Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 180 (2021) 517–524

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Industry 4.0 and Smart Manufacturing

Driver Shift Planning for an Online Store with Short Delivery Times

Matthias Horn[a,*], Nikolaus Frohner[a], Günther R. Raidl[a]

[a] *Institute of Logic and Computation, TU Wien, Favoritenstrae 9–11/192–01, 1040 Vienna, Austria*
Email addresses: {*horn,nfrohner,raidl*}*@ac.tuwien.ac.at*

Abstract

In this work we derive daily driver shift plans for an online store which delivers goods to customers within short times. The goal is to minimize the total labor time (total shift lengths) over all shifts. Thereby orders must be assigned to shifts s.t. all orders are delivered in time. We model this optimization problem by means of a mixed integer linear program using a time-index based formulation. This model features strengthening inequalities that allow to solve it also reasonably well with an open source branch-and-cut solver. Furthermore we use a coarse-grained variant of the model to quickly derive high-quality heuristic solutions within one minute even for larger instances with up to two thousand orders. On a realistic benchmark instance set the overall approach is able to obtain solutions with remaining optimality gaps below 1%.

Keywords: Shift Planning; Mixed Integer Programming; Scheduling Problem

1. Introduction

We study a single-day shift scheduling problem of an online store which delivers goods to customers within short times of usually one or two hours. The store uses its own warehouse where all goods are stored and all orders are picked. To deliver the orders to the customers, vans are loaded at the warehouse with multiple orders per tour. The store offers different kind of orders s.t. the customer can decide if the orders should be delivered within the next two hours (henceforth denoted as two-hour orders) or with some extra charge within one hour (denoted as one-hour orders). Since the costs for delivery are one of the most significant factors in the store's cost structure, an efficient delivery process is crucial. The first step in this delivery process is to design a driver shift plan that ensures that enough personnel will be available to satisfy all orders while minimizing the total labor time. The second step, which is not part of this work, is to perform the actual clustering and routing of orders during the operation of the warehouse based on the designed shift plan.

* Corresponding author. Tel.: +43(1)58801192132.
E-mail address: horn@ac.tuwien.ac.at

10.1016/j.procs.2021.01.270

Since the vast majority of orders are ad hoc orders there is no certainty about the order volume at any time. The shift plan, however, has to be prepared some days in advance. Therefore it is necessary to anticipate the volume of incoming orders. The prediction of the order volumes is done by a supervised regression model based on historic data. For this work we assume that this order prediction is already done on a hourly basis. Thus we know for each day and each hour the volume of orders that are expected to be due. Furthermore we know for each hour the average time a van needs to delivery one order, allowing us to estimate the average number of orders that can be delivered by a single driver at a specific hour.

Based on this predicted data a daily driver shift plan is created. Drivers are typically students not fully employed but hired just for certain hours. We model this optimization problem as *mixed integer linear program* (MIP) that allows to fulfill all expected orders over a day in time. Hereby the total labor time, i.e., the sum of all shift lengths shall be minimized. To obtain the shift plan for more than one day, the MIP model is solved independently for each day. To reduce the computation time of the MIP solver we try to reduce the size of the MIP model be identifying and removing binary decision variables that can only take the value zero to get a feasible model. Furthermore we provide the MIP solver with a high-quality heuristic solution which is obtained by a coarse-grained variant of the MIP model. The considered approach is already deployed at an online store to create daily shift plans for drivers. We discuss related work in Section 2 and give a formal definition of the considered shift scheduling problem in Section 3. The MIP model will be presented in Section 4 and obtained results are discussed in Section 5. Finally, Section 6 concludes the work and indicates possible future research directions.

2. Related Work

The requirement to plan shifts s.t. different kinds of demands are satisfied arise in many areas such as transportation, logistics, health care, call centers, or production [3, 5, 11]. A related problem to this work is the *shift scheduling problem* which has been considered by several authors. The task is to compute shifts and assign a number of employees to each shift over a single day. Usually breaks need to be placed during a shift and the objective is mainly to minimize the number of assigned employees without any under-staffing. The shift scheduling problem has been tackled mostly by integer programming (IP) approaches: Dantzig [4] developed a set-covering formulation where for each shift a single integer variable is used. The characteristics of the shifts (starting time, length, breaks) are predefined and feasible shifts must be enumerated and added to the model. This has the disadvantage that the number of used integer variables can rapidly increase if, e.g., different allowable break options should be considered. Therefore, Bechtold and Jacobs [2] proposed an IP model where breaks are modeled implicitly, allowing to obtain superior results compared to the set-covering model. Thompson [10] combines the work of Bechtold and Jacobs [2] and Moondra [6] to introduce a formulation in which also the length of shifts and the breaks are modeled implicitly. In Aykin [1] a comparison of different modeling approaches can be found. Rekik et al. [8] proposed an even more flexible model which allows to consider e.g. fractional breaks.

