A Heuristic Based on Vogel's Approximation Method for Sequencing Mixed-Model Assembly Lines

Hans-Otto Guenther, Rico Gujjula, Sebastian Werk

A Heuristic Based on Vogel's Approximation Method for Sequencing Mixed-Model Assembly Lines

Journal:	*International Journal of Production Research*
Manuscript ID:	TPRS-2010-IJPR-0296.R1
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	24-Aug-2010
Complete List of Authors:	Guenther, Hans-Otto; TU Berlin, Production Management Gujjula, Rico; TU Berlin, Production Management Werk, Sebastian; TU Berlin, Production Management
Keywords:	ASSEMBLY LINES, MIXED MODEL SEQUENCING
Keywords (user):	Mixed-model assembly lines, sequencing

International Journal of Production Research
Vol. 00, No. 00, 00 Month 200x, 1–18

RESEARCH ARTICLE

A Heuristic Based on Vogel's Approximation Method for Sequencing Mixed-Model Assembly Lines

Rico Gujjula, Sebastian Werk, Hans-Otto Günther†
Department of Production Management, TU Berlin, Germany
(*Received 00 Month 200x; final version received 00 Month 200x*)

Sequencing mixed-model assembly lines is a well researched topic in literature. However, many methods which were developed to solve this problem fail to cope with either the large size or specific characteristics of real-life problems. In this paper, a heuristic is proposed which is derived from Vogel's approximation method for transportation planning. The heuristic is able to handle large and supposedly difficult problem instances. Sophisticated test scenarios considering real-life aspects were generated to evaluate the performance of the heuristic for realistic problem instances. It is shown that the proposed heuristic significantly outperforms priority rule-based methods and requires only reasonable computational effort.

Keywords: Mixed-model assembly lines, sequencing, Vogel's approximation method

1. Introduction

One of the major challenges in the configuration of contemporary mass production systems is to cope with the huge portfolio of customer-specific product options while at the same time an efficient flow of products through the production system and a high utilisation of manufacturing equipment has to be retained. Nowadays, this goal is commonly achieved by the installation of so-called mixed-model assembly lines. These assembly lines employ flexible tools and automated assembly technology such that products may be manufactured in an almost arbitrary order.

Essential planning problems associated with mixed-model assembly line systems are the following:

- Line balancing – Determine the configuration of the assembly line including the determination of the number and the equipment of stations in the line, the assignment of tasks to stations, and the takt time at which products are to be launched onto the line.

†Corresponding author. Email: hans-otto.guenther@tu-berlin.de

ISSN: 0020-7543 print/ISSN 1366-588X online

DOI: 10.1080/00207540xxxxxxxxx
http://www.informaworld.com

- Master production scheduling – Assign production orders for individual workpieces to production intervals over a short-term planning horizon of several weeks.
- Production sequencing – Determine the sequence of workpieces for each production interval.
- Material flow control – Ensure the timely release of parts from suppliers and the in-time delivery of parts to the designated stations at the line.
- Resequencing – Reorder the production sequence in case of disruptions, for instance, due to unexpected part shortages.

This paper deals with the production sequencing problem for a standard paced mixed-model assembly line. Although this is a well-researched topic in literature, most sophisticated approaches are not able to cope with large problem instances or specific characteristics of real-life problems whereas simple methods often yield inconvenient results. In this paper a novel heuristic is presented which is derived from the well-known Vogel's approximation method for transportation planning. It is shown that this heuristic yields significantly better results than priority rule-based approaches and is able to cope even with excessively large and realistic problem instances within reasonable run time.

The remainder of this paper is organised as follows. In the next section, the sequencing problem is explained in detail. Specific assumptions are presented and a formal problem description is given by a mixed-integer linear program. Afterwards, an extensive literature overview is provided to point out the contribution of this paper. Section 4 presents a metaheuristic principle which is derived from Vogel's approximation method. It is shown how this principle can be applied for sequencing mixed-model assembly lines. The experimental design which is used to evaluate the proposed heuristic is explained in section 5. The paper concludes with results from numerical tests and a discussion.

2. Problem Description

Specific problem assumptions and line characteristics used in this paper are the following:

- The processing time of a workpiece at a station is deterministic.
- Workpieces are moved on a constant-speed conveyor belt. For practical reasons it is usually impossible to remove workpieces from their respective positions on the conveyor.
- The takt time of the line is fixed. This means that workpieces are placed at the same intervals on the conveyor and the gap between any two subsequent workpieces on the line is identical.
- Workers move downstream on the conveyor belt while they process a workpiece. When they have finished it, they walk back upstream to process the next workpiece in the sequence. Walking time of workers from one workpiece to the subsequent one is negligible.
- All stations are closed, i.e. workers can only work within the limits of their station. If the worker walks upstream to the next workpiece, no processing can start until this workpiece has entered the station. The waiting time of the worker at the upstream border is called *idle time*. If the worker reaches the downstream border while a workpiece is processed, the workpiece is left unfinished. This amount of unfinished work is called *utility work*. Utility workers are assigned to help at the station if utility work occurs.
- The speed of the assembly line is scaled to 1.

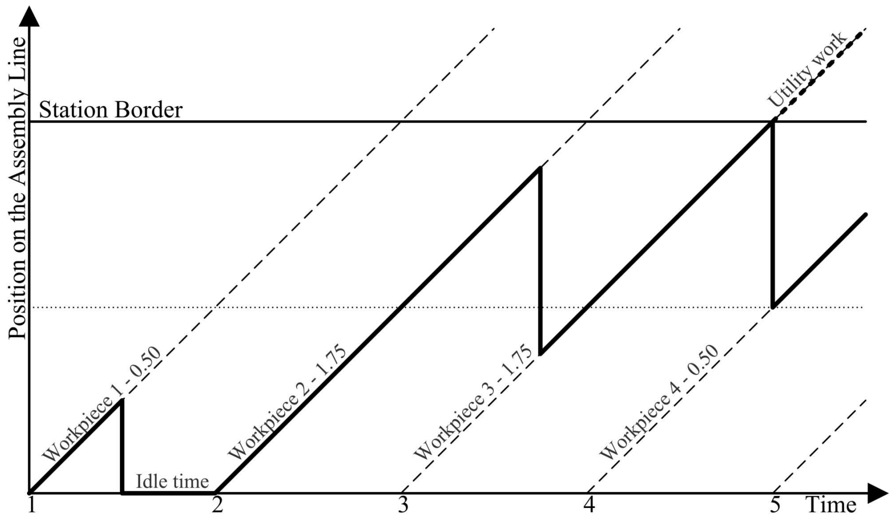


Figure 1. Example for idle time and utility work

A graphical explanation of the introduced terms is shown in Fig. 1. In this figure a station at the assembly line is depicted. The bold line represents the position of the worker within the limits of his/her station. Values on arcs indicate processing times of the respective workpieces. There are four workpieces which enter the station. The first workpiece enters the station at time 1 and leaves the station at time 3. It demands 0.5 time units processing time by the worker. The worker starts to process the workpiece at the moment it enters the station. After 0.5 time units the worker has finished his/her work, moves back to the border of the station and is idle until the next workpiece enters the station.

