Multi-Unit PSA Based on Multi-Unit Sequences

Ola Bäckström[a*], Pavel Krcal[b], Futoshi Tanaka[c], Pengbo Wang[a]
[a]RiskSpectrum AB, Stockholm, Sweden
[b]RiskSpectrum AB, Dresden, Germany
[c]Mitsubishi Heavy Industries LTD., Kobe, Japan

Abstract: The multi-unit PSA approach on which this work is based separates modeling of single units and of the scenarios which include several units. The intention is to fully employ and utilize PSA models and analyses of individual units. Multi-unit scenarios are defined separately, by specifying initiating events affecting multiple units, failures of shared equipment, shared parts of safety systems, common cause failures of components across units, and possible human failure events with consequences exceeding a single unit. These events are arranged in a so-called *multi-unit event tree*, where the failure and success of the multi-unit events form the sequences in the event tree. At the end of the sequence, we pass the information from the multi-unit event tree to models for individual units and let them quantify conditional consequence probabilities, e.g., core damage, for each unit separately. Finally, we compose quantification results of the multi-unit sequences and individual units to the overall multi-unit failure frequency (concurrent failure). This paper addresses algorithmic challenges when ensuring scalability of multi-unit PSA calculations with a higher number of units and/or complex shared systems. We also illustrate the concept of multi-unit analysis based on multi-unit sequences by some example calculations.

Keywords: Multi-Unit PRA, Complex Shared Systems, Scalable Calculations.

1. INTRODUCTION

Current Probabilistic Safety Assessment (PSA) models typically cover one unit and can answer questions about its safety or reliability. They encode failure scenarios that lead to a top failure of the unit or to undesired consequences such as an early release of radioactivity. An analysis tool can then estimate frequencies of these failures, their main contributors, possible ranking of components for potential improvements, etc. Often, a plant consists of several units located at the same site, sharing equipment and support systems, influencing each other both during a design basis operation and accident scenarios. The importance of the equipment is mostly understood based on the single unit metric – as the single unit metric is the baseline for decision making. Multi-Unit Probabilistic Safety Assessment (MUPSA) extends the standard unit PSA models to analyze accidents that involve more than one unit. It answers questions about the frequency of two or more units failing at the same time or becoming a threat to life and the environment. It aims at uncovering dependencies between units that contribute most to potential accidents affecting multiple units.

For nuclear power plants, an accident of a single unit leading to a core damage has been typically considered even for sites with multiple units. Firstly, such accidents are already very rare and not acceptable even if they affect only one unit in isolation. After the Fukushima accident, new designs of small modular reactors, considering other types of plants such as spent fuel processing, and other potential applications lead to increased interest in multi-unit analysis also from regulators (including for example the new safety series released by the IAEA [1]) and plant operators. Increasing calculation power makes such analyses principally feasible, but the details of the approach will determine its scalability and practical applicability. There are different possible approaches presented, see for example [1], [2]. One could, in one extreme, build a PSA model covering two or more units and analyze it using conventional methods. But maintaining such models becomes a major obstacle and the scalability of the analysis would restrict the amount of detail and the scope of the analysis.

The metric studied is also a key factor in the selection of method. The two mostly cited metrics regarding CDF are *Concurrent CDF* of multiple units OR *Site CDF*, meaning frequency of one or more unit failing [3].

The multi-unit PSA approach on which this work is based focuses on Concurrent CDF from several units and it separates modeling of single units and of the scenarios which include several units. A similar approach was discussed in the research project funded by NKS [2]. The intention is to fully employ and utilize PSA models and analyses of individual units. Multi-unit scenarios are defined separately, by specifying initiating events affecting multiple units, failures of shared equipment, shared parts of safety systems, common cause failures of components across units, and possible human failure events with consequences exceeding a single unit. These events are arranged in a so-called multi-unit event tree. Sequences in this tree mark which of the events that have effect on multiple units have occurred or not.

For example, a failure of diesel generators might occur at all units simultaneously because of a common cause. In this case, we quantify the multi-unit initiating event and the multi-unit failure events within the multi-unit event tree model. At the end of the sequence, we pass the information from the multi-unit event tree to models for individual units and let them quantify conditional consequence probabilities, e.g., core damage, for each unit separately. If diesel generators fail because of a multi-unit common cause, models for individual units will consider them as unavailable. On the other hand, if there is no multi-unit failure event of diesel generators, then their independent failures (which can still be a CCF event) can still occur within each unit. Finally, we compose quantification results of the multi-unit sequences and individual units to the overall multi-unit failure frequency.

