OPTIMIZATION OF DYNAMIC RISK ASSESSMENT IN NUCLEAR POWER PLANTS USING OPTIMIZED AND ACCELERATED SIMULATION INTERMEDIATE STATE STORAGE

Kyeongseo Kim, Junyong Bae, Wooseok Jo, Seung Jun Lee*

Ulsan National Institute of Science and Technology, 50 UNIST-gil, Ulju-gun, Ulsan, 44919, Republic of Korea

smart2439@unist.ac.kr; junyong8090@unist.ac.kr; cws5528@unist.ac.kr; sjlee420@unist.ac.kr

ABSTRACT

In Nuclear Power Plant, Risk Assessment is a process of evaluating safety by analyzing the possibility and impact of accidents in reactors and related systems. Through this, accidents are prevented radiation exposure and environmental damage are minimized to ensure the safe operation of nuclear power plants. Existing static risk assessment has limitations in that there is a lack of time association between components and systems and that there are limited accidents that can be analyzed because the analysis is performed under fixed conditions. Accordingly, a dynamic risk assessment through simulation of the thermal hydraulic system code emerged. However, there is a limitation of long simulation time, and to solve this problem, the study proposes Optimized and Accelerated Simulation Intermediate State Storage (OASIS). OASIS stores branch points according to changes in the operating status of NPP components, systems, and human operations, and if there is any overlap with the scenarios previously covered in the simulation of the scenario later, it is brought in for streamlined simulation. The problem of insufficient storage capacity is addressed by the branch point weighting method, which is expressed as the product of the time taken from each branch point to the storage point and the number of subsequent branches. The case study addresses the dynamic scenario of Loss of Offsite Power by optimizing scenario sampling using Deep Neural Network-based Search Algorithm of Informative Limit Surfaces and States (Deep-SAILS). To address storage limits, the number of branch point save files was restricted to 300. Under this configuration, the integrated use of Deep-SAILS and OASIS achieved a 96.30% reduction in simulation time, whereas the application of OASIS alone resulted in a 48.31% reduction. Additionally, the branch point weighting method manages insufficient storage space well.

Keywords: Dynamic PSA, Streamlined Simulation, Optimization, Weight, Database Management

I. INTRODUTION

The systematic evaluation of nuclear power plant risks is vital for ensuring public safety. To this end, risk assessment has been established as a rigorous, structured process for identifying hazards and implementing mitigation strategies [1]. Nuclear plant operation depends on the flawless interaction of multiple sophisticated systems; the Fukushima accident, however, revealed how a cascade of system failures can trigger complex nuclear accidents releasing radioactivity and inflicting long-lasting environmental degradation, area contamination, and ecological damage [2].

Consequently, early identification and quantification of potential plant risks are imperative. Risk assessment addresses this need by exhaustively modeling accident sequences and devising measures to enhance safety. In contrast to conventional static methods, dynamic risk assessment tracks the time-dependent interaction of system states, yielding more accurate and realistic risk profiles [3]. By integrating live operational data, reflecting continual system changes, dynamic risk assessment affords a deeper, more dependable understanding of safety margins. Yet, executing the vastly increased number of thermal-hydraulic (TH) simulations required by dynamic approaches demands far greater computational power than static analyses [4–6].

Achieving an optimal balance among computational load, simulation fidelity, and turnaround time remains a central challenge. To mitigate this, prior work has employed adaptive sampling, specifically the Deep Learning–based Searching Algorithm for Informative Limit Surfaces, States, and Scenarios (Deep-SAILS) [7]. Deep-SAILS efficiently distinguishes between safe and failed trajectories while limiting simulations to roughly 10% of all possible scenarios. However, it does not shorten the runtime of each individual TH simulation.

To fill this gap, we introduce the Optimized and Accelerated Simulation using Intermediate State Storage (OASIS) methodology. OASIS archives the temporal states of NPP components from completed runs and detects time-overlaps with new scenarios. By restarting each simulation at the relevant stored state rather than from time zero, OASIS drastically cuts per-scenario simulation time.

The study validates OASIS through a Loss of offsite power (LOOP) case study, comparing simulation duration and storage demands with and without OASIS integration. The findings confirm that OASIS offers a practical, high impact means of accelerating large scale dynamic risk assessments without compromising accuracy.

