Croatian Operational Research Review
CRORR **13**(2022), 99-111

Delivery pattern planning in retailing with transport and warehouse workload balancing

Adrian Alajkovic[1], **Mario Brcic**[2,*], **Viktorija Ivandic**[1], **Luka Bubalo**[1], **Mihael Koncic**[1], **and Mihael Kovac**[2]

[1] *KONZUM plus d.o.o.,Marijana Čavića 1/a, Zagreb, Croatia*
E-mail: ⟨{adrian.alajkovic, viktorija.ivandic, luka.bubalo, mihael.koncic}@konzum.hr⟩

[2] *University of Zagreb Faculty of Electrical Engineering and Computing, Unska 3, Zagreb, Croatia*
E-mail: ⟨{mario.brcic, mihael.kovac}@fer.hr⟩

Abstract. Goods from warehouses must be scheduled in advance, prepared, routed, and delivered to shops. At least three systems directly interact within such a process: warehouse workforce scheduling, delivery scheduling, and routing system. Ideally, the whole problem with the preceding inventory management (restocking) would be solved in one optimization pass. In order to make the problem simpler, we first decompose the total problem by isolating the delivery scheduling. Then we connect the optimization model to the rest of the system by workload balancing goal that is a surrogate of coordination and criterion for the system robustness. This paper presents the practical application of top-down discrete optimization that streamlines operations and enables better reactivity to changes in circumstances. We search for repetitive weekly delivery patterns that balance the daily warehouse and transportation utilization in the absence of capacity constraints. Delivery patterns are optimized for the quality criteria regarding specific store-warehouse pair types, with a special focus on fresh food delivery that aims at reducing inventory write-offs due to aging. The previous setup included semi-manual scheduling based on templates, historical prototypes, and domain knowledge. We have found that the system augmented with the new automated delivery scheduling system brings an improvement of 3% in the performance measure as well as speed in adjusting to the changes, such was the case with changes in policies during COVID-19 lockdowns.

Keywords: delivery, discrete optimization, retailing, scheduling

Received: April 15, 2022; accepted: May 25, 2022; available online: July 12, 2022

DOI: 10.17535/journal.2022.0007

1. Introduction

This paper presents a real-world application of operations research methods in the supply chain management of a major retailer. We present the system that schedules deliveries to the network of 420 shops in Croatia.

Our problem is set within the context of supply chain consisting of three major links:

1. distribution center (DC)
2. transportation system
3. stores.

*Corresponding author.

Distribution center. When orders are issued from the main system, they are processed in DC warehouses for different segments of goods. Workers have to traverse the warehouse and pick items from the order and take them to the loading area. The number of workers and hence picking volume is limited and usually, it is good practice to balance the work over days.

Transportation system. Items at the loading area are loaded into trucks that have limited capacity. Some trucks can only carry one type of goods. Multi-compartment trucks can load goods from different segments. After loading, trucks follow a predetermined route to visit all the shops on their itinerary. When a truck arrives at a shop the delivery is unloaded.

Stores. Upon unloading, store employees check the received items and administratively finalize the delivery. New items are first directly refilled to shelves and the surplus is put into the backroom from which shelf refilling can take place when the need arises. The additional refilling process is substantially costlier than direct shelf filling. Both shelf and backroom capacities are limited and store-specific.

1.1. Delivery patterns

Demand, which is usually patterned repetitively across a week, is consuming goods at stores. The replenishment must be done downstream, from the distribution center to stores using a transportation system. Therefore, it makes sense to model the replenishment according to repetitive weekly delivery patterns and this is in fact what is often done [10], as in our case. Hence, the usage of weekly delivery patterns, which are fixed weekly schedules of deliveries from each warehouse to the store that is repeated in all standard weeks within a horizon of several months (typically six) [6, 8]. Standard weeks are weeks without holidays, but with a common working pattern. The details such as the number of weekly deliveries are tuned to the sales process and store size.

Delivery schedules must respect the workload balance in transport and warehouses in order not to overload any subsystem. Weekly delivery patterns act as the baseline schedules. The baseline schedule is taken as a basis for schedules in all the successive weeks within some horizon. Potential reactive changes, based on the specifics of certain weeks (for example holidays or other changes in assumptions), are done with a local search in the neighborhood of the baseline schedule. Orchestrating and aligning all subsystems in supply chains is very demanding. Different systems have different speeds of adaptation to changes. Technological improvements such as IT technologies might decrease time-frames for adaptation. It is shown that the arrangement with a weekly baseline is beneficial to the alignment in the supply chain since there is eventual informational convergence between different actors [3, 6, 15].

1.2. Problem specifics

We will focus on the problem of replenishing stores from one DC that has several segments (delivered from different warehouses in DC), though the problem can be repurposed to multiple DCs and their warehouses.

