OPERATIONS RESEARCH
Vol. 62, No. 6, November–December 2014, pp. 1394–1415
ISSN 0030-364X (print) | ISSN 1526-5463 (online)

http://dx.doi.org/10.1287/opre.2014.1322

Information Relaxations, Duality, and Convex Stochastic Dynamic Programs

David B. Brown, James E. Smith
Fuqua School of Business, Duke University, Durham, North Carolina 27708
{dbbrown@duke.edu, jes9@duke.edu}

We consider the information relaxation approach for calculating performance bounds for stochastic dynamic programs (DPs). This approach generates performance bounds by solving problems with relaxed nonanticipativity constraints and a penalty that punishes violations of these nonanticipativity constraints. In this paper, we study DPs that have a convex structure and consider gradient penalties that are based on first-order linear approximations of approximate value functions. When used with perfect information relaxations, these penalties lead to subproblems that are deterministic convex optimization problems. We show that these gradient penalties can, in theory, provide tight bounds for convex DPs and can be used to improve on bounds provided by other relaxations, such as Lagrangian relaxation bounds. Finally, we apply these results in two example applications: first, a network revenue management problem that describes an airline trying to manage seat capacity on its flights; and second, an inventory management problem with lead times and lost sales. These are challenging problems of significant practical interest. In both examples, we compute performance bounds using information relaxations with gradient penalties and find that some relatively easy-to-compute heuristic policies are nearly optimal.

Subject classifications: dynamic programming; information relaxations; network revenue management; lost-sales inventory models.
Area of review: Decision Analysis.
History: Received October 2013; revision received May 2014; accepted September 2014. Published online in *Articles in Advance* October 29, 2014.

1. Introduction

Dynamic programming is a powerful framework for studying stochastic systems where decisions are made sequentially. Unfortunately, in practice, the complexity of dynamic programming models tends to grow rapidly with the number of variables considered. When optimal policies are difficult to identify, many researchers study stochastic dynamic systems using various forms of heuristic policies and evaluate the performance of these heuristics using Monte Carlo simulation. However, without knowing the optimal policy, it is hard to know how much better one might do with some other heuristic. We can always experiment with different forms of heuristics or different parameters for a given heuristic, but such experimentation can be time consuming and it is difficult to know when to stop. In these settings, it can be useful to have easy-to-compute upper bounds on the performance of an optimal policy: if the performance of a given heuristic is close to this upper bound, we might conclude that the heuristic is "good enough" and decide to not invest more effort in trying to improve the heuristic.

In this paper, we consider the information relaxation approach for calculating performance bounds for stochastic dynamic programs (DPs), following Brown, Smith, and Sun (2010; hereafter BSS). In BSS, bounds are generated by (1) relaxing the nonanticipativity constraints that require the decision maker (DM) to make decisions based only on the information available at the time the decision is made and (2) incorporating penalties that punish violations of these nonanticipativity constraints. For example, in this paper we will consider a network revenue management problem where an airline must decide whether to sell low-fare tickets long before the departure date or to reserve capacity (seats) for possible high-fare passengers who may request tickets later; the problem is complicated by the fact that many itineraries consume capacity on more than one flight. In a perfect information relaxation, we assume the DM knows exactly which requests will arrive before deciding whether to accept any request. In this case, the revenue management problem is a deterministic optimization problem that can be formulated as a linear program where one maximizes revenue subject to the capacity constraints. By randomly generating scenarios of requests and repeatedly solving this deterministic "inner problem," we obtain an upper bound on the performance with an optimal policy, namely, the value with perfect information. However, without any penalty for using this additional information, these perfect information bounds are often quite weak. Informally, we say a penalty is dual feasible if it does not punish any policy that is nonanticipative; the penalties may, however, punish policies that violate the nonanticipativity constraints.

The challenge in practice is to find penalties that provide good bounds and lead to inner problems that are easy to

solve. In BSS, we studied general DPs and presented a general approach for constructing "good" penalties (informally, dual feasible with no slack) from differences of approximate value functions. Here we focus on DPs that have a convex structure. With such DPs, penalties constructed from differences of approximate value functions may lead to inner problems that are not convex and may be difficult to solve. To address these computational challenges, we consider "gradient penalties" that are based on first-order linear approximations of approximate value functions and lead to inner problems that are deterministic convex optimization problems.

In this paper, we study the theoretical properties of these gradient penalties and demonstrate their use in two example applications that are of significant practical interest. In terms of theory, we show that, given the appropriate convex structure in the DP and the approximate value functions, these gradient penalties are dual feasible, there exists a gradient penalty that yields a tight zero-variance bound, and these gradients can be used to improve on bounds provided by other dynamic programming relaxations (e.g., Lagrangian relaxations). We first consider the case where the approximate value functions used to generate penalties are differentiable and then consider the more delicate case where these value functions may not be differentiable. Nondifferentiable value functions arise frequently: even if the reward functions and constraints are differentiable, the value functions may be nondifferentiable if the binding constraints on actions change in some scenarios. Both of our example applications involve value functions that are not differentiable.

The first example we consider is the network revenue management problem mentioned earlier. Topaloglu (2009) formulates the network revenue management problem as a stochastic DP that is difficult (or impossible) to solve. He then develops a Lagrangian relaxation approximation that can be solved and uses this to generate heuristics and performance bounds. We show how information relaxations and gradient penalties can be used to improve upon the Lagrangian relaxation bounds. In the numerical examples we consider, the performance bounds are significantly improved and, with these new bounds, we can show that the heuristics are within 1% of an optimal policy.

The second example we consider is an inventory management problem with lead times for delivery and where sales are lost if adequate inventory is not on hand. We follow Zipkin (2008a, b), and earlier researchers who formulated this problem as a stochastic dynamic program and consider heuristic policies. Specifically, we study a myopic heuristic and use it to generate gradient penalties. Here again, in our numerical examples, we obtain reasonably tight bounds and show that this myopic heuristic is nearly optimal.

In the remainder of this section, we provide a brief literature review. In §2, we introduce the general framework and some key results from BSS (2010). In §3, we consider the case where the DPs or approximating models have a convex structure and develop the theory of gradient penalties. In §§4 and 5, we consider the network revenue management and lost-sales inventory examples. Section 6 provides a few concluding remarks.

1.1. Literature Review

BSS (2010) builds on earlier work providing methods for calculating bounds for valuing American options developed by Rogers (2002), Haugh and Kogan (2004), and Andersen and Broadie (2004), among others. BSS generalized these methods from stopping problems to more general stochastic DPs. Rogers (2007) also considers information relaxation techniques for Markov decision problems. There is also a literature in stochastic programming that considers relaxations of nonanticipativity constraints with Lagrange multiplier penalties; see, e.g., Rockafellar and Wets (1976) and Shapiro et al. (2009). BSS compares and contrasts these related approaches in more detail. Of course, there are other approaches for generating performance bounds, including Lagrangian relaxations (see, e.g., Hawkins 2003 or Adelman and Mersereau 2008). We will integrate information relaxation bounds and Lagrangian bounds in the network revenue management example of §4.

Information relaxation bounds have been applied in a number of settings. For example, there are many applications in valuing American options, following the early work of Rogers (2002), Haugh and Kogan (2004), and Andersen and Broadie (2004). BSS (2010) provides examples in inventory management and option pricing. Lai et al. (2010) use information relaxation bounds in their study of heuristics for managing natural gas storage assets; see also Nadarajah et al. (2014). Devalkar et al. (2011) use this approach in their study of an integrated model procurement, processing, and trading of commodities in a multiperiod setting.

Brown and Smith (2011) consider an application of information relaxation bounds in portfolio management with transaction costs. There we used linear penalties based on a frictionless model that ignores transaction costs, as well as other approximate value functions. Here we develop the theory of this approach more fully and consider nondifferentiable as well as differentiable approximate value functions; we provide a more detailed comparison in an online appendix (available as supplemental material at http://dx.doi.org/10.1287/opre.2014.1322). Others have also used linear penalties in recent work. For example, Haugh et al. (2014) study a dynamic portfolio optimization problem like that considered in Brown and Smith (2011), but incorporating capital gains taxes; they provide information relaxation bounds with linear penalties based on a model that ignores taxes. Secomandi (2014) further studies policies for managing natural gas storage assets, using information relaxation bounds with linear penalties derived from a model that ignores inventory adjustment costs and losses and also

ignores limits on the rate of injection or withdraw of natural gas from storage. Haugh and Lim (2012) study linear penalties in linear-quadratic control problems.

2. The Basic Framework and Results

2.1. The Primal Problem

Uncertainty in the DP is described by a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is the set of possible outcomes or *scenarios* ω, $\mathcal{F}$ is a σ-algebra that describes the set of possible events, and $\mathbb{P}$ is a probability measure describing the likelihood of each event.

Time is discrete and indexed by $t = 0, \ldots, T$. The DM's state of information evolves over time and is described by a filtration $\mathbb{F} = (\mathcal{F}_0, \ldots, \mathcal{F}_T)$, where the σ-algebra $\mathcal{F}_t$ describes the DM's state of information at the beginning of period t; we will refer to $\mathbb{F}$ as the *natural filtration*. The filtrations must satisfy $\mathcal{F}_t \subseteq \mathcal{F}_{t+1} \subseteq \mathcal{F}$ for all $t < T$ so the DM does not forget what she once knew. We let $\mathbb{E}[-] = \mathbb{E}[-|\mathcal{F}]_0$ denote expectations conditioned on this initial state of information. We will assume that $\mathcal{F}_0 = \{\varnothing, \Omega\}$, so the DM initially "knows nothing" about the outcome of the uncertainties; this implies $\mathbb{E}[-]$ is a constant.

A function (or random variable) f defined on Ω is *measurable* with respect to a σ-algebra $\mathcal{F}_t$ (or $\mathcal{F}_t$-measurable) if for every Borel set R in the range of f, we have $\{\omega: f(\omega) \in R\} \in \mathcal{F}_t$. We can interpret f being $\mathcal{F}_t$-measurable as meaning the value of f depends only on the information known in period t. A sequence of functions $(f_0, \ldots, f_T)$ is *adapted* to a filtration $\mathbb{F}$ (or $\mathbb{F}$-adapted) if each function f_t is measurable with respect to $\mathcal{F}_t$.

The DM must choose an action a_t in period t from a set A_t; we let $A(\omega) \subseteq A_0 \times \cdots \times A_T$ denote the set of all feasible action sequences $\mathbf{a} = (a_0, \ldots, a_T)$ given scenario ω. The DM's choice of actions is described by a *policy* α that selects a sequence of actions $\mathbf{a}$ in A for each scenario ω in Ω (i.e., α: $\Omega \to A$). To ensure the DM knows the feasible set when choosing actions in period t, we assume that the set of actions available in period t depends on the prior actions $(a_0, \ldots, a_{t-1})$ and is $\mathcal{F}_t$-measurable for each set of prior actions. We let $\mathcal{A}$ denote the set of all *feasible* policies, i.e., those that ensure that $\alpha(\omega)$ is in $A(\omega)$.

In the primal problem, we require the DM's choices to be *nonanticipative* in that the choice of action a_t in period t depends only on what is known at the beginning of period t; that is, we require policies to be adapted to the natural filtration $\mathbb{F}$ in that a policy's selection of action a_t in period t must be measurable with respect to $\mathcal{F}_t$. We let $\mathcal{A}_\mathbb{F}$ be the set of feasible policies that are nonanticipative.

The DM's goal is to select a feasible nonanticipative policy to maximize the expected total reward. The rewards are defined by a $\mathbb{F}$-adapted sequence of reward functions $(r_0, \ldots, r_T)$, where the reward r_t in period t depends on the first $t+1$ actions $(a_0, \ldots, a_t)$ of the action sequence $\mathbf{a}$ and the scenario ω. We let $r(\mathbf{a}, \omega) = \sum_{t=0}^{T} r_t(\mathbf{a}, \omega)$ denote the total reward. The primal DP is then

$$\max_{\alpha \in \mathcal{A}_\mathbb{F}} \mathbb{E}[r(\alpha)]. \tag{1}$$

Here $\mathbb{E}[r(\alpha)]$ could be written more explicitly as $\mathbb{E}[r(\alpha(\omega), \omega)]$, where policy α selects an action sequence that depends on the random scenario ω and the rewards r depend on the action sequence selected by α and the scenario ω. We will typically suppress the dependence on ω and interpret $r(\alpha)$ as a random variable representing the total reward generated under policy α. Also note that we will assume that the maximum in (1) is attained and thus will write "max" in place of "sup" throughout.

It will be helpful to rewrite the primal DP (1) as a Bellman-style recursion in terms of the optimal value functions V_t. We let $\mathbf{a}_t = (a_0, \ldots, a_t)$ denote the sequence of actions up to and including period t. Since the period-t reward r_t depends only on the first $t+1$ actions $(a_0, \ldots, a_t)$, we will write $r_t(\mathbf{a})$ as $r_t(\mathbf{a}_t)$ with the understanding that the actions are selected from the full sequence of actions $\mathbf{a}$; we will use a similar convention for V_t. For $t > 0$, let $A_t(\mathbf{a}_{t-1})$ be the subset of period-t actions A_t that are feasible given the prior choice of actions $\mathbf{a}_{t-1}$: r_t and A_t are both implicitly functions of the scenario ω. We take the terminal value function $V_{T+1}(\mathbf{a}_T) = 0$ and, for $t = 0, \ldots, T$, we define

$$V_t(\mathbf{a}_{t-1}) = \max_{a_t \in A_t(\mathbf{a}_{t-1})} \left\{ r_t(\mathbf{a}_{t-1}, a_t) + \mathbb{E}[V_{t+1}(\mathbf{a}_{t-1}, a_t) | \mathcal{F}_t] \right\}. \tag{2}$$

Here both sides are random variables (and therefore implicitly functions of the scenario ω) and we select an optimal action a_t for each scenario ω.

2.2. Duality Results

In the dual problem, we relax the requirement that the policies be nonanticipative and impose penalties that punish violations of these constraints. We define relaxations of the nonanticipativity constraints by considering alternative information structures. We say that a filtration $\mathbb{G} = (\mathcal{G}_0, \ldots, \mathcal{G}_T)$ is a *relaxation* of the natural filtration $\mathbb{F} = (\mathcal{F}_0, \ldots, \mathcal{F}_T)$ if, for each t, $\mathcal{F}_t \subseteq \mathcal{G}_t$; we abbreviate this by writing $\mathbb{F} \subseteq \mathbb{G}$. $\mathbb{G}$ being a relaxation of $\mathbb{F}$ means that the DM knows more in every period under $\mathbb{G}$ than she knows under $\mathbb{F}$. For example, the perfect information relaxation is given by taking $\mathcal{G}_t = \mathcal{F}$ for all t. We let $\mathcal{A}_\mathbb{G}$ denote the set of feasible policies that are adapted to $\mathbb{G}$. For any relaxation $\mathbb{G}$ of $\mathbb{F}$, we have $\mathcal{A}_\mathbb{F} \subseteq \mathcal{A}_\mathbb{G}$; thus, as we relax the filtration, we expand the set of feasible policies.

The set of penalties Π is the set of functions π that, like the total rewards, depend on actions $\mathbf{a}$ and the scenario ω. As with rewards, we will typically write the penalties as an action-dependent random variable $\pi(\mathbf{a})$ $(= \pi(\mathbf{a}, \omega))$ or a policy-dependent random variable $\pi(\alpha)$ $(= \pi(\alpha(\omega), \omega))$, suppressing the dependence on the scenario ω. We define

the set $\Pi_{\mathbb{F}}$ of *dual feasible penalties* to be those that do not penalize nonanticipative policies in expectation, that is

$$\Pi_{\mathbb{F}} = \{\pi \in \Pi\colon \mathbb{E}[\pi(\alpha)] \leqslant 0 \text{ for all } \alpha \text{ in } \mathcal{A}_{\mathbb{F}}\}. \tag{3}$$

Policies that do not satisfy the nonanticipativity constraints (and thus are not feasible to implement) may have positive expected penalties.

We can obtain an upper bound on the expected reward associated with any nonanticipative policy by relaxing the nonanticipativity constraint on policies and imposing a dual feasible penalty, as stated in the following weak duality lemma from BSS (2010). We repeat the proof here, because it is short and instructive.

Lemma 2.1 (Weak Duality) *If α_F and π are primal and dual feasible, respectively (i.e., $\alpha_F \in \mathcal{A}_{\mathbb{F}}$ and $\pi \in \Pi_{\mathbb{F}}$) and $\mathbb{G}$ is a relaxation of $\mathbb{F}$, then*

$$\mathbb{E}[r(\alpha_F)] \leqslant \max_{\alpha_G \in \mathcal{A}_{\mathbb{G}}} \mathbb{E}[r(\alpha_G) - \pi(\alpha_G)].$$

Proof. With π, α_F, and $\mathbb{G}$ as defined in the lemma, we have

$$\mathbb{E}[r(\alpha_F)] \leqslant \mathbb{E}[r(\alpha_F) - \pi(\alpha_F)] \leqslant \max_{\alpha_G \in \mathcal{A}_{\mathbb{G}}} \mathbb{E}[r(\alpha_G) - \pi(\alpha_G)].$$

The first inequality holds because $\pi \in \Pi_{\mathbb{F}}$ (thus $\mathbb{E}[\pi(\alpha_F)] \leqslant 0$) and the second because $\alpha_F \in \mathcal{A}_{\mathbb{F}}$ and $\mathcal{A}_{\mathbb{F}} \subseteq \mathcal{A}_{\mathbb{G}}$. □

Thus any information relaxation with any dual feasible penalty will provide an upper bound on the expected reward generated by any primal feasible policy.

