ORIGINAL PAPER

Measuring and optimizing system reliability: a stochastic programming approach

Joshua L. Pulsipher[1] · Victor M. Zavala[1]

Abstract

We propose a computational framework to quantify (measure) and to optimize the reliability of complex systems. The approach uses a graph representation of the system that is subject to random failures of its components (nodes and edges). Under this setting, reliability is defined as the probability of finding a path between sources and sink nodes under random component failures and we show that this measure can be computed by solving a stochastic mixed-integer program. The stochastic programming setting allows us to account for system constraints and general probability distributions to characterize failures and allows us to derive optimization formulations that identify designs of maximum reliability. We also propose a strategy to approximately solve these problems in a scalable manner by using purely continuous formulations.

Keywords Reliability · Design · Network · Topology

Mathematics Subject Classification 68M15 · 90B15 · 90C15 · 68R10

1 Introduction

In this work, we investigate the problem of quantifying the reliability of complex systems and of designing systems of maximum reliability. Such problems have a wide range of applications such as supply chains, transportation networks, energy networks, process networks, sensor networks, and control networks (Kim and Kang 2013). In these applications, it is vital to design systems that maintain functionality in the face of natural and man-made events (e.g., mechanical failures, power outages, weather, and cyber-attacks) (Yan et al. 2012). Despite its practical importance, quantifying the reliability of complex systems remains a technical challenge.

✉ Victor M. Zavala
  victor.zavala@wisc.edu

1 Department of Chemical and Biological Engineering, University of Wisconsin-Madison, 1415 Engineering Dr, Madison, WI 53706, USA

Reliability has been traditionally defined as the probability that a system remains functional under component failures (Ogunnaike 2009). The most prominent model used in industry to quantify reliability is based on so-called reliability block diagrams (RBDs). Here, the system is modeled as a network (a directed graph) of series/parallel paths in which each path has a single source and sink node. The system is said to function under a given failure if there exists at least one path between the source and the sink node. The RBD approach exploits the simple topology of series/parallel systems to analytically compute the reliability of the overall system from the reliability of its individual components (Thomaidis and Pistikopoulos 1994). Here, it is also implicitly assumed that the probability of failure for every component can be chracterized using the same probability distribution. The availability of an analytical measure facilities the design of systems of maximum reliability (Ye et al. 2018). Unfortunately, the RBD approach is difficult to apply to more complex settings that involve, for instance, topologies with multiple source and sink nodes and loops and components with different probability distributions. As a result, analytical reliability measures cannot be easily derived under such settings.

The recursive decomposition algorithm (DFA) is a technique that aims to quantify reliability of more complex network topologies by systematically exploring paths between source and sink nodes (Bistouni and Jahanshahi 2014). This approach is more general but is not amenable for design tasks. Simulation-based approaches such as Monte Carlo (MC) sampling provide a general approach to quantify reliability. These approaches estimate reliability by "probing" the system against failure scenarios and then determine the probability that the system remains functional by averaging the number of scenarios the system is able to withstand (Li et al. 2013). These simulation-based approaches are computationally more expensive than the analytical RBD approach because they require repetitive simulations but can also enable the use of a wide range of stochastic programming formulations and solution techniques (Luedtke and Ahmed 2008). Specifically, we show that reliability can be computed by solving a stochastic mixed-integer program. This framework allows us to handle arbitrary system topologies, probability distributions to characterize different types of failures, and system constraints. Moreover, the stochastic program can be easily incorporated within optimal design formulations. We also provide evidence that accurate solutions for large systems can be obtained by solving purely continuous relaxations.

The paper is structured as follows: Sect. 2 establishes the definition of reliability guiding this work and introduces basic notation. Section 3 provides stochastic programming formulations to compute reliability and to design systems with maximum reliability. Section 4 presents case studies. Section 5 provides concluding remarks.

2 Problem definition and setting

In this section, we present a general graph abstraction to model complex systems. This abstraction is used to motivate and define reliability measures.

2.1 Graph abstraction and model

We model a system as a directed graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$ with components $\mathcal{N}$ (nodes) and $\mathcal{E}$ (edges). We use $n \in \mathcal{N}$ and $e \in \mathcal{E}$ to represent specific nodes and edges in the graph, respectively. The set of edges originating at node n is denoted as $\mathcal{E}_{\text{in}}(n) \subseteq \mathcal{E}$ and the set of edges ending a node n is denoted as $\mathcal{E}_{\text{out}}(n) \subseteq \mathcal{E}$. The set of supporting nodes for an edge e (the pair of nodes connected by the edge) is denoted $\mathcal{N}(e) \subseteq \mathcal{N}$. A schematic representation of the graph notation is provided in Fig. 1.

The topology of the system $\mathcal{G}(\mathcal{N}, \mathcal{E})$ is encoded in the incidence matrix $A \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{E}|}$ where $A_{ne} = 1$ if $e \in \mathcal{E}_{\text{in}}(n)$, $A_{ne} = -1$ if $e \in \mathcal{E}_{\text{out}}(n)$, or $A_{ne} = 0$ otherwise. The nominal topology A is subject to *failures* of its components (nodes and edges); as such, we define the *perturbed* incidence matrix as a random matrix $A(\xi_{\mathcal{N}}, \xi_{\mathcal{E}})$. Here, $\xi_{\mathcal{N}} \in \mathbb{R}^{|\mathcal{N}|}$ is the realization of a discrete (binary) random vector which indicates the set of nodes that function ($\xi_{\mathcal{N},n} = 1$ if node n functions) or do not function ($\xi_{\mathcal{N},n} = 0$ if n does not function). Similarly, $\xi_{\mathcal{E}} \in \mathbb{R}^{|\mathcal{E}|}$ denotes the realization of a binary random vector that indicates the set of nodes that function ($\xi_{\mathcal{E},e} = 1$) or do not function ($\xi_{\mathcal{E},e} = 0$). Under these definitions, the perturbed incidence matrix under realization $\xi := (\xi_{\mathcal{N}}, \xi_{\mathcal{E}})$ can be computed as:

$$A(\xi) := \Xi_{\mathcal{N}} A \Xi_{\mathcal{E}}, \tag{2.1}$$

where $\Xi_{\mathcal{N}} \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{N}|}$, $\Xi_{\mathcal{E}} \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$ are diagonal matrices of the form $\Xi_{\mathcal{N}} = \text{diag}(\xi_{\mathcal{N}})$ and $\Xi_{\mathcal{E}} = \text{diag}(\xi_{\mathcal{E}})$, respectively. In a stochastic programming context, one can interpret $A(\xi)$ as a random technology matrix (Birge and Louveaux 2011). The elements of the perturbed incidence matrix can also be written as:

$$A_{ne}(\xi) = A_{ne} \cdot \xi_{\mathcal{N},n} \cdot \xi_{\mathcal{E},e}, \quad n \in \mathcal{N}, \ e \in \mathcal{E}. \tag{2.2}$$

In other words, $A_{ne}(\xi) = 0$ (entry does not exist) if either node n or edge e fails (do not exist) in scenario ξ.

