Flow-Network-Based Method for the Reliability Analysis of Decentralized Power System Topologies With a Sequential Monte-Carlo Simulation

Matheus Montanini Breve, Bernd Bohnet, Gabriele Michalke, Julia Kowal, *Member, IEEE*, and Kai Strunz, *Senior Member, IEEE*

***Abstract*—This article introduces a sequential Monte-Carlo simulation method (MCS) based on flow networks to represent decentralized power systems and evaluate the reliability and availability of their topologies. It can be used to provide a comparison basis during the system design phase. The method can be employed to simulate a variety of types of decentralized local power systems. It is able to simulate bidirectional, radial, and meshed systems with both repairable and nonrepairable elements. Examples of elements that can be simulated include power sources, loads, and energy-storage units. The determination of system success is performed using a minimum-cost-maximum-flow graph algorithm that calculates the power delivered to the system loads given the state of the system components. If the loads are being properly supplied, the system is deemed functional. The simulation can accommodate additional sets of rules for system success if needed. The method can generate a variety of indices for system availability and reliability based on the system topology and the parameters of its individual components. Three example systems are presented and simulated to showcase the method's capabilities.**

***Index Terms*—Critical infrastructure, decentralized power systems, flow network, graph theory, microgrids, minimum-cost maximum flow (MCF), Monte-Carlo simulation (MCS), reliability analysis.**

I. Introduction

ACCORDING to the Emissions Gap Report of the United Nations Environment Programme released in 2020, the energy transformation sector is the largest contributor to greenhouse gas emissions worldwide. Electricity and heat generation alone, both belonging to the energy sector, account for 24 % of all emissions in the last decade [1]. Current research efforts are focused on reducing the sector footprint by either introducing novel technologies based on renewables or by optimizing the efficiency of existing ones.

These efforts go hand in hand with the ongoing electrification of other sectors. The most obvious example is the transport sector, where battery-electric or fuel-cell-based vehicles are expected to replace the majority of internal combustion vehicles in the next decades. The higher level of electrification poses challenges to the current centralized power generation, transmission, and distribution systems, one of the reasons decentralized energy resources [2] are considered to be future critical elements in the energy sector alongside demand-side energy management [3], [4] in a smart grid environment [5]. Decentralized energy systems allow power generation facilities to be geographically closer to the consumers, reducing transmission losses but at the same time requiring a more complex coordination of resources.

The higher level of decentralization and complexity of power systems and its effects on critical infrastructures are being investigated in multiple aspects [6], [7], [8]. The constituent systems of critical infrastructures must fulfill strict reliability and availability requirements, since unplanned downtimes can significantly disrupt businesses and individuals and cause far-reaching consequences, whose extent strongly depend on the type of infrastructure. Hospitals require a highly available electrical power grid to supply their essential medical equipment, using diesel generators to provide redundancy in the event of failures. The same applies to IT infrastructures, such as data centers, the backbone of the Internet. They provide a number of essential services, enabling for example the global exchange of information, a key aspect of the modern age, given that work, communication and entertainment are increasingly dependent on the Internet.

The design of reliable critical infrastructures is crucial to prevent or minimize the effects of disruptions. The increased complexity, decentralization, and the addition of backup energy supplies in power systems, such as batteries or diesel generators, calls for supplementary redundancy quantifications beyond those given by the $N-1$, $2N$, or similar criteria. An advanced and flexible method is needed to evaluate the

Manuscript received 20 May 2022; revised 20 December 2022 and 27 October 2023; accepted 21 November 2023. This work was supported by Robert Bosch GmbH. Associate Editor: Steven Li. *(Corresponding author: Matheus Montanini Breve.)*

Matheus Montanini Breve, Bernd Bohnet, and Gabriele Michalke are with the Department of Corporate Research at Robert Bosch GmbH, 71272 Renningen, Germany (e-mail: matheus.m.breve@gmail.com; bernd.bohnet@de.bosch.com; gabriele.michalke@de.bosch.com).

Julia Kowal and Kai Strunz are with the Institute of Energy and Automation Technology, Technical University of Berlin, 10587 Berlin, Germany (e-mail: julia.kowal@tu-berlin.de; kai.strunz@tu-berlin.de).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TR.2023.3336359.

Digital Object Identifier 10.1109/TR.2023.3336359

reliability and availability of power systems, while taking advantage of state-of-the-art computational methods. It must be able to evaluate systems with multiple dependencies, such as will be defined as systems whose analytical analysis is not feasible if no simplifications are assumed, for example, while using reliability block diagrams (RBDs) or fault tree analyses (FTAs). Examples of complex systems are, to name only a few, meshed power networks, systems with components with unique failure distributions, systems with multiple sources each with a different power production rating, or systems with multiple energy storage units.

As a solution to the challenges mentioned previously, this article presents a flow-network-based sequential Monte-Carlo simulation (SMCS) method for the topological analysis of reliability and availability of decentralized power supply systems (DPS), with special focus given to multienergy systems (MESs) with the size of a microgrid. As such, it combines two established techniques, graph theory and sequential stochastic simulations, to model a variety of topologies and components, such as sources, loads, and energy storage units. Stochastic failure and repair behaviors can be assigned to each component. Dependencies can be simulated as well, provided they can be described by the network graph structure, for example, if a component fails, the capacities of its edges can be set to null or, if it is in a damaged state, they can be set to a fraction of their nominal capacity. The presented method is meant to analyze decentralized local power systems in the design phases before all electrical parameters are known and set. Power lines in such systems are often short and they are typically designed not to impose a voltage constraint, meaning that only the maximum rated power is relevant for the proposed method. Flow networks are particularly useful in this case, since parameters, such as line impedances or voltage levels, are not required to simulate and evaluate the system topology at first. That is due to the fact that power, not current, is the single commodity of the flow network.[1]

The rest of this article is organized as follows. Section I explains the motivation and context. Section II is a short review of existing methods to assess the reliability and availability of systems with focus on DPS, their applications, and limitations. Section III describes the proposed model of DPS as flow networks and what characterizes their successful operation. Finally, it explains how the proposed method can be used to assess the reliability and availability of a variety of local power systems with an SMCS. Section IV contains a short description of how the method was implemented with the Python programming language. Section V shows simulation results of three scenarios: parallel arrangement of nonrepairable sources, parallel arrangement of repairable sources, and a complex repairable system with repairable components. Section VI discusses the advantages and limitations of the proposed method. Finally, Section VII concludes this article.

[1]The commodity of a flow network is what the flow in the graph represents, i.e., what the edges in the network carry. For example, in logistics studies, the network flow can represent the number of products or packages. In traffic engineering studies, it could be used to model the capacity of a road segment. Some systems might require multiple flow types and belong to a different class of problems, called "multicommodity flow problems."

II. Literature Review

There are essentially two types of reliability analysis: inductive and quantitative. The objective of inductive reliability methods, such as preliminary hazard analysis, failure mode and effects analysis, or hazard and operability analysis is to identify potential failures or situations that will cause a significant disruption or hazard in the operation of a system. Common causes for failures include functional causes, for example the failure of critical components, manufacturing errors, or human intervention [9]. These methods are not addressed in this work, as they do not provide quantitative reliability values, which can only be obtained via quantitative methods.

The objective of quantitative methods is, as indicated before, to obtain numerical values that characterize the reliability and availability of individual devices, collections of equipment, or subsystems. Common values used to characterize the reliability are: mean time between failures (MTBF); failure rate $\lambda(t)$; inherent and operational availabilities, A_i and A_o, respectively; and the time-dependent reliability in a given mission time t, $R(t)$. Quantitative methods can be subdivided into four additional categories: analytical, stochastic simulation-based, physical-model-based, and data-driven methods.

Analytical methods of reliability analysis rely on mathematical models to represent the structure of a system and to calculate its reliability. The most commonly employed methods in this category are RBD [10], FTA [11], [12], and Markov-based approaches [13], [14]. They can be used to analyze systems found in a variety of applications, e.g., in aviation, whose systems can be described by combinations of configurations such as series- and parallel-connected components, stand-by redundancies, or k-out-of-N connections. Some systems cannot be reduced to these configurations and, as a consequence, require other analytical techniques to be analyzed, such as the sum of disjoint products (SDPs) [15], [16], minimal cutsets (mincuts) [17], pivotal decomposition (PD, also called *binary decision diagram*) [18], or state enumeration.

The SDP, mincuts, and PD techniques represent the system as a graph of nodes and edges and checks its connectivity to predict whether the system is functional or not. The nodes correspond to the physical components, while the edges describe how nodes are interconnected. The system state is given by a structure function $\varphi(\mathbf{x}, t)$, where $\mathbf{x}$ is a vector containing the states of each individual system component at the time t. The first step to obtain the system reliability is to add two special nodes: a source σ and a target τ. The system is deemed functional if there is at least one viable path between σ and τ. Paths can be represented by a boolean product of component states. Often, to calculate the system reliability, this boolean expression has to be made disjoint, a problem that has attracted the attention of several researchers for at least five decades [19], [20], [21], [22], [23], [24], [25]. Alternatively, one can use the state enumeration method to verify all system states $\mathbf{x}$ and the respective values of $\varphi(\mathbf{x}, t)$. Summing the probabilities of all successful states leads to the system reliability. Notice, however, that the total number of states N grows exponentially with the number of components, $N = 2^{||\mathbf{x}||}$.

RBDs and FTAs are straightforward to understand and construct. They are powerful techniques to analyze and compare systems in terms of their reliabilities or availabilities. They can be employed as long as the events that cause a system failure are known beforehand and the system complexity is small. They do not take dependencies into account, such as repairable components, common-cause failures [26], or components with multiple states. Note, again, that obtaining a symbolic reliability expression requires knowing all simple paths between the source and target nodes. This task is, in itself, a known problem in graph theory and can quickly become impractical for large and densely connected graphs [27]. Dynamic RBDs and FTAs [28], [29] have been developed to address these limitations by extending the RBD and FTA techniques. They derive a Markov model from the fault tree of a system, adding information about the component recoveries. Markov-based approaches suffer from a similar limitation as they can easily become intractable, since their complexity grows exponentially with the number of components if no state reductions are possible [26].

Data-driven methods utilize data to derive a fault-tree or reliability-block representation of a given system. They can be classified in two ways, static and temporal methods. In the latter, system data are obtained in real time and used to continuously update and optimize the system representation so that it better reflects the system current state. They have been employed in the analysis of a production plant to provide decision support for maintenance optimization [30], [31] and for the analysis of an integrated energy system [32]. Static data-driven methods use qualitative or quantitative historical data of a system to create reliability models and indexes to support decision making. Examples include the analysis of a manufacturing line [33] and the reliability analysis of a commercial aircraft [34].