In this paper, we will focus on the perfect information relaxation, where the set of relaxed policies is the set of all policies $\mathcal{A}$ and actions are selected with full knowledge of the scenario ω. In this case, the weak duality lemma implies that for any α_F in $\mathcal{A}_{\mathbb{F}}$ and π in $\Pi_{\mathbb{F}}$,

$$\begin{aligned}\mathbb{E}[r(\alpha_F)] &\leqslant \max_{\alpha \in \mathcal{A}} \mathbb{E}[r(\alpha) - \pi(\alpha)] \\ &= \mathbb{E}\Big[\max_{\mathbf{a} \in A(\omega)} \{r(\mathbf{a}, \omega) - \pi(\mathbf{a}, \omega)\}\Big].\end{aligned} \tag{4}$$

If we take the penalty $\pi = 0$, this upper bound is the expected value with perfect information.

Note that the upper bound (4) is in a form that is convenient for Monte Carlo simulation: we can estimate the expected value on the right side of (4) by randomly generating scenarios ω and solving a deterministic *inner problem* of choosing a feasible action sequence $\mathbf{a}$ to maximize the penalized objective in scenario ω:

$$\max_{\mathbf{a} \in A(\omega)} \{r(\mathbf{a}, \omega) - \pi(\mathbf{a}, \omega)\}. \tag{5}$$

Here, unlike (1), we need only consider actions for a particular scenario ω and need not consider the nonanticipativity constraints that link actions across scenarios.

2.3. Penalties

BSS (2010) provides a general approach for constructing "good" penalties, based on a set of generating functions. We will show that we can, in principle, generate an optimal penalty using this approach.

Proposition 2.1 (Constructing Good Penalties). *Let $\mathbb{G}$ be a relaxation of $\mathbb{F}$ and let $(w_0, \ldots, w_T)$ be a sequence of generating functions defined on $A \times \Omega$, where each w_t depends only on the first $t+1$ actions $(a_0, \ldots, a_t)$ of $\mathbf{a}$. Define $\pi_t(\mathbf{a}) = \mathbb{E}[w_t(\mathbf{a}) \mid \mathcal{G}_t] - \mathbb{E}[w_t(\mathbf{a}) \mid \mathcal{F}_t]$ and $\pi(\mathbf{a}) = \sum_{t=0}^{T} \pi_t(\mathbf{a})$. Then, for all α_F in $\mathcal{A}_{\mathbb{F}}$, we have $\mathbb{E}[\pi_t(\alpha_F) \mid \mathcal{F}_t] = 0$ for all t, and $\mathbb{E}[\pi(\alpha_F)] = 0$.*

The result implies that the penalties π generated in this way will be dual feasible (i.e., $\mathbb{E}[\pi(\alpha_F)] \leqslant 0$ for α_F in $\mathcal{A}_{\mathbb{F}}$), but is stronger in that it implies the inequality defining dual feasibility (3) holds with equality: i.e., $\mathbb{E}[\pi(\alpha)] = 0$ for all α in $\mathcal{A}_{\mathbb{F}}$. In this case, we say the penalty *has no slack*. A penalty that has slack can certainly be improved by eliminating the slack. Good penalties are thus, by construction, dual feasible with no slack. We refer the reader to BSS for a proof and further discussion of this result.

Taking the information relaxation $\mathbb{G}$ to be the perfect information relaxation and considering a sequence of generating functions $(w_0, \ldots, w_T)$, we can write the dual problem recursively as follows. Take the terminal dual value function to be $\bar{V}_{T+1}(\mathbf{a}_T) = 0$. For $t = 0, \ldots, T$, we have

$$\begin{aligned}\bar{V}_t(\mathbf{a}_{t-1}) = \max_{a_t \in A_t(\mathbf{a}_{t-1})} \big\{&r_t(\mathbf{a}_{t-1}, a_t) - w_t(\mathbf{a}_{t-1}, a_t) \\ &+ \mathbb{E}[w_t(\mathbf{a}_{t-1}, a_t) \mid \mathcal{F}_t] + \bar{V}_{t+1}(\mathbf{a}_{t-1}, a_t)\big\}.\end{aligned} \tag{6}$$

The expected initial value, $\mathbb{E}[\bar{V}_0]$, provides an upper bound on the primal DP (1).

We can construct an optimal penalty using Proposition 2.1 by taking the generating functions to be based on the optimal DP value function given by (2). Specifically, if we take generating functions $w_t(\mathbf{a}) = V_{t+1}(\mathbf{a}_t)$, we obtain an optimal penalty of the form

$$\pi^\star(\mathbf{a}) = \sum_{t=0}^{T} V_{t+1}(\mathbf{a}_t) - \mathbb{E}[V_{t+1}(\mathbf{a}_t) \mid \mathcal{F}_t]. \tag{7}$$

It is easy to show by induction that the dual value functions are equal to the corresponding primal value functions, i.e., $\bar{V}_t = V_t$. This is trivially true for the terminal values (both are zero). If we assume inductively that $\bar{V}_{t+1} = V_{t+1}$, terms cancel and (6) reduces to the expression for V_t given in Equation (2). Thus, with this choice of generating function, we obtain an optimal penalty that we refer to as the *ideal penalty*: the inner problem is equal to V_0 in every scenario and, moreover, the primal and dual problems will have the same sets of optimal policies.

Of course, in practice, we will not know the true value function and cannot construct this ideal penalty. We can instead take the generating function to be the approximate

value functions $\hat{V}_{t+1}$, and consider a penalty function of the form

$$\hat{\pi}(\mathbf{a}) = \sum_{t=0}^{T} \{\hat{V}_{t+1}(\mathbf{a}_t) - \mathbb{E}[\hat{V}_{t+1}(\mathbf{a}_t) \mid \mathcal{F}_t]\}. \tag{8}$$

By Proposition 2.1, this penalty $\hat{\pi}$ is dual feasible with no slack, and leads to a valid upper bound on V_0. The key to obtaining a good bound from such an approximation value function is for the differences in (8) to provide a good approximation of the differences in (7) based on the true value function. For example, in the inventory example of BSS (2010), we often find that penalties based on limited lookahead approximate value functions do well. Though these limited lookahead approximations do not approximate the value functions very well (because they include only a few periods of rewards), they approximate the differences in (7) well.

Although the approximate value function $\hat{V}_{t+1}$ in (8) can be any function satisfying the conditions of Proposition 2.2, we can say more in the case where the approximate value function is an optimal value function for an approximating DP. Specifically, consider a DP defined on the same probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and filtration $\mathbb{F}$ as in the original model (as described in §2.1), but with total rewards $\hat{r}$ instead of r and constraint set $\hat{A}$ instead of A. We say this approximate model is a *relaxation* of the original model if $r(\mathbf{a}, \omega) \leqslant \hat{r}(\mathbf{a}, \omega)$ holds for all $\mathbf{a}$ in $A(\omega)$ and for all ω (i.e., for all actions that are feasible for the original model) and $A(\omega) \subseteq \hat{A}(\omega)$ for all ω; we will abbreviate this by writing $r \leqslant \hat{r}$ and $A \subseteq \hat{A}$, respectively. Because the rewards and feasible sets are no smaller in the relaxed model, it is easy to see that the optimal value in the relaxed model must be an upper bound on the optimal value in the original model, i.e.,

$$V_0 = \max_{\alpha \in \mathcal{A}_{\mathbb{F}}} \mathbb{E}[r(\alpha)] \leqslant \hat{V}_0 = \max_{\alpha \in \hat{\mathcal{A}}_{\mathbb{F}}} \mathbb{E}[\hat{r}(\alpha)], \tag{9}$$

where $\hat{\mathcal{A}}_{\mathbb{F}}$ denotes the set of nonanticipative, feasible policies for the relaxed problem.

What is perhaps not obvious is that the bound based on the penalty (8) from this relaxed value function $\hat{V}_t$ will be tighter than the bound (9) provided by the relaxed model itself. We summarize the results of this section and formalize this last observation in the following proposition.

Proposition 2.2. *Let $\hat{\pi}$ be the penalty given by* (8) *for approximate value functions $\hat{V}_t$.*

(i) Feasibility. *The penalty $\hat{\pi}$ is dual feasible and has no slack.*

(ii) Optimality. *If the approximate value functions $\hat{V}_t$ are the optimal value functions for the original model, then, for every scenario ω,*

$$\max_{\mathbf{a} \in A(\omega)} \{r(\mathbf{a}, \omega) - \hat{\pi}(\mathbf{a}, \omega)\} = V_0.$$

(iii) Improving bounds from other relaxations. *If the value functions $\hat{V}_t$ are the optimal value functions for a relaxed model with $A \subseteq \hat{A}$ and $r \leqslant \hat{r}$, then, for every scenario ω,*

$$\max_{\mathbf{a} \in A(\omega)} \{r(\mathbf{a}, \omega) - \hat{\pi}(\mathbf{a}, \omega)\} \leqslant \hat{V}_0.$$

Proof. Part (i) follows from Proposition 2.2 and part (ii) was established in the discussion preceding the proposition. Part (iii) follows from part (ii): using the result of part (ii) with the relaxed model, we know that, for every scenario ω,

$$\max_{\mathbf{a} \in \hat{A}(\omega)} \{\hat{r}(\mathbf{a}, \omega) - \hat{\pi}(\mathbf{a}, \omega)\} = \hat{V}_0.$$

Since $r(\mathbf{a}, \omega) \leqslant \hat{r}(\mathbf{a}, \omega)$ and $A(\omega) \subseteq \hat{A}(\omega)$, we have

$$\max_{\mathbf{a} \in A(\omega)} \{r(\mathbf{a}, \omega) - \hat{\pi}(\mathbf{a}, \omega)\} \leqslant \max_{\mathbf{a} \in \hat{A}(\omega)} \{\hat{r}(\mathbf{a}, \omega) - \hat{\pi}(\mathbf{a}, \omega)\} = \hat{V}_0. \quad \square$$

As indicated in the proof, the final result follows from the second result in that we can construct an ideal penalty for the relaxed model. We can then improve on the bound from the relaxed model by solving inner problems with the true (rather than relaxed) rewards and constraints.

3. Convex Dynamic Programs and Gradient Penalties

Though the results of §§2.2 and 2.3 hold for all DPs, our focus in this paper will be on the case where the DP or its approximating model has a convex structure. We will assume from now on that the actions are vectors of real numbers, i.e., $\mathbf{a} \in \mathbb{R}^n$ for some finite n, though the feasible set of actions may be restricted to some subset of $\mathbb{R}^n$, e.g., to integer or binary variables. A *convex dynamic program* is a DP where the reward functions $r_t(\mathbf{a}, \omega)$ are concave functions of the actions $\mathbf{a}$ for each ω and the feasible set of actions $A(\omega)$ is convex for each ω. With a convex DP, the primal DP (1) can be viewed as a (large) convex optimization problem with decision variables corresponding to choices of actions $\mathbf{a}$ for each scenario ω and a concave objective function, a convex set of constraints $A(\omega)$ for each scenario, and a large set of equality constraints that link actions across scenarios and represent the nonanticipativity constraints. We can also show that for a convex DP, the optimal value functions V_t given by the Bellman recursion (2) will be concave in actions.[1]

With convex DPs, though the rewards are concave and constraint sets are convex, with penalties like (8) based on approximate (or the true) value function, the penalized objective, $r(\mathbf{a}) - \hat{\pi}(\mathbf{a})$ may not be concave in $\mathbf{a}$ and, consequently, the resulting inner problem (5) may be difficult to solve. A natural way to address this issue is to replace the penalties with a first-order linear approximation. As discussed in §1.1, such linear penalties have been used in several recent applications.

3.1. Gradient Penalties: The Differentiable Case

Assuming the approximate value functions $\hat{V}_t$ are concave and (for now) differentiable in actions, we can take a first-order linear approximation around the nonanticipative (or $\mathbb{F}$-adapted) policy $\hat{\alpha}$:

$$\hat{V}_{t+1}(\mathbf{a}_t) \approx \nabla \hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t)^\top (\mathbf{a}_t - \hat{\boldsymbol{\alpha}}_t) + \hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t),$$

where $\nabla \hat{V}_{t+1}(\mathbf{a}_t)$ denotes the gradient of $\hat{V}_{t+1}(\mathbf{a}_t)$ with respect to the first $t+1$ actions, evaluated at the point $\mathbf{a}_t$ and $\hat{\boldsymbol{\alpha}}_t$ denotes the first $t+1$ actions selected under policy $\hat{\alpha}$. Note that $\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t)$ is a random variable (written more explicitly as $\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t(\omega), \omega)$), the gradients are calculated for each ω, and the resulting approximation is a random variable for each action sequence $\mathbf{a}_t$. We can then use this approximation as a generating function, taking

$$w_t(\mathbf{a}_t) = \nabla \hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t)^\top (\mathbf{a}_t - \hat{\boldsymbol{\alpha}}_t) + \hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t)$$

in Proposition 2.1 to generate the *gradient penalty*:

$$\hat{\pi}_\nabla(\mathbf{a}) = \sum_{t=0}^{T} \Big\{ \big(\nabla \hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) - \mathbb{E}[\nabla \hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) \,|\, \mathcal{F}_t]\big)^\top (\mathbf{a}_t - \hat{\boldsymbol{\alpha}}_t) + \big(\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) - \mathbb{E}[\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) \,|\, \mathcal{F}_t]\big) \Big\}. \quad (10)$$

(We use the assumption that $\hat{\boldsymbol{\alpha}}_t$ is $\mathcal{F}_t$-measurable to move $\hat{\boldsymbol{\alpha}}_t$ outside of the expectation.) This penalty is affine in actions $\mathbf{a}$ and, given a problem with concave rewards and convex action sets, the inner problem (5) with this penalty is a convex optimization problem. The final terms (inside the parentheses) are constant with respect to $\mathbf{a}$ and play the role of control variates: they have zero mean and thus do not affect the expected value in the bound (4). However, these terms may be correlated with the reward terms in (4) and including them in the penalty may help reduce the variance when estimating the bounds (4) using Monte Carlo simulation. We discuss this in more detail in §3.3.

What is striking about these gradient penalties is that the linear approximation, in principle, entails no loss in functionality when working with convex DPs. Just as in Proposition 2.2, the gradient penalties will be dual feasible and have no slack; when working with a convex DP, there exists a gradient penalty that generates a zero variance, tight bound; and, when working with an approximate value function from a relaxed model that is a convex DP, the gradient penalty will improve on the bound given by the relaxed model in every scenario. We formalize these results for the differentiable case as follows; we consider the nondifferentiable case in the next section.

Proposition 3.1. *Suppose the approximate value functions $\hat{V}_t$ are concave in actions and differentiable. Let $\hat{\pi}_\nabla$ be the gradient penalty defined by linearizing $\hat{V}_t$ around a $\mathbb{F}$-adapted policy $\hat{\alpha}$ as in* (10)*.*

(i) Feasibility. *The gradient penalty $\hat{\pi}_\nabla$ is dual feasible and has no slack.*

(ii) Optimality. *If the original model is a convex DP and the approximate value functions $\hat{V}_t$ and policies $\hat{\alpha}$ are the optimal value functions and an optimal policy for this model, then, for every scenario ω,*

$$\max_{\mathbf{a} \in A(\omega)} \big\{ r(\mathbf{a}, \omega) - \hat{\pi}_\nabla(\mathbf{a}, \omega) \big\} = V_0.$$

(iii) Improving bounds from other relaxations. *If the approximate value functions $\hat{V}_t$ are the optimal value functions for a relaxed model that is a convex DP with $A \subseteq \hat{A}$ and $r \leqslant \hat{r}$, and $\hat{\alpha}$ is an optimal policy for this relaxed model, then, for every scenario ω,*

$$\max_{\mathbf{a} \in A(\omega)} \big\{ r(\mathbf{a}, \omega) - \hat{\pi}_\nabla(\mathbf{a}, \omega) \big\} \leqslant \hat{V}_0.$$

Proof. This result is a special case of Proposition 3.2. □

Note that the last two results above hold "pathwise" (i.e., for every scenario ω), which implies the dual bounds given by taking expectations over scenarios,

$$\mathbb{E}\Big[\max_{a \in A(\omega)} \big\{ r(\mathbf{a}, \omega) - \pi_\nabla(\mathbf{a}, \omega) \big\} \Big],$$

will be equal to V_0 in part (ii) and less than or equal to $\hat{V}_0$ in part (iii). As in Proposition 2.2, the last result follows from the second result.

