A Complex Network Framework for Validated Assessments of Systems of Systems Robustness

Aleksandra Markina-Khusid, Ryan B. Jacobs, Laura Antul, Lance Cho, and
Huy T. Tran, *Member, IEEE*

***Abstract*—The complexity of modern systems of systems (SoS) requires the ability to quickly and effectively evaluate robustness in architecture alternatives. This article presents a framework for rapid, quantitative comparisons of robustness in SoS architectures that leverages complex network methods for assessing robustness and design of experiments (DoE) techniques for validating their use in the scenario of interest. We consider both single-layer and multilayer network representations of SoS and focus on algebraic connectivity, inverse average path length, and largest connected component size as measures of robustness. Two case studies are used to illustrate our framework and assess its utility: a command, control, communications, computer, intelligence, surveillance, and reconnaissance (C4ISR) simulation and a multilayer message-passing network simulation. We find that most of the considered network metrics capture expected robustness trends, though their ability to capture these trends is often affected by the scenario of interest. These results demonstrate the potential value of complex network methods for lightweight analysis of robustness in SoS architecture alternatives, when appropriately supported by DoE methods for understanding their limitations.**

***Index Terms*—Complex networks, design of experiments, robustness, systems of systems, systems of systems architecture.**

I. Introduction

MODERN systems are becoming increasingly networked to form systems of systems (SoS), providing novel capabilities across many domains. More specifically, an SoS is "a system-of-interest whose system elements are themselves systems; typically these entail large scale inter-disciplinary problems with multiple, heterogeneous, distributed systems" [1]. For example, consider a team of low-cost, heterogeneous autonomous systems with various sensors and data links working together to perform a surveillance mission. Other examples include intelligent transportation systems and smart cities utilizing data from a network of distributed sensors to improve operations. These networked systems have the potential to improve performance and efficiency relative to complex, monolithic systems of the past. However, reliance on this connectivity also introduces new vulnerabilities that must be considered. We must, therefore, understand how to engineer SoS that can provide desired capabilities in the presence of system or communication failures.

Here, we specifically aim to engineer robustness into SoS, where robustness is defined as "the property of a system that allows it to satisfy a fixed set of requirements, despite changes in the environment or within the system" [2], [3]. While similar, Robustness differs from other fail-safe properties like resilience, which is more focused on adaptation following a degradation in capability [3]–[7], and flexibility, which is more focused on handling changing requirements than environmental or system changes [2]. We focus on robustness because having inherent resistance to environmental or system changes is needed for safety-critical SoS that cannot allow even temporary capability loss, due to potentially severe impacts on those served by the SoS. Considering robustness within SoS engineering (SoSE) is challenging, particularly during early-concept studies of architecture alternatives which require analyses that capture SoS complexities while maintaining tractability for tradespace exploration. Tractability is required given the combinatorial nature of architecture design, which results in an immense number of architecture alternatives when considering possible structural and behavioral configurations. We aim to address this challenge of developing lightweight, yet informative techniques for representing SoS architecture alternatives and analyzing their robustness to potential failures.

Given the connectivity of modern SoS, complex network methods offer a potential approach for assessing robustness in architecture alternatives. This approach is particularly attractive because many calculations can be derived from an adjacency matrix, offering computationally lightweight metrics suitable for large, early-concept tradespace analyses. However, recent efforts have identified potential limitations of complex network methods when applied to real engineered systems [8], [9]. Thus, an additional challenge we aim to address is the development of techniques for validating the use of complex network methods for assessing SoS robustness in a domain of interest.

This article presents a framework for 1) quickly analyzing the robustness of SoS architecture alternatives using complex network methods and 2) efficiently validating the use of those methods for a domain of interest using design of experiments (DoE) techniques. We apply this framework to two case studies. Though we focus on robustness, our complex network approach provides a general mathematical framework that can be used to

Manuscript received July 22, 2020; revised November 27, 2020; accepted February 23, 2021. *(Corresponding author: Aleksandra Markina-Khusid.)*

Aleksandra Markina-Khusid, Ryan B. Jacobs, Laura Antul, and Lance Cho are with the MITRE Corporation, McLean, VA 22102 USA (e-mail: amarkina@gmail.com; rjacobs@mitre.org; lantul@mitre.org; mkcho@mitre.org).

Huy T. Tran is with the Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, Champaign, IL 61820 USA (e-mail: huytran1@illinois.edu).

Digital Object Identifier 10.1109/JSYST.2021.3064817

assess other common measures of merit, such as survivability, vulnerability, and reliability. The outcome of our framework is an understanding of which complex network metrics are most suitable for assessing the robustness of SoS alternatives in a given domain, and can subsequently be used for tractable architecture tradespace exploration and alternative down selection. The remainder of the article is organized as follows. Section II provides a summary of related work. Section III describes our proposed framework. Section IV presents a case study based on a command, control, communications, computer, intelligence, surveillance, and reconnaissance (C4ISR) simulation. Section V presents a second case study that considers multilayer networks rather than single-layer ones through the use of a generic message-passing network simulation. Section VI provides concluding thoughts and future research directions.

II. Background

Most SoS definitions stem from Maier's seminal work describing an SoS as a set of components, which may be individually regarded as systems, such that those components are independently operated and managed [10]. SoS are typically studied within the field of SoSE, which deals with "planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into an SoS capability greater than the sum of the capabilities of the constituent parts" [11]. A common way to view SoSE is through the wave model, which depicts major steps in a time-sequenced process [12]; a central theme in the wave model is the iterative and evolutionary development of SoS architectures. This need for iterative and evolutionary development has motivated many studies on SoS tradespace analysis. For example, tradespace analysis methods have been proposed that can handle different levels of model fidelity where architecture alternatives are evaluated in terms of their utility, cost, participation risk, and other value-sustaining "-ilities" [13], [14]. Others have investigated techniques for generating feasible SoS alternative architectures [15], optimizing architectures [16], [17], approximating SoS performance with surrogate models [18], and identifying important "-ilities" based on subjective input from decision-makers [19]. While these efforts support tradespace analysis, they do not provide direct means for assessing SoS robustness within those trade studies.

Regarding efforts to assess SoS robustness, one approach is to calculate performance with and without constituent systems [20], [21]; however, it is difficult to attain accurate performance measurements for an SoS in all configurations of interest. Network theory has instead been proposed as a more computationally feasible approach, building from recent studies of robustness in the complex networks community [22]–[26]. This approach has been used to study system resilience [27], SoS complexity and its relationship to fragility [28], robustness in complex engineered systems [29], [30], and robustness in combat networks [31], [32]. However, these studies either provide limited validation of their proposed metrics or do not provide a detailed approach for performing this validation. Such validation is important, given that recent studies have identified possible limitations of using network metrics to understand behaviors in real engineered systems [8], [9]. Furthermore, most systems engineering applications of complex network methods have focused on single-layer networks, with no consideration of multilayer representations. Many real-world SoS may be better modeled as having multiple layers that, for example, perform different functions or communicate using different types of data links. Single-layer models would only be able to analyze the impacts of disruptions within individual layers, potentially leading to misleading results that do not fully account for complexities related to multiple network layers.

Building on these efforts, we propose a cohesive framework for assessing the robustness of SoS architecture alternatives, which leverages single-layer and multilayer complex network methods, while also including the validation of those methods for the domain of interest. Our main contributions are as follows: 1) the consideration of single-layer and multilayer SoS architectures; 2) the inclusion of a process for validating the proposed complex network methods; and 3) the application of our framework to two SoS case studies. We hope to help bridge the gap between theoretical studies and real-world practitioners by improving our understanding of how well complex network methods translate to engineered systems, particularly large-scale SoS with multiple connectivity layers.

