INL/RPT-24-80380

Light Water Reactor Sustainability Program

Tools And Methods to Analyze Plant Outage Schedules and Assist Schedulers in Improving Outage Resilience

September 2024

U.S. Department of Energy

Office of Nuclear Energy

DISCLAIMER

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

INL/RPT-24-80380

Tools And Methods to Analyze Plant Outage Schedules and Assist Schedulers in Improving Outage Resilience

**D. Mandelli, S. St. Germain, E. Chen, C. Otani, J. Nielson,
J. Boyce, J. Mapes, A. Al Rashdan, C. Wang, and S. Lawrence**

September 2024

**Idaho National Laboratory
Light Water Reactor Sustainability
Idaho Falls, Idaho 83415**

http://lwrs.inl.gov

**Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517**

Page intentionally left blank

ABSTRACT

Nuclear power plant (NPP) refueling outages are considered one of the most critical phases in a plant's lifetime. In such instances, tens of thousands of activities (e.g., maintenance and surveillance) are performed in a short amount of time (typically 2–3 weeks unless major backfitting or modernization projects are carried out) by a large number of crews (e.g., electricians and mechanics) hired as contractors. As a result, plant outages can be expensive not only in terms of costs (e.g., contractor labor and material), but also in terms of loss generation since the plant is taken off the grid during the full outage duration (a representative metric is about 1.2M$/day of lost revenue). Thus, there is a continuous need to decrease outages' economic impact on plant finances by decreasing the frequency of plant outages (e.g., from 18 to 24 months), reducing the time to complete the outage, and reducing the risk of outage delays. The Optimization of Outage Activities project under the Risk Informed Systems Analysis Pathway (RISA) sponsored by the U.S. Department of Energy's Light Water Reactor Sustainability (LWRS) Program focuses on developing tools and methods to support NPPs with outage schedule optimization. The developed tools and methods are designed to analyze plant outage schedules so as to identify critical elements in the schedule that may pose a high risk of delay. These methods and tools can be considered resource-centric in the sense that they address outage challenges as a resource optimization problem. In this context, the resources are *time* and *crews*; outage delays occurs when either (or both) resource types are insufficient to complete the set of tasks assigned at a specific time instant of the outage. This report provides details on how plant resources (time and crews) can be allocated such that delays are minimized. In this respect, two classes of methods have been developed: one that focuses on the time resource and how variability in the time to complete outage tasks may impact outage delays, and one that minimizes the risk of outage delays by integrating available resources to assess when daily activities should be performed.

Page intentionally left blank

CONTENTS

FIGURES

TABLES

Page intentionally left blank

ACRONYMS

CP	critical path
IR	incident report
LWRS	Light Water Reactor Sustainability
MBSE	model-based system engineering
ML	machine learning
NLP	natural language processing
NPP	nuclear power plant
POS	part of speech
RAVEN	Risk Analysis Virtual Environment
RISA	Risk Informed System Analysis
TF	total float
UI	user interface
WCAT	work category
WO	work order

Page intentionally left blank

Tools And Methods to Analyze Plant Outage Schedules and Assist Schedulers in Improving Outage Resilience

1. INTRODUCTION

Nuclear power plant (NPP) refueling outages can be considered one of the most critical and labor-intensive phases in a plant's lifecycle (IAEA, 2022). During an NPP outage, 10,000–18,000 activities of various natures (e.g., maintenance, testing, and surveillance) are performed over the course of just a few weeks (typically 2–3 in the United States), unless major backfitting (e.g., steam generator replacement) or modernization projects (to extend plant life) must be carried out. This enormous array of activities is performed by a large number of crews (e.g., electricians and mechanics) hired as contractors during the outage planning phase. Consequently, plant outages can be costly (due to planning, contractor labor, material, and asset procurement), even without taking into account the actual loss of power generation due to the plant being taken off the grid for the full outage duration (Jeong, 2022); indicatively, the revenue lost when a plant is offline totals about 1.2M$/day.

Given these issues, NPP owners continuously require that the economic impact of outages on plant finances be decreased. This can be accomplished by:

1. *Decreasing the frequency of plant outages* (e.g., from 18 to 24 months): Note here that plant outage frequency is primarily dictated by fuel (mostly burnup considerations) and regulatory constraints (e.g., surveillance and testing of safety systems).
2. *Decreasing the risk of unplanned outages*: Unplanned outages are typically caused by either internal events (e.g., failure of safety-related equipment that might pose high risk of core damage) or external events (e.g., seismic activity, flooding, or hurricanes).
3. *Reducing the time to complete the outage*: This can be done by reducing the scope of the outages (i.e., the number of activities to be completed) or by maximizing resource usage in the outage planning phase.
4. *Reducing the risk of outage delays*: This is accomplished by identifying elements in the outage that might directly cause delays, and by proactively managing available resources so as to reduce such risk.

The Optimization of Outage Activities project under the Risk Informed Systems Analysis (RISA) pathway, which is sponsored by the Department of Energy's Light Water Reactor Sustainability (LWRS) Program[1] (Mandelli, 2023), addresses the fourth of the above-listed items by focusing on the development of computational tools and methods to support light-water reactor types such as pressurized- or boiling-water reactors, as well as on outage planners to identify outage schedule criticalities and optimize available resources (i.e., time and crews).

[1] LWRS official website: https://lwrs.inl.gov/SitePages/Home.aspx.

These methods and tools are designed to be resource-centric in the sense that they address plant outage challenges as a resource optimization and risk analysis problem. In the present work, we directly consider two types of resources: time and available labor. On this basis, it is natural to assert that when either (or both) of these resources is insufficient to complete the given set of tasks assigned on a specific day of the outage, a delay occurs. In this respect, the developed methods are designed to:

1. Analyze planned outage schedules to assess when these two kinds of resources may be insufficient to complete the set of activities assigned on a given day.
2. Provide insights into how available plant resources (time and crews) can be allocated such that the risk of delay is minimized.

This report details the developed methods, their use, and their potential impacts when employed to assist outage planners in finalizing outage schedules. In particular, two classes of methods have been developed:

- One that focuses on the time resource and, in particular, utilizes natural language processing (NLP) methods to evaluate the variability in the time to complete an outage task. It then evaluates the impact of activity duration variance on outage delays.
- One that integrates available resources in order to assess how daily activities should be performed such that the risk of outage delays is minimized.

Given the proprietary nature of the data used in this project, this report does not provide details on the considered plant or the specific outage instances. Similarly, the outcome of each analysis step reported herein has been digitally edited, obscured, or hidden to secure the provided proprietary information.

2. IMPACT OF PLANT OUTAGE ON PLANT OPERATION

Not only a plant outage is a critical element of a plant's lifecycle, they also directly impact the amount of power generation that is lost. As an example, the graph in Figure 1 shows the electric power (green line) that was lost over the 2007–2011 time frame, due to plant outages in the United States. NPPs typically plan for outages either in spring or fall so as to guarantee that ample electric power is available for meeting peak demand during summer (typically the result of AC cooling) and winter (typically the result of heating). Due to operational and regulatory constraints, NPP outages occur approximately every 18 (for pressurized-water reactors) or 24 months (for boiling-water reactors) of operation. The plot in Figure 1 partitions the power lost due to outages for systems that refuel every 18 or 24 months (the blue and the orange lines, respectively). Note that the provided data do not differentiate between planned vs. forced (i.e., unplanned) outages. A plant is considered to be undergoing an outage if its power is recorded to be 0%.

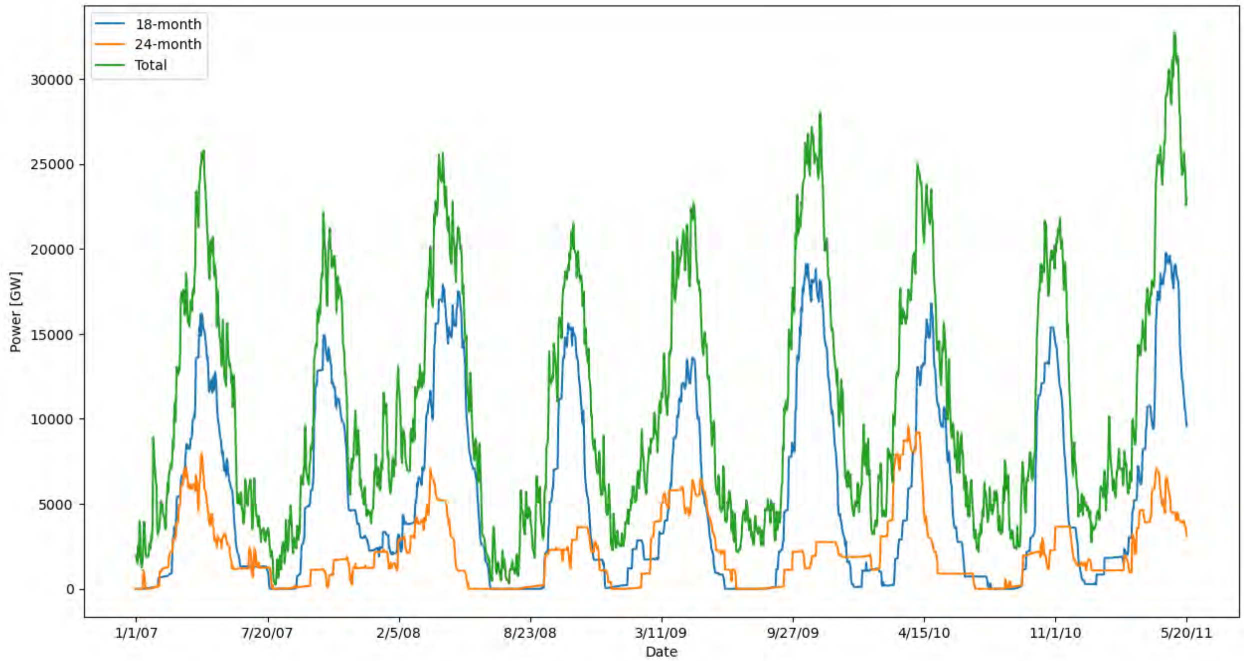


Figure 1. Impact of power generation lost due to outages for the U.S. fleet during the 2001–2011 time frame (data source: https://www.eia.gov/todayinenergy/detail.php?id=1490).

Figure 2 shows a more recent analysis of plant outages in the United States based on data available on the U.S. Nuclear Regulatory Commission website. It plots the number of reactors that underwent outages within the 2017–2024 time frame. Once again, the provided data do not differentiate between planned vs. forced (i.e., unplanned) outages. As also shown in Figure 2, it is rare not to have at least one reactor in the U.S. fleet undergoing an outage.

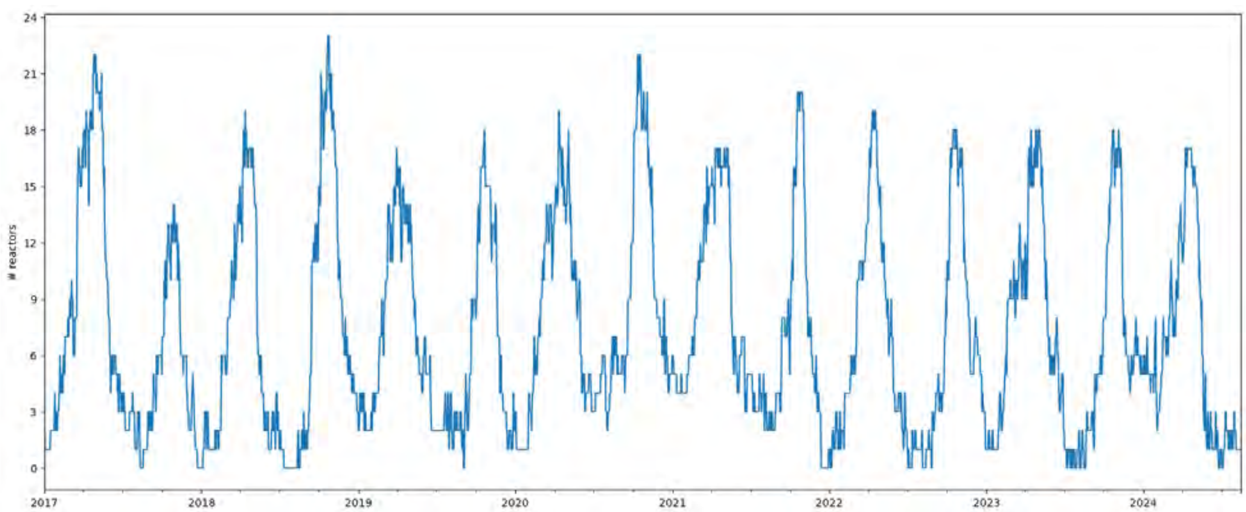


Figure 2. Number of U.S. reactors in outage on a daily basis during the 2017–2024 time frame. (data source: U.S. Nuclear Regulatory Commission power reactor status reports available at https://www.nrc.gov/reading-rm/doc-collections/event-status/reactor-status/2024/index.html).

To maximize plant availability, plant owners can plan for the following (not mutually exclusive) paths:

1. Reduce the risk of incurring outage delays
2. Reduce the time to perform an outage (e.g., by planning for shorter activity durations and by optimizing the resources available during outages)
3. Reduce the scope of an outage (e.g., by reducing the number of activities to be performed or by performing certain activities while the plant is online)
4. Decrease the frequency of plant outages (e.g., from 18 to 24 months)
5. Reduce the risk of unplanned outages.

This project directly tackles the first two paths by providing methods and tools to perform the following:

1. Assess completion times for outage tasks, based on past outage experience.
2. Identify critical elements in the outage—elements that may pose a risk of outage delays.
3. Identify the risk associated with unplanned activities that may arise during a planned outage.

It is important to highlight that this project is directly linked to another LWRS-RISA project (i.e., Advanced Modeling and Data Analytics for Optimization of Plant Processes, which focuses on the third and fifth paths listed above). This project has the scope to assess asset and system health from available monitoring data in order to provide information to system engineers regarding when certain maintenance activities are truly required.

3. OUTAGE SCHEDULE ANALYSIS USE CASES

The scope of this project addresses certain challenges pertaining to plant schedulers (St. Germain, 2013; St. Germain, 2015) in order to assess the critical points in planned outage schedules and to optimize resources (time and crews). In this respect, Table 1 provides an overview of the uses cases that have been tackled. In addition, each use case explicitly indicates the required data and the developed methods. Note that it is outside the scope of this project to develop methods for building plant outage schedules—as plant owners already rely on tools such as Primavera P6—but rather to analyze the critical points of the developed schedule.

Table 1. Use cases and required outage data.

ID	Use case	Required data	Methods
1	Assess activity duration variance	Activities performed during past outages (description and actual duration values)	NLP methods (see Section 6)
2	Assess schedule resilience (time related)	• Planned outage schedule • Planned activities (description and actual duration values) • [optional] Planned activity duration variance (see use case 1)	Schedule resilience methods (see Section 7)

ID	Use case	Required data	Methods
3	Post-outage analysis	• Planned outage schedule • Actual outage schedule • Planned activities (description and actual duration values) • Actual performed activities (description and actual duration values)	Post-outage analysis (see Section 5)
4	Analysis of unexpected events	• For each past outage: - Planned activities - Actual performed activities - Issue reports (IRs) - Work orders (WOs)	NLP methods (to be addressed in fiscal year [FY]-25)
5	Assess schedule resilience (crew related)	• Planned outage schedule • Planned activities (required crews and actual duration values) • Outage resource availability	Schedule resilience methods (to be addressed in FY-25)

One challenge when analyzing plant outages is data availability from plant owners, especially when dealing with use case 3. In this situation, the required data elements (e.g., IRs and WOs vs. outage activities) are stored in different databases not necessary linked to each other. The availability of such links is actually essential to assess the likelihood of unexpected events in future outages. Figure 3 provides a schematic view of the links between an outage unexpected event, the generated IR, and the corresponding WO.