Although we defer the formal proof of Proposition 3.1 until we consider the more general case that does not assume differentiability, it is helpful to provide some intuition about the proof of the second part of the proposition. To simplify the discussion, we will assume that the action choices are unconstrained. Consider a gradient penalty $\hat{\pi}_\nabla$ defined by linearizing $\hat{V}_t$ around policy $\hat{\alpha}$, as in (10). If we omit the terms inside the parentheses that are constant in actions (which, as discussed earlier, serve as control variates), the inner problem (5) for a given scenario reduces to

$$\begin{aligned} &\max_{\mathbf{a}} \left\{ \sum_{t=0}^{T} r_t(\mathbf{a}_t) - \big(\nabla \hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) - \mathbb{E}[\nabla \hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) | \mathcal{F}_t]\big)^\top (\mathbf{a}_t - \hat{\boldsymbol{\alpha}}_t) \right\} \\ &= \max_{\mathbf{a}} \left\{ \sum_{t=0}^{T} r_t(\mathbf{a}_t) - \left(\begin{pmatrix} \nabla \hat{V}_t(\hat{\boldsymbol{\alpha}}_{t-1}) \\ \mathbf{0} \end{pmatrix} - \mathbb{E}[\nabla \hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) | \mathcal{F}_t] \right)^\top \cdot (\mathbf{a}_t - \hat{\boldsymbol{\alpha}}_t) \right\}. \end{aligned} \quad (11)$$

Here, in rearranging terms, we use the fact that $\hat{V}_{T+1} = 0$ and thus $\nabla \hat{V}_{T+1} = \mathbf{0}$. In this expression, $\nabla \hat{V}_t$ has dimension corresponding to $\mathbf{a}_{t-1}$ and, hence, its gradient needs to be padded with a $\mathbf{0}$ of the dimension of a_t to match the dimensionality of $\nabla \hat{V}_{t+1}$, which corresponds to $\mathbf{a}_t$.

Now, if $\hat{\alpha}$ is an optimal policy and $\hat{V}_t$ are the optimal value functions and the choices of actions are unconstrained, we know that

$$\begin{aligned} \hat{V}_t(\hat{\boldsymbol{\alpha}}_{t-1}) &= r_t(\hat{\boldsymbol{\alpha}}_t) + \mathbb{E}[\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) | \mathcal{F}_t] \\ &= \max_{a_t} \big\{ r_t(\hat{\boldsymbol{\alpha}}_{t-1}, a_t) + \mathbb{E}[\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_{t-1}, a_t) | \mathcal{F}_t] \big\}, \end{aligned} \quad (12)$$

and the first-order conditions for optimality and the "envelope theorem" imply

$$\begin{pmatrix} \nabla \hat{V}_t(\hat{\boldsymbol{\alpha}}_{t-1}) \\ \mathbf{0} \end{pmatrix} = \nabla r_t(\hat{\boldsymbol{\alpha}}_t) + \mathbb{E}[\nabla \hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) \mid \mathcal{F}_t]. \tag{13}$$

Using this "consistency condition," we can rewrite the reduced inner problem (11) as

$$\max_{\mathbf{a}} \left\{ \sum_{t=0}^{T} r_t(\mathbf{a}_t) - \nabla r_t(\hat{\boldsymbol{\alpha}}_t)^\top (\mathbf{a}_t - \hat{\boldsymbol{\alpha}}_t) \right\}, \tag{14}$$

which, given the concavity of r_t, is minimized by taking $\mathbf{a}_t = \hat{\boldsymbol{\alpha}}_t$ for all t. Thus, the reduced inner problem (11) yields an optimal value of $\sum_{t=0}^{T} r_t(\hat{\boldsymbol{\alpha}}_t)$. Using this and incorporating the control variate terms that were omitted in the reduced inner problem (11), the inner problem (5) in this case is

$$\begin{aligned} &\max_{\mathbf{a}} \{r(\mathbf{a}) - \pi_\nabla(\mathbf{a})\} \\ &\quad = \sum_{t=0}^{T} \left\{ r_t(\hat{\boldsymbol{\alpha}}_t) - (\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) - \mathbb{E}[\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) \mid \mathcal{F}_t]) \right\} \\ &\quad = \hat{V}_0 + \sum_{t=0}^{T} \left\{ r_t(\hat{\boldsymbol{\alpha}}_t) - \hat{V}_t(\hat{\boldsymbol{\alpha}}_{t-1}) + \mathbb{E}[\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) \mid \mathcal{F}_t] \right\} \\ &\quad = \hat{V}_0. \end{aligned}$$

Here, in the second equality, we use $\hat{V}_{T+1} = 0$ and rearrange terms. In the third equality, we use the fact that $\hat{\boldsymbol{\alpha}}_t$ is optimal (so the first equality in (12) holds). Thus using a gradient penalty based on the optimal value function will generate a zero-variance tight bound.

In practice, with gradient penalties based on approximate value functions, the optimality conditions (12) and (13) may be approximated and the quality of the resulting bounds will depend on the quality of the approximations. As with the nongradient penalties and discussed following Equation (8), the key for a gradient penalty to provide good bounds is for the linear approximations of the approximate value functions to approximate the differences in the true value functions, i.e.,

$$\begin{aligned} &V_{t+1}(\mathbf{a}_t) - \mathbb{E}[V_{t+1}(\mathbf{a}_t) \mid \mathcal{F}_t] \\ &\quad \approx \left(\nabla \hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) - \mathbb{E}[\nabla \hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) \mid \mathcal{F}_t]\right)^\top (\mathbf{a}_t - \hat{\boldsymbol{\alpha}}_t) \\ &\qquad + \left(\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) - \mathbb{E}[\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) \mid \mathcal{F}_t]\right). \end{aligned}$$

With convex DPs, it suffices to construct a linear approximation that performs well in the vicinity of the true optimal solution. In particular, it is important for the difference in gradients to approximate $(\nabla V_{t+1}(\boldsymbol{\alpha}_t) - \mathbb{E}[\nabla V_{t+1}(\boldsymbol{\alpha}_t) \mid \mathcal{F}_t])$ well. In this case, the optimal solutions in the inner problem will match or closely approximate those of the true optimal solutions. Errors in the constant terms $(V_{t+1}(\boldsymbol{\alpha}_t) - \mathbb{E}[V_{t+1}(\boldsymbol{\alpha}_t) \mid \mathcal{F}_t])$ are less important, as they will average zero when calculating the bounds.

As an example of a setting where we can apply the results of Proposition 3.1 directly, we can point to Brown and Smith (2011), where we study a dynamic portfolio optimization problem with transaction costs. There, the approximate model is a portfolio optimization model that ignores transactions costs; this is a relaxation of the original model and is not difficult to solve to optimality. These frictionless value functions are differentiable and hence the results of Proposition 3.1 apply and, in particular, by part (iii), we can calculate information relaxation bounds that certainly improve on the bound provided by the frictionless model. However, the construction of penalties in Brown and Smith (2011) was different and, as discussed in the online appendix, we could have done somewhat better applying the approach of Proposition 3.1 instead.

3.2. Gradient Penalties: The General Case

As discussed in the introduction, the assumption that the approximate value functions are differentiable is a strong assumption that is not satisfied in many applications, including the revenue management and lost-sales applications considered in §§4–5. When the approximate value functions are not differentiable, the gradients are not uniquely defined and the choice of gradients may affect the quality of bounds.

We define the differential as the set of all gradients for a concave function f at a point x as

$$\partial f(x) = \{g\colon f(y) \leqslant f(x) + g^\top (y - x) \text{ for all } y\}.$$

(With convex functions, these are typically called subgradients and subdifferentials; with concave functions, these are sometimes called supergradients and superdifferentials. We will omit the "sub" and "super.") We note several basic properties of these differentials in the following lemma.

Lemma 3.1. *Let f, f_1, and f_2 be real-valued concave functions*:

(i) *$\partial f(x)$ is convex and nonempty.*
(ii) *If f is differentiable at x, then $\partial f(x) = \{\nabla f(x)\}$.*
(iii) *If $\gamma \geqslant 0$, then $\partial(\gamma f(x)) = \gamma \partial f(x)$.*
(iv) *$\partial(f_1(x) + f_2(x)) = \partial f_1(x) + \partial f_2(x)$.*
(v) *If $g(x) = f(Mx + b)$, then $\partial g(x) = M^\top \partial f(Mx + b)$.*
(vi) *$f(x^*) = \max_x f(x)$ if and only if $0 \in \partial f(x^*)$.*
(vii) *Let $f(x, \omega)$ be concave in x for each ω. Then $\partial \mathbb{E}[f(x, \omega)] = \mathbb{E}[\partial f(x, \omega)]$.*
(viii) *Let $f^*(y) = \max_x f(x, y)$ and let $x^*(y)$ denote an optimal solution, i.e., such that $f^*(y) = f(x^*(y), y)$. Then $(0, g) \in \partial f(x^*(y), y)$ if and only if $g \in \partial f^*(y)$.*

In (iv), the sum on the right side is a set-wise (or Minkowski) sum; similarly the expectation on the right in (vii) is a probability-weighted set-wise sum or integral. These properties of the gradients and differentials also apply to extended real-valued convex functions, provided the relevant differentials are not empty: $\partial f(x)$ will be nonempty if x is in the relative interior of the domain of f. The first six

results are standard results with proofs given in, for example, Bertsekas et al. (2003). The seventh result is proven in Bertsekas (1973). We refer to the last property as the "stacking gradients" result; it will play a role analogous to the consistency condition (13) for the differentiable case. We provide a proof in the appendix.

We define a generalized version of gradient penalties as follows. Consider approximate value functions $\hat{V}_t$ that are concave in actions and a policy $\hat{\alpha}$, which will serve as the basis for the approximation. Let $\boldsymbol{\delta} = (\boldsymbol{\delta}_0, \ldots, \boldsymbol{\delta}_T)$ be a *gradient selection*, where, for each t, $\boldsymbol{\delta}_t$ selects an element of the differential $\partial \hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t)$ for each scenario. Here $\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t)$ and $\boldsymbol{\delta}_t$ are random variables and we require $\boldsymbol{\delta}_t(\omega) \in \partial \hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t(\omega), \omega)$. Given a gradient selection $\boldsymbol{\delta}$, we take the generating functions in Proposition 2.2 to be

$$w_t(\mathbf{a}_t) = \boldsymbol{\delta}_t^\top(\mathbf{a}_t - \hat{\boldsymbol{\alpha}}_t) + \hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t)$$

and we have a generalized gradient penalty

$$\hat{\pi}_\partial(\mathbf{a}) = \sum_{t=0}^{T} \big\{ (\boldsymbol{\delta}_t - \mathbb{E}[\boldsymbol{\delta}_t \,|\, \mathscr{F}_t])^\top (\mathbf{a}_t - \hat{\boldsymbol{\alpha}}_t) + \big(\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) - \mathbb{E}[\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t) \,|\, \mathscr{F}_t]\big) \big\}. \tag{15}$$

Note that this definition of a gradient penalty reduces to that of Equation (10) if the approximate value functions $\hat{V}_t$ are differentiable as there is unique gradient in each scenario.

We can now generalize Proposition 3.1 as follows.

Proposition 3.2. *Suppose the approximate value functions $\hat{V}_t$ are concave in actions. Let $\hat{\pi}_\partial$ be the gradient penalty defined in* (15) *by linearizing $\hat{V}_t$ around a $\mathbb{F}$-adapted policy $\hat{\alpha}$ using gradient selection $\boldsymbol{\delta}$.*

(i) Feasibility. *For any gradient selection $\boldsymbol{\delta}$, the gradient penalty $\hat{\pi}_\partial$ is dual feasible and has no slack.*

(ii) Optimality. *If the original model is a convex DP and the approximate value function $\hat{V}_t$ and policy $\hat{\alpha}$ are optimal value functions and policies for this model, then there exists a gradient selection $\boldsymbol{\delta}$, such that for every scenario ω,*

$$\max_{\mathbf{a} \in A(\omega)} \{ r(\mathbf{a}, \omega) - \hat{\pi}_\partial(\mathbf{a}, \omega) \} = V_0.$$

Moreover, for any gradient selection $\boldsymbol{\delta}$, we have $V_0 \leqslant \max_{\mathbf{a} \in A(\omega)} \{ r(\mathbf{a}, \omega) - \hat{\pi}_\partial(\mathbf{a}, \omega) \}$ for every ω.

(iii) Improving bounds from other relaxations. *If the approximate value functions $\hat{V}_t$ are the optimal value functions for a relaxed model that is a convex DP with $A \subseteq \hat{A}$ and $r \leqslant \hat{r}$, and $\hat{\alpha}$ is an optimal policy for this relaxed model, then there exists a gradient selection $\boldsymbol{\delta}$ such that, for every scenario ω,*

$$\max_{\mathbf{a} \in A(\omega)} \big\{ r(\mathbf{a}, \omega) - \hat{\pi}_\partial(\mathbf{a}, \omega) \big\} \leqslant \hat{V}_0.$$

Proof. See Appendix A.2. □

Thus, any gradient selection $\boldsymbol{\delta}$ will provide a valid penalty (this follows from Proposition 2.1 as before) and therefore provide a valid bound. However we can only be sure that *there exists* a gradient selection that will be optimal given the optimal value function or will improve on a relaxed value function. If the value functions or approximate value functions are differentiable, there is a unique gradient at each point and this proposition reduces to Proposition 3.1, which assumes differentiability.

In the nondifferentiable case, we have some flexibility in the gradient selection and the choice of gradients may affect the quality of the bound. The proof of parts (ii) and (iii) of the proposition follows the same general form as that outlined following Proposition 3.1 for the differentiable case, with the stacking gradient result (Lemma 3.1(viii)) leading to a generalized version of the consistency condition (13). The proof of (ii) constructs an optimal gradient selection and thus provides guidance on how to select gradients in applications. Setting aside constraints on actions, we seek gradient selections $\boldsymbol{\delta}_{t-1}$ from $\partial \hat{V}_t(\hat{\boldsymbol{\alpha}}_{t-1})$, $\boldsymbol{\delta}_t^r$ from $\partial r_t(\boldsymbol{\alpha}_t)$, and $\boldsymbol{\delta}_t$ from $\partial \hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t)$ such that the analog of the consistency condition (13) holds, i.e.,

$$\begin{pmatrix} \boldsymbol{\delta}_{t-1} \\ \mathbf{0} \end{pmatrix} = \boldsymbol{\delta}_t^r + \mathbb{E}[\boldsymbol{\delta}_t \,|\, \mathscr{F}_t]. \tag{16}$$

We take constraints on actions into account in the proofs by incorporating them into the reward functions using characteristic functions that punish violations of the constraints. In the proof, we work forward in time, beginning with a gradient selection for period 0. Then, given a selection $\boldsymbol{\delta}_{t-1}$ for period $t-1$, we find selections $\boldsymbol{\delta}_t^r$ and $\boldsymbol{\delta}_t$ such that (16) holds. When working with an optimal value function as in parts (ii) and (iii) of the proposition, we can be sure that such gradient selections exist.

The consistency condition (16) is critical for obtaining tight bounds. To see this, consider gradient selections $\boldsymbol{\delta}_t^r$ from $\partial r_t(\hat{\boldsymbol{\alpha}}_t)$, and $\boldsymbol{\delta}_t$ from $\partial \hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t)$; we will assume any constraints on actions are included in the reward through the use of characteristic functions. Suppose (16) holds with error $\boldsymbol{\epsilon}_t$ in period t, i.e., $\boldsymbol{\epsilon}_t = (\boldsymbol{\delta}_{t-1}, \mathbf{0}) - \boldsymbol{\delta}_t^r - \mathbb{E}[\boldsymbol{\delta}_t \,|\, \mathscr{F}_t]$. Omitting the control variate terms from the penalty, the inner problem (5) with penalty (15) can be written:

$$\begin{aligned}
&\max_{\mathbf{a} \in A} \left\{ \sum_{t=0}^{T} r_t(\mathbf{a}) - (\boldsymbol{\delta}_t - \mathbb{E}[\boldsymbol{\delta}_t | \mathscr{F}_t])^\top (\mathbf{a}_t - \hat{\boldsymbol{\alpha}}_t) \right\} \\
&= \max_{\mathbf{a} \in A} \left\{ \sum_{t=0}^{T} r_t(\mathbf{a}) - (\boldsymbol{\delta}_t^r + \boldsymbol{\epsilon}_t)^\top (\mathbf{a}_t - \hat{\boldsymbol{\alpha}}_t) \right\} \\
&\leqslant \max_{\mathbf{a} \in A} \left\{ \sum_{t=0}^{T} r_t(\mathbf{a}) - \boldsymbol{\delta}_t^{r\top} (\mathbf{a}_t - \hat{\boldsymbol{\alpha}}_t) \right\} + \max_{\mathbf{a} \in A} \left\{ \sum_{t=0}^{T} -\boldsymbol{\epsilon}_t^\top (\mathbf{a}_t - \hat{\boldsymbol{\alpha}}_t) \right\} \\
&\leqslant \sum_{t=0}^{T} r_t(\hat{\boldsymbol{\alpha}}_t) + \max_{\mathbf{a} \in A} \left\{ \sum_{t=0}^{T} -\boldsymbol{\epsilon}_t^\top (\mathbf{a}_t - \hat{\boldsymbol{\alpha}}_t) \right\}.
\end{aligned}$$

The first equality follows from rearranging terms, the first inequality from optimizing separately rather than jointly, and the last from concavity of r_t. (If $\hat{\boldsymbol{\alpha}}_t$ is feasible (i.e., in A),

this last inequality is an equality.) Then, incorporating the control variate terms in the penalty, we find the inner problem can be written as

$$\max_{\mathbf{a}\in A}\{r(\mathbf{a})-\hat{\pi}_\partial(\mathbf{a})\}\leqslant \hat{V}_0+\max_{\mathbf{a}\in A}\sum_{t=0}^{T}-\boldsymbol{\epsilon}_t^\top(\mathbf{a}_t-\hat{\boldsymbol{\alpha}}_t),$$

where $\hat{V}_0$ is the value given by following policy $\hat{\alpha}$. Thus the error terms $\boldsymbol{\epsilon}_t$ affect the tightness of the bound. If we are working with optimal values $\hat{V}_0$ and policies $\hat{\alpha}$ for a relaxed model as in part (iii) of Proposition 3.2, we can be sure that there exists gradients such that these error terms are zero. If we can construct one, we can be sure that gradient penalty bound will improve on the bounds from the relaxed model, in every scenario. In our examples, we will attempt to select gradients so these errors will be zero when possible (as in the revenue management example of §4 where the penalty is based on a Lagrangian relaxation, without reoptimization) or small when not possible (as in the other cases).