Physical model-based methods implement detailed simulations of the electrical system to evaluate its state. They can be subdivided into two categories: deterministic and probabilistic. Deterministic methods determine the power-system state either with an electromagnetic transient (EMT) simulation or with a power-flow analysis. EMT simulations calculate the system state over time, accurately considering transients and involving a comparatively large number of time steps. Power-flow analysis, on the other hand, uses a static phasor-based representation of the power system to calculate line loadings and bus voltages. Power-flow analyses include methods, such as the optimum power flow (OPF) or probabilistic power-flow (PPF). OPF is an optimization problem that considers constraints and costs, such as voltage limits, power limits, and plant operation costs. Examples of OPF-based analyses for the reliability of power systems can be found in [35], [36], [37], [38]. PPF analyses lie in the intersection between deterministic and probabilistic methods. They are employed to estimate the statistical distributions of the line power flows and of the busbar voltages given a set of load and generation distributions. With that information, it is possible to determine how likely the power and voltage constraints will be satisfied and how likely the power system will be able to supply its loads. Examples of PPF-based analyses are found in [39], [40], [41]. An example of hybrid integrating both data-driven- and physical model-based methods can be found in [42].

Due to the limitations of analytical methods and other issues, such as the unavailability of historical data or the comparatively high computational complexity of detailed EMT simulations, researchers often resort to using stochastic simulation methods or power-flow-based methods to support the reliability analysis of power systems in general. The latter, as mentioned in the previous paragraph, is an established technique for the analysis of electric power networks up to a large scale. However, the application range of the proposed method is for smaller-scale decentralized power systems with comparatively shorter power lines. The considered short lines are assumed to not impose a voltage constraint, meaning that only the maximum rated power is relevant in this scenario. This rated power relates to the maximum current-carrying capacity. Consequentially, a capacitated flow network can be used instead of a power-flow-based electrical model to express the power-transfer capability of the analyzed DPS. In addition, the purpose of the flow-network-based method is to facilitate the comparison of reliability between DPS topologies during the system design phase, when line parameters are generally not known with certainty yet.

As for stochastic-simulation-based methods, examples include the flow-connectivity analyses, Petri nets, Markov chains, and Monte-Carlo simulations (MCSs) [43], [44], [45]. Among those, MCS are a popular choice thanks to their flexibility [46]. Two types of MCS exist: nonsequential and sequential. Nonsequential techniques include sampling techniques, such as state and transition sampling. State sampling methods randomly generate a number of system states, check whether they lead to system failure or success and calculate their overall probability. Differently, state transition sampling techniques focus on system transitions, with the limitation that all components must be modeled with constant failure rates [47].

SMCS, on the other hand, generate a fictitious history of a given system based on the simulated states of its components over time. As such, different failure distributions can be used to model individual components, configuring an advantage compared to the technique of state-transition sampling. It must be noted that sequential methods are more demanding in terms of memory and computing time compared to sampling methods. They can, however, provide a variety of reliability indices and they are flexible enough to model systems with unique operation characteristics and dependencies. Examples of SMCS applied in a context of power system reliability include: analysis of emergency and standby power systems, such as UPSs [48], [49], microgrids with photovoltaic and wind generators [50], analyses of power plant layouts and power distribution systems for data centers [51], [52], or the reliability analysis of composite power systems with time-varying loads [53].

The aforementioned papers use different approaches to determine if a critical system failure has occurred. Moazzami et al. [51] calculated the power flowing through two different busbar layouts via an ac optimal-power-flow algorithm, which requires parameters, such as bus voltages, generator impedances, transformer types, and line impedances. Gang et al. [49] employed a fault tree of the analyzed system, which has to be obtained beforehand to verify the system state during simulation. Ghahderijani et al. [50] checked if there

is enough power from sources to power the load by simply summing the available power from sources and comparing it to the power demand, ignoring any components that might be placed between them.

As observed, the methods introduced before rely on presetting the combinations of events that cause a critical system failure. Methods, such as optimal and nonoptimal power flow, require knowing the electrical parameters of every component for the calculation of power flow, which might not be available in the system design phase. Therefore, the set of success rules and the electrical parameters must be adapted, changed, or entirely rewritten for every change in the system topology. This can be avoided if a framework is used to represent the decentralized power system that considers a set of connected devices transferring power. Graph theory is a promising framework for that purpose.

A. Graph Theory for Modeling Power Systems

Thanks to the versatility and solid mathematical foundations of graph theory, it has been employed to model and analyze a variety of systems. Examples include a vulnerability analysis of the North American power grid [54], an evaluation of the robustness of the European gas network [55], and the cost optimization of the spatial distribution of electric vehicles charging stations [56]. It has been a central concept in the analysis of structural resilience or vulnerability of critical infrastructures [57], [58], such as gas, power, and communication networks.

There are essentially two approaches for analyzing the reliability of networks with graphs: simple connectivity and flow connectivity analyses. The former deems a system to be successful if there exists at least a path between two nodes, similar to the RBD and FTA approaches. The edges in simple connectivity analysis have capacities equal to one and, hence, do not provide a realistic representation of common systems, given that, for example, pipelines or power transmission lines have a maximum rated power or gas transmission capacity.

The flow connectivity approach, on the other hand, uses flow networks to model such systems. These networks have nonunitary edge capacities and can be used to describe capacity-constrained networks, such as gas, water, power, communication, or traffic networks. In these systems, the commodities flowing through the elements or through the connections between them are limited by physical constraints. System success is determined by verifying if the flow consumers are being supplied with their demands. Examples of flow networks used for power system reliability or vulnerability analysis are found in [59], [60]. Flow networks provide a more realistic description of commonly found systems and should therefore be further investigated in the context of network reliability.

B. Integration of Graph-Based and MCS Approaches for Reliability Analysis

Given the advantages of both graph theory and MCS, a natural next step would be to unite them, creating a general framework for the reliability analysis of flow-constrained systems. So far, such an integration has been barely followed upon.

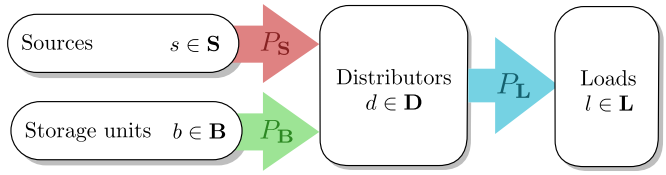

Fig. 1. Illustrative system $\mathcal{S}$, to serve as an abstraction of an electrical system.

Ferrario et al. [61] proposed hierarchical graphs as a way to model the dependencies between critical infrastructure systems, placing the inputs, transmission, and the demand nodes within different layers. To specify the capacity of the arcs and assess the robustness of the critical infrastructure, the work employs a sampling MCS method, meaning that time-dependent indices, such as availability and mean time to repair (TTR), cannot be evaluated with this approach. Energy storage units are not modeled in the article and the method can only be used to analyze radial networks with unidirectional flows, making it unfit to analyze modern systems, for example, those with decentralized generators, energy-storage units, and loads.

Similar ideas have been applied to gas networks, with Praks et al. proposing a flow-network-based method to analyze their availability using an SMCS [62]. Maximum-flow (MF) algorithms are used to calculate the gas flows in the network, giving priority to the demand nodes closer to the supply nodes. This article does not implement repairable pipelines and the storage units are treated as infinite-flow sources. It is, therefore, not applicable to the purpose of this article.

The method proposed in this work addresses the limitations of the aforementioned approaches, and therefore it goes beyond the state of the art and is valuable in the system design phase. It is able to model decentralized power systems with capacity-limited energy-storage units, meshed power networks, and cascading failures. It can be employed to obtain indices, such as system availability and mean time to system repair.

III. Modeling of Decentralized Power Systems as Flow Networks

Decentralized power systems are a collection of essentially four different types of components: sources, loads, energy storage units, and distributors. Sources generate and loads consume electrical power. Distributors are components that neither produce nor consume power, acting simply as power transfer or conversion devices, such as power-electronic converters, transformers, or cables. Energy storage units can function either as a source or as a load.

In order to show the need for a more detailed framework to model decentralized power systems for reliability analyses, consider an hypothetical electrical system $\mathcal{S}$ with a set of sources $\mathbf{S}$, a set of loads $\mathbf{L}$, a set of energy storage units $\mathbf{B}$, and a set of distributors $\mathbf{D}$. Here, the connections between the distributors are not analyzed. As an abstraction, this system is shown in in Fig. 1. The symbols P_s, P_b, and P_l represent, respectively, the power supplied by all sources, the power supplied by all storage units, and the power delivered to the loads.

Given that $\mathcal{S}$ is a closed system, the sum of the power provided by the sources and the storage units is equal to the power consumed by the loads, $P_{\mathbf{L}} = P_{\mathbf{B}} + P_{\mathbf{S}}$. This system is successful as long as $P_{\mathbf{L}}$ is larger than the minimum power required by the loads. The power produced by each source $s \in \mathbf{S}$ and by each storage unit $b \in \mathbf{B}$ are bound by their respective nominal power ratings. Sources are unidirectional, hence their power p_s must satisfy $0 \leq p_s(t) \leq c_s$, where c_s is the power rating of the source s. Energy storage units are bidirectional. Their injected or withdrawn power must satisfy $c_b^- \leq p_b(t) \leq c_b^+$, where c_b^+ is its maximum charging power and c_b^- its maximum discharging power. The battery power $p_b(t)$ takes a negative value when the unit b is withdrawing power, i.e., being charged.

Assume that sources and storage units have two states: ON, 1; or OFF, 0. Neglecting the structure formed by the distributors in $\mathbf{D}$, the success of the system depends solely on (1) being true

$$\mathbf{c}_s \mathbf{h}_s^T + \mathbf{c}_b \mathbf{h}_b^T = \mathbf{d}_l \mathbf{h}_l^T \tag{1}$$

where $\mathbf{c}_s$ and $\mathbf{c}_b$ are, respectively, vectors of the power production capacities of the sources and batteries, and $\mathbf{h}_s$ and $\mathbf{h}_b$ are vectors of the status of the sources and batteries, respectively. The vectors $\mathbf{d}_l$ and $\mathbf{h}_l$ represent, respectively, the power demands and states of the loads. This expression simply means that the power production capacity in the system must be equal to the power demands of the active loads. Ideally, the sources can supply the loads and share the total power demand of the loads equally among them. Storage units only need to be active if the power required by the loads is larger than the current power-production capacity of the sources, $\mathbf{c}_s \mathbf{h}_s^T$.

