Tour Scheduling in Attended Home Delivery

Tom Perroux, Fabien Lehuédé, Maria-Isabel Restrepo

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 55-10 (2022) 245–250

Tour Scheduling in Attended Home Delivery

Tom Perroux * **Fabien Lehuédé** * **María I. Restrepo** *

* *IMT Atlantique, Laboratoire des Sciences du Numérique de Nantes (LS2N, UMR CNRS 6004), Team Modelis, Nantes, France (e-mails: tom.perroux@gmail.com, fabien.lehuede@imt-atlantique.fr, maria-isabel.restrepo-ruiz@imt-atlantique.fr).*

Abstract: In this paper we study a tour-scheduling problem for an attended home delivery problem with uncertain order requests. The problem is modelled as a two-stage stochastic programming problem and solved using the multi-cut L-shaped method. Weekly working patterns are generated by means of a pricing heuristic. Numerical results on randomly generated instances show that including weekly working rules increases the total cost by only a small amount (up to 2.54%) when compared to an approach that only considers daily working rules for the generation of driver schedules.

Keywords: Personnel scheduling, Tour scheduling problem, Attended home delivery, L-shaped method

1. INTRODUCTION

Home service (HS) operations are becoming increasingly important due to social, demographic, and epidemiological trends in most countries. Attended home delivery (AHD) is an example of this type HS operations. In this problems, employees travel to perform work-related activities at geographically dispersed customer locations. For instance, drivers, deliver packages at customers' homes. As reported in recent social and demographic studies, several service operations, which require customers to travel and to visit a place indoor, are switching to become services provided at home. In fact, as from 2014 to 2019 e-commerce sales ratios nearly tripled globally (WEF, 2020). This accelerated growth creates important challenges and opportunities for AHD providers, and requires the development of decision support tools to optimise their operations and to organise them in a more sustainable way.

The most prevalent home service operation planning problems in real-life are *multi-period* (i.e., the planning horizon spans multiple days) because a demand for continuous operations have to be satisfied or because customers might request multiple services spread over different days. This important characteristic makes necessary the integration of different decision levels such as tactical employee scheduling (e.g., tour scheduling) and operational employee routing while optimising HS operations.

Many variants of employee scheduling and routing for home service operations have been studied for decades. However, the integration of both problems have only started to get the attention of the operations research and artificial intelligence communities. The majority of the works in the literature focuses on *single-period* optimisation problems (i.e. the planning horizon corresponds to one working day). A good deal of this literature considers the incorporation of working regulations an important factor. However, to the best of our knowledge, only few works have addressed multi-period personnel scheduling and routing problems including uncertainty (in order requests, crowd-source capacities,...) and working regulations. Examples include the studies in (Steeg and Schröder, 2008; Trautsamwieser and Hirsch, 2014; Wirnitzer et al., 2016; Cappanera et al., 2018; Restrepo, M. I. et al., 2020) for home healthcare, and the studies in (Zäpfel and Bögl, 2008; Goel and Irnich, 2017) for urban delivery and long-distance haulage applications, respectively. These works propose different approaches to generate multi-period plans including scheduling and routing decisions simultaneously. However, with the exception of the studies in (Zäpfel and Bögl, 2008; Trautsamwieser and Hirsch, 2014; Restrepo, M. I. et al., 2020), they fail to incorporate working regulations related to the allocation of days-off and rest times between consecutive shifts. In addition, as these problems are already difficult to solve, they do not consider uncertainty in the demands and/or supply capacities, except for (Cappanera et al., 2018; Restrepo, M. I. et al., 2020).

In this work, we study an integrated tour scheduling and order allocation problem problem for AHD. The problem is defined over a time horizon of one week, where days are divided into multiple time periods, and the majority of order requests are uncertain and must be fulfilled within a narrow time window. Customer orders can be delivered by company drivers or by external drivers (crowd-sourcing). Hence the focus of our research is to decide weekly tours for company drivers (working days, days-off, and start time and duration of shifts for each working day) and to assign customer orders to the drivers so as to fullfill all customer orders at minimum cost.

10.1016/j.ifacol.2022.09.396

2. PROBLEM DESCRIPTION AND SOLUTION APPROACH

We study the case of a last-mile delivery company that proposes two types of services to satisfy customers demands. The problem is closely related to the tactical problem addressed in Restrepo et al. (2019). Deliveries are composed by a number of packages, a time window, a pickup location and a delivery location. To receive their orders, customers can choose between one of two services. The first type of service corresponds to pre-ordered deliveries, called *appointment deliveries*, where the client can choose between some predefined time slots. The second type corresponds to *express deliveries* where the client must be served within a two-hour time window after having placed their order. To satisfy customer requests, the company can hire company drivers and external drivers. Company drivers are assigned to working shifts, and they are paid a fixed amount of money per hour worked, plus a variable amount of money depending on the number of deliveries made during the day. External drivers are more expensive than company drivers. However, they add flexibility and help to keep service levels when client requests cannot be satisfied by company drivers. We suppose that the company can hire an infinite amount of external drivers and can thus always satisfy the demand.