3.3. Control Variates and Pathwise Bounds

As mentioned earlier, the final terms in the expressions defining the gradient penalties (10) and (15) (inside the parentheses) have zero mean and are constant with respect to actions in the inner problem (5); thus these terms have no effect on expected value of the bound (4). However, because these terms are likely to be correlated with the period rewards, these terms may serve as helpful control variates and reduce the variance in simulation-based estimates of the upper bound (4). (Indeed various forms of control variates have been frequently used with information relaxations.)

These control variates may also be helpful when estimating the expected reward associated with a heuristic policy, i.e., in estimating a primal lower bound. For example, given a heuristic policy $\hat{\alpha}$ that is feasible for the primal DP (1), we can write the expected total reward as

$$\begin{aligned}\mathbb{E}[r(\hat{\alpha})]&=\mathbb{E}\left[\sum_{t=0}^{T}r_t(\hat{\boldsymbol{\alpha}}_t)\right]\\&=\mathbb{E}\left[\sum_{t=0}^{T}\{r_t(\hat{\boldsymbol{\alpha}}_t)-(\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t)-\mathbb{E}[\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t)\,|\,\mathcal{F}_t])\}\right],\end{aligned}\tag{17}$$

where this last form incorporates the control variate terms. This form of control variate is of the form considered in the "approximating martingale process" variance reduction approach of Henderson and Glynn (2002). It is not difficult to see that if the value functions $\hat{V}_t$ are value functions corresponding to policy $\hat{\alpha}$ (so $\hat{V}_t(\hat{\boldsymbol{\alpha}}_{t-1})=r_t(\hat{\boldsymbol{\alpha}}_t)+\mathbb{E}[\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t)\,|\,\mathcal{F}_t]$) adjacent terms in (17) cancel and the expectations reduce to the expectation of a constant, $\mathbb{E}[\hat{V}_0]=\hat{V}_0$. In this case, when estimating values by simulation, we obtain a zero-variance estimate of the expected reward associated with policy $\hat{\alpha}$. If the functions $\hat{V}_t$ approximate the values given by the policy $\hat{\alpha}$ (or, more precisely, approximate the differences in values appearing in (17)), we might expect to obtain low variance estimates of the value associated with a given policy.

These controlled estimates of the value with a given heuristic pair nicely with gradient penalties. If we use a gradient penalty based on approximate functions $\hat{V}_t$ with gradients taken around policy $\hat{\alpha}$, the inner problem (5) becomes

$$\begin{aligned}&\max_{\mathbf{a}\in A}\{r(\mathbf{a})-\hat{\pi}_\partial(\mathbf{a})\}\\&\quad=\max_{\mathbf{a}\in A}\sum_{t=0}^{T}r_t(\hat{\mathbf{a}}_t)-(\boldsymbol{\delta}_t-\mathbb{E}[\boldsymbol{\delta}_t\,|\,\mathcal{F}_t])^\top(\mathbf{a}_t-\hat{\boldsymbol{\alpha}}_t)\\&\qquad\qquad-\big(\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t)-\mathbb{E}[\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t)\,|\,\mathcal{F}_t]\big).\end{aligned}$$

If the policy $\hat{\alpha}$ chooses feasible actions, we can take $\mathbf{a}=\hat{\boldsymbol{\alpha}}$ as a feasible but not necessarily optimal choice in the optimization problem above and find

$$\begin{aligned}&\max_{\mathbf{a}\in A}\{r(\mathbf{a})-\hat{\pi}_\partial(\mathbf{a})\}\\&\quad\geqslant\sum_{t=0}^{T}r_t(\hat{\boldsymbol{\alpha}}_t)-\big(\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t)-\mathbb{E}[\hat{V}_{t+1}(\hat{\boldsymbol{\alpha}}_t)\,|\,\mathcal{F}_t]\big),\end{aligned}\tag{18}$$

where the right side here is the controlled estimate of value for policy in (17), for a given scenario. Thus, with this form of penalty, the inner problem values will be greater than or equal to the corresponding controlled estimate of the value under policy $\hat{\alpha}$ in every scenario. This relationship facilitates comparisons between heuristics and dual problems in each scenario and, when the estimates are both controlled in this way, we can obtain more precise estimates of the upper and lower bounds as well as the differences between them, i.e., the duality gap.

Note, however, that the policies $\hat{\alpha}$ used in the gradient penalties need not be feasible for the primal problem. For example, in the network revenue management problem, we will construct bounds that improve on a Lagrangian relaxation of the original model, using the result of Proposition 3.2(iii). In this case, the gradient penalties are taken around the optimal solution $\hat{\alpha}$ for the relaxed model, which, in general, will not be feasible for the original model. In this case (as will be evident in Figure 1), the inequality (18) need not hold in every scenario. When working with a reoptimized model in the network revenue management example, we take gradients around a feasible policy and (18) will be satisfied every scenario.

4. Example: Network Revenue Management

We consider a network revenue management application, following Topaloglu (2009). Although we present the problem in the context of an airline, the model also applies in other settings (e.g., railways, hotel chains). Topaloglu (2009) uses Lagrangian relaxation techniques (see, e.g., Hawkins 2003 or Adelman and Mersereau 2008) to approximate the network revenue management model.

4.1. The Model

Time is discrete and indexed as $t = 1, \ldots, T$. The airline has flights on a set $\mathcal{L} = \{1, \ldots, L\}$ of L *legs*. In each period, a customer requests one of I *itineraries* from the set $\mathcal{I} = \{1, \ldots, I\}$. We assume T, L, and I are finite. Itinerary i consumes f_{il} units of capacity on leg $l \in \mathcal{L}$; $\mathbf{f}_i \in \mathbb{Z}_+^L$ denotes the vector of capacity consumption for all legs of itinerary i. At any time t, the vector $\mathbf{c}_t \in \mathbb{Z}_+^L$ denotes the airline's remaining capacity on the legs; the initial capacity $\mathbf{c}_1$ is given.

Given an itinerary request i_t in period t, the airline decides whether to accept the request (taking $a_t = 1$) or reject it ($a_t = 0$). The airline can accept a request only if enough capacity remains, i.e., only if $\mathbf{c}_t \geqslant \mathbf{f}_{i_t}$. If the airline accepts the request, it receives r_{i_t} in immediate revenue and capacity becomes $\mathbf{c}_{t+1} = \mathbf{c}_t - \mathbf{f}_{i_t}$. If the airline rejects the request, it receives no revenue and capacity is unchanged, i.e., $\mathbf{c}_{t+1} = \mathbf{c}_t$. This problem can be formulated as a stochastic DP with state vector $(\mathbf{c}_t, i_t)$ describing the capacity remaining and itinerary request in period t. The Bellman recursion, for $t = 1, \ldots, T$, is

$$V_t(\mathbf{c}_t, i_t) = \max_{a_t \in A_t(\mathbf{c}_t, i_t)} \left\{ r_{i_t} a_t + \mathbb{E}[V_{t+1}(\mathbf{c}_t - \mathbf{f}_{i_t} a_t, \tilde{i}_{t+1})] \right\}, \quad (19)$$

with $V_{T+1} = 0$ and constraint set $A_t(\mathbf{c}_t, i_t) = \{a \in \{0, 1\}: \mathbf{f}_{i_t} a \leqslant \mathbf{c}_t\}$. Expectations are taken over the next-period itinerary request $\tilde{i}_{t+1}$. We assume itinerary requests are independent over time but allow the probabilities for the itinerary requests to vary over time. We adopt the convention that the first request arrives at $t = 1$, and let $V_0 = \mathbb{E}[V_1(\mathbf{c}_1, \tilde{i}_1)]$ be the expected revenue generated by an optimal policy.

Although the formulation (19) uses state vector notation, the general setup discussed in §2.1 assumes value functions are described as functions of past actions. For this problem, it is straightforward to express the value functions (19) as functions of prior actions, since $\mathbf{c}_t = \mathbf{c}_1 - \sum_{\tau=1}^{t-1} \mathbf{f}_{i_\tau} a_\tau$.

4.2. Lagrangian Relaxations

The state space for the DP (19) grows exponentially in the number of legs and the DP will be very difficult to solve with more than a few legs. The challenge comes from the fact that decisions are coupled across legs, as accepting an itinerary simultaneously reduces capacity across all legs on that itinerary. Topaloglu (2009) considers approximations of (19) that are based on a Lagrangian relaxation that relaxes this coupling constraint by allowing the airline to accept or reject individual legs of an itinerary; violations of the leg coupling constraints are "punished" with Lagrange multipliers. We will use the same Lagrangian relaxation as Topaloglu (2009), albeit with a slightly different form.

Before defining this Lagrangian relaxation, it is helpful to rewrite (19) in a form that has decision variables for each leg but requires these decisions to be the same for all legs. We let $\mathbf{a}_t = (a_{t1}, \ldots, a_{tL})$ denote the vector of decision variables for period t and let $\bar{a}_t$ denote the average of $\mathbf{a}_t$ over its L elements, i.e., $\bar{a}_t = L^{-1} \sum_{l \in \mathcal{L}} a_{lt}$; the coupling constraint requires $a_{lt} = \bar{a}_t$ for all l. The original model (19) can then be rewritten as

$$V_t(\mathbf{c}_t, i_t) = \max_{\mathbf{a}_t \in \mathbf{A}_t(\mathbf{c}_t, i_t)} \left\{ r_{i_t} \bar{a}_t + \mathbb{E}[V_{t+1}(\mathbf{c}_t - \mathbf{f}_{i_t} \circ \mathbf{a}_t, \tilde{i}_{t+1})] \right\}, \quad (20)$$

with $\mathbf{A}_t(\mathbf{c}_t, i_t) = \{\mathbf{a} \in \{0, 1\}^L: f_{i_t l} a_l \leqslant c_{tl},\ a_l = \bar{a}, \text{ for all } l \in \mathcal{L}\}$. Here $\mathbf{f}_{i_t} \circ \mathbf{a}_t$ denotes the component-wise product of the vectors $\mathbf{f}_{i_t}$ and $\mathbf{a}_t$.

The Lagrangian relaxation introduces Lagrange multipliers associated with the coupling constraints. Following Topaloglu (2009), we allow these Lagrange multipliers λ_{ilt} to depend on itinerary, leg, and time, but not capacity. The value function V_t^λ for this relaxation can be written as

$$V_t^\lambda(\mathbf{c}_t, i_t) = \max_{\mathbf{a}_t \in \hat{\mathbf{A}}_t(\mathbf{c}_t, i_t)} \left\{ r_{i_t} \bar{a}_t + \sum_{l \in \mathcal{L}} \lambda_{ilt} (a_{tl} - \bar{a}_t) + \mathbb{E}[V_{t+1}^\lambda(\mathbf{c}_t - \mathbf{f}_{i_t} \circ \mathbf{a}_t, \tilde{i}_{t+1})] \right\}, \quad (21)$$

where $\hat{\mathbf{A}}_t(\mathbf{c}_t, i_t) = \{\mathbf{a} \in [0, 1]^L: f_{i_t l} a_l \leqslant c_{lt} \text{ for all } l \in \mathcal{L}\}$ and $V_{T+1}^\lambda = 0$. Note that in this formulation, in addition to relaxing the coupling constraints, we also allow a_{lt} to be in $[0, 1]$ rather than $\{0, 1\}$. Thus, in addition to allowing the airline to accept or reject individual legs of an itinerary, in the relaxed model the airline can accept parts of requests. For example, if an itinerary consumes two units of capacity, in the relaxed model, the airline may accept half the itinerary. This relaxation convexifies the set of feasible actions and makes the Lagrangian relaxation a convex DP: the rewards are linear in actions and the action sets are convex. Moreover, the Lagrangian relaxation (21) is also a relaxation of the original problem (20) in the sense of our Proposition 3.2(iii): the rewards coincide for all feasible actions and the set of feasible actions in the Lagrangian relaxation includes that of the original problem.[2]

The dual problem associated with this Lagrangian relaxation is

$$\min_{\boldsymbol{\lambda}} V_0^\lambda = \min_{\boldsymbol{\lambda}} \mathbb{E}[V_1^\lambda(\mathbf{c}_1, \tilde{i}_1)]. \quad (22)$$

The following proposition provides some basic properties of this Lagrangian approximation. We let $\mathcal{L}(i)$ denote the set of legs on itinerary i, i.e., l such that $f_{il} > 0$.

Proposition 4.1. *Consider the Lagrangian relaxation* (21). *Let* $\Lambda \subseteq \mathbb{R}^{ILT}$ *denote the set of Lagrange multipliers satisfying, for all itineraries* i *and times* t: (a) $\lambda_{ilt} \geqslant 0$ *for all* $l \in \mathcal{L}$; (b) $\lambda_{ilt} = 0$ *if* $l \notin \mathcal{L}(i)$; *and* (c) $\sum_{l \in \mathcal{L}} \lambda_{ilt} = r_i$.

(i) Restricted Lagrangian relaxations. *For all capacities* $\mathbf{c}_t$ *and itineraries* i_t,

$$V_t(\mathbf{c}_t, i_t) \leqslant \min_{\boldsymbol{\lambda}} V_t^\lambda(\mathbf{c}_t, i_t) = \min_{\boldsymbol{\lambda} \in \Lambda} V_t^\lambda(\mathbf{c}_t, i_t). \quad (23)$$

(ii) Value function decomposition. *If* $\boldsymbol{\lambda} \in \Lambda$, *then for all capacities* $\mathbf{c}_t$ *and itineraries* i_t,

$$V_t^\lambda(\mathbf{c}_t, i_t) = \sum_{l \in \mathcal{L}} \vartheta_{lt}^\lambda(c_{lt}, i_t), \quad (24)$$

where

$$\vartheta^{\lambda}_{lt}(c_{lt}, i_t) = \max_{a_{lt} \in A_{lt}(c_{lt}, i_t)} \{\lambda_{i_t lt} a_{lt} + \mathbb{E}[\vartheta^{\lambda}_{l,t+1}(c_{lt} - f_{i_t l} a_{lt}, \tilde{i}_{t+1})]\},$$

and $A_{lt}(c_{lt}, i_t) = \{a \in [0,1]\colon f_{i_t l} a \leqslant c_{lt}\}$. *In addition,* ϑ^{λ}_{lt} *is nondecreasing, piecewise linear, and concave in* c_{lt} *for all* i_t.

Thus, with this Lagrangian relaxation, as shown in Topaloglu (2009), the problem decouples into the sum of L leg-specific value functions ϑ^{λ}_{lt} that depend only on the capacity of leg l itself. However, here we restrict the Lagrange multipliers be in the set Λ. This simplifies the representation (24) of the Lagrangian and reduces the number of nonzero Lagrange multipliers involved in the dual optimization problem (23). As shown in part (i) above, this restriction is without loss of optimality. This restriction also leads to a nice interpretation. In this approximate model (24), the airline can accept or reject individual legs of itineraries and receive revenue $\lambda_{ilt} \geqslant 0$ for legs on an itinerary. These fictitious revenues are zero for legs not on an itinerary and sum to r_i. Thus the Lagrange multipliers in Λ represent an allocation of the revenue r_i for an itinerary i to the legs on the itinerary; these allocations may vary over time. The dual problem (23) is to find a revenue allocation that minimizes the optimal expected revenue in this relaxed model.[3]

The approximate value functions from a Lagrangian relaxation can also be used to generate a heuristic policy that approximates the continuation value by V^{λ}_t in every period. Specifically, the *Lagrangian heuristic*, defined in Topaloglu (2009), will accept itinerary i_t arriving in period t with capacities $\mathbf{c}_t$ if it is feasible to accept the itinerary and the approximate value of accepting the itinerary exceeds that of rejecting, i.e., if

$$r_{i_t} + \sum_{l \in \mathcal{L}(i_t)} \mathbb{E}[\vartheta^{\lambda}_{l,t+1}(c_{lt} - f_{i_t l}, \tilde{i}_{t+1})] \geqslant \sum_{l \in \mathcal{L}(i_t)} \mathbb{E}[\vartheta^{\lambda}_{l,t+1}(c_{lt}, \tilde{i}_{t+1})]. \quad (25)$$

Following Topaloglu (2009), we can potentially improve the Lagrangian heuristic by reoptimizing the Lagrangian dual problem (22). Specifically, in each scenario, at prespecified times we minimize the Lagrangian dual (22) at the then-prevailing capacity. Reoptimizing may improve the heuristic by providing better approximations of the optimal value function as the capacities evolve over time.

