We consider a broad class of stochastic dynamic programming problems that are amenable to relaxation via decomposition. These problems comprise multiple subproblems that are independent of each other except for a collection of coupling constraints on the action space. We fit an additively separable value function approximation using two techniques, namely, Lagrangian relaxation and the linear programming (LP) approach to approximate dynamic programming. We prove various results comparing the relaxations to each other and to the optimal problem value. We also provide a column generation algorithm for solving the LP-based relaxation to any desired optimality tolerance, and we report on numerical experiments on bandit-like problems. Our results provide insight into the complexity versus quality trade-off when choosing which of these relaxations to implement.

Subject classifications: dynamic programming/optimal control: approximate dynamic programming, Lagrangian optimization, discounted infinite horizon; linear programming: column generation.

Area of review: Stochastic Models.

History: Received June 2006; revision received October 2006; accepted January 2007. Published online in Articles in Advance January 14, 2008.

1. Motivation

A classical result in integer programming, due to Geoffrion (1974), relates the optimal objective value of the Lagrangian dual with that of the linear programming (LP) relaxation. Consider an integer program that maximizes a linear objective function over linear constraints that can be divided into two groups: so-called “easy” constraints, and “complicating” constraints. The idea of Lagrangian relaxation is to dualize the complicating constraints with a penalty vector $\lambda$, i.e., add terms to the objective function penalizing constraint violations, and then optimize the resulting objective function over the remaining “easy” constraints. For any penalty vector, this gives an upper bound on the optimal objective value of the integer program, $z_{IP}$. The Lagrangian dual minimizes over $\lambda$ to find the smallest upper bound, $z_{LD}$. A result of Geoffrion (1974) implies that

$$z_{IP} \leq z_{LD} \leq z_{LP},$$

where $z_{LP}$ is the optimal objective value of the LP relaxation of the integer program. This result provides an important justification for considering Lagrangian relaxations of integer programs, and Geoffrion’s (1974) work paved the way for many papers (Fisher’s 1981) that used the Lagrangian approach.

In recent years, there has been considerable research interest in solving stochastic dynamic optimization problems. Because static, deterministic special cases of these problems may indeed be integer programs, the class of stochastic dynamic optimization problems is at least as difficult to solve as the class of integer programs. Although dynamic optimization problems can in theory be solved using exact dynamic programming (Bellman 1957), this proves intractable for many problems in practice, due to the so-called “curse of dimensionality.” Although the problems in this area are of significant practical interest, there is currently a scarcity of theory to guide the practitioner. Competing techniques abound, and practical dynamic optimization often requires problem-specific approaches.

Various proposals have been made to break the curse, all of which fall under the heading of “approximate dynamic programming” methods. However, to date there is no result like (1) that compares relaxations. The purpose of this paper is to investigate an analogous result for two relaxations of a special class of such stochastic, dynamic optimization problems, called weakly coupled dynamic programs. Such problems decompose into independent dynamic programs, or subproblems, except for a set of linking constraints on the actions. When these constraints bind weakly, we expect the relaxations to be close to the original problem.

Examples include:

- **Multiarmed and restless bandit problems.** The restless bandit problem (Whittle 1988) is a classical resource...
allocation problem in which several Markov decision problems are controlled subject to a single linking constraint. A special case, the multiarmed bandit problem, can be efficiently decomposed and solved using the so-called Gittins’ indices (Gittins and Jones 1974), but the more general problem (Whittle 1988) has been shown to be PSPACE-hard (Papadimitriou and Tsitsiklis 1994).

- **Inventory routing with direct shipment.** A firm replenishes inventory over time to a set of geographically dispersed customers, using a fleet of vehicles. Vehicles replenish exactly one customer at a time, known as direct shipment (Kleywegt et al. 2002). The problem decomposes by customer except for a linking constraint, representing the limited number of vehicles, that bounds the number of customers that can be replenished in a period.

- **Stochastic multiproduct dispatch problems.** Multiple types of product arrive stochastically over time and wait to be dispatched by a single vehicle with a finite capacity (Papadaki and Powell 2003). There is a fixed cost to dispatch, and there is a product-dependent holding cost per unit per period. The action space designates how many units of each product to dispatch, which are linked due to the limited carrying capacity of the vehicle.

Suppose that there are \( I \) subproblems, and suppose that the state space \( S \subseteq \mathbb{R}^I \) is the product space of subproblem state spaces. Let \( \bar{A}(s) \subseteq \mathbb{R}^I \) be the set of actions that can be taken from state \( s \in S \), and let \( r(s, a) \) be the reward, additively separable across subproblems, for taking action \( a \) in state \( s \). For each state \( s \), the action set \( \bar{A}(s) \) includes constraints that link actions across subproblems. Then the Bellman optimality equations in the discounted infinite-horizon case are

\[
J(s) = \max_{a \in \bar{A}(s)} \left[ r(s, a) + \beta E[J(s')] \mid (s, a) \right], \quad s \in S,
\]

where \( 0 < \beta < 1 \) is the discount factor. For a weakly coupled dynamic program, a natural approximation to the value function is the additively separable form

\[
J(s) \approx \theta + \sum_{i} V_i(s_i), \quad s = \{s_i\} \in S,
\]

(2)

where \( V_i(\cdot) \) is the approximate value function for subproblem \( i \), and \( \theta \) is a constant offset. Whereas the original value function \( J(s) \) is tractable to compute, or even to encode, for problems of practical interest, the approximate value function is not. Given such an approximation, it can then be used in the right-hand side of Bellman’s equation to obtain a control policy, which is tractable due to the decomposability of the expectation when the transition probabilities also decompose by subproblem. When there is special structure, the maximization can be solved efficiently.

There are two main classes of techniques for computing decomposed value function components like \( V_i(s_i) \), namely, simulation and mathematical programming. Examples of the former include the neuro-dynamic programming approach of Bertsekas and Tsitsiklis (1996) and the gradient-based techniques of Powell and Topaloglu (2003). The mathematical programming approach is due to Schweitzer and Seidmann (1985), and was recently reconsidered by de Farias and Van Roy (2003). The idea is to formulate the optimality equations as a linear program and substitute the approximate (2) in the formulation. This approach has been employed recently in the context of stochastic inventory-routing by Adelman (2004). An alternative mathematical programming approach is Lagrangian relaxation, whereby the linking constraints are dualized. This approach is the subject of focused study by Hawkins (2003), and has been applied in various contexts by Bertsimas and Mersereau (2007) for adaptive online marketing, by Talluri and van Ryzin (1998) for revenue management, and by Castaño (1997, 2005) and Yost and Washburn (2000) for military targeting. Whittle (1988) uses a similar Lagrangian relaxation to generate a heuristic index policy for the restless bandit problem.

Fix a state \( s \in S \). Let \( J^{LP}(s) \) be the approximate value of this state calculated using the LP approach, and let \( J^{\lambda}(s) \) be the approximate value of the state computed using the Lagrangian approach, both having the form (2). One of our key theorems implies that

\[
J(s) \leq J^{LP}(s) \leq J^{\lambda}(s).
\]

More generally, this result holds for the sum of value functions computed for and weighted by a fixed probability distribution \( \alpha(\cdot) \) on the initial state.

The LP-based relaxation thus gives a stronger bound, but we will see that it is also more difficult to compute. This motivates a need to understand whether the bounds are significantly different, and if so, when the difference is most significant. To that end, we provide

- necessary and sufficient conditions for the Lagrangian relaxation to give a tight bound on the optimal value, leading to insights on when we expect the Lagrangian to perform relatively well and relatively poorly;
- sufficient conditions for the two bounds to be equal, leading to an interpretation of the Lagrangian bound as dualizing constraints in the reduced profit maximization problem corresponding to the LP-based bound problem;
- an example instance in which the Lagrangian policy and bound perform arbitrarily poorly relative to both the optimal solution and the LP-based solution;
- an example instance in which the gap between the Lagrangian bound and the optimal value grows linearly with the number \( I \) of subproblems;
- theoretical conditions, in the LP-based relaxation and in a state-dependent variant of the Lagrangian relaxation, for the average relaxation gap to vanish as the number of subproblems grows.

On the algorithmic side, we show how to use column generation to solve the approximate linear program within any given optimality tolerance. We give computational results on bandit-like problems, an important subclass of
2. Weakly Coupled Dynamic Programs

We define the class of problems and the two relaxations we consider.

2.1. Problem Description and Notation

Suppose that there are \( I \) subproblems that are each Markov decision problems on disjoint state spaces. Corresponding to subproblem \( i \), define the following:

- State space \( S_i \), assumed finite.
- Control space \( A_i(s_i) \), depending on the current state \( s_i \) and assumed finite for all \( s_i \in S_i \).
- Markov transition probabilities \( p_i(s_i' \mid s_i, a_i) \) for all \( a_i \in A_i(s_i) \) and \( s_i, s_i' \in S_i \). Here we indicate the current state by \( s_i \) and the next state in time as \( s_i' \). Conditional on the local state \( s_i \) and action \( a_i \), transitions are assumed to be independent of other subproblems.
- Expected reward \( r_i(s_i, a_i) \) accruing in state \( s_i \) when control \( a_i \) is administered.

The overall problem is a collection of subproblems of this form solved simultaneously subject to a set of \( N \) linking constraints of the form \( \sum_{i=1}^{I} D_i(s, a) \leq b \), where \( b \in \mathbb{R}^N \) and \( D_i : \{ (s, a) : s_i \in S_i, a_i \in A_i(s_i) \} \rightarrow \mathbb{R}^N \). We denote column vector quantities using boldface type, and we use the symbol \( \tau \) to indicate the transpose. We use the following notation for the overall problem:

- State space \( S = \times_{i=1}^{I} S_i \), where \( \times \) indicates the Cartesian product.
- Control space \( \hat{A} = \{ a \in A(s) : D(s, a) \leq b \} \), where \( A(s) = \times_{i=1}^{I} A_i(s_i) \) and \( D(s, a) = \sum_{i=1}^{I} D_i(s, a_i) \). We assume \( A(s) \neq \emptyset \) for all \( s \in S \).
- Transition probabilities \( p(s' \mid s, a) = \prod_{i=1}^{I} p_i(s_i' \mid s_i, a_i) \) for all \( s \in S, a \in A(s) \).
- Expected rewards \( r(s, a) = \sum_{i=1}^{I} r_i(s_i, a_i) \) for all \( s \in S \) and \( a \in A(s) \).