Another (less) related problem is the *shift design problem* which differs from the shift scheduling problem by considering usually multiple consecutive days as planning horizon rather than one single day. In addition, the problem is usually cyclic, i.e., the end of the schedule is connected to its beginning. Moreover, under-staffing and over-staffing is typically possible and weighted in the objective function in shift design related problems. The shift design problem, originally introduced by Musliu et al. [7], has been considered by several authors and a wide range of methods are developed to tackle it including tabu search, simulated annealing, Benders decomposition based heuristics, and IP. For an overview we refer to Smet at al. [9].

This work schedule shifts for one single day s.t. the total shift lengths (labor time) over all shifts are minimized. We do not consider break placement during the shifts and we rather assign different orders to shifts that must be delivered by a single driver than assigning employees to shifts. The due date of the orders must always be satisfied.

3. Shift Planning

In our daily *driver shift plan* (DSP) problem we are given of predicted numbers of orders $\tilde{y}_{k,t} \geq 0$ for each category $k \in \{1, 2\}$ that are due within the hours $t = 0, \ldots, t_{\max}$. The category k corresponds to one-hour orders ($k = 1$) and two-hour orders ($k = 2$). Moreover, the predicted mean order delivery time $\phi_t \geq 0$ within the hours $t = 0, \ldots, t_{\max}$ are known. Orders from $\tilde{y}_{1,t}$ and $\tilde{y}_{2,t}$ can be due somewhere within the hour $[t, t+1)$. Therefore we create n equidistant time slots of length $1/n$ for each hour $t = 0, \ldots, t_{\max}$ and equally distribute the corresponding numbers of predicted orders $\tilde{y}_{1,t}$ and $\tilde{y}_{2,t}$ as well as the average number of orders $1/\tilde{\phi}_t$ that can be delivered by one single shift within hour t over those time slots.

More specifically: Let $\tau = \{t + i/n \mid t = 0, \ldots, t_{\max},\ i = 0, \ldots, n-1\}$ be the set of start times of each time slot and let $t_{n\max} = t + 1 - 1/n$ be the starting time of the last slot in τ. For each time slot $t \in \tau$ let $y_{1,t} = \tilde{y}_{1,\lfloor t \rfloor}/n$ be the number of orders which must be delivered within one hour and are supposed to be due in time slot t. In the same manner let $y_{2,t} = \tilde{y}_{2,\lfloor t \rfloor}/n$ be the number of orders which must be delivered within two hours and are due in time slot t. Note that $y_{k,t}$, $k \in \{1,2\}$, $t \in \tau$, represents here the expected number of orders for *each time slot* and not for each hour anymore. Let $\tau_{k,t} = \{t' \in \tau \mid t - k \leq t' \leq t\}$, $k \in \{1,2\}$, $t \in \tau$ be the set of all starting times of time slots where orders from $y_{k,t}$ may be delivered. Moreover, for each time slot $t \in \tau$ let $z_t = 1/(n\,\tilde{\phi}_{\lfloor t \rfloor})$ be the average number of orders which can be delivered in time slot t.

Let $U = \{1, \ldots, u_{\max}\}$ be the set of potentially available shifts (vans). The goal is to find a subset of shifts $U' \subseteq U$ and for each shift $u \in U'$ a start time q_u^{start} and an end time q_u^{end} such that all expected orders are feasibly assigned to a shift and the sum of the shift lengths is minimized. Since we assume that enhough cars are on standby, the number of available cars and thus used shifts is not restricted. However, lengths of used shifts must be at least $U_{\text{dur}}^{\min} = 4$ hours and at most $U_{\text{dur}}^{\max} = 7.5$ hours.