We use a network flow model to represent paths between nodes. Specifically, we define a set of *source* nodes as $\mathcal{N}_{\text{so}} \subseteq \mathcal{N}$ with associated source flows $d_n > 0$, a set of *sink* nodes as $\mathcal{N}_{\text{si}} \subseteq \mathcal{N}$ with associated sink flows $d_n < 0$, and a set of *relay* nodes as $\mathcal{N}_{\text{re}} \subseteq \mathcal{N}$ with associated flows $d_n = 0$. We observe that the source and sink flows are *fixed*. Under these definitions, the network flow representation can be expressed as:

Fig. 1 Representation of a system as a directed graph with node set $\mathcal{N} = \{n_1, n_2, n_3, n_4\}$ and edge set $\mathcal{E} = \{e_{12}, e_{13}, e_{23}, e_{24}, e_{34}\}$
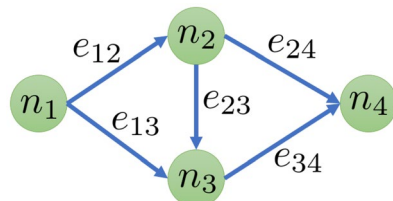
$$\sum_{e \in \mathcal{E}} A_{ne}(\xi)z_e + d_n = 0, \ n \in \mathcal{N} \tag{2.3}$$

where $z_e \in \mathbb{R}_+$ is the flow along edge $e \in \mathcal{E}$. The network flow model can also be expressed in compact form as:

$$A(\xi)z + d = 0. \tag{2.4}$$

In our framework, we expand this basic network flow model to capture the possibility of readjusting flows in order to maintain system functionality. This can be done by allowing some nodes $\mathcal{N}_u \subseteq \mathcal{N}$ to have controllable flows $u_n \in \mathbb{R}_+$. Moreover, in many applications, the edge flows z and the controls u have physical meaning and are, thus, subject to constraints; we capture such constraints using feasible sets $\mathcal{Z} \subseteq \mathbb{R}^{|\mathcal{E}|}$ and $\mathcal{U} \subseteq \mathbb{R}^{|\mathcal{N}_u|}$. With this, we define the extended network flow model as:

$$A(\xi)z + u + d = 0 \tag{2.5a}$$

$$u \in \mathcal{U}, \ z \in \mathcal{Z}. \tag{2.5b}$$

In this representation, the set $\mathcal{U}$ is constructed in a way that it restricts control at certain nodes. For instance, we consider the box control set:

$$\mathcal{U} = \{u \ : \ u_n = 0, \ n \notin \mathcal{N}_u \ \& \ \underline{u}_n \leq u_n \leq \overline{u}_n, \ n \in \mathcal{N}_u\}. \tag{2.6}$$

For simplicity, we assume that the feasible set for flows is also a box set of the form:

$$\mathcal{Z} = \{z \ : \ \underline{z}_e \leq z_e \leq \overline{z}_e, \ e \in \mathcal{E}\}. \tag{2.7}$$

2.2 Reliability measures

A *reliability measure* seeks to quantify the probability that a system remains functional under random component failures. Under a graph representation, the system is said to be functional if there exists at least one path that connects each sink node to a source node. For a particular realization ξ (with associated topology $A(\xi)$) and in the absence of controls and constraints, the functionality of a system can be checked using the *reliability function*:

$$\psi(A, \xi) := \begin{cases} 1 \text{ if } \exists z \ : \ A(\xi)z + d = 0 \\ 0 \text{ otherwise.} \end{cases} \tag{2.8}$$

This function uses the network flow representation to check if there exist a set of flows z that connect sinks and sources. This is based on the observation that, if a path does not exist between a sink and at least one source node (e.g., the network becomes disconnected in a given failure scenario), then there is no set of flows z that satisfies the flow constraint $A(\xi)z + d = 0$.

The traditional definition of reliability does not account for constraints and does not account for the possibility to control flows. To account for these features, we extend the *reliability function* as:

$$\psi(A, \xi, \mathcal{Z}, \mathcal{U}) := \begin{cases} 1 & \text{if } \exists\, z \in \mathcal{Z}, u \in \mathcal{U} : A(\xi)z + u + d = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.9}$$

We use this extended function to define the *reliability measure*:

$$R(A, \mathcal{Z}, \mathcal{U}) := \mathbb{P}(\psi(A(\xi), \mathcal{Z}, \mathcal{U}) = 1). \tag{2.10}$$

This measure function is the probability that the system remains functional. A similar measure has been proposed to measure system *flexibility* which, in our setting, would represent the ability of a system to withstand perturbations in the source and sink flows d (exogenous disturbances) (Straub and Grossmann 1993; Bansal et al. 1998; Swaney and Grossmann 1985; Pulsipher and Zavala 2018). Therefore, we highlight that a key distinction between flexibility and reliability is that the former deals with continuous perturbations, while the later deals with discrete perturbations.

2.3 Designs of maximum reliability

We are interested in using the reliability measure to find system designs that maximize reliability. In this task, one often needs to trade-off cost $c(A, \mathcal{Z}, \mathcal{U})$ and reliability, giving rise to the abstract problem:

$$\begin{aligned} \max_{A, \mathcal{Z}, \mathcal{U}} \;\; & R(A, \mathcal{Z}, \mathcal{U}) \\ \text{s.t.} \quad & c(A, \mathcal{Z}, \mathcal{U}) \leq \epsilon \end{aligned} \tag{2.11}$$

where $\epsilon \in \mathbb{R}$ is a cost budget that is spanned to find Pareto pairs (c^*, R^*). We highlight the dependence of the cost measure and reliability measure on the topological design (given by the incidence matrix A) and on the operational design (given by the constraint sets $\mathcal{Z}, \mathcal{U}$).

3 Stochastic programming formulations

In this section, we provide stochastic programming formulations to compute the proposed reliability measure and to design systems of maximum reliability. We show that these formulations can be easily derived from the network flow representation of the system.

3.1 Computing the reliability measure

We motivate the discussion by considering a simple setting with a single-input and single-output graph. Under this setting, the sets $\mathcal{N}_{\text{so}}$ and $\mathcal{N}_{\text{si}}$ are singletons and thus, the system is said to remain functional if there exists at least one path between the source and the sink node. Equivalently, given a fixed source flow, the

system is functional if we can find a set of edge flows that satisfy the fixed sink flow. Under this logic, we can compute $\psi(A, \xi)$ by finding a feasible solution for a network flow problem and this problem can be cast as a mixed-integer linear program (MILP) of the form:

$$\psi(A, \xi) = \max_{y,z} \; (1 - y)$$

$$\text{s.t.} \quad \sum_{e \in \mathcal{E}} A_{ne}(\xi) z_e = 0, \qquad\qquad\qquad n \in \mathcal{N}_{\text{re}}$$

$$\sum_{e \in \mathcal{E}} A_{ne}(\xi) z_e + d_n \cdot (1 - y) = 0, \; n \in \mathcal{N}_{\text{so}}$$

$$\sum_{e \in \mathcal{E}} A_{ne}(\xi) z_e + d_n \cdot (1 - y) = 0, \; n \in \mathcal{N}_{\text{si}} \tag{3.1}$$

$$z_e \geq 0, \qquad\qquad\qquad\qquad\qquad e \in \mathcal{E}$$

$$y \in \{0, 1\}.$$

Here, we arbitrarily set the source and sink flows to $d_n = 1$ and $d_n = -1$, respectively. This is done without loss of generality because the flows do not necessarily have physical meaning (in more general settings they might have meaning). We use the binary variable $y \in \{0, 1\}$ to relax the balances at the source and sink nodes (i.e., if $y = 0$ then the network flow system has a feasible solution and if $y = 1$ then it does not). If the network flow system does not have a solution, then we obtain the trivial flow solution $z_e = 0$ for all $e \in \mathcal{E}$. We, thus, have that the reliability measure is given by $\psi(A, \xi) = 1 - y^*$ and we note that the maximization problem is equivalent to minimize y. The MILP can be relaxed by setting $0 \leq y \leq 1$; interestingly, this LP is guaranteed to deliver an optimal (binary) solution for the MILP (see Appendix).

