Technische Universität Chemnitz
Fakultät für Mathematik
D-09107 Chemnitz, Germany

A Case Study of Joint Online Truck Scheduling and Inventory Management for Multiple Warehouses

C. Helmberg, S. Röhl

Preprint 2005-3

Preprintreihe der Fakultät für Mathematik
ISSN 1614-8835

A Case Study of Joint Online Truck Scheduling and Inventory Management for Multiple Warehouses*

C. Helmberg† S. Röhl‡

January 2005

Abstract

For a real world problem — transporting pallets between warehouses in order to guarantee sufficient supply for known and additional stochastic demand — we propose a solution approach via convex relaxation of an integer programming formulation, suitable for online optimization. The essential new element linking routing and inventory management is a convex piecewise linear cost function that is based on minimizing the expected number of pallets that still need transportation. For speed, the convex relaxation is solved approximately by a bundle approach yielding an online schedule in 5 to 12 minutes for up to 3 warehouses and 40000 articles; in contrast, computation times of state of the art LP-solvers are prohibitive for online application. In extensive numerical experiments on a real world data stream, the approximate solutions exhibit negligible loss in quality; in long term simulations the proposed method reduces the average number of pallets needing transportation due to short term demand to less than half the number observed in the data stream.

Keywords: convex relaxation, integer programming, stochastic demand, network models, large scale problems, bundle method, logistics, vehicle routing

MSC 2000: 90B06; 90C06, 90C90, 90B05

1 Introduction

Consider the following real world problem. Given several warehouses connected by a shuttle service of several trucks for shipping pallets of stored articles between them; given also an online stream of orders, that are stochastic in nature and that have to be handled within short time at specific warehouses. Is it possible to provide, online, a schedule of

*This work was supported by research grant 03HEM2B4 of the German Federal Ministry of Education and Research. Responsibility for the content rests with the authors.

†Fakultät für Mathematik, Technische Universität Chemnitz, D-09107 Chemnitz, Germany, helmberg@mathematik.tu-chemnitz.de

‡Fachhochschule Vorarlberg, Achstr. 1, A-6850 Dornbirn, Austria, stefan.roehl@fh-vorarlberg.ac.at

truck routes and truck loads so that all products are available at the right place ahead of processing time even if the realization of the schedule is subject to further uncertainties?

We suggest a solution approach based on convex relaxation and demonstrate its practical suitability on real world data of our industrial partner eCom Logistik GmbH & Co. KG. For up to three warehouses and roughly 40000 articles the method computes a schedule within five to twelve minutes. In long term simulations it reduces the average number of pallets that have to be transported on short notice due to demand to less than half the number of the current semi-automatic approach.

Several issues are of relevance in this problem: an appropriate stochastic optimization model is required that links the success probability of the inventory of the warehouses to the truck rides; the model must be solvable within short time in order to be suitable for online computations; the approach must be sufficiently robust to compensate frequent external changes in orders and the logistic transportation process.

In our method we follow the classical approach to model large scale transportation or network design problems as multicommodity flow problems (see e.g. [20, 17, 15]). These can be decomposed and solved efficiently via Lagrangian relaxation by combining min-cost flow algorithms (see e.g. [1]) and bundle methods (see e.g. [11, 5]). In particular, we model the rides of the trucks as well as the flow of pallets between warehouses by time discretized networks coupled via linear capacity constraints. Our main contribution is probably the development of a convex piecewise linear cost function, that models the stochastic quality of the warehouse configurations. Its primary aim is to minimize the expected number of pallets that have yet to be transported. Due to its favorable structure, even moderately accurate solutions seem to give rise to reasonable schedules. This allows the use of the aforementioned fast approximate solvers suitable for online optimization.

In practice, robustness within online environments hinges on reacting flexibly to new situations rather than sticking to past decisions. Consequently, our method does not keep any information on previous solutions but operates solely on status messages of the logistic operating system (the message system has been developed jointly with our industrial partner for this purpose). Therefore, the method is capable of continuing independent on what part of its proposed solution has been accepted by the human planner. We do not know how to quantify and measure the importance of this setup mathematically, but this concept appears to be vital for the success of the approach.

There is a vast literature on inventory management and logistics (see e.g. [8]), yet we found very few references that deal with both problems at the same time; none of them, however, treat both problems in sufficient detail for our purposes. In some works the transportation process is assumed to be instantaneous (see e.g. [13, 14, 6, 3, 22]), in others the stochastic part is fixed (see e.g. [2, 7]) or considerations are reduced to only one product [18]. To the best of our knowledge the approach proposed here is the first that deals jointly with inventory management of multiple products and inter warehouse logistics involving vehicle routing with transportation times.

The contents is structured as follows. Section 2 gives the necessary background on the real world problem. Next we present our optimization model in two steps: in Section 3 we formulate the set of feasible solutions by introducing the networks, variables, and constraints; Section 4 is devoted to the cost function. Implementational aspects such as the

generation of distribution data, the approach for solving the relaxation, and the rounding heuristic are described in Section 5. Extensive computational results on the real world data stream of our industrial partner are presented in Section 6; these include comparative runs with exact solution methods and a simulation run over 100 days for two and three warehouses. Finally, we offer some conclusions and outline possible enhancements in Section 7.

2 Problem Description

Our industrial partner, eCom Logistik GmbH & Co. KG, operates several warehouses in different locations within the same city and offers logistics services to business customers. In particular, it stores the products of a business customer and processes orders addressed to the business customer by picking and packing the ordered items into boxes or on pallets and passing them on to a shipping agency that takes care of the final delivery to the correct address. E.g., a startup company selling via the Internet could contract eCom Logistik for storing and delivering its products.

The business model implies important differences to standard inventory management problems. First, the task of our partner is to deliver, upon request, the goods stored but it is not its responsibility that sufficient goods are within its store, so the standard scenario of "ordering problems" does not apply. Second, there is no information available about the customers expectations on the development of demand. Therefore stochastic demand forecasts must be based on past demand for a particular product alone. The same is true for supply shipments by the customer for replenishing the store. These are mostly unpredictable in the sense that they are rare singular events of strongly varying size. Finally, knowledge about the products is restricted to an article identifier number and – at best – the number of items of this article to be expected on a typical pallet.

Due to the structure of the customers (a major customer is the Herlitz PBS AG, a large company producing stationery) a typical order is placed by small to medium sized retailers and consists of a long list of articles, that are quite divers in nature. Such an order must be picked and delivered to the shipping agency within the next work day, i.e., orders are accepted till 12 am, delivery of these orders starts at 2 pm and should be finished till 2 pm the next day. When such an order arrives, it is prescheduled to a certain warehouse and time slot for picking. At this time, all the items on the list have to be available in the picking lines of the selected warehouse so that the entire order can be shipped in one unit. The picking lines are replenished automatically or by local personnel by requesting a new pallet of the respective article from the automatic storage system. Due to size restrictions of the storage system or due to simultaneous demand at various locations it is not always possible to hold sufficient supply at all locations, i.e., pallets have to be shipped between the warehouses on time. For this purpose the company operates a shuttle service of trucks. The task we address in this paper is to determine a schedule for the trucks and their load of pallets so that "with high probability" the necessary supply is available in time. The current solution method used in practice is half automatic. Pallets are automatically put on a list if the available amount of an article falls short of a given minimum for this article. The minima are set by some automatic rules and are controlled by a human dispatcher,

who regularly initiates the transportation of pallets for known short term demand and for pallets on this list.

The logistic process of transporting the pallets entails further uncertainties which we currently cannot model to our full satisfaction, yet they have a strong influence on our approach. We start by describing the path of a pallet from one warehouse to another. When a specific pallet is to be transported, a retrieval order is added to the request queue of the automatic stacker crane in the respective aisle of the automatic storage system. Depending on the number of pending requests, retrieval of the pallet from the automatic storage system may need between two minutes and half an hour. Then the pallet is maneuvered by local routing decisions over a conveyer system towards one of several waiting queues in front of the automatic loading platforms for the trucks. As soon as the destination of a loading platform is set for the next transport, another algorithm selects pallets from the beginning of the waiting queues according to some rules depending on the space requirements of the specific pallets rather than on their due dates. If the platform is full and a truck has docked, all pallets are loaded automatically in one step; loading is not time critical. The driving time of the truck, however, depends strongly on whether streets are congested or not. Finally, after automatic unloading at the destination the storing time again depends on the position of the pallet within the truck-load and on the congestion of conveyor system and automatic storage system. In practice, typical travel times of a pallet vary between 1 to 6 hours plus driving time. By far the majority of these pallets is initiated by the dispatcher of the trucks. Some pallets, however, may also be started by other personnel for reasons not visible to the dispatcher; due to such pallets or due to packing problems it may happen that pallets cannot be loaded on the next truck. Such pallets may then block some of the waiting queues for quite some time if the next few transports do not serve the desired direction.

Together with our industrial partner a new data protocol was developed for efficient online updating of the current ordering and inventory status known to the system software. The latter need not reflect the true state of the system due to the asynchronous nature of the underlying logistic system, i.e., certain bookings may arrive significantly later, because they are entered by humans or because of communication delays between the warehouses. The messages of the protocol give a complete online account of

- article basics (in particular: the article ID and amount of the article that is on a typical pallet; it may be zero if the data is not supplied by customer),
- header information for orders (holds the scheduled time slot for picking),
- delivery items per order (article ID and amount to deliver),
- picks (reports that [a part of] the amount of a delivery item has been fulfilled),
- stock movements (between real and/or virtual storage systems),
- reinitializations of stock data,
- scheduled transportations of particular pallets (giving the article ID, planned time of retrieval from the automatic storage system and the source and destination warehouses),

- pallets currently in transport (those having reached a loading platform),
- available truck capacities (per truck a time period when it is available and the expected number of pallets it can load),
- and corresponding deletions.

"Current" stock and demand can be updated efficiently with these messages. We also use this data for generating demand distributions, see §5.1. Note, because of the business model there is certainly enough stock in the distributed storage system to cover the entire demand, even if the current figures show negative stock at single warehouses. The latter may occur due to the asynchronous nature of the logistic message system. Unfortunately, no information is available on the current position of the trucks or on the direction of their next ride.

In practice, our solution of the optimization process yields a suggestion for the dispatcher of the trucks who will then fix a route and select particular pallets for transportation. Depending on possible additional oral information not available in the operating system, the dispatcher may or may not follow the suggestions. The only feedback on these decisions are new messages on planned transportations of particular pallets.

For acceptance as well as practical reasons there is a rather strict priority order for the sequence in which pallets should be transported by the trucks.

Priority Level 1: Pallets that have been made known to the system via scheduled transportation messages should be served as quickly as possible. These pallets have been requested by a user of the logistic system at this time hopefully for a good reason (e.g. the user could be the dispatcher of the trucks). They will start moving at the prescheduled time, independent of what is decided elsewhere and if they are not transported, they will likely block the way of other pallets.

Priority Level 2: Pallets that are needed to cover the known demand of the next six days should be transported in the sequence of their due dates.

Priority Level 3: If there is still room on the trucks, further pallets may be transported for supply management based on demand forecasts. Among these the priority order should reflect the probability that the pallet is needed within the next three days, say.

No actual costs are known for delays in transportation or for the violation of due dates. For inventory management purposes stochastic models often assume a certain amount of available space and ask for the best use of this space in a probabilistic sense. It was the explicit wish of our industrial partner not to use such a concept, because the amount of available space is itself a highly uncertain parameter in an asynchronous logistic environment and depends on several other factors (e.g. depending on the size of a pallet there may be room in the automatic storage system or not; also, upon need the dispatcher may open up some intermediate storage facilities). Therefore they saw no possibility to provide the amount of free storage space automatically. Rather it was agreed that the amount of pallets transported for stock-keeping purposes should be controlled via an "upper probability level" $\overline{\pi} \in (0, 1)$ measuring the probability that available stock of an article at a

warehouse suffices for a given period of days with respect to an appropriate stochastic model of demand. If stock exceeds the upper level $\overline{\pi}$, no further pallets should be brought in for this article. Without information on available storage it is difficult make room by removing superfluous articles, and indeed our industrial partner did not wish to shift stock for such purposes without human interaction. We might add that our current approach could easily be extended to such tasks if appropriate information is provided.

There are a number of additional technical restrictions on feasible truck schedules (e.g. due to the number of automatic loading platforms and time requirements) as well as on the assignment of pallets to trucks that we will mention shortly in describing our optimization model, but a detailed description of these aspects does not seem appropriate.

3 Optimization Model, Part I: The Feasible Set

For modeling the route of trucks and the movement of pallets we use the standard approach of time discretized network flows coupled by linear constraints.

We assume the time discretization to be given (in minutes) as a finite sequence of nonnegative integers $0 \leq t_1 < \cdots < t_{n_\mathcal{T}}$, $n_\mathcal{T} \in \mathbb{N}$ and set $\mathcal{T} = \{t_i : 1 \leq i \leq n_\mathcal{T}\}$. For rounding up a $\tau \in \mathbb{R}$ with respect to $\mathcal{T}$ we define

$$\lceil \tau \rceil_\mathcal{T} = \begin{cases} \min\{t \in \mathcal{T} : t \geq \tau\} & \text{if } \tau \leq \max \mathcal{T} \\ \infty & \text{otherwise.} \end{cases}$$

Likewise, rounding down is defined as

$$\lfloor \tau \rfloor_\mathcal{T} = \begin{cases} \max\{t \in \mathcal{T} : t \leq \tau\} & \text{if } \tau \geq \min \mathcal{T} \\ -\infty & \text{otherwise.} \end{cases}$$

We denote the set of warehouses at different locations by $\mathcal{W}$ with $|\mathcal{W}| = n_\mathcal{W} \in \mathbb{N}$. To allow for a separate truck depot $d \notin \mathcal{W}$ we set $\mathcal{W}_d = \mathcal{W} \cup \{d\}$. Furthermore we denote the set of different products (articles) by $\mathcal{P}$ with $|\mathcal{P}| = n_\mathcal{P}$. We collect all trucks that have not to be discerned (e.g. in terms of capacity or compatibility with certain products) in truck classes and collect all truck classes in a set $\mathcal{R}$ with $|\mathcal{R}| = n_\mathcal{R} \in \mathbb{N}$ and $\mathcal{R} \cap \mathcal{P} = \emptyset$.

Next we specify the network structure via directed multigraphs $D = (\mathcal{V}, \mathcal{A})$ with $\mathcal{V}$ the set of nodes and $\mathcal{A}$ the set of arcs consisting of ordered pairs of nodes. The arcs will be supplemented with a subscript like $(v, w)_r$ if we have to discern several arcs running from a node v to another node w, but we simply write (v, w) if the meaning is clear from the context. Balances will be specified via a function $b : \mathcal{V} \to \mathbb{Z}$ and lower and upper bounds by means of functions $l : \mathcal{A} \to \mathbb{Z}$ and $u : \mathcal{A} \to \mathbb{Z} \cup \{\infty\}$.