The second workpiece enters the station at time 2 and leaves the station at time 4. It demands 1.75 time units of processing time. The worker starts working at the workpiece when it enters the station and finishes the work after 1.75 time units. In the meantime the third workpiece has entered the station and waits for processing. At time 3.75 the worker has finished the work on the second workpiece and moves directly to the third workpiece. He/she is able to process 1.25 time units on the workpiece until it leaves the station. The worker has to move to the next workpiece as he/she can't work outside of the station. The third workpiece remains unfinished as 0.5 time units of work were not processed. This amount of unfinished work is called utility work.

Utility work is considered in this paper as main performance parameter for sequencing mixed-model assembly lines. The objective of the proposed heuristic is to minimise utility work. However, assignment and scheduling of utility workers is not in the scope of this paper. Further considerations on this topic can be found, for instance, in Gujjula and Günther (2009).

Given the specific assumptions and considerations in this section, the problem characteristic used in this paper corresponds to [|||] according to the classification scheme introduced in Boysen *et al.* (2009b). This means that all conditions for stations, assembly line and objective are based on standard characteristics.

A mixed-integer linear program (MILP) is now presented as formal description of the sequencing problem. A list of used notation can be found in Table 1.

Table 1. List of sets, parameters and variables for the MILP

Sets	
W	The set of workpieces.
S	The set of stations at the assembly line.
C	The numbered positions in the sequence with $C = \{1, .., \|W\|\}$.
Parameters	
p_{ws}	The processing time of workpiece $w \in W$ at station $s \in S$.
l_s	The length of station $s \in S$.
tt	The takt time of the assembly line.
Variables	
x_{wc}	Binary variable which is set to 1 if and only if workpiece $w \in W$ is assigned to the c-th position in the sequence.
f_{cs}	Continuous variable which represents the position of the worker at station $s \in S$ before processing the workpiece at the c-th position.
u_{cs}	Continuous variable for the utility work occurring at station $s \in S$ after processing the workpiece at the c-th position.

Minimise:

$$\sum_{c \in C} \sum_{s \in S} u_{cs} \tag{1}$$

Subject to:

$$\sum_{c \in C} x_{wc} = 1 \qquad \forall w \in W \tag{2}$$

$$\sum_{w \in W} x_{wc} = 1 \qquad \forall c \in C \tag{3}$$

$$f_{1s} = 0 \qquad \forall s \in S \tag{4}$$

$$f_{cs} + \sum_{w \in W} p_{ws} \cdot x_{wc} - l_s \leq u_{cs} \qquad \forall c \in C;\ \forall s \in S \tag{5}$$

$$f_{cs} + \sum_{w \in W} p_{ws} \cdot x_{wc} - tt - u_{cs} \leq f_{c+1,s} \qquad \forall c \in C;\ \forall s \in S \tag{6}$$

$$f_{cs} \geq 0;\ u_{cs} \geq 0 \qquad \forall c \in C;\ \forall s \in S \tag{7}$$

$$x_{wc} \in \{0,1\} \qquad \forall w \in W;\ \forall c \in C \tag{8}$$

The objective of the MILP is to minimise utility work (1). The sequence of workpieces is represented by the variables x_{wc}. To construct a feasible sequence, it has to be ensured that each workpiece is assigned to exactly one position (2) and that each position has exactly one workpiece assigned (3). It is assumed that the assembly line is in an initial state before the production period starts. This means that workers are located at the starting positions of their stations (4). Utility work is set according to the work a worker can't process within the limits of his/her station (5). The work on a workpiece can't be started either before the worker hasn't finished the work on the previous workpiece or before an unfinished workpiece has left the station (6). The model formulation concludes with non-negativity (7) and binary (8) constraints.

The MILP can be solved to optimality for small-sized instances within reasonable time. However, for real-life instances the use of standard solvers is impractical. Therefore, a heuristic approach is proposed in this paper which is able to solve even excessively large instances within reasonable time.

3. Literature Overview

In conjunction with sequencing mixed-model assembly lines, three fundamental sequencing problems are discussed in literature: mixed-model sequencing, car sequencing and level scheduling. Firstly, *mixed-model sequencing* (MMS) is a detailed scheduling approach using individual processing times of workpieces and other relevant data to determine the work amount and movement of each worker. Secondly, *car sequencing* is an aggregated concept of MMS. This approach aims at determining a sequence which doesn't violate a set of spacing rules. A spacing rule is a constraint which allows only H_o workpieces within any N_o subsequent workpieces to use a certain option o. Thirdly, *level scheduling* is an approach to support the Just-In-Time (JIT) principle by smoothing parts usage at the assembly line. Boysen *et al.* (2009b) provide an excellent overview of literature for all three sequencing problems. As this paper deals with MMS, a detailed overview of the relevant literature is presented in the next paragraphs, whereas an overview for car sequencing and level scheduling is omitted. However, notable recent papers dealing with car sequencing and level scheduling are Erel *et al.* (2007), Alpay (2009), Boysen *et al.* (2009a), McMullen (2009), Boysen *et al.* (2010) and Golle *et al.* (2010).

MMS has been investigated using several different objectives. Most frequent are the ones which directly affect labour productivity. These are work overload, i.e. the amount of unfinished work which a worker can't process within the limits of his/her station, and idle time, i.e. the time the worker is waiting in his/her station for the next workpiece. If work overload is compensated by additional workers, this amount of work is also called utility work. Papers which deal with these objectives are, for instance, Wester and Kilbridge (1964), Thomopoulos (1967), Macaskill (1973), Yano and Bolat (1989), Yano and Rachamadugu (1991), Bolat and Yano (1992b), Tsai (1995), Scholl *et al.* (1998), Sumichrast *et al.* (2000), Sarker and Pan (2001), Kim and Jeong (2007) and Bautista and Cano (2008).

