An Approximate Dynamic Programming Approach to Attended Home Delivery Management

14 July 2016

Abstract

We propose a new method of controlling demand via delivery time slot pricing in attended home delivery management. The focus is on development of an approach that is suitable for industry-scale implementation. To that end, we exploit a relatively simple yet effective way of approximating the delivery cost by decomposing the overall delivery problem into a collection of smaller, area-specific problems. This cost estimation serves as an input to an approximate dynamic programming method which provides estimates of the opportunity cost associated with having a customer from a specific area book delivery in a specific time slot. These estimates depend on the area and on the delivery time slot under consideration.

Using real, large-scale industry data, we estimate a demand model including a multinomial logit model of the customers’ delivery time slot choice, and show in simulation studies that we can improve profits by over 2% in all tested instances relative to using a fixed price policy that is commonly encountered in e-commerce. These improvements are achieved despite having made strong assumptions in the delivery cost estimation. These assumptions allow us to reduce computational runtime to a degree suitable for real-time decision making on delivery time slot feasibility and pricing. Our approach provides quantitative insight to the importance of incorporating expected future order displacement cost into the opportunity cost estimation alongside marginal delivery costs.

1 Introduction

Online grocery sales are growing in double digits in the United Kingdom (12.5% in 2015), and new market entrants like Amazon are increasing competitive pressure on the incumbents to maintain their share of the market as reported by the market research company Mintel (2016). The fulfilment logistics are one of the main cost drivers in the business, and hence it is important for retailers to carefully balance the need for high customer service levels in terms of narrow home delivery time slots with the associated cost of service provision. Mintel (2016) find that 32% of current online grocery shoppers shop online because of improvements in delivery time slots. Most UK retailers have moved to one hour slots and offer them over
a wide range from around 6am to 11pm. Whilst customers expect convenient delivery slots, they are not prepared to pay much for peak-time delivery; around £6 to £7 is the maximum that UK retailers currently charge. However, Yang et al. (2016) show in an empirical study that even small fee differences can influence customers’ delivery time slots decisions, and may lead to overall improved profitability. Mintel (2016) further find that about 80% of online grocery customers in the UK pay per delivery, whilst the remaining 20% have delivery passes. Therefore, influencing customers’ delivery time slot choices via dynamic pricing seems a promising way of achieving more profitable delivery schedules.

This problem has been recently receiving considerable attention by the academic community (see literature review in §2), and various contributions have been made towards estimating choice behavior in this context, dynamic pricing or availability control of delivery time slots, and on the approximation of the delivery cost associated with a set of orders.

In this work, we develop a dynamic delivery slot pricing policy to manage demand over a finite booking horizon prior to the actual delivery date such that we maximize expected profit. We do not consider same-day delivery. The policy is based on a customer delivery slot choice model and reflects in its pricing the so-called opportunity cost associated with each delivery slot option. Opportunity cost of a delivery slot can be interpreted as its value, which is influenced by anticipated displaced order cost (meaning that some future orders are lost due to the capacity constraints if the customer selects this time slot) and by marginal delivery cost.

The main challenge is the calculation of the opportunity cost; once these are known, we only need a fast approach to evaluate slot feasibility and to optimize prices upon arrival of a customer wanting to book delivery. The opportunity cost and the feasibility check both require a way of approximating the NP-hard capacitated vehicle routing problem with time windows. We decompose this problem into a collection of smaller, independent subproblems corresponding to clusters of postcodes, and use a continuous approximation of the total traveling distance. Based on this decomposition, we propose an approximate dynamic programming model to estimate the opportunity costs. The latter are used as inputs to the real-time optimization of delivery time slot prices.

The main contribution of our work over the existing literature is the quantification of the profit impact of incorporating the expected order displacement cost into the opportunity cost estimate in a way suitable for large-scale applications. Real-time decision making is achieved by using a simple approximation of the vehicle routing problem that allows an easy and quick time slot feasibility check, and by exploiting results from the literature on how to solve the pricing problem efficiently. We demonstrate these contributions by applying the method to a simulation study based on real data from our industry partner, and show that our approach can produce significant profit improvements over fixed price benchmarks. Furthermore, the study reveals that dynamic pricing does not necessarily always improve over
fixed price strategies; their success hinges on good opportunity cost estimates that include both marginal delivery cost and expected order displacement cost.

This paper is organized as follows: in §2 we review and discuss related literature. In §3 we formally state the problem formulation, followed by more detailed explanations on how we approximate the delivery costs and value function in §4 and §5, respectively. Based on these approximations, we obtain a pricing policy as defined in §6. This policy is tested against fixed price benchmarks in a numerical study in §7, and we draw conclusions in §8.

2 Literature review

For a review on e-fulfillment from an operational research perspective, see Agatz et al. (2008). A recent overview of fulfillment and distribution from a qualitative point of view has been provided by Hübner et al. (2016).

In their seminal work, Campbell and Savelsbergh (2005) investigate a dynamic routing and scheduling problem of a grocery vendor who needs to decide which deliveries to accept or reject, and in which time slot to deliver the accepted orders. Customers are assumed to have a certain time slot profile that represents all slots that they are willing to accept; the firm assigns one of these slots to the order if the order is accepted.

The model of Campbell and Savelsbergh (2005) represents demand as an arrival process that is not influenced by the firm’s decisions. In a follow-up paper, Campbell and Savelsbergh (2006) use a relatively simple model of customer behavior to capture the effect of incentives (such as delivery charges) on the probability of a particular time slot being chosen. The focus is on influencing delivery time slot choices so that delivery cost is reduced (as opposed to improving total expected profit, which is the focus of our paper).

A more sophisticated model of customer choice, namely the multinomial logit (MNL), is used by Asdemir et al. (2009) in dynamic time slot pricing. They propose a dynamic programming (DP) formulation under the assumption that the problem can be addressed for each geographic area (e.g. a postcode) independently and that delivery capacity levels are committed a priori to each delivery time slot. Therefore, delivery costs are fixed, and the objective is to maximize expected profit from orders. The state space of their DP grows exponentially in the number of delivery time slots, which makes practical application difficult. In our work, we likewise use the MNL choice model and consider a DP formulation decomposed by geographic area, but we also discuss how to obtain these areas, and how to approximately solve the DP for industry-sized problems. Furthermore, the delivery cost approximation in our model is dynamic and not fixed as in Asdemir et al. (2009).

Agatz et al. (2011) address the issue of geographic dependence of the attended home delivery problem. Their approach is related to ours in that they also use the work of Daganzo (1987) to obtain a continuous delivery cost approximation. However, they consider the prob-
lem of which delivery time slots to offer in which area so as to reduce delivery costs while meeting service requirements. The work does not address the problem on how to influence customer choice behavior so as to improve expected profit, which is the aim of our work.

Ehmke and Campbell (2014) study an integrated routing and scheduling problem in the context of attended home delivery. The objective is to maximize the number of requests accepted for delivery subject to retaining feasible tours. Customers’ delivery slot choices are assumed to be independent of the firm’s decision making which is limited to accepting or rejecting delivery slot booking requests. Similarly, Cleophas and Ehmke (2014) discuss decision making of acceptance or rejection of delivery requests, but also propose to reserve transport capacities for specific delivery areas and time windows with a high expected order value.