Note, again, that in (1), neither the connections between the distributors nor their statuses are taken into account. The storage capacity of the storage units are not considered either, as they are just treated as a regular source. In short, this expression assumes that the structure of the distributor nodes and their statuses always allow the power from the sources and from the batteries to be transferred to the loads.

In a real setting, on the contrary, sources, loads, and storage units are all interconnected via a multitude of cables, transformers, switches, power converters, and busbars, each with a specific power rating and with a failure probability. Together, these components and their connections constitute the power transmission or power distribution network, a relevant part of any power system. This network, its structure, its energy storage elements, and the failure probabilities of all its components must be accounted for in the determination of the reliability of the power system topology. That can be achieved by using flow networks to model the abstraction of Fig. 1.

A. Decentralized Power Systems as Flow-Network Graphs

Being a collection of electrical components and their connections, the topology of decentralized power systems can be represented as directed graphs, with their nodes as the physical components and their edges as the existing physical interconnections. A directed graph is able to depict them accurately, yet it does not provide information about the physical constraints of their connections, such as their maximum power ratings or energy storage capacities. To address this issue, weights are assigned to each edge to define the maximum power that can flow through them. These will be referred to as edge capacities hereafter. In that sense, the power system is now modeled by a construction called flow-network graph. To model storage capacities, variables, such as state of charge, nominal and current capacity can be assigned to each of the graph nodes that represent energy-storage units.

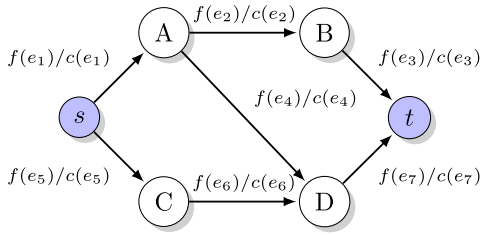

Fig. 2. Example of a flow-network graph. The expressions in the format $f(e_k)/c(e_k)$ next to the edges represent the flow f through the edge e_k, whose capacity is c.

1) Definitions and Constraints: Let a weighted directed graph $\mathcal{G} = (V, E)$ represent an electrical system with a set of nodes V and a set of edges E. Edges are defined as pairs of nodes (u, v) and each edge e has a given capacity $c(e)$. The set of nodes V is a superset containing all sources S, sinks (electric loads) L, storage units B and distributors nodes D. In a directed graph, each edge $e = (u, v)$ belonging to E has a direction, meaning that the pairs (u, v) only represent the arc leaving the node u and entering the node v.

To simplify the description of the graph constraints, two terms are hereby introduced: the total flow entering and leaving a node v. The total flow entering the node v, also called total inbound flow, is given by

$$F_i(v) = \sum_{e \in E_i(v)} f(e) \tag{2}$$

where $E_i(v)$ is the set of ingoing edges of the node v and $f(e)$ the flow through the edge e. The total flow leaving the node v is given by

$$F_o(v) = \sum_{e \in E_o(v)} f(e) \tag{3}$$

where $E_o(v)$ is the set of outgoing edges of the node v. As mentioned previously, the flows are constrained by the edge capacities. Therefore, both $F_i(v)$ and $F_o(v)$ have upper bounds. An example of a flow network, with the edge flows and capacities, is shown in Fig. 2.

The power flow through the edges of $\mathcal{G}$ must satisfy the constraints **C1**–**C5**, hereby referred to as flow-graph constraints.

C1. *Edge capacity constraint:* The power flow f through an edge e is positive and upper bounded by the edge capacity c

$$0 \leq f(e) \leq c(e) \quad \forall e \in E. \tag{4}$$

C2. *Nodal power flow conservation:* The total power flow entering a node v must be equal to the total power flow

leaving v. This constraint applies only to distributor nodes. Losses are not considered here

$$F_i(v) = F_o(v) \quad \forall v \in D. \tag{5}$$

C3. *Source power production constraint:* The sum of power flows leaving the sources cannot surpass their nominal power rating

$$F_o(v) \leq c(v) \quad \forall v \in S. \tag{6}$$

C4. *Load power demand constraint:* The total power flow entering the load nodes must not exceed their power demands

$$F_i(v) \leq c(v) \quad \forall v \in L. \tag{7}$$

C5. *Energy-storage unit constraints:*

a) If the storage unit v is fully charged, no power can flow into the unit

$$F_i(v) = 0 \quad \forall v \in B, \text{if SOC}(v) = 1. \tag{8}$$

b) Similarly, the storage unit cannot provide any power if it is fully discharged

$$F_o(v) = 0 \quad \forall v \in B, \text{if SOC}(v) = 0. \tag{9}$$

c) The total power flow leaving the unit must not surpass its maximum discharging power $c_-(v)$

$$F_o(v) \leq c_-(v) \quad \forall v \in B. \tag{10}$$

d) The power flows entering the storage unit must not surpass its maximum charging power $c_+(v)$

$$F_i(v) \leq c_+(v) \quad \forall v \in B. \tag{11}$$

e) Storage units can function either as a source (discharging) or as a load (charging). Thus, for any storage unit v, only the inbound or only the outbound edges may transfer power at any given point in time

$$(F_o(v) > 0) \oplus (F_i(v) > 0) \quad \forall v \in B. \tag{12}$$

2) On the Successful Operation of a Decentralized Power System Modeled as a Flow Network: For a decentralized power system to be successful, it must be able to supply the power demanded by the loads. Power is transferred over a variety of electrical devices and, to calculate the power magnitudes, power flow equations are often used to determine the power flowing through all devices by obtaining the complex voltages and currents across the system. These calculations require the knowledge of a multitude of parameters as a prerequisite, such as impedances of sources, lines, and transformers; the active and reactive powers of PQ-buses; and the amplitude and phase angle of the slack buses.

For the application presented in this article, in which power system are modeled as flow network graphs, knowing all the complex bus voltages and line currents is irrelevant, since only the power that reaches the loads is actually relevant for the evaluation of system reliability. This is analogous to the formulation of system success of (1). By using the graph representation with four different types of components, as shown previously, it is possible to determine if a power system is functional by mapping it to a problem known as the flow-circulation problem from graph theory. This problem deals with graph nodes that have either supplies or demands, δ. A solution consists in finding the flows $f(e)$ that satisfy the constraint $F_i(v) - F_o(v) = \delta(v)$ for all nodes, depending on their type. If nodes are power sources, then $\delta(v) < 0$. If nodes are loads, $\delta(v) > 0$. If nodes are distributors, $\delta(v) = 0$ [63].

Some types of power sources are not inherently controllable, for example, solar- and wind-power sources. Their production capacity depends on external factors, such as wind speed or solar irradiance. The same applies to loads, which vary over time in most power systems. However, in a flow-network framework, the power of all sources will be adjusted to match the power required by the loads at a given point in time, hence this framework is better suited to analyze the reliability of systems with controllable sources. To model that behavior, the constraint in the previous paragraph has to be changed to $F_i(v) = \delta(v), \forall v \in L$, that is, the power demands of the loads have to be met, but the power sources no longer have to deliver their rated power. This problem can be reduced to an MF problem. By doing so, only prior knowledge of the power system structure—such as node power demands, edge capacities, and the connections between nodes—is necessary to evaluate if the system is successful.

The solution to graph flow problems is well established [64], with one of the first algorithms developed in 1963, called the Ford–Fulkerson algorithm [65], to calculate the MF in a graph. Other max-flow algorithms have been developed since 1963, for example, the Edmonds–Karp or Dinic's algorithms. The objective of an MF algorithm is, as the name suggests, to find the MF between two nodes in a graph

$$\max \sum_{e \in E} f(e) \tag{13}$$

while satisfying the aforementioned flow network constraints **C1**–**C5**.

To calculate the power flow with the MF algorithm a few adaptations to the power-structure graph are to be made. First of all, the algorithm requires a fixed pair of nodes to represent the flow source and flow target. Since decentralized power systems normally contain multiple sources and multiple loads, two fictive nodes are added to the graph: the supersource and supersink. The supersource is connected to all source nodes, while the supersink receives a connection from all load nodes found in the graph. Storage units, due to their bidirectional nature, receive an edge from the supersource and an edge to the supersink. The addition of the supernodes effectively turns sources, storage units, and loads into distributor nodes, which are subject to the equality constraint (5).

Executing the MF algorithm, with $S_{\mathcal{G}}$ and $L_{\mathcal{G}}$, respectively, as the source and target nodes, yields a MF solution where a flow $f(e)$ is assigned to every edge e. With the calculated power flows it is possible to verify if the loads are being properly supplied. The following equation yields the ratio between the power delivered to the load node v and its power demand $\delta(v)$.

$$\gamma(v) = \frac{F_i(v)}{\delta(v)}, \quad v \in L. \tag{14}$$

A load node $v \in L$ is operational if it receives enough power. This condition can be expressed mathematically as follows,

$$\sigma(v) = \begin{cases} 0, & \text{if } \gamma(v) < p \\ 1, & \text{if otherwise} \end{cases} \tag{15}$$

where p is a threshold value for the minimum power that has to be supplied to the load node. If $p = 1$ then $F_i(v) \geq \delta(v)$, and it is assumed that the load node v is operational.

It must be noted that, because of the way the MF is calculated, the power demand is not proportionally shared with all the sources. For example, consider a system with N sources, each capable of supplying a power P. If the total power demand of the loads is P, then only one source will deliver that power. This effect can be seen clearly in the minimum-cost maximum flow (MCF) solution shown in Fig. 12, where the generator G_3 does not produce any power. For systems with only sources and no energy storage this limitation is not an issue, since, if a source fails, another will take over and supply the needed power. Since the intention is to model systems with energy storage elements though, given that their state of charge depends on the power balance of the system in a given time interval, this effect might cause their state of charge to change over a given time interval, even in situations where the sources alone could provide enough power to the loads. Hence, the power flow from the sources must be prioritized, which can be achieved by adding flow costs to each edge, mapping this case to a MCF problem.

3) Solving the Problem of Power Priority With the MCF: The MCF problem is an extension of the MF problem. It takes into account not only the capacities of each edge, but also the costs of transporting flow through them. The objective is to maximize the power flow while minimizing the total cost, as in

$$\min \sum_{e \in E} c(e)\, f(e) \tag{16}$$

where $c(e)$ and $f(e)$ are the capacity and flow of an edge e, making it essentially an optimization problem. Still, the problem of prioritizing taking power from sources remains. Which costs should the edges take to force loads to take power from storage units only in case it is needed? A solution consists in assigning higher costs to the edges belonging to the storage units, as shown in Fig. 3.