Given these definitions, we can state the Bellman optimality equations as

\[
J(s) = \max_{a \in A(s)} \left\{ r(s, a) + \beta \sum_{s' \in S} p(s' \mid s, a) J(s') \right\}, \quad s \in S. \tag{3}
\]

It will often be convenient to refer to sets of state-action pairs. For the sake of conciseness, we introduce the following sets:

\[
\mathcal{A}_i = \{ (s_i, a_i) : s_i \in S_i, a_i \in A_i(s_i) \},
\]

\[
\mathcal{A} = \{ (s, a) : s \in S, a \in A(s) \},
\]

\[
\mathcal{A}_i = \{ (s_i, a_i) : s_i \in S_i, a_i \in A_i(s_i) \},
\]

\[
\mathcal{A}_i = \{ (s_i, a_i) : s_i \in S_i, a_i \in A_i(s_i) \}.
\]

\[
\mathcal{A}(i) = \{ (s_i, a) : s_i \in S_i \}.
\]

For clarity of exposition, we focus on the discounted infinite-horizon case, but the relaxations are applicable in other settings as well.

2.2. The LP Approach to Dynamic Programming

We make use of the LP approach to dynamic programming (see Derman 1970). A linear program formulation corresponding to the optimality Equation (3) is

\[
H(\alpha) = \min_{V(s)} \sum_{s \in S} \alpha(s) V(s) \quad \text{s.t.} \quad V(s) \geq r(s, a) + \beta \sum_{s' \in S} p(s' \mid s, a) V(s'),
\]

\[
(s, a) \in \mathcal{A}. \tag{4}
\]

The coefficients \( \alpha(s) \) for all \( s \in S \) are nonnegative weights. Without loss of generality, we assume that these weights sum to one, in which case we can interpret them as a distribution on the initial problem state. Given this interpretation, we denote marginal initial state probabilities for subproblem \( i \) by \( \alpha_i(s_i) = \sum_{s_i' : s_i = s_i'} \alpha(s) \). The optimal objective of problem (4) is the weighted value function

\[
H(\alpha) = \sum_{s \in S} \alpha(s) J(s). \tag{5}
\]

It is natural to focus on the case \( \alpha(s) > 0 \) for all \( s \in S \) because under this assumption the value functions for all states can be computed by solving a single instance of problem (4). In this case, all of the decision variables \( V(s) \) in the linear program at optimality are equal to the value functions \( J(s) \). However, we will not always restrict ourselves to this case. If instead we merely have \( \alpha(s) \geq 0 \) for all \( s \in S \), then the optimal values of the decision variables in problem (4) will correspond to \( J(s) \) for at least those states \( s \) for which \( \alpha(s) > 0 \), and subsequent states. For details, we refer the reader to Derman (1970).

The computational effort required to solve problem (4) is considerable. The linear program has \( \prod_{i=1}^{I} |S_i| \) variables and \( |\mathcal{A}(i)| \) constraints. The latter number may be as high as \( \prod_{i=1}^{I} \sum_{s \in S_i} |A_i(s_i)| \). Thus, the numbers of constraints and variables grow exponentially with the number \( I \) of subproblems.

2.3. Lagrangian Relaxation

The first approximation approach we consider is Lagrangian relaxation, which leads to a problem decomposition. We relax the linking constraints and add Lagrangian terms to
the objective function. The resulting Bellman iteration is as follows for any \( \lambda \in \mathbb{R}_+^I \).

\[
J^\lambda(s) = \max_{a} \sum_{i=1}^{I} r_i(s_i, a_i) + \lambda^T \left[ b - \sum_{i=1}^{I} D_i(s_i, a_i) \right] \\
\quad + \beta \sum_{s' \in S} J^\lambda(s') \cdot \prod_{i=1}^{I} p_i(s'_i | s_i, a_i) \\
s.t. \ a_i \in A_i(s_i), \quad i \in \{1, \ldots, I\}. \tag{5}
\]

The Bellman equations decompose, as stated in the following proposition.

**Proposition 1.** For all \( s \in S \), the function \( J^\lambda(s) \) can be written as

\[
J^\lambda(s) = \frac{1}{1 - \beta} \lambda^T b + \sum_{i=1}^{I} V_i^\lambda(s_i), \tag{6}
\]

where the quantity \( V_i^\lambda(s_i) \) solves

\[
V_i^\lambda(s_i) = \max_{a_i \in A_i(s_i)} r_i(s_i, a_i) - \lambda^T D_i(s_i, a_i) \\
+ \beta \sum_{s'_i \in S} p_i(s'_i | s_i, a_i)V_i^\lambda(s'_i), \quad s_i \in S_i. \tag{7}
\]

Although a proof appears in Hawkins (2003), we present a simple alternative proof in online Appendix A. An electronic companion to this paper is available as part of the online version that can be found at http://or.journal.informs.org/.

As in §2.2, we can define a value function based on the initial state distribution as the weighted sum

\[
H^\lambda(\alpha) = \sum_{s \in S} \alpha(s) J^\lambda(s). \]

Clearly, \( H^\lambda(\alpha) \geq H(\alpha) \) for any \( \lambda \geq 0 \). In addition, the Lagrangian relaxation gives a statewise bound on the optimal value function \( J^\lambda(s) \) for any \( \lambda \geq 0 \).

**Proposition 2.** For any \( \lambda \geq 0 \), \( J^\lambda(s) \geq J(s) \) for all \( s \in S \).

**Proof.** First, observe that from feasibility in system (3) and because \( b - \sum_{i=1}^{I} D_i(s_i, a_i) \geq 0 \) for all \( a \in \tilde{A}(s) \), we have for any \( \lambda \geq 0 \) and any \( s \in S \),

\[
J(s) \leq \max_{a \in \tilde{A}(s)} \left\{ \sum_{i=1}^{I} r_i(s_i, a_i) + \lambda^T \left[ b - \sum_{i=1}^{I} D_i(s_i, a_i) \right] \\
\quad + \beta \sum_{s' \in S} p(s' | s, a) J(s') \right\}.
\]

Because \( \tilde{A}(s) \subseteq A(s) \), we have

\[
J(s) \leq \max_{a \in A(s)} \left\{ \sum_{i=1}^{I} r_i(s_i, a_i) + \lambda^T \left[ b - \sum_{i=1}^{I} D_i(s_i, a_i) \right] \\
\quad + \beta \sum_{s' \in S} p(s' | s, a) J(s') \right\}.
\]

Because this holds for any \( s \in S \), we have

\[
J \leq \sum_{i=1}^{I} J_i(\lambda),
\]

where \( \sum_{i=1}^{I} J_i(\lambda) \) is the Bellman operator corresponding to the Lagrangian problem (5). Thus, it follows from the monotonicity of the Bellman operator (see Puterman 1994, Theorem 6.2.2) that \( J^\lambda(s) \geq J(s) \), and thus that \( J^\lambda(s) \geq J(s) \).

It remains to identify a suitable \( \lambda \geq 0 \). We seek the tightest possible bound on \( H(\alpha) \). Define

\[
H^\lambda(\alpha) = \min_{\lambda \geq 0} \frac{1}{1 - \beta} \lambda^T b + \sum_{i=1}^{I} \alpha(s_i) V_i^\lambda(s_i)
\]

\[
= \min_{\lambda \geq 0} \frac{1}{1 - \beta} \lambda^T b + \sum_{i=1}^{I} \alpha(s_i) V_i^\lambda(s_i).
\]

It is straightforward to show that \( H^\lambda(\alpha) \) is a convex function of \( \lambda \).

Now consider the linear program for given \( i \in \{1, \ldots, I\} \):

\[
\min_{V_i(\cdot)} \alpha_i(s) V_i(s_i)
\]

s.t. \( V_i(s_i) \geq r_i(s_i, a_i) - \lambda^T D_i(s_i, a_i) \\
+ \beta \sum_{s'_i \in S} p_i(s'_i | s_i, a_i)V_i(s'_i), \quad (s_i, a_i) \in \mathbb{R}_i, \tag{9}
\]

which is the linear program formulation of the dynamic program (7). The solution to (7) solves this linear program (9). Also, if \( \alpha_i(s_i) > 0 \) for all \( s_i \in S_i \), then a solution to (9) solves (7). Combining these problems for all \( i \) gives the following linear program:

\[
(L\lambda): \quad H^\lambda(\alpha) = \frac{\lambda^T b}{1 - \beta} + \min_{V_i(\cdot)} \sum_{i=1}^{I} \alpha_i(s_i) V_i(s_i)
\]

s.t. \( V_i(s_i) \geq r_i(s_i, a_i) - \lambda^T D_i(s_i, a_i) \\
+ \beta \sum_{s'_i \in S_i} p_i(s'_i | s_i, a_i)V_i(s'_i), \quad (s_i, a_i) \in \mathbb{R}_i, \quad i \in \{1, \ldots, I\}.
\]

In contrast to (4), the linear program (L\lambda) has variables numbering \( \sum_{i=1}^{I} |S_i| \) and constraints numbering \( \sum_{i=1}^{I} \sum_{s_i \in S_i} |A_i(s_i)| \). Thus, the size of the linear program grows linearly with the number \( I \) of subproblems.

We can combine the outer minimization over \( \lambda \) with this linear program:

\[
(PL): \quad H^\lambda(\alpha) = \min_{V_i(\cdot), \lambda} \frac{\lambda^T b}{1 - \beta} + \sum_{i=1}^{I} \alpha_i(s_i) V_i(s_i)
\]

s.t. \( V_i(s_i) \geq r_i(s_i, a_i) - \lambda^T D_i(s_i, a_i) \\
+ \beta \sum_{s'_i \in S_i} p_i(s'_i | s_i, a_i)V_i(s'_i), \quad (s_i, a_i) \in \mathbb{R}_i, \quad i \in \{1, \ldots, I\}, \lambda \geq 0.
\]
The formulation of (PL) has computational implications (Hawkins 2003). That $H^\lambda(\alpha)$ is a (convex) piecewise-linear function of $\lambda$ follows from LP sensitivity analysis. (Add the constraint $\lambda = \lambda$ and consider the objective as a function of $\lambda$.) One can then solve the minimization (8) using stochastic subgradient methods, or else simply by using on LP solver to solve (PL) directly.

The dual of (PL) is particularly insightful:

\[
(DL): \quad H^\lambda(\alpha) = \max_{x} \sum_{i=1}^{I} \sum_{(s_i, a_i) \in X_i} r_i(s_i, a_i)x_{s_i a_i},
\]

s.t.

\[
\sum_{i=1}^{I} \sum_{(s_i, a_i) \in X_i} d_i(s_i, a_i)x_{s_i a_i} \leq \sum_{i=1}^{I} \sum_{(s_i, a_i) \in X_i} b_i - \beta \sum_{(s_i, a_i) \in X_i} p_i(s_i | s_i', a_i)x_{s_i' a_i} = \alpha(s_i), \quad s_i \in S_i, \quad i \in \{1, \ldots, I\},
\]

\[x_{s_i a_i} \geq 0, \quad s_i \in S_i, \quad a_i \in A_i(s_i), \quad i \in \{1, \ldots, I\}\.
\]

The dual variables can be interpreted as discounted total expected time spent in each state-action pair. The first set of constraints in (DL), corresponding with multipliers $\lambda$ in (PL), can be interpreted as an expectation version of the linking constraint. Thus, the Lagrangian relaxation essentially replaces constraints holding almost surely with constraints holding in expectation. The relationship between the Lagrangian relaxation and resource constraints holding in expectation has been recognized previously in the restless bandit case by Whittle (1988) and in sensor management problems by Yost and Washburn (2000) and Castaño (1997, 2005).

