JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 37, No. 2, March–April 2014

Lower Bounding Linear Program for the Perimeter Patrol Optimization Problem

Krishnamoorthy Kalyanam*
Infoscitex Corporation, Dayton, Ohio 45431
Myoungkuk Park† and Swaroop Darbha‡
Texas A&M University, College Station, Texas 77843
David Casbeer§ and Phil Chandler¶
U.S. Air Force Research Laboratory, Wright–Patterson Air Force Base, Ohio 45433
and
Meir Pachter**
U.S. Air Force Institute of Technology, Wright–Patterson Air Force Base, Ohio 45433

DOI: 10.2514/1.60487

In this article, a stochastic optimal control problem involving an unmanned aerial vehicle flying patrols around a perimeter is considered. To determine the optimal control policy, one has to solve a Markov decision problem, whose large size renders exact dynamic programming methods intractable. Therefore, a state aggregation based approximate linear programming method is used instead, to construct provably good suboptimal patrol policies. The state space is partitioned and the optimal cost-to-go or value function is restricted to be a constant over each partition. The resulting restricted system of linear inequalities embeds a family of Markov chains of lower dimension, one of which can be used to construct a lower bound on the optimal value function. In general, the construction of a lower bound requires the solution to a combinatorial problem. But the perimeter patrol problem exhibits a special structure that enables tractable linear programming formulation for the lower bound. This is demonstrated and numerical results that corroborate the efficacy of the proposed methodology are also provided.

I. Introduction

THE following base perimeter patrol problem is addressed: an unmanned aerial vehicle (UAV) and a remotely located operator cooperatively perform the task of perimeter patrol. Alert stations consisting of unattended ground sensors (UGSs) are located at key locations along the perimeter. Upon detection of an incursion in its sector, an alert is flagged by the UGS. The statistics of the alerts' arrival process are assumed known. A camera-equipped UAV is on continuous patrol along the perimeter and is tasked with inspecting UGSs with alerts. On arrival to an alert flagged UGS, the UAV orbits the UGS/alert station and a video feed is transmitted to the operator. The latter can steer the gimballed camera while looking for the cause of the alarm. Naturally, the longer a UAV dwells (loiters) at an alert site, the more information it gathers and transmits to the operator. We have an interesting and novel problem framework wherein the UGSs do the sensing and provide actionable intelligence in the form of alerts that the UAV can act upon. Once the UAV reaches a triggered UGS, it captures video or pictures of the vicinity until the controller dictates it to move on. This video imagery is passed on to a human operator, who decides the future course of action based on whether he perceives the source of the alert to be a threat or a nuisance. Arguably, if the UAV gets to the UGS location not long after it was tripped, the video imagery will have visual (identifiable) information on the intruder that triggered the UGS.

The objective, then, is to maximize the information gained and, at the same time, reduce the expected response time to alerts elsewhere. The problem is simplified by considering discrete-time evolution equations for updating the position of the UAV and also by considering only a finite (fixed) number m of UGS locations. It is assumed that the UAV has access to real-time information about the status of alerts (whether they have been attended to or not) at each alert station. Because the UAV is constantly on patrol or is servicing a triggered UGS, the framework considered here is analogous to the so-called cyclic polling system, which is a well-researched topic in queueing systems [1,2]. The perimeter patrol problem as envisaged here falls in the domain of discrete-time controlled queueing systems [3]. In general, a queueing system includes arriving customers, servers, and waiting lines/buffers, or queues, for the customers awaiting service [4]. In the context of perimeter patrol, the "customers" are the flagged UGSs/alerts waiting to be serviced and the UAV is the server. In queueing theory, the queues/buffers could be of finite or infinite capacity. We consider a unit/single buffer, for the UGS either flags an alert or it does not. Once it flags an alert, its state does not change, even if additional triggering events were to occur, until the status is reset by a loitering UAV. Hence, there is at most only one alert waiting at an alert site. Therefore, the perimeter patrol problem as envisaged here constitutes a multiqueue, single-server, unit-buffer queueing system with deterministic (interstation) travel and service times [1]. Connections to queueing theory and the dynamic vehicle routing problem [5] have been established in an earlier work (see [6] and references therein). In the literature, a persistent patrolling task is typically defined as continuously visiting the nodes on a graph so as to minimize the time lag between two visits [7]. Considerable work has been done on devising patrol strategies that optimize performance metrics, such as refresh time and latency (see [8,9] and references therein). We note that the framework considered herein is different in that we are also optimizing over the time the UAV spends at the alert station.

The control problem is to determine how long the UAV should dwell at a triggered alert station/UGS, and also which station to move toward next upon completion of a service. The present work

Presented as Paper 2012-2454 at the Infotech@Aerospace 2012, Garden Grove, CA, 19–21 June 2012; received 25 September 2012; revision received 16 April 2013; accepted for publication 22 April 2013; published online 5 February 2014.

*Research Scientist; krishnak@ucla.edu. Member AIAA.
†Graduate Student, Department of Mechanical Engineering.
‡Professor, Department of Mechanical Engineering.
§Research Engineer, Control Automation Branch. Member AIAA.
¶Tech Advisor, Control Automation Branch. Member AIAA.
**Professor, Electrical Engineering Department. Associate Fellow AIAA.

considers a metric that trades off the information gained by the UAV as a function of the time spent loitering at the alert sites gathering information on the nature of the alerts, and the time taken by the UAV to respond to other alerts; that is, the time it takes the UAV to reach a triggered UGS. Thus, the metric increases monotonically with the duration of time spent by the UAV at an alert station and decreases monotonically with the delay in responding to alerts. The problem is formulated as a stochastic dynamic program for which the optimal stationary policy can be obtained via dynamic programming (DP). Unfortunately, for practical values of m, the problem essentially becomes computationally intractable and hence, an approximate linear programming (LP) method is employed to arrive at good suboptimal policies. In an earlier work, the authors employed exact DP to solve the perimeter problem, where a different metric that minimizes the queue length (number of alerts in the system) was considered [6]. In the past, we have also considered a more generic robotic surveillance problem framework, where the approximate LP methods can be applied [10]. In the next section, the LP approach to DP is introduced before delving into the proposed methodology.

II. Linear Programming Approach to Dynamic Programming

The LP approach to solving dynamic programs (DPs) originated from the papers [11–14]. The basic feature of an LP approach for solving DPs corresponding to maximization of a discounted payoff is that the optimal solution of the DP (also referred to as the optimal value function) is the optimal solution of the LP for every nonnegative cost function. The constraint set describing the feasible solution of the LP and the number of independent variables are typically very large (curse of dimensionality) and hence, obtaining the exact solution of a DP (stochastic or otherwise) via an LP approach is not practical. Despite this limitation, the LP approach provides a tractable method for approximate DP [15–17].

One such method is to approximate the value (cost-to-go) function by a linear functional of a priori chosen basis functions [16]. This approach is attractive in that, for a certain class of basis functions, feasibility of the approximate (or restricted) LP is guaranteed [18]. A straightforward method for selecting the basis functions is through a state aggregation method. Here, the state space is partitioned into disjoint sets or partitions and the approximate value function is restricted to be the same for all the states in a partition. The number of variables for the LP therefore reduces to the number of partitions. State aggregation based approximation techniques were originally proposed by [19–21]. Since then, substantial work has been reported in the literature on this topic (see [22] and the reference therein). In this work, the state aggregation method is adopted. Although imposing restrictions on the value function reduces the size of the restricted LP, the number of constraints does not change. Because the number of constraints is at least of the same order as the number of states of the DP, one is faced with a restricted LP with a large number of constraints. Popular methods to prune the constraint set include aggregation of constraints, subsampling of constraints [18], and constraint generation methods [23,24].