Other objectives are concerned with the length of the assembly line (cf. Dar-El and

Cother 1975, Dar-El and Nadivi 1981), throughput (cf. Bard *et al.* 1992), worker displacement (cf. Okamura and Yamashina 1979, Tsai 1995) and line stoppages (cf. Xiaobo and Ohno 2000, Celano *et al.* 2004). A few papers integrate additional aspects into their objective like smoothing of parts usage (cf. Bard *et al.* 1994, Kotani *et al.* 2004, Akgündüz and Tunali 2009) or minimising setup costs (cf. Bolat 1994), whereas other papers extend the standard MMS approach by considering human factors (cf. Celano *et al.* 2004) or travelling routes of additional workers (cf. Yoo *et al.* 2005).

MMS is known to be NP-hard for most objectives like minimising utility work or worker displacement (cf. Tsai 1995). Therefore, the methodical approach in literature is rather focused on heuristics. Nevertheless, a few exact solution approaches have been presented. Common approaches are mathematical models which can be handled by commercial solvers (cf. Bard *et al.* 1992, Bolat and Yano 1992a, Bolat 1994, Scholl *et al.* 1998, Sarker and Pan 2001) and branch and bound procedures (cf. Bard *et al.* 1994, Bolat 1994, Bolat *et al.* 1994, Xiaobo and Ohno 1997). However, it has been reported that these approaches are capable of solving very small-sized instances to optimality but are unsuitable to deal with larger instances.

Reliable exact algorithms exist for restricted single-station instances (cf. Dar-El and Cucuy 1977, Yano and Rachamadugu 1991, Bolat and Yano 1992a, Tsai 1995). A few of them can be used to determine lower bounds for multiple-station instances if they follow the specific restrictions of the single-station problem (cf. Yano and Rachamadugu 1991).

The heuristics most easy to implement are greedy priority rule-based procedures which construct a sequence by successively filling it, starting from the first position. In each iteration a workpiece is chosen in accordance with a priority rule and appended to the sequence. In most cases, these priority rules are greedy in the sense that they prefer workpieces which cause the lowest increase in the objective value of the current sequence (cf. Thomopoulos 1967). Clearly, a major disadvantage of this concept is the so-called cherry picking effect. Namely, workpieces which tend to be difficult to integrate into a sequence are kept until the later stages of the heuristic and might cause an overly strong increase in the objective value in the last part of the sequence.

These shortcomings can be countervailed by applying less myopic strategies. One approach is to estimate the impact of appending a workpiece to a sequence by determining a lower bound of the objective value for the remaining workpieces. In each iteration of the heuristic, a workpiece is chosen which minimises the increase in the objective value plus the lower bound for the increase in the objective value caused by the remaining workpieces (cf. Yano and Bolat 1989, Yano and Rachamadugu 1991, Bolat 1994). However, the computational effort of these heuristic is higher than of the greedy ones and the performance improvement strictly relies on the quality of the lower bounds. It should be noted that in most cases the algorithms to determine lower bounds can be applied only to restricted instances of the MMS, for instance, the ones where each station has at most two different processing times. Bautista and Cano (2008) replace the lower bound calculation by an algorithm which determines an expected increase in the objective value. This algorithm overcomes restrictions of Yano and Rachamadugu (1991) and is able to handle multiple processing times per station at the cost of a much higher computational effort.

Another approach to avoid cherry picking is to fill every second position in the sequence not by the priority rule but rather by choosing a workpiece with a high cumulative processing time (cf. Macaskill 1973).

Improvement procedures like local search and tabu search have been successfully applied to MMS by using swap and insertion operators to manipulate a given sequence (cf. Okamura and Yamashina 1979, Bard *et al.* 1994). As for computational time and solution quality, these approaches are more suitable to handle larger instances than exact solution procedures. It has been observed in Okamura and Yamashina (1979) that the computational time of the local search approach increases not only with the number of workpieces and stations but also with the variability of processing times. Hence, this appears to be a critical factor for the performance of an algorithm.

Consider the case that workpieces can be partitioned into models, i.e. a set of interchangeable workpieces which share the same processing times for all stations. Dar-El and Cucuy (1977) as well as Scholl *et al.* (1998) exploit this input structure and propose a two-stage heuristic which constructs sequence patterns of models in the first stage and determines a selection of these patterns in the second stage in order to construct the final sequence. However, the benefit of this approach vanishes if the ratio of models and workpieces is large, which is the typically the case in modern car manufacturing.

Most approaches in literature were evaluated using randomly generated problem instances without considering specific characteristics of typical real-life problems. Neglected characteristics are, for instance, the following ones which can be observed at most car manufacturers:

- A large number of workpieces and stations have to be considered for production sequencing.
- Each workpiece within a production period is usually considered as an individual model, i.e. no two workpieces share the same processing time at all stations (cf. Decker 1993).
- For each workpiece, there is a number of stations where the required processing time exceeds the given takt time (”heavyweight”) and a number of stations where the processing time is less than takt time (”lightweight”). Although each workpiece generally tends to be either more heavy- or lightweight, it is not heavy- or lightweight at all stations, respectively.
- At each station, the capacity of workers is highly utilised and several heavyweight workpieces occur during a production period. Furthermore, station limits are tight so that many stations are prone to work overload occurrences.
- Many stations have to deal with a high variability of processing times.

This means that problem instances with such characteristics are generally too large to be handled by most approaches from literature since the data set can not be clustered or aggregated without a significant loss of accuracy. Traditional priority rule-based methods as proposed in Thomopoulos (1967) and Macaskill (1973) can cope with this situation and are easy to implement. However, the results of these methods can be inconvenient. Advanced approaches as in Bautista and Cano (2008) can yield better results but demand a higher computational effort.

To sum up, MMS is a problem which is not easy to tackle due to its complex structure and many methods except priority rule-based procedures fail to cope with the size or characteristics of real-life problem instances. Our contribution is the proposal of a novel heuristic to solve MMS by using the utility work objective. Our heuristic requires only reasonable computational effort but yields significantly better results than priority rule-based heuristics even for excessively large and supposedly difficult instances, i.e. instances with the aforementioned characteristics.

4. Heuristic Approach

The heuristic approach presented in this paper is derived from Vogel's approximation method (VAM). VAM is a well-known heuristic to solve the classic transportation problem (cf. Reinfeld and Vogel 1958). Recall that, given a single commodity, a set of suppliers, each with an individual capacity, a set of customers, each with an individual demand, and a transportation cost for every route from a supplier to a customer for the transport of one unit of the commodity, the classic transportation problem is to find shipping amounts from suppliers to customers such that the demand of each customer is satisfied and such that the overall cost of shipping is minimum.