Yang et al. (2016) estimate an MNL choice model from real e-grocer data and demonstrate numerically that using this model in time slot pricing to influence demand can improve overall profitability. They employ insertion heuristics to update a pool of feasible routes as orders are coming in over the booking horizon, and derive marginal delivery cost estimates from them that are being used as estimates of the opportunity cost of accepting an order into a particular time slot. In our work, we draw on their choice model but use a different (and computationally much more efficient) way of estimating marginal delivery costs. Furthermore, our opportunity cost estimates are not only based on delivery cost but also take potential future order displacement cost into account. The work of Cleophas and Ehmke (2014) is related to Yang et al. (2016) in that both papers combine demand fulfillment and revenue management, but the latter work is concerned with time slot pricing and incorporates customer choice modeling (in contrast to the static demand model of Cleophas and Ehmke (2014)).

A key difficulty in the attended home delivery problem is the estimation of routing cost before we know all orders (or even none of them). Bühler et al. (2016) propose several linear mixed integer programs to approximate delivery costs based on a fixed pool of potential routes. Klein et al. (2016) integrate a linear mixed integer program (MIP) formulation into the dynamic pricing approach of Yang et al. (2016) in an attempt to anticipate future customer requests. This approach aims at obtaining opportunity cost estimates that feature both delivery costs and revenue implications like the ones we produce in this work. In other words, their MIP formulation can be seen as an approximation of the value function. However, the MIP as proposed by Klein et al. (2016) suffers from computational challenges for industry-sized problems because the number of decision variables grows exponentially in the number of delivery time slots.

Klein et al. (2015) is related to our work in that they consider time slot pricing in attended home delivery under a model of customer choice. The objective and the problem setting is very similar to ours, but they tackle the problem with a different delivery cost estimation, a
different choice model, and a different approximation of the value function (namely using a MIP formulation).

3 Problem statement

We consider an e-grocer receiving orders via an online booking system which requires customers to book their delivery when completing their purchase. Orders for a specific delivery day can be received over a finite time horizon until a certain cut-off time after which no further order are accepted. Deliveries are made after the cut-off time. We model this booking horizon as being discrete starting at time $t = 1$ and ending in time period $t = T$. Customers arrive over this time horizon to book their delivery for this specific delivery day; we follow the demand model of Yang et al. (2016) in that we do not consider delivery time slot choice beyond a single day. The customer arrivals follow a homogeneous Poisson process $\lambda$. Note that this homogeneity can be achieved by appropriately defining a non-homogeneous time grid as explained in Yang et al. (2016). Each time period is sufficiently small such that the probability of more than one customer arrival is negligible. An arrived customer will request delivery in area $a$ with probability $\mu_a$. With a probability of $\sigma_{ai}$, the number of totes in an order from area $a$ is $i \in I_a$, where $I_a$ is a finite set of order sizes that may be encountered in area $a$ (obtained from historical data). By definition, $\sum_{i \in I_a} \sigma_{ai} = 1$. We assume that order size is independent of the time of order placement and of the delivery charge.

Upon customer arrival, we need to check which time slots $s \in \{1, \ldots, m\}$ out of the total of $m$ slots are feasible for the desired area given an order size of $i$ (measured in the number of required transport totes). We denote the resulting feasible set by $F_{ai} \subset \{1, \ldots, m\}$. Subsequently, we need to decide which delivery charges to impose on the feasible slots. Faced with the resulting set of feasible slot alternatives $s \in F_{ai}$ for area $a$ and given order size $i$ as well as delivery charges $\vec{d}_a$, the customer decides according to some choice model when to book delivery (or not to book at all). This choice model specifies the probability $P_{s,F_{ai}}(\vec{d}_a)$ that a customer chooses slot $s$ given the vector of delivery charges $\vec{d}_a$ for area $a$ over all feasible slots. If the customer books slot $s$, we receive a delivery charge of $d_{as}$ and a revenue of $ir$, where $i$ is the number of totes ordered and $r$ is the average profit per tote.

This control problem can be cast in the form of a dynamic programme over states labeled $(\vec{x}, \vec{y})$. Each $x_{tas}$ represents the number of orders accepted in time slot $s$ for area $a$ until time $t$ in the booking horizon, and $y_{ta}$ is the number of totes required for all these orders received until $t$ in area $a$. In the following, we omit the time index since it will be clear from the context. Note that this state definition does not capture all the information that could be relevant to the vehicle routing problem, such as which order requested how many totes. We use this somewhat reduced state because it contains sufficient information needed for the delivery cost approximation as discussed in the next section.
We let $C(\vec{x}, \vec{y})$ represent an oracle that returns an approximation of the minimum cost to the underlying vehicle routing problem with time windows for the set of orders $\langle \vec{x}, \vec{y} \rangle$ given a fixed fleet of vehicles with known capacities. If there is no feasible solution for a given state, then $C(\vec{x}, \vec{y}) := \infty$. We denote by $F_{ai}(\vec{x}, \vec{y}) := \{ s : C(\vec{x} + 1_{as}, \vec{y} + i1_a) < \infty \}$ all feasible time slots for area $a$ into which an order with $i$ totes can be feasibly inserted given that we are in state $\langle \vec{x}, \vec{y} \rangle$, where $1_{as}$ ($1_a$) is the unit vector with 1 in the $(a, s)$th $(a)$th position. For brevity of notation, in the following we use the short-hand notation $F_{ai}$ that has been mentioned above.

Let $V_t(\vec{x}, \vec{y})$ denote the value function at stage $t$ and state $\langle \vec{x}, \vec{y} \rangle$; it represents the maximum profit obtainable from the sales process from time $t$ until the cut-off time $T$. Then the dynamic programming recursion at stage $t \in \{1, 2, \ldots, T\}$ is:

$$V_t(\vec{x}, \vec{y}) = \max_{d} \lambda \sum_{a,i} \mu_a \sigma_{ai} \sum_{s \in F_{ai}} P_{s,F_{ai}}(\vec{d}_a) \left[ ir + d_{as} + V_{t+1}(\vec{x} + 1_{as}, \vec{y} + i1_a) \right] +$$

$$\left[ 1 - \lambda \sum_{a,i} \mu_a \sigma_{ai} \sum_{s \in F_{ai}} P_{s,F_{ai}}(\vec{d}_a) \right] V_{t+1}(\vec{x}, \vec{y})$$

$$= \max_{d} \lambda \sum_{a,i} \mu_a \sigma_{ai} \sum_{s \in F_{ai}} P_{s,F_{ai}}(\vec{d}_a) \left[ ir + d_{as} - \left( V_{t+1}(\vec{x}, \vec{y}) - V_{t+1}(\vec{x} + 1_{as}, \vec{y} + i1_a) \right) \right]$$

$$+ V_{t+1}(\vec{x}, \vec{y}) \quad \forall (\vec{x}, \vec{y}) \in X. \quad (3.1)$$

The value function after cut-off is then

$$V_{T+1}(\vec{x}, \vec{y}) = -C(\vec{x}, \vec{y}) \quad \forall (\vec{x}, \vec{y}) \in X, \quad (3.2)$$

where $X$ denotes the set of all states that allow a feasible delivery schedule.