III. Methods

Our framework is composed of following three steps: 1) we model the SoS of interest as a single-layer or multilayer network. 2) we implement a set of complex network metrics for lightweight measurements of robustness in alternative architectures for this SoS. 3) we use DoE techniques to efficiently validate and assess the utility of these metrics for the domain of interest. The remainder of this section describes these steps in detail.

A. *Complex Network Models*

We model SoS architectures as single-layer or multilayer networks with undirected and unweighted edges. Beginning with the single-layer case, we model SoS such that nodes represent constituent systems and edges represent relationships among those systems (e.g., communications, contractual agreements). An example architecture network is shown in the Department of Defense Architecture Framework (DoDAF) operational view 1 (OV-1) of the Naval Integrated Fire Control-Counter Air (NIFC-CA) architecture [11]. This multidomain SoS is composed of a variety of assets, including sea-based destroyers, ground-based operational centers, and air-based platforms, many of which are connected by various data links.

We represent these networks using their adjacency, degree, and Laplacian matrices, as the proposed robustness metrics can be calculated from these matrices. For an undirected, unweighted network, the adjacency matrix $\boldsymbol{A}$ is a symmetric matrix with elements of zero or one that specify the network's connectivity, defined as

$$\boldsymbol{A}_{ij} = \begin{cases} 1 & \forall (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

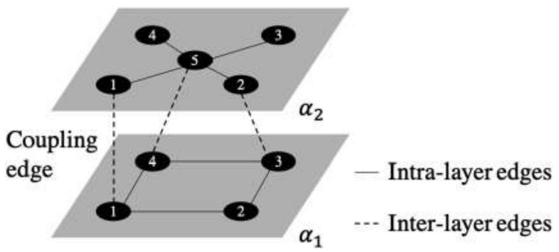


Fig. 1. Notional multilayer network with two layers, five nodes, eight intralayer edges, and three interlayer edges (one of which is a coupling edge).

where (i, j) is the edge connecting nodes i and j and E is the set of all edges in the network. The degree matrix $\boldsymbol{D}$ is a diagonal matrix containing the degrees of all nodes, defined as

$$\boldsymbol{D}_{ij} = \begin{cases} d_i & i = j \\ 0 & \text{otherwise} \end{cases}$$

where d_i is the degree (i.e., number of incident edges) of node i. The Laplacian $\boldsymbol{L}$ is defined as the difference between the degree and adjacency matrices of a network, such that $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$.

We model multilayer SoS architectures as multilayer networks, an approach that has provided notable insights into the robustness of many real-world networks [33]–[35]. Based on the notation described in [36], we define a multilayer network as a four-element tuple $M = (V_M, E_M, V, L)$. We define V as the set of all nodes in the network, such that a given node $i \in V$ can exist in multiple network layers. We define $L = \{\alpha_k\}_{k=1}^{b}$ as the set of layers considered, where α_k is the kth layer and $b = |L|$ is the number of layers. Note that we use L rather than the sequence $\{L_a\}_{a=1}^{d}$ because we choose to consider only one aspect (or network dimension) such that $d = 1$, with the aspect being the type of edges contained by a layer. Thus, we use L to represent the first set L_1 in the sequence $\{L_1\}$. Including other aspects (i.e., having $d > 1$) could allow higher dimensional analyses, for example, by including temporal changes to a network, in which the second aspect could define the time at which edges are present, with the first aspect defining the type of edges present. We define $V_M \subseteq V \times L$ as the set of *node-layer tuples*, where a node-layer tuple (i, α_1) indicates that node i exists within layer α_1. We define $E_M \subseteq V_M \times V_M$ as the set of all edges in the network, including intralayer edges and inter-layer edges. The set of all intralayer edges E_A is defined as $E_A = \{((i, \alpha_k), (j, \alpha_l))) \in E_M | k = l\}$; i.e., E_A is the set of all edges connecting two nodes within the same layer. The set of all interlayer edges E_C is defined as $E_C = E_M \setminus E_A$; i.e., E_C is the set of all edges connecting two nodes in different layers. The set of all coupling edges $E_{\tilde{C}} \subseteq E_C$ is defined as $E_{\tilde{C}} = \{((i, \alpha_k), (j, \alpha_l))) \in E_C | i = j \text{ and } k \neq l\}$; i.e., $E_{\tilde{C}}$ is the set of all intralayer edges connecting the same nodes in different layers. We then define an intralayer graph $\mathcal{G}_A = (V_M, E_A)$, an interlayer graph $\mathcal{G}_C = (V_M, E_C)$, a coupling graph $\mathcal{G}_{\tilde{C}} = (V_M, E_{\tilde{C}})$, and the overall graph $\mathcal{G}_M = (V_M, E_M)$ yielded by the multilayer network. See Fig. 1 for a notional view of a multilayer network.

We analyze a multilayer network using its supra-adjacency matrix $\boldsymbol{A}_M$, where $\boldsymbol{A}_M$ is the adjacency matrix of $\mathcal{G}_M$. That is, a supra-adjacency matrix is an adjacency matrix representing a flattened version of a multilayer network. Analogous to the single-layer network case, we can also analyze the supra-Laplacian $\boldsymbol{L}_M$ as $\boldsymbol{L}_M = \boldsymbol{D}_M - \boldsymbol{A}_M$, where $\boldsymbol{D}_M$ is the diagonal supra-matrix. Single-layer network metrics, such as those described in Section III-B, can then be applied to their counterpart supra matrices in the case of multilayer networks.

B. Network Metrics

We focus on three network metrics for assessing the robustness of an SoS architecture: algebraic connectivity, inverse average path length, and largest connected component size. We choose this set of metrics because it spans a variety of approaches to assessing robustness; more specifically, it considers a spectral approach (i.e., algebraic connectivity), an approach based on path lengths (i.e., inverse average path length), and an approach based on cluster size (i.e., largest connected component size). Furthermore, these metrics have provided valuable insights into the robustness of complex networks [22], [37]; we hypothesize that they also offer useful and scalable assessments of robustness in SoS architectures. The metrics are calculated from adjacency and Laplacian matrices for single-layer networks. We extend them for use with multilayer networks by calculating them from corresponding supra-adjacency and supra-Laplacian matrices, such that they preserve their single-layer meaning.

We use algebraic connectivity to measure the robustness of a network based on its spectral properties [38]. Algebraic connectivity is calculated as the second smallest eigenvalue of the Laplacian matrix and represents the average difficulty of isolating a node within a connected network. A network is connected if there exists a path between every pair of nodes in the network. A disconnected network has an algebraic connectivity of zero, while a connected one has a positive algebraic connectivity. Thus, a network with high algebraic connectivity indicates it is robust, since it is unlikely to lose connectivity with a node following node or edge removals. Algebraic connectivity inherently measures the robustness of a network through spectral analysis of its Laplacian matrix, without the need for node or edge removals; the other network metrics considered in this article require modeling of node or edge removals to measure robustness.

