On Solving MDPs With Large State Space: Exploitation of Policy Structures and Spectral Properties

Libin Liu, Arpan Chattopadhyay, and Urbashi Mitra, *Fellow, IEEE*

***Abstract*— In this paper, a point-to-point network transmission control problem is formulated as a Markov decision process (MDP). Classical dynamic programming techniques such as value iteration, policy iteration, and linear programming can be employed to solve the optimization problem, but they suffer from high-computational complexity in networks with large state space. To achieve complexity reduction, the structure of the optimal policy can be exploited and incorporated into standard algorithms. In addition, function approximation can also be applied, where the value function is approximated by the linear combination of some basis vectors in a lower dimensional subspace. The main challenge for function approximation lies in the absence of general guidelines for subspace construction. In this paper, a proper subspace for projection is first generated based on system information, and more general construction methods are proposed using tools from graph signal processing (GSP). Graph symmetrization methods are also used to tackle the directed nature of the probability transition graph so that the well-developed GSP theory for undirected graphs can be employed. The numerical results for a typical wireless system show that standard algorithms with structural information incorporated can achieve 50% complexity reduction without performance loss. The subspace generated from the system can achieve zero policy error with faster runtime, and the GSP approach can also provide a proper subspace for perfect reconstruction of the optimal policy. It is also shown that how the proposed method can be applied to other MDP problems.**

***Index Terms*— Wireless networks, Markov decision process (MDP), graph signal processing (GSP).**

I. Introduction

MARKOV Decision Processes (MDPs) are useful in modeling a wide range of network control and optimization problems, where the system evolves stochastically as a function of current system state and control input. The application of MDPs can be found in wireless networks [1], sensor networks [2]–[4], inventory problems [5], resource allocation [6], agriculture and population [7], *etc*.

The theory of MDPs is well established [8], [9]. Classical algorithms such as value iteration, policy iteration and linear programming serve as useful tools to solve the optimization problem. However, in practice, large scale networks are ubiquitous (*e.g.* wireless networks, sensor networks, biological networks, *etc*). The size of such networks usually scale polynomially or exponentially with the number of state variables. This is generally referred to as *the curse of dimensionality*, rendering the standard algorithms computationally prohibitive.

There are two main methodologies tailored for MDPs to tackle the curse of dimensionality. One is through *value function approximation*, which seeks to find compact representations of the value function in a lower dimensional subspace [8]–[10]. Finding a proper subspace can be non-trivial, our early work [11] presented several approaches and comparisons of subspace selection. Work reported in [12]–[14] showed that diffusion wavelets [15] could be useful for generating the subspace and solving the MDP problems efficiently, although the Probability Transition Matrix (PTM) is constrained to be symmetric. The challenges with subspace approaches are finding the right basis and ensuring the convergence of the associated iterative methods [9].

The other dimension reduction approach is realized through *model reduction*. This approach seeks compact representations of the system (Markov chain) rather than the value function via state aggregation and disaggregation. Model reduction can be achieved with different techniques and metrics. For example, the notion of *stochastic bisimulation* is proposed in [16] wherein two states are aggregated if they have the same cost functions and transition probabilities. In [17], the Kullback-Leibler (K-L) divergence is used as a metric to minimize the distance between the original Markov chain and the one after aggregation and disaggregation. The Kron-reduction technique can also be employed given the identification of an independent set [18]. Recently, stochastic factorization has been proposed in [19], where a low rank factorization of the PTM is performed to create a lower dimensional MDP by swapping the position of the two factored stochastic matrices. The disadvantages of these methods are: complexity (it is NP hard to find the right independent set for Kron-reduction), limitation to specific kinds of Markov Chains (K-L aggregation and

Manuscript received July 19, 2018; revised December 2, 2018 and February 1, 2019; accepted February 3, 2019. Date of publication February 15, 2019; date of current version June 14, 2019. This work was funded by the following grants: ONR N00014-15-1-2550, NSF CNS-1213128, NSF CCF-1410009, AFOSR FA9550-12-1-0215, and NSF CPS-1446901. It was also supported by the Royal Academy of Engineering, the Fulbright Foundation and the Leverhulme Trust. The associate editor coordinating the review of this paper and approving it for publication was N. Pappas. *(Corresponding author: Libin Liu.)*

L. Liu and U. Mitra are with the Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: libinliu@usc.edu; ubli@usc.edu).

A. Chattopadhyay is with the Electrical Engineering Department, IIT Delhi, New Delhi 110016, India (e-mail: arpanc@ee.iitd.ac.in).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCOMM.2019.2899620

stochastic factorization), and idealized conditions (stochastic bisimulation).

In addition to the approximation methods mentioned above, structural information can also be incorporated into the standard algorithms to achieve complexity reduction. For example, periodicity of the system is explored in [5] to reduce computation; as a generalization, graphs that are M-block cyclic [20] can potentially allow complexity reduction in analysis. Moreover, a threshold structure for optimal policies is a common phenomenon in many MDP applications in wireless networks [21]–[24]. Thresholded policies have thresholding states that characterize the optimal control decisions. The generalization of these types of networks is still challenging. In [25], a sufficient condition for the value function and optimal strategy to be even and quasi-convex is derived, but this work mainly focuses on MDPs where the state space is even (symmetric on the real line); although [25] also defines a folding operation to extend the analysis to the $\mathbb{R}^+$. However, the conditions are still too strong for typical wireless networks. For particular wireless networks exhibiting such thresholded policy structure, we will show that incorporating the policy structure in policy optimization can strongly reduce complexity.

In this paper, we consider a *value function approximation* method with Graph Signal Processing techniques employed. Graph Signal Processing (GSP) is a theory for analyzing signals defined on graphs. Early work [26]–[30] provided insightful extension of frequency and Fourier transforms from the classical signal domain to the graph domain. In our MDP problem, although the probability transition matrix can be viewed as a transition graph and the value function of each state can be viewed as graph signal, directly applying GSP theory appears to be challenging. Due to the nature of the MDP problem, our transition graph is a *directed* graph; GSP theory tailored for directed graphs [28]–[30] requires the Jordan decomposition of the PTM, which is much more complex than the Eigenvalue Decomposition (EVD), a technique that is usually used for undirected graphs. Luckily, various kinds of symmetrization techniques have been proposed [31]. Here, we adopt several methods and use classical GSP theory to obtain a good subspace for value function approximation.

Our main contributions in this paper are:

- The policy structure of a MDP for a wireless network example is derived.
- A modified policy iteration algorithm exploiting the policy structure is proposed. Numerical results show that it achieves reasonable complexity reduction with no performance loss.
- *Optimal subspace construction criteria for reduced dimension MDP is derived, and a good subspace is generated based on the system model that achieves zero policy error with faster runtime.*
- Various methods are proposed to generate the subspace using GSP techniques for value function approximation. One particular method based on the graph symmetrization technique in [31] achieves zero policy error.
- Finally, we show the application of our methods to another MDP admitting a thresholded policy [32].

The rest of the paper is organized as follows: Section II provides background on MDPs and GSP. In Section III, the wireless system is elaborated upon with the structure of the optimal policy derived, and a modified policy iteration algorithm is proposed. The projection method for MDPs is presented and the optimal criteria for subspace construction is derived in Section IV. Section V provides more detail on general subspace construction methods using GSP. Numerical validation is provided in Section VI and Section VII concludes the paper. Finally the proofs are provided in the appendices.

II. Preliminaries

In this section, we provide key background on MDPs and GSP. For notational clarity, column vectors are in bold lower case (*e.g.* $\mathbf{x}$); matrices are in bold upper case (*e.g.* $\mathbf{A}$); sets are in calligraphic font (*e.g.* $\mathcal{S}$); and scalars are non-bold (*e.g.* α, a).

A. Markov Decision Processes and Policy Iteration

Markov Decision Processes provide mathematical tools for modeling systems that involve decision making. Typically, a MDP is a 5-tuple $\{\mathcal{S}, \mathcal{U}, \mathbf{P}, c, \alpha\}$, where $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ denotes a finite state space, and $\mathcal{U} = \{u_1, u_2, \ldots, u_k\}$ denotes the finite action space. The probability transition matrix is given by $\mathbf{P}$ and we denote $s(t)$ and $u(t)$ as the state and action we take at time slot t respectively, where $t = \{1, 2, 3, \ldots\}$. The transition probability of going to state s', given the current state s and current action u, is given by:

$$p(s, u, s') = P\left(s(t+1) = s' | s(t) = s, \quad u(t) = u\right), \quad (1)$$

$\forall s, s' \in \mathcal{S},\ u \in \mathcal{U}$. For each transition, there is an associated instantaneous cost $\varphi_(s, u, s')$. The average cost (single stage cost) defined for each state s, given action u, is:

$$c(s, u) = \sum_{s' \in \mathcal{S}} \varphi(s, u, s') p(s, u, s') \quad \forall s, u. \quad (2)$$

In different networking settings, depending on the application, the cost function $c(s, u)$ can be used to describe performance metrics such as throughput, delay, failure probability, *etc*. The core problem of MDP optimization can be formulated as:

$$\begin{aligned} \mathbf{v}^*(s) &= \min_{\mu} \mathbf{v}_\mu(s) \\ &= \min_{\mu} \mathbf{E}_\mu \left\{ \sum_{t=0}^{\infty} \alpha^t c(s(t), \mu(s(t))) \middle| s(0) = s \right\}, \quad (3) \end{aligned}$$

where $\alpha \in (0, 1)$ is the discount factor, and $\mathbf{v}^*(s)$ is called the *value function* [8] measuring the expected sum of discounted costs over an infinite horizon starting from state s. A policy, $\mu : \mathcal{S} \to \mathcal{U}$, is a mapping from the state space to the action space, specifying the decision rule for each state. If under policy μ, the action taken at state s is u, then this action is represented as: $\mu(s) = u$. Denote μ_t a *stationary* policy if $\mu_t = \mu$ for all t; it is also called a *deterministic* policy if $P(\mu(s) = u) = 1, u \in \mathcal{U}$. Without loss of generality [8], we focus on stationary and deterministic policies that solve the optimization problem (3).

Policy iteration [8] is a classical algorithm to compute the solution to the optimization problem. It starts with any arbitrary policy μ^0 and iteratively generates a sequence of policies $\{\mu^{(k)}\}_{k\geq 1}$. Denote $\mathbf{P}_{\mu^{(k)}}$ and $\mathbf{c}_{\mu^{(k)}}$ as the PTM and cost function under policy $\mu^{(k)}$ respectively; and the value function under policy $\mu^{(k)}$ is denoted by $\mathbf{v}_{\mu^{(k)}}$. Then each iteration consists of two steps:

- *Policy evaluation*
  Under the current policy, $\mu^{(k)}$, the system evolves as a Markov chain. The value function is evaluated by solving the Bellman fixed point equation [8]:
  $$\mathbf{v}_{\mu^{(k)}} = \mathbf{c}_{\mu^{(k)}} + \alpha \mathbf{P}_{\mu^{(k)}} \mathbf{v}_{\mu^{(k)}}, \qquad (4)$$
  where $\mathbf{c}_{\mu^{(k)}}$ and $\mathbf{P}_{\mu^{(k)}}$ are the average cost and probability transition matrix under policy $\mu^{(k)}$.
- *Policy improvement*
  After obtaining $\mathbf{v}_{\mu^{(k)}}$, the policy is updated by the greedy search of the one step look-ahead of the current value function:

$$\mu^{(k+1)}(s) = \arg\min_{u\in\mathcal{U}} \left\{ c(s,u) + \alpha \sum_{s'\in\mathcal{S}} p(s,u,s')\mathbf{v}_{\mu^{(k)}}(s') \right\}. \qquad (5)$$

We keep iterating between steps (4) and (5) until two successive policies are the same, and we have the optimal policy and value function. It can be shown that after each iteration, the policy is improving the value function. Due to the finite nature of the state and action spaces, the policy iteration is guaranteed to converge to the optimal solution in finite number of iterations [8]. And the optimal solution satisfies the optimality equation:

$$\mathbf{v}^*(s) = \min_{u\in\mathcal{U}} \left\{ c(s,u) + \alpha \sum_{s'\in\mathcal{S}} p(s,u,s')\mathbf{v}^*(s') \right\}. \qquad (6)$$

B. Graph Signal Processing

Graphs provide an efficient representation tool for data in many domains. Many networks such as wireless networks, social networks and biological networks, have this kind of structure with data located on vertices and links representing their relationships (connectivity, similarity, *etc*). The graph can be either the physical graph of the system itself (*e.g.* a transportation network or social network) or the logical graph induced by the network protocol (*e.g.* Markov chains, Finite State Machines). When the vertices of the graph are appropriately labeled, we can form a column vector containing the data on the vertices, this vector is termed the *graph signal*.