Albeit the classic transportation problem can be easily solved to optimality by use of linear programming, VAM has received a significant amount of attention since the heuristic provides frequently optimal solutions and is easy to apply even for non-academic decision makers (cf. Shore 1970). In literature, the VAM approach has been extended and is also used as construction heuristic for other transportation problems (cf. Larson 1972, Sharma and Prasad 2003, Mathirajan and Meenakshi 2004).

Unlike other greedy approaches, VAM doesn't construct a solution based on a currently best option in terms of the objective value but chooses an option by the maximum amount of opportunity cost which could be caused if this option is not chosen.

Consider the set of customers C with unsatisfied demand and suppliers S with free capacity. In the first step of the method, for each customer $c \in C$, a priority value based on the difference between the unit transportation cost of second-cheapest and the cheapest route from suppliers in S is calculated. This is repeated for each supplier $s \in S$, but in this case, the priority value is determined by the difference of the second-cheapest and cheapest unit transportation cost for routes to customers in C.

In the next step of the method, the customer or supplier with the highest priority value is chosen and the maximum amount of the commodity is shipped on the corresponding route with the cheapest transportation cost to this customer or from this supplier, respectively. Customer and supplier who belong to this route get their demand and capacity updated, respectively. Unless all demand is satisfied, the method continues with the first step.

A general metaheuristic principle derived from VAM is presented in Fig. 2. Despite its simplicity and great peformance for the classic transportation problem, this concept has been applied only to a few other optimisation problems, for instance, the travelling salesman problem (cf. Stinson and Smith 1982). In this section, it is shown how the metaheuristic can be implemented to sequence mixed-model assembly lines.

Before the heuristic is presented, it is necessary to introduce additional notation. Let X and Y be sequences of workpieces from W. $X + Y$ denotes a sequence where Y is appended to X. Furthermore, $U(X)$ is the utility work caused by sequence X.

Based on the metaheuristic principle from Fig. 2, the following implementation for sequencing mixed-model assembly lines can be given: Start with $O_1 = O_2 = W$, and let *cost* be defined as in equation (9). P corresponds to the parameters from the MILP in section 2. After $x_o := (o_1, o_2)$ has been chosen, O_1 and O_2 get updated by removing o_1 and o_2 from each set and adding $o_1 + o_2$ to them.

$$cost(o_1, o_2) := \begin{cases} U(o_1 + o_2) & \text{for } o_1 \neq o_2 \\ \infty & \text{for } o_1 = o_2 \end{cases} \tag{9}$$

In order to calculate $U(X)$ in the course of the heuristic, equations (10) and (11) are

Input : Decision sets O_1, O_2, Parameter set P, Cost function *cost*
Output: Solution X

1 $\hat{X} \leftarrow \emptyset$
2 **while** $\hat{X}$ *does not induce a feasible solution* **do**
3 Calculate all $cost(o_1, o_2) : o_1 \in O_1; o_2 \in O_2$ considering parameters of P.
4 Determine for each $o_1 \in O_1$ the priority value pv_{o_1} by the difference of the second-cheapest and cheapest option using o_1 and let x_{o_1} denote this cheapest option
5 Determine for each $o_2 \in O_2$ the priority value pv_{o_2} by the difference of the second-cheapest and cheapest option using o_2 and let x_{o_2} denote this cheapest option
6 Choose $o \in O_1 \cup O_2$ such that pv_o is maximum
7 $\hat{X} \leftarrow \hat{X} \cup \{(x_o, P)\}$
8 Update O_1, O_2 and P if applicable
9 Construct solution X from $\hat{X}$
10 **return** X

Figure 2. Metaheuristic principle derived from Vogel's approximation method

applied such that $U(X) := \sum_{c=1}^{|X|} \sum_{s \in S} u_{cs}(X)$. Note that X_c is the workpiece in the c-th position of sequence X and that $f_{1s} := 0$ is assumed.

$$u_{cs}(X) := \max\{0, f_{cs} + p_{X_c,s} - l_s\} \tag{10}$$

$$f_{c+1,s}(X) := \max\{0, f_{cs} + p_{X_c,s} - tt - u_{cs}\} \tag{11}$$

With the proposed implementation, the algorithm merges in each iteration two sequences into a new sequence. The algorithm starts with $|W|$ sequences, this means that each workpiece is its own sequence. In the last iteration, the final sequence is constructed.

Very often in early iterations, the choice of x_o can be ambiguous, for instance, if many sequences can be joined with causing the same amount of utility work. In this case, a reasonable tiebreaker was chosen to solve this situation. For a sequence X an additional value $tb(X)$ is defined as in equation (12). The tiebreaker is to choose a joined sequence $X + Y$ such that $tb(X) + tb(Y) - tb(X + Y)$ is maximum.

$$tb(X) := \sum_{s \in S} \max\{0, f_{|X|s} + p_{X_{|X|},s} - tt\} \tag{12}$$

$tb(X)$ is defined such that the downstream position of the worker after processing the last workpiece in X is penalised, since this has a negative impact on workpieces which are appended to this sequence. If $tb(X)$ is 0, any sequence can be appended to X without hesitation. The proposed tiebreaker can be determined on-the-fly in the same run when utility work is calculated and, thus, has no significant impact on the computational time.

To demonstrate the work flow of the heuristic, an example is given in the next paragraphs. Relevant data for the scenario used in the example is given in Table 2. There are

Table 2. Data for the scenario used in the example

	a	b	c	d	length
Station 1	0.3	1.5	1.5	1.7	2.0
Station 2	0.5	0.5	1.5	1.5	2.0

Table 3. Example for the proposed heuristic

First iteration:

	a	b	c	d	pv
a	∞	0.0	0.0	0.0	0.0
b	0.0	∞	0.0	0.2	0.0
c	0.0	0.0	∞	0.2	0.0
d	$\boxed{0.0}$	0.2	0.2	∞	$\boxed{0.2}$
$pv.$	0.0	0.0	0.0	0.2	

Second iteration:

	b	c	da	pv
b	∞	0.0	0.2	0.2
c	$\boxed{0.0}$	∞	0.2	$\boxed{0.2}$
da	0.0	0.0	∞	0.0
pv	0.0	0.0	0.0	

Third iteration:

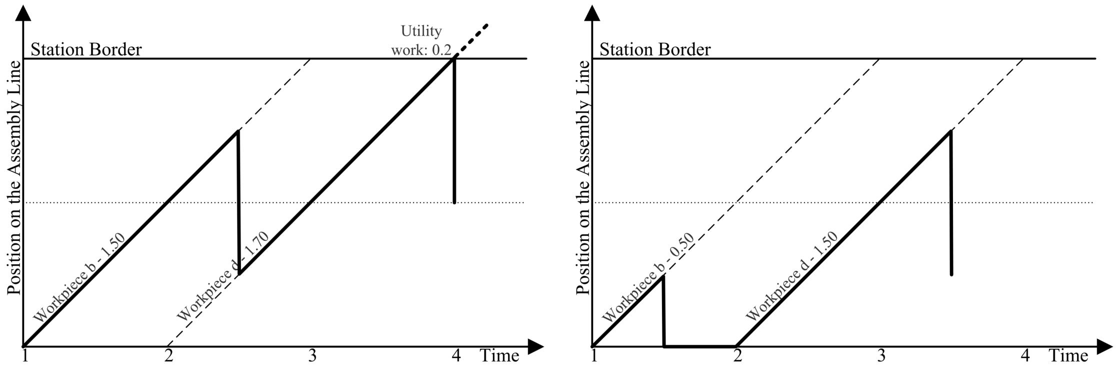
	cb	da	pv
cb	∞	0.7	∞ - 0.7
da	$\boxed{0.0}$	∞	$\boxed{\infty - 0.0}$
pv	∞ - 0.0	∞ - 0.7	

Final sequence: $dacb$

two stations, each with a length of 2, and four workpieces, namely, a, b, c and d. The takt time in the example is supposed to be 1. All iterations of the heuristic are listed in Table 3.

In the first iteration, utility work is calculated for every pair of workpieces, for instance, if b is succeeded by d, a utility work of 0.2 occurs (cf. Fig. 3 where a graphical evaluation of sequence bd is shown, section 2 contains an example how to read the figure). The cost of (a, a), (b, b), (c, c) and (d, d) is set to ∞ in accordance with equation (9). Afterwards, the priority value is determined for each row and each column. For example, the priority value of row d is the difference of the cost for the second-cheapest option (d, b) and cheapest option (d, a) so that $pv = 0.2 - 0.0 = 0.2$. Row and column d have the highest priority value and the associated options are (d, a) and (a, d), respectively. However, (d, a) is chosen due to the tiebreaker. a and d are removed from rows and columns and da is added to them.

In the second iteration, utility work is calculated for all options using the new sequence da. Afterwards, the priority value and cheapest solution is again determined for each row and each column. Rows b and c have the highest priority value and (c, b) is chosen due to the tiebreaker. b and c are removed from rows and columns and cb is added to them.



(a) Sequence *bd* at Station 1 (b) Sequence *bd* at Station 2

Figure 3. Evaluation of sequence *bd*

In the last iteration, row *da* and column *cb* have the highest priority value (note that $\infty - 0.0 > \infty - 0.7$ is assumed). However, both have the solution (da, cb) associated with them. Therefore, the final solution is *dacb*. It is an optimal solution since it causes no utility work.

Obviously, the construction of the initial matrix consumes a major part of the heuristic's effort. Utility work is determined for $|W|^2 - |W|$ sequences consisting of two workpieces. In each iteration of the heuristic, two rows and two columns are replaced by a new row and a new column. The values for these new entries have to be determined again, whereas the rest of the matrix can be reused from the previous iteration. Therefore, $2(|W| - m - 1)$ sequences have to be evaluated to compute the new matrix after the m-th iteration of the heuristic. As the heuristic takes $|W| - 1$ iterations, $|W|^2 - 3|W| + 2$ sequences have to be evaluated in order to update the matrix during the heuristic. Note that the computational time of each evaluation depends on the size of the evaluated sequence. This means that the heuristic tends to take more time, for instance, if a large sequence is generated in the early stages of the heuristic, whereas the heuristic takes less time if sequences are merged into many small sequences and are joined to a larger sequence in the final stages of the heuristic.

5. Experimental Design

To evaluate the potential of the proposed heuristic for practical application, the focus of the scenario generation is on realistic instances inspired by assembly lines of car manufacturers. This includes not only the size of an instance but also the characteristics listed in section 3. Therefore, a sophisticated data generation process has been developed to meet these characteristics.

Nine different scenarios have been considered for the numerical tests. Each scenario can be classified by the size of the corresponding problem instances (small, medium, large) and their level of difficulty (easy, medium, difficult). An overview of all scenarios is given in Table 4. Small scenarios comprise 100 stations and 200 workpieces. These numbers are doubled and quadrupled for medium and large scenarios, respectively.

There are four different processing times at each station s, namely, p_s^1, p_s^2, p_s^3, and p_s^4, which are sorted in ascending order. These processing times can have one of two characteristics: they are either clustered around the takt time tt or widely scattered in the interval $[0, l_s]$, i.e. processing times have either a low or high variability, respectively.

Table 4. Overview of scenarios used in this paper

Level of Difficulty	Size: Small	Medium	Large
Easy	100 Stations 200 Workpieces Low Variability	200 Stations 400 Workpieces Low Variability	400 Stations 800 Workpieces Low Variability
Medium	100 Stations 200 Workpieces Low/High Variability	200 Stations 400 Workpieces Low/High Variability	400 Stations 800 Workpieces Low/High Variability
Difficult	100 Stations 200 Workpieces High Variability	200 Stations 400 Workpieces High Variability	400 Stations 800 Workpieces High Variability

Table 5. Distribution of processing times for stations with low and high variability

Variability	Intervals	p_s^1	p_s^2	p_s^3	p_s^4
low	small	[0.6, 0.8]	[0.8, 1.0]	[1.0, 1.2]	[1.2, 1.4]
	large	[0.6, 1.0]	[0.6, 1.0]	[1.0, 1.4]	[1.0, 1.4]
high	small	[0.0, 0.5]	[0.5, 1.0]	[1.0, 2.0]	[2.0, 3.0]
	large	[0.0, 1.0]	[0.0, 1.0]	[1.0, 3.0]	[1.0, 3.0]

Easy scenarios comprise only stations of low variability whereas difficult scenarios contain only stations with highly variable processing times. Medium scenarios contain both types in equal parts. Four distribution templates have been applied to generate the processing times for the instances in this paper, two for stations of low and two for stations of high variability. The four templates can be found in Table 5. Note that $tt = 1$ is assumed. The templates for each class of station were assigned in equal parts. They are designed as follows. A template for each class applies either small disjoint or large overlapping intervals for each value. In either template, p_s^1, p_s^2 draw only values less or equal to 1 whereas p_s^3 and p_s^4 draw only values greater or equal to 1. Nevertheless, the average of the expected values for p_s^1 and p_s^2 as well as p_s^3 and p_s^4 is the same for both templates in each class.