We use inverse average path length to measure the robustness of a network based on its ability to sustain short path lengths within itself following node removals [23]. The average path length of a network is calculated as the average of the geodesic distances between all node pairs, where the geodesic distance $d_{i,j}$ is the length of the shortest path between nodes i and j. We use the inverse of the average path length to handle infinite path lengths associated with disconnected node pairs. More formally, we calculate the inverse average path length, $\langle d \rangle'_{\mathcal{G}}$, of a given graph $\mathcal{G}$ as

$$\langle d \rangle'_{\mathcal{G}} = \frac{1}{N(N-1)} \sum_{i,j \in V, i \neq j} \frac{1}{d_{i,j}} \tag{1}$$

where $N = |\mathcal{G}|$ is the number of nodes in $\mathcal{G}$ and V is the set of nodes in $\mathcal{G}$. We then use inverse average path length as a measure of robustness by calculating the delta of a network's inverse average path length following node removals or the mean of inverse average path length values over multiple node removal events, similar to [37]. We calculate the delta inverse average path length, $\Delta\langle d\rangle'$, as

$$\Delta\langle d\rangle' = \langle d\rangle'_{\mathcal{G}_0} - \langle d\rangle'_{\mathcal{G}_1} \tag{2}$$

where $\langle d\rangle'_{\mathcal{G}_0}$ is the inverse average path length of a network $\mathcal{G}_0$ with no nodes removed and $\langle d\rangle'_{\mathcal{G}_1}$ is the inverse average path length of the same network with nodes removed, $\mathcal{G}_1$. We calculate the mean of inverse average path lengths, $\overline{\langle d\rangle'}$, as

$$\overline{\langle d\rangle'} = \frac{1}{N_R}\sum_{i=1}^{N_R}\langle d\rangle'_{\mathcal{G}_i} \tag{3}$$

where N_R is the number of node removal events and $\langle d\rangle'_{\mathcal{G}_i}$ is the inverse average path length of the network $\mathcal{G}_i$ remaining after node removal event i. More robustness is associated with lower values of $\Delta\langle d\rangle'$ but higher values of $\overline{\langle d\rangle'}$, as these values indicate that a network is better able to sustain short path lengths following node removals. We use $\Delta\langle d\rangle'$ as a measure of robustness for case study 1 and $\overline{\langle d\rangle'}$ as a measure of robustness for case study 2.

We use the size of the largest connected component to measure the robustness of a network based on its ability to sustain connectivity following node removals [22]. We define $S_{\mathcal{G}}$ as the size of the largest connected component of network $\mathcal{G}$; we refer to $S_{\mathcal{G}}$ as a network's component size for simplicity. We use component size as a measure of robustness in a similar manner to that for inverse average path length; that is, we calculate the delta of a network's component size following node removals or the mean of component size values over multiple node removal events. We calculate the delta component size, ΔS, as

$$\Delta S = S_{\mathcal{G}_0} - S_{\mathcal{G}_1} \tag{4}$$

where $S_{\mathcal{G}_0}$ is the component size of a network $\mathcal{G}_0$ with no nodes removed and $S_{\mathcal{G}_1}$ is the component size of the same network with nodes removed, $\mathcal{G}_1$. We calculate the mean of component sizes, $\overline{S}$, as

$$\overline{S} = \frac{1}{N_R}\sum_{i=1}^{N_R} S_{\mathcal{G}_i} \tag{5}$$

where N_R is again the number of node removal events and $S_{\mathcal{G}_i}$ is the component size of the network $\mathcal{G}_i$ remaining after node removal event i. Similar to inverse average path length, more robustness is associated with lower values of ΔS but higher values of $\overline{S}$, as these values indicate a network is better able to sustain overall connectivity following node removals. We use ΔS as a measure of robustness for case study 1 and $\overline{S}$ as a measure of robustness for case study 2.

We consider two node removal methods for use with the inverse average path length and component size metrics. The first is a random removal where random nodes are selected for each node removal event [22]. Two examples of random removals are random failures in information technology support systems or power systems that result in the inability of an SoS to operate as intended. We also consider targeted node removals, where nodes are removed sequentially in order of their initial degree (i.e., number of neighbors) [22]. An example of targeted removals is an intentional attack on an SoS, as might occur in a warfighting scenario. We focus on these removal methods because they span a range of potential threat types, though our proposed methods can be used with any desired threat models. Table I summarizes our proposed network metrics.

C. Validation Through Design of Experiments

For the network metrics proposed in Section III-B to be useful for comparing SoS robustness, ranking of architecture alternatives using those metrics must be similar to rankings based on some presumed truth values for robustness. For this article, we use simulated measures of robustness to determine our presumed truth rankings. We calculate Spearman's rank correlation coefficients between each network metric and simulated robustness values associated with a set of architecture alternatives to assess the strength and direction of the relationship between these metrics. An ideal value of +1 would be achieved when the ranks of the architectures match perfectly for a network metric and simulated robustness, a value of 0 would indicate no monotonic relationship between the ranks, and a value of -1 would indicate an opposite ranking.

We then apply DoE techniques, specifically implementing a factorial design and analyzing main effects and interactions, to understand how the observed correlations between a given network metric and simulated robustness change as simulation parameters are varied [39]. Understanding the sensitivity of network metric correlations to these parameters is necessary because complex SoS simulations typically require specification of various input parameters defining the modeled scenario. We use DoE techniques because they allow us to efficiently determine the utility of network metrics by running a much smaller set of simulations than one would if fully evaluating the set of alternatives or using more complex sensitivity analysis methods [40]. We use a 2^3 factorial design for both case studies presented in this article, though more sophisticated experimental designs can be utilized as needed (e.g., Latin hypercubes [41] or sequential bifurcation [40], [42] in studies with a larger set of simulation parameters of interest). We treat the simulation parameters of interest as experiment factors and correlations for considered network metrics as responses. Due to the stochastic nature of many SoS simulations, we also include experiment replicates as needed.

We then analyze completed experiments by first examining distributions of metric correlations across all design points. Consistently high and tightly distributed correlation values provide an initial indication that the network metric in question is a good proxy for simulated robustness. We calculate main effects of factors on correlations as indicated by regression coefficients and use N-way ANOVA values to show the significance of those effects. We also examine interaction effects of factor pairs on correlations. These steps allow us to determine which, if any, of the proposed network metrics are useful proxies for simulated

TABLE I
SUMMARY OF NETWORK METRICS

Metric	Case studies used for	Description
Algebraic connectivity	1, 2	Robustness to node isolation
Delta inverse average path length, $\Delta\langle d\rangle'$	1	Robustness to losing short path lengths
Mean inverse average path length, $\overline{\langle d\rangle'}$	2	Robustness to losing short path lengths
Delta component size, ΔS	1	Robustness to losing overall connectivity
Mean component size, $\overline{S}$	2	Robustness to losing overall connectivity

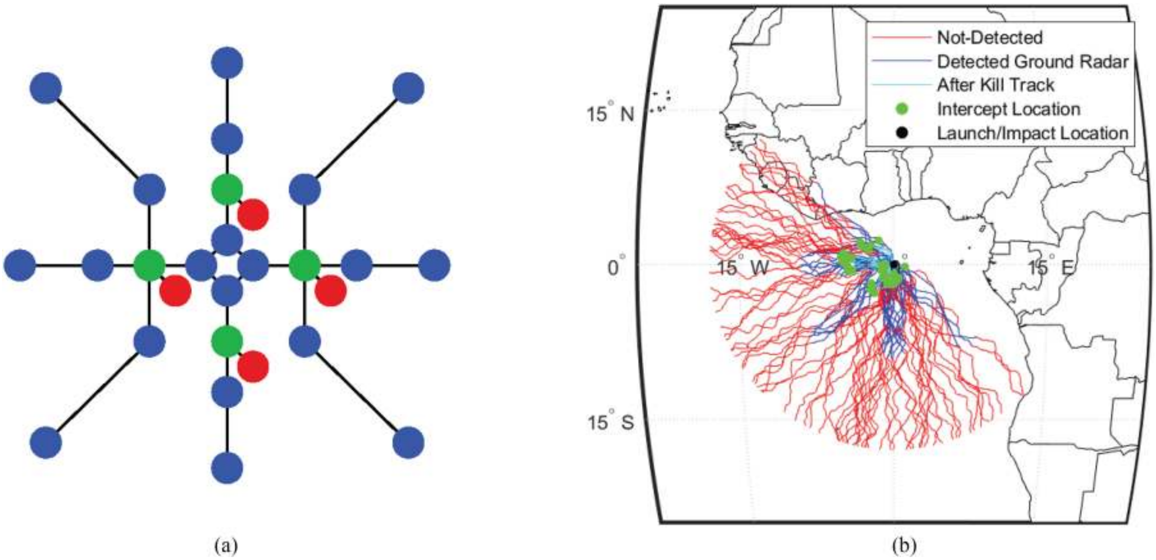


Fig. 2. (a) Network representation of the baseline architecture, with radars shown in blue, air operation centers shown in green, and fighters shown in red. Node are positioned such that their *x*- and *y*-coordinates match their latitude and longitude values. (b) Screenshot of the C4ISR simulation showing various threat trajectories in red with radar detections along those trajectories shown in blue.

robustness in a given study, and at what parameter values their utility holds or breaks down.