The analysis of a graph signal starts with the graph structure. Denote $\mathbf{x}$ as the graph signal and $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ as the corresponding graph, where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set. The relationship between nodes is represented by the adjacency matrix $\mathbf{W}$. If the graph is *undirected* ($\mathbf{W}$ is symmetric), then the graph Laplacian matrix [33] is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is the degree matrix, a diagonal matrix with $\mathbf{D}_{i,i} = \sum_{j\in\mathcal{V}} \mathbf{W}_{i,j}$, where $\mathbf{W}_{i,j}$ is the edge weight between node i and j. It is easy to see that $\mathbf{L}$ is a positive semidefinite (PSD) matrix. The spectral representation of the original graph signal is given by the eigenvalue decomposition of the matrix $\mathbf{L}$:

$$\mathbf{L} = \mathbf{B}\mathbf{\Lambda}\mathbf{B}^T = \sum_{i=1}^{|\mathcal{V}|} \lambda_i \mathbf{b}_i \mathbf{b}_i^T, \quad \lambda_i \geq 0 \quad \forall i, \qquad (7)$$

where $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{|\mathcal{V}|}]$ form an orthogonal eigenspace with projection matrix $\mathbf{b}_i\mathbf{b}_i^T$. Clearly, every graph signal defined on graph $\mathcal{G}$ can be decomposed as a linear combination of the eigenvectors $\mathbf{b}_i$.

Similar to classical signal processing, the total variation (*i.e.* graph frequency) [26] of the basis function $\mathbf{b}_i$ is defined as a quadratic sum:

$$\begin{aligned} \text{TV}(\mathbf{b}_i) &= \sum_{m,n} \mathbf{W}_{m,n}(\mathbf{b}_{i,m} - \mathbf{b}_{i,n})^2 \\ &= \mathbf{b}_i^T \mathbf{L} \mathbf{b}_i = \lambda_i. \end{aligned} \qquad (8)$$

Therefore, $\sigma(L) = \{0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{|\mathcal{V}|}\}$ represents the graph frequencies from low to high. It has been shown that the eigenvalues and eigenvectors of the Laplacian matrix $\mathbf{L}$ provide a harmonic analysis of graph signals [33], hence the *Graph Fourier Transform* is defined as the projection of the graph signal on the eigenspace of $\mathbf{L}$: $\tilde{\mathbf{x}} = \mathbf{B}^T\mathbf{x}$, where $\tilde{\mathbf{x}}$ is the vector of graph frequency coefficients. And the inverse transform is given by: $\mathbf{x} = \mathbf{B}\tilde{\mathbf{x}}$.

For directed graphs, due to the asymmetry induced by directivity of the adjacency matrix, the graph Laplacian concept can not be easily extended to directed graphs. While the graph Laplacian can be defined for directed graphs [34]; it is restricted to probability transition graphs (Markov chains) and the computation involves the calculation of the stationary distribution, which makes the computation more complicated. There are also other endeavors in analyzing graph signals defined on directed graphs, the authors in [28]–[30] directly analyzed the adjacency matrix and showed a nice frequency interpretation using Jordan decompositions, where the frequency is measured by the distance of between each eigenvalue and the point (1,0) on the complex plane. However, such methods also raise important issues requiring further attention. First, the generalized eigenvectors form a bi-orthogonal basis so that Parseval's identity does not hold. In addition, the total variation introduced in [30] does not guarantee that a constant graph signal has zero total variation. Furthermore, numerical complexity and instability can be an issue even for moderate matrix size and usually complex field analysis is required. Without a proper mapping from the complex field to real field, it is difficult to achieve good performance by approximating the graph signal in this manner.

III. Wireless Network Model and Policy Structure

The MDP model for a wireless network can vary depending on different applications and objective functions. For example, we can focus on the throughput of a network, whose state can be described by buffer length and retransmission index [1], [35]; or optimal transmission strategy in energy harvesting network, where the state of the network can be characterized by the energy storage in each device [36]. In this section, a wireless transmission example is examined; although it has a simple physical network structure, the probability transition graph induced by this MDP can have a large state space. We will demonstrate its thresholded policy structure and modify the policy iteration algorithm by incorporating such information. For the general case, a network control problem for a large network leads to an MDP with a large transition graph. Hence we consider an example of a MDP with large state space. The structure of the network enables us to solve the MDP theoretically and compare it with our proposed algorithms.

Consider a system of only one transmitter and one receiver. Time is discretized into slots with equal duration. In each time slot, the transmitter can decide whether to send a packet to the receiver or not. Packet arrival at the transmitter can be characterized by a Bernoulli proces with arrival probability p (this is not an uncommon assumption for queuing systems. Poisson arrivals can also be considered and our numerical results yielded similar performance behavior, see Sec. VI). Incoming packets will be stored in a buffer of capacity Q, and full occupancy of the buffer will lead to packet drop when there is a new packet arrival. Packets are transmitted through a channel between the transmitter and the receiver. The channel can be modeled as having path-loss, shadowing and fading [37]. The received power of a packet at the receiver is given by:

$$P_{rcv} = P_T C_0 l^{-\eta} h, \tag{9}$$

where P_T is the transmitting power at the transmitter, C_0 is a constant (path-loss at reference distance l), η is the path loss exponent, and the distance between transmitter and receiver is denoted by l. The Rayleigh fading gain is denoted by h, which is exponentially distributed with mean 1 and i.i.d. over time slots. We assume that the length of each time slot is greater than the channel coherence time thus the i.i.d assumption can hold. Under the assumption of flat channel in each time slot, the channel state h can be obtained via *channel estimation*, see *e.g.* [37]. It can be achieved by sending pilot signals at the beginning of each time slot and use the estimate as the reference channel condition. This channel estimation strategy can be found in multiple papers and books [38]–[41].

For the successful transmission of a packet, we assume that there exists a threshold P_{th} at the receiver. Thus, for any given channel state, there is also a power threshold at the transmitter to meet the successful transmission requirement. Therefore, given $P_{rcv} = P_{th}$, P_T is given by:

$$P_T(h) = \frac{P_{th}}{C_0 l^{-\eta} h}. \tag{10}$$

Notice that the channel state is continuous, discretization can be employed to simplify the analysis. The Gilbert-Elliot channel model [42], [43] is widely used for describing the burst error in transmission channels, where the evolution of the channel can be modeled as a Markov chain with two states (good and bad). In our model, the channel state is represented by the Rayleigh fading gain h, here we adopt the idea of discretization and partition the range of h into disjoint intervals with their midpoints representing the particular range of channel quality. Thus the probability density function (PDF) is changed to the probability mass function (PMF):

$$P(h = H_i) = \int_{a_i}^{b_i} \mathrm{e}^{-x} \mathrm{d}x, \quad \bigcup_i [a_i, b_i) = [0, +\infty), \tag{11}$$

where $H_i \in [a_i, b_i)$ is a particular number that represents the channel state within that interval.

The state of the system thus can be described by the pair (q, h), where $q \in \{0, 1, 2, \ldots, Q\}$ indicates the number of packets in the buffer and $h \in \{1, 2, \ldots, H\}$ represents the indices of the channel state. We consider a binary action space $\mathcal{U} = \{\text{transmit, idle}\} \triangleq \{1, 0\}$ for the transmitter. We also assume an i.i.d. channel and the channel coherence time equals the length of time slot. The transition probabilities are given by Equation (12), shown at the bottom of this page, and the Markov chain representation graph is shown in Fig. 1.

We consider a discounted cost infinite horizon problem (3) and the single-stage cost function in time slot t is defined as:

$$\begin{aligned} c(s(t), u(t)) = &\mathbf{1}\{\text{packet drop at time } t\} \\ &+ \beta P_T(s(t)) \cdot \mathbf{1}\{u(t) = \text{transmit}\}, \end{aligned} \tag{13}$$

where $\mathbf{1}(\cdot)$ is the indicator function, it is defined as follows:

$$\mathbf{1}(x) = \begin{cases} 1, & \text{if } x \text{ is true} \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

And β is the weighting factor measuring the emphasis on the transmitting power in the single stage cost.

$$(q_{t+1}, h_{t+1}) = \begin{cases} (0, h_{t+1}) & \text{w.p. } (1-p)P(h = h_{t+1}), \text{ if } q_t = 0 \\ (1, h_{t+1}) & \text{w.p. } pP(h = h_{t+1}), \text{ if } q_t = 0 \\ (q_t, h_{t+1}) & \text{w.p. } (1-p)P(h = h_{t+1}), \text{ if } 0 < q_t < Q \text{ and } U_t = 0 \\ (q_t + 1, h_{t+1}) & \text{w.p. } pP(h = h_{t+1}), \text{ if } 0 < q_t < Q \text{ and } U_t = 0 \\ (q_t, h_{t+1}) & \text{w.p. } pP(h = h_{t+1}), \text{ if } 0 < q_t < Q \text{ and } U_t = 1 \\ (q_t - 1, h_{t+1}) & \text{w.p. } (1-p)P(h = h_{t+1}), \text{ if } 0 < q_t < Q \text{ and } U_t = 1 \\ (q_t, h_{t+1}) & \text{w.p. } 1 \cdot P(h = h_{t+1}), \text{ if } q_t = Q \text{ and } U_t = 0 \\ (q_t - 1, h_{t+1}) & \text{w.p. } (1-p)P(h = h_{t+1}), \text{ if } q_t = Q \text{ and } U_t = 1 \\ (q_t, h_{t+1}) & \text{w.p. } pP(h = h_{t+1}), \text{ if } q_t = Q \text{ and } U_t = 1 \end{cases} \tag{12}$$

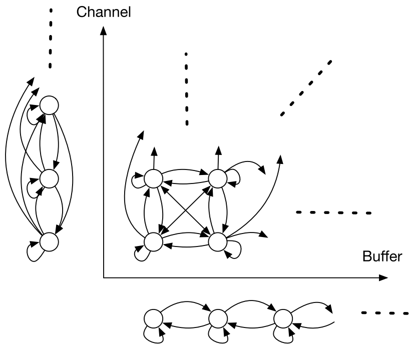


Fig. 1. Markov chain of the system, where the arrows show all possible transitions.

