A Factored MDP Approach to Optimal Mechanism Design for Resilient Large-Scale Interdependent Critical Infrastructures

Linan Huang, Juntao Chen, and Quanyan Zhu
Department of Electrical and Computer Engineering, Tandon School of Engineering
New York University, Brooklyn, NY 11201 USA
E-mail:{lh2328, jc6412, qz494}@nyu.edu

***Abstract*—Enhancing the security and resilience of interdependent infrastructures is crucial. In this paper, we establish a theoretical framework based on Markov decision processes (MDPs) to design optimal resiliency mechanisms for interdependent infrastructures. We use MDPs to capture the dynamics of the failure of constituent components of an infrastructure and their cyber-physical dependencies. Factored MDPs and approximate linear programming are adopted for an exponentially growing dimension of both state and action spaces. Under our approximation scheme, the optimally distributed policy is equivalent to the centralized one. Finally, case studies in a large-scale interdependent system demonstrate the effectiveness of the control strategy to enhance the network resilience to cascading failures.**

I. Introduction

With the development of Internet of Things (IoT), the physical world becomes increasingly connected due to the communication needs and cyber-physical reliances, among which the critical infrastructures (CIs) are fundamental and indispensable [1]. These strong connections and reliances make CIs interdependent, which on the one hand, enhance the system efficiency, yet on the other, make infrastructures valuable to faults and attacks. Cyber and mechanical outages in one component will affect others and can magnify to cause the cascading failures. During the Hurricane Sandy, failures inside the power grids led to a large-size blackout, and then the power outage propagated negatively to the dependent infrastructures, e.g., transportation and communications, which finally became a disaster causing a huge economic loss.

Therefore, our goal lies in enhancing the security and resilience of the interdependent infrastructures. To achieve this goal, we establish our model based on the following considerations.

1) Connectivity: The physical components and dependencies are represented by nodes and links in a network.
2) Resilience: A dynamic model is adopted to show how components recover with control policy as time evolves.
3) Stochastic dynamics: A probabilistic state transition scheme captures the randomness of the network.
4) Control policy: A decision model provides the optimal strategy to enhance the system performance.

Markov decision process (MDP) is an appropriate model to capture the four characteristics of the framework. Specifically, the state of the MDPs is a global combination of all the nodes' local states. A representation problem emerges as the system state grows exponentially with the number of nodes. To address this challenge, we use factored MDPs (FMDP) [2] and dynamic Bayesian network (DBN) [3] to reduce the complexity of the model with node dependencies. Approximate linear programming (ALP) is used to solve the MDP. The policy obtained from FMDP gives an optimal resilience mechanism for massive interdependent critical infrastructures.

A number of works have focused on understanding infrastructure interdependencies through concept identification [4], dependency classification [5], and model construction [6]. In [7], [8], game-theoretic models have been proposed to guide the design of the interdependent network. Cascading risks quantification in a power grid has been investigated in [9]. All these works successfully illustrate, analyze and simulate the interdependency of CIs. We contribute by adopting FMDP [2] to mathematically describe the network and reduce the computational complexity. Also, in contrast with [10], we consider restoration processes of the network during the system state transitions.

In this paper, we use the power and transit networks as a case study to illustrate the generic framework illustrated above. This application will be especially important for the urban area as it boasts a higher population density, more complex interdependent CIs, and a greater economic loss when cascading occurs.

The main contributions of this paper are summarized as follows.

1) We construct a real-time, extensible network model where the node failure probability is related to the physical factors including the capacity threshold, supply-demand relationship, and load shedding.
2) We adopt the FMDP, DBN, and ALP to tackle the curse of dimension for large-scale interdependent networks. We analyze two recovery scenarios and propose distributed policies that are equivalent to centralized ones.
3) We use numerical experiments to study the impact of topology and system parameters on the threshold policy, long-term benefits, and the risk avoidance. As a high dependency reduces the network resilience, our control strategy mitigates the adverse effect.

This research is partially supported by NSF grants EFRI-1441140, SES-1541164 and the DOE grant DE-FOA-0001281.

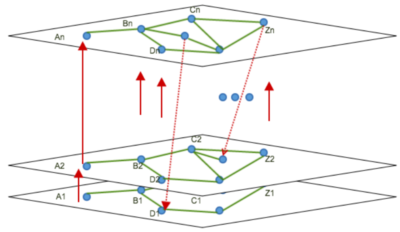

Fig. 1. The interdependent structure within and between different infrastructure networks is distinguished by solid green and red lines, respectively.

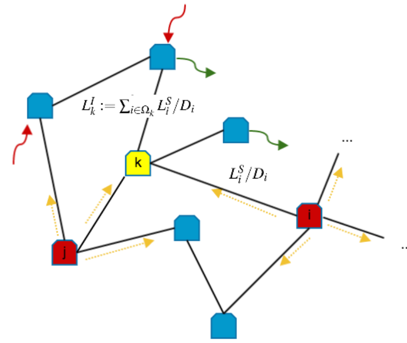

Fig. 2. The power injection of generator i, S_i^{in}, is denoted by a red arrow, and the local load of bus i, S_i^{out}, is denoted by a green arrow. The net power generation is $L_i^R = S_i^{in} - S_i^{out}$. When buses i and j are down, their extra loads flush to the normal neighboring buses, e.g., bus k. Then, the load of node k is increased by $L_k^I = L_i^S/D_i + L_j^S/D_j$. Other normal buses denoted by blue that are connected to k have a zero extra load increase.

