Critical Nodes in Interdependent Networks with Deterministic and Probabilistic Cascading Failures

Veremyev, Alexander; Pavlikov, Konstantin; Pasiliao, Eduardo; Thai, My T.; Boginski, Vladimir

Published in:
Journal of Global Optimization

DOI:
10.1007/s10898-018-0703-5

Publication date:
2019

Document version
Accepted manuscript

Citation for published version (APA):

Terms of use
This work is brought to you by the University of Southern Denmark through the SDU Research Portal. Unless otherwise specified it has been shared according to the terms for self-archiving. If no other license is stated, these terms apply:

• You may download this work for personal use only.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying this open access version

If you believe that this document breaches copyright please contact us providing details and we will investigate your claim. Please direct all enquiries to puresupport@bib.sdu.dk

Download date: 30. Apr. 2021
Critical Nodes in Interdependent Networks with Deterministic and Probabilistic Cascading Failures

Alexander Veremyev · Konstantin Pavlikov · Eduardo L. Pasiliao · My T. Thai · Vladimir Boginski

Abstract We consider optimization problems of identifying critical nodes in coupled interdependent networks, that is, choosing a subset of nodes whose deletion causes the maximum network fragmentation (quantified by an appropriate metric) in the presence of deterministic or probabilistic cascading failure propagations. We use two commonly used network fragmentation metrics: total number of disabled nodes and total number of disabled pair-wise connectivities. First, we discuss computational complexity issues and develop linear mixed integer programming (MIP) formulations for the corresponding optimization problems in the deterministic case. We then extend these problems to the case with probabilistic failure propagations using Conditional Value-at-Risk (CVaR) measure. We develop a scenario-based linear MIP model and propose an exact Markov chain-based algorithm to solve these problems. Finally, we perform a series of computational experiments on synthetic and semi-synthetic networks and discuss some interesting insights that illustrate the properties of the proposed models.

Keywords Combinatorial Optimization · Interdependent networks · Cascading Failures · Critical Nodes · Vulnerability Assessment · Conditional Value-at-Risk

A. Veremyev and V. Boginski
Department of Industrial Engineering and Management Systems, University of Central Florida, Orlando, FL 32816

K. Pavlikov
Department of Business and Economics, University of Southern Denmark, Campusvej 55, 5230 Odense M, Denmark

E.L. Pasiliao
Munitions Directorate, Air Force Research Laboratory, Building 13, Eglin AFB, FL 32542

M.T. Thai
Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611

This is a post-peer-review, pre-copyedit version of an article published in Journal of Global Optimization. The final authenticated version is available online at: http://dx.doi.org/10.1007/s10898-018-0703-5
1 Introduction

Interdependent networks arise in many application domains associated with infrastructure systems, such as power grids, telecommunication, and transportation networks. In a real-world setup, these systems interact with each other, so that disruptions/failures of components in one of the systems may affect the performance of other systems that depend on those components. Thus, failures can propagate through interdependent networked systems in a cascading fashion, where a failure of one component in one system may cause a failure of multiple components in another system, and so on. Cascading failure propagations in interdependent networks can be modeled using various assumptions. Deterministic failure propagation implies that a failure of a node would always cause failures of certain other nodes that “depend” on that node, whereas uncertain failure propagation implies that these failures would be caused with some probability. An interesting research question arising in this context is the problem of identifying the most “significant” (“critical”) nodes in an interdependent network, specifically, those whose failure (deletion) maximally degrades the network functionality quantified by some metric. In this paper, we focus on optimization problems that address this question from the attacker’s perspective. Formally, the considered problems consist in identifying a subset of up to $k$ nodes, whose removal would cause the largest “damage” to the network after a sequence of $S$ dynamic (deterministic or probabilistic) cascading failures, with the following two metrics used to quantify the damage:

- total number of operational nodes of the network after $S$ stages of cascading failures;
- total number of connected pairs of operational nodes in each layer of the network after $S$ stages of cascading failures.

Note that the parameter $S$ defines the number of cascading stages after which the network functionality is evaluated. It can be either small, if the decision maker is more concerned with the network functionality once the cascade started, or large ($S \to +\infty$), if the final state of the interdependent network after the cascading failures are longer occurring is of particular interest. The motivation for this consideration is the fact that in many cases cascading failures do not spread instantaneously, but rather in a stage-by-stage manner (see for instance [8,21]) and possibly can be stopped at earlier stages. Hence, depending on the application of interest, the decision maker might be interested in assessing network vulnerability after a certain number of cascade stages have occurred or when a cascade of failures is complete.

Clearly, identifying critical nodes in interdependent networks is also important from the defender’s perspective, as it would provide a decision maker not only valuable information about the vulnerability of the system, but also which nodes should be especially well-protected in order to prevent potential large-scale damage due to cascading failures, thus giving an opportunity to enhance robustness of an interdependent system.
1.1 Related Previous Work

Recent studies in modeling and simulation of interdependent networked systems cover general concepts and challenges \[43,1,31,44\], performance of interdependent networks under vertex or edge removals and related risk assessment issues \[15–17, 33,38,42\], as well as analytical and simulation approaches to studying cascading failures in interdependent networks \[52,32,9,20,5,40,50\]. A substantial amount of work in analyzing robustness of interdependent networks that is based on the size of largest remaining connected component in power networks after cascading failures has been done in \[8,21,24,39,49\]. Similar cascading failure models are described in \[58\]. Most of these studies focus on random interdependency between networks and node failures, and do not consider the detection of critical nodes in real networks and the effect of their failures. Recent studies \[20,13\] consider robustness of multiple interdependent networks (network of networks) under random failures or targeted attacks using a similar failure propagation model. Nguyen et al. \[35\] consider the so-called Interdependent Power Network Disruptor (IPND) optimization problem to identify critical nodes in an interdependent power network whose removals maximally destroy its functions due to both malfunction of these nodes and the cascading failures of its interdependent communication network. The problem was shown to be \(\mathcal{NP}\)-hard to approximate within the factor of \((2 - \epsilon)\), and a heuristic to solve this problem was proposed. Relative models of interdependent networks with cascading failure propagations and the corresponding mitigation/restoration strategies were considered in \[7,19,23,56\]. The study \[38\] used characteristic path length as a metric for assessing vulnerability of interdependent networks. For an extensive survey of recent studies of interdependent infrastructure models the reader is referred to \[37\].

An approach related to the one proposed in this paper was presented in \[25\], where the authors provide a model for vulnerability analysis of the system of interdependent infrastructures. The paper \[55\] considered a special-case setup of critical node detection problems in deterministic settings, which was represented as a generalization of the classical minimum vertex cover problem to coupled interdependent networks. This paper considers a more general framework for critical node detection problems in interdependent networks in both deterministic and probabilistic settings. Specifically, in the probabilistic framework, we will assume that a failure propagates through each interdependent link independently with a fixed probability \(p\), which is similar to the model described in \[10\]. This failure propagation mechanism (or “activation” in influence propagation applications) is also referred to as “forest fire”-based model of \[4\] with modifications by \[14\], cellular automata-based model \[34\], or Independent Cascade Model \[26\] and can be viewed as an extension of such models to interdependent network settings.

As a final remark of this section we note that the recent explosion of studies on interdependent networks in various disciplines resulted in the appearance of numerous relevant concepts and terminology (e.g., multilayer networks, multiplex networks, interdependent networks, networks of networks, etc.). In a recent survey \[28\], the authors attempted to unify the existing terminology, discussed a general framework for multilayer networks, and constructed a dictionary to relate the existing concepts to each other.
2 Critical Nodes in Interdependent Networks: Basic Model

Following the basic notations for the deterministic model introduced in [55], we define a mathematical model of a two-layer interdependent network as follows. The layers of an interdependent network are assumed to be simple undirected graphs $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ with sets of nodes $V_1 = \{1, \ldots, n_1\}$, $V_2 = \{1, \ldots, n_2\}$ ($n_1 + n_2 = n$), and edges $E_1, E_2$. It is also assumed that some nodes in one layer depend on nodes in another layer, and the corresponding sets of interdependent links (arcs) are denoted by $E_{12}, E_{21}$, meaning that if $(i, j) \in E_{12}$, then node $j \in V_2$ depends on node $i \in V_1$. For example, in the network depicted in Fig. 1, $E_{12} = \{(2, 1), (2, 2), (3, 3), (4, 3)\}$ and $E_{21} = \{(4, 4), (3, 2)\}$. The resulting coupled interdependent network is denoted as $G^{(2)} = ((V_1, E_1), (V_2, E_2), E_{12}, E_{21})$. The adjacency matrices corresponding to sets of edges $E_1, E_2, E_{12}, E_{21}$ are denoted by $A_1 = \{a_{ij}^1\}_{i,j \in V_1}, A_2 = \{a_{ij}^2\}_{i,j \in V_2}, A_{12} = \{a_{ij}^{12}\}_{i \in V_1, j \in V_2}, A_{21} = \{a_{ij}^{21}\}_{i \in V_2, j \in V_1}$, respectively.

The set of initially failed/attacked nodes may cause subsequent failures of other nodes due to the interdependence between nodes in layers. We assume that these failures propagate stage-by-stage, similar to the assumptions of [9]. As mentioned in the previous section, the motivation for this assumption is the fact that in many cases cascading failures do not spread instantaneously, but rather with some delay between cascade stages [6].

2.1 Deterministic Failure Propagation

In this setup, we assume that the consequent node failures propagate along the interdependent arcs with certainty. In particular, the failed nodes cause the failures of their dependent nodes according to “one-to-many” interdependence, which is defined in [55] as interdependence of Type 1:

Fig. 1: An example of a 2-layer interdependent network (layer 1 is on the left, layer 2 is on the right).
Definition 1 (Type 1 interdependence [55]) An interdependent network has 

**Type 1** failure causality if the failure of one node causes the failure of all of its dependent nodes.

We also assume that the failure process is observed until a pre-defined stage $S$. For a given interdependent network $G^{(2)}$, **cascades stages** and operational nodes are formally defined as follows:

**Definition 2 (Cascade Stage)** Let the initial set of failed nodes belong to stage 0. Then for any $s \in \{1, \ldots, S\}$, stage $s$ contains nodes that either belong to stage $(s - 1)$, or have failed due to the dependence on nodes in stage $(s - 1)$.

**Definition 3 (Operational nodes)** A node $i \in V_1 \cup V_2$ is operational (functional) at stage $s$ if it does not belong to stage $s$.

Moreover, we use the **depth of cascade** parameter defined as follows.

**Definition 4** The depth of cascade $S^*$ is the maximum possible length of the sequence of cascading failures in a given interdependent network $G^{(2)}$.

As mentioned in [55], the depth of cascade $S^*$ for any given interdependent network $G^{(2)}$ can be easily computed using the following explicit formula:

$$S^* = \max_{i,j \in V_1 \cup V_2, i \neq j} (\text{dist}(i, j) \mid \text{dist}(i, j) < +\infty)$$

where $\text{dist}(i, j)$ is the length of the shortest directed path from node $i$ to $j$ in a graph $G' = (V_1 \cup V_2, E_{12} \cup E_{21})$, and $\text{dist}(i, j) = +\infty$ if there is no such path.

Given a set $C = C_1 \cup C_2$ ($C_1 \subseteq V_1, C_2 \subseteq V_2$) of initially failed nodes in $G^{(2)}$ and a positive integer $S$, consider the remaining interdependent network when $S$ stages of cascading failures has occurred due to Type 1 interdependence. Let $v = \{v_{s, \ell}^i : s = 0, \ldots, S; i \in V_\ell, \ell = 1, 2\}$ be the set of binary variables such that $v_{s, \ell}^i = 1$ if and only if a node $i$ in layer $\ell$ belongs to stage $s$. As demonstrated in [55], these variables are useful for modeling stage-by-stage failure propagations as linear constraints in the corresponding MIP formulations. In this paper, we also consider the connectivity variables $u_{\ell}^{ij}$ ($i, j \in V_\ell, \ell = 1, 2$) such that $u_{\ell}^{ij} = 1$ if and only if nodes $i, j \in V_\ell$ are operational and connected by a path in layer $\ell$ going through operational (functional) nodes at stage $S$.