This paper addresses algorithmic challenges when ensuring scalability of multi-unit PSA calculations with a higher number of units and/or complex shared systems. By scalability we mean that real-life systems can be modeled and analyzed by this approach on an acceptable level of abstraction in a reasonable time (adequate to the purpose of the analysis) with standard computational resources. The complexity of the analysis grows exponentially with the number of units and events treated in the multi-unit event tree. Obtaining the exact solution might require prohibitive computational resources. The following features limit the combinatorial explosion: application of cutoff on the multi-unit event tree, selecting representative scenarios to cover all combinations of units, and grouping of multi-unit events to make the multi-unit event tree more compact. We will discuss advantages and limitations of the approach and specifically these efficiency improvements. We also illustrate the concept of multi-unit analysis based on multi-unit sequences by some example calculations. They also allow us to present the algorithmic challenges discussed in this work on concrete use cases. The work presented in this paper is further extensions to work previously presented at PSAM 14 [4] and relates to the paper from PSA 2023 [5] where the approach is used.

2. PRELIMINARIES

In this section, we present several concepts that are necessary for the multi-unit analysis. First, we discuss what we expect from models of individual units, then we introduce multi-unit events and finally, we connect them in multi-unit event trees.

2.1. Unit PSA Models

PSA models for individual units contain event trees and fault trees, with basic and CCF events as leaves. If we select sequences in event trees starting with certain initiators and leading to a certain unwanted damage or consequence that can be reached in plant units, we can build a possibly very large fault tree that captures combinations of events that will cause the consequence or damage to occur. This choice sequences and possibly some configuration conditions define an *analysis case*. In the remainder of the paper, we study without loss of generality a single analysis case. Clearly, the same method can be applied to an arbitrary number of such accident types.

An analysis of a specific analysis case starts with building a possibly very large fault tree. This fault tree can be decomposed to minimal cut sets. Quantification of a list of minimal cut sets is typically very quick, even if we use the quantification algorithm based on Binary Decision Diagrams (BDDs). We assume that we have a sufficiently detailed minimal cut set list for the analysis case of interest for each unit. Moreover, we can re-quantify this minimal cut set list also after modifying probabilities of some basic or CCF events.

2.2. Multi-Unit Events

In principle, PSA models for individual units can differ from each other. There might be events specific to only some units. Other units will not contain these events. Also, naming conventions may be different. The fact that these units are located on the same site creates dependencies between units. These dependencies translate to relations between events from individual units. For events from different units that are related to each other, we can define a new event, a *multi-unit event*, that models the fact that a root cause of these related events might be a single event reaching beyond a single unit. It affects multiple units by a mechanism similar to common cause failures which assigns a part of the probability of individual unit failures to the multi-unit one.

An example of a multi-unit event for two units is a failure in the pumping system where Pump A fails in both units. The corresponding basic events are called U1_PUMP_A_FAIL and U2_PUMP_A_FAIL. We call then multi-unit event MU_PUMP_A_FAIL. We consider different cases.

Assume that the pump is shared by both units. It is physically located outside of these units and pumps water to both. In this case, its failure affects each time both units. We assign it a coefficient 1.0, denoting that 100% of multi-unit failures lead to failures of U1_PUMP_A_FAIL and U2_PUMP_A_FAIL. We can write the multi-unit event as

MU_PUMP_A_FAIL; [U1_PUMP_A_FAIL; U2_PUMP_A_FAIL]; 1.0

In another case, there are two physical pumps in each unit, but in certain situations, both fail because of a common cause such as a maintenance failure or a manufacturing deficiency. This happens only in two percent of cases. Then we assign a coefficient 0.02 to this multi-unit failure. We can write the multi-unit event as

MU_PUMP_A_FAIL; [U1_PUMP_A_FAIL; U2_PUMP_A_FAIL]; 0.02

In the third case, there is again only one pump – and it can only be used for one plant at a time. An example of this type of equipment is a shared mobile diesel generator (compare FLEX equipment). In this case it will be necessary to specify the availability of the object considering the sequence, that is if it is likely that the component is also in use at the other unit. This type of dependency is not discussed further in this paper, even though the approach can include such relation.

