

Convexity and Feedback for Delivery Time Slot Pricing via Approximate Dynamic Programming

Denis Lebedev, Kostas Margellos, Paul Goulart

Abstract—We consider the revenue management problem of finding profit-maximising prices for delivery time slots in the context of attended home delivery. This multi-stage optimal control problem admits a dynamic programming formulation that is intractable for realistic problem sizes due to the so-called “curse of dimensionality”. Therefore, we study three approximate dynamic programming algorithms both from a control-theoretical perspective and numerically. Our analysis is based on real-world data, from which we generate multiple scenarios to stress-test the robustness of the pricing policies to errors in model parameter estimates. Our theoretical analysis and numerical benchmark tests indicate that one of these algorithms, namely gradient-bounded dynamic programming, dominates the others with respect to computation time and profit-generation capabilities of the pricing policies that it generates.

Index Terms—Approximate dynamic programming, revenue management, attended home delivery.

I. INTRODUCTION

Online grocery sales have been on the rise for the past few years. U.S. households are predicted to spend up to \$133.8 billion per year according to GlobalData [13] on online grocery shopping. However, one of the main holdbacks for growth of online supermarkets is the increased cost of home delivery compared with the logistics of brick-and-mortar supermarkets [6]. Moreover, another logistical problem for online supermarkets is that they have to fulfil *attended* home delivery, i.e. to deliver groceries to customers in pre-agreed delivery time windows. To this end, customers are asked to select a delivery time window as part of the purchase on the sales website. From the side of the company, this poses an optimisation question: How should one optimally adjust prices for delivery slots over time to maximise profits, by taking into account how customers respond to price changes and how customer choice affects delivery costs? We denote this as the *revenue management problem in attended home delivery*.

Broadly speaking, revenue management problems in attended home delivery can be viewed as optimal control problems. The dynamics of customers choosing delivery time windows on the booking website form the *plant* that we seek to control. We can measure the customer choice behaviour by keeping track of placed orders, which we can use as *state feedback*. An optimal control law would then use information from the states to update delivery slot prices, which serve as *control inputs* to our plant as shown schematically in Fig. 1.

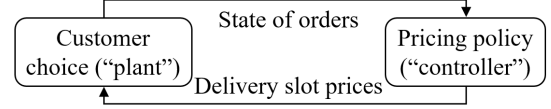


Fig. 1: A feedback control view of the pricing problem.

In principle, the exact state of orders is high-dimensional, e.g. since it represents locations of *all* customers and their chosen delivery time slot. For industry-sized problems the number of states becomes prohibitively large to compute the optimal pricing policy exactly. Therefore, simplified models have been proposed in the literature; see [15] for an overview.

In this paper, we focus on the state-space representation of [18], [19]. In this set-up, we split the delivery area into several sub-areas, each of which is served by a single delivery vehicle. We can then solve the optimal delivery slot pricing problem for each delivery sub-area separately. In that case, the dimensionality of the state is the number of delivery time slots of any delivery day. The state of orders can then be thought of as a vector whose entries represent the number of deliveries for every delivery time slot in a particular sub-area.

For this problem, [18] proposes a dynamic programming (DP) formulation and an approximate DP scheme. Their algorithm approximates the exact value function of the DP as an affine function in the vector of states. In this paper, we show that this method results in an open loop controller, thus motivating the use of non-linear value function approximations, which provide state feedback. In fact, [18], [7] suggest to explore non-linear approximate value functions. The recently developed approximate DP algorithm in [20] was derived outside the attended home delivery literature and it provides such non-linear approximate value functions. However, we show that this algorithm imposes difficulties in the computation of the optimal control policy, since it requires the solution of a non-convex optimisation problem.

Finally, we show our gradient-bounded DP algorithm from [11] overcomes the limitations of the other two algorithms, since it provides non-linear value function approximations, which can be computed using convex optimisation. We demonstrate the its efficacy in numerical examples based on data from [18], in which we benchmark the performance of all three algorithms when the model parameters are known exactly and when they are corrupted by estimation errors. This comparative analysis is new for this problem formulation and hence, it complements the numerical studies on attended home delivery conducted on different formulations, e.g. in [7].

The structure of the remaining paper is as follows: We define

Submitted for review on 14/07/2020. Research is supported by SIA Food Union Management. Corresponding author: Denis Lebedev.

The authors are with the Department of Engineering Science, University of Oxford, Oxford, OX1 3PJ, United Kingdom (e-mail: {denis.lebedev, kostas.margellos, paul.goulart}@eng.ox.ac.uk).

the optimisation problem and its DP formulation in Section II. Section III presents a general, sample-based approximate DP algorithm and we explain how the three algorithms considered in this paper are special cases of it. In that section, we also elaborate on the main theoretical limitations of the first two algorithms and how the third overcomes these. Since the profits generated by all three algorithms are random variables, we explain how we quantify their profit-generation performance in Section IV. We then demonstrate how the gradient-bounded DP algorithm dominates the two other algorithms by means of numerical examples in Section V. Finally, we conclude the paper and provide directions for future research in Section VI.

Notation: Given some $s \in \mathbb{N}$, let 1_s be a column vector of all zeros apart from the s^{th} entry, which equals 1. We define the convention that 1_0 is a vector of zeros. Let $\mathbf{1}$ denote a vector of ones. Let $\langle \cdot, \cdot \rangle$ denote the standard inner product. Let \mathbb{E} denote the expectation operator. Let $\Pr(\cdot)$ denote the probability of its argument. Let $\mathbb{1}(\cdot)$ denote the indicator function.