If we would somehow be able to solve this dynamic program (or at least to approximate the value function), then we could use the value function in a decision policy in the following form: given an arrival of a customer of a known area $a$ with order size $i$ during the booking horizon, we just need to solve the so-called online decision problem:

$$\{ d^*_{as} | s \in F_{ai} \} = \arg\max_{s \in F_{ai}} \sum_{a,i} P_{s,F_{ai}}(\vec{d}_a) \left[ ir + d_{as} - \left( V_{t+1}(\vec{x}, \vec{y}) - V_{t+1}(\vec{x} + 1_{as}, \vec{y} + i1_a) \right) \right].$$

The term $\left( V_{t+1}(\vec{x}, \vec{y}) - V_{t+1}(\vec{x} + 1_{as}, \vec{y} + i1_a) \right)$ represents the opportunity cost of having a customer from area $a$ book delivery in slot $s$ at time $t$ with an order of size $i$ given that we have currently orders $\langle \vec{x}, \vec{y} \rangle$ on the books. Depending on the choice model, this problem can be solved efficiently as we discuss in §6 provided that we have an approximation of the opportunity cost and a way of determining the feasible set of slots $F_{ai}$. Note that the opportunity cost reflects both revenue and cost implications of having a customer book delivery. Yang
et al. (2016) approximate opportunity cost only with the estimated delivery cost, but here we are taking both effects into account.

However, we stress that the suitability of the solution approach for practical, industry-sized application hinges on the ability to identify feasible slots $F_{a,i}$ and to solve the online pricing problem very quickly, namely in less than 100ms as advised by our industry partner. Furthermore, close to the delivery day we may have orders arriving in quick succession so that there may be no time for offline computations in between order arrivals.

In the following sections, we propose a solution approach that adheres to these tight practical limitations. The delivery cost function approximation is discussed in the next section, and the value function approximation thereafter.

4 Delivery cost approximation

Evaluation of the delivery cost function $C(\vec{x}, \vec{y})$ would require to solve a capacitated vehicle routing problem with time windows which is known to be NP-hard. However, we only require an estimate of the routing cost as opposed to the specific routes that vehicles are traveling on. Therefore, we propose to use a clustering-first, route-second strategy developed by Daganzo (1987) which simplifies the problem to a great extent whilst still retaining sufficient information to provide a useful cost estimation as we demonstrate in our numerical results in §7.

The idea is to decompose the problem geographically by assuming that each area has a single delivery van associated with it, and this van is driving to complete a full cycle in each time window within its designated area. Under some further assumptions, we can express the daily traveling distance $D_{a}$ within a given area $a$ by the simple formula given in (4.1). The assumptions are that customer locations are randomly and nearly uniformly scattered within the area, the time windows are equally long, customers only place their request into one of the time periods, and demand is balanced over time slots. The assumptions regarding uniformly distributed customers over space and time are strong. Whilst we would expect that our dynamic pricing approach will eventually lead to more uniformly distributed orders over time, some peaks and troughs are still likely to occur. Likewise the geographical distribution will likely be uneven.

Therefore, one would expect the resulting estimates to be of limited quality in as far as the actual vehicle routes would be much more flexible. However, our main objective is to devise an approach that requires very little computational time to evaluate routing cost, yet that does estimate them in a way such that it is still improving overall profitability. We define a single set of delivery areas for the delivery day under consideration, and we keep it fixed over the entire booking horizon. Whilst being restrictive, this approach allows us to (approximately) evaluate delivery cost very quickly.
Clustering: define delivery areas

First, we define rectangular delivery areas \( a \) with length \( L_a \) and width \( W_a \) that represent geographical clusters of customers to be served. The clustering problem features routing constraints and hence approaches like k-means cannot be applied in a straightforward manner; any area should be defined such that a single van can accommodate all expected orders (capacity constraint), and such that it is small enough to allow the van to complete full cycles and to visit and serve all customers in each delivery time window (time constraints).

We propose the following clustering approach: For a given weekday, e.g. Monday, we obtain the average number of daily deliveries \( N_z \) in a postcode \( z \) from the final delivery schedules of the past Mondays. Any postcode center located in a given rectangular area means that this area’s van needs to serve all daily orders associated with these postcodes. The total daily number of orders in area \( a \) is denoted by \( N_a = \sum_{z \in a} N_z \), where \( z \in a \) denotes all postcodes centered within area \( a \). These \( N_a \) orders form our expectation of total daily demand for this area; according to our assumptions above, we expect \( \lceil N_a/m \rceil \) orders in each of the \( m \) time slots.

The overall delivery region is partitioned into bands of equal size. A delivery area is defined as a piece of a band that satisfies both delivery time and capacity constraints. We only need to decide on the width of each area because its length is fixed by the latitude of each band. For illustration, Figure 1 in the numerical results section (§7) shows such a cluster derived for the Greater London region. In that numerical study, we tested different numbers of these horizontal bands and selected the one resulting in the smallest number of required vans.

For a given set of bands, we move from east to west within a given band to determine the maximum allowable width \( W_a \) of a delivery area \( a \) that satisfies:

- the time constraint that the total traveling and service time required to serve all \( \lceil N_a/m \rceil \) orders in a slot should not exceed the duration of that slot (e.g. 1 hour):
  \[
  (2L_a + W_a \lceil N_a/m \rceil /6)/v + \tau \lceil N_a/m \rceil \leq 1,
  \]
  where \( v \) represents the average traveling speed of the delivery van and \( \tau \) is the average service time at a customer location. The formula is explained further in the next subsection.

- the capacity constraint, namely that the total number of totes does not exceed the capacity \( \kappa \) of the delivery van: \( N_a \sum_{i \in I_a} i \sigma_{ai} \leq \kappa \).

This method produced realistic numbers of required vans in our numerical experiments.
Routing: approximate delivery cost

For a rectangular area \( a \) of width \( W_a \) and length \( L_a \), we define the stem distance \( \rho_a \) from the depot to the area’s center. For a given number of \( \sum_s x_{as} \) orders to be served on the delivery day under consideration, the daily traveling distance is given by

\[
D_a(\vec{x}_a) = (2\rho_a - L_a) + \sum_{s=1}^m (2L_a\delta_{as} + x_{as}W_a/6),
\]
where

\[
\delta_{as} = \begin{cases} 
0, & \text{if } x_{as} = 0, \\
1, & \text{if } 0 < x_{as} \leq M_a, \\
\infty, & \text{if } x_{as} > M_a,
\end{cases}
\]

and the maximum number of orders that can be served within a 1 hour time slot is denoted by \( M_a := \arg\max\{x \in \mathbb{Z}^+ | x + \frac{2L_a + \frac{W_a}{6}x}{1/\rho_a} \leq 1\} \). The first part of the formula expresses the stem distance traveled between depot and the area. In each time period \( s \), the vehicle travels the whole length \( L_a \) of the area twice. Note that the expected distance of two uniformly distributed points in the unit interval is 1/3. The van is assumed to travel in a full cycle covering the upper half of the rectangle on the first half cycle, and the other on its return, so the expected distance between two orders is \( W_a/6 \). We refer the interested reader to Daganzo (1987) for further details on the derivation of this formula.