Denote $V^*(q,h)$ as the optimal value function at state (q,h). Then the optimality equations are:

$$V^*(0,h) = \alpha\mathbf{E}_{a,h'}V^*(a,h'), \tag{15}$$

$$V^*(q,h) = \min\Big\{\beta P_T(h) + \alpha\mathbf{E}_{a,h'}V^*(q-1+a,h'),$$
$$\alpha\mathbf{E}_{a,h'}V^*(q+a,h')\Big\}\ 1 \le q \le Q-1, \tag{16}$$

$$V^*(Q,h) = \min\Big\{\beta P_T(h) + \alpha\mathbf{E}_{a,h'}V^*(Q-1+a,h'),$$
$$p + \alpha\mathbf{E}_{a,h'}V^*(Q,h')\Big\}, \tag{17}$$

where a packet arrival is denoted by a binary random variable a with with $P(a=1)=p$. The expectation is taken over a and h'. Equation (15) follows since no transmission is taken when the buffer is empty, (16) is the case where the buffer is not empty nor full, and the two terms correspond to the expected cost of transmission and silence, and (17) represents the situation when the buffer is full, being silent will incur no cost but will have additional expected cost p for packet dropping due to a new packet arrival.

There are three main propositions regarding this system:

Proposition 1: The value function $V^*(q,h)$ is nondecreasing in q and nonincreasing in h.

Proof: see Appendix A. □

Remark 1: Notice that the optimal value function can also be obtained by the value iteration algorithm. The main idea is to employ induction and show that such a monotonicity structure holds in each iteration.

Proposition 2: The one-step difference function $p\cdot\mathbf{1}(q=Q)+\mathbf{E}_{a,h'}V^*(\min\{q+a,Q\},h')-\mathbf{E}_{a,h'}V^*(q-1+a,h')$ is increasing in q.

Proof: see Appendix B. □

Remark 2: Two inequalities are required to show the monotonicity, one directly follows from the optimality equations, the other comes from induction.

Given Propositions 1 and 2, we are able to show the thresholded policy structure of the optimal policy.

Theorem 3: Thresholded policy: the optimal control policy is that, at state (q,h), we transmit only when $q \ge q_{th}(h)$ and $h \ge h_{th}(q)$, where q_{th} and h_{th} are threshold functions.

Proof: See Appendix C. □

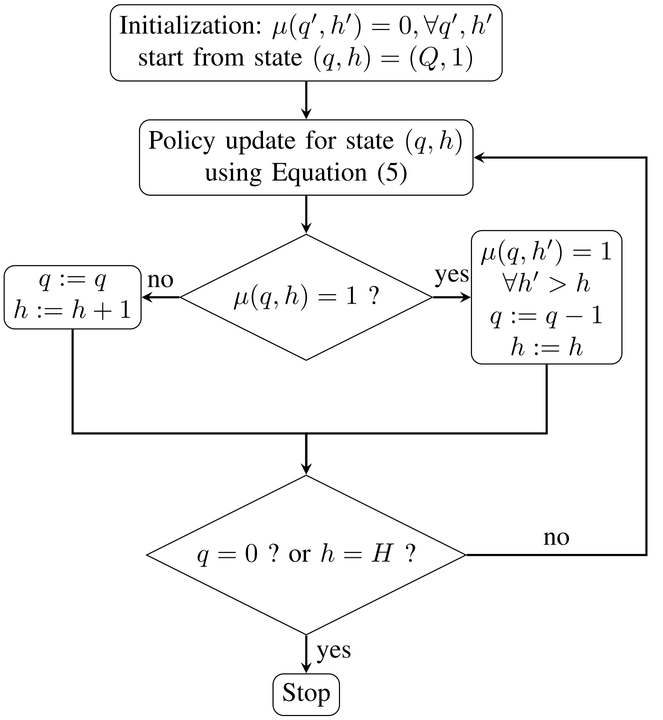


Fig. 2. Block diagram for modified policy improvement, the action $\{0,1\}$ denotes silence and transmit, respectively.

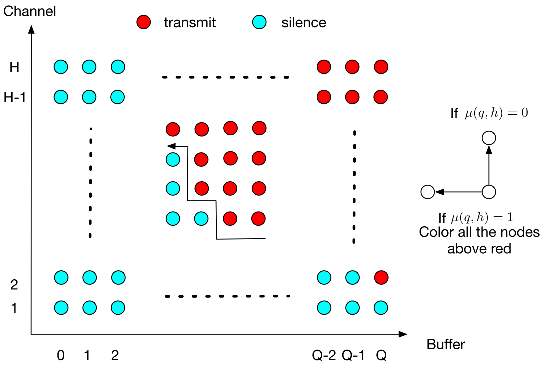


Fig. 3. Modified policy improvement step, where red nodes denote transmit, blue nodes denote silence; the action $\{0,1\}$ denotes silence and transmit, respectively.

Remark 3: The thresholded policy can be proved by directly applying proposition 1 and 2 to the optimality equations.

The structured policy revealed by Theorem 3 is appealing since it enables efficient computation in the policy iteration. Due to the particular form of the optimal policy, a specialized algorithm can be developed to search among policies that have the same form as the optimal policy, avoiding the need for checking all states to perform policy update, which is a typical step in the original policy iteration algorithm.

The flowchart and pictorial representation of the modified policy update method are shown in Figs. 2 and 3. The general idea of the modified policy update can be described as follows: with states placed on a 2D plane. We start from state $(Q,1)$ and go diagonally towards $(0,H)$. At particular state (q,h), we keep increasing h by 1 until we reach the state whose optimal action is to transmit. From the threshold property we know all the states above the threshold should transmit. Then, the buffer index q is decreased by 1 and the process is repeated until we reach the boundary states $q = 0$ or $h = H$.

It should be noted that this approach is a heuristic algorithm, but it is also to easy see that the original policy check is reduced to a "zig-zag" check. In addition, the original policy iteration needs to perform two policy checks for $(Q+1) \times H$ states; whereas in the zig-zag check, at most $Q + H$ number of states are needed (from state (Q, H) to state $(0, H)$). Therefore, the complexity for policy update reduces from $O(2 \cdot (Q+1) \cdot H)$ to $O(2 \cdot (Q+H))$.

It is clear that the algorithm essentially identifies the boundary states. Although we can update the boundary based on the previous execution of the zig-zag policy, we may need to check the states around the previous boundary to determine the new boundary, which may require visiting more than $(Q+H)$ states.

IV. Reduced Dimension MDP and Subspace Construction

In spite of the existence of a nice optimal policy structure, it is quite common that the state space and even the action space grow large in real systems, such a curse of dimensionality is still a major obstacle for the implementation of policy iteration. Therefore, a dimension reduction technique is highly desired. In this section, we first demonstrate the idea of projection for reduced dimension MDPs and derive the optimal subspace construction criteria. We will also further show that fast subspace construction is enabled under the system proposed in Sec. III.

To address the complexity challenge, *approximate dynamic programming* [9] has been proposed. We first notice that the system has composite states (q, h) and the state variable h is uncontrollable due to the i.i.d. channel property (It is an important property as we will explain later). A simplification method for uncontrollable state components has been proposed ([9], Sec. 6.1.5), where dimension reduction can be achieved by averaging out the uncontrollable state variable and hence a Bellman equation of lower dimension can be obtained. The details are omitted due to space constraints, but numerical results of this method will be provided in Sec. VI. While we will observe that the method has promise for one particular wireless network example, the method has limited applicability to general MDP problems. In contrast, the strategy we propose herein works well with general MDP problems as well.

A more general approach is through the *projected equation* method, where we seek compact representation of value function in a lower dimensional subspace $\hat{\mathbf{v}}_\mu = \mathbf{Mr}$. The subspace is denoted by the matrix $\mathbf{M}$ with dimension $N \times K$, where $N = |\mathcal{V}|$ is the size of the graph, $K \ll N$ is the size of the subspace, and $\mathbf{r}$ is the coefficient vector.

Without loss of generality, we assume that the columns of $\mathbf{M}$ are orthonormal. Since the original value function $\mathbf{v}_\mu$ is a fixed point of the Bellman operator, a way to find a good approximation is to force the linear approximation $\hat{\mathbf{v}}_\mu = \mathbf{Mr}$ to also be a fixed point under the Bellman operator. As a matter of fact, this is the main idea of *least-squares fixed point approximation* in [44].

Therefore, we seek an approximate value function $\hat{\mathbf{v}}_\mu$ that is invariant under the Bellman operator followed by projection, which will yield the following derivation:

$$\begin{aligned}\hat{\mathbf{v}}_\mu &= \mathbf{MM}^T(\mathbf{c}_\mu + \alpha \mathbf{P}_\mu \hat{\mathbf{v}}_\mu)\\ \Rightarrow \mathbf{Mr} &= \mathbf{MM}^T\mathbf{c}_\mu + \alpha \mathbf{MM}^T\mathbf{P}_\mu \mathbf{Mr}\\ \Rightarrow \mathbf{r} &= \mathbf{M}^T\mathbf{c}_\mu + \alpha \mathbf{M}^T\mathbf{P}_\mu \mathbf{Mr}\\ \Rightarrow \mathbf{r} &= (\mathbf{I} - \alpha \mathbf{M}^T\mathbf{P}_\mu \mathbf{M})^{-1}\mathbf{M}^T\mathbf{c}_\mu, \end{aligned} \tag{18}$$

where the third equation follows from the orthogonality of $\mathbf{M}$.

Then the approximated value function is:

$$\hat{\mathbf{v}}_\mu = \mathbf{M}(\mathbf{I} - \alpha \mathbf{M}^T\mathbf{P}_\mu \mathbf{M})^{-1}\mathbf{M}^T\mathbf{c}_\mu. \tag{19}$$

In standard policy iteration, for each iteration, the complexity is generally $O(N^3)$ due to matrix inversion in the policy evaluation step (4). But in the projected equation case, the pure matrix inversion is reduced to $O(K^3)$ and the complexity of matrix-vector multiplication is $O(N^2)$; therefore complexity reduction is achieved.

It should be noted that the convergence of such a reduced dimension method is not always guaranteed, since the approximated value function is used for policy improvement (5), and one possible reason is due to policy oscillation [9]. To ensure convergence to the true value function, one trivial condition is to have $\mathbf{v}_\mu = \hat{\mathbf{v}}_\mu$, then the criteria for subspace selection can be given by the following proposition:

Theorem 4: If $\mathbf{P}_\mu$ is low rank, to achieve $\mathbf{v}_\mu = \hat{\mathbf{v}}_\mu$, the subspace for Equation (19) is the set of orthonormal basis vectors that span the column space of $\mathbf{P}_\mu \oplus \mathbf{c}_\mu$, where $\oplus$ denotes direct sum.

Proof: See Appendix D. □

Remark 4: The proof can be seen by direct comparison of the original value function in full dimension and the approximated value function. This proposition provides a general strategy for optimal subspace construction and has potential application in many networks, since it is possible that the PTM induced by the protocol in a FSM is low rank.

One major shortcoming of such method lies in the changing subspace and possibly changing rank of $\mathbf{P}_\mu$ in each iteration. However, the model given in Section III can have further complexity reduction as a result of the following rank preservation property.

Theorem 5: Under the network setting in Section III, the rank of $\mathbf{P}_\mu$ is always equal to $Q + 1$, where Q is the buffer capacity. For any given channel state h_0, the columns associated with state $(q, h_0), q = \{0, 1, 2, \dots, Q\}$ in $\mathbf{P}_\mu$ form an independent set of basis that span the column space of $\mathbf{P}_\mu$.

Proof: The proof is provided in Appendix E. □

Remark 5: The proof is completed by exploiting the i.i.d. fading in the channel, as well as the independence between packet arrival and the channel. Then, it can be shown that states with common buffer length q share similar incoming transition pattern. The significance of the uncontrollability of h should also be emphasized. Since h is uncontrollable, Theorem 5 can also be interpreted as the rank of the PTM *equals the number of controllable elements of the state space, so that Theorem 4 can directly apply.*

Theorem 4 and 5 offer a fast basis construction method. We assume the states are ordered according to the lexicographical order (first h then q). In each iteration, the independent set of basis can be selected from the first $Q+1$ columns of P_μ and then concatenate with c_μ; and orthogonality can be simply achieved by Gram-Schmidt orthogonalization.