2.4. LP-Based ADP Relaxation

Here we force the value function to be additively separable through a substitution of the desired functional form in the LP formulation of the Bellman equations, problem (4).

Substituting the approximation $V(s) \approx \theta + \sum_{i=1}^{I} V_i(s_i)$ gives

\[
(LP\theta): \quad H^{LP\theta}(\alpha) = \theta + \min_{V(\cdot)} \sum_{i=1}^{I} \sum_{s_i \in S_i} \alpha_i(s_i)V_i(s_i),
\]

s.t.

\[
(1 - \beta) + \sum_{i=1}^{I} \sum_{s_i \in S_i} r_i(s_i, a_i) \geq \sum_{i=1}^{I} \sum_{s_i \in S_i} p_i(s_i | s_i', a_i)V_i(s_i'), \quad (s, a) \in \mathcal{X}.
\]

(LP\theta) has variables numbering $\sum_{i=1}^{I} |S_i|$, like (L\lambda), and constraints numbering as many as $\prod_{i=1}^{I} (\sum_{s_i \in S_i} |A_i(s_i)|)$, like (4). Thus, the problem (LP\theta) will in general have variables numbering linearly in $I$, but constraints numbering exponentially in $I$.

We will often denote by (LP\theta) the problem (LP$\theta$) with $\theta = 0$. Also, we will typically use the notation $H^{LP}(\alpha) = H^{LP\theta}(\alpha)$ because $H^{LP\theta}(\alpha)$ is constant with respect to $\theta$ by the following proposition.

PROPOSITION 3. $H^{LP\theta}(\alpha) = H^{LP\theta}(\alpha)$ for any choice of $\theta, \theta'$.

PROOF. Fix $\theta$ and $\theta'$ and take any optimal solution $V^{LP\theta}(\cdot)$ to (LP$\theta$). (We use the notation $V^{LP\theta}(\cdot)$ to denote a set of values of the form $\{V_i^{LP\theta}(s_i), s_i \in S_i, i \in \{1, \ldots, I\}\}$.) Then, $V^{LP\theta}(\cdot)$, where $V_i^{LP\theta}(s_i) = (\theta - \theta')/I + V_i^{LP\theta}(s_i)$ for all $s_i \in S_i, i \in \{1, \ldots, I\}$, is easily demonstrated to be feasible in (LP$\theta'$) with the objective value equal to $H^{LP\theta}(\alpha)$. If another feasible solution to (LP$\theta'$), $V^{LP\theta'}(\cdot)$, achieves a strictly smaller objective, then the solution $V^{LP\theta'}(\cdot)$ to (LP$\theta'$), where $V_i^{LP\theta'}(s_i) = (\theta - \theta')/I + V_i^{LP\theta}(s_i)$ for all $s_i \in S_i, i \in \{1, \ldots, I\}$, contradicts the optimality of $V^{LP\theta}(\cdot)$. Thus, $V^{LP\theta}(\cdot)$ is optimal in (LP$\theta'$), achieving objective $H^{LP\theta}(\alpha)$. □

Under the assumptions of $S$ finite and $A(s)$ finite for all $s \in S$, strong duality holds and we have the dual program

\[
(DLP): \quad H^\lambda(\alpha) = \max_{x} \sum_{(s, a) \in \mathcal{X}} r(s, a)x_{s, a},
\]

s.t.

\[
\sum_{(s', a') \in \mathcal{X}(s, a)} x_{s', a'} - \beta \sum_{(s', a') \in \mathcal{X}(s, a)} p_i(s_i | s_i', a_i)x_{s_i', a_i} = \alpha_i(s_i), \quad s_i \in S_i, \quad i \in \{1, \ldots, I\},
\]

\[x_{s, a} \geq 0, \quad (s, a) \in \mathcal{X}.
\]

We have $H^{LP}(\alpha) \geq H(\alpha)$, but we can also prove a stronger, statewise bound on the optimal value function. Fix $\alpha$ and let $\theta$ and $V^{LP\theta}(\cdot) = \{V_i^{LP\theta}(s_i), s_i \in S_i, i \in \{1, \ldots, I\}\}$ be a feasible solution to (LP$\theta$). Define

\[
J^{LP\theta}(s; \alpha) = \theta + \sum_{i=1}^{I} V_i^{LP\theta}(s_i),
\]

where the $\alpha$ indicates that the values of the decision variables are computed for initial state distribution $\alpha$.

PROPOSITION 4. For fixed $\alpha \geq 0$ and any $\theta$, $J^{LP\theta}(s; \alpha) \geq J(s)$ for all $s \in S$.

PROOF. The feasibility of $V^{LP\theta}(\cdot)$ in problem (LP$\theta$) implies that for any $s \in S$,

\[
(1 - \beta) + \sum_{i=1}^{I} V_i^{LP\theta}(s_i) \geq r(s, a)
\]

\[+ \beta \sum_{s' \in S} p(s' | s, a) \sum_{i=1}^{I} V_i^{LP\theta}(s_i), \quad a \in \tilde{A}(s).
\]

This is equivalent to the inequality

\[
J^{LP\theta} \geq \mathcal{L} J^{LP\theta},
\]

where $\mathcal{L}$ is the Bellman operator corresponding to the exact problem of Equation (3). The desired statewise result $J^{LP\theta}(s; \alpha) \geq J(s)$ for all $s \in S$ then follows from the monotonicity of the Bellman operator. □
2.5. Relation to Static Discrete Optimization

Integer programming Lagrangian relaxation is a special case of the relaxation we present in §2.3. That is, for \(|S| = 1\) (and suppressing the state notation), definition (5) reduces to

\[
(1 - \beta) J^L = \max \sum_{i=1}^{I} r_i(a_i) + \lambda^T (b - \sum_{i=1}^{I} D_i(a_i))
\]

s.t. \(a_i \in \mathcal{A}_i, \ i \in \{1, \ldots, I\}\),

which is the Lagrangian relaxation of a generic static discrete optimization problem with a set of additively separable linking constraints. The addition of state dynamics complicates the relaxation, and thus we require an additional fundamental assumption that \(\lambda\) is state independent (i.e., constant across states).

The LP-based relaxation we consider here is unrelated to the LP relaxation of an integer program. The LP-based relaxation for the case \(|S| = 1\) becomes

\[
\min \sum_{i=1}^{I} V_i
\]

s.t. \(\sum_{i=1}^{I} V_i \geq \frac{r(a)}{1 - \beta}, \ a \in \bar{A}, \)

where \(\bar{A}\) is the feasible region of the corresponding static problem. We can replace \(\sum_{i=1}^{I} V_i\) by a single decision variable \(V\) and see that problem (10) is equivalent to solving the integer program \(\max_{a \in \bar{A}} r(a)\).

3. Comparing the Relaxations

Relaxations (LP\(\theta\)) and (LP\(\theta\)) provide two alternatives for fitting an approximation of the form (2). In this section, we investigate the relative quality of the two approximators, ordering the two bounds and providing an insightful example in which the bounds and associated policies are arbitrarily different.

3.1. Comparing the Bounds

We begin by defining two families of polyhedra,

\[
\mathcal{P}_{LP}(\lambda) = \left\{ V(\cdot): \sum_{i=1}^{I} V_i(s) \geq \sum_{i=1}^{I} r_i(s, a_i) - \lambda^T b \right. \\
\left. + \beta \sum_{i=1}^{I} p_i(s' | s, a_i) V_i(s'), (s, a) \in \bar{\mathcal{R}} \right\},
\]

\[
\mathcal{P}(\lambda) = \left\{ V(\cdot): V(s) \geq r_i(s, a_i) - \lambda^T D_i(s, a_i) \right. \\
\left. + \beta \sum_{i \in S_i} p_i(s' | s, a_i) V_i(s'), (s, a) \in \bar{\mathcal{R}}, i \in \{1, \ldots, I\} \right\},
\]

where the notation \(V(\cdot)\) represents a set of values of the form \(\{V_i(s_i), s_i \in S_i, i \in \{1, \ldots, I\}\}\). Polyhedron \(\mathcal{P}_{LP}(\lambda)\) is the feasible space of problem (LP\(\theta\)) for \(\theta = (\lambda^T b)/(1 - \beta)\), and polyhedron \(\mathcal{P}(\lambda)\) is the feasible region of problem (LP\(\theta\)).

The following theorem provides the basis for our comparison of the two relaxations.

Theorem 1. For any \(\lambda \geq 0\), \(\mathcal{P}(\lambda) \subseteq \mathcal{P}_{LP}(\lambda)\).

Proof. Take some \(V(\cdot)\) feasible in \(\mathcal{P}(\lambda)\). Clearly, \(V(\cdot)\) will be feasible in an aggregated set of constraints obtained by summing the constraints of \(\mathcal{P}_{LP}(\lambda)\) over \(i\),

\[
\sum_{i=1}^{I} V_i(s) \geq \sum_{i=1}^{I} r_i(s, a_i) - \lambda^T \sum_{i=1}^{I} D_i(s, a_i) \\
+ \beta \sum_{i=1}^{I} \sum_{s' \in S_i} p_i(s' | s, a_i) V_i(s'), (s, a) \in \bar{\mathcal{R}}.
\]

Because \(\mathcal{R} \subseteq \mathcal{R}\), \(V(\cdot)\) will be feasible in

\[
\sum_{i=1}^{I} V_i(s) \geq \sum_{i=1}^{I} r_i(s, a_i) - \lambda^T \sum_{i=1}^{I} D_i(s, a_i) \\
+ \beta \sum_{i=1}^{I} \sum_{s' \in S_i} p_i(s' | s, a_i) V_i(s'), (s, a) \in \bar{\mathcal{R}}.
\]

Finally, for all \((s, a) \in \bar{\mathcal{R}}\), we have \(\sum_{i=1}^{I} D_i(s, a_i) \leq b\). This fact, together with \(\lambda \geq 0\), ensures that \(V(\cdot)\) satisfies

\[
\sum_{i=1}^{I} V_i(s) \geq \sum_{i=1}^{I} r_i(s, a_i) - \lambda^T b \\
+ \beta \sum_{i=1}^{I} \sum_{s' \in S_i} p_i(s' | s, a_i) V_i(s'), (s, a) \in \bar{\mathcal{R}},
\]

which is equivalent to its membership in \(\mathcal{P}_{LP}(\lambda)\). Thus, any values in \(\mathcal{P}(\lambda)\) are also in \(\mathcal{P}_{LP}(\lambda)\). \(\square\)

As the following corollary indicates, Theorem 1 implies that the LP-based approximation gives tighter bounds on the true optimal value function than does the Lagrangian relaxation method.