The costs w of all edges belonging to the sources, distributors, and loads are set to null. The cost of all edges connecting the storage units to the distributor network and vice-versa are set to one. The edges from the supersource to the storage units and those from them to the supersink are both assigned a cost of two. By doing so, it is possible to prioritize the power drawn from sources. In addition, one can determine if the storage units will function as actual sources or as loads, based on the flows flowing between the storage units, the supersource, and the supersink, as made clear by (19). Their operation mode depend on the power balance between sources and loads.

In order to understand the choice of edge costs, it is illustrative to consider a system consisting of one 10-kW source, one 10-kW battery, one bus, and one 10-kW load. The battery capacity is not relevant in this explanation. The system is shown in Fig. 4

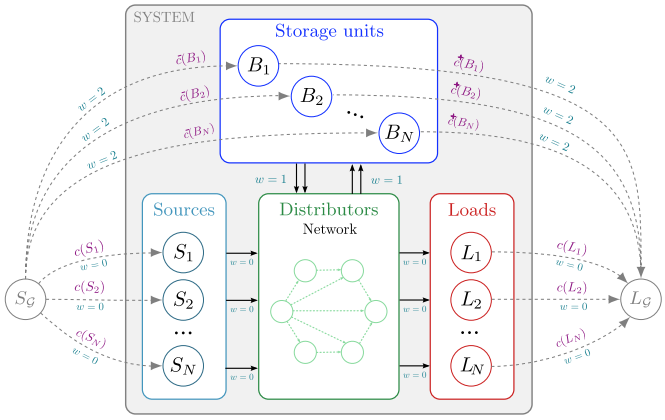

Fig. 3. Summary of working principle to enable the MCF algorithm.

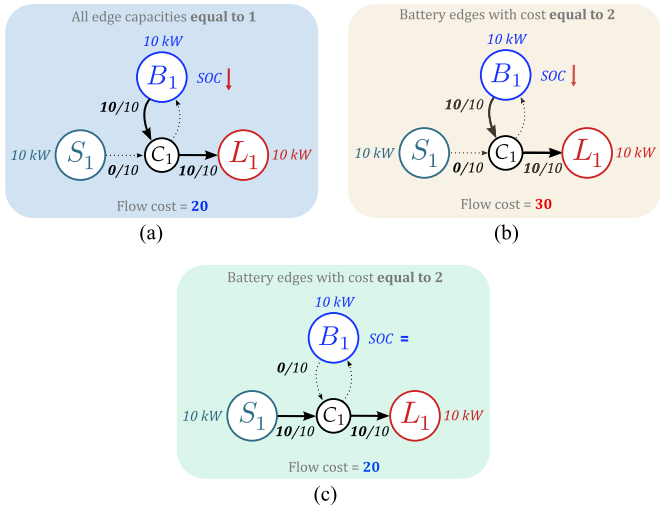

Fig. 4. System with its flow solutions for three cases to show how to prioritize sources by setting the edge costs.

in three different scenarios. In A, the system is shown with its MCF solution when all edges have a cost equal to one. In that case, it is not possible to guarantee that the source will supply the load, even though it can supply the load alone. By setting the costs of only the edges belonging to the battery equal to 2, and using the same flow solution from A, the total flow cost is equal to 30. Given that the MCF algorithm searches for the solution with the smallest cost, it will reach the flow solution shown in C.

4) Modeling Individual Components and Their Behavior in a Flow Graph: Modeling how components affect the power system is done by defining their behavior in the event of failure or repair. By assigning each component a time-to-failure (TTF) and a TTR probability distribution, it is possible to model how often they will affect the power system structure.

Sources, distributors, and loads can be in one of two states: functional (ON) or in repair (OFF)

$$\alpha(v) = \begin{cases} 0, & \text{off, failed or in repair} \\ 1, & \text{on, operational.} \end{cases}, \quad v \notin B \tag{17}$$

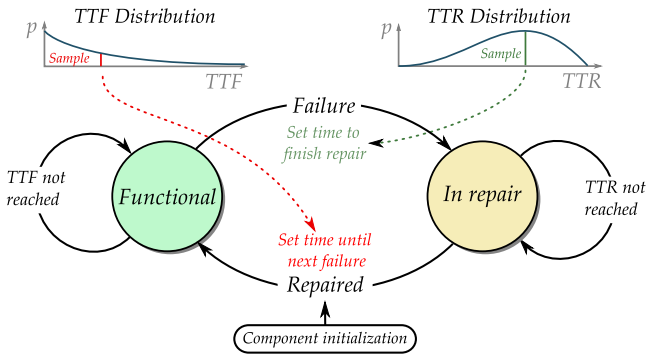

Fig. 5. State diagram of individual sources, distributors, and loads.

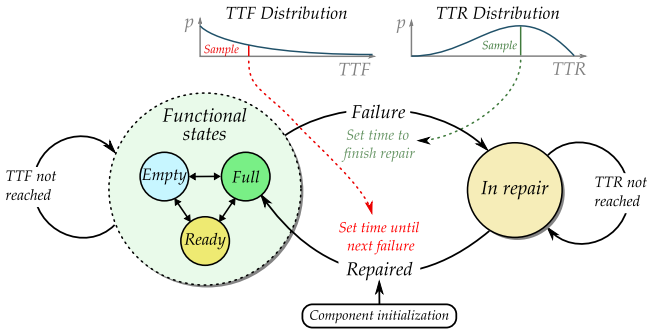

Fig. 6. State diagram of an energy storage component.

The transition between these states depends on the samples taken from their respective TTF and TTR distributions. Common distributions used to model time between failures are the exponential and Weibull distributions. A diagram showing the failure/repair model of a component is found in Fig. 5.

Energy storage units can be, in principle, in two states as well: either ON or OFF. However, if they are not in a failed state, they can also be either fully charged, fully discharged, or "ready"—here used for the state when the storage unit is neither full nor empty. Energy storage units are therefore modeled by four states

$$\alpha(v) = \begin{cases} 0, & \text{OFF, failed or in repair} \\ 1, & \text{ON, ready} \\ 2, & \text{ON, fully charged} \\ 3, & \text{ON, fully discharged.} \end{cases} \qquad v \in B \tag{18}$$

The transition between the ON and OFF states depends on the samples taken from the unit failure and repair distributions. Transitioning between any of the "ON" states depends on the state of charge of each unit. Fig. 6 shows the failure/repair model of an energy storage unit.

In the event of failure of a component v, v can no longer provide, draw, or transfer power. Modeling this behavior in the flow graph is done by setting the capacities of all its edges to 0: $c(e) = 0, \forall\, e \in \{E_i(v) + E_o(v)\}$. Reactivating it, once it has been repaired, can be done by setting the capacities of all its edges to their nominal values c_0: $c(e) = c_0(e), \forall\, e \in \{E_i(v) + E_o(v)\}$. Storage units have additional states, and their operation in a flow network must be modeled accordingly. If storage units are fully charged, the capacities of its ingoing edges are set to 0: $c(e) = 0, \forall\, e \in E_i(v), v \in B$. If they are, on the other hand, fully discharged, the capacities of its outgoing edges are set to 0, $c(e) = 0, \forall\, e \in E_o(v), v \in B$. If they are not in a failed state and neither full nor empty, the capacities of their edges are set back to their nominal values. If it has failed, it can be deactivated like any regular component, as described previously.

The state of charge of storage units depends on the power being drawn from or injected into the unit in the simulation time step, as well as on its nominal energy storage capacity. Given that storage units are connected to both supernodes $S_\mathcal{G}$ and $L_\mathcal{G}$, once the flow solution is obtained, the power balance p of an energy storage unit b is obtained by

$$p(b,t) = f(\{b, L_\mathcal{G}\}, t) - f(\{S_\mathcal{G}, b\}, t) \tag{19}$$

where $f(\{a,b\},t)$ is the flow from node a to node b at the time t. The power balance $p(b,t)$ is negative if b is functioning as a source, positive if it is functioning as a load and, if no power is being drawn or injected, $p(b,t) = 0$. Thus, if the storage unit b has a nominal energy storage capacity $k(b)$, its state of charge is given by

$$\mathrm{SOC}(b, t + \mathrm{d}t) = \mathrm{SOC}(b,t) + \frac{p(b,t)}{k(b)}\mathrm{d}t \tag{20}$$

with the additional constraint $0 \leq \mathrm{SOC}(b,t) \leq 1$.

B. Sequential MCS

Let $\Sigma(\mathcal{G})$ be the structure function of a power system $\mathcal{G}(V,E)$ represented as a graph. The function $\Sigma(\mathcal{G})$ should return 1 if $\mathcal{G}$ is successful and 0 if otherwise. System success is, therefore, given by boolean functions that take the graph edges and nodes and their respective data as inputs, as expressed in

$$\Sigma(\mathcal{G}) = \prod_j f_j(V,E). \tag{21}$$

As an example, assume that the system success depends on all loads being supplied with enough power. From (15)

$$f_1(V,E) = \prod_{v \in L} \sigma(v). \tag{22}$$

This condition alone might not address all possible events that lead to system failure. Assume, for example, that the system in question requires at least one functional storage unit as an energy buffer. This rule can be easily modeled by a second boolean function f_2. Based on (18)

$$f_2(V,E) = \sum_{v \in B} \left(\alpha(v) \notin \{0,3\} \right). \tag{23}$$

The system success function $\Sigma(\mathcal{G}(V,E))$ can be formulated to model systems of different requirements. Once it has been defined, the system can be simulated at discrete time intervals. Fig. 7 shows, in a simplified graphical manner, how the system is simulated.

In Fig. 7, the system remains active if all components are operational. Since the loads demand 10 kW of power, one source can supply them alone. If the two sources fail, the system remains active for as long as the storage unit can supply power, given its

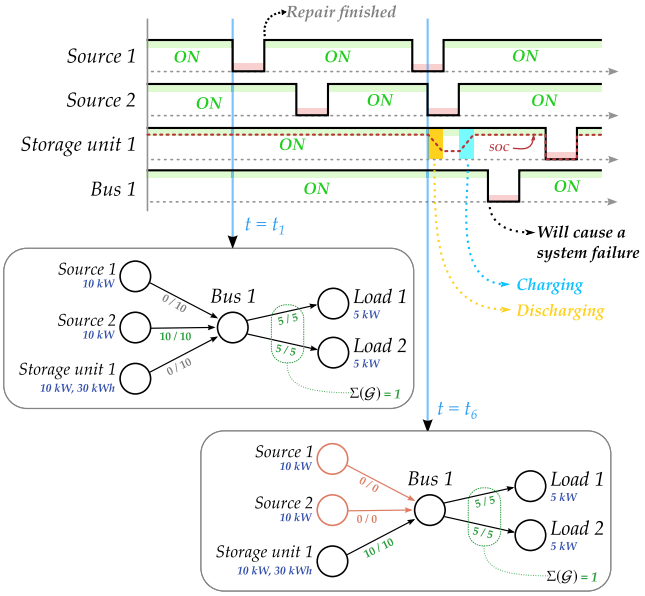

Fig. 7. Simple example of the SMCS method.