II. BACKGROUND

The accurate execution of the TH system codes is fundamental to analyzing accident scenarios in nuclear power plants [4]. In probabilistic risk assessment, these scenarios emerge from intricate sequences of both automated protections and operator interventions, necessitating a vast number of high-fidelity TH simulations. Consequently, carrying out millions of such simulations within a reasonable timetable is infeasible given computational limits of current study [5]. Unlike static risk assessment, which can rely on relatively simple Event Tree/Fault Tree logic, dynamic risk assessment must explicitly capture the time-dependent interactions among all system components. This temporal coupling greatly increases scenario complexity and expands the simulation domain [6]. Moreover, the sheer volume of time-resolved TH runs drives up both processing requirements and individual run durations compared to static methods.

To alleviate this burden, the Deep-SAILS adaptive sampling framework was introduced. Deep-SAILS is specifically tailored for NPP risk analysis, seeking to efficiently regions the Limit Surface (LS), the surface between success and failure region, using a Deep Neural Network metamodel augmented with Monte Carlo Dropout [7]. Rather than exhaustively simulating every possible scenario, Deep-SAILS iteratively targets samples near the evolving LS, focusing computational burden where it matters most. This targeted approach dramatically reduces the total number of scenarios, TH simulation executions required.

Figure 1 shows Deep-SAILS LS for a Loss of Coolant Accident (LOCA), constructed by discretizing critical parameters (e.g., injection flow rate and coolant injection delay). Each node represents a candidate scenario, and failure is defined by Peak Cladding Temperature (PCT) exceeding 1478 K. By concentrating on grid points adjacent to this boundary, Deep-SAILS delivers a high-resolution risk portrait with only a fraction of the total scenarios.

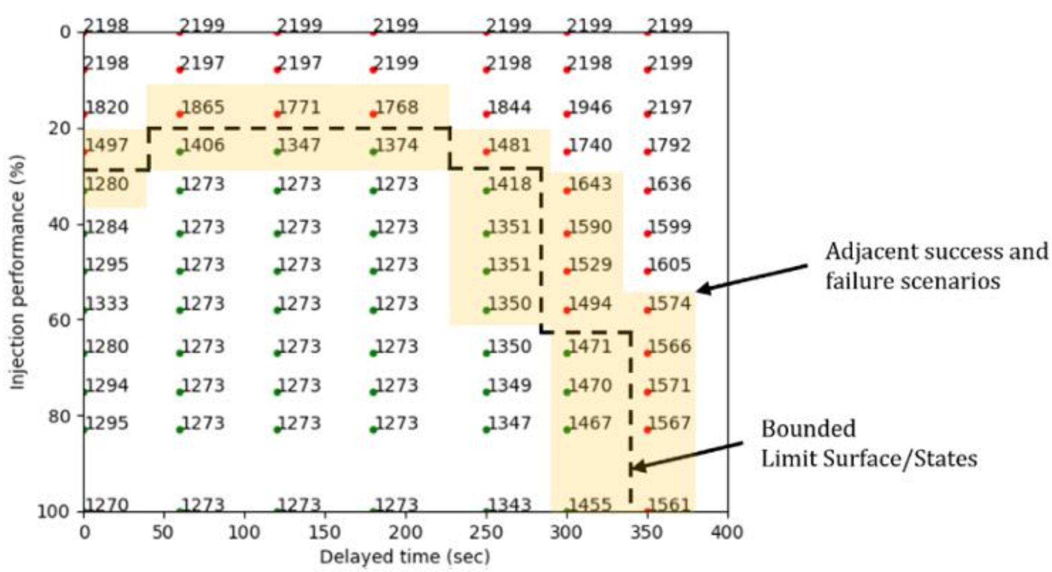

FIGURE 1. Scenario consequence of limit surface of Deep-SAILS of Two parameters (Injection performace, Delayed time of the cooling water) [6]

Delineating the LS is computationally intensive, owing to the nonlinear coupling among numerous NPP system parameters. To mitigate this, Iterative refinement strategy was adopted that incrementally hones the LS estimate while curtailing computational burden. In each cycle, scenarios predicted to lie near the provisional LS are preferentially sampled and subjected to high-fidelity simulation. These newly obtained results then augment the training set of the surrogate model (e.g., a deep neural network), progressively enhancing its boundary approximation. By concentrating on the most informative cases, this targeted sampling framework eliminates superfluous runs and markedly reduces total computational burden without sacrificing assessment accuracy.