We consider a fleet of C company drivers to schedule over a planning horizon of D days and I time periods on each day. The set of pre-defined shifts for company drivers is denoted by $\mathcal{S}$. Shifts are composed by consecutive working hours and must respect a minimum min_S and maximum max_S number of working time periods over the day. A shift is said to be compatible with a driver if the driver is available during the entire duration of the shift. We denote by $\mathcal{S}^c$ the set of shifts compatible with driver c ($\mathcal{S}^c \subseteq \mathcal{S}$). Set $\mathcal{T}$ define the possible weekly working patterns. A pattern is a sequence containing shifts and days-off. Each pattern must respect some working rules: a minimum $min_{\mathcal{T}}$ and a maximum $max_{\mathcal{T}}$ number of working hours over the week and a minimum $rest_{\mathcal{T}}$ number of days-off. A pattern is said to be compatible with a driver c if every shift $s \in \mathcal{S}^c$ in the sequence is compatible with driver c. We denote by $\mathcal{T}^c \subseteq \mathcal{T}$ the set of patterns compatible with driver c.

Since the problem we are addressing in this paper is tactical by nature, we do not consider an explicit representation of each customer location. Instead, we define a set of areas $\mathcal{A} = \{1..A\}$, such that all pickup and delivery locations in the same area $a \in \mathcal{A}$ are considered at the same location. We then define the set of origin-destination pairs (o-d pairs) $\mathcal{P}$ as the product $\mathcal{P} = \mathcal{A} \times \mathcal{A}$. We assume that a driver can visit only one o-d pair per time period and that he can always travel from the origin to the destination within each time period. Drivers can deliver several packages (limited by the capacity of their vehicle) in the same o-d pair at each time period.

Since not all customer requests are known in advance, we use a scenario-based approach to represent the uncertainty in the demand. Let Ω_i^d be the set of scenarios for the demand on day d and time period i. Each scenario ω in Ω_i^d is assigned a probability p_{di}^{ω} such that $\sum_{\omega \in \Omega_i^d} p_{di}^{\omega}=1$. The demand in a particular scenario is composed of a deterministic part plus a stochastic part. The demand is distributed among all o-d pairs following a probability distribution. The deterministic component of the demand (client requests known in advance) in a particular o-d pair p is denoted as g_{dip} and the stochastic demand as h_{dip}^{ω}. The total demand in o-d pair p under scenario ω is then equal to $b_{dip}^{\omega}= g_{dip} + h_{dip}^{\omega}$ with a probability p_{di}^{ω}.

We consider a fixed cost f_t^c associated to each pattern $t \in \mathcal{T}^c$ for driver c. This cost depends on a vehicle wear cost and a staffing cost per working time period. Deliveries made by company drivers have a variable cost l_{dip}^c that depends on the distance between o-d pairs and the type of vehicle used. Deliveries performed by external drivers have a cost c_{dip} that depends on the location of the destination. For instance, if the destination is far from the city center, the delivery is more expensive.

2.1 A two-stage stochastic approach

Let x_c^t be a binary variable assuming value 1 if weekly pattern t is assigned to company driver c, and 0 otherwise. Let y_{dip}^c be a binary variable taking value 1 if company driver c is allocated to o-d pair p during day d and time period i, it assumes value 0 otherwise. Decision variables $v_{dip}^{c\omega}$ and e_{dip}^{ω} denote the number of packages delivered by company drivers and external drivers respectively. Let δ_{dit}^c be a binary parameter taking value 1 if period $i \in I$ of day $d \in D$ is a working period for driver $c \in C$ in pattern $t \in \mathcal{T}^c$. It assumes value 0 otherwise. Let β^c the capacity in number of packages of driver's $c \in C$ vehicle, and let $\mu_{dip}^{c\omega} = min\{b_{dip}^{\omega}, \beta^c\}$ be the maximum number of packages that driver $c \in C$ can effectively deliver in o-d pair $p \in P$ on day $d \in D$ at time period $i \in I$ under scenario $\omega \in \Omega_i^d$. The deterministic equivalent program of the tour scheduling problem for attended home delivery is as follows:

$$\min \sum_{c \in C} \sum_{t \in \mathcal{T}^c} f_t^c x_t^c + \sum_{d \in D} \sum_{i \in I} \sum_{\omega \in \Omega_i^d} p_{di}^{\omega} \left(\sum_{c \in C} \sum_{p \in P} l_{dip}^c v_{dip}^{c\omega} + \sum_{p \in P} c_{dip} e_{dip}^{\omega} \right) \tag{1}$$

$$\sum_{t \in \mathcal{T}^c} x_t^c \leq 1, \; \forall\, c \in C \tag{2}$$

$$\sum_{p \in P} y_{dip}^c = \sum_{t \in \mathcal{T}^c} \delta_{dit}^c x_t^c, \; \forall\, c \in C,\, d \in D,\, i \in I \tag{3}$$

$$v_{dip}^{c\omega} \leq \mu_{dip}^{c\omega} y_{dip}^c, \quad \forall\, c \in C,\, d \in D,\, i \in I,\, \omega \in \Omega_i^d,\, p \in P \tag{4}$$

$$\sum_{c \in C} v_{dip}^{c\omega} + e_{dip}^{\omega} = b_{dip}^{\omega}, \quad \forall\, d \in D, i \in I, \omega \in \Omega_i^d, p \in P \tag{5}$$

$$x_t^c \in \{0,1\}, \; \forall\, c \in C,\, t \in \mathcal{T}^c \tag{6}$$

$$y_{dip}^c \in \{0,1\}, \; \forall\, c \in C,\, d \in D,\, i \in I,\, p \in P \tag{7}$$

$$v_{dip}^{c\omega} \in \mathbb{Z}_{\geq 0}, \forall\, c \in C, d \in D, i \in I, \omega \in \Omega_i^d, p \in P \tag{8}$$

$$e_{dip}^{\omega} \geq 0, \; \forall\, d \in D,\, i \in I,\, \omega \in \Omega_i^d,\, p \in P. \tag{9}$$

Constraint (2) limits the allocation of patterns to maximum one per company driver. Constraint (3) enforces that if driver $c \in C$ is working in pattern $t \in \mathcal{T}^c$ and period $i \in I$ of day $d \in D$, then driver c is assigned to an o-d pair. Constraint (4) limits the number of packages that driver $c \in C$ can deliver in o-d pair $p \in P$ at each time period. Constraint (5) ensures that the demand is satisfied either by company drivers or by external drivers. Note that there is no upper bound on the value of variables e^{ω}_{dip}. Therefore, problem (1)-(9) is always feasible. Constraints (6)-(9) are the domain constraints.

The number of constraints and variables in problem (1)-(9) grows with the number of drivers, days, patterns, time periods, o-d pairs, and scenarios. Thus the problem can become prohibitively large, especially when the number of scenarios is important. Therefore, we use a multi-cut L-shaped method (Birge and Louveaux, 1988) as a solution approach. Let M be the first-stage problem and $R(x, y)$ the second-stage problem (recourse problem). $R(x, y)$ can be decomposed into multiple sub-problems $R^{\omega}_{dip}(x, y)$ for each $d \in D$, $i \in I$, $\omega \in \Omega^d_i$, $p \in P$. The formulation of the second-stage problem, denoted as R^{ω}_{dip}, is as follows:

$$\mathcal{Q}(x, y_{dip}, \xi^{\omega}_{dip}) = \min \sum_{c \in C} l^c_{dip} v^{c\omega}_{dip} + c_{dip} e^{\omega}_{dip} \tag{10}$$

$$v^{c\omega}_{dip} \leq \mu^{c\omega}_{dip} y^c_{dip}, \; \forall\, c \in C \tag{11}$$

$$\sum_{c \in C} v^{c\omega}_{dip} + e^{\omega}_{dip} = b^{\omega}_{dip} \tag{12}$$

$$v^{c\omega}_{dip} \geq 0, \; \forall\, c \in C \tag{13}$$

$$e^{\omega}_{dip} \geq 0. \tag{14}$$

Note that integrality constraints on $v^{c\omega}_{dip}$ variables have been relaxed as problem (10)-(14) corresponds to a sorting problem and thus its solution is integer.