Corollary 1. \(H_{LP}(\alpha) \leq H^\mathcal{X}(\alpha)\) for all \(\lambda \geq 0\) and \(\alpha \geq 0\).

Proof. Fix \(\lambda \geq 0\) and let \(V^*(\cdot)\) be an optimal solution to problem (LP\(\theta\)). The objective of problem (LP\(\theta\)) is

\[
H^\mathcal{X}(\alpha) = \frac{\lambda^T b}{1 - \beta} + \sum_{i=1}^{I} \alpha_i(s_i) V^*_i(s_i).
\]

By Theorem 1, this solution is feasible in problem (LP\(\theta\)) for \(\theta = (\lambda^T b)/(1 - \beta)\), giving objective value equal to \(H_{LP}(\alpha)\). Hence, \(H_{LP}(\alpha) \leq H^\mathcal{X}(\alpha)\) because the objective in (LP\(\theta\)) is minimized. By Proposition 3, \(H_{LP}(\alpha) = H_{LP}(\theta)\), so we have \(H_{LP}(\alpha) \leq H^\mathcal{X}(\alpha)\). \(\square\)

An important implication of Corollary 1 is the bound \(H_{LP}(\alpha) \leq H^\mathcal{X}(\alpha)\).
3.2. Comparing the Policies

Given the value function estimates generated by the two methods, natural control policies are one-step greedy policies based on these estimates, in which the policy is obtained by plugging the value function into the right-hand side of Bellman’s equation. Corollary 1 orders the bounds produced by the two relaxations, but it does not imply a similar ordering of the corresponding policies. To our knowledge, bounds on the performance difference between one-step greedy policies are nonexistent in approximate dynamic programming.

Proposition 6.1 of Bertsekas and Tsitsiklis (1996) states that a greedy policy \( \mu \) based on a value function approximation \( \tilde{J} \) generates a value function \( \tilde{J}^\mu \) satisfying

\[
\| \tilde{J}^\mu - J \|_\infty \leq \frac{2\beta \| \tilde{J} - J \|_\infty}{1 - \beta}.
\]

Thus, if we can prove that the LP-based relaxation leads to tighter statewise value function bounds than the Lagrangian relaxation, Equation (11) gives a tighter guarantee on the performance of the resulting policy. Corollary 1 gives us such statewise value function bounds if we compute \( H^\lambda(\alpha) \) and \( H^{LP}(\alpha) \) for

\[
\alpha(s') = \begin{cases} 1 & \text{for } s' = s, \\ 0 & \text{for } s' \neq s, \end{cases}
\]

for each \( s \in S \). However, as the number of states becomes large, it becomes unrealistic to recompute the relaxations for each state. It is more realistic to solve each approximation once for a single \( \alpha \), and then use the resulting value function estimates for each state to generate policies. In spite of Propositions 2 and 4, we do not in general have \( J^{\lambda^*}(s; \alpha) \geq J^{LP}(s; \alpha) \) for arbitrary \( \alpha \). (We use the notation \( J^{\lambda^*}(s; \alpha) \) to emphasize that the optimal \( \lambda^* \) in the Lagrangian problem is a function of the initial state distribution \( \alpha \).) We include a counterexample in online Appendix B. We will thus rely on numerical experiments to evaluate the policies produced by the two relaxations.

3.3. Example in Which the Lagrangian Bound and Policy Can Be Arbitrarily Bad

A remaining question is whether the difference between the two bounds and associated policies is ever significant. In this section, we provide an affirmative answer by giving an example problem instance in which both the Lagrangian bound and policy perform arbitrarily poorly, whereas the LP-based relaxation gives the optimal solution.

Consider a problem with \( I = 1 \). The single subproblem has three possible states \( \{0, 1, 2\} \), and for each state \( s \), \( A(s) = \{0, 1\} \). Table 1 includes the rewards, constraint weights, and nonzero transition probabilities, and Figure 1 illustrates the possible transitions. The initial state is zero with probability one, and then the system transitions to state 1 or 2 depending on the action chosen. Both states 1 and 2 are absorbing states regardless of the action chosen. At each point in time, the controls are constrained by \( D(s, a) \leq 1 \). We assume that \( \beta > 1/2 \).

Control 1 in state 1 is infeasible, so the only attainable nonzero rewards are those arising from applying control 1 in state 2. Thus, the optimal policy is to use control 0 in state 0 and control 1 in state 2, yielding an optimal value of

\[
H(\alpha) = \frac{c\beta}{1 - \beta}.
\]

Because there is just a single subproblem, the LP-based relaxation yields the optimal solution. That is, \( H^{LP}(\alpha) = H(\alpha) = c\beta/(1 - \beta) \), and the implied one-step greedy policy is the optimal one.

The Lagrangian relaxation yields

\[
H^\lambda(\alpha) = J^{\lambda^*}(0) = \min_{\lambda \geq 0} \left\{ \frac{\lambda}{1 - \beta} + V^\lambda(0) \right\},
\]

where

\[
V^\lambda(0) = \max\{0 + \beta V^\lambda(2), 0 + \beta V^\lambda(1)\},
\]

\[
V^\lambda(1) = \max\{0 + \beta V^\lambda(1), c(2 + I) - 2\lambda + \beta V^\lambda(1)\},
\]

\[
V^\lambda(2) = \max\{-\epsilon\lambda + \beta V^\lambda(2), c - \lambda + \beta V^\lambda(2)\}.
\]

Table 1. A one-subproblem example with three states and two allowable actions per state.

<table>
<thead>
<tr>
<th>State</th>
<th>Control</th>
<th>Reward</th>
<th>Weight</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>( r(0, 0) = 0 )</td>
<td>( D(0, 0) = 0 )</td>
<td>( P(2 \mid 0, 0) = 1 )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>( r(0, 1) = 0 )</td>
<td>( D(0, 1) = 0 )</td>
<td>( P(1 \mid 0, 1) = 1 )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>( r(1, 0) = 0 )</td>
<td>( D(1, 0) = 0 )</td>
<td>( P(1 \mid 1, 0) = 1 )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>( r(1, 1) = c(2 + l) )</td>
<td>( D(1, 1) = 2 )</td>
<td>( P(1 \mid 1, 1) = 1 )</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>( r(2, 0) = 0 )</td>
<td>( D(1, 0) = \epsilon )</td>
<td>( P(2 \mid 2, 0) = 1 )</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>( r(2, 1) = c )</td>
<td>( D(1, 1) = 1 )</td>
<td>( P(2 \mid 2, 1) = 1 )</td>
</tr>
</tbody>
</table>

Note. \( I > 0 \) and \( c > 1 \) are positive constants, and \( 0 < \epsilon < l/(2 + l) \) is a small positive scalar. Transitions other than those indicated occur with probability zero.

Figure 1. Allowable state transitions in the example problem.
These equations imply
\[ V^1(1) = \max \left\{ 0, \frac{c(2 + l) - 2\lambda}{1 - \beta} \right\} \]
\[ = \begin{cases} \frac{c(2 + l) - 2\lambda}{1 - \beta} & \text{for } \lambda < c + cl/2, \\ 0 & \text{for } \lambda \geq c + cl/2, \end{cases} \]
\[ V^1(2) = \max \left\{ -\frac{\epsilon \lambda}{1 - \beta}, \frac{c - \lambda}{1 - \beta} \right\} = \begin{cases} \frac{c - \lambda}{1 - \beta} & \text{for } \lambda < \frac{c}{1 - \epsilon}, \\ -\frac{\epsilon \lambda}{1 - \beta} & \text{for } \lambda \geq \frac{c}{1 - \epsilon}. \end{cases} \]

The condition \( \epsilon < l/(2 + l) \) implies that \( c/(1 - \epsilon) < c + cl/2 \). Thus, to compute \( V^1(0) \), it suffices to consider three cases.

Case 1. \( \lambda < c/(1 - \epsilon) \):
\[ V^1(1) = \frac{c(2 + l) - 2\lambda}{1 - \beta}, \quad V^1(2) = \frac{c - \lambda}{1 - \beta}. \]
\[ V^1(0) = \frac{\beta}{1 - \beta} [c(2 + l) - 2\lambda] \]
because \( \lambda < c/(1 - \epsilon) \leq c + cl/2 < c(1 + l) \Rightarrow c(2 + l) - 2\lambda > c - \lambda \).

Case 2. \( c/(1 - \epsilon) \leq \lambda \leq c + cl/2 \):
\[ V^1(1) = \frac{c(2 + l) - 2\lambda}{1 - \beta}, \quad V^1(2) = \frac{-\epsilon \lambda}{1 - \beta}, \]
\[ V^1(0) = \frac{\beta}{1 - \beta} [c(2 + l) - 2\lambda] \]
because \( \lambda \leq c + cl/2 = (2c + cl)/2 < (2c + cl)/(2 - \epsilon) \Rightarrow c(2 + l) - 2\lambda > -\epsilon \lambda \).

Case 3. \( \lambda > c + cl/2 \):
\[ V^1(1) = 0, \quad V^1(2) = \frac{-\epsilon \lambda}{1 - \beta}, \quad V^1(0) = 0. \]

Then, the Lagrangian relaxation can be written as
\[ H^1(\alpha) = \min_{\lambda > 0} \frac{1}{1 - \beta} \left\{ \lambda + \beta [c(2 + l) - 2\lambda] \right\} \]
\[ = h^1(\alpha) = (c + cl/2)/(1 - \beta). \]

The term in braces has negative slope for \( \lambda < c + cl/2 \) and positive slope for \( \lambda > c + cl/2 \); thus, \( H^1(\alpha) = (c + cl/2)/(1 - \beta) \) minimizes. This yields a Lagrangian bound of \( H^1(\alpha) = (c + cl/2)/(1 - \beta) \). Furthermore, the value function corresponding to state 1 is larger than that corresponding to state 2; thus, a one-step greedy policy corresponding to this solution is to apply control 1 in state 0. Such a policy yields value zero regardless of c and l.