In all scenarios, the utilisation rate of stations, i.e. $\frac{\sum_{w\in W} p_{ws}}{tt\cdot|W|}$, is either 85% or 95% in equal parts as well. Furthermore, station lengths are tight, i.e. for all stations s they satisfy $l_s = \left\lceil \frac{p_s^4}{tt} \right\rceil \cdot tt$.

As for workpieces, four classes are considered: basic, standard, premium and luxury workpieces. Each class has a specific distribution of processing times as shown in Table 6, for instance, 33% of all workpieces are basic. Each basic workpiece takes the lowest processing time p_s^1 at 50%, the second-lowest processing time p_s^2 at 25% and the third-lowest processing time p_s^3 at 25% of all stations, respectively. However, the choice of these stations can be different for each workpiece. With this concept, there are usually no two workpieces which share the same processing times for all stations and each workpiece is light- and heavyweight for at least 25% of all stations, respectively.

The generation process of the data is as follows. At first, processing times are generated and workpieces are assigned to these times in accordance with the given distributions. Afterwards, the utilisation rate is assigned to a station and the processing times are scaled such that the utilisation rate is met. At last, the length of the stations is set.

Table 6. Classes of considered workpieces and distribution of processing times

	p_s^1	p_s^2	p_s^3	p_s^4	Ratio
Basic	50%	25%	25%	00%	33%
Standard	25%	50%	25%	00%	33%
Premium	00%	25%	50%	25%	17%
Luxury	00%	25%	25%	50%	17%

Ten instances for each scenario were randomly generated using the considerations from the previous paragraphs. Values were scaled such that $tt = 1$. For each instance, the heuristic was applied and the utility work of the final sequence as well as the computational time were recorded. As optimal benchmarks are lacking, two sequencing heuristics have been implemented and applied to gain benchmarking results. As mentioned in section 3, the choice of solution methods from literature which are able to cope with the sequencing instances in this paper is limited to priority rule-based heuristics. However, two less myopic procedures have been chosen since they outperform greedy approaches. Both benchmarking heuristics are explained in the following.

The first heuristic is a priority rule-based method using a lower bound determination as explained in section 3. It is referred to as H1 in the remainder of this paper. Approaches in literature to determine lower bounds of the increase in the objective value, which have been applied for sequencing, are generally too restricted to handle the instances used in this paper. Therefore, H1 makes use of simple lower bounds for utility work (13) and idle time (14). In both equations X denotes the current sequence and W denotes the set of unsequenced workpieces. Similar bounds can be found, for instance, in Bautista and Cano (2008). The computational effort for both bounds is low since intermediate results can be reused in each iteration of H1. The complete heuristic procedure is outlined in Fig. 4. Equation (15) can be applied to calculate idle time $I(X)$ of a sequence X such that $I(X) := \sum_{c=1}^{|X|} \sum_{s \in S} i_{cs}(X)$.

$$LBU(X;W) := \sum_{s \in S} \left(f_{|X|+1,s} + \sum_{w \in W} p_{ws} - |W| \cdot tt - l_s \right)^+ \tag{13}$$

$$LBI(X;W) := \sum_{s \in S} \left(f_{|X|+1,s} + \sum_{w \in W} p_{ws} - |W| \cdot tt \right)^- \tag{14}$$

$$i_{cs}(X) := \max\{0, tt - f_{cs} - p_{X_c,s}\} \tag{15}$$

The second heuristic was originally proposed in Bautista and Cano (2008) and is referred to as H2 in the remainder of this paper. The heuristic extends the approach from H1 by replacing the lower bound calculation with the so-called Ud-x procedure which determines an estimation for the increase of work overload for each station. The procedure work as follows: The unsequenced workpieces are sequenced such that they generate cycles of downstream and upstream movements until no further workpiece can be sequenced without causing either utility work or idle time. Afterwards, a workpiece is determined which is sequenced next and the amount of either utility work or idle time

$\widehat{W} := W$
$X := \emptyset$
for $i = 1, ..., |W|$ **do**
  Choose $w \in \widehat{W}$ such that $U(X+w) + LBU(X+w; \widehat{W} - w)$ is minimum. If more than one workpiece with this property exists, choose w out of these workpieces such that $I(X+w) + LBI(X+w; \widehat{W} - w)$ is minimum.
  $X := X + w$
  $\widehat{W} := \widehat{W} \backslash \{w\}$
return X

Figure 4. Algorithm of H1

caused by this workpiece is recorded. This approach is repeated until all workpieces are sequenced. An outline of the Ud-x procedure can be found in Bautista and Cano (2008).

Note that the original Ud-x procedure iterates over the different models of workpieces in order to determine the next workpiece to be sequenced. However, this approach would be too consuming for the problem instances in this paper since no two workpieces share the same processing times for all stations and, thus, each workpiece is its own model. Therefore, the implementation for the numerical tests was changed so that Ud-x iterates over the processing times p_s^1, p_s^2, p_s^3, and p_s^4 for each station s which allows H2 to solve the problem instances in this paper within reasonable time. Note that despite of the modifications the original intention of the Ud-x method is retained. Nevertheless, it should be noted that an increase in the number of processing times per station would have a direct impact on the running time of H2, whereas the running time of H1 and the proposed heuristic would not be directly affected.

In the course of the numerical experiments, the computational effort of H1, H2 and the proposed heuristic to determine a sequence can vary. In order to compensate a possible time gap between H1, H2 and the proposed heuristic, a neighbourhood search improvement method has been implemented. The method is a result of various tests which have been conducted to find out the best strategy for the problem instances used in this paper. The most successful strategy was a first improvement strategy using a swap neighbourhood, i.e. the set of sequences which can be generated by swapping two workpieces in a sequence. Considering the size of the test instances, a best improvement is less powerful, since the benefit of the best improvement in comparison to any improvement can not countervail the effort to evaluate the whole neighbourhood of a sequence.