II. PROBLEM STATEMENT

A. Multi-stage optimal control problem formulation

Revenue management in attended home delivery can be formulated as the following multi-stage optimal control problem for any delivery sub-area served by a single delivery vehicle: Customers are assumed to be allowed to make bookings in a finite time horizon and there is only a finite number of times that the online vendor can change delivery slot prices. Therefore, we consider a finite and discrete time horizon $T := \{1, 2, \dots, \bar{t}\}$. There is an additional time step $\bar{t} + 1$, at which no bookings happen anymore, which we will use to define the terminal condition of the problem. Suppose that the delivery day is split into n delivery time slots. Denote the set of delivery time slots by $S := \{1, 2, \dots, n\}$. As mentioned in Section I, we focus on an aggregated state-space representation, where for any time step $t \in T \cup \{\bar{t} + 1\}$ we define a state vector $x_t \in X \subset \mathbb{Z}^n$, whose entries are the number of orders placed in the respective delivery time slots. The set X is defined by the maximum state vector \bar{x} , i.e. $X := \{x_t \in \mathbb{Z}^n \mid 0 \leq x_t \leq \bar{x}\}$. For any $t \in T$, we define the delivery charge vector $d_t := [d_{1,t}, d_{2,t}, \dots, d_{n,t}]^\top$. Let the set of admissible delivery charge vectors be $D := \{d_t \mid d_{s,t} \in [\underline{d}, \bar{d}] \cup \{\infty\} \text{ for all } s \in S\}$.

For any $s \in S$, define the transition probability between two states x_t and $x_{t+1} = x_t + 1_s$ under delivery price vector d_t as $P_s(d_t)$, where we require $P_s(d_t) \geq 0$ for all $(s, d_t) \in S \times D$. We impose that $\sum_{s \in S} P_s(d_t) < 1$, such that the probability of the customer not choosing any slot is defined as $P_0(d_t) = 1 - \sum_{s \in S} P_s(d_t)$. This implies that transitions from x_t to x_{t+1} are only possible in the positive direction and by at most a unit step along one dimension. Furthermore, we restrict $P_s(d_t) = 0$ if $x_t + 1_s \notin X$, i.e. we do not allow unfeasible orders. Such models are typical for order-taking processes (see [1], [16], [18], [19]). Note that multi-product pricing, e.g. in [5], is stated in terms of inventory, which depletes over time, rather than orders, which accumulate. We assume that customer choice follows a multinomial logit model, like in [5], [18], [19], i.e.

$$P_s(d_t) := \frac{\exp(\beta_c + \beta_s + \beta_d d_{s,t})}{\sum_{k \in S} \exp(\beta_c + \beta_k + \beta_d d_{k,t}) + 1}, \quad (1)$$

for all $(s, d) \in S \times D$, where $\beta_c \in \mathbb{R}$ denotes a constant offset, $\beta_s \in \mathbb{R}$ represents a measure of the popularity for all delivery slots and $\beta_d < 0$ is a parameter for the price sensitivity. Note that the no-purchase utility is normalised to zero, i.e. for the no-purchase “slot” $s = 0$, we have a no-delivery “charge” $d_{0,t} = 0$, such that $\beta_c + \beta_0 + \beta_d d_{0,t} = \beta_c + \beta_0 = 0$ and hence, the 1 in the denominator of (1) arises from $\exp(\beta_c + \beta_0) = 1$. Furthermore, note that the constant offset β_c is not necessary, since it can be absorbed in the $\{\beta_s\}_{s \in S \cup \{0\}}$ parameters. However, β_c is often kept in practice to normalise one of the $\{\beta_s\}_{s \in S \cup \{0\}}$ parameters to zero (see e.g. [18]). Finally, we define an average revenue per order $r \in \mathbb{R}$, an expected customer arrival rate (on the booking system) per time step $\lambda \in (0, 1]$ and an approximate delivery cost function $C : \mathbb{Z}^n \rightarrow \mathbb{R} \cup \infty$, which we assume is Lipschitz continuous. Infinite delivery costs indicate infeasible states. We construct a multi-stage optimal control problem of the following form:

$$\begin{aligned} \max_{\{d_t \in D\}_{t=1}^{\bar{t}}} \mathbb{E} & \left[-C(x_{\bar{t}+1}) + \sum_{t \in T} \langle x_{t+1} - x_t, d_t + r \rangle \right], \\ \text{subject to } & x_{t+1} = x_t + \xi_t, \text{ for all } t \in T, \quad x_1 = 0, \end{aligned} \quad (2)$$

where \mathbb{E} is the expectation operator that is associated with the probability distribution of $\xi_t \in \{0, 1\}^n$, defined as follows: For all $(s, t) \in S \times T$, $\xi_t = 1_s$ with probability $\lambda P_s(d_t)$ and $\xi_t = 0$ with probability $1 - \sum_{s \in S} \lambda P_s(d_t)$. From an economic perspective, the objective value is the total expected operational contribution margin, i.e. revenue from sales and delivery charges minus delivery costs. For simplicity, we will refer to this as the *expected profit* that we seek to maximise.

B. Dynamic programming formulation

The above multi-stage optimal control problem is stage-wise independent, which makes it possible to derive the following DP recursion, analogously to [18], by introducing the value function $V_t : \mathbb{Z}^n \rightarrow \mathbb{R} \cup -\infty$, which represents the expected profit-to-go for any state-time pair $(x, t) \in X \times T$ as shown in (3). We assume that $V_t(x) = -\infty$ for all infeasible states $x \notin X$. Notice that we have dropped subscripts t for x and d to simplify notation and since the time step is evident from the value function. Furthermore, we adopt the convention that when for any $s \in S$, it happens that $d_s = \infty$ and $V_{t+1}(x + 1_s) = -\infty$, we have $P_s(d)(r + d_s + V_{t+1}(x + 1_s)) = 0$. This corresponds to the additional profit of accepting an unavailable slot, which is mathematically undefined in (3), but practically it is zero. To represent the DP in (3) more compactly, we define the Bellman operator \mathcal{T} through the relationship $V_t = \mathcal{T}V_{t+1}$, for all $t \in T$. It is not possible to solve the DP in (3) by direct computation of the value function for realistic problem instances due to the prohibitively large number of states. However, given an approximate value function, one can find approximately optimal prices relatively easy for the multinomial logit model using a Newton root search [5].