A. Notations

Throughout the paper, we use calligraphic letter $\mathscr{X}$ to define the set. The upper case letter X denotes a random variable and the lower case x represents its value. The boldface $\mathbf{X}$ denotes a vector of random variables and $\mathbf{x}$ for their values. $|A|$ denotes the cardinality of set A. In addition, we define the binary set $\mathscr{B} := \{0, 1\}$.

B. Organization of the Paper

The rest of the paper is organized as follows. Section II constructs a probabilistic failing model to describe the cascading effect. Section III formulates the problem in FMDP. Section IV solves the problem using ALP and describes the centralized and distributed optimal policy. Case studies are given in Section V, and Section VI concludes the paper.

II. Interdependent Networks Model

A general multi-layer interdependent network is illustrated in Fig. 1. We use the interdependencies between power grid and the subway transit systems as a running case study in this paper. Geographical and logical dependencies cause the interdependencies between these two layers. On the one hand, a subway station will be closed due to a power outage. On the other, the out-of-service of the subway will delay the repair of the failing bus in the power grid, thus increasing the failure probability of neighboring buses. This interdependency is illustrated in Fig. 1. Between the top two layers, a solid red link denotes a dependency from power to the subway, and a dotted red line denotes the reverse dependency.

By abstracting the system components and dependencies as nodes and links, respectively, we can view this two-layer coupled heterogeneous network as a single-layer network. The dependencies within and between layers are of different types, yet they can be generically captured using transition probabilities in an MDP framework. Hence the features of individual systems are captured by the construction of state transition probabilities. In Section II-A, we use electric power grid as an example to demonstrate construction details inside the single-layer network model.

A. Cascading Construction of Power Grid

Consider an n-bus power system where each bus i, $i \in \mathscr{N} := \{1, 2, ..., n\}$, is connected with a generator, e.g., the smart city where each building has its solar energy production, and buildings are connected for power transmission and delivery. Modern power networks have been designed with redundancies in case of anticipated failures. Therefore, bus $i \in \mathscr{N}$ has a regular net power generation of L_i^R under its certified capability $\bar{L}_i$. When there is a link from node i to j, we call node i the parent node of node j and the child node vice verse. In a bi-directional graph as in the power grid, we also say i and j are neighboring nodes.

Once one bus in the network fails due to random disturbance, it will cease to generate power. Then, its neighboring connected buses have to generate more power to compensate for the supply loss. Focusing on the generator breakdown failures [11], we assume that the transmission lines work normally. In addition, we assume that the same amount of time is given for neighboring buses to generate the extra power and they have the same generating efficiency. Then, the load L_i^S of the failure bus i is equally distributed to its neighboring buses in i's open neighborhood Ω_i. This assumption can be lifted by considering power flow constraints. Therefore, each bus k, $k \in \Omega_i$, has an L_i^S/D_i amount of load increase, where D_i is the degree of bus i. As illustrated in Fig. 2, bus k's total load increase can be calculated as $L_k^I := \sum_{i \in \Omega_k} L_i^S/D_i$ by taking all its failed neighboring nodes into account.

One interesting scenario is that two neighboring nodes are both down at time t due to the cascading failure. For example, nodes i and k are down at time t, and the healthy node j takes the extra load from k. Then, if j is down at time $t+1$, j is not able to get energy from k as k is down. Furthermore, j cannot provide the extra energy to k. Therefore, both nodes j and k have a demand larger than the supply, which leads to the load shedding.

With the introduction of load shedding, the load compensation of each node i will be limited to Ω_i and will not propagate to other non-neighboring nodes. Therefore, the dependency is limited to nodes in Ω_i, and the DBN and FMDP can be applied in Section III-B. In addition, the load L_i of each node i satisfies Markov property, i.e., it only depends on the current system state $\mathbf{x}$ to be discussed in Section III-A.

B. Probabilistic Model

Different from CASCADE [12] which assumes a deterministic failure threshold of each node, we adopt a probabilistic metric to model the node failure. Node i's load $L_i = L_i^R + L_i^I$ affects its failure probability P_i. Function $f : \mathbb{R}^+ \mapsto [0,1]$ used to quantify the failure probability $P_i = f(L_i)$ should satisfy the following requirements.

- Each node i possesses a small prior failure probability $\bar{P}_i$ under the normal condition, which is related to human errors, device malfunctions and malicious attacks.
- The rated capacity $\bar{L}_i$ serves as a threshold. The failure probability will not be affected remarkably by the load increase until the increase reaches a threshold, and then the failure probability grows exponentially afterwards.
- When $\bar{L}_i > 0$, the probability should quickly increase as loads grow, and the function f should be monotone.
- The function f has an upper bound of 1, and $\lim_{L_i \to \infty} f(L_i) = 1$.