An example of this type of scenario may be maintenance staff performing a clean-and-inspect WO on a circuit breaker cubical and noticing a cracked fuse block that must be replaced before returning the component to service. This will create a new WO that needs to be added to the schedule.

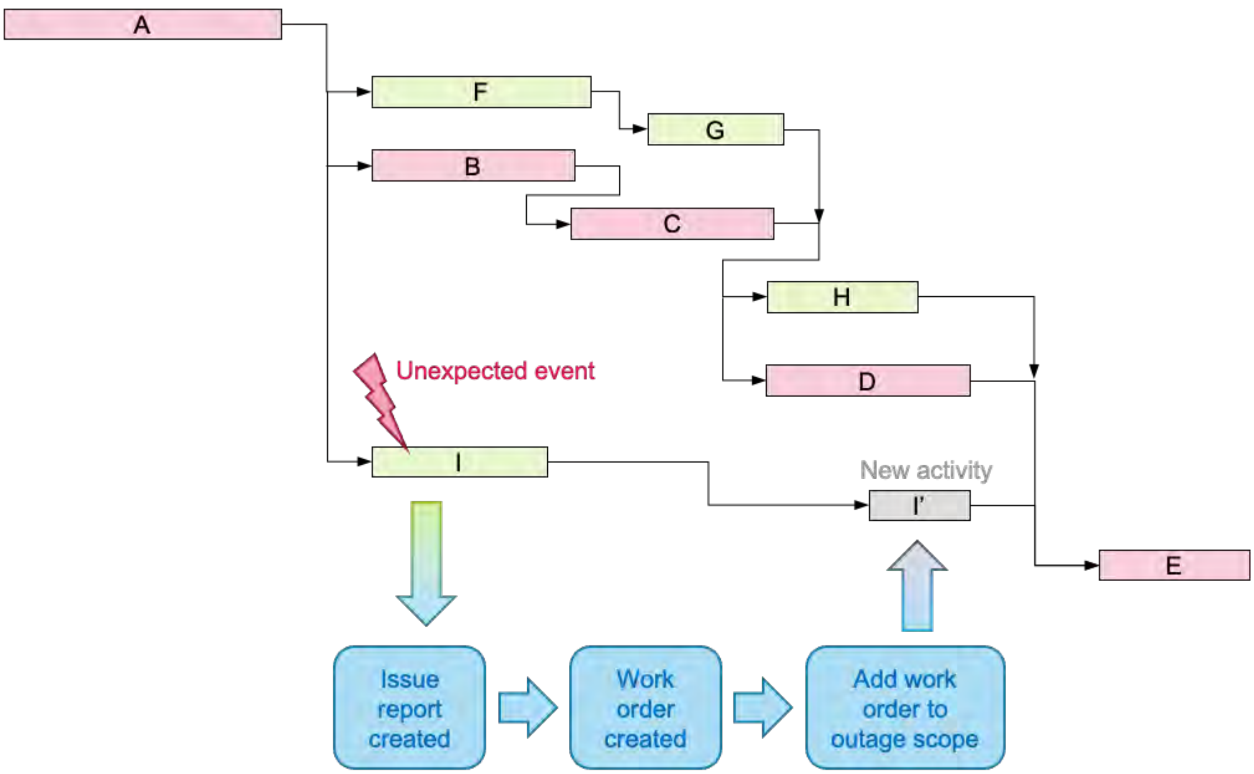


Figure 3. Relations among outage unexpected events, IRs, and WOs.

3.1 Plant Outage Scheduling

The process of scheduling and planning a plant outage is very complex, and typically begins 1–2 years prior to the actual start of the outage. The process requires a large amount of manual labor, even though computational tools are available for managing and scheduling the vast number of activities to be performed during an outage.

Another vital element in effectively planning an outage is the actual data required to assess what type of activity should be performed, and when. The set of data elements required for each activity is provided in Table 2.

Table 2. Required data for each planned outage activity.

Activity data element	Description
Duration	The estimated duration for each activity (typically in hours and minutes)
Resources	Required type of resources required to complete the activity (e.g., electrical, mechanical, scaffolding)
Dependencies	The set of activities that need to be completed prior to beginning the considered activity. Note that dependencies may be very heterogenous (e.g., system-logic driven, tech-specs driven, plant-risk[2] driven, or based on plant conditions)

[2] Typically, plant risk considerations are modeled through the plant probabilistic risk assessment (PRA) model.

With the data listed above, schedule optimization tools developed in this project can lay out the outage plan on a day-to-day basis, including the subset of activities to be completed each day and the plant personnel required.

Scheduling optimization tools such as Primavera P6[3] are designed to develop an outage schedule provided the above-listed data, such that the overall outage duration is minimized yet all the activities are completed. This minimization process carefully balances (through the schedule optimization algorithm) the activity duration, activity dependencies, and available resources.

The predicted completion time for a given outage activity is of particular interest, and is calculated by first determining the critical path (CP), which is the longest path (temporally speaking) from the initial to the final activity.

Other elements of interest—calculated after the optimization process—are how early/late an activity can start/finish. These values are often indicated as earliest start time, latest start time, earliest finish time, and latest finish time (see Figure 5). Lastly, for activities that are not part of the CP, the parameter *total float* (TF) is calculated. This parameter indicates the degree to which an activity can be extended before it becomes part of the CP (see Figure 4). Similarly, for activities that are part of the CP, the parameter *drag* is calculated. This parameter indicates the degree to which such an activity can be reduced before it gets moved out of the CP (see Figure 4).

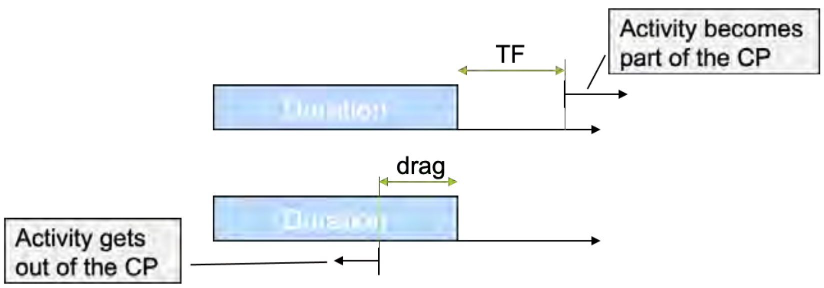


Figure 4. Graphical representation of the TF and drag associated with an activity.

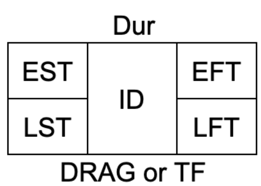


Figure 5. Summary of the defined and quantified parameters associated with an activity.

The above-described process of outage scheduling planning and optimization, also known as the CP method, is fairly mature and has been widely used in the nuclear industry. However, a few criticalities can emerge when this approach is applied in a real context:

- Activity duration is typically considered a point value, when in reality the actual duration is a variable based on past operational experience. Uncertainty in the activity duration can be expressed in various ways, such as:
  - Provide the average duration value accompanied by a measure of its variance.
  - Bound the range of possible duration values by the observed minimum and maximum duration.

[3] https://www.oracle.com/construction-engineering/primavera-p6/.

- There are multiple sources of duration uncertainties, such as number of people and the skills of the assigned crew, operational conditions (e.g., weather), and time of day when the activity is performed.
- Activity duration may be affected by the emergence of an event (which can be stochastic in nature) that must be addressed prior to completing the activity. This can impact the actual activity completion time.
- New activities can materialize once the outage has started. The emergent activities must be incorporated into the schedule, including dependencies with other activities.

4. OUTAGE SCHEDULE DATA RETRIVAL FROM PRIMAVERA P6

As was mentioned, Primavera P6 is a project management software for scheduling and tracking project activities, resources, and costs. Summarized in Table 3 are the required activity data elements and specific Primavera P6 parameters. Each of these parameters is tied to an activity characterized by an Activity ID and Activity Name. The ID is a code intended to efficiently refer to the activity, whereas the Name is a description of the activity itself. The activities are also categorized by a work category (WCAT) code. The various WCAT codes associated with plant operations are summarized in Table 4.

Table 3. List of required data for each planned outage activity and the specific Primavera P6 parameters.

Activity data element	Primavera P6 Parameters
Duration	Start (Scheduled) Finish (Scheduled) Actual Start Actual Finish Original Duration (Scheduled) Actual Duration
Resources	Resources
Dependencies	Predecessors Predecessors Details Successors Successors Details

Table 4. List of WCAT codes and their descriptions.

WCAT	Description
PM	Preventative Maintenance
CM	Corrective Maintenance
MM	Minor Maintenance
FO	Facility Operations
PO	Projects
RU	Routine
CO	Construction
OU	Outage
SH	Shipments

The duration for each activity is determined by its start and finish times. The scheduled and actual start/finish times are used to calculate the scheduled and actual durations, respectively. The resources are listed in comma-delimited fashion. The dependencies for each activity are defined by predecessors (parents), successors (children), and their respective details or relationships. The details are coded with respect to the start and finish of the activity and its predecessor or successor. Table 5 summarizes the detail codes, where *A* is the activity, *P* is the predecessor, and *S* is the successor. From a project perspective, the predecessor and successor data for each activity is repetitive. For example, if Activity 1 is a predecessor to Activity 2, Activity 1 will list Activity 2 as a successor and Activity 2 will list Activity 1 as a predecessor with identical detail codes.

Table 5. Dependency detail codes and their applications.

		Application to the Activity (*A*)	
Code	**Description**	**Predecessor (*P*)**	**Successor (*S*)**
FS	Finish to Start	*P* must finish before *A* can start	*A* must finish before S can start
FF	Finish to Finish	*P* must finish before *A* can finish	*A* must finish before S can finish
SF	Start to Finish	*P* must start before *A* can finish	*A* must start before S can finish
SS	Start to Start	*P* must start before *A* can start	*A* must start before S can start

These project activity data can be viewed or displayed in a layout table where each row is an activity and each column is an associated parameter. This layout, shown in Figure 6, is customizable to various activity data elements of interest. To customize it, click on the drop-down menu at the upper-left-hand corner of the table, then select “Columns.” This will open another window, enabling you to select which parameter columns you would like to add (see Figure 7). You can expand the “Available Options” categories by clicking the “+” icons. Select specific parameters and then click “►” to move them to the “Selected Options” list. To change the order of the columns, use “▲” and “▼.” Once you complete your selections, click “Ok” to close the column selector window, then your layout table will update itself. To export the layout table, click anywhere within the table, select all the contents (Ctrl + A), and then copy (Ctrl + C) and paste them into an Excel file. All the column headers will be pasted into that Excel file along with the data themselves.

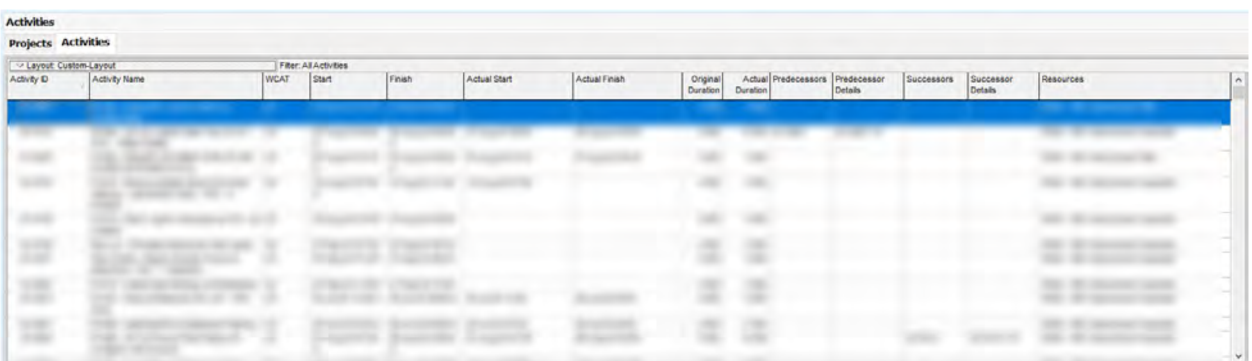


Figure 6. Primavera P6 layout table.

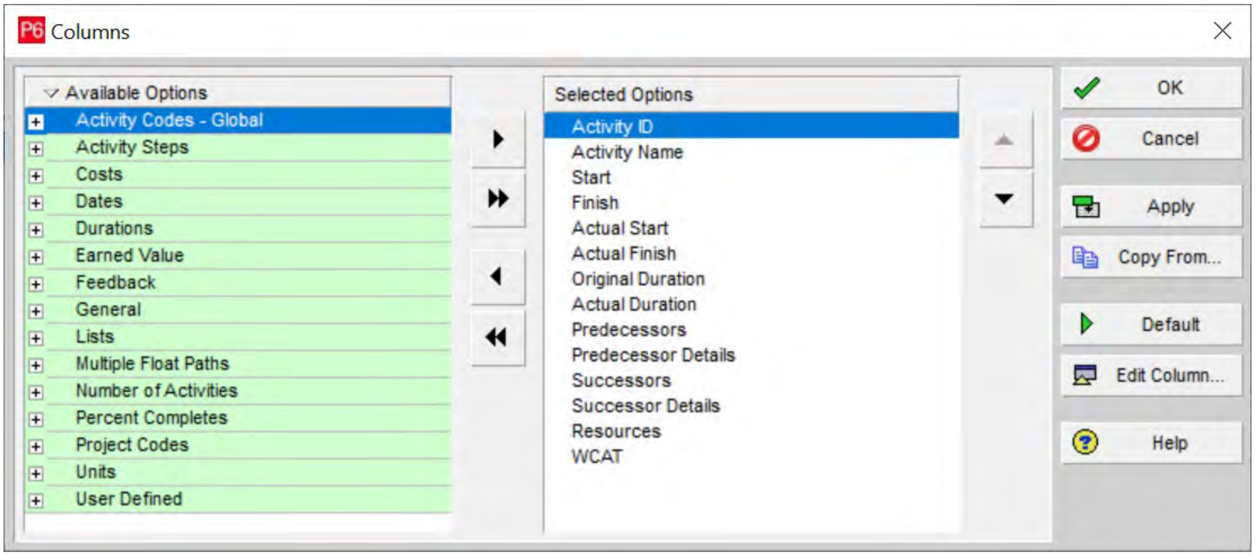


Figure 7. Primavera P6 column selector.

While the layout is the most efficient way to gather and then export your data of interest, other export options also exist. From the main Primavera P6 toolbar, select File, Export, etc., to open a window of file types via which to export your project information (see Figure 8).

