Abstract—The ever-increasing size of wireless networks poses a significant computational challenge for policy optimization schemes. In this paper, we propose a technique to reduce the dimensionality of the value iteration problem, and thereby reduce computational complexity, by exploiting certain structural properties of the logical state transition network. Specifically, our method involves approximating the original Markov chain by a simplified one whose state transition graph contains an independent set of a prespecified size, thus resulting in a sparsification of the transition probability matrix. As a result, value iteration needs to be performed only on the vertex cover of the network, from which the value function on the independent set can be obtained in a one-step process via interpolation. The Markov chain approximation process presented in this paper, for a given choice of independent set, involves minimizing matrix distance defined in terms of Frobenius norm or the Kullback-Leibler distance. This minimum distance then helps us to define a cost that can be minimized through an iterative greedy algorithm to obtain an approximately optimal independent set. Our method provides a tradeoff between accuracy and complexity that one can exploit by choosing the size of the independent set. Numerical results show that for a class of “collision” networks the value function approximation is accurate, even with a large independent set.

Index Terms—Markov chain, independent set, Markov decision processes, approximate value function, wireless networks

I. INTRODUCTION

Modern cyber-physical systems involve a complex interaction of communication networks, as well as control or action networks, thus making them immense in size. The seamless management of a system of this complexity, interconnection, and scale is daunting and requires a fundamentally new design paradigm.

The operation of stochastic systems, such as wireless networks and cyber-physical systems, is traditionally modeled as a finite state machine (FSM) with Markovian transitions, determined by networking protocols and environmental variables. Optimization of such networks is performed by identifying the control strategy maximizing the value function [1], i.e., values defined over the network states that measure the expected long-term reward, given a control strategy and instantaneous rewards accrued on transitions between states. These functions can be used to measure network performance metrics such as throughput, packet delivery rate, buffer congestion, delay, etc. [2]. However, computing the value function can be computationally prohibitive for modern wireless networks, due to the enormous size of the underlying Markov chain modeling the dynamics of the system (which grows exponentially with the number of users in the network).

In this paper, we propose a novel method to approximate the value function by imposing restrictions on the connectivity of the network states, thus making it possible to reduce the complexity of optimizing network behavior. This value function approximation can then be incorporated into policy iteration algorithms to enable policy optimization [1]. In particular, our method involves approximating the original Markov chain by a simplified one whose state transition graph contains an independent set of a pre-specified size (i.e., a set of states that do not have any transitions between them). By exploiting the partition of the state space into independent sets and vertex covers, we show that value iteration needs to be performed only on the vertex cover of the network. The value function on the independent set can then be recovered from the vertex cover in a one-step process via interpolation. Previous approaches to this problem typically involve constructing approximate representations of the Markov chain followed by approximate recovery of the value function over it, thereby making them sensitive to the choice of approximations involved. In comparison, although our method also involves introducing approximations in the Markov chain, it benefits from the fact that value function recovery (from the vertex cover) is exact in Markov chains with independent sets. Further, our approximation is particularly appealing since independent sets appear frequently in wireless networks; for example, in the FSMs of the random walk, the backward counter and the forward counter, the set constituted by every “odd” or “alternate” state can form an independent set. These FSMs typically occur as the canonical sub-chains of most networks and are used to model, for example, buffers for incoming and outgoing data packets, delay and packet retransmission counters. Therefore, in our method, the penalty of approximating the Markov chain to contain independent sets can be expected to be minimal for real-world wireless networks.

In our formulation, the independent set approximation removes edges (state transitions) from the original Markov chain, resulting in a sparsification of its transition probability matrix. Since value iteration is done on the vertex cover only, choosing its size to be much smaller than the size of the original state space of the network results in significant reduction of complexity. We consider two metrics for approximating the Markov chain: the Kullback-Leibler divergence (KLD) and the Frobenius norm. Given a budget for the maximum size of the vertex cover set, our method involves minimizing these metrics with respect to both the transition probabilities and the choice

This research was funded in part by one or more of the following grants: NSF CCF-1117896, NSF CNS-1213128, NSF CCF-1410009, AFOSR FA9550-12-1-0215, DOT CA-26-7084-00, NSF CPS-1446901 and ONR N00014-09-1-0700.
of states included in the vertex cover. We achieve this by first determining a closed-form expression of the approximate Markov chain that minimizes the metric for a given choice of independent set. We then use the value of the minimum metric to define the cost function of an independent set. We find an independent set that approximately minimizes this cost function with an iterative greedy algorithm.