The exact function f can vary for different applications. In this work, we let f be the sigmoid function as follows.

$$f(L_i) = \begin{cases} \bar{P}_i, & L_i \leq \bar{L}_i, \\ \frac{1}{1+(1/\bar{P}_i - 1)e^{-(L_i - \bar{L}_i)}}, & L_i > \bar{L}_i. \end{cases} \quad (1)$$

III. Problem Formulation

The physical characteristics of the power system have been modeled by the probability of failures in Section II-B. We can capture the dependencies of other infrastructure networks using a similar approach. Thus, we formulate the problem in a general network model as follows.

A. Factored MDPs

Recall that $\mathscr{B} := \{0,1\}$. At *stage* $t \in \{0,1,2,...\}$, the binary random variable $X_i^t \in \mathscr{B}$ denotes the *local state* of the node $i \in \mathscr{N}$ at stage t. Its value $x_i = 1$ if node i works normally and $x_i = 0$ otherwise. Similarly, the binary random variable $A_i^t \in \mathscr{B}$ is the local action of node i at stage t. Let $a_i = 1$ if the node is repaired and $a_i = 0$ otherwise. Repair can also refer to maintenance which can prevent the node failure at next stage if this node is currently working.

To study the n-node system, we define *system state* $\mathbf{X}^t$ as a vector of binary random variables, i.e., $\mathbf{X}^t := [X_1^t, X_2^t, ..., X_n^t]$. All possible values of $\mathbf{X}^t$ form a finite set $\mathscr{X}$, with $|\mathscr{X}| = N := 2^n$. The action of global network $\mathbf{A}^t = [A_1^t, A_2^t, ..., A_n^t]$ is also a vector of binary random variables with $\mathbf{A}^t \in \mathscr{A}$, where $\mathscr{A}$ is a finite set of *actions*; A stationary policy π is a mapping from the current state to the action set independent of time, $\pi : \mathscr{X} \mapsto \mathscr{A}$. Let $R : \mathscr{X} \times \mathscr{A} \mapsto \mathbb{R}$ be the reward function. $R(\mathbf{x}, \mathbf{a})$ shows the benefit of taking action $\mathbf{a}$ at state $\mathbf{x}$. P is the *transition probability* from state $\mathbf{x}$ to $\mathbf{x}'$ with action $\mathbf{a}$, i.e., $P(\mathbf{x}'|\mathbf{x}, \mathbf{a}) = \mathbb{P}\{\mathbf{X}^{t+1} = \mathbf{x}' | \mathbf{X}^t = \mathbf{x}, \mathbf{A}^t = \mathbf{a}\}$, $\forall t$.

B. Dynamic Bayesian Network of Transition Probability

Note that the total number of system state $N = 2^n$ grows exponentially with the number of nodes, which restricts the application of the large-scale system. The FMDP makes it possible to factorize the transition probability $P(\mathbf{x}'|\mathbf{x})$ in the form of $\prod_{i \in \mathscr{N}} P(x_i'|\mathbf{x}) := \prod_{i \in \mathscr{N}} P(X_i^{t+1} = x_i'|\mathbf{x})$. Because the transition probability is defined between a multiplication of the local state conditional probability rather than two global states, a DBN can be introduced as a two-layer directed acyclic graph with a set of binary states $\{X_1, ..., X_n, X_1', ..., X_n'\}$. Let $\mathbf{x}_{\Omega_i} \in \mathscr{B}^{|\Omega_i|}$ be the value of the vector of all the random variables X_i, where i belongs to the set Ω_i, e.g., $\mathbf{x}_{\Omega_i} = [x_i]_{i \in \Omega_i}$. The local state of node i at the next stage is only affected by the state of its parent node $j \in \Omega_i$. In a sparse network, the transition probability can be further simplified as $\prod_{i \in \mathscr{N}} P(x_i'|\mathbf{x}) = \prod_{i \in \mathscr{N}} P(x_i'|\mathbf{x}_{\Omega_i})$. Notice that although the interdependency between nodes makes the transition probability bi-directional in Fig. 2, the DBN graph is still directed and acyclic.

Our objective is to optimize the accumulated reward of the system evolving over an infinite horizon. Then, the cost function starting at different state $\mathbf{x}$ with policy π is

$$V_\pi(\mathbf{x}) = \mathbb{E}_\pi [\sum_{t=0}^{\infty} \gamma^t R\left(\mathbf{X}^t, \pi(\mathbf{X}^t)\right) | \mathbf{X}^0 = \mathbf{x}], \quad (2)$$

where $\gamma \in [0,1]$ is the *discount factor* that shows a discounted value of future rewards. The value function $V(\mathbf{x})$ is the optimal cost function over all the feasible policies $V(\mathbf{x}) = \max_\pi V_\pi(\mathbf{x})$. Then, using the dynamic programming [13], we can rewrite (2) as

$$T_\pi V(\mathbf{x}) = Q(\mathbf{x}, \mathbf{a}) := R(\mathbf{x}, \mathbf{a}) + \gamma \sum_{\mathbf{x}' \in \mathscr{X}} P(\mathbf{x}'|\mathbf{x}, \mathbf{a}) V(\mathbf{x}'), \quad (3)$$

where T_π is the DP operator and $Q(\mathbf{x}, \mathbf{a})$ is the expected value obtained at stage $\mathbf{x}$ with action $\mathbf{a}$.