We define θ^{ω}_{dip} variables and optimality cuts (18) to approximate the objective function of recourse problems $R^{\omega}_{dip}(x, y)$. The dual multipliers from constraints (11) and (12) are denoted by $\pi^{c\omega}_{dip}$ and λ^{ω}_{dip}, respectively. Note that there is no need to add feasibility cuts since recourse problems are always feasible and have fix recourse matrix. The formulation of the first-stage problem is as follows:

$$\min \sum_{c \in C} \sum_{t \in \mathcal{T}^c} f^c_t x^c_t + \sum_{d \in D} \sum_{i \in I} \sum_{\omega \in \Omega^d_i} \sum_{p \in P} \theta^{\omega}_{dip} \tag{15}$$

$$\sum_{t \in \mathcal{T}^c} x^c_t \leq 1, \; \forall\, c \in C \tag{16}$$

$$\sum_{p \in P} y^c_{dip} = \sum_{t \in \mathcal{T}^c} \delta^c_{dit} x^c_t, \; \forall c \in C, d \in D, i \in I \tag{17}$$

$$\theta^{\omega}_{dip} \geq p^{\omega}_{di} \left(\sum_{c \in C} \mu^{c\omega}_{dip} \pi^{c\omega}_{dip} y^c_{dip} + b^{\omega}_{dip} \lambda^{\omega}_{dip} \right), \quad \forall\, d \in D, \; i \in I, \; \omega \in \Omega^d_i, \; p \in P \tag{18}$$

$$x^c_t \in \{0, 1\}, \; \forall\, c \in C, \; t \in \mathcal{T}^c \tag{19}$$

$$y^c_{dip} \in \{0, 1\}, \; \forall\, c \in C, \; d \in D, \; i \in I, \; p \in P \tag{20}$$

$$\theta^{\omega}_{dip} \geq 0, \forall\, d \in D, \; i \in I, \; \omega \in \Omega^d_i, \; p \in P. \tag{21}$$

2.2 Weekly pattern generation

An important limitation to the efficiency of the resolution of problem (15)-(21) is related to the number of patterns $|\mathcal{T}|$ to generate. An upper bound to this number can be obtained with $|\mathcal{S}|^{|D|}$ which becomes prohibitively large as the number of days and shifts increase. The exhaustive enumeration of patterns would, in most situations, result in an intractable problem. Successful methods facing this kind of limitations often come with a Branch & Price strategy. Since we are solving the problem with a Branch & Cut method, implementing a good Branch & Price could be tedious and is thus out of the scope of this study. Hence, we propose a heuristic method (inspired by a column generation approach) to generate a reasonable number of patterns before problem (15)-(21) is solved with the multi-cut L-shaped method. Its procedure follows.

We denote by $\mathcal{T}' \subset \mathcal{T}$ the subset of patterns generated so far by the heuristic method, starting with a non empty set of initial patterns. Variable x^c_t, $\forall c \in C, t \in \mathcal{T}'$ represents the allocation of patterns to company drivers. Variables z^+_{di}, z^-_{di} represent respectively under-staffing and over-staffing at day d and time period i. The notation and mathematical model of the master problem for pattern-generation follows:

Table 1. Parameters for the pattern generation master problem.

f^c :	cost per working period for company driver c
c^+, c^- :	over-staffing and under-staffing costs, respectively
ρ^c_{dit}:	1 if i is a working period for driver c on day d, 0 otherwise
w_t:	number of working periods in pattern t
r_{di}:	average number of drivers required on day d at time period i ($r_{di} = \sum_{p \in P} \sum_{\omega \in \Omega} p^{\omega}_{di} b^{\omega}_{dip} / \bar{\beta}$). $\bar{\beta}$ is the mean capacity over all company drivers.

$$\min \sum_{c \in C} \sum_{t \in \mathcal{T}'} f^c w_t x^c_t + \sum_{d \in D} \sum_{i \in I} (c^+ z^+_{di} + c^- z^-_{di}) \tag{22}$$

$$\sum_{c \in C} \sum_{t \in \mathcal{T}'} \rho^c_{dit} x^c_t - z^+_{di} + z^-_{di} = r_{di}, \; \forall d \in D, i \in I \tag{23}$$

$$\sum_{t \in \mathcal{T}'} x^c_t \leq 1, \; \forall\, c \in C \tag{24}$$

$$x^c_t \geq 0, \; \forall\, c \in C, \; t \in \mathcal{T}' \tag{25}$$

$$z^+_{di}, \; z^-_{di} \geq 0 \; \forall d \in D, \; i \in I. \tag{26}$$

Constraints (23) ensure that the total number of drivers working at each time period is equal to the average number of required drivers subject to some adjustments related to excess or lack of employees. Constraints (24) limit the allocation of patterns to one per driver. The dual variables of constraints (23)-(24) are ϕ_{di} and ψ_c, respectively. The expression to compute the reduced cost of shift $s \in \mathcal{S}$ for driver c on day d is: $p^c_d = \sum_{i \in I} (f^c - \phi_{di}) \rho^c_{is} - \psi_c$, where ρ^c_{is} is equal to 1 if time period i is a working period for shift s, and 0 otherwise.