Problem (3.1) can be easily generalized to compute the reliability measure for graphs with multiple sources and sinks and with controllable flows. This can be done by solving the MILP:

$$\psi(A, \xi, \mathcal{Z}, \mathcal{U}) = \max_{y,z,u} \; (1 - y)$$
$$\text{s.t.} \quad \sum_{e \in \mathcal{E}} A_{ne}(\xi) z_e = 0, \qquad\qquad\qquad n \in \mathcal{N}_r$$
$$\sum_{e \in \mathcal{E}} A_{ne}(\xi) z_e + d_n \cdot (1 - y) + u_n = 0, \; n \in \mathcal{N}_{\text{so}}$$
$$\sum_{e \in \mathcal{E}} A_{ne}(\xi) z_e + d_n \cdot (1 - y) + u_n = 0, \; n \in \mathcal{N}_{\text{si}} \tag{3.2}$$
$$u \in \mathcal{U}, \; z \in \mathcal{Z}$$
$$y \in \{0, 1\}.$$

This MILP determines if all the sink flows can be satisfied via the source flows (i.e., each sink has at least one path to a source); this is true whenever $y = 0$ (which indicates that none of the source and/or sink nodes needs to be relaxed to achieve a feasible solution).

The MILP representation of the reliability function reveals that the measure $R(A, \mathcal{Z}, \mathcal{U})$ is a *joint chance constraint*. This chance constraint can be approximated using MC samples ξ^k, $k \in \mathcal{K}$ as (Kim et al. 2015):

$$R(A, \mathcal{Z}, \mathcal{U}) \approx \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \psi(A, \xi^k, \mathcal{Z}, \mathcal{U}). \tag{3.3}$$

By the law of large numbers, this sample average approximation becomes asymptotically exact as the number of samples increases (Hsu and Robbins 1947); moreover, the approximation converges exponentially (Kleywegt et al. 2002). Combining problems (3.3) and (3.2), we obtain the following approximation of the reliability measure:

$$
\begin{aligned}
R(A, \mathcal{Z}, \mathcal{U}) \approx \max_{y^k, z^k, u^k} \quad & \frac{1}{|K|} \sum_{k \in \mathcal{K}} (1 - y^k) \\
\text{s.t.} \quad & \sum_{e \in \mathcal{E}} A_{ne}(\xi^k) z_e^k = 0, && n \in \mathcal{N}_{\text{re}}, \ k \in \mathcal{K} \\
& \sum_{e \in \mathcal{E}} A_{ne}(\xi^k) z_e^k + d_n \cdot (1 - y^k) + u_n^k = 0, && n \in \mathcal{N}_{\text{so}}, \ k \in \mathcal{K} \\
& \sum_{e \in \mathcal{E}} A_{ne}(\xi^k) z_e^k + d_n \cdot (1 - y^k) + u_n^k = 0, && n \in \mathcal{N}_{\text{si}}, \ k \in \mathcal{K} \\
& z^k \in \mathcal{Z}, \ u^k \in \mathcal{U}, && k \in \mathcal{K} \\
& y^k \in \{0, 1\}, && k \in \mathcal{K}.
\end{aligned}
\tag{3.4}
$$

This problem is *fully decoupled* in the MC samples $k \in \mathcal{K}$ and, thus, can be trivially parallelized. It has been recently reported that a continuous relaxation of this problem (in combination with an appropriate rounding strategy) provides high-quality approximations of the exact solution (Pulsipher and Zavala 2019). Specifically, we can relax $y^k \in \{0, 1\}$ to $0 \leq y^k \leq 1$ and then round the optimized relaxed y^{k*} values to 1 if they are nonzero. This approach is analogous to employing slack variables to identify active and inactive sets of constraints. In the following section, we provide numerical evidence that this relaxation approach is effective. The exact relaxation result for the simple reliability problem (3.1) provides some intuition as to why this happens. However, establishing a theoretical justification in a more complex setting with constraints and controllable flows is difficult and is left as a topic of future work.

The MILP representation can be extended in a number of ways to capture desirable decision-making logic. For instance, one might want to relax the requirement that paths must exist to all sink nodes and instead require that only a subset of nodes are reachable. This can be done by introducing binary variables for all sink nodes y_n^k and by solving the problem:

$$R(A, \mathcal{Z}, \mathcal{U}) \approx \max_{y^k, z^k, u^k} \frac{1}{|K|} \sum_{k \in \mathcal{K}} L(y^k)$$

$$\begin{aligned}
\text{s.t.} \quad & \sum_{e \in \mathcal{E}} A_{ne}(\xi^k) z_e^k = 0, & n \in \mathcal{N}_{\text{re}},\ k \in \mathcal{K} \\
& \sum_{e \in \mathcal{E}} A_{ne}(\xi^k) z_e^k + d_n \cdot (1 - y_n^k) + u_n^k = 0, & n \in \mathcal{N}_{\text{so}},\ k \in \mathcal{K} \\
& \sum_{e \in \mathcal{E}} A_{ne}(\xi^k) z_e^k + d_n \cdot (1 - y_n^k) + u_n^k = 0, & n \in \mathcal{N}_{\text{si}},\ k \in \mathcal{K} \\
& z^k \in \mathcal{Z},\ u^k \in \mathcal{U}, & k \in \mathcal{K} \\
& y_n^k \in \{0, 1\}, & k \in \mathcal{K},\ n \in \mathcal{N}_{\text{si}} \cup \mathcal{N}_{\text{so}}.
\end{aligned}$$

$$(3.5)$$

Here, $L(y^k)$ is a logic function which is set to one if a subset of sinks of interest are reachable (or is set to zero otherwise).

