Friedrich-Schiller-Universität Jena
seit 1558

Jena Research Papers in Business and Economics

Optimally Loading Clocked Tow Trains for JIT-Supply of Mixed-Model Assembly Lines

Simon Emde, Malte Fliedner, Nils Boysen

10/2009

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

Optimally Loading Clocked Tow Trains for JIT-Supply of Mixed-Model Assembly Lines

Simon Emde, Malte Fliedner, Nils Boysen

Friedrich-Schiller-Universität Jena, Lehrstuhl für Operations Management, Carl-Zeiß-Straße 3, D-07743 Jena, Germany,
{simon.emde,malte.fliedner,nils.boysen}@uni-jena.de

Abstract
In today's mixed-model assembly production, there are two recent trends in particular, namely increasing vertical integration and the proliferation of product variety, which more and more shift focus to an efficient just-in-time part supply. In this context, many automobile manufacturers set up decentralized logistics areas referred to as "supermarkets". Here, small tow trains are loaded with parts and travel across the shop floor on specific routes to make frequent small-lot deliveries which are needed by the stations of the line. This paper investigates the loading problem of clocked tow trains, which aims at minimizing inventory near the line while avoiding material shortages given the limited capacity of tow trains. An exact solution procedure with polynomial runtime is presented and interdependencies with production planning, i.e., the sequencing problem of product models launched down the line, are investigated.

Keywords: Mixed model assembly lines; Just-in-Time; Clocked Tow Trains; In-Process Inventory

1 Introduction

With increasing vertical integration and ongoing proliferation of product variety, just-in-time (JIT) supply of final assembly lines more and more becomes one of the greatest challenges in today's automobile production. Thousands of materials and suppliers need to be coordinated, to ensure that final assembly never runs out of parts. In this context, a decentralized organization of frequent small-lot JIT-deliveries seems especially desirable in order to flexibly adjust part supply to unforseen events and to keep inventory near the line low. For this purpose, automobile producers more and more adopt the "supermarket-concept". Supermarkets are decentralized logistics areas, where all parts of adjacent line segments are intermediately stored, so that logistics workers can prepackage parts for assembly in a comfortable manner, in analogy to customers in traditional supermarkets. Part supply out of the supermarket is conducted by small tow trains (or tuggers) which serve a subset of stations. Typically, tow trains are clocked and operate upon a given schedule, so that they circulate through stations along their tours, where they substitute empty for filled bins of parts. Finally, an empty train returns to the supermarket to be reloaded for its next tour.

For a given schedule of a tow train and a given production sequence in final assembly the number of bins to be loaded on a train's next tour can be easily determined. However, space restrictions impede a perfectly balanced JIT-supply. On the one hand, tow trains must remain maneuverable when driven through sharp turns, so that the number of waggons per tow train is restricted to less than a handful. On the other hand, the space at the stations of the line is extremely scarce, so that the number of parts stored near the line is to be reduced to a minimum.

This paper introduces a solution procedure with polynomial runtime which calculates the optimal number of material bins per tour of a clocked tow train, so that inventory near the line is minimized given the limited capacity of vehicles. The research on this topic is inspired by an implementation of the supermarket-concept at a major German automobile producer and we are aware of multiple other automobile producers applying supermarkets and clocked tow trains for their in-house logistics. The concept of "kanban supermarkets" is not a novel phenomenon, but rather a core element of the famous Toyota Production System (see Vatalaro and Taylor, 2005, Holweg, 2007) with a long tradition in many industrial sectors (Rees et al., 1989, Hodgson and Wang, 1991, Spencer, 1995). Thus, we are convinced that our findings are generalizable and that our solution procedure can also be used in a variety of other implementations of the supermarket-concept coupled with clocked tow trains.

The remainder of the paper is organized as follows. Section 2 describes the in-house logistics process of supermarkets and the utilization of clocked tow trains in detail. Then, Section 3 introduces the loading problem of tow trains and states a suited mathematical model. The optimization procedure is described in Section 4. In a comprehensive computational study (Section 5) we investigate the elementary trade-off between tour frequency and in-process inventory near the line. Furthermore, we investigate whether leveled production sequences, as proposed by the famous Toyota Production System (see Monden, 1998), are indeed helpful to reduce in-process inventory. Finally, Section 6 concludes the paper.

2 JIT-supply from decentralized supermarkets via clocked tow trains

In the course of an extensive project to reduce in-house production depth in the final assembly, a major German automobile producer completely redesigned its in-house logistics concept. Instead of supplying final assembly directly from a centralized receiving store, multiple decentralized logistics areas, i.e., supermarkets, were introduced, which intermediately stock items for adjacent line segments. Notice that supermarkets can be interpreted as the in-house logistics equivalent of the cross-docking-concept (see, e.g., Apte and Viswanathan, 2000). These supermarkets are supplied from the receiving store with (comparatively) large industrial trucks, whereas line segments are served with small tow trains. In line with the JIT-philosophy, tow trains enable more frequent part deliveries at the stations of the line in smaller lots. Moreover, part supply can be adjusted more flexibly to unforeseen events, so that wrong deliveries compared to a large-lot-supply from centralized store can be reduced. Both advantages of the supermarket-concept are very important in today's automobile production as the space at the stations of the line is notoriously scarce (see Boysen et al., 2009a). Finally, small-lot-deliveries come along with smaller bins which can be stored in comfortable racks near the line, so that assembly workers can access parts in an ergonomic and efficient manner, which reduces the strain on the workforce and saves handling times when parts are fetched.

To facilitate a reliable and steady part supply, tow trains are typically operated upon a fixed schedule. This schedule predetermines the fixed tour by which each tow train cycles through the supermarket and related stations and determines the production cycles of each stopover. As the production sequence of the final assembly line is also fixed, the amount of parts required between any two visits of a vehicle can be calculated exactly. Thus, each tow train is to be loaded in its supermarket with the respective amount of bins required for the next tour. Then, a tow train successively visits the stations on its tour and finally returns to the supermarket. At each stopover, all bins for the respective station are unloaded while empty bins are returned. Especially for smaller parts in standardized bins unloading is fully automated by employing so-called "shooter-racks". At these special kind of gravity flow racks (see, e.g., Bartholdi and Hackman, 2008, Sec. 5.1.3) the waggons of a tow train are docked while driving by. As soon as the tow train comes to a halt, gates sideways of the waggon and at the back of the rack are opened automatically and loaded bins are injected by elastic springs into the rack while empty bins are returned to the waggon. These racks reduce the length of a stopover to merely a few seconds, so that reliable tow train schedules can be derived.

The planning and control of this in-house logistics concept amounts to a complex task where several interrelated decision problems have to be solved:

(i) Decide on the number and location of decentralized supermarkets.

(ii) Assign line segments to supermarkets and determine the number of tow trains per supermarket.

(iii) Determine each tow train's fixed delivery schedule.

(iv) Decide on the bins to be loaded per tour of a tow train

To the best of the authors' knowledge there exists no literature on the coordination of supermarkets and tow trains. However, these in-house logistics decision problems show some similarities to problems of designing and operating traditional distribution networks, e.g., location of distribution centers (see, e.g., Klose and Drexl, 2005), fleet sizing (see, e.g., Beaujon and Turnquist, 1991) and inventory routing (see, e.g., Cordeau et al., 2007). This paper investigates short-term problem (iv) and is, thus, related to the famous inventory routing problem (IRP), which deals with the repeated distribution of items from a single distribution center to multiple customers. Typically, the decision about the delivery schedule and the amount of items shipped per customer is supported by IRP (see Campbell et al., 1998), so that the loading problem of vehicles can be seen as a subproblem of IRP. However, IRP is a tactical problem and, thus, solved over a mid-term horizon, so that customer demands are usually either assumed to follow a constant rate (e.g., Fisher et al., 1982, Bell et al., 1983) or a given probability distribution (e.g., Kleywegt et al., 2002, 2004). In contrast to that, our problem is a short-term one, so that the production sequence of models and its part demand are known with certainty. As a consequence, the standard solution approaches for the IRP are not applicable. In the following we investigate the deterministic loading problem of tow trains in more detail.