The subspace ratio can be defined as the ratio between the size of the subspace and the original dimension. In our wireless network, the original dimension is $(Q+1)\times H$, whereas the size of the subspace is $Q+2$. Thus the ratio is $\frac{Q+2}{(Q+1)H} \approx \frac{1}{H}$ when Q, H are large. The efficiency increases as Q and H get larger.

In Section VI, we will numerically show that such a subspace selection method achieves zero policy error as well as faster runtime compared with the standard policy iteration algorithm.

V. General Subspace Construction Using GSP Methods

It can be observed that the design of subspace M is still application dependent. Furthermore, it may be too stringent of a condition for the PTM of the wireless system to be low rank, or to have the same rank in each iteration. This serves as the motivation for the investigation of general methods, and Graph Signal Processing can be one possible solution.

From the Bellman Equation (4), we notice that the probability transition matrix can be viewed as a transition graph, and the value function can be viewed as the corresponding graph signal. The theory of GSP in Section II-B provides a way of constructing a set of orthonormal basis vectors that can represent any graph signal. The decomposition of the graph signal is given by its graph Fourier transform, *i.e.*, the projection of the graph signal onto each basis. Compact representation of the graph signal can be achieved if it is smooth on the graph, *i.e.*, most of the energy will be concentrated on the low frequency components. It can also be seen from the Bellman Equation (4), where there is a smoothing operation with respect to the value function (the value function of each node is the averaging of value functions from its neighbor nodes), therefore we can assume that the value function is a relatively smooth signal across the graph so that it can be approximated by the low frequency basis.

In typical MDP settings, a major challenge is that the relationship captured by the edge directivity in directed graphs is fundamentally different from that of undirected graphs. To combat this challenge, the directed graph (PTM) is replaced with an undirected proxy that can both preserve system information and has computational advantage. **Therefore, the general strategy is to find a proxy graph on which the value function is smooth, thus the subspace can be formed by picking the low frequency eigenvectors of the Laplacian of the proxy graphs**. The construction of undirected proxies involves graph symmetrization [31], [35] or other methods so that the GSP techniques designed for undirected graphs can apply.

Another challenge is that the probability transition matrix changes after each policy update in policy iteration. Generating the subspace in each iteration accordingly will incur high complexity. To reduce complexity, a fixed subspace will be generated from functions of the averaged probability transition matrix $\bar{P}$, thus the key subspace is computed only once. The definitions of $\bar{\mathbf{P}}$ and $\bar{\mathbf{c}}$ are given by Equation (20).

$$\bar{\mathbf{P}}(s,s') = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} p(s,u,s'), \quad \bar{\mathbf{c}}(s) = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} c(s,u). \tag{20}$$

In the sequel, we demonstrate methods to construct undirected proxies given a directed graph $\mathbf{P}$, they are described as follows:

A. SYM: Natural Symmetrization

The undirected proxy can be obtained by $\mathbf{A}_1 = \frac{1}{2}(\mathbf{P}+\mathbf{P}^T)$. Then the Laplacian matrix is defined as $\mathbf{L}_1 = \mathbf{D}_1 - \mathbf{A}_1$, where $\mathbf{D}_1$ is the degree matrix of $\mathbf{A}_1$. And a set of eigenvectors are given by the eigenvalue decomposition of the Laplacian matrix $\mathbf{L}_1$. Given the subspace size K, the subspace $\mathbf{M}_1$ is then formed by selecting the eigenvectors associated with the smallest K eigenvalues (corresponding to low frequencies, see Equation (8)). The advantage of such method is of course simplicity, it retains the same set of edges but ignores the edge directivity.

B. BIB: Bibliometric Symmetrization

In the analysis of document citations, due to the directed nature of the citation graph, the symmetrized graph can be obtained as $\mathbf{P}\mathbf{P}^T$, *a.k.a. bibliographic coupling* matrix [45]; it measures the similarity of nodes that share similar out-links. Similarly, we can also obtain the *co-citation* matrix $\mathbf{P}^T\mathbf{P}$ [46], which emphasizes the significance of common in-links. Since there is no obvious reason to value the out-links more than the in-links, we take the sum of the two matrices and obtain the *bibliometric symmetrization* [31] matrix as $\mathbf{A}_2 = \mathbf{P}\mathbf{P}^T + \mathbf{P}^T\mathbf{P}$. Although this approach may result in an undirected graph with a completely different structure, the edge directivity is preserved. In addition, depending on the application and the emphasis on in-degree and out-degree, the bibliometric matrix can also be constructed as $\mathbf{P}_{2,\gamma} = \gamma\mathbf{P}\mathbf{P}^T + (1-\gamma)\mathbf{P}^T\mathbf{P}, \gamma \in [0,1]$. In this paper for simplicity we just set $\gamma = 0.5$.

Notice that $\mathbf{A}_2$ is a PSD matrix, thus picking the low frequency eigenvectors from its Laplacian matrix is equivalent to picking the eigenvectors of $\mathbf{A}_2$ that have large eigenvalues. Therefore, subspace $\mathbf{M}_2$ can be formed by selecting eigenvectors whose associated eigenvalues are the largest K eigenvalues.

C. AVF: Approximated Value Function Graph

The notion of *approximated value function* graph first appeared in [35]. It is a virtual graph that measures the similarity of value functions between states. Denote $d(s)$ as the minimum number of hops from state s to a predefined high cost region in the system. The main idea is that two

TABLE I

PROPOSED METHODS FOR SUBSPACE DESIGN

Basis	Prior	Proxy	Description
SYM	$\bar{\mathbf{P}}$	$\frac{1}{2}(\bar{\mathbf{P}}+\bar{\mathbf{P}}^T)$	EVD of the Laplacian matrix of proxy and pick eigenvectors with small eigenvalues
BIB	$\bar{\mathbf{P}}$	$\bar{\mathbf{P}}\bar{\mathbf{P}}^T+\bar{\mathbf{P}}^T\bar{\mathbf{P}}$	EVD of the bibliometric symmetrization and pick eigenvectors with large eigenvalues
AVF	sgn($\bar{\mathbf{P}}$), sgn($\bar{\mathbf{c}}$)	$\mathbf{W}_a$	EVD of the Laplacian matrix of proxy and pick eigenvectors with small eigenvalues
Jordan	$\bar{\mathbf{P}}$	No	Jordan decomposition of $\bar{P}$ and pick eigenvectors whose eigenvalues are closer to (1,0) on complex plane

Algorithm 1 AVF Graph Construction

```
1, Define the high cost region $\mathcal{H} = \{s : \bar{\mathbf{c}}(s) > \tau, \forall s \in \mathcal{S}\}$, s.t. card($\mathcal{H}$)/card($\mathcal{S}$) $= 0.1$, where $\tau$ is a threshold. Compute $d(s)$ for each state $s \in \mathcal{S}$.
2, Set all the non-zero entries of $\bar{\mathbf{P}}$, $\bar{\mathbf{c}}$ to 1 i.e., $\bar{\mathbf{P}} :=$ sgn($\bar{\mathbf{P}}$), $\bar{\mathbf{c}} :=$ sgn($\bar{\mathbf{c}}$), normalize $\bar{\mathbf{P}}$ so that it is row stochastic.
3, Set $\bar{\mathbf{v}}^{(0)} = \mathbf{1}$ and iteratively apply $\bar{\mathbf{v}}^{(k+1)} = \bar{\mathbf{c}} + \alpha\bar{\mathbf{P}}\bar{\mathbf{v}}^{(k)}$ until the following condition holds (the "SNR" for two consecutive value functions is above a threshold):
    $20\log_{10}\left(\frac{||\bar{\mathbf{v}}^{(k+1)}||_2}{||\bar{\mathbf{v}}^{(k+1)} - \bar{\mathbf{v}}^{(k)}||_2}\right) > 40.$
4, Set $\bar{\mathbf{v}} = \mathbf{v}^{(k+1)}$, and $\mathbf{W}_a(s,s') = \exp\{-[\bar{\mathbf{v}}(s) - \bar{\mathbf{v}}(s')]^2/(2\sigma^2)\}$ if $|d(s) - d(s')| \leq 1$ ($\sigma$ is the variance of $\bar{\mathbf{v}}$), $\forall s, s' \in \mathcal{S}$
```

Algorithm 2 Basis Construction With Jordan Form

```
1, Perform Jordan decomposition on $\bar{\mathbf{P}}$, $\bar{\mathbf{P}} = \mathbf{YJY}^{-1}$.
2, Subspace $\mathbf{T}_j$ is formed by the following rule:
    $\mathbf{y}_i \in \mathbf{T}_j \quad \text{if } |\lambda_i - 1| < \epsilon, \ \forall i$
   where $\epsilon$ is a threshold that affects the size of the subspace.
3, $\mathbf{M}_j = |\mathbf{T}_j|$.
4, Perform Gram-Schmidt orthogonalization on $\mathbf{M}_j$.
```

states s and s' are similar if $|d(s) - d(s')| \leq 1$, and there exists a virtual link connecting them with weights indicating the similarity, such a similarity matrix is denoted by $\mathbf{W}_a$. The construction of $\mathbf{W}_a$ requires the knowledge of the value function, but we can not obtain the value function without actually implementing standard algorithms. To avoid this complexity, we propose a method to estimate the value function and use the approximated value function to construct W_a. To this end sgn($\bar{\mathbf{P}}$) and sgn($\bar{\mathbf{c}}$) are required, where $\bar{\mathbf{P}}$ and $\bar{\mathbf{c}}$ are defined in Equation (20), and sgn($\cdot$) is the sign function. The construction of the AVF graph is described in Algorithm 1.

The AVF graph $\mathbf{W}_a$ serves as a hidden undirected graph for the system, therefore we can select eigenvectors from the corresponding Laplacian matrix to form the subspace. AVF was our early attempt to apply GSP to MDPs; we include this method [35] here for comparison.

D. Jordan Form

As a comparison to symmetrization, the Jordan form approach [29] is also considered. The Jordan decomposition of $\bar{\mathbf{P}}$ provides another set of basis vectors for signals defined on directed graphs. Therefore, a set of low frequency basis can be selected and complex numbers are mapped to real numbers using magnitude. Gram-Schmidt orthogonalization is also employed to tackle non-orthogonality of the basis vectors. The algorithm is described in Algorithm. 2.

The summary of all the methods is shown in Table I.

VI. NUMERICAL RESULTS

We simulate the system in Section III, the parameters for the channel come from a sensor network application [2], [47]; they are set as: $C_0 = 10^{0.17}$, $\eta = 4.7$, $l = 20$m, and $P_{rcv} = -97$dBm. The packet arrival rate is $p = 0.9$ and the buffer can store at most 50 packets.

In our simulations, the partition of channel is set to be $[0,1) \cup [1,2) \cdots [H-2, H-1) \cup [H-1, +\infty)$, with the midpoint of each interval representing the channel state. Therefore, the channel state can be characterized by a discrete random variable $h = \{0.5, 1.5, \ldots, H-0.5\}$, and the PMF of h can be calculated as:

$$\begin{aligned} P(h = n - 0.5) &= \int_{n-1}^{n} \mathrm{e}^{-x}\mathrm{d}x = \mathrm{e}^{-n+1} - \mathrm{e}^{-n} \\ &\qquad\qquad n = \{1, 2, \ldots, H-1\} \\ P(h = H - 0.5) &= \int_{H-1}^{\infty} \mathrm{e}^{-x}\mathrm{d}x = \mathrm{e}^{-H+1}. \end{aligned} \tag{21}$$

The channel is discretized into 40 states thus the total number of states is 2040. In later sections, the performances of different methods as a function of network size will be demonstrated. For different network (probability transition graph) sizes, Q and H are chosen in a way that they are close to each other. Typical parameter settings for the networks sizes are listed as follows: $[100, 500, 900, 1200, 1600, 2000] = [10 \cdot 10, 25 \cdot 20, 30 \cdot 30, 40 \cdot 30, 40 \cdot 40, 50 \cdot 40]$.