C. Reward and Penalty Functions

Define $\bar{R}_i(x_i)$ as the local reward function of node i at state x_i, and $C_i(a_i)$ as its local cost function of taking action a_i. Formally, $\bar{R}_i : \mathscr{B} \mapsto \mathbb{R}$ and $C_i : \mathscr{B} \mapsto \mathbb{R}$. A penalty is added for the load shedding. As presented in Section II-A, the load shedding is limited to node i and Ω_i. Denote the penalty function of node i by $F_i : \mathscr{B}^{|\Omega_i|+1} \mapsto \mathbb{R}$. Different penalty functions can be designed, and one of them is $F_i(x_i, \mathbf{x}_{\Omega_i}) = \frac{\beta}{D_i} \times \sum_{j \in \Omega_i} h_j(x_j)$, where $h_i(x_i)$ is an indicator function, i.e., for all i, $h_i(x_i) = 1$ when $x_i = 1$ and $h_i(x_i) = 0$ when $x_i = 0$. Note that β is a weighting constant. We define $R_i(x_i, \mathbf{x}_{\Omega_i}) := \bar{R}_i(x_i) - F(x_i, \mathbf{x}_{\Omega_i})$. The system reward $R(\mathbf{x}, \mathbf{a})$ can be first decoupled into two factored functions which are only related to vectors $\mathbf{x}$ and $\mathbf{a}$, respectively. Then, each function has a limited domain that relates only to part of elements of the vectors $\mathbf{x}$ and $\mathbf{a}$. Therefore, the final form of $R(\mathbf{x}, \mathbf{a})$ is written as $R(\mathbf{x}, \mathbf{a}) = \sum_{i=1}^n K_i \times [R_i(x_i, \mathbf{x}_{\Omega_i}) - C_i(a_i)]$, where K_i captures the importance of each node $i, i \in \mathscr{N}$.

IV. Problem Analysis and Resilient Mechanism Design for Interdependent Networks

One typical approach to address the MDPs is by exact or approximate linear programming (LP) [14]. We first focus on the scenario of repairing only one node at each stage in Section IV-C. The resulting new restricted action set $\mathscr{A}'$ contains n possible elements of $\mathbf{a}$. This setting is reasonable, e.g., when one repairing team moves between nodes at each stage to accomplish the work. Hence, multiple repairs cannot be processed simultaneously. We further assume that no cost is associated with the action, i.e., $C_i(a_i) = 0,\ \forall i \in \mathscr{N}$. With a tractable action space, we first solve the dimension problem of state space.

In Section IV-D, we generalize the model to the scenario that multiple repairs can be taken at the same stage with costs. Without loss of generality, a distributed policy is designed to address the curse of dimensionality of the action space. For consistency, we clarify three types of reward as follows:

1) The local reward is denoted by $R(\mathbf{x},\mathbf{a})$.
2) The global reward is the value function $V(\mathbf{x})$.
3) The system reward or total reward m is the objective function of the LP, e.g., $m = \sum_{\mathbf{x}\in\mathscr{X}} \alpha(\mathbf{x})V(\mathbf{x})$, where $\alpha(\mathbf{x}) > 0$ be the state-depending weighting function and $\sum_{\mathbf{x}\in\mathscr{X}} \alpha(\mathbf{x}) = 1$.

Resilience can be quantified by the cost and time a system takes to transit from a faulty state to normal. The global reward captures both two indices, i.e., cost and time after solving the MDPs problem. The total reward sum over all global rewards to reflect the effect of distinct initial states. Therefore, we use the global reward to represent the resilience starting at different initial states while taking the total reward as the system resilience measure.

A. Exact Linear Programming

The variables in LP are the global rewards $V(\mathbf{x}),\ \forall \mathbf{x} \in \mathscr{X}$. We select the state dependent weights to be uniformly distributed. Note that the action set is restricted to $\mathscr{A}' \subset \mathscr{A}$ with action constraints. Then, the exact LP is

$$\begin{aligned} \min_{V(\mathbf{x})}: \quad & \sum_{\mathbf{x}\in\mathscr{X}} \alpha(\mathbf{x})V(\mathbf{x}) \\ \text{s.t.} \quad & V(\mathbf{x}) \geq R(\mathbf{x},\mathbf{a}) + \gamma \sum_{\mathbf{x}'\in\mathscr{X}} P(\mathbf{x}'|\mathbf{x},\mathbf{a})V(\mathbf{x}), \\ & \forall \mathbf{x} \in \mathscr{X},\ \forall \mathbf{a} \in \mathscr{A}'. \end{aligned} \tag{4}$$

First, because we have unified multiple networks into one single network with more nodes in Section II, node n will be large in a multi-layer network application. Second, the number of system state is a vector whose value grows exponentially as $N = 2^n$. Thus we have exponentially many LP variables, i.e., $V(\mathbf{x})$. Finally, the amount of constraints in (4) is exponential. These three factors lead to the curse of dimension in state space, which are solved by ALP and variable elimination in the following part IV-B.