IV. Case Study 1: C4ISR Simulation

Our first case study builds on previous work in [43] and applies the proposed methods to single-layer architectures modeled using a C4ISR simulation. The C4ISR simulation offers an end-to-end analysis framework for mission assessments, including applications such as Integrated Air and Missile Defense. The simulation enables users to model aircraft, ballistic missile, and cruise missile threats against various defensive architectures and evaluate respective interceptor (e.g., aircraft or launcher) platform performance. Architectures are defined through the placement and constitution of a defensive laydown. For this case study, we consider architectures composed of three types of defensive systems: radars, air operations centers, and fighters. Radars are used to detect incoming threats, air operations centers are used to decide how to respond to detected threats, and fighters are used to engage threats. We simulate aircraft threat scenarios against a given architecture through the following steps. First, the probability of detection along the length of a threat trajectory by a radar system is calculated using embedded radar modeling tools, and used to simulate whether or not an individual threat is detected. Detections are then communicated to an air operations center, which in turn communicates engagement instructions to a fighter. The probability of kill for each threat trajectory versus a single defense platform is then calculated individually. Calculated probability of kill values are then statistically combined over all defense platforms to obtain an overall probability of engagement success (PES), assuming engagement independence. Each scenario is simulated with 80 different threat trajectories, where each threat trajectory can come from the north, east, south, or west quadrants of the target location. The detailed asset models and Monte–Carlo sampling used by this simulation result in high computational costs that prevent its use in early-concept tradespace analyses of architecture robustness, motivating the use of complex network methods instead.

A. Experimental Design

We apply our proposed framework to this case study as follows. We model C4ISR architectures as single-layer networks, where nodes represent defensive systems and edges represent communication links among those systems. We define a baseline architecture composed of 28 systems connected by 28 edges with the laydown shown in Fig. 2; this baseline is selected in a manner that provides dispersed coverage of the area of interest. We then generate architecture alternatives that are expected to have differing levels of robustness to the defined threat scenario,

TABLE II
C4ISR Experimental Design

Factors	Levels	Description
removalType	Targeted, Random	Type of node removal
radarType	Low, High	Power of modeled radar systems (i.e., sensing radius)
threatDistr	360, 180	Angular range from which threat trajectories can originate from

to understand how well our proposed network metrics capture simulated robustness trends. We generate these alternatives by adding edges to the baseline architecture. Each edge is added such that the head of the edge is the node that currently has the lowest degree and the tail is the node that is geographically closest to the head. We consider architecture alternatives generated with $0, 10, \ldots, 90$ percent of the baseline number of edges added to the baseline. This architecture generation model is stochastic in that ties between possible edges added are randomly broken.

We then calculate algebraic connectivity, delta inverse average path length, and delta component size metrics for each generated architecture and compare these values to a simulated measure of robustness. We measure the simulated robustness of an architecture by assessing its ability to sustain a high PES following node removals. We refer to the simulated robustness metric as delta PES, calculated as

$$\Delta\text{PES} = \text{PES}_{\mathcal{G}_0} - \text{PES}_{\mathcal{G}_1} \qquad (6)$$

where $\text{PES}_{\mathcal{G}_0}$ is the simulated PES of an architecture $\mathcal{G}_0$ with no nodes removed and $\text{PES}_{\mathcal{G}_1}$ is the simulated PES of the same architecture with nodes removed, $\mathcal{G}_1$. We use the random and targeted removal methods described in Section III-B to remove nodes for all metric calculations. Removals are modeled as a single removal event that removes 25% of the nodes in the original architecture (i.e., seven nodes), with two of those nodes being air operations centers and the remaining five being radar systems. We specify this distribution of removed systems in an effort to produce impactful removals.

Finally, we implement a 2^3 factorial experiment to understand how the correlation between network and simulated robustness metrics varies with changes to simulation factors. The three factors in this experiment are removal type, radar type, and threat distribution. Note that a 360° threat distribution includes threats from the north, east, south, and west directions, while a 180° range only includes threats from the south and west directions. The responses in this experiment are the Spearman's rank correlations between simulated robustness (i.e., delta PES) and algebraic connectivity, delta inverse average path length, and delta component size. Each experimental design point is evaluated by generating ten architecture alternatives (using the generative algorithm described above with $0, 10, \ldots, 90$ percent of the baseline number of edges added to the baseline) and calculating Spearman's rank correlation between each network robustness metric and the simulated robustness metric. The experimental design is summarized in Table II. We replicate the experiment 20 times to account for its stochastic nature.

B. Results

Fig. 3(a) shows distributions for Spearman's rank correlation between each of our proposed network metrics and simulated robustness, over all experimental design points and replicates (i.e., over 160 data points). The results show delta component size having the strongest correlation with simulated robustness with respect to the median and interquartile ranges; correlations for algebraic connectivity and delta inverse average path length are similar to each other and lower than those for component size. Correlations for delta component size are likely higher than those for algebraic connectivity because algebraic connectivity does not model robustness to a specific set of node removals; rather, it models the difficulty of disconnecting a network regardless of the removal method. Correlations for delta component size are likely higher than those for delta inverse average path length because the C4ISR simulation does not include high-fidelity modeling of communication systems; for example, the simulation does not explicitly model communication delays or failure rates and instead focuses more on modeling radar systems. The simplicity of the modeled communications thus makes the existence of connectivity among constituent systems sufficient for successful engagements, with no significant benefits provided by, for example, short communication paths. These characteristics likely lead to delta component size more closely capturing simulated robustness than delta inverse average path length, as component sizes measure path existence with no consideration of path lengths. We also see higher variability in correlations for algebraic connectivity relative to delta inverse average path length and delta component size; this trend is again explained by the lack of explicitly specifying node removals when evaluating robustness via algebraic connectivity, which may lead to more variance in how well the metric captures simulated robustness trends given the inclusion of removal type as a factor in the experimental design. These distributions suggest that spectral, path length, and cluster-based network metrics all have value toward measuring the robustness of sensor focused architectures, though there exist slight differences in correlation trends. We also explore the main effects of our three experiment factors on these correlation trends. Fig. 3(b) shows the main effect of each factor with respect to each network metric's correlation with simulated robustness. Removal type shows the largest and most statistically significant effect on correlations for all three network metrics, with targeted removals providing stronger correlations than random ones. Threat distribution shows the second largest effect, with widely distributed threats providing stronger correlations than focused ones. Radar type shows the smallest effect, with more powerful radar systems providing stronger correlations than weaker ones. One explanation for removal type having the largest effect is that targeted removals may have similar impacts among all metrics but random removals may introduce too much stochasticity for meaningful trends to be captured, thus leading to poor correlations. We also see that while algebraic connectivity generally shows lower correlations than delta component size [as shown in Fig. 3(a)], those correlations show lower sensitivity to the radar type and threat distribution factors. Given that radar type and threat distribution are specific