Nonetheless, as the scale of risk assessment grows, even such adaptive samplings confront limits in handling millions of candidate scenarios. To summarize this, the current study introduces OASIS.

As the scope of probabilistic risk assessments expands to encompass millions of candidate scenarios, even Deep-SAILS, adaptive sampling reaches its practical limits. To address this, the current study proposes OASIS, which identified points of control action, operator action and change of component state, from previously executed runs to detect and skip overlapping parts, thereby eliminating redundant computations and improving efficiency.

III. OPTIMIZED AND ACCELERATED SIMULATION USING INTERMEDIATE STATE STORAGE

The TH system code employed in this study is a specialized computational framework developed to capture the coupled thermal hydraulic phenomena occurring during accident transients in NPPs. Widely adopted for both routine safety evaluations and severe accident analyses, TH system codes facilitate risk assessment of NPPs responses across a broad spectrum of initiating events, thereby supporting advanced risk assessment methodologies and informing risk-informed decision-making in plant operation.

A major feature of TH system codes is ‘Restart’ function, which allows simulation states to be archived at selected time points and later reloaded to resume the transient from that point. By obviating the need to recompute the entire transient from initial conditions for each scenario, this feature shortens times.

In the proposed OASIS framework, the simulation “Restart” function is leveraged to streamline scenario execution. Defining the instant at which a control action occurs as the branch point, then identify overlapping parts between previously simulated scenarios and the current scenario. By bypassing these overlapping parts and starting simulation execution from the branch point, OASIS significantly reduces the computational burden while preserving the fidelity of the dynamic risk assessment.

Through systematic optimization of computational cost utilization, the algorithm eliminates unnecessary processing and enhances the efficiency of the simulation. The schematic of this methodology is illustrated in Figure 2.

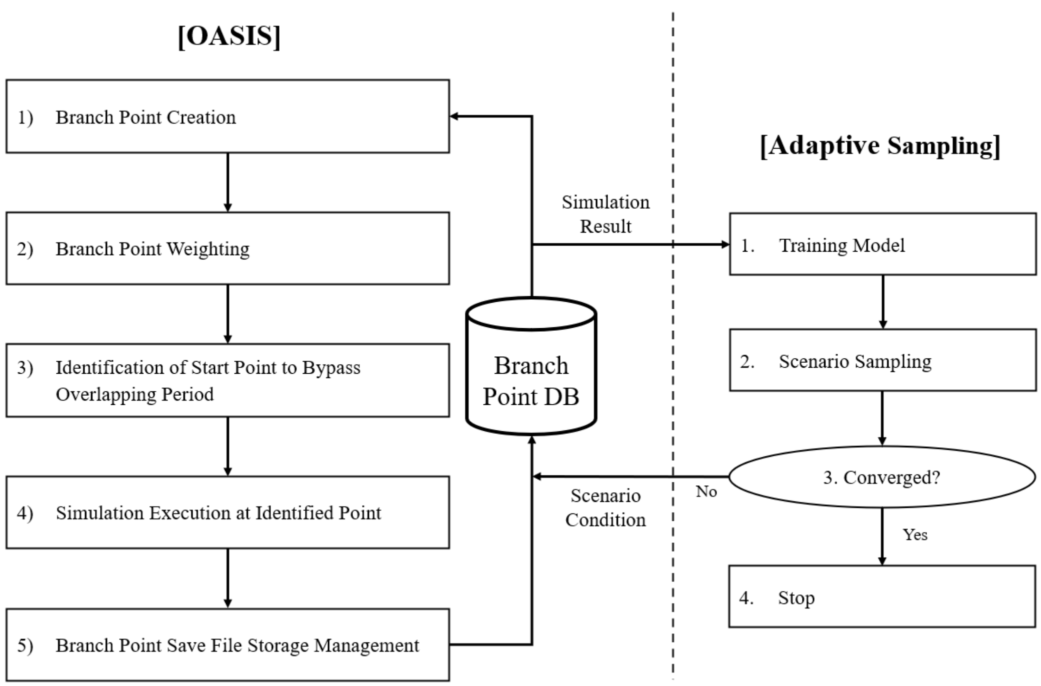

FIGURE 2. Framework of Optimizied and Accelerated Simulation using Intermidiate Storage