One can construct a suboptimal policy from the solution to the restricted LP by considering the policy that is greedy with respect to the approximate value function [25]. By construction, the expected discounted payoff for the suboptimal policy will be a lower bound to the optimal value function and hence, can be used to quantify the quality of the suboptimal policy. But, the lower bound computation is not efficient because the procedure involved is tantamount to policy evaluation, which involves the solution to a system of linear equations of the same size as the state-space. In an earlier work [26], we demonstrated (empirically and via simulation results) that the aggregation based method provides tractable upper and lower bounds to the optimal value function. But, the LP behind the lower bound was developed based on intuition, and proof that it indeed results in a lower bound was not provided. Instead, we provided a plausible explanation as to why it does so (see Sec. 4.2 of [26]). In this article, we rigorously construct the lower bounding LP from first principles and demonstrate via simulation results that it results in a tight lower bound. The novel contributions of this article may be summarized as follows:

1) A subset of the constraints of the restricted LP can be used for constructing a lower bound for the optimal value function. However, this involves solving a disjunctive LP, which is (in general) not tractable.

2) For the application considered herein, it is shown that the lower bounding disjunctive LP can be solved efficiently because it reduces to a standard LP formulation.

The rest of the paper is organized as follows: a mathematical model for the perimeter patrol problem is provided in Sec. III. The problem is then posed as a Markov Decision Problem (MDP) in Sec. IV. The state aggregation method and the restricted LP approach are detailed in Sec. V. A novel disjunctive LP that provides a lower bound on the optimal value function is developed in Sec. V.C, followed by its efficient LP characterization for the patrol problem. Simulation results confirming the efficacy of the proposed method are provided in Sec. VI, followed by some concluding remarks in Sec. VII.

III. Perimeter Patrol: Problem Statement

The patrolled perimeter is a simple closed curve with $N(\geq m)$ nodes that are spatially uniformly separated, of which m are the alert stations (UGS locations). Let the m distinct station locations be elements of the set $\Omega \subset \{0, \ldots, N-1\}$. A typical scenario shown in Fig. 1 has 15 nodes of which nodes $\{0, 3, 7, 11\}$ correspond to the alert stations. Here, station locations 3, 7, and 11 have no alerts, and station location 0 has an alert being serviced by the UAV. At time instant t, let $\ell(t)$ be the position of the UAV on the perimeter ($\ell \in \{0, \ldots, N-1\}$), $d(t)$ be the dwell time (number of loiters completed if at an alert site) and $\tau_j(t)$ be the delay in servicing an alert at location $j \in \Omega$. Let $y_j(t)$ be a binary but random variable indicating the arrival of an alert at location $j \in \Omega$ at time t. It is assumed that the statistics associated with the random variable $y_j(t)$ are known and that y_j; $j \in \Omega$ are independent. Each station has an independent Bernoulli arrival stream of alerts at the rate of α (alerts per unit time). Once a station has an alert waiting, no new alerts can arrive there until the current one is serviced. Hence, there are 2^m possibilities for the vector of alerts, $y(t) = [\, y_1(t) \quad y_2(t) \quad \cdots \quad y_m(t) \,]$ ranging from the binary equivalent of 0 to $2^m - 1$, indicating whether or not there is an alert waiting to be serviced at each of the m stations.

The control decisions are indicated by the variable u. If $u = 1$, then the UAV continues in the same direction as before; if $u = -1$, then the UAV reverses its direction of travel and, if $u = 0$, the UAV dwells at the current alert station. It is assumed that the UAV advances by one node in unit time if $u \neq 0$. It is also assumed that the time to complete one loiter is also the unit time. The UAV's direction of travel is denoted by ω, where $\omega = 1$ and $\omega = -1$ indicate the clockwise and

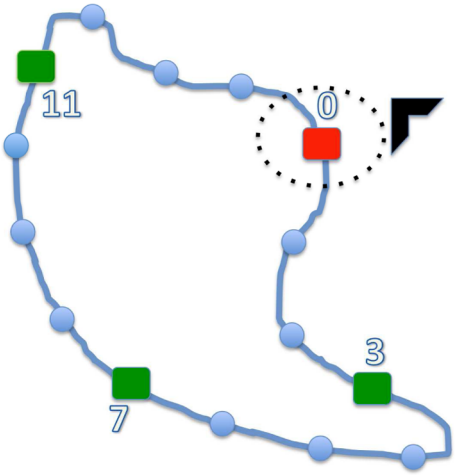

Fig. 1 Perimeter patrol scenario with UAV servicing alert at station 0.

counterclockwise directions, respectively. One may write the state update equations for the system as follows:

$$\begin{aligned}\ell(t+1) &= [\ell(t) + \omega(t)u(t)] \bmod N\\ \omega(t+1) &= \omega(t)u(t) + \delta[u(t)]\\ d(t+1) &= [d(t)+1]\delta[u(t)]\\ \tau_j(t+1) &= [\tau_j(t)+1]\{1-\delta[\ell(t)-j]\delta[u(t)]\}\\ &\quad \max\{\sigma[\tau_j(t)], y_j(t)\}, \quad \forall\ j \in \Omega\end{aligned} \tag{1}$$

where δ is the Kronecker delta function and $\sigma(\cdot) = 1 - \delta(\cdot)$. The status of the alert at station location $j \in \Omega$ at time t is given by $\mathcal{A}_j(t)$:

$$\mathcal{A}_j(t) = \begin{cases} 0, & \text{if } \tau_j(t) = 0 \\ 1, & \text{otherwise} \end{cases}, \quad \forall\ j \in \Omega \tag{2}$$

Also, the constraints in state and control are given by $u(t) = 0$ only if $\ell(t) \in \Omega$ and $d(t) \le D$. If $d(t) = D$, then $u(t) \neq 0$; that is, the UAV is forced to leave the station if it has already completed the maximum (allowed) number of dwell orbits. Combining the different components in Eq. (1), the evolution equations can be compactly represented as

$$x(t+1) = f[x(t), u(t), y(t)] \tag{3}$$

where $x(t)$ is the system state at time t with components $\ell(t)$, $\omega(t)$, $d(t)$, and $\tau_j(t)$, $\forall\ j \in \Omega$. Let us denote the 2^m possible values (from the m digit binary representation of 0 to $2^m - 1$) that $y(t)$ can take by the row vector $\tilde{y}_j \in \mathcal{R}^m$, $j = 1, \ldots, 2^m$. Given the Bernoulli alert arrival process with parameter α, the probability that there is no alert occurring in a unit time step, $q = e^{-\alpha}$. Hence, the probability that $y(k)$ takes any one of 2^m possible values is given by

$$p_j := \mathcal{P}\{y(t) = \tilde{y}_j\} = q^{(m-a_j)}(1-q)^{a_j}, \qquad j \in \{1, \ldots, 2^m\} \tag{4}$$

where $a_j = \sum_{i=1}^{m} \tilde{y}_j(i)$ denotes the number of stations with alerts for the alert arrival configuration indicated by $\tilde{y}_j$. We note that the solution method and the results hereafter do not rely on the Bernoulli assumption and only require that the probabilities p_j, $j \in \{1, \ldots, 2^m\}$ be provided.

