



Combinatorial Sequences for Disaster Scenario Generation

Bernhard Garn¹ · Klaus Kieseberg¹ · Dominik Schreiber¹ · Dimitris E. Simos¹ 

Received: 12 November 2021 / Accepted: 5 May 2023 / Published online: 7 June 2023
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2023

Abstract

Training exercises are an important tool in crisis management, as they can assist in a multitude of tasks, such as planning pre-crisis resource requirements and allocation, response planning and help train emergency personnel for actual crises. To be effective, the exercises have to utilize well constructed scenarios and be able to replicate certain characteristics of a crisis situation. In this paper, we propose a conceptual mathematical modeling approach for the automated generation of scenarios for disaster exercises via certain combinatorial sequence structures. The derived scenarios within an exercise collectively fulfill different notions of combinatorial sequence coverage, thereby providing the means to test existing response strategies for deficiencies as well as to train emergency personnel for their ability to handle different arrangements of events. This guaranteed diversity by construction can be used as a basis to obtain quantitative assurance statements when these scenarios have been successfully mastered by participants in exercises. We illustrate our proposed approach utilizing two different combinatorial structures for two example disasters.

Keywords Dynamics of disasters · Crisis management exercises · Combinatorial testing · Combinatorial sequence testing

Mathematics Subject Classification (2020) 05B99 · 90B99

This article is part of the Topical Collection on *Dynamics of Disasters*

✉ Dimitris E. Simos
dsimos@sba-research.org

Bernhard Garn
bgarn@sba-research.org

Klaus Kieseberg
kkieseberg@sba-research.org

Dominik Schreiber
dschreiber@sba-research.org

¹ SBA Research, 1040 Vienna, Austria

1 Introduction

Disasters and crises are extraordinary events or periods which possess great potential for harm [1]. Especially when it comes to disasters, damages caused by these events can be so severe, modern societies cannot cope with them by themselves, using only their resources. In particular, damages to critical infrastructures and disruptions of critical services can even lead to human fatalities. The management of disaster and crisis prevention, as well as corresponding relief measures, poses an enormous operational challenge to local governments and organizations tasked with it.

As they are not always predictable, crises, such as pandemics, natural disasters, and failure of critical infrastructure unfortunately cannot always be avoided. In 2020 alone, there have been 416 natural disasters worldwide, not including the SARS COVID-19 pandemic, resulting in global economic losses of about 268 billion US Dollars [2].

In order to minimize potential damages caused by a disaster, efforts must be made to ensure the functionality of important services and to help the appropriate endeavors for crisis relief. Here, one of the most important tools are exercises, as they can assist in planning pre-crisis resource requirements and allocation, response planning and provide a low-risk environment to help train personnel for actual crises. Exercises can also be used to assess emergency policies, procedures, novel tools, and methods used for crisis management [3].

The effectiveness of exercises has been subject of various research. While there exist a multitude of works in the literature focusing on designing and conducting crisis exercises, in our judgement, they mainly address the topic from an administrative point of view, focusing on organizational aspects as well as the formulation of high level exercise goals. Some literature on the generation of scenarios for crisis exercises also exists; however, the scenarios devised therein are generally generated with a very specific and predetermined exercise goal in mind, simulating a disaster or crisis under very specific conditions, which are already well known and whose results can be anticipated and thus not really lead to new insights [4]. Additionally, a consequence of the highly time-consuming and expensive endeavor of developing scenarios for exercises is often that only a limited number of scenarios are being developed and constantly reused [5], which reduces the overall scope of the training and, subsequently, diminishes training effectiveness [6], especially in the case of continuous training [7]. Building upon [6], we argue that it is therefore crucial to the training effort to have the means to structurally derive “robust libraries” of diverse disaster scenarios which cover potential events, event combinations, and event sequences *sufficiently well for some formal notion of coverage*, which should be supported by “research-based heuristics to ensure that scenario outputs are domain-valid and pedagogically effective” [6].

Based on the statements given above from the literature, we deduce that there is a clear need for more structural and comprehensive approaches in disaster scenario modeling as well as scenario generation for their subsequent use in disaster

exercises. In particular, the availability of automated scenario generation methods, which could guarantee full coverage for some formal notion of coverage in the generated disaster scenarios, seems to us to be especially desirable. Furthermore, we posit that such automatically generated diverse libraries of scenarios would not only provide more training diversity for the participants, but could also help identify deficiencies and weaknesses of disaster handling strategies, which might have otherwise been overlooked and only revealed in the case of a real disaster. Consequently, our aim is to propose a methodology for evaluating the resilience of existing strategies of affected stakeholders,¹ which also addresses the previously discussed challenges. More precisely, our goal is to provide effective and automated means for enabling stakeholders to conduct *testing and evaluation* of their response policies under as diverse scenarios as possible. In this work, we present a generic mathematical approach for generating disaster exercises that provides general abstract modeling concepts and capabilities as well as automated scenario generation methods. Our proposed approach can be applied to different *disasters under test* (DUTs) and is designed to incorporate expertise of domain experts, while also providing coverage guarantees for some selectable notions of combinatorial coverage in all generated disaster exercise plans.

The motivation for our automated scenario generation approach is based on the success and efficiency of automated software testing. In [8], the authors have developed a strategy for automated unit test generation for continuous integration testing, with their experimental results showing a significant improvement for branch coverage and test generation time. These properties also play an important role in disaster scenario training as it should cover as many responder strategies as possible, while reducing the effort of scenario generation. An overview of automated test generation for software and a discussion of recent developments and tools is provided in [9]. The application of automated software testing tools has found thousands of new bugs—many of them critical from a reliability or security point of view—in a lot of different application domains, including, but not limited to, embedded systems, application programming interfaces (APIs), and web and mobile applications. In recent years, a new field in the domain of software testing has opened up, successfully applying combinatorial methods for testing purposes [10]. The methodological origins of this field, which is termed combinatorial testing (CT) for software [11], can be traced to the field of *Design of Experiments* (DoE) within statistics [12], with the latest achievements and future directions of CT being outlined in [13, 14]. The central concepts within CT are variants of *combinatorial coverage*, which guarantee, for a positive natural number t , the coverage (i.e., appearance) of t -tuples or certain sequences of length t within mathematical artifacts and test sets generated from them. It is of crucial importance that these notions of coverage can actually be measured in existing test sets [15, 16] as well as actually enforced in newly generated *combinatorial test sets* [17, 18] (also called test sets of strength t or t -way test sets) in practical settings and on industrial scale. Empirical studies in software testing

¹ This includes, but is not limited to, first responders along with governmental agencies and additional parties.

have reported equivalent fault detection capabilities of higher-strength combinatorial test sets compared to exhaustive ones [19, 20] and thereupon the terminology *pseudo-exhaustive test sets* for combinatorial test sets and *pseudo-exhaustive testing* for CT as synonyms has arisen. Advances in algorithm design [21] and modern implementations [22] have made it possible to tightly compress the required t -tuples into t -way test sets, leading to a considerable reduction in the number of test cases of combinatorial test sets compared to exhaustive ones. Hence, CT provides highly efficient and effective software testing in practice at lower cost.

The approach presented in this paper is based on finite, nonempty sequences arising in discrete mathematics for modeling and automated generation of disaster scenarios via combinatorial methods for use in disaster exercises or crisis response policy evaluations. Our approach is designed to reveal weaknesses or even failures in responding strategies of affected stakeholders, caused by specific temporal interactions (i.e., orderings) of events within the entire temporal event sequence of a disaster or crisis. To this end, we leverage special combinatorial sequence structures from combinatorial design theory exhibiting different notions of combinatorial sequence coverage for the construction of disaster scenarios. The way in which these combinatorial design structures are employed in this work ensures that their defining abstract mathematical sequence coverage properties are transformed to the generated sets of disaster scenarios. This means that the obtained sets of disaster scenarios retain (i.e., fulfill themselves) these combinatorial sequence coverage properties, which on the side of the application domain of disaster management can be interpreted and expressed as coverage of temporal event orderings—to which we will later refer to as *stories*—by the generated disaster exercises. Given such a set of disaster scenarios (i.e., a disaster exercise generated via combinatorial methods) satisfying some notion of combinatorial sequence coverage, the *standard operating procedures* (SOPs) or *responding strategies* of stakeholders affected by the modeled DUT can then be evaluated against all individual disaster scenarios, to test how well they can handle and cope with them. The guaranteed coverage of some notion of combinatorial sequence coverage of such an exercise implies that if all constituting scenarios of the generated exercise were managed successfully, then this outcome could be interpreted to demonstrate and prove the effectiveness and resilience of existing disaster management strategies with respect to the specific occurring notion of combinatorial sequence coverage; if not, then those scenarios which revealed shortcomings could be used as basis to develop improvements to address any uncovered issues.

We would like to explicitly mention and emphasize that the underlying notion of combinatorial coverage together with its defining numerical values can be seen as a complexity measure of the exercises generated via combinatorial methods as proposed in this work. Moreover, this complexity can be freely selected and adjusted, providing exercise designers with an additional method alongside other various abstract properties of the employed building blocks, to assess and tune the difficulty of individual scenarios within the exercise. In particular, the complexity and difficulty of generated exercises can be raised to keep ahead of unfolding trends and discoveries or for testing policies also against more advanced scenarios with increased impact. Once the SOPs or disaster response policies of affected stakeholders for

some DUT have successfully managed a combinatorially generated training exercise with predetermined levels of complexity and difficulty for some notion of combinatorial sequence coverage, this achievement can be leveraged to derive a quantitative assessments in terms of an assurance for its resilience in addition to existing and established quantitative assessments of SOP or disaster response policies.

The mathematical artifacts employed in this work can be automatically generated and, consequently, also the disaster exercises satisfying combinatorial coverage notions derived from them. In particular, the optimization of the underlying combinatorial structures for minimizing their size (i.e., number of rows) has the further advantage of translating directly to a reduction in the number of scenarios of generated disaster exercises, while still providing full coverage for the underlying notion of combinatorial coverage. Hence, our work ameliorates automation for scenario generation and thereby also reduces the organizational cost of disaster exercise design and subsequent scenario generation.

Our proposed approach based on combinatorial sequence structures provides flexible and fine-grained modeling capabilities that are available to domain experts for instantiating these models for specific DUTs with expert knowledge by selecting—alongside other possible choices—appropriate events for the considered DUTs. For this reason—among others—is our proposed approach applicable to numerous different types of disasters, including man-made as well as non-man-made disasters. Furthermore, due to the design of our mathematical modeling approach, it is capable to take into account and comply with required and important properties and characteristics of single events, individual scenarios, and entire disaster exercises that we determined when analyzing the current research literature on the topic. We used the collected information to obtain a baseline to compare against and improve upon identified contemporary challenges and reported issues.