To measure the functionality and fragmentation of the remaining interdependent network, we can consider the following metrics:

- **Total number of operational nodes at stage $S$**:
  \[
  F_1(C, S) = n - \sum_{i \in V_1} v_{s, 1}^i - \sum_{j \in V_2} v_{s, 2}^j
  \]

- **Total number of connected pairs of operational nodes at stage $S$**:
  \[
  F_2(C, S) = \sum_{i_1, i_2 \in V_1, i_1 < i_2} u_{s, 1}^{i_1, i_2} + \sum_{j_1, j_2 \in V_2, j_1 < j_2} u_{s, 2}^{j_1, j_2}
  \]

In addition, we note that this list can include metrics considered in [54] (e.g., size of the largest connected component, the number of connected components, etc.) as they can be modeled using aforementioned pairwise connectivity variables and satisfy the monotonicity property defined as follows.
Fig. 2: Illustration of the stage-by-stage cascading failure process with $S = 2$. At stage 2, $F_1(\{A_2, B_3\}, 2) = 5$, and $F_2(\{A_2, B_3\}, 2) = 3$.

**Definition 5** Let $v^{iS} = \{v^{i1}_1, v^{i2}_2 : i \in V_1, j \in V_2\}$, $v^{nS} = \{v^{n1}_1, v^{n2}_2 : i \in V_1, j \in V_2\}$ be the values of variables $v$ that correspond to two states of the interdependent network at stage $S$, such that

$$
\begin{align*}
    v^{i1}_1 &\leq v^{n1}_1, & i \in V_1, \\
    v^{i2}_2 &\leq v^{n2}_2, & j \in V_2.
\end{align*}
$$

A measure $F$ is called monotonic if

$$
F(v^{iS}) \leq F(v^{nS}).
$$

In other words, if there are two instances of the remaining network, such that any failed node in instance 1 is also failed in instance 2, then the network functionality measured by metric $F$ in instance 1 is at most as large as in instance 2. Further in our computational study we focus on measures $F_1$ and $F_2$ only.

Let $c_{1i}$ ($i \in V_1$) and $c_{2j}$ ($j \in V_2$) be the costs associated with the removal of nodes in $G^{(2)}$ and $W$ be the total budget. The corresponding Critical Node Problem in Interdependent Network (CNPIN) is formally defined as:

**Definition 6 (CNPIN)** For a given interdependent network $G^{(2)}$ and a budget $W \geq 0$ find a subset of nodes $C = C_1 \cup C_2$ ($C_1 \subseteq V_1$, $C_2 \subseteq V_2$) such that

$$
\sum_{i \in C_1} c_{1i} + \sum_{j \in C_2} c_{2j} \leq W \text{ and } F_1(C, S) \text{ or } F_2(C, S) \text{ is minimized}.
$$

To distinguish between minimization of $F_1(C, S)$ and $F_2(C, S)$, the corresponding optimization problems are denoted by **CNPIN-1** and **CNPIN-2**. In this paper, we focus on the unweighted version of these problems, where $|C| = k$, and $c_{1i} = 1, \forall i \in V_1$, $c_{2j} = 1, \forall j \in V_2$. Fig. 2 illustrates an example of interdependent network (Fig. 2a, each node $i$ in layer 1 and in layer 2 is labelled by $A_i$ and $B_i$, respectively, for simplicity) and stage-by-stage node failures caused by disabling two nodes at stage 0 (filled, Fig. 2b) with $S = 2$. At stage 2, $F_1(\{A_2, B_3\}, 2) = 5$, and $F_2(\{A_2, B_3\}, 2) = 3$. 
2.2 Probabilistic Failure Propagation

In this setup, we assume that interdependent arcs can propagate cascading failures with some probability. For a given two-layer interdependent network \( G^{(2)} = ((V_1, E_1), (V_2, E_2), E_{12}, E_{21}) \), let \( \mathcal{P}_{12} = \{0 \leq p_{ij}^{(12)} \leq 1 : (i,j) \in E_{12}\} \) and \( \mathcal{P}_{21} = \{0 \leq p_{ji}^{(21)} \leq 1 : (j,i) \in E_{21}\} \) be the probability sets associated with the sets of interdependent links \( E_{12}, E_{21} \). Specifically, for any \((i,j) \in E_{12}\), \( p_{ij}^{(12)} \) is the probability that the failure of node \( i \in V_1 \) will cause the failure of node \( j \in V_2 \) at next stage.

For simplicity of presentation, we consider the case where \( p_{ij}^{(12)} = p, \forall (i,j) \in E_{12} \) and \( p_{ji}^{(21)} = p, \forall (j,i) \in E_{21} \), i.e., the interdependent links \( E_{12} \) and \( E_{21} \) propagate the failures with the same probability \( p \). As indicated above, this assumption is consistent with the “forest fire”, cellular automata, or independent cascade models that can be used as simple probabilistic representations of failure (activation) propagations between nodes. We will use this assumption in all our stochastic models; however, they can be easily generalized to the case when each arc in \( E_{12} \) and \( E_{21} \) has its own probability of failure propagation, although such generalization is outside the scope of this paper.

In presence of uncertainty, any subset \( C \subseteq V_1 \cup V_2 \) of \( k \) initially failed (attacked) nodes causes random network fragmentation, therefore, the functionality measured by functions \( F_1(\cdot) \) or \( F_2(\cdot) \) will be a random value. From a defender’s perspective, such randomness creates the risk that in certain states the network disruption will be very high. On the other hand, from an attacker’s perspective, the outcomes with low network disruption are less favorable. Let a loss \( L \) be a random variable equal to the random value of metric \( F \) (\( F_1 \) or \( F_2 \)) and consider the stochastic version of the critical node problem via the following risk minimization problem:

\[
\min_C \quad \mathcal{R}(L),
\]

where \( \mathcal{R}(\cdot) \) is a risk measure. A risk measure by definition is a map from a space of random variables to \( \mathbb{R} \), \( \mathcal{R} : \mathcal{L} \to \mathbb{R} \), i.e., a functional that aggregates a random value into a single number. In many situations, there exists a threshold constant \( L \), which a decision maker does not want the damage to exceed. Ideally it should never exceed \( L \), so in other words we would like the inequality \( F \leq L \) to hold almost surely or with probability one. However, this is not always possible. Instead, we might relax this requirement and require that a decision is acceptable if the average loss is less than or equal to \( L \), i.e., \( EF \leq L \). This relaxed requirement can then be interpreted using the expected value as the risk measure as follows:

\[
F \text{ “acceptably” } \leq L \iff EF \leq L.
\]  

Such requirement is not risk-averse, in other words, even if the random loss on average is less than or equal to \( L \), it can happen with high probability that the loss is greater than \( L \). Hence, another possibility is to require that \( F \leq L \) holds with probability at least \( \alpha \). This requirement can be achieved using the Value-at-Risk (VaR) measure:

\[
F \text{ “acceptably” } \leq L \iff \text{VaR}_\alpha(F) \leq L.
\]
Conditional Value-at-Risk (CVaR) is a widely used quantitative risk measure due to its practical interpretability and attractive mathematical properties that allow incorporating it into optimization problem formulations [29, 45, 46]. For a continuous random variable $X$ that describes a "loss", CVaR with confidence level $\alpha$ is defined as follows (see Fig. 3 for an illustration):

$$\text{CVaR}_\alpha(X) = E(X|X > q_\alpha),$$

where $q_\alpha$ is $\alpha$-quantile, defined by

$$q_\alpha(X) = \inf\{a \in \mathbb{R} : P(X > a) \leq 1 - \alpha\}.$$

For a discrete random variable $X$ and any $\alpha \in (0, 1)$, it is more convenient to use the generalized CVaR definition introduced in [45] as follows:

$$\text{CVaR}_\alpha(X) = \min_{\eta} \left( \eta + \frac{1}{1 - \alpha} E[|X - \eta|^+] \right).$$

When $X$ has a set of outcomes $(x_1, \ldots, x_Q)$ with the corresponding probabilities $(p_1, \ldots, p_Q)$, equation (6) can be rewritten as follows:

$$\text{CVaR}_\alpha(X) = \min_{\eta} \left( \eta + \frac{1}{1 - \alpha} \sum_{q=1}^{Q} p_q [x_q - \eta]^+ \right),$$
which can be linearized as

$$\min_{\eta, z} \left( \eta + \frac{1}{1 - \alpha} \sum_{q=1}^{Q} p_q z_q \right)$$

subject to

$$z_q \geq x_q - \eta, \quad q = 1, \ldots, Q,$$
$$z_q \geq 0, \quad q = 1, \ldots, Q.$$  

Assume that an attacker’s random “loss” associated with the disabling of a set of nodes \( C \subseteq V_1 \cup V_2 \) in the interdependent network \( G^{(2)} \) at stage \( S \) is the value of the function \( F_1(C, S) \) or \( F_2(C, S) \). For example, if no nodes are operational at stage \( S \) (i.e., \( F_1(C, S) = 0 \)), then the “loss” is zero, meaning that such an attack makes the remaining network completely dysfunctional. Conversely, if no nodes other than those in \( C \) fail as a result of the attack, then the loss is \( F_1(C, S) = |V_1| + |V_2| - |C| \) (maximum possible loss). In our study, we employ CVaR measure due to its simple representation through linear programming problem (8) - (10) and an intuitive practical interpretation. Since CVaR is the average “loss” across the worst \( 100 \cdot (1 - \alpha)\% \) outcomes, an attacker’s goal would be to minimize CVaR.

2.3 Scenario Generation for the Stochastic Environment

One possible way to estimate CVaR of the loss distribution is to use scenario-based approach. Since we assume that arcs in \( E_{12} \) and \( E_{21} \) propagate failures with the same probability \( p \), then we can view them as existing in the network with probability \( p \) and non-existing with probability \( 1 - p \). Thus, random sets \( E_{12} \) and \( E_{21} \) can be approximated by \( Q \) scenarios, i.e., \( Q \) deterministic sets \( E_{12}^q \) and \( E_{21}^q \), \( q = 1, \ldots, Q \) generated in such a way that every arc exists with probability \( p \), independently from other arcs. For any \( q = 1, \ldots, Q \), let \( L_q \) be the loss, which is equal to \( F_1(\cdot) \) or \( F_2(\cdot) \) corresponding to realizations \( E_{12}^q \) and \( E_{21}^q \) and a set \( C \) of initially failed nodes. Then CVaR of the random loss can be obtained using (8) - (10), assuming that every realization of \( L_q \) is equally probable:

$$\min_{\eta, z} \left( \eta + \frac{1}{Q(1 - \alpha)} \sum_{q=1}^{Q} z_q \right)$$

subject to

$$z_q \geq L_q - \eta, \quad q = 1, \ldots, Q,$$
$$z_q \geq 0, \quad q = 1, \ldots, Q.$$  

Note that such scenario-based approach gives an estimate of the CVaR. The larger the number of scenarios \( Q \), the closer the empirical distribution of possible outcomes to the real distribution, the better is the CVaR estimate. In the computational experiments section, we compare the CVaR values for optimal solutions estimated using various number of scenarios with the exact optimal CVaR value computed using a Markov chain-based approach which is described further in the paper. Our findings indicate that using relatively small number of scenarios gives rather accurate estimate of the minimum CVaR values that can be obtained using the exact Markov chain-based approach.
3 Computational Complexity

We present the computational complexity analysis, i.e., prove \(\mathcal{NP}\)-completeness for the problems CNPIN-1 and CNPIN-2 in the deterministic case for any fixed values \(S\) and \(S^*\), where \(0 < S \leq S^*\), meaning that the problem is not only \(\mathcal{NP}\)-complete for general interdependent networks \(G^{(2)}\), but also when restricted to the class of interdependent networks with depth of cascade equal to any given \(S^* > 0\). The \(\mathcal{NP}\)-completeness of stochastic case follows immediately since it contains deterministic as its special case. When \(S = 0\) (no cascading failures occur) or \(S^* = 0\) (no intra-layer interdependencies) the problem CNPIN-1 is trivial, whereas CNPIN-2, which minimizes the number of connected pairs of nodes in both layers \(G_1 = (V_1, E_1)\) and \(G_2 = (V_2, E_2)\), is a well-known Critical Node Detection problem shown to be \(\mathcal{NP}\)-hard [2].