The proposed framework is organized into a systematic five-stage sequence. During the simulation execution stage, a cyclic workflow is employed: as each scenario advances, newly identified branch points are recorded. Once the simulation is completed, these results are reloaded, allowing the procedure to loop back to the branchpoint creation stage. Upon reverting to the branch point creation stage, the conditions and results of the current iteration's scenarios are conveyed to the adaptive sampling module, which then prescribes the initial conditions for the scenarios to be executed in the subsequent iteration.

III.A. BRANCH POINT CREATION

In this stage, branch points within each scenario are identified. A branch point is defined as the instant at which a control action occurs, resulting in a change in either human operation or a component state.

Figure 3 illustrates the locations at which branch points are identified within a simulated scenario. Each branch point corresponds to the moment of events, control actions, takes place, and in this example three distinct branch points are identified within the scenario.

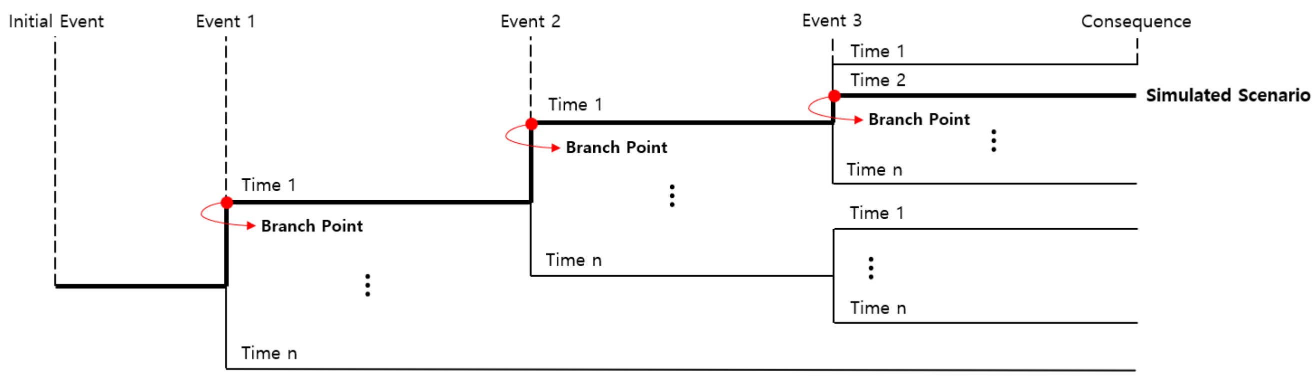

FIGURE 3. Example of Branch Point Identification

III.B. BRANCH POINT WEIGHTING

In this stage, each branch point identified during the generation stage is assigned a weight according to Eq. (1), where T denotes the simulation time elapsed at the branch point and B represents the count of subsequent branches emanating from that point. Specifically, the branch point weight is determined by multiplying T by B, thereby providing a metric for the potential reduction in computational time afforded by bypassing overlapping part of the simulations.

$$Scenario\ Weight = T\ \times B \tag{1}$$

When determining the count of subsequent branches, any consecutively occurring CD scenarios need not each undergo full simulation. Instead, these CD scenarios should be treated as a single merged CD branch to prevent inflation of the branch count. By merging them into one CD branch and excluding the redundant CD scenarios from simulation, the scenario-weight metric is not unduly skewed toward overly high values, and their relative importance is appropriately moderated. Figure 4 illustrates an example of this CD scenario merging.