We showcase our proposed approach for two different types of disasters, exemplified for the case of a non-man-made disaster by a flooding and for the case of a man-made disaster by a cyber crisis involving critical infrastructure. In both examples, we make use of two different combinatorial sequence structures—both having already been successfully applied in the domain of combinatorial testing for software—that allow for the modeling and generation of disaster exercise scenarios that will not only satisfy the important and indispensable characteristics identified from the literature pointed out before, but will also give rise to disaster exercises, comprised of one or more disaster scenarios, that will guarantee 100% coverage for certain notions of combinatorial sequence coverage. We also highlight key differences in the generated disaster exercises resulting from the different properties of the underlying notions of combinatorial sequence coverage and point out their respective advantages and disadvantages.

Contribution. In particular, this paper makes the following contributions:

- Analysis of disaster exercise scenarios and scenario generation methods reported in the literature,
- Proposes the usage of certain discrete sequence structures with specific combinatorial sequence coverage properties for disaster scenario generation for exercises,

- Showcases the exercise generation process based on the proposed approach for two different example disasters.

This paper is structured as follows. Related work is discussed in Sect. 2. In Sect. 3, we analyze the properties of exercise scenarios and their events from the literature. Our proposed discrete mathematical modeling and generation approach for disaster exercises is presented in Sect. 4. In Sect. 5, we provide two walkthrough examples of how our proposed approach could be applied in practice. We conclude the paper in Sect. 6 and also provide an outlook on future work.

2 Related Work

It is important to note that while the literature in this area is in agreement that crises and disasters are not the same (although they are closely related to each other, as, depending on the definition used, a disaster might be an agent of a crisis or a crisis might turn into a disaster), no universally accepted definitions have yet been agreed upon [23]. This is attributed to the dependence of these terms to the specific domain they are occurring in. In [24], the authors give an outline of different definitions of the term disaster in general as well as for specific disciplines. Similarly, [25] systematically reviews different definitions of the terms *disaster* and *crisis* found in the literature in order to establish differences and similarities between those two terms. Due to the blurred semantic lines separating the terms of *disaster* and *crisis*, the mainstream literature often uses the two terms interchangeably. Subsequently, the same can be observed with the terms *crisis management* and *disaster management*. As our proposed approach operates at a very abstract level where disasters and crises feature similar characteristics and can both be modeled by a sequence of crisis events, we will join in with the mainstream literature and, for the scope of this work, use the terms *disaster* and *crisis* interchangeably throughout this paper.

Traditionally, a lot of the research regarding the topic of crisis management is focused on the study of former real life crises. In [26], the authors show how disaster case studies can help to better understand risks posed by extreme weather- and climate-related events. Case studies not only allow to retrospectively analyze the development of a crisis and identify its major drivers, they are also a valuable means to assess the effectiveness of the relief measures utilized and for providing scenarios for the testing of new tools, methods, and potential relief plans. In [27], the data extracted from flood maps created with information from former floodings is used to evaluate a simulation tool developed to assess the viability of the use of sandbags for various flood scenarios.

One shortfall of case studies is the focus on past real-life disaster scenarios; hence, for preparing for future and yet unknown crises, other, complementary, methods are needed. Here, the design and effectiveness of exercises for preparing against hitherto unexplored and unforeseen crisis scenarios have been subjects of various research. In [28], the author discusses the benefits and problems of using exercise scenarios as a way of effectively preparing against unpredictable variables of crises. The author comes to the conclusion that the development of information and

communication technologies plays a big part in the effectiveness of scenarios and games for efforts preparing for crisis situations. Similarly, [29] takes a deeper look at the role and potential benefits of exercises in disaster management. Another work, [30], analyzes the effect of exercises on emergency personnel and describes the factors limiting the effectiveness of crisis exercises by observing exercises and conducting interviews with participants.

Stimulated by the results of the research on the importance and effectiveness of exercises, a few programs have been developed aiming to raise the resilience of countries and organizations by providing them with a library of fundamental principles for exercise programs and introducing them to the most common approaches to exercise design and management [3, 31]. It is, however, important to note that, due to their operational character, these programs are of a very abstract nature and focus heavily on the administrative side of exercise management. Topics like scenario design (the selection of events as well as their temporal arrangement) and generation are discussed only superficially therein, and they content themselves with pointing to domain experts for these tasks without providing specific guidance.

Still, there do exist some works dealing with novel ideas on the design and generation of exercise scenarios. An in-depth guideline for the development of effective scenario blueprints, so called *Master Scenario Event Lists*, is provided in [32]. Another work, [33], introduces a novel five-step method for extracting functional requirements for exercises from user-requirements utilizing storyboarding. In [34], the authors introduce a new theoretical method for exercise designers to create scenarios for functional exercises. The framework proposed there features a transdisciplinary approach, bridging the gap between crisis management, dramaturgy, and modeling derived from system engineering.

We point to [35] for a survey reviewing and consolidating the existing research on scenario generation techniques and related crisis simulation frameworks. The research for novel and modern IT-supported methods for generating scenarios is especially prevalent in simulation training, where a variety of different scenarios covering multiple training aspects is beneficial for providing effective training. As the manual creation of viable training scenarios is time consuming and expensive, researchers are especially interested in coming up with automated methods for scenario generation. This challenge has been approached with a narrative-based generation methodology in [36], while the usage of cellular automata has been proposed in [37]. Personalized or adaptive forms of training in a virtualized learning environment, which includes serious games simulations, have been treated in [38, 39]. In [7] (see also [6, 40]), the authors review some of the research and methods introduced for automated scenario generation in typical simulation training domains such as military training and defense planning along with presenting a novel method for procedural scenario generation using functional L-systems. The underlying conceptual model features an approach utilizing so called *baseline scenarios*, which satisfy previously selected training objectives. With the help of *scenario vignettes*, the authors are then able to modify the complexity of the scenarios. Another work, [41], also presents a generation algorithm that can generate a new scenario from an already existing one. The approach attempts to modify and combine aspects of different members of the initial scenario to find the best

sequence of events that creates the appearance of a realistic mission, achieves a set of training objectives suitable for a given individual trainee while also being tailored to the individual trainee's abilities. Additionally, they defined a set of metrics, to evaluate the success of their scenario generation system.

The authors of [42] propose a new formulation of scenario complexity, which is intended to be used with automated scenario generation systems. The goal is that trainees advance through increasingly complex training scenarios, which help them to gradually gain more experience in an efficient manner by ensuring that trainees receive scenarios appropriate to their experience level. In [43], the authors extend the traditional task complexity formulation by proposing the inclusion of cognitive task elements into the complexity metric, which are important for scenarios that are focused on higher-order thinking skills. Similar to [42], the goal is to improve the definition of scenario complexity and to find optimal training scenarios that are ideally fitted to the skills of the participants.

In [44], the authors propose an architecture to facilitate learning through the principles of deliberate practice, enabled by interoperating next-generation exercise management tools with mixed-reality simulations. An intuitive interface for exercise managers using block-based programming has been proposed in [45] and the design of an externalized AI-based scenario event controller in [46]. In [47], the authors examine the degree to which an integrated training approach using scenario training technologies provided the fidelity necessary to support military casualty care learning objectives. We also remark that with the help of modern smartphone technology, software-based gamification has also been used to directly educate citizens on disaster prevention information [48].

Like the works mentioned above, most of the approaches presented in the literature follow strict rules, where the scenarios are designed with a specific crisis and objective already in mind. While generally these approaches are certainly useful for the training of specific relief measures (i.e., emergency drills), they do not necessarily provide—or even take into account explicitly—notions of coverage for varying situations. In order to use exercises not only for assessing, but also explicitly for stress-testing disaster management plans, it is, therefore, imperative to construct scenarios not confined to specific objectives, but, in contrast, featuring a broad range of “chaotic” sequences in order to identify problems introduced by certain sequences of events.

3 Exercise Scenarios for Crisis and Disaster Management

In this section, we identify and highlight important key concepts and essential properties of disaster exercises and disaster management in general. In Sect. 3.1, we provide an overview of disaster management activities. Disaster exercises in general and different types of exercises are discussed in Sect. 3.2 and Sect. 3.3, respectively. We analyze exercise scenarios in Sect. 3.4 and focus on the characteristics of scenarios and events in Sect. 3.5.

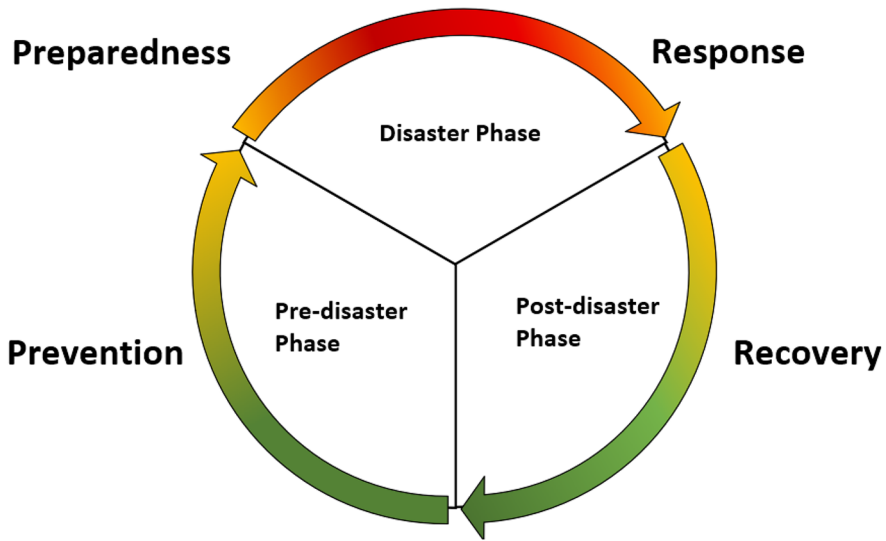


Fig. 1 The four stages of the DRM cycle

3.1 Disaster Management

The process of disaster management is generally conceptualized as the *disaster-risk-management-cycle* [49], or DRM-cycle, as shown in Fig. 1. The DRM-cycle consists of four stages, with each of these stages representing activities for either aiding ex-ante crisis prevention/mitigation (stage 1) and preparedness (stage 2), or ex-post disaster response (stage 3) and recovery (stage 4). The focus of the first stage lies in the prevention or mitigation of the impact of disasters or crises and in particular encompasses all activities that may be undertaken to avert a disaster. For example, the construction of appropriate flood protection and drainage infrastructure in areas at risk of flooding are activities with the goal of preventing a flooding disaster in the first place. The next stage, stage 2, consists of activities and measures, such as designing of evacuation plans and carrying out emergency drills, undertaken with the goal of strengthening preparedness by striving towards ensuring an effective response to future disastrous events and reducing their impact. Activities like evacuation, rescue efforts, or first aid, which are carried out immediately before or after a disaster occurred, are grouped in the response stage of the DRM cycle (stage 3). Finally, the last stage deals with repair- and reconstruction-efforts of the damages incurred during the disaster. As crises and disasters are very similar to each other in terms of these four *developing phases* [23], a similar DRM cycle as shown in Fig. 1 may also be used to describe the timeline of a crisis. The methodology proposed in this paper is positioned in the two pre-disaster phases of the DRM cycle (i.e., ex-ante), enhancing the efforts of preventing and mitigating disasters (stage 1) as well as raising preparedness (stage 2).