As \( l \to \infty \), \( H^1(\alpha) = (c + cl/2)/(1 - \beta) \) grows without bound, whereas \( H(\alpha) = H^{L1}(\alpha) = c\beta/(1 - \beta) \) remains constant. As \( c \to \infty \), the value \( c\beta/(1 - \beta) \) of the optimal and LP-based policies grow without bound, whereas the value of the Lagrangian policy remains constant at zero. Thus, in this example both the Lagrangian bound and policy can be made arbitrarily poor. We will explore why in the next section.

4. Analysis of Relaxation Gaps

Having ordered the two bounds and shown that the difference between them can be significant, it remains to explore when we might expect them to perform well relative to each other and relative to the optimal value \( H(\alpha) \). In this section, we explore when the two bounds are tight. We also look at performance of the relaxations as the number of subproblems grows. We provide various qualitative results about the relaxation gaps, and in some cases quantitative bounds. We will see that the state independence of the multipliers \( \lambda \) and the manner in which the constraints are formulated play key roles in the relative performance of the relaxations.

4.1. Gap \( H^1(\alpha) - H(\alpha) \)

In this section, we present necessary and sufficient conditions for the Lagrangian relaxation to provide a tight estimate of \( H(\alpha) \). The theory is useful for understanding the behavior of the Lagrangian relaxation, and by Corollary 1 also provides sufficient conditions for the LP-based relaxation to be tight.

We will require the following additional notation. Let \( d : S \to \bigcup_{s \in S} \tilde{A}(s) \), where \( d(s) \in \tilde{A}(s) \) for all \( s \in S \) indicate a feasible deterministic control policy, and let \( d(s) \) indicate the ith component of \( d(s) \). We will denote by \( J^d(\alpha) \) value of state \( s \) under \( d \) in the Lagrangian problem. That is, \( J^d(\cdot) \) solves

\[ J^d(\alpha) = r(s, d(s)) + \lambda^T \left[ -b - \sum_{i=1}^l D_i(s_i, d(s_i)) \right] + \beta E[J^d(\alpha)] \]

where the expectation is with respect to the conditional distribution of the ensuing state under policy \( d \). Let \( H^d(\alpha) = \sum_{s \in S} \alpha(s) J^d(\alpha) \). For fixed \( d \), we can also write \( J^d(\alpha) = J^d(s) + W^d(\alpha) \), where

\[ J^d(s) = r(s, d(s)) + \beta E[J^d(s^\prime) | s, d(s)] \]

is the discounted sum of expected rewards under \( d \), and

\[ W^d(s) = \lambda^T \left[ -b - \sum_{i=1}^l D_i(s_i, d(s_i)) \right] + \beta E[W^d(s^\prime) | s, d(s)] \]

is the discounted total expected accumulation of penalty terms under \( d \).

The following result gives equivalent necessary and sufficient conditions for the Lagrangian relaxation to give a tight bound on the optimal objective value.

Theorem 1. The following three statements are equivalent, assuming \( \alpha > 0 \):
(i) \( H^1(\alpha) = H(\alpha) \),
(ii) there exist \( \lambda > 0 \) and \( d(s) \in \tilde{A}(s) \) such that \( \lambda^T [b - D(s, d(s))] = 0 \) for all \( s \in S \) and \( H^d(\alpha) = H^1(\alpha) \).
Lemma 1. Following result, which says that the first condition of the
Proof.

\[ H(\alpha) = H^\lambda(\alpha) \text{ for } \alpha > 0 \text{ if and only if } J(s) = J^\lambda(s) \text{ for all } s \in S. \]

We include a proof in online Appendix C. It uses the following result, which says that the first condition of the

\[ \text{Lemma 1. } H(\alpha) = H^\lambda(\alpha) \text{ for } \alpha > 0 \text{ if and only if } J(s) = J^\lambda(s) \text{ for all } s \in S. \]

\[ \text{Proof. } \text{Necessity follows directly from the definitions of } H^\lambda(\alpha) \text{ and } H(\alpha). \text{ For sufficiency, suppose that } H^\lambda(\alpha) = H(\alpha) \text{ for } \alpha > 0, \text{ but } J^\lambda(s) \neq J(s) \text{ for some } s \in S. \text{ Then, there exists at least one } s \in S \text{ such that } J(s) < J^\lambda(s), \text{ and at least one } s \in S \text{ such that } J(s) > J^\lambda(s). \text{ The existence of the latter contradicts Proposition 2.} \]

Theorem 2 is analogous to results proven for static optimization problems by Everett (1963). A discussion of Everett’s results can be found in Nemhauser and Wolsey (1988, Section III.2.6). Condition (iii) is essentially a saddle point optimality condition (see Bertsekas 1982, Proposition 5.2) for dynamic programming.

For \(|S| = 1\), conditions (ii) and (iii) are trivially equivalent because there are no state dynamics. When we include state dynamics, conditions (ii) and (iii) are equivalent by Theorem 2, but they bring different insights. Condition (ii) is a statement about the Lagrangian problem, and describes a certificate of Lagrangian bound tightness based on the solution to the Lagrangian problem. Condition (iii), on the other hand, describes a certificate using the true optimal value function. We can interpret this condition as saying that the constraints can be perfectly dualized in a reduced profit maximization corresponding to the dual of problem (4).

Theorem 2, together with the example of §3.3, characterizes what can go wrong for the Lagrangian relaxation. In the example problem, the only feasible action in state 1 is control 0, formulated by the constraint \(2a_i \leq 1\). This is not a tight formulation of the feasible space, and leads to a positive penalty term of \((1 - 2a_i)\lambda\) that violates the conditions of Theorem 2 when \(a_i = 0\) and \(\lambda > 0\). Of course, if we apply the Lagrangian relaxation to an equivalent formulation of the example, in which \(A(1)\) excludes control 1 and all remaining \(D(s, \cdot)\)s are set to one, then the penalty terms vanish. Theorem 2 is satisfied, and the Lagrangian gives the optimal solution. Problems with similar structures to our example are known to plague Lagrangian approaches to the static knapsack problem (Martello and Toth 1990), which is a special case of a weakly coupled dynamic program (see §2.5).

Another reason why the Lagrangian relaxation might give a loose bound is peculiar to the dynamic case, where multiple states are visited. Optimal policies may lie in the interior of the space \(D(s, \cdot)\) for some states (requiring \(\lambda > 0\) to make the corresponding penalty term 0), while on the boundary for other states (for which we may require \(\lambda > 0\) to ensure feasibility), thus making Theorem 2 condition (iii) impossible to satisfy with a single state-independent vector of multipliers. Such problems are those for which we might expect the value functions to be poorly approximated by a separable form. We give an example and further discussion in §4.3.

4.2. Gap \(H^\lambda(\alpha) - H^{LP}(\alpha)\)

Corollary 1 orders the bounds produced by the two relaxations, but tells us nothing about when we should expect the two bounds to be equivalent or different. It is simple to show an example for which \(H^\lambda(\alpha) = H^{LP}(\alpha)\). For example, both methods yield the optimal value function in cases in which the linking constraints are irrelevant (i.e., never violated for any solution feasible in the individual subproblems).

The following gives a more general sufficient condition for \(H^{LP}(\alpha)\) to equal \(H^\lambda(\alpha)\) in the spirit of Theorem 2. (We do not have a direct converse of Theorem 3 because we do not have a result analogous to Lemma 1 that holds here.) Let \(V_i^{LP}(s) = \sum_{i=1}^{J} V_{i}(s)\) for all \(s \in S\), where \(V_{LP}(\cdot)\) is an optimal solution to (LP).

\[ \text{Theorem 3. Suppose that there exists } \lambda_{LP} \geq 0 \text{ and a policy } d(s) \in A(s) \text{ for all } s \in S \text{ such that } \lambda_{LP}^{T} = H^{\lambda_{LP}}(s) \text{ for all } s \in S \text{ and} \]

\[ d(s) = \arg\max_{a \in A(s)} r(s, a) + \lambda_{LP}^{T} [b - D(s, d(s))] + \beta E[J(s') | s, a]. \]

Then, \(H^\lambda(\alpha) = H^{LP}(\alpha)\).

We include a proof in online Appendix C.

Although in §2.3 we motivated the Lagrangian bound as a relaxation and decomposition of the original problem (3), Theorem 3, together with Theorem 1, motivates a view of the Lagrangian bound as a relaxation of the LP-based approximation. The LP-based approximation decomposes the value function by subproblem, and the Lagrangian bound further relaxes the problem by dualizing the constraints in the reduced profit maximization corresponding to the problem (DLP).

Using a different proof technique, we have been able to show the equivalence of the Lagrangian bound and the LP-based bound for a class of restless bandit problems (Whittle 1988), including the multiarmed bandit problem. We provide details in online Appendix D.

4.3. Bounds on the Relaxation Gaps

It is known (Bertsekas 1982, Proposition 5.26) that, under mild conditions, the Lagrangian duality gap of a weakly coupled static and deterministic math program is bounded
above by a constant that does not depend on the number of subproblems. Hence, as the number of subproblems grows to infinity, the average duality gap per subproblem goes to zero asymptotically. Furthermore, if the optimal value of the primal problem is proportional to the number of subproblems, the ratio of the duality gap to the optimal value goes to zero asymptotically. In this section, we first give an example demonstrating that this asymptotic result does not extend to our Lagrangian relaxation of weakly coupled stochastic dynamic programs. We then prove that a modified asymptotic result does hold when the Lagrange multipliers are allowed to be state dependent, in which case we denote the optimal Lagrangian relaxation objective value by \( H^{\lambda^*}(\alpha) \) to stress that \( \lambda^*(\cdot) \) is a function of state \( s \). Such a Lagrangian relaxation is not numerically practical because the problem does not decompose when the multipliers are dependent state. However, the result gives us insight into when good asymptotic behavior is likely to occur, namely, when a state-independent Lagrange multiplier is a good approximation to optimal state-dependent multipliers.

4.3.1. Example in Which the Lagrangian Gap Grows Linearly with the Number of Subproblems. Assume a collection of \( I \) identical subproblems, each of the following form. Each subproblem \( i \) has two states, i.e., \( S_i = \{1, 2\} \). In each state \( s_i \) there are two actions, i.e., \( A_i(s_i) = \{0, 1\} \). If \( s_i = 1 \), then the reward for taking action \( a_i \in A_i(1) \) is \( r_i(1, a_i) = a_i \), and the rewards in state \( s_i = 2 \) are zero. All state-action pairs transition into state 2, with probability one. Choose the initial state distribution \( \alpha \) so that \( s_i = 1 \), with probability one.

The subproblems are linked by a single constraint of the form \( \sum_{i=1}^I D_i(s_i, a_i) \leq b \), where \( b = 1 \); and for subproblem \( i \) the constraint coefficient is

\[
D_i(s_i, a_i) = \begin{cases} 2a_i & \text{if } s_i = 1, \\ 1 - a_i & \text{if } s_i = 2, \end{cases}
\]

for all \((s_i, a_i) \in \mathcal{X}_i\). Because of this constraint, the only feasible action in the initial state is to set \( a_i = 0 \). After that, the system absorbs into state 2 and never earns rewards again. Therefore, the optimal policy value is \( H(\alpha) = 0 \).