The procedure to generate feasible patterns is as follows:

- Step 0: Compute one initial feasible pattern for each driver $c \in C$
- Step 1: Solve problem (22)-(26) and get the value of dual variables ϕ_{di}, ψ_c.
- Step 2: Compute the reduced cost of each shift $s \in \mathcal{S}$ for each day $d \in D$ and for each driver $c \in C$
- Step 3: Generate the lowest reduced cost pattern respecting the working rules for each driver $c \in C$ based on the shifts reduced costs computed at Step 2
- Add the new patterns (not already in $\mathcal{T}'$) generated to $\mathcal{T}'$. If a new pattern is added for any of the drivers $c \in C$ go to Step 1, else stop.

3. NUMERICAL RESULTS

In this section, we evaluate the performance of the two-stage stochastic model under different configurations. We also compare the solutions and costs to draw conclusions concerning the interest of this study and how it could be used to take managerial medium term decisions. Experiments were run on an Ubuntu 20.4 system with an Intel Xeon Gold 6230 2.10 GHz CPU and 4Go to 10Go RAM allocated depending on the size of the instance. All CPU times are given in seconds. We first introduce the instance generation process.

We consider a planning horizon of 6 days, as the 7th day of the week is considered a day-off. A day is divided into 10 time periods of one-hour length each. The demand is given by O, the total number of packages to deliver per day. To have more realistic instances, we use a random perturbation factor $\gamma \in [0.8, 1.5]$ to compute the total number of packages to deliver on day d. The total demand $\tau_d = O \times \gamma$ on day d is then distributed among the different time periods following a probability distribution. We use a gamma distribution with parameters $k = 2$ and $\Theta = 2$ to represent a demand with more requests arriving during the first time periods of the day, and a normal distribution with parameters $\mu = 5$ and $\sigma = 2$ to represent requests arriving in increasing numbers until the middle of the day and then decreasing.

We denote by τ_{di} the total demand at time period $i \in I$ on day $d \in D$. The demand at each time period is then split between the pre-ordered requests and the express requests using a variability coefficient var_{di} on each day and time period. var_{di} represents the percentage of the demand on day d and time period i that is not known in advance. We use two types of variability: a medium variability where the percentage of unknown requests increases throughout the day and taking a higher value at each new day of the week, and a high variability where the uncertainty increases drastically throughout the first day and increase slightly over the rest of the week. The unknown requests are the stochastic part of the demand which is represented with scenarios generated as follows: let $\bar{\xi}_{di} = \frac{\tau_{di} \times var_{di}}{100}$ be the mean number of unknown packages to deliver at time period i on day d. All possible demand realisations ξ^{ω}_{di} are generated and a perturbation is applied to the demand $\kappa^{\omega}_{di} = \xi^{\omega}_{di} - \bar{\xi}_{di}$ (i.e., the perturbation is positive if $\xi^{\omega}_{di} \geq \bar{\xi}_{di}$ and negative if $\xi^{\omega}_{di} < \bar{\xi}_{di}$). The probability p^{ω}_{di} of each scenario $\omega \in \Omega_{di}$ is computed following a probability mass function with a Poisson distribution. Finally, the orders are distributed among the o-d pairs by randomly choosing an origin and a destination pair. The deterministic demand g_{dip} is the sum of all the pre-ordered requests allocated to o-d pair p on day d and time period i, and the stochastic demand h^{ω}_{dip} is the sum of all the express requests deliveries allocated to o-d pair p in this particular scenario ω. The total demand as used in the model is $b^{\omega}_{dip} = g_{dip} + h^{\omega}_{dip}$, the sum of the deterministic demand and the stochastic demand in scenario ω.

For each instance, we consider a fleet of company drivers C. Each driver can use one of the three different types of vehicles with parameters described in Table 2. Column labelled Cap. gives is the maximum number of packages that can be delivered per driver per time period. The wear cost is the cost of using the vehicle per time period, the delivery and pickup costs are the costs per package delivered using the vehicle. Additionally, there is a cost of 10 per hour for each internal driver, thus the total fixed cost f^c is 10 plus the wear cost of the vehicle allocated to driver c. The fixed cost of a pattern f^c_t is defined as the sum of the fixed costs at every working hour plus an extra 100 cost over the week for managing the driver. The variable cost associated to the delivery of a package is the sum of the delivery cost (presented in column Del. cost) and the pickup cost. External deliveries cost 50 per package.