3.1 Truck Graphs

For each class of trucks $r \in \mathcal{R}$ the basic structure of the graph is the same, up to slight differences in the arc sets. The latter are due to differences in driving speed and loading or unloading properties. We consider a fixed $r \in \mathcal{R}$ in the following. Let two driving time functions

$$\underline{T}_r : \mathcal{W}_d \times \mathcal{W}_d \times \mathcal{T} \to \mathbb{R} \qquad \text{and} \qquad \overline{T}_r : \mathcal{W} \times \mathcal{W} \times \mathcal{T} \to \mathbb{R} \tag{1}$$

be given that specify the expected driving time of a truck of truck class r from one warehouse to another one that start at a time step $t \in \mathcal{T}$. The function $\underline{T}$ yields the driving time for empty transfers while $\overline{T}$ gives the time if the truck transports pallets including loading and unloading times; since, in addition, a loaded truck goes slower, the differences between $\underline{T}_r$ and $\overline{T}_r$ are significant.

The node set $\mathcal{V}_r$ of the graph $D_r = (\mathcal{V}_r, \mathcal{A}_r)$ includes, for each time step $t \in \mathcal{T}$ and each warehouse $w \in \mathcal{W}$, three node types discerned by the names C (courtyard), L (loading), and U (unloading) and, for each time step $t \in \mathcal{T}$, a depot node type named C, so

$$\mathcal{V}_r = \mathcal{W} \times \{C, L, U\} \times \mathcal{T} \quad \cup \quad \{d\} \times \{C\} \times \mathcal{T}.$$

For extracting components of $v = (w, a, t) \in \mathcal{V}_r$ we introduce the short hand notation $v_W = w$, v_N=a, and $v_T = t$. An illustration of the basic graph structure is given in Fig. 1.

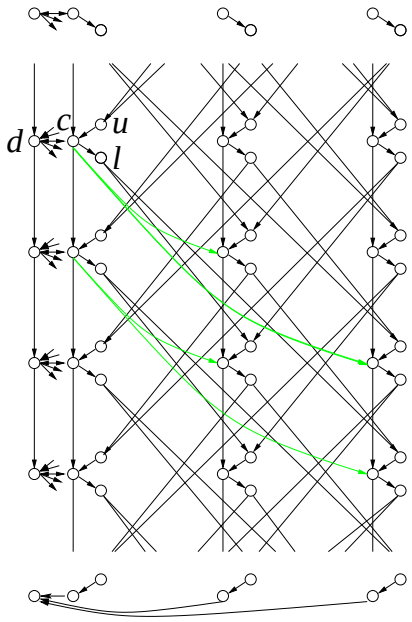

Figure 1: Basic structure of a truck graph for three warehouses (not all nodes and edges are shown). Node c corresponds to the node (w_1, C, t_i) representing the courtyard of warehouse w_1 at time t_i, node l represents the loading node (w_1, L, t_i), u the unloading node (w_1, U, t_i), and d the depot node (d, C, t_i).

For future reference we introduce the arc set via several different classes of arcs. First, we collect all arcs that just lead on to the next time step for the courtyard node and the depot node,

$$\mathcal{A}_r^B = \{(u, v) : u, v \in \mathcal{V}_r, u_W = v_W, u_N = v_N = C, v_T = \lceil u_T + 1 \rceil_{\mathcal{T}}\}.$$

Next, we collect unloading arcs in

$$\mathcal{A}_r^U = \{(u, v) : u, v \in \mathcal{V}_r, u_W = v_W, u_N = U, v_N = C, v_T = u_T\}.$$

Likewise, the loading arcs are

$$\mathcal{A}_r^L = \{(u,v) : u,v \in \mathcal{V}_r, u_W = v_W, u_N = C, v_N = L, v_T = u_T\}.$$

Arcs corresponding to empty transfers go from one courtyard to that of another warehouse,

$$\mathcal{A}_r^E = \{(u,v) : u,v \in \mathcal{V}_r, u_W \neq v_W, u_N = v_N = C, v_T = \lceil u_T + \underline{T}_r(u_W, v_W, u_T)\rceil_{\mathcal{T}}\}.$$

Finally, the arcs corresponding to pallet transports are

$$\mathcal{A}_r^P = \{(u,v) : u,v \in \mathcal{V}_r, u_W \neq v_W, u_N = L, v_N = U, v_T = \left\lceil u_T + \overline{T}_r(u_W, v_W, u_T)\right\rceil_{\mathcal{T}}\}.$$

This completes the arc set for truck graph D_r,

$$\mathcal{A}_r = \mathcal{A}_r^B \cup \mathcal{A}_r^U \cup \mathcal{A}_r^L \cup \mathcal{A}_r^E \cup \mathcal{A}_r^P.$$

Next, we specify lower and upper bounds on the arc values as well as the balances. Coupling constraints between the graphs will be discussed in §3.4. Lower bounds are uniformly set to zero,

$$l_r(a) = 0 \quad \text{for all } a \in \mathcal{A}_r.$$

Upper bounds are set to infinity except for the unloading arcs and the loading arcs. For the latter two the numbers must correspond to the number of trucks that can be unloaded/loaded within the given previous/following time span by the available automatic loading platforms. For simplicity, we assume that these numbers are independent of the truck class (which is not necessarily the case in the practical application) and given by functions depending on the warehouse and the time step,

$$\lambda_U : \mathcal{W} \times \mathcal{T} \to \mathbb{N}_0 \quad \text{and} \quad \lambda_L : \mathcal{W} \times \mathcal{T} \to \mathbb{N}_0. \tag{2}$$

With these functions the upper bounds read

$$\begin{aligned} u_r(a) &= \infty \quad \text{for all } a \in \mathcal{A}_r^B \cup \mathcal{A}_r^E \cup \mathcal{A}_r^P, \\ u_r((u,v)) &= \lambda_U(u_W, u_T) \quad \text{for all } (u,v) \in \mathcal{A}_r^U, \\ u_r((u,v)) &= \lambda_L(u_W, u_T) \quad \text{for all } (u,v) \in \mathcal{A}_r^L. \end{aligned}$$

Balances reflect the availability of trucks and are also given by user data. For completeness, let the function

$$n_r : \mathcal{W}_d \times \mathcal{T} \to \mathbb{Z}$$

specify the number $n_r(w,t)$ of trucks of class r that are added/subtracted at location $w \in \mathcal{W}_d$ and time $t \in \mathcal{T}$, then

$$\begin{aligned} b_r(v) &= 0 \quad \text{for all } v \in \mathcal{V}_r \text{ with } v_N \neq C, \\ b_r(v) &= n_r(v_W, v_T) \quad \text{for all } v \in \mathcal{V}_r \text{ with } v_N = C. \end{aligned}$$

In practice the exact starting or ending position of the trucks may not be given or may not be known because the time window for using the truck exceeds the time span of $\mathcal{T}$. In this case, we let trucks start or end in the depot.

3.2 Article Graphs

There is no use in modeling every single pallet that is currently stored in the automatic storage system, because any particular pallet seemingly available at the beginning of the calculations may no longer be available when the schedule has been computed. Therefore we only consider, for each article ID, idealized identical pallets that carry the amount promised by the article basics (and a single pallet carrying the entire stock of one warehouse if no such data is available)[1]. So for each article $p \in \mathcal{P}$ the stock at each warehouse is discretized to standard pallets and there is one network per article p that models the movement of these pallets. Let $p \in \mathcal{P}$ be fixed for the rest of this section.

The node set $\mathcal{V}_p$ of the graph $D_p = (\mathcal{V}_p, \mathcal{A}_p)$ contains, for each time step $t \in \mathcal{T}$ and each warehouse $w \in \mathcal{W}$, two nodes discerned by the names A (automatic storage system) and B (transportation buffer), and one artificial node named $(d, C, t_{n_\mathcal{T}})$ (collect),

$$\mathcal{V}_p = \mathcal{W} \times \{A, B\} \times \mathcal{T} \cup \{(d, C, t_{n_\mathcal{T}})\}.$$

The reason for introducing a transportation buffer is that our industrial partner wanted to be able to restrict the total increase of pallets in the automatic storage system suggested by the automatic planning tool. Like for the truck graphs, we reference the components of a node $v = (w, a, t) \in \mathcal{V}_p$ by $v_W = w$, $v_N = a$, and $v_T = t$. An illustration of the basic graph structure is given in Fig. 2.

We start with the arcs leading from one time step to the next in the automatic storage system,

$$\mathcal{A}_p^A = \{(u, v) : u, v \in \mathcal{V}_p, u_W = v_W, u_N = v_N = A, v_T = \lceil u_T + 1 \rceil_\mathcal{T}\},$$

and in the buffer,

$$\mathcal{A}_p^B = \{(u, v) : u, v \in \mathcal{V}_p, u_W = v_W, u_N = v_N = B, v_T = \lceil u_T + 1 \rceil_\mathcal{T}\}.$$

Next, there are arcs leading from the automatic storage system to the transportation buffer,

$$\mathcal{A}_p^{AB} = \{(u, v) : u, v \in \mathcal{V}_p, u_W = v_W, u_N = A, v_N = B, u_T = v_T\}.$$

Pallets that moved into the transportation buffer can only be removed from the buffer via balances reflecting demand or by transportation to the transportation buffer of another warehouse. Remaining pallets will finally be collected in the node $(d, C, t_{n_\mathcal{T}})$.

Let the time needed to prepare a pallet for loading and to store it after unloading be specified by the functions

$$T^L : \mathcal{W} \to \mathbb{R}_+ \quad \text{and} \quad T^U : \mathcal{W} \to \mathbb{R}_+.$$

More precisely, $T^L(w)$ gives, for warehouse $w \in \mathcal{W}$, the time needed to retrieve a pallet from the automatic storage system and to maneuver it to the automatic loading platform,

[1] An attractive alternative would be to compute a statistically representative amount from previous pallet data. Our industrial partner, however, strongly preferred to rely on the given article basics that are within the responsibility of the customer.

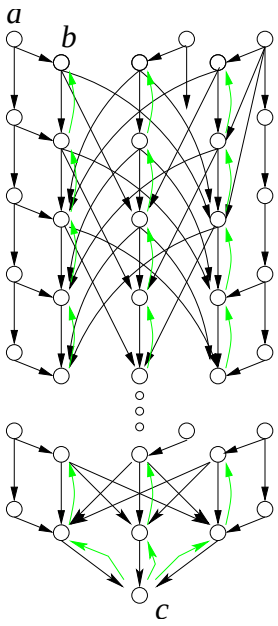

Figure 2: Basic structure of an article graph for three warehouses (not all nodes and edges are shown). Node a corresponds to the node (w_1, A, t_1) representing the automatic storage system of warehouse w_1 at time t_1, node a represents the buffer node (w_1, B, t_1), and c the collect node $(d, C, t_{n_\mathcal{T}})$.

$T^U(w)$ is the time needed to move the pallet from the loading platform to the automatic storage system and store it there.

Denote by $\mathcal{R}(p) \subseteq \mathcal{R}$ the truck classes that can be used to transport article p. The transportation arcs between the buffer nodes are introduced for each arc $a \in \mathcal{A}_r^P$, $r \in \mathcal{R}(p)$, with the property that starting time minus preparation time and ending time plus storing time fit into the given time steps of $\mathcal{T}$,

$$\begin{aligned}\mathcal{A}_p^P = \{(u,v)_r : \quad & u, v \in \mathcal{V}_r, r \in \mathcal{R}(p), (u^r, v^r) \in \mathcal{A}_r^P, u_W = u_W^r, v_W = v_W^r, u_N = v_N = B, \\ & u_T = \left\lfloor u_T^r - T^L(u_W) \right\rfloor_{\mathcal{T}}, v_T = \left\lceil u_T^r + \overline{T}_r(u_W, v_W, u_T^r) + T^U(v_W) \right\rceil_{\mathcal{T}}\}.\end{aligned} \tag{3}$$

After the last time step, all remaining pallets are collected in the artificial node $(d, C, t_{n_\mathcal{T}})$,

$$\mathcal{A}_p^C = \{((w, B, t_{n_\mathcal{T}}), (d, C, t_{n_\mathcal{T}})) : w \in \mathcal{W}\}.$$

Quite often it is impossible to find a schedule satisfying the balances given in practice, because either time limits are simply too tight or the number of available pallets does not suffice to cover the demand. Therefore a reasonable strategy to cope with infeasibilities is essential. For the violation of due dates it was agreed that the delay of pallets should be as small as possible; infeasibilities caused by lack of supply are compensated by artificial pallets supplied in the artificial node $(d, C, t_{n_\mathcal{T}})$. Both concepts can be incorporated in the graph by allowing for arcs going backwards in time from $(d, C, t_{n_\mathcal{T}})$ to the last buffer nodes,

$$\mathcal{A}_p^I = \{((d, C, t_{n_\mathcal{T}}), (w, B, t_{n_\mathcal{T}})) : w \in \mathcal{W}\}$$

and reverse arcs to $\mathcal{A}_p^B$ for lateness,

$$\mathcal{A}_p^L = \{(u,v) : u,v \in V_p, u_W = v_W, u_N = v_N = B, u_T = \lfloor v_T - 1 \rfloor_{\mathcal{T}}\}.$$

This completes the set of arcs for article graph D_p,

$$\mathcal{A}_p = \mathcal{A}_p^A \cup \mathcal{A}_p^B \cup \mathcal{A}_p^{AB} \cup \mathcal{A}_p^P \cup \mathcal{A}_p^C \cup \mathcal{A}_p^I \cup \mathcal{A}_p^L.$$

Remark 1 *The current infeasibility setting entails the danger, that demand that cannot be satisfied in time in one warehouse is compensated by pallets from a second warehouse, that has itself insufficient pallets but that can be refilled quicker from a third warehouse. We do not know how to avoid such behavior without making explicit or implicit use of bilinear constraints and that type of constraints is computationally too expensive. In our current practical instances such configurations are impossible because each product is kept in two warehouses at most. Likewise such configurations cannot appear if for each article a special supply warehouse is declared, so that supply may only be brought from there. The latter is a typical constraint in practice. A different approach useful in some applications is that pallets are not allowed to be late at all. In such a setting, infeasibilities are easily modeled by adding an artificial initial node that provides sufficient supply over "infeasible" arcs. The ultimate decision on which setting produces the most realistic solutions must be found in cooperation with the industrial partner.*

The lower and upper bounds on the arc capacities are all set to the usual values,

$$l_p(a) = 0, u_p(a) = \infty \quad \text{for all } a \in \mathcal{A}_p.$$

These values could also be used to install article dependent bounds on the maximum amount to be stored in each warehouse.

