

Research Article

Role of Recovery in Evolving Protection against Systemic Risk: A Mechanical Perspective in Network-Agent Dynamics

Chulwook Park ^{1,2}

¹International Institute for Applied Systems Analysis (IIASA), Laxenburg A-2361, Austria

²Department of Physical Science, Seoul National University, Seoul 08826, Republic of Korea

Correspondence should be addressed to Chulwook Park; pcw8531@gmail.com

Received 19 March 2020; Revised 22 August 2020; Accepted 25 September 2020; Published 28 April 2021

Academic Editor: Tomas Veloz

Copyright © 2021 Chulwook Park. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a model of evolving protection against systemic risk related to recovery. Using the failure potential in network-agent dynamics, we present a process-based simulation that provides insights into alternative interventions and their mechanical uniqueness. The fundamental operating principle of this model is that computation allows greater emphasis on optimizing the recovery within the general regularity of random network dynamics. The rules and processes that are used here could be regarded as useful techniques in systemic risk measurement relative to numerical failure reduction analyses.

1. Introduction

1.1. Background. Various contemporary studies have argued that systemic risk and abrupt failure events are related to the highly interconnected systems and networked structures created by agents [1]. Based on a simple set of properties, observations derived from several models indicate that the proportion of protection between nodes in the network can be described as a probability. This is related to how systemic risk should be coped with rather than simply predicting it based on the probability of failure (e.g., a failed bank in a financial system, asymptomatic transmission of a disease, and cognitive bias in decision-making). The risk of propagation is higher than that of independent failure events, extending to interdependent ones, which we refer to as cascading failures among system components [2]. Many scenarios that arise in simulations should be regarded not as indicating uncertainty or mistakes but rather as the consequences of inappropriate settings and interactions. In particular, proper protection against systemic risk with system components can be evolved heuristically through strategy dynamics (social learning and exploration) as a potential means of limiting failure. The present model expands this concept to investigate nonlinear randomness effects due to delayed responses, which may result in

sensitivity to small changes that are difficult to prepare or manage [3]. To assess the profound implications that this approach may have for our understanding of dynamic behavior, including protection processes, this study investigates the influence of the necessary heuristics through which a proper response could mediate risk diffusion before a system completely fails [4], as discussed below.

1.2. Literature Review. By investigating how the complexity of networked structures underpins real-world systemic phenomena, various simulation studies have identified implications for individual robustness, the propagation of systemic risk, protection flow, and collective behavior across networks [5]. A distinguishing feature of such phenomena is that they emerge from the complex interactions among individual elements in a system or from their associations with each other [3]. The effect of context-varying mechanical flux on a system's risk is highly complex. The possibility of quantifying such a risk needs to be evaluated in consideration of the distortions and patterns of such effects [4]. The investigations involve a variety of information, e.g., social contacts that are favored as the infection and spreading route, which in turn can be used to infer the characteristics of the underlying networks [6]. As demonstrated by the

existence history (e.g., the financial crisis in 2008, the outbreak of COVID-19 in 2019), a reproduction (i.e., failure, bias, or virus) varies greatly from individual to individual, which is generally believed to affect the spreading dynamics significantly [7]. Whether such individual inhomogeneity aggravates the outbreak is a challenging question, and the answer depends on the specific model [8, 9]. In particular, the connectivity patterns of individuals are key to understanding [10] how networks are structured and communicated with each other [11]. Other network properties that have been investigated include the concept of evolutionary dynamics, which helps characterize and understand the architecture of artificial systems in relation to the network properties [12]. As most tools for laying out networks are variants of an algorithm, it is difficult to use them to explore how the conditions of a network affect the network's dynamics [13]. The assessment process can be used to make macroscale observations for input performance, while approaches for microscale evaluations to simultaneously obtain more detailed insights must be treated within the structure of the network itself [14]. Several studies have reported such structures in terms of both microscale (e.g., individual incentives and relative gain versus effort) and macroscale (e.g., institutional competition and central intervention) behavior. For example, evolutionary explanations of systemic risk demonstrate how optimal decision makers are constrained when creating biased estimates of their capability and show how individuals alter their strategies in response to perceptions of resource value [15]. Standard evolutionary models in complex environments show that potentially different biases in decision-making expose different experimental groups at different transition probabilities [16]. A recent study found that by employing strong mitigation (i.e., social distancing and isolation of confirmed cases as guided by risk diffusion testing) related to the different response strategies, an outbreak can be suppressed to levels below the normal critical care capacity [10]. Although a triggered cascade can evolve over a certain time scale (i.e., days), it can be mitigated with intervention by the central system [17]. Evidence from many nowcasting and forecasting estimates indicates that in the absence of prevention and control measures other than simply isolating the risk cases, the probability of continued transmission with the projected trajectory remains high (exponential growth of the number of infections) [7, 8]. Thus, there is an urgent need to reduce propagation rates and control the growth of this risk to reduce not only the peak demand on the system but also the total number of eventually affected individuals [18].

1.3. Gap Statement. The computational modeling technique shows no bridge between the dynamics of agent nodes (with the vertex as a fundamental element) and the emergent properties of failure in recovery [19] (note that “recovery” in the context of financial systemic risk often refers to the fraction of a loan that is recovered after the default of the counterparty; here, it refers to a different quantity called “recovery time delay” similar to the concept of intervention). Most tools for laying out networks are variants of an

algorithm and hence cannot easily be used to explore how the conditions affect the dynamics of the network [20] owing to the following factors: (1) Many of them take the form of a theoretical explanatory insight constructed in response to a hypothetical assumption. (2) The type and number of individuals are arbitrary or left undefined. (3) Validation with respect to stylized mechanical parameters cannot explain their potential over-parameterization. (4) There is an extended transient or burn-in phase that is discarded before analysis [21]. (5) Most importantly, the time units for many of these models may have no clear interpretation. To address these issues, we extend the model to fit an estimation of macro-/microscale variables, such as protections and interventions. The assessment process can be used to make macro- or microscale observations of input performance, while approaches for improving the recovery delay and obtaining more detailed insights should be investigated in the structure of the network itself [22]. This requires the combination of large repositories to construct representations of trajectories that can be analyzed at different scales and from different perspectives [23]. Indeed, the mechanisms and serial algorithms that underpin our understanding of systemic risk in networked agents must be evaluated through various means. Accordingly, we can establish a common ground for the integration of knowledge and methodologies with consistent definitions and reconcile the approaches for studying networks from various fields, which will intuitively enable us to face all the difficulties and pitfalls that are inherent in interdisciplinary work.

1.4. Purpose. This study develops a modeling framework that can account for quantitative measurement in agented networks, allowing us to explore how the recovery time delay affects the risk potential in both macro- and microscale cases. To regard agent dynamics as a random network, this model follows the standard approach in agent-network modeling, where by default, a small event (agent n is hit by a shock at time t) can trigger the initial passage in a risk diffusion process. The mechanism tests the clear implications for different values of the interaction, including interruption, and how protection may be related to a set of interconnectedness with mitigation entities against failure potential, rather than solely focusing on cascading events.

1.5. Value. With the objectives of better risk assessment and effective risk reduction, this model will enable us to not only directly observe the spread of failure in agented network industries but also better understand how protection can be accomplished through intervention. This work is related to the mainstream of research that contributes to the discussion of systemic robustness.

2. Model (Process-Based Methodology)

The concept of network generation in this study is based on random processes, such as those described in graph theory [24]. Although an arbitrary construction cannot fully capture the local characteristics of individuals observed in real-

world networks, everyone in the world is connected to everyone else through a chain of mutual acquaintances or even stronger relationships [25]. To examine the systemic risks that may result from failures originating and cascading on such contact networks, together with how the networked agents may be expected to protect themselves against failure cascades, we consider an agent-based systematic-risk model with evolving protection strategies that have been developed at the International Institute for Applied Systems Analysis (IIASA; Ulf Dieckmann). This model enables agent-based simulations [26], beginning with the simple assumption that pairs of nodes can be randomly connected by an edge with a given connection probability (Table 1). Using a parameter to evaluate the impact of risk on the networked agents, we can estimate the influence of primary risk along the structure as a general failure property [27]. Through scaling for the different evolutionary (Table 2) and nonevolutionary components (Table 3), each step computes a new entity and generates a new proportion in relation to the intervention [3] (Table 4).

2.1. Operating Principles. First, each agent is characterized by two values: capital and strategy. An agent that has lost all of its capital is regarded as failed. Initially, only one agent is given as failed. Each agent is assigned to a node of the network, which is given with a number n of nodes, a connection probability p_d , and a resultant adjacency matrix A . Second, at each time step, units of payoff (c) are added to each agent's capital, of which fractions f_m and f_p are spent on maintenance and protection. The updated capital is expressed as $1 + (1 - f_m - f_p)c$. Third, a failure potential can originate at each node with probability p_n and can propagate through each link with probability p_l . The system may reach one of the two states after the appearance of a random probability (initially failed node of p_n through the link of p_l) and before the second probability occurs. Each failed node with a failure potential becomes a failed node with probability p_r according to the equation below. A failed node losses its capital and remains in a failed state for one time step $p_p = p_{p,\max}/(1 + c_{p,1/2}/(f_p c))$. We can measure capital in units of $c_{p,1/2}$ to obtain $p_p = p_{p,\max}/(1 + 1/(f_p c))$. Fourth, each agent's strategy values, namely, f_{p0} and f_{p1} , are updated through social learning and exploration. Social learning represents the process of choosing a random role model with probability p and imitating the role model's strategy. The probability of imitation is $p_i = 1/(1 + \exp(-s\Delta c))$, where Δc is the difference between the role model's capital and the focal agent's capital and s is the selection intensity. Then, exploration is performed by altering a randomly chosen strategy with probability p_e with a random value chosen from a normal distribution with mean 0 and standard deviation $\sigma_e [f(x|\mu, \sigma^2)]$. The protection level for each agent is determined according to the heuristics $f_p = f_{p0} + f_{p1}C$, where C is the eigenvector centrality of the graph, which is a measure of the influence of a node in a network. If A denotes the adjacency matrix of the graph, the eigenvector centrality or x must satisfy the equation $Ax = \lambda x$, where the vector is normalized to 1. We can normalize this vector to a maximum

TABLE 1: Network property (two parameters).

Number of individuals (nodes)	n	$\in (1, \infty)$
Connection probability (degrees)	$d(p)$	$\in (0, 1)$

TABLE 2: Evolutionary part (four parameters).

Imitation probability	p_r	$\in (0, 1)$
Selection intensity	s	$\in (0, 1)$
Exploration probability	p_e	$\in (0, 1)$
Normally distributed increment	σ	$\in (0, 1)$

TABLE 3: Nonevolutionary part (five parameters).

Maintenance	f_m	$\in (0, 1)$
Propagation probability for each node	p_n	$\in (0, 1)$
Propagation probability through each link	p_l	$\in (0, 1)$
Protection maximum	$p_{p,\max}$	$\in (0, 1)$
Reference point	$c_{p,1/2}$	$\in (0, 1)$

TABLE 4: Time-dependent part (four parameters).

Time periods	t	$1 \sim \infty$
Recovery rate	r_t	1
Recovery time delay	t_r	$1 \sim 10$
Realization	tt	$1 \sim 100$

value, which brings the vector components closer to 1; here, f_{p1} is important for the measurement of f_p (note that we did not explicitly demonstrate this because it has been shown in a previous study [1]). The value for f_p must be truncated to the interval $(0, 1 - f_m)$. Finally, the failure potential, which lasts for only one time step ($r_t = 1$), is controlled by another variable, which we refer to as the recovery time delay ($t_r \in [0, \infty]$). The details of these process-based mechanisms are presented below.

2.2. Process 1. Basic Properties of Created Networks. Individuals in this model are considered as vertices (nodes), and sets of two elements are drawn as a line (edge) connecting two vertices. Data are stored in the nodes, and the edges represent the connections between them, although they can also store data. The edges between the nodes can describe any connection (adjacency) between them (Figure 1(a)). The nodes can contain any amount of data that is assigned to them, and the edges include the data of the strength of the connection they represent.

Connectivity is another essential property of this structure. A disconnected network has some vertices that cannot be reached by the edges from any other vertex (Figure 1(b)).

A disconnected network might have one vertex connected to no edges at all, or it might have two connected networks that have no connection between them. Similarly, a connected network has no disconnected vertices; thus, a metric called connectivity is used to describe a network as a whole, and it depends on the information being presented, usually identified by $(n, d[p])$. Networks also have additional properties, i.e., edges can have a direction, such that a relationship

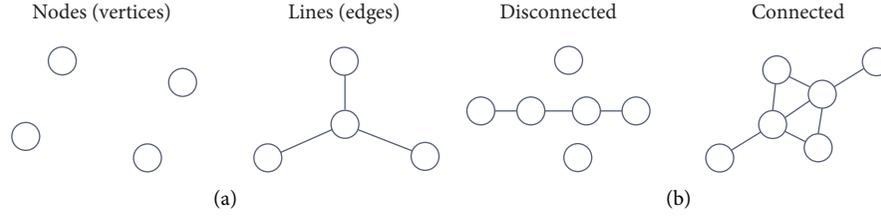


FIGURE 1: Schematic representation of nodes and lines (a) and connectivity (b).

between two nodes is only one-way and not reciprocal. However, in the present model, we used an undirected network, the edges of which have no direction, because in our case, edges are drawn between two individual nodes who have met; hence, all relationships being represented are reciprocal. Thus, the relationship created (network) is undirected and begins with edges randomly drawn between one pair of nodes at a time. For example, four nodes may have edges between them, as shown in Figure 2.