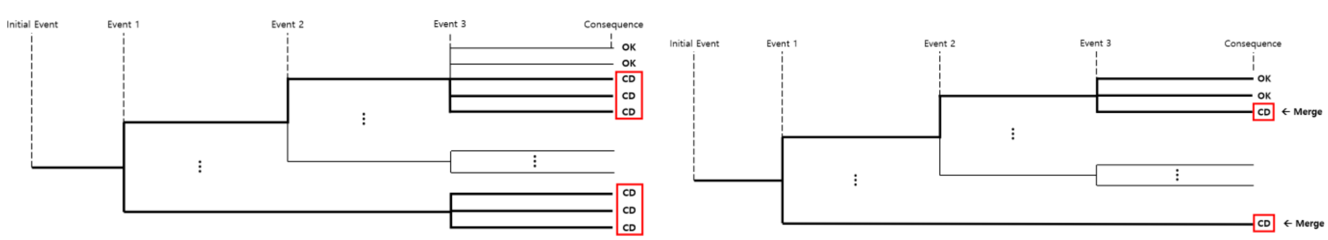

FIGURE 4. Example of CD scenario merging (Left: Event tree before merging; Right: Event tree after merging)

After merging consecutive CD scenarios into unified branches, the branch-point weighting metric is applied to compute each branch point's weight. Figure 5 provides an illustrative example of this calculation. In the example, Event 3, corresponding to the control action occurred in the event tree of Figure 4, is selected as the branch point. The simulation time up to this branch point is 12 hours, and the number of subsequent branches originating from this point is three. Substituting these values into the weighting metric (12 × 3) yields a branch-point weight of 36, which is then assigned to that branch point.

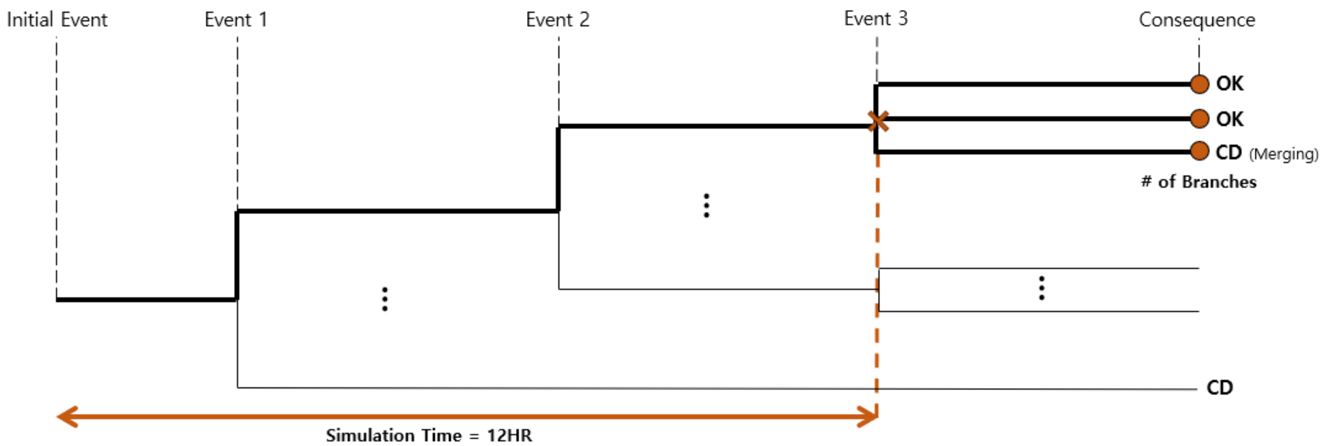

FIGURE 5. Example of branch point weight calculation

III.C. IDENTIFICATION OF START POINT TO BYPASS OVERLAPPING PERIOD

In this stage, archived branch point save file from prior simulations are leveraged to determine the optimal simulation execution point for the currently selected scenario. The branch point save file that promises maximal computational savings is chosen and designated as the simulation execution point.

Figure 6 illustrates the overlapping parts between the current scenario and two previous scenarios, as well as their respective outermost overlapping points. By comparing the outermost time stored at each prior branch point, the overlapping parts between previous scenario 1 and the current scenario is identified as point (1), and that between previous scenario 2 and the current scenario as point (2). Given that point (2) affords greater reduction in simulation time, it is selected as the simulation execution start point.