4.3. Gradient Penalties and Inner Problems

We can use these Lagrangian relaxations to generate gradient penalties and performance bounds. With perfect information, the itineraries $i_1, \ldots, i_T$ are known in advance, and the inner problem has T decisions $a_t \in \{0, 1\}$ representing whether to accept or reject itinerary i_t. In these inner problems, unlike the Lagrangian relaxations, we impose the leg coupling constraints. Let $\mathbf{r}$ be the vector of length T with tth element being r_{i_t} and $\mathbf{F}$ be the $L \times T$ matrix with tth column corresponding to $\mathbf{f}_{i_t}$. With $\hat{\pi}^{\lambda}_{\partial}$ denoting a gradient penalty, the inner problem is

$$\begin{aligned} &\underset{\mathbf{a} \in \{0,1\}^T}{\text{maximize}} \quad \{\mathbf{r}^{\top}\mathbf{a} - \hat{\pi}^{\lambda}_{\partial}(\mathbf{a})\} \\ &\text{subject to} \quad \mathbf{F}\mathbf{a} \leqslant \mathbf{c}_1. \end{aligned} \quad (26)$$

Since the penalty is affine in $\mathbf{a}$, (26) is a binary linear program: there is a single binary decision variable (accept or reject the realized itinerary) for each period and the objective is to maximize the (penalized) revenue. The constraints require the total capacity consumed by all accepted itineraries to be less than the initial capacity of each leg. These inner problems thus scale linearly with the number of periods T and legs L in the problem and do not depend on the number of itineraries I or the capacity available on a leg.

From Proposition 4.1(ii), the Lagrangian relaxation value functions can be written as a sum of leg-specific value functions. Let λ denote a set of Lagrange multipliers and let $\alpha^{\lambda}_l = (\alpha^{\lambda}_1, \ldots, \alpha^{\lambda}_T)$ be the leg l decisions for this policy. We can write a gradient penalty associated with this Lagrangian relaxation as

$$\hat{\pi}^{\lambda}_{\partial}(\mathbf{a}) = \sum_{t=0}^{T} \sum_{l \in \mathcal{L}} \{(\delta^{\lambda}_{lt} - \mathbb{E}[\delta^{\lambda}_{lt}])(a_{lt} - \alpha^{\lambda}_{lt}) + (\vartheta^{\lambda}_{l,t+1}(c_{lt}(\alpha^{\lambda}_l), i_t) - \mathbb{E}[\vartheta^{\lambda}_{l,t+1}(c_{lt}(\alpha^{\lambda}_l), \tilde{i}_{t+1})])\}, \quad (27)$$

where δ_{lt} is a gradient selection for $\vartheta^{\lambda}_{l,t+1}(c_{lt}(\alpha^{\lambda}_l), i_t)$ with respect to the leg l acceptance decisions and $c_{lt}(\alpha^{\lambda}_l)$ denotes the leg l capacity in period t as a function of the decisions α^{λ}_l for this leg under policy α. As noted in Proposition 4.1(ii), the leg-specific value functions $\vartheta^{\lambda}_{l,t+1}(c_{lt}, i_t)$ are increasing, piecewise linear, and concave in capacity c_{lt} and hence will be nondifferentiable where the slopes change as we change "pieces" in these piecewise linear functions. These changes in slopes reflect changes in the set of binding constraints in future periods as we change future decisions (according to the optimal policy for the leg-specific problem) in response to changes in the current capacity.

Because these leg-specific value functions may be nondifferentiable, the choice of gradients is generally not unique. Since the Lagrangian relaxation is a convex DP with rewards and constraint sets that are weakly larger than those of the original model, Proposition 3.2(iii) ensures that there exists a gradient selection for (27) such that the optimal value of the inner problems will be less than or equal to V^{λ}_0 in every scenario; with such a selection, the upper bound from this approach will be (weakly) tighter than the upper bound from the corresponding Lagrangian relaxation. However, some care is required to construct such a gradient selection. We use a procedure that selects gradients to satisfy condition (16) for the Lagrangian value

function, working forward in time, as discussed following Proposition 3.2; this procedure is described in detail in the appendix. This procedure ensures (16) is satisfied and thus results in inner problems whose objective function will be less than or equal to V_0^λ in every scenario.

We can simplify the inner problems (26) by relaxing the binary constraints requiring a_t to be in $\{0, 1\}$ to allow a_t to be in $[0, 1]$, so (26) is a linear program (LP) rather than an integer program. The average of these relaxed inner problems would still provide a valid upper bound and, with the gradient selection discussed above, would still improve on the Lagrangian bound in every scenario (because the Lagrangian relaxation also allows a_t to be in $[0, 1]$). Although this LP relaxation could in principle lead to a weaker upper bound than that given by enforcing the binary constraints, in the numerical experiments of §4.5, the LP relaxations have had binary optimal solutions in every scenario and, thus, this relaxation made no difference in the bounds obtained.

As with the heuristic, we can also use reoptimization to potentially improve the upper bounds. We do this using a gradient penalty analogous to (27) with the reoptimized Lagrangian relaxation value functions. In this variant, we select gradients around actions chosen by the Lagrangian heuristic (rather than the optimal policy α_{lt}^λ for the Lagrangian relaxation as in (27)). By Proposition 3.2(i), such a penalty is dual feasible and thus leads to a valid upper bound. Because this penalty is built from Lagrangian relaxations that change over time, within each scenario, we do not have a theoretical result like Proposition 3.2(iii) that ensures that there will be a gradient selection that results in an upper bound that is better than the Lagrangian bound V_0^λ. However, we might expect these reoptimized gradient penalties to lead to better bounds because the value functions are better approximated in downstream states. We will use a gradient selection procedure that is like that used in the case without reoptimization; the details are in the appendix. However, here, unlike the case without reoptimization discussed above, there is no guarantee that we can find a gradient selection such that (16) holds or that the resulting upper bound will be better than V_0^λ.

4.4. Examples

We will consider two numerical examples, one with one hub and one with two hubs. These examples are from data sets that were developed and studied by Huseyin Topaloglu.[4]

The one-hub example considers a network with one hub and eight satellite cities: the airline has flights from each satellite city to the hub and back. The itineraries are all possible combinations of starting points and final destinations (each reachable with at most two legs) and come in low- and high-fare classes: thus there are $L = 16$ legs and $I = 144$ itineraries ($= 2$ fare classes $\times$ 9 cities $\times$ 8 possible destinations from each city). All itineraries request at most one unit of capacity per leg, i.e., $f_{il} \in \{0, 1\}$. There are $T = 200$ periods; the total capacity on all 16 legs is 358 and the maximum initial capacity for any leg is 31. The probabilities for the itineraries vary over time, with the probabilities of low-fare itinerary requests decreasing and high-fare requests increasing as time passes. A full DP model for this example has approximately 2×10^{26} states and would be very difficult to solve exactly.

The two-hub example has $L = 14$ legs, $I = 113$ itineraries, and the maximum number of legs on any itinerary is three; again, there are low- and high-fare itineraries for each route flown. There are $T = 400$ periods; the total capacity on all 14 legs is 621 and the maximum initial capacity for any leg is 82. Again, all itineraries request at most one unit of capacity per leg and the probabilities vary over time with a pattern similar to that in the one-hub example. The full DP has approximately 3×10^{27} states.

For each example, we first optimize the Lagrangian relaxations (i.e., solve (22)) to find a good set of Lagrange multipliers. This is done once, before running the simulation. We use a subgradient optimization algorithm, starting with the $\boldsymbol{\lambda} \in \Lambda$ that splits revenue for each itinerary equally among all legs on the itinerary. We then run this subgradient algorithm for 200 iterations. The resulting Lagrange multipliers $\boldsymbol{\lambda}^\star$ need not be exactly optimal, but nevertheless can be used to generate a heuristic and, as indicated in Proposition 4.1(i), will provide an upper bound $V_t^{\lambda^\star}$ on the optimal value function.

Then, in a Monte Carlo simulation, we generate 100 sample itinerary scenarios. In each sample scenario, we do the following:

(i) We evaluate the Lagrangian heuristic using the approximate value functions $V_t^{\lambda^\star}$ and calculate the total reward collected using this heuristic in this scenario. These sample values are then adjusted using control variates as discussed in §3.3.

(ii) We calculate a gradient selection and a corresponding gradient penalty based on the Lagrangian relaxation; we then solve the inner problem (26). The gradient selection is constructed as discussed in the appendix to ensure that the upper bounds are weakly tighter than $V_0^{\lambda^\star}$ in every scenario.

Averaging the values from (i) and (ii) across these 100 sample scenarios provides estimates of lower and upper bounds on the optimal revenue V_0.

In addition, we consider heuristics and bounds based on the reoptimized Lagrangian heuristic, as discussed in §4.3. Following Topaloglu (2009), we reoptimize every $T/5$ time periods in each scenario. Because these reoptimizations are time consuming, we use just 25 iterations in the subgradient optimization algorithm when reoptimizing. Of course, reoptimizing more frequently or more precisely (e.g., with more iterations) may lead to better results.

Finally, we compute upper and lower bounds that provide useful benchmarks for evaluating the other heuristics and bounds. For a lower bound, we consider a *naive heuristic* that accepts itineraries whenever capacity is available on

all involved legs. For an upper bound, we solve inner problems (26) with zero penalty; the average of the optimal values for the inner problem gives an estimate of the expected value with perfect information. Both of these benchmark bounds are easy to compute.

4.5. Results

Table 1 shows the results, including run times, for both examples. All computations are on a desktop computer (a Dell PC with a 3.07 GHz Intel Xeon quad-core CPU and 12.0 GB of RAM) running Windows 7, using MATLAB 7.12.0 (R2011a) with the MOSEK 7.0 Optimization Toolbox to solve the inner problems (26). Mean standard errors (MSEs) are provided for the bounds that are estimated using simulation. The MSEs are calculated in the usual way as $\sigma/\sqrt{n}$, where σ is the standard deviation of the heuristic or dual values generated in the simulation, and $n = 100$ is the number of trials in the simulation. The MSEs for the heuristics are MSEs for the heuristic values after adjusting them using the control variates.

The naive heuristic, although very easy to evaluate, performs quite poorly. This is not surprising: in these examples, the most valuable itineraries have a high probability of arriving near the end of the time horizon and this naive heuristic often leaves insufficient capacity to accept them.

The heuristics and bounds from the Lagrangian relaxations are much better. It takes a few minutes to run the subgradient algorithm to find a good choice of Lagrange multipliers; the two-hub example takes longer because it has twice the number of time periods, more capacity, and the itineraries consume capacity on more legs. Given the leg-specific value functions for the Lagrangian relaxation, the heuristic is relatively easy to evaluate. Reoptimization increases the run times for the heuristic substantially (to 45 minutes for the one-hub example and three hours for the two-hub example), but improves its performance significantly.

In terms of the upper bounds, the Lagrangian relaxation upper bounds are good and yield a duality gap less than 3% when compared to the Lagrangian heuristic and about 2% when compared to the heuristic with reoptimization. The perfect information bounds with zero penalty are relatively weak (much worse than the Lagrangian bounds); this shows the need to use some form of penalty. The computational effort associated with calculating the information relaxation upper bounds is not great: most of the work is in calculating the Lagrangian relaxation and reoptimizing. In Table 1,

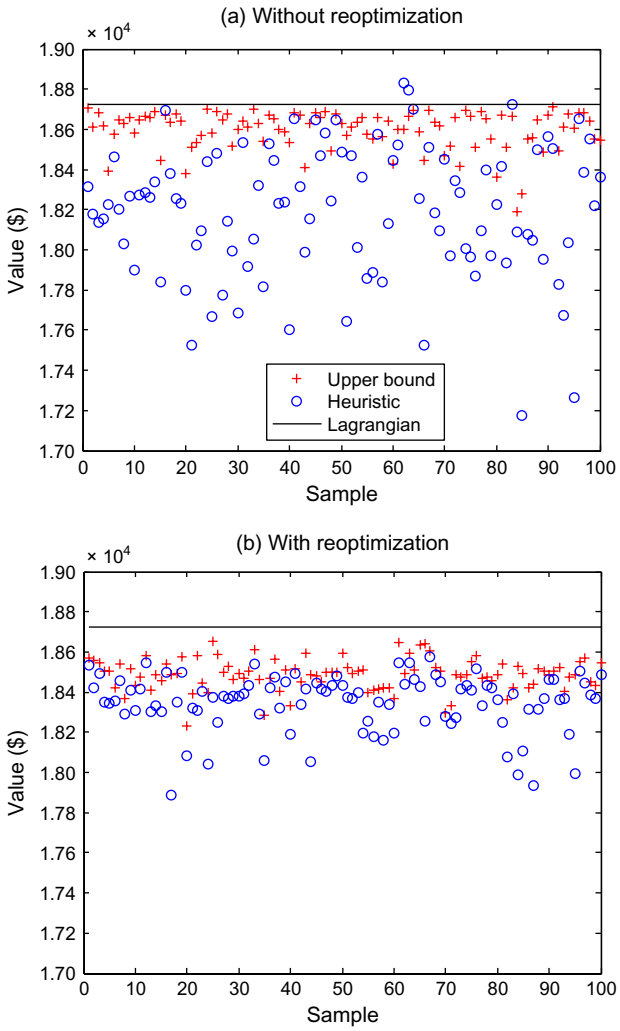
Table 1. Bounds and run times for network revenue management examples (100 samples).

Lower bounds	One-hub example		Two-hub example	
Naive heuristic				
Mean (MSE), $	9,355	(30)	27,259	(50)
Run time, seconds	0.01		0.01	
Lagrangian heuristic (a)				
Mean (MSE), $	18,191	(33)	44,245	(77)
Run time, seconds	33		85	
Lagrangian heuristic with reoptimization (b)				
Mean (MSE), $	18,348	(14)	44,480	(42)
Run time, seconds	2,729		10,853	
Upper Bounds				
Lagrangian relaxation (c)				
Value, $	18,726		45,291	
Run time, seconds	124		571	
Perfect information with zero penalty				
Mean (MSE), $	19,342	(30)	46,067	(50)
Run time, seconds	0.27		0.35	
Perfect info. + Lagrangian gradient penalty (d)				
Mean (MSE), $	18,597	(10)	45,000	(15)
Run Time, seconds	35		69	
Perfect info. + Reopt. Lagrangian gradient penalty (e)				
Mean (MSE), $	18,488	(8)	44,844	(19)
Run Time, seconds	38		69	
Gaps				
Without reoptimization				
Lagrangian relaxation to heuristic, i.e., (c)–(a)	535	2.94%	1,046	2.36%
Perfect info. + LR Grad. penalty to heuristic, i.e., (d)–(a)	406	2.23%	755	1.71%
With reoptimization				
Lagrangian relaxation to heuristic, i.e., (c)–(b)	378	2.06%	811	1.82%
Perfect Info. + LR Grad. penalty to heuristic, i.e., (e)–(b)	140	0.76%	364	0.82%

we see that the bounds with gradient penalties improve on the Lagrangian relaxation bound. Without reoptimization, the duality gap is reduced from 2.94% to 2.23% and 2.36% to 1.71% for the two examples. With reoptimization, the upper bounds are improved and the reoptimized Lagrangian heuristics are within 1% of the new upper bound.

Figure 1 shows a plot of the sample values for the Lagrangian heuristic and the inner problem values (26) for the 100 scenarios in the one-hub example; the plots for the two-hub example are similar. (The values for the heuristics shown here are adjusted using control variates.) In the results without reoptimization, we see that the value of the inner problem with the gradient penalty is no worse than the upper bound $V_0^{\lambda^\star}$ from the Lagrangian relaxation in every scenario; the gradient selection was constructed to ensure this, as in Proposition 3.2(iii). With reoptimization, we see in Figure 1(b) that the inner problem values are better than $V_0^{\lambda^\star}$ in these 100 scenarios, though that need not hold in all cases. We also note with reoptimization, in each scenario, the inner problem values (26) are greater than or equal to the corresponding values of the Lagrangian heuristic. As discussed in §3.3, this must be the case when the gradients are taken around the actions chosen by the Lagrangian heuristic. This is not true in the case without reoptimization as the gradients are taken around an infeasible policy.