2.3. Multi-Unit Event Trees

The multi-unit events can be organized to an event tree, so called multi-unit event tree. Branching points in this event tree model the decision whether failures in individual units occur independently or whether they were caused by a common multi-unit event. Each sequence in this event tree then determines which multi-unit events occurred. The remaining failures happen independently in all units included in the analysis.

Figure 1 shows a multi-unit event tree with an initiating event and three multi-unit events. If the calculation type of the unit PSA models is probability, we can leave the initiating event empty. Otherwise, it maps initiating events of the frequency type relevant to the analysis case under study across all units. After the initiating event, we have a list of multi-unit events that either occur and affect several units or do not occur and then the corresponding failures in individual units are considered independent of each other. Here, we have a loss of offsite power which would typically be a multi-unit event as it can affect all units. Then we consider two pumps that can fail either independently or because of a multi-unit event.

2.4. Basic Method Overview

For a single sequence in the multi-unit event tree, we can quantify the probability of the analyzed consequence for each unit conditional on the information included in this sequence. The conditional probability of all units reaching the consequence is the product of the unit probabilities. Finally, we multiply this probability by the probability or frequency of the sequence in the multi-unit event tree.

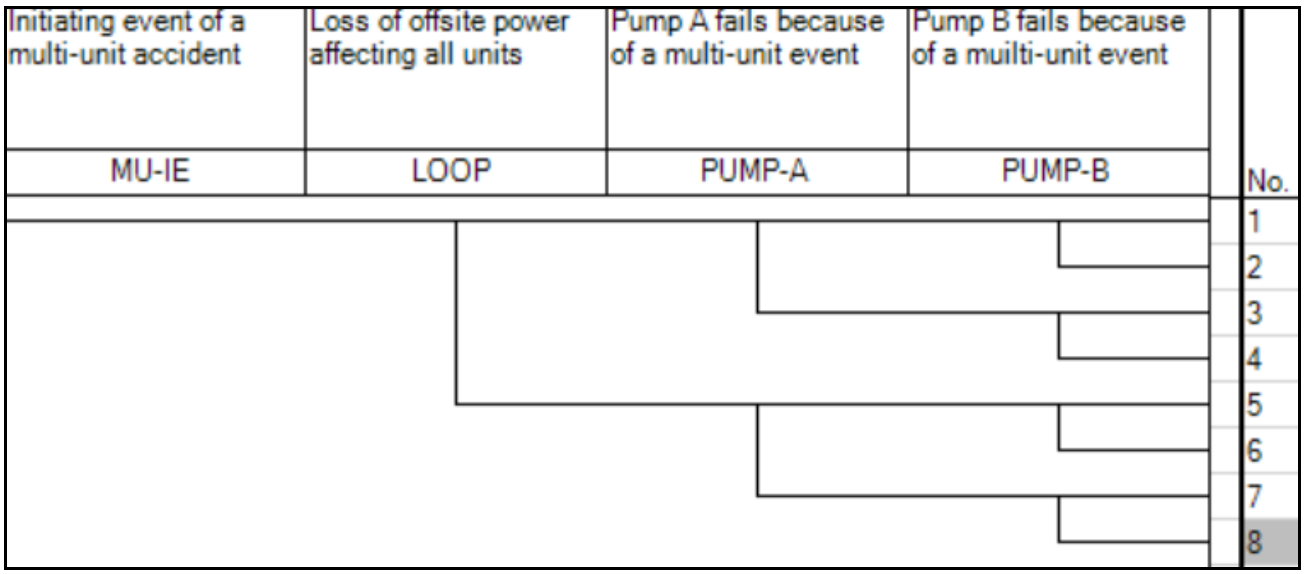

Figure 1. An example of a multi-unit event tree

Let us assume a quantification of multi-unit events that distinguishes between independent failures in individual units and a common failure in all units caused by a multi-unit event. This corresponds to the Beta model for Common Cause Failures. This gives us a single parameter for each multi-unit event, denoted by *x*. Let us further denote by *FailProb* the fraction of the event failure probability assigned to the multi-unit event. This is calculated by

$$FailProb(A) = x \cdot P(A) \quad (1)$$

Where *A* is an event from a multi-unit definition. We assume that all events in a multi-unit definition have the same failure probability. If this is not the case then we take the minimum probability of all events from the multi-unit definition (as if the beta factor should be 1.0, the maximum likelihood the FailProb(A) could have would be the lowest probability of any of the included events).