We assume that driving distance is the only cost incurred by accepting deliveries (since the fleet and the drivers’ salaries are assumed to be fixed cost), so the total cost of area \( a \) is \( C_a(\vec{x}_a, y_a) = \xi D_a(\vec{x}_a) \) if the van capacity is not exceeded, i.e. \( y_a \leq \kappa \) (and \( \xi \) is a known cost per mile factor), or \( C_a(\vec{x}_a, y_a) = -\infty \) otherwise.

In summary, this delivery cost estimation has the advantage that the resulting overall cost function \( C(\vec{x}, \vec{y}) = \sum_a C_a(\vec{x}_a, y_a) \) is decomposable by delivery area, and can be quickly evaluated. We exploit these features in our approximation of the dynamic programming value function.

5 Value function approximation

For the final stage \( T+1 \), the value function decomposes by the areas: \( V_{T+1}(\vec{x}, \vec{y}) = -\sum_a C_a(\vec{x}_a, y_a) \). Since the slot pricing decisions are independent between different areas, the dynamic program as a whole decomposes by area:

\[
V^a_t(\vec{x}, y_a) = \max_{\vec{d}_a} \mu_a \sum_{i} \sigma_{ai} \sum_{s \in F_{ai}} P_{s,F_{ai}}(\vec{d}_a) \left[ ir + d_{as} - \left( V^a_{t+1}(\vec{x}_a, y_a) - V^a_{t+1}(\vec{x}_a + 1_s, y_a + i) \right) \right] + V^a_{t+1}(\vec{x}, y_a) \quad \forall (\vec{x}_a, y_a) \in \mathcal{X}_a,
\]

(5.1)
where $\mathcal{X}_a = \{(\vec{x}_a, y_a) \in (\mathbb{N}^m \times \mathbb{N}) | 0 \leq x_{as} \leq M_a, y_a \leq \kappa\}$. In the following, we omit the area index $a$ since we focus on the solution of this single-area dynamic program. It is still intractable because of the large state space which grows exponentially in the number of time slots. Therefore, we propose to approximate its value function with a function linear in $\vec{x}$ and time $t$. Note that we omit dependence on the total number of totes $y$ to be delivered in the area under consideration. This number is of relevance in the cost function to determine whether all orders fit into the van, and is used in the determination of slot feasibility. However, the time constraints are usually much more restrictive to the problem than van capacity if the delivery time slots are narrow (say, 1 hour), so we ignore $y$ in the approximation:

$$V_t(\vec{x}, y) \approx \bar{V}_t(\vec{x}) := \gamma_0 - \sum_s \gamma_s x_s + (T + 1 - t)\theta, \quad \forall (\vec{x}, y) \in \mathcal{X}. \quad (5.2)$$

The parameter $\gamma_s$ can be interpreted as an estimate of the opportunity cost of accepting an order in slot $s$ (regardless of the order size $i$) since $V_{t+1}^a(\vec{x}_a, y_a) - V_{t+1}^a(\vec{x}_a + \mathbf{1}_s, y_a + i) \approx \gamma_s$.

In Algorithm 1 we outline the approximate dynamic programming procedure that we propose to find the parameters $\vec{\gamma}$ and $\theta$. We sample $k_{\text{max}}$ paths of order arrivals indexed by $k$, and use our current best knowledge of the parameters to approximate the value-to-go in the dynamic programming recursion. This allows us to step forward in time, and at each time step $t$ we calculate the value $\hat{V}_t^{(k)}$ of being in state $\vec{x}_t^{(k)}$. Next, we update the parameters with a stochastic gradient step. Specifically, we seek to find parameters that bring our value function approximation $\hat{V}_t^{(k)}$ closer to the observed value $\hat{V}_t^{(k)}$:

$$\min_{\vec{\gamma}, \theta} \frac{1}{2} \mathbb{E} \left[ \hat{V}_t^{(k)}(\vec{x}_t^{(k)}) - \hat{V}_t^{(k)} \right]^2.$$

The updating scheme (with fixed step sizes $\alpha_1, \alpha_2, \alpha_3$ along the negative gradient directions) is thus the following:

$$\gamma_0^{(k)} = \gamma_0^{(k-1)} - \alpha_1 [\bar{V}_t^{(k-1)}(\vec{x}_t^{(k)}) - \hat{V}_t^{(k)}],$$
$$\gamma_s^{(k)} = \gamma_s^{(k-1)} - \alpha_2 [\bar{V}_t^{(k-1)}(\vec{x}_t^{(k)}) - \hat{V}_t^{(k)}] x_{ts}^{(k)}, \quad \forall s,$$
$$\theta^{(k)} = \theta^{(k-1)} - \alpha_3 [\bar{V}_t^{(k-1)}(\vec{x}_t^{(k)}) - \hat{V}_t^{(k)}](T + 1 - t).$$

We stop the procedure after $k_{\text{max}}$ iterations (corresponding to the $k_{\text{max}}$ sample paths), and use the final value function approximation in the real-time control policy to make pricing decisions as discussed in the next section.
Algorithm 1 Approximate Dynamic Programming Procedure.

1: Initial value function parameters: $\gamma \leftarrow 0, \theta \leftarrow 0$ to define $\hat{V}_t^{(0)}(\bar{x})$ in (5.2) for all $t, \bar{x}$.
2: Boundary condition: $\bar{V}_{T+1}^{(k)}(\bar{x}) = -\xi D(\bar{x})$ for all $\bar{x}, k = 1, \ldots, k_{\text{max}}$, where $\xi$ is cost per mile and $D(\bar{x})$ is the total mileage driven.
3: Iteration counter: $k \leftarrow 1$
4: Initial state: $\bar{x}_1^{(k)} = 0, y_1^{(k)} = 0$ (initially no orders on record)
5: while $k \leq k_{\text{max}}$ do
6: Generate sample path of order arrivals (for area under consideration): $(\bar{R}_1^{(k)}, \ldots, \bar{R}_T^{(k)})$, where $\bar{R}_t^{(k)}$ is a vector containing either zeros (no order), or information on order size $i_t^{(k)}$, and its profit $r_t^{(k)}$ before delivery cost.
7: for all $t = 1, 2, \ldots, T$ do
8: Define feasible set $F_i := \{s \mid \bar{x}_{ts}^{(k)} + 1 \leq M_a\}$ if there is sufficient van capacity ($y_t^{(k)} + i_t^{(k)} \leq \kappa$), or $F_i = \emptyset$ otherwise. Solve for optimal $\bar{d}$ in:
9: $\hat{V}_t^{(k)} = \max_d \sum_{s \in F_i} P_{s,F} (d) \left( r_t^{(k)} + d_s - \left[ \hat{V}_{t+1}^{(k-1)}(\bar{x}_t^{(k)}) - \hat{V}_{t+1}^{(k-1)}(\bar{x}_t^{(k)} + 1_s) \right] + \bar{V}_{t+1}^{(k-1)}(\bar{x}_t^{(k)}) \right)$
10: Update value function parameters $\bar{\gamma}, \theta$ using $\hat{V}_t^{(k)}$ with a stochastic gradient step to define the new approximation $\bar{V}_t^{(k)}(\bar{x})$ for all $\bar{x}$.
11: Simulate customer’s decision under prices $\bar{d}$ and available slots $F_i$, and accordingly define next state $\bar{x}_{t+1}^{(k)}$ and $y_{t+1}^{(k)}$.
12: $k \leftarrow k + 1$
13: end while

6 Real-time control policy

As soon as a customer request arrives, we need to determine which delivery slots are feasible and then to decide on the delivery pricing. In fact, for the sake of practical relevance, this decision needs to be made almost instantaneously. We propose to determine the area clusters and the corresponding value function approximations before the start of the booking horizon.