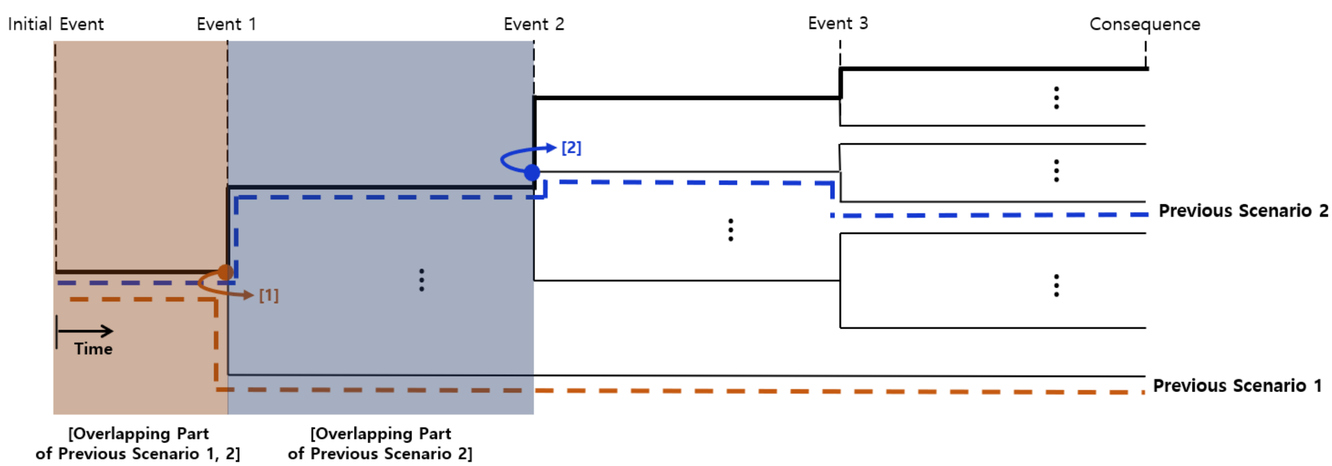

FIGURE 6. Example of identifying the outermost overlapping point (1: Outermost overlapping point of previous scenario 1; 2: Outermost overlapping point of previous scenario 2)

III.D. SIMULATION EXECUTION AT IDENTIFIED POINT

In this stage, the simulation is executed from the simulation execution point identified during the 'Identification of start point to bypass overlapping period' stage. Figure 7 illustrates an example of this simulation execution. In this case, the simulation execution point was designated as point (2) during the 'Identification of start point to bypass overlapping period' stage, and thus, the simulation is initiated from point (2).