The conditional probability of individual units can be calculated by updating the probabilities of basic/CCF events affected by the multi-unit events. Let us assume that we want to quantify a unit conditionally on a sequence in the multi-unit event tree. For all multi-unit events along this sequence that have occurred (the sequence passes the failed branch under this multi-unit event), we change the value of the event which belongs to the quantified unit to *1.0*. For all multi-unit events along this sequence that have not occurred (the sequence passes the success branch under this multi-unit event), we change the value of the event which belongs to the quantified unit to *(P(A) - FailProb(A))/(1 - FailProb(A))*. The denominator *(1 - FailProb(A))* avoids optimistic estimate of the multi-unit values. It might in some cases lead to slight over-estimates of the multi-unit top values. This conservatism stays negligible when *x*P(A)* is a small number. For a unit *U*, we denote its conditional failure probability in a sequence *S* by *CFP(S, U)*.

The probability or frequency of a sequence in a multi-unit event tree can be calculated by collecting the probabilities of multi-unit events along this sequence. For a multi-unit event that occurs (and affects all units), we take the probability *FailProb(A)*, where *A* is an event from this multi-unit definition. For a multi-unit event that does not occur and the respective failures in all units are independent, we take the value *1 - FailProb(A)*. For a sequence *S*, we denote the sequence probability by *P(S)*.

Finally, for a sequence *S*, we can calculate failure probabilities of individual units (a single-unit scenario for a unit *U*, denoted *SU(S, U))* and for all units (a multi-unit scenario, denoted *MU(S)*).

$$MU(S) = P(S) \cdot \prod_U CFP(S, U) \quad (2)$$

$$SU(S, U) = P(S) \cdot CFP(S, U) \quad (3)$$

A multi-unit (*MU*) or single-unit (*SU(U)*) failure probability is a sum of probabilities over all sequences.

$$MU = \sum_S MU(S) \quad (4)$$

$$SU(U) = \sum_S SU(S, U) \quad (5)$$

Notice that the sum of all sequences will keep the over-approximation of the multi-unit probability within reasonable bound, because we quantitatively consider both the failure branch and the success branch of each multi-unit event.

Using Figure 1 as an example, there are eight multi-unit event sequences. For each of the sequences a probability/frequency is calculated and then summed up. If we as an example consider sequence S6 (LOOP, -PUMP A, PUMP B), the probability of this sequence in the multi-unit event tree can be directly calculated. The multi-unit probability of LOOP and PUMP B respectively is calculated by multiplying the probability in the referred PSA models with the specified factor for its dependency. PUMP A does in this sequence not suffer from a multi-unit event and is hence calculated as the success. With this information we can calculate the P(S6).

Let us assume that we have two models M1 and M2 where the concurrent failure is of interest. To quantify the MU(S6) we need to quantify each of the referred PSA models M1 and M2 conditional to the sequence. To do this we need to change the probability of LOOP and PUMP B to 1.0 (as they have failed in the multi-unit sequence) and we need to reduce the probability of failure for PUMP A (as part of its failure probability was considered in the multi-unit event tree). When such modifications are performed for models M1 and M2 respectively, the CFP(S6, M1) and CFP(S6, M2) can be calculated. With this, the MU(S6) can be calculated by: P(S6) · CFP(S6, M1) · CFP(S6, M2).

In case that the model contains frequency events as initiators, we need to treat frequency events within the multi-unit event tree. For this, we need to map all frequencies from individual units to multi-unit frequencies. We do not go into technical details of frequency handling here as it does not add any computational complexity to the algorithm. The probabilistic case is sufficient to explain all algorithmic advantages and challenges of this method.

3. ALGORITHMIC ADVANTAGES AND CHALLENGES

The approach presented in the previous chapter has multiple advantages, where the two main advantages are:

- Use of the existing “standard” models
- Use of existing MCS lists

The use of the standard PSA models mean that you can continue to maintain your single unit PSA model and there is no need to create a specific, separate, PSA model to perform the multi-unit calculations. This is time saving when the model is set up – but, more importantly, as you do not have a separate model to maintain, it will reduce the effort to maintain the required PSA models.