In what follows, we first prove that the decision version of CNPIN-1 and CNPIN-2 is \(\mathcal{NP}\)-complete for the class of interdependent networks \(G^{(2)}\) with the depth of cascade \(S^* = 1\) and with considered number of failure stages \(S = 1\). Then, we show how this proof can be extended for other classes of interdependent networks \(G^{(2)}\) with any given \(S^*\) and any considered number of failure stages \(S\), such that \(S^* \geq S \geq 1\).

For the \(\mathcal{NP}\)-completeness proof, we use the reduction to the classical maximum coverage problem, which is known to be \(\mathcal{NP}\)-complete [3,27]. Following the standard approach [22], define the decision versions of CNPIN-1 and CNPIN-2 as follows:

**CPIN-1D/CNPIN-2D**

**INSTANCE:** An interdependent network \(G^{(2)} = ((V_1, E_1), (V_2, E_2), E_{12}, E_{21})\) with the depth of cascade \(S^*\), positive integers \(S, k (1 \leq S \leq S^*)\), and a positive number \(D\).

**QUESTION:** Is there a subset of nodes \(C = C_1 \cup C_2\), such that \(C_1 \subseteq V_1, C_2 \subseteq V_2, |C| = k\) and

\[
\text{CNPIN-1D}: \mathcal{F}_1(C, S) \leq D, \quad (14)
\]

\[
\text{CNPIN-2D}: \mathcal{F}_2(C, S) \leq D. \quad (15)
\]

The decision version of the maximum coverage problem is defined as follows:

**MC-D**

**INSTANCE:** A universe \(U = \{1, \ldots, m\}\) with \(m\) elements, a family \(A = \{A_1, \ldots, A_t\}\) of \(t\) subsets of \(U\), and positive integers \(\alpha < t\) and \(\beta\).

**QUESTION:** Does there exist \(A' \subseteq A\) with \(|A'| = \alpha\), such that the total number of elements of \(U\) in \(A'\) is at least \(\beta\).

**Proposition 1** CNPIN-1D is \(\mathcal{NP}\)-complete for \(S = S^* = 1\).

**Proof** Observe that the value of \(\mathcal{F}_1(C, S)\) can be computed in polynomial time; hence, CNPIN-1D is in the class \(\mathcal{NP}\). Consider an instance of MC-D with given \(U, A, \alpha\) and \(\beta\) where \(|U| = m\) and \(|A| = t\). Without loss of generality, assume that \(\forall i \in U, \exists j: i \in A_j\). We construct an interdependent network \(G^{(2)} = ((V_1, E_1), (V_2, E_2), E_{12}, E_{21})\) with the depth of cascade \(S^* = 1\) and prove that there exists \(A' \subseteq A\) with \(|A'| = \alpha\), such that the total number of elements of \(U\) in
Observe that $S$ the total number of elements of $A$ dependent on the first layer to be the set of nodes corresponding to the universe set $E$ (Fig. 4). Formally, $C$ is at least $\beta$ if and only if there is a subset of nodes $C = C_1 \cup C_2$, such that $C_1 \subseteq V_1$, $C_2 \subseteq V_2$, $|C| = \alpha$ and $F_1(C, S) \leq m + t - \alpha - \beta$, where $S = 1$.

The main idea for the interdependent network construction is to define the first layer to be the set of nodes corresponding to the universe set $U$ (i.e., $V_1 = U = \{1, \ldots, m\}$), the second layer of nodes to be the set of indices of the family $A = \{A_1, \ldots, A_t\}$ (i.e., $V_2 = \{1, \ldots, t\}$), and construct links of interdependencies $E_{21}$ in such a way that if element $i \in U$ belongs to subset $A_j$ then $(j, i) \in E_{21}$ (Fig. 4). Formally,

\begin{align*}
V_1 &= \{1, \ldots, m\}, \ E_1 = \emptyset, \quad (16) \\
V_2 &= \{1, \ldots, t\}, \ E_2 = \emptyset, \quad (17) \\
E_{21} &= \{(j, i) : j \in \{1, \ldots, t\}, i \in \{1, \ldots, m\}, \text{ and } i \in A_j\}, \ E_{12} = \emptyset, \quad (18) \\
G^{(2)} &= ((V_1, E_1), (V_2, E_2), E_{12}, E_{21}). \quad (19)
\end{align*}

Observe that $S^* = 1$. Suppose that there exists $A' \subseteq A$ with $|A'| = \alpha$, such that the total number of elements of $U$ in $A'$ is at least $\beta$. Let $C \subseteq V_2$ be the set of indices of elements in $A'$. Obviously, $|C| = \alpha$ and $F_1(C, S) \leq m + t - \alpha - \beta$, since the removal of $\alpha$ nodes in layer 2 ($V_2$) causes the failure of at least $\beta$ nodes in layer 1 ($V_1$).

Conversely, suppose that there exists a subset of nodes $C = C_1 \cup C_2$, such that $C_1 \subseteq V_1$, $C_2 \subseteq V_2$, $|C| = \alpha$ and $F_1(C, S) \leq m + t - \alpha - \beta$, where $S = 1$. Note that removing a node $i \in V_1$ does not do any better than removing node $j \in V_2 : (j, i) \in E_{21}$ (by our assumption, such node exists for any $i \in V_1$). Therefore, without loss of generality, we can assume that $C_1 = \emptyset$. Let $A' = \{A_j : j \in C_2\}$, then $|A'| = \alpha$. Since, $F_1(C, S) \leq m + t - \alpha - \beta$, and the number of remaining operational nodes in $V_2$ is $t - \alpha$, then the number of operational nodes in $V_1$ is at most $m - \beta$, which means that there are at least $\beta$ elements of $U$, covered by $A'$.

**Corollary 1** **CNPIN-1D** is $\mathcal{NP}$-complete for any fixed positive integers $S, S^* : 1 \leq S \leq S^*$.

**Proof** The proof follows immediately from generalizing the proof of Proposition 1 by adding extra nodes to the constructed graph in order to increase $S$ and $S^*$ to
the desired level, similarly to the technique in the proof of Proposition 3 in [55]. □

**Proposition 2** CNPIN-2D is \(\text{NP}\)-complete for any fixed positive integers \(S, S^* : 1 \leq S \leq S^*\).

**Proof** Observe that the value of \(F_2(C, S)\) can be written as
\[
F_2(C, S) = \sum_{l=1}^{L} \frac{\sigma_l(\sigma_l - 1)}{2},
\]
where \(\sigma_l, l = 1, \ldots, L\) are the sizes of remaining connected components and \(L\) is the total number of these components in both network layers. Since the number of remaining connected components and their size can be computed in polynomial time, then \(F_2(C, S)\) can be computed in polynomial time; hence, CNPIN-2D is in the class \(\text{NP}\).

For \(S = S^* = 1\) the proof is similar to the proof of Proposition 1 with the only difference that in the constructed graph \(G^{(2)}\), we define \(E_1 = \{(i, j) : \forall i, j \in V_1\}\) instead of the empty set. In this case, layer 1 contains \(b\) operational nodes if and only if the total number of connected pairs of operational nodes is \(\frac{b(b-1)}{2}\). Therefore, it can be shown that that there exists \(A' \subseteq A\) with \(|A'| = \alpha\), such that the total number of elements of \(U\) in \(A'\) is at least \(\beta\) if and only if there is a subset of nodes \(C = C_1 \cup C_2\), such that \(C_1 \subseteq V_1\), \(C_2 \subseteq V_2\), \(|C| = \alpha\) and \(F_2(C, S) \leq \frac{(m-\beta)(m-\beta-1)}{2}\), where \(S = 1\).

The generalization to any fixed positive integers \(S, S^*: 1 \leq S \leq S^*\) is also straightforward. □

**4 MIP Formulations: Deterministic Setup**

4.1 Minimizing the remaining number of operational nodes

An IP formulation for minimizing \(F_1(C, S)\) is straightforward since \(F_1(C, S) = n - \sum_{i \in V_1} v_1^S - \sum_{j \in V_2} v_2^S\) is a linear function of binary variables \(v_1^s, v_2^s\) \((i \in V_1, j \in V_2, s = 0, \ldots, S)\), and the cascading failure process can be modeled using linear recursive constraints described in [55].
Problem 1 \((F_1-D)\)

\[
\min \quad F_1(C, S) = n - \sum_{i \in V_1} v_{1i}^S - \sum_{j \in V_2} v_{2j}^S
\]  

subject to

\[
\sum_{i \in V_1} v_{1i}^0 + \sum_{j \in V_2} v_{2j}^0 \leq k,
\]

\[
v_{1i}^s \geq \frac{1}{n_2 + 1} \left( v_{1i}^{s-1} + \sum_{j \in V_2} a_{21}^{ij} v_{2j}^{s-1} \right), \quad i \in V_1, \ s = 1, \ldots, S,
\]

\[
v_{1i}^s \leq v_{1i}^{s-1} + \sum_{j \in V_2} a_{21}^{ij} v_{2j}^{s-1}, \quad i \in V_1, \ s = 1, \ldots, S,
\]

\[
v_{2j}^s \geq \frac{1}{n_1 + 1} \left( v_{2j}^{s-1} + \sum_{i \in V_1} a_{12}^{ij} v_{1i}^{s-1} \right), \quad j \in V_2, \ s = 1, \ldots, S,
\]

\[
v_{2j}^s \leq v_{2j}^{s-1} + \sum_{i \in V_1} a_{12}^{ij} v_{1i}^{s-1}, \quad j \in V_2, \ s = 1, \ldots, S,
\]

\[
v_{1i}^s, v_{2j}^s \in \{0, 1\}, \quad i \in V_1, \ j \in V_2, \ s = 0, \ldots, S.
\]

In the above formulation, constraint (21) is a budget constraint that requires that at most \(k\) nodes in both network layers are failed at stage 0; and constraints (22) - (23) and (24) - (25) enforce cascading failure process in layer 1 and layer 2, respectively. Note that the formulation \(F_1-D\) contains \(n(S+1)\) binary variables and \(O(nS)\) constraints.

Observe that the objective (20) in \(F_1-D\) seeks to maximize the value of \(\sum_{i \in V_1} v_{1i}^S + \sum_{j \in V_2} v_{2j}^S\); hence, constraints (22), (24) are redundant; moreover, integrality of \(v_{2j}^s, j \in V_2, s = 1, \ldots, S, \ell = 1, 2\) can be relaxed. Then, \(F_1-D\) allows the following reformulation.