By contrast, if we take an adjacency matrix, we may consider the rows and columns of the matrix to be labeled by the vertices (nodes), giving us one, two, three, or four vertices here. We may use any actual labeling; here, we denote our adjacency matrix by A . The definition of our matrix is that an entry in row m and column n is equal to 1 if there is an edge between m and n , and 0 otherwise:

$$A_{m \times n} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix} \end{matrix}, A = \begin{cases} 1, & \text{if } mn \in \text{edge,} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Note that the adjacency matrix of a network contains all the information contained in that network. Similarly, note that the presented network is given a random appearance because of the way the computer generates the visuals. If the program was to run a second time, a different picture would be generated; however, regardless of how it is run, the same relationship holds between the vertices (nodes) and the edges (line), resulting in the same degrees:

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{matrix}, \deg(v_{2m}) = 3, \deg(v_{2n}) = 3, A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & 1 & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}. \quad (2)$$

The resulting $m \times n$ matrix is obtained as $[A = G(n, p[d])]$, and any node (n) can be randomly linked to any other node. Because the collection of nodes influences the connection probability $[p[d] \in (0, 1)]$, the model investigates the distribution of the connection in the network (probability of degree) (Figure 3).

Figure 4 is based on the standard representation in network theory. Note that the vertices have *degree* d $[A = G(n, d)]$. This d -regular network has degree d for all vertices, and the resulting matrix can be compared to another well-known random network called the Erdős–Rényi network [28], obtained from $[A = G(n, p)]$, shown in Figure 5, where any node (n) can be randomly exposed to any other nodes, creating random connections.

Technically, with respect to the collection of nodes influenced by the connection probability $[p \in (0, 1)]$, it can be observed that both cases with a higher degree (on the right-hand side of the plots) show a higher probability of being connected to an agent. However, for simplicity, the model presents a random regular structure owing to the connection degree $[d \in (0, n - 1)]$ because the network has

all vertices of a certain degree (the same connection probability). In general, various network types (e.g., institutions, firms, banks, food distributors, and supply chains) have ambiguous effects in relation to individual ties because they allow for different ways of diversifying risk; this is particularly true as they influence each other in the same procedure of originating strategies and distributing security [18]. This is considered in the literature to provide strong empirical evidence for a technical approach for comparison with another network (note the lattice and calculation of eigenvector centrality (Table 5)) to obtain a simple numerical solution (Figure 6).

In the following section, analyses are conducted for the case of a random regular network such that agents within the strategy have the same level of diversification [17].

2.3. Process 2. Primary Risk Influence. Next, to observe the propagation process, the model uses an array (vector) to represent the probability of failure $[p \in (0, 1)]$ with given initially influenced nodes $(1 \leq j \leq N)$ denoted merely by

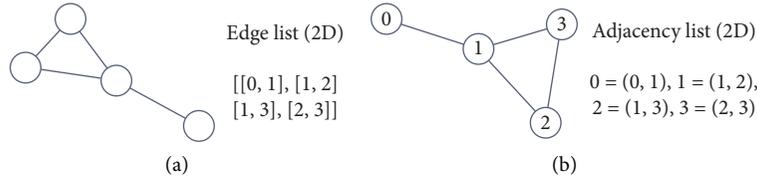


FIGURE 2: Schematic representation of the edge list (a) and the adjacency list (b).

```

=====
# CODE BOOK: Generating the graph
=====
dim = 10                                # linear dimension of the network
n = dim * dim                            # number of nodes
edge = []                                 # edges of the network

# Here, we use a simple lattice as aspects of the graph.

for node in range(n):
    row = int(node / dim)
    col = int(node % dim)

    adj = [ [(row + 1) % dim, col],
            [ row, (col + 1) % dim],
            ]

    for i in [0,1]:
        neighbour = adj[i][0] * dim + adj[i][1]
        edge.append ([node, neighbour])

G = nx.Graph()

G.add_nodes_from(range(n))
G.add_edges_from(edge)

=====
# CODE BOOK: Generating a Random Regular d-Graph
=====
G = nx.random_regular_graph(d = 9, n = n, seed = None) # n = nodes, d = connection degree [p ∈ (0,1)]
edge = np.array(G.edges())                            # create random graph

A = np.array(nx.adjacency_matrix(G).todense())        # resultant adjacency matrix
=====

```

FIGURE 3: Code book: generating the graph and a random regular d-graph.

(p_j). Each node can be in one of the two states: not failed or failed. All nodes are initially without failure.

Extending the above output, given the failure dynamics, the following is obtained:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} \vec{a}_1 \\ \vec{a}_2 \\ \cdots \\ \vec{a}_n \end{bmatrix} = \begin{bmatrix} a_{11}\vec{a}_1 + a_{12}\vec{a}_2 + \cdots + a_{1n}\vec{a}_n \\ a_{21}\vec{a}_1 + a_{22}\vec{a}_2 + \cdots + a_{2n}\vec{a}_n \\ \cdots \\ a_{m1}\vec{a}_1 + a_{m2}\vec{a}_2 + \cdots + a_{mn}\vec{a}_n \end{bmatrix} = B\vec{a} = \begin{bmatrix} B\vec{a}_1 \\ B\vec{a}_2 \\ \cdots \\ B\vec{a}_n \end{bmatrix}. \quad (3)$$

That is, the states matrix:

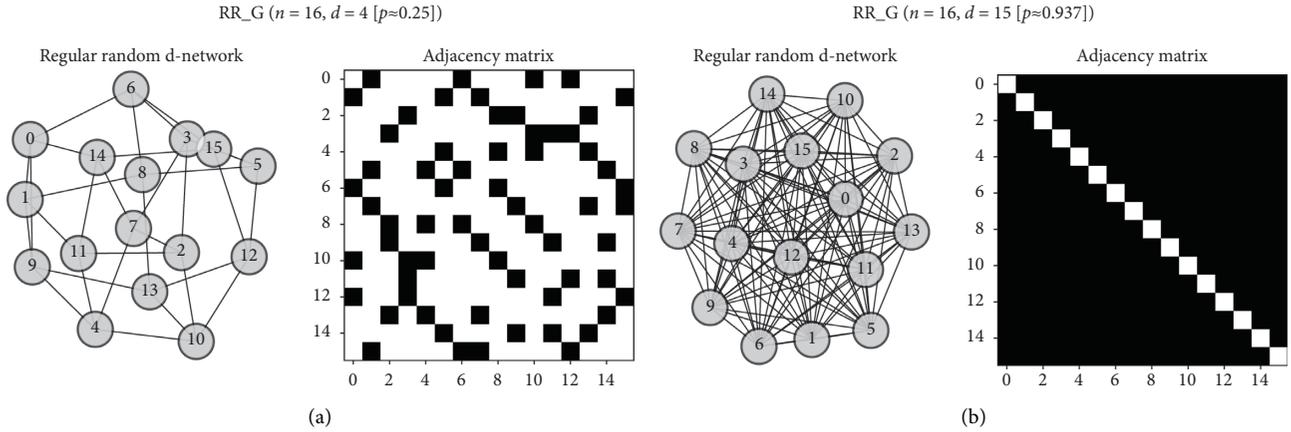


FIGURE 4: Prototype of a random regular network with its properties. The number of nodes $n = 4 * 4$; connection degree (probability) $d = 15$ (b) and $d = 4$ (a). For each section, the plots of the left-hand side show the random (regular) network that was created. The circles represent nodes with an arbitrarily assigned label from 0 to 15, and each line represents a link. The plot on the right-hand side of each section shows the adjacency matrix with entries in row m and column n (either 1 (black) or 0 (white)).

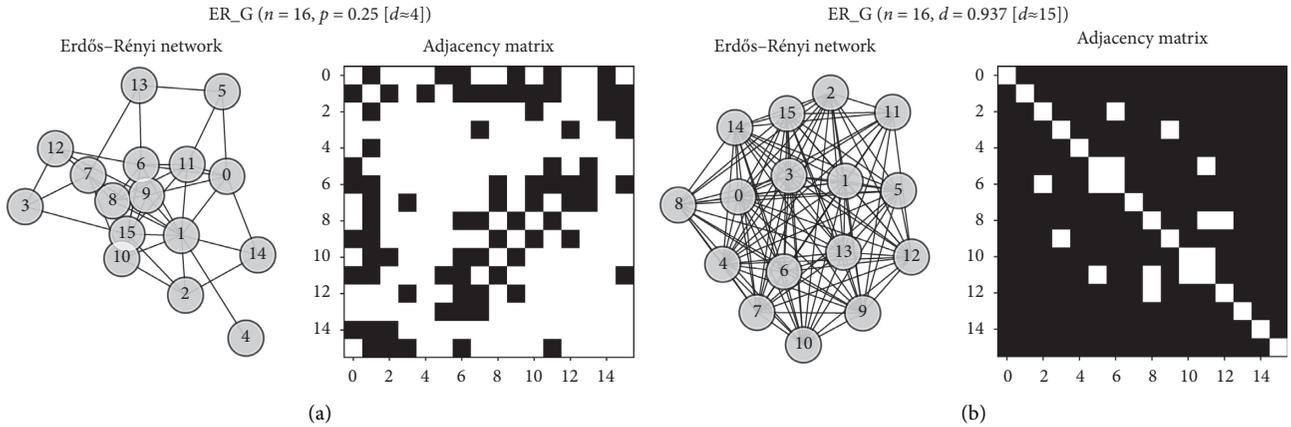


FIGURE 5: Prototype of a random network with its properties. The number of nodes $n = 16$; connection probability $p = 0.937$ (b) and $p = 0.25$ (a). The plot on the left-hand side of each half of the figure shows the random (Erdős-Rényi) network that was created. Each node is represented by a circle, with an arbitrarily assigned label from 0 to 15, and each line represents a link. The plot on the right-hand side of each section shows the adjacency matrix with entries in row m and column n (either 1 (black) or 0 (white)).

TABLE 5: Numerical result for eigenvector centrality [$Ax = \lambda x$].

Node (i)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$d = 4$ [$p=0.25$]	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
$p = 0.25$ [$d=4$]	0.33	0.237	0.165	0.207	0.320	0.223	0.150	0.171	0.212	0.100	0.353	0.227	0.382	0.231	0.110	0.339

d = random regular property obtained from Figure 4(a); p = random property obtained from Figure 5(a).

i.e., States Matrix

$$\begin{aligned}
 & [[1, 1, 1, 1, 0, 0, 1, 1, 1, 1,], \\
 & [0, 0, 0, 1, 1, 1, 1, 1, 1, 1,], \\
 & [0, 0, 0, 0, 0, 0, 0, 0, 0, 0,], \\
 & [0, 0, 0, 0, 1, 1, 1, 1, 1, 0,], \\
 & [0, 0, 0, 0, 1, 1, 1, 1, 1, 1,], \\
 & [0, 0, 0, 1, 1, 1, 1, 1, 1, 0,], \\
 & [0, 0, 0, 0, 0, 0, 0, 0, 0, 0,], \\
 & [0, 0, 1, 1, 1, 1, 1, 1, 1, 1,], \\
 & [0, 0, 0, 0, 0, 0, 0, 0, 0, 1,], \\
 & [0, 0, 0, 0, 0, 0, 1, 1, 1, 1,]
 \end{aligned} \tag{4}$$

```

=====
# CODE BOOK: Create centrality
=====
dim = 10                                # linear dimension of the network
n = dim * dim                            # number of nodes
edge = []                                 # edges of the network

# Here, we use a simple lattice as aspects of the graph.

for node in range(n):
    row = int(node / dim)
    col = int(node % dim)

    adj = [ [ (row + 1) % dim, col],
             [ row, (col + 1) % dim],
             ]

    for i in [0, 1]:
        neighbour = adj[i][0] * dim + adj[i][1]
        edge.append([node, neighbour])

G = nx.random_regular_graph(d=9, n=n, seed=None) # n = nodes, d = connection degree [p ∈ (0,1)]
edge = np.array(G.edges())                       # create random graph

C = nx.eigenvector_centrality(G)                 # calculate eigenvector_centrality normalized [C ∈ (0,1)]
=====

```

FIGURE 6: Code book: create centrality.

```

=====
# CODE BOOK: Risk influence
=====
# for risk dynamics
pn = 0.1                                    # with this, a failure potential can originate at each node
pl = 0.3                                    # with this, a failure potential can propagate along each link

""" Failure potential can originate at each node with (pn) and can produce a failure with (1-pp). The failure potential
propagates along each link with (pl) from a failed node. """

R = (np.random.random(n) ≤ pn)              # randomly choose a certain percentage with a conditional
failure_potential[R] = 1                    # conditional if this occurs
for i in range(n):                          # loop for every individual within the loop with conditional
    if B[i, 3] > 0:                           # if the node fails
        neighbors = nx.all_neighbors(G, i)    # apply network property
        for j in neighbors:                   # find the link for the node (neighbor linked by connection)
            R = np.random.random()           # randomly choose a certain percentage
            if R ≤ pl:                         # if it is (conditional)
                failure_potential[j] = 1     # do this (failure potential goes to)
=====

```

FIGURE 7: Code book: risk influence.

Note that the matrix is denoted by $B\vec{a}$ instead of A because it no longer represents the adjacency matrix. $B\vec{a}$ is continued to be labeled as rows and columns with values of 1 and 0; the key difference is the possibility of showing the state of each node (m : 1 = failure and 0 = absence of failure) according to the time step (n) (Figure 7).

In relation to the fundamental characteristic of the model, we stipulate that an individual (node) can fail if one of its neighbors is infected with failure through a network (Figure 8). An elementary level of risk (or cascades of failure)

depends on the co-occurrence of i and j of the nodes. This implies that individuals are more biased against other individuals that are highly linked in their network.

