weights all parts equally, whereas the boxes of HSP contain different numbers of parts and are differently weighted. Consequently, while there is a positive leveling effect between batches caused by ORV-objectives, smoothing inside batches may even complicate an efficient JiT-supply. This can be explained by the fact that Level Scheduling approaches aspire production sequences where demand changes between neighboring cycles are minimized. This may cause higher inventory levels for HSP-instances with larger boxes. While this effect can be already observed for the applied min-max objective, we assume that it would be even more relevant for the min-sum case.

All these points highlight differences in the underlaying assumptions between both models. Consequently, in order to facilitate material supply, alternative sequencing procedures need to

be developed and analyzed. Therefore, strategic management is held to analyze in detail whether Level Scheduling policies actually support an efficient material supply at the line. Our findings emphasize that there are real-world part supply settings where it seems much more promising to focus on schedules where part demand balancing is done in batches.

7 Summary and Conclusions

This paper considers the problem of finding a schedule of material boxes for efficiently supplying the production process of a mixed-model assembly line. Specifically, this problem was motivated by an industrial application in the automotive industry, where the final assembly line production process is supplied with parts from a central storage using a hoist system. Since stoppage- or off-line repair costs at the line are prohibitive, demand restrictions are hard, i.e., due date violations are not allowed. A solution of the problem has to define a sequence of boxes to be transported by the hoist system.

In order to solve this problem, this paper introduces its mathematical definition. After proving its NP-hardness in the strong sense, different exact and heuristic solution procedures are proposed. By generating a first comprehensive test bed for the new problem, the efficiency of the new solution methods is validated. Specifically, it was shown that a Bounded Dynamic Programming approach provides optimal solutions for instances of moderate and medium size (with up to 20 boxes). Furthermore, it can be conjectured that a proposed Simulated Annealing approach is able to generate almost optimal solutions for large instances (with up to 80 boxes). Note that the consumed computational time of the SA-approach for scheduling 80 boxes is in the average less than 1.5 seconds and increases only linearly with the number of boxes. Therefore, this heuristic can also be applied to extremely large problem instances.

Finally, this paper analyzes whether strategic management can support an efficient material supply of the assembly line production process by applying Level Scheduling policies. Based on the proposed new approach, a detailed statistical analysis is provided for the first time. In contrast to the causality that is frequently conjectured in literature, no correlation can be proven. On the contrary, indicators have been found that leveling material demands inside each batch induced by the assembly sequence – like it is proposed by the Toyota Production System and its Level Scheduling approach – may complicate an efficient part supply in our problem setting.

Consequently, future research should focus on sequencing approaches for mixed-model assembly lines that pursue a leveling of material demands in batches. Note that, due to a reduced vertical integration in recent times, stages are more loosely coupled and therefore deliveries occur in larger batches. Consequently, since leveling can be focused on batches, an application of Level Scheduling policies unnecessarily reduces flexibility inside sequencing. In addition to it, batch-oriented policies may actually support an efficient internal material supply, since their objective function is stronger related to the HSP.

Additionally, future research should focus on extended versions of the HSP. In particular, due

to their practical relevance, extensions towards parallel machine-cases, where multiple forklifts and storage systems jointly supply parts for a specific line segment, seems an appropriate field for future research.

Appendix

We will now prove NP-hardness for HSP by a transformation from the 3-Partition Problem, which is well known to be NP-hard in the strong sense (see Garey and Johnson, 1979).

3-Partition Problem: Given $3q$ positive integers r_i $(i = 1, \ldots, 3q)$ and a positive integer R with $R/4 < r_i < R/2$ and $\sum_{i=1}^{3q} r_i = q \cdot R$, does there exist a partition of the set $\{1, 2, \ldots, 3q\}$ into q sets $\{A_1, A_2, \ldots, A_q\}$ such that $\sum_{i \in A_j} r_i = R, \forall j = 1, \ldots, q$?

Transformation of 3-Partition into HSP: Generate $3q$ boxes, which contain a_p units of a single part ($|M| = 1$), where a_p equals the integer values r_p of 3-Partition, with $p = 1, \ldots, 3q$. A priority weight of $w_1 = 1$ is assumed for the single part. With regard to the forklift, we assume delivery times of all boxes equal one, i.e., it holds $p_m = 1$, $\forall m \in M$ whereas pick and drop time is negligible, i.e., $p^l = 0$. For final assembly we define $T = 6 \cdot q$ cycles. Within these cycles a demand of R additional items occurs any six cycles starting with cycle 6. Hence, exactly three forklift visits with boxes are possible between any two demand events. Specifically, we define $D_{1,\tau \cdot 6+t} = \tau \cdot R, \forall t \in \{0, \ldots, 5\}$, $\tau = 0, \ldots, q$. We label a demand event and the belonging boxes delivered between previous and current demand event as a "demand cycle". The question we ask is whether we can find a solution for HSP with objective value $Z = R$.