This paper is organized as follows: In Section II, we provide some background on value functions and their role in Markov decision processes, followed by related work on approximate value iteration in the literature. In Section III, we describe how the presence of independent sets simplifies the calculation of the value function. Section IV presents a framework for finding an independent set that approximately minimizes this cost function. We achieve this by using the undirected graph Laplacian as the basis functions. These basis functions yield a smaller subspace characterized by a set of parametric basis functions. Several works follow these lines to propose different classes of basis functions. For example, the work of [5] proposes Kernel-based approaches, whereas the authors in [6] propose using diffusion wavelets [7] and eigenfunctions of the undirected graph Laplacian as the basis functions. These approximations, however, can be sensitive to the choice of basis functions. Further, designing the basis functions to obtain better performance can be challenging.

More recently, there have been approaches for reducing the dimensionality of value function approximation by sampling the value function on a subset of states. The complete value function is then recovered using tools from graph signal processing [8]. For example, the authors in [9] consider the trajectory of the logical state of the network as a graph and propose a directed graph lifting transform for estimating the value function from its samples on a subset of states. Similarly, the work in [10] considers the usage of the multi-scale graph lifting analysis for estimation of the value function. Using prior knowledge, the work of [11] designs an undirected graph over which an interpolation approach is proposed to recover the value function from a subset of states. Most of these approaches involve approximating both the probability transition matrix and the value function. Our approach, however, is different in the sense that we only approximate the Markov chain with another containing a large independent set. The computation

\[ v = g + \gamma P v, \]

yielding

\[ v = (I - \gamma P)^{-1} g. \]

The difficulty in solving (7) is related to the inversion of matrix \( I - \gamma P \), whose complexity scales as \( O(n^3) \). Therefore, in large systems, it is important to focus on faster ways of performing policy evaluation, in order to reduce the overall complexity of policy optimization schemes. One way to compute \( v \) numerically is via the value iteration algorithm [1], which sets \( v[0] \) as an initial estimate of \( v \) to iteratively compute

\[ v[i] = g + \gamma P v[i-1], \quad i \geq 1. \]

As \( i \to \infty \), \( v[i] \) converges to \( v \) given by (7).

B. Related Work

The problem of value function approximation and approximate dynamic programming has been widely studied to date [3]. Most of the previous approaches suffer from the curse of dimensionality, due to the exponential dependence of the state space size on number of network components. In [4], value functions are approximated using linear projections onto a smaller subspace characterized by a set of parametric basis functions. Several works follow these lines to propose different classes of basis functions. For example, the work of [5] proposes Kernel-based approaches, whereas the authors in [6] propose using diffusion wavelets [7] and eigenfunctions of the undirected graph Laplacian as the basis functions. These approximations, however, can be sensitive to the choice of basis functions. Further, designing the basis functions to obtain better performance can be challenging.

More recently, there have been approaches for reducing the dimensionality of value function approximation by sampling the value function on a subset of states. The complete value function is then recovered using tools from graph signal processing [8]. For example, the authors in [9] consider the trajectory of the logical state of the network as a graph and propose a directed graph lifting transform for estimating the value function from its samples on a subset of states. Similarly, the work in [10] considers multi-scale graph lifting analysis for estimation of the value function. Using prior knowledge, the work of [11] designs an undirected graph over which an interpolation approach is proposed to recover the value function from a subset of states. Most of these approaches involve approximating both the probability transition matrix and the value function. Our approach, however, is different in the sense that we only approximate the Markov chain with another containing a large independent set. The computation
of the complete value function on this new Markov chain from the smaller set of states is exact.

There have also been approaches for dimensionality reduction by aggregating the states of the network to produce a smaller and more tractable Markov chain [12]. Finding an optimal state aggregation involves solving a combinatorial optimization problem whose complexity grows exponentially fast in the size of the state space. The authors in [13] propose a spectral theory-based heuristic to solve this problem. Our approach however operates on a new Markov chain of the same size, whose structure allows us to perform value iteration on a small subset of states. Another approach for mitigating the complexity of Markov chains involves finding a low-rank approximation of the original probability transition matrix in terms of KLD [14].