IV. Markov Decision Problem

Our objective is to find a suitable policy that simultaneously minimizes the service delay and maximizes the information gained upon loitering. The information gain $\mathcal{I}$, which is based on an operator error model, is plotted as a function of dwell time in Fig. 2. The operator modeling and derivation of the information gain function have been left out for brevity (for details, see Sec. II.A of [6]). The one-step payoff/reward function is modeled as

$$R_u(x) = [\mathcal{I}(d_x + 1) - \mathcal{I}(d_x)]\delta(u) - \rho \max\{\bar{\tau}_x, \Gamma\}, \qquad x \in \mathcal{S} \tag{5}$$

where d_x is the dwell associated with state x, and $\bar{\tau}_x = \max_{j\in\Omega} \tau_{j,x}$ is the worst service delay (among all stations) associated with state x. The parameter $\Gamma (\gg 0)$ is a judiciously chosen maximum penalty. The positive parameter ρ is a constant weighing the incremental information gained upon loitering once more at the current location against the delay in servicing alerts at other stations. Let $\mathcal{S}$ denotes the set of all possible system states. From the state definition in Eq. (3), the total number of states is given by

$$|\mathcal{S}| = 2 \times N \times (\Gamma+1)^m + D \times m \times (\Gamma+1)^{m-1} \tag{6}$$

where the factor 2 comes from the UAV being bidirectional. For the loiter states, directionality is irrelevant and hence, when $d \ge 1$, ω is reset to 1. From the reward function definition, Eq. (5), we see that the state-space $\mathcal{S}$ only includes states x with $\tau_i \le \Gamma$, $\forall\ i \in \Omega$ and so, is finite. Any stationary policy π specifies for each state $x \in \mathcal{S}$ a control action $u = \pi(x)$. The transition probability matrix for a fixed u is given by P_u, where $P_u(i, j)$ indicates the probability of transitioning from state i to state j under the influence of u (the states are ordered arbitrarily). From the Bernoulli distribution in Eq. (4), one can write

$$P_u(x, y) = \begin{cases} 0, & \text{if } y \neq f(x, u, \tilde{y}_l) \text{ for any } l \in \{1, \ldots, 2^m\} \\ \sum_{j\in\mathcal{C}} p_j, & \text{where } \mathcal{C} = \{l | y = f(x, u, \tilde{y}_l)\} \end{cases} \tag{7}$$

The column vector of immediate payoffs associated with the policy π is given by R_π, where $R_\pi(x) = R_{\pi(x)}(x)$. Solving the stochastic control problem amounts to selecting a policy that maximizes the infinite-horizon discounted reward starting from x_0,

$$V_\pi(x_0) = \mathbf{E}\left[\sum_{t=0}^{\infty} \lambda^t R_\pi(x(t)) | x(0) = x_0\right]$$

where $\lambda \in [0, 1)$ is a temporal discount factor. The optimal policy is obtained by solving Bellman's equation,

$$V^*(x) = \max_u \left\{ R_u(x) + \lambda \sum_{l=1}^{2^m} p_l V^*(f(x, u, \tilde{y}_l)) \right\}, \quad \forall\ x \in \mathcal{S} \tag{8}$$

where $V^*(x)$ is the optimal value function (or optimal discounted payoff) starting from state x. The optimal policy then is given by

$$\pi^*(x) = \arg\max_u \left\{ R_u(x) + \lambda \sum_{l=1}^{2^m} p_l V^*(f(x, u, \tilde{y}_l)) \right\}, \quad \forall\ x \in \mathcal{S} \tag{9}$$

Recall that the problem size $|\mathcal{S}|$ is an mth-order polynomial in Γ [Eq. (6)]. Therefore, solving for the optimal value function and policy using exact DP methods is rendered intractable for practical values of Γ and m. For this reason, one is interested in tractable approximate methods that yield suboptimal solutions with some guarantees on the deviation of the associated approximate value function from the optimal one. Before venturing into the approximation scheme, some preliminary results that help us in establishing the aggregation based LP are provided.

Bellman's equation suggests that the optimal value function satisfies the following set of linear inequalities:

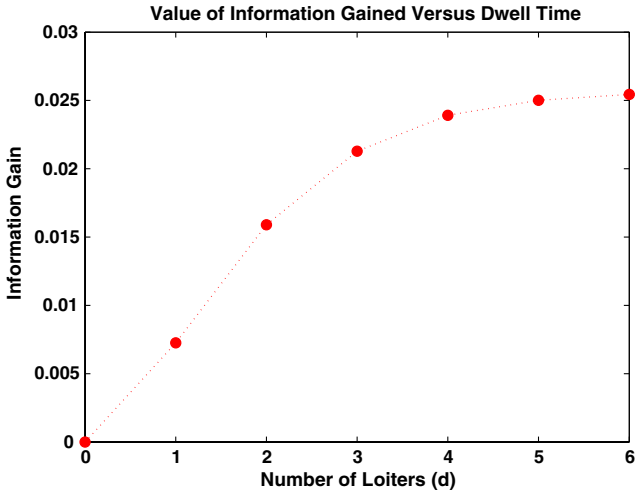

Fig. 2 Information gain as a function of dwell time.

$$V(x) \geq R_u(x) + \lambda \sum_{l=1}^{2^m} p_l V(f(x,u,\tilde{y}_l)), \quad \forall\, u, \quad \forall\, x \in \mathcal{S} \quad (10)$$

They are hereafter referred to as the Bellman inequalities. The Bellman inequalities may be compactly represented as

$$V \geq R_u + \lambda P_u V, \quad \forall\, u \quad (11)$$

It is a well-known result that any V that satisfies the Bellman inequalities is an upper bound to the optimal value function V^* [27].

Lemma 1: If V satisfies Eq. (11), then $V \geq V^*$.

The preceding property leads to the so-called "exact LP" that can be used to compute the optimal value function (for proof, see [16]).

Corollary 1: The optimal solution to the LP,

$$\text{ELP} := \min\ c^T V, \qquad \text{subject to} \quad V \geq R_u + \lambda P_u V, \quad \forall\, u \quad (12)$$

is the optimal value function V^* for every $c > 0$.

V. State Aggregation Based Approximate DP

In the perimeter patrol problem considered herein, by Eq. (5), there is a structure to the reward function $R_u(x)$. To explain this structure, consider a station where an alert is being serviced by the UAV. The information gained by the UAV about the alert is only a function of the service delay at the station and the amount of time the UAV dwells at the station servicing the alert. Therefore, no matter what the delays are at the other stations, the reward is the same as long as the maximum delay and the dwell time of the UAV at the station are the same. This leads to a natural partitioning of the state-space; that is, aggregate all the states that have the same ℓ, ω, d, $\mathcal{A}_j$, $\forall\ j \in \Omega$ and $\bar{\tau} = \max_{j\in\Omega} \tau_j$ into one partition. As a result of aggregation, the number of partitions can be shown to be

$$M = 2N + 2N(2^m - 1)\Gamma + mD + mD(2^{m-1} - 1)\Gamma \quad (13)$$

which is linear in Γ and hence, considerably smaller than the total number of states [Eq. (6)]. As per the preceding scheme, $\mathcal{S}$ is partitioned into M disjoint sets $\mathcal{S}_i$, $i = 1, \ldots, M$ and the set $\mathcal{S}_i$ is called the ith partition. Henceforth, the following notation shall be used: if $f(x,u,Y)$ represents the state the system transitions to starting from x and subject to a control input u and a stochastic disturbance Y, then $\bar{f}(x,u,Y)$ represents the partition to which the final state belongs.

A. Partial Ordering of States

Let $\ell_x, d_x, \omega_x, \tau_{j,x}$ and $\mathcal{A}_{j,x}$ represent the location, dwell, direction of UAV's motion, the service delay, and alert status, respectively, at station location $j \in \Omega$ corresponding to some state $x \in \{1, \ldots, |\mathcal{S}|\}$. Also, let $\ell(i)$, $d(i)$, $\omega(i)$, $\bar{\tau}(i)$, and $\mathcal{A}_j(i)$ denote the location, dwell, direction, maximum delay (among all stations), and the alert status, respectively, at station location $j \in \Omega$ that correspond to some partition index $i \in \{1, \ldots, M\}$. A partial ordering of the states is introduced according to $x \geq y$ iff $\ell_x = \ell_y$, $d_x = d_y$, $\omega_x = \omega_y$ and $\tau_{j,x} \geq \tau_{j,y}$,$\forall\ j \in \Omega$. By the same token, partitions are also partially ordered according to $\mathcal{S}_i \geq \mathcal{S}_j$ iff, for every $z \in \mathcal{S}_j$, there $\exists x \in \mathcal{S}_i$ such that $x \geq z$. The partial ordering brings about the following monotonicity property in the optimal value function that is central to our result (for proof, see the Appendix).