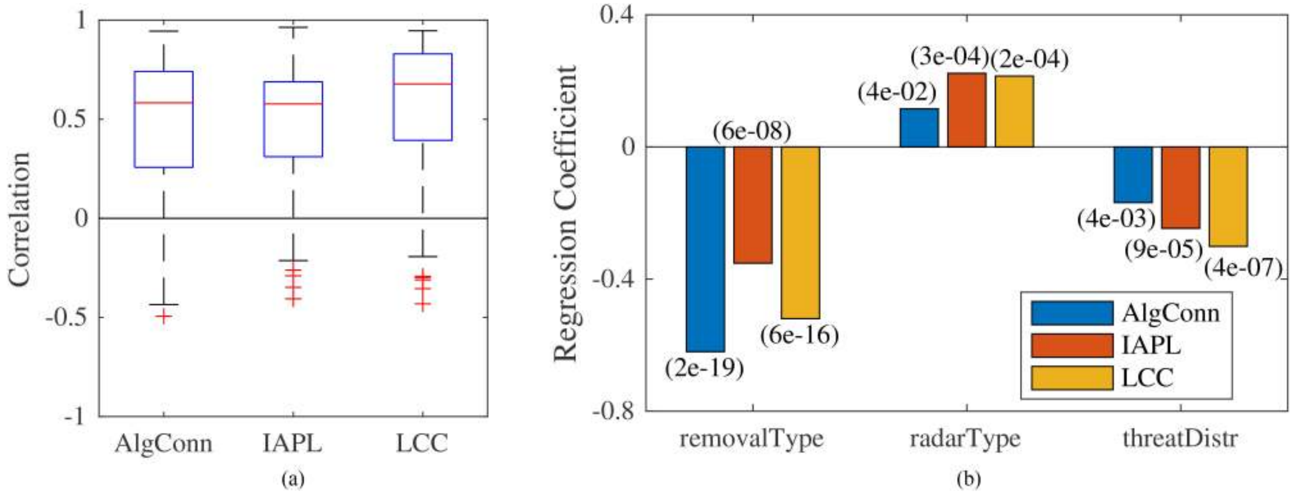


Fig. 3. (a) Box plots of Spearman's rank correlation between simulated robustness and algebraic connectivity (AlgConn), delta inverse average path length (IAPL), and delta component size (LCC) over all experimental design points and replicates from case study 1. Red lines denote the median, while the boxes denote the interquartile range (i.e., the 25th and 75th percentiles). The whiskers extend to either the most extreme observation or to a distance of 1.5 times the interquartile range. The + markers denote data outside of this range. (b) Main effects of factors on Spearman's rank correlation between each of the network metrics and simulated robustness, shown as the magnitude of linear regression coefficients. Corresponding p-values for those regression coefficients are shown with each bar.

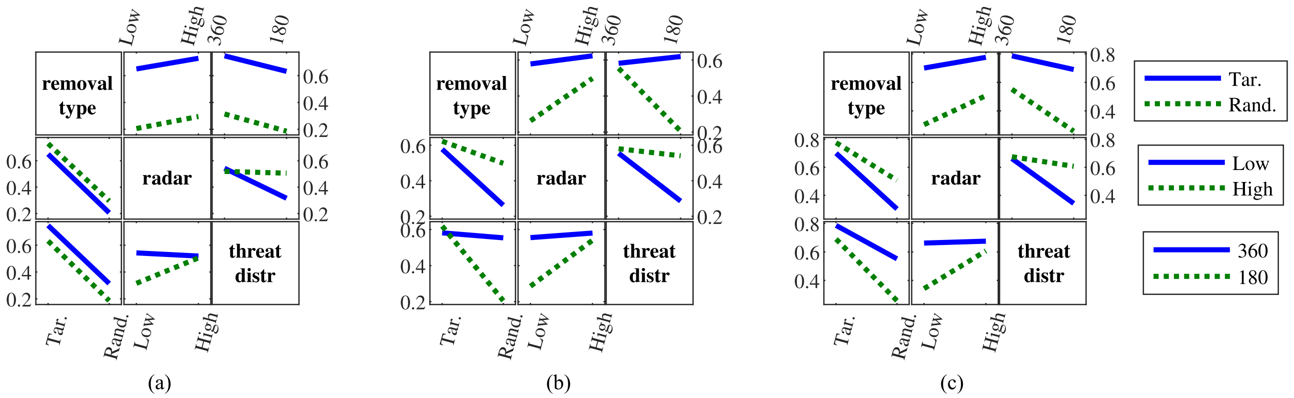


Fig. 4. Interaction plots for Spearman's rank correlation between simulated robustness and (a) algebraic connectivity, (b) delta inverse average path length, and (c) delta component size for the experiment factors in case study 1.

to the simulation, this result suggests algebraic connectivity may provide more consistent correlations with respect to changes in simulation specific factors. These results demonstrate the importance of understanding which simulation factors have the strongest impact on the utility of robustness metrics, as that utility is highly sensitive to some factors and relatively insensitive to others. That is, practitioners should be aware of which conditions their scenario of interest satisfies before applying these metrics in tradespace analyses.

Fig. 4 shows the corresponding interaction effects for each factor with respect to each network metric's correlation with simulation robustness. We see little interaction effects for removal type, suggesting that the effects of radar type and threat distribution on metric correlations do not depend on the removal type modeled. However, we do see interaction effects for radar type and threat distribution, as the effect of each of these two factors strongly depends on the level of the other. These results further demonstrate the importance of understanding how factors of interest may influence the utility of various network metrics.

V. Case Study 2: Message-Passing Multilayer Network Simulation

Our second case study applies the proposed methods to multilayer architectures modeled using a message-passing network simulation. This simulation models general networked systems relying on information exchange among nodes to perform desired missions, expanding on [44] and [45] to consider multilayer networks. Simulations begin with all nodes in a network being active. At each time step t, each active node has a probability μ of generating a new message; and the target node of that message is randomly selected among the set of active nodes. Messages travel through the network along the shortest path from source to target, one hop at a time. Messages fail with a probability of ρ at each time step. Shortest paths are recalculated at each time step to only include currently active nodes. The objective of the network is to successfully pass messages from their source to target node. We run each simulation for 500 time steps with 10 nodes simultaneously removed every 100 time steps for this article.

A. Experimental Design

We apply our proposed framework to this case study as follows. We model message-passing architectures as two-layer networks, where nodes represent systems that generate and receive messages. Each node within a given layer has a corresponding node in the other layer; i.e., we model architectures as multiplex networks [36]. Note that this case study could be extended to general multilayer networks with little modification. We model edges as communication links among systems, where intralayer edges represent communication links of the same type within a given layer and interlayer edges represent communication links between layers. The existence of an interlayer edge represents an interlayer connection between nodes that allows messages to be encoded/decoded as needed to pass through layers. We consider architectures with 100 nodes in each layer (i.e., 200 nodes in total). We focus on synthetically generated random multilayer architectures, having network topologies that range from being Erdős–Rényi to Barabási–Albert (BA) scale-free [22]. We generate these architectures by first generating each network layer independently using the single-layer network generation algorithm proposed in [45]. We use the α_{topology} parameter of that algorithm to define how intralayer edges are added, with each edge being added in a random manner with a probability of α_{topology} and added using preferential attachment with a probability of $1 - \alpha_{\text{topology}}$. We then add interlayer edges between nodes with a probability of $\alpha_{\text{intralayer}}$ based on the algorithm proposed in [35]. We generate architecture alternatives for this case study by using this multilayer generative algorithm and increasing α_{topology} from $0, 0.1, \ldots, 1$. We set $\alpha_{\text{intralayer}} = 0.25$ to provide enough interlayer connectivity to define functional architectures, but not so much that node removals have no impact.