3 The tow train loading problem

3.1 Problem description and mathematical model

Part demand for the tow train loading (TTL) problem is determined by the production sequence of the final assembly, which is well known and communicated to all suppliers hours before TTL is to be solved. Nowadays, versatile products like cars can be specified by the customers according to their individual needs so that customizable product options, e.g., sunroof or leather trim, can be (de-)selected. Thus, it is the sequence of customized product models launched down the line which exactly specifies the number of parts required in each production cycle. Additionally, the fixed schedule of each tow train exactly determines the cycles in which a vehicle substitutes material bins at a station. Thus, the demand for bins at any station between to stops of the tow train can be easily calculated, which is shown by the following example.

Example: Consider a production system where copies of three models ($m = 1, 2, 3$) are assembled at two stations ($s = 1, 2$) . The demand of parts d_{sm}^{mod} per station and model is given in Table 1a. Assuming a (given) production schedule of $< 1, 1, 3, 2]$, the demands per cycle ($c = 1, \ldots, 5$) and station can easily be calculated as in Table 1b; notice that workpieces move down the line sequentially, so that the first copy of model 1 will reach the second station as late as cycle 2. Table 1c displays an externally given schedule which states the production cycle at which the tow train arrives at a station. On each tour ($t = 1, 2, 3$) the tow train visits both stations in adjacent cycles, so that to station 1 the train will come in cycles 0, 1 and 3 and to station 2 in cycles 0, 2 and 4, where cycle 0 refers to the required inventories at the start of production. It is assumed that all bins delivered in a cycle c are available for assembly not before cycle $c + 1$. Furthermore, we assume that bins for station 1 have a capacity of two parts while those for station 2 have one of three. In this introductory example we restrict ourselves to the case of only one kind of part per station but extending the problem to account for multiple parts with differing bin capacities is quite easy. On its first tour the tow train must deliver one bin containing 2 parts to station 1 in cycle 0. This amount is sufficient to serve the demand $d_{11}^{cyc} = 1$ of the first cycle $c = 1$, because the demand of the remaining cycles can be supplied in later tours. On tour 2 in cycle 1, the remaining part in stock at station 1 is not enough to fulfil the demand ($d_{12}^{cyc} + d_{13}^{cyc} = 1 + 2 = 3$) of cycles 2 and 3 up to the next tour, so that another bin is to be delivered. On its last visit, demand up to the last cycle has to be satisfied, which amounts to $d_{14}^{cyc} + d_{15}^{cyc} = 5 + 0 = 5$. As no parts are in stock at cycle 4 another 3 bins are to be delivered. Table 1d summarizes the resulting demands d_{st} for bins per station s and tour t.

d_{sm}^{mod}	1	2	3
1	1	5	2
2	2	1	5

(a) Example demands of parts per station and model d_{sm}^{mod}.

d_{sc}^{cyc}	1	2	3	4	5
1	1	1	2	5	0
2	0	2	2	5	1

(b) Example demands of parts per station and cycle d_{sc}^{cyc}.

c_{st}	1	2	3
1	0	1	3
2	0	2	4

(c) Example schedule of the tow train.

d_{st}	1	2	3
1	1	1	3
2	1	2	1

(d) Example demand of bins in between tours.

With these demands on hand, the JIT-principle could be easily met by simply loading each tow train exactly with the number of desired bins. However, each additional tow train causes one-time investment cost and daily labor costs for its operator, so that manufacturers aim at keeping the fleet of tow trains small. Furthermore, each vehicle's capacity is comparatively low. For instance at our OEM exactly three waggons are allowed per tow train. Otherwise, maneuverability in the sharp turns of the shop floor cannot be guaranteed. Thus, TTL aims at a delivery schedule in line with the JIT-principle, which considers the limited tow train capacity.

To concisely model this problem, the situation at our OEM allows for the following simplifying assumptions:

- In order to reduce handling times of assembly workers to a minimum, each bin needs to be stored directly next to their respective station. Thus, there exist no bins which are accessed by two or more stations.
- All bins are of identical standardized size, which is a requirement of the aforementioned shooter racks.
- With a given number of waggons per tow train and standardized bins the capacity restriction of vehicles can be measured one-dimensionally by limiting the number K of bins to be loaded. As all tuggers are sufficiently powerful an additional weight restriction is a non-issue.

- Seeing that the stock at any station can only reach its minimum right before and its maximum right after the tow train is scheduled to visit, considering every cycle in the planning horizon is not necessary. We therefore focus on the state right before a tugger arrives at a station.

Making us of the notation defined in Table 1 the restrictions of TTL can now be formalized as follows:

$$\sum_{s=1}^{S} x_{st} \leq K \qquad \forall t = 1, \ldots, T \tag{1}$$

$$\sum_{t'=1}^{t} x_{st'} \geq \sum_{t'=1}^{t} d_{st'} \qquad \forall t = 1, \ldots, T, s = 1, \ldots, S \tag{2}$$

$$x_{st} \in \mathbb{N}_0 \qquad \forall t = 1, \ldots, T, s = 1, \ldots, S \tag{3}$$

Restriction (1) ensures that the capacity of the tow train is not exceeded. Constraint (2) states that the sum of bins that have been delivered to any station s is sufficient to meet the accumulated demand in every period t. Note that this does not preclude bins from being brought to the station in an earlier period than that in which they are consumed; they will then be lying in stock in the meantime. Finally, (3) makes it impossible to deliver a non-integral or negative number of containers to any station. Note that initial stock at each station need not be modeled explicitly, but can simply be considered by reducing part/bin demands appropriately.

Concerning the objective, it seems clear that as few parts as possible should be delivered on the whole. The very heart of the JIT philosophy is to avoid waste and surplus stocks, therefore only the bins that are actually needed should be brought to the stations. To meet this goal, objective function (4) aims to minimize the sum of all bins in stock over all stations and periods.

$$\text{Minimize } f_{sum} = \sum_{t=1}^{T} \sum_{s=1}^{S} \sum_{t'=1}^{t} (x_{st'} - d_{st'}) \tag{4}$$

As a second consideration, space at the stations is notoriously scarce. If it is necessary to stock bins, they should at least be divided as equally among the stations as possible so as not to clog up one or a few stations. The maximum amount of containers stashed at any one station should be minimal, as stated by equation (5).

$$\text{Minimize } f_{max} = \max \left\{ \sum_{t'=1}^{t} (x_{st'} - d_{st'}) \middle| \, t = 1, \ldots, T; \; s = 1, \ldots, S \right\} \tag{5}$$

Example (cont.): Consider a tow train with a capacity of 3 bins per tour. Thus, for our example a "perfect" JIT-solution obviously does not exist because the total demand after the last tour ($\sum_{s=1}^{2} d_{s3}$) exceeds the capacity of the train. A feasible solution X for the example problem is given in Table 2a which leads to one container of parts being

S	number of stations (index $s = 1, \ldots, S$)
T	number of tours of the tow train (index $t = 1, \ldots, T$)
K	maximum number of containers that can be loaded onto the tow train
d_{st}	number of containers that are used up at station s in the interval between tours t and $t + 1$
x_{st}	amount of containers for station s to be loaded in tour t

Table 1: Notation

kept in stock at station 1 up until the third tour (Table 2b), resulting in $f_{sum}(X) = 2$ and $f_{max}(X) = 1$.

x_{st}	1	2	3
1	2	1	2
2	1	2	1

(a) Example solution.

l_{st}	1	2	3
1	1	1	0
2	0	0	0

(b) Example stocks of the solution.

Table 2: An example problem.

3.2 Properties of the objective functions

We have now established two desirable objectives, namely minimizing the sum of all bins stocked (Equation (4)) and minimizing the maximum number of bins stocked at any one station (Equation (5)). We will now show that these two objectives are not conflicting but can, in fact, both be optimized simultaneously.