The neighbourhood is evaluated in the following order. Starting from the first position of the sequence, the first swap candidate position x is defined and the workpiece at this position is successively swapped with all workpieces which are sequenced at a later position y. If the sequence can be improved by swapping the workpieces at x and y, both workpieces remain at their new positions. Otherwise they are swapped back to their original positions. In either case the procedure continues until all swap candidates at $y > x$ are evaluated. Afterwards, x is incremented by 1 and the swapping procedure is repeated. x is set to the first position in the sequence once it has reached the last position.

This approach allows an efficient evaluation of each sequence since intermediate results can be reused. The method is interrupted if either a local optimum is reached or after reaching a time limit which is checked at the beginning of each iteration over x. The time limit has been set so that the CPU time gap between either H1 or H2 and the proposed heuristic was closed. Note that the implementation of an advanced metaheuristic was

Table 7. Average amount of utility work and computational time for each scenario and heuristic

Level of Difficulty	Small			Medium			Large		
	Heu	UW	T	Heu	UW	T	Heu	UW	T
Easy	H1:	1.1	0.3	H1:	102.1	1.9	H1:	320.8	13.7
	H1-I:	0.0	1.5	H1-I:	12.8	14.5	H1-I:	138.8	139.3
	H2:	2.3	1.0	H2:	31.3	7.0	H2:	127.8	53.7
		—		H2-I:	21.3	14.3	H2-I:	101.2	137.7
	VA:	0.0	1.0	VA:	0.0	14.1	VA:	0.0	136.8
Medium	H1:	297.7	0.3	H1:	1438.4	2.0	H1:	6320.6	14.3
	H1-I:	226.0	1.5	H1-I:	1295.5	8.7	H1-I:	6023.2	75.3
	H2:	252.8	1.2	H2:	1379.0	7.5	H2:	6316.9	58.0
		—		H2-I:	1360.1	9.0	H2-I:	6212.0	77.1
	VA:	194.0	0.7	VA:	1059.5	7.9	VA:	4951.2	74.1
Difficult	H1:	726.5	0.3	H1:	3155.6	2.1	H1:	13484.3	15.0
	H1-I:	595.2	1.5	H1-I:	2913.4	8.6	H1-I:	12994.8	76.5
	H2:	634.7	1.2	H2:	3016.0	7.5	H2:	13269.8	58.2
		—		H2-I:	2971.1	9.1	H2-I:	13027.4	79.9
	VA:	532.6	0.7	VA:	2515.0	8.0	VA:	11250.1	75.3

Heu - Heuristic UW - Utility work T - Time in s

not necessary for two reasons. Firstly, any local optimum which has been reached in the course of the numerical experiments was also a global optimum and, secondly, the improvement was still very steep when the neighbourhood search was interrupted.

The proposed heuristic as well as the benchmarking heuristics H1, H2 and the neighbourhood search improvement method were implemented using a generic framework for sequencing mixed-model assembly lines. The frameworks contains components to generate instances based on considerations from this section and sequence evaluators for several objectives. This allows a fair and meaningful comparison between all three heuristics. All tests were carried out on Intel Dual Xeon Quad Core 2.5GHz 4GB RAM using one core.

6. Numerical Tests

The results of the numerical tests can be found in Table 7. For each scenario, the average amount of utility work and computational time is listed for each heuristic. VA denotes the proposed heuristic from this paper. H1-I and H2-I denote the results from H1 and H2 after applying the neighbourhood search improvement.

On average, VA outperforms H1 and H2 as well as H1-I and H2-I. Furthermore, VA outperforms the other heuristics for each instance. The only exception is the small/easy scenario, where H1 determines the same result as VA for one instance. Additionally, H1-I determines the same result as VA for all instances within this scenario. Therefore, a detailed statistical analysis is omitted.

In all easy scenarios, VA determines for all instances solutions which cause no utility work, i.e. it solves the instances to optimality. However, the computational effort is much higher than for medium or difficult scenarios of the same size. An explanation of this behaviour can be found in section 4. Clearly, the dependency of VA's running time on the variability of processing times is a weakness of the proposed method. Nevertheless, the effort of VA is still reasonable even for these scenarios since also H1-I and H2-I are not able to outperform VA.

7. Discussion

This paper deals with sequencing mixed-model assembly lines to minimise utility work. Although this is a well researched topic in literature, only a few methods are known which are able to handle large real-life instances within reasonable time. Therefore, this paper presents a novel heuristic which is an implementation of a metaheuristic principle derived from the well-known Vogel's approximation method for transportation planning.

Several test scenarios which resemble data from car manufacturers have been generated to evaluate the heuristic. Additionally, two other heuristics have been implemented to benchmark the results. The first one adopts a concept of combining a priority rule-based method with lower bounds evaluation from literature. However, as specific approaches in literature are either too restricted or too time consuming, lower bounds have been used which can be determined with low computational effort. The second benchmarking heuristics is a method from literature which had to be slightly adjusted to handle the instances in this paper.

It is shown, that the proposed heuristic significantly outperforms both benchmarking heuristics. Furthermore, it yields optimal results for a few scenarios. Since the computational effort is higher than for the benchmarking heuristics, a neighbourhood search improvement procedure has been implemented to close the CPU time gap. However, the enhanced benchmarking procedures are still outperformed. This means that the solution quality of the proposed heuristic justifies the longer computational time.

In this paper the proposed heuristic has been evaluated only for one objective. It remains open if the good performance can be retained if the heuristic is applied to sequencing problems with different or multiple objectives or if it can be adapted to solve related sequencing problems like level scheduling or sequencing models on a line with limited flexibility. For future research it seems to be worth investigating if the metaheuristic principle derived from Vogel's approximation method can be successfully implemented to solve other combinatorial problems.

References

Akgündüz, O.S. and Tunali, S., 2009. An adaptive genetic algorithm approach for the mixed-model assembly line sequencing problem. *International Journal of Production Research* , doi: 10.1080/00207540903117857.

Alpay, S., 2009. GRASP with path relinking for a multiple objective sequencing problem for a mixed-model assembly line. *International Journal of Production Research*, 47 (21), 6001–6017.

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

Bard, J.F., Shtub, A., and Joshi, S.B., 1994. Sequencing mixed-model assembly lines to level parts usage and minimize line length. *International Journal of Production Research*, 32 (10), 2431–2454.

Bautista, J. and Cano, J., 2008. Minimizing work overload in mixed-model assembly lines. *International Journal of Production Economics*, 112 (1), 177–191.

Bolat, A., 1994. Sequencing jobs on an automobile assembly line: objectives and procedures. *International Journal of Production Research*, 32 (5), 1219–1236.