3.2 Exercises

Crisis management exercises are proactive measures which are used to strengthen the crisis management capability of countries or organizations by providing them the means to devise, assess, and update their disaster management plans and generally test their abilities to deal with catastrophic situations in a low-risk environment [50]. Crisis management exercises can also be used for the evaluation of novel tools and methods, before they are integrated into SOPs.

“Exercises are based on scenarios that correspond to a sequence of events organized in a specific space-time framework [51].”

Different types of disaster exercises have been identified in the literature on disaster management (which we discuss in detail in Sect. 3.3 below), and depending on its type, an exercise may consist of one extensive or multiple shorter scenarios. An exercise might also play through the same scenario multiple times, with or without slight changes. This helps attendees to solidify the lessons learned during the past iterations, deepens the understanding and engagement of the trainees in the crisis management process, and allows them to focus on the most important task (or tasks) in each iteration, which might not be the same in all iterations. Making small changes to the scenario between iterations can be used to intentionally shift the focus of the exercise-iterations to different areas or can be used to gradually increase the scope of the exercise with each played iteration.

3.3 Types of Exercises

The literature on the topic of disaster management—for the most part—differentiates between three major types of exercises: tabletop, functional, and full-scale exercises [52].

“Tabletop exercises” are *discussion-based exercises* which take the form of facilitated discussions like seminars, workshops, and games. Here, participants can familiarize themselves with or discuss existing emergency procedures on an operational and jurisdictional level or develop and test out new ones on a strictly theoretical and strategic basis. All actions taken during the exercise are fictional, and any communication to entities outside is simulated. As they mostly take place inside a single location or a joined network with all participants present, tabletop exercises are relatively easy and inexpensive to plan and conduct [53].

“Functional exercises” are *operational exercises*, which are conducted to test and evaluate a single or a small number of very specific functions and processes of a disaster response plan, only involving one or a few emergency responder organizations [52]. The focus of this type of exercise lies mostly on the leadership, management, and coordination of the entities involved. These exercises are more complex and operate at a higher level of realism than tabletop exercises, as they are conducted in real time with operational personnel using proper equipment.

“Full-scale exercises” are also *operational exercises* and are conducted to test all or most major processes or functions of an emergency response plan. To this end,

full-scale exercises involve the whole crisis management as well as multiple agencies, organizations, and jurisdictions [31], making full-scale exercises not only very complex to design and manage, but also very expensive.

3.4 Exercise Scenarios

In [28], scenarios are defined as descriptions of the conditions under which the crisis management system or crisis management policy to be designed, tested, or evaluated is assumed to perform. In particular, this means that a scenario specifies a possible, but not necessarily probable, context and series of events.

Generally, a good exercise scenario must satisfy several requirements. Most importantly, there should be an achievable number of clear and specific goals [54]. It should be designed within the expertise of the participants and allow them to test and further develop their specific skillsets. Since they are supposed to simulate events and processes occurring during a crisis, scenarios have to reproduce certain crisis characteristics and effects [34]. To this end, scenario designers might inject events which create urgency and time constraints, add elements of surprise, or present the participants with a dilemma. Moreover, the difficulty of a scenario can be increased further by the implementation of completely unrelated events or red herrings.

Providing a certain level of realism can help with immersion and convey the necessity of such exercises to the trainees, especially in operation-based exercises like functional or full-scale exercises [54]. However, exercise designers must find the right balance between keeping up the illusion of a crisis situation and finding ways to minimize efforts needed for scenario generation. For example, while filling a scenario with red herrings might significantly boost the stress level of the trainees during an exercise, it might interfere with the training of specific skills the scenario was originally meant to train. Similarly, using actors to portray victims or other affected entities during exercise might heighten the level of immersion, but raises the amount of organizational effort needed.

Depending on the type of the exercise, the characteristics and complexity of its scenarios as well as their implementation can vary:

- In *tabletop exercises*, the scenario is typically presented in a narrative form [29], where both the setup and the information about the levels of the escalation process are presented to the participants by a moderator using photos or slides accompanied by text handouts [54]. The scenario itself consists of a series of conceptual events discussed from a high-level point of view, whereby the specific details of the exact procedures of certain sub-processes or subfunctions are neither considered nor taken into account. The abstract nature of such exercises allows for time-outs to assess the steps taken so far, as well as jumps in the scenario-timeline to discuss long-term impacts and developments [29]. Tabletop exercises can easily be adapted and reiterated multiple times, since all actions performed are fictional and a reset to the starting point can be done instantaneously.

- *Functional exercises* are more complex than tabletop exercises [52]. The complexity in the design of suitable scenarios is the result of the realism simulated in these exercises as well as the amount of in-depth domain knowledge needed. In functional exercises, a scenario only focuses on what affects a specific subfunction of the emergency response plan to be tested, while the disaster itself as well as difficulties and implications that are not related to the immediate subfunction are only mentioned in a narrative form. Difficulties related to the immediate domain might be simulated using controlled failures of hardware or fictional data from infrastructure. To convey a certain sense of urgency and to make the exercises seem more realistic, actors are often used simulating victims or other relevant entities like agency officials [31]. The training with real tools and equipment also helps deepen the immersion, but needs to be justified within the exercise goals.
- Scenarios for *full-scale exercises* are very elaborate, since they require multiple objectives to test most processes and subfunctions of the disaster response plan [29]. To enact a realistic setting, multiple responder agencies are required to participate with each of them having their own exercise objective while executing their functions in the disaster plan on the operational field in real time. As with functional exercises, realism is increased further by the use of actors, real tools, and equipment. Planning and conducting full-scale exercises is an enormous and expensive endeavor, demanding significant investments in terms of time and resources.

3.5 Characteristics of Scenarios and Events

We analyzed scenarios and their constituent events from disaster management guidelines [3, 50, 54] and expanded on their identified and extracted characteristics and properties using input from domain experts to obtain an abstract overview of the disaster management perspective to be used as basis for our mathematical modeling approach proposed in Sect. 4 of this work. We would like to highlight that the characteristics of scenarios and events listed below are general in nature and that their concepts occur within all types of exercises. However, the degree in which they are of consequence to a particular exercise can vary based on the type and purpose of the exercise in question.

Exercise scenarios can be seen as a combination of two components, the context and the crisis. The *context* gives a description of the environment the crisis takes place in. It may include background information such as the geographical setting as well as the demographic of a country, what organizations are involved, their structure, relationships, etc.

The context is the environmental framework into which many different crises might be embedded for study [28].

The *crisis* is the literal storyboard of the scenario—a script of crisis-events within a specific time-frame consisting of several types of events with different consequences [34]. This not only includes events happening during the exercise, but all

Table 1 Events occurring in our hypothetical flooding disaster

| | Name | Description |
|-------|-------------------|--|
| Setup | | Heavy rainfalls during the thawing season causing water levels of rivers to rise to a critical level |
| Event | Water_Rising | Water levels rises 1.5m |
| Event | Leakage | Leakage of flood control measures near a low density residential area |
| Event | Flooding | Flooding near critical infrastructure |
| Event | Structural_Damage | Structural damages to bridge vital to response efforts in the area affected |

events leading up to the crisis as well. In this paper, we refer to the former as *crisis events* and to the latter as *scenario setup events*.

We list the determined characteristics of events in Sect. 3.5.1 and of scenarios in their entirety in Sect. 3.5.2.

3.5.1 Characteristics of Events

As outlined before, events are the building blocks of scenarios, which make up exercises. For an entire exercise to satisfy certain properties, we must therefore take a look at the characteristics of individual events first. To motivate and illustrate the analyzed characteristics of events, we introduce at this point two specific sets of events corresponding to our two chosen hypothetical example disasters of a flooding and a cyber crisis due to a cyber attack on a power plant. The events considered in the flooding disaster are given in Table 1 and those in the cyber crisis in Table 2. It is important to note that the events used in these two hypothetical disasters are merely serving as lucid examples for possible event choices. In a real-world setting, when designing an exercise, the included events would of course be chosen by domain experts for the specific domain of the DUT (e.g., non-man-made natural disaster like flooding, landslide, volcanic eruption, pandemic, or man-made disasters like a cyber crisis).

Visibility of Events In crisis management, especially in the case of cyber-crises, not all events are immediately recognizable to the participants. This concerns not only

Table 2 Events occurring in our hypothetical cyber-attack crisis

| | Name | Description |
|-------|------------------|---|
| Setup | | By using a compromised email account and exploiting a vulnerability in an old program, an attacker successfully compromises the internal network of a power plant, gaining access to critical control systems |
| Event | PO_Drop | Drop in power-output level |
| Event | Malicious_SW | IT department discovers malicious attachments in emails sent from an employee |
| Event | Denying_Measures | Systems deny deployment of reactive measures |
| Event | Fire-Alarm | Fire alarm is set off at the turbine hall |

the crisis events, but also events of the scenario setup. In our examples, the setup for the cyber-attack scenario might not be fully visible to the participants right from the beginning. We define the visibility of an event as the potential of the participants to recognize the occurrence of an event in such a way that the participant's knowledge of its existence can force a direct or indirect reaction. A fully visible event is any event the participants can directly react to, whereas a partially visible event might be discovered as being something that happened due to visible repercussions but which is not fully identifiable. An invisible event is any event that the participants do not discover. Whether or not an event is visible to the participants is up to exercise designers.