As the probability to have a concurrent core damage of two or more units shouldn’t be higher than the risk of a single unit failure – an MCS list which is considered sufficient to represent a single unit failure must be sufficient to represent a multi-unit failure. This is assuming that the concurrent CDF studied is still relevant (and not negligible compared to the single unit CDF). If the concurrent CDF were much lower than the single unit CDF, such that the cutoff for the single unit MCS should be much lower to represent it relevantly, then the concurrent CDF will surely not be of interest (it would be negligible).

The use of existing MCS lists will also provide a sound basis for scalability of the approach. As the complexity of solving a fault tree increases with the number of events and operators, it is clear that a combination of two or more unit PSA models will rapidly increase the complexity of the problem to be solved. Using pre-calculated MCS lists will not only provide fast means for calculation, but relevant cutoff levels also (on a single unit basis) can be used and thereby provide high precision of the results within a reasonable calculation time.

The approach also has some challenges of which can be mentioned,

- Each dependency is specified and defined separately
- Calculation complexity grows with number of dependencies specified
- How to manage success of the multi-unit event in the conditional quantification
- Cutoff impact harder to evaluate

As each dependency is specified explicitly, it presumes that the analyst identifies the dependencies that are worth considering. Simple methods, based on the importance for a single unit, can be used. An example is presented in the NKS report [3], which is based on the Fussel-Vesely importance. To identify the relevant dependencies can be time consuming and is expected to be the most time-consuming task in this approach.

Also, the calculation complexity grows with the number of dependency events identified. It could lead to a combinatorial explosion and is therefore an important aspect for the approach to consider. The complexity grows exponentially with the multi-unit events and the number of units considered. As we calculate with success, all combinations of failure/success (a multi-unit event affecting all units and no multi-unit event with failures in units occurring independently) are considered. This means that we calculate conditional failure probabilities of all units for each such combination. Even if each quantification takes a fraction of a second, the exponential growth of the number of quantifications can result in calculation times of several hours. Each new multi-unit event doubles the number of sequences to calculate and hence the calculation time is also doubled.

The complexity also grows if we want to consider not only failure of all units, but a sub-set of units. This assessment could be relevant to consider because the user may be interested in frequency of failure of two concurrent plants, three concurrent plants etc. Different impact could also be relevant due to CCF aspects for the multi-unit events. If we can assume the beta CCF model for multi-unit events, the complexity does not depend on the number of units. Either the multi-unit event does not occur and the corresponding events in the units occur independently. Or, the multi-unit event occurs and affects all units. One can extend this to other CCF models such as alpha, where one, e.g., for a plant with four units considers also multi-unit events that affect three of them. Even though the events in the plants that fail because of the multi-unit event are completely symmetrical (have the same reliability model and reliability parameters), this can be relevant because the plants themselves do not have to be symmetrical. For some of them, the importance of this event (the risk increase factor) can be greater than for others. Such scenarios might be of special interest to analysts, due to the non-symmetrical behavior.

What is mentioned in the section for Basic method overview above about setting the success to *(P(A) - FailProb(A))/(1 - FailProb(A))* is not directly obvious. The obvious answer would be to set the success probability to *P(A) - FailProb(A).* This will however have the unwanted effect that when the FailProb(A) is reduced for every conditional quantification we will be non-conservative. The term added *(1 - FailProb(A)* is hence to consider for this. This will be exact when the single unit risk is estimated using the multi-unit event tree, but will be slightly conservative (negligible) when combination of several units is considered.

As the calculation approach is based on using MCSs instead of original models, the cutoff effect would be different compared to if we run the analysis from the original master fault trees again – considering the conditional effect. The cutoff of the combined MCS lists will not be easily determined. But in practice, the dominating scenarios and the point estimate of the top event will be very accurate as discussed above.

3.1. Some Solutions to Manage Potentially Time-Consuming Calculations

A potentially significant issue in the approach (described above) is the number of dependencies to consider. There are two main features that are evaluated in the approach:

- Use of cutoff within the multi-unit event tree
- Group dependencies

Their purposes are the same – to reduce or limit the size of the multi-unit event tree, and hence limit the number of conditional quantifications of the unit PSA MCS lists.