We then calculate algebraic connectivity, mean inverse average path length, and mean component size metrics for each generated architecture and compare these values to a simulated measure of robustness. We measure the simulated robustness of an architecture by assessing how well it sustains the ability to successfully pass messages from their source to target node following node removals. That is, we calculated simulated robustness as the total number of messages successfully received over all times steps of a simulation. We refer to the simulated robustness as the area under the curve (AUC) of $y(t)$, where $y(t)$ is the total number of messages received by target nodes at time t, calculated as

$$y(t) = \sum_{i=1}^{N_t} |\mathcal{M}_i(t)| \tag{7}$$

where N_t is the number of active nodes at time t and $\mathcal{M}_i(t)$ is the set of messages received by target node i at time t. We use $y(t)$ to calculate the AUC as

TABLE III
MESSAGE-PASSING NETWORK EXPERIMENTAL DESIGN

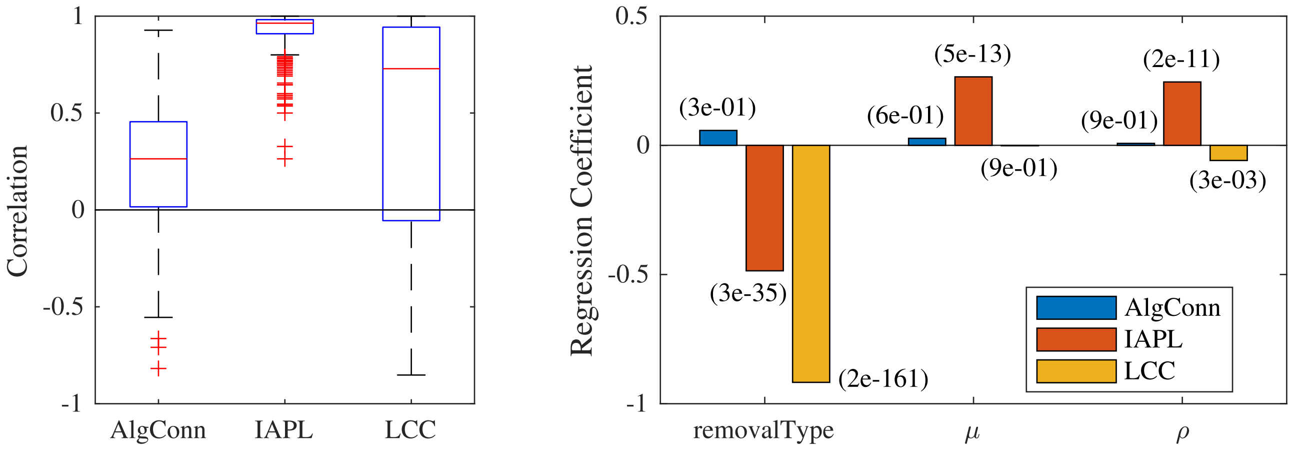
Factors	Levels	Description
removalType	targeted, random	Type of node removal
μ	0.1, 0.3	Probability of a node generating a new message
ρ	0.1, 0.3	Probability of a message randomly failing

$$\text{AUC} = \sum_{t=1}^{500} y(t). \tag{8}$$

As with case study 1, we then implement a 2^3 factorial experiment to understand how the correlation between network and simulated robustness metrics varies with changes to simulation factors. The three factors in this experiment are removal type, the message generation rate (μ), and the message failure rate (ρ). The message generation rate defines the workload of the architecture, while the message failure rate defines its reliability. The responses in this experiment are the Spearman's rank correlations between simulated robustness (i.e., AUC) and algebraic connectivity, mean inverse average path length, and mean component size. Each experimental design point is then evaluated by generating eleven architecture alternatives (using the generative algorithm described above with $\alpha_{\text{topology}} = 0, 0.1, \ldots, 1$) and calculating Spearman's rank correlation between each network robustness metric and the simulated robustness metric. The experimental design is summarized in Table III. We replicate the experiment 50 times to account for its stochastic nature.

B. Results

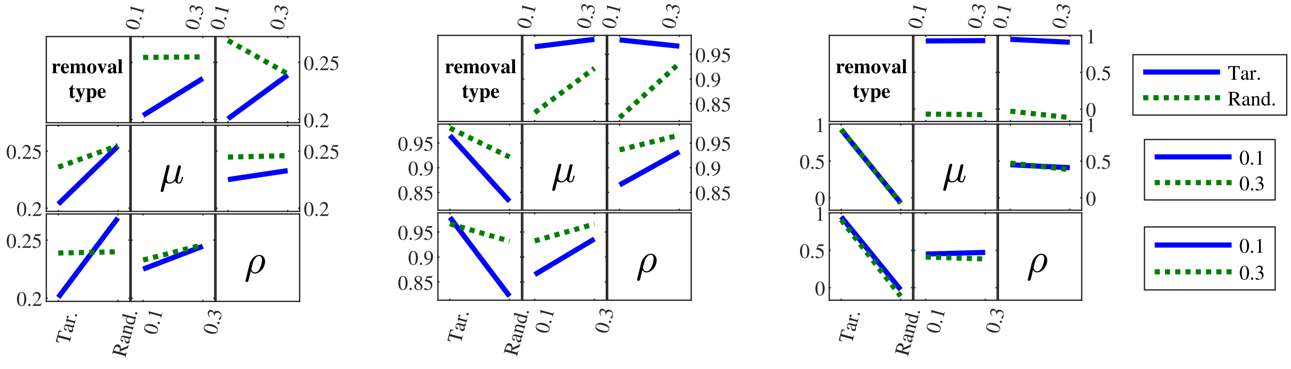
Fig. 5(a) shows the distributions for Spearman's Rank correlation between each of our proposed network metrics and simulated robustness, over all experimental design points and replicates (i.e., over 400 data points). The results show that mean inverse average path length has the strongest correlation with simulated robustness with respect to the median, followed by mean component size, and then algebraic connectivity. Mean inverse average path length also shows the smallest interquartile ranges, followed by algebraic connectivity, then mean component size. Correlations for mean inverse average path length are significantly higher than those for algebraic connectivity and mean component size because of the inclusion of a message failure rate in this simulation; a non-zero message failure rate results in path lengths playing a critical role in the probability of a message successfully reaching its target node, thus leading to high correlations between mean inverse average path length and simulated robustness. Conversely, algebraic connectivity and mean component size do not account for path lengths. Furthermore, algebraic connectivity does not model specific node removals (as discussed in Section IV-B), explaining its low correlations for this case study. These distributions suggest that path length-based robustness metrics are best suited for measuring the robustness of architectures highly dependent on timely



(a)

(b)

Fig. 5. (a) Box plots of Spearman's rank correlation between simulated robustness and algebraic connectivity (AlgConn), mean inverse average path length (IAPL), and mean component size (LCC) over all experimental design points and replicates from case study 2. The red line within each box marks the median; the bottom and top of the boxes mark the 25th and 75th percentiles, respectively. The whiskers extend to either the most extreme observation or to a distance of 1.5 times the interquartile range; the red + markers beyond the whiskers fall outside of this range. (b) Main effects of factors on Spearman's rank correlation between each of the network metrics and simulated robustness, shown as the magnitude of linear regression coefficients. Corresponding p-values for those regression coefficients are shown with each bar.