On the other hand, after dualizing the constraint \( \sum_{i=1}^I D_i(s_i, a_i) \leq b \), the optimal Lagrangian value is

\[
H^{\lambda^*}(\alpha) = \min_{\lambda > 0} \max_{\{a_i \in [0, 1]\}} \lambda \left( \sum_{i=1}^I a_i + 1 - \sum_{i=1}^I 2a_i \right) + (I+1)\beta \frac{1}{1-\beta}
\]

Assume that \( 3\beta - 2 \geq 0 \). Because all \( \lambda \) coefficients are nonnegative, \( \lambda^* = 0 \) minimizes. Therefore, \( H^{\lambda^*}(\alpha) = \max\{I, 0\} = I \). Consequently, \( H^{\lambda^*}(\alpha) - H(\alpha) = I \), and so the gap grows linearly with \( I \).

4.3.2. Gaps \( H^{\lambda^*}(\alpha) - H(\alpha) \) and \( H^{LP}(\alpha) - H(\alpha) \). The trouble in the above example, as we see next, is that the multiplier \( \lambda \) does not depend on the system state. We first explain Bertsekas’s (1982) duality gap result for a general myopic \( (\beta = 0) \) problem solved at a fixed state \( s \). The primal problem, written in vector form, is

\[
(P(s)) : \max_{a \in A(s)} r(s, a)
\]

s.t. \( D(s, a) \leq b \).

The Lagrangian dual is

\[
(D(s)) : \min_{\lambda > 0} \max_{a \in A(s)} r(s, a) + \lambda^* [b - D(s, a)].
\]

Now let \( \mathcal{E} \) be a scalar measuring the lack of convexity in elements of \( \{r_i(s_i, \cdot)\}_i \), viewed as functions of \( a_i \). We define it formally in online Appendix E, but for a fixed state \( s \) it is essentially a generalization to a nonconvex domain of the metric (Aubin and Ekeland 1976)

\[
\max_{i \in [1, \ldots, I]} \sup_{\gamma_i} \left\{ r_i(s, \sum_j \gamma_j a''_j) - \sum_j \gamma_j r_i(s, a''_j) \right\},
\]

over all finite families \( \gamma_i \in (0, 1), a''_i \in A_i(s_i) \), with \( \sum_j \gamma_j = 1 \), assuming that the domain of \( r_i(s_i, \cdot) \) is convex. The result of Bertsekas (1982), restated as Theorem 8 in the online appendix, says that

\[
\min D(s) - \max P(s) \leq (N+1)\mathcal{E}(s).
\]

We henceforth assume that all \( r_i \) functions are taken from a fixed bounded set, and therefore \( \mathcal{E}(s) \) is uniformly bounded over all \( s \) and \( I \). Hence, the duality gap is bounded above by a constant independent of \( I \).

Now consider a Lagrangian relaxation of the Bellman equations, similar to (5) except that the Lagrange multiplier vector now can depend on state \( s \), denoted by \( \lambda(s) \):

\[
J^{\lambda^*}(s) = \max_{a \in A(s)} \left\{ r(s, a) + \lambda(s)^T [b - D(s, a)] + \beta \mathcal{E}[J^{\lambda^*}(s') | s, a] \right\}, \quad s \in S. \quad (13)
\]

Given an initial state distribution \( \alpha \), the Lagrangian problem becomes

\[
H^{\lambda^*}(\alpha) = \min_{\lambda(s) > 0} \sum_{s \in S} \alpha(s) J^{\lambda^*}(s).
\]

We can interpret problem (13) as dualizing constraints that hold almost surely at each state. We distinguish this
from (5) in §2.3, where we interpreted a state-independent $\lambda$ as dualizing constraints that hold in expectation.

Define $\bar{S}(\alpha) \subseteq S$ to be the set of states attainable by (Markov randomized) policies under the initial state distribution $\alpha$, ignoring the constraints $D(s, a) \leq b$. $\bar{S}(\alpha)$ is generally difficult to obtain in practice, but here it is of theoretical interest. We then have the following theorem for the duality gap.

**Theorem 4.**

$$H^{\star}(\alpha) - H(\alpha) \leq \frac{(N+1)\varepsilon^* + \Gamma}{1 - \beta},$$

(14)

where $\varepsilon^* = \max_{s \in \bar{S}(\alpha)} \varepsilon(s)$ and

$$\Gamma = \max_{\alpha} \max_{r(s, a)} \min_{a \in \Delta(s)} r(s, a) - \min_{a \in \Delta(s)} \max_{r(s, a)} r(s, a).$$

**Proof.** Without loss of accuracy, we may restrict the state space to $\bar{S}(\alpha)$ because neither $H(\alpha)$ nor $H^{\star}(\alpha)$ depend on their respective value functions for states $s \in S \setminus \bar{S}(\alpha)$. Let $r^*(s) = \max_{a \in \Delta(s)} r(s, a)$ and $s_{\min} = \arg \min_{s \in \bar{S}(\alpha)} r^*(s)$. Then, from Example 6.6.2 in Puterman (1994), it follows that

$$J(s) \geq \frac{1}{1 - \beta} r^*(s_{\min}), \quad s \in \bar{S}(\alpha).$$

This implies

$$H(\alpha) \geq \frac{1}{1 - \beta} r^*(s_{\min}) = \frac{1}{1 - \beta} \max P(s_{\min}).$$

(15)

Next, for any $\lambda(\cdot) \geq 0$, let

$$r^{\lambda}(s) = \max_{a \in \Delta(s)} r(s, a) + \lambda^T [b - D(s, a)].$$

Also, choose

$$s_{\max} = \arg \min_{s \in \bar{S}(\alpha)} \left\{ \min_{a \in \Delta(s)} r(s, a) + \lambda^T [b - D(s, a)] \right\}.$$  

Again, from Example 6.6.2 in Puterman (1994), it follows that for any $\lambda(\cdot) \geq 0$,

$$J^{\lambda}(s) \leq \frac{1}{1 - \beta} \max_{s \in \bar{S}(\alpha)} r^{\lambda}(s), \quad s \in \bar{S}(\alpha).$$

Therefore,

$$H^{\star}(\alpha) = \min_{\lambda(\cdot) \geq 0} \sum_{s \in \bar{S}(\alpha)} \alpha(s) J^{\lambda}(s) \leq \min_{\lambda(\cdot) \geq 0} \frac{1}{1 - \beta} \max_{s \in \bar{S}(\alpha)} r^{\lambda}(s)$$

$$= \frac{1}{1 - \beta} \max_{s \in \bar{S}(\alpha)} \left( \min_{a \in \Delta(s)} r(s, a) + \lambda^T [b - D(s, a)] \right)$$

$$= \frac{1}{1 - \beta} \min D(s_{\max}),$$

where the second equality follows because $\lambda(\cdot)$ is a function of state $s$. Consequently,

$$H^{\star}(\alpha) - H(\alpha) \leq \frac{1}{1 - \beta} (\min D(s_{\max}) - \max P(s_{\min}))$$

$$= \frac{1}{1 - \beta} [\min D(s_{\max}) - \max P(s_{\max})$$

$$+ \max P(s_{\min})]$$

$$\leq \frac{1}{1 - \beta} [(N+1)\varepsilon^*(s_{\max}) + \Gamma]$$

$$\leq \frac{(N+1)\varepsilon^* + \Gamma}{1 - \beta}.$$  

□

When $\lambda$ is a constant, we are no longer able to reverse the max and the min in the last part of the proof. In the example of §4.3.1, we have that $\Gamma = 0$. Therefore, the difficulty lies in that $\lambda$ is not state dependent.

We can also derive a similar bound for $H^{LP}(\alpha)$. Let $H^{LP}(\alpha; \tilde{S})$ denote the optimal objective value of (LP)$\theta$ when only the constraints for states $\tilde{S} \subseteq S$ are included. That is, we have a constraint only for $(s, a) \in \tilde{S}$, such that $s \in \tilde{S}$. We recover $H^{LP}(\alpha)$ when $\tilde{S} = S$. When $\tilde{S} \supseteq \bar{S}(\alpha)$, all of our results involving $H^{LP}(\alpha)$, including those in §2.4, can be modified to hold for $H^{LP}(\alpha; \tilde{S})$. Also, we obviously have $H(\alpha) \leq H^{LP}(\alpha; \tilde{S}) \leq H^{LP}(\alpha)$.  

**Theorem 5.** Consider any set of states $\tilde{S}$ such that $\bar{S}(\alpha) \subseteq \tilde{S} \subseteq S$. Then,

$$H^{LP}(\alpha; \tilde{S}) - H(\alpha) \leq \frac{\Gamma(\tilde{S})}{1 - \beta},$$

(16)

where

$$\Gamma(\tilde{S}) = \max_{s \in \tilde{S}} \min_{a \in \Delta(s)} r(s, a) - \min_{a \in \Delta(s)} \max_{s \in \tilde{S}} r(s, a).$$

**Proof.** Consider (LP)$\theta$ restricted to states $s \in \tilde{S}$. Construct a feasible solution by setting $V = 0$ and

$$\theta = \left( \frac{1}{1 - \beta} \right) \max_{s \in \tilde{S}} \min_{a \in \Delta(s)} r(s, a).$$

Because (LP)$\theta$ minimizes, it follows that $H^{LP}(\alpha; \tilde{S}) \leq \theta$. Let $s_{\min} = \arg \min_{s \in \tilde{S}} r^*(s)$. Because $H(\alpha)$ does not depend on $J(s)$ for states $s \in S \setminus \bar{S}(\alpha)$, and $\min_{s \in \tilde{S}} r^*(s) \leq \min_{s \in \bar{S}(\alpha)} r^*(s)$, we can combine the previous bound with (15) to yield (16).  

□

When we take $\tilde{S} = \bar{S}(\alpha)$, we have $\Gamma(\tilde{S}) = \Gamma = 0$ for the example of §4.3.1. Therefore, the bound (16) implies that $H^{LP}(\alpha; \bar{S}(\alpha))$ equals $H(\alpha)$, i.e., the gap is zero. Theorem 5 can be loosened by using $\tilde{S}$ as in Theorem 4. The essential difference between the bounds (14) and (16) is the additional term involving $\varepsilon^*$ due to dualization, which indicates that the quality of the Lagrangian relaxation is dependent on lack of convexity in the problem.
It follows from the uniform boundedness of $\mathcal{E}(s)$ over all $s$ and $I$ that $\mathcal{E}^*$ is bounded above by a constant that does not depend on $I$. However, the constants $\Gamma$ and $\Gamma(\tilde{S})$ may in general depend on the number of subproblems $I$. When they are sublinear in $I$, our theorems imply that

$$H^{\alpha}(\alpha) - H(\alpha) \xrightarrow{I} 0$$

and

$$H^{LP}(\alpha; \tilde{S}) - H(\alpha) \xrightarrow{I} 0,$$

i.e., the average relaxation gaps go to zero as the number of subproblems increases.