$$V_t(x) := \max_{d \in D} \left\{ \lambda \sum_{s \in S} P_s(d) (r + d_s + V_{t+1}(x + 1_s)) + \left(1 - \lambda \sum_{s \in S} P_s(d) \right) V_{t+1}(x) \right\} \quad \forall (x, t) \in X \times T, \quad V_{\bar{t}+1}(x) = -C(x) \quad \forall x \in X \quad (3)$$

III. GENERAL, SAMPLE-BASED APPROXIMATION

A key to solving the revenue management problem in attended home delivery is to approximate the value function in (3) effectively. A popular strategy, not only for this problem (see [7], [18]), but also for other stochastic multi-stage problems (see [12], [14]), is to use a sample-based approach and to refine the value function along states that are likely to be visited under the approximately optimal decision policy. Approximations can then be improved by iterating between generating samples and refining the value function along the obtained sample paths.

We now state a general, sample-based, iterative approximate DP procedure in Algorithm 1. The three algorithms that we investigate in this paper are special cases of this general algorithm and they differ only in step 11 of Algorithm 1. We detail how this step is computed for each of the individual specific algorithms further below.

Algorithm 1 General, sample-based approximation

- 1: Initialise parameters: $X, D, P_s, T, r, \lambda, C$ and i_{\max}
 - 2: Initialise $Q_t^0(x) \leftarrow (\bar{d} + r)\langle \mathbf{1}, \bar{x} - x \rangle - C(\bar{x})$, for all $(x, t) \in X \times T$
 - 3: Initialise $Q_{\bar{t}+1}^0(x) \leftarrow -C(x)$, for all $x \in X$
 - 4: **for** $i \in \{1, 2, \dots, i_{\max}\}$ **do**
 - 5: $x_1^i \leftarrow \mathbf{0}$
 - 6: **for** $t \in \{1, 2, \dots, \bar{t}\}$ **do** ▷ “Forward sweep”
 - 7: $d_t^i \leftarrow d^*(x_t^i)$, the solution of (4)
 - 8: $x_{t+1}^i \leftarrow x_t^i + \text{sample} \{P_s(d_t^i)\}$
 x_{t+1}^i
 - 9: **end for**
 - 10: **for** $t \in \{\bar{t}, \bar{t} - 1, \dots, 1\}$ **do** ▷ “Backward sweep”
 - 11: $Q_t^i \leftarrow \text{update } Q_t^{i-1}$
 - 12: **end for**
 - 13: **end for**
 - 14: **return** $\{Q_t^{i_{\max}} \mid t \in \{1, 2, \dots, \bar{t}\}\}$
-

We first initialise all parameters of the DP in (3) (step 1). Denote the maximum number of iterations by $i_{\max} \in \mathbb{N}$ and let $I := \{0, 1, \dots, i_{\max}\}$. Let the value function approximation be denoted by Q_t^i for all $(i, t) \in I \times T$. We could initialise Q_t^0 to any value as long as that does not violate any assumptions on the approximation algorithm used, as discussed further below. However, one effective way to satisfy the assumptions of all three algorithms considered and to speed up computation, is to initialise Q_t^0 for all $t \in T$ using the unique fixed point of DP, V^* (step 2). In [8], it is shown that the fixed point is given analytically (under mild technical assumptions) as

$$V^*(x) := (\bar{d} + r)\langle \mathbf{1}, \bar{x} - x \rangle - C(\bar{x}), \quad \text{for all } x \in X. \quad (3)$$

Note that V^* is an upper bound to V_t for all $t \in T$, since \mathcal{T} is a monotone operator (see [3, Chapter 3]). Furthermore, notice that the fixed point is affine in x and that the components of the gradient are given by $\bar{d} + r$. Using (3), we can compute the

optimal delivery slot prices at the fixed point, **which is \bar{d}** for all feasible slots. The intuition behind this is that the fixed point corresponds to the limit of the value function as t tends to $-\infty$. Hence, going backwards infinitely many time steps, the probability of selling out the entire delivery capacity tends to one for all prices $d \in D$. **To maximise profits in this scenario, one would** charge customers the maximum admissible delivery charge \bar{d} for all delivery time slots. Finally, we also initialise $Q_{\bar{t}+1}^i(x) := V_{\bar{t}+1}(x) = -C(x)$ for all $(x, i) \in X \times I$ (step 3).

Now fix any iteration $i \in I \setminus \{0\}$. In each “forward sweep”, we solve an approximate version of (3) forward in time by replacing V_t with its approximation Q_t^{i-1} (step 7), i.e.

$$d^*(x_t^i) := \operatorname{argmax}_{d \in D} \left\{ \lambda \sum_{s \in S} P_s(d) (r + d_s + Q_{t+1}^{i-1}(x_t^i + 1_s)) + \left(1 - \lambda \sum_{s \in S} P_s(d) \right) Q_{t+1}^{i-1}(x_t^i) \right\}. \quad (4)$$

Notice that [5, Theorem 1] have shown that for the multinomial choice model, the maximisers are unique for the above expression, hence we use equality in (4). By (4), we compute a suboptimal d_t^i for all $t \in T$ and simulate state transitions by sampling from the transition probability distribution given the approximately optimal decisions (step 8). This defines a sample path x_t^i for all $t \in T \cup \{\bar{t} + 1\}$.

In each “backward sweep”, we update the approximation using the chosen algorithm (step 11). These two sequences – “forward sweep” and “backward sweep” – are repeated for i_{\max} iterations. In the next sections, we describe the exact mechanisms of the three algorithms considered in this paper, making step 11 in Algorithm 1 explicit.