Fig. 4 shows the value function and optimal policy for a particular β, where in the figure for the optimal policy, $\{0, 1\}$ indicate silence and transmission, respectively. As shown in the figure, the monotonicity of value function and the thresholded policy structure are clearly observed. For Poisson arrivals, the numerical results are shown in Fig. 5. It is observed that the optimal policy also has a threshold form.

As channel state information can be obtained via channel estimation, we can examine the effect of channel estimation error. The channel estimation error is modeled as: $e \sim \mathcal{N}(0, \sigma^2)$ and the estimated channel state is $h' = h + e$. Let μ^* be the optimal policy and μ' the policy determined under channel estimation errors. We compare the value function under these two policies $\mathbf{v}_{\mu^*}$, $\mathbf{v}_{\mu'}$. We use the normalized error (NE) to measure the error between $\mathbf{v}_{\mu^*}$ and $\mathbf{v}_{\mu'}$; it is defined

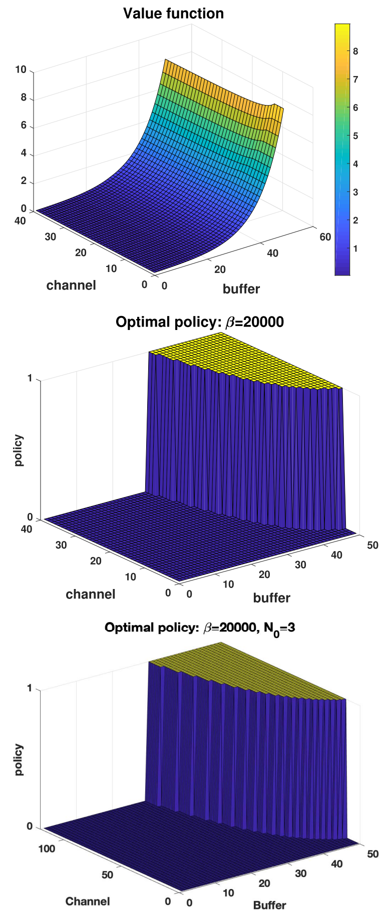


Fig. 4. Pictorial representation of the value function and optimal policy. For optimal policy, $\{0, 1\}$ denote silence and transmission, respectively.

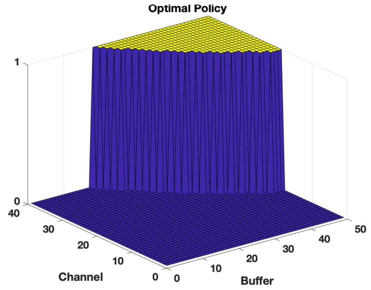


Fig. 5. Poisson arrivals: optimal policy. For optimal policy, $\{0, 1\}$ denote silence and transmission, respectively.

as: $NE = ||\mathbf{v}_{\mu^*} - \mathbf{v}_{\mu'}||/||\mathbf{v}_{\mu^*}||$. A typical signal to noise ratio in the channel is 3dB to 10dB, $\text{SNR}_{dB} = 10\log_{10}(1/\sigma^2)$, *i.e.*, $\sigma^2 \in [0.1, 0.5]$. The normalized error is shown in Fig. 6. It can be observed that if good channel estimation is achieved, the performance (value function) is close to the optimal one.

For simplicity, channel quantization is applied, we can consider finer quantization:

$$\left[0, \frac{1}{N_0}\right) \cup \left[\frac{1}{N_0}, \frac{2}{N_0}\right) \cdots \left[(H-1)\frac{N_0-1}{N_0}, H-1\right) \cup [H-1, +\infty), \quad (22)$$

where N_0 tunes the "resolution" of quantization. A larger N_0 yields more accuracy in the optimal policy since the Rayleigh fading variable is continuous. Fig. 4 also shows a finer boundary when $N_0 = 3$; for example, given $q = 40$, it can be calculated that the threshold states for original quantization and finer quantization are 12 and $\frac{35}{3}$. Even in the absence of quantization, we have a thresholded policy. The state space will become uncountable which will be prohibitive for value or policy iteration computations.

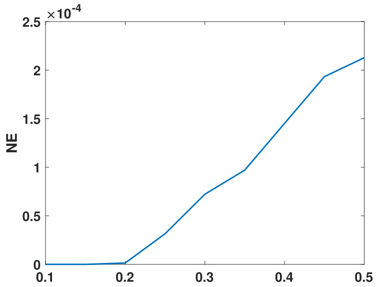


Fig. 6. Normalized error versus σ^2, network size $= 1200$.

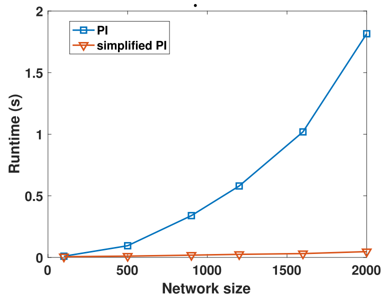


Fig. 7. Runtime comparison between original PI and simplified PI.

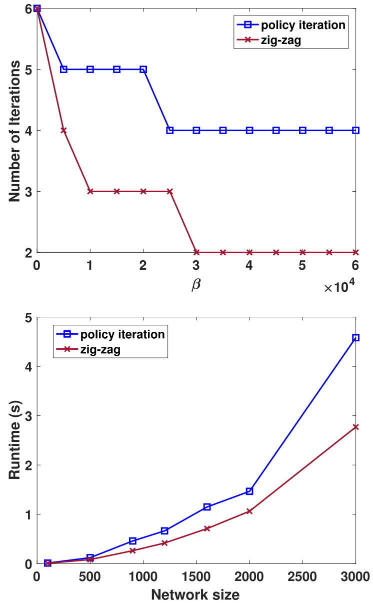


Fig. 8. Number of iterations of zig-zag policy update, network size $= 2040$, versus β (left figure); runtime comparison versus network size (right figure).

As mentioned in Sec. IV, the simplification method for uncontrollable state components [9] can be applied, the numerical result is shown in Fig. 7. It can be clearly observed that the simplification can dramatically reduce runtime complexity. However, the simplification method only works for special types of MDP problems where there are uncontrollable state components. For general MDP problems provided in

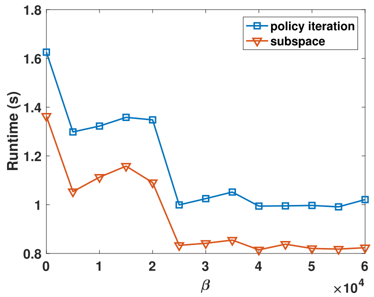


Fig. 9. Performance of the optimal basis selection method, network size = 2040, versus β.

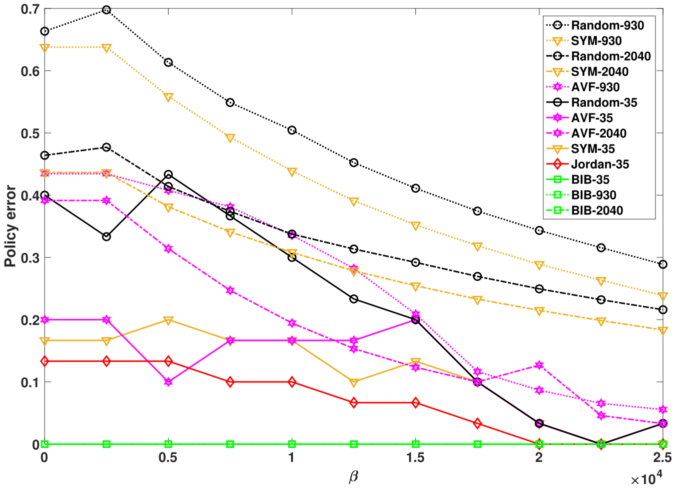


Fig. 10. Performance of subspace approach using GSP methods, with different network sizes, versus β.

Sec. VI-D, this method can not be applied, whereas our GSP approach still works and achieves good performance.

In the sequel, we will numerically evaluate the "zig-zag" policy update (Section III), low rank subspace construction (Section IV) and GSP subspace construction (Section V). To evaluate the performance under different network settings, the weighting factor β in Equation (13) is tuned, yielding different cost functions and thus leading to different policies. The cost for packet drop in cost function (13) is 1 and it can be calculated that the order of transmitting power is $O(10^{-4})$mw; thus for fair comparison the range of β is tuned from $O(1)$ to $O(10^4)$ (increase of emphasis on transmitting power). Since our major concern is optimal control, the policy error is defined as:

$$\text{policy error} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(\hat{\mu}(i) \neq \mu^*(i)), \tag{23}$$

where N is the total number of states, and the approximated policy and optimal policy are denoted by $\hat{\mu}$ and μ^*.

A. Zig-Zag Policy Update

The complexity of the modified algorithm can be measured by both the number of iterations for convergence and the runtime w.r.t. network size. From Fig. 8, it can be observed that, compared with the original policy iteration, the zig-zag policy update reduces the number of iterations by 50% for a fixed network size; as the network size increases, we have an even larger runtime reduction.

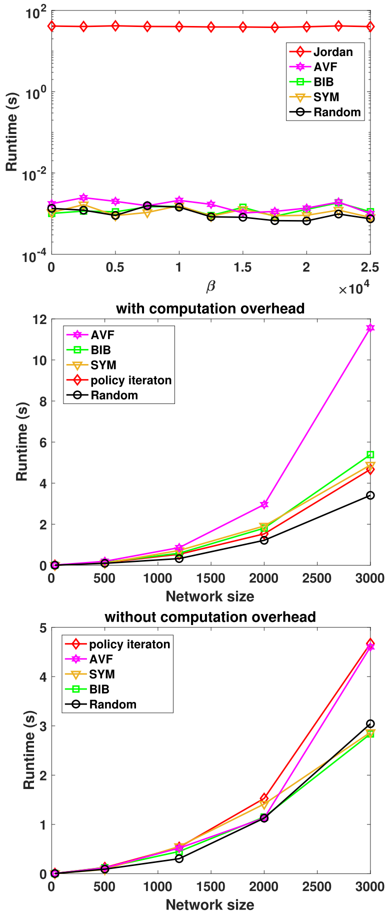


Fig. 11. Left figure: runtime of subspace approaches with network size = 35, versus β. Right two figures: runtime of subspace approaches (with/without computation overhead for subspace construction), versus network size.

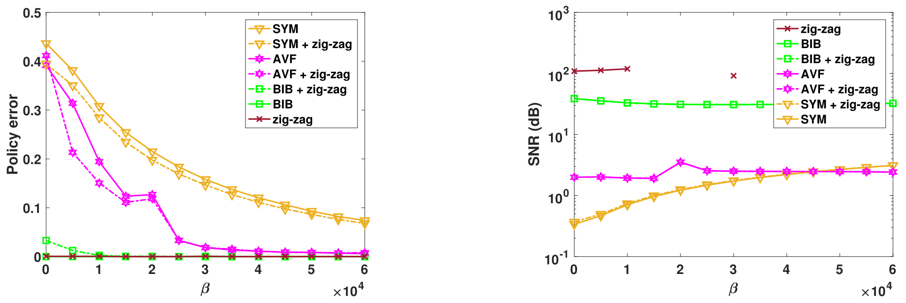
In the zig-zag policy update algorithm, the exact value function is used for zig-zag policy update. It is also worthwhile to investigate the performance of zig-zag policy update with an approximated value function. This performance will be shown in the sequel.