The setting of our realistic problem sets the context within which we seek the ideal model. The transportation system uses a separate software for routing that also takes into account the truck capacities. It works upon the input delivery schedule. Since the optimization within the transportation system is a black box from which we did not receive feedback, we decided to strive for balancing the whole system to reduce all costs due to variance that results in overtime and/or part-time hiring. Additionally, it reduces peak work on unloading and shelving, and necessary peak store capacity, since our pattern taboo procedure balances inter-delivery periods.

1.3. Contributions

Contributions of our paper are:

1. Model for creating delivery patterns with transport and warehouse workload balancing which uses sub-pattern matching and penalizing or forbidding certain patterns. This is in contrast with most of the related work that assigns preselected delivery patterns to the stores.

2. Schedule quality function shaping using combinatorial constructs that, in the absence of justified and group-adjusted individual shop delivery pattern preferences, is an objective way of evaluating schedules through flexibility

3. Case study of implementing and using the model in operations of the real-world supply chain that resulted in 3% improvement in the objective and decrease in the number of separate shippings, leading to better efficiency, resulting in a reduction of fresh food waste.

The paper is organized as follows. Section 2 presents related work. Section 3 details the preprocessing of the input to the model. In Section 4 we present our model. In Section 5 we list our results and measured benefits of using in operations. Finally, in Section 6 we offer our conclusion with future work.

2. Related work

Our work falls into the area of delivery scheduling, especially when using delivery patterns.

Ronen and Goodhart [12] cluster similar stores together. Additionally, they preselect delivery patterns that they assign to stores using mixed integer programming (MIP) and then build transportation routes using PVRP. The optimization goes in stages without reoptimizing and feedback between them. Instore logistics costs are ignored. Sternbeck and Kuhn [14] propose a MIP model that assigns preselected patterns to the stores while taking into account the costs across all three major links, including in-store logistics. This model does not take into account bundling effects across stores. A real industry case example was solved with 1000 stores, 6 DCs, and 3 segments to achieve 5.3% savings in operational costs.

Holzapfner et al. [7] have improved upon [14] by taking into account bundling effects and where clustering is done as a part of optimization. Delivery patterns are assigned from a preselected collection and travel costs are approximated by distance. A real industry case example was solved with 48 stores, 1 DC, and 1 segment to achieve 2.5% savings in costs in comparison to [14].

Frank et al. [5] have created simultaneous optimizing routing and delivery patterns when using multi-compartment vehicles with the assumption of a homogenous fleet. Each truck is assumed to have one delivery tour per day. The approach uses a hybrid iterative multi-stage algorithm with adaptive large neighborhood search (ALNS) for selection of delivery patterns, large neighborhood search (LNS) for routing decisions, and simulated annealing for selection of search parameters for previous stages. A real industry case example was solved with 376 stores, 1 DC, and 3 segments on average in 2.89hours to achieve 7.68% savings in costs in comparison to the status quo approach. The pre-existing solution at the retailer did not coordinate delivery patterns with routing, but everything was done in one sequential pass. Also, this approach improves the results of [7] on benchmarks with up to 40 stores, single segment, and single compartment vehicles by 1.25% on average. This improvement is attributed to actually solving the vehicle routing problem.

All the aforementioned papers were created in succession and were built directly on top of each other by improving the capabilities of previous works. We were mostly inspired by work [7]. However, we were making the solution under a different setting (see subsection 1.2) and

in times of turmoil so we wanted to make a model quickly. For that reason, we have decided not to collect lists of predefined patterns for shops but instead to use flexibility-based metrics for sub-pattern quality. Also, we wanted a model that can be quickly solved on available MIP solvers to forgo the time for algorithm design. Due to the latter, we took inspiration from [12] and we decided to use clustering in the preprocessing phase, instead of during the optimization phase. This made the model have a favorable structure as evidenced in section 5.

3. Preprocessing

In preprocessing we have decided to fix certain data, as well as make certain simplifying assumptions. Here we list each of those elements.

3.1. Clustering

To reduce the problem complexity, unlike [7], we have decided to learn shop assignments to route clusters by clustering from the historical data and correcting with expert interventions. Our procedure used hierarchical graph clustering. At the lowest level, we created routes that are close to the typical usage. At the higher level, we have created group clusters for balancing the workload. The clustering results are saved within the incidence matrices H and J (see Table 1).