Figure 1. (Color online) Sample values for the Lagrangian heuristic and the inner problems for the one-hub example.



Finally, we consider the impact of the gradient selection on the quality of the bounds. As discussed earlier, the results in Table 1 were calculated using a gradient selection designed to ensure that the consistency conditions (16) are satisfied, if possible. If we instead take the gradient selection δ_t in (27) to be a simple 50–50 mix of left and right derivatives of $\vartheta_{l,t+1}^{\lambda}$ for each capacity level, we obtain weaker bounds, particularly in the case with reoptimization. In the one-hub example, without reoptimization, the upper bound with the simple 50–50 gradients is \$18,656 (13) as compared to \$18,597 (10) with the more sophisticated gradient selection; with reoptimization, the upper bound is \$18,929 (31), as compared to \$18,488 (8). (MSEs are shown in parentheses.) The results for two-hub example are similar: with 50–50 gradients, the bounds are \$45,068 (20) without reoptimization and \$45,444 (84) with reoptimization, as compared to \$45,000 (15) and \$44,844 (19), respectively, with the more sophisticated gradient selection. These results illustrate the importance of selecting gradients carefully, so the consistency conditions (16) are satisfied or approximately so. This is particularly important in the case with reoptimization as we have gradients of different approximate models (e.g., with different Lagrange multipliers) in different periods and the consistency condition (16) links gradients across periods.

To get a sense of just how well the heuristic with reoptimization is performing, let us reconsider the duality gaps of Table 1. In the one-hub example, the best duality gap (\$140) is less than the average value (\$225) of an itinerary arriving in the final period and less than one-third of the highest priced itinerary (\$456). In the two-hub example, the gap (\$364) is close to the average value of an itinerary in the last period (\$331) and less than half the maximum itinerary value (\$775). Hence, in these two examples with 358 and 621 seats to sell, the gradient penalty bounds show that this heuristic is within the value of a single ticket of an optimal policy! These bounds thus make it clear that we cannot improve significantly on this heuristic.

5. Example: Inventory Management with Lost Sales

The lost-sales inventory problem is a classic problem that has received renewed attention in recent years. In this model, there is a lead time between orders being placed and delivered and sales are lost rather than backordered when inventory is not sufficient to meet demand. Among many references, the lost-sales model was originally formulated in Karlin and Scarf (1958), further explored in

Morton (1971), and recently reexamined in Zipkin (2008a) and (2008b).

5.1. The Model

We consider a finite-horizon, discrete-time, single-item inventory system with stochastic demands, a constant lead time, and lost sales. We let T be the time horizon and L be the order lead time; L is a positive integer. Given the lead times, we will assume that the system operates over periods t ranging from $t = 0, \ldots, T + L$, with (random) demand d_t ($d_t \geqslant 0$) realized in period t. Demands are assumed to be integer valued and independent and identically distributed over time. We let a_t be the quantity ordered in period t and arriving in period $t + L$; we assume that a_t is integer valued and $a_t \geqslant 0$. In terms of the general notation of §2.1, the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is the standard crossproduct space with scenarios ω representing the vectors of demands $(d_0, \ldots, d_{T+L})$. In the natural filtration, the DM learns demand d_t in period t, after placing an order in that period.

The period-t state vector $\mathbf{x}_t = (x_{t0}, \ldots, x_{t(L-1)})$ is an L-vector where the first element (x_{t0}) represents the inventory level in period t and the remaining elements (x_{tl}) represent the order arriving in period $t + l$. Given period-t state $\mathbf{x}_t$, demand d_t, and order a_t, we can write the next period state as

$$\mathbf{x}_{t+1}(\mathbf{x}_t, d_t, a_t) = \big([x_{t0} - d_t]^+ + x_{t1}, x_{t2}, \ldots, x_{t(L-1)}, a_t\big). \quad (28)$$

With this transition, the next-period inventory is given by the leftover inventory ($[x_{t0} - d_t]^+$) plus the arriving order (x_{t1}); the remaining orders "shift left" to become one period closer to arrival and the current order (a_t) becomes the last order in process. We assume that the systems begins with no orders in process.

Let c ($c \geqslant 0$) be the unit cost of procurement, h ($h \geqslant 0$) be the unit cost of holding inventory, p ($p \geqslant 0$) be the unit penalty for lost sales, and γ ($0 \leqslant \gamma \leqslant 1$) be the discount factor. We can then write the period-t costs as

$$q(\mathbf{x}_t, d_t, a_t) = \gamma^L c a_t + h[x_{t0} - d_t]^+ + p[d_t - x_{t0}]^+.$$

Here we are assuming that the cost of ordering is paid when the goods are delivered in period $t + L$. We take the terminal value to be $f_{T+L+1}(\mathbf{x}_{T+L}) = 0$. We can write earlier optimal-cost-to-go functions as

$$f_t(\mathbf{x}_t) = \min_{a_t \geqslant 0} \mathbb{E}\big[q(\mathbf{x}_t, \tilde{d}_t, a_t) + \gamma f_{t+1}(\mathbf{x}_{t+1}(\mathbf{x}_t, \tilde{d}_t, a_t))\big], \quad (29)$$

where the expectations are taken over the demand $\tilde{d}_t$ in period t. Naturally, it is optimal to take $a_t = 0$ for $t = T + 1, \ldots, T + L$ since these orders would not arrive within the timeframe considered in the model.

5.2. Structural Properties of the Lost-Sales Model

Some care is required to place this problem in the framework of the convex DPs of §3. First, rather trivially, we are minimizing costs rather than maximizing rewards and, hence, we want to have convex cost functions rather than concave value functions and the gradients involved will be subgradients rather than supergradients. Second, as with the network revenue management example, though the problem is naturally described as a function of the state variables, to place it in the framework of §3, we need to write the value functions as functions of the past order quantities rather than the state vector considered above. If we define period costs as the expectations of q as in (29), the cost-to-go functions may not be convex in prior order quantities (more than L periods ago). The inventory level (x_{t0}) in any given period is a convex function of prior order quantities and the cost-to-go function (29) is convex in inventory, but the composition of these two convex functions need not be convex.

To ensure the cost functions are convex in order quantities, we will consider two transformations of the original problem. First, we work with accumulated orders rather than orders: let $z_t = \sum_{\tau=0}^{t} a_t$ denote the accumulated order quantities up to time period t. We can write this in vector form as $\mathbf{z}_t = \mathbf{M}_t \mathbf{a}_t$, where $\mathbf{z}_t = (z_0, \ldots, z_t)$, $\mathbf{a}_t = (a_0, \ldots, a_t)$, and $\mathbf{M}_t$ is a $(t+1)$-by-$(t+1)$ matrix with ones on and below the diagonal; the fact that the order quantities a_t must be nonnegative implies that the accumulated order quantities z_t must be nondecreasing. Second, we take the terminal cost function to be the total (discounted) cost over all $T + L + 1$ periods and all earlier cost functions to be zero.

We can write the terminal cost function $J_{T+L+1}(\mathbf{z}_{T+L})$ for given demand scenario and accumulated order sequence $(\mathbf{z}_{T+L})$ as a linear program where the decision variables $\mathbf{s} = (s_0, \ldots, s_{T+L})$ can be interpreted as the cumulative amount distributed:

$$J_{T+L+1}(\mathbf{z}_{T+L}) = \min_{\mathbf{s}} \sum_{t=0}^{T+L} \gamma^t \big(\gamma^L c(z_t - z_{t-1}) + h(z_{t-L} - s_t) + p(d_t - s_t + s_{t-1})\big)$$

subject to

$$\begin{aligned} &s_t \geqslant s_{t-1} \quad \text{for } t = 0, \ldots, T + L, \\ &s_t \leqslant z_{t-L} \quad \text{for } t = 0, \ldots, T + L, \end{aligned} \quad (30)$$

where we take $s_\tau = 0$ and $z_\tau = 0$ when $\tau < 0$. The constraints in (30) require the amount distributed in period t $(s_t - s_{t-1})$ to be nonnegative and the total distributed to be less than the total received. In the objective, $(z_t - z_{t-1})$, is the amount ordered in period t, $(z_{t-L} - s_t)$ is the leftover inventory, and $(d_t - s_t + s_{t-1})$ is the unmet demand. This linear programming formulation is a relaxation of the original problem (29) in that the DM could choose to hold leftover in inventory (taking $s_t < z_{t-L}$) while simultaneously

leaving demand unmet (taking $s_t - s_{t-1} < d_t$), whereas in the original problem, it is implicitly assumed that demand must be met before any items could be held over as inventory for the next period. However, it is not difficult to see that with our assumptions on model parameters ($h, p \geqslant 0$ and $\gamma \leqslant 1$), it will be optimal to meet demand whenever possible. Moreover, if the demands and order quantities are all integers, the optimal s_t will also be integers.

With this terminal cost function, earlier cost functions are given recursively as

$$J_t(\mathbf{z}_{t-1}) = \min_{\{z_t : z_t \geqslant z_{t-1}\}} \mathbb{E}[J_{t+1}(\mathbf{z}_{t-1}, z_t)], \tag{31}$$

where the expectations are taken over period-t demand. Note that these cost functions represent the expected total costs including past, present, and future costs, rather than the expected costs-to-go as in Equation (29). We take $J_t(\mathbf{z}_{t-1}) = +\infty$ for infeasible order sequences.

Following Zipkin (2008a), we now show that the cost functions for the lost-sales problem are L-natural-convex ($L^\natural$-convex) functions of accumulated orders.[5] An extended real-valued function $g\colon \mathbb{Z}^n \to \mathbb{R}^1 \cup \{+\infty\}$ is submodular if and only if

$$g(\mathbf{p}) + g(\mathbf{q}) \geqslant g(\mathbf{p} \vee \mathbf{q}) + g(\mathbf{p} \wedge \mathbf{q}) \quad \text{for all } \mathbf{p}, \mathbf{q} \in \mathbb{Z}^n$$

where $\vee$ and $\wedge$ denote component-wise maximization and minimization, respectively. An extended real-valued function g defined on $\mathbb{Z}^n$ is $L^\natural$-convex if $\hat{g}(p_0, \mathbf{p}) = g(\mathbf{p} - p_0\mathbf{1})$ is submodular for all $p_0 \in \mathbb{Z}^1$. Here $\mathbf{1}$ is a vector of n ones.

Proposition 5.1. (i) *For all t, $J_t(\mathbf{z}_{t-1})$ is $L^\natural$-convex in $\mathbf{z}_{t-1}$.*

(ii) *Let g be an extended real-valued $L^\natural$-convex function defined on $\mathbb{Z}^n$, let N be the index set $N = \{1, \ldots, n\}$, and let $\mathbb{1}(-)$ be the set-indicator function defined on subsets of indices $X \subseteq N$ where $\mathbb{1}(X)$ is an n-vector whose ith element of $\mathbb{1}(X)$ is 1 if $i \in X$ and 0 otherwise. Then, for any $\mathbf{z} \in \mathbb{Z}^n$ such that $g(\mathbf{z})$ is finite,*

$$\partial g(\mathbf{z}) = \Bigg\{ x \in \mathbb{R}^n \colon g(\mathbf{z}) - g(\mathbf{z} - \mathbb{1}(X)) \leqslant \sum_{i \in X} x_i \leqslant g(\mathbf{z} + \mathbb{1}(X)) - g(\mathbf{z}) \text{ for all } X \subseteq N \Bigg\}.$$

$L^\natural$-convexity implies $J_t(\mathbf{z}_{t-1})$ has a convex extension; that is, there exists a convex function $\bar{J}_t$ such that $\bar{J}_t(\mathbf{z}_{t-1}) = J_t(\mathbf{z}_{t-1})$ for all integer $\mathbf{z}_{t-1}$. Specifically, the convex extension $\bar{J}_t$ is defined as the point-wise supremum over the set of all hyperplanes that lie beneath J_t for all integer order quantities (see Murota 2003). So, in this sense, $\bar{J}_t$ can be viewed as an extended real-valued "convex" function on $\mathbb{R}^t$. The characterization of the differential follows from results in Murota (2003). The differential $\partial g(\mathbf{z})$ is closely related to a base polyhedral set and its extreme points may be found using a variation of the greedy algorithm; details are provided in the online appendix. Using this variation of the greedy algorithm, we can identify an extreme point (or extreme ray) of the differential $\partial g(\mathbf{z})$ by evaluating the function $g(\mathbf{z})$ a total of $t+1$ times.

5.3. Heuristics

Given the difficulty of solving the dynamic program (29), it is natural to consider simpler heuristics. We will focus on a myopic heuristic studied by Morton (1971), Zipkin (2008b), and others. In this heuristic, the DM chooses order quantities a_t in period t to minimize costs from period t to period $t+L$ (when the order a_t arrives) without considering the evolution of the system after that point or future orders. Thus, in each period, we solve a one-dimensional optimization problem to find the myopic order quantity a_t for a given state $\mathbf{x}_t$:

$$\hat{f}_t^L(\mathbf{x}_t) = \min_{a_t \geqslant 0} \mathbb{E}\big[q(\mathbf{x}_t, \tilde{d}_t, a_t) + \gamma \hat{f}_{t+1}^{L-1}(\mathbf{x}_{t+1}(\mathbf{x}_t, \tilde{d}_t, a_t))\big]. \tag{32}$$

For the lookahead values $\hat{f}_t^l(\mathbf{x}_t)$ for $0 < l < L$, we assume there are no additional orders and take

$$\hat{f}_t^l(\mathbf{x}_t) = \mathbb{E}\left[q(\mathbf{x}_t, \tilde{d}_t, 0) + \gamma \hat{f}_{t+1}^{l-1}(\mathbf{x}_{t+1}(\mathbf{x}_t, \tilde{d}_t, 0))\right]. \tag{33}$$

The terminal lookahead value is given by $\hat{f}_t^0(\mathbf{x}_t) = \rho([x_{t0} - d_t]^+)$. Here $\rho(\cdot)$ is a function that approximates the residual value of inventory remaining at the end of the myopic lookahead horizon. Zipkin's version of the myopic heuristic takes $\rho(x) = -cx$, so leftover inventory is valued as if it substitutes for future purchases. We have found that we can obtain better performance by taking the residual value to be $\rho(x) = -\kappa x$ and doing a grid search to identify good κ for a given problem. We typically find that the best κ satisfies $\kappa < c$, which suggests that it is better to assume that leftover inventory is valued at less than the ordering cost. Of course, other functional forms for the residual value, perhaps including time or state dependence, may perform even better.

There are a number of other heuristics one might explore. For example, Zipkin (2008b) considers a "myopic-2" heuristic that looks $L+2$ periods ahead and finds that it tends to outperform the myopic heuristic. In work in progress, Sun et al. (2014) consider heuristics based on $L^\natural$-convex quadratic approximations of the value functions. We will focus solely on the myopic heuristic, although the techniques we use could be used with these other heuristics as well.

5.4. Gradient Penalties and Dual Bounds

With perfect information about demands, we can write the inner problem (5) as a linear program, as in (30) but with cumulative order quantities z_t as decision variables as well as cumulative amount distributed s_t:

$$\begin{aligned} \min_{\mathbf{z},\mathbf{s}} \sum_{t=0}^{T+L} \gamma^t \big(\gamma^L c(z_t - z_{t-1}) + h(z_{t-L} - s_t) \\ + p(d_t - s_t + s_{t-1})\big) \\ \text{subject to } s_t \geqslant s_{t-1} \quad \text{for } t = 0, \ldots, T+L, \\ s_t \leqslant z_{t-L} \quad \text{for } t = 0, \ldots, T+L, \end{aligned} \tag{34}$$

where we take $s_\tau = 0$ and $z_\tau = 0$ when $\tau < 0$. If we include a penalty that is linear in the order quantities z_t (such as our gradient penalties), we simply add terms that are affine in z_t to the objective function. As with (30), given integer demands, the optimal z_t and s_t will be integers.

As discussed in BSS (2010), if we know the optimal policies for a dynamic program satisfy certain structural properties, we can restrict the dual problem to consider solutions that also satisfy these properties. Zipkin (2008a, Lemma 6) shows that when starting from a zero initial inventory position and following an optimal ordering policy, the optimal order quantities must satisfy critical fractile bounds, where the critical fractiles are defined for the total demand over several periods. Here we can add these linear constraints on order quantities to the linear program (34) for the inner problem and improve the bounds. We will include these critical fractile constraints in our numerical experiments, but we do not include the more complicated constraints on weighted sums of inventory levels that Zipkin (2008a, Corollary 10) develops.

We will consider gradient penalties based on the approximate value functions associated with the myopic heuristic. Specifically, we take the approximate value functions $\hat{V}_t$ for period t in the gradient penalty (15) to be the value function (30)–(31), but with the horizon T set to $t + L$; this is equivalent to adding the previously incurred costs to the myopic cost-to-go function (32). As in the myopic heuristic, we augment the terminal value (30) to include a residual term $\rho(-)$ to value leftover inventory. As this is a value function for a lost-sales model (albeit with a different time horizon), these value functions will be $L^\natural$-convex and we can use the result of Proposition 5.1(ii) to characterize the gradients. In addition to considering gradient penalties based on these myopic approximate value functions, we will also consider penalties based on optimal value functions in cases with a short lead time ($L = 4$), which can be solved exactly. We will take gradients around the order quantities selected in the myopic heuristic when working with the myopic value functions and around the optimal order quantities when working with the optimal value function.