Lemma 2: If $\mathcal{S}_i \geq \mathcal{S}_j$, $\min_{x\in\mathcal{S}_i} V^*(x) \leq \min_{z\in\mathcal{S}_j} V^*(z)$.

From a given state $x \in \mathcal{S}_i$, define $z_x^{i,u}$ to be the tuple of partition indices that the system can transition to under control action u; that is, $z_x^{i,u} = [\bar{f}(x,u,\tilde{y}_1), \ldots, \bar{f}(x,u,\tilde{y}_{2^m})]$. Let $\mathcal{T}(i,u)$ denote the set of all distinct $z_x^{i,u}$ for a given partition index i and control u. For the sake of notational simplicity, let the lth component of any tuple $k \in \mathcal{T}(i,u)$ be denoted by k_l. Let the cardinality of the set $\mathcal{T}(i,u)$ be denoted by $|\mathcal{T}(i,u)|$. The partitions belong to one of two types:

Type 1: If the UAV is at a station with an alert, the dwell time is zero and there is an alert at some other station; that is,

$$\ell(i) \in \Omega, \qquad d(i) = 0, \qquad \mathcal{A}_{\ell(i)}(i) = 1, \quad \text{and} \quad \mathcal{A}_j(i) = 1, \quad \text{for some } j \in \Omega, \qquad j \neq \ell(i) \quad (14)$$

Else, it is of *Type 2*.

Let $\mathcal{P}_1$ denote the set of all partition indices of Type 1; that is, if a partition index i is of Type 1, then $i \in \mathcal{P}_1$.

Lemma 3:

$$|\mathcal{T}(i,u)| = \begin{cases} \bar{\tau}(i), & i \in \mathcal{P}_1 \text{ and } u = 0 \\ 1, & \text{otherwise} \end{cases}$$

Proof: Consider partition index $i \in \mathcal{P}_1$ and control input $u = 0$. Because the UAV has decided to loiter at the current station $\ell(i) \in \Omega$, the service delay at that station $\tau_{\ell(i)}$ will be reset to zero in the next time step. Hence, the future state (and partition) maximum delay will be determined by the highest of the service delays, say $\tilde{\tau}$, among the other stations with alerts (at least one such station exists because partition $i \in \mathcal{P}_1$). Therefore, $\forall\ j \in \{1, \ldots, \bar{\tau}(i)\}$, $\exists x_j \in \mathcal{S}_i$ such that $\tilde{\tau}_{x_j} = j$. The corresponding tuple of future partition indices $z_{x_j}^{i,0} = [\bar{f}(x_j,u,\tilde{y}_1), \ldots, \bar{f}(x_j,u,\tilde{y}_{2^m})]$ will have maximum delay $j+1$, and so, $\mathcal{T}(i,0) = \cup_{j=1}^{\bar{\tau}(i)} \{z_{x_j}^{i,0}\} \Rightarrow |\mathcal{T}(i,0)| = \bar{\tau}(i)$. For all other control choices $u \neq 0$, all the states $x \in \mathcal{S}_i$ will transition to future states with the same maximum delay $\bar{\tau}(i) + 1$. Therefore, for $u \neq 0$, $\mathcal{T}(i,u)$ is a singleton set and hence, $|\mathcal{T}(i,u)| = 1$. If $\bar{\tau}(j) > 0$, the corresponding partition, $\mathcal{S}_j$ is a singleton set as per the aggregation scheme (see Sec. V.A) and hence, $|\mathcal{T}(j,u)| = 1, \forall\ u$.

B. Restricted Linear Program

Let us restrict the ELP [Eq. (12)] by requiring further that $V(x) = v(i)$ for all $x \in \mathcal{S}_i$, $i = 1, \ldots, M$. Augmenting these constraints to the exact LP, one gets the following restricted LP for some $c > 0$:

$$\text{RLP} := \min \sum_{i=1}^{M} \underbrace{\sum_{x\in\mathcal{S}_i} c(x)}_{\bar{c}(i)}\, v(i) \text{ subject to}$$

$$v(i) \geq R_u(x) + \lambda \sum_{l=1}^{2^m} p_l v(\bar{f}(x,u,\tilde{y}_l)), \quad \forall\, x \in \mathcal{S}_i, \quad i = 1, \ldots, M, \quad \forall\, u \quad (15)$$

The restricted LP deals with a smaller number of variables, $M (\ll |\mathcal{S}|)$. In a related work, the authors have demonstrated how the upper bound formulation RLP can be solved efficiently for the perimeter patrol problem [28]. In this article, the focus instead is on a lower bound formulation. Now, an approximate value function can be constructed from every feasible solution to RLP according to $\tilde{V}(x) = v(i)$, $\forall\ x \in \mathcal{S}_i$, $i = 1, \ldots, M$. Because the approximate value function satisfies, by construction, the Bellman inequalities [Eq. (10)], it is automatically an upper bound to V^* (by Lemma 1). Therefore, if v_c^* is the optimal solution to RLP [Eq. (15)] for some cost vector $\bar{c} > 0$, then clearly $v_c^*(i) \geq V^*(x), \forall\ x \in \mathcal{S}_i, i = 1, \ldots, M$. It has been shown that v_c^* is independent of the choice of cost vector $\bar{c}$ and indeed it is the least upper bound for the partitioning scheme given by $\mathcal{S}_i$, $i = 1, \ldots, M$ [29]. Using any approximate value function (not necessarily an upper bound) $\tilde{V}$, one can construct the corresponding suboptimal "greedy" policy according to

$$\pi(x) = \arg\max_u \left\{ R_u(x) + \lambda \sum_y P_u(x,y) \tilde{V}(y) \right\}, \quad \forall\, x \in \mathcal{S}$$

Let the corresponding improvement in value function be defined as, $\alpha(x) := R_\pi(x) + \lambda \sum_y P_\pi(x,y) \tilde{V}(y) - \tilde{V}(x)$. The following bounds hold [25,27]: $\forall\ x \in \mathcal{S}$,

$$\tilde{V}(x) + \frac{\min_y \alpha(y)}{1-\lambda} \le V^*(x) \le \tilde{V}(x) + \frac{\max_y \alpha(y)}{1-\lambda} \tag{16}$$

Furthermore, the expected discounted payoff V_π corresponding to the suboptimal policy π satisfies the following bound [25]:

$$\tilde{V}(x) + \frac{\min_y \alpha(y)}{1-\lambda} \le V_\pi(x) \le V^*(x), \quad \forall\, x \in \mathcal{S}$$

In the literature, typically V_π is used as a candidate for a lower bound on the optimal value function. The rationale for obtaining a lower bound is clear: if one obtains tight lower and upper bound approximations, then the "distance" (based on some suitable norm) between the two would clearly indicate the quality of the approximations. In this context, the lower bound provided by Eq. (16) is typically very conservative. On the other hand, computation of $V_{\tilde{\pi}}$ involves solving a linear system of equations of size $|\mathcal{S}|$, which is as bad as solving the original Markov Decision Problem. Therefore, one is interested in alternate efficient methods to compute a lower bound. In the next section, such a method is established and its applicability to the perimeter patrol problem is shown.