Priority of Events It is vital for an exercise to reproduce certain characteristics of crises like stress or confusion. To this end, scenario designers might inject events into the scenario that have the sole purpose of distracting the participants or which force a decision process, where the participants have to re-evaluate their current approach. For instance, in the first of our sets of example events, we can see that event *Flooding*, which deals with occurrences of floodings near critical infrastructure, has more potential for damage than event *Leakage*, which concerns leakage of flood control measures near a low density residential area. Also, some of the events used in a disaster-exercise might be injected purely to create confusion or stress for the participants. In our stated example before of a cyber attack on a power plant, the motive behind the event *Fire-alarm* is to cause a distraction and to divert manpower to inspect and solve this issue. With that in mind, we can differentiate between key events which are responsible for critical and mandatory exercise goals, events of lesser importance that may provide optional objectives, and events used to either increase the workload and stress level of the participants or to simulate a real exercise environment.

Repeatability of Events The repetition of certain events in a scenario can not only affect the priority of the event itself, but can even have an effect on the overall impact of the disaster scenario. Additionally, the repetition of an event can be used to negate measures taken by the participants as well as to highlight insufficient ones. For example, in our set of events for a flooding scenario, a second appearance (i.e., repetition) of event *Water_Rising* could render measures already taken by the participants in response to the first appearance of this event insufficient. Although the repetition of events can help participants practice difficult tasks and procedures [55], depending on the exercise type and setting, scenario designers must be careful with its implementation in the scenario. While the repetition of minor events in functional and full-scale exercises might help with providing a more effective exercise scenario, the repetition of major events would require a lot of effort. Also, depending on the scenario setting, the repetition of major events might not always make sense from a storyline point of view.

Constraints An event might have an attribute of dependence with regards to another event in the scenario. For example, certain events can only happen in a fixed order,

as they are a prerequisite to other events. For example, in the case of our events for the cyber scenario, the event *Denying_Measures*, where the systems under attack deny the deployment of reactive measures, has to follow up another event and cannot be the first event of a scenario. Another example is given in the event list for our flooding scenario: here, event *Water_Rising* always precedes event *Leakage* since without the rise of water, the flood control measures already in place should be sufficient to hold off the water. Events belonging to the scenario setup might also be modeled as events with constraints fixing their position at the beginning of a scenario, since by design they have to occur before any crisis event.

3.5.2 Characteristics of Scenarios

The impact of a crisis not only depends on the individual events happening during the crisis and their characteristics, but also on how many events appear in total (i.e., including multiplicities), the order in which they happen during the crisis, and how they actually unfold. Therefore, the overall response strategy (i.e., regarding the distribution, allocation, and coordination of resources and emergency services and personnel) has to be adapted accordingly on a case-by-case basis. Similar considerations apply to the characteristics of scenarios in exercises, which we discuss below in detail.

Scenario Length The scenario length is given by the number of crisis events making up the exercise scenario. These events have to fit the exercise goals and can be extracted from either historical data or are set forth by domain experts. Here, the inclusion of specific events, which introduce crisis-characteristics like stress or uncertainty and which might force participants to select between different strategies in their crisis management approach, is one of the most effective methods in providing a realistic exercise experience. However, scenario designers must be careful to utilize this method in moderation, as exercises that are too extensive and feature too many different paths or non-essential objectives not only become very difficult to design, but also tedious to partake in and monitor, and, depending on the purpose of the exercise, can also have negative training effects as the implementation of new events may unintentionally shift the training goals in unwanted directions. In other words, the scenario length majorly determines the impact and focus of the resulting exercise.

Order of Events Some events might change or even amplify the damage of following events; therefore, the order in which disastrous events occur can change both the impact and potential for damages of the overall crisis. Also, while some events might not be problematic on their own, they could become devastating prefacing or following some specific event or specific sequence of events. Studying and assessing different sequences of disaster events is therefore vital for crisis management, since it may reveal major problems in disaster preparedness and response plans.

Unfolding of Events Just like the order in which crisis events happen can influence the severity of disaster scenarios, so can their temporal and geographic spread. For example, events happening long after each other may disturb an already in progress relief process which did not account for these specific events, whereas events occurring in rapid succession might overburden emergency services.

Similarly, crisis events happening geographically far apart from each other might lead to a long response time, whereas multiple events happening in very close vicinity to each other might increase their collective impact and overwhelm infrastructures or organizations aiding with relief efforts, like hospitals or emergency services.

Together with the scenario length, both the temporal and geographical spread have considerable influence on the overall duration of the scenario as well as the duration of the exercise.

4 Mathematical Problem Formulation

In this section, we describe our proposed mathematical approach based on discrete mathematics for modeling and generating disaster (event) scenarios in detail. We begin, in Sect. 4.1, with presenting suitable mathematical sequence structures and discuss those properties of them that will be leveraged for disaster event scenario modeling for exercises in this work. Next, in Sect. 4.2, we establish the link between the previously discussed combinatorial sequence structures and the modeling of disaster event scenarios for exercises.

4.1 Combinatorial Methods for Sequence Structures

In the following, we present certain combinatorial sequence structures arising in discrete mathematics, which have found real-world applications in (at least) the testing of event-based software systems. Since both event-based software systems and disaster scenarios have an intrinsic temporal nature with events potentially impacting their respective successors in the timeline, they both can be modeled with mathematical artifacts known as sequence covering arrays and related structures.

The notion of *sequences* permeates all branches of mathematics, with sequences appearing often and in various circumstances. Within discrete mathematics, the specific class of finite (empty or nonempty) sequences with elements taken from a fixed, finite, nonempty set is used explicitly or implicitly in many different ways, and this class has also received considerable attention due to its many applications to real-world problems in a myriad of diverse application domains. The important point that is paramount to highlight for all these real-world applications is that mathematical objects can be *instantiated* with many disparate real-world entities, and consequently, the inherent abstract mathematical properties of the considered mathematical objects or structures are passed on. These properties then appear again within the application domain, where they have instantiation-specific impacts and implications depending on the considered real-world application domain or use-case, which are

subsequently used to *reason about or—in the best case even—solve* the considered real-world problems.

4.1.1 Sequence Covering Arrays

We motivate and explain the definition and structural properties of certain combinatorial sequence structures with their use in the application domain of combinatorial sequence testing for software. In the modern interconnected computing landscape of today,² devices of any kind have to be able to receive, parse, and react to external events, as well as to trigger events themselves based on their own tasks. In particular, the current state of a device and the actions taken by it often depend on (or are the result of) the specific history of prior events pertaining to that device before. Similarly, if a failure manifests itself in a device, this failure can also be the direct result of the history of events that the device has experienced up to this point in time. Hence, in this setting, the temporal orders in which events appear determine significantly whether some specific, critical condition or state is attained so that at least one failure manifests itself. This means that these critical conditions or states, which lead to at least one failure manifesting itself, arise as the direct result of whether some event (or events) has (have) taken place prior to some other event (or events) or not.

The previously described conceptual connection between specific temporal orderings of events and failures in an event-driven software system has been empirically observed in [56] for the specific case of non-repeating events and analyzed from a software testing perspective, where the authors also proposed a corresponding combinatorial testing method including an algorithm, which ensures that any t events will be tested in every possible t -way order at least once in a constructed test set by that algorithm. The event sequence software testing problem considered in [56] dealt with the implications of the order of connecting many peripherals to a system and the resulting failures that manifested themselves in this system for some orderings. The failures that were observed can be exemplary illustrated as follows: a failure might manifest itself when connecting peripheral B before peripheral A is fully connected, or another failure might manifest itself only when peripherals B and C are both already connected when trying to connect peripheral A.

When testing such event-driven systems, it is therefore vital to assess (i.e., test) whether specific orderings of events—as part of event sequences—will lead to failures manifesting themselves in a system or not. It has been a key observation in [56] that combinatorial methods originating in discrete mathematics can be used to efficiently address such non-repeating event sequence testing problems. The abstract mathematical artifact proposed to this end in [56], from which the developed so-called t -way event sequence test sets are derived, is called a *sequence covering array* (SCA). This structure exhibits full t -way permutation coverage for all t -combinations from the set of all considered events for some fixed, but arbitrarily choosable, non-zero

² As of this writing.

natural number t less or equal to the cardinality of the set of all considered events. The formal definition in the sense of [56] is as follows:

Definition 1 For³ given finite nonempty set S of symbols of cardinality $|S| = s \in \mathbb{N}^\times$; $N, t \in \mathbb{N}^\times$ with $t \leq s$; a *sequence covering array* (SCA) of strength t is an $N \times s$ array with entries from the set S , such that every t -way permutation of symbols from S occurs in at least one row and each row is a permutation of the s symbols. The t symbols in the permutation are not required to appear adjacent to each other. That is, for every t -way arrangement⁴ $\langle x_1, \dots, x_t \rangle$ of pairwise distinct symbols x_1, \dots, x_t from the set S , the regular expression $\cdot * x_1 \cdot * x_2 \cdots \cdot * x_t \cdot *$ matches at least one row in the array.

For any given finite nonempty set S of symbols of cardinality $\mathbb{N}^\times \ni s > 0$ and $t \in \mathbb{N}^\times$, $t \leq s$, there always exists at least one SCA of strength t for the symbol set S , but this object is not necessarily unique. To see this, consider an array representation of all permutations of the set S , while observing that this array will not only cover all possible s -way permutations of S , but also all τ -way permutations of S for all $\tau \in \mathbb{N}^\times$ with $\tau \leq s$. For this reason, the construction of SCAs with as few rows as possible is of high interest both from a theoretical packing-optimization as well as from an applied resource-minimization perspective. In [56], it has been shown that the actual construction of SCAs in practice can be achieved with greedy algorithms and that the size of generated SCAs (i.e., the number of test sequences in a SCA) grows logarithmically in the number of events. Note that for any given finite nonempty set S , a SCA of strength two with two rows can be directly constructed by using any enumeration of the events in the set S in any order as first row and this enumeration in reversed order as second row. Then, any ordered selection of any two events from the set S (i.e., an ordered pair from the set S) will either appear within the first or second row of the constructed array; thus, it provides full 2-way (i.e., pairwise) permutation coverage and additionally also has optimal size (i.e., minimum number of rows).

When SCAs are used as sequence test sets in an applied setting for the (event) sequence testing of a system, their symbol set is usually referred to as the *set of events*. For a given sequence testing problem with nonempty event set \mathcal{E} of cardinality $\mathbb{N}^\times \ni \mu > 0$ and $t \in \mathbb{N}^\times$ such that $t \leq \mu$, each row of a SCA of strength t with symbol set \mathcal{E} individually encodes exactly one test sequence and all rows of the SCA collectively guarantee 100% t -way permutation coverage with regard to the set \mathcal{E} according to the definition of a SCA. This has as consequence that once all sequence test cases derived from the rows of the SCA have been successfully handled by the system when performing the actual test execution for the given sequence

³ We denote the set of natural numbers, under which we understand in this work the non-negative integers, with the symbol \mathbb{N} and use the symbol \mathbb{N}^\times to denote the positive natural numbers.