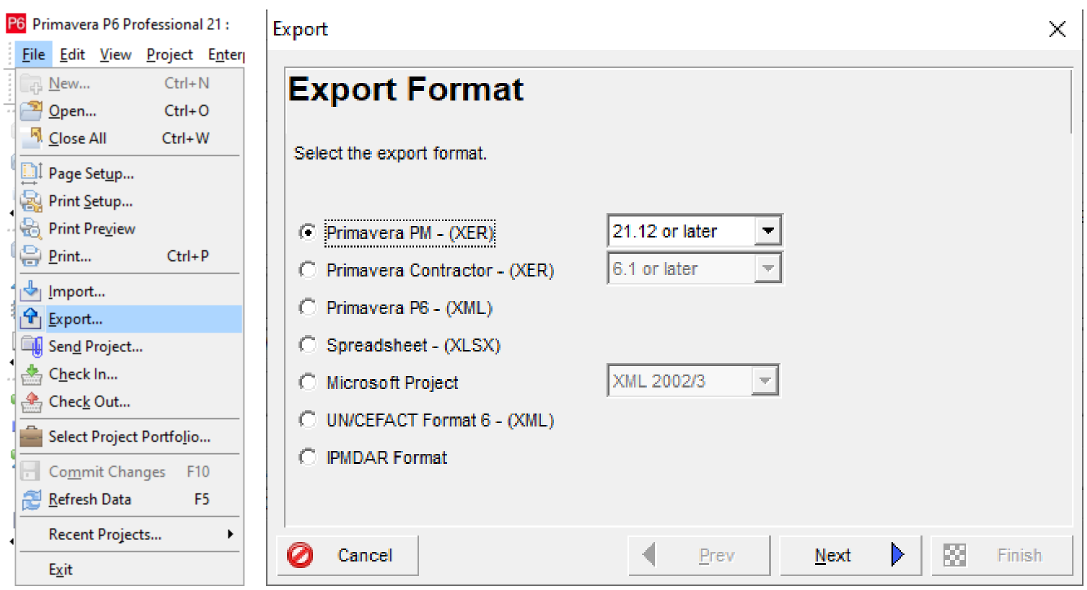


Figure 8. Primavera P6 exporting.

The different formats all offer various advantages, but none of these advantages are greater than the above extraction method to accurately and directly extract data for use in other tools or contexts. The XER formats allow for data sharing with Primavera software (e.g., moving data between projects or between software, such as from P6 to Contractor). The difficulty with the XER format is that all the data are labeled using the Primavera variable name, which is not an obvious indicator as to the meaning of the actual parameter. The XLSX format also allows for export to Excel, but its disadvantage lies in the ease of formatting the spreadsheets. The Microsoft Project option would allow you to transfer the data for use in Microsoft Project, but this must be done carefully, as the data types do not precisely align with each other. In particular, the dates would need to be checked. The XML option is a software-independent format that could be used to transfer the project scheduling data if it is convenient to be in XML format. This is also disadvantageous in terms of using the Primavera variable names. The IPMDAR format is useful for reporting Department of Defense acquisition requirements,[4] but such reporting is not required for this project goal.

5. ANALYSIS OF OUTAGE DATA

As indicated in Section 3, the outage data contained in P6 provide a large amount of information regarding activities performed, timing values, and resources employed. As an example, Figure 9 shows the distribution of the planned outage activities (left) (i.e., before the outage actually started) against their actual distribution (right) (i.e., after the outage was complete). To hide any proprietary data, the actual distribution was altered and the dates intentionally hidden. The right-hand plot shows the planned outage time window in green, whereas the activities performed outside the planned time window are located in the red region. From this plot, note that few activities were performed prior the actual start date, whereas about 300 activities were performed after the planned outage end date. These activities were distributed over a 6-day period, which corresponds to the observed outage delay.

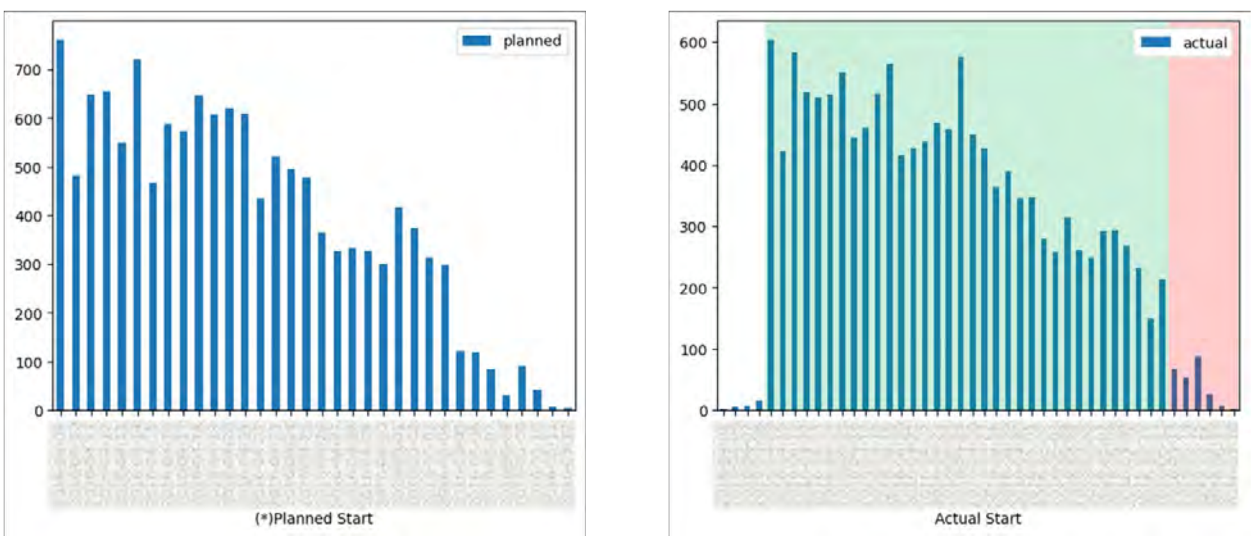


Figure 9. Distribution of activities throughout the outage: planned (left) vs. actual (right). The planned outage time window is shown in green on the right-hand plot; the activities performed outside the planned time window are highlighted by the region shown in red.

[4] https://tensix.com/exporting-primavera-p6-professional-files/

If planned and actual outage schedule datasets are available from P6, then it is possible to determine the following: the set of planned activities that were and were not completed, and the set of unplanned (also referred to here as emergent) activities that were completed during the same outage. Such calculations can be performed via basic set theory operations. From the outage data shown in Figure 9, such calculations were performed. They were then condensed in the form of a Sankey plot (see Figure 10).

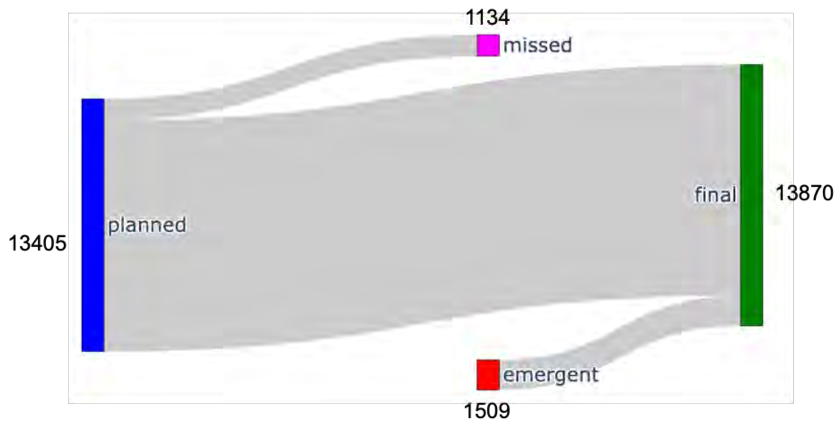


Figure 10. Sankey plot of the activities that were either planned, unplanned, missed, or completed.

In FY-25, a focus for this project will be to analyze the nature of the emergent activities in order to understand the risk they pose, thus enabling them to be fully accounted for when planning future outages. For this task, the planned activity that will actually trigger the unexpected one must first be identified. Unfortunately, the outage data stored in P6 do not contain that information. On the other hand, another source of information can be the plant databases that contain the IRs and corresponding WOs generated throughout the outage. At this point, the emergent activity can be traced back to the originally planned activity by looking at the corresponding WOs and IRs, along with the system/component and date of the planned activities (see Figure 3).

Another analysis step on which we are focusing is analysis of the resources employed throughout the outage. In this respect, the P6 data provide information on the activity type, which can be linked to the actual resource employed to complete the activity. The set of types is plant-dependent, though certain commonalities do exist. While this analysis might not be very useful when applied to planned activities, it provides insights when applied to emergent ones. For the outage shown in Figure 9, Figure 11 shows the distribution of emergent activity types after selecting only the emergent activities. The same analysis can be performed on a day-by-day basis, as shown in Figure 12. From here, outage schedulers can observe the distribution associated with activity types for past outages, identify trends, and proactively plan additional resources for upcoming outages.

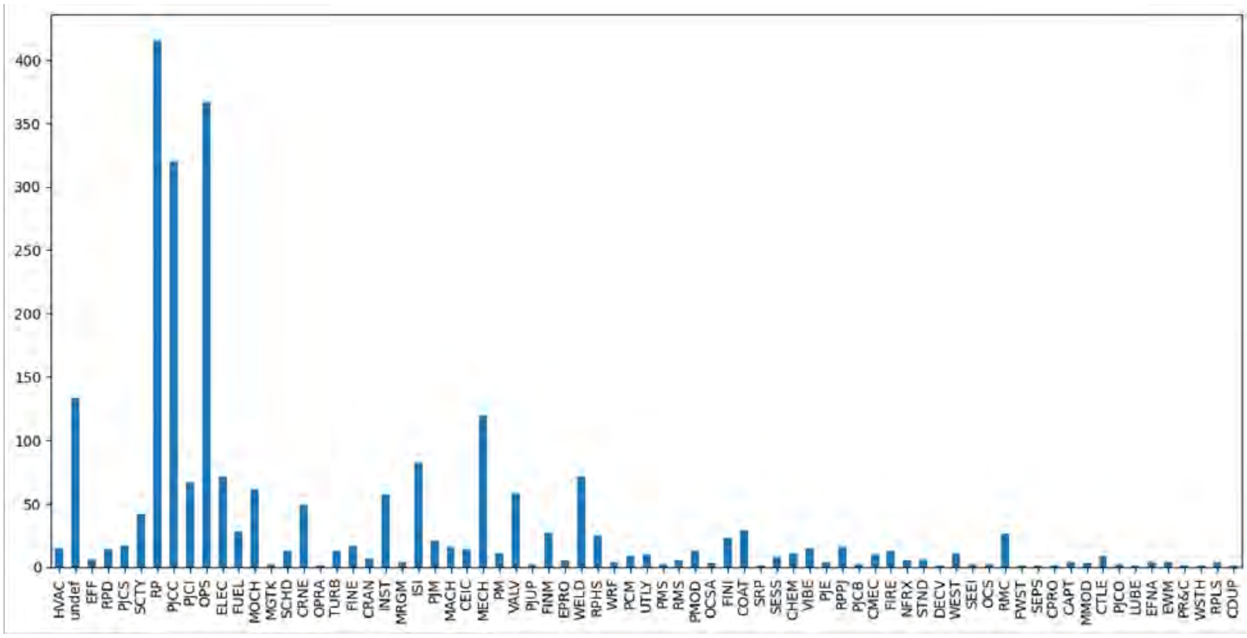


Figure 11. Distribution of emergent activities by type.

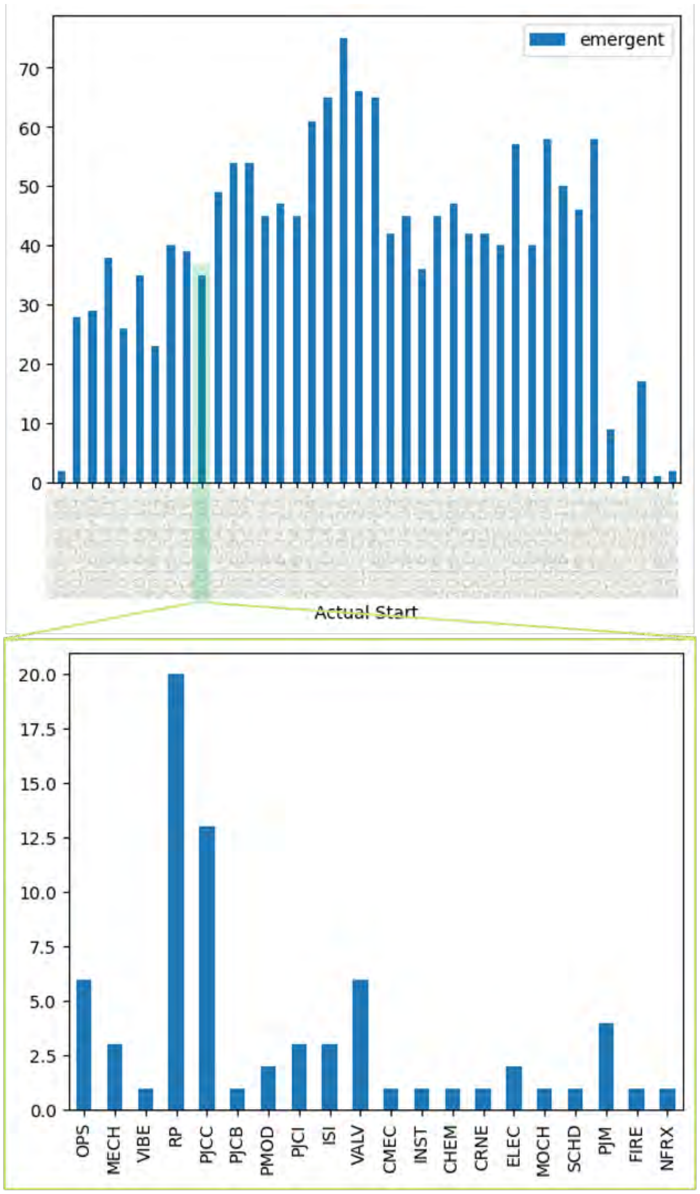


Figure 12. Distribution of emergent activities throughout the outage (top plot), and daily distribution of emergent activities by type (bottom plot).

6. ANALYSIS OF ACTIVITY DURATION VARIANCE

This section presents two approaches that utilize NLP algorithms to determine the variance associated with activity completion time.

6.1 Knowledge-based Approach

This approach uses text semantic similarity to evaluate activity completion time. The basic idea is to identify, from the subset of activities performed in previous outages, activities similar to the one being queried. The temporal distribution of the queried activity can then be determined by collecting the historical completion time from the subset of past activities.

An example of textual similarity is shown in Figure 13, which compares two activities with similar semantic meanings, bringing up the importance of data cleaning and curation. The example provided in Figure 13 suggests that, were we to perform a simple word-to-word similarity comparison between those two activities, they would prove very dissimilar. On the other hand, if the historical activity were to be cleaned (e.g., through spell checking and abbreviation identification and expansion), it would be transformed into "[ACC01-B] PRESSURE TRANSMITTER CALIBRATION." Consequently, the two activities would be very similar. The elements required for the semantic similarity analysis are:

- The set of past outage activities. This set might be partitioned across several datasets, one for each outage. The outage of different plant units, different plants, or different utilities can be gathered to improve the analysis results.
- A computational method designed to compute the semantic similarity between two activities (i.e., the queried and the historic activity) would generate a point value that measures how similar the two activities are. An important note here is that the computational time for such a method must be very small, as the similarity search for a queried activity in a database of tens of thousands of past activities must be performed within minutes.

For this project, we focused on developing the semantic similarity method and testing it on several outage databases (Li, 2003; Li, 2006). The following sections detail the development and present a high-level overview (to mask proprietary data) of the obtained results.

Figure 13. Example of semantic similarity between a queried and a historical outage activity.