3.2. Business rules

We have agreed upon an internal list of business rules that should be enforced in deliveries, which were informal rules of best practices in the operations, but there was no strict checking. The rules were of two types: fixed delivery schedule elements and delivery coupling/decoupling. These business rules describe the current way of operating and reflect current geographical and operational decisions. The new model that mirrors many of the aspects, enables fast what-if analyses with existing and modified rules in the system. It also enables the identification of inherent constraints of the system and constraints that were adopted due to the problem complexity for humans and that can be relaxed when the problem is solved by the computational procedure. The effects of rules are saved within the parameters F, C, and U (see Table 1).

3.3. Fixed decisions

Furthermore, several decisions are already made before optimization, based on forecasting and historical performance. The number of weekly deliveries is fixed by several parameters, expressed as business rules that reflect historical best practices. Additionally, activation of delivery routes throughout a week is also fixed at the input. These fixed decisions are input into the model as parameters $N_{i,j}$ and $RAD_{i,j}$ (see Table 1).

4. The model

The model consists of the variables and parameters with domains and semantics given in Table 1 and Table 2.

4.1. Objective function

We use a single objective function where multiple contributions are weighted.

parameter	definition
W	set of warehouses
W_{fresh}	set of warehouses for fresh food
S	set of shops
D	set of days
R	set of routes
G	set of groups of routes
H	incidence matrix between G and R
J	incidence matrix between R and S
$N_{i,j} \in \mathbb{N}_0$	number of deliveries between warehouse i and store j in the delivery pattern
$F \subseteq \{(i,j,k,\{0,1\})\|i \in W, j \in S, k \in D\}$	fixed schedule elements
$C \subseteq \{(i,j,k)\|i,j \in W, k \in S\}$	combining deliveries (i,k) with (j,k)
$U \subseteq \{(i,j,k)\|i,j \in W, k \in S\}$	separating deliveries (i,k) from (j,k)
$I : W \times S \times D \longrightarrow \mathbb{N}_0$	daily delay of work on warehouses for the delivery to shops
$AA_{i,j} \in \mathbb{R}$	average number of articles per delivery for warehouse i and shop j
$AP_{i,j} \in \mathbb{R}$	average number of palettes per delivery for warehouse i and shop j
$UWL_{i,j}, LWL_{i,j} \in \mathbb{R}$	upper and lower limit of warehouse i on day j
$UGL_{i,j}, LGL_{i,j} \in \mathbb{R}$	upper and lower limit of route group i on day j
$WAD_{i,j} \in \{0,1\}$	1 if warehouse i is active on day j, 0 otherwise
$RAD_{i,j} \in \{0,1\}$	1 if route i is active on day j, 0 otherwise
$SM \in \{1,...,7\}$	size of minimal working days set

Table 1: Parameters in the model

$$minimize \sum_{i,j,k} WS_{i,j} \cdot (y11_{i,j,k} + z111_{i,j,k} + WL_{i,j} \cdot z101_{i,j,k}) + \sum_{(i,j,k) \in U, l \in D} WC_{i,j,k} \cdot m11_{i,j,k,l} \quad (1)$$

On the top level, it is a combination of individual (the first composite term under the sum) and warehouse-interaction taboo patterns (the second term) for each store. The weighting is based on importance and flexibility and it is further explained in the following subsection.

4.1.1. Weighting

For future work, we intend to elicit group preferences from all the supply chain participants and use these parameters (in addition to the financial ones) as weighting factors between different aspects of the quality of the solution. This means collecting preferences from all the perspectives: transport, warehousing, and shop personnel which include intimate local expertise not captured in the data.

As the first approximation, we have used the weighting based on the schedule flexibility of different shops. We will treat the shop's schedule as more flexible if it has many possible solutions and less flexible if it has only a few. Hence, we want to focus our attention on flexible shops and give them more weight.

For individual-taboo weights in equation (2) we use number of combinations $\binom{\#\text{ open days}}{\#\text{ deliveries}}$ normalized with the total number of deliveries. In equation (3), pattern $'101'$, which groups the only two deliveries too-close, has smaller weight (arbitrary constant selected and tuned by experts) than adjacent groupings expressed with patterns $'11'$ and $'111'$.

variable	definition
$x_{i,j,k} \in \{0,1\}$	decision to deliver from warehouse i to store j on day k
$w_{i,j} \in \mathbb{R}^+$	amount delivered from warehouse i on day j
$r_{i,j} \in \mathbb{R}^+$	amount delivered over route i on day j
$g_{i,j} \in \mathbb{R}^+$	amount delivered over group of routes i on day j
$y\mathbf{pp}_{i,j,k} \in \{0,1\}$	pattern **pp** in deliveries from warehouse i to store j on two consecutive days $k, k+1$
$z\mathbf{ppp}_{i,j,k} \in \{0,1\}$	pattern **ppp** in deliveries from warehouse i to store j for three consecutive days $k, k+1, k+2$
$m\mathbf{pp}_{i,j,k,l} \in \{0,1\}$	interaction pattern **pp** in deliveries from warehouses i and j to store k on day l