III. APPROXIMATE VALUE FUNCTION EVALUATION

In order to propose an approximate value function evaluation procedure, we first define a weighted directed graph \( G = (S, E, P) \), with vertices \( S = \{1, \ldots, n\} \) and edges \( E = \{(i, j) \in S \times S : p_{ij} \neq 0, i \neq j\} \). We wish to do processing over a subset of nodes \( C \subset S \), in order to reduce the complexity of computing the inverse in (7). To this end, we define a partition of the state space \( S \) into independent set and vertex cover sets, whose definitions are given below.

**Definition 1** (Independent set [15]). The set \( T \subset S \) is an independent set for a Markov chain with graph \( G = (S, E, P) \) if \( (i, j) \notin E \), for all \( i \in T, j \in T \setminus \{i\} \). The set \( C = T^c \) is termed vertex cover, where \( A^c \equiv S \setminus A \) denotes the complement set of \( A \subseteq S \).

According to this definition, if a Markov chain graph has set \( T \) that is an independent set, then no state transitions may occur between states in \( T \). Since we do not consider the diagonal elements of the stochastic matrix when defining the Markov chain graph, the diagonal elements \( p_{ii} \geq 0 \) for \( i \in T \). An example of independent set and vertex cover is shown in Figure 1.

We now show how to compute the value function by exploiting such partition of the state space. In what follows we denote by \( G = (S, E, P) \) an arbitrary Markov chain graph, and by \( \hat{G} = (S, E, \hat{Q}) \) a Markov chain graph on the same set of states, that has an independent set \( T \subset S \), so that \( q_{ij} = 0 \) for \( i, j \in T, i \neq j \). Let \( C = S \setminus T \) be the corresponding vertex cover. Let \( \hat{v} \) be the value function computed over such Markov chain, from (6), satisfies the system of equations

\[
\hat{v} = g + \gamma \hat{Q} \hat{v}.
\]

In particular, for \( i \in C \) we obtain

\[
\hat{v}_i = g_i + \gamma \sum_{j \in C} q_{ij} \hat{v}_j + \gamma \sum_{j \in T} q_{ij} \hat{v}_j.
\]

On the other hand, for \( i \in T \) we obtain

\[
\hat{v}_i = g_i + \gamma \sum_{j \in C} q_{ij} \hat{v}_j + \gamma \sum_{j \in C} q_{ij} \hat{v}_j,
\]

where we have used the fact that \( T \) is an independent set and thus \( q_{ij} = 0 \) for \( i, j \in T \) with \( i \neq j \). We can rewrite (11) as

\[
\hat{v}_i = g_i + \gamma \sum_{j \in C} q_{ij} \hat{v}_j.
\]

Then, substituting (12) in (10) we obtain, for \( i \in C \),

\[
\hat{v}_i = g_i + \gamma \sum_{j \in C} q_{ij} \hat{v}_j + \gamma \sum_{j \in T} q_{ij} \left[ g_j + \gamma \sum_{m \in C} q_{jm} \hat{v}_m \right] - \gamma q_{ij} \hat{v}_j
\]

\[
= g_i + \gamma \sum_{j \in T} q_{ij} \hat{v}_j
\]

\[
+ \gamma \sum_{j \in C} q_{ij} \left[ g_j + \gamma \sum_{m \in T} q_{jm} \hat{v}_m \right] - \gamma q_{ij} \hat{v}_j.
\]

We can rewrite it in matrix form as follows. Let \( Q_{C,T} \) be the sub-matrix of \( Q \) containing the transitions probabilities from states in \( C \) to states in \( C \). Let us denote \( Q_{T,C} \) and \( Q_{T,T} \) in the same way. Note that, since \( T \) is an independent set, \( Q_{T,T} \) is a diagonal matrix with the transition probabilities \( q_{i,i}, i \in T \) along its diagonal. Let \( g_C \) and \( g_T \) be the reward vectors in the cover set \( C \) and in the independent set \( T \) respectively; and let \( \hat{v}_C \) and \( \hat{v}_T \) be the value function restricted to \( C \) and \( T \) in the same way. Then, we can rewrite (12) and (13) as

\[
\hat{v}_T = (I - \gamma Q_{T,T})^{-1} [g_T + \gamma Q_{T,C} \hat{v}_C],
\]

and

\[
\hat{v}_C = g_C + \gamma Q_{C,T} (I - \gamma Q_{T,T})^{-1} g_T
\]

\[
+ \gamma Q_{C,C} (I - \gamma Q_{T,T})^{-1} Q_{T,C} \hat{v}_C.
\]