Table 2. Vehicle parameters

Type	Cap.	Wear cost	Pickup cost	Del. cost
Car	4	15	2.5	2.5
Motorcycle	3	10	2.5	2.5
Bicycle	3	0	7.5	7.5

We define a class of instances for each combination of total orders, probability distribution (labelled in Table 3 as Dist.) and variability (labelled in Table 3 as Var.), and generate a set of 5 instances for each class. Instances are summarised in Table 3.

Table 3. Class instances summary

Class	Total orders	Dist.	Var.	Nb. instances
0	50	normal	medium	5
1	50	normal	high	5
2	50	gamma	medium	5
3	50	gamma	high	5
4	100	normal	medium	5
5	100	normal	high	5
6	100	gamma	medium	5
7	100	gamma	high	5

The set of possible shifts $\mathcal{S}$ is generated based on three parameters: the minimum and maximum length (in time periods) of a shift (min_S and max_S, respectively) and the interval for the start between two consecutive shifts of the same length (Δ_S). For instance, if $\Delta_S = 2$ and a shift with a length of 4 time periods starts at time period 0, then the next shift with a 4-period length will start at 2. We consider an additional shift of length 0 to represent a day-off. The number of possible shifts depends on parameters min_S, max_S, Δ_S and the number of time periods $|I|$. We use the combination of parameters $min_S = 6$, $max_S = 8$, $\Delta_S = 2$ to generate a set $\mathcal{S}_r$ of 6 shifts with low flexibility, and parameters $min_S = 4$, $\max_S = 8$, $\Delta_S = 1$ to generate a set $\mathcal{S}_f$ of 26 shifts with more flexibility.

The set of possible weekly patterns $\mathcal{T}$ is generated from the set of shifts $\mathcal{S}$ and three parameters: the minimum

and maximum working hours over the week ($min_{\mathcal{T}}$ and $max_{\mathcal{T}}$, respectively) and the minimum number of resting days ($rest_{\mathcal{T}}$). We set $rest_{\mathcal{T}} = 1$ to impose at least one day-off between the 1st day and the 6th day of work. As for the weekly working hours, we set $min_{\mathcal{T}} = 10$ and $max_{\mathcal{T}} = 40$. We use two strategies to generate the set of patterns $\mathcal{T}$: we do the complete enumeration of feasible patterns using either $\mathcal{S}_r$ or $\mathcal{S}_f$ and the values of $min_{\mathcal{T}}$, $max_{\mathcal{T}}$,and $rest_{\mathcal{T}}$, or we generate patterns with the pricing heuristic presented in the Section 2.2.

To measure the impact of using different pattern generation methods on the objective value and the CPU time, we solve the set of instances with the set of patterns $\mathcal{T}$ and with the set of patterns generated with the heuristic pricing procedure denoted as $\mathcal{T}'$. We denote by $\mathcal{T}_r$ and $\mathcal{T}'_r$ the set of patterns generated from the set of shifts $\mathcal{S}_r$. Similarly, we denote by $\mathcal{T}_f$ and $\mathcal{T}'_f$ the set of patterns generated from the set of shifts $\mathcal{S}_f$. We decompose the problem into daily problems and solve them using shifts $\mathcal{S}$ and $\mathcal{S}_f$ to compute a lower bound that is not feasible in terms of working rules for the weekly version of the problem. The difference between the weekly solution and the lower bound measures the cost of respecting the working rules over the week. Additionally we build a feasible solution from the lower bound by replacing shifts with external drivers until the solution respects the weekly working rules. This is used to measure the interest of solving a large weekly model compared to a simple heuristic procedure to obtain a feasible solution from an infeasible one.

Results are presented in Table 4. For each class of instances we give the mean value over all instances of each measure with the full patterns $\mathcal{T}_r$ (upper row) and the priced patterns $\mathcal{T}'_r$ (bottom row). The set $\mathcal{T}_r$ is the same for all instances, whereas $\mathcal{T}'_r$ is different for each instance since it depends on the solution of the pricing problem presented in Section 2.2. Values t_h and $|\mathcal{T}'_r|$ represent respectively the time in seconds spent generating the set $\mathcal{T}'_r$ and its size. The total time in seconds required to define and to solve the two-stage problem with the L-shaped method is given by *time*. Z is the objective value of the solution. We set a time limit of one and two hours for instances with 50 and 100 orders per day, respectively. Δ_{LB} measures the percentage difference between the best integer solution found with the two-stage method and a lower bound related to the solution of the problem without considering weekly working rules. In a similar way, Δ_{UB} measures the percentage difference between the best integer solution found with the two-stage method and an upper bound found with the solution obtained after making the lower bound feasible with respect to the weekly working rules by means of external drivers. Finally, Δ_Z measure the cost difference (in percentage) between the solution to the problem with full patterns and with priced patterns.