Table 2: Decision variables in the model

$$WS_{i,j} = \frac{\left(\frac{\sum_{k \in D} WAD_{i,j}}{N_{i,j}}\right)}{N_{i,j}} \tag{2}$$

$$WL_{i,j} = 0.2 \cdot \mathbb{1}_{N_{i,j}=2} \tag{3}$$

For warehouse-interaction taboo weights, we have followed a similar idea in eq.(4). We calculated the number of overlaps between the delivery schedules from different warehouses to the same shops which we have normalized. Finally, we calculated the square root in order to put the weight on a similar scale as individual-taboo weights.

$$WC_{i,j,k} = \sqrt{\frac{\#\text{overlapping combinations}}{\min(N_{i,k}, N_{j,k})}} \tag{4}$$

4.2. Input data

Most of the elements in Table 1 are self-explanatory. Set of a group of routes G defines the elements over which the balancing is done for the transport system. Incidence matrices H and J define the results of hierarchical clustering of the shops into two levels: routes and then groups of routes. Effects of different business rules are input through the fixations to the schedule (F), and by combining and separating deliveries (C and U). Combining deliveries use triplet specification to define asymmetric relation whereby certain delivery from warehouse i must be combined with some of the deliveries from warehouse j. This enables modeling deliveries with multi-compartment vehicles even when the number of deliveries from warehouses i and j differ. Separating deliveries use triplet specification to define which pairs of deliveries are penalized in the objective function given in equation (1). Delivery that must be delivered on the day k incurs earlier work on the warehouse. Daily delay $I(i,j,k)$ states the delivery delay for the work done in warehouse i for shop j on day k. Average weekly numbers for delivery sizes are given in pallets ($AP_{i,j}$) and articles ($AA_{i,j}$) per shop. These delivery sizes are used for balancing work in warehouses (articles) and transport (pallets). Balancing is controlled through tolerance bands. We use pairs of parameters to describe width and displacement of tolerance bands relative to the weekly average; $(UWL_{i,j}, LWL_{i,j})$ for warehouses and $(UGL_{i,j}, LGL_{i,j})$ for groups of routes. Finally, warehouses do not work on all days, and routes are not active on all days, which is input through the $WAD_{i,j}$ and $RAD_{i,j}$ parameters. Minimal working days set is the set of days when all the warehouses are active - and plays a role when balancing the transport (see 4.4.2).

4.3. Decision variables

Decision variables in the model are listed in the Table 2. The main focus of the optimization is the schedule which is within the binary decision variables x. Decision variables w keep track of amounts delivered from warehouses and they are used for balancing the workload. Decision variables r and g keep track of amounts delivered over routes and groups of routes in order to balance the workload in transport. We use soft and hard taboos in order to shape the delivery schedule. Soft taboos penalize certain patterns in the schedule through the objective function. Hard taboos forbid certain patterns in the schedule through the constraints. Variables with prefixes y,z, and m are used as pattern recognizers within the schedule. They are further named by the pattern that they recognize. For example, variables $y11$ recognize deliveries on two consecutive working days for the same pair of warehouse and store. We instantiate variables y for patterns in $\{11\}$ and use them as soft taboos. We instantiate variables z for patterns in $\{000, 111, 101\}$. Variables $z000$ are used as hard taboos, the others are used as soft taboos. We instantiate variables m for patterns in $\{11\}$ and use them for coupling and decoupling deliveries. Coupling is done through the constraints, while decoupling is achieved through soft taboos.

4.4. Constraints

We try to model the preferability of certain patterns that appear in the delivery schedule by negative discrimination in the form of soft and hard taboo sub-patterns that are undesirable or even forbidden. Hard taboo sub-patterns are covered by constraints that forbid them. Soft taboo constraints are counted by their respective decision variables, which are designated pattern matchers, and these variables enter the objective function in the form of penalties in the objective function (see subsection 4.1).

$$\sum_{k \in D} x_{i,j,k} = N_{i,j}, \forall i \in W, \forall j \in S \tag{5}$$

$$y\mathbf{pp}_{i,j,k} = \begin{cases} 1, & (x_{i,j,k}, x_{i,j,k+1}) = \mathbf{pp} \\ 0, & otherwise \end{cases}, \forall i \in W, \forall j \in S, \forall k \in D \tag{6}$$

$$z\mathbf{ppp}_{i,j,k} = \begin{cases} 1, & (x_{i,j,k}, x_{i,j,k+1}, x_{i,j,k+2}) = \mathbf{ppp} \\ 0, & otherwise \end{cases}, \forall i \in W, \forall j \in S, \forall k \in D \tag{7}$$