4. Mixed Integer Linear Programming Model

We formulate the problem as a MIP model with binary decision variables $s_{u,t}$, $x_{u,t}$, $u \in U$, $t \in \tau$, and continuous decision variables $a_{k,t,t'}$, $k \in \{1,2\}$, $t \in \tau$, $t' \in \tau_{k,t}$. Variable $a_{k,t,t'}$ indicates the amount of orders from $y_{k,t}$ delivered in time slot t'. The binary variable $s_{u,t}$ indicates if shift u starts at time t (=1) or not (=0), whereas binary variable $x_{u,t}$ indicates if shift u is active at time t (=1) or not (=0). Our DSP model is:

$$
\begin{aligned}
\min\quad & \tfrac{1}{n} \textstyle\sum_{u \in U} \sum_{t \in \tau} x_{u,t} && && \text{(1a)}\\
\text{s.t.}\quad & \textstyle\sum_{t \in \tau} s_{u,t} \leq 1 && u \in U && \text{(1b)}\\
& x_{u,t} \geq s_{u,t'} && u \in U, && \\
& && t, t' \in \tau \mid t \leq t' < t + U_{\text{dur}}^{\min} && \text{(1c)}\\
& x_{u,t} \leq x_{u,t-1/n} + s_{u,t} && u \in U,\ t \in \tau \setminus \{0\} && \text{(1d)}\\
& x_{u,0} \leq s_{u,0} && u \in U && \text{(1e)}\\
& \textstyle\sum_{t' \in \tau | t' \leq t - U_{\text{dur}}^{\max}} s_{u,t'} + x_{u,t} + \sum_{t' \in \tau | t' > t} s_{u,t'} \leq 1 && u \in U,\ t \in \tau && \text{(1f)}\\
& \textstyle\sum_{t' \in \tau_{k,t}} a_{k,t,t'} \geq y_{k,t} && k \in \{1,2\},\ t \in \tau && \text{(1g)}\\
& \textstyle\sum_{u \in U} x_{u,t} \geq \sum_{k \in \{1,2\}} \sum_{t' \in \tau | t \in \tau_{k,t'}} \frac{a_{k,t',t}}{z_t} && t \in \tau && \text{(1h)}\\
& s_{u,t} \leq \textstyle\sum_{t' \in \tau | t' \leq t} s_{u-1,t'} && u \in U \setminus \{1\},\ t \in \tau && \text{(1i)}\\
& x_{u-1,t} \leq x_{u,t} + 1 - \textstyle\sum_{t' \in \tau | t' \leq t} s_{u,t'} && u \in U \setminus \{1\},\ t \in \tau && \text{(1j)}\\
& s_{u,t} \in \{0,1\} && u \in U,\ t \in \tau && \text{(1k)}\\
& x_{u,t} \in \{0,1\} && u \in U,\ t \in \tau && \text{(1l)}\\
& a_{k,t,t'} \geq 0 && k \in \{1,2\},\ t \in \tau,\ t' \in \tau_{k,t} && \text{(1m)}
\end{aligned}
$$

The objective is to minimize the sum of active shift times (1a). Constraints (1b) ensure that each shift can only start at one time, whereas constraints (1c) imply that if a shift starts at time t then it must also be active for at least $U_{\text{dur}}^{\min}$ time units. Constraints (1d) and (1e) guarantee that a shift can only be active at time t if the shift is active also in the preceding time slot $t - 1/n$ or the shift starts at time t. Inequalities (1f) ensure that each active shift satisfies the maximum shift durations. Constraints (1g) ensure that all $y_{k,t}$ orders are delivered by time slot t, whereas constraints (1h) ensure that enough shifts are provided to deliver all orders assigned at time slot t. In order to break symmetries, Constraints (1i) force that the starting times of the shifts must be in non-decreasing order, i.e. a shift u can only start if the preceding shift $u-1$ starts no later than u. Constraint (1j) guarantee that a shift u cannot end later than the next shift $u+1$.

Having solved the MIP model, the actual set of used shifts U' comprises those with some variable $s_{u,t} = 1$, i.e., $U' = \{u \in U \mid \sum_{t \in \tau} s_{u,t} > 0\}$. For such a shift $u \in U'$, the start time is given by $q_u^{\text{start}} = \min\{t \in \tau \mid s_{u,t} = 1\}$ and the end time by $q_u^{\text{end}} = \max\{t + 1/n \mid t \in \tau \land x_{u,t} = 1\}$.

Reducing the size of model DSP. To reduce the runtime of the MIP solver it may be beneficial to reduce the number of binary decision variables $s_{u,t}$ and $x_{u,t}$ from the model DSP in a pre-processing step. In particular as we sort in our symmetry breaking the shifts according to non-decreasing starting times, there are often time slots in shifts that cannot be feasibly used. For example, if the last shift with index $u_{\max}$ starts at the earliest possible time slot, this implies that all other shifts also have to start at the earliest time slot. This will most likely not be a feasible solution, since orders that must be delivered in a period at the end of the time horizon can probably not be taken into account by any shift.