First off, in order to ease notation, we introduce auxiliary variables l_{st}, which denote the amount of parts in stock at station s in period t and are defined as

$$l_{st} = \sum_{t'=1}^{t} (x_{st'} - d_{st'}) \; \forall t = 1, \ldots, T, s = 1, \ldots, S. \tag{6}$$

Next, the notion of an excess of demand e_t is introduced. If in any one period t the cumulated demand $\sum_{s=1}^{S} d_{st}$ of bins is greater than the capacity K of the tow train, there is an excess of

$$e_t = \sum_{s=1}^{S} d_{st} - K \; \forall t = 1, \ldots, T \tag{7}$$

that cannot be fulfilled just-in-time. If K is greater than the demand in a given period, there is an overcapacity and e_t is negative. Whenever e_t is positive, the bins in excess must have already been stocked in the preceding period $t - 1$. Of course, if the capacity of the tugger is insufficient to supply e_t plus the total demand of bins in this period

$\sum_{s=1}^{S} d_{s,t-1}$, then the difference will need to have been stocked in yet an earlier period. It follows that the number of bins for all parts required in stock g_t in any period t can be calculated recursively by

$$g_t = \max\{0; g_{t+1} + e_{t+1}\} \ \forall t = 1, \ldots, T-1, \tag{8}$$

with $g_T = 0$ (since it follows from the assumptions that it is pointless to stockpile bins in the last period). Table 3 shows the excess and required stocks in the above example.

d_{st}	1	2	3
1	1	1	3
2	1	2	1
e_t	-1	0	1
g_t	1	1	0

Table 3: e_t and g_t in the example

We can now prove the following:

Lemma 3.1. *In any feasible solution it holds that* $\sum_{s=1}^{S} l_{st} \geq g_t \quad \forall t = 1, \ldots, T$.

Proof. Directly follows from the definition of g_t. □

Lemma 3.2. *For any feasible solution X where $\sum_{s=1}^{S} l_{st'} > g_{t'}$ for a period t', it also holds that $\exists t = t'+1, \ldots, T : \sum_{s=1}^{S} x_{st} < K \vee \sum_{s=1}^{S} l_{sT} > 0$.*

Proof. Let $\sum_{s=1}^{S} x_{st} \geq K \quad \forall t' + 1, \ldots, T$. According to (7) and (8) it holds that $g_{t+1} + \sum_{s=1}^{S} d_{s,t+1} - g_t \leq K \quad \forall t = 1, \ldots, T-1$, since $g_t \geq g_{t+1} + e_{t+1} = g_{t+1} + \sum_{s=1}^{S} d_{s,t+1} - K \quad \forall t = 1, \ldots, T-1$. It follows that if $\sum_{s=1}^{S} l_{st} > g_t$ and $\sum_{s=1}^{S} x_{s,t+1} \geq K$ then $\sum_{s=1}^{S} l_{s,t+1} > g_{t+1}$, so that whenever $\sum_{s=1}^{S} l_{st'} > g_{t'}$ and $\sum_{s=1}^{S} x_{st} \geq K \quad \forall t = t' + 1, \ldots, T$ then $\sum_{s=1}^{S} l_{sT} > g_T \geq 0$ which completes the proof. □

Lemma 3.3. *Any feasible solution X with a period t' for which $\sum_{s=1}^{S} l_{st'} > g_{t'}$ holds, can be improved to a solution X', so that $f_{sum}(X') < f_{sum}(X)$ and $f_{max}(X') \leq f_{max}(X)$.*

Proof. Let $\sum_{s=1}^{S} l_{sT} > 0$ hold for solution X. It immediately follows that $l_{s'T} > 0$ for a station s'. Let t_0 denote the last period where station s' was supplied, $t_0 = max\{t = 1, \ldots, T | x_{s't} > 0\}$, it then follows that $x_{s't_0}$ can be reduced by at least one unit without compromising the feasibility of the solution, since the delivery was never consumed by a station. This reduction diminishes (at least) $\sum_{s=1}^{S} l_{sT}$, therefore improving the f_{sum}-objective, while not increasing the f_{max}-objective.

If instead $\sum_{s=1}^{S} l_{sT} = 0$ holds for X, then let t_1 be the earliest period after t' with excess capacity, $t_1 = min\left\{t = t' + 1, \ldots, T | \sum_{s=1}^{S} x_{st} < K\right\}$. According to Lemma 3.2 such a period needs to exist and since $\sum_{s=1}^{S} x_{st} \geq K \quad \forall t = t' + 1, \ldots, t_1 - 1$ it follows that $\sum_{s=1}^{S} l_{s,t_1-1} > g_{t_1-1} \geq 0$. Let s' be a station for which $l_{s',t_1-1} > 0$ holds, then $x_{s't_1}$

could be increased by at least one unit thereby diminishing l_{s',t_1-1} without compromising the feasibility of the solution. Again the resulting solution would be better with regard to f_{sum} and at least as good regarding f_{max}. □

Corollary 3.4. *A solution is f_{sum}-optimal if and only if $\sum_{s=1}^{S} l_{st} = g_t, \forall t = 1, \ldots, T$ holds.*

Proof. The "if"-part follows from the lower bound derived from Lemma 3.1, the "only if" directly follows from Lemma 3.3. □

Theorem 3.5. *For every feasible problem instance, there exists a solution which is both f_{sum} and f_{max} optimal.*

Proof. Let X be a feasible solution that is optimal with regard to f_{max}. By Lemma 3.1 it holds that no feasible solution with $\sum_{s=1}^{S} l_{st} < g_t$ for any $t = 1, \ldots, T$ can exist. Assume that in at least one period t' $\sum_{s=1}^{S} l_{st'} > g_{t'}$, then according to Lemma 3.3 the solution can be improved with respect to f_{sum} without compromising f_{max}-optimality. It follows that there is always an f_{max}-optimal solution for which $\sum_{s=1}^{S} l_{st} = g_t, \forall t = 1, \ldots, T$ holds which by Corollary 3.4 is also f_{sum}-optimal. □

Theorem 3.5 opens up the possibility to formulate problem TTL as an integer program with both objectives in one joint objective function (9) subject to (1) - (3).

$$\text{Minimize } f_j = f_{sum} + f_{max} \tag{9}$$

4 Outline of the Algorithm

With these properties in mind, we set forth to develop an algorithm to solve the TLL-problem. Theorem 3.5 states that in each period t no more (and no less) than g_t bins need to be stocked to optimize f_j. A solution that satisfies this condition will definitely be optimal with regard to f_{sum}. However, there remains the problem of distributing the bins among the stations $s = 1, \ldots, S$ in each period $t = 1, \ldots, T$ such that f_{max} is minimal.

The idea of the algorithm presented in this paper is to distribute the required stock g_t in each period as evenly as possible, i.e., so that no more than g_t bins are stocked in any period and the solution is still feasible. What makes this difficult is the fact that the decision to stock bins in one station in one period may have repercussions on all the following. Bins, once stocked, cannot be removed from a station except by consumption in the station itself. A simple example: assume that in one period t 10 bins are stocked in one station s, which does not pose a problem with regard to g_t. In the next period no bins are consumed in that station, which means that the 10 containers will of course still be lying in stock in $t + 1$. If $g_{t+1} < 10$ the solution cannot be optimal - and might not even be feasible anymore. Consequently, in order to construct a feasible and optimal solution, whenever an algorithm assigns bins to a station it will have to look ahead at how this affects later periods.

Unlike the f_{sum} objective, whose optimal value is already known in advance, the optimum of the f_{max} function is much harder to obtain. We will show that if f^*_{max} was known in advance, it would, however, be rather easy and computationally inexpensive to construct a feasible solution, as care would only have to be taken to distribute no more than g_t containers per period t while not exceeding f^*_{max} in any station. As a consequence, the optimization is transformed into a series of feasibility problems, where different values for f_{max} are systematically tested. If no feasible solution can be found, the value was too low. Otherwise, it might have been too high and a lower one needs to be checked. The determination of f^*_{max} can be implemented efficiently in the form of a binary search, where the search space between the lower and upper bound for f^*_{max} is continuously halved until only one, the correct, value remains. Notice that f^*_{max} is bounded from above by $R = max\{g_t|\forall t = 1, \ldots, T\}$, since, in the worst case, all containers required in a period are stashed at just one station. It is further bounded from below by $L = \lceil \frac{R}{S} \rceil$, as, in the best case, all containers are divided equally among the stations.