3.2 Optimal design

The design problem (2.11) aims to make topological and capacity changes to a nominal network to maximize reliability (under a given cost budget). To formulate this problem, we recall that the base topology of the system is given by the graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$ with associated incidence matrix A, nodes $\mathcal{N}$, and $\mathcal{E}$. Our goal is this formulation to expand the number of edges to maximize reliability. This is done by defining an expanded set of edges $\bar{\mathcal{E}}$ such that $\mathcal{E} \subset \bar{\mathcal{E}}$. The expanded set of edges has an associated incidence matrix $\bar{A}$. In other words, the new incidence matrix has an expanded set of connections between the nodes. We represent the added set of edges as $\hat{\mathcal{E}} := \bar{\mathcal{E}} \backslash \mathcal{E}$. In our design problem, we also seek to expand the set of feasible edge flows and control flows (to model capacity expansions). The design problem is cast as the following MILP:

$$\max_{v, \underline{z}, \overline{z}, \underline{u}, \overline{u}, z^k, y^k, u^k} \frac{1}{|K|} \sum_{k \in \mathcal{K}} L(y^k)$$

$$\begin{aligned}
\text{s.t.} \quad & c(v, \underline{z}, \overline{z}, \underline{u}, \overline{u}) \leq \epsilon \\
& \sum_{e \in \bar{\mathcal{E}}} \bar{A}_{ne}^k z_e^k = 0, & n \in \mathcal{N}_{\text{re}},\ k \in \mathcal{K} \\
& \sum_{e \in \bar{\mathcal{E}}} A_{ne}^k z_e^k + d_n \cdot (1 - y_n^k) + u_n^k = 0, & n \in \mathcal{N}_{\text{so}},\ k \in \mathcal{K} \\
& \sum_{e \in \bar{\mathcal{E}}} A_{ne}^k z_e^k + d_n \cdot (1 - y_n^k) + u_n^k = 0, & n \in \mathcal{N}_{\text{si}},\ k \in \mathcal{K} \\
& A_{ne}^k = \bar{A}_{ne} \cdot \xi_{\mathcal{N},n}^k \cdot \xi_{\mathcal{E},e}^k, & n \in \mathcal{N},\ e \in \bar{\mathcal{E}},\ k \in \mathcal{K} \\
& A_{ne}^k = \bar{A}_{ne} \cdot \xi_{\mathcal{N},n}^k \cdot \xi_{\mathcal{E},e}^k \cdot v_e, & n \in \mathcal{N},\ e \in \hat{\mathcal{E}},\ k \in \mathcal{K} \\
& \underline{z} \leq z^k \leq \overline{z},\ \underline{u} \leq u^k \leq \overline{u}, & k \in \mathcal{K} \\
& y_n^k \in \{0, 1\}, & n \in \mathcal{N}_{\text{si}} \cup \mathcal{N}_{\text{so}},\ k \in \mathcal{K} \\
& v_e \in \{0, 1\}, & e \in \hat{\mathcal{E}} \\
& \underline{z}, \overline{z} \in \overline{\mathcal{Z}},\ \underline{u}, \overline{u} \in \overline{\mathcal{U}}.
\end{aligned}$$

$$(3.6)$$

Here, the sets $\overline{\mathcal{Z}}$ and $\overline{\mathcal{U}}$ include possible design values for flow and control bounds. Also, $v \in \{0, 1\}^{|\hat{\mathcal{E}}|}$ denote topological design variables for selecting which of the candidate edges are included in the new design (if none are added then $v_e = 0$ for all $e \in \hat{\mathcal{E}}$ and the network retains its nominal topology). We note that the abstract design cost function $c(A, \mathcal{Z}, \mathcal{U})$ can now be expressed in the parametric form $c(v, \underline{z}, \overline{z}, \underline{u}, \overline{u})$.

The proposed design formulation seeks to highlight the modeling flexibility provided by the proposed stochastic programming framework.

4 Case studies

We analyze the behavior of the proposed framework by applying it to distribution networks. We consider a simple three-node network and the IEEE-14 power distribution network. We also consider a simple parallel–series RBD system to illustrate how the proposed stochastic programming framework is consistent with the analytical RBD solution. All formulations are implemented in `JuMP` 0.18.5 (Dunning et al. 2017) and are solved using `Gurobi` 7.5.1 on a Intel® Core™ i7-7500U machine running at 2.90 GHz with 4 hardware threads and 16 GB of RAM. All results can be reproduced using the scripts provided in https://github.com/zavalab/JuliaBox/tree/master/ReliableDesign.

4.1 Reliability of parallel–series systems

We consider a simple parallel–series system to highlight that the stochastic programming approach is consistent. The system of interest is represented by the reliability block diagram shown in Fig. 2. This system seeks to pump a flow stream using two pumps and valves in parallel. The parallel design topology enhances the reliability of the system (compared to a topology with a single pump and valve).

Traditional RBD methods can be leveraged to obtain an analytic representation for the overall reliability of the system since this system features a single source and sink (Bistouni and Jahanshahi 2014). In particular, the analytic reliability measure is computed by aggregating the component reliabilites according to their respective connectivities. Specifically, the reliability of m components in series configurations is given by:

$$R_s = \prod_i^m R_i. \tag{4.1}$$

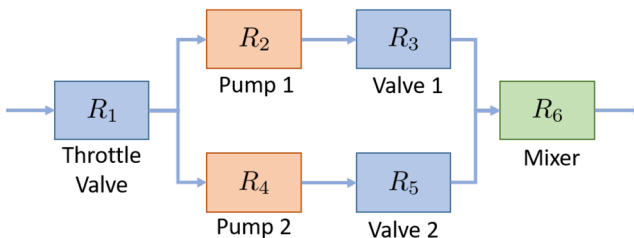The reliability of a parallel configuration is given by:

Fig. 2 Reliability block diagram for a pump system

$$R_p = 1 - \prod_i^m (1 - R_i). \tag{4.2}$$

Following these basic rules, the reliability of the system of interest can be computed as:

$$R_{overall} = R_1(1 - (1 - R_2 R_3)(1 - R_4 R_5))R_6. \tag{4.3}$$

For simplicity, we let each component lifetime be described by an exponential distribution with a mean lifetime of 100 years and we evaluate reliability after 5 years of operation. The reliability for each component is given by the exponential cumulative distribution function evaluated at 5 years (i.e., $R_i = \exp(-5/100)$). From Eq. (4.3), we, thus, obtain the overall reliability $R_{overall} = 89.66\%$.

To demonstrate the equivalence of the proposed stochastic programming setting, we use MC samples drawn from the component distribution functions (a component fails if the lifetime is above the desired threshold of 5 years). We use a total of 10,000 MC samples and solve Problem (3.4). Here, we let the throttle valve be the source node and the mixer be the sink node with $d_n = 1$ and $d_n = -1$, respectively. Furthermore, we set $\mathcal{U} = \emptyset$ (no controls) and $\mathcal{Z} = \mathbb{R}_+^{|\mathcal{E}|}$. Using this approach, the reliability measure is $R(\mathcal{A}, \mathcal{Z}, \mathcal{U}) = 89.69\%$, which is close to the analytical solution.

4.2 Network models

The systems under study are illustrated in Figs. 3 and 4. We consider a simple 3-node distribution network and the IEEE 14-node power network benchmark problem. In these cases, the sink nodes $n \in \mathcal{N}_{si}$ have a fixed flow d_n and the source nodes $n \in \mathcal{N}_{so}$ are controllable with capacity $\bar{u}$. Furthermore, the edges have a finite capacity $\bar{z}$. The 3-node network features a single source (a power plant) and three sink nodes (power consumers). The IEEE 14-node network exhibits a more complex topology with multiple sinks and sources. The data for this problem are obtained from MATPOWER (Zimmerman et al. 2010).