By solving (15), we finally obtain

\[
\hat{v}_C = \left( I - \gamma \left[ Q_{C,C} + \gamma Q_{C,T} (I - \gamma Q_{T,T})^{-1} Q_{T,C} \right] \right)^{-1}
\]

\[
\times [g_C + \gamma Q_{C,T} (I - \gamma Q_{T,T})^{-1} g_T].
\]

Note that, in order to compute \( \hat{v}_C \), we need to invert a matrix of size \( |C| \times |C| \), whose complexity scales as \( O(|C|^3) \). Thus, if the vertex cover \( C \) is such that \( |C| \ll n \), we obtain a computational advantage with respect to computing the value function over the entire state space. Once \( \hat{v}_C \) has been determined, the value function over the independent set, \( \hat{v}_T \), is obtained via interpolation of \( \hat{v}_C \) as in (14).

Alternatively, the value function \( \hat{v}_C \) can be determined with the following iterative algorithm. Starting from an initial
estimate $v^{[0]}_C$, we obtain further refinements via the value iteration updates
\[
v^{[i]}_C = b + A v^{[i-1]}_C,
\]
where we define
\[
A \triangleq \gamma \left[ Q_{C, C} + \gamma Q_{C, T} (I - \gamma Q_{T, T})^{-1} Q_{T, C} \right]
\]
\[
b \triangleq g_C + \gamma Q_{C, T} (I - \gamma Q_{T, T})^{-1} g_T
\]
and therefore using the definition of spectral norm, the $\ell_2$ relative error can be bounded as
\[
\| \hat{v} - v \|_2 \leq \gamma \| (I - \gamma Q) \|_2 \| P - Q \|_2 \| v \|_2.
\]

If $|C| \ll n$, these updates can be done with complexity $O(|C|^2)$ per iteration, much lower than that of running the value iteration algorithm over the original state space ($O(n^2)$), and computing $A$ and $B$ is $O(|T||C|^2)$ (done only once).

For convergence, we need to make sure that the spectral radius of $A$ is less than 1. This is proved in the following lemma.

**Lemma 1.** The spectral radius of $A$, denoted as $\rho(A)$, is such that $\rho(A) < 1$.

**Proof.** Let, $\forall i \in C$,
\[
C_i \triangleq A_{i,i},
\]
\[
R_i \triangleq \sum_{j \in C, j \neq i} |A_{i,j}|.
\]
Then, from Gershgorin circle theorem [16], we obtain
\[
\min_{i \in C} \{C_i - R_i\} \leq \rho(A) \leq \max_{i \in C} \{C_i + R_i\}.
\]

It is thus sufficient to prove that $C_i + R_i < 1$ and $C_i - R_i > -1, \forall i \in C$. From (18) we have
\[
C_i = \gamma q_{i,i} + \gamma^2 \sum_{j \in T} q_{i,j} q_{j,i},
\]
\[
R_i = \gamma - \gamma \sum_{j \in T} \left( \frac{1 - \gamma}{1 - \gamma q_{j,j}} - C_i \right).
\]
Then, using the fact that $\gamma < 1$, we obtain
\[
C_i + R_i = \gamma - \gamma \sum_{j \in T} \left( \frac{1 - \gamma}{1 - \gamma q_{j,j}} \right) \leq \gamma < 1,
\]
and, using the fact that $C_i \geq 0$, $C_i - R_i \geq -(C_i + R_i) > -1$.