Setting up the balances for these graphs involves rounding the available amounts and demands to pallets. This requires some care and several decisions that are motivated heuristically or by practical experience. In order to describe our rounding method we consider for article p a single warehouse $w \in \mathcal{W}$. For this warehouse we form a list of pairs (α_i, τ_i), $i \in \{1, \ldots, k\}$ with $k \in \mathbb{N}$, consisting of positive or negative amounts $\alpha_i \in \mathbb{R}$ and sorted times $\tau_i \in \mathbb{R}_+$ satisfying $\min \mathcal{T} = \tau_1 \leq \cdots \leq \tau_k$. By α_0 we denote the amount currently available at warehouse w; note, that α_0 may be negative due to asynchronous reports of the logistic system. The case of $\alpha_0 < 0$ should lead to the transportation of pallets only in connection with actual demand or at the priority level 3 in connection with stock keeping. Further negative numbers $\alpha_i < 0$ represent demands scheduled for time τ_i either due to orders or prescheduled transport pallets (see §3.3) and positive numbers $\alpha_i > 0$ correspond to expected additions to the store (again caused by prescheduled transport pallets). In rounding, we need to discern the case where the article basics to p list the standard amount θ_p stored on a pallet of p, i.e., $\theta_p > 0$, and the case that no such data is known, $\theta_p = 0$.

If $\theta_p > 0$ then the conceptual setting is as follows. As long as there is no demand at w, the complete amount is available for transport. The number of pallets is the round up of the available amount divided by θ_p. In general the pallet count includes one pallet that

only holds a fraction of θ_p. We call this the *fractional pallet* and denote the difference of the sum of the pallets times θ_p to the true amount by $\underline{\alpha}_p^w$ (since initially we may count too much or there may be negative stock, this number may be negative). Upon the first occurrence of demand at warehouse w a pallet is moved from the automatic storage system to the picking lines, this pallet is then no longer available for transportation. The first pallet fetched is defined to be the fractional pallet. Further pallets will be retrieved whenever the fractional amount in the picking lines falls below zero, further pallets will be added to the automatic storage system whenever the fractional amount exceeds θ_p. We give the algorithm in pseudo code, using the convention of the programming language C that += means adding the right hand side number to the variable on the left.

Algorithm 2 *(computes balances for article p and warehouse w)*
Input: $\theta_p > 0$, *initial supply* $\alpha_0 \in \mathbb{R}$, *demands/supplies* (α_i, τ_i), $i \in \{1, \ldots, k\}$
Output: balances and amount $\underline{\alpha}_p^w$ *accounting for the difference in a fractional pallet*
1. *set* $b_p(v) = 0\ \forall v \in \{v' \in \mathcal{V}_p : v'_W = w\}$ *and* $i = 1$;
2. *set* $d = \max\{0, \lceil \alpha_0/\theta_p \rceil\}$, $\rho = \alpha_0 - d\theta_p$ *and* $b_p((w, A, t_1)) = d$;
3. *while* $(i \leq k)$ *and* $(\alpha_i >= 0)$ *do*
   $d = \max\{0, \lceil (\rho + \alpha_i)/\theta_p \rceil\}$;
   ρ += $\alpha_i - d \cdot \theta_p$;
   $b_p((w, A, \lfloor \tau_i \rfloor_{\mathcal{T}}))$ += d;
   i += 1;
   end while
4. *while* $(i \leq k)$ *do*
   $d = \lfloor (\rho + \alpha_i)/\theta_p \rfloor$;
   ρ += $\alpha_i - d\theta_p$;
   if $(d > 0)$ *then* $b_p((w, A, \lfloor \tau_i \rfloor_{\mathcal{T}}))$ += d;
   else $b_p((w, B, \lfloor \tau_i \rfloor_{\mathcal{T}}))$ += d;
   i += 1;
   end while
5. *set* $\underline{\alpha}_p^w = \rho$.

Note, that all additions to the warehouse are entered at the nodes A corresponding to the automatic storage system, while retrievals (negative d) are fetched from the buffer node B. The purpose is to remove pallets from the transport buffer as soon as a pallet of this article is requested.

In the case of $\theta_p = 0$ (i.e., the article basics of the customers do not give any information on the standard size of a pallet of article p), it was agreed to handle this equivalently to $\theta_p = \infty$ meaning there is one pallet carrying the entire amount of the warehouse. This way at most one pallet is scheduled for transport in every call to the planning routine and if the pallet delivered does not suffice, the next pallet will be scheduled on the next call.

Algorithm 3 *(computes balances for article p and warehouse w in the case $\theta_p = 0$)*
Input: $\theta_p = 0$, *initial supply* $\alpha_0 \in \mathbb{R}$, *demands/supplies* (α_i, τ_i), $i \in \{1, \ldots, k\}$
Output: balances and the sum of the amounts $\underline{\alpha}_p^w$
1. *set* $b_p(v) = 0\ \forall v \in \{v' \in \mathcal{V}_p : v'_W = w\}$;

2. if ($\alpha_0 > 0$) set $d = 1$ else set $d = 0$;
3. set $\rho = \alpha_0$ and $b_p((w, A, t_1)) = d$;
4. while ($i \leq k$) and ($\alpha_i >= 0$) do
if ($\rho \leq 0$) and ($\rho + \alpha_i > 0$) set $b_p((w, A, \lfloor \tau_i \rfloor_{\mathcal{T}})) = 1$;
ρ += α_i;
i += 1;
end while
5. if ($i \leq k$) and ($\rho > 0$) then $b_p((w, B, \lfloor \tau_i \rfloor_{\mathcal{T}})) = -1$;
6. while ($i \leq k$) and ($\rho >= 0$) do
ρ += α_i;
if ($\rho < 0$) then $b_p((w, B, \lfloor \tau_i \rfloor_{\mathcal{T}}))$ += -1;
i += 1;
end while
7. while ($i \leq k$) do
ρ += α_i;
i += 1;
end while
8. set $\underline{\alpha}_p^w = \rho$.

This fixes the balances for all nodes except for the artificial node $(d, C, t_{n_{\mathcal{T}}})$. This balance is set so that the sum over all balances is zero,

$$b_p((d, C, t_{n_{\mathcal{T}}})) = - \sum_{v \in \mathcal{V}_p \setminus \{(d,C,t_{n_{\mathcal{T}}})\}} b_p(v).$$

3.3 Prescheduled Pallets

Here we deal with pallets that have been announced by the logistic system as prescheduled for transportation at a certain time. For these pallets we have the following data: the article type p and the amount that is loaded on the pallet, the time of retrieval from the automatic storage system, the source and the destination warehouses. There is, however, no information giving the reason for this transport. Therefore, it was agreed to interpret such pallets as realizations of a previously suggested schedule that were considered sufficiently urgent by the truck dispatcher; this should be correct in most cases. Since such pallets have higher priority in transportation than all other pallets, they cannot be included within the anonymous pallet setting of the article graphs of §3.2 directly. So separate graphs are set up for transporting them. None the less we have to account for the amount of article p that they transport. For this purpose we add entries (α_i, τ_i) to the lists of demands/supplies of the source and destination warehouses with α_i the amount carried on the pallet, the starting time as given by the pallet information and the destination time being the sum of the starting time and a user specified constant for the expected transportation time. Having done this, prescheduled pallets with the same source and destination and the same sets $\mathcal{R}(p)$ of compatible truck classes do not have to be discerned any longer. To simplify notation we assume $\mathcal{R}(p) = \mathcal{R}$ for all $p \in \mathcal{P}$. For each transport direction $\vec{w} = (w_1, w_2) \in \vec{\mathcal{W}} = \{(w_1, w_2) : w_1, w_2 \in \mathcal{W}, w_1 \neq w_2\}$ we set up a graph $D_{\vec{w}} = (\mathcal{V}_{\vec{w}}, \mathcal{A}_{\vec{w}})$ as follows.

Keeping close to the notation of the article graphs, the set of nodes consists of a buffer node for each time step and the two warehouses w_1 and w_2 and one additional artificial node named $(d, C, t_{n_\mathcal{T}})$,

$$\mathcal{V}_{\vec{w}} = \{w_1, w_2\} \times \{B\} \times \mathcal{T} \cup \{(d, C, t_{n_\mathcal{T}})\}.$$

In analogy to §3.2 we define the arc sets leading on to next time step, but for later reference we split them into the source and the destination parts,

$$\mathcal{A}^S_{\vec{w}} = \{(u, v) : u, v \in \mathcal{V}_{\vec{w}}, u_W = v_W = w_1, u_N = v_N = B, v_T = \lceil u_T + 1 \rceil_\mathcal{T}\},$$

$$\mathcal{A}^D_{\vec{w}} = \{(u, v) : u, v \in \mathcal{V}_{\vec{w}}, u_W = v_W = w_2, u_N = v_N = B, v_T = \lceil u_T + 1 \rceil_\mathcal{T}\}.$$

The arc set for collecting the remaining pallets reads

$$\mathcal{A}^C_{\vec{w}} = \{((w, B, t_{n_\mathcal{T}}), (d, C, t_{n_\mathcal{T}})) : w \in \mathcal{W}\}.$$

The arcs corresponding to transportation arcs of the truck graphs differ slightly, because after the initial preparation time the pallets are now waiting at the loading platform, ready for immediate transportation,

$$\begin{aligned}\mathcal{A}^P_{\vec{w}} = \{(u, v)_r : \quad & u, v \in \mathcal{V}_r, r \in \mathcal{R}, (u^r, v^r) \in \mathcal{A}^P_r, u_W = u^r_W, v_W = v^r_W, u_N = v_N = B, \\ & u_T = u^r_T, v_T = \left\lceil u^r_T + \overline{T}_r(u_W, v_W, u^r_T) + T^U(v_W) \right\rceil_\mathcal{T}\}.\end{aligned} \tag{4}$$

The final arc set is

$$\mathcal{A}_{\vec{w}} = \mathcal{A}^S_{\vec{w}} \cup \mathcal{A}^D_{\vec{w}} \cup \mathcal{A}^C_{\vec{w}} \cup \mathcal{A}^P_{\vec{w}}.$$

Lower and upper bounds are

$$l_{\vec{w}}(a) = 0, u_{\vec{w}}(a) = \infty \quad \text{for all } a \in \mathcal{A}_{\vec{w}}.$$

The balances are zero for the nodes in the destination warehouse w_2,

$$b_{\vec{w}}(v) = 0 \qquad \text{for all } v \in \{v' \in \mathcal{V}_{\vec{w}} : v_W = w_2\}.$$

For a node $v = (w_1, B, t)$ with $t \in \mathcal{T}$ the balance $b_{\vec{w}}(v)$ counts the number of prescheduled pallets in this direction with retrieval time τ such that $t = \lceil \tau + T^L(w_1) \rceil_\mathcal{T} \leq t_{n_\mathcal{T}}$ (for the definition of T^L see page 9). Finally, the artificial node ensures, that all balances sum up to zero,

$$b_{\vec{w}}((d, C, t_{n_\mathcal{T}})) = - \sum_{v \in \mathcal{V}_{\vec{w}} \setminus \{(d, C, t_{n_\mathcal{T}})\}} b_{\vec{w}}(v).$$

3.4 Coupling Constraints

Next we define the set of variables — these correspond mainly to the flow along the arcs of the networks — and the additional coupling constraints. The usual capacity constraints and flow conservation constraints for the networks will not be listed explicitly, but they are of course a central ingredient in the model.

In setting up the cost function, we will need for each article $p \in \mathcal{P}$, for each warehouse $w \in \mathcal{W}$, and for a specified subset $\widehat{\mathcal{T}} \subset \mathcal{T}$ of points in time $t \in \widehat{\mathcal{T}}$ some variables that measure the number of surplus pallets of article p that are available at w at time t after subtracting those that are reserved due to future balances. We refer to these variables by the index set

$$\mathcal{A}_f = \{(p, w, t) : p \in \mathcal{P}, w \in \mathcal{W}, t \in \widehat{\mathcal{T}}\}. \tag{5}$$

For convenience, all variable indices are collected in a super set

$$\mathcal{A} = \mathcal{A}_f \cup \bigcup_{j \in \mathcal{R} \cup \mathcal{P} \cup \vec{\mathcal{W}}} \mathcal{A}_j.$$

The vector of primal variables is

$$x \in \mathbb{Z}^{\mathcal{A}}.$$

The coupling constraints on top of the capacity and flow constraints fall into four categories: the constraints on the capacity of the loading and unloading platforms, the constraints linking transportation arcs of pallets and trucks, the capacity constraints for the transportation buffer, and the constraints determining the values of the variables of $\mathcal{A}_f$.

The capacity constraints on the loading platforms make use of the functions λ_U and λ_L defined in (2), i.e., for all $t \in \mathcal{T}$ and for all $w \in \mathcal{W}$:

$$\begin{aligned} \sum_{a \in \{(u,v) \in \mathcal{A}_r^U : r \in \mathcal{R}, u_W = w, u_T = t\}} x_a &\leq \lambda_U(w, t), \\ \sum_{a \in \{(u,v) \in \mathcal{A}_r^L : r \in \mathcal{R}, u_W = w, u_T = t\}} x_a &\leq \lambda_L(w, t). \end{aligned}$$

For brevity, we collect these loading constraints in a matrix A_L and a right hand side vector b_L,

$$A_L x \leq b_L. \tag{6}$$

For a truck class $r \in \mathcal{R}$ the arc set $\mathcal{A}_r^P$ contains the arcs corresponding to potential pallet transports. For each $a \in \mathcal{A}_r^P$ there exists for each article $p \in \mathcal{P}$ and for each direction $\vec{w} \in \vec{\mathcal{W}}$ at most one corresponding arc $a(p) \in \mathcal{A}_p^P$ (see (3)) and $a(\vec{w}) \in \mathcal{A}_{\vec{w}}^P$ (see (4)). To simplify notation, let $\mathcal{A}^P(a)$ denote this set of arcs. With $\kappa_r \in \mathbb{N}$ denoting the number of pallets that can be loaded on a truck of truck class r in average, the coupling constraints are

$$\text{for all } r \in \mathcal{R}, \text{ and for all } a_r \in \mathcal{A}_r^P : \sum_{a \in \mathcal{A}^P(a_r)} x_a \leq \kappa_r x_{a_r}.$$

In words, flow over an arc $a_r \in \mathcal{A}_r^P$ opens up capacity on the corresponding arcs $a \in \mathcal{A}^P(a_r)$ in the article and pallet graphs. We collect these constraints in a system with matrix A_K and a right hand side b_K,

$$A_K x \leq b_K. \tag{7}$$

The constraints for restricting the number of pallets in the transportation buffer of each warehouse $w \in \mathcal{W}$ and each period between two consecutive time steps $t_i, t_{i+1} \in \mathcal{T}$

to some given constants $n_w \in \mathbb{N}$ read

$$\text{for } i = 1, \ldots, n_{\mathcal{T}} - 1 \text{ and } w \in \mathcal{W}: \quad \sum_{a \in \{(u,v) \in \mathcal{A}_j^B : j \in \mathcal{P} \cup \vec{\mathcal{W}}, u_W = w, u_T = t\}} x_a \leq n_w.$$

The constraints for the buffer capacity will be represented by a matrix A_B and a right hand side b_B,

$$A_B x \leq b_B. \tag{8}$$

Finally, the last set of constraints computes the (possibly negative) number of pallets $x_{(p,w,t)}$ of article $p \in \mathcal{P}$ that would be stored at warehouse $w \in \mathcal{W}$ at the end of the planning horizon if transportation is stopped after time step $t \in \widehat{\mathcal{T}}$. For this, the sum of future balances

$$b_{(p,w,t)} = \sum_{v \in \{u \in \mathcal{V}_p : u_W = w, u_T > t\}} b_p(v)$$

has to be added to the flow of p at w following time step t,

$$\begin{aligned} &\text{for } (p,w,t) \in \mathcal{A}_f, t < t_{n_{\mathcal{T}}}: \\ &x_{(p,w,t)} = \sum_{a \in \{(u,v) \in \mathcal{A}_p^A \cup \mathcal{A}_p^B : u_W = w, u_T = t\}} x_a - \sum_{a \in \{(u,v) \in \mathcal{A}_p^L : v_W = w, v_T = t\}} x_a + b_{(p,w,t)}, \\ &\text{for } (p,w,t_{n_{\mathcal{T}}}) \in \mathcal{A}_f: \\ &x_{(p,w,t_{n_{\mathcal{T}}})} = \sum_{a \in \{(u,v) \in \mathcal{A}_p^C : u_W = w, u_T = t_{n_{\mathcal{T}}}\}} x_a - \sum_{a \in \{(u,v) \in \mathcal{A}_p^I : v_W = w, v_T = t_{n_{\mathcal{T}}}\}} x_a. \end{aligned} \tag{9}$$

We represent these constraints for extracting the remaining flow by a matrix A_F and a right hand side b_F,

$$A_F x = b_F. \tag{10}$$

4 Optimization Model, Part II: The Cost Function

Recall, that no actual costs are known that could be assigned to delays in the delivery of prescheduled pallets or pallets transported to satisfy current demand. So the priority rules must serve as a guideline for the design of the cost function. There is a large number of possibilities to do so and the final decision is always a bit arbitrary. Still, we believe that our approach satisfies a number of reasonable criteria that could be put to such a quality measure.