To identify such infeasible starting times we follow two strategies. The first strategy is to compute an upper bound $y_{u,t}^{\text{UB}}$ on the maximum number of orders which can be delivered if shift u starts at time slot t. If this upper bound is less than the total number of orders that must be delivered over the whole planning horizon then no feasible solution exists and the corresponding decision variable $s_{u,t}$ can be removed from the DSP model. The second strategy determines for each shift u a lower bound on the earliest time $s_u^{\text{ear}} \in \tau$ as well as an upper bound on the latest time $s_u^{\text{lat}} \in \tau$ at which shift u has to start such that a feasible solution may exist. Hence, all decision variables $s_{u,t}$ with $t < s_u^{\text{ear}}$ or $t > s_u^{\text{lat}}$ are removed from the DSP model.

More precisely: For the upper bound $y_{u,t}^{\text{UB}}$ consider

$$v_{t_1,t_2}^{\max} = \max_{t \in \tau \mid t_1 \le t \le t_2} \sum_{t' \in \tau \mid t \le t' < t + U_{\text{dur}}^{\max}} z_{t'} \tag{2}$$

as the maximum amount of orders that *can be delivered during a single shift* starting at some time slot from t_1 to t_2. Further consider

$$y_{t_1,t_2}^{\max} = \sum_{k \in \{1,2\}} \sum_{t \in \tau \mid \tau_{k,t} \cap [t_1, t_2 + U_{\text{dur}}^{\max}) \neq \emptyset} y_{k,t} \tag{3}$$

as the maximum amount of orders which *must be delivered* by shifts starting at time slot t_1 to t_2. Then an upper bound on the maximum number of orders that can be delivered if shift u starts at time slot t is

$$y_{u,t}^{\text{UB}} = \min\left\{(u-1)\, v_{0,t}^{\max}, y_{0,t}^{\max}\right\} + \min\left\{v_{t,t}^{\max}, y_{t,t}^{\max}\right\} + \min\left\{(u_{\max} - u)\, v_{t,t_{n\max}}^{\max}, y_{t,t_{n\max}}^{\max}\right\}. \tag{4}$$

The expression $(u-1)\, v_{0,t}^{\max}$ is an upper bound on the number of orders that can be delivered by shifts $1, \ldots, u-1$ in period $0, \ldots, t$. This upper bound is further capped by the maximum amount of orders that must be delivered within this period. In the same manner the second term refers to shift u, starting at time t and the third term refers to all remaining shifts starting within the period $t, \ldots, t_{n\max}$.

For the earliest and latest possible start times let $y_{t_1,t_2}^{\min} = \sum_{k \in \{1,2\}} \sum_{t \in \tau \mid \tau_{k,t} \subseteq [t_1,t_2]} y_{k,t}$ be the minimum amount of orders that must be delivered in time slots starting in $[t_1, t_2]$ to still obtain a feasible solution. Let $a_{t_1,t_2}^{\max} = \sum_{t \in \tau \mid t_1 \le t \le t_2} z_t$ be the maximum amount of orders that can be delivered by one single shift starting within period $[t_1, t_2]$. Then for each $t \in \tau$, $\lceil y_{t-U_{\text{dur}}^{\max}+1,t}^{\min} / a_{t-U_{\text{dur}}^{\max}+1,t}^{\max} \rceil$ is the minimum number of shifts that must start at some time $t' \in \tau \mid t - U_{\text{dur}}^{\max} < t' \le t$ to deliver all orders. The bound can be further strengthened by considering the maximum over all t', thus let the minimum number of shifts that must start in $(t - U_{\text{dur}}^{\max}, t]$ be $u_t^{\text{lat}} = \max_{t' \in \tau \mid t - U_{\text{dur}}^{\max} < t' \le t} \left\lceil y_{t',t}^{\min} / a_{t',t}^{\max} \right\rceil$. In the same manner we can derive the minimum number of shifts $u_t^{\text{ear}} = \max_{t' \in \tau \mid t \le t' < t + U_{\text{dur}}^{\max}} \left\lceil y_{t',t}^{\min} / a_{t',t}^{\max} \right\rceil$ that must start no earlier than $t - U_{\text{dur}}^{\max} + 1$ to serve all orders in time slots starting in $[t, t + U_{\text{dur}}^{\max})$.