The true value function is given by (7), whereas the approximate one, calculated under the transition probability matrix $Q$, is given by the solution of the system of equations (15) in the cover set $C$, and by (14) in the independent set $T$. Alternatively, using (7) with $Q$ in place of $P$, we obtain the approximate value function
\[
\hat{v} = (I - \gamma Q)^{-1} g.
\]

Then, the approximation error is
\[
\hat{v} - v = (I - \gamma Q)^{-1} g - (I - \gamma P)^{-1} g
\]
\[= \gamma (I - \gamma Q)^{-1} (Q - P) (I - \gamma P)^{-1} g
\]
\[= \gamma (I - \gamma Q)^{-1} (Q - P) v,
\]

and therefore using the definition of spectral norm, the $\ell_2$ relative error can be bounded as
\[
\| \hat{v} - v \|_2 \leq \gamma \| (I - \gamma Q) \|_2 \| P - Q \|_2 \| v \|_2.
\]

It can be shown that for any stochastic matrix $Q$, we have
\[
\| (I - \gamma Q) \|_2 \leq \frac{1}{1 - \gamma},
\]
for any $0 \leq \gamma < 1$. Then the relative error is bounded by
\[
\frac{\| \hat{v} - v \|_2}{\| v \|_2} \leq \frac{\gamma}{1 - \gamma} \| P - Q \|_2.
\]

Interestingly, this bound is independent of the reward vector $g$, and only depends on the $\ell_2$ error between $P$ and $Q$, and on the discount factor $\gamma$. Also, this bound is naive in the sense that it does not reflect the dependence between $\| P - Q \|_2$ and $\gamma$. We take advantage of that and only focus on reducing the matrix approximation error independently of the value of $\gamma$.

**IV. MARKOV CHAIN APPROXIMATION USING INDEPENDENT SET**

In general, a Markov chain with transition probability matrix $P$ and graph representation $G = (S, E, P)$ may not contain an independent set $T$ large enough to make (16) computationally efficient. In this case, one may want to approximate the original Markov chain with another Markov chain with graph representation $\hat{G} = (\hat{S}, \hat{E}, \hat{Q})$, where $\hat{Q}$ approximates the original transition probability matrix $P$ and has an independent set sufficiently large to make (16) computationally efficient.

In this section, we study the following problem: given some $m < n$ (typically, $m \ll n$), what is the best way to approximate $P$ with a transition probability matrix $Q$, where $Q$ has a vertex cover $C$ of size $m$.

**A. Markov chain approximation with independent sets**

Let $P$ be a transition probability matrix. Let $\Phi(P, Q)$ be an approximation error function such that $\Phi(P, Q) = 0$ if $P = Q$, and $\Phi(P, Q) > 0$, otherwise. Then, the optimization problem is defined as
\[
Q^* = \arg\min_{Q \in \mathcal{V}_C \mid |C| \leq m} \Phi(P, Q),
\]
where $\mathcal{V}_C \equiv \{ Q : Q1 = 1, Q_{ij} \geq 0, q_{ij} = 0, \forall i \in T, \forall j \in T \setminus \{i\}, \}$, with $T \subseteq C^c$, is the set of transition probability matrices $Q$ with vertex cover set $C$, and the minimization is jointly over $Q \in \mathcal{V}_C$ and the cover set $C$ of size at most $m$. We can solve the joint optimization problem by first solving with respect to $Q \in \mathcal{V}_C$ for a fixed cover set $C$, and then optimize overall vertex covers $C$ of size $|C| \leq m$. For a fixed cover set $C$, the optimal stochastic matrix is the minimizer of
\[
\phi(C) \triangleq \min_{Q \in \mathcal{V}_C} \Phi(P, Q).
\]
Then, the optimal vertex cover set can be obtained by solving
\[
\min_{|C| \leq m} \phi(C).
\]

The solutions of (33) and (34) depend on the cost function $\Phi(\cdot)$ employed. Solving (34) is, in general, NP hard, thus we need to design an approximation algorithm. In the next sections, we
consider the KLD and the Frobenius norm as cost functions. Then, we will present an heuristic algorithm to approximate the solution of (34).

We define the KLD between stochastic matrices $P$ and $Q$ as follows.

**Definition 2** (KLD). Let $P, Q$ be irreducible stochastic matrices, $Q$ be absolutely continuous with respect to $P$ (i.e., $p_{ij} = 0 \Rightarrow q_{ij} = 0$), and $\pi$ be a non negative vector that sums to one. Then, the KLD between $\Phi$ is given by

$$D(Q||P) \triangleq \sum_{i \in S} \sum_{j \in S} \pi_i q_{ij} \log \left( \frac{q_{ij}}{p_{ij}} \right),$$

where we assume $0 \cdot \log(0) = 0$.