Problem 2 \((F_1-DR)\)

\[
\min \quad F_1(C, S) = n - \sum_{i \in V_1} v_{1i}^S - \sum_{j \in V_2} v_{2j}^S
\]  

subject to

\[
\sum_{i \in V_1} v_{1i}^0 + \sum_{j \in V_2} v_{2j}^0 \leq k,
\]

\[
v_{1i} \leq v_{1i}^{s-1} + \sum_{j \in V_2} a_{21}^{ij} v_{2j}^{s-1}, \quad i \in V_1, \ s = 1, \ldots, S,
\]

\[
0 \leq v_{1i}^0 \leq 1, \quad i \in V_1, \ s = 1, \ldots, S,
\]

\[
v_{2j} \leq v_{2j}^{s-1} + \sum_{i \in V_1} a_{12}^{ij} v_{1i}^{s-1}, \quad j \in V_2, \ s = 1, \ldots, S,
\]

\[
0 \leq v_{2j}^0 \leq 1, \quad j \in V_2, \ s = 1, \ldots, S,
\]

\[
v_{1i}^0, v_{2j}^0 \in \{0, 1\}, \quad i \in V_1, \ j \in V_2.
\]

In our computational experiments, we use the formulation \(F_1-DR\) instead of \(F_1-D\) to solve CNPIN-1 since it has fewer number of constraints and demonstrated better performance.
4.2 Minimizing the total number of connected pairs of operational nodes

Recall that in our model, the remaining number of connected pairs of operational nodes

\[ F_2(C, S) = \sum_{i_1, i_2 \in V_1: i_1 < i_2} u_{i_1 i_2}^1 + \sum_{j_1, j_2 \in V_2: j_1 < j_2} u_{j_1 j_2}^2 \]

is written in terms of binary connectivity variables \( u_{ij}^l \) \((i, j \in V, \ell = 1, 2)\), such that \( u_{ij}^l = 1 \) if and only if nodes \( i, j \in V \) are operational at stage \( S \) and connected by a path in layer \( \ell \) going through operational (functional) nodes at stage \( S \). The first linear MIP approach to ensure such connectivity requirements that have been proposed in the literature, see, e.g., [2, 36], is based on the idea of triangle inequalities that enforce transitive relationships between nodes in the graph, i.e., if \( i_1 \) is connected to \( i_2 \) and \( i_2 \) is connected to \( i_3 \) in layer 1, then \( i_1 \) should be connected to \( i_3 \), resulting in the following set of inequalities:

\[
\begin{align*}
& u_{i_1 i_2}^1 + u_{i_2 i_3}^1 - u_{i_1 i_3}^1 \leq 1, \quad \forall i_1, i_2, i_3 \in V_1, \quad (32) \\
& u_{i_1 i_2}^1 + v_{i_1 i_3}^S + v_{i_2 i_3}^S \geq 1, \quad \forall (i_1, i_2) \in E_1, \quad (33)
\end{align*}
\]

where the additional set of constraints (33) is based on that fact that nodes \( i_1 \) and \( i_2 \) are connected in layer 1 if there is an edge between them and both nodes are operational at stage \( S \). The drawback of such approach is that the number of constraints that need to be imposed is \( O(n^3) \), where \( n \) is the number of nodes in a graph. It makes the problem size and running time of the corresponding problem formulations almost intractable even on small sparse networks [2]. In another study [12], the authors show that the number of triangular constraints can be reduced to \( O(nm) \), where \( m \) is the number of edges in a graph. In a recent work [53] (and its further generalization in [54]), another linearization technique has been proposed which reduces the number of constraints to \( O(n^2) \). The authors report that such linearization not only significantly improves the running time for the corresponding problem formulations, but also allows to solve problems for much larger sparse network instances. The idea of this linearization is that instead of triangle inequalities (32), another set of recursive connectivity constraints can be used for layers 1 and 2:

\[
\begin{align*}
& u_{i_1 i_3}^1 \geq \frac{1}{|N_1(i_1)|} \sum_{i_2 \in N_1(i_1)} u_{i_2 i_3}^1 - v_{i_1 i_3}^S, \quad \forall i_1, i_3 \in V_1, \quad (34) \\
& u_{j_1 j_3}^2 \geq \frac{1}{|N_2(j_1)|} \sum_{j_2 \in N_2(j_1)} u_{j_2 j_3}^2 - v_{j_1 j_3}^S, \quad \forall j_1, j_3 \in V_2. \quad (35)
\end{align*}
\]

where \( N_\ell(i) = \{ j : (i, j) \in E_\ell, j \in V, \} \) denotes the set of neighbors of node \( i \) in layer \( \ell \in \{1, 2\} \). These constraints ensure that, for example, in layer 1, \( u_{i_1 i_3}^1 = 1 \), i.e., nodes \( i_1, i_3 \in V_1 \) are operational and connected at stage \( S \), if \( i_1 \) is operational at stage \( S \) and there is at least one neighbor \( i_2 \in N_1(i_1) \) of node \( i_1 \) (i.e., \( (i_1, i_2) \in E_1 \)) that is operational at stage \( S \) and connected to node \( i_3 \) (i.e., \( u_{i_2 i_3}^1 = 1 \)). In addition, the following set of constraints enforcing connectivities along the edges is required for the valid formulation:

\[
\begin{align*}
& u_{i_1 i_2}^1 + v_{i_1 i_2}^S + v_{i_2 i_3}^S \geq 1, \quad \forall (i_1, i_2) \in E_1, \quad (36) \\
& u_{j_1 j_2}^2 + v_{j_1 j_2}^S + v_{j_2 j_3}^S \geq 1, \quad \forall (j_1, j_2) \in E_2. \quad (37)
\end{align*}
\]
Moreover, additional (not required for a valid formulation) inequalities can be used to potentially improve the solver performance:

\[
    \begin{align*}
    u_{i_1i_3}^1 & \leq 1 - v_{i_1i_3}^S, \quad u_{i_1i_3}^1 & \leq 1 - v_{i_1i_3}^S, \quad \forall i_1, i_3 \in V_1, \\
    u_{i_1i_3}^1 & \leq \sum_{i_2 \in N_2(i_1)} u_{i_2i_3}^1, \quad \forall i_1, i_3 \in V_1, (i_1, i_3) \notin E_1, \\
    u_{j_1j_3}^2 & \leq 1 - v_{j_1j_3}^S, \quad u_{j_1j_3}^2 & \leq 1 - v_{j_1j_3}^S, \quad \forall j_1, j_3 \in V_2, \\
    u_{j_1j_3}^2 & \leq \sum_{j_2 \in N_2(j_1)} u_{j_2j_3}^2, \quad \forall j_1, j_3 \in V_2, (j_1, j_3) \notin E_2.
    \end{align*}
\]

The problem formulation that minimizes \( F_2(C, S) \) is as follows:

**Problem 3 (F2-D)**

\[
    \begin{align*}
    \min \ F_2^{G^{(2)}}(C, S) = & \sum_{i_1, i_2 \in V_1 : i_1 < i_2} u_{i_1i_2}^1 + \sum_{j_1, j_2 \in V_2 : j_1 < j_2} u_{j_1j_2}^2, \\
    \text{subject to} & \sum_{i \in V_1} v_{i_1}^0 + \sum_{j \in V_2} v_{j_2}^0 \leq k, \\
    \text{cascade propagation constraints : } & (22) - (25), \\
    \text{connectivity constraints : } & (34) - (41), \\
    u_{i_1i_2}, u_{j_1j_2} & \in \{0, 1\}, \quad \forall i_1, i_2 \in V_1, j_1, j_2 \in V_2, \\
    v_{i_1}^s, v_{j_2}^s & \in \{0, 1\}, \quad \forall i \in V_1, j \in V_2, s = 0, \ldots, S.
    \end{align*}
\]

The formulation \( F_2-D \) contains \( O(n^2) \) binary variables and \( O(n^2) \) constraints. Similarly to the previous subsection, we can relax the integrality for the variables \( v_{i_1}^s \) and \( v_{j_2}^s \) for \( 1 \leq s \leq S \). However, according to our observations from computational experiments conducted using XPress-MP solver [57] with the default settings, such relaxation does not provide any significant performance improvements or may even lead to a slower performance. This can be attributed to the fact that the number of variables and constraints related to connectivity modeling comprise the major part of the formulation and branching on these variables pose the most significant challenge to the optimization solver. Indeed, the number of variables \( v_{i_1}^s, v_{j_2}^s \) is in the order of \( O(n) \) for small \( S \), which is substantially smaller the number of connectivity variables. Moreover, by not declaring \( v_{i_1}^s, v_{j_2}^s \) variables as binary, the solver does not execute a number of heuristic procedures associated with them, which might result in a larger number of explored branch-and-bound nodes and slower overall performance. Hence, in our computational experiments, we use the formulation \( F_2-D \) to solve \textbf{CNPIN-2}.

As a final remark, we note that there are some other studies on critical node identification in the existing literature, which consider MIP techniques for handling connectivities in residual graphs, see e.g., [11,51,41,48], and these models can be used to solve \textbf{CNPIN-2}. However, they are either more complex or require some preprocessing procedures; therefore, for our experiments that we conduct mostly for demonstrating purposes of vulnerability of interdependent networks, we use constraints \( (34) - (41) \) as they are fairly easy to implement and the number of constraints is relatively small. For the recent survey overviewing the literature in this area we refer the reader to [30].
5 Optimization Models: Stochastic Setup

5.1 Scenario-based approach

As described in Section 2.3, in this approach we generate $Q$ possible realizations of cascade propagation links $E_{12}$ and $E_{21}$: $E_{12}^q$, $E_{21}^q$, $q = 1, \ldots, Q$. Each scenario has the probability $1/Q$ to occur. Then, the value of the fragmentation metric $F_1(C, S)$ or $F_2(C, S)$ of the remaining network at stage $S$ in each scenario $q (q = 1, \ldots, Q)$ is determined by the initial set of failed nodes $C$ and by the deterministic set of links $E_{12}^q, E_{21}^q$ that propagate failures in this scenario. We introduce variables $v_{sq\ell i} (s = 1, \ldots, S; q = 1, \ldots, Q; \ell = 1, 2; i \in V_\ell)$ which denote a set of failed nodes in layer $\ell$ at stage $s$ in scenario $q$. We use the same set of constraints (22) - (25) to model the propagation of cascading failures in each scenario. Using the same technique as in deterministic settings for minimizing $F_2(C, S)$, we also introduce connectivity variables $u_{\ell q ij} (i, j \in V_\ell; \ell = 1, 2; q = 1, \ldots, Q)$ to model pair-wise connectivities between operational nodes $i, j \in V_\ell$ in layer $\ell$ in each scenario $q$ at stage $S$ and use the same linearization approach. Using formulation (8) - (10) for minimizing CVaR, given loss $L_q = F_1(C, S)$ or $L_q = F_2(C, S)$ in each scenario $q$, the critical nodes problem formulations for two considered network fragmentation metrics can be written as follows:

**Problem 4 (CVaR-$F_1$-$S$)**

$$\min_{\eta, z, v} \left( \eta + \frac{1}{Q(1 - \alpha)} \sum_{q=1}^{Q} z_q \right)$$

subject to

$$z_q \geq n - \sum_{i \in V_1} v_{1q}^0 - \sum_{j \in V_2} v_{2q}^0 - \eta, \quad q = 1, \ldots, Q, \quad (45)$$

$$z_q \geq 0, \quad q = 1, \ldots, Q, \quad (46)$$

$$\sum_{i \in V_1} v_{1q}^0 + \sum_{j \in V_2} v_{2j}^0 \leq k, \quad q = 1, \ldots, Q, \quad (47)$$

cascade propagation constraints :

(28) - (31),

$v_{1q}^0, v_{2j}^0 \in \{0, 1\}, \quad q = 1, \ldots, Q, \quad i \in V_1, j \in V_2.$
Problem 5 (CVaR-$F_2$-S)

\[
\min_{\eta, x, v, u} \left( \eta + \frac{1}{Q(1-\alpha)} \sum_{q=1}^{Q} z_q \right) \tag{48}
\]

subject to
\[
z_q \geq \sum_{i_1, i_2 \in V_1: i_1 < i_2} u_{i_1 i_2}^1 q + \sum_{j_1, j_2 \in V_2: j_1 < j_2} u_{j_1 j_2}^2 q - \eta, \quad q = 1, \ldots, Q, \tag{49}
\]
\[
z_q \geq 0, \quad q = 1, \ldots, Q, \tag{50}
\]
\[
\sum_{i \in V_1} v_i^0 + \sum_{j \in V_2} v_j^0 \leq k, \tag{51}
\]

cascade propagation constraints:

(22) – (25), \quad q = 1, \ldots, Q,

connectivity constraints:

(34) – (41), \quad q = 1, \ldots, Q,

\[u_{i_1 i_2}^1, u_{j_1 j_2}^2 \in \{0, 1\}, \quad i_1, i_2 \in V_1, j_1, j_2 \in V_2, q = 1, \ldots, Q, \]

\[v_{i_1}^q, v_{j_2}^q \in \{0, 1\}, \quad i \in V_1, j \in V_2, s = 1, \ldots, S, q = 1, \ldots, Q, \]

\[v_{i_1}^0, v_{j_2}^0 \in \{0, 1\}, \quad i \in V_1, j \in V_2. \]

Note that in CVaR-$F_1$-S we use the formulation with relaxed integrality (similar to $F_1$-DR) as it demonstrates better performance; however, CVaR-$F_2$-S is based on $F_2$-D since relaxing integrality does not always help to improve the running time. Similarly to the notations used in the previous section, we will also refer to the problems defined by formulations CVaR-$F_1$-S and CVaR-$F_2$-S as CNPIN-1S and CNPIN-2S, respectively.