To derive the latest start times, initialize $s_u^{\text{lat}} := t_{n\max}$ for all shifts $u \in U$ and let $\hat{u} := 0$ be the minimum number of shifts needed so far. Then consider each time slot $t \in \tau$ in increasing order. Let $n_t = |\{u' \in U \mid u' \le \hat{u} \land t < s_u^{\text{lat}} + U_{\text{dur}}^{\max}\}|$ be the number of currently opened shifts that may serve orders within the currently considered period $(t - U_{\text{dur}}^{\max}, t]$. If $u_t^{\text{lat}} > n_t$ then the currently open shifts cannot serve all orders and we have to open at least $u_t^{\text{lat}} - n_t$ new shifts that must start no later than time t to serve all remaining orders within period $(t - U_{\text{dur}}^{\max}, t]$. Hence, we set the latest start times of the new shifts to $s_{\hat{u}+i}^{\text{lat}} := \min(s_{\hat{u}+i}^{\text{lat}}, t)$, $i = 1, \ldots, u_t^{\text{lat}} - n_t$, and increase the number of currently open shifts to

$\hat{u} := \hat{u} + u_t^{\text{lat}} - n_t$. In this way we determine for each shift the latest possible start time such that still all orders can be delivered within the due times. In the same manner we can derive the earliest start times s_u^{ear} for all shifts $u \in U$ by considering each time slot $t \in \tau$ in decreasing order and starting with the last shift with index $u_{\max}$.

We can now remove all binary decision variables $s_{u,t}$, $u \in U$, $t \in \tau$, from the DSP model for which $y_{u,t}^{\text{UB}} \leq \sum_{k \in \{1,2\}} \sum_{t' \in \tau} y_{k,t'}$ or $t < s_u^{\text{ear}}$ or $t > s_u^{\text{lat}}$ or $t > t_{n\max} - U_{\text{dur}}^{\min} + 1$ holds. Proceeding from the reduced $s_{u,t}$ variables, for each shift $u \in U$, let t_1 be the earliest time slot for which a variable s_{u,t_1} still exists and let t_2 be the latest time slot for which a variable s_{u,t_2} still exists. Remove all $x_{u,t}$, $t \in \tau$ variables for which $t < t_1$ or $t \geq t_2 + U_{\text{dur}}^{\max}$ holds.

Further strengthening. The model DSP can be strengthened by taking the lower bound on the minimum number of shifts $u_{\min} = \left\lceil y_{0,t_{n\max}}^{\max} / v_{0,t_{n\max}}^{\max} \right\rceil$ needed to serve all orders from $y_{k,t}$, $k \in \{1, 2\}$, $t \in \tau$, into account. It follows that the first $u_{\min}$ shifts from U must be active, since a shift with index $u \in U$ can only start when all other shifts with a lower index start before or at the same time and there are at least $u_{\min}$ shifts required to obtain a feasible solution. Hence, constraints (1b) can be replaced by

$$\sum_{t \in \tau} s_{u,t} = 1 \qquad u \in U \mid u \leq u_{\min}, \tag{5}$$

$$\sum_{t \in \tau} s_{u,t} \leq 1 \qquad u \in U \mid u > u_{\min}, \tag{6}$$

respectively. For similar reasons we can replace constraints (1i) by

$$s_{u,t} \leq \sum_{t' \in \tau | t' \leq t} s_{u-1,t'} \qquad u \in U \setminus \{1\},\ t \in \tau \mid u - 1 > u_{\min} \vee s_{u-1}^{\text{lat}} \geq t \tag{7}$$

since the sum $\sum_{t' \in \tau | t' \leq t} s_{u-1,t'}$ is always one if shift $u - 1 \leq u_{\min}$ cannot start later than $s_{u-1}^{\text{lat}} < t$. In the same manner constraints (1j) can be replaced by

$$x_{u-1,t} \leq x_{u,t} \qquad u \in U \setminus \{1\},\ t \in \tau \mid u \leq u_{\min} \wedge s_u^{\text{lat}} < t \tag{8}$$

$$x_{u-1,t} \leq x_{u,t} + 1 - \sum_{t' \in \tau | t' \leq t} s_{u,t'} \qquad u \in U \setminus \{1\},\ t \in \tau \mid u > u_{\min} \vee s_u^{\text{lat}} \geq t. \tag{9}$$

4.1. Initial Heuristic Solution

The advantage of creating an initial heuristic solution to the shift planning problem is twofold. First, the maximum number of allowed shifts $u_{\max}$ in the MIP model from Section 4 is estimated by the number of shifts obtained by the heuristic solution. Second, providing the MIP solver the initial solution may shorten the computation time of the solver.