Next, the probability of failure can be determined by the number of links from the node of the specification scaled by R/S . If we retain the individual characteristics as constant (K), R/S is equal to the risk (failure probability: $p \in (0, 1)$) as a function of the connectivity created by the connections (λx), and R is set to K/S . Nodes at lower (higher) links can be expected to have a lower (higher) connectivity to their risk.

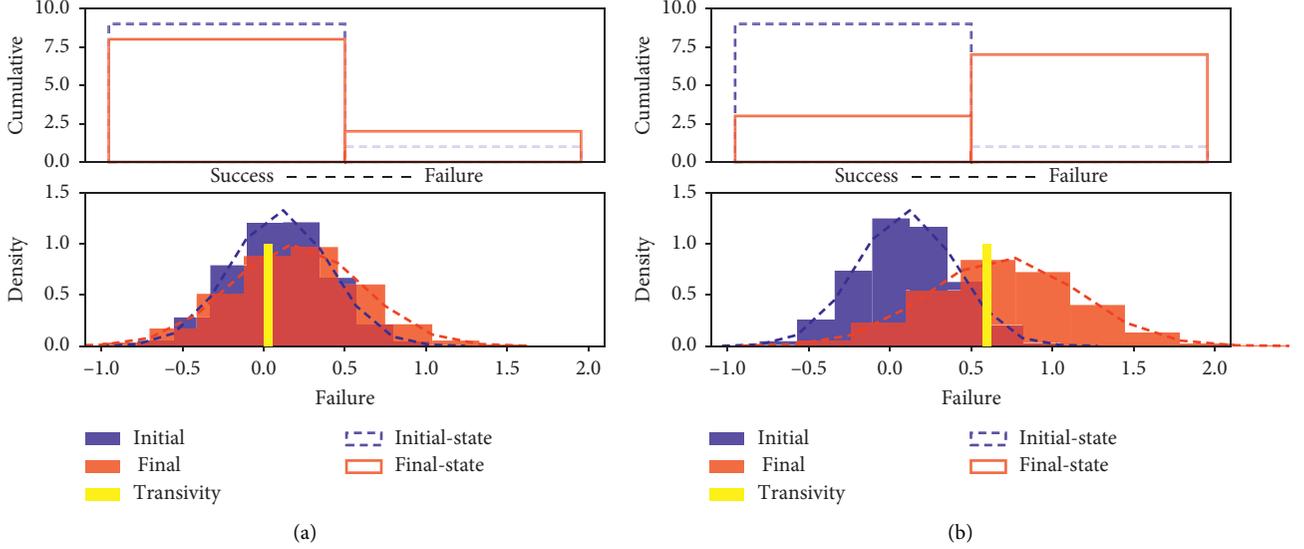


FIGURE 8: Schematic description of failure with distribution. (a) The upper left-hand side plot represents the cumulative number of absences of failure (success) nodes from the initial (blue bar) and final (red bar) cases, and (b) the upper right-hand side plot represents the increasing number of failure nodes from the initial and final cases. The lower plots represent the probability distribution between the initial failure (blue) and the final failure (red), with the likelihood as the transitivity (yellow bar). Note that the influences of these failures are computed as the frequencies of the failure nodes in their initial and final states, i.e., as relative frequencies $[f_j = h_j/N, h_j = \sum_{k=1, \dots, N; X^{(k)}=j} 1]$, where h_j is the frequency of the failure state (j), and the proportion (percentage) of occurrences of failure in the statistical ensemble corresponds to the relative frequency.

In other words, if we remove nodes from the network, the bias is reduced where the links are lower, even if they retain their individual characteristics throughout the process.

2.4. Process 3. Impose Protection against Systemic Risk. Along with the basic intuitions mentioned above, protection dynamics is also implemented. First, we divide the program into the subdynamics (*payoff*, *failure*, and *strategy*). Then, the result of each subdynamics is saved. These subdynamics are trivial problems that add complexity to the dynamics. To implement each of these, we use simple equations. These equations combine the previously computed variables with the newly added or computed variables, for example, $a \rightarrow$ store in the table, $b \rightarrow$ compute c , $d \rightarrow$ store in the table, $a + b = c \rightarrow$ lookup a , $b \rightarrow$ compute c , $d \rightarrow$ store in the

table, $a + d = e \rightarrow$ lookup a , and $d \rightarrow$ compute e . In the example given in Figure 9, we use the values already stored in the table to compute new variables. This technique is often referred to as memorization (Figure 9).

2.4.1. Payoff Dynamics. An agent is associated with each node and is characterized by its capital and strategy as follows: for each time step, each agent receives one unit of payoff, which is added to that agent's capital c , of which fractions f_m and f_p are spent on maintenance and protection, respectively; thus, the capital value is updated as $1 + (1 - f_m - f_p)c$. We used an elementwise computation with arrays for vectorization (\vec{v}) instead of a loop:

$$f_p = \vec{v}_i = \begin{bmatrix} \vec{v}_{i1} \\ \vec{v}_{i2} \\ \dots \\ \vec{v}_{in} \end{bmatrix}, f_m = \vec{v}_{ii} = \begin{bmatrix} \vec{v}_{ii1} \\ \vec{v}_{ii2} \\ \dots \\ \vec{v}_{iin} \end{bmatrix}, \vec{v}_{ii} + (-)\vec{v}_i = \begin{bmatrix} \vec{v}_{ii1} + (-)\vec{v}_{i1} \\ \vec{v}_{ii2} + (-)\vec{v}_{i2} \\ \dots \\ \vec{v}_{iin} + (-)\vec{v}_{in} \end{bmatrix} = \begin{bmatrix} \vec{v}_{iii1} \\ \vec{v}_{iii2} \\ \dots \\ \vec{v}_{iiiin} \end{bmatrix}, c = \begin{bmatrix} \vec{v}_{iii1} \\ \vec{v}_{iii2} \\ \dots \\ \vec{v}_{iiiin} \end{bmatrix} = \begin{bmatrix} c\vec{v}_{iii1} \\ c\vec{v}_{iii2} \\ \dots \\ c\vec{v}_{iiiin} \end{bmatrix} = \begin{bmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \dots \\ \vec{v}_n \end{bmatrix}. \quad (5)$$

The previous initial random network property is as follows:

```

=====
# CODE BOOK: Initial parameters
=====
# Formemorization

B = np.zeros((n, 4))           # create matrix to store parameters
failure_potential = np.zeros(n) # create another matrix to store influence
protection_potential = np.zeros(n) # create another matrix to store protection potential
fp = np.zeros(n)             # create another matrix to store protection level
capital = 1                   # initial capital

# For strategies
fp0 = np.random.normal(0.4, 0.01, size = [n]) # strategy of the one (mean, SD, size = n)
fp1 = np.random.normal(0.5, 0.01, size = [n]) # strategy of the other (mean, SD, size = n)
=====

```

FIGURE 9: Code book: initial parameters.

$$A \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} + kA \begin{bmatrix} ka_{11} & ka_{12} & \dots & ka_{1n} \\ ka_{21} & ka_{22} & \dots & ka_{2n} \\ \dots & \dots & \dots & \dots \\ ka_{m1} & ka_{m2} & \dots & ka_{mn} \end{bmatrix} = B, k \in [0, 1]. \quad (6)$$

Thus, the applied output given the payoff dynamics becomes

$$B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix} \begin{bmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \dots \\ \vec{v}_n \end{bmatrix} = \begin{bmatrix} b_{11}\vec{v}_1 + b_{12}\vec{v}_2 + \dots + b_{1n}\vec{v}_n \\ b_{21}\vec{v}_1 + b_{22}\vec{v}_2 + \dots + b_{2n}\vec{v}_n \\ \dots \\ b_{m1}\vec{v}_1 + b_{m2}\vec{v}_2 + \dots + b_{mn}\vec{v}_n \end{bmatrix} = B\vec{v} = \begin{bmatrix} B\vec{v}_1 \\ B\vec{v}_2 \\ \dots \\ B\vec{v}_n \end{bmatrix}, \quad (7)$$

where the vector (\vec{v} = payoff_dynamics) components are equal to matrix B . This product is equal to $B\vec{v}$ (Figure 10).

2.4.2. Failure Dynamics. The failure potential can originate at each node with probability $p_n \in [0, 1]$, and it also propagates along each link with probability $p_l \in [0, 1]$ at each time step. The failure potential turns into failure with probability $1 - p_p$ depending on the agent's investment in protection: a possible choice is $[p_p = p_{p,\max}/(1 + c_{p,1/2}/(f_p c))]$, where protection (p_p) is equal to the applied (saturation) function. Here, $p_{p,\max}$ is a designated protection maximum, $c_{p,1/2}$ denotes an allocated reference point, and

$f_p c$ represents an evolutionary protection level multiplied by the updated capital:

$$p_p = \vec{u}_i = \begin{bmatrix} \vec{u}_{i1} \\ \vec{u}_{i2} \\ \dots \\ \vec{u}_{in} \end{bmatrix}, f_p = \vec{v}_i = \begin{bmatrix} \vec{v}_{i1} \\ \vec{v}_{i2} \\ \dots \\ \vec{v}_{in} \end{bmatrix}, f_p c = c \begin{bmatrix} c\vec{v}_{i1} \\ c\vec{v}_{i2} \\ \dots \\ c\vec{v}_{in} \end{bmatrix}. \quad (8)$$

The applied output given the failure dynamics is as follows:

$$B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix} \begin{bmatrix} \vec{u}_1 \\ \vec{u}_2 \\ \dots \\ \vec{u}_n \end{bmatrix} = \begin{bmatrix} b_{11}\vec{u}_1 + b_{12}\vec{u}_2 + \dots + b_{1n}\vec{u}_n \\ b_{21}\vec{u}_1 + b_{22}\vec{u}_2 + \dots + b_{2n}\vec{u}_n \\ \dots \\ b_{m1}\vec{u}_1 + b_{m2}\vec{u}_2 + \dots + b_{mn}\vec{u}_n \end{bmatrix} = B\vec{u} = \begin{bmatrix} B\vec{u}_1 \\ B\vec{u}_2 \\ \dots \\ B\vec{u}_n \end{bmatrix}, \quad (9)$$

```

=====
# CODE BOOK: Payoff dynamics
=====
"""Each agent chooses its protection level according to the heuristics  $f_p = f_{p0} + f_{p1} * C$ , truncated to the interval
(0,1- $f_m$ ), where  $C$  is a measure of the centrality of the agent's node normalized to the interval (0,1)."""

 $f_p = B[:,1] + B[:,2] * \text{Centrality}$  # for heuristics of the protection level
 $f_p[f_p < 0] = 0$  # for truncation of the interval (0)
 $f_p[f_p > (1-f_m)] = 1-f_m$  # for truncation of the interval (0, 1- $f_m$ )

"""Each agent receives one unit of payoff, which is added to its capital  $c$ , of which fractions  $f_m$  and  $f_p$  are spent on
maintenance and protection, respectively, resulting in the updated capital  $1 + (1-f_m-f_p) * c$ ."""

 $B[:,0] = 1 + (1-f_m-f_p) * B[:,0]$  # resulting in the updated capital
=====

```

FIGURE 10: Code book: payoff dynamics.

```

=====
# CODE BOOK: Protection potential
=====
# For protection
Centrality = np.array(list(C.values())) # copying C's values to make an array from a dictionary
 $f_p = B[:,1] + B[:,2] * \text{Centrality}$  # for protection level
 $f_m = 0.1$  # for maintenance

# For saturation function
 $p_{max} = 1$  # for protection maximum
 $cp = 1$  # for reference point (=cp, 1/2)

""" Failure potential turns into a failure with probability 1- $pp$ , depending on an agent's investment into protection  $pp = pp_{max} / ((1 + cp, 1/2) / (f_p c))$ ."""

 $B[:,3] = 0$  # node failed
protection_potential = np.zeros(n) # conditional
index = (failure_potential > 0) # return as true or false for all individuals
protection_potential[index] = ( $p_{max} / (1 + cp / (f_p[index] * B[index,0]))$ ) # index true goes to
protection_potential[ np.isnan(protection_potential) ] =  $p_{max}$  # conditional
 $R = ((np.random.random(n) \leq 1 - \text{protection\_potential}) \& \text{index})$  # randomly choose a certain % with conditional

 $B[R, 3] = 1$  # individuals with index true becomes 1
 $B[R, 0] = 0$  # individuals with index true becomes 0
=====

```

FIGURE 11: Code book: protection potential.

where the vector (\vec{u} = failure_dynamics) components are equal to matrix B . This product is equal to $B\vec{u}$. In this section, a prewritten function (regular) is used to create a short iterate $1 = D$ array for vectorization (\vec{u}) instead of using the adjacency matrix directly. This substitution shortens the loop. The failure lasts for one time step (default) and results in the loss of an agent's capital (Figure 11).