4.1 Priority Level 1: Prescheduled Pallets

We first discuss the top priority level, namely the prescheduled pallets. Working on the assumption, that these pallets are prescheduled according to some concept by the truck dispatcher (which is not always true), we expect that finding a feasible schedule is not a major problem. We simply impose a significant linear cost for each time step that the

prescheduled pallets have to wait at the loading platform of their source warehouse. Indeed, for our instances from practice this seemed sufficient to produce acceptable solutions. In particular, we fixed a constant $\gamma \in \mathbb{R}_+$ large enough and used, for each direction $\vec{w} \in \vec{\mathcal{W}}$ this constant times the waiting time as cost coefficient for the arcs leading on to the next time step at the source warehouse,

$$c_a = \begin{cases} \gamma \cdot \frac{v_T - u_T}{t_{n_\mathcal{T}} - t_1} & \text{for all } a = (u,v) \in \mathcal{A}^S_{\vec{w}}, \\ 0 & \text{for all } a \in \mathcal{A}_{\vec{w}} \setminus \mathcal{A}^S_{\vec{w}}. \end{cases}$$

Remark 4 *Penalizing the sum of the waiting times entails a certain danger of starvation for single pallets at remote warehouses without much traffic. If such effects are observed it might be worth to replace the sum of the waiting times by a penalty function that increases significantly with waiting time. In fact, for a given flow on a waiting arc $a \in \mathcal{A}^S_{\vec{w}}$ we know exactly how long each pallet has waited already, therefore one could set up an appropriate convex piecewise linear cost function. So far this appears not to be necessary.*

4.2 Priority Level 2: Pallets Satisfying Demand

On the next priority level — the pallets that have to be transported to satisfy current demand — the same approach is taken for the arcs $\mathcal{A}^L_p$ modeling lateness and the arcs $\mathcal{A}^I_p$ modeling the failure to deliver a needed pallet of article p within the planning horizon. The cost should now be balanced with respect to the cost of the first priority level. Having no reliable measure for the importance of a priority 1 pallet in comparison to violation of a priority 2 pallet, we set

$$c_a = \begin{cases} \frac{1}{4}\gamma \cdot \frac{v_T - u_T}{t_{n_\mathcal{T}} - t_1} & \text{for all } a = (u,v) \in \mathcal{A}^L_p, \\ \frac{1}{4}\gamma & \text{for all } a \in \mathcal{A}^I_p. \end{cases}$$

The same remark on the danger of starvation and its prevention applies as for the case of prescheduled pallets. The buffer and transport arcs are assigned some marginal costs with the goal to keep pallets from using these arcs without reason. The costs are designed so that it still should be cheaper to use transportation earlier if needed at all. For concreteness, let $\varepsilon > 0$ be a small constant compared to γ, then

$$c_a = \begin{cases} \varepsilon \cdot \frac{1}{2} \cdot \frac{v_T - u_T}{t_{n_\mathcal{T}} - t_1} & \text{for all } a = (u,v) \in \mathcal{A}^B_p, \\ \varepsilon \cdot \left(1 + \frac{v_T - t_1}{t_{n_\mathcal{T}} - t_1}\right) & \text{for all } a = (u,v) \in \mathcal{A}^P_p, \\ 0 & \text{for all } a \in \mathcal{A}_p \setminus (\mathcal{A}^L_p \cup \mathcal{A}^I_p \cup \mathcal{A}^B_p \cup \mathcal{A}^P_p). \end{cases}$$

For the two first priority levels there is not too much choice, because all pallets are needed. In most practical instances, it is not difficult to find a feasible solution transporting all pallets with little delay. If, however, no such solution can be found, then at least one delivery will fail and only a human dispatcher could decide which one hurts the least. In such cases minimizing the number of missing pallets is typically a reasonable criterion for automatically generated solutions, because it also reduces the number of pallets the human dispatcher has to care for.

4.3 Priority Level 3: Transports for Stock-keeping

The situation is distinctly different for the third priority level. There is no immediate pressure to transport particular pallets and ample room for decisions. Yet, these decisions will have a strong influence on the difficulty of future instances. Therefore it is in our view the most demanding task to find a reasonable criterion for this third level. Again it should form a compromise between transporting those pallets that are needed with the highest probability and transporting as many pallets as possible to reduce the overall load.

Our general goal is to reduce the expected number of pallets that will need transportation within the next days (in practice we settled for three days), but because a new schedule is to be determined every two to three hours with new information, we prefer schedules that transport those pallets early, that have high probability to be needed.

As a probability model we assume that for each article $p \in \mathcal{P}$ and each warehouse $w \in \mathcal{W}$ a probability distribution $F_p^w : \mathbb{R} \to [0,1]$ is given that assigns to an arbitrary amount α of article p the probability that demand will not exceed α for a specified period of time. Stated differently, α suffices to cover demand with probability $F_p^w(\alpha)$. In particular, if p is certainly not needed at w, then in our application the distribution function should satisfy $F_p^w(x) = 1$ for $x \geq 0$ and $F_p^w(x) = 0$ for $x < 0$. Thus, contrary to the usual definition of distribution functions, we will assume here that probability distributions are continuous from the right. In addition, we require the distributions F_p^w to be zero on $\mathbb{R}_- \setminus \{0\}$ and that there exists $\alpha \in \mathbb{R}_+$ with $F_p^w(\alpha) = 1$. The assumptions are certainly valid for the distributions we generate; a detailed description of these F_p^w is given in §5.1.

Let us fix an article $p \in \mathcal{P}$, a warehouse $w \in \mathcal{W}$, and a time step $t \in \widehat{\mathcal{T}}$. Then the number of pallets of p remaining at w at the end of the planning horizon under the assumption that no further transports take place after t is given by variable $x_{(p,w,t)}$ (see (9)). Assuming $\theta_p > 0$ (the case $\theta_p = 0$ will be treated later) and making use of the fractional amount $\underline{\alpha}_p^w$ (see Algorithm 2), $\underline{\alpha}_p^w + \theta_p x_{(p,w,t)}$ is a good guess[2] on the actual amount of p that would be available at the end of the planning horizon under these circumstances. With this, $\pi = F_p^w(\underline{\alpha}_p^w + \theta_p x_{(p,w,t)})$ yields the probability, that the pallets transported so far suffice for the next days. The next pallet is therefore needed with probability $1 - \pi$, the one after that with probability $1 - F_p^w(\underline{\alpha}_p^w + \theta_p x_{(p,w,t)} + \theta_p)$ and so on. Hence, there is no difficulty in computing the expected number of pallets of article p needed at warehouse w when stopping with the current solution after time step t.

Observation 5 *Let a probability distribution $F_p^w : \mathbb{R} \to [0,1]$ with $F_p^w(x) = 0$ for $x < 0$ and $F_p^w(\bar{\alpha}) = 1$ for some $\bar{\alpha} > 0$ specify the additional demand for p at w. Let the fractional pallet $\underline{\alpha}_p^w$ of Algorithm 2 and $x_{(p,w,t)} \in \mathbb{Z}$ pallets of size $0 < \theta_p \in \mathbb{R}$ be available at w after time $t_{n_\mathcal{T}}$ if transports are stopped after t, then*

$$f_p^w(x_{(p,w,t)}) = \sum_{x_{(p,w,t)} \leq i \leq \lfloor (\bar{\alpha} - \underline{\alpha}_p^w)/\theta_p \rfloor} [1 - F_p^w(\underline{\alpha}_p^w + i\theta_p)]$$

gives the expected number of pallets of size θ_p needed at w for sufficient supply. Moreover,

[2]Recall, that the pallets transported may deviate from θ_p and that the computation of $\underline{\alpha}_p^w$ involves further assumptions on the use of fractional pallets.

the extension $f_p^w : \mathbb{R} \to \mathbb{R}$ defined by setting $f_p^w(x) = f_p^w(\lfloor x \rfloor) + (x - \lfloor x \rfloor)[f_p^w(\lceil x \rceil) - f_p^w(\lfloor x \rfloor)]$ for $x \in \mathbb{R}$ is a piecewise linear convex function with Lipschitz constant 1.

Proof. Let $\hat{\alpha} \in \mathbb{R}$ be the available amount and let, for $\alpha \in \mathbb{R}$, $X_{\hat{\alpha}}(\alpha) = \max\{0, \lceil (\alpha - \hat{\alpha})/\theta_p \rceil\}$ denote the random variable counting the number of pallets needed to cover the unknown additional demand. Then the expected value of $X_{\hat{\alpha}}$ is

$$E(X_{\hat{\alpha}}) = \sum_{i=1}^{\infty} i[F_p^w(\hat{\alpha} + i\theta_p) - F_p^w(\hat{\alpha} + (i-1)\theta_p)] = \sum_{0 \le i \le \lfloor (\bar{\alpha} - \hat{\alpha})/\theta_p \rfloor} [1 - F_p^w(\hat{\alpha} + i\theta_p)]$$

Set $\hat{\alpha} = \underline{\alpha}_p^w + x_{(p,w,t)}\theta_p$ to obtain the formula above. Since the differences of consecutive values are nondecreasing, $f_p^w(j) - f_p^w(j-1) = F_p^w(\underline{\alpha}_p^w + (j-1)\theta_p) - 1 \le F_p^w(\underline{\alpha}_p^w + j\theta_p) - 1 = f_p^w(j+1) - f_p^w(j)$ for $j \in \mathbb{Z}$, the function is convex. The Lipschitz property follows from $|f_p^w(j) - f_p^w(j-1)| \le 1$. ■

Remark 6 *In the sequel we will often make use of the following helpful interpretation of function f_p^w. It may be viewed as assigning a priority value*

$$\pi_p^w(j) = 1 - F_p^w(\underline{\alpha}_p^w + j\theta_p) \in [0, 1], \quad j \in \mathbb{Z} \tag{11}$$

to the j-th pallet of article p remaining at w at the end of the planning horizon (negative j correspond to removals or missing pallets). As noted above, pallet j is assigned the probability, that all up to the j-th pallet are needed to cover additional future demand. Correspondingly, pallets with higher priority value should be transported first. For pallets that are needed with certainty (according to F_p^w) the priority will be 1 and using F_p^w alone we cannot discern their importance. For all other pallets of interest (with $\pi_p^w(j) > 0$ and arbitrary p and w) the priority order will be unique with high probability because of differing distributions and differing fractional supply $\underline{\alpha}_p^w$.

Setting $\widehat{\mathcal{T}} = \{t_{n_{\mathcal{T}}}\}$ in $\mathcal{A}_f$ of (5) , a possible candidate for a cost function would thus be the convex and piecewise linear function

$$\sum_{(p,w,t_{n_{\mathcal{T}}}) \in \mathcal{A}_f} f_p^w(x_{(p,w,t_{n_{\mathcal{T}}})}).$$

Under the assumption that abundant supply is available, it would measure the expected number of pallets, that still need transportation at the end of the planning horizon. For this cost function, however, it is not important whether among the selected pallets those are transported first that are needed with high probability. Furthermore, consider a pallet that is needed almost surely but entails a poorly filled truck ride. Such a pallet may be ignored in favor of a truck ride transporting a large number of pallets with small probabilities. Both of these shortcomings are not acceptable. In practice, it is important to improve the worst probabilities first, because these typically lead to the most urgent demand situations, while there is usually more time for the dispatcher to bring in supply for stock that falls short with low probability (e.g. by chartering an additional truck).

Furthermore, only the first few rides of the solution will be realized in practice and then a new solution will be computed, which increases the danger of repeatedly postponing the transportation of important pallets.

A first step to improve the situation is to apply the cost function not only at the end but at several points in time by specifying a larger set $\widehat{\mathcal{T}} \subset \mathcal{T}$. Since choosing $\widehat{\mathcal{T}} = T$ would be computationally too expensive and might also favor greedy solutions too much, we decided for

$$\widehat{\mathcal{T}} = \left\{ t_{\lfloor \frac{i}{3} n_{\mathcal{T}} \rfloor} : i \in \{1, 2, 3\} \right\}.$$

With the definition of $\mathcal{A}_f$ as in (5) this would lead to the cost function

$$\sum_{(p,w,t) \in \mathcal{A}_f} f_p^w(x_{(p,w,t)}).$$

In this setting, solutions are preferred that minimize the expected number of required pallets already at early stages, at the price that the final constellation at time $t_{n_{\mathcal{T}}}$ might get a bit worse. Unfortunately, this does not yet resolve the problem of ignoring a few pallets needed almost surely in favor of many pallets needed with rather low probabilities.

To address this issue, observe that the gain of a truck ride may be quantified as the sum of the priorities of the pallets arriving at the destination warehouse minus the hopefully small priorities subtracted at the source warehouse. Therefore transportation of pallets with high probability values can be made more rewarding while keeping the priority order suggested by the probabilities by applying consistently the same strictly increasing map to all probabilities.