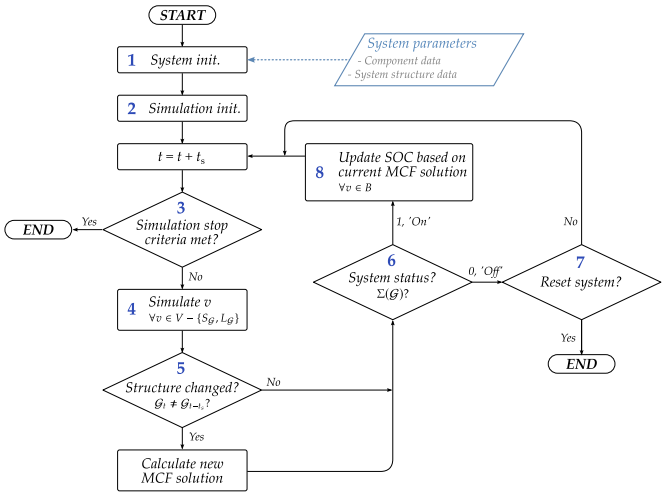

Fig. 8. Simulation flow chart.

constraints for energy-storage capacity. In the example of the figure, the storage unit was able to supply the power while the two sources were in a failed state. The system would fail if the two sources stayed in a failed state for a longer time.

Fig. 8 shows a flow chart containing the simulation procedures and decisions. The subroutines (SRs) are detailed hereafter.

SR 1. *System initialization*: In this step, the component (power ratings, TTF and TTR distributions, storage capacities, etc.) and system structure data (edges and their capacities) are loaded. A directed-graph representation of the system is built with the data. The supersource and supersink are added and the edge costs set—as shown in Fig. 3.

SR 2. *Simulation initialization*: The simulation time step t_s is set. All components are initialized by assigning each component a time to next failure (TTF), where each TTF is a sample taken from the TTF distribution of the respective component. The first MCF solution is calculated in this step.

SR 3. *Simulation stop criteria*: The stop criteria of the simulation are checked after every time step. The simulation ends if any of the predefined stop criteria are met. Examples of possible criteria include: a simulation time limit, a specific number of system critical failures, maximum computing time, etc.

SR 4. *Component-level simulation*: Every component, except for the supersource and supersink, are simulated according to their individual failure and repair behavior—as shown in Figs. 5 and 6. For example, a component will fail once the elapsed time since the TTF sample has been taken equals the TTF sample. The edge capacities are set based on the state of the component, as explained in Section III-A4.

SR 5. *Changes in the system structure*: In this step, changes in the system structure are detected. Any change in the edge capacities characterizes a change in the system structure. For example, the failure of a component is modeled by setting the capacities of its edges to 0, therefore causing a change in the system structure.

SR 6. *Definition of system status*: The success or failure of the system is verified in this step, based on the predetermined conditions, as expressed in (21).

SR 7. *In the event of system failure*: There are two options once the system has suffered a critical failure, either reset the system or allow the simulation to continue.

- Resetting the system means that the simulation ends once the system has suffered a critical failure. The components are restored to their original states and the simulation is restarted after a critical failure. In this case, only the mean TTF of the system can be obtained. Because the system is reset after a critical system failure, the system availability cannot be calculated.
- Allowing the simulation to continue means that, even if the system has suffered a critical failure, the simulation continues. As soon as component repairs allow the system to supply the loads, the system state is changed back to functional. The simulation runs until the stop criteria in SR 3 are met. Unlike when the system is reset, it is possible to obtain the system MTBF and its availability.

SR 8. *Update of the state of charge of storage units:* In this step, the state of charge of each functional storage unit is updated using (20), with $\mathrm{d}t$ representing the simulation time step t_s.

IV. Implementation

The proposed simulation method was implemented using the Python programming language. The package NetworkX [66] was used for the representation and manipulation of power-system graphs, as well as for the calculation of the flow solution. This package is especially suitable for the proposed method, given that it allows many different Python objects (numbers,

dictionaries, class instances, etc.) to be used as the node or edge data, making it possible to store all necessary information within the objects representing the graph nodes or edges.

The Pandas module [67] was used to load the node and edge data, stored in text format as tables. NumPy was used to draw samples from failure and repair distributions. The component models, as shown in the Figs. 5 and 6, were implemented as Python objects with their respective methods, such as the ones used to check for failure or completed repairs, or to update the state of charge of energy storage components. These component-model objects store the failure and repair timestamps of their respective components.

The system, which contains all the components and their structure, was modeled as a Python object. The structure functions that dictate system failure are stored as class methods, and are used to check if a critical system failure has occurred, based on the system state at each time step. System-level results, for example, the timestamps of critical failures, timestamps of system repairs, and the system availability, are stored in the system object. These data are exported as a table for analysis, once the simulation stop criteria are met.

V. Simulation Results

In what follows, the results obtained via the proposed method for three different scenarios are presented. The first two scenarios are relatively simple and may also be studied analytically. Both serve as a proof of concept for the proposed simulation method. The third scenario, on the other hand, involves the simulation of a rather comprehensive system with more dependencies to demonstrate the method's practicality.

A. Three Nonrepairable Sources Connected in Parallel

In this scenario, three identical 10-kW sources are connected in parallel to supply a 10-kW load. The failure rate of the sources was chosen arbitrarily. Their times to failure are modeled by exponential distributions with a failure rate of $\lambda = 0.10536$ failures per year, equivalent to saying that the sources have a 90 % reliability in one year and an MTBF of 9.491 years. It is assumed that the sources cannot be repaired in this scenario. Once the system has failed, it is reset and simulated again. The chosen simulation time step was one hour and the stop criteria was arbitrarily set to 500 simulated critical system failures.

The MTBF value obtained through simulation of 500 critical failures, shown in Fig. 9(a), matches the actually expected value for an arrangement of n identical parallel-connected components whose time intervals between failures are modeled by an exponential distribution. The MTBF of parallel-connected components modeled by an exponential distribution is given by $\text{MTBF}_e = H_n\lambda^{-1}$, where H_n is the nth harmonic number and n the number of identical parallel-connected components. For this scenario, a MTBF value of 17.4005 years was predicted, a difference of 39 h compared to the simulated value of 17.4050 years. It can be observed as well that the CDF of the simulated intervals between failures matches the analytical reliability curve, $R(t) = 1 - (1 - e^{-\lambda t})^3$, as displayed in Fig. 9(b).

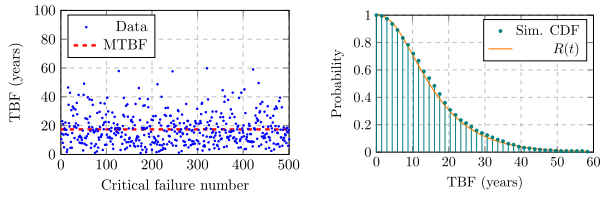

Fig. 9. Simulated times between failures, MTBF, and the theoretical reliability curve compared to the obtained cumulative distribution of TBF. (a) Blue dots: simulated times between system failures. Dashed red line: mean value, MTBF = 17.4050 years. (b) Comb plot: cumulative distribution calculated from the simulated system TBF values. Solid line: analytical reliability curve $R(t) = 1 - (1 - e^{-\lambda t})^3$.

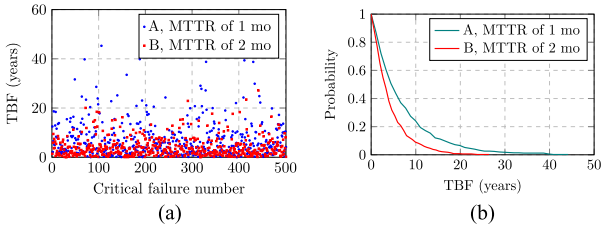

Fig. 10. Simulated times between failures of both scenarios and the reliability curves. (a) Round blue dots: simulated TBF of scenario A. Square red dots: simulated TBF of Scenario B. (b) Reliability curves for both scenarios, based on the cumulative distributions of their simulated TBF values.

B. Two Repairable Sources Connected in Parallel

Two identical 1-kW sources are connected in parallel to supply a 1-kW load. The time intervals between failures of the sources are modeled by an exponential distribution with a failure rate of $\lambda = 1$ failure per year, equivalent to saying that the sources have a reliability of 36.79% in an year. This value was, once again, chosen arbitrarily for illustration purposes.

Once the system has failed, it is reset and simulated again. The simulation time step was set to 1 h, and 500 critical system failures were simulated. Two scenarios were simulated: *A*, sources are repaired within 720 h (MTTR = 1 month) and *B*, sources are repaired within 1440 h (MTTR = 2 months).

The simulation yields an MTBF of 4.22 years for scenario *B* (sources with an MTTR of 2 months) and an MTBF of 6.96 years for scenario *A* (sources with an MTTR of 1 month).

Using (24) for the MTBF of a k-out-of-n system with repairable components, as derived in [68]

$$\text{MTBF}_{k,n} = \frac{\sum_{i=0}^{n-k} \binom{n}{i} \left(\frac{\lambda}{\mu}\right)^i}{k\lambda \binom{n}{k} \left(\frac{\lambda}{\mu}\right)^{n-k}} \tag{24}$$

yields the following MTBF values for the two scenarios: A, 7.12 years when sources are repaired within 1 mo and B, 4.04 years when sources are repaired within 2 months. In (24), λ is the failure rate and μ the repair rate of the identical components. The repair rate is calculated via $\mu = \text{MTTR}^{-1}$. As it can be observed, the values obtained through simulation match the ones

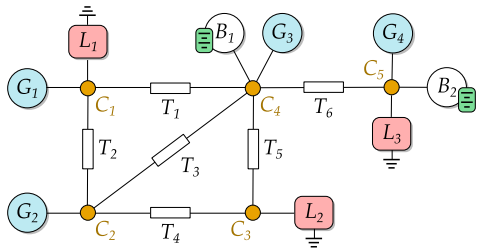

Fig. 11. Example of a decentralized meshed power system.