⁴ Finite, nonempty sequences are denoted in ascending order between angle brackets, e.g., $\langle 0, 1, 2 \rangle$, for notational accentuation of the importance of the given *order* of elements in the sequence. This notation is also further used to describe a single permutation consisting of pairwise distinct elements.

Table 3 SCA of strength two for the symbol set $\{A, B, C, D\}$ of cardinality four (taken from [56])

| | | | | |
|---------------|---|---|---|---|
| Test 1 | A | B | C | D |
| Test 2 | D | C | B | A |

Table 4 SCA of strength three for the symbol set $\{A, B, C, D\}$ of cardinality four (taken from [56])

| | | | | |
|---------------|---|---|---|---|
| Test 1 | A | D | B | C |
| Test 2 | B | A | C | D |
| Test 3 | B | D | C | A |
| Test 4 | C | A | B | D |
| Test 5 | C | D | B | A |
| Test 6 | D | A | C | B |

testing problem, then there is no failure that can be triggered by some not necessarily adjacent order of any t or fewer events; simply because if there were one, at least one test sequence in the SCA—due to its complete t -way permutation coverage—would have triggered it, meaning that the failure would have manifested itself as the result of executing this test sequence.⁵ In this sense, t -way sequence testing is able to expose all failures that manifest themselves as the result of the interaction between at most t events in some specific not necessarily adjacent order. Therefore, the testing of event-based systems with SCAs can be used to obtain quantitative assurance guarantees.

In Tables 3 and 4, we showcase SCAs for the symbol set $\{A, B, C, D\}$ of cardinality four for strengths two and three, respectively. These two example SCAs can still be verified by hand to indeed constitute SCAs in manageable time-wise effort by checking among other properties that in particular all required 2-way and 3-way permutations actually appear within the given arrays. Compared to an exhaustive 4-way sequence test set, the given SCAs of strength two and three achieve a reduction in the number of test sequences of $1 - \frac{2}{24} = 91.7\%$ and $1 - \frac{6}{24} = 75\%$, respectively, illustrating the efficiency gains that can be attained when using SCAs as part of the practical sequence testing of real-world systems.

4.1.2 Adding Constraints to Test Sequences

The permutation-based sequence testing approach presented previously had been developed for a specific applied sequence testing problem dealing with the order of connecting peripherals and proved to be successful for this specific sequence testing problem (see [56]). However, it turned out that the underlying permutation model of SCAs is quite restrictive as a formal basis when considering the use of SCAs in general sequence testing problems. Consequently, other sequence

⁵ Within the scope of this work, we assume that all considered *systems* for testing are deterministic and that all failures occurring during the execution of sequence test cases are actually observable by the experimenters.

Table 5 Description of sequencing constraints given in [57]

| Sequencing operator | Explanation |
|----------------------|---|
| $_e$ or (e) | e always happens |
| $e_1^* - e_2$ | If e_1 happens, then e_2 must immediately happen after e_1 |
| $e_1 -^* e_2$ | If e_2 happens, then e_1 must immediately happen before e_2 |
| $e_1 \sim e_2$ | e_2 never immediately happens after e_1 (or e_1 never immediately happens before e_2) |
| $e_1^{\dots*} e_2$ | If e_1 happens, then e_2 must happen after e_1 but not necessarily immediately happen after e_1 |
| $e_1^{\dots*} e_2$ | If e_2 happens, then e_1 must happen before e_2 but not necessarily immediately happen before e_2 |
| $e_1 \cdot \sim e_2$ | e_1 never happens before e_2 (or e_2 never happens after e_1) |

testing approaches, which are still centered around some variant of the notion of full t -way coverage, have been proposed. Of particular relevance to this work is the approach presented in [57] proposing t -way sequence testing with constraints together with a corresponding t -way test sequence set generation algorithm, which makes use of a set of events and a set of constraints. The developed notation for specifying constraints in [57] supports three types of constraints: repetition, length, and sequencing constraints. In particular, the approach to sequence testing with constraints given in [57] does not necessarily require—in contrast to the one given in [56]—that individual test sequences are permutations, thus enabling and providing much more structural flexibility for modeling and—as a result—in the generated sets of t -way test sequences. A repetition constraint denotes an upper limit on the number of times an event can appear in a test sequence, but can also be formulated as a default constraint meaning that the upper limit applies to all events in the sequence. A length constraint determines the minimum and maximum length of a test sequence. Note that since every test sequence should cover at least one target sequence of length t , the minimal length for a test sequence must therefore also be t . Sequencing constraints are used to restrict or enforce the order in which events appear within test sequences and are built using sequencing and Boolean operators. For the sake of completeness, we reproduce the description of sequencing constraints given in [57] in Table 5.

The introduced approach to sequence testing with constraints in [57] entailed a necessary change in the objective of the advocated criteria of *full coverage* from

Table 6 Excerpt of test sequences of length four as part of a 3-way test set with constraints

| | | | | |
|----------------|----|----|----|----|
| Test 1 | A | A | B | F |
| Test 2 | A | A | E | F |
| ... | .. | .. | .. | .. |
| Test 20 | C | D | B | F |
| Test 21 | C | D | E | F |
| ... | .. | .. | .. | .. |
| Test 29 | E | E | A | F |
| Test 30 | E | E | B | F |

Table 7 Excerpt of test sequences of length five as part of a 3-way test set with constraints

| | | | | | |
|----------------|----|----|----|----|----|
| Test 1 | A | A | B | E | F |
| Test 2 | A | A | C | D | F |
| ... | .. | .. | .. | .. | .. |
| Test 27 | B | C | D | A | F |
| Test 28 | B | C | D | B | F |
| ... | .. | .. | .. | .. | .. |
| Test 38 | E | C | D | B | F |
| Test 39 | E | C | D | E | F |

the notion of full t -way *permutation coverage* given in [56] to full t -way *target sequence coverage* given in [57]. A t -way (event) target sequence is defined in [57] as an event sequence of length t , which can be *covered* (i.e., its elements appearing in the same order, but not necessary element-wise consecutively) within a valid test sequence (i.e., a test sequence satisfying all the given constraints) that can be executed by the system. The accompanying actual sequence test set generation approach given in [57] also follows a greedy strategy, where starting from the empty set, each newly added test sequence to the already existing sequence test set provides the maximum t -way target sequence coverage increase, until all required t -way target sequences with respect to the given constraints are covered in the constructed sequence test set.

To make the notion of full t -way target sequence coverage more tangible, we present a concrete example for a t -way test set with constraints, generated using a tool from the authors of [57]. The events considered in this example are given by the set $E = \{A, B, C, D, E, F\}$ of cardinality six, and we fix the strength to be three, i.e., we require full 3-way target sequence coverage. Additionally, all the generated sequences must satisfy the following constraints:

1. The length of every sequence has to be either four, five, or six.
2. Every sequence has to end with the event F .
3. All events may happen up to 2 times during a sequence except event F , which may only happen once.
4. Events C and D may only occur together, and C must immediately happen before D .

Table 8 Excerpt of test sequences of length six as part of a 3-way test set with constraints

| | | | | | | |
|----------------|----|----|----|----|----|----|
| Test 1 | A | A | B | C | D | F |
| Test 2 | A | A | E | B | E | F |
| ... | .. | .. | .. | .. | .. | .. |
| Test 17 | B | C | D | A | B | F |
| Test 18 | B | C | D | C | D | F |
| ... | .. | .. | .. | .. | .. | .. |
| Test 24 | E | C | D | B | A | F |
| Test 25 | E | C | D | C | D | F |
| Test 26 | C | D | C | D | B | F |
| Test 27 | C | D | C | D | E | F |

All these four constraints can be constructed by combining appropriate sequencing constraints listed in Table 5 together with Boolean operators. In total, the tool generated 96 test sequences of different length, which cumulatively cover every 3-way target sequence of events while also satisfying the constraints listed above.

We display in Tables 6, 7, and 8 excerpts from the generated sequences of length four, five, and six, respectively. In detail, the tool generated 30 sequence tests of length four covering 52 3-way target sequences, 39 test sequences of length five covering 113 3-way target sequences, and 27 sequence tests of length six covering 143 3-way target sequences.

It is important to note that although overall all events in the set E appear in at least one generated test sequence of some length, not all generated sequences contain every element of the set E . This not only holds—obviously—for sequences of length four and five, but also for some of the generated sequences of length six, since neither of the sequences $\langle C, D, C, D, E, F \rangle$ and $\langle E, C, D, C, D, F \rangle$, which occur both within the generated sequences of length six (as Test 27 and Test 25 in Table 8, respectively), contain the elements A or B .

Furthermore, it is also noteworthy that a 3-way target sequence can be covered by multiple generated sequences. In this example, this is the case for the 3-way target sequence $\langle C, D, B \rangle$, which is covered by the sequences $\langle E, C, D, B, A, F \rangle$, $\langle C, D, C, D, B, F \rangle$ as well as sequence $\langle B, C, D, A, B, F \rangle$ (appearing as Test 24, Test 26, and Test 17 in Table 8, respectively) in the sense of not necessarily adjacent order coverage.

This example demonstrates that t -way test sequence sets with constraints may contain sequences of different length, provide more flexibility than SCAs for elements to appear multiple times or not at all within one test case, and satisfy additional constraints among the elements of their sequences. In particular, for any given nonempty set of constraints, it is a priori not clear whether there are any valid sequences at all, i.e., whether this set of constraints is satisfiable, in contrast to the universal existence of SCAs.

4.2 Combinatorics for Disaster Scenario Generation

In this section, we link the scenario generation for disaster exercises to the problem of constructing certain mathematical artifacts with specific combinatorial (sequence) coverage properties. We further explain how the characteristics of disaster events and scenarios from the disaster domain manifest themselves through our proposed linking in properties of the employed mathematical structures and vice versa.

To give a brief overview before we present the details of our proposed mapping below, recall that exercises are proactive measures consisting of one or more scenarios which participants from one or more organizations take part in. In the scenarios themselves, the scenario context as well as the setup events collectively comprise the background information regarding the surrounding circumstances in which the crisis events are supposed to occur. The actual disaster or crisis depicted by a scenario is then given by a finite nonempty (temporal) sequence of crisis events.