Bolat, A., Savsar, M., and Al-Fawzan, M.A., 1994. Algorithms for real-time scheduling of jobs on mixed model assembly lines. *Computers & Operations Research*, 21 (5), 487–498.

Bolat, A. and Yano, C.A., 1992a. Scheduling algorithms to minimize utility work at a single station on a paced assembly line. *Production Planning & Control*, 3 (4), 393–405.

Bolat, A. and Yano, C.A., 1992b. A surrogate objective for utility work in paced assembly lines. *Production Planning & Control*, 3 (4), 406–412.

Boysen, N., Fliedner, M., and Scholl, A., 2009a. Level scheduling of mixed-model assembly lines under storage constraints. *International Journal of Production Research*, 47 (10), 2669–2684.

Boysen, N., Fliedner, M., and Scholl, A., 2009b. Sequencing mixed-model assembly lines: Survey, classification and model critique. *European Journal of Operational Research*, 192 (2), 349–373.

Boysen, N., Fliedner, M., and Scholl, A., 2010. Level Scheduling for batched JIT supply. *Flexible Services and Manufacturing Journal* , doi: 10.1007/s10696-009-9058-z.

Celano, G., *et al.*, 2004. Human factor policy testing in the sequencing of manual mixed model assembly lines. *Computers & Operations Research*, 31, 39–59.

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

Dar-El, E.M. and Cucuy, S., 1977. Optimal Mixed-Model Sequencing for Balanced Assembly Lines. *OMEGA*, 5 (3), 333–342.

Dar-El, E.M. and Nadivi, A., 1981. A mixed-model sequencing application. *International Journal of Production Research*, 19 (1), 69–84.

Decker, M., 1993. Capacity smoothing and sequencing for mixed-model lines. *International Journal of Production Economics*, 30–31, 31–42.

Erel, E., Gocgun, Y., and Sabuncuoglu, I., 2007. Mixed-model assembly line sequencing using beam search. *International Journal of Production Research*, 45 (22), 5265–5284.

Golle, U., Boysen, N., and Rothlauf, F., 2010. Analysis and Design of Sequencing Rules for Car Sequencing. *European Journal of Operational Research* , doi: 10.1016/j.ejor.2010.03.019.

Gujjula, R. and Günther, H.O., 2009. Scheduling Utility Workers at Mixed-Model Assembly Lines. *In*: H. Sun, R. Jiao and M. Xie, eds. *Proceedings of the 2009 IEEE International Conference on Industrial Engineering and Engineering Management*, 1092–1096.

Kim, S. and Jeong, B., 2007. Product sequencing problem in Mixed-Model Assembly Line to minimize unfinished works. *Computers & Industrial Engineering*, 53, 206–214.

Kotani, S., Ito, T., and Ohno, K., 2004. Sequencing problem for a mixed-model assembly line in the Toyota production system. *International Journal of Production Research*, 42 (23), 4955–4974.

Larson, R.E., 1972. Normalizing Vogel's Approximation Method. *Mathematics Magazine*, 45 (5), 266–269.

Macaskill, J.L.C., 1973. Computer Simulation for Mixed-Model Production Lines. *Management Science*, 20 (3), 341–348.

Mathirajan, M. and Meenakshi, B., 2004. Experimental Analysis of some Variants of Vogel's Approximation Method. *Asia-Pacific Journal of Operational Research*, 21 (4), 447–462.

McMullen, P.R., 2009. JIT mixed-model sequencing with batching and setup considerations via search heuristics. *International Journal of Production Research* , doi: 10.1080/00207540903321640.

Okamura, K. and Yamashina, H., 1979. A heuristic algorithm for the assembly line model-mix sequencing problem to minimize the risk of stopping the conveyor. *International Journal of Production Research*, 17 (3), 233–247.

Reinfeld, N.V. and Vogel, W.R., 1958. *Mathematical Programming*. Englewood Cliffs, NJ: Prentice-Hall.

Sarker, B.R. and Pan, H., 2001. Designing a Mixed-Model, Open-Station Assembly Line Using Mixed-Integer Programming. *Journal of the Operational Research Society*, 52 (5), 545–558.

Scholl, A., Klein, R., and Domschke, W., 1998. Pattern Based Vocabulary Building for Effectively Sequencing Mixed-Model Assembly Lines. *Journal of Heuristics*, 4, 359–381.

Sharma, R.R.K. and Prasad, S., 2003. Obtaining a good primal solution to the uncapacitated transportation problem. *European Journal of Operational Research*, 144, 560–564.

Shore, H.H., 1970. The Transportation Problem and the Vogel Approximation Method. *Decision Sciences*, 1 (3–4), 441–457.

Stinson, J.P. and Smith, A.W., 1982. A heuristic programming procedure for sequencing the static flowshop. *International Journal of Production Research*, 20 (6), 753–764.

Sumichrast, R.T., Oxenrider, K.A., and Clayton, E.R., 2000. An Evolutionary Algorithm for Sequencing Production on a Paced Assembly Line. *Decision Sciences*, 31 (1), 149–172.

Thomopoulos, N.T., 1967. Line Balancing-Sequencing for Mixed-Model Assembly. *Management Science*, 14 (2), B59–B75.

Tsai, L.H., 1995. Mixed-model Sequencing to Minimize Utility Work and the Risk of Conveyor Stoppage. *Management Science*, 41 (3), 485–495.

Wester, L. and Kilbridge, M., 1964. The assembly line model-mix sequencing problem. *Proceedings of the third international conference on Operation Research*, 247–260.

Xiaobo, Z. and Ohno, K., 1997. Algorithms for sequencing mixed models on an assembly line in a JIT production system. *Computers & Industrial Engineering*, 32 (1), 47–56.

Xiaobo, Z. and Ohno, K., 2000. Properties of a sequencing problem for a mixed model assembly line with conveyor stoppages. *European Journal of Operational Research*, 124, 560–570.

Yano, C.A. and Bolat, A., 1989. Survey, Development, and Applications of Algorithms for paced assembly lines. *Journal of Manufacturing and Operations Management*, 2 (3), 172–198.

Yano, C.A. and Rachamadugu, R., 1991. Sequencing to Minimize Work Overload in Assembly Lines with Product Options. *Management Science*, 37 (5), 572–586.

Yoo, J.K., Shimizu, Y., and Hino, R., 2005. A Sequencing Problem for Mixed-Model Assembly Line with the Aid of Relief-Man. *JSME International Journal*, C 48 (1), 15–20.