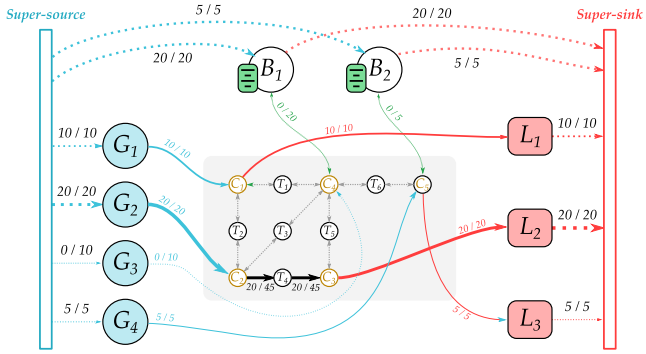

Fig. 12. Graph representation of the system shown in Fig. 11 with the initial MCF solution. The loads are being properly supplied by the sources and the batteries are not needed. The capacities of all arcs between the distributor nodes is set to 45 kW to correspond to the the sum of the maximum power production from the sources.

predicted by the closed-form analytical expression. Note that this expression is only valid for this specific scenario and only yields one reliability index, the MTBF. Besides that, it can only be used if the sources are identical and if their failure behavior are modeled by exponential distributions.

C. Comprehensive System With Repairable Components

In this scenario, an illustrative system without power losses, multiple sources, loads, busbars, energy-storage units, and bidirectional power lines is simulated. This system was designed to serve as an example and to showcase the different types of components this method can integrate and simulate. Typical test systems for reliability analyses, such as the IEEE 34, the IEEE RTS 24, or the CIGRE B4 dc, do not contain energy-storage devices and were therefore not considered for analysis, given that the objective of this article is to demonstrate the usefulness of the proposed algorithm. These test systems could however be adapted to integrate energy-storage units, or they could be simulated directly with this method as well. The system in this scenario can be thought as representative of modern power systems, such as microgrids, with decentralized generators, energy-storage units, and loads that are geographically distributed and connected through meshed power lines.

It must be made clear that the graph nodes representing sources and loads can also represent a collection of them. In order to clarify what that means, assume that the designed system is meant to represent a neighborhood. In this scenario, G_2 could represent the national grid connection, G_1 could represent a residential fuel-cell system supplying its residential load L_1; G_3 and B_1 could represent a household with a V2G-able battery-electric vehicle and a wind power turbine, respectively; and G_4, B_2, and L_3 could, respectively, represent the solar system of a building with a backup diesel generator and its load.

Even such a small-scale system makes the process of obtaining closed analytical solutions impossible. Therefore the results shown here will not be compared with their theoretical values, unlike the previous examples.

The simulated system is shown in Fig. 11.

The system is not reset when a critical failure occurs. It is further simulated until the repairs of its components allow the loads to be supplied again. With the simulated system downtime it is possible to obtain the system availability, calculated via

$$A = \frac{T_{\text{off}}}{T_{\text{off}} + T_{\text{on}}} \tag{25}$$

which is updated after every critical failure. The component parameters, which include their maximum power, failure probabilities, and times to repair, were chosen arbitrarily. These are listed in the following bullet points.

1) G_1: 10-kW generator. Its TTF is modeled by an exponential distribution with an MTBF of three years. Mean TTR of four days.
2) G_2: 20-kW generator. Its TTF is modeled by an exponential distribution with an MTBF of ten years. Mean TTR of six days.
3) G_3: 10-kW generator. Its TTF is modeled by an exponential distribution with an MTBF of five years. Mean TTR of five days.
4) G_4: 5-kW generator. Its TTF is modeled by an exponential distribution with an MTBF of four years. Mean TTR of two days.
5) Loads, L_1: 10 kW; L_2: 20 kW; L_3: 5 kW. Loads are assumed to have no failure behavior, i.e., they cannot fail.
6) B_1: battery, can deliver up to 20 kW and be charged with 20 kW as well. Its nominal energy-storage capacity is 60 kWh. Its failure behavior is modeled by an exponential distribution with an MTBF of seven years. Mean TTR of three days.
7) B_2: battery that can deliver up to 5 kW and store 10 kWh of energy. It can be charged with up to 5 kW. Its failure and repair behavior is the same of the battery B_1.
8) Distribution lines T: MTBF of 20 years and MTTR of one day.
9) Busbars C: nodes assumed to have the same failure and repair behaviors of distribution lines.

Fig. 12 shows how the system can be represented as a graph, together with its initial MCF solution. As it can be noted, all loads are being properly supplied by the sources G_1, G_2, and G_4. As a result, no power is drawn from the batteries, B_1 and B_2, when the system is in its initial state.

The simulation results for this scenario are shown in Fig. 13.

The MTBF values obtained through the simulation for this scenario were 35 072 h in the first run, and 36 591 h for the second run, corresponding to an average of 4.09 years between critical

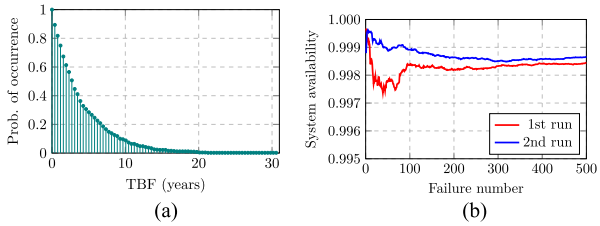

Fig. 13. Cumulative distribution of the simulated TBF values and system availability. (a) Cumulative distribution of times between critical system failures based on TTF values from the first simulation run. This curve is analogous to the reliability curve of the system. (b) Calculated system availability during both simulation runs. The system availability is updated at every critical system failure using (25).

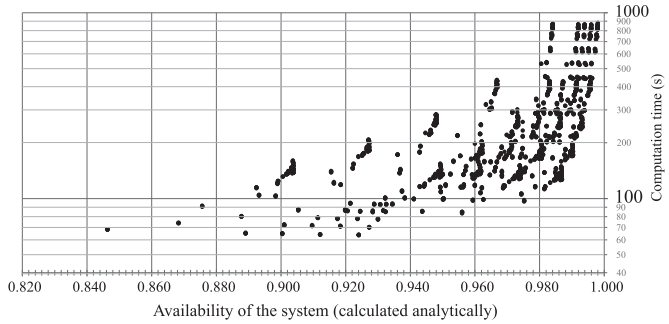

Fig. 14. Sensitivity analysis of the computation time in respect to the analytical system availability.

TABLE I
TABLE CONTAINING SELECTED RESULTS FOR THE FIRST 15 CRITICAL SYSTEM FAILURES, OBTAINED FROM THE SECOND SIMULATION RUN OF THE SYSTEM OF FIG. 12

Failure number	Failure timestamp	Repair timestamp	TTR (h)	TBF (h)	Cumulative uptime (h)	Cumulative downtime (h)	Availability
1	19 845	19869	24	19845	19845	24	0.998792
2	57 194	57218	24	37325	57170	48	0.999161
3	157 206	157230	24	99988	157158	72	0.999542
4	240 479	240503	24	83249	240407	96	0.999601
5	323 946	323970	24	83443	323850	120	0.999630
6	358 111	358135	24	34141	357991	144	0.999598
7	362 702	362726	24	4567	362558	168	0.999537
8	434 514	434538	24	71788	434346	192	0.999558
9	499 086	499110	24	64548	498894	216	0.999567
10	539 879	539903	24	40769	539663	240	0.999555
11	573 852	573876	24	33949	573612	264	0.999540
12	651 499	651523	24	77623	651235	288	0.999558
13	663 472	663496	24	11949	663184	312	0.999530
14	693 363	693387	24	29867	693051	336	0.999515
15	721 597	721741	144	28210	721261	480	0.999335

system failures. In terms of system repair, the first run yielded a mean time of 54 h, and the second run a value of 49 h to repair the system. Although the system reliability curve, displayed in Fig. 13(a), falls relatively quickly, the system availability, shown in Fig. 13(b), converges to a value of around 99.85 % for both simulations runs.

In terms of simulation time, more than 2000 years were simulated to generate 500 critical system failures, corresponding to around 17 million time steps, each of 1 h. The simulation time is only shown indirectly in Fig. 13(b), as the horizontal axis represents the number of critical system failures. The simulation results are, as mentioned before, in tabular form. An example of a table of results yielded by the MCS for this scenario is found in Table I. Each simulation run took around 260 s of computation time to complete, as measured with the Python *time* module. The measurement of the elapsed computation time starts with the SR to load the system data, and ends as soon the predefined stop criteria are met, in this case, the number of simulated critical system failures of 500. The following hardware was used: i7-10610 U CPU, 32 GB of RAM, and no dedicated graphics card. To avoid confusion, the simulation time refers to the time within the simulation, i.e., within the fictitious history of the system generated by the MCS. The computation time refers to the time taken to generate the entire fictitious history of the system with the Monte-Carlo method.

D. Brief Sensitivity Analysis of the Computation Time

The total computation time depends on a variety of factors. For instance, it is influenced by the topology of the simulated system, by the reliabilities and availabilities of its components, by the length of the simulation time step, and, of course, by the computation power of the hardware. Therefore, it is difficult to properly compare the computation times.

To show the impact of only one system parameter on the computation time, Fig. 14 shows the computation times in relation to the availability of one specific power system. The simulated system contains ten repairable sources in a eight-out-of-ten structure and one energy-storage unit, both connected in series. The MTBF and MTTR of the sources and of the energy-storage unit were varied yielding different system availabilities for each scenario, which can be calculated analytically.

The system with its different MTBF and MTTR values was simulated with a 1-h simulation time step until 2000 failures were obtained for each scenario.

The dots on the graph show the computation time needed to simulate 2000 failures of each scenario in relation to the expected availability of the system.

As it can be observed, the computation time increases exponentially once the system availability approaches 100%, yet the results do not strictly follow an exponential curve, demonstrating that not only system availability plays a role in the computation time. Further investigations should be undertaken to determine which and how strongly certain system parameters influence the computation time.

VI. DISCUSSION

Even an illustrative small-scale system, such as the one shown in Fig. 11, cannot be analyzed analytically with the boolean or sampling Monte-Carlo techniques. Although an SMCS is an established state-of-the-art technique for MESs, the simulation methods of the current state-of-the-art literature that employs graph theory deal poorly with components such as batteries or diesel generators, i.e., components with a limited energy-storage capacity, either by considering them as either infinite sources, as in [62], or by neglecting them altogether, as in [61]. Clearly one could also use the approach of [69], where a microgrid with photovoltaics, energy storage, time-varying loads, and microturbine

systems was simulated using probabilistic models and an SMCS, without, however, simulating the system structure in-between these components. The state-of-the-art literature does not offer a general framework for the analysis of decentralized power systems, unlike the graph-based representation presented here.