Within the booking horizon, we can then check the “feasibility” of delivery in a given area $a$ by checking whether (a) the current number of totes to be delivered in this area exceed the van’s capacity, and (b) whether we exceed the maximum number of orders $M_a$ in any slot. Both conditions are simple comparisons of known numbers.

The proposed “feasibility” check is unrealistic in as far as the actual routing would be looking very different from the assumed area-based routing approximation. However, it can be seen as a conservative estimate, so whilst slots deemed “infeasible” may actually be feasible in the actual vehicle routing process, “feasible” slots would be expected to indeed be feasible. Keeping sets of feasible vehicle routings as done by Yang et al. (2016) are likely to be too
time-consuming for real-time decision making.

Next, given the set of feasible slots $F_{ai}(\vec{x}, \vec{y})$ for the incoming request in area $a$ for $i$ totes and associated order profit $ir$ given state $(\vec{x}, \vec{y})$, we need to find the optimal delivery charges to offer. We want to solve the following problem:

$$\vec{d} = \arg\max_{s \in F_{ai}} P_{s,F_{ai}}(\vec{d}_a) \left[ ir + d_{as} - \gamma_s \right].$$

(6.1)

The difficulty of solving this problem depends on the choice model underpinning the customers’ time slot decisions as well as on the range of feasible price vectors. Note that essentially the same problem needs to be solved repeatedly in the approximate dynamic programming iterations as described in Algorithm 1. Various constellations have been investigated and tractable formulations proposed, for example, using the multinomial logit (MNL) choice model and continuous prices (Dong et al., 2009), MNL with a discrete price set (Davis et al., 2013), nested MNL with bounded continuous prices (Rayfield et al., 2013).

We are using an MNL choice model with continuous prices as in Yang et al. (2016), i.e. the probability of a customer to choose delivery slot $s$ given that the set $F$ of slots is available at prices $\vec{d}$ is defined by

$$P_{s,F}(\vec{d}) = \frac{\exp(\beta_0 + \beta_s + \beta_d d_s)}{\sum_{k \in F} \exp(\beta_0 + \beta_k + \beta_d d_k) + 1},$$

where $\beta_0$ is an offset parameter, $\beta_s$ measures the attractiveness of slot $s$ and $\beta_d$ is the price sensitivity. The no-purchase utility has been normalized to zero.

**Proposition 6.1.** For the MNL choice model and continuous prices, the optimal solution to (6.1) is given by

$$d_s^* = \gamma_s - ir - \frac{h}{\beta_d}, \quad \forall s \in F,$$

where $h$ is the unique solution of

$$(h - 1) \exp(h) = \sum_{s \in F} \exp(\beta_0 + \beta_s + \beta_d (\gamma_s - ir)).$$

(6.2)

**Proof.** Theorem 1 in Dong et al. (2009).

Standard Newton root search can be employed to find $h$ in (6.2).

7 Numerical results

We test our approach in a simulation study based on real data from our industrial partner. With these experiments, we seek answers to the following questions:
1. Does the clustering result in a number of areas that is realistic, i.e. it does not exceed fleet size?

2. Does the proposed policy deliver a consistent improvement in profitability over various demand scenarios compared to benchmarking policies? How does dynamic pricing with different opportunity cost estimates perform, and what is the value of including order displacement cost in the opportunity cost estimate?

3. Is computational speed sufficient for potential commercial application?

4. Can we glean managerial insights from the new pricing approach regarding how whether delivery areas that are far from the depot should be priced differently than closer areas, and how order volume influences the average delivery charge?

We discuss these in the following study and summarize our findings at the end of the section.

7.1 Data description

The data has been provided by a major e-grocer in the United Kingdom, specifically focusing on delivery operations in the Greater London area. The same data was used by Yang et al. (2016) to estimate time-dependent customer arrival processes and to calibrate a multinomial logit (MNL) choice model. It contains anonymized customer booking requests over 6 months from the beginning of June to the end of November 2011, all made through the company’s website. Each customer is identified by an ID number; they have to be logged into their personal account in order to book delivery. The postcode of each customer is contained in the data. Each request for display of available delivery slots has been stored, specifically which customer is requesting, the order size in terms of number of totes, the time of the request, which delivery day, which slots were displayed as available and at what charges. The customer’s delivery slot decision is likewise being recorded. Expenditure is not included in the data set, but a fixed average revenue per tote of £30.39 was provided. Furthermore, since we aim to maximize profit, we assume the average profit before delivery cost to be 30% of revenue.

Regarding the range of data that we used, only customers who may need to pay per delivery were considered, i.e. customers who do not hold a subscription for free delivery. Furthermore, we focussed our attention on customers wanting to book for a Monday delivery. Customers who looked at a Monday delivery but then decided for delivery on another day are considered to be lost sales because we optimize for each day individually. It would be desirable to include multiple days in the choice model, but this increases its complexity to a great degree and is outside the scope of this paper. There are 26 booking histories associated with Monday deliveries over the full time horizon available, each of them containing several thousand customer arrivals. We only use the latest instance of a booking request of a
particular customer for a particular delivery day (sometimes a customer looked at the options for the same delivery day at different times). For the sake of simplicity, we did not include cancelations in the model; thus we removed canceled orders from the data.

The arrival process was estimated over $T = 6990$ non-homogenous periods with arrival probability $\lambda = 0.824$ in each period. Note that the time period sizes were chosen in a way to allow the constant probability, i.e. early periods are wide whereas periods close to the delivery day are narrow. We refer to Yang et al. (2016) on details of the estimation procedure. Since the data contains the postcode of each customer, we can calculate the probability $\mu_z$ that a new delivery request hails from a particular postcode $z$ as the proportion of requests from $z$ relative to all requests. This probability will be used in the next section to derive the probability $\mu_a$ that an arrived customer belongs to area $a$ (that consists of a collection of postcodes).

Regarding the delivery slot choices, the online grocer is using 27 partly overlapping delivery time slots of one hour duration starting either at the full hour or at half past the hour. However, our routing cost estimation assumes that we have non-overlapping slots. Therefore, we transformed the data by randomly changing any half-past-the-hour slot request to either the preceeding or succeeding overlapping slot with equal probability. We estimated the MNL choice model based on this modified data and obtained parameters as reported in Table 1. For details on how to derive the MNL parameters and the arrival process, we refer to Yang et al. (2016). Note that the base utility parameter is smaller relative to the utility of the non-purchase option (which has been set to zero). This is because the data contains many cases where customers looked at a particular day but then went on to either select a slot on a different day or not to book delivery at all. The $\beta$ parameters reflect the popularity of different slots; e.g., 9-11am, noon-1pm, or 9-10pm are particularly attractive. The price sensitivity parameter $\beta_d$ is negative in line with expectation. All delivery charges contained in the data belong the set $\{£0, £1, \ldots £7\}$.