2.4.3. *Strategy Dynamics.* Each agent chooses its protection level according to the heuristics $f_p = f_{p0} + f_{p1}C$ truncated to the interval $(0, 1 - f_m)$:

$$\vec{v} \longrightarrow \vec{f} \longrightarrow f(\vec{v}),$$

$$f(\vec{v}) = \begin{cases} 0 < f(\vec{v}) < 0.9, & f_m = 0.9, \\ 0 < f(\vec{v}) < 0.1, & f_m = 0.1, \end{cases} \quad \vec{v} |_{f_p = f_{p0} + f_{p1}C} \quad (10)$$

For initialization of the strategy values, two arrays are added for vectorization ($(f_{p0} = w_i), (f_{p1}C = Cw_i)$):

$$f_{p0} = \vec{w} = \begin{bmatrix} \vec{w}_1 \\ \vec{w}_2 \\ \dots \\ \vec{w}_n \end{bmatrix}, f_{p1}C = C\vec{w} = \begin{bmatrix} C\vec{w}_1 \\ C\vec{w}_2 \\ \dots \\ C\vec{w}_n \end{bmatrix}, \quad C \in [0, 1], \quad (11)$$

$$f_{p0} = \vec{w}_i = \begin{bmatrix} \vec{w}_{i1} \\ \vec{w}_{i2} \\ \dots \\ \vec{w}_{in} \end{bmatrix}, f_{p1}C = C\vec{w}_i = \begin{bmatrix} C\vec{w}_{i1} \\ C\vec{w}_{i2} \\ \dots \\ C\vec{w}_{in} \end{bmatrix}, \vec{w}_i + C\vec{w}_i = \begin{bmatrix} \vec{w}_{i1} + C\vec{w}_{i1} \\ \vec{w}_{i2} + C\vec{w}_{i2} \\ \dots \\ \vec{w}_{in} + C\vec{w}_{in} \end{bmatrix} = f_p = \begin{bmatrix} \vec{v}_{i1} \\ \vec{v}_{i2} \\ \dots \\ \vec{v}_{in} \end{bmatrix}. \quad (12)$$

The eigenvector centrality for node I is $Ax = \lambda x$, where A is the matrix of the network with eigenvalue λ . The strategy values f_{p0} and f_{p1} evolve through social learning and strategy exploration as follows: at each time step, each agent randomly chooses another agent as a role model with probability $p_r \in [0, 1]$ and imitates that agent's strategy values with the following probability:

$$p_i = [1 + e^{-\omega\Delta\pi}]^{-1}, \pi_r - \pi_f = \Delta\pi|_{\pi_r = \text{role model}}, \quad (13)$$

where p_i is the probability of acceptance of the role model for imitation, π_f is the capital of the focal individual, π_r is the capital of the role individual, e denotes the exponential, and ω is the intensity of the selection ($\omega < 1 = \text{weak selection}$; $\omega \infty = \text{strong selection}$). The focal individual imitates the strategy of the nearby role individual, comparing its new capital (large $\Delta\pi = \text{large capital difference}$; small $\Delta\pi = \text{small capital difference}$); then, the focal individual chooses to imitate the strategy of the role individual. In relation to this imitation, a temporary matrix is employed to avoid changing and using one matrix in the loop (Figure 12).

Finally, at each time step, each agent with probability $p_e \in [0, 1]$ randomly chooses one of its two strategy values and alters it with a normally distributed increment with mean 0 and the following standard deviation (Figure 13):

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{(x-\mu)^2}{2\sigma^2}} |x = \text{individual capital},$$

$$\mu \in R = \text{mean (location)}, \sigma^2 > 0 = \text{variance (squared scale)}. \quad (14)$$

2.5. Process 4. Reset Failure (and/or Protection) Potential. A failure lasts for one time step and results in the loss of an agent's capital, according to the reset failure potential and/or failure, as follows:

where \vec{w}_{i1} is a vectorization as the designated strategy of f_{p0} and \vec{w}_{ii} is a vectorization as the designated strategy of f_{p1} , multiplied by the eigenvector centrality from the random graph (C), which is a measure of the centrality of the agent's node, normalized to the interval $(0, 1)$:

$$\text{random}(n) < r_t = 1 \longrightarrow 1 - p_p = 0, \quad (15)$$

where $\text{random}(n) < \text{rec} = 1$ denotes randomly chosen individuals with a certain probability [$\text{random}(n)$], and the failure potential of any individual is $(1 - p_p)$, which is chosen according to a certain probability that approaches 0:

$$t_r < 1 = \text{strong intervention}, \quad (16)$$

$$t_r \longrightarrow \infty = \text{weak intervention}, \quad t_r \in [0, \infty],$$

By default, this recovery rate is implemented by resetting the failure potential after every $r_t = 1$ time step. At the same time, to control this intervention, we allowed the number of time steps to be controlled by another parameter ($t_r \in [0, \infty]$), which represents the recovery time delay (Figure 14).

2.6. Mechanical Insight of the Programming. The broader objective of this step-by-step procedure is to show the computerized process underlying the fundamental mechanisms that are used. The Tables 1–5 and matrix mentioned above account for the details of the rules and procedures (see Appendix 6 for details on the coding). In relation to the technical insights of the implementation, we present the details in Figure 15 that may interest program developers (Figure 15).

The versions on the left- and right-hand sides in Figure 15 represent vectorization using a loop and an array, respectively. Both versions are used in the development of our programming, as can be seen in the code book, which presents simple calculations. Specifically, following the step-by-step procedures of the operation, the version on the left-hand side can be recommended in general; however, it requires more time to simulate many individuals across time steps (see Output in Figure 15). Thus, in some dynamics implementations of models, we used the version on the right-hand side owing to its efficiency.

```

=====
# CODE BOOK: Imitation
=====
# For imitation dynamics
s = 1                                # selection intensity
pr = 0.9                              # imitation probability

"""Each agent randomly chooses another agent as a role model with probability pr and imitates that agent's strategy
values with probability pi."""

R = np.random.random()                # randomly choose a certain percentage only once
if R <= pr:                            # conditional
ff = i                                # focal model (each node i)
while True:                            # it is true
rr = np.random.choice(n)               # randomly choose role model
if ff != rr:                           # until focal choose a different role model
break                                  # exit out of the loop
pi = 1 / (1 + (np.exp(-s * (B[rr, 0] - B[ff, 0]))) # calculate (fermi) function
R = np.random.random()                # randomly choose a certain percentage only once
if R <= pi:                            # conditional
temp[ff, 1:3] = B[rr, 1:3]             # imitate the role model
B[:, 1:3] = temp[:, 1:3]               # update strategy values
=====

```

FIGURE 12: Code book: imitation.

```

=====
CODE BOOK: Exploration
=====
# For exploration dynamics
pe = 0.1                              # exploration probability
mu = 0.0                              # mean for normal increment
sigma = 0.1                            # standard deviation for normal increment

"""Each agent with probability pe randomly chooses one of its two strategy values and alters it by a normally
distributed increment with mean (0) and SD (sigma)."""

temp = B[:, 1:3]                       # a temporary variable to save strategies
R = np.random.random(size = [n, 2]) <= (0.5 * pe) # randomly choose a certain % with conditional
temp[R] += np.random.normal(mu, sigma, size = [n, 2])[R] # normal distributed increment
B[:, 1:3] = temp
=====

```

FIGURE 13: Code book: exploration.

3. Performance (Results)

Given the set of features in relation to the dynamics presented according to the model description, the results of the simulation show the fundamental characteristics of risk diffusion in a randomly networked system and present a framework that enables us to examine the assumptions efficiently while imposing realistic protections against failure. In particular, the simulation characterizes the role of the recovery time delay from the observations of applied dynamics over time to indicate how failure spreads.

First, individuals in the model are considered as vertices, and a set of two elements is drawn as edges connecting two vertices in relation to the information given in the graph. This representation involves two parameters: the number of nodes (n) and the probability that an edge is present (d). In network analysis, indicators of centrality identify the most important vertices within a graph, and their applications

include the identification of the most influential node(s) in a network. The eigenvector centrality of a node is defined as $Ax = \lambda x$, where A represents the adjacency matrix of the network with eigenvalue λ . The principal has an entry for each of the n vertices. The larger the entry for each vertex, the higher it's the ranking with respect to the eigenvector centrality (Figure 16).

Figure 16(a) represents the influences of failures, computed as the frequency of failure nodes in relation to their initial and final states, with the computed relative frequencies $f_j = h_j/N$, $h_j = \sum_{k=1, \dots, N; X^{(k)}=j} 1$, where h_j is the frequency of the failure state (j) and the percentage of occurrences of that failure outcome in the statistical ensemble corresponds to the relative frequency f_j . Figure 16(b) is created to be run in a number of such simulations according to the time step, as one could spread the failure through the network $P(X \in [a, b]; t) = 1/N \sum_{k=1, \dots, N; X^{(k)}(t) \in [a, b]} 1$, with a time series of the number of

```

=====
CODE BOOK: Recovery delay
=====
# for recovery rate
rec1 = 1.0 # always reset (failure potential)
rec2 = 1.0 # never reset (failure)

# fore recovery time delay
failtime = 1 # determines the number of timesteps
failtimear = np.zeros(n) # during which a node is failed
failidx = [] # index of failed nodes

""""A failure lasts for one time step and causes the loss of an agent's capital: reset failure potential and/or failure.""""

R = np.random.random(n) # randomly choose individuals with a certain(%)
index = R7 < rec1 # indexing with conditional
failure_potential[index] = 0 # individuals with index true, becomes 0 (reset the value)

failtimear[B[:, 3] == 0] += 1 # count the failed steps
failidx = ((failtimear % failtime) == 0)
=====

```

FIGURE 14: Code book: recovery delay.

With for_loop	With vector_array
<pre> # for initial parameters B = np.zeros((n,4)) pe = 0.5 mu = 0.0 sigma = 0.01 for i in range(n): R = np.random.random() if R ≤ pe: choose = np.random.randint(1,3) norInc = np.random.normal(mu, sigma, size = None) B[i, choose] = B[i, choose] + norInc </pre>	<pre> # for initial parameters B = np.zeros((n,4)) pe = 0.5 mu = 0.0 sigma = 0.01 temp =B[:,1:3] R = np.random.random(size = [n, 2]) ≤ (0.5 * pe) temp[R] += np.random.normal(mu, sigma, size = [n,2])[R] B[:,1:3] = temp </pre>
↓	↓
<pre> R is: 0.15185496744882165 choose is: 2 0.0104693604787 R is: 0.7683043808698666 R is: 0.13780958929673937 choose is: 1 -0.0147493833134 R is: 0.3237906283769204 choose is: 2 -0.00149137311491 R is: 0.47586950300136266 choose is: 2 -0.00348010349487 R is: 0.3097678759688134 choose is: 2 -0.000587484956137 R is: 0.07287711142245623 choose is: 2 -0.00493456667833 R is: 0.12517600508302607 choose is: 1 -0.00438073778316 R is: 0.04830857294798152 choose is: 1 -0.00497596487175 R is: 0.5264128287834354 [[0. 0. 0.01046936 0.] [0. 0. 0. 0.] [0. -0.01474938 0. 0.] [0. 0. -0.00149137 0.] [0. 0. -0.0034801 0.] [0. 0. -0.00058748 0.] [0. 0. -0.00493457 0.] [0. -0.00438074 0. 0.] [0. -0.00497596 0. 0.] [0. 0. 0. 0.]] </pre>	<pre> [[True True] [True False] [True True] [False False] [True False] [False False] [False False] [False True] [False False] [False False]] [[0.00938266 -0.00177942] [-0.01086615 0.00553024] [0.01279481 -0.00112887] [0.00011118 -0.00223817] [0.00566594 0.00394121] [-0.00104304 0.01093303] [-0.00413213 0.00267854] [0.00614522 -0.00477365] [-0.00086985 -0.00050924] [-0.01461356 0.01471215]] [[0. 0.00938266 -0.00177942 0.] [0. -0.01086615 0. 0.] [0. 0.01279481 -0.00112887 0.] [0. 0. 0. 0.] [0. 0.00566594 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. -0.00477365 0.] [0. 0. 0. 0.] [0. 0. 0. 0.]] </pre>

FIGURE 15: Numerical results using a loop (left side) and an array (right side).

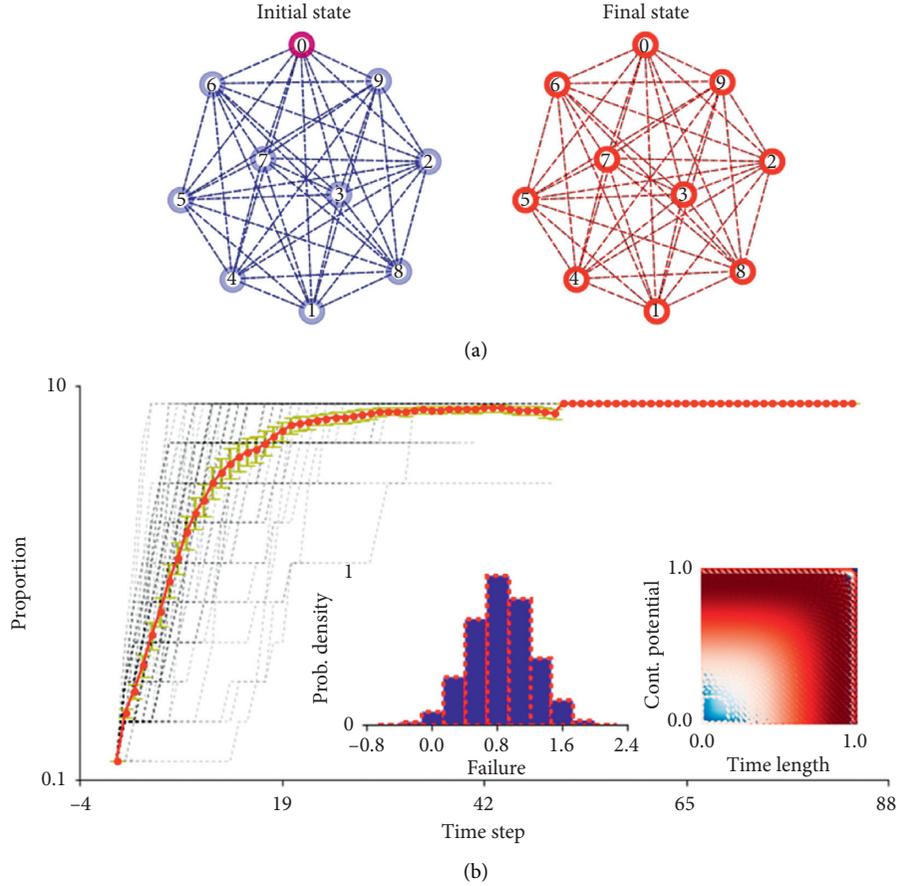


FIGURE 16: Visualization of the propagation of failure over time. (a) The individuals' state from the absence of failure (—blue) to failure (—red). (b) The time series of the number of failures for 100 realizations (black lines) until the number of failed nodes is stabilized. The horizontal axis represents the time step, and the vertical axis represents the proportion of failed nodes (the red dots denote the averages, and the yellow error bars denote the standard deviations). The histogram in the inset on the left-hand side shows the numerical distributions (failure probabilities), and the plot in the inset on the right-hand side shows the potential of failure corresponding to the time length.

failures for 100 realizations (black lines) until the number of failed nodes is stabilized. This implies that the greater the propagation from a higher rank (with respect to degree or centrality), the greater the area drawn between the failed nodes (red) and the nonfailed nodes (blue). This result intuitively implies that, primarily, higher-degree agents have greater exposure to cascading failure risk than lower-degree agents; this increases the potential for cascading failure [29].