B. Approximate Linear Programming (ALP)

First, we present the definition of linear value functions.

Definition 1 (Linear Value Function)**.** *A linear value function over basis functions* $H = \{h_1(\cdot),\dots,h_k(\cdot)\}$ *is a function* $V(\mathbf{x}) = \sum_{j=1}^{k} w_j h_j(\mathbf{x})$ *for some coefficients* $\mathbf{w} = (w_1,\dots,w_k)'$.

Therefore, the global reward can be represented compactly as $V \in \mathscr{H}$, where $\mathscr{H}$ is the linear subspace of $\mathbb{R}^N$ spanned by H. In our setting, we keep each *basis element* h_i the same as the indicator function in III-C.

Using the linear value function, the ALP with the variables $w_1,\dots,w_k$ is

$$\begin{aligned} (\text{ALP}):\ \min_{\mathbf{w}} \ & \sum_{\mathbf{x}\in\mathscr{X}} \alpha(\mathbf{x}) \sum_{i\in\mathscr{N}} w_i h_i(\mathbf{x}) \\ \text{s.t.} \ & \sum_{i\in\mathscr{N}} w_i h_i(\mathbf{x}) \geq R(\mathbf{x},\mathbf{a}) + \gamma \sum_{\mathbf{x}'\in\mathscr{X}} P(\mathbf{x}'|\mathbf{x},\mathbf{a}) \sum_{i\in\mathscr{N}} w_i h_i(\mathbf{x}'), \\ & \forall \mathbf{x} \in \mathscr{X},\ \forall \mathbf{a} \in \mathscr{A}'. \end{aligned}$$

This ALP reduces the number of free variables in the exact LP to k. However, the number of constraints is still $N \times |\mathscr{A}'| = 2^n \times |\mathscr{A}'|$. Based on the method introduced in Sections 4 and 5 in [15], we rewrite the constraints in ALP by defining $G(\mathbf{x},\mathbf{a}) = \sum_i w_i\ g_i(\mathbf{x},\mathbf{a})$, where $g_i(\mathbf{x},\mathbf{a}) = \sum_{\mathbf{x}'} P(\mathbf{x}'|\mathbf{x},\mathbf{a})h_i(\mathbf{x}')$ is the expectation of each basis element. Because the basis elements of restricted domain constrain the expectation's domain, we can simplify $g_i(\mathbf{x},\mathbf{a})$ as

$$\begin{aligned} g_i(\mathbf{x},\mathbf{a}) &= \sum_{\mathbf{x}'\in\mathscr{X}} P(\mathbf{x}'|\mathbf{x},\mathbf{a})h_i(x_i') = \sum_{i\in\mathscr{N}} \prod_{j\in\mathscr{N}} P(x_j'|x_j,\mathbf{x}_{\Omega_j},\mathbf{a})h_i(x_i') \\ &= P(X_i^{t+1} = 1|x_i,\mathbf{x}_{\Omega_i},\mathbf{a}). \end{aligned}$$

Denote $\bar{g}_i(\mathbf{x},\mathbf{a}) := \gamma g_i(\mathbf{x},\mathbf{a}) - h_i(\mathbf{x})$. The constraints of ALP can be written as

$$0 \geq \max_{\mathbf{x}\in\mathscr{X}}\ R(\mathbf{x},\mathbf{a}) + \sum_{i\in\mathscr{N}} w_i\ \bar{g}_i(\mathbf{x},\mathbf{a}), \quad \forall \mathbf{a} \in \mathscr{A}'. \tag{5}$$

For each feasible action, (5) is equivalent to

$$0 \geq \max_{i\in\mathscr{N}}\ \sum_i R_i(x_i,\mathbf{x}_{\Omega_i}) + \sum_i w_i\ \bar{g}_i(x_i,\mathbf{x}_{\Omega_i},\mathbf{a}). \tag{6}$$

The essence of the variable elimination is that instead of maximizing over all possible $\mathbf{x} \in \mathscr{X}$ in (5), which grows exponentially with the number of nodes, we maximize over one local state x_i and introduce a new LP variable depending on the remaining local state variables at each step. In this way, we still achieve the global optimality while the computational complexity reduces to polynomial. For example, suppose that we eliminate x_k first. Then, (6) can be rewritten as

$$\begin{aligned} 0 \geq \max_{x_{i,i\neq k,i,k\in\mathscr{N}}} \ & \sum_{i\neq k} R_i(x_i,\mathbf{x}_{\Omega_i}) + \sum_{i\neq k} w_i\ \bar{g}_i(x_i,\mathbf{x}_{\Omega_i},\mathbf{a}) \\ & + \max_{x_k}\ R_k(x_k,\mathbf{x}_{\Omega_i}) + \bar{g}_k(x_k,\mathbf{x}_{\Omega_k},\mathbf{a}). \end{aligned} \tag{7}$$