<table>
<thead>
<tr>
<th>$\beta_6$</th>
<th>$\beta_7$</th>
<th>$\beta_8$</th>
<th>$\beta_9$</th>
<th>$\beta_{10}$</th>
<th>$\beta_{11}$</th>
<th>$\beta_{12}$</th>
<th>$\beta_{13}$</th>
<th>$\beta_{14}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.0305</td>
<td>-0.3591</td>
<td>0.3107</td>
<td>0.5922</td>
<td>0.6154</td>
<td>0.0796</td>
<td>0.5356</td>
<td>-0.2415</td>
<td>-0.6286</td>
</tr>
<tr>
<td>$\beta_{15}$</td>
<td>$\beta_{16}$</td>
<td>$\beta_{17}$</td>
<td>$\beta_{18}$</td>
<td>$\beta_{19}$</td>
<td>$\beta_{20}$</td>
<td>$\beta_{21}$</td>
<td>$\beta_{22}$</td>
<td></td>
</tr>
<tr>
<td>-1.6736</td>
<td>-0.4351</td>
<td>-0.161</td>
<td>0</td>
<td>0.2533</td>
<td>0.0736</td>
<td>0.562</td>
<td>0.2346</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: MNL parameters estimated for Monday deliveries. Base utility $\beta_0 = -2.5087$, price sensitivity parameter $\beta_d = -0.0766$. The slot preference $\beta_6$ represents the 6-7am slot, etc. Slot 18 has been used as reference point and has hence been set to zero.

Finally, as inputs for the delivery cost estimation we used $\xi = 0.25$ as the cost per mile, and $\kappa = 80$ totes as van capacity. Stem distances $\rho_a$ from the depot to the center of a given rectangular area $a$ were calculated as the crow flies. Average van speed was assumed to be $v = 25.4$ miles per hour as provided by our industry partner. Deliveries are made in two
sequential shifts (6:00 to 15:00, 15:00 to 23:00). We make the simplifying assumption that vans are ready for delivery at the beginning of each shift; however, we do include the driving cost to and from of each area to the depot in the cost estimation. This is because we are mainly interested in obtaining reasonable delivery cost estimations and are less concerned with feasible schedules; our area-based cost estimation is anyway a very conservative approximation so that feasibility under this scheme is likely to imply feasibility under a more sophisticated routing.

7.2 Clustering

We derived the delivery area definitions as described in §4. The overall delivery region has been divided into 16 bands; this number resulted from experimenting with different numbers to see which results in the least number of areas required to cover all postcodes $z$ with positive average total daily number of orders, denoted by $N_z$. For our data, using 16 bands and applying our ad-hoc area definition method resulted in 111 areas, which is a realistic fleet size for retailers like our industry partner. The resulting rectangular areas and customer locations covered by each of them are shown below in Figure 1. Some areas in the outskirts of London are wide and contain few customer locations, others in central London are very small due to the high density of customers there. Each dot represents a postcode $z$ with a certain underpinning average daily number of orders $N_z$. We assume that these orders are all uniformly distributed over the $m$ delivery slots (needed for the cost estimation framework of Daganzo (1987)), so here we do not use information on which slots were booked historically. Note that some areas span white space because we need to be able to attribute an order from a postcode that is in the delivery region but from which we have not yet received orders to a specific area. The area definition will remain static throughout the booking horizon and therefore needs to be able to accommodate orders from any part of the delivery region.

All experiments were conducted on the basis of this area decomposition. The probability that an arrived customer is from area $a$ is defined by $\mu_a := \sum_{z \in a} \mu_z$, where $z \in a$ is shorthand notation to represent all postcode centers within area $a$. For each postcode $z$, we know the average order size $i_z$ (measured in the number of totes and rounded to the nearest integer) for the delivery day under consideration. The order size distribution $\sigma$ was assumed to be the same for each cluster, and is defined by $\sigma_i := \sum_{z: i_z = i} \mu_z$, for all $i \in \{1, 2, \ldots, 10\}$.

7.3 Simulation results

Using the arrival rates $\lambda$, $\mu_a$ and the order size distributions $\sigma$, we generate 1,000 arrival streams over the entire booking horizon $T$. Order value is derived from the sampled order size multiplied by the average revenue per tote. For these 1,000 demand scenarios, we test the following policies:
Figure 1: Scatter plot of customer location clusters.

- **VS**: Value-based, Static pricing. Delivery charge is based on order value, namely £3 for goods worth £50 or more, and £5 otherwise.

- **F4, F5**: Fixed prices at £4 and £5, respectively, for all time slots.

- **OC-0**: Given a request for delivery of an order of size $i$ to area $a$ (and feasible set of slots $F_{ai}$), assume that the **opportunity cost** is zero and solve (6.1) with $\gamma_s$ replaced by 0 for all slots $s$.

- **OC-C**: Given a request for delivery of an order of size $i$ to area $a$ (and feasible set of slots $F_{ai}$), assume that the **opportunity cost** equals the marginal estimated delivery cost and solve (6.1) with $\gamma_s$ replaced by $\xi(2L_a + W_a/6)$ for the first order in slot $s$, and by $\xi W_a/6$ subsequently.

- **OC-CR**: Given a request for delivery of an order of size $i$ to area $a$ (and feasible set of slots $F_{ai}$), assume that the **opportunity cost** equals $\gamma_s$, i.e. it consists of both cost and revenue effects, and solve (6.1). This is our proposed approach. The underpinning approximate dynamic program uses a step sizes $\alpha$ of 0.0001, 0.00014 and 0.00025 for $\gamma_0$, $\theta$ and $\gamma_s$, respectively (note that different scaling of these variables requires different step sizes). We use $k^{\text{max}} = 3,000$ iterations.

For all policies, feasibility of delivery in a particular area and slot at a particular time $t$ in the booking horizon is being evaluated based on the fixed area definition as described in §6. Customers’ choices are sampled from the MNL model in dependence of the delivery charges that we limit to the interval [-£10, £10] as in Yang et al. (2016). We follow their approach of projecting optimal prices onto this interval; so the resulting price set will not necessarily be optimal any more.
We remark that $OC-0$ is a myopic policy that assumes that there are no delivery capacity constraints and no delivery cost; accordingly this policy should do well in areas where expected demand is much lower than capacity and where delivery costs are low. Policy $OC-C$ should do better in that it accounts for delivery cost, and overall is expected to work well if again expected demand is much less than capacity.

Final delivery costs are calculated in a two-stage process: we first assume that indeed each vehicle serves one area only and insert as many accepted orders as possible into the single-cluster delivery route using a greedy insertion heuristic. If any orders remain unserved, we then try in a second stage to insert these orders into the delivery routes of vans in adjacent areas that still have available capacity. If there are still orders left unserved, then we add a fixed penalty cost of £5 for each of them (corresponding to a standard second class delivery via the mail service).