A. Affine value function approximation update

This approach is proposed in [18]. The idea is to approximate the value function by an affine function of the form

$$Q_t^i(x) := \gamma_0^i + (\bar{t} + 1 - t)\theta^i - \sum_s \gamma_s^i x_s, \quad (5)$$

for all $(x, i, t) \in X \times I \times T$ and where γ_s^i , for all $s \in S \cup \{0\}$ and θ^i are **real scalar** parameters, for all $i \in I$. The updating rule in step 11 of Algorithm 1 is a gradient descent step to minimise $(Q_t^i(x_{t+1}^i) - \mathcal{T}Q_{t+1}^{i-1}(x_{t+1}^i))^2$, which thus becomes

$$\begin{aligned} \gamma_0^i &= \gamma_0^{i-1} - \alpha_1 (Q_t^{i-1}(x_{t+1}^i) - \mathcal{T}Q_{t+1}^{i-1}(x_{t+1}^i)) \\ \gamma_s^i &= \gamma_s^{i-1} - \alpha_2 (Q_t^{i-1}(x_{t+1}^i) - \mathcal{T}Q_{t+1}^{i-1}(x_{t+1}^i)) x_{s,t+1}^i, \\ &\quad \text{for all } s \in S \\ \theta^i &= \theta^{i-1} - \alpha_3 (Q_t^{i-1}(x_{t+1}^i) - \mathcal{T}Q_{t+1}^{i-1}(x_{t+1}^i)) (\bar{t} + 1 - t), \end{aligned} \quad (6)$$

where α_1, α_2 and α_3 are (positive) step sizes, which are chosen to be sufficiently small for convergence of the above iterative procedure (see e.g. [2, Lemma 8.2]).

One important observation from a control perspective is that a value function approximation that is affine in x implies that the pricing control will have *no state feedback* for all states x such that $x + 1_s \in X$ for all $s \in S$. To see this, notice that (4) can be re-written in terms of differences $Q_{t+1}^{i-1}(x_t^i) - Q_{t+1}^{i-1}(x_t^i + 1_s)$ for all $s \in S$ as follows:

$$\begin{aligned} & \operatorname{argmax}_{d \in D} \left\{ \lambda \sum_{s \in S} P_s(d) (r + d_s \right. \\ & \quad \left. + Q_{t+1}^{i-1}(x_t^i + 1_s) - Q_{t+1}^{i-1}(x_t^i)) + Q_{t+1}^{i-1}(x_t^i) \right\} \\ &= \operatorname{argmax}_{d \in D} \left\{ \lambda \sum_{s \in S} P_s(d) (r + d_s - \gamma_s^{i-1}) + Q_{t+1}^{i-1}(x_t^i) \right\}, \end{aligned}$$

for all $x \in X$, such that $x + 1_s \in X$ for all $s \in S$, and where we have first substituted for Q_{t+1}^{i-1} from (5). Notice that $Q_{t+1}^{i-1}(x_t^i)$ is independent of d since and thus irrelevant for the argmax operator. Hence, the approximately optimal pricing policy does not depend on the state x_t^i for all $(x, t) \in X \times T$ such that $x + 1_s \in X$ for all $s \in S$. This ultimately means that the affine value function approximation generates a *feedforward* pricing policy, which is incapable of adjusting prices based on changes in the vector of orders. This insight also provides theoretical support for the suggestions of [7] and [18] to explore non-linear value function approximations: The preceding discussion shows that **non-linear value functions** make it possible to include state feedback in the pricing policy.

B. Non-linear stochastic dual dynamic programming update

In contrast to the affine value function update above, non-linear stochastic dual DP generates non-linear value function approximations, which by the above discussion make it possible to include state feedback in the pricing policy. Similarly to [20] and [21], this update is computed in step 11 of Algorithm 1 as $Q_t^i \leftarrow \min\{H^*, Q_t^{i-1}\}$, where the minimum is taken pointwise and the so-called Lagrange dual cut H^* is defined as $H^*(x) := v^* - \langle \mu^*, x_{t+1}^i - x \rangle$, for all $x \in X$ and where

$$\begin{aligned} v^* := \min_{\mu \in \mathcal{M}} \max_{d \in D, z \in X} & \left\{ \lambda \sum_{s \in S} P_s(d) (r + d_s + Q_{t+1}^{i-1}(z + 1_s)) \right. \\ & \left. + \left(1 - \lambda \sum_{s \in S} P_s(d) \right) Q_{t+1}^{i-1}(z) + \langle \mu, x_{t+1}^i - z \rangle \right\} \quad (7) \end{aligned}$$

and where μ^* is the minimiser of (7). In [20, Proposition 5], it is shown that the resulting value function approximation is an upper bound to the exact value function if we additionally assume that the initial approximation Q_t^0 is an upper bound to V_t for all $t \in T$. The formulation in [20] is slightly different from (7), since it includes an additional regularisation term. Please see the extended version of this article [?] for the equivalent proof for this formulation.

From a control perspective, the benefit of having a non-linear value function approximation, in comparison with the affine value function approximation from Section III-A, comes at a different cost: The problem of finding the optimal cut coefficients μ in (7) is a non-convex optimisation problem and

there are consequently no guarantees that it can be solved to global optimality. For the particular form of the problem in this paper, we can find the cut coefficients from a reformulation of the problem, which results in a bi-concave objective function, which can be exploited to solve this problem as outlined in [?, Appendix ??]. Nevertheless, since global optimality is required to ensure that the approximate value function constitutes an upper bound on the exact value function, we cannot guarantee that **all approximations are indeed upper bounds on the exact value function**. We illustrate how this may result in computational problems in Section V.