Figure 1 outlines the procedure for constructing a feasible solution – if one exists – for a given $p := f_{max}$. First, in Line 1, the g_t are calculated as with Equation (8). Then, the algorithm iterates through all periods (Line 2). In each period the number of containers to be stored k is calculated (Line 3). This number may be less than g_t (although of course not less than zero) for any given t because decisions made in earlier periods may cut into the available storage space, like in the example above where 10 containers are already stocked in period $t + 1$ and can thus not be reassigned. This same idea is also applied when calculating the upper bounds in Line 5. The question here is: What is the maximum number of bins that can be stocked in each station without making it impossible to stay below g_t bins in future periods? To answer this, first, a vector $z_{t'}$, $\forall t' = t + 1, \ldots, T$, is assembled which contains the residues of the $g_{t'}$ that are still disposable. The upper bound ub_{st} for a station s is then defined as the minimum of the sum of the accumulated demands $\sum_{t''=t+1}^{t'} d_{st''}$ (bins that have been consumed will not lie in stock anymore), the $z_{t'}$ (the number of bins that may still be stocked) and the $l_{st'}$ (the amount of bins that have already been stocked in the station) of all the periods $t' = t + 1, \ldots, T$ following the current period. Also, no upper bound can be greater than p (i.e., the given f_{max}) or $l_{st} + k$ (because no more than k bins may be distributed), of course. As long as these bounds are not exceeded, the solution will obviously be f_{sum}-optimal and feasible, the given p permitting. The pseudo-code from Line 10 to 12 serves to assign as much of k as possible to the station whose current stock level is farthest away from its maximum, dictated by ub_{st} (Line 10). If all stations have already reached their limit, the solution is infeasible because p is too small (Line 7). After each portion of k has been assigned, i.e., one station has been loaded to the maximum, the future stock levels $l_{st'}$, $\forall t' = t + 1, \ldots, T$ have to be updated in Line 13, which in turn makes a recalculation of the upper bounds necessary.

```
Calculate the g_t for all periods;
for t = 1 to T do
    Set k to the part of g_t that is still unassigned;
    while k > 0 do
        Calculate the upper bounds;
        if all stations already reached their upper bounds then
            p is infeasible;
            Exit;
        end
        Set maxStation to the station farthest away from its upper bound;
        Load maxStation to its upper bound;
        Reduce k by the amount assigned in the previous step;
        Update the stock levels in the succeeding periods;
    end
end
p is feasible;
Return the optimal stock levels;
```

Figure 1: An algorithm to check if a given $f_{max} =: p$ is feasible.

A formal description of the whole algorithm as well as an example can be found in the appendix.

Strictly speaking, the algorithm in Figure 1 only outputs the amount of containers l_{st} stored in each period and station while actually the loading of the tow train x_{st} is what is wanted. However, the x_{st} can easily be calculated by rearranging Equation (6):

$$x_{st} = l_{st} + d_{st} - l_{s,t-1} \qquad \forall s = 1, \dots, S, t = 2, \dots, T \tag{10}$$

with

$$x_{s1} = l_{s1} + d_{s1} \qquad \forall s = 1, \dots, S$$

In the appendix it is proven that the proposed algorithm solves any instance of TTL to optimality in polynomial time.

5 Computational study

5.1 Instance generation

As there is no established test data for the TTL, we will first describe how the instances for this paper were generated.

TTL instances are derived from the parts usages of different models at the stations. Depending on the production sequence of the models, the number of parts and, consequently, containers consumed at any station in between any two tours will fluctuate.

For each station count from Table 4, the models are randomly generated by assigning a demand d_{mp}^{par} for parts $p \in P$ to each model $m \in M$. The parts usages d_{mp} are calculated as $\lfloor rnd(u_m, u_m) \rceil \forall m \in M; p \in P$ where $u_m = rnd(0.5, 0.5) \forall m \in M$ and $rnd(\mu, \sigma) \sim N(\mu, \sigma)$ is a normally distributed random number capped at 0 and $\lfloor \cdot \rceil$ denotes rounding to the nearest integer. Bins have a differing capacity depending on what kind of part $p \in P$ is stored in them, namely a uniformly distributed random number from the interval $[1; 20]$, to allow for the fact that different parts may have different sizes while the size of the bins is standardized. At each station two different kinds of parts p are used.

Symbol	description	values
$\lvert M \rvert$	number of distinct models	**100**
D	sequence length	**400**
T	number of tours	10, **25**, 50, 75, 100, 125, 150, 175, 200
S	number of stations	10, 50, 100, 200, **400**

Table 4: Parameters for instance generation

Next, one sequence $Y_0 = (y_1, y_2, \ldots, y_D)$ is generated for each station count from Table 4 by assigning to each sequence position y_i, $\forall i = 1, \ldots, D$ a model $rnd(|M|/2, |M|/4)$ where $rnd(\mu, \sigma)$ is a normally distributed random number from the interval $[1; |M|]$ rounded to the next integer. Of this sequence, 99 random permutations $Y_r \forall r = 1, \ldots, 99$ are generated, leading to a total of 100 sequences for each parameter set. With these data in mind, it is easy to calculate the number of bins required d_{st}^r in between any two tours t at any station s for each sequence r by following the procedure outlined in Section 3. For the sake of this study, it is assumed that the train cycles through the stations in a consecutive manner, needing exactly 1 cycle to get from one station to the next. The tuggers will start their tour every $\lfloor D/T \rfloor$ cycles. All stations will also receive a delivery before actual production begins (in cycle 0), i.e., an initial stock that is also planned like a normal tour.

The tugger capacity K is calculated as the minimum K necessary to still be able to reach a feasible solution given the "worst" sequence for each parameter set, i.e.,

$$K = \max \left\{ \max \left\{ \left\lceil \frac{\sum_{t'=1}^{t} \sum_{s=1}^{S} d_{st'}^r}{t} \right\rceil \middle| t = 1, \ldots, T \right\} \middle| r = 0, \ldots, 99 \right\}. \qquad (11)$$

The instances are divided into two groups: First, we analyze a case developed on the basis of real-world data from a major German car manufacturer. The parameters used for this series of test are printed in bold in Table 4. Furthermore, for this case, we assume that each tow train will serve three stations which it will visit $T = 25$ times during the time frame of $D = 400$ cycles. Note that the problem decomposes into smaller sub-problems in this case, because, as there are no overlapping tugger routes, what one tow train delivers to one set of stations does not concern another tow train which serves another (distinct) set of stations. Second, we try all the other parameter combinations from Table 4 to see how tweaking the problem parameters affects results and performance. For this latter group of tests, we assume that the tow train always cycles through all stations. This is obviously an unrealistic assumption in many cases but it presents an interesting test case as it makes the problem more challenging.

5.2 Computational results

The algorithm was implemented in C# 2008 and tests have been run on an x86 PC with an Intel Core 2 Quad Q9550 2.8 GHz CPU and 4096 MB of RAM.

Apart from solving the TTL, interdependencies between part supply and production sequencing are investigated. As part of the famous Toyota Production System, level scheduling has received widespread attention both in research (for surveys, see Kubiak (1993), Dahmala and Kubiak (2005) and Boysen et al. (2009b)) as well as practical application (see, e.g., Duplaga et al. (1996), Monden (1998)). Level scheduling consists of finding a production sequence such that the material requirements are smoothed over time with the goal of facilitating JIT supply of material and reducing safety stocks – a goal that would seem to correspond well with the objectives of TTL. However, by measuring the correlation between classic level scheduling goals and objective values for the TTL, the promise of level scheduling facilitating part supply is tested in our specific supply setting. In order to do this, a well-known objective functions for the Output Rate Variation (ORV) problem is utilized (Kubiak, 1993). The target consumption rate r_p per part p, which is to be approximated by actual part demand, is defined as

$$r_p = \frac{\sum_{m \in M} d^{par}_{mp} \cdot b_m}{D} \quad \forall p \in P, \tag{12}$$

where b_m represents the number of models of type m to be produced over a planning horizon consisting of D production cycles. This leads to an objective function

$$Z_1(Y) = \sum_{t=1}^{D} \sum_{p \in P} \left(\sum_{m \in M} d^{par}_{mp} \cdot \sum_{t'=1}^{t} y_{mt'} - t \cdot r_p \right)^2 \tag{13}$$

with binary variables y_{mt} indicating whether ($y_{mt} = 1$) or not ($y_{mt} = 0$) model m is produced in cycle t. While the general goal of the classic ORV is probably compatible with the goals of the TTL, one problem with this function Z_1 can immediately be identified: The fluctuations of parts usage rates are summed up over time. Therefore, a sequence that entails a usage rate which consistently oscillates around the desired rate will possibly produce the same objective value as a sequence that behaves well most of the time but has one or a few periods with extreme deviation. It stands to reason, however, that the latter sequence would have far worse repercussions on the TTL. As a consequence, we will also take another aggregation function of the ORV into consideration:

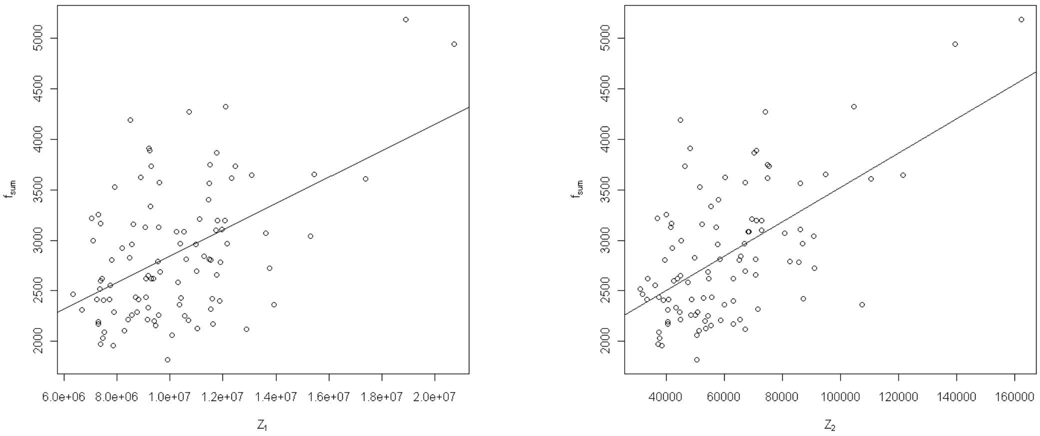
$$Z_2(Y) = \max \left\{ \sum_{p \in P} \left(\sum_{m \in M} d^{par}_{mp} \cdot \sum_{t'=1}^{t} y_{mt'} - t \cdot r_p \right)^2 \middle| \, t = 1, \ldots, D \right\}. \tag{14}$$

With these functions (13) and (14) the generated model sequences of our test instances can be evaluated with respect to ORV. For each parameter set, these objective values and the ones obtained through solving the TTL are compared. The results are listed in Table 5 for the real-world case. f_{sum} and f_{max} show the corresponding optimal objective values, the first row showing the minimum, the second the average and the third the maximum over all 100 sequences, ρ_{Z_1} and ρ_{Z_2} stand for Pearson's product-moment correlation between f_j and Z_1 and Z_2, respectively, and CPU time denotes the average time in seconds the algorithm needed to solve the problem. The correlation is also marked with an asterisk (*) if it is significant at the 95% confidence level and with two asterisks (**) if it is significant at the 99% confidence level.

f_{sum}	f_{max}	ρ_{Z_1}	ρ_{Z_2}	K	CPU time
1817	4				
2860,21	6,98	0.50**	0.59**	10	<1
5187	13				

Table 5: Minimum, average and maximum objective values and correlation with level scheduling for the real-world case.

While the numbers for the f_{sum}-objective can be hard to translate, the meaning of the f_{max} values is easy to interpret: On average, a maximum of about 7 bins was stored at any station, right before the tow train arrived. The values of both objective functions are rather volatile, ranging from 1817 to 5187 and from 4 to 13, respectively. Keeping in mind that the only thing that changes between instances is the order in which the very same models are assembled, it becomes obvious that a "good" production sequence can go a long way toward easing just-in-time supply of the assembly line. As to finding such a sequence, classic ORV methods seem to be applicable, seeing that there is a highly significant – albeit not very strong – correlation between ORV and TTL objectives. That there is great room for improvement yet to be exploited is noticeable when comparing the correlation coefficients ρ_{Z_1} and ρ_{Z_2}: Just as we surmised, the Z_2 objective is better suited to predicting attainable TTL objective values than the Z_1 objective is, although the correlation is still far from perfect, as can also be seen in Figure 2.



(a) The relation between Z_1 and f_{sum}.

(b) The relation between Z_2 and f_{sum}.

Figure 2: Correlation between ORV and TTL for the real-world case.

Table 6 shows the results for all test cases. Here, the above observations regarding correlation between ORV and TTL are confirmed. In only 6 out of the 45 problems no significant correlation between either Z_1 or Z_2 and f_j could be found. In most cases the correlation is only moderate (0.49 is the maximum for Z_1, 0.53 for Z_2) but highly significant (99% confidence in 31 cases for Z_1 and 35 cases for Z_2). In 37 cases Z_2 proved to be the better predictor of TTL success than Z_1. In the remaining 8 cases they were tied or at least very close together.

S	T	f_{sum}	f_{max}	ρ_{Z_1}	ρ_{Z_2}	K	CPU time
	10	184,68	4,06	0.33**	0.28**	59	<1
	25	103,15	2,06	0.21**	0.26**	32	<1
	50	90,25	1,45	0.18*	0.27**	24	<1
	75	183,56	1,86	0.052	0.17*	20	<1
10	100	194,34	1,78	0.10	0.16	18	<1
	125	151,01	1,5	0.19*	0.22**	17	<1
	150	150,68	1,4	0.20**	0.18*	16	<1
	175	95,65	1,17	0.24**	0.24**	16	<1
	200	63,36	1,03	0.34**	0.31**	16	<1
	10	843,17	3,77	0.24**	0.25**	255	<1
	25	677,63	2,1	0.43**	0.50**	144	<1
	50	1018,08	1,92	0.42**	0.50**	107	<1
	75	883,73	1,74	0.49**	0.51**	93	<1
50	100	1029,39	1,87	0.33**	0.35**	83	<1
	125	822,3	1,56	0.27**	0.33**	78	<1
	150	813,29	1,49	0.19*	0.22**	73	<1
	175	928,03	1,57	0.13	0.21**	68	<1
	200	766,29	1,39	0.038	0.13	66	<1
	10	2127,32	4,58	0.10	0.23**	546	<1
	25	1412,95	2,2	0.35**	0.46**	302	<1
	50	2116,99	2,06	0.29**	0.40**	221	<1
	75	2314	2,07	0.23**	0.38**	188	<1
100	100	2985,43	2,16	0.20**	0.34**	165	<1
	125	2814,19	2,07	0.26**	0.38**	151	<1
	150	2586,69	1,92	0.22**	0.34**	141	<1
	175	3433,65	2,11	0.33**	0.41**	128	<1
	200	2763,28	1,95	0.20**	0.31**	124	<1
	10	5646,86	5,5	0.13	0.11	1167	<1
	25	4679,07	2,71	0.33**	0.40**	634	<1
	50	4288,44	2,11	0.43**	0.53**	460	<1
	75	6272,87	2,27	0.32**	0.42**	381	<1
200	100	5004,77	2,02	0.26**	0.31**	343	<1
	125	8514,39	2,31	0.11	0.12	298	2
	150	7638,43	2,21	0.15	0.20*	277	2
	175	8072,71	2,15	0.076	0.056	257	3
	200	7019,28	2,07	0.11	0.095	246	3
	10	7420,27	4,1	0.096	0.093	1836	<1
	25	7000,11	2,17	0.26**	0.30**	1070	<1
	50	7175,36	1,8	0.38**	0.49**	813	2
	75	9872,19	1,98	0.41**	0.46**	686	3
400	100	11385,88	2,05	0.42**	0.44**	603	5
	125	11675,51	2,03	0.39**	0.44**	547	8
	150	9940,91	1,78	0.23**	0.25**	512	10
	175	10664,13	1,75	0.31**	0.37**	476	13
	200	8674,95	1,62	0.29**	0.33**	459	13

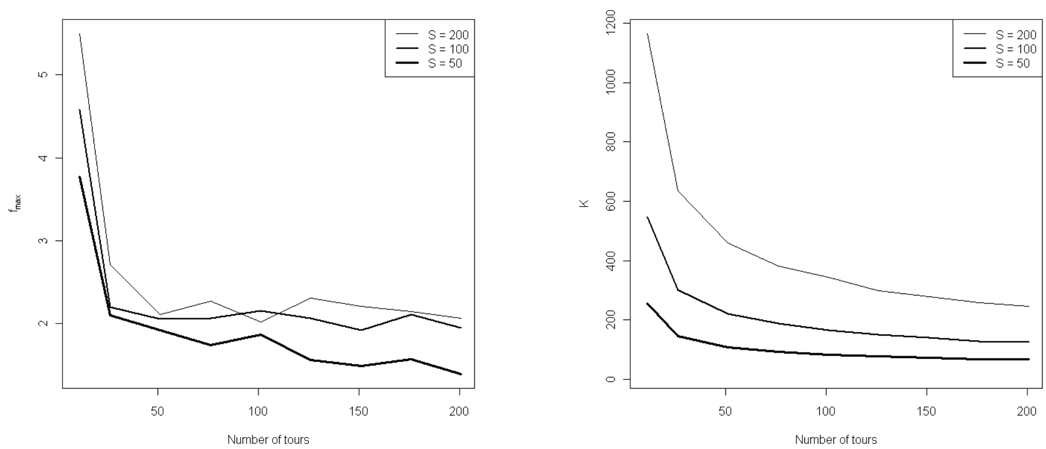
Table 6: Average objective values and correlation with level scheduling.