The term $\max_{x_k}\ R_k(x_k,\mathbf{x}_{\Omega_i}) + \bar{g}_k(x_k,\mathbf{x}_{\Omega_k})$ can be replaced by a group of linear constraints with a new LP variable $e_k(\mathbf{x}_{\Omega_k})$ introduced. Remind that Ω_k is a set containing all the parent nodes of k, and $|\mathbf{x}_{\Omega_k}| = 2^{|\Omega_k|}$. For each value of $\mathbf{x}_{\Omega_k}$, it satisfies

$$\begin{aligned} e_k(\mathbf{x}_{\Omega_k}) &\geq R_k(X_k^t = 1,\mathbf{x}_{\Omega_k}) + \bar{g}_k(X_k^t = 1,\mathbf{x}_{\Omega_k},\mathbf{a}), \forall \mathbf{x}_{\Omega_k} \in \mathscr{B}^{|\Omega_i|}, \\ e_k(x_{\Omega_k}) &\geq R_k(X_k^t = 0,\mathbf{x}_{\Omega_k}) + \bar{g}_k(X_k^t = 0,\mathbf{x}_{\Omega_k},\mathbf{a}), \forall \mathbf{x}_{\Omega_k} \in \mathscr{B}^{|\Omega_i|}. \end{aligned}$$

Therefore, at each step, we add $2^{|\Omega_k|+1}$ new LP constraints and eliminate one variable x_k for this example. We repeat the above process until the last variable is eliminated.

C. Control Policy with a Single-Repair Scenario

After figuring out the coefficient $\mathbf{w}$, we obtain the global reward $V(\mathbf{x})$. Then, we use the following greedy search method of Q function to find the optimal policy:

$$\begin{aligned}\mathbf{a}^* &\in \arg\max_{\mathbf{a}\in\mathscr{A}'} Q(\mathbf{x},\mathbf{a}) \\ &= \arg\max_{\mathbf{a}\in\mathscr{A}'} [R(\mathbf{x},\mathbf{a}) + \gamma \sum_{\mathbf{x}'\in\mathscr{X}} P(\mathbf{x}'|\mathbf{x},\mathbf{a})V(\mathbf{x}')].\end{aligned} \tag{8}$$

Note that the linear value function approximation can still be used here to limit the domain of the transition probability. Recall that the control policy is a mapping from the state space to the action space. Thus, we need to traverse the entire state space which explodes exponentially with the increase of the number of nodes. For each state, we test all possible actions to find the optimal one with the computational complexity $|\mathscr{X}| \times |\mathscr{A}'|$. For the single repair scenario, the action space is relatively small. Each action corresponds to a bunch of states due to a specific pattern: the most important node should be repaired when it fails regardless of other nodes' state. Therefore, all states $\mathbf{x}$ with $x_i = 0$ should correspond to the action $a_i = 1$, if node i is the most important. In the following theorem, we prove that w_i can be used to determine the importance of the node and hence decide the control policy.

Theorem 1. *Assuming the elements in $\mathbf{w}$ satisfy $w_i > w_j > w_n > ... > w_m > w_k$, where $i, j, ..., m, k \in \mathscr{N}$. Let P_i^{dep} be the probability of node i working normally when its parent nodes are down, and $P_i^{dep} = P_d$, $\forall i \in \mathscr{N}$. In a directed ring topology, the action priority, i.e., the order of node repairing when multiple outages occur, depends on the magnitude of the elements in $\mathbf{w}$ if conditions in (9) are satisfied.*

$$\begin{cases} (1-P_d)\cdot w_i > w_j, \\ (1-P_d)\cdot w_j > w_n, \\ \quad ... \\ (1-P_d)\cdot w_m > w_k. \end{cases} \tag{9}$$

Sketch of the Proof: We compute the objective function of the greedy search $Q(\mathbf{x},\mathbf{a})$ under different categories of states including conditions of single and multiple outages to summarize the pattern. We need the group of constraints (9) to avoid extreme cases when node i produces much less reward comparing to node j, i.e., $w_j \gg w_i$. Hence, we prefer to repair a working node j to prevent it from an outage in the future.