C. Gradient-bounded dynamic programming update

Gradient-bounded DP was introduced in [11] for the specific application of revenue management in attended home delivery. This manifests itself in **two assumptions on the exact value function of the DP that need to be satisfied**:

First, the value function needs to be *concave extensible*. This means that its concave closure $\tilde{V}_t : \mathbb{R}^n \rightarrow \mathbb{R}$, defined as the smallest concave upper bound on the exact value function, coincides with the exact value function for all state-time pairs, i.e. $\tilde{V}_t(x) = V_t(x)$ for all $(x, t) \in X \times (T \cup \{\bar{t} + 1\})$. Second, the exact value function needs to be *submodular*, i.e.

$$V_t(\max(y_1, y_2)) + V_t(\min(y_1, y_2)) \leq V_t(y_1) + V_t(y_2), \quad (8)$$

for all $(y_1, y_2, t) \in X \times X \times (T \cup \{\bar{t} + 1\})$.

These two assumptions result in a particular segmentation of the convex hull of V_t : For any $t \in T$, construct the unique hyperplane H through the set of pairs $(y, V_t(y))$ for all $y \in Y_+(x_{t+1}^i) := \{x_{t+1}^i + 1_s\}_{s \in (S \cup \{0\})}$. Then H is a separating hyperplane, i.e. $H(x) \geq V_t(x)$ for all $x \in X$, where the inequality holds with equality for all $y \in Y_+(x)$. The gradient-bounded DP algorithm exploits this property. We refer the interested reader to [11] for details on the above-mentioned assumptions and to [9] for proofs that these assumptions hold for the revenue management problem under study.

For gradient-bounded DP, let the value function approximation Q_t^i for all $(i, t) \in I \times T$ be the pointwise minimum of a finite number of affine functions, i.e. $Q_t^i(x) := \min_{j \in \{0, 1, \dots, i\}} H_t^j(x)$, for all $x \in X$, where $H_t^j : X \mapsto \mathbb{R}$ describes a hyperplane, i.e. $H_t^j(x) := \langle a_t^j, x \rangle + b_t^j$, for all $x \in X$, with $a_t^j \in \mathbb{R}^n, b_t^j \in \mathbb{R}$ for all $(t, j) \in T \times I$. Furthermore, this approximation is an upper bound on the exact value function, i.e. $Q_t^i(x) \geq V_t(x)$ for all $(x, t, i) \in X \times T \times I$. To this end, it is important to initialise Q_t^0 for all $t \in T$ at an upper bound. The gradient-bounded DP update then ensures that the approximate value functions remain upper bounds to the exact value function for all iterations $i \in I$ as shown in [11, Proposition 1]. In step 11 of Algorithm 1, gradient-bounded DP generates updates for the approximate value function as shown in Algorithm 2 and explained further below.

Fix any iteration $i \in I$. We first check if Q_{t+1}^{i-1} is submodular (see (8)) on the set $Z(x_{t+1}^i) := \{x_{t+1}^i + 1_s + 1_{s'}\}$, for all $(s, s') \in (S \cup \{0\}) \times (S \cup \{0\})$, i.e. if and only if $0 \leq Q_{t+1}^{i-1}(y_1) + Q_{t+1}^{i-1}(y_2) - Q_{t+1}^{i-1}(\min\{y_1, y_2\}) - Q_{t+1}^{i-1}(\max\{y_1, y_2\})$ holds for all $(y_1, y_2) \in Z(x_{t+1}^i) \times Z(x_{t+1}^i)$ (steps 1 and 2). Note that this is not necessarily the case for the approximate value

Algorithm 2 Gradient-bounded dynamic programming update

```

1:  $Z(x_{t+1}^i) \leftarrow \{x_{t+1}^i + 1_s + 1'_s\}_{s \in S \cup \{0\}, s' \in S \cup \{0\}}$ 
2: if  $Q_{t+1}^{i-1}$  is submodular on  $Z(x_{t+1}^i)$  then
3:    $H^* \leftarrow$  unique hyperplane through
      $\{(y, (\mathcal{T}Q_{t+1}^{i-1})(y))\}_{y \in Y_+(x_{t+1}^i)}$ 
4: else
5:    $j^* \in \operatorname{argmin}_{j \in J_{t+1}^{i-1}} \{(\mathcal{T}H_{t+1}^{j-1})(x_{t+1}^i)\}$ 
6:    $H^* \leftarrow \mathcal{T}H_{t+1}^{j^*-1}$ 
7: end if
8:  $Q_t^i \leftarrow \min\{H^*, Q_t^{i-1}\}$ 

```

function, even if the exact value function is submodular. We then distinguish between two cases:

Case I: If Q_{t+1}^{i-1} is submodular on $Z(x_{t+1}^i)$, we locally compute the exact DP stage problem on the set $Y_+(x_{t+1}^i)^{i-1}$, i.e. $\{\mathcal{T}Q_{t+1}^{i-1}(y)\}_{y \in Y_+(x_{t+1}^i)}$, to construct the hyperplane through $\{(y, (\mathcal{T}Q_{t+1}^{i-1})(y))\}_{y \in Y_+(x_{t+1}^i)}$ (step 3).

Case II: If Q_{t+1}^{i-1} is not submodular on $Z(x_{t+1}^i)$, we need to compute a submodular upper bound on Q_{t+1}^{i-1} , which is readily given by the hyperplanes from which Q_{t+1}^{i-1} is constructed. Therefore, we select the hyperplane $H_{t+1}^{j^*-1}$ that minimises the value at the evaluation point x_{t+1}^i , i.e. $Q_t^i = \min\{\mathcal{T}H_{t+1}^{j^*-1}, Q_t^{i-1}\}$, where $j^* \in \operatorname{argmin}_{j \in J_{t+1}^{i-1}} \{(\mathcal{T}H_{t+1}^{j-1})(x_{t+1}^i)\}$ and where J_{t+1}^{i-1} is the set of sup-

porting hyperplanes, i.e. $J_{t+1}^{i-1}(x) := \operatorname{argmin}_{j \in \{0,1,\dots,i-1\}} H_{t+1}^j(x)$, for all $(i, t, x) \in I \times T \times X$ (steps 5 and 6). Therefore, this creates the locally tightest upper bound. Finally, we take the pointwise minimum of the approximate value function at the previous iteration Q_t^{i-1} and the newly created hyperplane H^* to obtain the new value function approximation (step 8).