3.1. Protection against Failure (Imposing Realistic Dynamics).

In line with our proposition, protection dynamics was applied in reference to the risk of failure. This model allows an agent to make a large investment in protection. Note that we assume that the failure potential from the dynamics becomes failure with probability $1 - p_p$ depending on the agent's investment in protection; $p_p = p_{p,\max}/(1 + c_{p,1/2}/(f_p c))$ when a failure lasts for one time step (Figure 17). The probabilities of these scenarios can be proven as follows.

First, scenario A's failure potential $1 - p_p$ is 0.527 because the investment in protection p_p is 0.473, based on the

possible choice p_p : $p_{p,\max}/(1 + c_{p,1/2}/(f_p c)) = 1/(1 + 1/(0.9*1))$, for certain parameter values ($p_{p,\max} = 1$, $c_{p,1/2} = 1$, $p_r, p_e = 0.9$, $C = 1$). Second, scenario B's failure potential $1 - p_p$ becomes 0.953 because the investment in protection p_p is 0.047 based on the possible choice p_p : $p_{p,\max}/(1 + c_{p,1/2}/(f_p c)) = 0.1/(1 + 1/(0.9*1))$, for certain parameter values ($p_{p,\max} = 0.1$, $c_{p,1/2} = 1$, $p_r, p_e = 0.9$, $C = 1$). As can be seen from the set on the left-hand side of the plot, this scenario has a better result than the set on the right-hand side, corresponding to the coexistence of failure and the absence of the 20 eulist with time. The results imply that contagion and systemic risk are likely to be intensified by the parameter setting (i.e., $p_{p,\max}$, $c_{p,1/2}$, and f_p), resulting in a significant failure cost [30]. Note that the failure potential $1 - p_p$ also changes in relation to the choice of strategy values f_{p0} and f_{p1} based on social learning ($p_r \in [0, 1]$) and exploration ($p_e \in [0, 1]$) because these probabilities can lead to different levels of protection in f_p [1].

3.2. Role of Recovery (Imposing Recovery Time Delay).

Here, we present a specific characterization to determine whether an alternative intervention could affect the failure

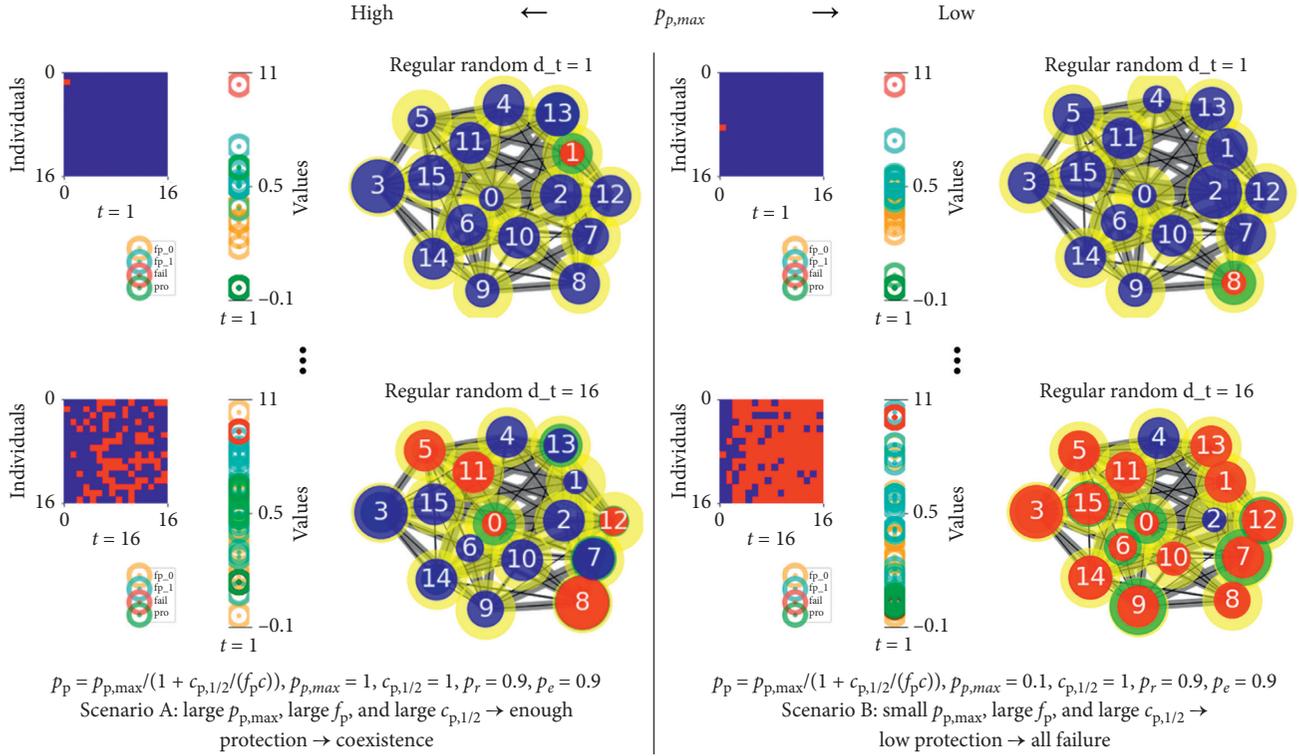


FIGURE 17: Protection dynamics against systemic risk. In each section, the plot on the left-hand side shows a matrix (horizontal axis = time step from 1 to 16; vertical axis = individuals [16]; color of the matrix = failure state: failure [red] and absence of failure [blue]). The plot in the middle shows each individual's parameter values at a given time step ($t = 16$: orange: fp_0 = strategy [1]; cyan: fp_1 = strategy [2]; red: fail = failure; green: protection potential [p_p]). The plot on the right-hand side represents individuals' dynamics within a random network; node number = random label for each node, line width = eigenvector centrality, and node color = states (failure [red] ← → [blue] absence of failure; green = protection potential; yellow = initial structure of the state without failure and protection). The initialized parameters of the simulations are as follows: nodes $n = 10$, connection $d = 0.9$ (% = 5.5), $p_{p,max} = 1$, $f_{p0} = 0.4$, $f_{p1} = 0.5$, $f_m = 0.1$, $s = 1$, $\mu = 0.0$, $\sigma = 0.1$, $p_n = 0.1$, $p_l = 0.3$, and $t = 10$ (see Appendixes 1 and 2 for the results of the entire simulation).

trends observed in coexistence scenario A ($p_{p,max} = 1$, $c_{p,1/2} = 1$, $p_r, p_e = 0.9$, $C = 1$). The model shown in Figure 18 implies that contagion and systemic risk are also likely to be intensified with the recovery time delay, resulting in a significant failure cost. We observe that immediate or delayed intervention is associated with microscale propagation criteria for each node (Figure 18).

In Figure 18, the individuals can be distinguished clearly. Obviously, nodes that have immediate recovery (plots on the left-hand side) appear to have the potential to protect against the propagation of failure; on the contrary, nodes with a malfunction (plots on the right-hand side) do not have such potential. The simulation consisted of repeated trials, and each trial had two possible outcomes: failure (= red) and absence of failure (= blue). The probability of failure is the same for every trial, as in flipping a coin n times, based on the binomial variable that we defined in the model's basic structure. The probability of failure in each trial is given by $P(X) = n!/r!(n-r)!p^r(1-p)^{n-r} = C(n, r)p^r(1-p)^{n-r}$, where n is the total number of trials, r is the total number of failure events, and p is the probability of failure in a single trial.

In the plots on the left-hand side in Figure 18, the probability of absence of failure is approximately 0.6, and the probability of failure is 0.4 (the case probability for Scenario

A is calculated by the number of failures from the time steps: $f_j = h_j/N$, $h_j = \sum_{k=1, \dots, N; X^{(k)}=j} 1$). We assume a random variable X , which is equal to the number of failures after a certain number of time steps. One of the first conditions of this result is that it consists of a finite number of independent trials. This means that the probability of obtaining failure or absence of failure in each trial is independent of whether we obtained failure in a previous trial. Thus, in the case of the plots on the left-hand side with recovery at every time step (immediate intervention), the results of the simulation show independent trials. Another condition is that each trial clearly has one of the two discrete outcomes in which the variable X should be clearly classified as showing either failure or absence of failure with a given fixed number of trials. Then, the final condition of the probability of failure and absence of failure for each trial is constant, as we have already obtained measurements for each trial from the cases of Scenarios A and B.

Proof of the role of the recovery rate. We now examine the frequency of failed agents observed on the left-hand side in Figure 18. Suppose that at a given time, there are failed nodes (N_f) and nonfailed nodes ($N - N_f$). The number of failed nodes at $t + 1$ is given by

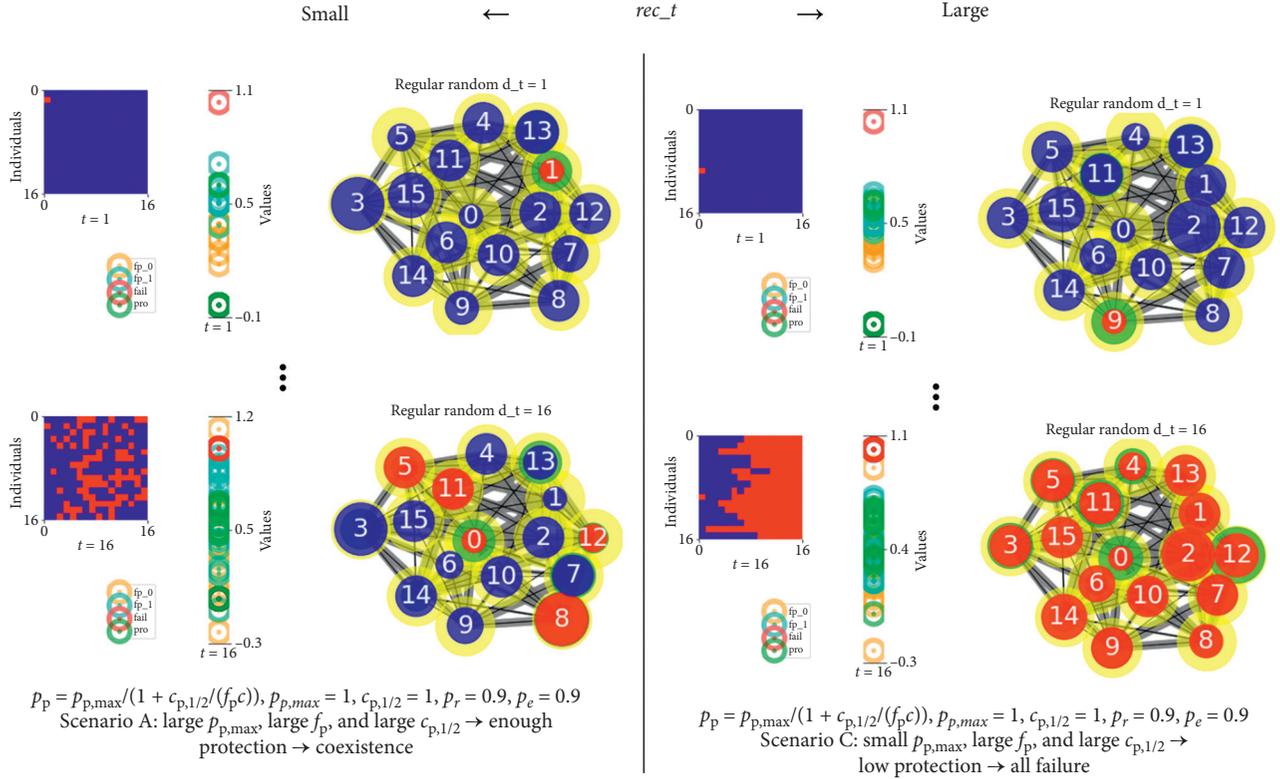


FIGURE 18: Dynamics of protection against systemic risk with recovery time delay. Using the same initial parameter values ($p_{p,max} = 1$, $c_{p,1/2} = 1$, $p_r = 0.1$, $p_e = 0.1$, $C = 1$), this set of the plots represents the time delay case (left-hand side: $r_t = 1$; right-hand side: $r_t = 5$). In each section, the plot on the left-hand side shows the matrix (horizontal axis = time step from 1 to 16; vertical axis = individuals from 1 to 16; color of the matrix = failure state: failure [red] and absence of failure [blue]) corresponding to each individual's parameter values at the given time step ($t = 16$: orange: fp_0 = strategy [1]; cyan: fp_1 = strategy [2]; red: fail = failure; green: pro = protection potential [p_p]). The graph on the right-hand side represents the dynamics in a random regular network, such that the node number = random label for each node and the node color = state (failure = red, absence of failure = blue) (see Appendix 3 for the results of the entire simulation).