C. Lower Bound for the Optimal Value Function

Recall that each $x \in \mathcal{S}_i$, $V^*(x)$ satisfies the Bellman inequality, Eq. (10):

$$\begin{aligned} V^*(x) &\ge R_u(x) + \lambda \sum_{l=1}^{2^m} p_l V^*(f(x,u,\tilde{y}_l)), \quad \forall\, u \\ &\ge R_u(x) + \lambda \sum_{l=1}^{2^m} p_l \min_{y \in \bar{f}(x,u,\tilde{y}_l)} V^*(y), \quad \forall\, u \end{aligned} \tag{17}$$

Let $\underline{w}(i) := \min_{x \in \mathcal{S}_i} V^*(x)$, $i = 1, \ldots, M$. Then, it follows that

$$\underline{w}(i) \ge \min_{x \in \mathcal{S}_i} \left\{ R_u(x) + \lambda \sum_{l=1}^{2^m} p_l \underline{w}(\bar{f}(x,u,\tilde{y}_l)) \right\}, \quad \forall\, u \tag{18}$$

The preceding set of inequalities motivates the following nonlinear program:

$$\text{NLP:} = \min\ \bar{c}^T w, \qquad \text{subject to} \tag{19}$$

$$w(i) \ge \min_{x \in \mathcal{S}_i} \left\{ R_u(x) + \lambda \sum_{l=1}^{2^m} p_l w(\bar{f}(x,u,\tilde{y}_l)) \right\}, \quad \forall\, u,\ i = 1, \ldots, M \tag{20}$$

Let w_c^* be the optimal solution to the NLP for some $\bar{c} > 0$. By construction, w is a feasible solution to the NLP and hence,

$$\bar{c}^T w_c^* \le \bar{c}^T \underline{w} = \sum_{i=1}^{M} \bar{c}(i) \min_{x \in \mathcal{S}_i} V^*(x)$$

Therefore, by choosing $\bar{c}(i) = 1$ and $\bar{c}(j) = 0$ for all $j \ne i$, one can obtain a lower bound to the optimal value function for all the states in the ith partition. Unfortunately, the NLP is combinatorial in nature and hence, intractable for a general MDP. However, the perimeter patrol problem exhibits a special structure that reduces the NLP to a standard LP formulation.

Theorem 1: The NLP [Eq. (20)] reduces to the following LP:

$$\text{LBLP:} = \min\ \bar{c}^T w, \qquad \text{subject to}$$

$$w(i) \ge r_u(i) + \lambda \sum_{l=1}^{2^m} p_l w(k_l), \quad \forall\, u, \qquad i = 1, \ldots, M \tag{21}$$

where the tuple $k \in \mathcal{T}(i,u)$, if $|\mathcal{T}(i,u)| = 1$, else $k = k^*$, where $k^* \in \mathcal{T}(i,u)$ is the tuple of partition indices such that $\bar{\tau}(k_l^*) = \bar{\tau}(i) + 1$, $l = 1, \ldots, 2^m$.

Proof: Recall the nonlinear constraints [Eq. (18)] satisfied by $\underline{w}(i)$, $i = 1, \ldots, M$, that motivated the NLP formulation

$$\underline{w}(i) \ge \min_{x \in \mathcal{S}_i} \left\{ R_u(x) + \lambda \sum_{l=1}^{2^m} p_l \underline{w}(\bar{f}(x,u,\tilde{y}_l)) \right\}, \quad \forall\, u \tag{22}$$

which, given the definition of $\mathcal{T}(i,u)$, can be written in the following equivalent form: $\forall\, i = 1, \ldots, M$,

$$\underline{w}(i) \ge r_u(i) + \lambda \min_{k \in \mathcal{T}(i,u)} \sum_{l=1}^{2^m} p_l \underline{w}(k_l), \quad \forall\, u \tag{23}$$

where $r_u(i)$ is the reward associated with partition index i and, given the partitioning scheme, satisfies $R_u(x) = r_u(i), \forall\, x \in \mathcal{S}_i$. Given the structure in the perimeter patrol problem, Eq. (23) will collapse to a single linear inequality constraint for every partition index i and control u. Let us focus our attention on partition index $i \in \mathcal{P}_1$ and control action $u = 0$. For this choice, the cardinality of $\mathcal{T}(i,0)$ is $\bar{\tau}(i)$ (from Lemma 3). Indeed, $\exists \bar{x} \in \mathcal{S}_i$ such that the corresponding tuple of future partition indices $k^* = [\bar{f}(\bar{x},0,\tilde{y}_1), \ldots, \bar{f}(\bar{x},0,\tilde{y}_{2^m})]$ has the highest possible maximum delay; that is, $\bar{\tau}(k_l^*) = \bar{\tau}(i) + 1$, $l = 1, \ldots, 2^m$. Because $k_l^* \ge k_l$, $l = 1, \ldots, 2^m$, $\forall\, k \in \mathcal{T}(i,u)$, Lemma 2 gives us

$$\underline{w}(k_l^*) \le \underline{w}(k_l), \qquad l = 1, \ldots, 2^m, \quad \forall\, k \in \mathcal{T}(i,u)$$

Therefore, the nonlinear inequality corresponding to partition index $i \in \mathcal{P}_1$ and control $u = 0$ becomes

$$\underline{w}(i) \ge r_0(i) + \lambda \sum_{l=1}^{2^m} p_l \underline{w}(k_l^*) \tag{24}$$

If $u \ne 0$, then $|\mathcal{T}(i,u)| = 1$ as per Lemma 3. Therefore, there exists exactly one tuple $\underline{k}$ in $\mathcal{T}(i,u)$ and hence, the nonlinear constraint Eq. (23) reduces to the linear inequality:

$$\underline{w}(i) \ge r_u(i) + \lambda \sum_{l=1}^{2^m} p_l \underline{w}(\underline{k}_l) \tag{25}$$

Again, for partition indices j of Type 2, $|\mathcal{T}(j,u)| = 1$, $\forall\, u$ as per Lemma 3. Therefore, as before, the nonlinear inequality in Eq. (23) collapses to the linear inequality:

$$\underline{w}(j) \ge r_u(j) + \lambda \sum_{l=1}^{2^m} p_l \underline{w}(\underline{k}_l) \tag{26}$$

In summary, regardless of which partition one considers, the corresponding nonlinear constraint in Eq. (18) collapses to a linear constraint and hence, NLP for the perimeter patrol problem can be rewritten as follows:

$$\text{LBLP:} = \min\ \bar{c}^T w, \quad \text{subject to}$$

$$w(i) \ge r_u(i) + \lambda \sum_{l=1}^{2^m} p_l w(k_l), \quad \forall\, u, \qquad i = 1, \ldots, M \tag{27}$$

where the tuple $k \in \mathcal{T}(i,u)$ if $|\mathcal{T}(i,u)| = 1$, else $k = k^*$, where $k^* \in \mathcal{T}(i,u)$ is the (unique) tuple of partition indices such that $\bar{\tau}(k_l^*) = \bar{\tau}(i) + 1$, $l = 1, \ldots, 2^m$.

Corollary 2: The optimal solution w^* to LBLP is independent of the cost $\bar{c} > 0$ and is a lower bound to the optimal value function V^*; that is, $\forall\, i \in \{1, \ldots, M\}$, $w^*(i) \le \min_{x \in \mathcal{S}_i} V^*(x)$.