The former is achieved by creating a heuristic solution by a greedy construction algorithm. However, preliminary experiments showed that the quality of this solution is usually not sufficient to shorten the computation time of the MIP solver when providing it as initial solution. To obtain a better solution, we therefore use a coarse-grained variant of the DSP model, denoted as CDSP, where the start times of each $u \in U$ are restricted to some individual subset $\tau_u \subseteq \tau$. Time slots are again defined to last from the start time until the start time of the next time slot, and can thus be a multiple of $1/n$. In this way the model uses less binary decision variables $s_{u,t}$, $x_{u,t}$, $u \in U$, $t \in \tau_u$ and it is more likely that the MIP solver will quickly find a feasible solution. Since the starting times are more restricted these solutions are clearly heuristic ones and their quality depends strongly on the chosen subsets of restricted start times, which are derived from the heuristic solution obtained by the greedy construction algorithm.

In the following the greedy construction algorithm is described. Afterwards the selection of the restricted start times as well as the coarse-grained model are explained in more detail.

Greedy construction algorithm. The heuristic assigns orders from $y_{k,t}$ to shifts in increasing due-time order t and decreasing order of k. As long as $y_{k,t} > 0$, all possible free time slots for all currently open shifts are considered. Each order is assigned to a time slot/shift that causes the smallest increase of the objective value. Ties are resolved by selecting always the shift with the smallest starting time and the earliest slot. The selected slot is filled up with orders until the slot is full w.r.t. the average number of orders that can be delivered during this slot. Variable $y_{k,t}$ is decreased by the corresponding amount of orders assigned to the selected slot. If there is no free slot where orders can be placed then a new shift is opened.

Coarse-grained MIP model. Assume that a heuristic solution is obtained by the greedy construction algorithm with start times q_u^{start} and end times q_u^{end} for each shift u. Then for each shift u a restricted set of starting times of time slots is derived by

$$\tau_u = \{t \mid t = 0, \dots, t_{\max}\} \cup \{q_u^{\text{start}} + i/n, q_u^{\text{end}} + i/n \mid i = -h, \dots, h\}. \tag{10}$$

The first term adds a slot starting at every full hour. This is reasonable since the predicted mean order delivery time changes hourly. To ensure that at least the solution obtained by the greedy construction heuristic is feasible w.r.t. the CDSP model, the second term divides the time intervals further such that the start time q_u^{start} and the end time q_u^{end} of shift u in the heuristic solution are also included. To give the solver further freedom, also the h nearest slots of length $1/n$ around q_u^{start} and q_u^{end} are added. The CDSP model essentially corresponds to the DSP model with the exception of the extension to the individual subsets of starting times τ_u for the shifts. Moreover, the CDSP model does not enforce that a feasible solution must have shifts with non-decreasing start and end times in order to not restrict the solution space further. However, a corresponding solution with non-decreasing start and end times can always be obtained by reordering/reshaping the shifts accordingly.

The solution obtained by the CDSP model is then provided to the MIP solver as initial solution to solve the full DSP model with the fined-grained planning horizon τ.

5. Results

To test the DSP model we created randomly eight different instance sets with different characteristics in terms of in-homogeneity of the mean order delivery times as well as different total amounts of orders to deliver. In practice the online store usually experiences a peak of orders at the end of the day when customers go home from work. Therefore we use function $f(t) = (t - t_{\max})/25\, e^{(t_{\max}-t)/50}$ with a planning horizon of $t_{\max} = 16$ hours to simulate such a peak and overlay $f(t)$ with some random noise modeled by a normal distribution $\mathcal{N}(0, 0.01)$ with mean value zero and a standard deviation of 0.01. To create a new instance we sample $\tilde{y}_{1,t}$ and $\tilde{y}_{2,t}$ from the distribution $\rho/2 \max(0, f(t) + \mathcal{N}(0, 0.01))$ for each hour t. Variable ρ approximately indicates the total amount of orders that should be delivered. Furthermore for each hour we sample the average number or orders that can be delivered within this hour from a normal distribution $\mathcal{N}(3.5, \sigma)$ with mean value 3.5 and different values for the standard deviation σ. In this way eight instance sets $A_{\rho,\sigma}$ were created with $\rho \in \{200, 500, 1000, 2000\}$ and $\sigma \in \{0.5, 1.0\}$. For each instance set, 30 instances were sampled such that in total 240 instances were obtained.

The DSP as well as the CDSP model are solved with the open source COIN-OR Branch-&-Cut (CBC) solver, version 2.10, available at `https://github.com/coin-or/Cbc`. The experiments were conducted on a computing cluster of 16 machines, each with two *Intel Xeon E5-2640 v4* CPUs with 2.40 GHz in single threaded mode and 15 GB RAM. For all considered experiments the number of slots per hour was set to $n = 4$. Consequently, a single time slot has a duration of 15 minutes.