$$m\mathbf{pp}_{i,j,k,l} = \begin{cases} 1, & (x_{i,k,l}, x_{j,k,l}) = \mathbf{pp} \\ 0, & otherwise \end{cases}, \forall (i,j,k) \in U, \forall l \in D \tag{8}$$

$$x_{i,j,k} = v, \forall (i,j,k,v) \in F \tag{9}$$

$$x_{i,k,l} \leq x_{j,k,l}, \forall (i,j,k) \in C, \forall l \in D \tag{10}$$

$$x_{i,j,k} \leq RAD_{l,k}, \forall i \in W, \forall (l,j | J_{l,j} = 1), \forall k \in D \tag{11}$$

$$z000_{i,j,k} = 0, \forall i \in W_{fresh}, \forall j \in S, \forall k \in D \tag{12}$$

We set the required number of deliveries per shop-warehouse pair in (5). Constraints (6) are used to match and find all patterns of interest over the successive pairs of days (i.e. pair

taboos). Therein, **pp** is used as a placeholder for exact pattern of interest. For example, variables $y11_{i,j,k}$ would locate situations (i.e. be equal to 1) with two deliveries on successive days k and $k+1$ to a shop j from the warehouse i. We currently use only one pair taboo, that is $pp \in \{11\}$ for (6).

In a similar fashion we locate patterns (i.e. triplet taboos) over successive triplets of days in (7). Therein, **ppp** is used as a placeholder for exact pattern of interest. For example, variables $z101_{i,j,k}$ would locate situations (i.e. be equal to 1) with two deliveries on days k and $k+2$ and no delivery on day $k+1$ between the shop j and the warehouse i. We currently use three triplet taboos, that is $ppp \in \{111, 101, 000\}$ for (7).

Constraint (12) forbids a pattern of three consecutive days without delivery of fresh food by using the $z000$ variables. $W_{fresh} \subseteq W$ is the set of warehouses for fresh food. This is important for reducing inventory write-offs due to aging.

Constraints (8) match and find all the patterns of interaction between the same-day deliveries from different warehouses (i.e. interactive pair taboos). These patterns can be used to force packing together or to separate different deliveries. For example, variables $m11_{i,j,k,l}$ would locate situations (i.e. be equal to 1) with two deliveries on a day l to a shop k from the warehouses i and j. We currently use only one interactive pair taboo, that is $pp \in \{11\}$ for (8).

Constraints (9) are used for fixed schedule elements. Each such constraint forces a fixed prespecified decision v on delivery from warehouse i to shop j on day k.

Constraints (10) are used to combine deliveries from two warehouses, i and j to the same shop k on the same day l. Deliveries from warehouse i are combined with the deliveries from warehouse j, whereby j may have more deliveries.

Constraints (11) deactivate deliveries depending on openness of delivery routes. For example, if shop j is on a route l and that route is closed on day k, then deliveries from any warehouse i to shop j on day k must be 0.

4.4.1. Balancing warehouses

We balance the work on warehouses within the tolerance band relative to the average work (proportional to the number of pallets) whose width and displacement we shape by parameters $UWL_{i,j}$ and $LWL_{i,j}$. The intention is to reduce the variation across days so that the least amount of overwork or part-time workers is needed to cover such deviations.

$$w_{i,j} = \sum_{s \in S} WAD_{i,j} \cdot AA_{i,s} \cdot x_{i,s,j+I(i,s,j)}, \forall i \in W, \forall j \in D \tag{13}$$

$$w_{i,j} \leq UWL_{i,j} \cdot \frac{\sum_{k \in D} w_{i,k}}{\sum_{k \in D} WAD_{i,k}}, \forall i \in W, \forall j \in D \tag{14}$$

$$w_{i,j} \geq LWL_{i,j} \cdot \frac{\sum_{k \in D} w_{i,k}}{\sum_{k \in D} WAD_{i,k}}, \forall i \in W, \forall j \in D \tag{15}$$

Constraint (13) calculates the total work on warehouse i in day j. Constraints (14) and (15) set up a tolerance band relative to the average work on the warehouse i throughout the whole week. The tolerance band is described with upper and lower limits through the parameters $UWL_{i,j}$ and $LWL_{i,j}$ that the user sets to appropriate values. As an example, they could be set for a certain pair (i, j) to values $UWL_{i,j} = 1.15$ and $LWL_{i,j} = 0.95$ for a tolerance band of width 20% around the average.