Observation 7 *Let $g : [0,1] \to [0, \bar{\gamma}]$ with $\bar{\gamma} > 0$ be a fixed non decreasing function. For p, w, F_p^w, $\bar{\alpha}$ as in Observation 5, the function $\tilde{f}_p^w : \mathbb{R} \to \mathbb{R}_+$ defined by*

$$\tilde{f}_p^w(x) = \sum_{x \le i \le \lfloor (\bar{\alpha} - \underline{\alpha}_p^w)/\theta_p \rfloor} g(1 - F_p^w(\underline{\alpha}_p^w + i\theta_p)) \quad \text{for all } x \in \mathbb{Z}$$

and

$$\tilde{f}_p^w(x) = \tilde{f}_p^w(\lfloor x \rfloor) + (x - \lfloor x \rfloor)[\tilde{f}_p^w(\lceil x \rceil) - \tilde{f}_p^w(\lfloor x \rfloor)] \quad \text{for all } x \in \mathbb{R}$$

is again convex and piecewise linear with Lipschitz constant $\bar{\gamma}$.

Proof. Because of the monotonicity of g and $1 - F_p^w$, the linear pieces satisfy $\tilde{f}_p^w(x) - \tilde{f}_p^w(x-1) \le \tilde{f}_p^w(x+1) - \tilde{f}_p^w(x)$ for $x \in \mathbb{Z}$, thus $\tilde{f}_p^w$ is convex. Furthermore $|\tilde{f}_p^w(x) - \tilde{f}_p^w(x-1)| \le \bar{\gamma}$ for $x \in \mathbb{Z}$, so the Lipschitz constant is $\bar{\gamma}$. ■

By choosing an appropriate g we could, in principle, enforce strict priorities between pallets on different probability levels. For example, if a truck ride with at least one pallet having $\pi_p^w(j) = 1$ should be preferred to truck rides without such pallets, let $\hat{\pi} = \max\{\pi_p^w(j) < 1 : p \in \mathcal{P}, w \in \mathcal{W}, j \in \mathbb{Z}\}$ be the highest probability less than 1 assigned to the pallets. By our assumptions on the distributions F_p^w we have $\hat{\pi} < 1$. Denote by κ the largest capacity of all trucks. Then the function $g : [0,1] \to [0,1]$ with $g(1) = 1$ and $g(x) = x/(\hat{\pi}\kappa)$ for $x \in [0, 1)$ would have the desired effect.

In practice we take a less restrictive approach. In order to motivate our choice we first introduce a merit function to measure the quality of a feasible constellation. Suppose that at time t the set $S = \{(p, w, j) : p \in \mathcal{P}, s \in \mathcal{W}, j \in \mathbb{Z}, \pi_p^w(j) < 1\}$ describes the pallets, that are *not* available at the respective warehouses at the end of the planning period if no further transports occur after t (for the moment we ignore j's with $\pi_p^w(j) = 1$). Consider the function

$$1 - \prod_{(p,w,j)\in S} (1 - \pi_p^w(j)) = 1 - \prod_{(p,w,j)\in S} F_p^w(\underline{\alpha}_p^w + j\theta_p).$$

Its value will be close to 1 if many of the pallets are needed with high priority $\pi_p^w(j)$, and its value will decrease whenever an element from S is deleted or replaced by an element having lower priority value. For illustration purposes let us make the absolutely invalid assumption, that the $\pi_p^w(j)$ specify the probabilities of independent events that pallet (p, w, j) will be needed at w in the next time period. Then this number would give the probability that at least one of the pallets not available will have to be transported in the next period. So we would like to find a constellation that minimizes this number or, equivalently, maximizes $\prod_{(p,w,j)\in S} F_p^w(\underline{\alpha}_p^w + j\theta_p)$. Using the logarithm for linearization, a reasonable objective could read

$$\min_S \quad - \sum_{(p,w,j)\in S} \log(F_p^w(\underline{\alpha}_p^w + j\theta_p)).$$

With respect to Observation 7, this suggests the choice $g(\cdot) = \min\{\bar{\gamma}, -\log(\cdot - 1)\}$ for some $\bar{\gamma} > 0$; the priority order between the pallets is preserved. Because $\log(1 + x) \le x$ for all $x \in \mathbb{R}$, we have

$$-\log(F_p^w(\underline{\alpha}_p^w + j\theta_p)) \ge 1 - F_p^w(\underline{\alpha}_p^w + j\theta_p)$$

and transporting a pallet with high priority level has become more attractive than before.

We continue with a detailed specification on how to construct the cost function from this and how to deal with the cases $\pi_p^w(j) = 0$ and $\theta_p = 0$; the reader not interested in impelementational details may safely skip this part and continue with the next section.

In order to cope with pallets (p, w, j) having $\pi_p^w(j) = 0$ or that correspond to negative amounts $\underline{\alpha}_p^w + \theta_p j < 0$, we introduce a lower probability level $\underline{\pi}$ satisfying $0 < \underline{\pi} < \overline{\pi} < 1$ (for the definition of $\overline{\pi}$ see page 5; beyond this level no pallets should be transported). Define, for a given weight $\bar{\gamma} > 0$, $p \in \mathcal{P}$, $w \in \mathcal{W}$ the constants

$$\text{for } j \in \mathbb{Z}: \quad g_p^w(j) = \begin{cases} 2\bar{\gamma}(-\log \underline{\pi}) & j < 0, \\ \frac{3}{2}\bar{\gamma}(-\log \underline{\pi}) & j \ge 0 \text{ and } \underline{\alpha}_p^w + j\theta_p < 0, \\ \bar{\gamma}(-\log \underline{\pi}) & 0 \le 1 - \pi_p^w(j) < \underline{\pi} \text{ and } \underline{\alpha}_p^w + j\theta_p \ge 0, \\ \bar{\gamma}(-\log[1 - \pi_p^w(j)]) & \underline{\pi} \le 1 - \pi_p^w(j) < \overline{\pi}, \\ 0 & \overline{\pi} \le 1 - \pi_p^w(j). \end{cases}$$

In practice we set $\bar{\gamma} = \frac{1}{4}\gamma/(-4|\widehat{\mathcal{T}}| \log \underline{\pi})$ and $\underline{\pi} = 10^{-2}$. This choice corresponds to our

actual implementation up to a minor technical detail[3], that we ignore here. Let

$$\begin{array}{lcll} \hat{\jmath}_p^w & = & \max\{j \in \mathbb{Z} : \pi_p^w(j) < 1 - \overline{\pi}\} & p \in \mathcal{P}, w \in \mathcal{W}, \\ \underline{\jmath}_p & = & \sum_{w\in\mathcal{W},t\in\mathcal{T}} b_p((w,B,t)) & p \in \mathcal{P}, \\ \overline{\jmath}_p & = & \sum_{w\in\mathcal{W},t\in\mathcal{T}} b_p((w,A,t)) & p \in \mathcal{P} \end{array}$$

denote the last pallet of article p at warehouse w with $g_p^w(j) > 0$, the sum of all negative demands, and the sum of all positive supplies of article p, respectively. Then for p with $\theta_p > 0$ and $w \in \mathcal{W}$ we define one component of the cost function by

$$\breve{f}_p^w(x) = \begin{cases} \infty & x \in (-\infty, \underline{\jmath}_p) \cup (\overline{\jmath}_p, \infty), \\ \sum_{x\le j\le \hat{\jmath}_p^w} g_p^w(j) & x \in \mathbb{Z} \cap [\underline{\jmath}_p, \hat{\jmath}_p^w], \\ \varepsilon[x - (\hat{\jmath}_p^w + 1)] & x \in [\hat{\jmath}_p^w + 1, \overline{\jmath}_p], \\ \breve{f}_p^w(\lfloor x\rfloor) + (x - \lfloor x\rfloor)[\breve{f}_p^w(\lceil x\rceil) - \breve{f}_p^w(\lfloor x\rfloor)] & x \in [\underline{\jmath}_p, \hat{\jmath}_p^w + 1] \setminus \mathbb{Z}. \end{cases}$$

So the function is convex with domain $[\underline{\jmath}_p, \overline{\jmath}_p]$, it is nonnegative and piecewise linear on its domain and zero at $\hat{\jmath}_p^w + 1$. In theory there is no need for restricting the domain nor for having the cost function slightly increase for $x > \hat{\jmath}_p^w + 1$, but it is advantageous in connection with solving the Lagrangian relaxation by bundle methods. Note also, that among the pallets the same priority order is maintained as induced by the probabilities, but the weighting differs to the effect, that truck loads containing just a few high priority pallets will now be preferred to truck loads containing many medium priority pallets.

Similar considerations motivate our choice for the cost function for $p \in \mathcal{P}_0 = \{p \in \mathcal{P} : \theta_p = 0\}$. Recall, that for such articles, by convention, any pallet transported carries an infinite amount and at most one should be brought to any warehouse upon demand. Likewise each warehouse can supply at most one such pallet for transportation. Thus, for $p \in \mathcal{P}_0$, (the sum in the if condition indicates the existence of demand)

$$\begin{array}{lcll} \pi_p^w(j) & = & 1 & \text{for } 0 > j \in \mathbb{Z}, \\ \pi_p^w(0) & = & 1 - F_p^w(\underline{\alpha}_p^w) & \text{if } \sum_{t\in\mathcal{T}} b_p((w,B,t)) < 0, \\ \pi_p^w(0) & = & 1 - F_p^w(0) & \text{if } \sum_{t\in\mathcal{T}} b_p((w,B,t)) = 0, \\ \pi_p^w(1) & = & 1 - F_p^w(\underline{\alpha}_p^w) & \text{if } \sum_{t\in\mathcal{T}} b_p((w,B,t)) = 0, \\ \pi_p^w(j) & = & 0 & \text{otherwise.} \end{array}$$

The constants $g_p^w(j)$ and the cost function $\breve{f}_p^w(\cdot)$ are now constructed according to the same rules as above. The complete cost function for priority level 3 reads

$$\sum_{(p,w,t)\in\mathcal{A}_f} \breve{f}_p^w(x_{(p,w,t)}).$$

4.4 Costs on the Truck Graphs

The costs defined on the arcs of the truck graphs do not have a major influence in the current application. We impose some costs on the transport and transfer arcs so that

[3]If there is no demand for p at w while there is strictly positive supply, then we set $g_p^w(0)$ depending only on $\pi = F_p^w(0)$ independent of the sign of $\underline{\alpha}_p^w$ in order to not inhibit emptying the warehouse.

trucks do not ride without need. Because the current application does not require the minimization of the number of trucks in use, the depot is only useful as artificial starting and stopping location. Since trucks should not keep waiting at the depot, we set the costs

$$c_a = \begin{cases} 1 & a \in \mathcal{A}_r^P, r \in \mathcal{R}, \\ \frac{1}{10} & a \in \mathcal{A}_r^E, r \in \mathcal{R}, \\ 10 & a = (u, v) \in \mathcal{A}_r^B \text{ with } u_W = d, r \in \mathcal{R}, \\ 0 & \text{otherwise.} \end{cases}$$

5 Implementation

5.1 Generation of the Probability Distributions

We assume, that for most articles there is no reasonable trend of the demand during the short period of the next n working days for small $n \in \mathbb{N}$. In our application this assumption makes sense due to the typical structure of the life-cycle of the articles. The life-cycle consists of a starting phase with slowly increasing demand, a running phase with high sales and a stopping phase with slowly decreasing sales or an abrupt stopping because it is sold out and a follower article is introduced. This life-cycle normally runs over a year or longer, but we are interested in the demand a few days ahead. So it doesn't seem advantageous to use classical time series methods to forecast the future demand. For the situation of some strong seasonal articles see below.

Also, for most articles there is a strong volatility of the daily sales, so it might be possible to misinterpret a random fluctuation as a substantial trend. Because of this volatility it is very difficult to observe dependencies of the sales of different articles. The experiences of our industrial partner and our statistical examinations indicate that short term trends and dependencies between articles are not of major relevance for our optimization problem. So we assume, that the daily demands of an article for the next n days are independent and identically distributed and that they are also independent from the demand of all other articles.

We denote the random demand of article p at warehouse w for one day by D_p^w and its distribution function by G_p^w. Note that, for reasons explained in §4.3, we do not follow the usual convention, that distribution functions are continuous from the left, but require for this particular application that they are continuous from the right. Let $D_p^{w,m},\ m \leq n$, denote the accumulated demand of the following m working days and $G_p^{w,m}$ its distribution function.

As pointed out in §4.3, our main interest is in obtaining an estimation for the probability that a given amount suffices to satisfy demand, so we concentrate on the distribution function G_p^w. For estimating G_p^w we use the empirical distribution of the daily demand of p at w for a fixed number T of working days backwards. Let $d_{i,p}^w$ denote the demand for p at w at the i-th previous working day. Our approach consists of applying decreasing weights $z_t,\ t = 1, \ldots, T$ with $\sum_{t=1}^T z_t = 1$, to the past daily demands $d_{1,p}^w, \ldots, d_{T,p}^w$, i.e., we take

$$\hat{G}_p^w(x) = \sum_{t=1}^{T} z_t \mathbb{1}_{\{d_{t,p}^w \leq x\}}$$

as an estimation for $G_p^w(x)$. Note, that we consider only working days of our industrial partner. We take $T = 25$.

The empirical distribution has a lot of favorable properties, especially as an estimator for distribution functions in the i.i.d. case. For some convergence results see for instance [19] Chapter 5.1.1. Also, it plays an important role for many statistical methods, among others for the bootstrap method, see [9]. For applications in finance it is often used as the best choice, e.g. for calculations of the value at risk, cf. [4].

In particular, empirical distributions are considered a suitable choice for estimations if distributions exhibit heavy tails or if it is difficult to identify parametric families of distribution functions in the model. Quite frequently, we could observe heavy tails for the daily demand at our industrial partner as a consequence of the mixture of the demands of many small retailers and very few huge demands by big chains of stores (e.g. ALDI, Metro etc.). Regarding parametric families of distributions, such distributions would have to be assigned automatically and on short term, since customers change their article identifiers frequently (often due to minor changes in design) and no information is available on the nature of the product. We currently see no hope to set up such a system, even though some clustering analysis of articles that appear repeatedly jointly in several orders might help in this respect.

A further advantage of the weighted empirical distribution is that the concept is quite intuitive and allows a lot of flexibility, so that users should be able to adjust the weights reasonably according to their own preferences.