(a)

(b)

(c)

Fig. 6. Interaction plots for Spearman's rank correlation between simulated robustness and (a) algebraic connectivity, (b) mean inverse average path length, and (c) mean component size for the experiment factors in case study 2.

and reliable message passing, with cluster-based metrics also providing some value. Spectral-based measures show limited value for such scenarios.

We also explore the main effects of our three experiment factors on these correlation trends. Fig. 5(b) shows the main effect of each factor with respect to each network metric's correlation with simulated robustness. Similar to case study 1, removal type shows the largest and most statistically significant effect on correlations for mean inverse average path length and mean component size; however, its effect on algebraic connectivity is not significant at a 0.05% level, likely due to the consistently low correlations of algebraic connectivity regardless of factor levels. Message generation rate and failure rate show smaller, but still statistically significant, main effects on correlations for inverse average path length. These factors show small effects on the correlations of algebraic connectivity and mean component size though. As with case study 1, these main effect results demonstrate the importance of understanding the potential impacts of simulation factors on the utility of robustness metrics.

Fig. 6 shows the corresponding interaction effects for each factor. Here, we see notable interactions for each factor when assessing correlations with algebraic connectivity. However, interaction effects are smaller for correlations with mean inverse average path length and mean component size. These results suggest that further studies of mean inverse average path length and mean component size may not need to consider interaction effects among these factors.

VI. Conclusion

Analyzing the robustness of SoS architecture alternatives is challenging due to the need for lightweight analysis methods that can support early-concept tradeoff studies while also capturing complexities of modern SoS. We present a framework for analyzing SoS robustness that uses complex network methods able to capture single-layer and multilayer architectures, supported by a DoE approach for validating the use of those methods for a scenario of interest. We apply this framework to two case studies: a single-layer C4ISR simulation and a multilayer message-passing network simulation. Our results from the C4ISR case study suggest that spectral, path length, and cluster-based metrics can provide value toward assessing the robustness of alternative defensive architectures that are not limited by network speed or reliability, as all three metrics show relatively high correlations within our experimental design. In comparison, our results from the multilayer message-passing network case study suggest that path length-based metrics are best suited for scenarios requiring timely and reliable message passing, as our path length metric shows consistently high correlations. These results demonstrate that complex network methods can be useful when comparing the robustness of architecture alternatives, but they must be used within a validative framework that quantitatively evaluates their utility and how it may change as simulation parameters are varied.

Future directions for this article include more formalized integration with existing SoSE tools and processes, including model-based engineering methods. Furthermore, the work presented in this article abstracts SoS architectures to be undirected, unweighted networks. Extending the proposed framework to consider directed and weighted networks may further extend its utility toward evaluating SoS robustness to a malicious attack or propagating failures.

Acknowledgment

The authors would like to thank Judith Dahmann at MITRE for offering her invaluable SoSE expertise that was critical to this article.

References

[1] D. Walden, G. Roedler, K. Forsberg, R. Hamelin, and T. Shortell, *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 4th Ed. Wiley, 2015.

[2] J. H. Saleh, G. Mark, and N. C. Jordan, “Flexibility: A multi-disciplinary literature review and a research agenda for designing flexible engineering systems,” *J. Eng. Des.*, vol. 20, no. 3, pp. 307–323, 2009.

[3] P. Uday and K. Marais, “Designing resilient systems-of-systems: A survey of metrics, methods, and challenges,” *Syst. Eng.*, vol. 18, no. 5, pp. 491–510, 2015.

[4] A. W. Righi, T. A. Saurin, and P. Wachs, “A systematic literature review of resilience engineering: Research areas and a research agenda proposal,” *Rel. Eng. System Saf.*, vol. 141, pp. 142–152, 2015.

[5] D. D. Woods, “Four concepts for resilience and the implications for the future of resilience engineering,” *Rel. Eng. Syst. Saf.*, vol. 141, pp. 5–9, 2015.

[6] S. Hosseini, K. Barker, and J. E. Ramirez-Marquez, “A review of definitions and measures of system resilience,” *Rel. Eng. Syst. Saf.*, vol. 145, pp. 47–61, 2016.

[7] N. Yodo and P. Wang, “Engineering resilience quantification and system design implications: A literature survey,” *J. Mech. Des.*, vol. 138, no. 11, pp. 111 408–1-13, 2016.

[8] D. L. Alderson, “Catching the “network science” bug: Insight and opportunity for the operations researcher,” *Operations Res.*, vol. 56, no. 5, pp. 1047–1065, Oct. 2008.

[9] P. Hines, E. Cotilla-Sanchez, and S. Blumsack, “Do topological models provide good information about electricity infrastructure vulnerability?” *Chaos*, vol. 20, no. 3, 2010, Art. no. 033122.

[10] M. W. Maier, “Architecting principles for systems-of-systems,” *Syst. Eng.*, vol. 1, no. 4, pp. 267–284, 1998.

[11] O. o. t. D. U. S. o. D. f. A. Engineering, T. Systems, and Software, *Systems Engineering Guide for Systems of Systems, Version 1.0*, 1st ed. Washington, DC: ODUSD (A&T) SSE, 2008.

[12] J. Dahmann, G. Rebovich, J. Lane, R. Lowry, and K. Baldwin, “An implementers’ view of systems engineering for systems of systems,” in *Proc. IEEE Int. Syst. Conf.*, Apr. 2011, pp. 212–217.

[13] N. Ricci, M. E. Fitzgerald, A. M. Ross, and D. H. Rhodes, “Architecting systems of systems with ilities: An overview of the SAI method,” *Procedia Comput. Sci.*, vol. 28, no. Conf. Syst. Eng. Res., pp. 322–331, 2014.

[14] A. M. Ross and D. H. Rhodes, “An approach for system of systems tradespace exploration,” in *Modeling and Simulation Support for System of Systems Engineering Applications*, L. B. Rainey and A. Tolk, Eds. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2015, pp. 75–98.

[15] T Lochow, “Tools and techniques - DANSE,” *Insight*, vol. 19, no. 3, 2016.

[16] N. Davendralingam and D. DeLaurentis, “An analytic portfolio approach to system of systems evolutions,” *Procedia Comput. Sci.*, vol. 28, pp. 711–719, 2014.

[17] D. M. Curry and C. H. Dagli, “A computational intelligence approach to system-of-systems architecting incorporating multi-objective optimization,” *Procedia Comput. Sci.*, vol. 44, pp. 86–94, 2015.

[18] T. Ender *et al.*, “Systems-of-systems analysis of ballistic missile defense architecture effectiveness through surrogate modeling and simulation,” *IEEE Syst. J.*, vol. 4, no. 2, pp. 156–166, Jun. 2010.

[19] A. Renault and C. Dagli, “Genetic algorithm optimization of SoS meta-architecture attributes for fuzzy rule based assessments,” *Procedia Comput. Sci.*, vol. 95, Pub. 6, pp. 95–102, 2016.

[20] L. Pape and C. Dagli, “Assessing robustness in systems of systems meta-architectures,” *Procedia Comput. Sci.*, vol. 20, pp. 262–269, 2013.

[21] A. J. Turner, W. Monahan, and M. Cotter, “Quantifying the ilities: A literature review of robustness, interoperability, and agility,” in *Proc. 15th Annu. Conf. Syst. Eng. Res.*, Redondo Beach, CA, pp. 1–10, 2017.

[22] R. Albert, H. Jeong, and A.-L. Barabasi, “Error and attack tolerance of complex networks,” *Nature*, vol. 406, no. 6794, pp. 378–382, Jul. 2000.