Word, sentence, and document similarity analyses are an active part of recent NLP method development and play a crucial role in text analytics (e.g., text summarization and representation, text categorization, and knowledge discovery). A wide variety of methodologies have been proposed over the last two decades. For the most part, these techniques can be classified into five different types: the lexical knowledge-based approach, statistical corpus approach (word co-occurrence), machine learning (ML) and deep learning approach, sentence-structure-based approach, and hybrid approach. However, the major drawbacks common to these approaches are that they are computationally inefficient and lack the desired automation, adaptability, and flexibility. In the present research, we try to address these drawbacks by developing a tool that is generally usable for any application requiring a similarity analysis.

As shown in Figure 14, we are leveraging parts of speech (POS), disambiguation, lexical database, domain corpus, word embedding and vector similarity, sentence word order, and sentence semantic analyses to calculate sentence similarity. POS is used to parse a sentence and tag each word and token with a POS tag and syntactic dependency tag. This information provides sentence syntactic structure data (i.e., negation, conjecture, and syntactic dependency) for guiding the similarity measuring process. The disambiguation approach is employed to determine the best sense of a given word, especially when coupled with a specific domain corpus.

Next, a predefined word hierarchy from the lexical database (i.e., WordNet) is used to compute the word similarity. However, some words are not contained in the lexical database, since it only connects four POS types: nouns, verbs, adjectives, and adverbs. Moreover, these words are grouped separately and do not feature interconnections. For instance, nouns and verbs are not interlinked (i.e., the similarity score between "calibration" and "calibrate" is 0.091 when using WordNet). In this case, ML-based word embedding is introduced to enhance the similarity calculation. For the "calibration" and "calibrate" example, the similarity score instead becomes 0.715. The next step is to compute sentence similarity by leveraging both sentence semantic information and syntactic structure. The semantic vectors are constructed using the previously introduced word similarity approach, while the syntactic similarity is measured based on word order similarity.

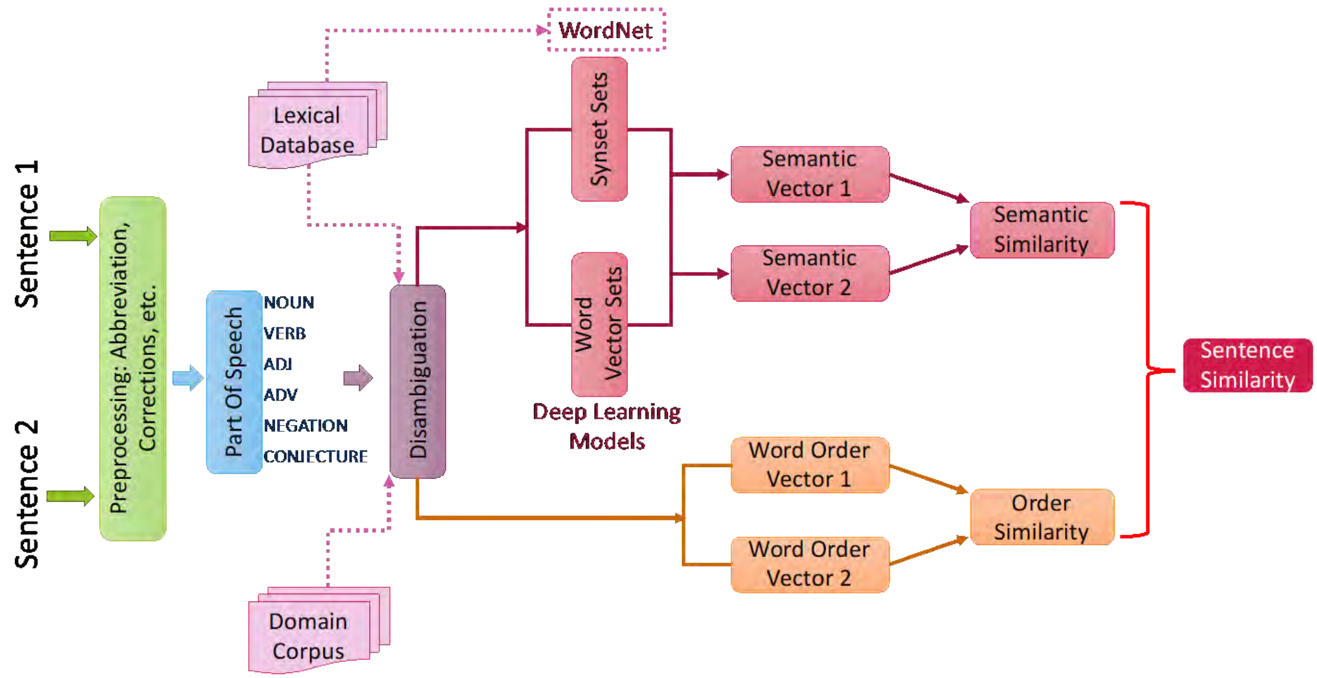


Figure 14. Illustration of the sentence similarity calculation.

Here, we wish to highlight the importance of data curation (e.g., cleaning and reconstruction) of the textual elements that describe each activity, and how it might impact the search for similar activities. More specifically, the data curation process for all historical outage activities entails the following steps:

1. *Remove component IDs*. Specific asset or system IDs (e.g., Accumulator ACC-01B in Figure 13) do not necessarily provide any type of information from a semantic point of view, meaning they can be removed from the actual text by either parsing the activity text or providing the algorithm with a list containing the full assortment of plant asset or system IDs. During our testing, such a list was unavailable, and we relied on an empirical method designed to remove all words containing a mixture of characters, numbers, and symbols.

2. *Handle abbreviations*. NPP outage activities usually correspond to short sentences that often contain abbreviations, negatively impacting the ability to extract knowledge from such texts. Hence, we developed an NLP pipeline designed to identify abbreviations and fully spell them out. The starting point is a library of abbreviations collected from documents available online. This library is basically a dictionary that relates an identified abbreviation to the corresponding set of words. A challenge here is that a single abbreviation may have multiple words associated with it. Similarly, there may be multiple ways to reduce a given word. Abbreviations are handled in each sentence by first identifying misspelled words, each of which is then searched in the developed library. If an abbreviation in the library matches the misspelled word, it is replaced by the corresponding complete word. If no abbreviation in the library matches the word, we proceed by searching for the closest abbreviation. If multiple words match the obtained abbreviation, the one that best aligns with the sentence context is chosen.

3. *Run a spell check*. After the abbreviations have been handled, the remaining misspelled words are parsed through our spellchecking methods for a final correction.

Once the historical plant outage data have been cleaned, the similarity value between the queried activity and each historical activity is determined. This results in an array of similarity values with dimensionality identical to the number of historical activities and the corresponding array (with identical dimensionality) containing the activity durations.

The duration of the queried activity is computed by considering both the similarity and duration arrays. More precisely, setting a similarity threshold (typically in the 0.7–0.9 range[5]) enables us to collect only those elements of the duration array that have a corresponding similarity measure greater than 0.7. Of note here is that if the queried activity has never been completed in past outages, no similar past activities with a similarity value above 0.7 will be found. This approach does not, in fact, involve performing any type of regression.

An example application for the developed methods was identified by using a dataset provided by an existing U.S. NPP. This dataset covered activities performed over the course of five outages (see Figure 15), and all the data it contained were cleaned. A relevant feature of the provided dataset is that some activities were categorized using plant-specific labels. A label indicates the type of work performed in an activity (e.g., electrical, chemical, or instrumentation and control). Note that a large portion of outage activities (about 30%) are unlabeled. For those activities, the label NaN was assigned. In total, about a hundred unique labels were identified.

Figure 16 gives an example of knowledge-based analysis. For a new activity being queried, a similarity value is associated with each activity contained in the database of past outages. As already mentioned, the similarity value is between 0 and 1; however, activities with very cryptic descriptions (e.g., characterized by many unknown acronyms and/or very few—if any—actual English words) may lead to a NaN similarity value that simply indicates that no actual similarity value can be determined.

From here, the first step in the analysis is to make a 2-D plot of each past activity in terms of activity duration vs. similarity (see Figure 17). Such a plot will show in graphical form the outcome of the search for similar activities, and how the queried activity fits within the set of past completed activities. In particular, it provides insights into the selection process for the subset of similar activities that should be considered. In the example given in Figure 17, provided the queried activity, about 20 similar completed activities (a similarity value of about 0.8) are highlighted in the red box.

---

[5] Recall that a similarity measure is in the (0,1] range where perfect similarity is indicated with the unitary value, while very low similarity values (near 0) will be assigned for dissimilar textual elements.

Next, it is possible to select the subset of similar activities and collect their duration values. This type of process generates a histogram representing the duration variance so as to complete the queried activity provided past outage data (see Figure 18). Given these results, the actual duration of similar activities may then be able to be statistically analyzed to identify possible outliers obtained from the similarity search, track the historic trend for activity completion time, and evaluate the impact of employed human resources on completion time.

	Activity Name	Original Duration	Actual Duration
0	[redacted]	1.00h	1.00h
1		0.00h	0.00h
2		7.00h	7.00h
3		167.00h	167.00h
4		6.00h	6.00h
...		...	...
384		5.00h	5.00h
385		5.00h	5.00h
386		24.00h	24.00h
387		18.00h	18.00h
388		32.00h	32.00h

Figure 15. Snapshot of imported data generated from five past outages.

	Activity Name	Original Duration	Actual Duration	Cleaned Name	Processed Name	Synset Conversion	Similarity
0	[redacted]					[Synset('align.v.01'), Synset('stead.n.01'), S...	0.112481
1						[Synset('scram.v.01')]	0.000000
2						[Synset('while.n.01'), Synset('mobilize.v.04')...	0.000000
3						[Synset('radiotherapy.n.01'), Synset('waste.n....	0.770492
4						[Synset('fuel.n.01'), Synset('cooling_system.n...	0.000000

Figure 16. Snapshot of the NLP analysis (last column) outcomes. A similarity value is associated with each past activity.

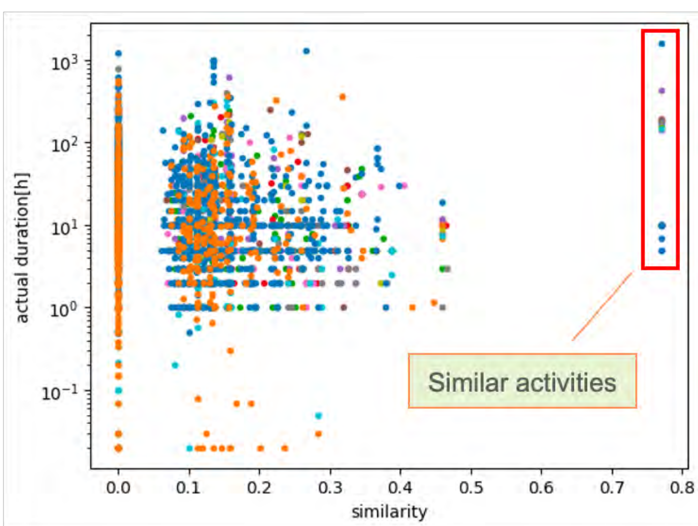


Figure 17. Outcome of the knowledge-based NLP analysis. Past activities are shown in a 2-D plot (actual duration and similarity). Those activities with high similarity values are highlighted in the red box.

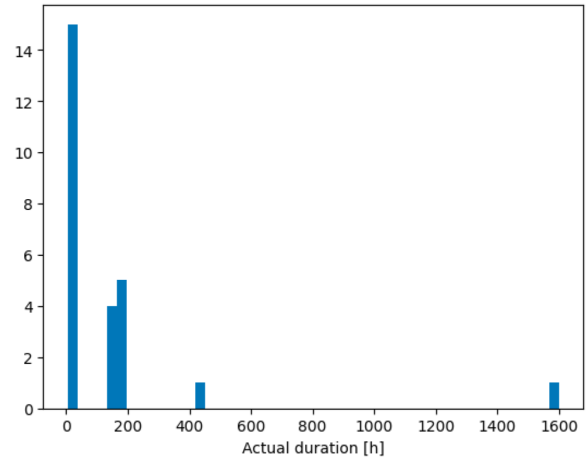


Figure 18. Histogram representing the duration variance in completing the queried activity, based on the similar activities reflected in Figure 17.

6.2 Natural Language Processing and Regression-based Approach

For this section, a text vectorization method that relies on word frequencies was used in tandem with a regression model to predict the activity durations. More specifically, Term Frequency-Inverse Document Frequency (TF-IDF) and ridge regression methods were used. Ridge regression is a form of linear regression model that address multicollinearity, meaning it models word interdependencies. TF-IDF is expected to be able to outperform a simple bag-of-words approach but does not add context to the words. When paired with ridge regression, the regression model directly correlates the words' occurrences to time estimates. The following sections discuss each of those methods separately. Furthermore, two common concepts in ML are discussed: k-fold cross validation and hyperparameter tuning. K-fold cross validation validates the results by iteratively isolating parts of the dataset for testing, while using the rest to train a model. Hyperparameter tuning involves finding the optimal set of tunable model parameters so as to achieve the best performance. R^2 (i.e., a common metric for measuring regression model performance) was used in this effort.

6.2.1 TF-IDF

TF-IDF is a tool for expanding the dimensionality of text. The IDF part represents the rarity of words in a given corpus. The TF part represents word frequencies in a given document. TF is weighted by IDF, which helps measure the importance of a word in a document (relative to a collection of documents) and removes frequent words that add no context (e.g., stop words). TF-IDF provides a vector for a document (i.e., an activity, in this application), meaning a list of numbers that is equal in length to the vocabulary size of the document corpus. Each number represents an item in the vocabulary. One key aspect of vectorizing words is the use of n-grams. An n-gram is a concatenation of words into a single word representing the items in the vocabulary. For instance, the text snippet "remove motor bearing life" contains four unigrams (words), and we see the bigram "motor bearing," the trigram "motor bearing life," etc. TF-IDF was chosen such that unigrams, bigrams, trigrams, and 4-grams are all considered. Hereafter, this concept is referred to as "up to 4 grams."

6.2.2 Ridge Regression

Ridge regression is an established technique for dealing with the multicollinearity that occurs in regression problems. In this case, the regression problem relates word frequencies to time needed. For example, the words “engineering” and “evaluate” could both correlate with time needed, and thus this multicollinearity would be alleviated by ridge regression. According to (McDonald, 2009), ridge regression has proven valuable to applied statisticians for over three decades. Typically, every word’s TF-IDF value is multiplied by a coefficient; these coefficients are the n-gram weights. Another unique property of ridge regression is smaller n-gram weights in comparison to ordinary least squares regression. Smaller n-gram weights are desirable because they are a form of regularization. Regularization is typically beneficial in ML (Yingjie and Zhang, 2022), has been defined as “any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error (Goodfellow, Bengio and Courville, 2016),” and is “one of the central concerns of the field of machine learning, rivaled in its importance only by optimization” (Goodfellow, Bengio and Courville, 2016). This implies that reducing training error is not the sole goal of ML, as it typically does not improve performance on new test data; however, decreasing the generalization error does.