5. Solving the LP-Based Relaxation

Corollary 1 tells us that the LP approximation gives superior bounds to the Lagrangian method. However, the number of constraints in the LP approximation grows exponentially with the number $I$ of subproblems, whereas the Lagrangian relaxation can be solved using a linear program that has constraints numbering linearly in $I$ (recall problem (DL) in §2.3). Thus, in general, the Lagrangian relaxation will be easier to compute than the LP-based relaxation. In this section, we discuss computation of the LP-based relaxation.

Because of the large number of constraints in (LP), the LP-based relaxation method is viable for large instances only if there are efficient methods for handling the large number of constraints. The results of de Farias and Van Roy (2004) give us hope that solving a problem with a sampled set of constraints might give strong results, although their result relies on a sampling distribution based on knowledge of the optimal solution. We explore a practical column generation method for solving the problem (DLP). Lübbecke and Desrosiers (2005) provide a recent review of column generation. Adelman (2004) reports on the application of column generation for LP-based approximate dynamic programming in an inventory/routing context.

In some contexts, columns can be generated very efficiently, yielding promising algorithms for solving (LP). Corresponding to a current column set $C \subseteq \mathcal{F}$, the master problem corresponding to a column generation procedure for solving (DLP) is

$$H^{LP}_{C}(\alpha) = \max_{(s, a) \in C} \sum r(s, a)x_{sa}$$

s.t.

$$\sum_{(s', a) \in C: s' = s} x_{sa} - \beta \sum_{(s', a) \in C} p_i(s_i | s', a_i)x_{sa} = \alpha_i(s),$$

$$x_{sa} \geq 0, \quad (s, a) \in C.$$

Let $\{V^C_i(s_i), i \in \{1, \ldots, I\}, s_i \in S_i\}$ be the shadow prices of the constraints in problem (17), and define $f_i(s_i, a_i) = r_i(s_i, a_i) - V^C_i(s_i) + \beta \sum_{s_i' \in S_i} p_i(s_i' | s_i, a_i) V^C_i(s_i')$.

To prove that an optimal solution to (17) also solves (DLP), it is sufficient to find the maximum reduced profit

$$R_C(\alpha) = \max_{(s, a) \in C} \sum_{i=1}^{I} f_i(s_i, a_i)$$

s.t. $\sum_{i=1}^{I} D_i(s_i, a_i) \leq b$ \hspace{1cm} (18)

and verify that it is nonpositive. In the course of the column generation approach, we do not necessarily need to solve (18) to optimality at each iteration. Any $(s, a) \in \mathcal{F}$ with $\sum_{i=1}^{I} D_i(s_i, a_i) \leq b$ and $\sum_{i=1}^{I} f_i(s_i, a_i) > 0$ gives a candidate column to add to $C$.

Problem (18) is a generalized multidimensional knapsack problem. Although it is in general NP-complete, there are many examples for which it is efficiently solvable. For example, if there is a single constraint with integral weights, then (18) can be solved using dynamic programming in time proportional to $bI \max_{s_i \in S_i} [s_i]$, as is commonly done with knapsack problems. For the restless bandit problem, (18) can be solved with a simple greedy algorithm. See online Appendix D.

A potential drawback of the LP-based method is that it does not necessarily give a value function bound unless the linear program (DLP) is solved to optimality. However, the following theorem establishes a bound at each stage of the column generation algorithm, without solving (DLP) to optimality. (Such results are standard in column generation; see Lasdon 1970.)

Theorem 6. For $C \subseteq \mathcal{F}$ and assuming (17) is feasible,

$$H^{LP}_{C}(\alpha) \leq H^{LP}(\alpha) \leq H^{LP}_{C}(\alpha) + \frac{1}{1 - \beta} R_C(\alpha).$$

Proof. The first inequality follows from the definition of $H^{LP}_{C}(\alpha)$. To prove the second inequality, let $(x_{sa}, (s, a) \in \mathcal{F})$ be an optimal solution to (DLP), and let $\{V^C_i(s_i), i = 1, \ldots, I, s_i \in S_i\}$ be the shadow prices of (17) for column set $C$. Adding the constraints of (DLP) over $s_i \in S_i$ for some $i$ yields

$$\sum_{s_i \in S_i} \sum_{(s, a) \in \mathcal{F}} x_{sa} - \beta \sum_{(s, a) \in \mathcal{F}} x_{sa} \sum_{s_i \in S_i} p_i(s_i | s', a_i) = \sum_{s_i \in S_i} \alpha_i(s_i).$$

Hence

$$\sum_{(s, a) \in \mathcal{F}} x_{sa} = \frac{1}{1 - \beta}.$$

Now multiply the (DLP) constraints by the corresponding $V^C_i(s_i)$, sum, and subtract from $H^{LP}(\alpha) = \sum_{(s, a) \in \mathcal{F}} r(s, a)x_{sa}$ as follows:

$$H^{LP}(\alpha) - \sum_{i=1}^{I} \sum_{s_i \in S_i} \alpha_i(s_i)V^C_i(s_i)$$

$$= \sum_{(s, a) \in \mathcal{F}} r(s, a)x_{sa} - \sum_{i=1}^{I} \sum_{s_i \in S_i} V^C_i(s_i).$$

Adelman and Mersereau: Relaxations of Weakly Coupled Stochastic Dynamic Programs
\[
\begin{align*}
\mathcal{H}^L_\alpha &= \sum_{s_i \in S} \sum_{\alpha_i \in \mathcal{A}} x_{s_i\alpha_i} - \beta \sum_{s_i \in S} p_i(s_i, s'_i, \alpha_i) x_{s_i\alpha_i} \\
&= \sum_{(s, \alpha) \in \mathcal{X}} x_{s\alpha} \sum_{i=1}^I f_i(s_i, \alpha_i).
\end{align*}
\]

Hence,
\[
\mathcal{H}^{LP}_\alpha \leq \sum_{i=1}^I \sum_{s_i \in S} \alpha_i(s_i) V^C_i(s_i) + \left( \sum_{(s, \alpha) \in \mathcal{X}} x_{s\alpha} \right) \cdot \left( \max_{(s, \alpha) \in \mathcal{X}} \sum_{i=1}^I f_i(s_i, \alpha_i) \right)
\]
\[
= \mathcal{H}^{LP}_C(\alpha) + \frac{1}{1 - \beta} R_C(\alpha). \quad \square
\]

Figure 2 illustrates the evolution of the lower bound \( \mathcal{H}^{LP}_C(\alpha) \) and upper bound \( \mathcal{H}^{LP}_C(\alpha) + 1/(1 - \beta) R_C(\alpha) \) as columns are generated in the course of the column generation algorithm for a generated problem with 10 subproblems and five states per subproblem.

In addition to eliminating the need to solve (DLP) to optimality, Theorem 6 also leads to a natural stopping criterion by comparing the upper and lower bounds.


The example of §3.3 shows that the Lagrangian bound and policy can be arbitrarily poor, but the question remains how well the Lagrangian and LP-based relaxations perform on an important class of problems. In this section, we report on a computational simulation on derivatives of the restless bandit problem. Our objectives are to measure the difference between the bounds empirically, to compare the quality of the policies produced by the two relaxations, and to explore the inherent trade-off between bound quality and computational effort.

Our simulation study includes nine sets of random problems, generated according to the distributions indicated in Table 2. There are two controls (“active” and “passive”) available for each subproblem at each period. Multiarmed bandit problems comprise Set 1: passive rewards and weights are zero, passive transition probabilities are given by the identity matrix, and there is a single cardinality constraint on the number of active subproblems each period. Active reward parameters are sampled from the uniform distribution on [0, 1]. Initial state probabilities and active transition probabilities are sampled from a uniform distribution (and then appropriately normalized). The remaining problem sets differ from set 1 in their sizes and in their generation of “active” control constraint coefficients and rewards. Rewards in sets 4, 5, 8, and 9 are sampled from an exponential distribution with mean varying with the weight. This induces a natural correlation between weight and reward.

For each set of problems, we generate 200 instances. On each instance, we generate bounds using both the Lagrangian and LP-based methods. The Lagrangian bounds are computed using the LP formulation (PL) given in §2.3, and the LP-based bounds are computed using column generation, solving the knapsack problem (18) using CPLEX’s integer programming solver to generate columns. We terminate the column generation method when the gap between the upper bound \( \mathcal{H}^{LP}_C(\alpha) + 1/(1 - \beta) R_C(\alpha) \) and the lower bound \( \mathcal{H}^{LP}_C(\alpha) \) is 0.5% or less. We denote the final upper bound by \( \mathcal{H}^{LP}_U(\alpha) \).

We simulate the policies implied by the bounds using 200 sample paths for each instance, and for either 40 time steps (for \( \beta = 0.80 \)) or 175 time steps (for \( \beta = 0.95 \)), noting that \( 0.80^{40} \approx 0.95^{175} \lesssim 2 \times 10^{-4} \). Results of the simulation study, averaged for each problem set, are presented in Table 3.

We observe that the LP-based relaxation nearly always gives both significantly stronger bounds and better-performing policies, with the magnitude of the improvements

| Problem set | \( I \) | \( |S| \) | \( b \) | \( \mathbf{D}_i(\cdot, \text{“active”}) \) | \( r_i(\cdot, \text{“active”}) \) |
|-------------|-------|---|-----|-----------------|-----------------|
| 1           | 8     | 4 | 1   | 1               | U[0, 1]         |
| 2           | 8     | 4 | 1   | 0.51            | U[0, 1]         |
| 3           | 8     | 4 | 1   | U[0, 1]         | U[0, 1]         |
| 4           | 8     | 4 | 1   | U[0, 1]         | Exp(1/D)       |
| 5           | 8     | 4 | 1   | U[0.4, 0.6]     | Exp(0.2/(D - 0.4)) |
| 6           | 16    | 6 | 1   | 0.51            | U[0, 1]         |
| 7           | 16    | 6 | 1   | U[0, 1]         | U[0, 1]         |
| 8           | 16    | 6 | 1   | U[0, 1]         | Exp(1/D)       |
| 9           | 16    | 6 | 1   | U[0.4, 0.6]     | Exp(0.2/(D - 0.4)) |
dependent on both the nature of the problem and the discount factor. One interesting comparison is between the results for problem sets 1 and 2. The set of feasible actions is the same in both problem sets, but the representation of the feasible set differs. In line with our discussions of Theorems 2 and 3, the performance of the Lagrangian relaxation is dependent on this representation, whereas the LP-based relaxation is not. The gap between the Lagrangian and LP-based bounds is often substantial, whereas the policy performances are closer together.