5.2 Markov chain-based approach

The idea behind this approach is that the CVaR value of the network fragmentation metric $F_1(C, S)$ or $F_2(C, S)$ at stage $S$ for all possible sets of initially failed nodes $C$ can be computed explicitly. Then, the set of nodes with the worst value of the CVaR needs to be selected. Indeed, the process of cascading failures under uncertainty can be represented as a Markov chain where an interdependent network moves from one state to another in a discrete time (cascade stages). Recall that for any $s \geq 1$, all nodes belonging to stage $s$ are completely determined by the nodes belonging to stage $s - 1$ and nodes, which fail exactly at stage $s - 1$; therefore, this process is "memoryless", i.e., has the Markov property. For a given interdependent network $G^{(2)} = ((V_1, E_1), (V_2, E_2), E_{12}, E_{21})$, let the random vector

\[X_s = (v_{1^s}^{s-1}, v_{2^s}^{s-1}, v_{1^s}^s - v_{1^s}^{s-1}, v_{2^s}^s - v_{2^s}^{s-1}), \tag{52}\]

where $v_{\ell}^s = (v_{\ell 1}^s, \ldots, v_{\ell n_\ell}^s)$, $\ell = 1, 2$ and $1 \leq s \leq S$, be a state of a Markov chain. The first two components of $X_s$ are two sequences of $0 - 1$ of length $n_1$ and $n_2$ representing the nodes, failed by stage $s - 1$, whereas the last two components of $X_s$ represent the nodes failed exactly at stage $s$. The initial state $X_0$ has the
first two sequences of 0, and the last two components represent the initially failed
nodes $C_1 \subseteq V_1$ and $C_2 \subseteq V_2$.

To illustrate this concept consider the interdependent network depicted on Fig. 5 with 5 nodes in each layer ($A_1$-$A_5$ and $B_1$-$B_5$) and the probability $p$ that each
interdependent arc causes the failure “transmission”. Let nodes $A_2$ and $A_4$ belong
to stage 0, i.e., initially failed nodes. Then, in the above terminology

$$X_0 = (00000, 00000, 01010, 00000).$$ (53)

Due to the uncertain failure propagation, there are 4 possible realizations of cascade at stage 1 (Fig. 7) and the corresponding transition probabilities:

1. **Fig. 7a:** No nodes fail at stage 1.
   $$P\{X_1 = (01010, 00000, 00000, 00000)\} = (1 - p)^3.$$

2. **Fig. 7b:** $B_1$ fails due to dependence on $A_2$, no other nodes fail.
   $$P\{X_1 = (01010, 00000, 00000, 10000)\} = p(1 - p^2).$$

---

**Fig. 5:** Initial network

**Fig. 6:** Stage 0 nodes = $A_2 + A_4$

**Fig. 7:** Cascade failure realizations at stage 1.
3. **Fig. 7c**: B3 fails due to dependence on A2 or A4, no other nodes fail.

\[ P\{X_1 = (01010, 00000, 00000, 00100)\} = (1 - p)(1 - (1 - p)^2). \]

4. **Fig. 7d**: B1 and B3 fail due to dependence on A2 and/or A4.

\[ P\{X_1 = (01010, 00000, 00000, 10100)\} = p(1 - (1 - p)^2). \]

The realization of cascade at stage 2 can be described in a similar manner starting from each realization at stage 1. Hence, we can iteratively find all the realizations at any stage \( s \geq 2 \) up to the desired stage \( S \), and calculate the corresponding probabilities and values of the considered fragmentation metric. Then, the value of CVaR at stage \( S \) for the initially failed nodes A2 and A4 can be computed explicitly. For example, Table 1 presents the calculation of the actual values of CVaR for \( p = 0.5, \alpha = 50\% \) for two fragmentation metrics \( F_1(\{A2, A4\}, 1) \) and \( F_2(\{A2, A4\}, 1) \).

<table>
<thead>
<tr>
<th>State</th>
<th>( F_1({A2, A4}, 1) )</th>
<th>( F_2({A2, A4}, 1) )</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 7a</td>
<td>8</td>
<td>13</td>
<td>0.125</td>
</tr>
<tr>
<td>Fig. 7b</td>
<td>7</td>
<td>6</td>
<td>0.125</td>
</tr>
<tr>
<td>Fig. 7c</td>
<td>7</td>
<td>4</td>
<td>0.375</td>
</tr>
<tr>
<td>Fig. 7d</td>
<td>6</td>
<td>3</td>
<td>0.375</td>
</tr>
<tr>
<td>CVaR0.5</td>
<td>7.25 (72.5%)</td>
<td>6.75 (33.75%)</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Illustration of calculation of the CVaR of the number of operational nodes and the number of connected pairs for the example from Fig. 7.

The last step is to consider all possible sets of initially failed nodes, compute the corresponding CVaR values, and select the set whose deletion gives the minimum possible value of the CVaR of fragmentation metric \( F_1(\cdot) \) or \( F_2(\cdot) \) at stage \( S \). From the illustration above it is clear that although the number of possible states in the cascading failure process is \( 2^{2n_1 + 2n_2} \), the actual number of states, which need to be considered, can be much smaller. Moreover, since we are interested only in the worst outcomes, we demonstrate that this procedure can be simplified even further. The drawback of the Markov chain approach is that the number of iterations and memory usage grows exponentially with the network size and the number of initially failed nodes. Nevertheless, it gives the exact optimal solution as opposed to scenario-based approach which might be useful for the performance evaluation and design of effective heuristics.

5.3 Implementation

Below we formally describe the procedure that implements the aforementioned Markov chain approach for finding \( k \) nodes whose failure causes the maximum loss to the network functionality, measured by CVaR of \( F_1(\cdot) \) or \( F_2(\cdot) \) at stage \( S \).

**Implementation Procedure**
1. Create a set of all possible $C_{n_1+n_2}^k$ choices of $k$ out of $n = n_1 + n_2$ nodes to fail at Stage 0 and denote it as $C = \{ C^\eta : \eta = 1, \ldots, C_n^k \}$.

2. For every $C^\eta$ ($\eta = 1, \ldots, C_n^k$), i.e., $k$ nodes to be failed at Stage 0
   (a) Identify a set of all possible network states $\tau_1$ at Stage 1, $\xi_1^1, \ldots, \xi_1^t$ and compute the corresponding set of transition probabilities $p_1^1, \ldots, p_1^t$.
   (b) For every network state $\xi_1^t$ ($t = 1, \ldots, \tau_1$) find a set of all possible network states $\tau_2$ at Stage 2, $\xi_2^1, \ldots, \xi_2^t$ and compute the corresponding set of transition probabilities $p_2^1, \ldots, p_2^t$.
   (c) Create the set of all possible networks states at Stage 2 and the corresponding unconditional probabilities using the probabilities of networks states $\xi_1^1, \ldots, \xi_1^t$ at Stage 1, $p_1^1, \ldots, p_1^t$.
   (d) Using the same technique, repeat Steps (c), (d) until the cascade stage $S$ is reached, and all possible networks states and the corresponding transition probabilities are identified.
   (e) For every possible network state at Stage $S$, compute $F_1(\cdot)$ or $F_2(\cdot)$. Using the probability of each corresponding value $F_1(\cdot)$ or $F_2(\cdot)$, compute the associated CVaR value, $\text{CVaR}(C^\eta)$.

3. Select $\eta^* = \text{argmin}_{\eta = 1, \ldots, C_n^k} \text{CVaR}(C^\eta)$.

6 Computational Experiments

In this section, we present the results of numerical experiments to demonstrate the performance of the proposed MIP models and techniques in deterministic and stochastic settings. Specifically, we aim to illustrate robustness and vulnerability of sampled interdependent networks with respect to the number of cascade stages, number of interdependent links, number of critical nodes, and the probability of failure propagation, as well as numerically investigate the difference between scenario-based and Markov chain-based exact algorithm for the CVaR optimization models.

6.1 Hardware, Software, Test Instances

The proposed MIP models were implemented and solved using XPress-MP solver [57] on an Intel Core(TM) i5-4200M CPU 2.5GHz, RAM 6GB machine, equipped with Windows 8.1x64 operating system. The Markov Chain approach was implemented in MATLAB R2011b on an Intel Xeon X5355 2.66 GHz, 16GB RAM machine, equipped with Windows 7x64 operating system.

To the best of our knowledge, there are no publicly available real-world instances of interdependent networks; therefore, we used a combination of real-world and randomly generated networks in our computational experiments. Each test instance is characterized by the number of nodes in each layer ($|V_1|$ and $|V_2|$), as well as the number of interdependent arcs ($|E_{12}|$ and $|E_{21}|$). For convenience of presentation in the tables, each instance is labeled in the format $|V_1|\{V_2|E_{12}|E_{21}|$. As in many other related studies, random network instances were generated constructed according to a classical Erdős-Rényi $G(n, p)$ model [18]. By definition, a uniform random graph $G(n, p)$ has $n$ nodes, and every pair of nodes is con-
connected by an edge with probability $p$.\(^1\) In the considered random instances, each of the layers $(V_1, E_1)$ and $(V_2, E_2)$ was generated using the $G(n, p)$ model, where the parameter $p$ was set to 0.55. This choice of the parameter $p$ is motivated by the idea that we would like to test the proposed techniques on sufficiently dense graphs, but at the same time, these graphs should not be too dense; otherwise the solutions (sets of critical nodes) of CNPIN-1 and CNPIN-2 would likely coincide. Using a similar uniform random generation procedure, arcs in $E_{12}$ and $E_{21}$ are added randomly in a way that every possible arc has the same probability of existence and arcs are added until the specified number of interdependent arcs is achieved. Moreover, in order to illustrate the performance of the proposed models in realistic settings, we also conduct computational experiments on the two-layer network, which consists of the well-known IEEE 118-bus power grid network (constructed using publicly available data), which is a portion of the American Electric Power System in the Midwestern US as of December, 1962, and a randomly generated SCADA network (with the assumed power-law distribution, which is common for communication networks) similar to the one depicted in [55]. In these cases, the test instances are labeled `ieee118_118comm_[E_{12}]_[E_{21}]`. In addition, multiple random network instances were generated for each combination of the aforementioned parameters. These instances are labeled in the format `|V_1|_|V_2|_|E_{12}|_|E_{21}|_1`, `|V_1|_|V_2|_|E_{12}|_|E_{21}|_2`, etc.

6.2 Results and Discussion

6.2.1 Deterministic Setup

The first part of the experiments in the deterministic setup deals with the problem CNPIN-1. Specifically, the goal is to illustrate the relation between the size of the survived part of the interdependent network at stage $S$ and the number of critical nodes $k$, especially with respect to the number interdependent arcs in $E_{12}$ and $E_{21}$. In this case, it turns out that reasonably large instances can be solved to optimality using the aforementioned formulations. Tables 2 and 3 present objective values (in percentages of total number of nodes) and CPU times, respectively, for CNPIN-1. One can observe that the number of arcs in $E_{12}$ and $E_{21}$ plays an important role in keeping the network robust with respect to cascading failures: the larger the number of arcs in $E_{12}$ and $E_{21}$, the smaller fraction of the network survives in the end of cascading stage $S$. Intuitively, this is not surprising, since the more interdependent arcs the network has, the greater the average number of node neighbors in the dependent layer; thus, the failure of a node in one layer can spread to many nodes in the other layer.