For our design studies, we consider a cost function of the form:

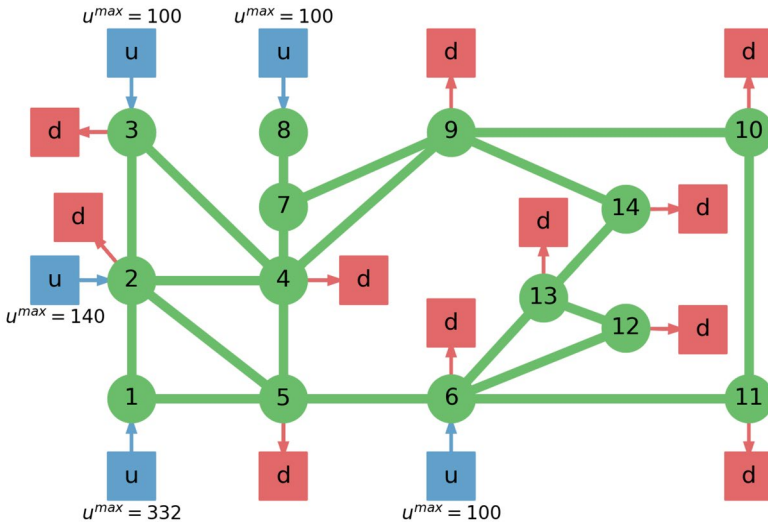

Fig. 3 Schematic of 3-node distribution network
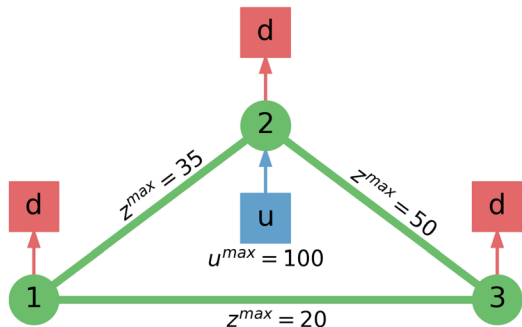
Fig. 4 Schematic of IEEE 14-node distribution network

$$c(v, \bar{z}, \bar{u}) := \sum_{e \in \mathcal{E}} (100 \cdot v_e + \bar{z}_e) + \sum_{n \in \mathcal{N}_u} \bar{u}_n. \tag{4.4}$$

We consider random failures for relay nodes, source nodes, and sink nodes. Here, we model the lifetimes of the components as exponential random variables with mean lifetimes of 100, 80, and 50 years, respectively. MC failure scenarios $\xi_{\mathcal{N}}^k$, $\xi_{\mathcal{E}}^k$ are generated by sampling the exponential distributions and the components are set to failure mode if their lifetime is above a certain threshold value. The thresholds for the 3-node and IEEE 14-node networks were set to 5 and 2 years, respectively. Also, we use nominal line capacity limits of $z^{max} = 100$ for the IEEE 14-node network since these are not provided by MATPOWER.

4.3 Design for maximum reliability (capacity expansion)

We first consider a design problem in which capacity is expanded (i.e., topological expansion variables v are omitted). For the 3-node power network, we solve the MILP formulation to obtain 6 Pareto pairs and we use 1000 MC samples. The Pareto pairs are plotted in Fig. 5. Here, we note that the Pareto frontier shows abrupt changes; this is because this system is simple and, thus, the solution space is small. A manifestation of this limited spaces is that the maximum possible reliability for this system is just 51.4%. This indicates that, regardless of how much capacity is provided (unlimited budget), this network will never achieve a higher reliability because of its limiting topology. In other words, the only way to increase reliability is to add edges.

We explore the Pareto solutions obtained with $\epsilon = 30$ and $\epsilon = 45$. The optimized capacities for these solutions are shown in Fig. 6. Here, the increased
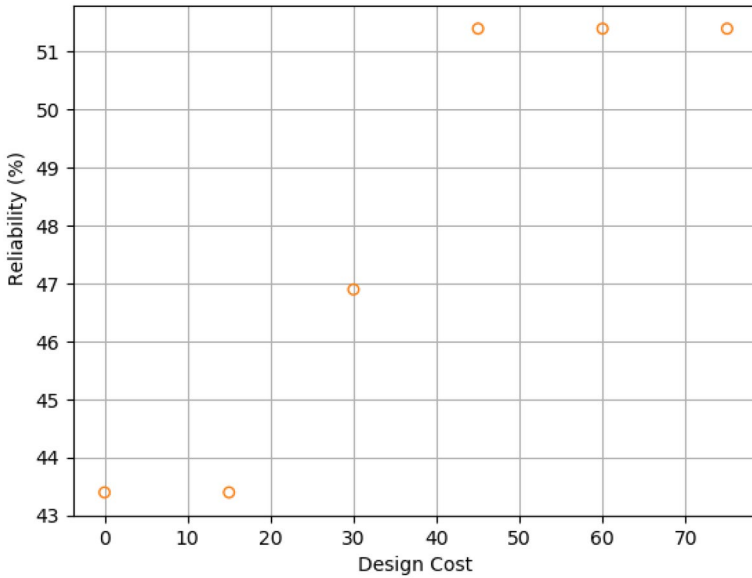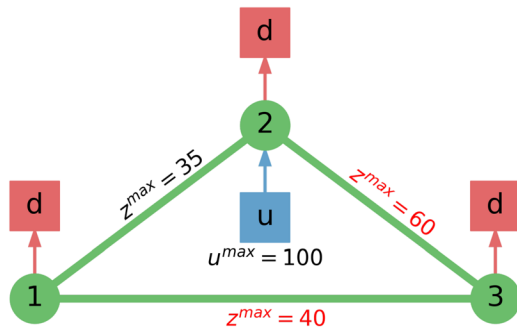
Fig. 5 Pareto frontier for optimal capacity design of 3-node network using 1000 MC samples

Fig. 6 Schematic of the 3-node network corresponding to Pareto pair shown in Fig. 5 with a design cost of 30

capacities relative to the base design are highlighted in red. In the first case, enough capacity is added to the edges connecting nodes 2, 3, and 1 to permit the network to function in the event that the edges connecting nodes 1 and 2 fails. In the other case, enough capacity is added to the edges to permit feasible operation if any one edge fails.

We apply the same design formulation to the IEEE-14 power network problem. We compute a total of 13 Pareto pairs by varying the budget ϵ from 0 to 1800 and we use 2000 MC samples. The solutions obtained with the MILP formulation are presented in Fig. 7. We see that this system shows a smoother Pareto frontier because the solution space for this more complex system is larger. For this system, the largest possible reliability is 78.7% (this system has more degrees of freedom).
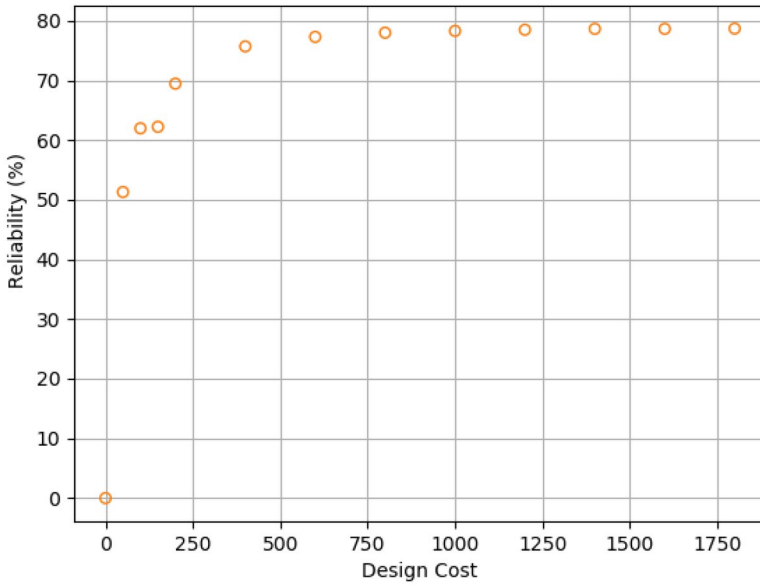
Fig. 7 Pareto frontier for optimal design problem of IEEE 14-node network