The only moderate correlation between ORV and TTL objectives can be explained by some of the characteristics of ORV. For instance, the employed objective functions do not account for the different bin sizes of parts, which could, however, be readily considered by introducing weighting factors for parts. More severely, classic ORV objectives weight negative and positive deviations from a leveled part demand equally. In the context of in-house supply, it is the demand peaks which trigger a supply run and the more peaks occur in consecutive production cycles the more difficult a synchronous supply becomes.

In the symmetric objective function of the ORV an overshooting of the demand rate in one cycle accompanied by an undershooting in the next cycle, is counted twice, even though it might have little impact on scheduled tow train loads. Finally, since the parts share a common bottleneck, i.e., the tow train capacity, supply problems especially occur when more than one part exceeds the ideal target rate in the same cycle. As Z_1 aggregates deviations over time, deviations of parts in the same cycle are not considered differently as the same deviations spread over several cycles. We can draw two main conclusions from this evaluation: On the one hand, classic ORV objectives are in principle suited to ease tow train deliveries in the described setting. On the other hand, the sequencing objectives discussed in the literature are far from ideal if the sole aim is an improved in-house supply. An important step in this direction would be the consideration of delivery schedules, which is discussed in detail in Boysen et al. (2009c).

In addition to the relationship to level scheduling, we further explore the connection between the number of tours T and the f_{max}-objective in Figure 3a. Unsurprisingly, there is a definite trend towards better, i.e., lower, objective values when the tow train visits the stations more often. However, this trend is anything but linear. The greatest gains are made when T jumps from 10 to 25, whereas in the interval between 100 and 200 not much happens. This is understandable, seeing that the whole production sequence takes only 400 cycles to process, meaning that with $T = 200$, the tugger will come to each station every other cycle, i.e., the supply is already almost as just-in-time as possible.

T has a very similar effect on tow train capacity K as can be seen from Figure 3b. Remember that K is systematically calculated as the bare minimum tugger capacity needed to reach a feasible solution in even the most unfavorable of the randomly generated sequences. The trend here is also undeniable: The more tours, the less burdened the tugger will be, but after a certain point (about 50 tours in the test cases), the added benefit of an even denser schedule becomes so marginal as to not matter.



(a) Influence of T on f_{max}. (b) Influence of T on K.

Figure 3: Influence of tour frequency on f_{max} and K.

Another thing that comes to attention when evaluating the test results is the relative similarity in the f_{sum}- and f_{max}-values within one parameter set. Sequences that negatively affect one objective almost invariably seem to similarly affect the other, too, and vice versa. Indeed, a correlation of up to 0.8 can be attested for the test cases (the exact strength of the correlation mainly depends on the difficulty of the parameter set – if the instance is too "easy", objective values are low all around, making it harder to detect trends), which hints at an even distribution of the bins among the stations. This means that it is apparently not the case that there is just one station which causes a lonely peak in the f_{max} objective while all the others are more or less empty (which would imply a low f_{sum}-score), which was the very reason for introducing the f_{max} objective in the first place.

Finally, concerning performance, the algorithm is obviously well-suited to solving even the most complex instances since it never needed more than an average of thirteen seconds to solve even the most difficult problems.

6 Conclusion

The paper presents an exact solution procedure with polynomial runtime for the tow train loading problem, which decides on the number of bins to be loaded per tour of a clocked and capacity constrained tow train when supplying parts consumed by a mixed-model assembly line. In a comprehensive computational study the tradeoff between the number of tours, inventory near the line and tow train capacity is investigated. In line with intuition, increasing the delivery frequency reduces inventory or requires fewer waggons per train. This effect is shown to diminish with delivery schedules becoming denser. Furthermore, interdependencies between part supply and production scheduling are investigated. The Toyota Production System promotes leveled production sequences on the final assembly line, so that resulting part demands are evenly spread over the planning horizon. This way, part supply is (said to be) facilitated and safety stocks can be reduced. This effect is confirmed by our computational study. However, it was also shown that by slightly changing the aggregation function of level scheduling this effect increases. Thus, future research should further investigate the best type of level scheduling (or alternative sequencing approaches) for different supply settings.

With regard to supermarkets and clocked tow trains future research should also tackle the superordinate problems of the planning hierarchy, e.g., location of supermarkets and determining tow train schedules (see Section 2). However, to exactly quantify the impact of different decisions within these problems solving the subordinate problem of loading tow trains becomes an essential part. As optimally solving this problem requires merely polynomial time, an integration of our solution procedure should be easily possible.

Appendix

Formal description of the algorithm

$e_t := \sum_{s=1}^{S} d_{st} - K \ \forall t = 1, \ldots, T$;
$g_T := 0$;
for $t = T - 1$ **down to** 1 **do**
    $g_t := \max\{0; e_{t+1} + g_{t+1}\}$;
end
$R := \max\{g_t \mid t = 1, \ldots, T\}$;
$L := \lceil \frac{R}{S} \rceil$;
$R := R + 1$;
while $L < R$ **do**
    $p := L + \lfloor (R - L)/2 \rfloor$;
    $pTooSmall := False$;
    $l_{st} := 0 \ \forall s = 1, \ldots, S; t = 1, \ldots, T$;
    for $t = 1$ **to** T **do**
        $k := g_t - \sum_{s=1}^{S} l_{st}$;
        while $k > 0$ **and** $pTooSmall = False$ **do**
            $z_{t'} := g_{t'} - \sum_{s=1}^{S} l_{st'} \ \forall t' = t + 1, \ldots, T$;
            $ub_{st} :=$ $\min\left\{\min\left\{\sum_{t''=t+1}^{t'} d_{st''} + z_{t'} + l_{st'} \mid t' = t + 1, \ldots, T\right\}; l_{st} + k\right\} \ \forall s = 1, \ldots, S$;
            $maxMargin := \max\{\min\{ub_{st} - l_{st}; p - l_{st}\} \mid s = 1, \ldots, S\}$;
            $maxStation := \underset{s=1,\ldots,S}{\operatorname{argmax}} \{\min\{ub_{st} - l_{st}; p - l_{st}\}\}$;
            if $maxMargin \leq 0$ **then**
                $pTooSmall := True$;
            end
            $l_{maxStation,t} := l_{maxStation,t} + maxMargin$;
            $k := k - maxMargin$;
            $l_{maxStation,t'} := \max\left\{0; l_{maxStation,t'-1} - d_{maxStation,t'}\right\} \ \forall t' = t + 1, \ldots, T$;
        end
    end
    if $pTooSmall = True$ **then**
        $L := p + 1$;
    else
        $R := p$;
        $l^*_{st} := l_{st} \ \forall s = 1, \ldots, S; t = 1, \ldots, T$;
    end
end
Return l^*;

Figure 4: Formal description of the algorithm to optimally load tow trains.

Example problem for the algorithm

Consider the example problem given in Table 7a. The table also shows the excess (or overcapacity) e_t and the amount of parts that have to be stocked g_t per period as described by Equations (7) and (8), respectively. In a first step, the lower and upper bounds for the binary search are computed: $R = \max\{8; 14; 9; 10; 0\} + 1 = 15$ and $L = \lceil \frac{14}{4} \rceil = 4$. The first value of f^*_{max} that is tried is $p = 4 + \lfloor (15-4)/2 \rfloor = 9$.