6.2.3 k-Fold Cross Validation

A bias exists when using the entire dataset for training and testing. This is because the performance on the training set is not a good indicator of predictive performance on new data. This concept is referred to as over-fitting (Bishop, 2006). Normally, to avoid such over-fitting, an 80/20 split is performed, meaning that 80% of the data are allocated for training and the remaining 20% are allocated for testing. The disadvantage of this type of split is that 80% of the data are not used in the testing set and 20% are not used for training the model. But there is a limited supply of data, and using as much as possible improves the quality of the results. To overcome this challenge, cross validation is performed. In 5-fold cross validation, five different 80/20 splits are used. Each of these splits tests on a different 20%. There are four training segments in each fold and one testing segment. By rotating through the segments, all the data receive a prediction without bias. The folds can overlap (i.e., the 20% used for testing could overlap with the other 20% sets, or may be fully explicit).

6.2.4 Hyperparameter Tuning

In hyperparameter tuning, model variables are changed to improve the model’s ability to represent the data patterns. For a small set of variables, a grid search with cross validation can be used, as this is not computationally expensive. Ideally, hyperparameter tuning should show consistent performance for a hyperparameter adjustment. A bias may result from hyperparameter tuning, due to a human examining the data to find an optimal hyperparameter, then using the optimal hyperparameters to test the same data. In this effort, the ridge hyperparameter alpha was tuned. Additionally, the number of n-grams used was tuned as a hyperparameter, starting with the TF-IDF default setting, which is unigrams.

Continuous regression models often use R^2, which is a statistic that directly measures model predictions against known values in order to determine goodness of fit. It normally ranges from 0% to 100%. R^2 is interpreted as follows: values closer to 100% mean the predictions are identical to the expected output, whereas values of 0% indicate no better than random predictions. A value of 50% is not random, but means that half the data variability has been explained and half has not.

6.2.5 Dataset

Given the limited number of words used in outage activities, a surrogate dataset was utilized instead (i.e., condition report data from a nuclear utility). The condition report data contain longer text, as well as the target prediction for the time taken to resolve the condition. The hypothesis tested using this dataset is that outage task durations can be forecasted if they were longer (by means of adding work management data). The utilized condition reports corresponded to dates ranging from 2017 to 2024. Cleaning was done to remove negative and zero durations (due to data quality issues). Any condition report without either an origination date or a completion date was also removed. This still left over 70,000 condition reports post-cleaning, resulting in a vocabulary of over 16 million unique n-grams Each condition report was based on free-form text, thus representing the type of data typically available in outage records and indicating that fields containing categorical data (e.g., condition report priority) were not used.

6.2.6 Results

Figure 19 shows the result of applying TF-IDF and ridge regression. It maps the actual and the predicted number of days reflected in a condition report—from origination to completion—in a logarithmic scale. The ideal line (i.e., a line with a slope of 1 and an offset of 0) is shown in black. The fitted line is shown in red, and a 95% percentile interval is shown in orange. For the percentile interval, a sliding window of 1,000 condition reports' predicted duration was used. The line fitness resulted in a R^2 of 33% between the predicted and the actual number of days reflected in the condition reports (in log scale).

The hyperparameter bias was found insignificant. The ridge hyperparameter alpha was tuned and very little change was noticed, thus its default hyperparameter was not changed. The results also showed that using up to 4 grams resulted in a 22% increase in R^2 in comparison to the default unigram setting. Moving to 5-grams did not improve performance. Also, little difference was seen between up to 4 grams and up to bigrams/trigrams.

In Figure 20, 1,000 condition reports were simultaneously fitted—using a sliding window of 1,000 condition reports—and the resulting R^2 plotted. The results show that the initial predictions achieved an R^2 of ~30%–45% over the first several years, after which the model degraded, with some time periods showing performance no better than random chance ($R^2 = 0$).

6.2.7 Discussion

From this effort, it was concluded that even when longer text inputs were provided, the observed slope of the fitted line (1.12) suggested that the correlation was close to the ideal fit (with a slope of 1). However, due to the log scale of the fitness metric, accurately predicting the work duration proved challenging. Several factors may have contributed to this:

- Contextual Information: The most likely cause is related to context. Describing a condition without giving the necessary background information could hinder model performance. Similarly, without context, outage activities may vary greatly in time. For example, overpacking a minor valve in a plant may take significantly less time than handling a major one. Failure of the activity description to capture these nuances can affect model accuracy. The same issue was observed with conditions.
- Data Quality: The quality of data significantly impacted model performance, as evident from the degraded R^2 performance over time, suggesting that the quality of the condition reports may have deteriorated.
- Complexity of Prediction: Predicting completion times for conditions is much harder than for outage activities, especially because conditions involve multiple organizations, decisions, and actions. This results in larger variations, even for identical equipment under identical conditions.

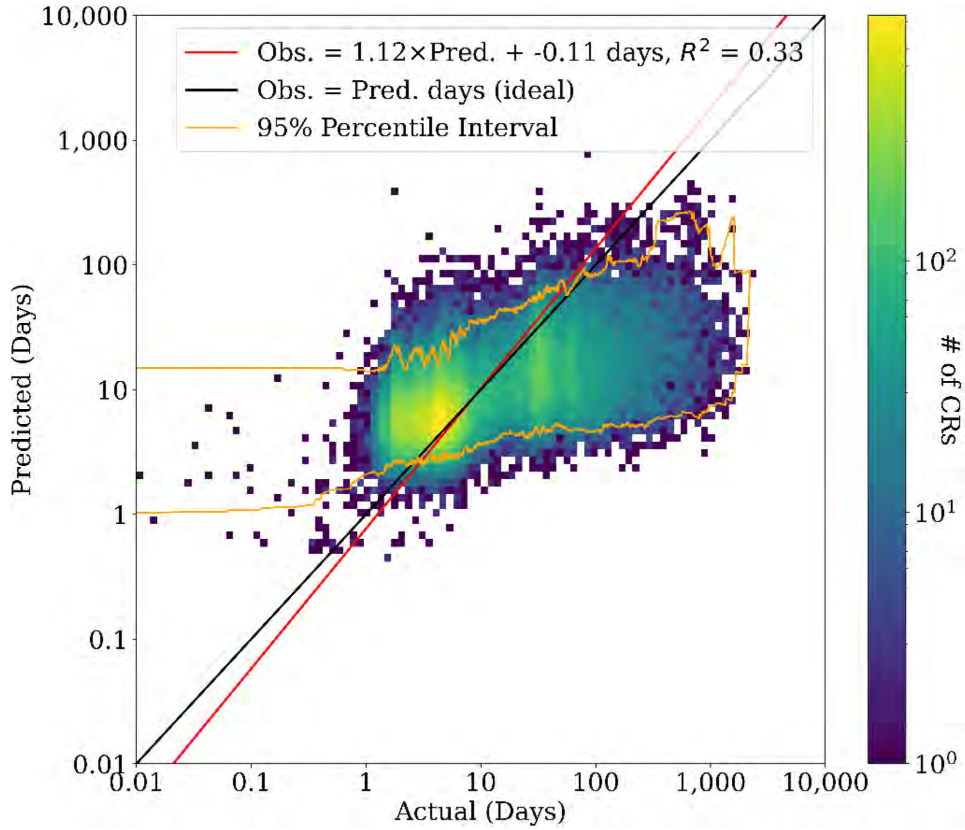


Figure 19. Actual vs. predicted origination-to-completion time (in days) for condition reports.

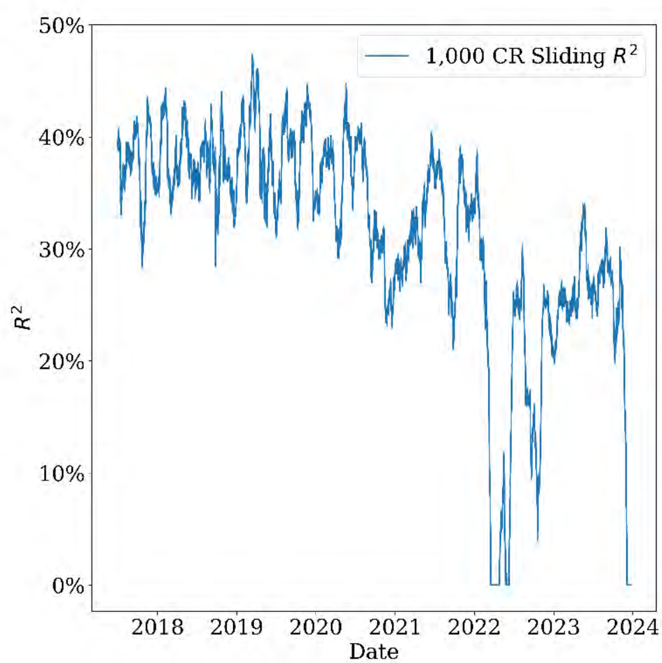


Figure 20. R2 for a sliding window of 1,000 condition reports.

7. ANALYSIS OF OUTAGE SCHEDULE RESILIENCE

Once the duration variance associated with a subset of activities in a planned outage schedule has been determined via the methods explored in Section 6, the same variance can be propagated through the outage schedule in order to assess the duration variance of the full outage schedule, using the methods shown in (Mandelli, 2023) (see Figure 21). Furthermore, the float values of those activities not part of the CP can be used to assess the fraction of the activity duration variance that directly impacts the outage schedule (i.e., delays), as shown in the duration variance histogram associated with activity *F* (displayed at the top of Figure 21).

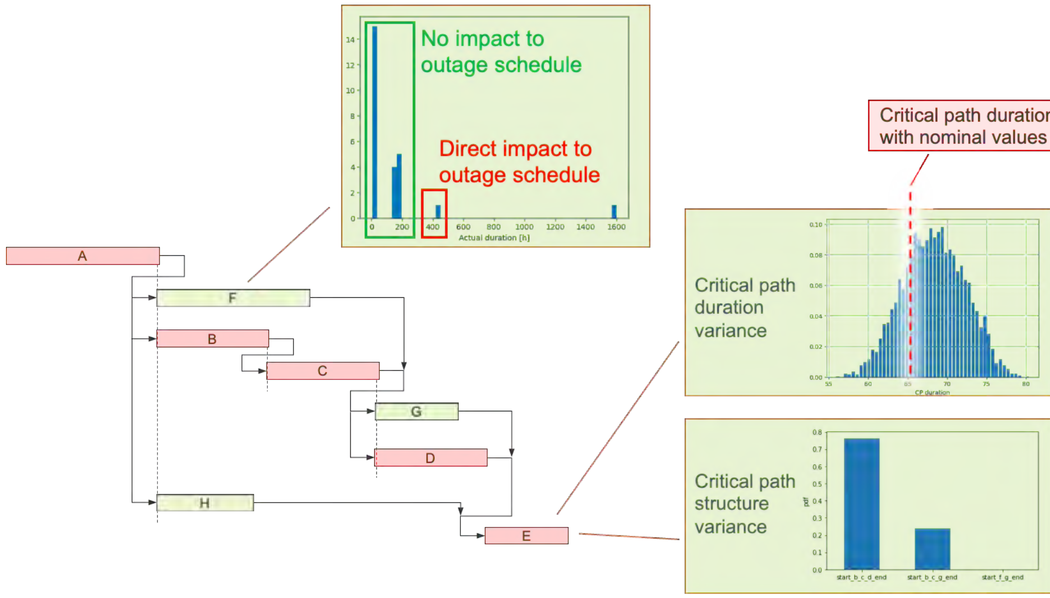


Figure 21. Integration of activity duration variance into the outage schedule so as to assess CP duration and CP structure variance.

A typical plant outage touches almost every part of the plant, as thousands of maintenance and surveillance activities must be performed by a large number of crews. Consequently, the structure of a typical outage schedule can be very complex and overwhelming to manually analyze in order to understand which critical elements of the schedule may cause delays. Most schedule analysis methods based on the CP methods rely on the float value associated with activities that are not part of the CP. However, even the process of ranking activity importance based on float values entails the following disadvantages:

- It still ranks tens of thousands of activities such that a fraction (e.g., hundreds) might have very small float values.
- It can be misleading because it does not consider the subset of activities that are part of the same sub-path.
- Resource availability (e.g., crews and material) might delay the start of activities, thus reducing the actual float value of such activities.

Here, we now move from a "one activity at a time" mindset to a "one sub-path at a time" mindset in order to tackle the three above-listed disadvantages via the following steps:

1. Reduce the outage schedule by combining activities that are in series. More specifically, two activities, $Act_1$and Act_2, can be combined into Act' ($Act' = Act_1 \rightarrow Act_2$), if Act_2 is solely dependent Act_1
2. Perform the CP calculation on the reduced schedule obtained via Step 1.
3. Determine the set of sub-paths from the reduced schedule. For each, calculate the portion of the CP that is parallel.

An example of outage schedule reduction is shown in Figure 22. Note that even activities that are part of the CP (highlighted in red) are also merged. In fact, the following activities have been combined: activities F and F', and activities B and C. At this point, Step 2 (described above) can be performed by assigning a duration value for each set of merged activities equal to the sum of the duration values of all the activities contained within that set. Step 3 is actually performed using classical graph theory operations (a schedule can be in fact be viewed as an acyclic graph). The set of obtained sub-paths is shown in Table 6.

Next, the outage schedule can now be analyzed by comparing the duration of each sub-path with the duration of the parallel CP sub-path. Such an approach heavily reduces the number of elements that are part of that ranked list (i.e., sub-paths rather than activities). Furthermore, it provides an actual snapshot of the impact of the duration variation of multiple activities that are part of the same sub-path.

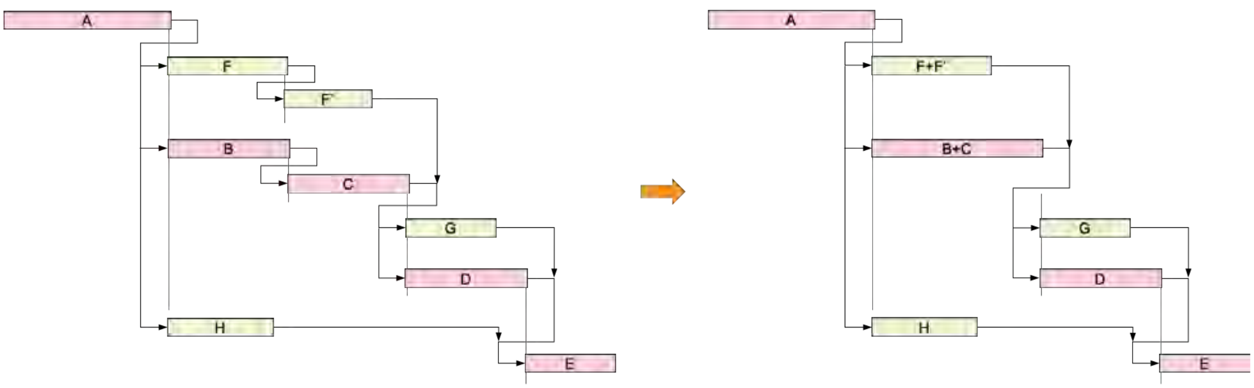


Figure 22. Examples of outage schedule reduction: original (left) and reduced (right).

Table 6. List of sub-paths and the corresponding parallel CP sub-paths for the example shown in Figure 22.

Sub-path	Parallel CP sub-path
F-F'	B-C
G	D
H	B-C-D

A slightly more complex example of sub-path analysis is applied to the schedule reflected in Figure 23, while the corresponding set of sub-paths is shown in Figure 24. At this point, schedule resilience analysis can be performed by:

- Calculating the TF of the determined sub-paths (rather than focusing on the TF values of each single activity)
- Ranking the importance (from a resilience point of view) of each sub-path, based on the TF value.