As an estimation of $G_p^{w,n}$ for a fixed $n \in \mathbb{N}$ we use the n-th convolution $(\hat{G}_p^w)^{*(n)}$, i.e., for $x \in \mathbb{R}$ and $n > 1$

$$\begin{aligned}
(\hat{G}_p^w)^{*(n)}(x) &= \int_{-\infty}^{\infty} (\hat{G}_p^w)^{*(n-1)}(x-z)\, d\hat{G}_p^w(z) \\
&= \sum_{j=1}^{l} (\hat{G}_p^w)^{*(n-1)}(x-y_j) \left(\hat{G}_p^w(y_j) - \hat{G}_p^w(y_j-)\right) \\
&= \sum_{\{(i,j):\, x_i+y_j \le x\}} \left((\hat{G}_p^w)^{*(n-1)}(x_i) - (\hat{G}_p^w)^{*(n-1)}(x_i-)\right) \cdot \left(\hat{G}_p^w(y_j) - \hat{G}_p^w(y_j-)\right),
\end{aligned}$$

where x_i, $i = 1, \ldots, k$, denote the jump points of $(\hat{G}_p^w)^{*(n-1)}$ in ascending order, y_j, $j = 1, \ldots, l$, the jump points of $\hat{G}_p^w$, and $\hat{G}_p^w(y-)$ the left hand limit of $\hat{G}_p^w(\cdot)$ at y. We have chosen $n = 3$. By assuming that the daily demand of the following working days is independent, we obtain an estimation of the distribution function of the theoretical demand of the n following working days.

In principle, this approach is valid only for the case that no information is available on future demand, but, of course, we know all orders before their picking time. In order to avoid investigations on conditional distribution functions considering the demand that is currently known, we think of $(\hat{G}_p^w)^{*(n)}$ as an estimation of all additional orders that will appear during the next n working days. Any other approach would require additional information on the ordering behavior of the customers and on the technical details about

the passing of the orders from the different customer management systems to the warehouse management system of our partner.

To obtain a continuous distribution function, we compute a piecewise linearization $\tilde{G}_p^{w,n}$ of $(\hat{G}_p^w)^{*(n)}$. Let z_i, $i = 1, \ldots, m$, denote the jump points of $(\hat{G}_p^w)^{*(n)}$ in ascending order. We define

$$\tilde{G}_p^{w,n}(x) := \begin{cases} 0 & \text{for } x < z_1 \\ (\hat{G}_p^w)^{*(n)}(z_i) + [(\hat{G}_p^w)^{*(n)}(z_{i+1}) - (\hat{G}_p^w)^{*(n)}(z_i)] \, \frac{x - z_i}{z_{i+1} - z_i} & \text{for } z_i \le x < z_{i+1} \\ 1 & \text{for } x \ge z_m. \end{cases}$$

Then $\tilde{G}_p^{w,n}$ is a strictly increasing piecewise linear function on $[z_1, \alpha]$, where

$$\alpha = z_m = n \cdot \max_{t=1,\ldots,T} d_{t,p}^w,$$

and has at most one jump at $x = z_1$. We use it as an estimation of the distribution of the additional orders of the next n working days at warehouse w.

Remark 8 *Note that, in contrast to $(\hat{G}_p^w)^{*(n)}$, the linearized $\tilde{G}_p^{w,n}$ is no longer an unbiased estimation of the distribution function of $D_p^{w,n}$. Although it slightly overestimates it, this problem doesn't play an important role because the choice of n is a vague estimation of our industrial partner. The main advantages of $\tilde{G}_p^{w,n}$ are, that it is continuous on (z_1, ∞) and that the inverse $(\tilde{G}_p^{w,n})^{-1}$ exists on $(\tilde{G}_p^{w,n}(z_1+), 1)$ also in a strong functional sense and is continuous on this interval.*

Remark 9 *The use of a smooth kernel K_n is a common alternative to obtain a continuous distribution function $\bar{G}_p^{w,n}$ by defining*

$$\bar{G}_p^{w,n}(x) := \int_{-\infty}^{\infty} K_n(x - y) \, d(\hat{G}_p^w)^{*(n)}(y).$$

It can be proven, that unter slight assumptions $\bar{G}_p^{w,n}(x)$ is an unbiased estimator, see e.g. [21]. A more difficult problem might be the existence of the inverse $(\bar{G}_p^{w,n})^{-1}$ in the strong sense for all samples $d_{1,p}^w, \ldots, d_{T,p}^w$.

There is no explicit seasonal approach in our model but we are able to observe long-term trends. This is influenced by the choice of the weights z_i. We use constant weights $1.25 \cdot T^{-1}$ up to the switching point $\lceil 0.6 \cdot T \rceil$, and after that point linear decreasing weights, i.e., we take

$$z_t = \frac{25}{8}(T - t + 1)T^{-2} \qquad \text{for } \lceil 0.6 \cdot T \rceil < t \le T.$$

A better adjustment of the weights might be possible based on a careful evaluation of the numerical results by our industrial partner. For example, one might think of using exponentially decreasing weights, which is a popular approach in time series analysis. Further improvement might be gained by adjusting T on dependence of the article p or the warehouse w.

To avoid a misinterpretation of a special situation at our industrial partner, we have chosen $(\tilde{F}_p^w(x))^{(n)} = 1$ for $x \geq 0$ if there is only one $i \in \{1, \ldots, T\}$ with $D_{i,p}^w > 0$. This situation occurs if there is a single huge demand of a big chain of stores preparing a special offer. In most cases it is not useful to provide additional stock of article p at warehouse w on top of the demand caused by the special offer.

For all articles p and warehouses w we use the distribution function $\tilde{G}_p^{w,3}$ for our numerical experiments and denote it by F_p^w (see also §4.3).

Some articles are tightly linked with a deadline after which they are unlikely to be sold again, e.g. greeting cards and wrapping paper for Christmas or Easter. For such articles it would be desirable that the user sets $F_p^w(x) = 1$ for $x \geq 0$, because we are not able to recognize this on past demands alone.

5.2 Lagrangian Relaxation and Bundle Method

By Lagrangian relaxation of the coupling constraints of §3.4 the problem decomposes into $|\mathcal{R}|+|\mathcal{P}|+|\vec{\mathcal{W}}|$ independent min cost flow problems and $|A_f|$ minimization problems of one dimensional convex piecewise linear functions (one function $\breve{f}_p^w$ for each $(p, w, t) \in \mathcal{A}_f$). These subproblems can be solved efficiently by specialized methods yielding objective value and subgradient (or supergradient) for the dual problem of determining optimal Lagrange multipliers. The latter are computed by a bundle method that also produces approximations to primal optimal solutions.

For concreteness, let A_i denote the node-arc-incidence matrix of the digraph $D_i = (\mathcal{V}_i, \mathcal{A}_i)$ and $b_i \in \mathbb{R}^{\mathcal{V}_i}$ the corresponding balances for $i \in \mathcal{R} \cup \mathcal{P} \cup \vec{\mathcal{W}}$, then the complete problem description reads

$$
\begin{array}{lllllllll}
\min & \displaystyle\sum_{r\in\mathcal{R}} c_{\mathcal{A}_r}^T x_{\mathcal{A}_r} + \sum_{\vec{w}\in\vec{\mathcal{W}}} c_{\mathcal{A}_{\vec{w}}}^T x_{\mathcal{A}_{\vec{w}}} + \sum_{p\in\mathcal{P}} c_{\mathcal{A}_p}^T x_{\mathcal{A}_p} + \sum_{(p,w,t)\in\mathcal{A}_f} \breve{f}_p^w(x_{(p,w,t)}) & & & & & & & & \\
\text{s.t.} & A_r x_{\mathcal{A}_r} & & & & & & = & b_r & r \in \mathcal{R} \\
& & A_{\vec{w}} x_{\mathcal{A}_{\vec{w}}} & & & & & = & b_{\vec{w}} & \vec{w} \in \vec{\mathcal{W}} \\
& & & A_p x_{\mathcal{A}_p} & & & & = & b_p & p \in \mathcal{P} \\
& A_L x & & & & & & \leq & b_L & \\
& & A_K x & & & & & \leq & b_K & \\
& & & A_B x & & & & \leq & b_B & \\
& & & & A_F x & & & = & b_F & \\
& l_{\mathcal{A}_i} \leq x_{\mathcal{A}_i} \leq u_{\mathcal{A}_i} & i \in \mathcal{R} \cup \vec{\mathcal{W}} \cup \mathcal{P}, & & x \in \mathbb{Z}^{\mathcal{A}}. & & & & &
\end{array}
$$

Note, that the loading constraints A_L of (6) affect the variables belonging to truck graphs only, the capacity constraints A_K of (7) involve almost all graphs but none of the variables A_f, the buffer constraints A_B of (8) deal with arcs of article graphs exclusively, the constraints A_F of (10) for computing the remaining flow involve only variables of article graphs and the set $\mathcal{A}_f$.

In order to describe the relaxation, let m_L, m_K, m_B, m_F denote the number of rows

of the matrices A_L, A_K, A_B, A_F, let

$$m = m_L + m_K + m_B + m_F, \quad \bar{A} = \begin{bmatrix} A_L \\ A_K \\ A_B \\ A_F \end{bmatrix}, \text{ and } \bar{b} = \begin{bmatrix} b_L \\ b_K \\ b_B \\ b_F \end{bmatrix}.$$

Feasible Lagrange multipliers are $y \in Y := \mathbb{R}_-^{m_L+m_K+m_B} \times \mathbb{R}^{m_F}$. For defining the dual function $\varphi(y)$, set for $i \in \mathcal{R} \cup \vec{\mathcal{W}} \cup \mathcal{P}$

$$\varphi_i(y) = \min\{(c_{\mathcal{A}_i} - [\bar{A}^T y]_{\mathcal{A}_i})^T x_{\mathcal{A}_i} : A_i x_{\mathcal{A}_i} = b_i, l_{\mathcal{A}_i} \le x_{\mathcal{A}_i} \le u_{\mathcal{A}_i}, x_{\mathcal{A}_i} \in \mathbb{Z}^{\mathcal{A}_i}\},$$

and for $a = (p, w, t) \in \mathcal{A}_f$

$$\varphi_a(y) = \min\{\breve{f}_p^w(x_a) - [\bar{A}^T y]_a x_a : x_a \in \mathbb{R}\}$$

then the dual problem reads

$$\max_{y \in Y} \ \varphi(y) = \bar{b}^T y + \sum_{i \in \mathcal{R} \cup \vec{\mathcal{W}} \cup \mathcal{P} \cup \mathcal{A}_f} \varphi_i(y).$$

Note, that for given $y \in Y$, $i \in \mathcal{R} \cup \vec{\mathcal{W}} \cup \mathcal{P}$ determining an optimizer for $\varphi_i(y)$,

$$x_{\mathcal{A}_i}(y) \in \text{Argmin}\{(c_{\mathcal{A}_i} - [\bar{A}^T y]_{\mathcal{A}_i})^T x_{\mathcal{A}_i} : A_i x_{\mathcal{A}_i} = b_i, l_{\mathcal{A}_i} \le x_{\mathcal{A}_i} \le u_{\mathcal{A}_i}, x_{\mathcal{A}_i} \in \mathbb{Z}^{\mathcal{A}_i}\},$$

amounts to computing an optimal solution to a min-cost flow problem for the graph $D_i = (\mathcal{V}_i, \mathcal{A}_i)$. For this we employ the code MCF of Andreas Löbel [16], which is a network simplex code that supports warm starts for changing cost coefficients. For $a = (p, w, t) \in \mathcal{A}_f$, finding an (integral) optimizer for $\varphi_a(y)$,

$$x_a(y) \in \text{Argmin}\{\breve{f}_p^w(x_a) - [\bar{A}^T y]_a x_a : x_a \in \mathbb{R}\}$$

is easy, since the function is piecewise linear and convex, so it can be done by binary search on the (integral) break points. Collecting all primal optimizers in $x(y) \in \mathbb{Z}^{\mathcal{A}}$, the primal violation $s(y) = \bar{b} - \bar{A}x(y)$ is a subgradient of $\varphi(\cdot)$ in y. Thus, the dual function value and a subgradient can be computed efficiently. This allows the use of bundle methods, see [11]. We use our own code ConicBundle, which is an outgrowth of [10]. Under mild assumptions, that are satisfied here, the method generates a sequence $y^k \in Y$ of dual feasible points converging to the optimum. At the same time, by taking convex combinations of the subgradients $s(y^k)$, it yields a sequence of aggregate subgradients $\bar{s}^k$ that converge to zero. It can be shown (see e.g. [5, 10]), that by propagating the same aggregation process to the $x(y^k)$ one obtains a sequence $\bar{x}^k$ of aggregate primals, whose cluster points lie in the set of optimal solutions of the primal linear programming relaxation. The code ConicBundle generates these $\bar{x}^k$. The final $\bar{x}^k$ is used in the heuristic for generating primal feasible solutions.

In principle, the code ConicBundle would allow the use of separate cutting plane models for each function φ_i, i.e., separate collections of subgradients and function values that

approximate φ_i from below. In practice, however, this would lead to very large quadratic subproblems and it turned out to be computationally more efficient to collect the functions in four groups with a separate cutting plane model for each group. In particular, we have separate models for the four functions

$$\varphi_{\mathcal{R}}(y) = \sum_{r\in\mathcal{R}} \varphi_r(y),\ \varphi_{\vec{\mathcal{W}}}(y) = \sum_{\vec{w}\in\vec{\mathcal{W}}} \varphi_{\vec{w}}(y),\ \varphi_{\mathcal{P}}(y) = \sum_{p\in\mathcal{P}} \varphi_p(y),\ \varphi_{\mathcal{A}_f}(y) = \sum_{a\in\mathcal{A}_f} \varphi_a(y).$$

Splitting in this way seemed superior to several other choices. Maybe this is due to the fact, that some constraint classes of the coupling constraints act exactly on one of these subgroups. For each group we used the minimal bundle size, i.e., one new subgradient and the old aggregate subgradient.

Remark 10 *In order to increase efficiency it might be worth to approximate $\varphi_{\mathcal{A}_f(y)}$ by a single second order cone constraint. Our first experiments in this direction entailed some numerical difficulties and we did not pursue this further.*

5.3 Rounding Heuristic

For generating feasible solutions we make use of the primal (approximate or exact) fractional solution vector $\bar{x}$. In practice, $\bar{x}$ is generated by aggregation in the bundle method. For comparative numerical experiments we will also use an exact optimal solution produced by a simplex method. Based on this vector we first fix pallet candidates for transportation, later we assign these candidates to truck rides. The emphasis of the heuristic is on transporting for each article the same amounts as the fractional solution along the same directions. The heuristic works reasonably well (see §6) but it is still somewhat simplistic and ad hoc, so we refrain from giving a detailed description and outline only its main steps.