The design obtained with a budget of $\epsilon = 400$ is shown in Fig. 8. The expanded capacities are highlighted in red. The capacity of the supplier attached to relay node 6 is significantly expanded (which occurs because it is the only supplier that serves
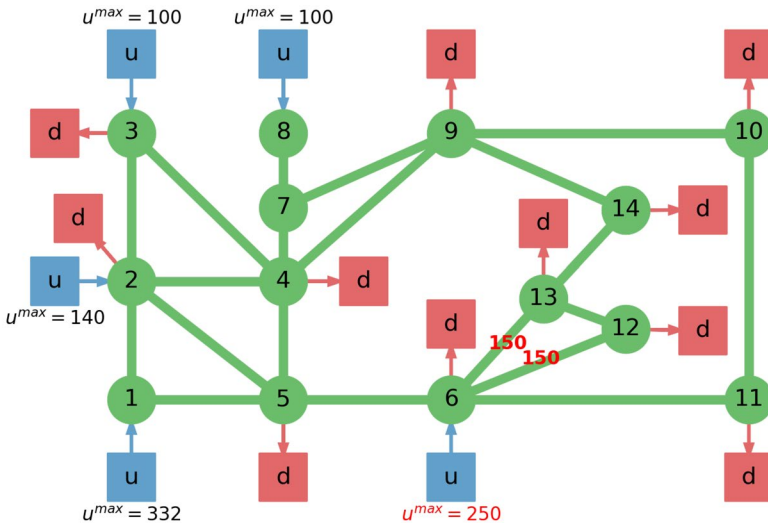

Fig. 8 Schematic of the IEEE 14-node network corresponding to Pareto pair shown in Fig. 7 with a cost of 400

the right side of the network). The capacities of two edges attached to node 6 are also increased such that the internal demands can be satisfied if either edge fails. It is interesting to note that these 3 simple changes to the network design significantly increase the overall reliability of the system (they increase it by 24.4%).

4.4 Continuous approximation for design problem

We consider a continuous relaxation of the design problem; here, integer solutions are obtained by solving the relaxation and then using simple rounding. This technique is first applied to the 3-node power network using the same samples and ϵ values considered above in Sect. 4.3. In Fig. 9, we juxtapose the resulting Pareto pairs. We observe that 5 out of 6 pairs are exactly recovered and a pair is underestimated. In Pulsipher and Zavala (2019) it is hypothesized that the quality of the approximations is the result of degeneracy associated with the joint chance constraint (i.e., multiple solutions yield the same optimal value). This simple network exhibits little degeneracy at that solution because its solution space is small.

Table 1 summarizes the performance of the MILP formulation and the continuous relaxation for the 3-node network. We observe that 5 of the 6 pairs are exactly equivalent since they have no differences in the active constraints. Also, the third pair only differs by 3.5% (which is a small gap). For this small network, the solution times are negligible; so, the benefits of the relaxation are not obvious.
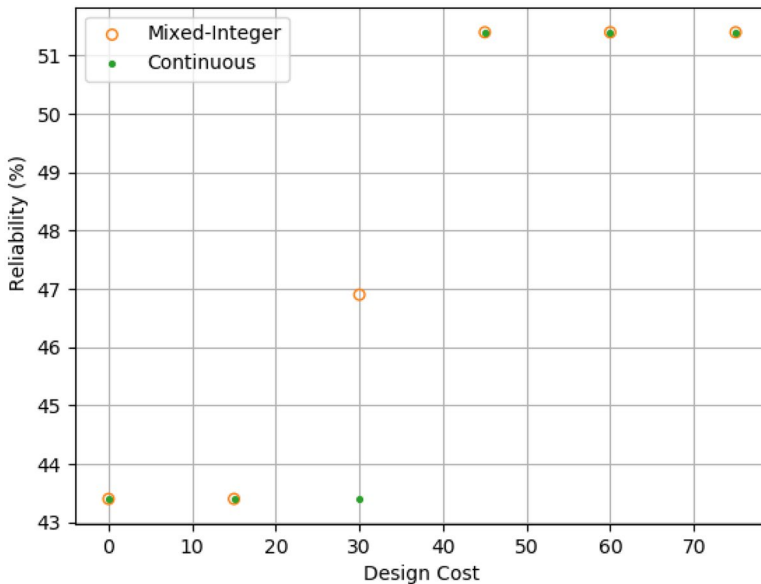

Fig. 9 Pareto frontier for optimal capacity design of the 3-node network juxtaposing the pairs obtained from the full MILP formulation and its continuous relaxation

Table 1 Performance results obtained for 3-node network using the MILP design formulation and continuous relaxation

Cost	MILP R (%)	LP R (%)	MILP time (s)	LP time (s)	Active difference (%)
0	43.4	43.4	0.0467	0.0349	0
15	43.4	43.4	0.0400	0.0468	0
30	46.9	43.4	0.0407	0.0478	3.5
45	51.4	51.4	0.0562	0.0478	0
60	51.4	51.4	0.0526	0.0478	0
75	51.4	51.4	0.0443	0.0458	0

The relaxation strategy was also applied to the IEEE 14-node power network using the same conditions specified above in Sect. 4.3. A juxtaposition of the Pareto pairs is shown in Fig. 10. We observe that the frontier is approximated well (the majority of the pairs being exactly reproduced). Some of the minor discrepancies are attributed to numerical precision. A summary of the results is shown in Table 2. With the exception of the third pair, the Pareto pairs only exhibit differences in the active constraints of less than 1%. We can, thus, see that the relaxation indeed delivers high-quality approximation. Importantly, we observe that the computational time is reduced by 96%. This enables us to handle much larger networks than would be possible using the full MILP formulation.
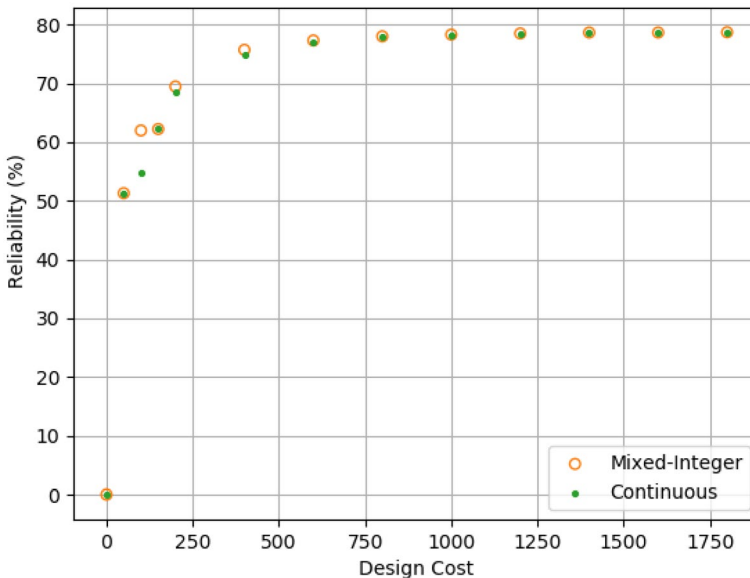