Note that, if $\pi$ is the stationary distribution of $Q$, then our definition coincides with the KLD rate [17].

Suppose $\pi$ and the vertex cover set $C$ are fixed and we want to optimize the non-zero entries of $Q$ according to (33), where $\Phi(P, Q) = D(Q||P)$. For a matrix $Q \in V_C$, we can write the KLD cost as

$$D(Q||P) = \sum_{i \in C} \sum_{j \in S} \pi_i q_{ij} \log \left( \frac{q_{ij}}{p_{ij}} \right) + \sum_{i \in T} \sum_{j \in C} \pi_i q_{ij} \log \left( \frac{q_{ij}}{p_{ij}} \right) + \sum_{i \in T} \sum_{j \in T} \pi_i q_{ij} \log \left( \frac{q_{ij}}{p_{ij}} \right),$$

where $T = C^c$ is the independent set and we have used its definition to set $q_{ij} \log \left( \frac{q_{ij}}{p_{ij}} \right) = 0$ for $i, j \in T$ with $i \neq j$. Since the domain of the log function is positive reals, we do not need to impose non negativity constraints on $Q$, then we only need to solve

$$\min_{Q1=1} = D(Q||P).$$

It can be shown that the optimal $Q$ has a closed form expression given by

$$q_{ij} = \begin{cases} p_{ij}, & \text{if } i \in C, j \in S, \\ \frac{p_{ij}}{p_{ii} + p_{ji}}, & \text{if } i \in T, j \in C, \\ \frac{p_{ij}}{p_{ii} + p_{ji}}, & \text{if } i \in T, j = i, \\ 0, & \text{otherwise}, \end{cases}$$

where we have defined $p_{ij} \triangleq \sum_{j \in C} p_{ij}$. Note that the optimal solution is equivalent to setting to zero the transition probabilities in the independent set $T$ (except for the transition probability from $i \in T$ to itself) and then normalizing the remaining non zero transitions so that $Q1 = 1$.

Since we have an optimal stochastic matrix for a given cover set $C$, we can update the KLD for that solution as follows. Plugging the optimal $Q$ into (36), we obtain

$$\phi(C) = -\sum_{i \in T} \pi_i \log(p_{ii} + p_{ii}).$$

The Frobenius norm of matrix $P$ is $\|P\|_F = \left( \sum_{i,j} p_{ij}^2 \right)^{1/2}$, then we minimize

$$\min_{Q \in V_C} \frac{\|P - Q\|_F}{\|P\|_F},$$

**Algorithm 1** Greedy Independent set optimization

1: for $i = 1$ to $n_{iter}$ do
2: pick randomly $s \in C$ and $t \in T$
3: $C \leftarrow C \setminus \{s\} \cup \{t\}$
4: if $\phi(C) < \phi(C)$ then
5: $C \leftarrow C$
6: return $C$

For simplicity do not impose non-negativity on the entries of $Q$. We will show that the solution is, in fact, non-negative. For a fixed cover set $C$ and independent set $T = C^c$, (39) is equivalent to

$$\min_{Q1=1} \frac{1}{\|P\|_F^2} \left[ \sum_{i \in C} \sum_{j \in S} (q_{ij} - p_{ij})^2 + \sum_{i \in T} \sum_{j \in C} (q_{ij} - p_{ij})^2 + \sum_{i \in T} \sum_{j \in T} p_{ij}^2 + \sum_{i \in T} (q_{ii} - p_{ii})^2 \right],$$

Where we have zeroed the entries $q_{ij}$ in the independent set and squared the objective function in (39). It can be shown that the optimal $Q$ has non negative entries

$$q_{ij} = \begin{cases} p_{ij}, & \text{if } i \in C, j \in S, \\ p_{ij} + \frac{p_{ci} - p_{ci}}{|C|+1}, & \text{if } i \in T, j \in C, \\ p_{ii} + \frac{p_{ci} - p_{ci}}{|C|+1}, & \text{if } i \in T, j = i, \\ 0, & \text{otherwise}. \end{cases}$$