B. Optimal Basis Selection

Fig. 9 shows the performance of the optimal basis selection method in Section IV. In iteration k, the subspace $\mathbf{M}^{(k)}$ is formed by picking the independent columns of $\mathbf{P}_{\mu_k} \oplus \mathbf{c}_{\mu_k}$ and perform Gram-Schmidt orthogonalization. We see that the runtime varies for different β, since for each β we start with a random initial policy, the initial policy is same for both policy iteration and the subspace approach. Still, the runtime is roughly improved by 20%. Also, it should be notice that **we have perfect reconstruction of the policy, since the algorithm forces** $\hat{\mathbf{v}} = \mathbf{v}$. The subspace ratio is given by $\frac{Q+2}{(Q+1)\times H} = \frac{52}{51\times 40} = 0.0255$, obviously we are constructing a subspace that is much smaller than the original size.

C. Subspace Construction With GSP

The policy error is shown in Fig. 10, and different GSP approaches on different network sizes are indicated by the



(a) (b)

Fig. 12. Performance of subspace approaches using GSP methods, network size = 2040, versus β. In the SNR figure, the discontinuities represent infinite SNR.

labels. To further analyze the performance, a reference line labelled "Random" is introduced, whose method is to generate the subspace randomly.

Due to the complexity of the Jordan decomposition, it is only implemented on a newtork with small state space (size 35, $Q = 6, H = 5$), with subspace size 40% of the original size (see Fig. 11). It can be clearly observed that the Jordan form, while having reasonably good performance, suffers from high complexity and thus scaling to larger networks is problematic. It can also be observed from Fig. 11 that although the GSP methods incur some additional cost over the policy iteration due to the computation overhead for subspace construction, the subspace is constructed only once and thus the overall runtime is close to the classical policy iteration.

In networks with large state space (see Fig. 10) where the network size is 930 ($Q = 30, H = 30$) and 2040 ($Q = 50, H = 40$), the subspace is chosen to be 10% of the original size. Clearly, the natural symmetrization performs better than the random approach, but still has high policy error due to the fact that it ignores the directivity of the graph. The performance is further improved by the AVF graph approach, because instead of simple symmetrization, the undirected proxy graph is constructed by looking at the similarity of value function between nodes. Finally and surprisingly, the best performance (zero policy error) is achieved by bibliometric symmetrization. We conjecture that a possible reason for the good performance may be that not only it considers the similarity between nodes in terms of transition probability, but also preserves information of directivity by looking at both in-degree and out-degree links.

Fig. 12 shows the overall performance (policy error and SNR) of all methods (network size = 2040). The policy error is defined in Equation (23) and the Signal to Noise Ratio (SNR) is defined as:

$$\text{SNR} = 20\log_{10}\left(\frac{||\mathbf{v}^*||_2}{||\mathbf{v}^* - \hat{\mathbf{v}}||_2}\right), \tag{24}$$

where $\mathbf{v}^*$ and $\hat{\mathbf{v}}$ denote the original value function and the approximated value function, respectively.

We have already observed that the zig-zag policy update with exact value function can reduce the number of iteration for convergence, in addition, the SNR is really high (the discontinuities represent infinite SNR) so that the policy error is negligible. However, when combined with the subspace approach, the error is not negligible, due to the fact that the approximated value function is used, which has low SNR in natural symmetrization and AVF method. Still, we have better performance compared to the pure subspace approach.

D. More General Results

Two more examples will be given for further validation of the subspace methods using GSP. We notice that the simplification method [9] can not be applied here due to the absence of uncontrollable state components. The first example is an equipment replacement model in [32, Ch. 4.7.5], where the set of integers $s = \{1, 2, 3, \dots\}$ represents the condition of the equipment from good to bad. At each state the decision maker can choose from action set $u = \{u_0, u_1\}$, where u_0 corresponds to operating the equipment for an additional period; and u_1 corresponds to replacing it immediately. The transition probabilities satisfy:

$$p(s, u, s') = \begin{cases} 0 & s' \leq s, u = u_0 \\ g(s' - s) & s' > s, u = u_0 \\ g(s') & u = u_1 \end{cases}, \tag{25}$$

where $g(\cdot)$ is a predefined probability distribution, in our simulation, g is set to be an uniform distribution from integer 1 to 10.

The reward function consists of three parts: a fixed income of R units; a nondecreasing state-dependent cost $c_1(s) = \zeta s$, $\zeta = 0.01$, and a replacement cost $c_r = 5$. To convert the reward maximization problem to a cost minimization problem and apply the definition of value function in Equation (3), the cost function is obtained by taking the negative value of the reward:

$$c(s, u) = \begin{cases} -[R - c_1(s)] & u = u_0 \\ -[R - c_r - c_1(s)] & u = u_1. \end{cases} \tag{26}$$

It can be observed that the transition probability has structured form, and it also has been proven that such example exhibits a thresholded policy, *i.e.* equipment will be replaced only when its condition is worse than a threshold state s^*. Therefore, a modified policy iteration algorithm incorporating threshold information can also be developed, and we also apply our general subspace design method to different network sizes and tune the reward function R (so that we have different policies), the subspace is set to be 10% of the original size and the performance is shown in the left figure in Fig. 13. Although the bibliometric method does not achieve zero policy, it is still the best among all methods.

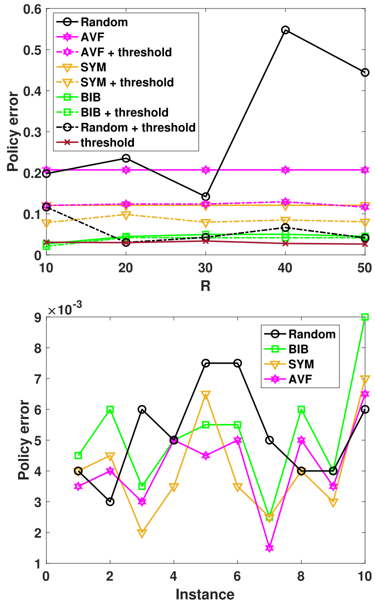


Fig. 13. Left figure: policy error for equipment replacement model, network size $= 2000$, versus R. Right figure: policy error for random graph model.

Previous examples focus on MDPs where the PTM is a structured graph. Simulation on random graphs is also conducted and the result is shown in the right figure in Fig. 13. The network size is 2000, the graph structure, transition probabilities and the cost functions are all generated randomly so that there is no policy structure to exploit. Therefore the subspace methods are pure GSP approach and it can be observed that all methods achieve negligible policy error.

VII. Conclusions

In this paper, a point-to-point transmission control problem is formulated as a MDP. The structure of the optimal policy is examined and modified policy iteration algorithm is proposed, with which the number of iterations is reduced by half. Based on the application, a fast subspace construction method is also presented, and we are able to attain both complexity reduction (runtime improved by roughly 20%) and perfect reconstruction of value function and policy. Furthermore, general subspace construction methods using GSP are also proposed. Among all the methods, the subspace obtained by the eigenvalue decomposition of the bibliometric symmetrization of the averaged PTM gives best performance (zero policy error).

There are a few directions for future work. First, the subspaces obtained through GSP generally involves eigenvalue decomposition of some PSD matrices, whose complexity is generally $O(N^3)$. It is important to have more efficient methods for subspace construction. Second, the GSP technique can be viewed as Fourier analysis of graph signal, it is also worthwhile to explore other suitable transformation methods (*e.g.* multiresolution analysis).

Appendix A
Proof of Proposition 1

A. Monotonicity of $V^*(q,h)$ in h

Since the channel states are i.i.d. over time, therefore $\mathbf{E}_{a,h'}V^*(q-1+a,h')$ and $\mathbf{E}_{a,h'}V^*(q+a,h')$ are constants w.r.t h, which can also be viewed as nonincreasing function of h. In addition, $\beta P_T(h)$ is a strictly decreasing function of h. Therefore, two functions in the min function are nonincreasing in h, so $V^*(q,h)$ is nonincreasing in h for given q.

B. Monotonicity of $V^*(q,h)$ in q

Notice that optimal value function can also be obtained by value iteration. We can start with $V^0(q,h) = 0$ for all q,h (*i.e.*, a zero vector), it automatically satisfies the nondecreasing property in q for any given h. Consider $1 \le q \le Q-1$ and assume that in kth iteration $V^{(k)}(q,h)$ is nondecreasing in q given h. In $(k+1)$th iteration we have:

$$\begin{aligned} &V^{(k+1)}(q+1,h)\\ &= \min\Big\{\beta P_T(h) + \alpha\mathbf{E}_{a,h'}V^{(k)}(q+a,h'),\\ &\quad p\cdot\mathbf{1}[q{=}Q-1] + \alpha\mathbf{E}_{a,h'}V^{(k)}(\min\{q{+}1{+}a,Q\},h')\Big\}\\ &\ge \min\Big\{\beta P_T(h) + \alpha\mathbf{E}_{a,h'}V^{(k)}(q-1+a,h'),\\ &\quad \alpha\mathbf{E}_{a,h'}V^{(k)}(\min\{q+a,Q\},h')\Big\}\\ &= V^{(k+1)}(q,h) \quad 1 \le q \le Q-1. \end{aligned} \tag{27}$$

where the inequality follows from the induction hypothesis.

For $q = 0$:

$$\begin{aligned} V^{(k+1)}(0,h) &= \alpha\mathbf{E}_{a,h'}V^{(k)}(a,h')\\ &\le \min\Big\{\beta P_T(h) + \alpha\mathbf{E}_{a,h'}V^{(k)}(a,h'),\\ &\qquad \alpha\mathbf{E}_{a,h'}V^{(k)}(1+a,h')\Big\}\\ &= V^{(k+1)}(1,h). \end{aligned} \tag{28}$$

the inequality follows because $\alpha\mathbf{E}_{a,h'}V^{(k)}(a,h') \le \beta P_T(h) + \alpha\mathbf{E}_{a,h'}V^{(k)}(a,h')$, and induction hypothesis: $\alpha\mathbf{E}_{a,h'}V^{(k)}(a,h') \le \alpha\mathbf{E}_{a,h'}V^{(k)}(1+a,h')$.

Hence, $V^{(k+1)}(q,h)$ is increasing in q for any given h. Hence, $V^{(j)}(q,h)$ is increasing in q for any given h, for all $j = 0,1,2,\ldots$. Since $V^{(j)} \to V^*$, monotonicity holds for V^* as well.