The use of cutoff in the multi-unit event tree (*multi-unit sequence cutoff*) does in its simplest form represent that the scenario has a probability (frequency) that is so low, so it is no longer of interest to the quantification. This is a simple, yet very efficient approach expected to have negligible impact on the result. As the setup of the multi-unit event tree will consider all combinations of all multi-unit events – it can be easily understood that a cutoff limit which excludes a lot of combinations (for example a cutoff set an order of magnitude lower than single unit CDF) will yet be very efficient removing a lot of irrelevant multi-unit event tree sequences. Remember, the combinations will still need to be combined with the conditional probability of the single unit MCS lists. The impact of cutoff is further discussed in the section for *Experimental Evaluations*.

To reduce the number of dependencies explicitly specified in the multi-unit event tree, we expect that it could be efficient to represent several dependencies by a gate. It is perhaps a failure of a complete system that has an impact on multiple units. We could break it down into the most important reasons and model them by basic events. These basic events are then included in the definitions of multi-unit events (and then several events could represent the same impact). This could be simplified by just including the system top gates in the multi-unit definitions, and hence reduce the amount of failure causes in the multi-unit event tree. This will however push the complexity to the calculation, as the gate can represent different types of situations (compare OR gate, AND gate etc.). A gate, or a MCS list, cannot be handled as easily as a single event and the calculation has to be modified (all events cannot be set to failed, and what does it mean when a multi-unit event does not occur?).

If the assessment covers more than two units, and the analyst may be interested in failures of two units, three units and four units etc., this may increase the calculation time significantly. This because each sequence from the multi-unit event tree will have to be multiplied with all potential failure combinations. In reality, it is anticipated that the dominating scenario would be a multi-unit event affecting all (affected) units. If there are some combinations of unit failures that are of especial interest, such can be modelled separately. In case there are many units, an approach possible could be to define representative combinations. One possibility is to identify symmetrical units. Another possibility is to identify worst case combinations based on the risk increase factors and focus on these combinations.

4. EXPERIMENTAL EVALUATION

In this section we focus on some experimental evaluations of the approach. Firstly, the multi-unit sequence quantification was applied to a level 2 PSA model to study potential impact when multi-unit dependency was considered. Conditional release probabilities given a site level loss of offsite power (LOOP) event, for multi-unit and single units release sequences were quantified, and the results are shown in Figure 2. In the test case, multi-unit CCFs and multi-unit human error dependencies were considered. An increase in the multi-unit release probability can be seen when multi-unit dependency is considered. The only inputs necessary for the quantification were the MCSs of single units and a list of multi-unit events along with their coefficients described in Section 2.2.

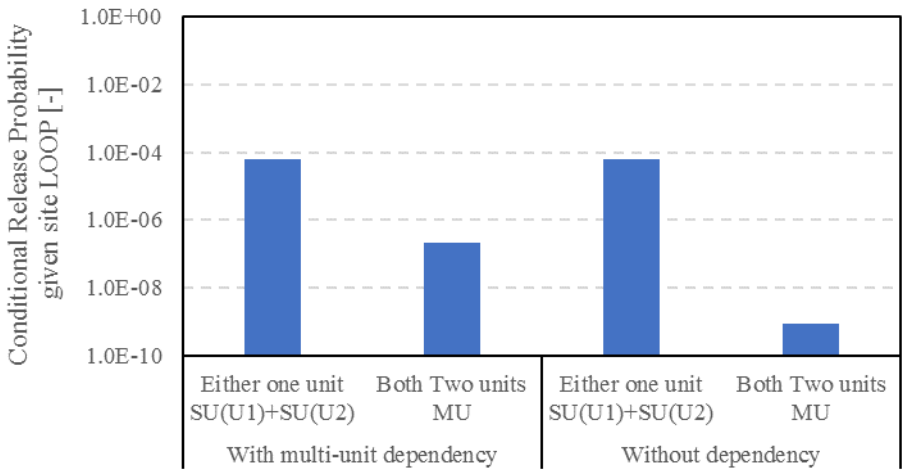

Figure 2. Example of level 2 PSA multi-unit sequence evaluation results

Secondly, an impact assessment of the multi-unit sequence cutoff was studied. The effectiveness of multi-unit sequences cutoff, discussed above as the solution to reduce calculation time, has been tested against a two unit model with 30 multi-unit events. Without cutoff, the number of multi-unit sequences generated by the multi-unit event tree will be $2^{30}(\fallingdotseq 10^9)$ and calculation time will exceed days even if each sequence is calculated within fractions of seconds. With multi-unit sequence cutoff, the multi-unit accident the quantification of trivial sequences that do not affect the overall results are prevented, and the calculation time has been significantly reduced without degrading accuracy. Figure 3 shows the quantification results converging at cutoff values that can be handled within a minute. The analysis was performed using a personal computer equipped with an Intel Core i5-12600 processor and 64 GB random access memory. Achievement of short calculation times implies that we can use a more detailed MCS list as an input, and increase accuracy of the results as if we were using the original model instead of MCSs as inputs.