Note that the TF value associated with a sub-path clearly represents how "close" that sub-path (e.g., sub-path F-G) is to a sub-path that is part of the CP (i.e., B-C). Thinking in terms of sub-paths rather than activities greatly reduces the complexity of analyzing large outage schedules (hundreds of sub-paths vs. tens of thousands of activities). During a plant outage, outage managers can perform the following activities:

1. Monitor the status of each sub-path by updating its TF value by integrating all the observed delays stemming from those activities that belong to that sub-path.
2. Assign available resources to activities that are part of sub-paths that are closer to the CP.

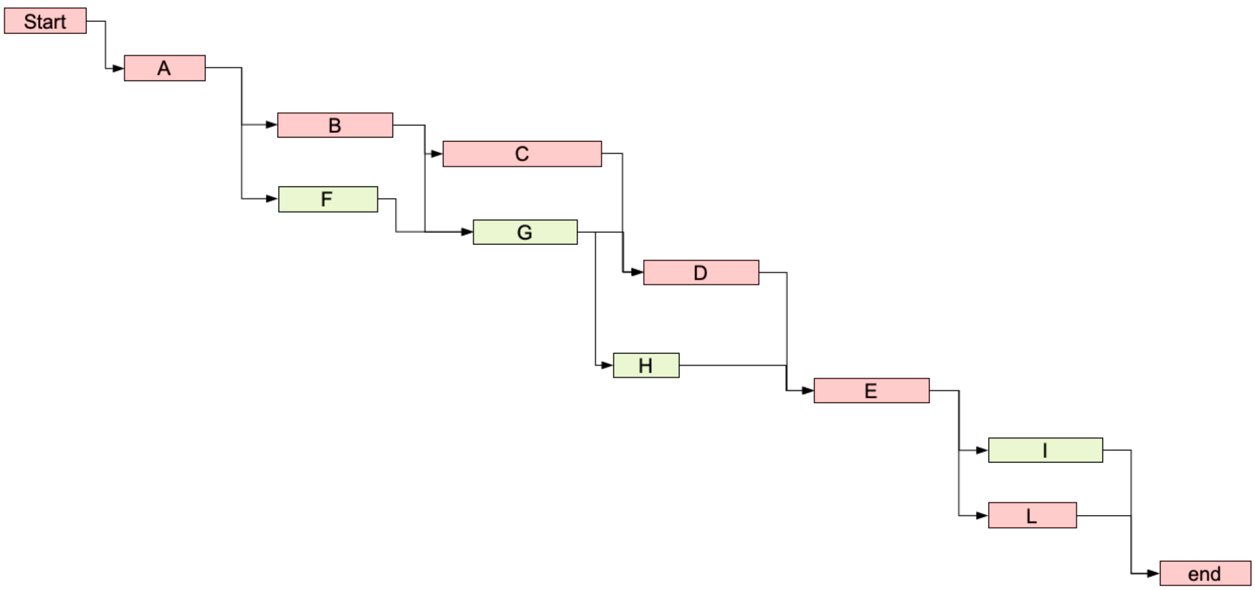


Figure 23. Example schedule for the sub-path analysis.

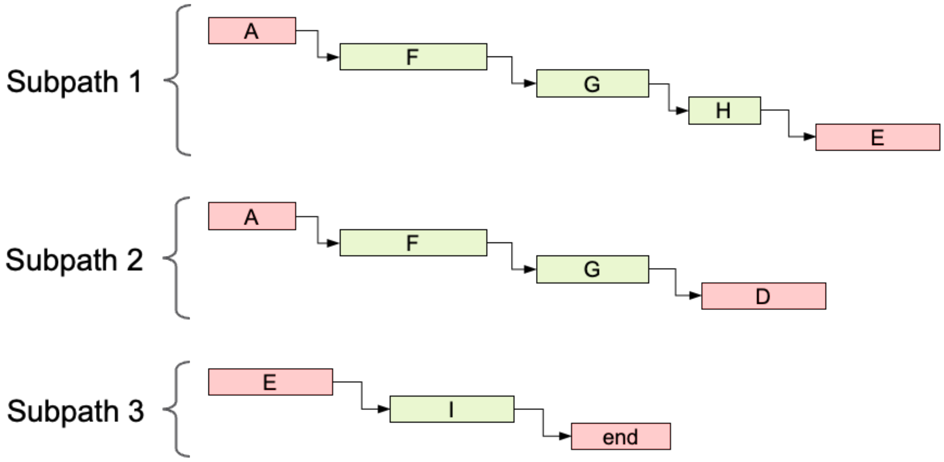


Figure 24. List of sub-paths generated by the sub-path analysis, given the example shown in Figure 23.

Table 7. Ranking the sub-paths obtained from the schedule shown in Figure 23, based on sub-path TF values.

Sub-path	Parallel Portion of the CP	Sub-path TF [h]
F-G-H	B-C-D	8
F-G	D-C	12.5
I	L	13

To test the developed computational methods on much larger-scale models, we developed a benchmark plant outage schedule based on the data available in the open literature, as well as on interactions with the industry partner. This outage schedule consists of 10,310 activities and contains no proprietary data nor any reference to a specific existing plant. In addition, the developed outage targets refueling along with standard maintenance and surveillance activities. In other words, it does not contain any major backfittings or plant modernization activities that usually considerably extend the outage length.

The developed benchmark outage consists of the following main elements:

- Reactor power operations (shutdown, cooling, repressurization, and restart)
- Core refueling and fuel shuffling operations
- Containment surveillance
- Primary loop surveillance and maintenance (e.g., valves, pressurizer, core vessel, instrumentation and control systems)
- Secondary loop surveillance and maintenance (e.g., chillers, pumps, heaters and steam generators)
- Electric grid maintenance (e.g., transformers, switchgears, converters, busses, batteries).

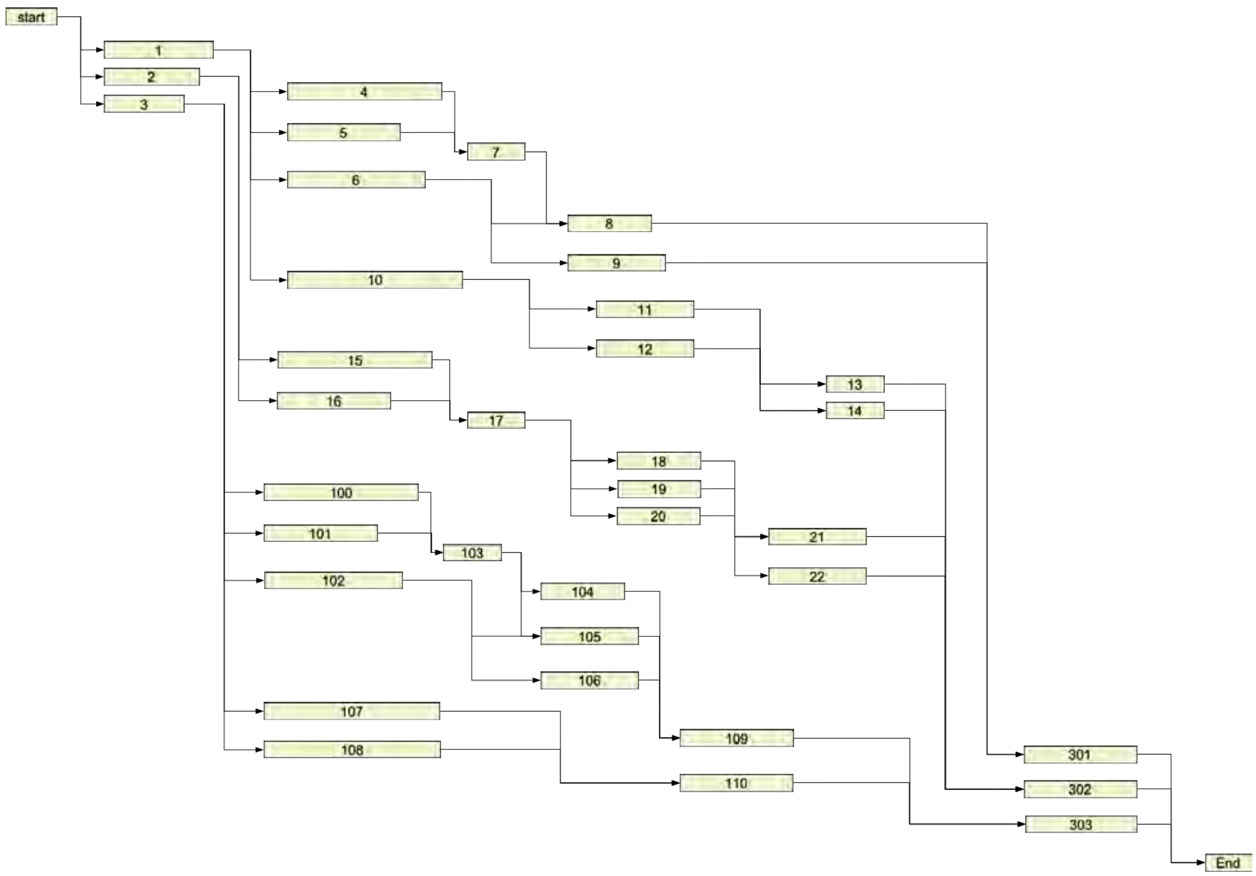


Figure 25. Representation of the developed test outage schedule following the sub-path reduction process.

7.1 Schedule Outage Analysis Tool

Given the analysis methodologies presented in the previous sections, a LOGOS user interface (UI) tool is under development to improve ease of usage. The UI is intended to serve as a streamlined outward-facing interface where users can easily upload and modify an outage schedule without having to interface with underlying source code. This includes preprocessing data in a command line terminal or via another such approach. This section presents the requirements, setup, and construction of the UI tool. Note that the presented UI is in the pre-alpha phase and that all the listed tasks require additional development before public release will be possible.

7.1.1 Requirement Specifications

High-level specifications for the UI were determined during its conceptual phase. These requirements identify the general operating and implementation environment of the UI. The listed requirements are given in Table 8, along with an accompanying description.

Table 8. High-level requirements for the LOGOS UI.

Identified Requirement	Description
Operating system compatibility with Windows & macOS	The UI should be accessible and usable on both the Windows and macOS systems.
Licensing compatibility with APACHE 2.0	The UI and all imported/external modules and libraries should be compatible with the APACHE 2.0 license.
Web-based architecture	The UI should be able to run on all major browsers, including Google Chrome, Microsoft Edge, Mozilla Firefox, and Apple Safari.
Generic hardware	The UI should not utilize any unique hardware aside from a desktop computer/laptop, a display monitor, a mouse, and a keyboard.

Functional requirements were also identified that specify how the UI is intended to function. Implementation of the functional requirements is separated into three phases. Phase 1, shown in Table 9, focuses on implementing the visualization component of the outage schedule within the UI. Column 1 provides a reference ID for requirements tracking, column 2 identifies the functional requirement, column 3 describes the requirement, and column 4 identifies the development status of the requirement. The development status corresponds to one of five categories: complete (C), under development (U), under design (D), and incomplete (I). Complete signifies that the requirement is implemented and ready for further testing and evaluation. Under development indicates it is under code development. This is in contrast to under design, which means the requirement is still under discussion and the exact function has yet to be finalized. Lastly, incomplete means the requirement has not been included in the UI and the requirements are too high-level to implement in code.

Table 9. Phase 1 functional requirements for UI.

ID	Functional Requirement	Description	Status
1.1	Import outage schedule	Loads the outage schedule from the JSON file	C
1.2	Export outage schedule	Saves changes to the outage schedule to a new JSON file	I
1.3	Visual outage schedule as Gannt chart	Displays the interactive Gantt chart and all activities included in the loaded file	C
1.4	Temporal location tracking	Activities displayed in the Gantt chart should be listed chronologically	C
1.5	Logical link tracking	Arrows between linked activities are present in the Gantt chart to illustrate dependency relationships	C
1.6	Highlight CP	The CP through the Gantt chart should be identified; both critical activities and critical links between critical activities should be unique	C
1.7	Group activities into categories	Activities should be grouped by category (i.e., project)	D

ID	Functional Requirement	Description	Status
1.8	Visual properties of an activity	When a user hovers over an activity, that activity's properties should be shown	C
1.9	Simplified Gantt chart	When a user specifies the option to simplify the Gantt chart, the displayed Gantt chart should be simplified	D
1.10	Customization of viewing date options	Allows users to specify the date range for the Gantt chart	C

Phase 2 of the UI development includes implementing the backend LOGOS analysis of the Gantt chart and connecting the output to the UI visualization functions. Phase 2 is critical to the UI, as it implements the analytical functionality desired for the UI. Due to the critical nature of Phase 2, most of the functional requirements identified are under detailed design so as to develop the correct functionality, and thus are not visualized on the UI.

Table 10. Phase 2 functional requirements for the UI.

ID	Functional Requirement	Description	Status
2.1	Calculate simplified Gantt chart	Simplifies the chart by combining a series of activities, and renaming them as a single activity; provides the simplified chart structure to the UI.	D
2.2	Group activities together by project	Allows activities to be grouped into similar categories or projects	D
2.3	Calculate outage uncertainty	Calculates the uncertainty in the project/activity end date	D

Phase 3 of the UI development includes allowing user modifications and interaction with UI components (e.g., the Gantt chart). Nearly all the functional requirements in Phase 3 are currently under design and will be considered low priority until the prior phases are complete. The functional requirements for Phase 3 are shown in Table 11.

Table 11. Phase 3 functional requirements for UI.

ID	Functional Requirement	Description	Status
3.1	Add new activity to the Gantt chart	Adds a new activity to the Gantt chart and causes it to appear in local data tables of the UI	D
3.2	Remove activities from the Gantt chart	Removes an existing activity from the Gantt chart and removes it from local data tables of the UI	D
3.3	Add new resource to the Gantt chart	Adds a new resource that can be selected by activities and causes it to appear in local data tables of the UI	D
3.4	Remove resource from the Gantt chart	Removes an existing resource from activities in the Gantt chart and removes it from local data tables of the UI	D
3.5	Edit activity attributes	Edits activity details, including duration, grouping, start/end date, and resource utilized	D

3.6	Edit activity dependencies	Edits which dependencies are part of an activity chain	D
3.7	Move activity in the Gantt chart in time along the temporal axis	Activities shown in the Gantt chart can be moved along the temporal axis in drag-and-drop fashion	I
3.8	Enable groupings (and ungrouping) of activities	Visual display option where activities can be grouped and ungrouped in the Gantt chart	D

Lastly, these functional requirements were identified for the pre-alpha version of the LOGOS UI. Thus, they currently lack a well-defined success criterion.

7.1.2 User Interface Implementation Platform

The UI was implemented on Anvil, a free web-based development environment based on Python. The Anvil app development environment is found at https://anvil.works/. All apps developed in Anvil can be freely used and do not interfere with the licensing requirements for the UI. Each Anvil app is comprised of three layers: visual UI for user interactions, client-side code that implements the visual aspects of the UI, and server-side code that implements private functions called from the client-side code. Two additional layers are implemented as necessary to support LOGOS deployment. An external code interface connects function calls from the Anvil app to the Risk Analysis Virtual Environment (RAVEN) interface, and RAVEN performs the analytical computations. The external code interface is necessary, as the Anvil environment is incompatible with RAVEN in its current deployment form. The interface was shaped via a feature in Anvil called Uplink, which treats the external code module as server code by assigning the code a server address to which Anvil can connect.