Table 4 shows that even if the set of patterns $\mathcal{T}'_r$ is approximately 100 times smaller than the full set of patterns $\mathcal{T}_r$, the problem is not much easier to solve when we look at the values of *time*. We remark that the value of the optimality gap was lower than 0.1% for all the instances solved. It is thus preferable to solve the problem with full patterns when possible. On the other hand, Δ_Z shows that the solution obtained with priced patterns is

Table 4. Solution comparison with different pattern generation methods

Cl	t_h	$\lvert\mathcal{T}'_r\rvert$	*time*	Z	Δ_{LB}	Δ_{UB}	Δ_Z
0	-	-	1382	8001	-1.80	8.49	0.93
	144	266	2613	8076	-2.70	7.49	
1	-	-	1305	7974	-1.61	9.20	0.80
	246	339	1949	8038	-2.39	8.34	
2	-	-	4025	9,619	-2.42	6.07	0.41
	68	171	3875	9663	-2.82	5.64	
3	-	-	3916	11148	-2.57	5.79	0.56
	118	205	2572	11211	-3.11	5.21	
4	-	-	8532	16699	-1.72	5.30	0.45
	245	343	8607	16773	-2.16	4.83	
5	-	-	6927	16374	-1.73	4.51	0.79
	228	341	9085	16508	-2.50	3.69	
6	-	-	8050	21055	-2.54	4.02	0.47
	190	274	8040	21154	-3.00	3.54	
7	-	-	7736	22464	-2.51	4.36	0.58
	254	306	6924	22594	-3.06	3.76	

less than 1% from the optimal solution obtained with full patterns, and thus the heuristic generates good quality patterns. Additionally we can observe that the distance to the lower bound is small, -1.6% to -2.57% with full patterns, thus the impact of considering the weekly rules is not very high on these instances. This difference is expected to be larger if longer planning horizons and additional working rules over the week are considered. Conversely, the distance to the upper bound shows that when weekly rules are not considered and the resulting solution is made feasible by using external drivers, the impact on the objective value is significant, up to 8.49% when considering full flexibility patterns on instance class 0.

We solve the same instances as before but including more flexibility. Patterns are generated with the pricing heuristic using shifts $\mathcal{S}_f$. Results are presented in Table 5. The time in seconds for generating the patterns and the number of patterns generated are presented in columns labelled t_h and $|\mathcal{T}'_f|$. The time in seconds for defining and solving the problem, the value of the objective function and the optimality gap of the two-stage stochastic problem are presented in columns *time*, Z_f, and *Gap*. The time limit is set to one and two hours for instances with 50 and 100 orders, respectively. $\Delta_{Z_f - Z_r}$ measures the difference (in percentage) between the best integer solution found with $\mathcal{T}_r$ and the best integer solution found with $\mathcal{T}'_f$.

First, we can observe that with increased flexibility the number of patterns is 3 to 4 times higher than in Table 4. *Gap* shows that the problem is more difficult to solve with $\mathcal{T}'_f$ than with $\mathcal{T}_r$ as the optimality gap is larger for the problem considering patterns with more flexibility. Negative values of $\Delta_{Z_f - Z_r}$ reveal that our approach for generating patterns can lack of quality when the number of possible patterns become too important. Although we cannot solve the problem with $\mathcal{T}_f$ and thus we cannot

Table 5. Results with a high flexibility set of patterns generated heuristically

Class	t_h	$\lvert\mathcal{T}'_f\rvert$	$time$	Z_f	Gap	$\Delta_{Z_f-Z_r}$
0	470	970	3678	8147	9.28e-3	-1.83
1	352	782	3687	8116	1.17e-2	-1.78
2	284	769	3670	8954	5.81e-3	6.90
3	262	764	3701	10308	7.35e-3	7.52
4	742	1140	8451	16755	1.89e-2	-0.33
5	700	1270	8519	16700	2.94e-2	-1.99
6	1108	1430	8580	19970	9.22e-3	5.15
7	667	1163	8499	20993	8.54e-3	6.54