In the second part of the experiments, it turns out that CNPIN-2 problem is more computationally challenging than CNPIN-1; therefore the test instances for this problem are selected to be smaller than those used for CNPIN-1. Tables 4 and 5 present objective values (in percentages of total number of connected pairs of nodes in both layers in the initial network) and CPU times, respectively, for CNPIN-2. Based on the results presented in these tables, the following general

\(^1\) In this subsection only, the parameter $p$ is used in the context of the classical $G(n, p)$ model, rather than the parameter denoting the probability of failure propagation throughout other sections of the paper.
observations can be made regarding the computational time and solution properties of the problems in the deterministic setup:

- The number of interdependent arcs significantly impacts the size of survived part of the network (as it can be observed from Table 2);
- In some cases (especially those with IEEE 118 bus topology), it is sufficient to attack only \( \sim 2.5\% \) of nodes in the entire network and allow 4 cascade stages to occur in order to disable \( \sim 70\% \) of the nodes in the network, which indicates that interdependent networks with such topology and type of dependence can be rather vulnerable to targeted attacks;
- CPU time of \textbf{CNPIN-1} generally increases with the increase of the number of interdependent arcs (see Table 3);
- The size of remaining (survived) network after the deletion of critical nodes appears to be a convex function of the number of attacked nodes (as illustrated in Fig. 8). This implies that an incremental increase of parameter \( k \) for lower values of \( k \) has a more pronounced effect on the network damage, rather than the same increase for higher values of \( k \);
- CPU time of \textbf{CNPIN-2} drastically increases compared to time of \textbf{CNPIN-1} and substantially depends on the network size \( n \).

![Fig. 8: Number of a) operational nodes and b) connected pairs of nodes (reported as a fraction of initial values) after the failure of \( k \) critical nodes in one of the instance of ieee118_118comm_200_200 network. The total initial number of nodes is 236 (118 nodes in each layer), and the number of connected node pairs in the initial network is 13806.](image)

6.2.2 Stochastic Setup

The next part of the computational experiments deals with the same problems in the stochastic setup (\textbf{CNPIN-1S} and \textbf{CNPIN-2S}). Since this setup requires
to generate multiple random arc failure scenarios and incorporate them into an optimization problem, it results in a large number of additional variables and constraints in the formulations. Therefore, the network instances that can be solved to optimality in the stochastic setup are substantially smaller than those in the deterministic setup; moreover, the considered number of cascade stages is also smaller (limited to \( S = 3 \) in this part of the experiments). Tables 6–9 present the corresponding results. We also illustrate the robustness of a network depending on the parameter \( p \) (probability that a failure is propagated through an interdependent arc). Not surprisingly, we observe that the fraction of survived nodes or survived connections in the network in the end of a cascading failure process inversely depends on the value of the parameter \( p \), which underscores the importance of designing networks with secured interdependencies, i.e., networks with low probability of failure propagations. The observations that can be made from the obtained results are summarized below.

- It is important to compare scenario-based solutions with exact solution obtained by the Markov chain (MC)-based algorithm. It turns out that although the considered number of scenarios for each problem is moderate (compared to the total number of possible scenarios), the objective values obtained by exact and scenario-based approaches are rather close (as illustrated in Fig. 9);
- The optimal objective values of **CNPIN-1S** and **CNPIN-2S** obtained by the exact MC-based approach appear to be concave functions of parameter \( p \) (see Fig. 9). This implies that, for instance, a 0.1 decrease of the value of the parameter \( p \) from 0.9 to 0.8 has a greater impact on the fraction of the survived network than a 0.1 decrease of the value of the parameter \( p \) from 0.4 to 0.3;
- CPU time for both problems drastically increases with the number of scenarios;
- MC-based exact algorithm is computationally challenging, and it can outperform MIP only for small-size and/or sparse network instances;
- In the considered instances, the value of \( p \approx 0.6 \) generally appears to be a “tipping point” between a relatively robust state of the network (over a half of the whole network is operational after the attack) and a “degraded” state of a network (a small fraction of a network is operational).

### 7 Conclusion

In this paper, we considered the models of cascading failure propagation in coupled interdependent networks in both deterministic and probabilistic settings, as well as the corresponding optimization problems of critical node detection in such networks. We proved the \( \mathsf{NP} \)-completeness of the decision versions of the considered problems, and provided the corresponding mathematical programming formulations. We also defined critical node detection problems in interdependent networks under uncertain failure propagations and used Conditional Value-at-Risk (CVaR) as the metric for quantifying losses associated with the failure of critical nodes. We demonstrated that the proposed CVaR-based optimization problems can be addressed using both exact (Markov chain) and scenario-based (linear 0-1 programming) approaches and our computational experiments indicate that using a small number of scenarios can be enough to find a pretty accurate estimate of the optimal CVaR value. As a direction for future research, it would be interesting to
develop efficient heuristic algorithms for the defined problems, which would allow one to tackle substantially larger network instances in real-world settings. In addition, the proposed formulations may yield extensions to the weighted versions of the problems (with weights associated with nodes and/or links). A detailed study of these problems would also be of interest.

Acknowledgements

This material is based upon work supported by the AFRL Mathematical Modeling and Optimization Institute. M.T. Thai’s and V. Boginski’s research is supported in part by NSF award EFRI-1441231.
Table 2: Optimal objective values for CNPIN-1 on the corresponding test instances. The objective values after the failure of $k$ critical nodes and $S$ cascade stages are reported as fractions (percentages) of total number of nodes in the network ($n = n_1 + n_2 = 236$).

<table>
<thead>
<tr>
<th>Network</th>
<th>$S$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>iecel18,118 comm_100,100,1</td>
<td>$k = 1$</td>
<td>0.7000</td>
<td>0.9620</td>
<td>0.9450</td>
<td>0.9280</td>
<td>0.9110</td>
<td>0.8940</td>
<td>0.8770</td>
<td>0.8640</td>
<td>0.8520</td>
</tr>
<tr>
<td>iecel18,118 comm_100,100,2</td>
<td>$k = 2$</td>
<td>0.9530</td>
<td>0.9240</td>
<td>0.8940</td>
<td>0.8640</td>
<td>0.8350</td>
<td>0.8090</td>
<td>0.7840</td>
<td>0.7590</td>
<td>0.7370</td>
</tr>
<tr>
<td>iecel18,118 comm_200,200,1</td>
<td>$k = 3$</td>
<td>0.9340</td>
<td>0.8860</td>
<td>0.8490</td>
<td>0.8290</td>
<td>0.7910</td>
<td>0.7630</td>
<td>0.7330</td>
<td>0.7030</td>
<td>0.6730</td>
</tr>
<tr>
<td>iecel18,118 comm_200,200,2</td>
<td>$k = 4$</td>
<td>0.9210</td>
<td>0.8900</td>
<td>0.8650</td>
<td>0.8400</td>
<td>0.8150</td>
<td>0.7910</td>
<td>0.7670</td>
<td>0.7430</td>
<td>0.7170</td>
</tr>
<tr>
<td>iecel18,118 comm_400,400,1</td>
<td>$k = 5$</td>
<td>0.9110</td>
<td>0.8490</td>
<td>0.7750</td>
<td>0.7120</td>
<td>0.6700</td>
<td>0.6390</td>
<td>0.6060</td>
<td>0.5800</td>
<td>0.5550</td>
</tr>
<tr>
<td>iecel18,118 comm_400,400,2</td>
<td>$k = 6$</td>
<td>0.9750</td>
<td>0.9340</td>
<td>0.8910</td>
<td>0.8580</td>
<td>0.8180</td>
<td>0.7760</td>
<td>0.7310</td>
<td>0.6830</td>
<td>0.6310</td>
</tr>
<tr>
<td>iecel18,118 comm_600,600,1</td>
<td>$k = 7$</td>
<td>0.9580</td>
<td>0.9200</td>
<td>0.8840</td>
<td>0.8480</td>
<td>0.8140</td>
<td>0.7800</td>
<td>0.7460</td>
<td>0.7160</td>
<td>0.6860</td>
</tr>
<tr>
<td>iecel18,118 comm_600,600,2</td>
<td>$k = 8$</td>
<td>0.9620</td>
<td>0.9240</td>
<td>0.8860</td>
<td>0.8480</td>
<td>0.8140</td>
<td>0.7800</td>
<td>0.7460</td>
<td>0.7160</td>
<td>0.6860</td>
</tr>
<tr>
<td>iecel18,118 comm_600,600,3</td>
<td>$k = 9$</td>
<td>0.9620</td>
<td>0.9240</td>
<td>0.8860</td>
<td>0.8480</td>
<td>0.8140</td>
<td>0.7800</td>
<td>0.7460</td>
<td>0.7160</td>
<td>0.6860</td>
</tr>
</tbody>
</table>

References

5. Barrett, C., Beckman, R., Channakeshava, K., Huang, F., Kumar, V., Marathe, A., Marathe, M., Pei, G.: Cascading Failures in Multiple Interdependent Networks: From Transporta-
Table 3: CPU time (in seconds) for the results reported in Table 2 which are obtained using $F_3$-DR.
<table>
<thead>
<tr>
<th>Network</th>
<th>$S$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>50_50_25_1</td>
<td>1</td>
<td>94.0</td>
<td>88.2</td>
<td>82.7</td>
<td>77.3</td>
<td>71.0</td>
<td>64.6</td>
<td>59.0</td>
<td>54.5</td>
<td>50.9</td>
</tr>
<tr>
<td>50_50_25_2</td>
<td>1</td>
<td>94.0</td>
<td>88.2</td>
<td>82.7</td>
<td>77.3</td>
<td>71.0</td>
<td>64.6</td>
<td>59.0</td>
<td>54.5</td>
<td>50.9</td>
</tr>
<tr>
<td>50_50_25_3</td>
<td>1</td>
<td>94.0</td>
<td>88.2</td>
<td>82.7</td>
<td>77.3</td>
<td>71.0</td>
<td>64.6</td>
<td>59.0</td>
<td>54.5</td>
<td>50.9</td>
</tr>
<tr>
<td>50_50_25_4</td>
<td>1</td>
<td>94.0</td>
<td>88.2</td>
<td>82.7</td>
<td>77.3</td>
<td>71.0</td>
<td>64.6</td>
<td>59.0</td>
<td>54.5</td>
<td>50.9</td>
</tr>
<tr>
<td>100_100_50_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100_100_50_2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100_100_50_3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100_100_50_4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Optimal objective values for CNPIN-2 on the corresponding test instances. The objective values after the failure of $k$ critical nodes and $S$ cascade stages are reported as fractions (percentages) of the total number of connected pairs of nodes in two layers of the network.