Appendix B
Proof of Proposition 2

We need to show for $1 \le q \le Q-2$ and for $q = Q-1$:

$$\begin{aligned} &\alpha\mathbf{E}_{a,h'}V^*(q+1+a,h') - \alpha\mathbf{E}_{a,h'}V^*(q+a,h')\\ &\quad \ge \alpha\mathbf{E}_{a,h'}V^*(q+a,h') - \alpha\mathbf{E}_{a,h'}V^*(q-1+a,h'), \end{aligned} \tag{29}$$

$$\begin{aligned} &p + \alpha\mathbf{E}_{a,h'}V^*(Q,h') - \alpha\mathbf{E}_{a,h'}V^*(Q-1+a,h')\\ &\quad \ge \alpha\mathbf{E}_{a,h'}V^*(Q-1+a,h') - \alpha\mathbf{E}_{a,h'}V^*(Q-2+a,h'), \end{aligned} \tag{30}$$

which is equivalent to:

$$\begin{aligned} &\mathbf{E}_{a,h'}V^*(q+1+a,h') + \mathbf{E}_{a,h'}V^*(q-1+a,h')\\ &\quad \ge 2\mathbf{E}_{a,h'}V^*(q+a,h'), \end{aligned} \tag{31}$$

$$\begin{aligned} &p + \alpha\mathbf{E}_{a,h'}V^*(Q,h') + \alpha\mathbf{E}_{a,h'}V^*(Q-2+a,h')\\ &\quad \ge 2\alpha\mathbf{E}_{a,h'}V^*(Q-1+a,h'). \end{aligned} \tag{32}$$

Sufficient conditions for the above inequalities to hold are $1 \leq q \leq Q-1$ and $\forall h'$ we have:

$$\mathbf{E}_{h'}V^*(q+1,h') + \mathbf{E}_{h'}V^*(q-1,h') \geq 2\mathbf{E}_{h'}V^*(q,h'), \tag{33}$$

$$1 + \alpha\mathbf{E}_{h'}V^*(Q-1,h') \geq \alpha\mathbf{E}_{h'}V^*(Q,h'). \tag{34}$$

To verify inequality (31), simply multiply inequality (33) by p with $q := q+1$ and by $1-p$ with $q := q$ and add them up, a comparison with the expansion of inequality (31) w.r.t. a can directly show that they are equivalent. Inequality (32) can be verified by multiplying inequality (33) by $\alpha(1-p)$ with $q := Q-1$ and adding with inequality (34) multiplied by p. In the sequel, we focus on proving inequality (31) since the derivations for inequality (32) are similar.

Recall from the optimality equation (16):

$$\begin{aligned} V^*(q,h) &= \min\Big\{\beta P_T(h) + \alpha\mathbf{E}_{a,h'}V^*(q-1+a,h'), \\ &\qquad \alpha\mathbf{E}_{a,h'}V^*(q+a,h')\Big\} \\ &\leq \alpha\mathbf{E}_{a,h'}V^*(q+a,h') \quad \text{(equality if } q=0) \\ &= \alpha p\mathbf{E}_{h'}V^*(q+1,h') + \alpha(1-p)\mathbf{E}_{h'}V^*(q,h'). \end{aligned} \tag{35}$$

Take expectation on both side w.r.t. h and it follows that:

$$\mathbf{E}_{h'}V^*(q+1,h') \geq \frac{1+\alpha p-\alpha}{\alpha p}\mathbf{E}_{h'}V^*(q,h'). \tag{36}$$

To show inequality (33), we only need to show:

$$\mathbf{E}_{h'}V^*(q-1,h') \geq \frac{\alpha p+\alpha-1}{\alpha p}\mathbf{E}_{h'}V^*(q,h'). \tag{37}$$

In value iteration, we start with $V^{(0)} = \mathbf{1}$, which automatically satisfies condition (37). We assume that in the kth iteration we have

$$\mathbf{E}_{h'}V^{(k)}(q-1,h') \geq \frac{\alpha p+\alpha-1}{\alpha p}\mathbf{E}_{h'}V^{(k)}(q,h'). \tag{38}$$

Then in the $(k+1)$th iteration:

If $q=0$.

$$\begin{aligned} V^{(k+1)}(0,h) &= \alpha\mathbf{E}_{a,h'}V^{(k)}(a,h') \\ &\geq \frac{\alpha p+\alpha-1}{\alpha p}\alpha\mathbf{E}_{a,h'}V^{(k)}(1+a,h') \\ &\geq \frac{\alpha p+\alpha-1}{\alpha p}\min\Big\{\beta P_T(h)+\alpha\mathbf{E}_{a,h'}V^{(k)}(a,h'), \\ &\qquad \alpha\mathbf{E}_{a,h'}V^{(k)}(1+a,h')\Big\} \\ &= \frac{\alpha p+\alpha-1}{\alpha p}V^{(k+1)}(1,h). \end{aligned} \tag{39}$$

The first inequality follows because a and h are independent random variables so we can directly use the induction hypothesis.

Similar analysis can be applied to the case where $1 \leq q \leq Q-1$. Equation (38) holds in each iteration and therefore holds in the optimal value function V^* since $V^{(k)} \to V^*$. As a result, we have proved Equation (37).

By simple summation, inequality (36) and inequality (37) will give us inequality (33), which is a sufficient condition for inequality (31). Together with inequality (32), we prove that $p\cdot\mathbf{1}(q=Q)+\mathbf{E}_{a,h'}V^*(\min(q+a,Q),h')-\mathbf{E}_{a,h'}V^*(q-1+a,h')$ is increasing in q.

Appendix C
Proof of Theorem 3

If the optimal action is to transmit, the expected cost of transmission should not be greater than that of silence:

$$\beta P_T(h) \leq \alpha\mathbf{E}_{a,h'}V^*(q+a,h') - \alpha\mathbf{E}_{a,h'}V^*(q-1+a,h')$$
$$1 \leq q \leq Q-1 \tag{40}$$

$$\beta P_T(h) \leq p + \alpha\mathbf{E}_{a,h'}V^*(Q,h') - \alpha\mathbf{E}_{a,h'}V^*(Q-1+a,h') \tag{41}$$

A. Thresholded Policy in h

When $q=0$, we do not transmit, which can be viewed as a special thresholding policy in h.

Given $1 \leq q \leq Q$, the R.H.S of Equation (40) and (41) are just a constant w.r.t. h, and $\beta P_T(h)$ is a decreasing function of h. Therefore there is a threshold value h_{th} such that the inequality holds when $h \geq h_{th}$. Therefore we have thresholded policy in h given q.

B. Thresholded Policy in q

When $q=0$, we do not transmit. If $1 \leq q \leq Q-1$, given h, $\beta P_T(h)$ in Equation (40) is just a constant. Proposition 2 reveals that $p\cdot\mathbf{1}(q=Q)+\alpha\mathbf{E}_{a,h'}V^*(\min(q+a,Q),h') - \alpha\mathbf{E}_{a,h'}V^*(q-1+a,h')$ is increasing in q, thus there exists a threshold q_{th} such that the inequality holds when $q \geq q_{th}$.

Appendix D
Proof of Theorem 4

The original value function and the approximated value function are given by $\mathbf{v} = (\mathbf{I}-\alpha\mathbf{P})^{-1}\mathbf{c}$ and $\hat{\mathbf{v}} = \mathbf{M}(\mathbf{I}-\alpha\mathbf{M}^T\mathbf{P}\mathbf{M})^{-1}\mathbf{M}^T\mathbf{c}$, where $\mathbf{M}$ is the subspace.

One sufficient condition for $\mathbf{v}=\hat{\mathbf{v}}$ is to have $(\mathbf{I}-\alpha\mathbf{P})^{-1} = \mathbf{M}(\mathbf{I}-\alpha\mathbf{M}^T\mathbf{P}\mathbf{M})^{-1}\mathbf{M}^T$. From this equality we have:

$$(\mathbf{I}-\alpha\mathbf{P})^{-1} = \mathbf{M}(I-\alpha\mathbf{M}^T\mathbf{P}\mathbf{M})^{-1}\mathbf{M}^T \tag{42}$$

$$\Rightarrow (\mathbf{I}-\alpha\mathbf{P})^{-1}\mathbf{M} = \mathbf{M}(\mathbf{I}-\alpha\mathbf{M}^T\mathbf{P}\mathbf{M})^{-1} \tag{43}$$

$$\Leftrightarrow \mathbf{M}(\mathbf{I}-\alpha\mathbf{M}^T\mathbf{P}\mathbf{M}) = (\mathbf{I}-\alpha\mathbf{P})\mathbf{M} \tag{44}$$

$$\Leftrightarrow \mathbf{M}\mathbf{M}^T\mathbf{P}\mathbf{M} = \mathbf{P}\mathbf{M}, \tag{45}$$

where the second equality follows from the orthogonality of $\mathbf{M}$.

Notice that $\mathbf{M}\mathbf{M}^T$ is a projection matrix, thus the columns of $\mathbf{P}\mathbf{M}$ lie in the column space of $\mathbf{M}$. Also, columns of $\mathbf{P}\mathbf{M}$ is in the column space of $\mathbf{P}$. Since $\mathbf{P}$ is low rank, a sufficient condition for Equation (45) to hold is to have columns of $\mathbf{M}$ spanning the column space of $\mathbf{P}$.

There will be an addition problem going back from Equation (43) to Equation (42) due to the existence of a

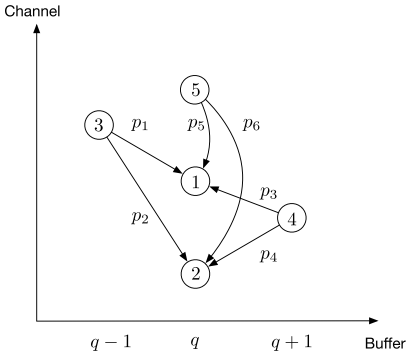


Fig. 14. Pictorial representation of low rank property. State 1 and 2 have same ancestor states, and their incoming transition probabilities are scalar multiple of each other.

projection matrix. However, if $\mathbf{c}$ also lies in the column space of $\mathbf{M}$, then from Equation (43) we have:

$$\begin{aligned}\hat{\mathbf{v}} &= \mathbf{M}(\mathbf{I}-\alpha\mathbf{M}^T\mathbf{P}\mathbf{M})^{-1}\mathbf{M}^T\mathbf{c}\\ &= (\mathbf{I}-\alpha\mathbf{P})^{-1}\mathbf{M}\mathbf{M}^T\mathbf{c}\\ &= (\mathbf{I}-\alpha\mathbf{P})^{-1}\mathbf{c}\\ &= \mathbf{v}. \qquad (46)\end{aligned}$$

Therefore, for $\mathbf{v} = \hat{\mathbf{v}}$, subspace $\mathbf{M}$ should be the set of orthonormal basis that span the column space of $\mathbf{P} \oplus \mathbf{c}$.

Appendix E
Proof of Theorem 5

One possible case for $\mathbf{P}$ being low rank is that it has multiple similar columns. From the perspective of Markov chain, it means that several states have similar in-degree.

In our model, the state of the system is represented by the pair (q, h), the state can be placed on a 2D plane (shown in Fig. 14).

Notice that given the protocol, for a state with buffer length q, it can only be reached from the states that have buffer length $q-1$ (packet arrival and no transmission), or buffer length q (no packet arrival and no transmission), or buffer length $q+1$ (no packet arrival and transmission). Due to the i.i.d property of the channel, for state i and state j that have the same buffer length, if there exists a transition from state k to state i, there also exists a transition from state k to state j. In other words, state i and state j have same set of ancestor states.

Denote $\mathbf{P}[(q',h') \to (q,h)]$ as the transition probability from state (q', h') to (q, h). Consider two states that have the same buffer length (q, h_1), (q, h_2). The in-degree/incoming probabilities for state (q, h_1) are as follows:

$$\mathbf{P}[(q',h') \to (q,h_1)] = \begin{cases} p \cdot P_h(h=h_1), & q'=q-1,\ \mu(q',h')=0.\\ (1-p)\cdot P_h(h=h_1), & q'=q,\ \mu(q',h')=0.\\ (1-p)\cdot P_h(h=h_1), & q'=q+1,\ \mu(q',h')=1.\\ 0, & \text{otherwise,}\end{cases}$$

where p is the probability for packet arrival, P_h is the probability mass function for channel state, and $\mu(q,h) = \{0,1\} \triangleq \{\text{silence}, \text{transmit}\}$.