We will consider two different methods for selecting gradients for the penalty. In both approaches, we use a variation of the greedy algorithm (see online appendix) to identify extreme points (or extreme rays) of the differential $\partial \hat{V}_t(\mathbf{z}_{t-1})$ given in Proposition 5.1(ii); finding such an extreme point requires evaluating the approximate value function $\hat{V}_t(\mathbf{z}_{t-1})$ a total of $t+1$ times. In the first approach, we use a simple 50–50 convex combination of two "extreme" extreme points of the differential. These gradients are easy to compute but generally will not satisfy the consistency condition (16). In the second approach, we use a more sophisticated procedure where we try to find an element of the differential that satisfies condition (16). Here we work forward in time as discussed following Proposition 3.2, selecting gradients for one period to match the selected gradient for the previous period. For each period, we sequentially generate extreme points of the differential (using the same variation on the greedy algorithm) until the convex hull includes an element satisfying the desired condition. If no such point exists, we instead take the point in the differential that is closest to satisfying (16). This approach is more time consuming than the first approach but generates gradients such that (16) is satisfied or approximately so if it is not possible to satisfy the condition exactly. The details of these two procedures are provided in the appendix.

5.5. Numerical Examples

We will consider four numerical examples, all adapted from Zipkin (2008b). In all cases, we assume a time horizon of $T = 40$, ordering costs $c = 0$, holding costs $h = 1$, penalty $p = 9$, and discount factor γ=1.[6] We will consider lead times $L = 4$ and 10; the model with $L = 4$ is small enough to solve exactly, whereas with $L = 10$, the model is much too large to solve exactly. In both cases, we consider Poisson and geometric demand distributions with mean 5. The geometric distribution includes more extreme demand scenarios and leads to a more difficult inventory management problem and, as we will see, much higher expected costs. We will study the myopic heuristic and dual bounds generated by gradient penalties based on the myopic heuristic and, for the $L = 4$ cases, based on the optimal value function. The results are summarized in Table 2.

For each set of parameters, we first run a few small-sample simulations to identify a good value κ to use in the residual value function $\rho(x) = -\kappa x$ that approximately capture the value of leftover inventory in the myopic heuristic and associated bounds. These values are reported at the top of Table 2. We also calculate exact value functions for the $L = 4$ cases. We then run a Monte Carlo simulation with 100 sample scenarios, with $T + L + 1$ demand realizations for each scenario. In each sample scenario, we do the following:

(i) We evaluate the myopic heuristic and adjust these values using control variates, as discussed in §3.3.

(ii) We then select gradients based on the myopic value function and optimal value function (for the $L = 4$ cases), using both the simple and sophisticated approaches. Using these selections with gradient penalties, we then solve the inner problem (34) (with the penalty term included in the objective) as a linear program. We also consider a bound where there is no penalty. In all cases, we enforce the critical fractile constraints discussed in §5.4 when solving the inner problems.

The average of the values from (i) and (ii) provide estimates of the upper and lower bounds on the optimal costs. These values and the mean standard errors (MSEs) associated with these estimates are shown in Table 2; the MSEs are calculated in the same way as in the network revenue management examples.

Table 2. Bounds and run times for lost sales example (100 samples).

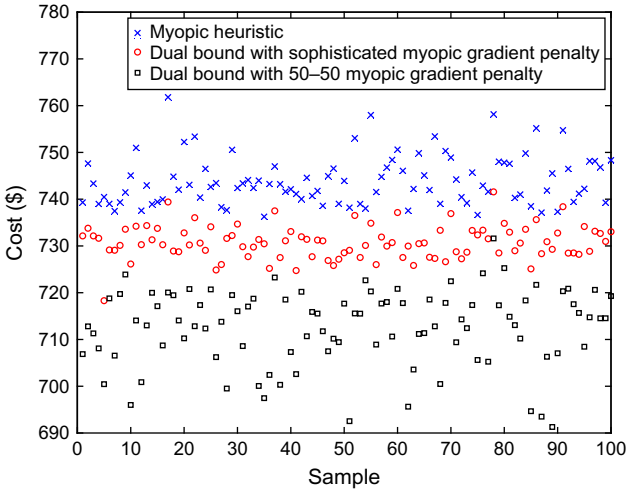
	Poisson demand		Geometric demand	
Upper bounds	$L=4$	$L=10$	$L=4$	$L=10$
Best κ value	0.95	1.325	0.675	0.9
Myopic heuristic				
Mean (MSE), \$	448 (0.16)	744 (0.53)	833 (0.65)	1,139 (1.28)
Run time, seconds	2	6	3	11
Exact value				
Value, \$	448	—	832	—
Run time, seconds	416	1,365		
Lower bounds				
Zero penalty				
Mean (MSE), \$	225 (4.45)	500 (5.94)	315 (8.36)	593 (11.0)
Gap as % of myopic heuristic (%)	49.8	32.8	62.2	47.9
Run time, seconds	2	2	2	2
50–50 gradient penalty based on myopic value function				
Mean (MSE), \$	430 (0.26)	713 (0.82)	806 (0.39)	1,092 (0.72)
Gap as % of myopic heuristic (%)	4.0	4.2	3.2	4.1
Run time, seconds	53	94	57	116
50–50 gradient penalty based on optimal value function				
Mean (MSE), \$	429 (0.41)	—	818 (0.35)	—
Gap as % of myopic heuristic (%)	4.4		1.8	
Run time, seconds	55		57	
Soph. gradient penalty based on myopic value function				
Mean (MSE), \$	440 (0.22)	731 (0.36)	816 (0.46)	1,099 (0.68)
Gap as % of myopic heuristic (%)	1.8	1.8	1.9	3.5
Run time, seconds	408	2,690	382	1,879
Soph. gradient penalty based on optimal value function				
Mean (MSE), \$	448 (0.00)	—	832 (0.00)	—
Gap as % of myopic heuristic (%)	0.1		0.1	
Run time, seconds	429	338		

As in the network revenue management examples, all computations were done in MATLAB using a standard desktop PC. We use Mosek to solve the inner problems (a linear program) as well as a quadratic program that arises in the sophisticated gradient selection procedure. The run times (shown in Table 2) are dominated by the time required to evaluate the (approximate) value functions when calculating gradients. The simple 50–50 gradient selection procedure requires evaluating the value functions in $2(t+1)$ times in period t. The sophisticated gradient selection procedure requires more evaluations as we search for a gradient satisfying condition (16). We have not attempted to optimize these computations: there are many redundant calculations and the run times could probably be significantly improved.

In terms of performance, we see that the myopic heuristic is very close to the exact value (within approximately 0.12%) for the cases where we can compute the exact value. The lower bound based on the optimal value functions with the sophisticated gradient selection procedure is sharp, yielding the exact value with zero variance, as guaranteed by Proposition 3.1(ii) and our choice of gradients. Reviewing these results and those for the other lower bounds, we see the importance of gradient selections: the sophisticated procedure outperforms the simple 50–50 procedure in every case. The gaps between the myopic heuristic values and the corresponding sophisticated gradient bounds based on the myopic heuristic are under 2% in all but the most difficult case, with geometric demand and $L=10$, where the gap is 3.5%. It is impossible to know whether this 3.5% gap (or the 1.9% gap for the other $L=10$ case) is mostly due to the suboptimality of the heuristic or suboptimality of the bound; we suspect that there is some of each here. In all cases, the zero penalty bounds are quite weak; this reflects the difficulty of the inventory management problem (with perfect information, the holding and penalty costs are nearly zero, except for the L initial "start up" periods where no orders have arrived and penalties are unavoidable) and the importance of using an effective penalty.

Figure 2 shows the sample results for the myopic heuristic (adjusted by the control variates) and the inner problems for the sophisticated and 50–50 gradient penalties based on the myopic value function, for the case with $L=10$, with Poisson demand. Here we see that the (adjusted) heuristic values are greater than both inner problem values in every sample, as they must be given the discussion in §3.3. We also see that the dual bounds based on the sophisticated gradients outperform the simpler 50–50 gradient selection

Figure 2. (Color online) Sample values for the myopic heuristic and inner problems for the lost-sales problem.



in every scenario. We have no theoretical guarantee that this will be the case (and indeed it is not always the case in some other trials), but these results reinforce the importance of selecting gradients carefully.

To examine the importance of the residual value function used to value leftover inventory in the myopic heuristic, we also evaluated this heuristic and associated gradient bounds using $\rho(x) = -cx$, as assumed in Zipkin (2008b) and elsewhere, rather than using $\rho(x) = -\kappa x$ and choosing a coefficient κ for the particular problem at hand. With $\kappa = c$, we find that the expected costs for the myopic heuristic increase by amounts ranging from 3.1% in the $L = 4$ case with Poisson demand to 7.3% in the $L = 10$ case with geometric demand. The expected costs given by the gradient bounds decrease by amounts ranging from approximately 12% to 24%, with the cost decreases being slightly larger in the cases with the simple 50–50 gradient selection. The results for the dual bounds are particularly sensitive as this change in residual coefficient has a significant effect on the gradients of the approximate value functions.

As shown in Table 2, with these optimized value of κ for the myopic heuristics and gradient bounds, we find duality gaps of less than 2% in three of the four cases and 3.5% in the fourth. These results may be "good enough" for most applications. If not, one can experiment with alternative forms for residual functions or more sophisticated heuristics in effort to narrow these gaps.

6. Conclusions

In this paper, we have studied the problem of calculating performance bounds for stochastic DPs with a convex structure through the use of information relaxations and gradient penalties. The main motivation for studying these penalties was computational tractability: the fact that gradient penalties are linear in actions implies that the inner problems, with perfect information, may be formulated and solved as deterministic convex optimization problems. In terms of quality of the resulting performance bounds, the analysis suggests these gradient penalties are effective: we can in theory obtain a tight, zero variance bound with the right choice of gradient penalty, and we can improve upon performance bounds from other relaxations using this approach.

We have considered two example applications that are interesting in their own right and demonstrate the usefulness of the information relaxation approach. The network revenue management application demonstrated how the dual bounds could be used to improve upon bounds given by a relaxed approximating model, here a Lagrangian relaxation, that can be solved to optimality. In the lost-sales application, the penalties were based on the limited lookahead value functions that are used with a myopic heuristic. In the network revenue management model, the rewards are linear but nondifferentiability arises through changes (or potential downstream changes) in binding constraints. In the lost-sales model, the nondifferentiability is embedded in the inventory cost functions and in the state dynamics where sales are lost if sufficient inventory is not on hand. In both cases, we exploit structural properties of the approximating model to calculate gradients and penalties. In the network revenue management model, the key feature is the ability to decompose the Lagrangian relaxation into leg-specific subproblems. In the lost-sales model, we use $L^\natural$-convexity of the approximating value functions to characterize the set of gradients. In other applications, one may need to think carefully about calculating gradients: it would be nice to have some general techniques that do not require knowledge of such problem-specific structure.

In recent related work, Desai et al. (2012) study performance bounds for a class of convex stochastic DPs. Their approach involves using a perfect information relaxation and a set of penalties that are linear in actions; the penalties are parametric functions of the uncertainties and they use large-scale optimization to find the best parameters for this set of penalties. We view their approach as complementary to the use of gradient penalties. The challenge with this optimization-based approach is that is not clear what parametric function to use; the analysis of gradient penalties provides insights into what these functions should look like. The challenge with the gradient penalty approach is that we may want to improve the bounds by searching over penalties from different approximate models; this search can be tedious. It would be interesting to consider a hybrid method that efficiently searches over penalties by starting with a gradient penalty from a given approximate model and then refines the bound using optimization.

Supplemental Material

Supplemental material to this paper is available at http://dx.doi.org/10.1287/opre.2014.1322.

Appendix. Proofs for §3

A.1. Proof of Lemma 3.1(viii)

PROOF. $\Rightarrow$: Given $(0, g) \in \partial f(x^*(y), y)$, for any w we have

$$\begin{aligned} f^*(w) &= f(x^*(w), w) \\ &\leqslant f(x^*(y), y) + 0^\top (x^*(w) - x^*(y)) + g^\top (w - y) \\ &= f^*(y) + g^\top (w - y). \end{aligned}$$

Thus $g \in \partial f^*(y)$.

$\Leftarrow$: Given $g \in \partial f^*(y)$, for any v, w we have

$$\begin{aligned} f(v, w) &\leqslant f^*(w) \leqslant f^*(y) + g^\top (w - y) \\ &= f(x^*(y), y) + 0^\top (v - x^*(y)) + g^\top (w - y). \end{aligned}$$

The first inequality follows from the definition of $f^*(w)$ as a maximum; the second inequality follows from the assumption that $g \in \partial f^*(y)$. Thus $(0, g) \in \partial f(x^*(y), y)$. □

A.2. Proof of Proposition 3.2

PROOF. Part (i) follows directly from Proposition 2.2. Part (iii) follows from part (ii) in the same way as in Proposition 2.2.

(ii) For the proof of part (ii), it is helpful to represent the constraints using characteristic functions. For a set X, let $\mathbb{1}_{A_t}(\mathbf{a}_t) = 0$ if $a_t \in A_t(\mathbf{a}_{t-1})$ and $-\infty$ otherwise; since A_t is a convex set, $\mathbb{1}_{A_t}$ is a concave function taking values on the extended real line.

Since, by assumption, the approximate value functions and policy correspond to those from the original model (i.e., $\hat{V}_t = V_t$ and $\hat{\alpha} = \alpha$), we will omit the "hats" from V and α in this proof. With this notation, we can rewrite the inner problem for a particular scenario as follows:

$$\begin{aligned} &\max_{\mathbf{a} \in A} \{ r(\mathbf{a}) - \hat{\pi}_\partial(\mathbf{a}) \} \\ &\quad = \max_{\mathbf{a}} \sum_{t=0}^{T} \big\{ r_t(\mathbf{a}_t) + \mathbb{1}_{A_t}(\mathbf{a}_t) + \big(\mathbb{E}[\boldsymbol{\delta}_t \,|\, \mathcal{F}_t] - \boldsymbol{\delta}_t\big)^\top (\mathbf{a}_t - \boldsymbol{\alpha}_t) \\ &\qquad\qquad + \big(\mathbb{E}[V_{t+1}(\boldsymbol{\alpha}_t) \,|\, \mathcal{F}_t] - V_{t+1}(\boldsymbol{\alpha}_t)\big) \big\}. \end{aligned} \tag{1}$$

We use $\boldsymbol{\alpha}_t$ to denote the sequence of actions up to time t chosen by policy α in this scenario ω, i.e., $\boldsymbol{\alpha}_t = (\alpha_0(\omega), \ldots, \alpha_t(\omega))$, and let $\boldsymbol{\alpha} = \boldsymbol{\alpha}_T$. We will show that $\mathbf{a} = \boldsymbol{\alpha}$ is an optimal solution to (1). Dropping terms that do not depend on $\mathbf{a}$ from the maximization problem on the right side of (1) and rearranging, this maximization problem is equivalent to

$$\max_{\mathbf{a}} \sum_{t=0}^{T} \big\{ r_t(\mathbf{a}_t) + \mathbb{1}_{A_t}(\mathbf{a}_t) + \big(\mathbb{E}[\boldsymbol{\delta}_t \,|\, \mathcal{F}_t] - (\boldsymbol{\delta}_{t-1}, \mathbf{0})\big)^\top \mathbf{a}_t \big\} \tag{2}$$

in that (1) and (2) have the same sets of optimal solutions. To parse the above notation, recall that V_{t+1} depends on actions $\mathbf{a}_t = (a_0, \ldots, a_t)$, and thus $\boldsymbol{\delta}_t$ has dimension equal to the dimension of $\mathbf{a}_t$. The $\mathbf{0}$ above has dimension equal to the dimension of a_t, so the dimensions of $\boldsymbol{\delta}_t$ and $(\boldsymbol{\delta}_{t-1}, \mathbf{0})$ are the same. We use the convention that $\boldsymbol{\delta}_{-1} = \varnothing$, so $(\boldsymbol{\delta}_{-1}, \mathbf{0}) = \mathbf{0}$.

Problem (2) is an unconstrained convex optimization problem with the constraints of the original problem captured through the characteristic functions in the objective. We will show that there exists a gradient selection $\boldsymbol{\delta} = (\boldsymbol{\delta}_0, \ldots, \boldsymbol{\delta}_T)$, where $\boldsymbol{\delta}_{t-1}(\omega) \in \partial V_t(\boldsymbol{\alpha}_{t-1}(\omega), \omega)$, such that $\mathbf{0}$ is in the differential of the objective function in (2) at $\mathbf{a} = \boldsymbol{\alpha}$. Since this objective is concave in actions, by Lemma 3.1(vi), this implies that $\mathbf{a} = \boldsymbol{\alpha}$ is an optimal choice of actions in (2).