These crisis events, which are chosen by the exercise organizers with possible consultation with domain experts, and—in particular—their granularity can depend

on the type of the exercise conducted. However, our proposed mapping is independent of these semantic considerations, since it operates only on the set of selected crisis events, independent of how they were determined in the first place. The finite nonempty set of selected crisis events together with an appropriately chosen value $t \in \mathbb{N}^{\times}$ will give rise to a certain mathematical artifact (e.g., SCA), with each of its elements (e.g., single rows in the case of a SCA) encoding the temporal crisis event ordering of one scenario of the disaster exercise generated via combinatorial methods. In other words, the constructed structure as a whole corresponds to an entire exercise, while the elements of the constructed structure correspond to the individual scenarios an exercise is composed of. Note that in the simplest form of our approach, it is only necessary to specify a set of crisis events and the desired strength of t -way permutation coverage to generate a combinatorial disaster exercise.

The usage of mathematically designed test plans is central in the field of DoE which is a well-established and recognized branch within statistics and which enables experimenters to obtain scientific objectivity for conclusions drawn based upon experiments. For over a century now, designed experiments have been successfully applied in various industries, ranging from agriculture, manufacturing, service, and even governmental sectors. The availability of end-user software with a focus to be usable for the knowledgeable, non-expert user supported the now wide-spread usage of DoE methods. The increasing accessibility to computing power and data analytic capabilities in the last years has led to several software companies developing dedicated statistical software, aiming to provide easily accessible statistical methods to non-expert end users [58, 59]. These software products support conducting designed experiments from start to finish by providing the end-user with an (text or graphical) interface to specify parameters, values, constraints, and goal of the experiment and then supply to the user a corresponding experimental design, which can be—among others—a (fractional) factorial design, an orthogonal array, or a response surface design, depending on the application need and goal. Similarly, the software can provide the user with a detailed analysis of the results of the experiments, often focusing on the visual exploration of the obtained results. Such end-user focused software has also been developed for CT, for example, *IBM Functional Coverage Unified Solution* [60] or the software offered by the company Hexawise [61]. The proposed disaster exercise generation methodology described in this paper is currently only implemented in research prototype tools, but we have already presented a holistic overall conceptual framework for disaster exercises in [62] comprising the modeling, generation, and post-analysis of conducted exercises. In order to enable low-barrier access and easy usage of *combinatorial sequence testing methods* within various disaster management organizations, we envision a future implementation of this framework—containing, in particular, an implementation of the methodology proposed in this work—in an end-user focused software solution, combining concepts of situational awareness dashboards and statistical experiments software. We further envision that these efforts will lead to a single capable software solution providing not only scenario generation for exercises, but also comprehensible feedback loops for exercise managers through utilization of interactive visualizations of scenario parameters and exercise progress. Hence, disaster response organizations will only have to deal with one software and will be able to focus on leveraging

Table 9 Mapping of terminology between disaster exercise and combinatorial sequence testing domains

| Disaster exercise domain | Combinatorial sequence testing domain |
|---------------------------------------|--|
| Exercise | Sequence test set (i.e., SCA of strength t or t -way sequence test set with constraints) |
| Exercise complexity | Strength t |
| Scenario | Element in sequence test set |
| Scenario length | Length of test sequence |
| Story | t -way permutation or target sequence of length t |
| Events | Event symbols |
| Multiplicities of events in scenarios | Repetition of event symbols in sequences by constraints |
| Constraints between events | Constraints between event symbols |

mathematically designed disaster exercises as part of their operational training use case for disaster preparation.

A high-level overview of the mapping between the terminology used in the domain of disaster exercises and the domain of combinatorial sequence testing is provided in Table 9. Note that in our proposed linking, we do not consider setup events when enforcing certain notions of coverage. Below, we give the details of our proposed linking, i.e., how we map the discussed concepts from the domain of disaster exercises to concepts in combinatorial sequence testing.

| | |
|----------------------|---|
| Exercise: | In the domain of disaster research, a crisis exercise is a proactive measure consisting of one or more exercise scenarios which are used to train emergency response personnel or assess and evaluate (existing or novel) response strategies. We map the concept of a disaster exercise to a sequence test set derived from a combinatorial design, which can—for instance—be a SCA or sequence test set with constraints. |
| Exercise complexity: | The complexity of disaster exercises is reflected in our mapping in the strength of the generated sequence test set in the domain of combinatorial sequence testing. |
| Scenario: | In disaster management, exercises consist of scenarios, which are sequences of events to which the participants in an exercise must respond. Since entire exercises map to sets of test sequences, it follows naturally that individual scenarios of exercises map to the individual sequences (i.e., members) in sequence test sets. |
| Scenario length: | For both the disaster management domain and the combinatorial sequence testing domain, the length of a disaster scenario, respective a test sequence, is given by the number of its elements (including multiplicities). Based on this consistency, our proposed mapping is naturally extended to relate the concept of length in both domains. |

- Story:** The order of events in a disaster scenario can have significant impact on the severity of the outcome of a crisis: Even usually harmless events might entail serious ramifications when sequentially arranged in an unfavorable order. In the disaster exercise domain and from a high-level point of view in abstract terms, we define the notion of a story as a *specific (temporal) unfolding of some crisis events as part of how a DUT may temporally manifest itself as a whole*. It follows that one individual scenario as part of a disaster exercise can have one or more stories embedded into it when crisis event successions from the respective stories are interleaved within the scenario (i.e., the ordering of the crisis events of each story is considered in not necessarily adjacent order as part of the overall scenario). Observe that in a disaster exercise consisting of multiple scenarios, some stories might appear more than once in total as part of different individual scenarios of the disaster exercise.
- In the abstract mathematical domain of finite, nonempty sequence modeling, the notion of a subsequence of a sequence provides a precise concept to assess sets of sequences⁶ for their coverage of certain event orderings for some events in the not necessarily adjacent sense (i.e., subsequences of sequences in the considered set).
- These two notions match conceptually very well, and therefore, we link stories from the disaster exercise domain to (sub-) sequences in the combinatorial sequence testing domain. Accordingly, the embedded stories within an individual disaster exercise scenario in the disaster exercise domain correspond to covered event-subsequences of an individual sequence test case in the combinatorial sequence testing domain. Note that there exist different variants of the notion of (*t*-way) sequence coverage in the domain of combinatorial sequence testing (e.g., *t*-way permutation coverage or *t*-way target sequence coverage).
- Repeating events:** A specific event can appear multiple times in a disaster scenario, and this multiplicity can be expressed in the combinatorial sequence testing domain with event repetition constraints for suitable combinatorial sequence structures.
- Constraints:** Constraints formulated in the disaster domain can be modeled in the combinatorial sequence testing domain as constraints for compatible combinatorial sequence structures (for example, such as those featured in [57]).

⁶ Note that single sequences appear as the special case of a singleton set.

5 Examples for Combinatorial Disaster Exercises

In this section, we present two illustrative examples for how we propose to use combinatorics-inspired automated generation of test sequences to derive multiple disaster scenarios which will collectively fulfill certain combinatorial coverage requirements. We also indicate how the generated sets of test sequences would be used in practical emergency drills or planspiels. Specifically, depending on the extent of the exercise and the number of scenarios generated, as well as industry and stakeholder requirements, we suggest splitting the scenarios of one exercise up into multiple batches of scenarios. This would not only help with the execution of exercises featuring a large amount of scenarios, but also allow supervisors to analyze the progress made and if deemed necessary add some adjustments to the successive batches. We state these two examples for the underlying combinatorial structure of a SCA, which we use in Sect. 5.1 for the case of a flooding disaster of strength three and in Sect. 5.2 for a cyber crisis of strength two. Additionally, we provide sets of disaster scenarios, which are constructed featuring sequential and repetition constraints, for both examples. These scenarios have been generated by using a tool presented in [57], implementing the framework introduced in the same work.

5.1 Flooding

This example is concerned with the flooding disaster mentioned before in Sect. 3.5. Assume a situation in which a recent natural disaster threat analysis has pointed to increasing risk of damages caused by floods in some area. The local government wants to be able to appropriately react and enhance its preparation efforts. To that end, first responder organizations and other stakeholders are gathered for creating a committee to design a disaster exercise of type tabletop. Domain experts are tasked with generating a list of relevant events, which we assume are those given in Table 1. In this example, upon the actual execution of the exercise, the setup events are fully visible, not only to the exercise lead, but to the participating emergency response personnel as well.

In the following, we present two ways how combinatorial methods can be used to create disaster exercises.

5.1.1 Exercise via a SCA

Assume that it was decided that this exercise should be derived from a SCA of strength three, hence having complexity three. The events that have been identified by the domain experts (see Table 1) are regarded as an abstract symbol set, which, together with the desired complexity three of the exercise, constitute all the information required to construct a SCA of strength three which is given in Table 4. This combinatorial object is then used to derive the individual scenarios of the exercise, which are given by the individual rows (i.e., test sequences) of the SCA. The scenarios generated for this tabletop disaster exercise by a SCA are depicted in Table 10.

Table 10 Flooding disaster exercise plan derived from a SCA of strength three

| | | | | |
|-------------------|-------------------|-------------------|----------|-------------------|
| Scenario 1 | Water_Rising | Structural_Damage | Leakage | Flooding |
| Scenario 2 | Leakage | Water_Rising | Flooding | Structural_Damage |
| Scenario 3 | Leakage | Structural_Damage | Flooding | Water_Rising |
| Scenario 4 | Flooding | Water_Rising | Leakage | Structural_Damage |
| Scenario 5 | Flooding | Structural_Damage | Leakage | Water_Rising |
| Scenario 6 | Structural_Damage | Water_Rising | Flooding | Leakage |

All these scenarios have length four and collectively by construction provide full story coverage for any three distinct events. The exercise consists of six scenarios, and since SCAs were selected as underlying combinatorial sequence structure, in each obtained scenario, each event appears exactly once, and there are no constraints.

The exercise plan will be played in three batches of two scenarios per day, with 1 week between the batches. Depending on how well the participants perform in the execution of the exercise, it might be recreated with increased complexity four in case major difficulties are uncovered or be repeated after 1 year with only complexity two when it is clear that the participants are already sufficiently skilled in dealing with the presented scenarios.

5.1.2 Exercise via a Sequence Test Set with Constraints

Assume now that it was decided that this exercise should instead be derived from a sequence test set with constraints and that the following rules and conditions were considered: First, we will establish that a scenario may not feature less than 3 or more than 4 events and that this disaster exercise should also be of complexity three. We will enforce a repetition constraint, which states that all events may happen only once during a sequence, except the event *Water_Rising*, which may occur indefinitely often. We will also establish that in each sequence, at least one of the events *Leakage*, *Flooding*, or *Structural_Damage* must occur at least once. Finally, we will declare that all sequences must begin with the event *Water_Rising*.