Fixing pallet candidates for transportation is done separately for each article $p \in \mathcal{P}$ as follows. The variables $x_{\mathcal{A}_p}$ represent the flow in the article graph D_p of §3.2. We decompose the flow along transportation arcs into paths from one warehouse to the next, these are then aggregated to pallets. For this, the arcs are sorted in non decreasing order with respect to the time stamp of the tail node. The arcs are summed up in this sequence, each arc being added to the appropriate sum of the corresponding direction; likewise, at the warehouses the balances are summed up in the same order. If the sum of a direction exceeds the lower threshold 0.1 and a positive balance is available at the tail warehouse, a new pallet is created; at the same time, one unit is subtracted from the direction sum and one unit is subtracted from the available amount at the tail warehouse. The new pallets release date is eventually set to the maximum of the arrival times (at the tail warehouse) of the flow pieces that contribute to it. The pallet contributes one unit of flow to the head warehouse at the time when the first truck, that serves this direction after the pallets creation, arrives there. The due dates of the pallet is eventually set to the first event, that needs part of the pallets flow. Such an event is either a negative balance at the head warehouse or another pallet starting from the head warehouse.

The generation of pallet candidates for prescheduled pallets of §3.3 follows the same pattern but is much simpler, we skip it here.

Once the pallet candidates are fixed and equipped with release and due dates, we generate the truck rides. For each warehouse w, each direction $(w, \bar{w}) \in \vec{\mathcal{W}}$ leaving this warehouse, and each truck class $r \in \mathcal{R}$, we consider the first ten possible transport rides of a truck of type r in this direction. Each ride is filled in a greedy manner by the most important pallets that are yet to be transported in this direction, the priority being available prescheduled pallets, then regular pallets ordered by due dates, and among pallets with the same due date we choose by the improvement in distribution considering both warehouses. A newly generated ride is compared to the previously selected ride for this truck class and direction. When a ride of this type has been fixed, the selected ride is compared to the previously chosen candidate for this direction; the winner of the direction is compared to the current favorite of the truck rides leaving w; finally, this candidate is compared to the previously selected ride for the warehouses treated so far. Each comparison considers only two concrete truck rides with their assigned load and takes into account attributes such as the yet uncovered amount of flow in the relevant directions summed over all truck classes, the uncovered amount for the particular truck class, the number of prescheduled pallets loaded, the number of pallets with tight due date, and the departure and arrival times of the trucks. The chosen ride is then fixed, all pallets loaded are removed from the lists of waiting pallets and the process starts anew, till no further truck rides can be generated or no pallets are waiting for transport.

The heuristic is not very sophisticated and relies on the hope that the amounts and timing suggested by the primal solution $\bar{x}$ provide a good guideline for setting up a schedule. So far we have not implemented any improvement heuristics that enhance this initial solution via local search methods (this clearly could improve the solution a bit in several cases). Our current main concern, however, is not a perfect solution for the given problem, but a reasonably fast method suitable for use in an online environment. In particular, any sophisticated post optimizations will most likely have no effect in practice, because in general the realization of the schedule differs considerably from the planned solution.

6 Numerical Experiments

For our experiments we use more than half a year of real data, stemming from the application at our industrial partner. The online stream of data described in §2 is available in full. We generate our instances by running through it and stopping at 6:00, 9:00, 12:00, and 15:00 every day for recomputing the online schedule (we do *not* include later data in this computation). The planning horizon is one day; more precisely, each run includes at most $|\mathcal{T}| \leq 144$ time steps of 10 minutes each, depending on the availability of trucks for transportation. We ignore instances where no truck data is available (in some cases this data is obviously missing by mistake). After preprocessing, between 500 and 1200 articles of the 40000 products are in need of transportation.

The company currently operates two warehouses, call them A and B, within a distance of roughly 40 minutes driving time; including the loading process, the estimate is 50 minutes. Correspondingly, we set $\underline{T}_r(A, B, t) = 40$ and $\overline{T}_r(A, B, t) = 50$ in (1) for all truck classes $r \in \mathcal{R}$ and time steps $t \in \mathcal{T}$ in our experiments.

In order to test the algorithm also for its performance on three warehouses, we modify

the data stream as follows. In addition to the two real warehouses A and B we introduce a third warehouse C. The data and parameters of the third warehouse correspond to a real warehouse that the company previously operated along with A and B. The driving distance is 20 minutes between C and A and 30 minutes between C and B. Including loading time, transportation time is 30 and 40 minutes, respectively. We set the functions $\underline{T}_r$ and $\overline{T}_r$ correspondingly. Next, we generate a data stream for the three warehouses out of the real world data stream by reassigning articles to these warehouses as follows. Upon the first occurrence of an article identifier for a $p \in \mathcal{P}$ in the real world data stream, the article is randomly assigned a map $w_p : \{A, B\} \to \{A, B, C\}$ that maps the original warehouses A and B to two warehouses $w_p(A) \neq w_p(B)$ out of A, B, and C. All following messages of the data stream that relate to this article are then remapped with this same map, so that e.g. all orders originally referring to this article in warehouse A are now orders for this article at $w_p(A)$, or a transport of a pallet of p from A to B is mapped to a transport from $w_p(A)$ to $w_p(B)$.

So we present results for two scenarios: The first operates on a real world data stream on two warehouses and will be denoted by 2-WH, the second uses a highly realistic data stream for three warehouses, its name is 3-WH. Each instance consists of roughly 144 time steps, 4-6 truck graphs, 2 respectively 6 pallet graphs and between 500 and 1200 article graphs (for an average of 800), the total number of variables ranging between 300000 and 1.4 million. For academic purposes an anonymized version of the split data stream, the scripts for generating the instances and our compiled code can be downloaded from

`http://www.tu-chemnitz.de/mathematik/discrete/projects/warehouse_trucks/software/`

In our application it is not useful to spend much time on computing an exact optimal solution, because the data is uncertain and the optimal solution is in danger of being outdated when it is found. Rather, we want to produce a new solution of reasonable quality quickly. Therefore our parameter setting in the bundle method forces the code to take aggressive steps in a wide neighborhood and to stop computation very early. In particular, we stop if the norm of the aggregate subgradient (which is the norm of the primal violation of the approximate primal solution) is less than 5 and the relative precision criterion indicates a relative precision of 5%. On top of this we stop the code after at most 2000 evaluations. In order to study the effect of this crude approximation we compare, in Table 1 the value of the relaxation to the exact optimum of the relaxation computed by a simplex code (we use the dual optimizer of ILOG CPLEX 8.1 [12] as it performed better than primal simplex or barrier on several test instances). Likewise we compare the quality of the heuristic of §5.3 when applied to approximate solutions generated by the bundle method and when applied to the exact primal LP solution.

The first two columns of Table 1 give the name of the scenario (2-WH refers to the scenario with two warehouses, 3-WH to three warehouses) and the number of instances therein. Each instance corresponds to one planning run with available truck data. In each following column we list the average and, in parenthesis, the deviation of the respective values over all instances. Computation times refer to a Linux PC with Pentium 4, 3.2 GHz processor, 1 GByte RAM, and 1 MByte cache. For the LP relaxation computed by the dual simplex method of [12] we display average and sample standard deviation

Table 1: Average and deviation of running time and relative precision

		LP by simplex		bundle		
scenario	#inst.	time(sec)	heur-gap(%)	time(sec)	relax-gap(%)	heur-gap(%)
2-WH	942	2688 (1815)	5.32 (5.96)	234 (97)	3.37 (3.44)	5.75 (6.13)
3-WH	942	7297 (5993)	19.4 (12.8)	316 (131)	6.45 (4.34)	18.8 (11.7)

of computation time in seconds and the relative precision of the heuristic generated by rounding the primal solution. For the bundle solution we display time, relative precision of the relaxation, and the relative precision of the heuristic solution in comparison to the value exact LP-relaxation,

$$\text{relax-gap} = 100 \cdot \left(1 - \frac{\text{bundle-solution}}{\text{LP-solution}}\right), \quad \text{heur-gap} = 100 \cdot \left(1 - \frac{\text{LP-solution}}{\text{heuristic-solution}}\right).$$

Observe that for scenario WH-2 the bundle approach needs 4–5 minutes to compute a solution within 3.5±3.5% of the true value of the relaxation, while it takes the simplex method roughly $\frac{1}{2}$ to 1 hour to determine the exact solution. The poor quality of the approximate primal solution generated by the bundle method does not lead to significant deteriorations in quality of the rounded solution, since the gaps of the heuristic solutions (both measured with respect to the exact LP-solutions) generated from the true primal optimal solution and the approximation differ by only 0.5%. The results are even more striking for scenario 3-WH. The bundle approach still requires 4–8 minutes while the simplex method needs 2–3 hours. Surprisingly, the heuristic solution generated from the approximate solutions is on average even better than the solution generated from the exact solution.

Clearly, the simplex approach is far too slow for real world online computations, while the bundle method is fast enough without significant differences in solution quality for our current rounding heuristic. The gap between heuristic solution and provable bound is still considerable. There should be some room for improvement on the side of the heuristic, but in fact the quality of the lower bound might be the bigger problem. Indeed, since the cost function is nonlinear and convex, it must be expected that the the solution of the relaxation is in the relative interior of one of the faces of the feasible polytope. This holds even if the latter would form the convex hull of all feasible integral points. In other words, the use of a strict convex combination of feasible truck rides will in general allow significantly better solutions than pure integral solutions. In order to improve the bound one would therefore have to generate the convex hull of the epigraph of the cost function evaluated at feasible points.

The decisive criterion for the success of the method is whether the generated solutions lead to significant improvements in the availability of products at the warehouses. In order to study this aspect we present simulation results for a consecutive time span of 100 days (June 1, 2004 to September 15, 2004) in the data stream where the truck data is available throughout[4]. We removed all transportation messages and inserted the transports

[4]The actual starting point of the simulation was set to one week earlier so that the initial crossover phase does not enter the results.

suggested by the planning algorithm instead. In particular, for each new schedule all decisions were accepted that had to be initiated according to this schedule before the start of the next planning run, independent on whether the realization of these decisions would extend into the next planning period or not. Trucks were routed consistently over time (without passing positional information to the data stream, since this is not happening in practice either) and "surprise pallets" of the original data stream (some pallets are started by third parties due to production processes rather than external orders) were included in transportation so that major reassignments of pallets to trucks on a first come first serve basis were necessary during simulation. Thus, the actual realization of the transports often differed considerably (and quite realistically) from the intended schedule. The generated data stream, however, differs only in transportation data from the original one. All orders, picks, and internal movements/retrievals are preserved. This may cause some additional negative supplies, but negative supplies occur regularly in practice due to asynchronous communication or manual booking errors and they can be handled by the approach without problems.

All simulation runs were computed on a Linux PC with a Pentium 4, 2.8 GHz processor, 512 KB cache and 1 GByte of RAM. On this machine reading the current configuration and processing the next part of the data stream for generating the input took about one and a half minute. Computation time statistics for solving the relaxation by the bundle method and running the heuristic for the simulation on two warehouses (2-WH-sim) give an average of 329 CPU-seconds with a deviation of 89 seconds and a maximum of 487 seconds, for three warehouses (3-WH-sim) the average was 416 CPU-seconds, deviation 101, and maximum 599 seconds. Thus computation time is always below 10 minutes for each run of 2-WH-sim and below 12 minutes for 3-WH-sim.

It is difficult to give good criteria for comparing original to simulated warehouse configurations on basis of online data, since the original data stream was definitely influenced by the availability of certain products at the warehouses. We decided to compare, for various supply levels, the number of pallets that would need transportation to achieve the respective supply level at all warehouses ("missing" pallets at one warehouse are counted if they are available in excess of the desired level in at least one of the other warehouses). This number can only be computed for articles p whose pallet size $\theta_p > 0$ is given in the article basics. Fortunately, they account for 75% of all articles and form the main bulk of pallets transported, so the measure should be quite reliable. Table 2 lists for each scenario the average and deviation values over all instances of the number of pallets needed according to the following levels (each level includes the count of the previous levels). Column *negative* counts the pallets needed to get all supplies non-negative. Column *demand* are the pallets required to cover negative supply and known demand of the next six days. In column $F_p^w \geq 0.3$ we list the pallets needed so that for all articles p and warehouses w we have $F_p^w(\alpha_p^w) \geq 0.3$ where α_p^w denotes the amount of p at w available on top of the known 6-days-demand. In words, for each article and warehouse, supply should suffice for an additional stochastic demand of three more days with a probability of 30% . Column $F_p^w \geq 0.9$ is defined correspondingly for the supply level of 90%. The last column *#trans* displays the number of all pallets actually transported by the trucks during the entire period.

In order to present a clear picture of the development over time, we also include plots of the actual numbers of pallets needed in figures 3 and 4. The number of pallets transported up to day $i \in \{1, \ldots, 100\}$ by the various scenarios is displayed in Figure 5.

Table 2: Average and deviation of the number of pallets needed

scenario	#inst.	negative	demand	$F_p^w \geq .3$	$F_p^w \geq .9$	#trans
2-WH-orig	400	157.4(16.8)	296.8(133.8)	671.7(268.5)	2284.5(565.1)	63433
2-WH-sim	400	2.5(4.2)	49.8(65.4)	80.3(86.7)	865.3(568.2)	61364
3-WH-sim	400	2.1(3.4)	37.2(61.9)	45.2(64.1)	201.4(115.1)	62345

The results of Table 2 show a clear superiority of the automatic planning tool (2-WH-sim and 3-WH-sim) versus the human planner (2-WH-orig). One should, however, be careful in interpreting these numbers. We first comment on the 2-WH-sim scenario and discuss the 3-WH-sim scenario afterwards.

In 2-WH-sim it is reassuring that the number of pallets to compensate negative supply is almost zero, so the correct supply is made available by the automatic planning tool without any interaction with the actual retrieval process (remember that retrieving non existent pallets is allowed and simply generates negative amounts!). In contrast, the number of pallets needed to compensate the negative amounts in 2-WH-orig is constantly quite high. This may be due to some human insight, that for a significant number of these articles negative amounts do not require action. Constant offsets have no influence on the deviation, and the data of Figure 3 indicates that at most two thirds of the "negative" pallets (100 pallets say) belong to this class of neglectable negatives. Even after subtracting this number from all averages in the first row, the simulation results are considerably better both in average and deviation while our solution needs fewer transports in total. The most relevant and reliable data is probably column *demand* after subtracting column *negative*, since these pallets need immediate transportation to satisfy known demand. Here, the real world solution shows a need of 139±117 pallets while our computed solution reduces this to 37±62 pallets. Thus this number is more more than halfed. Notice also, that in the plots of Figure 3 the height of the the peeks diminishes for $F_p^w \geq 0.9$ over time for the simulation run, which is not the case for the original data stream. This indicates that the generated distribution functions do their job reasonably well and on long term the warehouses supply structure should get even more favorable. The main advantage of the automatic planning tool over manual planning might rely on the fact, that human planners tend to order large amounts of a few articles that currently need transportation rather than ordering appropriate amounts for all orders that are close to being low on supply.