Fig. 10 Pareto frontier for optimal capacity design of the IEEE 14-node network juxtaposing the pairs obtained from the full MILP formulation and its continuous relaxation

Table 2 The performance results obtained for the IEEE 14-node network using the mixed-integer and continuous capacity design formulations

Design cost (–)	MILP R (%)	LP R (%)	MILP time (s)	LP time (s)	Active difference (%)
0	0	0	5.42	1.23	0
50	51.3	51.3	58.84	12.84	0
100	62	54.9	78.62	3.68	7.1
150	62.25	62.25	95.09	5.75	0.5
200	69.45	68.4	83.35	4.75	1.05
400	75.7	74.85	124.46	4.84	0.85
600	77.3	77.1	168.61	7.39	0.8
800	78	77.9	185.11	3.97	0.2
1000	78.3	78.1	205.75	7.2	0.5
1200	78.5	78.45	186.34	3.27	0.05
1400	78.65	78.55	125.08	4.14	0.1
1600	78.65	78.65	151.48	2.57	0.1
1800	78.7	78.7	108.57	1.72	0

4.5 Design for maximum reliability (topological expansion)

We apply the MILP formulation to the 3-node power network including the use of the topological design variables v. We recall that these design variables determine if a particular edge is added to the system. In other words, this more complex formulation chooses an optimal design configuration of edges and capacities where it enforces a fixed upfront cost for the use of each edge. A total of 7 Pareto solutions were computed by varying the budget ϵ from 0 to 750 and the same samples mentioned above are used. These solutions are presented in Fig. 11. A nonzero R index is not obtained until a budget of 600 is employed since at least 6 edges are required to allow the network to function and the capital cost of each line is 100. After this, capacity increases help improve the network until the R index is maximized by adding all the edges and adding extra capacity resulting in the same best possible optimal design considered shown in Fig. 6.

The optimal design for a budget of 650 is depicted in Fig. 12 (left) and this is compared against the design with maximum budget (right). The edges that are switched off are plotted in gray. Here, we observe that the trade-off design is able to effectively operate relative to the sample set without one of the relay edges with the addition of some capacity. The design of maximum budget employs all of the edges with enough capacity to operate if any relay edge fails (making it the most robust design).

We also considered topology and capacity design decisions for the IEEE 14-node power network. A total of 27 Pareto pairs were obtained by varying the budget ϵ from 0 to 4700. The solutions are shown in Fig. 13. The Pareto frontier was obtained using the continuous relaxation. An average of 56 seconds were
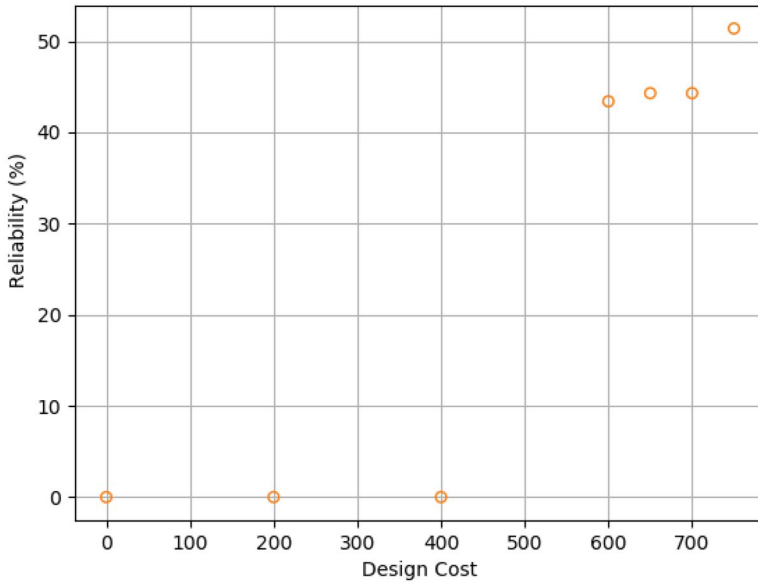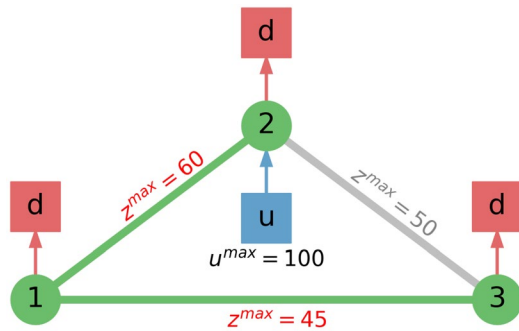
Fig. 11 Pareto frontier for optimal topological and capacity design of the 3-node network obtained from the full MILP formulation

Fig. 12 Schematic of the 3-node network corresponding to the Pareto pair shown in Fig. 11 with a design cost of 650

needed to compute the frontier. The reduced solution times allow us to explore more designs and to explore the reliability limits of the system. In other words, even if the relaxation cannot be guaranteed to provide an exact solution, it captures general behavior and, thus, can be used as a exploratory tool.

The Pareto pair with a budget of 2500 is shown in Fig. 14. The modified capacities are highlighted in red and the edges not in use are colored gray. Here, we observe that reliable performance can be obtained simply by adding capacity to the right supplier and most of the edges, except the edges connecting relay nodes 1–2 and 4–7. Interestingly, this analysis shows that these edges do not impact reliability and can be eliminated (this elimination is not obvious). This highlights
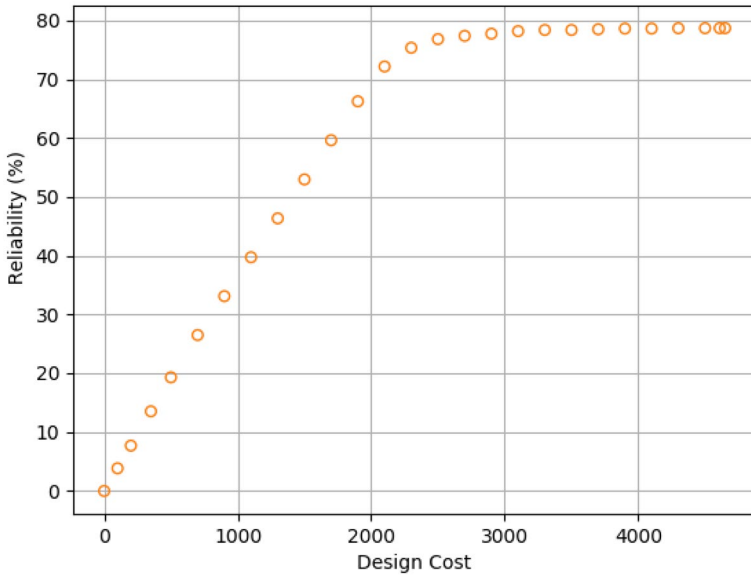
Fig. 13 Pareto frontier for optimal topological and capacity design for IEEE 14-node network (using continuous relaxation)
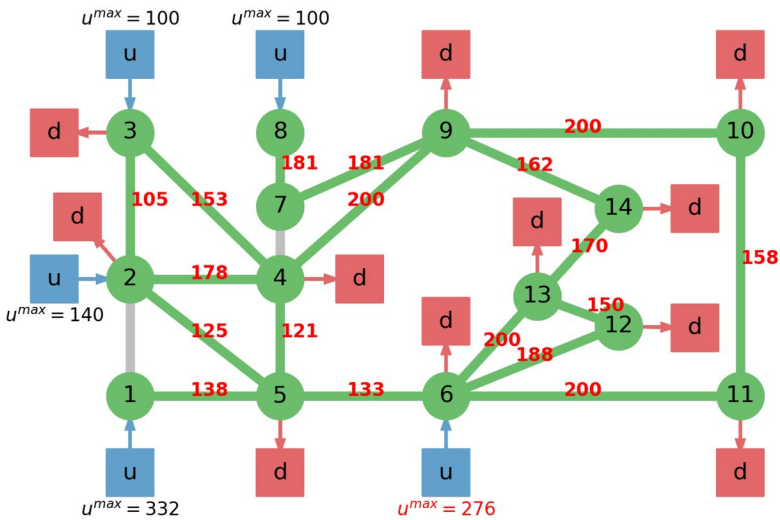