The LP-based relaxation gives better performance, but at a computational cost. To examine this trade-off computationally, we take problems generated as in problem set 4, but for various numbers $I$ of subproblems. Table 4 gives the resulting bounds and computation times. Computation times are the total CPU time (on a 2.4 GHz Intel Xeon processor) required for the optimization solver (CPLEX 10.0 via AMPL), averaged over 50 problem instances. Table 4 is reported for $\beta = 0.80$. For $\beta = 0.95$, we have found the LP-based bound to require moderately more time, with the relative time difference increasing with $I$. We have also investigated computation times when we stop the column generation once the LP-based upper bound beats the Lagrangian bound. In this problem set, the Lagrangian bound performs quite well, so the LP-based upper bound beats the Lagrangian bound in just $10\%–30\%$ less time than in Table 4.

The linear programs needed to solve the Lagrangian relaxations are small (just 401 variables and 800 constraints for $I = 100$), and so it is not surprising that the Lagrangian method requires a fraction of the time required to generate the LP-based bounds, whose computation time grows exponentially with $I$. Nevertheless, in settings where the optimization must be accomplished either periodically or just once, the computation times for the LP-based bounds are quite acceptable on problems that are far beyond the reach of exact dynamic programming ($10^{60}$ states for $I = 100$).

Figure 3 presents two graphical views of the data in Table 4. The upper plot shows the difference in the two bounds as a function of $I$, whereas the lower plot traces

<table>
<thead>
<tr>
<th>Problem</th>
<th>$\beta$</th>
<th>Myopic policy</th>
<th>Lagrangian policy</th>
<th>LP policy</th>
<th>$\tilde{H}^{L_P}(\alpha)$</th>
<th>$H^{*}(\alpha)$</th>
<th>% diff: Lag. vs. LP bounds</th>
<th>% diff: Lag. vs. LP policies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.80</td>
<td>3.87</td>
<td>3.91</td>
<td>3.91</td>
<td>4.08</td>
<td>4.08</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>0.95</td>
<td>14.36</td>
<td>14.93</td>
<td>14.93</td>
<td>15.03</td>
<td>15.01</td>
<td>$-0.1$</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>0.80</td>
<td>3.89</td>
<td>3.89</td>
<td>3.93</td>
<td>4.11</td>
<td>7.27</td>
<td>76.9</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>0.95</td>
<td>14.09</td>
<td>14.37</td>
<td>14.70</td>
<td>14.81</td>
<td>27.09</td>
<td>82.9</td>
<td>2.2</td>
</tr>
<tr>
<td>3</td>
<td>0.80</td>
<td>7.11</td>
<td>7.44</td>
<td>7.43</td>
<td>8.35</td>
<td>8.68</td>
<td>4.0</td>
<td>$-0.1$</td>
</tr>
<tr>
<td>3</td>
<td>0.95</td>
<td>24.01</td>
<td>26.76</td>
<td>26.73</td>
<td>29.53</td>
<td>31.81</td>
<td>7.7</td>
<td>$-0.1$</td>
</tr>
<tr>
<td>4</td>
<td>0.80</td>
<td>7.08</td>
<td>7.60</td>
<td>7.61</td>
<td>8.44</td>
<td>9.06</td>
<td>7.3</td>
<td>0.1</td>
</tr>
<tr>
<td>4</td>
<td>0.95</td>
<td>24.86</td>
<td>28.44</td>
<td>28.55</td>
<td>30.84</td>
<td>34.24</td>
<td>11.0</td>
<td>0.4</td>
</tr>
<tr>
<td>5</td>
<td>0.80</td>
<td>6.49</td>
<td>7.07</td>
<td>7.21</td>
<td>7.96</td>
<td>9.32</td>
<td>17.1</td>
<td>1.9</td>
</tr>
<tr>
<td>5</td>
<td>0.95</td>
<td>23.32</td>
<td>27.12</td>
<td>28.12</td>
<td>29.99</td>
<td>36.78</td>
<td>22.6</td>
<td>3.6</td>
</tr>
<tr>
<td>6</td>
<td>0.80</td>
<td>4.20</td>
<td>4.20</td>
<td>4.21</td>
<td>4.45</td>
<td>8.03</td>
<td>80.4</td>
<td>0.3</td>
</tr>
<tr>
<td>6</td>
<td>0.95</td>
<td>14.66</td>
<td>14.68</td>
<td>15.17</td>
<td>15.55</td>
<td>28.42</td>
<td>82.8</td>
<td>3.3</td>
</tr>
<tr>
<td>7</td>
<td>0.80</td>
<td>9.43</td>
<td>9.78</td>
<td>9.78</td>
<td>10.93</td>
<td>10.91</td>
<td>$-0.2$</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>0.95</td>
<td>28.15</td>
<td>31.97</td>
<td>32.03</td>
<td>35.63</td>
<td>36.30</td>
<td>1.9</td>
<td>0.2</td>
</tr>
<tr>
<td>8</td>
<td>0.80</td>
<td>8.79</td>
<td>9.31</td>
<td>9.32</td>
<td>10.56</td>
<td>10.82</td>
<td>2.4</td>
<td>0.1</td>
</tr>
<tr>
<td>8</td>
<td>0.95</td>
<td>27.89</td>
<td>32.54</td>
<td>32.77</td>
<td>35.96</td>
<td>38.11</td>
<td>6.0</td>
<td>0.7</td>
</tr>
<tr>
<td>9</td>
<td>0.80</td>
<td>8.45</td>
<td>8.83</td>
<td>9.06</td>
<td>10.35</td>
<td>11.74</td>
<td>13.4</td>
<td>2.6</td>
</tr>
<tr>
<td>9</td>
<td>0.95</td>
<td>25.63</td>
<td>30.37</td>
<td>31.75</td>
<td>34.75</td>
<td>40.36</td>
<td>16.2</td>
<td>4.6</td>
</tr>
</tbody>
</table>

| Problem | $|S|$ | $\tilde{H}^{L_P}(\alpha)$ | $H^{*}(\alpha)$ | Solver CPU secs. (LP-based bound) | Solver CPU secs. (Lagrangian bound) |
|---------|-----|-----------------|---------------|-------------------------------|---------------------------------|
| 2       | 16  | 3.82            | 4.32          | 0.0474                        | 0.0015                          |
| 4       | 256 | 6.04            | 6.68          | 0.1837                        | 0.0020                          |
| 8       | 65,536 | 8.33         | 8.87          | 0.8437                        | 0.0021                          |
| 12      | $\sim 10^7$ | 9.81          | 10.33         | 2.3833                        | 0.0026                          |
| 16      | $\sim 10^9$ | 11.05         | 11.51         | 5.2125                        | 0.0027                          |
| 20      | $\sim 10^{12}$ | 11.64        | 12.09         | 10.001                        | 0.0026                          |
| 24      | $\sim 10^{14}$ | 12.54         | 12.96         | 16.811                        | 0.0030                          |
| 30      | $\sim 10^{18}$ | 13.57         | 13.95         | 36.391                        | 0.0031                          |
| 40      | $\sim 10^{24}$ | 14.60         | 15.00         | 99.820                        | 0.0036                          |
| 75      | $\sim 10^{55}$ | 17.20         | 17.46         | 1,659.5                       | 0.0057                          |
| 100     | $\sim 10^{60}$ | 18.48         | 18.70         | 6,722.0                       | 0.0064                          |
the difference in policy performance as a function of $I$. For this set of problems, the policy differences are quite small, but both the bounds and policy performances seem to converge toward 0 as $I \to \infty$. Thus, in contrast to the experience in the example of §4.3.1, for this problem set the Lagrangian seems to be a good choice for large instances because it achieves similar bounds and policies for a fraction of the computational effort required to compute the LP-based relaxation.

Figure 4 plots an upper bound on the average optimality gap per subproblem for each method as a function of the number of subproblems $I$. The “best policy” is for each problem instance and serves as a proxy for the optimal value $H(\alpha)$, which cannot be computed except for the smallest problems. We see that, in keeping with Theorems 4 and 5, the average optimality gaps approach zero as $I$ increases. Of course, Theorem 4, which requires $\lambda$ to be state dependent, does not guarantee such behavior in this case because we force $\lambda$ to be state independent.

7. Concluding Remarks

Although the Lagrangian relaxation we consider here is a generalization (see §2.5) of that found in the literature on techniques for static integer programming (Fisher 1981), our LP-based relaxation is specific to the dynamic programming context. Nevertheless, the lessons learned in this paper parallel those found in the static integer programming literature. When it is numerically possible to solve an integer program exactly, then that is by all means preferable to solving its Lagrangian relaxation. In the case of stochastic dynamic programming, the same is true. Because solving a dynamic program is fundamentally harder than solving an integer program, we consider the LP-based relaxation, which is intermediate in complexity between the exact solution and the Lagrangian relaxation. We prove that the bound obtained is stronger than the Lagrangian bound, and empirically we find that the resulting policies are stronger as well.

However, on problems with very high-dimensional state spaces, the only viable option is to solve the Lagrangian relaxation, which is dramatically easier to solve because it decomposes the problem into multiple subproblems. Just as in integer programming, we find that the Lagrangian relaxation can provide excellent bounds and policy performance as the number of subproblems increases, depending on the problem structure. Nevertheless, we have provided an example in which the asymptotic performance of the Lagrangian policy and bound can be arbitrarily poor compared with those from the LP-based relaxation. For this reason, we advise solving the LP-based relaxation instead of the Lagrangian whenever possible.

In short, we have provided useful theory and insights into two practical relaxations of a broad, important class of stochastic dynamic programs. Such comparisons have not been performed previously. Our hope is that this work will stimulate the increased usage of dynamic programming methods on real-world industrial problems and spawn
future research comparing competing practical approaches to approximate dynamic programming.

8. Electronic Companion

An electronic companion to this paper is available as part of the online version that can be found at http://or.pubs.informs.org/ecompanion.html.

Endnote

1. The term “weakly coupled dynamic program” has previously appeared in the literature. The problems we consider match the discussion of Hawkins (2003) and the problem with “instantaneous” constraints as discussed in Meuleau et al. (1998).

Acknowledgments

The authors thank three anonymous referees and the associate editor for constructive reviews that significantly improved this paper. Both authors gratefully acknowledge the financial support of University of Chicago Graduate School of Business. The second author acknowledges support from the FMC Faculty Research Fund at University of Chicago Graduate School of Business.

References