4.4.2. Balancing transport

We balance the work in transportation within each group of routes by allowing it only to be within a tolerance band that is defined relative to the average work. We can shape the width

and displacement of the tolerance band by parameters $UGL_{i,j}$ and $LGL_{i,j}$. The intention is to reduce the variation across days so that the least amount of overwork or part-time transporters is needed to cover such deviations.

$$r_{i,j} = \sum_{s \in S} RAD_{i,j} \cdot AP_{i,s} \cdot J_{i,s} \cdot x_{i,s,j}, \forall i \in R, \forall j \in D \tag{16}$$

$$g_{i,j} = \sum_{k \in R} H_{i,k} \cdot r_{k,j}, \forall i \in G, \forall j \in D \tag{17}$$

$$g_{i,j} \leq UGL_{i,j} \cdot \frac{\sum_{k \in D} g_{i,k}}{SM}, \forall i \in G, \forall j \in D \tag{18}$$

$$g_{i,j} \geq LGL_{i,j} \cdot \frac{\sum_{k \in D} g_{i,k}}{SM}, \forall i \in G, \forall j \in D \tag{19}$$

Constraints (16) calculate total work on route i in day j. Similarly, constraints (17) calculate total work on group i in day j.

Constraints (18) and (19) set up an arbitrarily shifted tolerance band around the average work on the group of routes i through the whole week. The tolerance band is described with upper and lower limits through the parameters $UGL_{i,j}$ and $LGL_{i,j}$ that the user sets to appropriate values. As an example, they could be set for a certain pair (i, j) to values $UGL_{i,j} = 1.05$ and $LGL_{i,j} = 0.85$ for a tolerance band of width 20% around the average.

5. Results and benefits

The instances of the model have 40666 decision variables (mostly binary) and 45952 constraints. We have used historical weekly data to create 36 instances. Our computation setting is using a 12-core 10th gen i7 with 16GiB RAM with Linux OS. We have tried solving our instances using the three best solvers within their respective class of licensing (based on MIPLIB2017 benchmarking done in [11]) to compare the performance:

- Gurobi 9.1 - arguably the best performing commercial solver
- SCIP 7.0.2 - the best performing source-available solver
- CBC 2.10 - the best performing free and open source solver

On all tests, we have used a fraction gap of 0.8% as a stopping condition. CBC is severely underperforming as solving instances takes several hours (up to 10h), so we will not take it into further consideration. It takes 19.83s on average for Gurobi to solve these instances and 223.62s for SCIP, when using heuristics and 1 CPU thread. We did an additional test on Gurobi with 8threads and turned off heuristics and that resulted in a substantial increase in runtime where it took on average 1210.82 seconds for solving our instances, with substantially increased variance. In line with [11], we report on geometric mean: 17.02s for 1 threaded Gurobi, 185.4s for 1 threaded SCIP, and 740.95s for 8 threaded Gurobi without heuristics. The details of the performance of solvers regarding duration can be seen in Figure 1 and Figure 2.

In Figure 3 we can see the detailed performance that disregards durations and compares the progress of incumbent solutions (measured in the fractional gap to the lower bound) with a number of processed branch&bound tree nodes. Therein, for each problem we get to see the number of processed nodes for each achieved optimality gap which corresponds to finding new, better performing feasible solutions. Gurobi (Figure 3a) manages to solve 17 out of 36 instances without processing any tree nodes, just by using heuristics. And in all other cases, it manages to find high-quality feasible solutions, which it improves rapidly. We can see in Figure