D. Generalized Control Policy with Multiple Repairs

With the development of IoT technologies and smart cities, each node can be equipped with self-reconfigurable and self-healing capabilities. Therefore each node can take local actions to repair itself. We generalize the previous constrained action space $\mathbf{a} \in \mathscr{A}'$ to an exponentially-large dimension $\mathbf{a} \in \mathscr{A}$ by removing constraints, where $|\mathscr{A}'| = n$ and $|\mathscr{A}| = 2^n$. Therefore, in this scenario, the total computational complexity of greedy search is $|\mathscr{X}| \times |\mathscr{A}| = 2^{2n}$, which is computationally complex for even a small-scale problem. Moreover, it also creates difficulties in solving the ALP. To address this challenge, we use the same variable elimination method for the factorization term concerning $\mathbf{a}$. Specifically, the expectation of each basis function can be simplified as

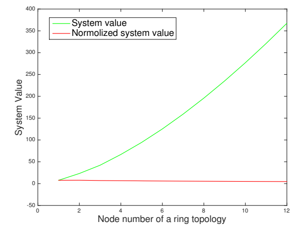

Fig. 3. The green curve shows the sensitivity of system value m with respect to the number of nodes in a ring topology. The red line is normalized with the number of nodes.

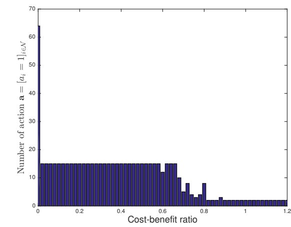

Fig. 4. The ratio of cost and reward affects the policy discretely. The probability of taking the action $\mathbf{a} = [a_i = 1]_{i\in\mathscr{N}}$ is larger when the cost is low.

$$\begin{aligned} g_i(x_i, \mathbf{x}_{\Omega_i}, \mathbf{a}) &= g_i(x_i, \mathbf{x}_{\Omega_i}, a_1, a_2, ..., a_n) \\ &= g_i(x_i, \mathbf{x}_{\Omega_i}, a_i, \mathbf{a}_{\Omega_i}), \quad \mathbf{a}_{\Omega_i} \in \mathscr{B}^{|\Omega_i|}. \end{aligned} \tag{10}$$

Therefore, based on (10), a centralized decision maker is not required. Each node can decide their action in a distributed way based on the local information from neighboring nodes. With the factored action space, we can use the same method stated in Section IV-B to solve the ALP. The constraint in the ALP becomes

$$0 \geq \max_{\mathbf{a},\mathbf{x}} \sum_i R_i(x_i, \mathbf{x}_{\Omega_i}) + \sum_i w_i \; \bar{g}_i(x_i, \mathbf{x}_{\Omega_i}, a_i, \mathbf{a}_{\Omega_i}). \tag{11}$$

The order of variable elimination is heuristic and will affect the complexity. One way to choose the sequence is to minimize variables involved at each step. Because a_i and x_i share the same structure, the order of elimination of a_i and x_i should be in succession. Moreover, the setting can be simplified further because the transition probability of node i only depends on its own action a_i. The action of i's parent node at stage t does not influence the state of itself until the next stage $t+1$, and will not influence node i until stage $t+2$. Hence, the executed action on the parent nodes has a delayed effect. With a natural assumption that $P(x_i'|x_i, \mathbf{x}_{\Omega_i}, \mathbf{a}) = P(x_i'|x_i, \mathbf{x}_{\Omega_i}, a_i)$, the expectation of basis function then is simplified as $g_i(x_i, \mathbf{x}_{\Omega_i}, \mathbf{a}) = g_i(x_i, \mathbf{x}_{\Omega_i}, a_i)$. The sequence of variable elimination can be first binary action variables and then binary state variables. Note that when computing the optimal value of a_i, we only need to investigate the value of a few state variables including x_i and x_j, $j \in \Omega_i$. Therefore, the computational complexity in finding control policy is reduced to $n \cdot (2^{1+|\Omega_i|})$ which is much less than the original $|\mathscr{X}| \times |\mathscr{A}| = 2^{2n}$.

V. Case Studies

A. Single-Repair Case

We first investigate two essential components of a real network: ring and line. For a directly ring topology, the system value increases as the number of the nodes increases as shown in Fig. 3, which demonstrates that increasing dependencies set back resilience. A similar pattern appears for directed line topology. However, with the same system parameters, a network of line topology always has a larger reward than its ring network counterpart as the terminal nodes in the line network are not connected.

B. Multiple-Repair Case

In the setting of multiple repairs, we aim to investigate the distributed control in a two-layer network. The system contains a four-node ring (node 1-4) representing the power grid and a two-node line (node 5,6) representing the subway line. The dependency probabilities within the network are set to 0.5, and the prior probability is set to 0.01. The dependency probability of link 6 to 1 is changing from 0.99 to 0 so that the two-layer network changes from completely independent to tightly coupled.

The first observation is the threshold policy which is fully decided by information of its parents' states. Let $\neg$ and $\bigoplus$ be logic notations of bitwise 'negation' and 'and', respectively. Nodes 2 to 5 have the policy fully determined by its own state $a_i = \neg x_i,\ i \in \{2,3,4,5\}$ while policy of node 6 is determined also by its parent, i.e., $\neg a_6 = x_5 \bigoplus x_6$. As the coupling turns to be tighter, node 1's policy changes from $a_1 = \neg x_1 + x_1 \neg x_4$ to $a_1 = \neg x_1 + x_1 \neg (x_4 \bigoplus x_6)$ at a turning point of probability 0.5. The reason lies in that after reaching 0.5, node 6 exerts more influence in making the policy.