Similarly to the non-linear stochastic dual DP update from Section III-B and in contrast to the affine value function approximation update from Section III-A, the approximate value function generated by the gradient-bounded DP update is non-linear in x as it is given by the pointwise minimum of affine functions in x . Assuming that the initialiser is an upper bound on the exact value function, which can be satisfied if we choose the fixed point for this purpose (as discussed at the beginning of Section III), the approximate value function is an upper bound to the exact value function as shown in [11, Proposition 1]. Finally, the advantage of gradient-bounded DP over non-linear stochastic dual DP is that only convex optimisation problems need to be solved to compute the update, which makes gradient-bounded DP more resilient against computational stability problems than non-linear stochastic dual DP as shown in Section V.

IV. PROFIT-GENERATION PERFORMANCE CRITERION

Since the profits that all three algorithms generate are random variables, we can quantify their performance with probabilistic guarantees by performing validation runs, i.e. by simulating customer decisions forward in time and pricing

based on the most refined approximate value function. Let the profit that we obtain in each of k_{\max} validation runs be $l_v(k)$ for all $k \in K := \{1, \dots, k_{\max}\}$. Let $[l_-, l_+]$ denote the (finite) support of the distribution of $l_v(k)$ for any $k \in K$. In our case, $l_+ = V^*(0) = (\bar{d} + r)\langle \mathbf{1}, \bar{x} \rangle - C(\bar{x})$, where we use the fixed point from (3) and $l_- = -C(0)$. We then compute the empirical mean \bar{l}_v and empirical standard error σ_v as $\bar{l}_v := 1/k_{\max} \times \sum_{k=1}^{k_{\max}} l_v(k)$ and $\sigma_v := \sqrt{1/k_{\max} \times \sum_{k=1}^{k_{\max}} (l_v(k) - \bar{l}_v)^2}$. For any of the three algorithms considered, we can then quantify the performance of a pricing policy using the maximum of the two following bounds, presented in [10, Proposition 7], which state the expected profit which can be guaranteed with confidence $(1 - \alpha) \in (0, 1)$ after observing k_{\max} validation samples. Recall that \mathbb{E} denotes the expectation operator and that $\Pr(\cdot)$ denotes the probability of its argument.

Proposition 1. Fix any significance level $\alpha \in (0, 1)$. Then $\Pr(\mathbb{E}\bar{l}_v \geq l^*) \geq 1 - \alpha$, for all $l^* \in \{l_B, l_D\}$, where:

$$l_B := \bar{l}_v - \sqrt{\frac{2\sigma_v \ln(\frac{2}{\alpha})}{k_{\max}}} - \frac{7(l_+ - l_-) \ln(\frac{2}{\alpha})}{3(k_{\max} - 1)} \text{ and} \quad (9a)$$

$$l_D := \int_{l=0}^{\infty} 1 - \min \left\{ 1, F_K(l) + \sqrt{\frac{\ln(\frac{1}{\alpha})}{2k_{\max}}} \right\} dl, \quad (9b)$$

where F_K denotes the empirical cumulative distribution function of $\{l_v(k)\}_{k \in K}$, i.e. $F_K(l) := k_{\max}^{-1} \sum_{k \in K} \mathbb{1}(l_v(k) \geq l)$, where $\mathbb{1}$ denotes the indicator function.

The proof can be found in [10, Appendix A.6]. Strictly speaking, we assume non-positive fixed costs $C(0)$ for the bounds to hold (see [10, Assumption 5]). Hence, we set $l_- = -C(0) = 0$ for our case study. However, since the fixed costs do not impact the pricing policy of any of the algorithms considered, these become irrelevant for our analysis.

V. NUMERICAL EXAMPLES

In the following two sections, we present the numerical analysis that compares the three value function approximation algorithms stated and analysed in Section III. To this end, we generate particular instances of the revenue management problem in attended home delivery presented in Section II. We use the parameter values in [18] as a base case and modify these parameters to simulate various scenarios and conduct a sensitivity analysis. In Section V-A, we analyse the performance of the three algorithms under the assumption that the model parameters are known accurately. Then in Section V-B, we simulate how well the algorithms perform when they are trained on the data in V-A, but being tested on scenarios, where the customer choice parameters differ from the model.

A. Exact model analysis

In this section, we adapt the numerical case study parameters from [18] to arrive at the set-up defined in Table I. We use the same step sizes for the affine value function update (see (6) in Section III-A) as in [18], i.e. $\alpha_1 := 0.0001$, $\alpha_2 := 0.00025$, $\alpha_3 := 0.00014$. Additionally, we consider two parameters, which we vary as described further below.

12, its profit generation performance decreases over time as can be seen in Fig. 4. This might appear counter-intuitive at first, but is in line with our theoretical analysis from Section III-B: We conjecture that this is due to the difficulty of finding global maxima of non-convex optimisation problems. If the algorithm converges to a local maximum, the value function approximation is no longer guaranteed to be an upper bound on the exact value function. Over time, this then leads to a compounding of errors caused by suboptimality, i.e. instead of increasing, the expected profit decreases as more cuts are added to the approximate value function. A practical way to circumvent this problem is to compute the expected profit with 99% confidence after each iteration and to pick the iteration which produces the maximum expected profit with 99% confidence. In the example of Fig. 4, the best solution is found after the first iteration – the optimal policy is dominated by pricing all slots at the maximum charge \bar{d} for all time steps, since the high demand factor 8 almost guarantees that all slots will be sold out for any choice of [admissible](#) prices. Hence, over time invalid cuts accumulate, which results in a degradation of the profit performance.