$$\begin{aligned} p_A(n) &= N_f = 0.731 = N_f + p_p(N - N_f) \\ &= 0.731 + 0.5(1 - 0.731) = 0.865 = N_{NoF}, \end{aligned} \quad (17)$$

where p_p denotes the protection probability (arbitrarily designated as 0.5 only for this numerical calculation); conversely, the number of failed nodes is

$$(1 - p_p)(N - N_f) = (1 - 0.5)(1 - 0.731) = 0.135 = N_f. \quad (18)$$

Failure is propagated through an existing link with probability p_i ; at the same time, a link exists with a certain probability through the created network (p_{RR} , RR = random regular graph). Therefore, if we impose the condition that the failure potential can propagate through each link, and the propagation probability p_l is less than 1 (when the original probability p_n = constant), this equation can be modified simply by replacing p_p with

$$p'_p = 1 - (1 - p_p)(1 - (1 - p_l p_{RR}))^{N_f}. \quad (19)$$

For example, when we arbitrarily consider the protection $p_p = 0.5$ and failure propagation (p_l) through a random

network (p_{RR}) to be sufficiently high ($p_l p_{RR} = 0.9$), the protection influenced by the failure propagation p'_p becomes small because $1 - (1 - 0.5)(1 - (1 - 0.9))^{N_f} = 0.09$. On the contrary, if a weak failure propagation occurs, such as $p_l p_{RR} = 0.1$, the protection influenced by the failure propagation p'_p increases because

$$1 - (1 - 0.5)(1 - (1 - 0.1))^{N_f} = 0.95. \quad (20)$$

However, in the plots on the right-hand side in Figure 18, the probability is no longer the same but changes from trial to trial. We have the variable $X(p_l)$, which is equal to the number of failures from a designated population. It seems to represent the same operation because, with it, each trial can be classified as either a failure or an absence of failure over a fixed number of trials ($t = 16$). At the same time, there is the probability that the variable $p_l p_{RR}$ is not constant for each trial owing to the recovery time delay, which does not consist of independent trials. The probability of failure or absence of failure in the first trial would be equal to the total number of individuals between the two simulation cases:

$$p_l p_{RR}(k \text{ on } 1^{\text{st}} \text{ trial}) = 0.1. \quad (21)$$

However, the probability of the second trial (and the following trial) would not be the same because the simulation of the right-hand side case depends on what happened in the first (previous) trial:

$$p_l p_{RR}(k \text{ on } 2^{\text{nd}} \text{ trial}) \neq \text{constant}. \quad (22)$$

In other words, each trial is carried out without replacement, which results in an exponentially large difference between the two cases. Thus, this result does not meet the independent condition: the probability present in the following trial depends on what happened in the previous one. Because replacement does not take place, the probability of failure for each trail is also not constant, unlike the simulation on the left-hand side, where the probability of failure is constant:

$$p_l p_{RR}(k \text{ on } 2^{\text{nd}} \text{ trial}) = \text{constant}. \quad (23)$$

Inspired by the plausible scenarios included in the recovery delay for cases A and C in Figure 18, we focus on the comparison of the parameter $p_{p,\max}$ (when recovery delay constant = 1) with more time steps ($t=100$). This is because of the applied function $p_p [= p_{p,\max}/(1 + c_{p,1/2}/(f_p c))]$ for this application, which will primarily be decided by $p_{p,\max}$ when we consider $c_{p,1/2}/(f_p c)$ and the recovery time as constants (Figure 19). This refers to the individual for control by interventions to be protected against the risk of failure when the failure propagation mechanism influences agents' decisions as they are generated.

Figure 19 shows a scatter plot constructed using $p_{p,\max}$ (Figure 19(a)) with a clear negative correlation between failure and capital. Although it is constructed by the recovery time delay (Figure 19(b)), strong power law relationships are still obeyed in cascading failure, even if the capital values remain strong (see the averaged values for the failure and capital in the lower part of Figure 10). In the following results, this trend (failure and capital) is determined by two parameters ($p_{p,\max}$, rec_t); however, their influences do not have the same weight. Here, we should note that intervention applied to recovery can persist for a short period of time, which we could cite as a bias or rationality.

3.3. Generalize with Stationarity. Let us assume that the system has evolved and reaches stationarity, which implies that all variables have nearly constant fluctuation around their mean values. In stationarity, the state of the system can be considered independently of the initial conditions. It is useful to obtain a relationship among the variables in stationarity. First, we denote the number and fraction of failed agents by N_f and f , respectively. For stationarity, the average number of failed agents remains constant. Consider the two successive steps t and $t + 1$. Suppose that at time t , there are N_f and $N - N_f$ failed and nonfailed nodes, respectively. Therefore, the number of nonfailed nodes at $t + 1$ is $N_f + p_p(N - N_f)$, and the number of failed nodes is $(1 - p_p)(N - N_f)$. Stationarity means that the number of failed (or nonfailed) nodes remain constant (on average), i.e.,

$$N_f = (1 - p_p)(N - N_f), N_f = \begin{cases} p_p \approx 0.9, & N_f = \downarrow, \\ p_p \approx 0.1, & N_f = \uparrow, \end{cases} \quad (24)$$

$$f = \frac{1 - p_p}{2 - p_p} = \frac{(1 - p_{p,\max})f_p c + 1}{(2 - p_{p,\max})f_p c + 2}, f = \begin{cases} (2 - p_{p,\max}) \approx \uparrow, & f = \downarrow, \\ (2 - p_{p,\max}) \approx \downarrow, & f = \uparrow. \end{cases}$$

Let us define $q_p = 1 - p_p$. The above formula for f can be written as $f = q_p/(1 + q_p)$, which is simple. Here, p_p is the average probability of protection among the nonfailed nodes. Note that the protection probability for the failed nodes is zero because they have no capital. In other words, $p_p = \sum p_p(i)/N - N_f$, where $p_p(i)$ represents the protection probability for node i . In the second equality, f_p and c are defined similarly.

This relationship shows that the stationary value for f is bounded in the interval $[0, 1]$ and is a decreasing function of p_p . The limiting values seem promising. For $p_p = 1$, i.e., full protection, there is no failed agent. For $p_p = 0$, i.e., no protection, there is a failed agent. Note that at the end of each time step, all the failed agents recover (as an initial setting). If the propagation probability $p_l p_{RR}$ is less than 1, this equation should be modified by replacing p_p as follows:

$$p'_p = 1 - (1 - p_p)(1 - (1 - p_l p_{RR})^{N_f}),$$

$$p'_p = \begin{cases} p_l p_{RR} \approx 0.9, p'_p = \downarrow, \\ p_l p_{RR} \approx 0.1, p'_p = \uparrow. \end{cases} \quad (25)$$

Again, by defining $q'_p = 1 - p'_p$, we implicitly obtain f . To check the validity of these formulae with a simulation, we increase the total simulation time ($t=1,000$); for smaller $p_l p_{RR} < 1$, a longer time is required to reach stationarity (Figure 20).

Proof of the stationary state. we calculate the average value of the capital in the stationary state to prove the long-term role of the recovery time delay. First, consider the case $p_l p_{RR} = 1$. It can be easily seen that the average value of

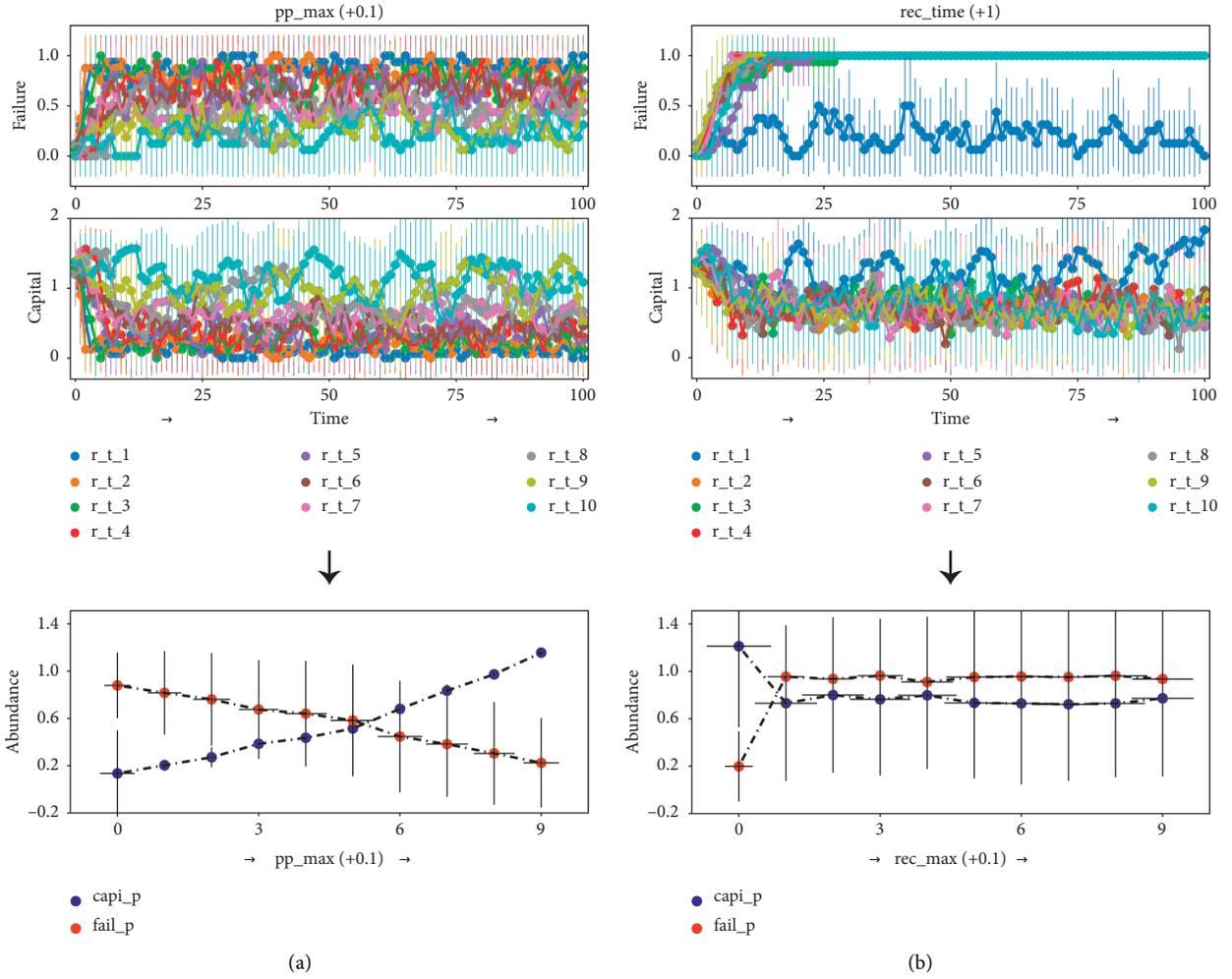


FIGURE 19: Representation of the outcomes according to the recovery time delay (b) and protection maximum (a). Using the same initialized parameter values (no delay [$r_t = 1$]), the plots represent the time delay ($r_t = 1 \sim r_t = 10$) case. The upper plot shows failure (vertical axis = value of the failure corresponding to the time step (horizontal axis)). The middle plot shows the capital (vertical axis = value of the capital corresponding to the time step [horizontal axis]). The lower plot shows the averaged outcome values (vertical axis) for capital and failure according to the recovery time delay ($rec_time[+1]$) and the protection maximum ($pp_max[+0.1]$) (see legend). All the dashed lines denote variability.

the capital in the stationary state satisfies the following equation:

$$c_{in} + p_p(1 - f_p - f_m)c = c,$$

$$c = \begin{cases} p_p \approx 0.9, & c = \uparrow, \\ p_p \approx 0.1, & c = \downarrow, \end{cases} \quad (26)$$

$$c = \begin{cases} f_p \approx \uparrow, & c = \downarrow, \\ f_p \approx \downarrow, & c = \uparrow, \end{cases}$$

where c is the average value of the capital among nonfailed individuals. Combining the above equations, we can obtain c and f_p as a function of f in the stationary state. However, it is easier to obtain c as a function of p_p , such that the protection probability should be replaced by $p'_p = 1 - (1 - p_{lPRR})(1 - (1 - p_{lPRR})^{N_f})$ when $p_{lPRR} < 1$. Therefore, we

obtain the following (note that in the simulation, we use a truncation of f_p to constrain it to the interval $[0, 1 - f_m]$ [see code book]):

$$c = \frac{p_p p'_p - c_{in}(p_{p,max} - p_p)}{(p_{p,max} - p_p)(p'_p(1 - f_m) - 1)}, \quad (27)$$

$$f_p = \frac{p_p(p'_p(1 - f_m) - 1)}{p_p p'_p - c_{in}(p_{p,max} - p_p)}$$

According to the trajectories (upper part of Figure 20), the outcomes such as failure (or capital) reach a stationary state even if the trajectories from the strategies continue to evolve and exhibit different behaviors (see the inset of the upper plots). Therefore, the presence of the detailed dynamic features of the system (capital and failure) indicates