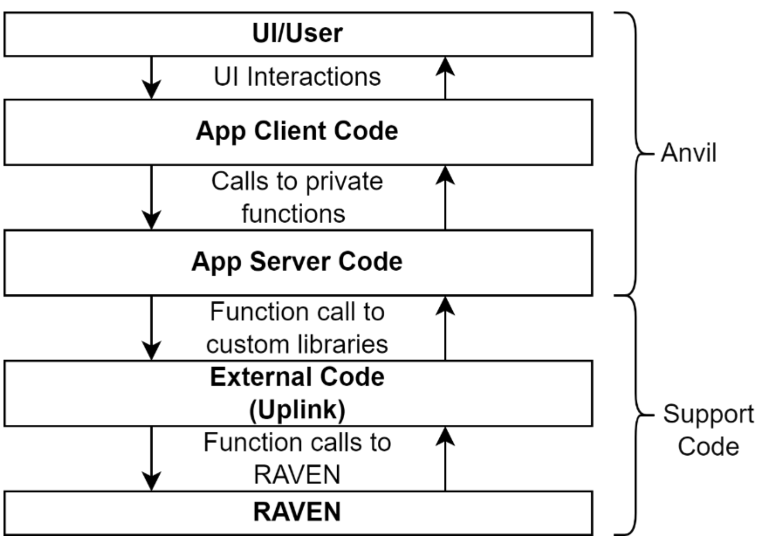


Figure 26. Software stack for the LOGOS UI.

7.2 Pre-alpha Demonstration of UI Functional Requirements

This section provides a demonstration of the UI. The reference ID for the identified functional requirements (Table 9 and Table 11) is marked on the UI so as to identify where it is being implemented. Note that Phase 2 functional requirements do not yet have a visual component in the UI and cannot be identified in the UI demonstration. The current UI has six menu tabs: Home (Figure 27), View Schedule (Figure 28), Gantt Activities (Figure 29), Activity Groups (Figure 30), and Gantt Resources (Figure 31). Organization of these pages is based on interpretations of the functional requirements specified in Table 9 and Table 11. Home is the landing page that users will first encounter. The left-hand side has a menu bar that allows for navigating to the different pages. View Schedule is the primary page to which users will navigate to observe the Gantt chart and select viewing options. Gantt Activities, Activity Groups, and Gantt Resources allow the user to modify the options and attributes for specified activities in the Gantt chart. The user can add, edit, and remove attributes via these pages.

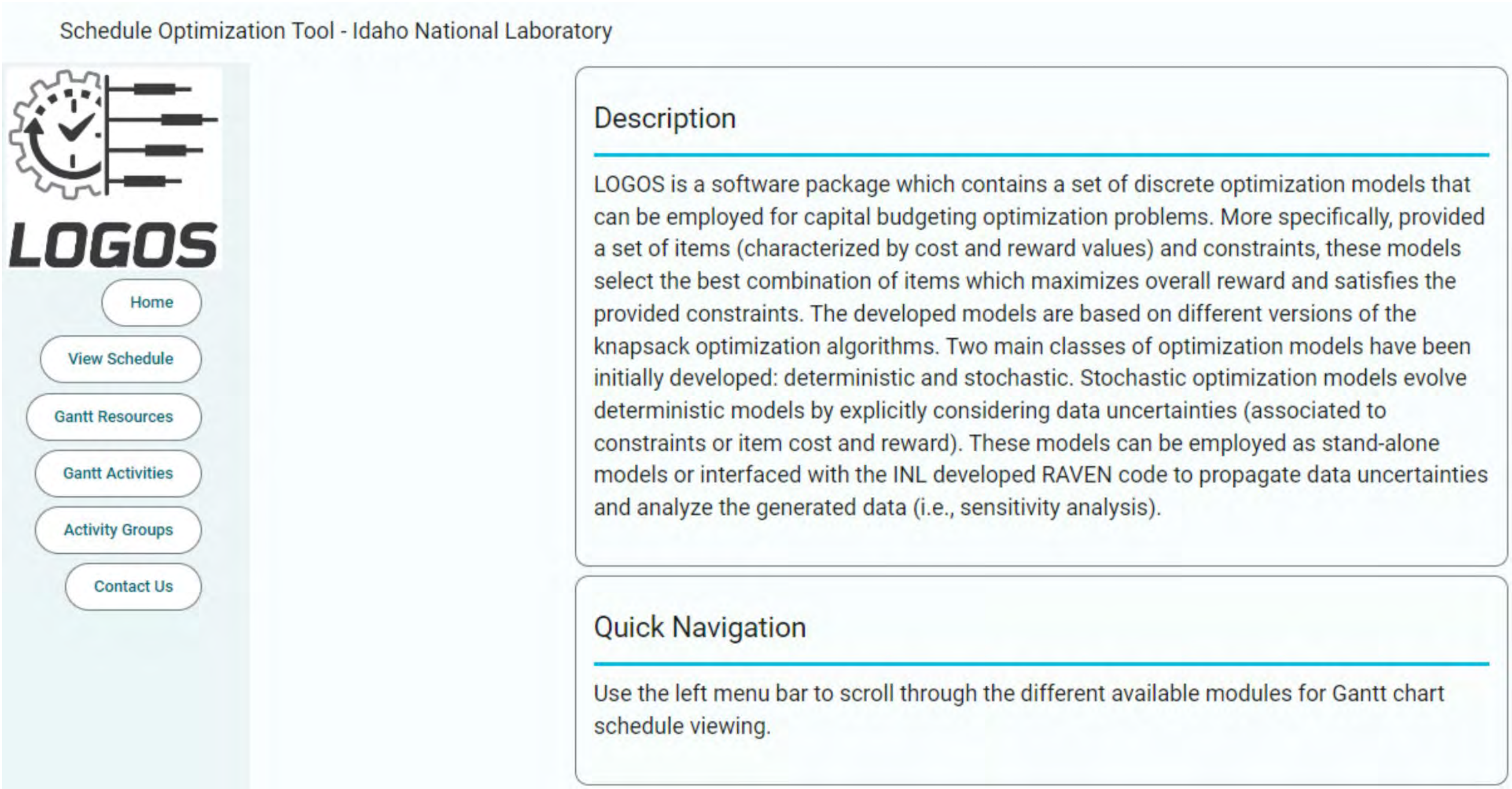
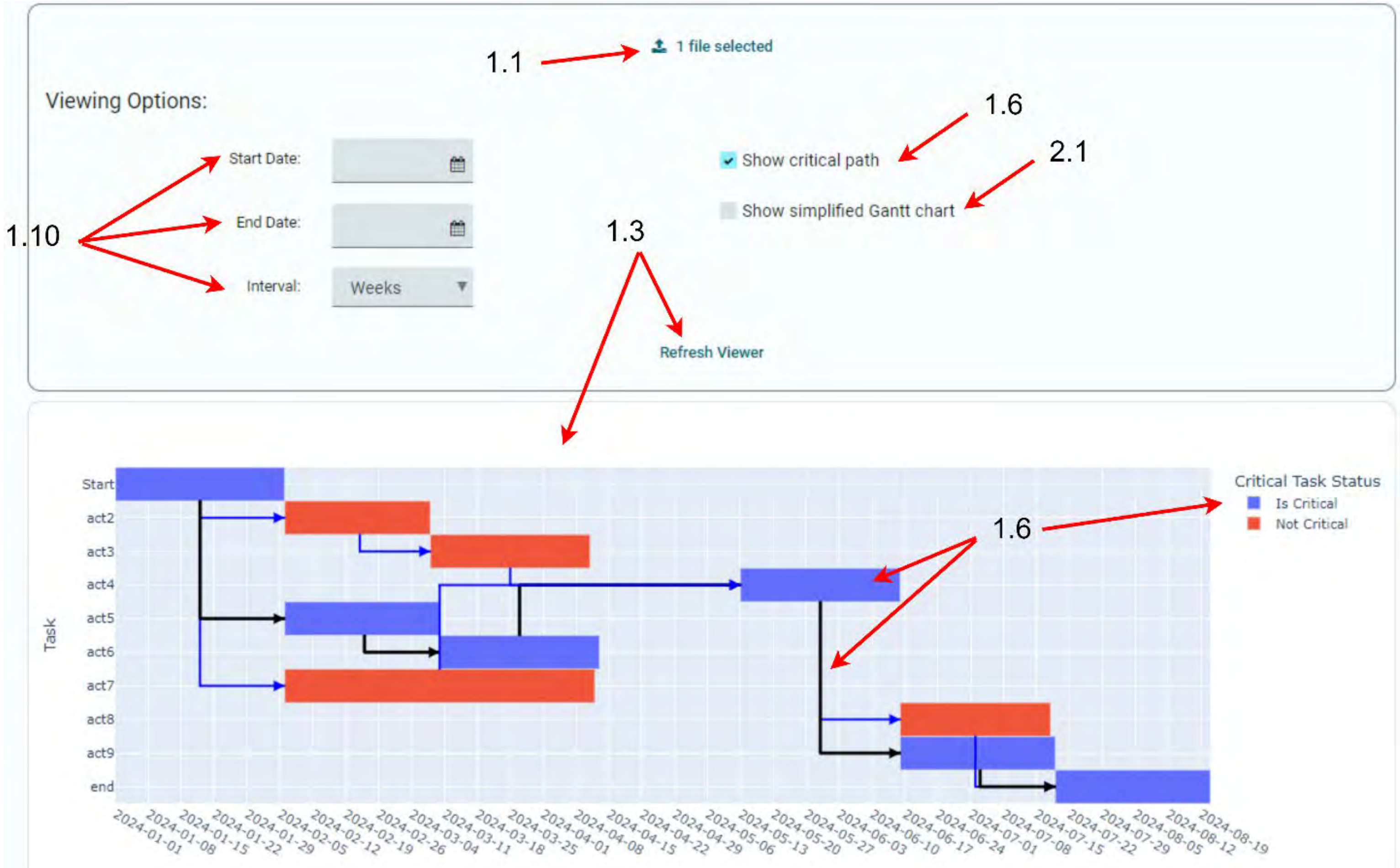
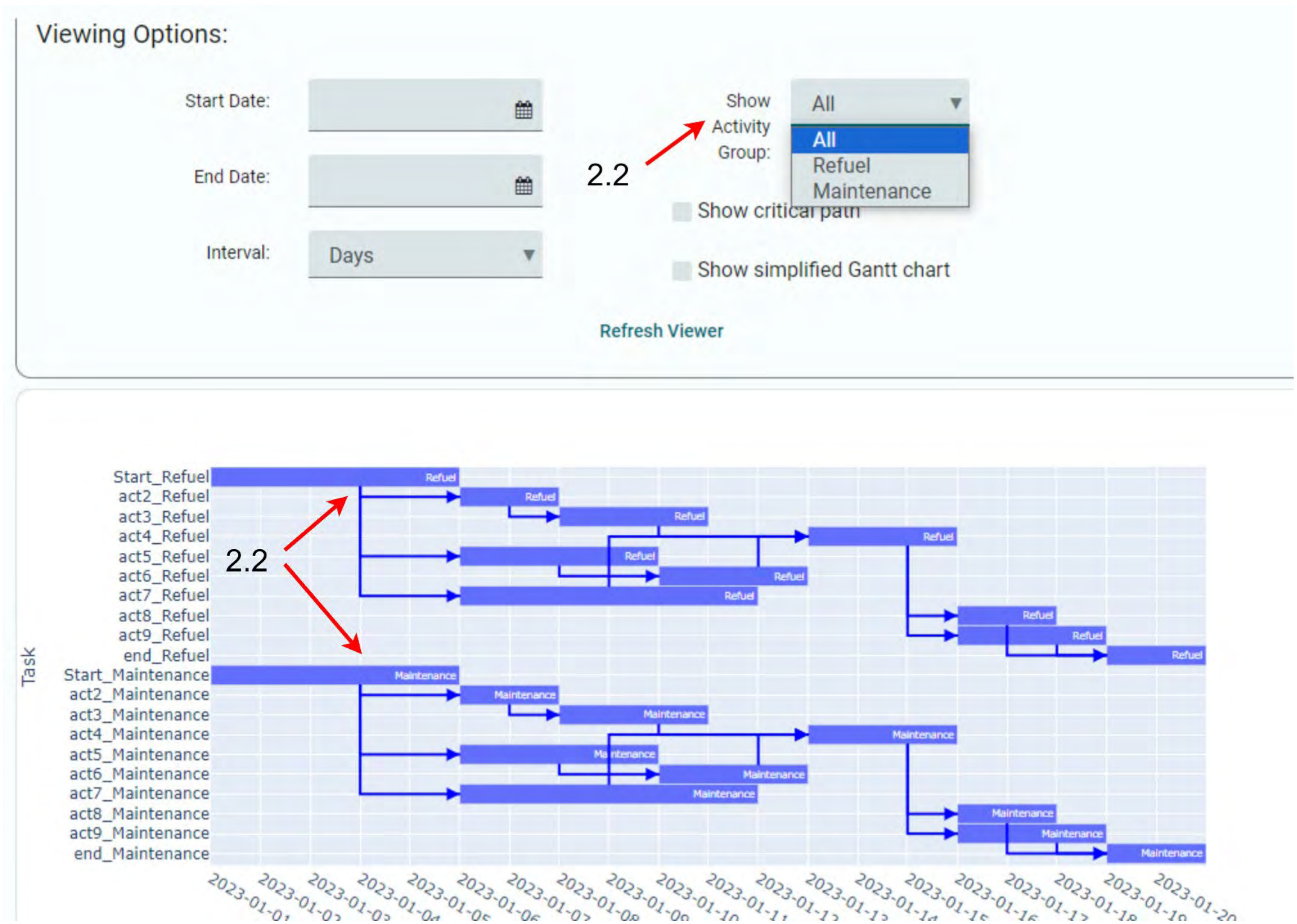


Figure 27. “Home” page of the LOGOS UI.


1.1
1 file selected
Viewing Options:
1.6
2.1
Start Date:
End Date:
Interval:
Weeks
1.10
Show critical path
Show simplified Gantt chart
1.3
Refresh Viewer
Start
act2
act3
act4
act5
act6
act7
act8
act9
end
Task
Critical Task Status
Is Critical
Not Critical
1.6
2024-01-01
2024-01-08
2024-01-15
2024-01-22
2024-01-29
2024-02-05
2024-02-12
2024-02-19
2024-02-26
2024-03-04
2024-03-11
2024-03-18
2024-03-25
2024-04-01
2024-04-08
2024-04-15
2024-04-22
2024-04-29
2024-05-06
2024-05-13
2024-05-20
2024-05-27
2024-06-03
2024-06-10
2024-06-17
2024-06-24
2024-07-01
2024-07-08
2024-07-15
2024-07-22
2024-07-29
2024-08-05
2024-08-12
2024-08-19

(a)



(b)

Figure 28. "View Schedule" page of LOGOS UI for (a) CP and (b) grouped activities.