Application of multi-unit sequence cutoff significantly expands the number of multi-unit events that can be handled. However, if the multi-units share a complex system modeled as a detailed fault tree, the number of multi-unit events could be hundreds. Or in cases where the probabilities of multi-unit events are high, multi-unit sequence cutoff will not effectively reduce the sequence to be quantified, and calculation time could still be an issue. To use the approach of gates instead of basic events is one of the solutions to such cases.

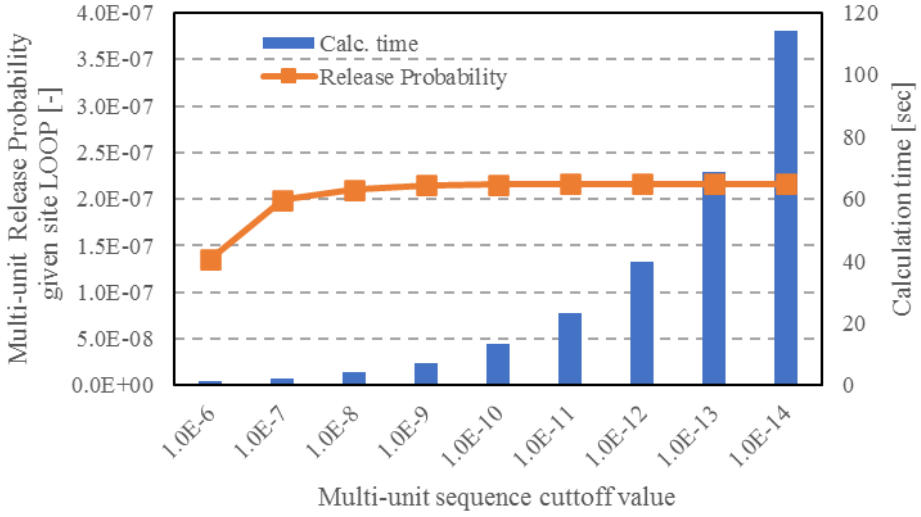

Figure 3. Effect of mutli-unit cutoff fucntion when applied to a model with 30 multi-unit events

5. CONCLUSION

We have presented a multi-unit approach that separates modeling of single units and of the scenarios which include several units. The approach has several advantages where the two main advantages are the use of the existing single unit models and scalability of the approach.

Even though the approach seems straightforward and has properties that make the approach scalable – when put into practical tests of real models there are challenges. For example, how to treat success of multi-unit events when the events are considered in the conditional quantification AND how many dependencies can be represented without a combinatorial explosion? This paper discusses the main challenges faced and comes with practical suggestions on how to manage such challenges. Cutoff in the multi-unit event tree is an efficient approach to ensure scalability. The positive impact of cutoff on calculation efficiency for multi-unit analyses has been confirmed by evaluation on a real-life case study. There are further extensions for continued work indicated, where the use of gates or MCS lists is considered a main future improvement.

References

[1] International Atomic Energy Agency. Multi Unit Probabilistic Safey Assessment. IAEA, Safety Report Series No.110, 2023.
[2] Holmberg J et al. Site risk analysis for nuclear installations, NKS, NKS-419, February 2019
[3] Modarres M, A Review of Multi-Unit Nuclear Power Plant Probabilistic Risk Assessment Research, 26th International Conference on Nuclear Engineering (ICONE26), July 22-26, 2018
[4] Bäckström O et al., SITRON – Site risk assessment approach developed for Nordic countries, PSAM 14, September 16-21 2018
[5] Tanaka F, Yamamoto Y and Shioya R. Treatment of Adjacent Unit Release Effects in Multi-Unit PRA, PSA 2023 proceedings, pages 826-835, July 15-20 2023