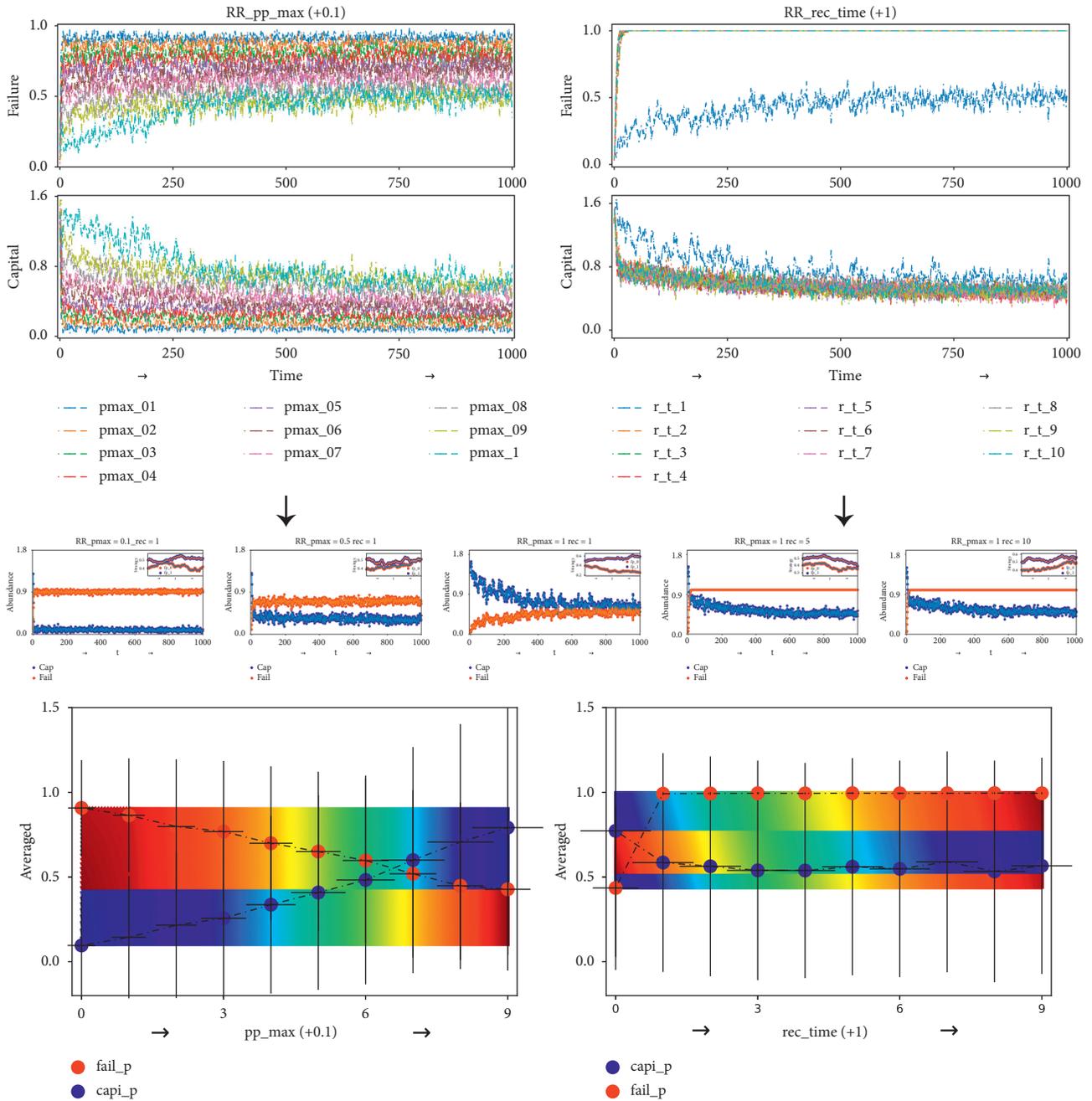


FIGURE 20: Stationarity in time series with the random regular graph (nodes = $10 * 10$, connection $d=9$, time steps = 1,000). The set on the left-hand side shows the results controlled by the protection maximum, and the set on the right-hand side shows the results with a recovery time delay (dots = averaged values; dashed lines = variation). The upper plot shows the failure and capital combined (vertical axis = value corresponding to the time step [horizontal axis]). The plot in the middle shows failure (red) and capital (blue) (vertical axis = value corresponding to the time step [horizontal axis]). The inset plot shows the trend of the two strategies (f_{p0} = red; f_{p1} = blue) (see Appendixes 4 and 5 for the results of the entire simulation). The lower plot shows the averaged outcomes (vertical axis) for capital (blues) and failure (reds), according to the recovery time delay (rec_time[+1]) and protection maximum [pp_max(+0.1)] (see legend). The background contour shows the dominance between the outcomes, and all the dashed lines denote the variability.

the possibility of stationarity in the coevolutionary process. This observation can enable us to gain a sense of the qualitatively different nature of propagation (compared to $p_{p,max}$) within systems, such that failure is not limited to protection or strategy but is due to regulation over time. In other words, there is another rate that continues to

increase the bias as the length of the delay causes the cascading failure, even where the individuals still have potential (Figure 21).

As indicated by the protection (Figure 21; bottom, marker of green, nodes that are controlled by $p_{p,max} = 0.1$ (left-hand side) seem to lose their ability to protect, owing

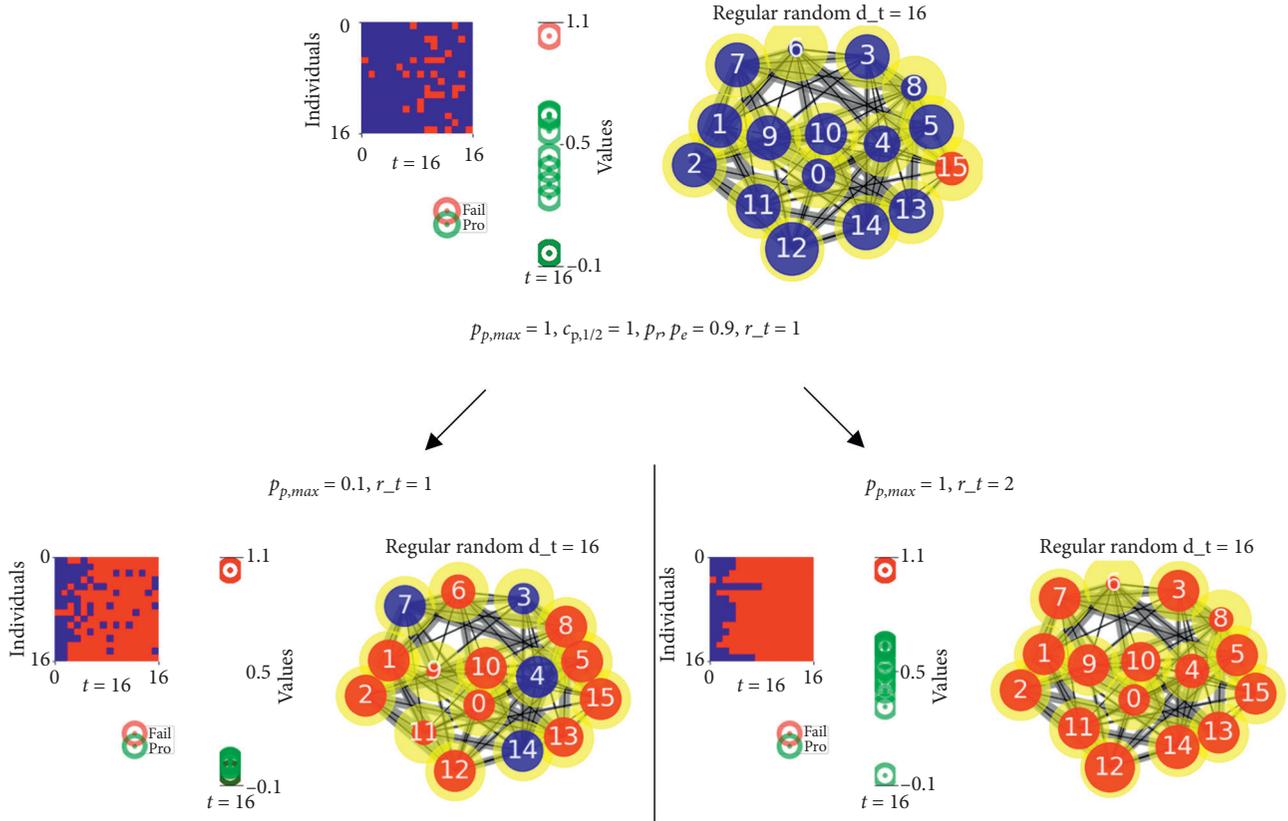


FIGURE 21: Representation of protection potential constructed by $p_{p,max}$ (lower set) and t_r (lower right-hand side). With the same initialized parameter values (upper plots: $p_{p,max} = 1$, $c_{p,1/2} = 1$, $p_r = 0.9$, $p_e = 0.9$, $C = 1$), the set of plots represents the time delay (left-hand side: $r_t = 1$; right-hand side: $r_t = 2$) case. In each section, the plot on the left-hand side shows a matrix (horizontal axis = time step from 1 to 16; vertical axis = individuals from 1 to 16; color of the matrix = failure state: failure [red] and absence of failure [blue]) corresponding to the parameter values of each individual at the given time step ($t = 16$: red: fail = failure; green: pro = protection potential). The graph on the right-hand side represents their dynamics in a random regular network; node number = random label of each node marked with a yellow background; lines = connections embedded by eigenvector centrality as the thickness; node color = states (failure = red; absence of failure = blue).

to the propagation of failure. On the contrary, nodes that are controlled by $t_r = 2$ (right-hand plots) seem to maintain their potential, even if they all feature cascading failure. We noted that the different levels of potential (between the two parameters $[p_{p,max}, t_r]$) came from their capital, as marked in blue in Figure 20 (see bottom plots of the figure).

Thus, to reduce the potential ramifications (p_p') of such additional losses to others, an immediate recovery intervention may not only be preferred to the potential damage ($1 - p_p'$) from individual failures but also guarantee that it is strategically possible for even large insolvent individuals to recover losses to uninsured connectors. Before the potential for failure can advance the propagation value, it must be identified, and the recovery value of the individual protection potential (i.e., capital) must be estimated.

4. Discussion

We presented a simple general model to quantify the protection for mitigating systemic risk, with the recovery time delay as the result. Using a simple set of properties, the

observations from this model indicate how probability describes the proportion of protection that can be characterized by how systemic risk should be coped with rather than being predicted by the probability of failure [31].

4.1. Summary of the Model. We proposed the following process-based steps (1–4) to create a model. First, *network-agent properties* were established: when the basic data structure was constructed, the functionalities of the mechanism began with a specific undirected relationship between agents through agent-based simulations, in a coexisting macroscale structure with individuals interconnected at the microscale level in the network. Next, *the primary risk influence* was established using a parameter to evaluate the impact of risk on the networked agents; the influence of primary risk was estimated along the structure as a general failure property. Then, it was necessary to implement *protection against systemic risk* by embedding protection dynamics that emphasizes the roles of payoff, failure, and strategy dynamics. Finally, *recovery* was considered through scaling for the different evolutionary and nonevolutionary components; this is crucial to the way this system functions, where each step computes a new entity and

generates a new proportion in relation to the intervention. The implications of this model are as follows.

4.2. Theoretical Implications. A suite of plausible dynamics and decentralized bottom-up mechanisms was constructed by establishing appropriate rules for the interaction, within which the system components can self-organize, including mechanisms for ensuring rule compliance (vectorized microscale implementation). This evolutionary heuristic promotes balance with respect to the interactions [32]. For example, when investment in protection is weak (low investment), a pattern of strong systemic risk emerges in the form of agent failure in networked conditions. By contrast, when investment in protection is strong (high investment), a pattern of protection emerges, with little diversification against all challenges (Figure 17). The simulation also reflected a clear correlation between a set of parameter values (between those for capital and failure and between those for the strategies of f_{p0} and f_{p1}). Thus, the strategy of social learning could be another crucial factor in resource provision that violates expectations and leads to novel trends with high impact. These results shed light on the propagation modes. The observed contagion and persistence patterns can be regarded not as a direct causal link but in relation to an accumulated rational driven by interconnectedness [33].

More importantly, in this study, the failure potential also reflects the time delay after the official failure of an individual [34]. Newly damaged individuals may affect others or may recover from the next time step onwards (recovery occurs with the rate t_r). The potential needed for recovery is related to the number of healthy individuals $n(t)$, and if individuals recover at time $t_r = 1$, the protection potential changes as follows:

$$c(t+1) = c(t) + n(t) - [1 - pp(t)]. \quad (28)$$

Once the capital is exhausted, the systemic risk increases rapidly. This is based on suitable real-time intervention related to short-term anticipation of risk flows [35]. However, because regulation occurs over time, another rate itself increases the failure potential even if the capital is still exhausted (note that the recovery rate mainly influences failure and does not have the same weight for capital). From our observations, we assume that the protection (or robustness) of an agent π_i decreases (or increases) by an amount proportional to the relative exposure of risk sharing η_{ij} . Consistent with the network of dynamics described above, the dynamics, including the recovery time delay (in time series), can be formalized as follows:

$$\begin{aligned} \pi_i &= \pi_i(0) - \frac{a}{k} \sum_j \eta_{ij} \chi_j(\tau), \\ \pi_i &= \begin{cases} \tau \approx \uparrow, & \pi_i = \downarrow, \\ \tau \approx \downarrow, & \pi_i = \uparrow. \end{cases} \end{aligned} \quad (29)$$

For convenience, with respect to the results, k substitutes the protection potential, defined by $c(t+1) = c(t) + n(t) - [1 - pp(t)]$, such that the function a determines the extent of the loss caused by the default $\chi_j(\tau)$ when we fix risk sharing η_{ij} . This indicates that the agent has defaulted at a previous time, denoted by the time variable τ . In the simulations, there was no delay during the development of the cascade ($\tau \approx 0, \tau = t_2 - t_1$). However, this dynamic differs across the default of the recovery [36]. If the system delays the intervention, the cascade tends to be much larger (following the power law) because the defaults of the neighbors of a given agent are statistically dependent on j , such that their default is more likely to cause the default of others:

$$f'(x) = \left(\frac{k}{x}\right) + c, \quad x = r_{t(t+1)}, c = 1. \quad (30)$$

In Table 6, such a range $[f'(x)]$ will be determined by failure propagation through each link in the random regular network ($p_I p_{RR}$), which was determined by the recovery time delay. We simply proved that this numerical trend of the protection probability defined by $p'_p = 1 - (1 - p_p)(1 - (1 - p_I p_{ER})^{N_f})$ could not be the same as that of $p_{p-\max}$ [when $p_I p_{ER} = \text{constant}$ (Figure 20)]. Thus, the failure influences in these simulations are not identical, as $p_{p-\max}$ changes according to these parameters, including where increases or decreases occur following a nonlinear curvature [37].