Proof: To arrive at the corollary, observe that LBLP is the exact LP [Eq. (12)] corresponding to a reduced order MDP defined over the space of partitions with reward r_u and transition probability between partitions i and j given by

$$\bar{P}_u(i,j) = \begin{cases} 0, & \text{if } j \neq k_l, \text{ for any } k \in \mathcal{T}(i,u),\ l = 1, \ldots, 2^m \\ \sum_{q \in \mathcal{C}_1} p_q, & \text{where } \mathcal{C}_1 = \{l | j = k_l, k \in \mathcal{T}(i,u)\}, \text{ if } |\mathcal{T}(i,u)| = 1 \\ \sum_{q \in \mathcal{C}_2} p_q, & \text{where } \mathcal{C}_2 = \{l | j = k_l^*, k^* \in \mathcal{T}(i,u), \\ & \bar{\tau}(k_l^*) = \bar{\tau}(i) + 1,\ l = 1, \ldots, 2^m\}, \quad \text{otherwise} \end{cases} \tag{28}$$

Therefore, Lemma 1 implies that the optimal solution w^* is independent of $\bar{c} > 0$ and also is dominated by every feasible solution, including $\underline{w}$. Hence, $w^*(i) \leq \underline{w}(i) = \min_{x \in \mathcal{S}_i} V^*(x) \leq V^*(y)$, $\forall\ y \in \mathcal{S}_i$, $i = 1, \ldots, M$. Therefore, for the perimeter patrol problem, the lower bound approximation to the optimal value function can be computed cheaply via the LBLP formulation. In the next section, the efficacy of the proposed method is demonstrated via simulation results.

VI. Simulation Results

Consider a perimeter with $N = 15$ nodes of which node numbers $\{0, 3, 7, 11\}$ are alert stations and a maximum allowed dwell of $D = 5$ orbits. The other parameters were chosen to be, weighing factor $\rho = .005$ and temporal discount factor $\lambda = 0.9$. From practical experience, the alert arrival rate is chosen to be $\alpha = 1/60$. This reflects a rather low arrival rate where two alerts occur on average in the time taken by the UAV to complete an uninterrupted patrol around the perimeter. The maximum delay time that is kept track of is set to be $\Gamma = 15$, and so the total number of states $|\mathcal{S}| = 2,048,000$. To show that the proposed approximate methodology is effective, the approximate value function is computed via LBLP [Eq. 27] and compared with the optimal value function. In addition, a greedy suboptimal policy corresponding to the approximate value function is compared with the optimal policy in terms of the two performance metrics: alert service delay and information gained upon loitering.

The states in the example problem are aggregated based on the reward function (see Sec. V.A for details). This results in $M = 8900$ partitions, which is considerably smaller than the original number of states $|\mathcal{S}|$. The LBLP formulation is solved and gives us the lower bound V_{low} to the optimal value function V^*, where $V_{\text{low}}(x) = w^*(i)$, $\forall\ x \in \mathcal{S}_i$, $i = 1, \ldots, M$. Because the size of the example problem is reasonably small, the optimal value function V^* is also computed and used for comparison with the approximation. Note that, for higher values of m and Γ, the problem essentially becomes intractable and one would not have access to the optimal value function. A representative sample of the approximation results is given by choosing all the states in partitions corresponding to alert status $\mathcal{A}_j = 1, \forall\ j \in \Omega$ (all stations have alerts) and maximum delay $\bar{\tau} = 2$. Figure 3 compares V^* with V_{low} for this subset of the state-space. The first 15 partitions shown in the x axis of Fig. 3; that is, partition indices $i = 1, \ldots, 15$, correspond to the clockwise states

$$\ell = i - 1, \qquad d = 0, \qquad \omega = 1,$$
$$\bar{\tau} = \max_{j \in \Omega} \tau_j = 2, \quad \text{and} \quad \mathcal{A}_j = 1, \quad \forall\ j \in \Omega \tag{29}$$

and the last 15 partitions shown in the x axis; that is, partition indices $i = 16, \ldots, 30$, correspond to the counterclockwise states:

$$\ell = i - N - 1, \qquad d = 0, \qquad \omega = -1,$$
$$\bar{\tau} = \max_{j \in \Omega} \tau_j = 2, \quad \text{and} \quad \mathcal{A}_j = 1, \quad \forall\ j \in \Omega \tag{30}$$

Recall that our objective is to obtain a provably good suboptimal policy and so, consider the policy that is greedy with respect to V_{low}: $\forall\ x \in \mathcal{S}$,

$$\pi_s(x) = \arg\max_u \left\{ R_u(x) + \lambda \sum_{l=1}^{2^m} p_l V_{\text{low}}(f(x,u,\tilde{y}_l)) \right\} \tag{31}$$

To assess the quality of the suboptimal policy, the expected discounted payoff V_{sub} that corresponds to the suboptimal policy π_s is also computed via

$$(I - \lambda P_{\pi_s}) V_{\text{sub}} = R_{\pi_s} \tag{32}$$

Because V_{sub} corresponds to a suboptimal policy, the following inequality holds:

$$V_{\text{sub}} \leq V^*$$

In Fig. 4, V_{sub} is compared with the optimal value function V^* for the clockwise states defined in Eq. (29). Finally, the performance of the suboptimal policy π_s is compared with that of the optimal strategy π^*

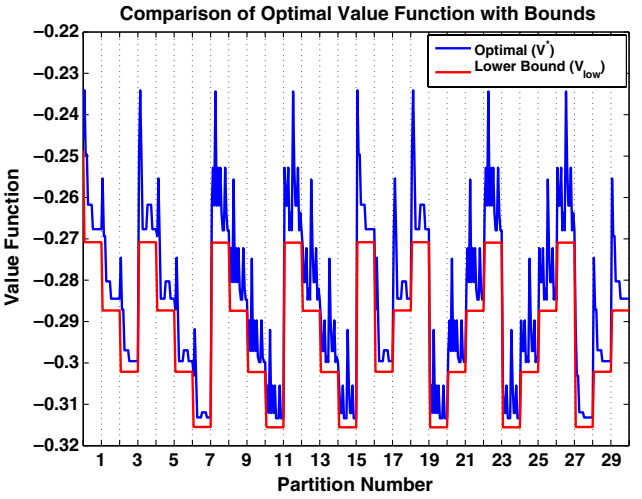

Fig. 3 Comparison of approximate value function with the optimal.

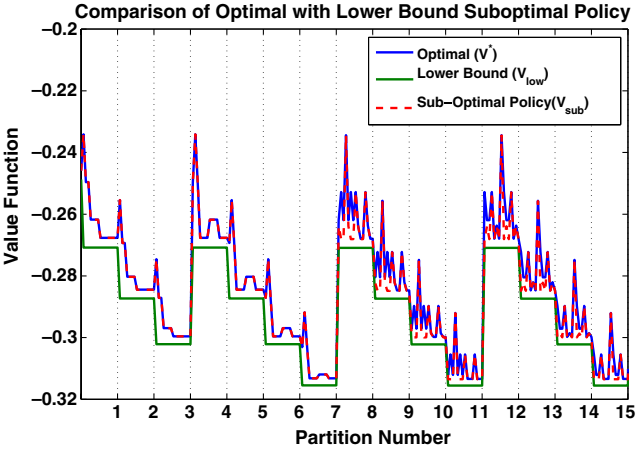

Fig. 4 Comparison of value function corresponding to suboptimal policy π_s with the optimal.

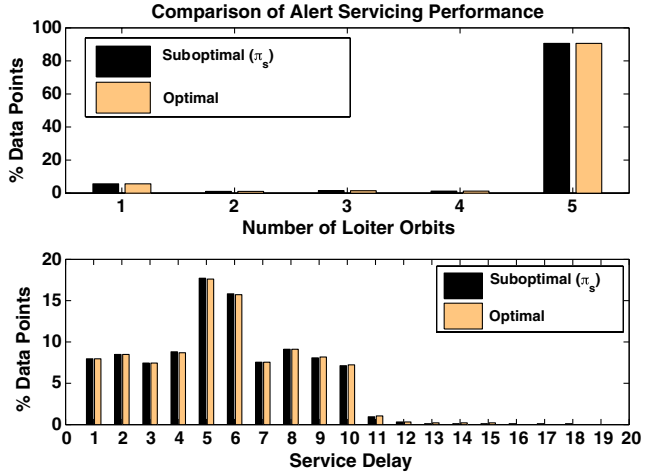

Fig. 5 Comparison of service delay and number of loiters between optimal and suboptimal policies.