Second, in general, increasing the action cost decreases the times of repairing. However, there still exist some states preferring to fix even when repairing cost is higher than the current reward as shown in Fig. 4. This interesting yet surprising result makes sense in the light of long-term benefit received in the future stages if the node works regularly. Moreover, the computation results also point out the risky nodes so that we can maintain them in advance to avoid future failing risks.

C. Large-Scale Networks Simulation

We construct a 100-node network with a random topology following the rule in Section II. Note that at each time step, the number of working nodes are calculated over 100 iterations to reduce the randomness of transition probability. The initial contingency of a normal power grid should be a rare event, and thus the prior failure probability is set to 0.001. The system starts at a state where 10% of the nodes are down. As shown in Fig. 5, the number of normal nodes reaches to zero quickly due to the cascading failure although the prior failure probability is relatively small. In comparison, the control policy greatly mitigates the cascading effect of the network. The blue line in Fig. 5 converges to a dynamic balance of keeping approximate 80% of the nodes to work.

VI. Conclusion and Future Work

In this paper, we have used a factored MDP approach to enable the resilient design of large-scale interdependent critical infrastructures. Approximate policy can be characterized using the weight of each node instead of a brute-force or greedy search. The proposed distributed policy takes the form of

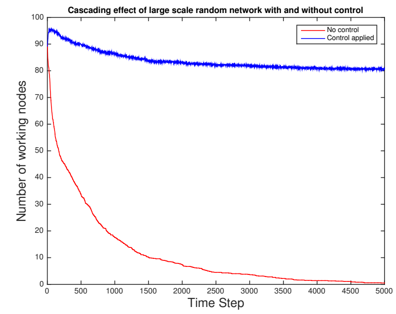

Fig. 5. Cascading effect of a large-scale random network with and without optimal control. The designed policy can always mitigate the cascading effect.

a threshold policy, which can mitigate the negative effect on system resilience introduced by increasing dependency. The future work would be the investigation of the control policies to mitigate cascading failures on some other structured networks, e.g., chordal and imperfect networks.

References

[1] J. D. Moteff, *Critical infrastructures: Background, policy, and implementation*. DIANE Publishing, 2010.

[2] C. Boutilier, R. Dearden, M. Goldszmidt *et al.*, "Exploiting structure in policy construction," in *IJCAI*, vol. 14, 1995, pp. 1104–1113.

[3] C. Boutilier, T. Dean, and S. Hanks, "Decision-theoretic planning: Structural assumptions and computational leverage," *Journal of Artificial Intelligence Research*, vol. 11, no. 1, p. 94, 1999.

[4] J. Moteff and P. Parfomak, "Critical infrastructure and key assets: definition and identification." DTIC Document, 2004.

[5] S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly, "Identifying, understanding, and analyzing critical infrastructure interdependencies," *IEEE Control Systems*, vol. 21, no. 6, pp. 11–25, 2001.

[6] P. Pederson, D. Dudenhoeffer, S. Hartley, and M. Permann, "Critical infrastructure interdependency modeling: a survey of us and international research," 2006.

[7] J. Chen and Q. Zhu, "Resilient and decentralized control of multi-level cooperative mobile networks to maintain connectivity under adversarial environment," in *IEEE Conference on Decision and Control (CDC)*, 2016, pp. 5183–5188.

[8] J. Chen and Q. Zhu, "Interdependent network formation games with an application to critical infrastructures," in *American Control Conference (ACC)*, 2016, pp. 2870–2875.

[9] I. Dobson, K. R. Wierzbicki, J. Kim, and H. Ren, "Towards quantifying cascading blackout risk," in *Bulk Power System Dynamics and Control-VII. Revitalizing Operational Reliability, 2007 iREP Symposium*. IEEE, 2007, pp. 1–12.

[10] Z. Wang, A. Scaglione, and R. J. Thomas, "A markov-transition model for cascading failures in power grids," in *System Science (HICSS), 2012 45th Hawaii International Conference on*. IEEE, 2012, pp. 2115–2124.

[11] J. D. Glover, M. S. Sarma, and T. Overbye, *Power System Analysis & Design, SI Version*. Cengage Learning, 2012.

[12] I. Dobson, B. A. Carreras, and D. E. Newman, "A loading-dependent model of probabilistic cascading failure," *Probability in the Engineering and Informational Sciences*, vol. 19, no. 01, pp. 15–32, 2005.

[13] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 1995, vol. 1, no. 2.

[14] D. P. de Farias and B. Van Roy, "The linear programming approach to approximate dynamic programming," *Operations Research*, vol. 51, no. 6, pp. 850–865, 2003.

[15] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman, "Efficient solution algorithms for factored mdps," *Journal of Artificial Intelligence Research*, vol. 19, pp. 399–468, 2003.