Fig. 14 Schematic of the IEEE 14-node network corresponding to Pareto pair shown in Fig. 13 with a cost of 2500

that the use of systematic reliability analysis techniques in being can help to determine which components are truly needed and avoid over-engineering.

5 Conclusions

We propose stochastic programming formulations to compute the reliability of complex systems. Specifically, the proposed reliability measure uses a graph representation of the system and aims to identify the probability that sink nodes are reachable by source nodes. This measure can be computed by solving a network flow problem, can be easily extended to incorporate constraints, and can be easily embedded in design formulations. We also show that the reliability measure can be computed by solving a stochastic mixed-integer program and that a continuous relaxation of this problem provides high-quality solutions. Case studies are provided to demonstrate the developments.

Appendix: Quality of relaxation for simple setting

Theorem 1 *The relaxation of problem* (3.1) *is exact.*

Proof We express problem (3.1) (denoted as P) in vector form as:

$$\psi(A, \xi) = \max_{y,z} \ (1 - y)$$
$$\text{s.t.} \quad A(\xi)z = \hat{d} \cdot (1 - y)$$
$$z \geq 0$$
$$y \in \{0, 1\},$$

where we define $\hat{d} := -d$ to express the constraints in a standard linear form. The relaxed problem (denoted as $\bar{P}$) is obtained by replacing $y \in \{0, 1\}$ with $\bar{y} \in [0, 1]$. We denote optimal solutions for $\bar{P}$ and P as $\bar{y}^*$ and y^*, respectively. We show that $\bar{P}$ delivers optimal solutions for P by analyzing three possible cases. First consider the case in which $\hat{d} \in \mathcal{R}(A(\xi))$ (where $\mathcal{R}(\cdot)$ denotes the range of the input matrix) and there exists a nontrivial flow solution ($z_j^* > 0$ for some j) such that $A(\xi)z^* = \hat{d}$. This implies that all values of y are feasible since $(1 - y)\hat{d} \in \mathcal{R}(A(\xi))$, $\forall y \in \mathbb{R}$. Thus, $y^* = \bar{y}^* = 0$ must be optimal solutions (yielding the largest possible objective), since any other feasible value of y would have a lower objective. The second case is that in which $\hat{d} \in \mathcal{R}(A(\xi))$ and there does not exist a nontrivial solution ($z_j^* > 0$ for some j) such that $A(\xi)z^* = \hat{d}$. In this case, the only feasible solution is the trivial solution $z^* = 0$ and thus $y^* = \bar{y}^* = 1$. The third and final case corresponds to $\hat{d} \notin \mathcal{R}(A(\xi))$; it follows that $(1 - y)\hat{d} \in \mathcal{R}(A(\xi))$ if and only if $y = 1$ since any scalar multiple of $\hat{d}$ will lie outside of $\mathcal{R}(A(\xi))$ except for the trivial case that $\hat{d} = 0$. Thus, the only feasible (and, therefore, optimal) solution to both problems is $y^* = \bar{y}^* = 1$. $\qquad\square$

References

Bansal V, Perkins JD, Pistikopoulos EN (1998) Flexibility analysis and design of dynamic processes with stochastic parameters. Comput Chem Eng 22:S817–S820

Birge JR, Louveaux F (2011) Introduction to stochastic programming. Springer, Berlin

Bistouni F, Jahanshahi M (2014) Analyzing the reliability of shuffle-exchange networks using reliability block diagrams. Reliabil Eng Syst Saf 132:97–106

Dunning I, Huchette J, Lubin M (2017) Jump: a modeling language for mathematical optimization. SIAM Rev 59(2):295–320

Hsu P-L, Robbins H (1947) Complete convergence and the law of large numbers. Proc Nat Acad Sci USA 33(2):25

Kim Y, Kang W-H (2013) Network reliability analysis of complex systems using a non-simulation-based method. Reliabil Eng Syst Saf 110:80–88

Kleywegt AJ, Shapiro A, Mello TH (2002) The sample average approximation method for stochastic discrete optimization. SIAM J Optim 12(2):479–502

Li W et al (2013) Reliability assessment of electric power systems using Monte Carlo methods. Springer, Berlin

Luedtke J, Ahmed S (2008) A sample approximation approach for optimization with probabilistic constraints. SIAM J Optim 19(2):674–699

Ogunnaike BA (2009) Random phenomena: fundamentals of probability and statistics for engineers. CRC Press, Boca Raton

Pulsipher JL, Zavala VM (2018) A mixed-integer conic programming formulation for computing the flexibility index under multivariate Gaussian uncertainty. Comput Chem Eng 119:302–308

Pulsipher JL, Zavala VM (2019) A scalable stochastic programming approach for the design of flexible systems. Comput Chem Eng 20:20

Straub DA, Grossmann IE (1993) Design optimization of stochastic flexibility. Comput Chem Eng 17(4):339–354

Sujin K, Raghu P, Henderson Shane G (2015) A guide to sample average approximation. Handbook of simulation optimization. Springer, Berlin, pp 207–243

Swaney RE, Grossmann IE (1985) An index for operational flexibility in chemical process design. Part I: Formulation and theory. AIChE J 31(4):621–630

Thomaidis TV, Pistikopoulos EN (1994) Integration of flexibility, reliability and maintenance in process synthesis and design. Comput Chem Eng 18:S259–S263

Yan Y, Qian Y, Sharif H, Tipper D (2012) A survey on cyber security for smart grid communications. IEEE Commun Surv Tutor 14(4):998–1010

Ye Y, Grossmann IE, Pinto JM (2018) Mixed-integer nonlinear programming models for optimal design of reliable chemical plants. Comput Chem Eng 116:3–16

Zimmerman RD, Murillo-Sánchez CE, Thomas RJ (2010) Matpower: steady-state operations, planning, and analysis tools for power systems research and education. IEEE Trans Power Syst 26(1):12–19