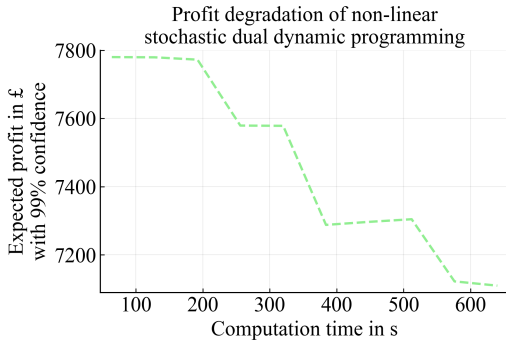


Fig. 4: Expected profits of non-linear stochastic dual DP (demand factor 8, slot capacity 12) decrease over time.

Comparing the expected profits obtained between the three algorithms, we observe that gradient-bounded DP always generates the highest expected profit with 99% confidence or is within 1% of the optimal value, when the demand factor is so high that demand saturates and all three algorithms perform very similarly. This saturation behaviour can be seen in Fig. 5, where we also show that for demand factors 1 and lower, gradient-bounded DP produces between 10 and 15% more expected profit with 99% confidence than affine value function approximation. At the same time, gradient-bounded DP performs similarly to non-linear stochastic dual DP in most scenarios. However, gradient-bounded DP generates up to 10% more profit than non-linear stochastic dual DP for small demand factor 1/8 and capacity 6 as well as for large demand factor 8 across all slot capacities.

Overall, we conclude that gradient-bounded DP performs best in this exact model experiment, because it outperforms affine value function approximation in terms of profit generation while being similarly fast and at the same time, gradient-bounded DP is more than four times faster than non-linear stochastic dual DP while generating very similar profit.

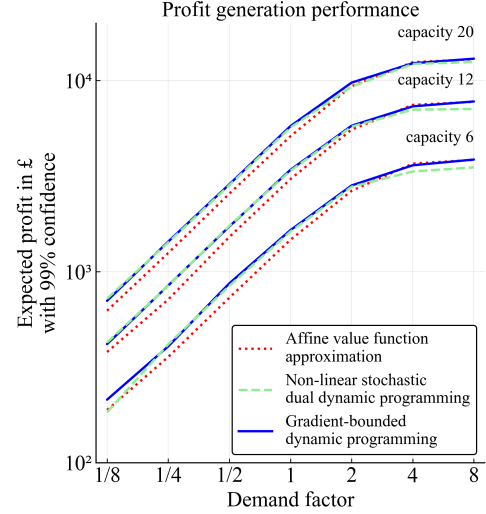


Fig. 5: Expected profits with 99% confidence of the three algorithms against demand factor, for all delivery slot capacities.

B. Parameter sensitivity analysis

We assume in the previous section that the customer choice model parameters β_c, β_d and $\{\beta_s\}_{s \in S}$, are known exactly, which is not the case in practice. Hence, we now investigate how well the pricing policies obtained by the three algorithms in the previous section perform on perturbed models. Consider the case that customer preferences across slots and prices are misspecified. To model this, we now corrupt the parameter estimates β_c, β_d and $\{\beta_s\}_{s \in S \cup \{0\}}$ by additive Gaussian noise. This choice of distribution is justified because, in the limit as the number of data points used for estimating the customer choice parameters tends to infinity, the error between estimated and true customer choice parameter value vector is a Gaussian with zero mean [17, Chapter 8.6].

We consider three scenarios in which we [set the](#) variance of the Gaussian to $\sigma^2 \in \{0.01, 0.1, 1\}$. With these noise levels, we sample the sets of customer choice parameters, which we hold fixed for all validation runs in this experiment. Note that we do not have to worry about normalising the probability distribution, since the multinomial choice model is normalised for all possible parameter values. The numerical values used in our analysis are documented in [the extended version of this article \[?, Appendix ??\]](#). We [show](#) how the profit generation performance of the three algorithms degrades in comparison with the ideal scenario in the previous section in Fig. 6.

As we see in Fig. 6(b) and (c), non-linear stochastic dual DP and gradient-bounded DP are both robust against [model](#) uncertainty. Only for $\sigma^2 = 1$, there is a substantial degradation in profit generation performance. In contrast, Fig. 6(a) shows that even small uncertainties in the customer choice model have substantial negative impact on the affine value function approximation algorithm, decreasing expected profit with 99% confidence by about an order of magnitude for $\sigma^2 = 1$. We conjecture that this is due to the lack of state feedback in the affine value function approximation solution as detailed in Section III-A: For any $t \in T$, the suggested optimal slot price vector is identical for all x strictly inside the set of feasible