A. Advantages Over Existing Methods

The graph representation facilitates changes in the system structure and makes it possible to quickly analyze different topologies. For example, adding a new source or energy storage unit or changing their nominal power ratings can be done by simply altering the data loaded during the system initialization, as shown in Fig. 8. No electrical parameters are required, as it is assumed that the electrical network has been properly designed. This, on the other hand, means that the simulation cannot be used to calculate the real power flowing through the system components. It does not guarantee that the calculated power flow is economically feasible. This method is only meant to predict, in terms of the structure and status of the system, if the loads can be properly supplied and thus exclusively meant for the evaluation of power-system reliability.

The main strength of this method is the variety of components that can be included in the simulation under the same framework, including energy-storage units with limited energy-storage capacity, bidirectional power lines, and components with different repair and failure behaviors modeled by unique distributions. Therefore this method allows not only the simulation of MESs as in [69], but also the simulation of the system structure itself with a scalable framework that can be applied directly to other system topologies.

Multistate components can be integrated in the presented method as well, as long as the behaviors of such components can be modeled by changing the capacities of their edges in the graph. These features allow time-varying loads and sources, such as wind and solar power, to be integrated in the simulation, since their produced or consumed power can be modeled by setting their edge capacities in function of the simulation time. Note, however, that changing the load or source powers changes the system structure, and hence requires a new MCF solution, potentially making the simulation additionally burdensome in terms of computational effort.

Indices, such as loss of load expectation and loss of load probability can also be simulated by changing how the simulation deals with a critical system failure in SR 6. Instead of using a threshold, as expressed in (15), and requiring that all loads are properly supplied, one can allow the simulation to run even when not all the load demands have been met. With the simulated outages, it is possible to calculate the frequency and duration of the loss of load indices.

B. Limitations

This method cannot simulate specific interactions between components other than those modeled by the graph structure itself. For example, when components are connected in a chain, the failure of one component will prevent any flow from going through the chain. This is the expected behavior of a flow network. If the failure of a given component causes another to fail and this cascaded-failure behavior is not covered by the graph, it must be accounted for and programmed manually. An additional graph layer could be used to model specific interactions to, for example, specify components that fail simultaneously. Stand-by redundancies must also be modeled manually, either by considering that components not producing or transferring power cannot fail or, again, by using an additional graph layer to model these interactions, similar to the approach described in [45].

Given that the presented method of reliability analysis is an SMCS, it has the same limitations of similar stochastic approaches. If components have very small failure rates, that can lead to long simulation times. They can be reduced if longer time steps t_s are used. Longer time steps, however, interfere with the stepwise calculation of the state-of-charge of energy storage units—see (20)—making them less precise. Note however that, if the energy storage units are small and cannot provide power beyond the time needed to repair sources, they will have little to no impact on the system reliability.

A flow-network representation of the power system is employed. It uses the general concept of power as the graph commodity. Since line impedances and voltage limits are not considered, that means the power flows calculated by the MCF method are not as accurate as those obtained with a more detailed power-flow analysis. Hence, the presented method can be considered a flow-connectivity analysis of power systems, making it more suitable to evaluate and compare the reliability and availability of topologies of decentralized and local power systems in the design phase, when detailed line parameters are not available with certainty yet.

C. Future Improvements and Extensions

The MCF solution only changes if a change in the system structure occurs. Since the samples that dictate the times between failures and repairs of all components are known, it would be possible to skip simulation time steps when no failures or repairs are predicted to happen. This would reduce the total computing time needed for the simulation and would transform it into an event-based SMCS, similar to the next-event approach shown in [52].

Another option to reduce simulation times is storing the calculated MCF solutions and mapping them to the respective system states, therefore avoiding having to recalculate an MCF solution for system states that were previously encountered during simulation.

The method may also be enhanced to include numerical power-flow calculations to determine the real loading of the lines and components in the power system, given it is already modeled with a graph structure that can be converted to an admittance matrix if the electrical system parameters are known. This will enlarge the scope of application to further extensions of an existing decentralized power system to the design of larger power systems with typical line parameters. With known parameters, it would be possible to implement dynamic thermal

line models, as shown in the studies [14], [35], [37], to evaluate how much of an impact the weather conditions would have on the reliability and availability of an existing DPS.

D. Algorithmic Complexity

One cannot forget to mention the polynomial computational complexity of MCF algorithms. One of the most advanced algorithms, developed by Orlin in 1997, runs in $\mathcal{O}(\min(n^2 m \log(nC), n^2 m^2 \log(n)))$ [70], where n is the number of nodes, m the number of edges, and C the maximum edge capacity found in the graph, which must be an integer. The Python library NetworkX uses a primal network-simplex method for the calculation of an MCF solution. This effectively makes the method described in the present article perfect to analyze small-scale decentralized local systems, but unfit for large and dense power systems. It must be noted, however, that this limitation is also present in other similar graph-based methods.

VII. Conclusion

This article introduced a reliability analysis method for decentralized and local power system topologies integrating flow-network graphs and an SMCS. The method is able to model multistate sources, loads, distributors, energy-storage elements along with their states of charge, bidirectional power flows, and radial or meshed networks. The possibility of simulating all these different components under the same framework using the flow-graph representation is a distinguishing feature of this method. It therefore extends the functionality of similar approaches found in the literature.

This graph-based framework can be used to simulate multiple power-system topologies with components with unique failure and repair distributions, as well as to include unique events that cause a critical system failure. The graph representation allows quick manipulation of system parameters without having to rewrite the rules for system success, as it is calculated via the MCF algorithm automatically. Hence, adding new sources, changing edge capacities or setting longer component repair times is no longer a burdensome activity. It can be used in early system design phases, when electric parameters, such as impedances, voltages, and currents, are still not known. It is especially suitable to analyze modern power systems with decentralized generators, energy-storage units, and loads that are geographically distributed and connected through meshed power lines, e.g., microgrids.

The sequential Monte-Carlo method employed for simulation provides a number of reliability indices, such as discrete probability density functions for the mean time between critical failures, mean time to system repair, and system availability. These and other indices can be used to compare different power system topologies, a process facilitated by the flow-network graph implementation. The simulation results of the three different scenarios, two of those analyzed analytically as well as a proof of concept, showcase the method capabilities.

References

[1] "United nations environment programme (UNEP) with DTU partnership. Emissions GReport," 2020. Accessed: Sep. 30, 2021. [Online]. Available: https://www.unep.org/emissions-gap-report-2020

[2] A. M. Adil and Y. Ko, "Socio-technical evolution of decentralized energy systems: A critical review and implications for urban planning and policy," *Renewable Sustain. Energy Rev.*, vol. 57, pp. 1025–1037, May 2016.

[3] G. Strbac, "Demand side management: Benefits and challenges," *Energy Policy*, vol. 36, no. 12, pp. 4419–4426, Dec. 2008.

[4] P. Palensky and D. Dietrich, "Demand side management: Demand response, intelligent energy systems, and smart loads," *IEEE Trans. Ind. Informat.*, vol. 7, no. 3, pp. 381–388, Aug. 2011.

[5] H. Farhangi, "The path of the smart grid," *IEEE Power Energy Mag.*, vol. 8, no. 1, pp. 18–28, Jan./Feb. 2010.

[6] A. Gholami, F. Aminifar, and M. Shahidehpour, "Front lines against the darkness: Enhancing the resilience of the electricity grid through microgrid facilities," *IEEE Electrific. Mag.*, vol. 4, no. 1, pp. 18–24, Mar. 2016.

[7] R. Brännvall, M. Siltala, J. Gustafsson, J. Sarkinen, M. Vesterlund, and J. Summers, "EDGE," in *Proc. 11th ACM Int. Conf. Future Energy Syst.*, 2020.

[8] P.R. Padma and D. Rekha, "Sustainability modelling and green energy optimisation in microgrid powered distributed fogmicrodatacenters in rural area," in *Wireless Networks*, vol. 27, pp. 5519–5532, Jan. 2020.

[9] J.-P. Signoret and A. Leroy, "The inductive approaches," in *Springer Series in Reliability Engineering*. Berlin, Germany: Springer Int. Publishing, 2021, pp. 139–143.

[10] *Reliability Block Diagrams* Standard IEC 61078:2016, International Electrotechnical Commission, Geneva, Switzerland, 2016. [Online]. Available: https://webstore.iec.ch/publication/25647

[11] W. S. Lee, D. L. Grosh, F. A. Tillman, and C. H. Lie, "Fault tree analysis, methods, and application a review," *IEEE Trans. Rel.*, vol. R-34, no. 3, pp. 194–203, Aug. 1985.

[12] *Fault Tree Analysis (FTA)* Standard IEC 61025:2006,International Electrotechnical Commission, Geneva, Switzerland, 2006. [Online]. Available: https://webstore.iec.ch/publication/4311

[13] *Application of Markov Techniques* Standard IEC 61165:2006, International Electrotechnical Commission, Geneva, Switzerland, 2016. [Online]. Available: https://webstore.iec.ch/publication/4721

[14] O. A. Lawal and J. Teh, "A framework for modelling the reliability of dynamic line rating operations in a cyber-physical power system network," *Sustain. Energy, Grids Netw.*, vol. 35, Sep. 2023, Art. no. 101140.

[15] J. Abraham, "An improved algorithm for network reliability," *IEEE Trans. Rel.*, vol. R-28, no. 1, pp. 58–61, Apr. 1979.

[16] K. Heidtmann, "Smaller sums of disjoint products by subproduct inversion," *IEEE Trans. Rel.*, vol. 38, no. 3, pp. 305–311, Aug. 1989.

[17] C.-C. Jane, J.-S. Lin, and J. Yuan, "Reliability evaluation of a limited-flow network in terms of minimal cutsets," *IEEE Trans. Rel.*, vol. 42, no. 3, pp. 354–361, Sep. 1993.

[18] Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Trans. Comput.*, vol. C-35, no. 8, pp. 677–691, Aug. 1986.

[19] O. Wing and P. Demetriou, "Analysis of probabilistic networks," *IEEE Trans. Commun.*, vol. 12, no. 3, pp. 38–40, Sep. 1964.

[20] L. Fratta and U. Montanari, "A Boolean Algebra method for computing the terminal reliability in a communication network," *IEEE Trans. Circuit Theory*, vol. 20, no. 3, pp. 203–211, May 1973.

[21] L. Page and J. Perry, "A practical implementation of the factoring theorem for network reliability," *IEEE Trans. Rel.*, vol. 37, no. 3, pp. 259–267, Aug. 1988.

[22] S. Soh and S. Rai, "CAREL: Computer aided reliability evaluator for distributed computing networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 2, no. 2, pp. 199–213, Apr. 1991.