Now, to see if a feasible solution can be constructed with this upper bound p on the station load, the algorithm plans the stocks in each period. For $t = 1$, first the available storage z_t in the following periods are updated. As no capacity has yet been used, $z_t = g_t, \forall\ t = 1, \dots, T$. The z_t are needed to calculate the upper bounds ub_{s1} as in Line 17. For example,

$$\begin{aligned} ub_{11} &= \min\left\{ \min\left\{ \sum_{t''=2}^{t'} d_{1t''} + z_{t'} + l_{st'} \mid t' = 2, \dots, 5 \right\}; l_{11} + k \right\} \\ &= \min\left\{ \min\left\{ 7+14+0; 7+9+0; 15+10+0; 15+10+0 \right\}; 0+8 \right\} \\ &= \min\left\{ 16; 8 \right\} \\ &= 8. \end{aligned} \tag{15}$$

The bounds for the other stations s, periods t and iterations i can be found in Table 7b. Now, $k = g_1 - \sum_{s=1}^{S} l_{s1} = 8 - 0 = 8$ units of storage space yet to be assigned have to be distributed among the stations such that no station exceeds either its upper bound or p. For this, the station with the greatest margin $maxMargin = \max\{\min\{ub_{s1} - l_{s1}; p - l_{s1}\} \mid s = 1, \dots, 4\}$ of available storage space is selected. In period one, with $p = 9$, all stations have the same margin of $maxMargin = 8$ which is incidentally equal to k. Therefore the whole of $k = 8$ containers can be loaded into station 1 without either breaking the upper bound or p. Period $t = 1$ has thus been planned. A look at Table 8 reveals that the decision to stock 8 containers in station 1 in period 1 affects periods 2 and 3 as well because not all containers are consumed until $t = 4$. As no upper bound was exceeded, however, we can be sure that this has adverse effects on either feasibility or optimality.

d_{st}	1	2	3	4	5
1	0	7	0	8	0
2	0	7	0	8	10
3	6	0	10	3	10
4	6	0	15	0	10
e_t	-8	-6	5	-1	10
g_t	8	14	9	10	0

(a) An example problem.

t	1	2		3	4		5
i	1	1	2	1	1	2	1
$s = 1$	8	8	5	8	0	0	0
$s = 2$	8	8	4	4	10	10	0
$s = 3$	8	13	13	4	10	1	0
$s = 4$	8	13	4	4	10	1	0

(b) Upper bounds ub_{st} per station s, period t and iteration i ($p = 9$).

Table 7: The example problem and upper bounds.

l_{st}	1	2	3	4	5
1	8	1	1	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
z_t		13	8	10	0

Table 8: Stock after $t = 1$ has been planned ($p = 9$).

In the second period, $maxMargin = \max\{\min\{7;8\};\{8;9\};\{13;9\};\{13;9\}\} = 9$ is less than $k = g_2 - \sum_{s=1}^{S} l_{s2} = 14 - 1 = 13$. Looking at the upper bounds in Table 7b ($t = 2, i = 1$), it would seem possible to distribute the entire $k = 13$ containers in one step by stocking 7 additional containers in station 1 and 6 in station 2. The individual bounds as well as p would not be violated and no more than $k = 13$ units would be distributed on the whole. However, those 13 containers would still be in stock in period 3 because they are not consumed, thus leading to an overstock in this later period. Consequently, only *one* station (the one with the greatest margin, i.e., either 3 or 4) can safely be loaded to the maximum (station 3 in Table 9a) before the bounds must be recalculated. Note that any station thus loaded will not have to be considered again in this period because it has either been loaded to its maximum regarding p or its individual upper bound or because the margin was great enough to take the whole of k. After the upper bounds have been updated, the remaining $k = g_2 - \sum_{s=1}^{S} l_{s2} = 14 - (1 + 9) = 4$ units are assigned to station 1 with $maxMargin = 4$ (Table 9b). The final stocks with $p = 9$ after all periods have been planned can be found in Table 9c.

l_{st}	1	2	3	4	5
1	8	1	1	0	0
2	0	0	0	0	0
3	0	9	0	0	0
4	0	0	0	0	0
z_t		4	8	10	0

(a) Stock after the first stage of $t = 2$ has been planned.

l_{st}	1	2	3	4	5
1	8	5	5	0	0
2	0	0	0	0	0
3	0	9	0	0	0
4	0	0	0	0	0
z_t		0	4	10	0

(b) Stock after the second stage of $t = 2$ has been planned.

l_{st}	1	2	3	4	5
1	8	5	5	0	0
2	0	0	4	9	0
3	0	9	0	1	0
4	0	0	0	0	0

(c) Final stocks ($p = 9$).

Table 9: Stock after both iterations of $t = 2$, and the final stocks ($p = 9$).

Since a feasible solution with $p = f_{max} = 9$ could be found, this p might have been too high; it might also be exactly optimal but that cannot be confirmed until all lower values for p have been discarded. To this end, the upper bound of the binary search is set to $R := p = 9$. Note that this cuts the search space in half. The new p is $4 + \lfloor(9-4)/2\rfloor = 6$. Running the algorithm with this value will show that it also leads to a feasible solution, thus R is set to $p = 6$, making the new $p := 4 + \lfloor(6-4)/2\rfloor = 5$ which will also work making the final $p := 4 + \lfloor(5-4)/2\rfloor = 4$ which leads to the feasible solution shown in

Table 10a. Now $R = L = 4$, successfully ending the search. The actual tow train load can easily be derived by applying Equation 10, resulting in the loads in Table 10b. The optimal objective value in the example is $f_j^* = f_{sum}^* + f_{max}^* = 41 + 4 = 45$.

l_{st}	1	2	3	4	5
1	4	4	4	0	0
2	4	4	4	4	0
3	0	4	1	4	0
4	0	2	0	2	0

(a) Final stocks ($p = 4$).

x_{st}	1	2	3	4	5
1	4	7	0	4	0
2	4	7	0	8	6
3	6	4	7	6	6
4	6	2	13	2	8

(b) Final loading of the tugger ($p = 4$).

Table 10: Optimal result in the example with $p = f_{max}^* = 4$.

Proof of Correctness

Proposition A.1. *$\sum_{s=1}^{S} l_{st} \leq g_t, \forall\ t = 1, \ldots, T$ holds if and only if $l_{st'} \leq ub_{st'}, \forall s = 1, \ldots, S, t' \leq t$ in each iteration.*

Proof. First off, we will have to take a look at how the stock flow between periods works. Notice that during the course of the optimization l_{st} and ub_{st} are updated in each iteration. For ease of presentation we omitted an iteration index and assume that l_{st} and ub_{st} stem from the same iteration. By definition (Equation (6)),

$$l_{st} = \sum_{t'=1}^{t} (x_{st'} - d_{st'}) \tag{16}$$

$$\Leftrightarrow \quad l_{st} = \sum_{t'=1}^{t-n} (x_{st'} - d_{st'}) + \sum_{t'=t-n+1}^{t} (x_{st'} - d_{st'}) \tag{17}$$

$$\Leftrightarrow \quad l_{st} = l_{s,t-n} + \sum_{t'=t-n+1}^{t} (x_{st'} - d_{st'}) \tag{18}$$

$$\Rightarrow \quad l_{st} \geq l_{s,t-n} - \sum_{t'=t-n+1}^{t} d_{st'} \qquad \forall n \in \mathbb{N}, 1 \leq n < t, s = 1, \ldots, S \tag{19}$$

"$\rightarrow$" Assume that there is a solution with $l_{st'} > ub_{st'}$. By insertion, this inequality can be rearranged for at least one t, with $t' \leq t \leq T$:

$$
\begin{aligned}
& l_{st'} > ub_{st'} && (20)\\
\Leftrightarrow \quad & l_{st'} > \min\left\{ \min\left\{ \sum_{t'''=t'+1}^{t''} d_{st'''} + z_{t''} + l_{st''} \,\middle|\, t'' = t'+1,\dots,T \right\}; l_{st'} + k \right\} && (21)\\
\Rightarrow \quad & l_{st'} > \sum_{t''=t'+1}^{t} d_{st''} + z_t + l_{st} \qquad \vee \qquad l_{st'} > l_{st'} + k && (22)
\end{aligned}
$$

The inequality still holds if we replace l_{st} by applying Inequality (19):

$$
\begin{aligned}
& l_{st'} > \sum_{t''=t'+1}^{t} d_{st''} + z_t + l_{st'} - \sum_{t''=t'+1}^{t} d_{st''} && \vee && l_{st'} > l_{st'} + k && (23)\\
\Leftrightarrow \quad & 0 > z_t && \vee && 0 > k && (24)\\
\Leftrightarrow \quad & 0 > g_t - \sum_{s=1}^{S} l_{st} && \vee && 0 > g_{t'} - \sum_{s=1}^{S} l_{st'} && (25)\\
\Leftrightarrow \quad & \sum_{s=1}^{S} l_{st} > g_t && \vee && \sum_{s=1}^{S} l_{st'} > g_{t'} && (26)
\end{aligned}
$$