Similarly, for the in-degree for state (q, h_2), we have:

$$\mathbf{P}[(q',h') \to (q,h_2)] = \begin{cases} p \cdot P_h(h=h_2), & q'=q-1,\ \mu(q',h')=0.\\ (1-p)\cdot P_h(h=h_2), & q'=q,\ \mu(q',h')=0.\\ (1-p)\cdot P_h(h=h_2), & q'=q+1,\ \mu(q',h')=1.\\ 0, & \text{otherwise.}\end{cases}$$

By simple comparison, it can be found that $\mathbf{P}[(q',h') \to (q,h_2)] = \frac{P_h(h=h_2)}{P_h(h=h_1)}\mathbf{P}[(q',h') \to (q,h_1)]$. Therefore, all the states sharing the same buffer length have similar in-degree distribution, *i.e.*, their in-degrees are just scalar multiples of each other. As a result, the rank of the probability transition matrix is always $Q+1$.

References

[1] M. Levorato, S. Narang, U. Mitra, and A. Ortega, "Reduced dimension policy iteration for wireless network control via multiscale analysis," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2012, pp. 3886–3892.

[2] A. Chattopadhyay, M. Coupechoux, and A. Kumar, "Sequential decision algorithms for measurement-based impromptu deployment of a wireless relay network along a line," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2954–2968, May 2015.

[3] A. Munir and A. Ross, "An MDP-based dynamic optimization methodology for wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 4, pp. 616–625, Apr. 2012.

[4] M. Abu Alsheikh, D. T. Hoang, D. Niyato, H.-P. Tan, and S. Lin, "Markov decision processes with applications in wireless sensor networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1239–1267, 3rd Quart., 2015.

[5] J. Riss, "Discounted Markov programming in a periodic process," *Operations Res.*, vol. 13, no. 6, pp. 920–929, 1965.

[6] N. Buras, "A three-dimensional optimization problem in water-resources engineering," *Oper. Res. Quart.*, vol. 16, no. 4, pp. 419–428, 1965.

[7] R. Mendelssohn, "Managing stochastic multispecies models," *Math. Biosci.*, vol. 49, nos. 3–4, pp. 249–261, 1980.

[8] D. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1. Belmont, MA, USA: Athena Scientific, 2005.

[9] D. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 2. Belmont, MA, USA: Athena Scientific, 2005.

[10] D. P. de Farias and B. V. Roy, "The linear programming approach to approximate dynamic programming," *Oper. Res.*, vol. 51, no. 6, pp. 850–865, 2003.

[11] L. Liu, A. Chattopadhyay, and U. Mitra, "On exploiting spectral properties for solving MDP with large state space," in *Proc. 55th Annu. Allerton Conf. Commun., Control Comput.*, Oct. 2017, pp. 1213–1219.

[12] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2011.

[13] M. Maggioni and S. Mahadevan, "Fast direct policy evaluation using multiscale analysis of Markov diffusion processes," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 601–608.

[14] M. Levorato, U. Mitra, and A. Goldsmith, "Structure-based learning in wireless networks via sparse approximation," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, p. 278, Aug. 2012.

[15] R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Appl. Comput. Harmon. Anal.*, vol. 21, no. 1, pp. 53–94, 2006.

[16] R. Givan, T. Dean, and M. Greig, "Equivalence notion and model minimization in Markov decision processes," *Artif. Intell.*, vol. 147, nos. 1–2, pp. 163–223, 2013.

[17] K. Deng, P. Mehta, and S. Meyn, "Optimal Kullback-Leibler aggregation via spectral theory of Markov chains," *IEEE Trans. Autom. Control*, vol. 56, no. 12, pp. 2793–2808, Dec. 2012.

[18] E. Pavez, N. Michelusi, A. Anis, U. Mitra, and A. Ortega, "Markov chain sparsification with independent sets for approximate value iteration," in *Proc. 35rd Annu. Allerton Conf.*, 2015, pp. 1399–1405.

[19] A. Barreto, J. Pineau, and D. Precup, "Policy iteration based on stochastic factorization," *J. Artif. Intell. Res.*, vol. 50, pp. 763–803, Aug. 2014.

[20] O. Teke and P. P. Vaidyanathan, “Extending classical multirate signal processing theory to graphs—Part I: Fundamentals,” *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 409–422, Jan. 2017.

[21] D. S. Zois and U. Mitra, “Active state tracking with sensing costs: Analysis of two-states and methods for n-states,” *IEEE Trans. Signal Process.*, vol. 65, no. 11, pp. 2828–2843, Jun. 2017.

[22] J. Chakravorty and A. Mahajan, “On the optimal thresholds in remote state estimation with communication costs,” in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 1041–1046.

[23] M. H. R. Khouzani, S. Sarkar, and E. Altman, “Dispatch then stop: Optimal dissemination of security patches in mobile wireless networks,” in *Proc. 49th IEEE Conf. Decis. Control*, Dec. 2010, pp. 2354–2359.

[24] M. Levorato, U. Mitra, and M. Zorzi, “Cognitive interference management in retransmission-based wireless networks,” *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3023–3046, May 2012.

[25] J. Chakravorty and A. Mahajan. (2017). “Sufficient conditions for the value function and optimal strategy to be even and quasi-convex.” [Online]. Available: https://arxiv.org/abs/1703.10746

[26] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[27] S. K. Narang and A. Ortega, “Perfect reconstruction two-channel wavelet filter banks for graph structured data,” *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2786–2799, Jun. 2012.

[28] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs,” *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[29] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs: Graph Fourier transform,” in *Proc. Int. Conf. Acoustic, Speech Signal Process. (ICASSP)*, 2013, pp. 6167–6170.

[30] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs: Frequency analysis,” *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, Jun. 2014.

[31] V. Satuluri and S. Parthasarathy, “Symmetrizations for clustering directed graphs,” in *Proc. 14th Int. Conf. Extending Database Technol.*, 2011, pp. 343–354.

[32] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2005.

[33] F. K. Chung, “Spectral graph theory,” in *Proc. CBMS Regional Conf. Math.*, Providence, RI, USA: AMS, 1997.

[34] F. K. Chung, “Laplacian and Cheeger inequality for directed graphs,” *Ann. Combinatorics*, vol. 19, no. 1, pp. 1–19, 2005.

[35] M. Levorato, S. Narang, U. Mitra, and A. Ortega, “Optimization of wireless networks via graph interpolation,” in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2013, pp. 483–486.

[36] N. M. D. Testa and M. Zorzi, “Optimal transmission policies for two-user energy harvesting device networks with limited state-of-charge knowledge,” *IEEE Trans. Wireless Commun.*, vol. 15, no. 2, pp. 1393–1405, Feb. 2016.

[37] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[38] M. Dong and L. Tong, “Optimal design and placement of pilot symbols for channel estimation,” *IEEE Trans. Signal Process.*, vol. 50, no. 12, pp. 3055–3069, Dec. 2002.

[39] E. Ertin, U. Mitra, and S. Siwamogsatham, “Maximum-likelihood-based multipath channel estimation for code-division multiple-access systems,” *IEEE Trans. Commun.*, vol. 49, no. 2, pp. 290–302, Feb. 2001.

[40] S. Beygi and U. Mitra, “Multi-Scale Multi-Lag channel estimation using low rank approximation for OFDM,” *IEEE Trans. Signal Process.*, vol. 63, no. 18, pp. 4744–4755, Sep. 2015.

[41] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[42] E. N. Gilbert, “Capacity of a burst-noise channel,” *Bell Syst. Tech. J.*, vol. 39, no. 5, pp. 1253–1265, 1960.

[43] E. O. Elliott, “Estimates of error rates for codes on burst-noise channels,” *Bell Syst. Tech. J.*, vol. 42, no. 5, pp. 1977–1997, Sep. 1963.

[44] M. G. Lagoudakis and R. Parr, “Least-squares policy iteration,” *J. Mach. Learn. Res.*, vol. 4, pp. 1107–1149, Dec. 2003.

[45] M. Kessler, “Bibliographic coupling between scientific papers,” *Amer. Documentation*, vol. 14, no. 1, pp. 10–25, 1963.

[46] H. Small, “Co-citation in the scientific literature: A new measure of the relationship between two documents,” *J. Amer. Soc. Inf. Sci.*, vol. 24, no. 4, pp. 265–269, 1973.

[47] A. Chattopadhyay *et al.*, “Impromptu deployment of wireless relay networks: Experiences along a forest trail,” in *Proc. 11th Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Oct. 2014, pp. 232–236.

Libin Liu received the B.E. degree in electronic engineering and information science from the University of Science and Technology of China, China, in 2015. He is currently pursuing the Ph.D. Degree with the Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA, USA. His research interests include graph signal processing, network control, and reinforcement learning.

Arpan Chattopadhyay received the B.E. degree in electronics and telecommunication engineering from Jadavpur University, Kolkata, India, in 2008, and the M.E. and Ph.D. degrees from the Electrical Communication Engineering Department, Indian Institute of Science, Bengaluru, in 2010 and 2015, respectively. He is currently with the Electrical Engineering Department, IIT Delhi, as an Assistant Professor. He has previously held postdoctoral positions at the Electrical Engineering Department, University of Southern California, Los Angeles, and the DYOGENE Group, INRIA/ENS, Paris, France. His research interests include wireless networks, cyber-physical systems, IoT, machine learning, and networked estimation and control.

Urbashi Mitra (F’07) received the B.S. and M.S. degrees from the University of California at Berkeley, Berkeley, CA, USA, and the Ph.D. degree from Princeton University, Princeton, NJ, USA. She is currently the Gordon S. Marshall Chair of engineering with the University of Southern California. Her research interests include wireless communications, biological communication, underwater acoustic communications, communication and sensor networks, and the detection, estimation, and the interface of communication, sensing, and control. She is currently the Co-Director of the Communication Sciences Institute, University of Southern California. She was a recipient of the 1996 National Science Foundation CAREER Award and the 1997 OSU College of Engineering MacQuigg Award for Teaching, the 2000 OSU College of Engineering Lumley Award for Research, the 2001 Okawa Foundation Award, the Texas Instruments Visiting Professorship (Rice University, Fall 2002), the 2009 DCOSS Applications and Systems Best Paper Award, the Best Paper Award from the 2012 GLOBECOM Signal Processing Symposium for Communications, the 2012 U.S. National Academy of Engineering Lillian Gilbreth Lectureship, the 2014 to 2015 IEEE Communications Society Distinguished Lecturer, the Women in Communications Engineering Technical Achievement Award from the IEEE Communications Society in 2017, the 2016 U.K. Royal Academy of Engineering Distinguished Visiting Professorship, the 2016 USA Fulbright Scholar Award, and the 2016 to 2017 U.K. Leverhulme Trust Visiting Professorship. She was the General Co-Chair of the first ACM Workshop on underwater networks at Mobicom 2006, Los Angeles, CA, USA. She was the Co-Chair of the IEEE Communication Theory Symposium at ICC 2003, Anchorage, AK, USA, the 2012 IEEE International Conference on Signal Processing and Communications, Bengaluru, India, the 2014 IEEE International Symposium on Information Theory, Honolulu, HI, USA, the 2014 IEEE Information Theory Workshop, Hobart, TAS, Australia, and the 2018 IEEE International Workshop on Signal Processing Advances in Wireless Communications, Kalamata, Greece. She is currently the Co-Chair of (Technical Program) the 2019 IEEE Communication Theory Workshop, Sefloss Iceland. She was the inaugural Editor-in-Chief of the IEEE TRANSACTIONS ON MOLECULAR, BIOLOGICAL, AND MULTI-SCALE COMMUNICATIONS. She was an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2012–2015), the IEEE TRANSACTIONS ON INFORMATION THEORY (2007–2011), the IEEE JOURNAL OF OCEANIC ENGINEERING (2006–2011), and the IEEE TRANSACTIONS ON COMMUNICATIONS (1996–2001).