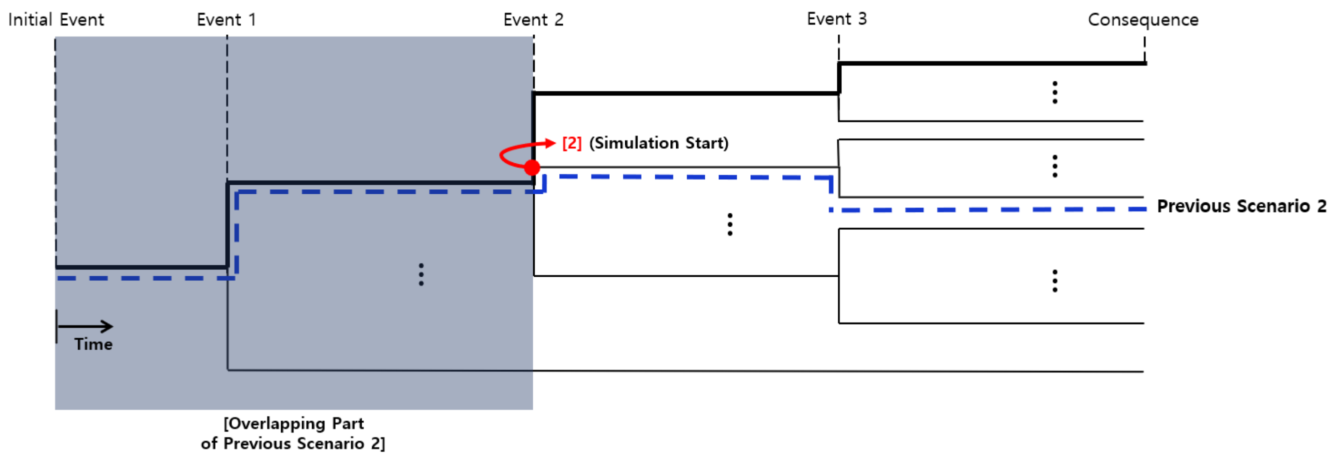

FIGURE 7. Example of simulation execution at the outermost overlapping point

III.E. BRANCH POINT SAVE FILE STORAGE MANAGEMENT

This stage aims to address the storage limitations for saving branch point information data. The branch point information data are sorted in ascending order based on their weights. Then, the weight of the candidate branch point data is compared to those of the existing stored branch point data. If there is any stored branch point data with a weight lower than that of the candidate data, the stored branch point data with the lowest weight is removed and replaced with the candidate data. Subsequently, as the newly saved branch point data may enable merging of CD scenarios, the entire set of stored branch point data is updated by reflecting the possibility of merging, using the weight and the results of the stored branch points.

IV. CASE STUDY

To evaluate the effectiveness of OASIS in reducing computation time, a simplified LOOP (Loss of Offsite Power) scenario was used. This case study employed the OPR 1000 plant type, which is a Korean-designed pressurized water reactor with an approximate power output of 1,000 MWe. Figure 8 presents a simplified event tree of the conventional complex LOOP scenario, incorporating the newly introduced Multi-barrier Accident Coping Strategy (MACST). In this Event Tree, the mitigation components considered were the Alternative AC-Diesel Generator (AAC-DG), the Atmosphere Dump Valve (ADV), and the Portable Low Pressure Pump (PLPP). While the Event Tree was designed for sequential operation of AAC-DG, ADV, and PLPP, to better reflect the dynamic nature of the scenario, PLPP was modeled as an independent variable capable of operating at any time.

In this case study, the simulation time was set to a maximum of 72 hours to account for the extended mission time associated with the use of the PLPP, one of the MACST facilities. The scenario assumption for this case study defined that core damage is considered to have occurred if the Peak Cladding Temperature exceeds 1478 K, at which point the simulation is terminated. Additionally, the reactor cooling rate was assumed to be maintained at or below 56°C/hr when using the ADV. Furthermore, the operating pressure of the PLPP was assumed to be no higher than 25 kg/cm², and the number of branch point save files was limited to 300.

Table 1 presents the discretized operating conditions of the components in the LOOP scenario used in the case study. The AAC-DG fail time and the PLPP start time are each discretized into 24 one-hour intervals spanning from 1 to 24 hours, if these events occur after the initial LOOP event. The ADV opening time is discretized into 25 one-hour intervals ranging from 0 to 24 hours, under the assumption that these events occur after the AAC-DG fail time.

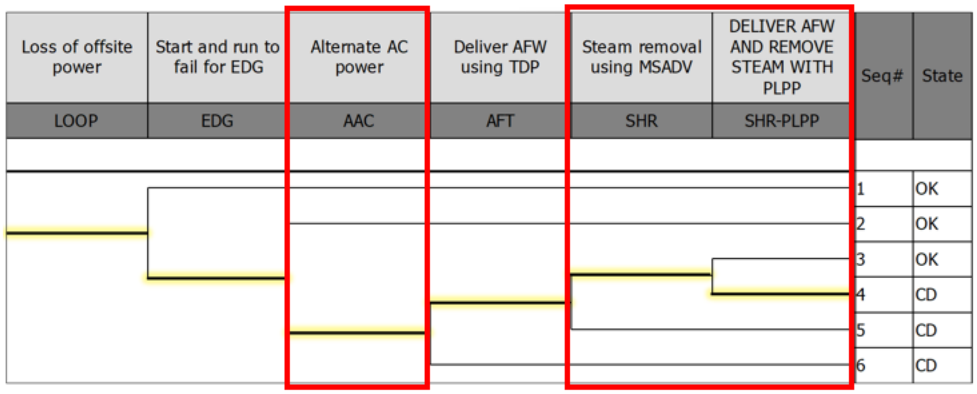

FIGURE 8. Event Tree for LOOP (with MLP Facility)