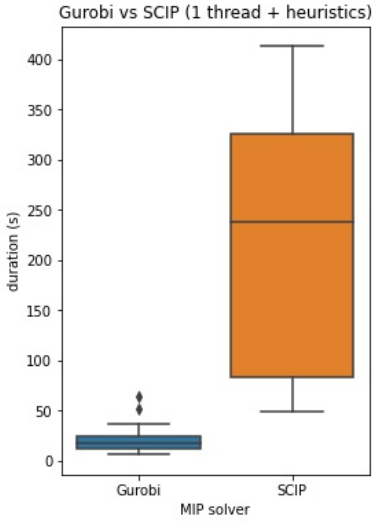


Figure 1: Comparison between Gurobi and SCIP with 1 thread and heuristics on

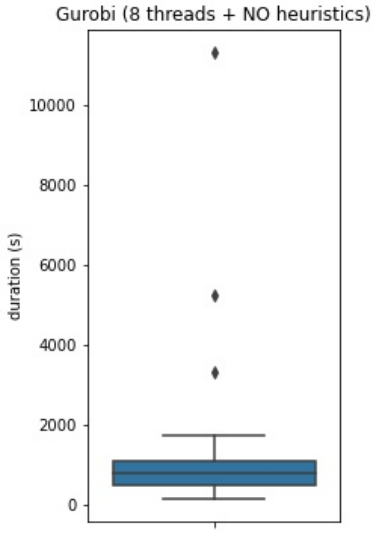
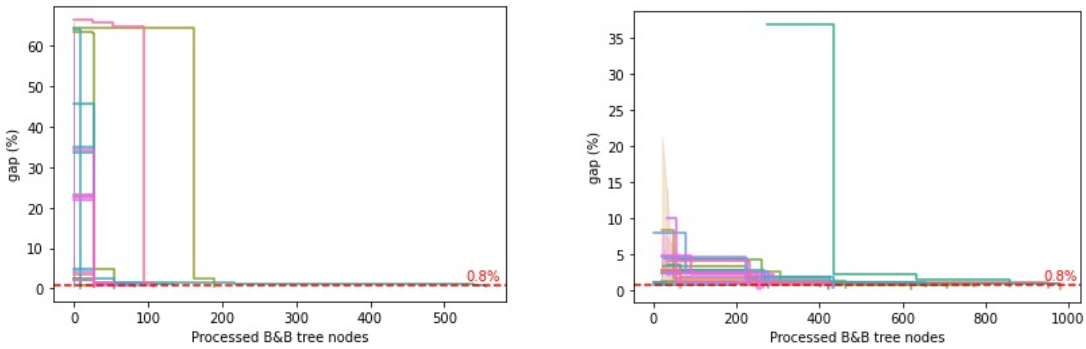


Figure 2: Gurobi performance with 8 threads and heuristics off

4 the performance hit Gurobi solver takes when it is not allowed to use heuristics. In that case, it takes much longer to find the first feasible solution, and later it also takes much longer to improve the gap. In Figure 3b, we can see the performance of SCIP. It is evident that it never managed to solve any instance without processing any nodes. For SCIP, heuristics are a major tool for finding high-quality solutions as well. SCIP reports on heuristics that found the solutions and most often reported heuristics were (in order): adaptive large neighborhood search (ALNS) [13], and local search heuristics such as large neighborhood search (LNS) methods such as RINS [4] and RENS [1].

Faster time to solution enables better adaptivity to changes. This is a crucial benefit in the uncertain pandemic times which are characterized by a shortage of goods, openness to new trends, and swinging demand. We avoid unfavorable or forbidden patterns for shops that cluster the deliveries instead of separating them. This balances the unloading work throughout the week. Additionally, explicit balancing of the work in warehouses and transportation gives



(a) Gurobi gap performance

(b) SCIP gap performance

Figure 3: Gap performance on 1 thread with heuristics on

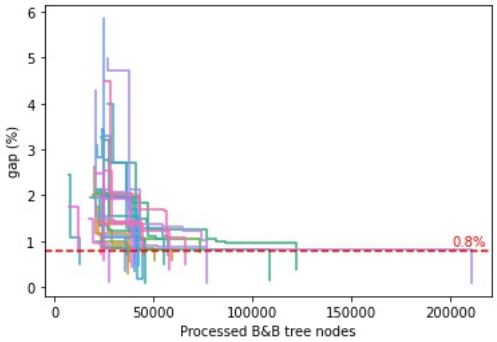


Figure 4: Gap performance on 8 threads without heuristics

balanced utilization of resources across the supply chain.

We have noticed a 3% improvement in the quality of delivery schedules (back-measured against historical data using our objective). Also, a number of deliveries are forced now to go together which results in a smaller number of shop visits (by pairing up the segment deliveries), and by balancing the routes through the week we provide a smaller impact on the roads at all times (no peaks) which reduces peak number of active deliveries on routes. Additionally, the automated decision-making support enables what-if analyses.

6. Conclusion

We have optimized weekly delivery patterns to the network of 420 shops of a major Croatian retailer. At the same time, we balanced the utilization of warehouses and transportation. We put the new delivery scheduling system into operation during the pandemic when the changes in the uncertain times became extreme, and the reaction times to changes had to become faster. There were lockdowns and many legal changes regarding the working days, which changed the definition of standard weeks. We have taken a proactive-reactive approach. We keep the supply chain aligned through a baseline delivery pattern schedule that is active for a long period. We track the performance of the baseline pattern and trigger the re-optimization in the case of degradation. The baseline is a basis of weekly reactive changes for nonstandard weeks - such as holidays and changes in demands, different sales promotions, etc. When rescheduling the baseline, we try to keep the new baseline as close as possible to the previous, while respecting the new needs. Alignment is always an important aspect. Our system is operational since the May 2020 and it has measurable benefits such as a 3% improvement in the objective function compared to the previous approach. Also, by balancing the transportation utilization we reduced the peak and total number of deliveries by pairing them together from different warehouses, when possible. We have seen that our MIP model has the property that it is substantially easier to find high-quality feasible solutions than to find near-optimal solutions. This implies that the problem is not tightly constrained and that the landscape is "flat". In that sense, local search heuristics such as large neighborhood search (LNS) methods such as RINS [4] and RENS [1], as well as adaptive large neighborhood search (ALNS) [13], have proven very effective at finding such high-quality solutions. This is in line with the choice of using ALNS and LNS methods for optimizing similar problems in [5].