"$\leftarrow$" The inverse is also true. Assume that there is a solution with $\sum_{s=1}^{S} l_{st} > g_t$. This can have two reasons: First, too many bins have been stocked in this very period t. In this case:

$$
\begin{aligned}
& \sum_{s=1}^{S} l_{st} > g_t && && (27)\\
\Leftrightarrow \quad & 0 > k && && (28)\\
\Leftrightarrow \quad & l_{s't} > k + l_{s't} && \forall s' = 1,\dots,S && (29)\\
\Leftrightarrow \quad & l_{s't} > ub_{s't} && \forall s' = 1,\dots,S && (30)
\end{aligned}
$$

Or, bins have been poorly assigned in a preceding period $t' < t$, such that it becomes impossible to stock no more than g_t bins in t. In that case:

$$
\begin{aligned}
& \sum_{s=1}^{S} l_{st} > g_t && (31)\\
\Leftrightarrow \quad & 0 > z_t && (32)
\end{aligned}
$$

Let us assume, without loss of generality, that the offending period t' is $t-1$. According to Inequality (19),

$$
\begin{aligned}
& l_{s,t-1} \geq l_{st} + d_{st} && \forall s = 1, \dots, S && (33)\\
\Rightarrow \quad & \sum_{s=1}^{S} l_{s,t-1} \geq \sum_{s=1}^{S} l_{st} + \sum_{s=1}^{S} d_{st} && && (34)
\end{aligned}
$$

Inserting z_t from Inequality (32), which was shown to be strictly less than zero, the inequality becomes strict:

$$
\begin{aligned}
& \sum_{s=1}^{S} l_{s,t-1} > \sum_{s=1}^{S} (l_{st} + d_{st} + z_t) && && (35)\\
\Leftrightarrow \quad & \sum_{s=1}^{S} l_{s,t-1} > \sum_{s=1}^{S} ub_{s,t-1} && && (36)\\
\Rightarrow \quad & l_{s',t-1} > ub_{s',t-1} && \exists s' = 1, \dots, S && (37)
\end{aligned}
$$

□

Proposition A.2. *The algorithm optimally solves the problem.*

Proof. It is obvious that the optimal f_{max} objective value must lie in the interval between L and R, therefore the binary search will always find it, provided the inner loop is capable of constructing a corresponding feasible solution. This is indeed the case for any feasible value of p as the following proof by induction shows.

In the first period $t = 1$, the statement holds obviously true. Exactly $k = g_t$ (thus optimizing f_{sum}) containers are stocked such that p is not exceeded in any station while also observing the upper bounds ub_{st}. If that is not possible, then either the problem instance is inherently infeasible or p is too low; either way, p will be rejected and the binary search will try a larger value if possible. Apparently, in some individual period a lower p might be attainable if the upper bounds could be neglected. This is, however, neither possible nor necessary as Theorem 3.5 in conjunction with Proposition A.1 states.

Given a feasible assignment of l_{st} in period t, the algorithm will also correctly and optimally determine the $l_{s,t+1}$. Since $l_{st} \leq ub_{st}$, $\forall s = 1, \dots, S$, it follows, by Proposition A.1, that $\sum_{s=1}^{S} l_{s,t+1} \leq g_{t+1}$. Thus, there is no problem stocking exactly g_{t+1} parts, again observing the upper bounds as well as p.

The part of any $l_{s,t+1}$ that has already been determined in the earlier period is $l_{s,t+1} = l_{st} - d_{s,t+1}$ (see Inequality (19)). In consequence, no station can have more than p containers in stock in $t + 1$ because of a decision made earlier. □

Proof of polynomial runtime complexity

Proposition A.3. *The algorithm solves the problem in polynomial time.*

Proof. In its outer loop the algorithm performs a binary search in the interval $[L; R]$. The smallest value that L can take is 0, the value of R is $\max\{g_t \mid t = 1, \ldots, T\}$ which is bounded in the sum of the demands $\sum_{s=1}^{S} d_{st}$ for any given period t, ergo the interval to be examined grows exponentially with input size. The binary search algorithm, on the other hand, has a worst case performance of $O(log\ n)$, where n is the number of elements in the interval. The number of iterations through the outer loop is therefore polynomially bounded.

The runtime of the inner part of the algorithm that deals with constructing a feasible solution for a given p is obviously polynomially bounded as it contains only "for"-loops over either the number of stations or periods, the sole exception being the "while"-loop in Line 4. This loop, too, can only be iterated through S times at the most because after S iterations either $k = 0$ or all stations have already been loaded to their respective upper bounds, in which case *pTooSmall* becomes true and the loop exits. □

References

Apte, U.M.; Viswanathan, S. (2000): Effective cross docking for improving distribution efficiencies. International Journal of Logistics 3, 91–302.

Bartholdi, J.J. III; Hackman, S.T. (2008): Warehouse & Distribution Science, Release 0.89. http://www.warehouse-science.com.

Beaujon, G.J.; Turnquist, M.A. (1991): A Model for Fleet Sizing and Vehicle Allocation. Transportation Science 25, 19–45.

Bell, W.J.; Dalberto, L.M.; Fisher, M.L.; Greenfield, A.J.; Jaikumar, R.; Kedia, P.; Mack, R.G.; Prutzman, P.J. (1983): Improving the Distribution of Industrial Gases with an On-Line Computerized Routing and Scheduling Optimizer. Interfaces 13, 4–23.

Boysen, N.; Fliedner, M.; Scholl, A. (2009a): Level scheduling under storage constraints. International Journal of Production Research 47, 2669–2684

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

Boysen, N.; Fliedner, M.; Scholl, A. (2009c): Level Scheduling for batched JIT supply. Jena Research Papers in Business and Economics (JBE) 01/2009, FSU Jena, Germany.

Campbell, A.; Clarke, L.; Kleywegt, A.; Savelsbergh, M.W.P. (1998): The Inventory Routing Problem. In: Crainic, T.G.; Laporte, G. (eds.): Fleet Management and Logistics. Kluwer Academic Publishers, Dordrecht, Netherlands, 95–113.

Cordeau, J.-F.; Laporte, G.; Savelsbergh, M.W.P.; Vigo, D. (2007): Vehicle Routing. In: Barnhart, C.; Laporte, G. (eds.): Handbook in Operations Research & Management Science 14, Elsevier, Amsterdam, Netherlands, 367–428.

Dahmala, 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.

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.

Fisher, M.L.; Greenfield, A.J.; Jaikumar, R.; Lester, J.T. III (1982): A Computerized Vehicle Routing Application. Interfaces 12, 42–52.

Hodgson, T.J.; Wang, D. (1991): Optimal hybrid push/pull control strategies for a parallel multistage system: part I. International Journal of Production Research 29, 1279–1287

Holweg, M. (2007): The genealogy of lean production. Journal of Operations Management 25, 420–437.

Kleywegt, A.J.; Nori, V.S.; Savelsbergh, M.W.P. (2002): The Stochastic Inventory Routing Problem with Direct Deliveries. Transportation Science 36, 94–118.

Kleywegt, A.J.; Nori, V.S.; Savelsbergh, M.W.P. (2004): Dynamic programming approximations for a stochastic inventory routing problem. Transportation Science 38, 42–70.

Klose, A.; Drexl, A. (2005): Facility location models for distribution system design. European Journal of Operational Research 162, 4–29.

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

Monden, Y. (1998): Toyota Production System: An integrated approach to just-in-time, third ed. Engineering & Management Press, Norcoss, GA.

Rees, L.P.; Huang, P.Y.; Taylor III, B.W. (1989): A comparative analysis of an MRP lot-for-lot system and a Kanban system for a multistage production operation. International Journal of Production Research 27, 1427–1443.

Spencer, M. (1995): Production planning in a MRP/JIT repetitive manufacturing environment. Production Planning & Control 6, 176–184.

Vatalaro, J.; Taylor, R. (2005): Implementing a Mixed Model Kanban System: The Lean Replenishment Technique for Pull Production. Productivity Press, Portland, OR.