Because the objective in (2) is a sum of concave functions and differentials commute with summation for concave functions (Lemma 3.1(iv)), to show that $\mathbf{a} = \boldsymbol{\alpha}$ is optimal in (2), it is sufficient to show that there exist gradients $\boldsymbol{\delta}_t$ such that $\mathbf{0}$ is in the differential for each term in the sum in (2); that is, it is sufficient to show that there exists a gradient selection $\boldsymbol{\delta} = (\boldsymbol{\delta}_0, \ldots, \boldsymbol{\delta}_T)$ such that, for all t,

$$\mathbf{0} \in \partial \{ r_t(\boldsymbol{\alpha}_t) + \mathbb{1}_{A_t}(\boldsymbol{\alpha}_t) \} + \big(\mathbb{E}[\boldsymbol{\delta}_t \,|\, \mathcal{F}_t] - (\boldsymbol{\delta}_{t-1}, \mathbf{0})\big),$$

or, equivalently,

$$(\boldsymbol{\delta}_{t-1}, \mathbf{0}) \in \partial \big\{ r_t(\boldsymbol{\alpha}_t) + \mathbb{1}_{A_t}(\boldsymbol{\alpha}_t) \big\} + \mathbb{E}[\boldsymbol{\delta}_t \,|\, \mathcal{F}_t]. \tag{3}$$

Condition (3) generalizes the "consistency condition" (13) to the case with constraints and nondifferentiable rewards. We show (3) by forward induction: we first show this condition holds for $t = 0$ and then, assuming it holds for the first t periods, we show that it also holds for the first $t + 1$ periods.

For $t = 0$, since $\boldsymbol{\delta}_{-1} = \varnothing$, (3) becomes

$$\mathbf{0} \in \partial \{ r_0(\alpha_0) + \mathbb{1}_{A_0}(\alpha_0) \} + \mathbb{E}[\boldsymbol{\delta}_0 \,|\, \mathcal{F}_0]; \tag{4}$$

we will show that there exist $\boldsymbol{\delta}_0$ satisfying this condition. By definition, α_0 is optimal for the problem

$$\max_{a_0} \big\{ r_0(a_0) + \mathbb{1}_{A_0}(a_0) + \mathbb{E}[V_1(a_0) \,|\, \mathcal{F}_0] \big\}.$$

Since $V_1(a_0)$ is concave in a_0, this is a convex optimization problem and, by Lemma 3.1(vi), optimality of α_0 implies

$$\mathbf{0} \in \partial \big\{ r_0(\alpha_0) + \mathbb{1}_{A_0}(\alpha_0) + \mathbb{E}[V_1(\alpha_0) \,|\, \mathcal{F}_0] \big\}. \tag{5}$$

By Lemma 3.1(iv), this means that there exists gradients $\mathbf{g}_1$ and $\mathbf{g}_2$ such that

$$\mathbf{g}_1 \in \partial \big\{ r_0(\alpha_0) + \mathbb{1}_{A_0}(\alpha_0) \big\}, \quad \mathbf{g}_2 \in \partial \mathbb{E}[V_1(\alpha_0) \,|\, \mathcal{F}_0], \quad \text{and} \quad \mathbf{g}_1 + \mathbf{g}_2 = \mathbf{0}.$$

By Lemma 3.1(vii), we can interchange expectations and the differential operator and we have

$$\mathbf{g}_2 \in \partial \mathbb{E}[V_1(\alpha_0) \,|\, \mathcal{F}_0] = \mathbb{E}[\partial V_1(\alpha_0) \,|\, \mathcal{F}_0].$$

This implies the existence of a selection of gradients $\boldsymbol{\delta}_0(\omega) \in \partial V_1(\alpha_0(\omega), \omega)$ such that $\mathbf{g}_2 = \mathbb{E}[\boldsymbol{\delta}_0 \,|\, \mathcal{F}_0]$. Since $\mathbf{g}_1 + \mathbf{g}_2 = \mathbf{0}$, (4) holds for the selection of gradients $\boldsymbol{\delta}_0$.

Now assume that (3) holds for t periods for some gradient selection $(\boldsymbol{\delta}_0, \ldots, \boldsymbol{\delta}_{t-1})$ with $\boldsymbol{\delta}_{t-1} \in \partial V_t(\boldsymbol{\alpha}_{t-1})$. We will show that there is a $\boldsymbol{\delta}_t$ such that (3) holds for $t + 1$ periods for the gradient selection $(\boldsymbol{\delta}_0, \ldots, \boldsymbol{\delta}_{t-1}, \boldsymbol{\delta}_t)$. The proof is similar to that for the $t = 0$ case. By definition, $a_t = \alpha_t$ is optimal for the problem

$$\begin{aligned} V_t(\boldsymbol{\alpha}_{t-1}) = \max_{a_t} \big\{ & r_t(\boldsymbol{\alpha}_{t-1}, a_t) + \mathbb{1}_{A_t}(\boldsymbol{\alpha}_{t-1}, a_t) \\ & + \mathbb{E}[V_{t+1}(\boldsymbol{\alpha}_{t-1}, a_t) \,|\, \mathcal{F}_{t-1}] \big\}. \end{aligned}$$

Then, since this is a convex optimization problem, by the "stacking gradient" result (Lemma 3.1(viii)), we know that $\boldsymbol{\delta}_{t-1} \in \partial V_t(\boldsymbol{\alpha}_{t-1})$ implies

$$\begin{aligned}(\boldsymbol{\delta}_{t-1}, \mathbf{0}) &\in \partial\big\{r_t(\boldsymbol{\alpha}_{t-1}, \alpha_t) + \mathbb{1}_{A_t}(\boldsymbol{\alpha}_{t-1}, \alpha_t) \\ &\qquad + \mathbb{E}[V_{t+1}(\boldsymbol{\alpha}_{t-1}, \alpha_t) \,|\, \mathcal{F}_t]\big\} \\ &= \partial\big\{r_t(\boldsymbol{\alpha}_t) + \mathbb{1}_{A_t}(\boldsymbol{\alpha}_t) + \mathbb{E}[V_{t+1}(\boldsymbol{\alpha}_t) \,|\, \mathcal{F}_t]\big\}.\end{aligned}$$

By Lemma 3.1(iv), this implies that there exists gradients $\mathbf{g}_1$ and $\mathbf{g}_2$ such that

$$\mathbf{g}_1 \in \partial\{r_t(\boldsymbol{\alpha}_t) + \mathbb{1}_{A_t}(\boldsymbol{\alpha}_t)\}, \quad \mathbf{g}_2 \in \partial\mathbb{E}[V_{t+1}(\boldsymbol{\alpha}_t) \,|\, \mathcal{F}_t], \quad \text{and}$$
$$\mathbf{g}_1 + \mathbf{g}_2 = (\boldsymbol{\delta}_{t-1}, \mathbf{0}).$$

By Lemma 3.1(vii), we can interchange expectations and the differential operator and we have

$$\mathbf{g}_2 \in \partial\mathbb{E}[V_{t+1}(\boldsymbol{\alpha}_t) \,|\, \mathcal{F}_t] = \mathbb{E}[\partial V_{t+1}(\boldsymbol{\alpha}_t) \,|\, \mathcal{F}_t].$$

This implies the existence of gradients $\boldsymbol{\delta}_t$ from $\partial V_{t+1}(\boldsymbol{\alpha}_t)$ such that $\mathbf{g}_2 = \mathbb{E}[\boldsymbol{\delta}_t \,|\, \mathcal{F}_t]$. Since $\mathbf{g}_1 + \mathbf{g}_2 = (\boldsymbol{\delta}_{t-1}, \mathbf{0})$, (3) holds for the selection of gradients $(\boldsymbol{\delta}_0, \ldots, \boldsymbol{\delta}_t)$. This completes our inductive proof and we have established that $\mathbf{a} = \boldsymbol{\alpha}$ is an optimal choice of actions for (2) and, therefore, also for (1).

Now we can rewrite the inner problem for a given scenario and simplify as follows:

$$\begin{aligned}&\max_{\mathbf{a}\in A}\{r(\mathbf{a}) - \hat{\pi}_\partial(\mathbf{a})\} \\ &= \max_{\mathbf{a}} \sum_{t=0}^{T} \big\{r_t(\mathbf{a}_t) + \mathbb{1}_{A_t}(\mathbf{a}_t) + (\mathbb{E}[\boldsymbol{\delta}_t \,|\, \mathcal{F}_t] - \boldsymbol{\delta}_t)^\top(\mathbf{a}_t - \boldsymbol{\alpha}_t) \\ &\qquad\qquad + (\mathbb{E}[V_{t+1}(\boldsymbol{\alpha}_t) \,|\, \mathcal{F}_t] - V_{t+1}(\boldsymbol{\alpha}_t))\big\}, \\ &= \sum_{t=0}^{T} \big\{r_t(\boldsymbol{\alpha}_t) + \mathbb{1}_{A_t}(\boldsymbol{\alpha}_t) + \big(\mathbb{E}[V_{t+1}(\boldsymbol{\alpha}_t) \,|\, \mathcal{F}_t] - V_{t+1}(\boldsymbol{\alpha}_t)\big)\big\} \\ &= \sum_{t=0}^{T} \big\{V_t(\boldsymbol{\alpha}_{t-1}) - V_{t+1}(\boldsymbol{\alpha}_t)\big\} \\ &= V_0. \qquad (6)\end{aligned}$$

The first equality repeats (1). The second equality uses the previously established fact that $\mathbf{a} = \boldsymbol{\alpha}$ is an optimal choice of actions for (1). For the next equalities, we use the definition of the value function and the fact that $V_{T+1} = 0$. This completes the proof that there exists a gradient selection such that the inner problem yields V_0 for each scenario.

To see that the inner problem yields values greater than V_0 for each scenario for any gradient selection based on V_t and α, reconsider the sequence of equalities (6). For any gradient selection, the first, third, and fourth equalities continue to hold. The second equality holds with inequality ($\geqslant$) rather than equality, because $\mathbf{a} = \boldsymbol{\alpha}$ is a feasible but not necessarily optimal choice for the optimization problem in the second line. □

Endnotes

1. This can be established with a recursive proof: $V_{T+1} = 0$ is trivially concave. If V_{t+1} is concave in $\mathbf{a}_t$, then so is $r_t + \mathbb{E}[V_{t+1} \,|\, \mathcal{F}_t]$, and, finally, V_t, as the partial maximization of a concave function over a convex set, is also concave.

2. When the itineraries use at most one unit of capacity on each leg (i.e., $f_{il} \in \{0, 1\}$), it can be shown that the optimal decisions in (21) will always be 0 or 1 for each leg even if noninteger values are allowed. Thus, in this case, convexifying the set of actions is without loss of optimality. This convexification of the action set is not required for the results of Proposition 4.1, but is required to make the Lagrangian relaxation a convex DP.

3. The representation in Topaloglu (2009) includes terms of the form $\max\{r_{i_t} - \sum_{l\in\mathcal{L}} \lambda_{i_t l t}, 0\}$; these terms are equal to zero when $\boldsymbol{\lambda} \in \Lambda$ due to condition (c). Kunnumkal and Talluri (2012) study the relationship between the Lagrangian relaxations and approximate dynamic programming approaches for this network revenue management model. Among other things, they establish a result showing that a Lagrangian relaxation similar to (21) can be written as the sum of piecewise linear leg-specific value functions, similar to (24).

4. The one-hub example is problem instance $(200, 8, 1.0, 4)$ in Topaloglu (2009) and is available at http://people.orie.cornell.edu/huseyin/research/rm_datasets/rm_datasets.html. The data set for the two-hub example was provided by Huseyin Topaloglu in a private communication (June 2013); we are grateful for his help in sharing these examples.

5. Zipkin (2008a) showed that $f_t(\mathbf{x}_t)$ defined by Equation (29) is $L^\natural$-convex in an accumulated version of the state variable x_t, whereas we establish $L^\natural$-convexity of J_t as a function of all prior accumulated order quantities. We use a similar argument to establish this result.

6. Zipkin (2008b) notes that any lost-sales model with nonzero costs is equivalent (after transformation of other parameters) to a model with $c = 0$. However, this equivalent $c = 0$ model understates the total inventory costs for the original system, as it does not include ordering costs. Therefore, in percentage terms, the duality gaps in our experiments are larger than they would be if the ordering costs were included.

References

Adelman D, Mersereau AJ (2008) Relaxations of weakly coupled stochastic dynamic programs. *Oper. Res.* 56(3):712–727.

Andersen L, Broadie M (2004) Primal-dual simulation algorithm for pricing multidimensional American options. *Management Sci.* 50(9): 1222–1234.

Bertsekas D (1973) Stochastic optimization problems with nondifferentiable cost functionals. *J. Optim. Theory Appl.* 12(2):218–231.

Bertsekas D, Nedić A, Ozdağlar A (2003) *Convex Analysis and Optimization* (Athena Scientific, Belmont, MA).

Brown DB, Smith JE (2011) Dynamic portfolio optimization with transaction costs: Heuristics and dual bounds. *Management Sci.* 57(10): 1752–1770.

Brown DB, Smith JE, Sun P (2010) Information relaxations and duality in stochastic dynamic programs. *Oper. Res.* 58(4):785–801.

Desai V, Farias VF, Moallemi CC (2012) Pathwise optimization for linear systems with convex costs. Working paper, Graduate School of Business, Columbia University.

Devalkar S, Anupindi R, Sinha A (2011) Integrated optimization of procurement, processing, and trade of commodities. *Oper. Res.* 59(6): 1369–1381.

Haugh MB, Kogan L (2004) Pricing American options: A duality approach. *Oper. Res.* 52(2):258–270.

Haugh MB, Lim AEB (2012) Linear-quadratic control and information relaxations. *Oper. Res. Lett.* 40(6):521–528.

Haugh MB, Iyengar G, Wang C (2014) Tax-aware Dynamic Asset Allocation. Working paper, Columbia University, New York.

Hawkins J (2003) A Lagrangian decomposition approach to weakly coupled dynamic optimization problems and its applications. Ph.D. thesis, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA.

Henderson SG, Glynn PW (2002) Approximating martingales for variance reduction in Markov process simulation. *Math. Oper. Res.* 27(2): 253–271.

Karlin S, Scarf H (1958) Inventory models of the Arrow-Harris-Marschak type with time lag. Arrow K, Karlin S, Scarf H, eds. *Studies in the Mathematical Theory of Inventory and Production* (Stanford University Press, Stanford, CA), 155–178.

Kunnumkal S, Talluri KT (2012) Equivalence of piecewise-linear approximation and Lagrangian relaxation for network revenue management. Working paper, Universitat Pompeu Fabra, Barcelona, Spain.

Lai G, Margot F, Secomandi N (2010) An approximate dynamic programming approach to benchmark practice-based heuristics for natural gas storage valuation. *Oper. Res.* 58(3):564–582.

Morton T (1971) The near-myopic nature of the lagged-proportional-cost inventory problem with lost sales. *Oper. Res.* 19(7):1708–1716.

Murota K (2003) *Discrete Convex Analysis* (SIAM, Philadelphia).

Nadarajah S, Margot F, Secomandi N (2014) Relaxations of approximate linear programs for the real option management of commodity storage. Working paper. Tepper School of Business, Carnegie Mellon University, Pittsburgh.

Rockafellar R, Wets R (1976) Nonanticipativity and $\mathscr{L}^1$-martingales in stochastic optimization problems. *Math. Programming Study* 6: 170–187.

Rogers LCG (2002) Monte Carlo valuation of American options. *Math. Finance* 12:271–286.

Rogers LCG (2007) Pathwise stochastic optimal control. *SIAM J. Control Optim.* 46:1116–1132.

Secomandi N (2014) Analysis and enhancement of practice-based policies for the real option management of commodity storage assets. Working paper, Carnegie Mellon University, Pittsburgh.

Shapiro A, Dentcheva D, Ruszczyński A (2009) *Lectures on Stochastic Programming: Modeling and Theory*, MOS-SIAM Series on Optimization, Vol. 9 (SIAM, Philadelphia).

Sun P, Wang K, Zipkin P (2014) Heuristics for lost-sales inventory systems based on quadratic approximation of *L*-natural convex value functions. Working paper, Fuqua School of Business, Duke University, Durham, NC.

Topaloglu H (2009) Using Lagrangian relaxation to compute capacity-dependent bid prices in network revenue management. *Oper. Res.* 57(3):637–649.

Zipkin P (2008a) On the structure of lost-sales inventory models. *Oper. Res.* 56(4):937–944.

Zipkin P (2008b) Old and new methods for lost-sales inventory systems. *Oper. Res.* 56(5):1256–1263.

David B. Brown is an associate professor of Decision Sciences at the Fuqua School of Business, Duke University. His research focuses on developing methods for solving optimization problems involving uncertainty.

James E. Smith is J.B. Fuqua Professor of Business Administration at the Fuqua School of Business, Duke University. His research interests are primarily in decision analysis and focus on developing methods for formulating and solving dynamic decision problems. He is an INFORMS Fellow.