The resulting individual scenarios, when translating the obtained sequence test set with constraints to a disaster exercise, are given in Table 11. As can be seen there, the exercise plan produced by the tool consists of 23 test scenarios, which will be played over a time-frame of 4 weeks, playing 2 scenarios per day for 3 days each week. All of the scenarios are of length four and satisfy the repetition and sequencing constraints given above by construction while also collectively fully covering all 3-way target sequences of events.

5.2 Cyber Attack

Cyber disasters are a serious and real threat to a large variety of industries, especially critical infrastructure organizations, with the focus of our example from Sect. 3.5 being that of a power plant. In order to ensure that the employees of such

Table 11 Flooding disaster exercise plan derived from a three-way test set with constraints

| | | | | |
|--------------------|--------------|-------------------|-------------------|-------------------|
| Scenario 1 | Water_Rising | Water_Rising | Water_Rising | Leakage |
| Scenario 2 | Water_Rising | Water_Rising | Flooding | Structural_Damage |
| Scenario 3 | Water_Rising | Leakage | Water_Rising | Flooding |
| Scenario 4 | Water_Rising | Leakage | Structural_Damage | Water_Rising |
| Scenario 5 | Water_Rising | Flooding | Water_Rising | Leakage |
| Scenario 6 | Water_Rising | Structural_Damage | Leakage | Flooding |
| Scenario 7 | Water_Rising | Leakage | Water_Rising | Water_Rising |
| Scenario 8 | Water_Rising | Leakage | Water_Rising | Structural_Damage |
| Scenario 9 | Water_Rising | Leakage | Flooding | Water_Rising |
| Scenario 10 | Water_Rising | Leakage | Flooding | Structural_Damage |
| Scenario 11 | Water_Rising | Leakage | Structural_Damage | Flooding |
| Scenario 12 | Water_Rising | Flooding | Water_Rising | Water_Rising |
| Scenario 13 | Water_Rising | Flooding | Water_Rising | Structural_Damage |
| Scenario 14 | Water_Rising | Flooding | Leakage | Water_Rising |
| Scenario 15 | Water_Rising | Flooding | Leakage | Structural_Damage |
| Scenario 16 | Water_Rising | Flooding | Structural_Damage | Water_Rising |
| Scenario 17 | Water_Rising | Flooding | Structural_Damage | Leakage |
| Scenario 18 | Water_Rising | Structural_Damage | Water_Rising | Water_Rising |
| Scenario 19 | Water_Rising | Structural_Damage | Water_Rising | Leakage |
| Scenario 20 | Water_Rising | Structural_Damage | Water_Rising | Flooding |
| Scenario 21 | Water_Rising | Structural_Damage | Leakage | Water_Rising |
| Scenario 22 | Water_Rising | Structural_Damage | Flooding | Water_Rising |
| Scenario 23 | Water_Rising | Structural_Damage | Flooding | Leakage |

an organization have the ability and experience in dealing with various crisis situations, it is necessary to have them participate regularly in specific crisis exercises. Depending on the field of work and organizational level of the trainees, these exercises can range from tabletop exercises to functional exercises and can consist of multiple training sessions with each session playing through one or more scenarios. In this example, we will focus on a set of scenarios for the use in functional exercises. For this matter, we assume that domain experts have created a list of relevant exercise events, which are given in Table 2. Considering the properties and cyber-physical environment of this exercise, we will determine that the setup events are only known to the supervisors and that the trainees are only confronted with the actual crisis events.

In the following, we again present two ways for the structured generation of scenarios for exercises utilizing combinatorial methods.

5.2.1 Exercise via a SCA

Assume that a functional exercise used for lower-level training against cyber attacks, derived from a SCA of a certain strength t , with t representing the complexity of

Table 12 Cyber disaster exercise plan derived from a SCA of strength two

| | | | | |
|-------------------|------------|------------------|------------------|------------|
| Scenario 1 | PO_Drop | Malicious_SW | Denying_Measures | Fire-Alarm |
| Scenario 2 | Fire-Alarm | Denying_Measures | Malicious_SW | PO_Drop |

the exercise, should be conducted. We further assume that it was decided that the exercise should have complexity two and that exercise designers have created a list of relevant exercise events which are given in Table 2. By deriving the exercise from a SCA of strength 2, specifically from the one given in Table 3, which per definition covers all 2-way permutations of any two unequal events listed in Table 2, we obtain in total the two scenarios provided in Table 12 that the employees should be trained on. During the regularly scheduled training exercises, the participating employees are presented with the events from the currently played scenario one after the other with some time in between, giving them an opportunity to reevaluate their chosen strategy and react to the occurring problems. Actions taken by the participants during the exercise are documented and can be seen immediately by the supervisors. Depending on the decisions and actions of the participants, the supervisors can guide the trainees in desired directions.

5.2.2 Exercise via a Sequence Test Set with Constraints

For exercise scenarios regarding a cyber attack on a power plant featuring constraints, we will enforce the following rules and conditions: First, we will establish that a scenario may not feature less than 3 or more than 4 events. Similar to the previously generated scenarios utilizing SCAs, the scenarios with constraints will also feature a complexity of 2. Regarding possible repetitions of events, we will establish that all events except one might occur up to two times during a scenario. The event *PO_Drop* is the exception to this constraint as it may only happen once. Furthermore, we will establish that either one of the events *PO_Drop* or *Fire-Alarm* has to happen during a scenario as they are the driving force behind it. Additionally, the event *Denying_Measures* may not happen as the first event of the sequence but may only happen after either the event *PO_Drop* or *Fire-Alarm*, since it constitutes an event following actions from the participants.

The resulting individual scenarios, when translating the obtained sequence test set with constraints to a disaster exercise, are given in Table 13. The obtained exercise consists of a total of 6 scenarios with each one being of length 4. This exercise

Table 13 Cyber disaster exercise plan derived from a two-way test set with constraints

| | | | | |
|-------------------|--------------|------------------|------------------|------------------|
| Scenario 1 | PO_Drop | Malicious_SW | Denying_Measures | Fire-Alarm |
| Scenario 2 | Malicious_SW | PO_Drop | Malicious_SW | Denying_Measures |
| Scenario 3 | Fire-Alarm | PO_Drop | Malicious_SW | Denying_Measures |
| Scenario 4 | Fire-Alarm | Fire-Alarm | PO_Drop | Malicious_SW |
| Scenario 5 | PO_Drop | Denying_Measures | Malicious_SW | Denying_Measures |
| Scenario 6 | Fire-Alarm | Denying_Measures | PO_Drop | Malicious_SW |

can now be integrated into a long-term security training cycle which considers the overall safety and security of the plant and where different types of crisis exercises are played through every month.

6 Conclusion

In this paper, we have proposed the use of combinatorial methods for the modeling and automated generation of scenarios for disaster exercises with the underlying objective that the resulting guaranteed fulfillment of different notions of combinatorial sequence coverage in generated exercises reveals weaknesses or even failures in responding strategies of affected stakeholders. Our approach is inspired by combinatorial testing for software, and we have expounded on how to link concepts from the disaster management domain to properties of combinatorial sequence structures. Both the field of software testing as well as disaster and crisis management benefit from several attributes of the leveraged combinatorial structures such as automated construction approaches, guaranteed combinatorial coverage, and minimization of the size of generated structures (i.e., translating to the number of tests or scenarios, respectively). The two combinatorial sequence structures considered in this work—SCAs and sequence test sets with constraints—have distinct intrinsic properties, and for both, there are tools available which implement algorithms for their automated construction. Their different properties offer flexible and fine-grained modeling capabilities to exercise designers and are also reflected in generated disaster exercises. The desired strength of coverage for the respective notion of coverage provided by the two considered structures can be freely chosen by exercise designers and also adjusted over time to account for changes in training requirements. We illustrated our proposed approach for disaster exercise design and generation with two synthetic examples consisting of the case of a non-man-made disaster given by a flooding and the case of a man-made disaster in the form of a cyber crisis in a critical infrastructure. For both cases, we indicated how to potentially identify events together with constraints and showcased how combinatorial construction approaches could create disaster exercises based upon them. We also suggested how the obtained exercises could be conducted in practice by the respective stakeholders in a real-world setting. Combinatorial methods expand modeling approaches, enable automation in generation of individual scenarios as well as complete exercises, and increase the efficiency of exercise design and generation by reducing the overall organizational cost for disaster management. Hence, the integration of combinatorial methods advances the domain of disaster management.

Our presented approach is conceptual in nature and can be extended in multiple ways in future work. First, we want to evaluate our proposed approach in practice for different kinds of disasters and affected stakeholders. Second, we are interested in broadening our combinatorial modeling and generation approach by analyzing more combinatorial sequence structures for usage in disaster scenario generation. In particular, we expect that real-world evaluations of our approach could indicate how to extend or adapt our considered and employed combinatorial structures, for example, with regard to prioritization of scenarios within generated exercise plans, since, in a

real-world setting, resource-constraints might make it impossible to execute generated exercise plans entirely. Third, we see potential benefits in the implementation of our approach in a user-friendly software solution for disaster exercise management. Fourth, the precise mathematical coverage guarantees of disaster exercises generated via our proposed approach could be integrated into general vulnerability, risk, and impact analysis assessments of disaster threats.

Funding SBA Research (SBA-K1) is a COMET Center within the framework of COMET - Competence Centers for Excellent Technologies Program and funded by BMK, BMDW, and the federal state of Vienna. The COMET Program is managed by FFG.

Moreover, this work was performed partly under the following financial assistance award 70NANB21H124 from U.S. Department of Commerce, National Institute of Standards and Technology.

Data Availability The datasets generated during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of Interest The authors declare no competing interests.