[23] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, “Attack vulnerability of complex networks,” *Phys. Rev. E., Stat., Nonlinear, Soft Matter Phys.*, vol. 65, no. 5 Pt 2, May 2002, Art. no. 056109.

[24] S. Iyer, T. Killingback, B. Sundaram, and Z. Wang, “Attack robustness and centrality of complex networks,” *PLoS ONE*, vol. 8, no. 4, 2013, Art. no. e59613.

[25] A. Jamakovic and P. V. Mieghem, “On the robustness of complex networks by using the algebraic connectivity,” in *NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, A. Das, H. K. Pung, F. B. S. Lee, and L. W. C. Wong, Eds. Berlin Heidelberg: Springer, 2008, pp. 183–194.

[26] J. Gao, X. Liu, D. Li, and S. Havlin, “Recent progress on the resilience of complex networks,” *Energies*, vol. 8, no. 10, pp. 12 187–12 210, 2015.

[27] P. R. Garvey and C. A. Pinto, “Introduction to functional dependency network analysis,” in *Proc. 2nd Int. Symp. Eng. Syst.*, Cambridge, MA, USA, 2010, pp. 1–17.

[28] M. Efatmaneshnik, J. Bradley, and M. J. Ryan, “Complexity and fragility in system of systems,” *Int. J. Syst. Syst. Eng.*, vol. 7, no. 4, pp. 294–312, 2016.

[29] H. Mehrpouyan, B. Haley, A. Dong, I. Y. Tumer, and C. Hoyle, “Resiliency analysis for complex engineered system design,” *Artif. Intell. Eng. Des., Anal. Manuf.*, vol. 29, no. 1, pp. 93–108, 2015.

[30] B. M. Haley, A. Dong, and I. Tumer, “A comparison of network-based metrics of behavioral degradation in complex engineered systems,” *J. Mech. Des.*, vol. 138, pp. 1–11, 2016.

[31] H. T. Tran, J. C. Domerçant, and D. N. Mavris, “A Network-based cost comparison of resilient and robust system-of-systems,” *Procedia Computer Science*, vol. 95. Elsevier Masson SAS, 2016, pp. 126–133. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1877050916324759

[32] J. Li, Y. Tan, K. Yang, X. Zhang, and B. Ge, “Structural robustness of combat networks of weapon system-of-systems based on the operation loop,” *Int. J. Syst. Sci.*, vol. 48, no. 3, pp. 659–674, 2017.

[33] 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. 1025–8, Apr. 2010.

[34] X. Liu, H. E. Stanley, and J. Gao, “Breakdown of interdependent directed networks,” *Proc. Nat. Acad. Sci.*, vol. 113, no. 5, 2016, Art. no. 201523412.

[35] M. Korkali, J. G. Veneman, B. F. Tivnan, J. P. Bagrow, and P. D. Hines, “Reducing cascading failure risk by increasing infrastructure network interdependence,” *Sci. Rep.*, vol. 7, pp. 1–14, 2017.

[36] M. Kivela, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. a. Porter, “Multilayer networks,” *J. Complex Netw.*, vol. 2, no. 2, Jul. 2014.

[37] M. Ventresca and D. Aleman, “Network robustness versus multi-strategy sequential attack,” *J. Complex Netw.*, no. 3, pp. 126–146, 2015.

[38] J. Wu, M. Barahona, Y.-J. Tan, and H.-Z. Deng, “Spectral measure of structural robustness in complex networks,” *IEEE Trans. Syst., Man, Cybern. - Part A: Syst. Hum.*, vol. 41, no. 6, pp. 1244–1252, 2011.

[39] D. C. Montgomery, *Design and Analysis of Experiments*, 8th ed. Hoboken, NJ, USA: Wiley Sons, Inc., 2013.

[40] E. Borgonovo and E. Plischke, “Sensitivity analysis: A. review of recent advances,” *Eur. J. Oper. Res.*, vol. 248, no. 3, pp. 869–887, 2016.

[41] T. M. Cioppa and T. W. Lucas, “Efficient nearly orthogonal and space-filling latin hypercubes,” *Technometrics*, vol. 49, no. 1, pp. 45–55, Feb. 2007.

[42] H. Shen and H. Wan, “Controlled sequential factorial design for simulation factor screening,” *Eur. J. Oper. Res.*, vol. 198, no. 2, pp. 511–519, 2009.

[43] L. Antul *et al.*, “Toward scaling model-based engineering for systems of systems,” in *Proc. IEEE Aerosp. Conf. Proc.*, Big Sky, MT, pp. 1–9, 2018.

[44] H. T. Tran, M. Balchanos, J. C. Domercant, and D. N. Mavris, “A framework for the quantitative assessment of performance-based system resilience,” *Rel. Eng. Syst. Saf.*, vol. 158, pp. 73–84, 2017.

[45] H. T. Tran, J. C. Domerc, and D. N. Mavris, “Parametric design of resilient complex networked systems,” *IEEE Syst. J.*, vol. 13, no. 2, pp. 1496–1504, Jun. 2019.

Aleksandra Markinda-Khusid received the B.S. degree in physics, M.S. and Ph.D. degrees in electrical engineering, and the M.S. degree in engineering and management, all from the Massachusetts Institute of Technology, Cambridge, MA, USA 1999, 2001, 2005, and 2015 respectively.

She leads the Systems and Mission Analysis Department, MITRE Corporation, McLean, VA, USA. Her research interests include analytical and quantitative systems engineering and mission engineering, including systems of systems engineering, trade space analysis, and decision support as enabled by the modern digital engineering approaches.

Ryan B. Jacobs received the B.S. degree in aerospace engineering from Embry-Riddle Aeronautical University, Daytona Beach, FL, USA, in 2008, the M.S. and Ph.D. degrees from the Georgia Institute of Technology, Atlanta, GA, USA, in aerospace engineering, in 2012 and 2016, respectively.

He is the Principal Engineer with the Systems Engineering Innovation Center, MITRE, McLean, VA, USA. His research interests include systems-of-systems engineering, artificial intelligence applied to systems engineering, and tradespace exploration.

Laura Antul received the B.S. degrees in mathematical sciences from Worcester Polytechnic Institute, Worcester, MA, USA, in 2016.

She is currently a Lead Systems Engineer with the MITRE Corporation, McLean, VA, USA. Her research interests include the application of graph theoretic concepts to allow for lightweight rapid analysis of system of systems (SoS) architectures to provide insight into a variety of different problems in the SoS engineering domain. She has been expanding her work from model-based systems analysis to a modular analysis architecture geared toward analyzing data from multiple different sources in a digital engineering environment.

Lance (Mann Kyo) Cho received the B.S. and M.S. degrees in aerospace engineering from the Georgia Institute of Technology, Atlanta, GA, USA 2014 and 2015, respectively.

He is a Senior Systems Engineer with the MITRE Corporation, McLean, VA, USA.

Huy T. Tran received the B.S. degree in mechanical engineering from North Carolina State University, Raleigh, NC, USA, in 2008, and the M.S. degree in mechanical engineering from the University of Wisconsin-Madison, Madison, WI, USA, in 2010, the M.S. and Ph.D. degrees in aerospace engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2014 and 2015.

He is currently a Research Assistant Professor with the Aerospace Engineering Department, University of Illinois at Urbana-Champaign, Urbana, IL, USA. He is an affiliate of the University of Illinois’ Center for Autonomy, Intelligent Robotics Laboratory, Smart Transportation Infrastructure Initiative, and Applied Research Institute. His research interests include autonomy in unstructured environments and prediction in complex systems, through a combination of reinforcement learning, deep learning, network science, and optimization.