Concerning the reduction of binary decision variables we were able to remove on average 41.72% of the binary decision variables from the MIP models. More specifically, 58.86% of the $s_{u,t}$ variables and 24.58% of the $x_{u,t}$ variables could on average removed.

Next we investigate different choices for h, the number of slots which are additionally added in the CDSP model. For each instance, a heuristic solution is created by the greedy construction heuristic described in Section 4.1, and this solution is used to derive the specific time slot starting times τ_u of each potential shift u as described in the previous Section. The heuristic solution obtained by the CDSP model is then provided to the MIP solver in order to ultimately solve the DSP model. Figure 1 reports the obtained average optimality gaps from the CDSP as well as from the DSP model for each instances class. The gaps are related to the lower bound obtained by the DSP model. The total time limit was set to 10 CPU minutes. At most one CPU minute was spend to solve the CDSP model and the rest of the time to solve the DSP model. For all considered instance classes the solutions obtained by the CDSP model exhibit an average optimality gap below 0.5% for $h = 0$. Since the solver has more freedom with increasing h it is not surprising that in general the obtained solution quality is increasing with increasing h. Final solutions obtained by the DSP model have average optimality gaps below 0.1% in case of the instance sets with a standard deviation of $\sigma = 0.5$. For instance sets with $\sigma = 1.0$, we obtained gaps below 0.2%.

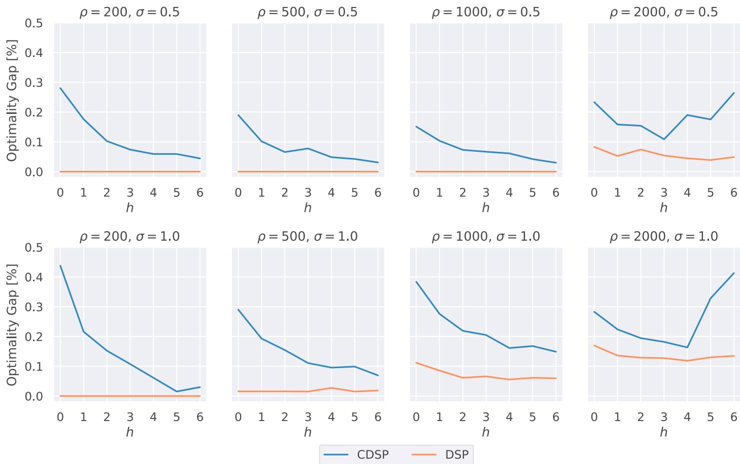

Fig. 1. Average optimality gaps obtained by the CDSP and DSP models with different values of h. The gaps are related to the lower bound obtained by solving the DSP model.

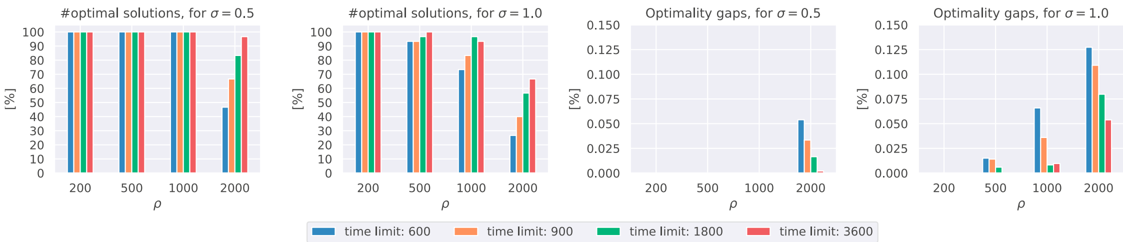

Fig. 2. Optimality gaps and number of instances solved to proven optimality for different time limits with $h = 3$.

Figure 2 shows the average number of instances solved to proven optimality by solving the DSP model as well as the obtained average optimality gaps for different time limits set to 600, 900, 1800, and 3600 seconds. For each instance the CDSP model with $h = 3$ was solved with a time limit of 60 seconds to obtain the respective initial solution for the DSP model. As one may expect, with a higher time limit more instances can be solved to proven optimality and smaller optimality gaps are obtained, but in general the performance of CBC also decreases with increasing instance size. For smallest instances with $\rho = 200$ all instances can be solved to proven optimality. However, for larger instances with $\rho = 2000$ and $\sigma = 1.0$ only 60% can be solved to proven optimality. Moreover, it seems that instances where the mean delivery time has a higher variance are harder to solve.