<table>
<thead>
<tr>
<th>Network</th>
<th>S</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>50,50,25,25,1</td>
<td>4.6</td>
<td>7.9</td>
<td>9.7</td>
<td>9.9</td>
<td>20.3</td>
<td>21.7</td>
<td>14.4</td>
<td>9.0</td>
<td>27.0</td>
<td></td>
</tr>
<tr>
<td>50,50,25,25,2</td>
<td>7.2</td>
<td>7.1</td>
<td>7.1</td>
<td>9.4</td>
<td>11.5</td>
<td>7.8</td>
<td>12.9</td>
<td>16.3</td>
<td>16.7</td>
<td></td>
</tr>
<tr>
<td>50,50,25,25,3</td>
<td>5.9</td>
<td>7.4</td>
<td>7.0</td>
<td>8.4</td>
<td>9.8</td>
<td>17.9</td>
<td>17.0</td>
<td>26.9</td>
<td>25.0</td>
<td></td>
</tr>
<tr>
<td>50,50,25,25,4</td>
<td>7.6</td>
<td>6.2</td>
<td>6.0</td>
<td>7.2</td>
<td>6.8</td>
<td>5.2</td>
<td>7.6</td>
<td>13.6</td>
<td>13.0</td>
<td></td>
</tr>
<tr>
<td>50,50,25,25</td>
<td>6.1</td>
<td>6.0</td>
<td>6.8</td>
<td>8.1</td>
<td>6.2</td>
<td>5.1</td>
<td>8.0</td>
<td>7.7</td>
<td>7.2</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: CPU time (in seconds) for the results reported in Table 4 which are obtained using $\mathcal{F}_2$-D.
<table>
<thead>
<tr>
<th>Network</th>
<th>$Q$</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>10_10_20_20</td>
<td>100</td>
<td>90.0</td>
<td>90.0</td>
<td>86.0</td>
<td>84.0</td>
<td>75.0</td>
<td>65.0</td>
<td>50.5</td>
<td>44.5</td>
<td>28.0</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>90.0</td>
<td>90.0</td>
<td>86.6</td>
<td>83.6</td>
<td>75.0</td>
<td>66.4</td>
<td>56.6</td>
<td>43.2</td>
<td>27.9</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>90.0</td>
<td>90.0</td>
<td>87.1</td>
<td>83.5</td>
<td>76.4</td>
<td>67.2</td>
<td>56.3</td>
<td>44.9</td>
<td>30.0</td>
</tr>
<tr>
<td>$\alpha^*$</td>
<td>100</td>
<td>90.0</td>
<td>90.0</td>
<td>86.0</td>
<td>83.5</td>
<td>76.0</td>
<td>65.8</td>
<td>66.3</td>
<td>44.6</td>
<td>29.6</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>90.0</td>
<td>90.0</td>
<td>87.0</td>
<td>83.9</td>
<td>76.5</td>
<td>66.6</td>
<td>62.9</td>
<td>49.0</td>
<td>35.3</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>90.0</td>
<td>90.0</td>
<td>87.5</td>
<td>84.4</td>
<td>77.0</td>
<td>67.6</td>
<td>60.5</td>
<td>45.9</td>
<td>34.8</td>
</tr>
<tr>
<td>10_10_20_2</td>
<td>100</td>
<td>90.0</td>
<td>90.0</td>
<td>86.0</td>
<td>84.0</td>
<td>75.0</td>
<td>65.0</td>
<td>50.5</td>
<td>44.5</td>
<td>28.0</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>90.0</td>
<td>90.0</td>
<td>87.0</td>
<td>84.2</td>
<td>81.0</td>
<td>72.9</td>
<td>61.9</td>
<td>50.2</td>
<td>35.3</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>90.0</td>
<td>90.0</td>
<td>88.4</td>
<td>85.1</td>
<td>80.1</td>
<td>72.3</td>
<td>60.6</td>
<td>49.0</td>
<td>34.5</td>
</tr>
<tr>
<td>$\alpha^*$</td>
<td>100</td>
<td>90.0</td>
<td>90.0</td>
<td>86.0</td>
<td>86.0</td>
<td>80.3</td>
<td>71.2</td>
<td>60.5</td>
<td>48.7</td>
<td>34.5</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>90.0</td>
<td>90.0</td>
<td>87.0</td>
<td>85.1</td>
<td>81.4</td>
<td>72.4</td>
<td>61.2</td>
<td>49.4</td>
<td>34.1</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>90.0</td>
<td>90.0</td>
<td>86.5</td>
<td>84.0</td>
<td>76.5</td>
<td>66.0</td>
<td>54.5</td>
<td>43.0</td>
<td>28.6</td>
</tr>
<tr>
<td>10_10_20_3</td>
<td>100</td>
<td>90.0</td>
<td>90.0</td>
<td>87.0</td>
<td>83.1</td>
<td>77.1</td>
<td>69.1</td>
<td>58.5</td>
<td>44.3</td>
<td>29.7</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>90.0</td>
<td>90.0</td>
<td>87.0</td>
<td>83.2</td>
<td>77.5</td>
<td>69.4</td>
<td>58.0</td>
<td>46.8</td>
<td>31.2</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>90.0</td>
<td>90.0</td>
<td>87.1</td>
<td>83.6</td>
<td>77.4</td>
<td>68.2</td>
<td>57.4</td>
<td>49.4</td>
<td>31.0</td>
</tr>
<tr>
<td>$\alpha^*$</td>
<td>100</td>
<td>90.0</td>
<td>90.0</td>
<td>87.0</td>
<td>83.1</td>
<td>77.5</td>
<td>69.0</td>
<td>50.0</td>
<td>39.0</td>
<td>21.5</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>90.0</td>
<td>90.0</td>
<td>87.4</td>
<td>83.9</td>
<td>75.3</td>
<td>64.8</td>
<td>54.0</td>
<td>37.4</td>
<td>22.1</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>90.0</td>
<td>90.0</td>
<td>87.3</td>
<td>83.8</td>
<td>74.0</td>
<td>64.0</td>
<td>51.2</td>
<td>39.5</td>
<td>24.7</td>
</tr>
<tr>
<td>$\alpha^*$</td>
<td>100</td>
<td>90.0</td>
<td>90.0</td>
<td>87.0</td>
<td>83.6</td>
<td>74.9</td>
<td>64.9</td>
<td>51.4</td>
<td>37.7</td>
<td>24.3</td>
</tr>
<tr>
<td>20_20_40_1</td>
<td>100</td>
<td>95.0</td>
<td>94.0</td>
<td>91.8</td>
<td>89.5</td>
<td>81.8</td>
<td>76.5</td>
<td>64.0</td>
<td>53.5</td>
<td>38.3</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>95.0</td>
<td>94.8</td>
<td>92.7</td>
<td>88.8</td>
<td>82.4</td>
<td>75.4</td>
<td>63.8</td>
<td>51.4</td>
<td>30.5</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>95.0</td>
<td>95.0</td>
<td>94.1</td>
<td>89.6</td>
<td>81.3</td>
<td>76.8</td>
<td>65.9</td>
<td>54.3</td>
<td>30.5</td>
</tr>
<tr>
<td>$\alpha^*$</td>
<td>100</td>
<td>95.0</td>
<td>95.0</td>
<td>94.1</td>
<td>88.6</td>
<td>81.7</td>
<td>74.5</td>
<td>60.8</td>
<td>40.1</td>
<td>22.1</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>95.0</td>
<td>95.0</td>
<td>94.0</td>
<td>89.4</td>
<td>82.4</td>
<td>74.8</td>
<td>64.3</td>
<td>50.4</td>
<td>21.7</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>95.0</td>
<td>95.0</td>
<td>94.5</td>
<td>90.6</td>
<td>84.9</td>
<td>76.5</td>
<td>65.7</td>
<td>54.2</td>
<td>22.5</td>
</tr>
<tr>
<td>$\alpha^*$</td>
<td>100</td>
<td>95.0</td>
<td>95.0</td>
<td>94.1</td>
<td>89.5</td>
<td>85.5</td>
<td>79.8</td>
<td>72.0</td>
<td>58.5</td>
<td>48.3</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>95.0</td>
<td>95.0</td>
<td>93.7</td>
<td>90.6</td>
<td>85.1</td>
<td>80.8</td>
<td>72.1</td>
<td>60.3</td>
<td>46.6</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>95.0</td>
<td>95.0</td>
<td>94.0</td>
<td>91.8</td>
<td>86.8</td>
<td>80.8</td>
<td>72.3</td>
<td>60.7</td>
<td>47.1</td>
</tr>
<tr>
<td>$\alpha^*$</td>
<td>100</td>
<td>95.0</td>
<td>95.0</td>
<td>94.9</td>
<td>92.4</td>
<td>88.3</td>
<td>76.9</td>
<td>65.9</td>
<td>45.6</td>
<td>28.5</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>95.0</td>
<td>95.0</td>
<td>94.1</td>
<td>91.9</td>
<td>88.5</td>
<td>77.5</td>
<td>63.3</td>
<td>52.0</td>
<td>38.7</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>95.0</td>
<td>95.0</td>
<td>94.3</td>
<td>92.7</td>
<td>89.0</td>
<td>81.2</td>
<td>75.6</td>
<td>64.6</td>
<td>55.1</td>
</tr>
<tr>
<td>$\alpha^*$</td>
<td>100</td>
<td>95.0</td>
<td>95.0</td>
<td>94.0</td>
<td>93.0</td>
<td>88.5</td>
<td>81.8</td>
<td>74.4</td>
<td>57.5</td>
<td>40.9</td>
</tr>
</tbody>
</table>

Table 6: Optimal objective values of CNPIN-1S (the CVaR of the total number of operational nodes across $Q$ scenarios with 90% confidence level) after the failure of $k = 2$ critical nodes and $S = 3$ cascade stages. The objective values are reported as fractions (percentages) of operational nodes in two layers of the initial network. The lines labeled by $\alpha^*$ present the solutions obtained by the Markov chain-based exact algorithm. “− − −” denotes that the optimal solution could not be obtained within the time limit of 50,000 seconds.

<table>
<thead>
<tr>
<th>Network</th>
<th>$Q$</th>
<th>$0.1$</th>
<th>$0.2$</th>
<th>$0.3$</th>
<th>$0.4$</th>
<th>$0.5$</th>
<th>$0.6$</th>
<th>$0.7$</th>
<th>$0.8$</th>
<th>$0.9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10_10_20_20_1</td>
<td>100</td>
<td>1.1</td>
<td>2.6</td>
<td>2.9</td>
<td>4.9</td>
<td>7.3</td>
<td>5.6</td>
<td>7.3</td>
<td>7.0</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>6.6</td>
<td>10.0</td>
<td>22.2</td>
<td>30.8</td>
<td>40.9</td>
<td>52.7</td>
<td>44.0</td>
<td>43.9</td>
<td>34.6</td>
</tr>
<tr>
<td></td>
<td>$\infty$</td>
<td>54.1</td>
<td>54.8</td>
<td>54.4</td>
<td>54.0</td>
<td>54.4</td>
<td>60.5</td>
<td>139.0</td>
<td>184.6</td>
<td>186.1</td>
</tr>
<tr>
<td>10_10_20_20_2</td>
<td>100</td>
<td>1.2</td>
<td>2.3</td>
<td>4.0</td>
<td>4.5</td>
<td>5.3</td>
<td>5.3</td>
<td>5.4</td>
<td>5.0</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>7.0</td>
<td>14.8</td>
<td>22.9</td>
<td>30.6</td>
<td>39.8</td>
<td>31.0</td>
<td>42.1</td>
<td>39.1</td>
<td>24.8</td>
</tr>
<tr>
<td></td>
<td>$\infty$</td>
<td>42.3</td>
<td>43.9</td>
<td>45.1</td>
<td>44.4</td>
<td>47.2</td>
<td>145.6</td>
<td>130.1</td>
<td>135.5</td>
<td></td>
</tr>
<tr>
<td>10_10_20_20_3</td>
<td>100</td>
<td>28.6</td>
<td>28.5</td>
<td>28.6</td>
<td>28.6</td>
<td>29.0</td>
<td>32.2</td>
<td>55.1</td>
<td>141.7</td>
<td>123.4</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>1.4</td>
<td>2.5</td>
<td>3.3</td>
<td>5.5</td>
<td>6.4</td>
<td>5.4</td>
<td>5.6</td>
<td>5.8</td>
<td>5.1</td>
</tr>
<tr>
<td></td>
<td>$\infty$</td>
<td>6.7</td>
<td>41.2</td>
<td>47.6</td>
<td>35.2</td>
<td>30.6</td>
<td>34.0</td>
<td>44.7</td>
<td>34.2</td>
<td>37.2</td>
</tr>
<tr>
<td>10_10_20_20_4</td>
<td>100</td>
<td>1.6</td>
<td>2.4</td>
<td>4.1</td>
<td>5.4</td>
<td>6.4</td>
<td>8.8</td>
<td>5.8</td>
<td>6.5</td>
<td>3.7</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>7.7</td>
<td>15.7</td>
<td>26.7</td>
<td>38.8</td>
<td>46.5</td>
<td>47.0</td>
<td>50.4</td>
<td>38.9</td>
<td>37.1</td>
</tr>
<tr>
<td></td>
<td>$\infty$</td>
<td>47.4</td>
<td>49.4</td>
<td>57.2</td>
<td>59.4</td>
<td>61.3</td>
<td>141.8</td>
<td>117.4</td>
<td>140.1</td>
<td></td>
</tr>
<tr>
<td>20_20_40_40_1</td>
<td>100</td>
<td>5.8</td>
<td>12.1</td>
<td>15.4</td>
<td>26.3</td>
<td>31.0</td>
<td>32.0</td>
<td>33.9</td>
<td>42.9</td>
<td>34.8</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>44.5</td>
<td>108.8</td>
<td>147.7</td>
<td>210.5</td>
<td>243.7</td>
<td>336.6</td>
<td>339.8</td>
<td>391.4</td>
<td>416.6</td>
</tr>
<tr>
<td></td>
<td>$\infty$</td>
<td>149.4</td>
<td>409.6</td>
<td>484.9</td>
<td>674.1</td>
<td>698.4</td>
<td>1014.4</td>
<td>1177.3</td>
<td>1806.8</td>
<td>1812.2</td>
</tr>
<tr>
<td>20_20_40_40_2</td>
<td>100</td>
<td>4.4</td>
<td>12.7</td>
<td>18.0</td>
<td>27.0</td>
<td>24.0</td>
<td>34.7</td>
<td>28.1</td>
<td>25.7</td>
<td>28.7</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>47.2</td>
<td>118.5</td>
<td>175.4</td>
<td>269.0</td>
<td>271.5</td>
<td>314.8</td>
<td>326.6</td>
<td>387.2</td>
<td>262.4</td>
</tr>
<tr>
<td></td>
<td>$\infty$</td>
<td>148.3</td>
<td>349.3</td>
<td>583.2</td>
<td>790.8</td>
<td>793.3</td>
<td>895.2</td>
<td>1162.0</td>
<td>1385.0</td>
<td>962.4</td>
</tr>
<tr>
<td>20_20_40_40_3</td>
<td>100</td>
<td>7.5</td>
<td>13.0</td>
<td>21.0</td>
<td>30.4</td>
<td>35.2</td>
<td>46.5</td>
<td>52.5</td>
<td>55.4</td>
<td>42.8</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>49.3</td>
<td>105.6</td>
<td>183.1</td>
<td>253.9</td>
<td>309.7</td>
<td>422.8</td>
<td>415.6</td>
<td>466.6</td>
<td>424.9</td>
</tr>
<tr>
<td></td>
<td>$\infty$</td>
<td>134.7</td>
<td>313.3</td>
<td>499.7</td>
<td>734.3</td>
<td>958.9</td>
<td>1104.4</td>
<td>1029.0</td>
<td>1271.3</td>
<td>1407.9</td>
</tr>
<tr>
<td>20_20_40_40_4</td>
<td>100</td>
<td>5.9</td>
<td>10.6</td>
<td>14.8</td>
<td>21.8</td>
<td>25.3</td>
<td>35.2</td>
<td>33.7</td>
<td>29.3</td>
<td>26.9</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>41.3</td>
<td>95.9</td>
<td>149.2</td>
<td>205.6</td>
<td>226.3</td>
<td>275.2</td>
<td>292.7</td>
<td>366.3</td>
<td>289.7</td>
</tr>
<tr>
<td></td>
<td>$\infty$</td>
<td>126.8</td>
<td>287.0</td>
<td>426.0</td>
<td>541.9</td>
<td>697.6</td>
<td>953.9</td>
<td>1051.9</td>
<td>1165.9</td>
<td>908.2</td>
</tr>
</tbody>
</table>