Note, that for the scenario 3-WH-sim we do not have an authentic real world transportation schedule to compare to. As a remedy we remap the original transports for two warehouses in the same way that the articles are reassigned to the three warehouses. Conceptually this corresponds to splitting a single original truck ride into up to 6 virtual truck rides, each taking care of the transports going into the newly assigned directions. Obviously this does not yield a feasible truck schedule, but it certainly gives rise to feasible and real-

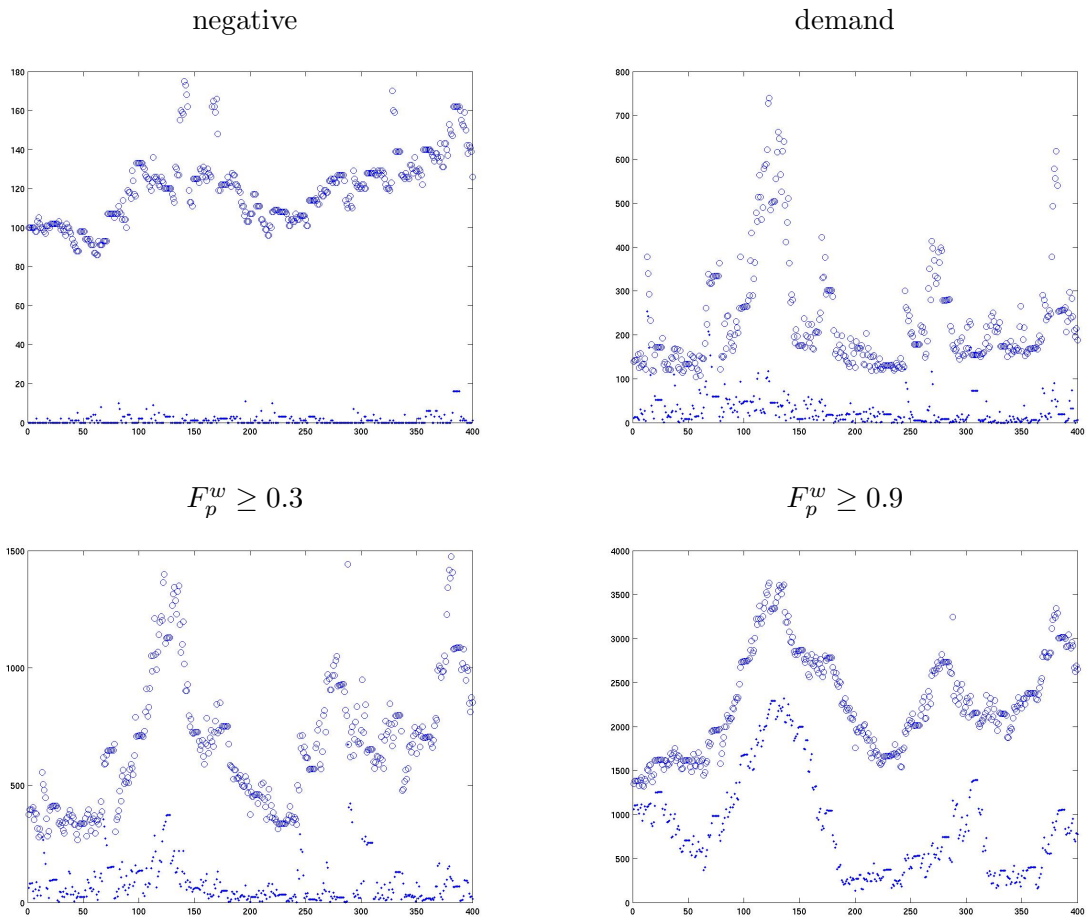

Figure 3: Comparing 2-WH-orig to 2-WH-sim with respect to pallets needed per run for each supply level. Data of 2-WH-orig is shown as "○", data of 2-WH-sim as "·".

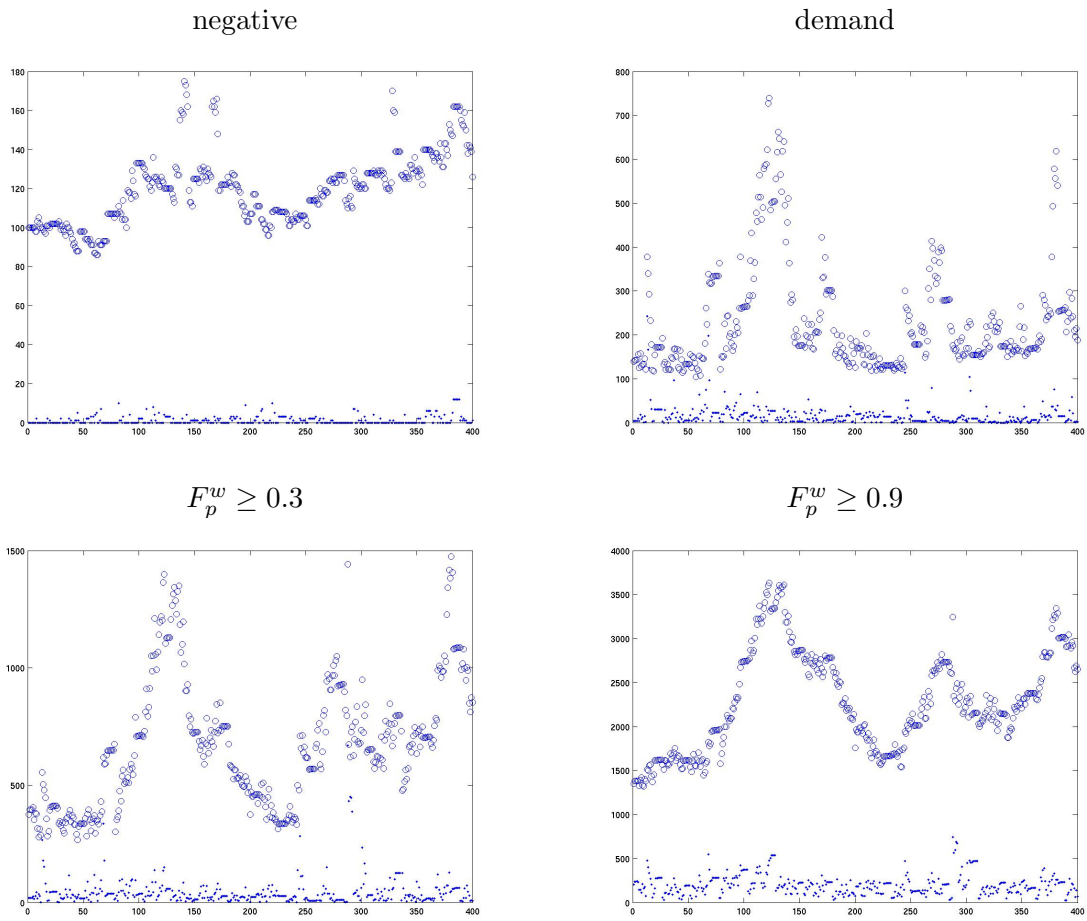

Figure 4: Comparing 2-WH-orig to 3-WH-SIM with respect to pallets needed per run for each supply level. Data of 2-WH-orig is shown as "○", data of 3-WH-SIM as "·".

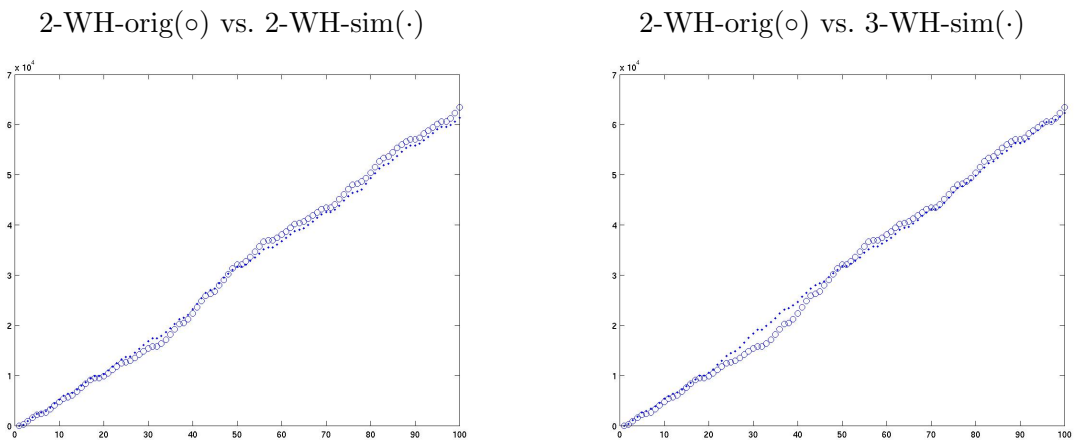

Figure 5: Comparing 2-WH-orig and simulations with respect to the number of pallets transported. The data displayed is the sum of pallets transported up to day $i = 1, \ldots, 100$.

istic warehouse-configurations at the three warehouses. Indeed, per article and remapped warehouse we obtain exactly the same demand and supply values as in the original 2-WH-scenario. Therefore the sum of the needed pallets over all articles and warehouses is the same for 2-WH-orig and the 3-WH-scenario with remapped original transports. In this light it is reasonable to compare the 3-WH-sim scenario to the 2-WH-orig scenario as it is done in the plots of figures 4 and 5.

The approach seems to work even better for 3-WH-sim than for 2-WH-sim, but this might be mostly due to the random (but fixed) reassignment of the original two locations of each article to two new locations. This way demand is randomly (and thus more evenly) distributed over all three locations. In consequence the need for truck rides between the warehouses is also more evenly spread and this makes it easier to satisfy demand on time. Even though the comparison to the remapped original solution does not allow to draw any conclusion on the advantage of automatic versus human solution, the results show that the proposed approach also works very well in a realistic online scenario for three warehouses.

7 Conclusions

The presented approach for joint online truck scheduling and inventory management is fully suitable for online use in practice and significantly outperforms the manual half automatic planning approach of our industrial partner. We see our main contribution in the development of a suitable convex and piecewise linear cost function, that renders the corresponding LP-formulation sufficiently robust so that even rough approximations to the primal optimal solution give excellent results in actual online runs. The suitability for fast approximate solvers such as the bundle method is vital since exact LP-solvers turn out to be far too slow for online applications of this size.

Still, much remains to be done: further improvements in quality could be expected from including in the model uncertainties in driving time, loading capacity, and positioning of the trucks; it might also help to analyze the observed quantities stored on pallets or the actual transportation time of the pallets; in generating the distributions, enhancements are conceivable via better use of statistical data, e.g. by exploiting knowledge on joint appearance of articles in orders, etc.

Acknowledgements. This work was supported by research grant 03HEM2B4 of the German Federal Ministry of Education and Research[5] and would not have been possible without our industrial partners at eCom Logistik GmbH & Co. KG and Herlitz PBS AG, in particular F. Eckle, H. Gebel, and W. Rüstau. Much of the research was carried out while both or part of the authors were employed at the Konrad-Zuse-Zentrum für Informationstechnik Berlin. Special thanks go to Nicole Kruse who helped in setting up the project. We are also grateful to our students Philipp Friese (ZIB Berlin) and Frank Fischer (Chemnitz University of Technology) for their excellent work that helped significantly in visualizing and testing our approach, and to the respective institutions, that provided financial support for these students.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1993.

[2] S. Anily and A. Federgruen. Two-echelon distribution systems with vehicle routing costs and central inventories. *Oper. Res.*, 41(1):37–47, 1993.

[3] T. W. Archibald, S. A. E. Sassen, and L. C. Thomas. An optimal policy for a two depot inventory problem with stock transfer. *Manage. Sci.*, 43(2):173–183, 1997.

[4] J. Boudoukh, M. Richardson, and R. Whitelaw. *The Best of Both Worlds: A Hybrid Approach to Calculating Value at Risk*. Risk, 1998. Reprint in Internal Modeling and CADII: Qualifying and Quantifying Risk within a Financial Institution (Risk Books, London, England), 1999.

[5] S. Feltenmark and K. C. Kiwiel. Dual applications of proximal bundle methods, including Lagrangian relaxation of nonconvex problems. *SIAM J. Optim.*, 10(3):697–721, 2000.

[6] E. H. Frazelle, S. T. Hackman, U. Passy, and L. K. Platzman. The forward-reserve problem. In *Proceedings of the 1992 IBM Europe Institute on optimization solutions*, pages 43–61, Oberlech, Austria, 1994.

[7] F. Fumero and C. Vercellis. Synchronized development of production, inventory, and distribution schedules. *Transp. Sci.*, 33(3):330–340, 1999.

[8] S. Graves, A. Rinnooy Kan, and P. Zipkin, editors. *Logistics of production and inventory*, volume 4 of *Handbooks in Operations Research and Management Science*. North-Holland, Amsterdam, 1993.

[5] Responsibility for the content rests with the authors.

[9] P. Hall. *The Bootstrap and Edgeworth Expansions*. Springer, New York, 1992.

[10] C. Helmberg and K. C. Kiwiel. A spectral bundle method with bounds. *Math. Programming*, 93(2):173–194, 2002.

[11] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II*, volume 306 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, Heidelberg, 1993.

[12] ILOG S.A., Gentilly, France. *ILOG AMPL CPLEX System, Version 8.1, User's Guide*, Dec. 2002. Information available at `http://www.ilog.com`.

[13] P. Köchel. Ein dynamisches Mehr-Lager-Modell mit Transportbeziehungen zwischen den Lagern. *Math. Operationsforsch. Statist., Ser. Optimization*, 13:267–286, 1982.

[14] P. Köchel. Optimal adaptive inventory control for a multi-location model with redistribution. *Optimization*, 19:525–537, 1988.

[15] B. Korte and J. Vygen. *Combinatorial optimization. Theory and algorithms*, volume 21 of *Algorithms and Combinatorics*. Springer, Berlin, 2 edition, 2002.

[16] A. Löbel. *MCF Version 1.2 – A network simplex Implementation*. Konrad-Zuse-Zentrum für Informationstechnik Berlin, Jan. 2000. Available at `http://www.zib.de/Optimization/Software/Mcf` (free of charge for academic use).

[17] M. Padberg. *Linear optimization and extensions*, volume 12 of *Algorithms and Combinatorics*. Springer, 2 edition, 1999.

[18] M. I. Reiman, R. Rubio, and L. M. Wein. Heavy traffic analysis of the dynamic stochastic inventory-routing problem. *Transp. Sci.*, 33(4):361–380, 1999.

[19] J. Shao. *Mathematical Statistics*, Springer Texts in Statistics. Springer-Verlag, New York, Berlin, Heidelberg, 2 edition, 2003.

[20] A. Schrijver. *Combinatorial Optimization*, volume 24 of *Algorithms and Combinatorics*. Springer, 2003.

[21] B.W. Silverman. *Density estimation for statistics and data analysis*, Monographs on Statistics and Applied Probability. Chapman and Hall, London, New York, 1986.

[22] J. P. van den Berg, G. P. Sharp, A. J. R. M. (Noud) Gademann, and Y. Pochet. Forward-reserve allocation in a warehouse with unit-load replenishments. *Europ. J. Oper. Res.*, 111:98–113, 1998.