At the level of long-term planning, this aggregate model is doing a good job at improving current operations. However, we see further potential for significantly increasing efficiency through more detailed tactical-level planning. In future work, we would like to make decisions at a greater resolution, while now we do so at the granularity of single delivery which uses **average** delivery size indicators. We plan to improve or even remove the need for shop clustering within

preprocessing. Also, we will include storeroom and truck capacities to fine-tune the decisions. We will try to incorporate vehicle traveling costs in form of distance traveled into the model by optimizing the routes. We have tried learning a better branching policy for SCIP solver using a graph convolutional neural network in [9] to accelerate solving this problem. We have had moderate success with respect to the number of processed nodes. Plain and accelerated SCIP each had 2 wins between themselves and in 2 instances they were tied on test instances. Working on learned diving, branching policies, and cuts generation are promising areas that explore inductive limits inherent in hard combinatorial optimization problems, many of which are listed in [2]. Family of *no free lunch theorems* is the most known representative of such limits.

Acknowledgements

The authors would like to acknowledge Fortenova group for their support and approval.

References

[1] Timo Berthold. *Primal heuristics for mixed integer programs*. PhD Thesis, Zuse Institute Berlin (ZIB), 2006.

[2] Mario Brcic and Roman V. Yampolskiy. Impossibility Results in AI: A Survey. *arXiv:2109.00484 [cs]*, September 2021. doi: 10.48550/arXiv.2109.00484.

[3] Gerard Cachon. Managing a retailer's shelf space, inventory, and transportation. *Manufacturing & Service Operations Management*, 3(3):211–229, 2001. Publisher: INFORMS.

[4] E. Danna, E. Rothberg, and C. L. Pape. Exploring relaxation induced neighborhoods to improve MIP solutions. *Math. Program.*, 2005. 10.1007/s10107-004-0518-7.

[5] Markus Frank, Manuel Ostermeier, Andreas Holzapfel, Alexander Hübner, and Heinrich Kuhn. Optimizing routing and delivery patterns with multi-compartment vehicles. *European Journal of Operational Research*, 293(2):495–510, September 2021. doi:10.1016/j.ejor.2020.12.033.

[6] Vishal Gaur and Marshall L. Fisher. A Periodic Inventory Routing Problem at a Supermarket Chain. *Operations Research*, 52(6):813–822, 2004. Publisher: INFORMS.

[7] Andreas Holzapfel, Alexander Hübner, Heinrich Kuhn, and Michael G. Sternbeck. Delivery pattern and transportation planning in grocery retailing. *European Journal of Operational Research*, 252(1):54–68, July 2016. doi: 10.1016/j.ejor.2015.12.036.

[8] Alexander H. Hübner, Heinrich Kuhn, and Michael G. Sternbeck. Demand and supply chain planning in grocery retail: an operations planning framework. *International Journal of Retail & Distribution Management*, 2013. Publisher: Emerald Group Publishing Limited.

[9] Jana Juros, Mario Brcic, Mihael Koncic, and Mihael Kovac. Exact solving scheduling problems accelerated by graph neural networks. In *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, pages 865–870, 2022. doi: 10.23919/MIPRO55190.2022.9803345.

[10] Heinrich Kuhn and Michael G. Sternbeck. Integrative retail logistics: An exploratory study. *Operations Management Research*, 6(1):2–18, June 2013. doi: 10.1007/s12063-012-0075-9.

[11] Hans Mittelman. MILP benchmarks.

[12] D Ronen and C A Goodhart. Tactical store delivery planning. *Journal of the Operational Research Society*, 59(8):1047–1054, August 2008. Publisher: Taylor & Francis _eprint: doi:10.1057/palgrave.jors.2602449.

[13] Stefan Ropke and David Pisinger. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, 40(4):455–472, November 2006. 10.1287/trsc.1050.0135.

[14] Michael G. Sternbeck and Heinrich Kuhn. An integrative approach to determine store delivery patterns in grocery retailing. *Transportation Research Part E: Logistics and Transportation Review*, 70:205–224, October 2014. doi: 10.1016/j.tre.2014.06.007.

[15] Karel H. van Donselaar, Vishal Gaur, Tom van Woensel, Rob A. C. M. Broekmeulen, and Jan C. Fransoo. Ordering Behavior in Retail Stores and Implications for Automated Replenishment. *Management Science*, 56(5):766–784, 2010. https://www.jstor.org/stable/40660793.