Table 1 reports the main results obtained from the DSP model with an overall time limit of 600 seconds. The CBC solver was provided with a heuristic solution obtained from the CDSP model with a time limit of 60 seconds and $h = 3$. Column opt lists the percentage of instances that could be solved to proven optimality whereas columns gap DSP and gap CDSP report the averaged remaining optimality gaps. In case of CDSP the obtained gaps are related to the lower bound obtained by the DSP model. The numbers in brackets indicate the corresponding standard deviation. The median computation times are reported by column time and the average reduction rates of the binary decision variables are shown by column red. After 600 seconds the approach has solved on average all considered instances with a remaining optimality gap below 0.13.

6. Conclusion

In this work we considered a daily driver shift scheduling problem for an online store with short delivery times of one and two hours. The goal was to generate a shift plan s.t. all expected orders are delivered before their due dates and the total labor time is minimized. The problem was solved by a MIP based on a time indexed formulation. The

Table 1. Main results obtained with a time limit of 600 seconds and $h = 3$.

σ	ρ	opt [%]	gap DSP [%]	gap CDSP [%]	time [s]	red [%]
0.5	200	100.0	0.00 (0.00)	0.07 (0.17)	8	45.39
	500	100.0	0.00 (0.00)	0.08 (0.12)	38	41.97
	1000	100.0	0.00 (0.00)	0.07 (0.10)	149	40.61
	2000	46.7	0.05 (0.06)	0.11 (0.08)	619	39.87
1.0	200	100.0	0.00 (0.00)	0.11 (0.20)	15	44.70
	500	93.3	0.02 (0.06)	0.11 (0.18)	108	41.45
	1000	73.3	0.07 (0.12)	0.20 (0.15)	346	40.18
	2000	26.7	0.13 (0.11)	0.18 (0.12)	635	39.62

computation time of the MIP solver was reduced by removing binary decision variables from the model that can only take the value zero to get a feasible model. Furthermore a coarse-grained variant of the model was utilized to quickly obtain a high-quality heuristic solution which is then provided to the MIP solver as initial solution. To conclude, although not all benchmark instances could be solved to proven optimality, very high quality solutions with small remaining optimality gaps are obtained in relatively short time.

Next steps are to incorporate more constraints that are possible relevant in practice. It may be, for instance, important to restrict the number of shifts that start at the same time or to restrict the number of shifts that are active at the same time. Moreover, it may be promising to pursue the coarse-grained approach further. Until now we derive quickly heuristic solutions by considering a coarse-grained time horizon. In a similar manner it should be possible to derive also a feasible lower bound on the total labor time from a coarse-grained time horizon. Alternatively, an iterative approach may be devised where time points are iteratively added to the time horizon until the optimality of a feasible solution can be proven. For future work it may also be interesting to compare the CBC solver with commercial solvers like Gurobi or Cplex.

Acknowledgements

This project is partially funded by the Doctoral Program "Vienna Graduate School on Computational Optimization", Austrian Science Foundation (FWF) Project No. W1260-N35.

References

[1] T. Aykin. A comparative evaluation of modeling approaches to the labor shift scheduling problem. *European Journal of Operational Research*, 125(2):381–397, 2000.
[2] S. E. Bechtold and L. W. Jacobs. Implicit modeling of flexible break assignments in optimal shift scheduling. *Management Science*, 36(11):1339–1351, 1990.
[3] E. K. Burke, P. De Causmaecker, G. V. Berghe, and H. Van Landeghem. The state of the art of nurse rostering. *Journal of Scheduling*, 7(6):441–499, 2004.
[4] G. B. Dantzig. A comment on Edie's "traffic delays at toll booths". *Journal of the Operations Research Society of America*, 2(3):339–341, 1954.
[5] P. De Bruecker, J. Van den Bergh, J. Belin, and E. Demeulemeester. Workforce planning incorporating skills: State of the art. *European Journal of Operational Research*, 243(1):1–16, 2015.
[6] L. Moondra. An LP model for work force scheduling for banks. *Journal of Bank Research*, 7:299–301, 1976.
[7] N. Musliu, A. Schaerf, and W. Slany. Local search for shift design. *European Journal of Operational Research*, 153(1):51–64, 2004.
[8] M. Rekik, J.-F. Cordeau, and F. Soumis. Implicit shift scheduling with multiple breaks and work stretch duration restrictions. *Journal of Scheduling*, 13(1):49–75.
[9] P. Smet, A. Lejon, and G. Vanden Berghe. Demand smoothing in shift design. *Flexible Services and Manufacturing Journal*, 2020.
[10] G. M. Thompson. Improved implicit optimal modeling of the labor shift scheduling problem. *Management Science*, 41(4):595–607, 1995.
[11] J. Van den Bergh, J. Belin, P. De Bruecker, E. Demeulemeester, and L. De Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385, 2013.