4.3. Practical Implications. We propose a modified version of the protection model against systemic risk in terms of its failure recovery mechanics, as demonstrated above. The mechanisms here are based on a few simple basic rules: a node fails at a time with a probability of failure if its failure potential is greater than or equal to that of its nearest neighbors [1], and it fails owing to the interconnected potential, although it spontaneously recovers at an external recovery or according to an intervention probability [3]. The consequences of the damage are crucial for systemic risk and controllability; however, they have not been explored systematically thus far [38]. We show how the process of embedding and the related recovery times impact the dynamics of failure processes in a network (see the section on the proof of the role of the recovery time delay). Therefore, we propose that the extent of the intervention regime in the system can be a source of evolutionary (in)stability, wherein the immediate recovery of components can mitigate damage and propagate failure in dynamic networks [17]. These decentralized management principles could be applied to logistical and production systems, or even to administrative processes and governance [18].

Governments are often reluctant to resolve insolvency in institutions [e.g., banks, firms, supply chains, and infected virus individual (i.e., Ebola, SARS, and COVID-19)], and they permit them to continue operating despite their negative effects. The length of the attendant delay causes a cascading failure, regardless of each individual's capital (i.e., immunity), which then leads to reductions in network welfare. These delays may propagate the damage to many

TABLE 6: Numerical result of the function $[f'(x)]$.

Domain ($x = r_t$)	1	2	3	4	5	6	7	8	9	10
Numerical $f'(x)$	2	1.333	1.2	1.142	1.111	1.09	0.1076	1.066	1.058	1.050
Fixed mean (capital)	0.772	0.585	0.563	0.539	0.538	0.561	0.548	0.589	0.533	0.566

other individuals in the network, increasing their fragility and probability of failure [39]. Evidence from governing systems suggests that if individuals' troubles are estimated before their capital becomes negative, institutions could, based on their risk potential, weed out the inefficient or unfortunate individuals to avoid more serious adverse effects [40]. This indicates the importance of resolving challenges to individuals as quickly as possible and developing more rapid responses to certify protected individuals by providing immediate intervention. On the contrary, prompt corrective action can increase the willingness to supply protection to reduce the chances of systemic risk [41]. Given the evolutionary mechanisms shown in this simulation model, we observed that the evolutionary response often obtains a critical value for a plausible protection potential. We demonstrated that, although structures have high potential for individual protection and a strategy to maintain their capital, the function of the interconnected recovery delay makes them weak amplifiers, where unprofitable intervention shows a high bias.

5. Concluding Remarks

Cascading failure was used to assess how rules and processes can collectively add up to a series of interconnected unsuitable outcomes. In the simulations, although every object was considered to be doing well individually, systemic risks could emerge, as the whole was vast. In other words, even if every single object is considered to be highly fitted and behaving properly (i.e., capital), there is no standard solution for proper management for everyone's benefit. Therefore, the elimination of resources needed for recovery requires a strong effort to halt cascades as they begin, when the damage is still small, and the problem may not yet be perceived as threatening. These issues can and must be treated with a proper (re)design of the system and adoption of management principles, as shown by the simulation results [42].

Many disasters in the real-world result from incorrect thinking and inappropriate system design. To address such risks more completely, a better understanding of proper intervention and resilience is crucial. However, there remains a lack of an effective method for calculating networked risk. The proposed model can facilitate realistic calculation of the interdependence and propagation of risks in a network and how they can be absorbed, and it can mediate such that both the system components and the plausible systemic intervention and outcomes will work well.

We derived a unifying framework for the interplay of observations that embed realistic dynamics such as payoff, failure, strategy, and recovery in a random regular network. The mechanics showed that the complexity should be

described by the essential features of the model's processes that capture evolutionary damage spread owing to the co-existence of crucial hypotheses in the system. Thus, it may be possible to create an account where protection and failure are not static quantities, and propagation is likely to be understood in terms of how frequently the system is in a condition that leads to a large diffusion. This availability of mechanisms raises additional expectations that predictability and controllability are a simple matter of proper system design and operation [43]. More intuitively, this study can provide a better understanding of recovery, where real-time management can overcome instabilities caused by delays in feedback or lack of information.

Data Availability

The data used to support this study can be obtained from the corresponding author upon request.

Ethical Approval

This study was approved by the local ethics committee (SNUIRB No.1509/002-002) and conformed to the ethical standards of the 1964 Declaration of Helsinki (Collaborative Institutional Training Initiative Program, report ID 20481572).

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (Grant Number: 2020R111A1A01056967, PI: Chulwook Park).

Supplementary Materials

Appendix 1: the results of the entire simulation for protection dynamics against systemic risk (left-hand side of Figure 17). Appendix 2: the results of the entire simulation for protection dynamics against systemic risk (right-hand side of Figure 17). Appendix 3: the results of the entire simulation for protection dynamics against systemic risk (right-hand side of Figure 18). Appendix 4: stationarity controlled by pp,max: (nodes = $10 * 10$, connection $d = 9$, time step = 1,000). Appendix 5: stationarity controlled by rec_t: (nodes = $10 * 10$, connection $d = 9$, time steps = 1,000). Appendix 6: code book. (*Supplementary Materials*)

References

- [1] C. Park, "Network and agent dynamics with evolving protection against systemic risk," *Complexity*, vol. 2020, Article ID 2989242, 16 pages, 2020.
- [2] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, no. 7291, pp. 1025–1028, 2010.
- [3] D. Helbing, "Globally networked risks and how to respond," *Nature*, vol. 497, no. 51, p. 7447, 2013.
- [4] P. C. Trimmer, A. I. Houston, J. A. R. Marshall, M. T. Mendl, E. S. Paul, and J. M. McNamara, "Decision-making under uncertainty: "biases and Bayesians," *Animal Cognition*, vol. 14, no. 4, pp. 465–476, 2011.
- [5] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, "Epidemic processes in complex networks," *Reviews of Modern Physics*, vol. 87, no. 3, pp. 925–979, 2015.
- [6] Å. Brännström, H. Sjödin, and J. Rocklöv, "A Method for Estimating the True Number of Infections from the Reported Number of Deaths with Application to COVID-19. Submitted to Eurosurveillance," 2020.
- [7] K. Sneppen and L. Simonsen, "Impact of Superspreaders on dissemination and mitigation of COVID-19," 2020.
- [8] J. Lloyd-Smith, S. Schreiber, P. Kopp, and W. Getz, "Superspreading and the effect of individual variation on disease emergence," *Nature*, vol. 438, pp. 355–359, 2020.
- [9] T. Britton, "The Disease-Induced Herd Immunity Level for Covid-19 Is Substantially Lower than the Classical Herd Immunity Level," 2020.
- [10] H. Sjödin, A. Johansson, Å. Brännström et al., "COVID-19 healthcare demand and mortality in Sweden in response to non-pharmaceutical (NPIs) mitigation and suppression scenarios," *Tentatively Accepted for Publication by International Journal of Epidemiology. medRxiv*, vol. 2020, 2020.
- [11] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [12] O. Sporns, D. Chialvo, M. Kaiser, and C. Hilgetag, "Organization, development and function of complex brain networks," *Trends in Cognitive Sciences*, vol. 8, no. 9, pp. 418–425, 2004.
- [13] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.
- [14] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, 2010.
- [15] J. A. R. Marshall, P. C. Trimmer, A. I. Houston, and J. M. McNamara, "On evolutionary explanations of cognitive biases," *Trends in Ecology & Evolution*, vol. 28, no. 8, pp. 469–473, 2013.
- [16] T. W. Fawcett, B. Fallenstein, A. D. Higginson et al., "The evolution of decision rules in complex environments," *Trends in Cognitive Sciences*, vol. 18, no. 3, pp. 153–161, 2014.
- [17] S. Battiston, D. Delli Gatti, M. Gallegati, B. Greenwald, and J. E. Stiglitz, "Liaisons dangereuses: increasing connectivity, risk sharing, and systemic risk," *Journal of Economic Dynamics and Control*, vol. 36, no. 8, pp. 1121–1141, 2012.
- [18] N. Ferguson, D. Laydon, G. Nedjati Gilani et al., "Report 9: Impact of Non-pharmaceutical Interventions (NPIs) to Reduce COVID19 Mortality and Healthcare Demand," 2020.
- [19] L. Böttcher, M. Luković, J. Nagler, S. Havlin, and H. J. Herrmann, "Failure and recovery in dynamical networks," *Scientific Reports*, vol. 7, 2017.
- [20] N. Dehmamy, S. Milanlouei, and A.-L. Barabási, "A structural transition in physical networks," *Nature*, vol. 563, no. 7733, pp. 676–680, 2018.
- [21] S. Poledna, M. G. Miess, and C. H. Hommes, "Economic forecasting with an agent-based model," 2019.
- [22] X. Guardiola, R. Guimera, A. Arenas, A. Diaz-Guilera, D. Streib, and L. A. N. Amaral, "Macro-and Micro-structure of Trust Networks," 2002.
- [23] M. A. Di Muro, C. E. La Rocca, H. E. Stanley, S. Havlin, and L. A. Braunstein, "Recovery of interdependent networks," *Scientific Reports*, vol. 6, no. 1, 2016.
- [24] N. C. Wormald, "Models of Random Regular Graphs. London Mathematical Society Lecture Note Series," 1999.
- [25] A. Vespignani, "Twenty years of network science," *Nature*, vol. 558, no. 10, pp. 528–529, 2018.
- [26] E. Bonabeau, "Agent-based modeling: methods and techniques for simulating human systems," *Proceedings of the National Academy of Sciences*, vol. 99, no. 3, pp. 7280–7287, 2002.
- [27] J. M. Pacheco, A. Traulsen, and M. A. Nowak, "Coevolution of strategy and structure in complex networks with dynamical linking," *Physical Review Letters*, vol. 97, no. 25, p. 258103, 2006.
- [28] P. Erdos and A. Rényi, "On Cantor's series with convergent," *Physical Review Letters*, vol. 2, pp. 93–109, 1959.
- [29] M. O. Jackson, B. W. Rogers, and Y. Zenou, "The economic consequences of social-network structure," *Journal of Economic Literature*, vol. 55, no. 1, pp. 49–95, 2017.
- [30] S. L. Schwarcz, "Systemic Risk," 2008.
- [31] K. N. Laland, "Social learning strategies," *Animal Learning & Behavior*, vol. 32, no. 1, pp. 4–14, 2004.
- [32] M. Barthélémy and L. A. N. Amaral, "Small-world networks: evidence for a crossover picture," *Physical Review Letters*, vol. 82, no. 15, pp. 3180–3183, 1999.
- [33] D. Kahneman, "A perspective on judgment and choice: mapping bounded rationality," *American Psychologist*, vol. 58, no. 9, pp. 697–720, 2003.
- [34] J. Dermine, "Comment," *Swiss Journal of Economics and Statistics*, vol. 58, 1996.
- [35] S. Lämmer and D. Helbing, "Self-control of traffic lights and vehicle flows in urban road networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 4, 2008.
- [36] P. Gai and S. Kapadia, "Contagion in financial networks," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 466, no. 2120, pp. 2401–2423, 2010.
- [37] M. Bardoscia, F. Caccioli, J. I. Perotti, G. Vivaldo, and G. Caldarelli, "Distress propagation in complex networks: the case of non-linear DebtRank," *PLoS one*, vol. 11, no. 10, pp. 1–12, 2016.
- [38] J. Lorenz, S. Battiston, and F. Schweitzer, "Systemic risk in a unifying framework for cascading processes on networks," *The European Physical Journal B*, vol. 71, no. 4, pp. 441–460, 2009.
- [39] E. J. Kane, "Principal-agent problems in S&L salvage," *The Journal of Finance*, vol. 45, no. 3, pp. 755–764, 1990.
- [40] A. Gupta and L. Misra, "Failure and Failure Resolution in the US Thrift and Banking Industries," 1999.
- [41] G. G. Kaufman and S. A. Seelig, "Post-resolution treatment of depositors at failed banks: implications for the severity of banking crises, systemic risk, and too-big-to-fail," 2006.
- [42] Coronavirus, "Three things all governments and their science advisers must do now," *Nature*, vol. 579, 2020.
- [43] D. Helbing and A. Johansson, "Cooperation, norms, and revolutions: a unified game-theoretical approach," *PLoS One*, vol. 5, no. 10, 2010.