We report the results over all delivery areas in Table 2 for different scalings of the arrival rate $\lambda$ so as to gain insights into how robust the profit improvements of our proposed policy $OC-CR$ over the benchmark $VS$ is with respect to changes in demand intensity. For each scaling scenario, the approximate dynamic program is only solved once and the resulting parameters $\vec{\gamma}$ are used in the $OC-CR$ policy in all simulations.

Table 2 provides many insights: first and most importantly, we observe that $OC-CR$ performs consistently the best over all scenarios with improvements over the benchmark in the range of 2.2%–2.5%. Secondly, we always observe $OC-0 < OC-C < OC-CR$ as one would expect since we incorporate increasingly more accurate estimates of the opportunity cost.

In particular, $OC-CR$ is significantly and consistently better than $OC-C$ in all scenarios, which demonstrates the value of incorporating the impact of future profit opportunities from orders in the opportunity cost rather than just the estimated marginal delivery cost. $OC-CR$ achieves these profit improvements despite collecting fewer orders than the other two dynamic policies, and at a higher cost per order. This is due to our focus on optimizing total profit rather than minimizing costs. The algorithm anticipates future order values and in which time slot they are likely to occur, so that the pricing can influence demand accordingly.

All dynamic policies attract higher average order value compared to static policies, but overall profitability of $OC-0$ and $OC-C$ may suffer compared to the benchmark even if they attract more orders. This is because they set the delivery charge too low (due to underestimated opportunity cost), and hence the resulting loss in delivery charge income more than compensates for the additional profit from orders to result in an overall loss. This demonstrates the importance of the revenue stream from delivery pricing (in the absence of other potential forms of control such as slot availability control). Dynamic pricing per se does not improve profitability over simple fixed-price policies; its success hinges on good opportunity cost estimates that capture both marginal delivery cost and future order displacement cost.

The smaller demand relative to available capacity (i.e. the smaller the scaling parameter),
Table 2: “#Deliv” is the average number of deliveries under the respective policy, “TotalCost” the average total delivery cost, “MeanCost” is TotalCost/#Deliv, “MeanPrice” the average delivery charge, “MeanValue” the average order value in terms of profit before delivery costs, “TotalProfit” the average total profit after distribution, “StdDev” is the standard deviation of profits, and “Gap” is the percentage gap to the total profit achieved by policy VS. All percentage improvements that are statistically significant at the 95% level are indicated by an asterisk.

<table>
<thead>
<tr>
<th>Scaling</th>
<th>Policy</th>
<th>#Deliv</th>
<th>TotalCost</th>
<th>MeanCost</th>
<th>MeanPrice</th>
<th>MeanValue</th>
<th>TotalProfit±StdDev</th>
<th>Gap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>VS</td>
<td>2408</td>
<td>4942</td>
<td>2.05</td>
<td>3.01</td>
<td>32.62</td>
<td>80856±48.84</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>F4</td>
<td>2334</td>
<td>4874</td>
<td>2.09</td>
<td>4</td>
<td>32.61</td>
<td>80581±52.33</td>
<td>-0.34</td>
</tr>
<tr>
<td>0.9</td>
<td>F5</td>
<td>2562</td>
<td>5159</td>
<td>1.95</td>
<td>3.01</td>
<td>32.59</td>
<td>89236±48.37</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>F4</td>
<td>2576</td>
<td>5088</td>
<td>1.98</td>
<td>4</td>
<td>32.6</td>
<td>89198±49.84</td>
<td>-0.04</td>
</tr>
<tr>
<td>1.1</td>
<td>F5</td>
<td>2793</td>
<td>5284</td>
<td>1.89</td>
<td>4</td>
<td>32.56</td>
<td>96810±48.69</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>F4</td>
<td>2793</td>
<td>5284</td>
<td>1.89</td>
<td>4</td>
<td>32.56</td>
<td>96772±51.04</td>
<td>0.21</td>
</tr>
<tr>
<td>1.2</td>
<td>F5</td>
<td>2982</td>
<td>5461</td>
<td>1.83</td>
<td>4</td>
<td>32.52</td>
<td>102884±46.19</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>F4</td>
<td>2982</td>
<td>5461</td>
<td>1.83</td>
<td>4</td>
<td>32.52</td>
<td>103470±47.36</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>OC-0</td>
<td>3015</td>
<td>5524</td>
<td>1.83</td>
<td>4</td>
<td>32.52</td>
<td>103728±47.74</td>
<td>0.82</td>
</tr>
<tr>
<td></td>
<td>OC-C</td>
<td>2936</td>
<td>5556</td>
<td>1.82</td>
<td>0.01</td>
<td>34.29</td>
<td>95356±40.38</td>
<td>-1.26</td>
</tr>
<tr>
<td></td>
<td>OC-CR</td>
<td>2802</td>
<td>5357</td>
<td>1.91</td>
<td>2.61</td>
<td>34.55</td>
<td>98745±46.29</td>
<td>2.25</td>
</tr>
<tr>
<td></td>
<td>VS</td>
<td>3052</td>
<td>5534</td>
<td>1.81</td>
<td>3.01</td>
<td>32.52</td>
<td>102884±46.19</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>F4</td>
<td>2982</td>
<td>5461</td>
<td>1.83</td>
<td>4</td>
<td>32.52</td>
<td>103470±47.36</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>OC-0</td>
<td>3015</td>
<td>5524</td>
<td>1.83</td>
<td>4</td>
<td>32.52</td>
<td>103728±47.74</td>
<td>0.82</td>
</tr>
<tr>
<td></td>
<td>OC-C</td>
<td>2936</td>
<td>5556</td>
<td>1.82</td>
<td>0.01</td>
<td>34.29</td>
<td>95356±40.38</td>
<td>-1.26</td>
</tr>
<tr>
<td></td>
<td>OC-CR</td>
<td>2802</td>
<td>5357</td>
<td>1.91</td>
<td>2.61</td>
<td>34.55</td>
<td>98745±46.29</td>
<td>2.25</td>
</tr>
<tr>
<td></td>
<td>VS</td>
<td>3177</td>
<td>5706</td>
<td>1.86</td>
<td>2.76</td>
<td>34.53</td>
<td>105231±43.99</td>
<td>2.28</td>
</tr>
<tr>
<td></td>
<td>F4</td>
<td>2982</td>
<td>5461</td>
<td>1.83</td>
<td>4</td>
<td>32.52</td>
<td>103470±47.36</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>OC-0</td>
<td>3100</td>
<td>5542</td>
<td>1.79</td>
<td>-0.09</td>
<td>34.27</td>
<td>100413±38.59</td>
<td>-2.4</td>
</tr>
<tr>
<td></td>
<td>OC-C</td>
<td>2970</td>
<td>5521</td>
<td>1.86</td>
<td>2.76</td>
<td>34.53</td>
<td>105231±43.99</td>
<td>2.28</td>
</tr>
<tr>
<td></td>
<td>OC-CR</td>
<td>2970</td>
<td>5521</td>
<td>1.86</td>
<td>2.76</td>
<td>34.53</td>
<td>105231±43.99</td>
<td>2.28</td>
</tr>
</tbody>
</table>

the better OC-0 and OC-C perform because the true opportunity costs moves towards the marginal delivery cost (note that with demand ≪ capacity, it is unlikely that future orders will be displaced by an order). Accordingly, we expect that if demand is being scaled down further, then at some point OC-C and OC-CR will perform similarly well.