Table 6. Value of the stochastic solution for each class of instance

Class	$VSS_{fix}(\%)$	$VSS_{var}(\%)$	$VSS_{ext}(\%)$	$VSS_t(\%)$
0	-2.10	-0.18	2.66	0.57
1	-2.93	-0.38	3.74	0.74
2	-2.90	-0.29	4.54	1.34
3	-4.79	-0.60	7.49	2.10
4	-7.33	-0.75	10.41	2.32
5	-8.60	-1.20	12.30	2.49
6	-2.68	-0.28	4.94	1.98
7	-4.04	-1.03	9.83	4.76

estimate the quality of patterns generated, $\Delta_{Z_f-Z_r}$ shows that increasing flexibility can lead to important cost savings (up to 7.52% for instances from Class 3). In addition, we find smaller objective values with instances having a gamma distribution for the demand and negative values with instances having a normal distribution. Theses results reveal that flexibility can really matter depending on the distribution of the requests, here with a gamma distribution flexibility is really important. We cannot conclude that with a normal distribution of the demand the flexibility does not have impact since the negative values of $\Delta_{z_f-Z_r}$ could come from the pattern generation method not performing well with this kind of distribution.

To estimate the interest of solving a two-stage stochastic problem we measure the value of the stochastic solution (VSS) on our set of instances. Let us recall that $VSS = EEV - RP$ where RP is the objective function of the two-stage stochastic problem and EEV is obtained after solving the mean value problem and then solving the two-stage problem with the first stage decisions from the mean value problem. The mean scenario is obtained by aggregating all scenarios into one single scenario considering their corresponding probability. We use the set of patterns $\mathcal{T}'_f$.

Results are presented in table 6. We decompose the total value of the stochastic solution (VSS_t) into VSS_{fix}, VSS_{var}, VSS_{ext} (for the fixed staffing costs, variable costs, and external driver costs, respectively). Negative values of VSS indicate that the EEV solution finds lower costs. Note that VSS_t cannot be negative as EEV cannot be better than RP. All values of VSS_{fix} and VSS_{var} are negative which indicates that the mean value problem allocates less working hours to company drivers and thus more deliveries are done by external drivers as can be seen with the values of VSS_{ext}. The external cost can be significantly higher for some class of instances: 10.41% and 12.30% for classes 4 and 5 but this comes with high fixed costs savings -7.33% and -8.6%. We can observe that VSS_t increases with the total number of orders and is in general higher with the gamma distribution. As expected, VSS_t also increases with high variability instances (classes 1, 3, 5, and 7).

4. CONCLUSION

In this paper we addressed a tour-scheduling problem for attended home delivery with uncertain order requests. This problem is important as numerous attended home delivery providers rely on a flexible workforce where drivers have individual preferences, several skills, and different working contracts. This flexibility adds complexity to attended home delivery operations planning because of the large number of working regulations that need to be considered in the problem, especially in multi-period problems. If these aspects are not included in the optimisation phase of the operations planning, poor-quality schedules are affected to employees, deteriorating their working conditions. We defined a two-stage stochastic programming model to represent the problem and solved it using a multicut L-shaped method. Numerical results showed that the weekly working rules considered increased the cost by only a small amount (up to 2.54%), and that

Future research perspectives include the implementation of a Branch & Price method to generate the set of weekly patterns.

REFERENCES

Birge, J.R. and Louveaux, F.V. (1988). A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34(3), 384–392.

Cappanera, P., Scutellà, M.G., Nervi, F., and Galli, L. (2018). Demand uncertainty in robust Home Care optimization. *Omega*, 80, 95–110.

Goel, A. and Irnich, S. (2017). An exact method for vehicle routing and truck driver scheduling problems. *Transportation Science*, 51(2), 737–754.

Restrepo, M.I., Semet, F., and Pocreau, T. (2019). Integrated shift scheduling and load assignment optimization for attended home delivery. *Transportation Science*, 53(4), 1150–1174.

Restrepo, M. I., Rousseau, L.M., and Vallée, J. (2020). Home healthcare integrated staffing and scheduling. *Omega*, 95, 102057.

Steeg, J. and Schröder, M. (2008). A hybrid approach to solve the periodic home health care problem. In *Operations Research Proceedings 2007*, 297–302. Springer.

Trautsamwieser, A. and Hirsch, P. (2014). A Branch-Price-and-Cut approach for solving the medium-term home health care planning problem. *Networks*, 64(3), 143–159.

Wirnitzer, J., Heckmann, I., Meyer, A., and Nickel, S. (2016). Patient-based nurse rostering in home care. *Operations Research for Health Care*, 8, 91–102.

Zäpfel, G. and Bögl, M. (2008). Multi-period vehicle routing and crew scheduling with outsourcing options. *IJPE*, 113(2), 980–996.