Clearly, a feasible solution for an instance of 3-Partition can be directly transformed into a feasible solution of the corresponding HSP-instance. For each set $A_i = \{r_{i_1}, r_{i_2}, r_{i_3}\}$ we just transport the corresponding boxes containing a_{i_1}, a_{i_2}, and a_{i_3} items in one demand cycle. As the integer values of each set amount to R, obviously any demand $D_{mt} = R$ can be fulfilled while a maximum inventory level of $Z = R$ is not exceeded.

On the other hand, we can also prove that each feasible solution of an HSP-instance is also a feasible solution for 3-Partition. This holds true because of the following cognitions. First, due to the restriction on the processing time values $R/4 < a_p = r_p < R/2$, we need exactly three boxes per demand cycle. Clearly, any solution with more or less than three boxes per demand cycle results in an inventory level unequal to R.

Moreover, we have to show that the total number of items transported within each demand cycle equals R. Thus, we consider an arbitrary cycle with three boxes containing a_{i_1}, a_{i_2}, and a_{i_3} items. Since demand is always fulfilled in a feasible schedule, we conclude that if $a_{i_1} + a_{i_2} + a_{i_3} < R$, demand can only be fulfilled if inventory was taken over from a previous demand cycle. However, this requires for the previous demand cycle that more than R items lay in stock prior to the belonging demand event, which may not occur because of $Z \leq R$. Thus, in any feasible

HSP-solution with $Z \leq R$, it holds $a_{i_1} + a_{i_2} + a_{i_3} = R$ for all demand cycles. By assigning the corresponding elements in the 3-Partition instance to an identical set A_i, we have a feasible solution to 3-Partition. This obviously concludes the proof. $\square$

References

[1] Aarts, E.H.L., Korst, J.H.M., van Laarhoven, J.M., 1997. Simulated Annealing, In: Aarts, E.H.L., Lenstra, J.K. (eds.) Local search in combinatorial optimization, Chichester et al., 91–120.

[2] Abdul-Razaq, T., Potts, C., 1988. Dynamic programming state-space relaxation for single machine scheduling, Journal of the Operational Research Society 39, 141–152.

[3] Baek, J.-G., Baek, J.-K., Kim, S.-S., 1997. A batch scheduling scheme for the workcenter that supplies parts to a mixed-model assembly line, Computers & Industrial Engineering 33, 757–760.

[4] Battini, D., Faccio, M., Persona, A., Sgarbossa, F., 2009. Design of the optimal feeding policy in an assembly system, International Journal of Production Research (to appear).

[5] Boysen, N., Fliedner, M., Scholl, A., 2007. A classification of assembly line balancing problems. European Journal of Operational Research 183, 674–693.

[6] Boysen, N., Fliedner, M., Scholl, A., 2008. Sequencing mixed-model assembly lines to minimize part inventory cost, OR Spectrum 30, 611–633.

[7] Boysen, N., Fliedner, M., Scholl, A., 2009a. Assembly line balancing: Joint precedence graphs under high product variety, IIE Transactions 41, 183–193.

[8] Boysen, N., Fliedner, M., Scholl, A., 2009b. Level scheduling under storage constraints, International Journal of Production Research 47, 2669–2684.

[9] Boysen, N., Fliedner, M., Scholl, A., 2009c. Sequencing mixed-model assembly lines: Survey, Classification and Model Critique, European Journal of Operational Research 192, 349–373.

[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] Choi, W., Lee, Y., 2002. A dynamic part-feeding system for an automotive assembly line, Computers & Industrial Engineering 43, 123–134.

[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] 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.

[14] Garey, M.R., Johnson, D.S., 1979. Computers and intractability: A guide to the theory of NP-completeness, Freeman, New York.

[15] Held, M., Karp, R.M., 1962. A dynamic programming approach to sequencing problems, Journal of the Society for Industrial and Applied Mathematics 10, 196–210.

[16] Józefowska, J., 2007. Just-in-Time Scheduling, Springer.

[17] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing, Science 220, 671–680.

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

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

[20] Li, G., 1997. Single machine earliness and tardiness scheduling, European Journal of Operational Research 96, 546–558.

[21] Liaw, C.F., 1999. A branch-and-bound algorithm for the single machine earliness and tardiness scheduling problem, Computers & Operations Research 26, 679–693.

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

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

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

[25] Ow, P.S., Morton, T.E., 1989. The single machine early/tardy problem, Management Science 35, 177–191.

[26] Solnon, C., Cung, V.D., Nguyen, A., Artigues, C., 2008. 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 191, 912–927.