Table 7: CPU time (in seconds) for the problems presented in Table 6.
<table>
<thead>
<tr>
<th>Network</th>
<th>Q</th>
<th>( p )</th>
<th>( 0.1 )</th>
<th>( 0.2 )</th>
<th>( 0.3 )</th>
<th>( 0.4 )</th>
<th>( 0.5 )</th>
<th>( 0.6 )</th>
<th>( 0.7 )</th>
<th>( 0.8 )</th>
<th>( 0.9 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10_10_20_20_1</td>
<td>100</td>
<td>80.0</td>
<td>80.0</td>
<td>73.1</td>
<td>66.1</td>
<td>51.1</td>
<td>41.4</td>
<td>27.8</td>
<td>16.4</td>
<td>4.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>80.0</td>
<td>80.0</td>
<td>71.1</td>
<td>71.3</td>
<td>54.9</td>
<td>44.1</td>
<td>33.3</td>
<td>18.4</td>
<td>7.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \infty^* )</td>
<td>80.0</td>
<td>80.0</td>
<td>76.3</td>
<td>69.4</td>
<td>58.6</td>
<td>45.1</td>
<td>31.5</td>
<td>18.4</td>
<td>6.4</td>
<td></td>
</tr>
<tr>
<td>10_10_20_20_2</td>
<td>100</td>
<td>80.0</td>
<td>80.0</td>
<td>78.2</td>
<td>71.1</td>
<td>61.3</td>
<td>50.1</td>
<td>32.9</td>
<td>19.6</td>
<td>10.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>80.0</td>
<td>80.0</td>
<td>80.0</td>
<td>69.3</td>
<td>61.3</td>
<td>46.9</td>
<td>34.1</td>
<td>20.7</td>
<td>8.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \infty^* )</td>
<td>80.0</td>
<td>80.0</td>
<td>78.4</td>
<td>74.4</td>
<td>63.6</td>
<td>51.4</td>
<td>36.8</td>
<td>22.5</td>
<td>10.4</td>
<td></td>
</tr>
<tr>
<td>10_10_20_20_3</td>
<td>100</td>
<td>73.3</td>
<td>73.3</td>
<td>73.3</td>
<td>67.2</td>
<td>57.3</td>
<td>46.3</td>
<td>36.2</td>
<td>24.4</td>
<td>6.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>73.3</td>
<td>73.3</td>
<td>73.3</td>
<td>68.3</td>
<td>60.4</td>
<td>46.8</td>
<td>39.9</td>
<td>20.1</td>
<td>5.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \infty^* )</td>
<td>73.3</td>
<td>73.3</td>
<td>73.3</td>
<td>68.8</td>
<td>63.4</td>
<td>44.2</td>
<td>30.5</td>
<td>17.5</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>10_10_20_20_4</td>
<td>100</td>
<td>73.3</td>
<td>73.3</td>
<td>73.3</td>
<td>70.3</td>
<td>50.3</td>
<td>40.0</td>
<td>30.2</td>
<td>20.2</td>
<td>7.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>73.3</td>
<td>73.3</td>
<td>73.3</td>
<td>67.3</td>
<td>55.8</td>
<td>41.9</td>
<td>26.2</td>
<td>10.5</td>
<td>3.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \infty^* )</td>
<td>73.3</td>
<td>73.3</td>
<td>73.3</td>
<td>69.0</td>
<td>56.2</td>
<td>40.7</td>
<td>24.8</td>
<td>11.7</td>
<td>3.5</td>
<td></td>
</tr>
<tr>
<td>15_15_30_30_1</td>
<td>50</td>
<td>86.7</td>
<td>84.1</td>
<td>84.5</td>
<td>74.6</td>
<td>67.7</td>
<td>43.9</td>
<td>44.4</td>
<td>19.9</td>
<td>10.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>86.7</td>
<td>80.0</td>
<td>81.7</td>
<td>76.7</td>
<td>61.4</td>
<td>49.1</td>
<td>32.1</td>
<td>23.0</td>
<td>10.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \infty^* )</td>
<td>86.7</td>
<td>86.7</td>
<td>80.3</td>
<td>75.5</td>
<td>64.1</td>
<td>49.8</td>
<td>35.0</td>
<td>22.8</td>
<td>13.2</td>
<td></td>
</tr>
<tr>
<td>15_15_30_30_2</td>
<td>50</td>
<td>86.7</td>
<td>86.7</td>
<td>86.7</td>
<td>76.7</td>
<td>67.9</td>
<td>54.7</td>
<td>45.3</td>
<td>24.8</td>
<td>11.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>86.7</td>
<td>86.1</td>
<td>81.8</td>
<td>76.2</td>
<td>68.9</td>
<td>48.8</td>
<td>39.8</td>
<td>20.0</td>
<td>11.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \infty^* )</td>
<td>86.7</td>
<td>86.7</td>
<td>83.2</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>15_15_30_30_3</td>
<td>50</td>
<td>86.7</td>
<td>86.7</td>
<td>87.2</td>
<td>79.9</td>
<td>61.9</td>
<td>50.9</td>
<td>35.8</td>
<td>20.1</td>
<td>13.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>86.7</td>
<td>86.7</td>
<td>81.7</td>
<td>78.0</td>
<td>61.6</td>
<td>45.9</td>
<td>39.7</td>
<td>22.7</td>
<td>11.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \infty^* )</td>
<td>86.7</td>
<td>86.7</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>15_15_30_30_4</td>
<td>50</td>
<td>86.7</td>
<td>84.5</td>
<td>81.7</td>
<td>68.8</td>
<td>49.7</td>
<td>37.0</td>
<td>27.1</td>
<td>14.6</td>
<td>5.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>86.7</td>
<td>86.7</td>
<td>80.1</td>
<td>73.4</td>
<td>62.2</td>
<td>37.4</td>
<td>28.7</td>
<td>13.4</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \infty^* )</td>
<td>86.7</td>
<td>88.7</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td></td>
</tr>
</tbody>
</table>

Table 8: Optimal objective values of CNPIN-2S (the CVaR of the total number of survived nodes across \( Q \) scenarios with 90% confidence level) after the failure of \( k = 2 \) critical nodes and \( S = 3 \) cascade stages. The objective values are reported as fractions (percentages) of the number of connected pairs of nodes in the initial network. The lines labeled by \( \infty^* \) present the solutions obtained by the Markov chain-based exact algorithm. "--" denotes that the optimal solution could not be obtained within the time limit of 50,000 seconds.

<table>
<thead>
<tr>
<th>Network</th>
<th>Q</th>
<th>CPU Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10_10_20_20_1</td>
<td>100</td>
<td>482.9</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>804.7</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>1167.3</td>
</tr>
<tr>
<td>∞</td>
<td>500</td>
<td>1572.0</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>2104.4</td>
</tr>
<tr>
<td></td>
<td>1500</td>
<td>2735.0</td>
</tr>
<tr>
<td></td>
<td>2000</td>
<td>3496.2</td>
</tr>
<tr>
<td></td>
<td>2500</td>
<td>4421.7</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>10000.0</td>
</tr>
<tr>
<td>10_10_20_20_2</td>
<td>100</td>
<td>409.0</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>799.9</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>1372.9</td>
</tr>
<tr>
<td>∞</td>
<td>500</td>
<td>2737.7</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>2856.0</td>
</tr>
<tr>
<td></td>
<td>1500</td>
<td>3702.2</td>
</tr>
<tr>
<td></td>
<td>2000</td>
<td>4394.4</td>
</tr>
<tr>
<td></td>
<td>2500</td>
<td>5319.9</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>10000.0</td>
</tr>
<tr>
<td>10_10_20_20_3</td>
<td>100</td>
<td>114.8</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>259.1</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>406.5</td>
</tr>
<tr>
<td>∞</td>
<td>500</td>
<td>502.2</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>863.3</td>
</tr>
<tr>
<td></td>
<td>1500</td>
<td>1537.4</td>
</tr>
<tr>
<td></td>
<td>2000</td>
<td>2422.4</td>
</tr>
<tr>
<td></td>
<td>2500</td>
<td>3898.3</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>10000.0</td>
</tr>
<tr>
<td>10_10_20_20_4</td>
<td>100</td>
<td>259.1</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>579.3</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>914.4</td>
</tr>
<tr>
<td>∞</td>
<td>500</td>
<td>1144.4</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>1616.9</td>
</tr>
<tr>
<td></td>
<td>1500</td>
<td>2678.0</td>
</tr>
<tr>
<td></td>
<td>2000</td>
<td>4487.5</td>
</tr>
<tr>
<td></td>
<td>2500</td>
<td>7392.8</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>10000.0</td>
</tr>
</tbody>
</table>

Table 9: CPU time (in seconds) for the problems presented in Table 8.