Table 1 Comparison of alert servicing performance between optimal and suboptimal policies

Policy	Mean loiters	Mean delay	Worst delay
π^*	4.7	5.6	15
π_s	4.7	5.6	18

in terms of the two important metrics: service delay and information gain (measured via the dwell time). To collect the performance statistics, Monte-Carlo simulations with alerts at each station generated from independent Bernoulli processes with rate $\alpha = 1/60$ were done. All simulations had a run time of 60,000 time units. Both the optimal and suboptimal policies were tested against the same alert sequence. Figure 5 shows histogram plots for the service delay (top plot) and the dwell time (bottom plot) for all serviced alerts in the simulation run. The corresponding mean and worst-case service delays and the mean dwell time are also shown in Table 1. There is hardly any difference in terms of either metric between the optimal and the suboptimal policies. This substantiates the claim that the aggregation approach gives us a suboptimal policy that performs almost as well as the optimal policy itself. This is to be expected given that the value functions corresponding to the optimal and suboptimal policies are close to each other (see Fig. 4). Because the false alarm rate α is fairly low, roughly 90% of the alerts were cleared within 10 time steps (see bottom plot of Fig. 5). Also, from the top plot of Fig. 5, it can be seen that maximum information was gained five loiters completed) on almost 90% of the serviced alerts.

VII. Conclusions

A state aggregation based restricted linear programming (LP) to construct suboptimal policies for the perimeter patrol stochastic optimal control problem has been provided. As a general result, a novel disjunctive LP that can be used to compute a nontrivial lower bound to the optimal value function has been provided. In particular, for the perimeter patrol problem, it is shown that the lower bound formulation simplifies to an exact LP corresponding to a lower dimensional Markov chain defined over the space of partitions. This comes about because of the monotonicity of the optimal value function, which is a product of the structure in the problem. Simulation results show that the performance of the suboptimal policy, obtained via the lower bound approximate value function, is comparable to that of the optimal policy.

Appendix: Monotonicity of the Optimal Value Function

Let $x(t; x_0, \boldsymbol{u}_t, \boldsymbol{y}_t)$ denote the state at time $t > 0$ if the initial state at $t = 0$ is x_0 and the sequence of inputs $\boldsymbol{u}_t = \{u(0), u(1), \ldots, u(t-1)\}$ and disturbances $\boldsymbol{y}_t = \{y(0), y(1), \ldots, y(t-1)\}$.

Claim 1: If $x_1 \geq x_2$, then for the same sequence of inputs $\boldsymbol{u}_t$ and disturbances $\boldsymbol{y}_t$, $x(t; x_1, \boldsymbol{u}_t, \boldsymbol{y}_t) \geq x(t; x_2, \boldsymbol{u}_t, \boldsymbol{y}_t)$ for every $t > 0$.

Proof: The proof is by induction. Clearly, at $t = 0$, $x_1 \geq x_2$. By the semigroup property of state transitions, it is sufficient to show that the result holds for $t = 1$. Let the state x of the patrol system be of two types. If the following holds,

$$\ell_x \in \Omega, \qquad d_x = 0, \qquad \mathcal{A}_{\ell_x,x} = 1, \quad \text{and} \quad \mathcal{A}_{j,x} = 1, \quad \text{for some } j \in \Omega, \qquad j \neq \ell_x \tag{A1}$$

that is, the UAV is at a station with an alert, the dwell time is zero, and also there is an alert at some other station, then the state x is of Type 1. Else, it is of Type 2. Note that, if $x_1 \geq x_2$, then the states x_1 and x_2 are necessarily of the same type. The key property that will be used in proving Claim 1 is the following: Service delay at a station either remains at zero (if no new alert has occurred there) or it goes up by one (if there is an unserviced alert there) or it is reset to zero (if a UAV decides to loiter there).

If x_1 and x_2 are of Type 1 and the UAV chooses to loiter; that is, $u(0) = 0$, it is clear that neither the location nor the dwell will differ at $t = 1$. Furthermore, the delays at $t = 1$ associated with the stations corresponding to initial state x_1 will be no less than the delays associated with stations corresponding to initial state x_2 because $x_1 \geq x_2$. If $z_1 = x(1; x_1, 0, y(0))$ and $z_2 = x(1; x_2, 0, y(0))$, then $\ell_{z_1} = \ell_{z_2}$, $d_{z_1} = d_{z_2}$, $\omega_{z_1} = \omega_{z_2}$, and $\tau_{j,z_1} \geq \tau_{j,z_2}$, $\forall\, j \in \Omega$ for all possible $y(0)$, and so $z_1 \geq z_2$. The same relationship holds for other possible control choices, $u(0) \neq 0$, as well. By a similar argument, one can show that $x(1; x_1, u(0), y(0)) \geq x(1, x_2, u(0), y(0))$ holds regardless of the control choice, even if the states x_1, x_2 are of Type 2.

Lemma 2: If $\mathcal{S}_i \geq \mathcal{S}_j$, $\min_{x \in \mathcal{S}_i} V^*(x) \leq \min_{z \in \mathcal{S}_j} V^*(z)$.

Proof: Let π^* be the optimal policy; accordingly, $\pi^*(x)$ is fixed for every $x \in \mathcal{S}$. Then, for every $t > 0$, $x(t; x_1, \boldsymbol{u}_t^*, \boldsymbol{y}_t)$ can be determined for some sequence of disturbances $\boldsymbol{y}_t$ where the optimal input sequence $\boldsymbol{u}_t^* = \{u^*(0), \ldots, u^*(t-1)\}$ (starting with x_1) can be recursively obtained as

$$u^*(t) = \pi^*(x(t-1; x_1, \boldsymbol{u}_{t-1}^*, \boldsymbol{y}_{t-1})) \tag{A2}$$

with the initialization $u^*(0) = \pi^*(x_1)$. For the preceding $\boldsymbol{u}^*$ and $\boldsymbol{y}$, one can then determine the evolution of the states corresponding to initial state x_2. Because $x(t; x_1, \boldsymbol{u}_t^*, \boldsymbol{y}_t) \geq x(t; x_2, \boldsymbol{u}_t^*, \boldsymbol{y}_t)$ by Lemma 1, the reward $R_{u^*}(x(t; x_1, \boldsymbol{u}_t^*, \boldsymbol{y}_t)) \leq R_{u^*}(x(t; x_2, \boldsymbol{u}_t^*, \boldsymbol{y}_t))$ for every $t \geq 0$ (because the one-step reward is based only on the maximum delay, dwell time, and control input, the inequality follows). Because the preceding holds for any given disturbance sequence, the expected discounted payoff associated with the state starting from x_1; that is, $V^*(x_1)$, is no more than the expected discounted payoff associated with the state starting from x_2, which is denoted by $V_{\boldsymbol{u}^*}(x_2)$. As a result, $V^*(x_1) \leq V_{\boldsymbol{u}^*}(x_2) \leq V^*(x_2)$. The second part of the inequality holds because $\boldsymbol{u}_t^*$ as defined in Eq. (A2) is a suboptimal control policy for the state evolution starting from x_2 and hence, the expected discounted payoff associated with that policy is necessarily dominated by the optimal value function starting from x_2. Let partitions $\mathcal{S}_i$ and $\mathcal{S}_j$ be such that $\mathcal{S}_i \geq \mathcal{S}_j$. Let $\bar{z} = \arg\min_{z \in \mathcal{S}_j} V^*(z)$. Because $\mathcal{S}_i \geq \mathcal{S}_j$, $\exists \bar{x} \in \mathcal{S}_i$ such that $\bar{x} \geq \bar{z}$. It has been shown that, for this case, $V^*(\bar{x}) \leq V^*(\bar{z}) = \min_{z \in \mathcal{S}_j} V^*(z) \Rightarrow \min_{x \in \mathcal{S}_i} V^*(x) \leq \min_{z \in \mathcal{S}_j} V^*(z)$.