References

1. Sommer S, Pearson C (2011) Infusing creativity into crisis management: an essential approach today. *Organ Dyn* 40
2. Benfield A (2021) Weather, climate & catastrophe insight. 2020 Annual Report. Aon Benfield. <https://www.aon.com/global-weather-catastrophe-natural-disasters-costs-climate-change-2020-annual-report/index.html>
3. Department of Homeland Security. Homeland Security Exercise and Evaluation Program (HSEEP). <https://www.fema.gov/sites/default/files/2020-04/Homeland-Security-Exercise-and-Evaluation-Program-Doctrine-2020-Revision-2-2-25.pdf>. Accessed 6 Jun 2021
4. Automated risk scenario generation using system functional and structural knowledge, volume Engineering/Technology Management of ASME International Mechanical Engineering Congress and Exposition (2005)
5. Martin G, Schatz S, Hughes C, Nicholson D (2010) What is a scenario? Operationalizing training scenarios for automatic generation. *Int J Cogn Ergon* 746–753
6. Martin G, Schatz S, Bowers C, Hughes CE, Fowlkes J, Nicholson D (2009) Automatic scenario generation through procedural modeling for scenario-based training. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol 53. SAGE Publications Sage CA: Los Angeles, CA, pp 1949–1953
7. Martin GA, Hughes CE (2010) A scenario generation framework for automating instructional support in scenario-based training. In: *Proceedings of the 2010 Spring Simulation Multiconference, SpringSim '10*, San Diego, CA, USA. Society for Computer Simulation International
8. Campos J, Arcuri A, Fraser G, Abreu R (2014) Continuous test generation: enhancing continuous integration with automated test generation. In: *Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering*. pp 55–66
9. Candea G, Godefroid P (2019) Automated software test generation: some challenges, solutions, and recent advances. In: *Computing and Software Science*. Springer, pp 505–531
10. Kacker RN, Kuhn DR, Lei Y, Simos DE (2021) Factorials experiments, covering arrays, and combinatorial testing. *Math Comput Sci* 1–25
11. Kuhn DR, Kacker RN, Lei Y (2013) *Introduction to combinatorial testing*. CRC Press
12. Montgomery DC (2019) *Design and analysis of experiments*. John Wiley & Sons

13. Tzoref-Brill R (2019) Chapter two - advances in combinatorial testing. volume 112 of *Advances in Computers*. Elsevier, pp 79–134
14. Cemal Yilmaz, Sandro Fouché, Cohen Myra B, Adam Porter, Gulsen Demiroz, Ugur Koc (2014) Moving forward with combinatorial interaction testing. *Computer* 47(2):37–45
15. Kacker R, Kuhn DR, Lei Y, Simos D (2021) Measurement of the adequacy of a test suite with respect to the modeled test space. *IEEE Softw* 0–0
16. Rick Kuhn, Kacker Raghu N, Lei Yu, Dimitris Simos (2020) Input space coverage matters. *Computer* 53(1):37–44
17. Wagner M, Kleine K, Simos DE, Kuhn R, Kacker R (2020) CAGEN: a fast combinatorial test generation tool with support for constraints and higher-index arrays. In: 2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), pp 191–200
18. Yu L, Lei Y, Kacker RN, Kuhn DR (2013) ACTS: a combinatorial test generation tool. In: 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation (ICST), pp 370–375
19. Kuhn DR, Okun V (2006) Pseudo-exhaustive testing for software. In: 2006 30th Annual IEEE/ NASA Software Engineering Workshop. pp 153–158
20. Kuhn R, Kacker R, Lei Y, Hunter J (2009) Combinatorial software testing. *Computer* 42(8):94–96
21. Kleine K, Simos DE (2018) An efficient design and implementation of the in-parameter-order algorithm. *Math Comput Sci* 12(1):51–67
22. Bombarda A, Crippa E, Gargantini A (2021) An environment for benchmarking combinatorial test suite generators. In: 2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW). pp 48–56
23. Shaluf I, Fakhru'l-Razi A, Said AM (2003) A review of disaster and crisis. *Disaster Prev Manag* 12:24–32
24. Al-Madhari AF, Keller AZ (1997) Review of disaster definitions. *Prehosp Disaster Med* 12(1):17–21
25. AL-Dahash H, Thayaparan M, Kulatunga U (2016) Understanding the terminologies: disaster, crisis and emergency
26. Murray V, McBean G, Bhatt M, Borsch S, Cheong TS, Erian WF, Llosa S, Nadim F, Nunez M, Oyun R et al (2012) *Case studies*. Cambridge University Press, pp 487–542
27. Lin P, Ralph H, Dharendra S, Trivess M (2014) Planning for sandbagging as a response to flooding: a tool and case study. *Aust J Emerg Manag* 29(4):26–31
28. Walker WE (1995) *The use of scenarios and gaming in crisis management planning and training*, vol 7897. Rand Santa Monica, CA
29. Peterson DM, Perry RW (1999) The impacts of disaster exercises on participants. *Disaster Prevention and Management: An International Journal*
30. Berlin JM, Carlström ED (2014) Collaboration exercises - the lack of collaborative benefits. *Int J Disaster Risk Sci* 5(3):192–205
31. Department of Health and State of New Hampshire Human Services, Health Facilities Administration. Basics of exercise design and administration. <https://www.dhhs.nh.gov/oos/bhfa/documents/beda.pdf>. Accessed 7 Jun 2021
32. Wakelee J, Bradshaw MBAJ, Hites MEPL (2009) Steps in writing an effective master scenario events list. *J Emerg Manag* 7(6)
33. Rankin A, Kovordanyi R, Field JN, Morin M, Jenvald J, Eriksson H (2010) A scenario-based modelling method for simulation systems
34. Limousin P, Tixier J, Bony-Dandrieux A, Chapurlat V, Sauvagnargues S (2016) A new method and tools to scenarios design for crisis management exercises. In: CISAP7: 7th International Conference on Safety and Environment in Process Industry, vol 53. pp 319–324
35. Praiwattana P, El Rhalibi A (2016) Survey: development and analysis of a games-based crisis scenario generation system. In: *International Conference on Technologies for E-Learning and Digital Entertainment*. Springer, pp 85–100
36. Ahmed M, Saleh K, Abobakr A, Nahavandi S (2019) Scenario generation-based training in simulation: pilot study. In: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC). pp 1239–1244
37. Praiwattana P, El Rhalibi A (2016) Scenario generation with cellular automaton for game-based crisis simulation system. In: 2016 9th International Conference on Developments in eSystems Engineering (DeSE). pp 157–162

38. Ferdinandus GR, Peeters M, van den Bosch K, Meyer J-JC (2013) Automated scenario generation - coupling planning techniques with smart objects. In: Proceedings of the 5th International Conference on Computer Supported Education - vol 1: CSEDU. INSTICC, SciTePress, pp 76–81
39. Streicher A, Schönbein R, Pickl SW (2021) A general framework and control theoretic approach for adaptive interactive learning environments. Springer International Publishing, Cham, pp 243–257
40. Martin GA, Hughes CE, Schatz S, Nicholson D (2010) The use of functional L-systems for scenario generation in serious games. In: Proceedings of the 2010 Workshop on Procedural Content Generation in Games, PCGames '10, New York, NY, USA. Association for Computing Machinery
41. Zook A, Lee-Urban S, Riedl MO, Holden HK, Sottolare RA, Brawner KW (2012) Automated scenario generation: toward tailored and optimized military training in virtual environments. In: Proceedings of the International Conference on the Foundations of Digital Games, FDG '12, New York, NY, USA. Association for Computing Machinery, pp 164–171
42. Dunne R, Schatz S, Flore SM, Nicholson D (2011) Optimizing decision preparedness by adapting scenario complexity and automating scenario generation. In: Selected Papers and Presentations Presented at MODSIM World 2010 Conference Expo
43. Dunne R, Schatz S, Fiore SM, Martin G, Nicholson D (2010) Scenario-based training: scenario complexity. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol 54. SAGE Publications Sage CA: Los Angeles, CA, pp 2238–2242
44. Hannay JE, Kikke Y (2019) Structured crisis training with mixed-reality simulations. In: 16th International Conference on Information Systems for Crisis Response and Management
45. van Schijndel DKR, Stolpe A, Hannay JE (2020) Using block-based programming and sunburst branching to plan and generate crisis training simulations. In: Stephanidis C, Antona M (eds) HCI International 2020 - Posters. Springer International Publishing, Cham, pp 463–471
46. Rouwendal DK, Stolpe A, Hannay JE (2021) Toward an AI-based external scenario event controller for crisis response simulations. In: Proceedings of the 18th International Conference on Information Systems for Crisis Response and Management. International Association for Information Systems for Crisis Response and Management
47. Milham LM, Phillips HL, Ross WA, Townsend LN, Riddle DL, Smith KM, Butler PV, Wolf RJ, Irizarry DJ, Hackett MG et al (2017) Squad-level training for tactical combat casualty care: instructional approach and technology assessment. *J Def Model Simul* 14(4):345–360
48. Matsuno Y, Fukunuma F, Tsuruoka S (2021) Development of flood disaster prevention simulation smartphone application using gamification. Springer International Publishing, Cham, pp 147–159
49. Wisner B, Adams J (2003) Environmental health in emergencies and disasters
50. European Centre for Disease Prevention and Control (2014) Handbook on simulation exercises in EU public health settings – how to develop simulation exercises within the framework of public health response to communicable diseases. <https://www.ecdc.europa.eu/en/publications-data/handbook-simulation-exercises-eu-public-health-settings>. Accessed 10 Jun 2021
51. Bouget G, Chapuis J, Vincent J (2009) Conception de scénarios d'attaque de systèmes complexes. In: Workshop 3SGS'09
52. Perry RW (2004) Disaster exercise outcomes for professional emergency personnel and citizen volunteers. *J Conting Crisis Manag* 12(2):64–75
53. Leonard JJ Jr, Roberson MG (1999) Adding realism to tabletop exercises. In: International Oil Spill Conference, vol 1999. American Petroleum Institute, pp 555–560
54. Tonje G, Håvard F (2017) Planning and conducting crisis management exercises for decision-making: the do's and don'ts. *EURO J Decis Process* 5(1):79–95
55. Stern E (2014) Designing crisis management training and exercises for strategic leaders
56. Kuhn DR, Higdon JM, Lawrence JF, Kacker RN, Lei Y (2012) Combinatorial methods for event sequence testing. In: 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation. pp 601–609
57. Duan F, Lei Y, Kacker RN, Kuhn DR (2019) An approach to T-way test sequence generation with constraints. In: 2019 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW). pp 241–250
58. The R Foundation R. <https://www.r-project.org>. Accessed 27 Nov 2022
59. JMP Statistical Discovery LLC. JMP. https://www.jmp.com/en_us/home.html. Accessed 27 Nov 2022
60. IBM. IBM Functional Coverage Unified Solution. https://researcher.watson.ibm.com/researcher/view_group.php?id=1871. Accessed 27 Nov 2022
61. Hexawise. <https://hexawise.com>. Accessed 27 Nov 2022

62. Garn B, Kieseberg K, Celic B, Simos DE (2022) From design of experiments to combinatorics of disasters: a conceptual framework for disaster exercises. In: Simos DE, Rasskazova VA, Archetti F, Kotsireas IS, Pardalos PM (eds) Learning and Intelligent Optimization. Springer International Publishing, Cham, pp 15–26

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.