Then, the error corresponding to the vertex cover set $C$, obtained by plugging the optimal solution into the Frobenius norm cost, is given by

$$\phi(C) = \left( \frac{1}{|C|+1} \sum_{i \in T} (1-p_{ci} - p_{ci})^2 + \sum_{i \in T} p_{ij}^2 \right)^{1/2}.$$

Since optimizing (34) is NP-hard, we propose a greedy algorithm. It starts with an arbitrary partition of the states $S = C \cup T$ with fixed size $|C| = m$; at each iteration, it randomly switches elements between $C$ and $T$ and updates the current sets if the cost $\phi(C)$ decreases. The pseudo-code is shown in Algorithm 1.

**V. Numerical Results**

We evaluate our methods in several physical networks with $N = 4$ users and different collision graphs. In a collision graph, simultaneous packet transmissions will fail between two connected users. We consider 4 collision graphs:

1) **Disconnected:** no edges between users,
2) **Broadcast:** 1 user is connected to all other users,
3) **Circular:** users are connected in a circular graph,
4) **Full:** all users are connected to each other.

Each user has a data queue of size $b = 3$, then the network state is the number of packets in each queue, of size $|S| = N^{b+1} = 256$. Data packets arrive from the upper layer with probability $p=0.3$ in each slot for each user, and are buffered in the corresponding queue. Data transmission occurs with
probability $q=p$ in each slot; retransmissions are allowed in case of transmission failure. As rewards we consider network throughput, collected each time slot after transmissions.

We simulate each network and compute its state transition matrix $P$ and average reward $g$. Then for each $P$ we run Algorithm 1 for KLD and Frobenius cost. We perform a comparison with the case where the vertex cover is selected at random. We repeat the experiment with 100 different random generations of the vertex cover for vertex cover relative sizes $\frac{|S|}{n} \in \{0.05, 0.1, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40\}$, and compute both the average KLD cost given by (38) and the Frobenius cost given by (40). We plot the results in Figure 2. We can see that the matrix approximation cost decreases as the vertex cover gets larger. Independently of the method to choose the vertex cover (random or greedy) and the metric employed, for any given value of $\frac{|S|}{n}$ the physical networks that have more connected collision graphs have lower cost. Ordering the networks by number of edges in the collision graph Disconnecte, Broadcast, Circular, Full is the same as ordering them by decreasing cost function. Finally, the greedy vertex cover selection method always performs better than the random method.

In Figure 3, we show the average relative error in value function approximation $\|v - \hat{v}\|_2/\|v\|_2$ for the circular and full networks. Again, in each plot we compare Frobenius and KLD cost function with random and greedy vertex cover selections. The greedy selection is consistently better than random vertex cover selection method. For the full network, the Frobenius cost is better than KLD cost for all ratios $\frac{|S|}{n}$, and, as the vertex cover size increases, all methods achieve a smaller error. For the circular network topology, Frobenius is better than KLD cost only for vertex cover sets of smaller size. In general, all methods have relative errors below 10%.

In Figure 4, some examples of true and approximated value functions are shown for vertex cover sizes of 20% of the number of states. We compare the greedy KLD and Frobenius methods for the Broadcast and Circular physical networks. For the circular network, we showed in Figure 3 that for vertex cover of size of around 20% of the size of the state space,
the greedy KLD and Frobenius methods have similar error, and, in Figure 4, we observe that the errors are of different types. For the KLD case, the approximate value function is more prone to spikes, while for the Frobenius case, the error is more spread out.

VI. CONCLUSION

In this paper, we presented a framework for reducing the dimension of the value iteration problem by sparsifying the state transition network. Specifically, our method involves approximating the original Markov chain by a sparser one whose state transition graph contains an independent set, i.e., set of states with no transitions between them, which allows to significantly reduce the complexity of evaluating the value function. We have considered both the Frobenius norm and the KL divergence metrics between the corresponding probability transition matrices, and we have proposed a greedy algorithm to determine an approximately optimal independent set. Numerical results for different classes of collision networks show that our approximation is accurate, even for a small vertex cover.

REFERENCES


Fig. 4: Plots of value functions for broadcast and circular collision networks