[23] A. Rauzy, E. Châtelet, Y. Dutuit, and C. Bérenguer, "A practical comparison of methods to assess sum-of-products," *Rel. Eng. System Saf.*, vol. 79, no. 1, pp. 33–42, 2003.

[24] W.-C. Yeh, "An improved sum-of-disjoint-products technique for the symbolic network reliability analysis with known minimal paths," *Rel. Eng. Syst. Saf.*, vol. 92, no. 2, pp. 260–268, Feb. 2007.

[25] J. Xing, C. Feng, X. Qian, and P. Dai, "A simple algorithm for sum of disjoint products," in *Proc. Proc. Annu. Rel. Maintainability Symp.*, 2012, pp. 1–5.

[26] R. A. Sahner and K. S. Trivedi, "Reliability modeling using SHARPE," *IEEE Trans. Rel.*, vol. R-36, no. 2, pp. 186–193, Jun. 1987.

[27] L. G. Valiant, "The complexity of enumeration and reliability problems," *SIAM J. Comput.*, vol. 8, no. 3, pp. 410–421, Aug. 1979.

[28] J. Dugan, S. Bavuso, and M. Boyd, "Dynamic fault-tree models for fault-tolerant computer systems," *IEEE Trans. Rel.*, vol. 41, no. 3, pp. 363–377, Sep. 1992.

[29] S. Distefano and A. Puliafito, "Dynamic reliability block diagrams VS dynamic fault trees," in *Proc. Proc. Annu. Rel. Maintainability Symp.*, 2007, pp. 71–76.

[30] J. Friederich and S. Lazarova-Molnar, "Towards data-driven reliability modeling for cyber-physical production systems," *Procedia Comput. Sci.*, vol. 184, pp. 589–596, 2021.

[31] P. Niloofar and S. Lazarova-Molnar, "Data-driven extraction and analysis of repairable fault trees from time series data," *Expert Syst. Appl.*, vol. 215, Apr. 2023, Art. no. 119345.

[32] L. Chi et al., "Data-driven reliability assessment method of integrated energy systems based on probabilistic deep learning and gaussian mixture model-hidden Markov model," *Renewable Energy*, vol. 174, pp. 952–970, Aug. 2021.

[33] H. A. Khorshidi, I. Gunawan, and M. Y. Ibrahim, "Data-driven system reliability and failure behavior modeling using FMECA," *IEEE Trans. Ind. Informat.*, vol. 12, no. 3, pp. 1253–1260, Jun. 2016.

[34] G. Pandian, M. Pecht, E. Zio, and M. Hodkiewicz, "Data-driven reliability analysis of boeing 787 dreamliner," *Chin. J. Aeronaut.*, vol. 33, no. 7, pp. 1969–1979, Jul. 2020.

[35] J. Teh, C.-M. Lai, and Y.-H. Cheng, "Impact of the real-time thermal loading on the bulk electric system reliability," *IEEE Trans. Rel.*, vol. 66, no. 4, pp. 1110–1119, Dec. 2017.

[36] M. A. Rios, J. L. Vera, and M. F. Perez, "Availability assessment of voltage source converter HVDC grids using optimal power flow-based remedial actions," *High Voltage*, vol. 5, no. 5, pp. 523–533, Oct. 2020.

[37] C.-M. Lai and J. Teh, "Network topology optimisation based on dynamic thermal rating and battery storage systems for improved wind penetration and reliability," *Appl. Energy*, vol. 305, Jan. 2022, Art. no. 117837.

[38] Y. Su and J. Teh, "Two-stage optimal dispatching of AC/DC hybrid active distribution systems considering network flexibility," *J. Modern Power Syst. Clean Energy*, vol. 11, no. 1, pp. 52–65, 2023.

[39] G. Stefopoulos, A. Meliopoulos, and G. Cokkinides, "Probabilistic power flow with nonconforming electric loads," in *Proc. Int. Conf. Probabilistic Methods Appl. Power Syst.*, 2004, pp. 525–531.

[40] M. Hajian, W. D. Rosehart, and H. Zareipour, "Probabilistic power flow by Monte Carlo simulation with latin supercube sampling," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1550–1559, May 2013.

[41] G. E. Constante-Flores and M. S. Illindala, "Data-driven probabilistic power flow analysis for a distribution system with renewable energy sources using Monte Carlo simulation," *IEEE Trans. Ind. Appl.*, vol. 55, no. 1, pp. 174–181, Jan./Feb. 2019.

[42] H. Wang, B. Wang, P. Luo, F. Ma, Y. Zhou, and M. A. Mohamed, "State evaluation based on feature identification of measurement data: For resilient power system," *CSEE J. Power Energy Syst.*, vol. 8, no. 4, pp. 983–992, 2022.

[43] G. Satumtira and L. Dueñas-Osorio, *Synthesis of Modeling and Simulation Methods on Critical Infrastructure Interdependencies Research*. Berlin, Germany: Springer, 2010.

[44] E. Zio, "Stochastic simulation of critical infrastructures for electric power transmission," in *Energy Security*. Dordrecht, The Netherlands: Springer, 2011, pp. 109–124.

[45] J. Johansson, H. Hassel, and E. Zio, "Reliability and vulnerability analyses of critical infrastructures: Comparing two approaches in the context of power systems," *Rel. Eng. Syst. Saf.*, vol. 120, pp. 27–38, Dec. 2013.

[46] A. A. Kadhem, N. I. A. Wahab, I. Aris, J. Jasni, and A. N. Abdalla, "Computational techniques for assessing the reliability and sustainability of electrical power systems: A review," *Renewable Sustain. Energy Rev.*, vol. 80, pp. 1175–1186, Dec. 2017.

[47] R. Billinton and A. Sankarakrishnan, "A comparison of Monte Carlo simulation techniques for composite power system reliability assessment," in *Proc. IEEE WESCANEX 95. Commun. Power Comput. Conf.*, 1995, pp. 145–150.

[48] C. Singh and J. Mitra, "Monte Carlo simulation for reliability analysis of emergency and standby power systems," in *Proc. IEEE Conf. Rec. Ind. Appl. Conf. 13th IAS Annu. Meeting*, 1995, pp. 2290–2295.

[49] W. Gang, D. Mao-Sheng, and L. Xiao-Hua, "Analysis of UPS system reliability based on Monte Carlo approach," in *Proc. IEEE Region 10 Conf. TENCON*, 2004, pp. 205–208.

[50] M. M. Ghahderijani, S. M. Barakati, and S. Tavakoli, "Reliability evaluation of stand-alone hybrid microgrid using sequential Monte Carlo simulation," in *Proc. 2nd Iranian Conf. Renewable Energy Distrib. Gener.*, 2012, pp. 33–38.

[51] M. Moazzami, R. Hemmati, F. H. Fesharaki, and S. R. Rad, "Reliability evaluation for different power plant busbar layouts by using sequential Monte Carlo simulation," *Int. J. Elect. Power Energy Syst.*, vol. 53, pp. 987–993, Dec. 2013.

[52] Y. Lei and A. Q. Huang, "Data center power distribution system reliability analysis tool based on Monte Carlo next event simulation method," in *Proc. IEEE Energy Convers. Congr. Expo.*,2017, pp. 2031–2035.

[53] A. Sankarakrishnan and R. Billinton, "Sequential Monte Carlo simulation for composite power system reliability analysis with time varying loads," *IEEE Trans. Power Syst.*, vol. 10, no. 3, pp. 1540–1545, Aug. 1995.

[54] R. Kinney, P. Crucitti, R. Albert, and V. Latora, "Modeling cascading failures in the north American power grid," *Eur. Phys. J. B*, vol. 46, no. 1, pp. 101–107, Jul. 2005.

[55] F. Monforti and A. Szikszai, "A MonteCarlo approach for assessing the adequacy of the European gas transmission system under supply crisis conditions," *Energy Policy*, vol. 38, no. 5, pp. 2486–2498, May 2010.

[56] I. Morro-Mello, A. Padilha-Feltrin, J. D. Melo, and F. Heymann, "Spatial connection cost minimization of EV fast charging stations in electric distribution networks using local search and graph theory," *Energy*, vol. 235, Nov. 2021, Art. no. 121380.

[57] R. Albert, I. Albert, and G. L. Nakarado, "Structural vulnerability of the North American power grid," *Phys. Rev. E*, vol. 69, no. 2, Feb. 2004, Art. no. 025103.

[58] G. A. Pagani and M. Aiello, "The power grid as a complex network: A survey," *Physica A: Stat. Mechanics Appl.*, vol. 392, no. 11, pp. 2688–2700, Jun. 2013.

[59] T. Werho, V. Vittal, S. Kolluri, and S. M. Wong, "Power system connectivity monitoring using a graph theory network flow algorithm," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4945–4952, Nov. 2016.

[60] J. Fang, C. Su, Z. Chen, H. Sun, and P. Lund, "Power system structural vulnerability assessment based on an improved maximum flow approach," *IEEE Trans. Smart Grid*, vol. 9, no. 2, pp. 777–785, Mar. 2018.

[61] E. Ferrario, N. Pedroni, and E. Zio, "Evaluation of the robustness of critical infrastructures by hierarchical graph representation, clustering and Monte Carlo simulation," *Rel. Eng. Syst. Saf.*, vol. 155, pp. 78–96, Nov. 2016.

[62] P. Praks, V. Kopustinskas, and M. Masera, "Probabilistic modelling of security of supply in gas networks and evaluation of new infrastructure," *Rel. Eng. Syst. Saf.*, vol. 144, pp. 254–264, Dec. 2015.

[63] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Hoboken, NJ, USA: Prentice Hall, 1993.

[64] A. Schrijver, "On the history of the transportation and maximum flow problems," *Math. Program.*, vol. 91, no. 3, pp. 437–445, Feb. 2002.

[65] L. R. Ford and D. R. Fulkerson, *Flows in Networks*. Princeton, NJ, USA: Princeton Univ. Press, 1963.

[66] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proc. 7th Python Sci. Conf.*, 2008, pp. 11–15.

[67] W. McKinney, "Data structures for statistical computing in python," in *Proc. 9th Python Sci. Conf.*, 2010, pp. 56–61.

[68] J. Angus, "On computing MTBF for a k-out-of-n: G repairable system," *IEEE Trans. Rel.*, vol. 37, no. 3, pp. 312–313, Aug. 1988.

[69] X. Song, Y. Zhao, J. Zhou, and Z. Weng, "Reliability varying characteristics of PV-ESS-based standalone microgrid," *IEEE Access*, vol. 7, pp. 120872–120883, 2019.

[70] J. B. Orlin, "A polynomial time primal network simplex algorithm for minimum cost flows," *Math. Program.*, vol. 78, no. 2, pp. 109–129, Aug. 1997.