TABLE 1. Parameters of LOOP Scenarios and Discretized Operating Conditions

Parameter (Axis)		**Operating Condition**	**Resolution (HR)**
1	**AAC Fail Time**	**1~24HR** after Initial Event LOOP	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 **(24)**
2	**ADVs Opening Time**	**0~24HR** after AAC Fail Time	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 **(25)**
3	**PLPP (Portable Low-Pressure Pump) start time for deployment and connection**	**1~24HR** after Initial Event LOOP	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 **(24)**

In the case study, a total of 14,400 scenarios were considered. Based on the Deep-SAILS calculations, 1,033 scenarios equivalent to 7.17% were identified as critical scenarios located near the LS, and these were simulated. Using Deep-SAILS alone, it took 75 hours and 25 minutes to evaluate the results for all 14,400 scenarios. When Deep-SAILS was coupled with OASIS, the total simulation time was reduced to 38 hours and 59 minutes, representing a 48.31% reduction compared to using Deep-SAILS alone. Extrapolating from the Deep-SAILS efficiency, it was estimated that simulating all scenarios without applying any method would have taken 1,051 hours and 50 minutes. Therefore, the combination of Deep-SAILS and OASIS enabled the evaluation of all scenario results using only 3.3% of the originally required simulation time.

V. CONCLUSION AND DISCUSSION

To address the challenge of large computational costs in dynamic PSA, which arise from the numerous scenarios and extensive thermal-hydraulic (TH) simulation times, this study proposes an algorithm that leverages streamlined simulation and intermediate state storage management. By combining adaptive sampling with the TH system code's 'Restart' function, the algorithm identifies and skips the overlapping portions between previously simulated scenarios and the current scenario, thereby reducing simulation time.

In the case study, the initial event was set as LOOP, and the TH system code was coupled with Deep-SAILS, an adaptive sampling technique. A total of 14,400 scenarios were considered. When Deep-SAILS and OASIS were coupled, simulation time was reduced by 96.30% compared to the baseline with no method applied.

Future research is expected to focus on the validation of the algorithm's effectiveness in capturing the dynamic behavior of scenario progression under more realistic conditions, the validation of the proposed methodology through comparison with alternative approaches, and the optimization of time efficiency while considering the limitation on the number of branch point save files.

ACKNOWLEDGMENTS

This work was supported by Korea Institute of Energy Technology Evaluation and Planning(KETEP) grant funded by the Korea government(MOTIE)(RS-2024-00403194, Next-Generation Nuclear Technology Creation IP-R&D Talent (Human Resources). Development Project) This research was supported by the National Research Council of Science & Technology(NST) grant by the Korea government (MSIT) (No. GTL24031-400).

REFERENCES

[1] T. Bedford and R. Cooke, Probabilistic Risk Analysis: Foundations and Methods. Cambridge, U.K.: Cambridge Univ. Press, 2001.

[2] C. Seong, G. Heo, S. Baek, J. W. Yoon, and M. C. Kim, "Analysis of the technical status of multiunit risk assessment in nuclear power plants," Nuclear Engineering and Technology, vol. 50, pp. 1035–1043, 2018.

[3] D. Mandelli, C. Wang, C. Parisi, D. Maljovec, A. Alfonsi, Z. Ma, and C. Smith, "Linking classical PRA models to a dynamic PRA," Annals of Nuclear Energy, vol. 149, p. 107746, 2020.

[4] S. Ryu, H. Kim, S. G. Kim, K. Jin, J. Cho, and J. Park, "Probabilistic deep learning model as a tool for supporting the fast simulation of a thermal–hydraulic code," Expert Systems with Applications, vol. 200, p. 116966, 2022.

[5] J. Park and H. Kim, "A case study to address the limitation of accident scenario identifications with respect to diverse manual responses," Reliability Engineering & System Safety, vol. 251, no. 110406, 2024.

[6] E. Zio, "Integrated deterministic and probabilistic safety assessment: concepts, challenges, research directions," Nuclear Engineering and Design, vol. 280, pp. 413–419, 2014.

[7] J. Bae, J. W. Park, and S. J. Lee, "Limit surface/states searching algorithm with a deep neural network and Monte Carlo dropout for nuclear power plant safety assessment," *Applied Soft Computing*, vol. 124, p. 109007, 2022.