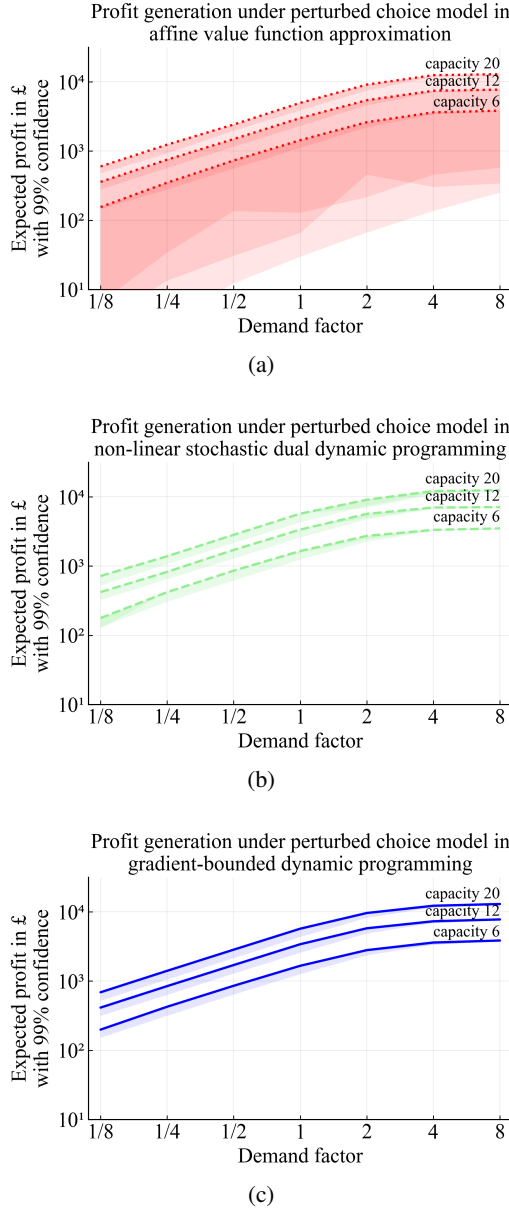


Fig. 6: Expected profits under perturbed model parameters. Lines: $\sigma^2 = 0.01$. Shaded regions: σ^2 increases to 0.1 and 1.

states X , because the affine value function approximation has constant gradient for all these points. Since the other two algorithms both generate a piecewise affine approximate value function, gradients and hence optimal delivery prices vary depending on the particular state-time pair $(x, t) \in X \times T$.

Overall, we conclude that both gradient-bounded DP and non-linear stochastic dual DP increase their relative profit-generation advantage over affine value function approximation under imperfect customer choice model parameter estimates. identified limitations in the affine value function approxima-

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we analysed three approximate dynamic programming algorithms to find approximately optimal delivery slot prices in the revenue management problem in attended home delivery. From a control-theoretical perspective, we

tion algorithm and the non-linear stochastic dual dynamic programming algorithm. We showed in numerical examples how gradient-bounded dynamic programming can overcome these limitations. Possible directions for future work include investigating the numerical performance of these algorithms for other revenue management problems and extending the promising gradient-bounded dynamic programming approach to other customer decision models than multinomial logit.

REFERENCES

- [1] K. Asdemir, V. S. Jacob, and R. Krishnan. Dynamic pricing of multiple home delivery options. *European Journal of Operational Research*, 196(1):246 – 257, 2009.
- [2] A. Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.
- [3] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. II*. Athena Scientific, 4th edition, 2012.
- [4] C. F. Daganzo. Modeling distribution problems with time windows: Part i. *Transportation Science*, 21(3):171–179, 1987.
- [5] L. Dong, P. Kouvelis, and Z. Tian. Dynamic pricing and inventory control of substitute products. *Manufacturing & Service Operations Management*, 11(2):317–339, 2009.
- [6] N. Galante, E. G. López, and S. Monroe. The future of online grocery in europe. *McKinsey & Company*, pages 22–31, 2013. Available: https://www.mckinsey.com/~media/McKinsey/Industries/Retail/Our%20Insights/The%20future%20of%20online%20grocery%20in%20Europe/The_future_of_online_grocery.ashx.
- [7] S. Koch and R. Klein. Route-based approximate dynamic programming for dynamic pricing in attended home delivery. *European Journal of Operational Research*, 2020.
- [8] D. Lebedev, P. Goulart, and K. Margellos. A concave value function extension for the dynamic programming approach to revenue management in attended home delivery. In *2019 18th European Control Conference (ECC)*, pages 999–1004, June 2019.
- [9] D. Lebedev, P. Goulart, and K. Margellos. Dynamic programming for optimal delivery time slot pricing. Technical report, 2019. Available: <https://arxiv.org/abs/1910.11757>.
- [10] D. Lebedev, P. Goulart, and K. Margellos. Gradient-bounded dynamic programming for submodular and concave extensible value functions with probabilistic performance guarantees. Technical report, 2020. Available: <https://arxiv.org/pdf/2006.02910.pdf>.
- [11] D. Lebedev, P. Goulart, and K. Margellos. Gradient-bounded dynamic programming with submodular and concave extensible value functions. In *21st IFAC World Congress 2020*, 2020. Available: <https://arxiv.org/pdf/2005.11213.pdf>.
- [12] M. V. F. Pereira and L. M. V. G. Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52(1):359–375, May 1991.
- [13] N. Saunders. Online grocery & food shopping statistics. <https://www.onespace.com/blog/2018/08/online-grocery-food-shopping-statistics/>, 2018. Accessed on 16/06/2020.
- [14] A. Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209(1):63–72, feb 2011.
- [15] A. K. Strauss, R. Klein, and C. Steinhardt. A review of choice-based revenue management: Theory and methods. *European Journal of Operational Research*, 2018.
- [16] M. Suh and G. Aydin. Dynamic pricing of substitutable products with limited inventories under logit demand. *IIE Transactions*, 43(5):323–331, 2011.
- [17] K. E. Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, 2 edition, 2009.
- [18] X. Yang and A. K. Strauss. An approximate dynamic programming approach to attended home delivery management. *European Journal of Operational Research*, 263(3):935–945, 2017.
- [19] X. Yang, A. K. Strauss, C. S. M. Currie, and R. Eglese. Choice-based demand management and vehicle routing in e-fulfillment. *Transportation Science*, 50(2):473–488, 2016.
- [20] S. Zhang and X. A. Sun. Stochastic dual dynamic programming for multistage stochastic mixed-integer nonlinear optimization. Technical report, 2019. Available: <https://arxiv.org/pdf/1912.13278.pdf>.
- [21] J. Zou, S. Ahmed, and X. A. Sun. Stochastic dual dynamic integer programming. *Mathematical Programming*, 175(1):461–502, May 2019.