Acknowledgments

This work was partly supported by the U.S. Air Force Research Laboratory Summer Faculty Program and Air Force Office of Scientific Research award number FA9550-10-1-0392.

References

[1] Takagi, H., "Queueing Analysis of Polling Models," *ACM Computing Surveys*, Vol. 20, No. 1, March 1988, pp. 5–28. doi:10.1145/62058.62059

[2] Levy, H., and Sidi, M., "Polling Systems: Applications, Modeling and Optimization," *IEEE Transactions on Communications*, Vol. 38, No. 10, Oct. 1990, pp. 1750–1760. doi:10.1109/26.61446

[3] Sennott, L. I., *Stochastic Dynamic Programming and the Control of Queueing Systems*, Wiley-Interscience, New York, 1999, pp. 1–14.

[4] Kleinrock, L., *Queueing Systems, Vol I: Theory*, Wiley, New York, 1975, pp. 3–9.

[5] Pillac, V., Gendreau, M., Gueret, C., and Medaglia, A. L., "A Review of Dynamic Vehicle Routing Problems," *European Journal of Operational Research*, Vol. 225, No. 1, 2013, pp. 1–11. doi:10.1016/j.ejor.2012.08.015

[6] Krishnamoorthy, K., Pachter, M., Chandler, P., and Darbha, S., "Optimization of Perimeter Patrol Operations Using Unmanned Aerial Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 2, 2012, pp. 434–441. doi:10.2514/1.54720

[7] Chevaleyre, Y., "Theoretical Analysis of the Multi-Agent Patrolling Problem," *Proceedings of the IEEE International Conference on Intelligent Agent Technology*, Sept. 2004, pp. 302–308.

[8] Cassandras, C. G., Ding, X. C., and Lin, X., "An Optimal Control Approach for the Persistent Monitoring Problem," *Proceedings of the IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Dec. 2011, pp. 2907–2912.

[9] Pasqualetti, F., Durham, J. W., and Bullo, F., "Cooperative Patrolling via Weighted Tours: Performance Analysis and Distributed Algorithms," *IEEE Transactions on Robotics*, Vol. 28, No. 5, 2012, pp. 1181–1188. doi:10.1109/TRO.2012.2201293

[10] Park, M., Darbha, S., Krishnamoorthy, K., Khargonekar, P. P., Pachter, M., and Chandler, P., "Sub-Optimal Stationary Policies for a Class of Stochastic Optimization Problems Arising in Robotic Surveillance Applications," *Proceedings of the 5th Annual Dynamic Systems and Control Conference*, ASME Paper DSCC2012-8610, Oct. 2012.

[11] Manne, A. S., "Linear Programming and Sequential Decisions," *Management Science*, Vol. 6, No. 3, 1960, pp. 259–267. doi:10.1287/mnsc.6.3.259

[12] d'Epenoux, F., "A Probabilistic Production and Inventory Problem," *Management Science*, Vol. 10, No. 1, 1963, pp. 98–108. doi:10.1287/mnsc.10.1.98

[13] Denardo, E. V., "On Linear Programming in a Markov Decision Problem," *Management Science*, Vol. 16, No. 5, 1970, pp. 282–288. doi:10.1287/mnsc.16.5.281

[14] Hordijk, A., and Kallenberg, L. C. M., "Linear Programming and Markov Decision Chains," *Management Science*, Vol. 25, No. 4, 1979, pp. 352–362. doi:10.1287/mnsc.25.4.352

[15] Mendelssohn, R., "Improved Bounds for Aggregated Linear Programs," *Operations Research*, Vol. 28, No. 6, 1980, pp. 1450–1453. doi:10.1287/opre.28.6.1450

[16] Schweitzer, P. J., and Seidmann, A., "Generalized Polynomial Approximations in Markovian Decision Processes," *Journal of Mathematical Analysis and Applications*, Vol. 110, No. 2, 1985, pp. 568–582. doi:10.1016/0022-247X(85)90317-8

[17]] Trick, M., and Zin, S., "Spline Approximation to Value Functions: A Linear Programming Approach," *Macroeconomic Dynamics*, Vol. 1, No. 1, 1997, pp. 255–277. doi:10.1017/S1365100597002095

[18] De Farias, D. P., and Van Roy, B., "The Linear Programming Approach to Approximate Dynamic Programming," *Operations Research*, Vol. 51, No. 6, 2003, pp. 850–865. doi:10.1287/opre.2003.51.issue-6

[19] Axsäter, S., "State Aggregation in Dynamic Programming: An Application to Scheduling of Independent Jobs on Parallel Processors," *Operations Research Letters*, Vol. 2, No. 4, 1983, pp. 171–176. doi:10.1016/0167-6377(83)90050-0

[20] Bean, J. C., Birge, J. R., and Smith, R. L., "Aggregation in Dynamic Programming," *Operations Research*, Vol. 35, No. 2, 1987, pp. 215–220. doi:10.1287/opre.35.2.215

[21] Mendelssohn, R., "An Iterative Aggregation Procedure for Markov Decision Processes," *Operations Research*, Vol. 30, No. 1, 1982, pp. 62–73. doi:10.1287/opre.30.1.62

[22] Van Roy, B., "Performance Loss Bounds for Approximate Value Iteration with State Aggregation," *Mathematics of Operations Research*, Vol. 31, No. 2, 2006, pp. 234–244. doi:10.1287/moor.2006.31.issue-2

[23] Grötschel, M., and Holland, O., "Solution of Large-Scale Symmetric Travelling Salesman Problems," *Mathematical Programming*, Vol. 51, Nos. 1–3, 1991, pp. 141–202. doi:10.1007/BF01586932

[24] Schuurmans, D., and Patrascu, R., *Direct Calue-Approximation for Factored MDPs*, Vol. 14, Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2001, pp. 1579–1586.

[25] Porteus, E. L., "Bounds and Transformations for Discounted Finite Markov Decision Chains," *Operations Research*, Vol. 23, No. 4, 1975, pp. 761–784. doi:10.1287/opre.23.4.761

[26] Krishnamoorthy, K., Pachter, M., Darbha, S., and Chandler, P., "Approximate Dynamic Programming with State Aggregation Applied to UAV Perimeter Patrol," *International Journal of Robust and Nonlinear Control*, Vol. 21, No. 12, 2011, pp. 1396–1409. doi:10.1002/rnc.1686

[27] MacQueen, J. B., "A Modified Dynamic Programming Method for Markovian Decision Problems," *Journal of Mathematical Analysis and Applications*, Vol. 14, No. 1, 1966, pp. 38–43. doi:10.1016/0022-247X(66)90060-6

[28] Krishnamoorthy, K., Park, M., Pachter, M., Chandler, P., and Darbha, S., "Bounding Procedure for Stochastic Dynamic Programs with Application to the Perimeter Patrol Problem," *Proceedings of the American Control Conference*, Montreal, QC, June 2012, pp. 5874–5880.

[29] Park, M., Krishnamoorthy, K., Pachter, M., Darbha, S., and Chandler, P., "State Partitioning Based Linear Program for Stochastic Dynamic Programs: An Invariance Property," *Operations Research Letters*, Vol. 40, No. 6, Nov. 2012, pp. 487–491. doi:10.1016/j.orl.2012.08.006