We emphasize that the observation that OC-C can be worse than VS does not contradict the findings of Yang et al. (2016) who propose a policy that approximates opportunity cost only with estimated marginal delivery cost and who observe that this policy improves over the same fixed price policy. Their opportunity cost estimation relies a computationally more intensive way of estimating the marginal delivery cost. This is likely to produce better results than OC-C and the static pricing policies, but it will make it difficult to be implemented in a real-time decision making environment. We limit our comparison to policies that we deem suitable for real-time decision making in large-scale applications.
Figure 2 depicts how the delivery charges vary with order size when using the \textit{OC-CR} policy. The average delivery charge of \textit{OC-CR} was calculated over all requests with the same order size (expressed in the number of totes) for the scenario with scaling parameter of 1. As one would expect, the larger the order, the lower the charges tend to be — in fact, for larger orders we often even charge negative delivery fees, i.e. discounts. The graph reflects that the algorithm differentiates the value of orders and adjusts pricing accordingly.

The opportunity cost estimates \( \gamma_{as} \) of \textit{OC-CR} can be interpreted as the average profit value that slot \( s \) has in \( a \); in other words, we don’t want to have a customer book that slot from whom we are not making at least \( \gamma_{as} \) profit from the order and the delivery charge combined. The estimates take the popularity of time slots into consideration, as well as the likelihood that a future order will be displaced by accepting an order into a time slot. Figure 3 shows that popular slots (9am-10am, noon-1pm, 7-10pm) receive high opportunity cost estimates in areas close to the depot. The latter is located in London, so delivery areas close to the depot (i.e. within around 30 miles) have a denser customer population than more remote ones and hence demand for peak time delivery slots is likely to reach full capacity.

Figure 3 also illustrates that remote areas have very low opportunity cost estimates across all slots. This may be somewhat counterintuitive since they also have high stem driving cost associated with them, which is taken into account by the \textit{OC-CR}. However, demand in these areas is much smaller than available van capacity, therefore the opportunity cost is mostly influenced by delivery cost and not so much by order displacement cost. The latter has often the biggest impact on \( \gamma_{s} \), and accordingly \textit{OC-CR} produces low estimates for remote areas. The delivery charges will accordingly tend to be lower than in busy areas so as to attract...
more orders, a feature of our approach that can help to develop rural markets. Note that
delivery charges are one of the main inhibitors of online shopping; “26% of consumers who
have either stopped or are shopping less for groceries online said they are doing so because of
higher delivery charges” as stated by Mintel (2016).

The fact that the opportunity cost estimates $\gamma_{as}$ are close to zero in remote areas also
means that $OC-0$ should give similar results like $OC-CR$ for these areas. Therefore, one could
consider to use the simpler policy $OC-0$ in these areas so as to further reduce computational
effort.

In terms of computational effort, the parameter estimation via approximate dynamic
programming for all area clusters takes around 15 to 17 minutes (depending on the scaling
of the arrival rate) in total on a standard desktop PC. More importantly, the speed of the
online feasibility and pricing decisions that need to be made at the arrival time of a customer
booking request. The critical threshold for practical implementation has been stated to
be at 100 milliseconds by our industrial partner; our method achieves it on average in 0.4
microseconds.

In summary, our findings regarding the research questions listed at the outset of this
section are as follows:

1. Realistic number of areas: The clustering approach results in a realistic number of areas
for our industry data set.
2. Consistent improvement of \textit{OC-CR}: We observe profit improvements of over 2\% across all demand scenarios over the benchmark policy \textit{VS}. This suggests that the method returns stable improvements relative to uncertain demand intensity. Dynamic pricing policies \textit{OC-O} and \textit{OC-C} can perform worse than static pricing because they may set prices too low if the opportunity cost is influenced much by order displacement cost (especially when capacity is tight).

3. Computational speed: The crucial online calculations underpinning the feasibility check and pricing decisions are virtually instantaneous owing to the simplistic routing model and efficient price optimization.

4. Managerial insights: Although one might intuitively expect that delivery charges in areas distant from the depot should be higher than area nearby, our approach is designed in a way that the best pricing policy is to keep charges low in remote areas so as to increase overall demand there. This is beneficial also with respect to perceived fairness; customers would not accept to be penalized for living further away from the company’s depot.

Charges should reflect the value of a delivery time slot both in terms of marginal delivery cost and future order displacement cost. Dynamic pricing may perform much worse than static pricing if the opportunity cost does not include order displacement costs, especially when demand is similar to or exceeds capacity.

If profit is the overall objective, performance should not be measured in terms of cost per order, number of deliveries made or on average order profit (before delivery). The delivery charges can make a substantial contribution to overall profit.

8 Conclusions and future research

We propose a new method of controlling demand via delivery time slot pricing in attended home delivery management. The focus is on development of an approach that is suitable for industry-scale implementation. To that end, we exploit a relatively simple yet effective way of approximating the delivery cost by decomposing the overall delivery problem into a collection of smaller, area-specific problems. This cost estimation serves as an input to an approximate dynamic programming method which provides estimates of the opportunity cost of accepting a given customer booking into a specific time slot. These estimates depend on the area and on the delivery time slot under consideration.

Using real, large-scale industry data, we estimate a demand model including a multinomial logit choice model, and show in simulation studies that we can improve profits by over 2\% in all instances relative to using an order-value dependent fixed price policy. These improvements are achieved despite having made strong assumptions on the delivery cost.
estimation that were needed to reduce computational runtime to a degree that allows real-time decision making. Our approach provides quantitative insight into the importance of incorporating expected future order displacement cost into the opportunity cost estimation alongside marginal delivery costs.

There are some limitations of our study that warrant further research. First, it would be insightful to conduct a simulation study where the final delivery cost incurred at the end of each simulated booking horizon is based on a industry-standard vehicle routing solution. Secondly, our proposed approach uses opportunity cost estimates that do not depend on time of booking within the booking horizon nor on the level of orders accepted; Meissner and Strauss (2012) show that incorporating this intuitive dependence can further improve the results at the cost of significantly increased computational burden. Our model could take time-dependence to some extent into account by re-solving several times over the booking horizon with the most recent information on accepted orders, and thereby we would obtain opportunity cost estimates that may change over time. Thirdly, one would expect a better value function approximation to return stronger results, such as a piecewise linear approximation. Finally, it would be desirable to include choices between adjacent delivery days in addition to choice between delivery time slots. As we remarked in §6, a nested logit model could be used to that end, at least as long the resulting pricing problem can still be solved sufficiently quickly.

References


W. Zachary Rayfield, Paat Rusmevichientong, and Huseyin Topaloglu. Approximation methods for pricing problems under the nested logit model with price bounds. Cornell University, September 2013.