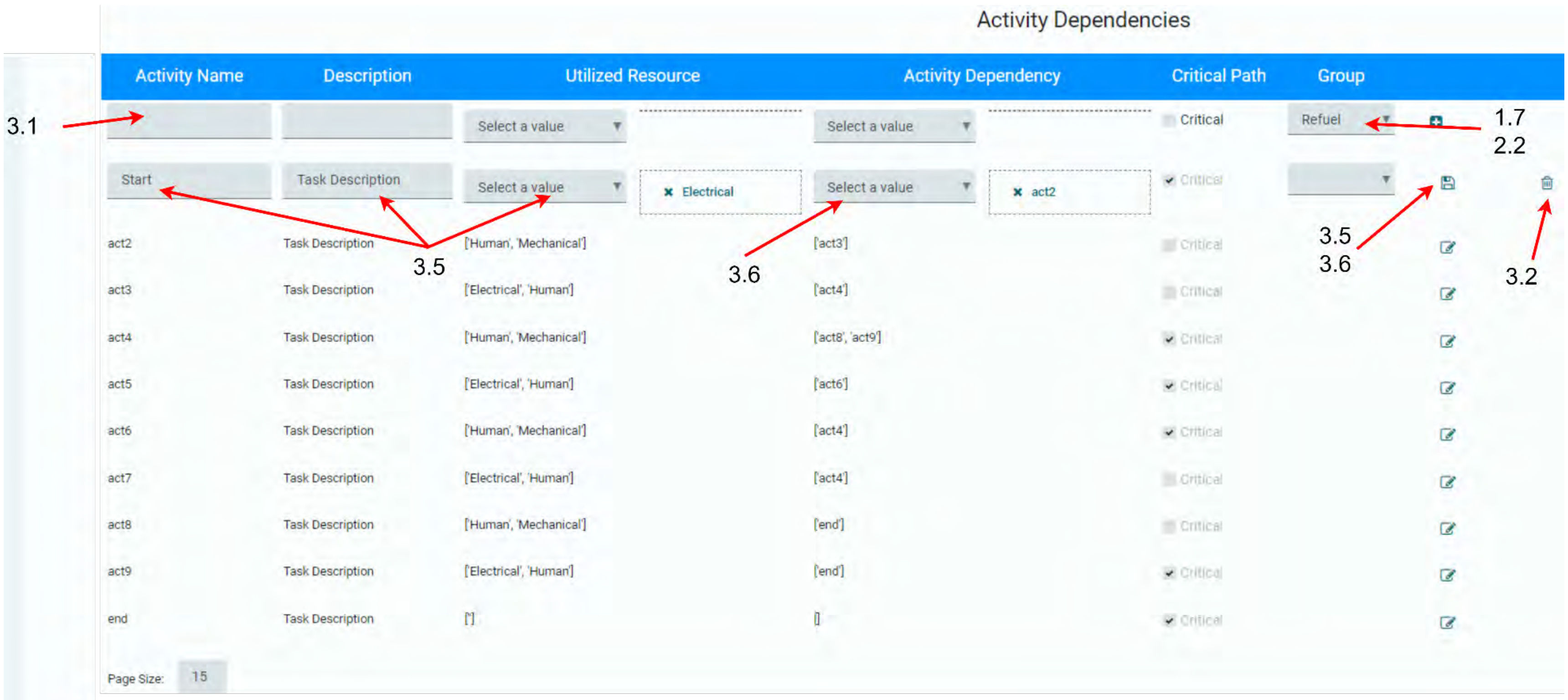


Figure 29. “Gantt Activities” page of the LOGOS UI.

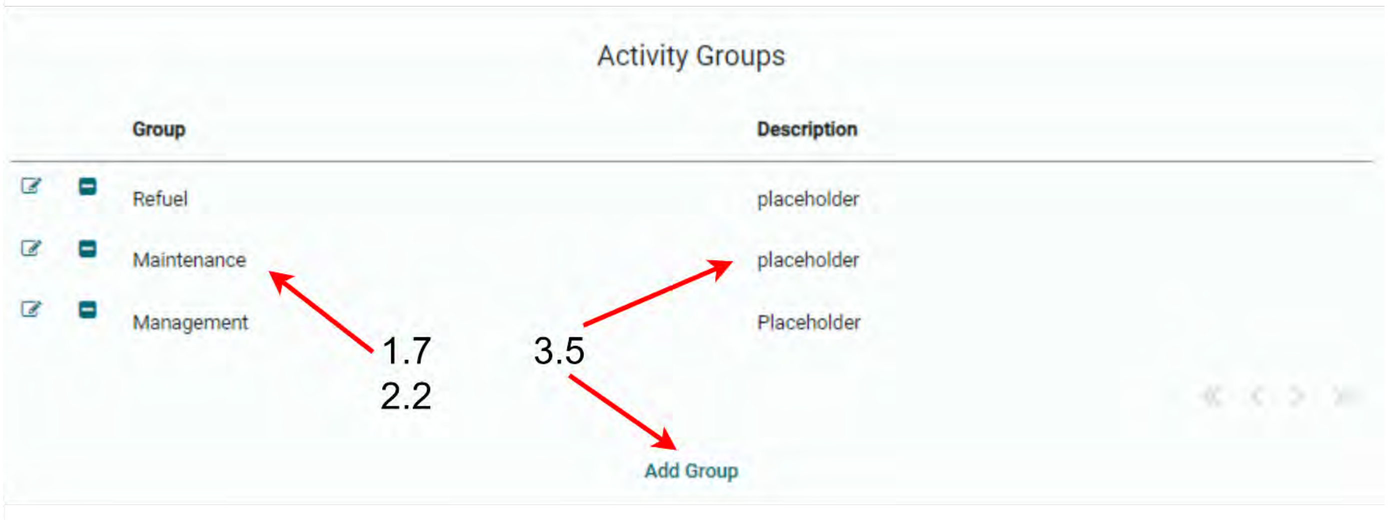


Figure 30. “Activity Group” page of the LOGOS UI.

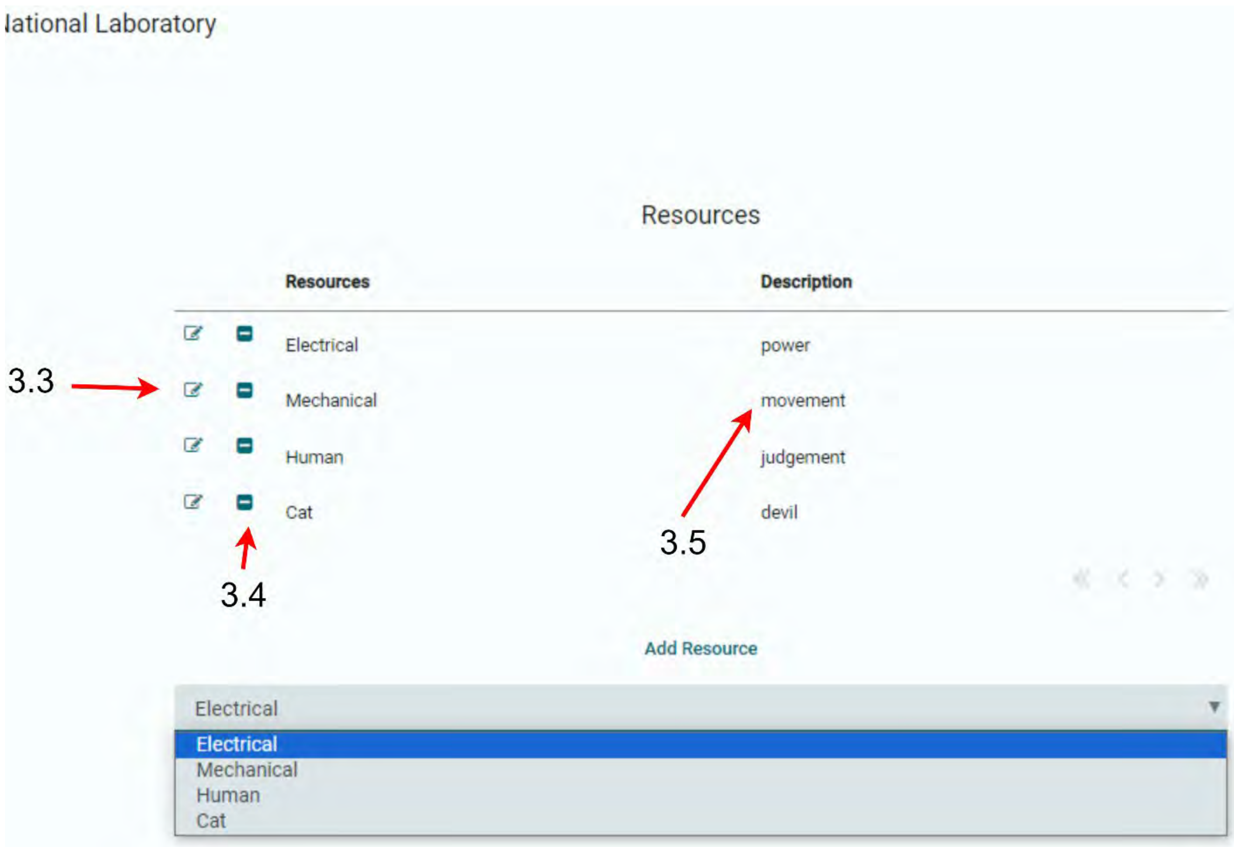


Figure 31 “Gantt Resources” page of the LOGOS UI.

8. TOOLS DEVELOPMENT

8.1 RAVEN

RAVEN is a flexible, multipurpose modeling and simulation platform for performing uncertainty quantifications, regression analyses, data analyses, and model optimization analyses (see Figure 32). Depending on the tasks to be accomplished and the probabilistic characterization of the problem, RAVEN perturbs a system response by altering its parameters via Monte Carlo, Latin hypercube, and other reliability surface search sampling methods.

The data generated via the sampling process are analyzed using classical and more advanced data mining approaches. RAVEN also manages parallel dispatching (i.e., on desktop, workstation, and large high-performance computing machines) of the software representing the physical model. It heavily relies on artificial intelligence algorithms to construct surrogate models of complex physical systems in order to perform uncertainty quantifications, reliability analyses (e.g., limit state surface), and parametric studies.

Figure 32. Graphical representation of RAVEN's computational capabilities.

Table 12 summarizes the subset of methods and models contained in RAVEN that are being used within this project. More specifically, RAVEN is actively used to:

- Link together models (e.g., ones contained in LOGOS) by using, for example, the ensemble and logical model
- Sample the constructed models (e.g., for data uncertainty propagation and model optimization) by using RAVEN samplers, optimizers (e.g., genetic algorithms), and the "RAVEN running RAVEN" capability (e.g., a mix of uncertainty propagation and model optimization)
- Postprocess the generated data via classical statistical and data mining methods.

Table 12. List of RAVEN models/methods and their corresponding use cases.

Model/Method/Capability	Use Case
Samplers (e.g., Monte Carlo, Latin hypercube, grid)	Methods designed to perform uncertainty quantifications by sampling model input variables according to a specific strategy
Pareto frontier	Method designed to determine the Pareto frontier from a multidimensional dataset
Optimizers (e.g., gradient-based, genetic algorithms, simulated annealing)	Methods designed to determine the global minima and maxima of a model's output variable by smartly changing input variable values
Ensemble model	Capability to link models together in a linear data workflow
Logical model	Capability to choose which of two models to run, based on a set of logical conditions
Basic statistics	Postprocessor designed to analyze large datasets, the goal being to determine the statistical moments (e.g., mean and variance) and perform sensitivity analysis
Data mining	Postprocessor designed to perform clustering of large datasets in order to evaluate similarities and patterns from the generated data

8.2 LOGOS

LOGOS (https://github.com/idaholab/LOGOS) contains a set of discrete optimization models employable for plant resource optimization problems. LOGOS integrates economic and reliability risks into a single analysis framework. More specifically, when provided structure, system, and component health information (e.g., failure rates or failure probabilities), operation and maintenance costs, replacement costs, other costs associated with component failure, and plant budget constraints, LOGOS determines the optimal set of projects (e.g., structure, system, or component replacement/refurbishment) that maximizes profit and satisfies the provided requirements/constraints.

The input data listed above can be either deterministic or stochastic in nature, meaning they can be point value forecasts or probability distribution functions. In the latter case, several scenarios are generated by sampling the provided distributions. The developed models are based on enhanced versions of the knapsack optimization problem. The two primary application classes are capital budgeting optimization and task schedule optimization. The goal of the capital budgeting class is to determine the optimal set of projects (i.e., project prioritization) and related schedules that maximize the overall profit. The goal of the task scheduler is to determine the optimal project schedule that minimizes the overall completion time.

The developed plant models can be either deterministic or stochastic in nature. Deterministic models are simpler, as they require point value data associated with constraints such as project costs and rewards. Stochastic optimization models extend deterministic models by explicitly considering data uncertainties. These formulations can be employed as standalone models or interfaced with the Idaho-National-Laboratory-developed RAVEN code to propagate data uncertainties and analyze generated data (i.e., sensitivity analysis and uncertainty quantification).

Table 13 provides a more detailed overview of the actual models/methods contained in LOGOS. The first two entries listed in Table 13 are designed for use in a model-based optimization setting in which the user specifies the architecture of the overall system model, which might include economic, reliability, and LOGOS sub-models (e.g., a project scheduling model). The user would then link models together via the RAVEN ensemble model capability and perform model optimization on the integrated model by using the optimization engines (e.g., genetic algorithms) available in RAVEN.

The third entry in Table 13 is mainly designed for use in a standalone configuration to perform data-based optimization-type analyses. If desired, these models and methods can be linked to RAVEN to propagate data uncertainties.

Table 13. List of LOGOS models/methods and their corresponding use cases.

Model/Method Class	Use Case
CP model	CP calculations and sub-path analysis provided an outage schedule
Scheduling model	Data-based task scheduling and resource optimization
Simple, multiple knapsack optimization model	Data-based project prioritization, selection, and scheduling using PYOMO (which includes its own distributionally robust and value-at-risk formalism)

8.3 DACKAR

Digital Analytics, Causal Knowledge Acquisition and Reasoning (https://github.com/idaholab/DACKAR) is a library of software tools for analyzing equipment reliability data, both numeric (e.g., monitoring data) and textual (e.g., IRs and WOs)In addition, these methods focus on integrating all these data elements so as to assist system engineers in analyzing component, asset, and system historic performances and optimizing maintenance activities. This integration is performed by extracting knowledge from textual data via technical language processing methods and quantifying system, asset, and component health from numeric, condition-based data. We rely on model-based system engineering (MBSE) models of systems and assets to identify their architecture and functional (i.e., cause and effect) relations. Numeric and textual data elements are then associated with an MBSE graph element, based on their nature. This bonding of MBSE models and data elements constitutes a first-of-its-kind knowledge graph of an NPP system, with data elements being organized in a structured manner that enables system engineers to identify cause-effect trends in data elements and to carry out appropriate actions in response.

During FY-23 and FY-24, this project relied on the DACKAR NLP methods presented in Section 6.1 to assess activity duration variance (performed using the knowledge-based approach). The NLP knowledge extraction methods were initially developed to extract the nature of the unexpected events, and more development and testing will be performed in FY-25. In the near future, we envision that the developed knowledge graph will contain not only WOs and IRs, but also outage data (e.g., activities performed during multiple outages, and unexpected events that emerged during past outages).

9. OUTAGE SCHEDULE ANALYSIS TOOL

The Outage Schedule Analysis Tool is ultimately intended to improve outage efficiency, reduce costs, and enhance schedule resilience for NPP operators. It will combine advanced analytical techniques with practical operational support so as to address the complex challenges of outage management.

A proof-of-concept application for this tool is currently in the beginning stages of development (see Figure 33). A basic UI was developed to serve as the foundation for our tool. This interface is designed to be intuitive and user friendly, enabling users to easily interact with the various features of the tool. Though still in its early stages, our proof-of-concept application demonstrates potential for outage schedule importation, basic schedule visualization, and initial analysis of activity durations. As development continues, more advanced features will be added.

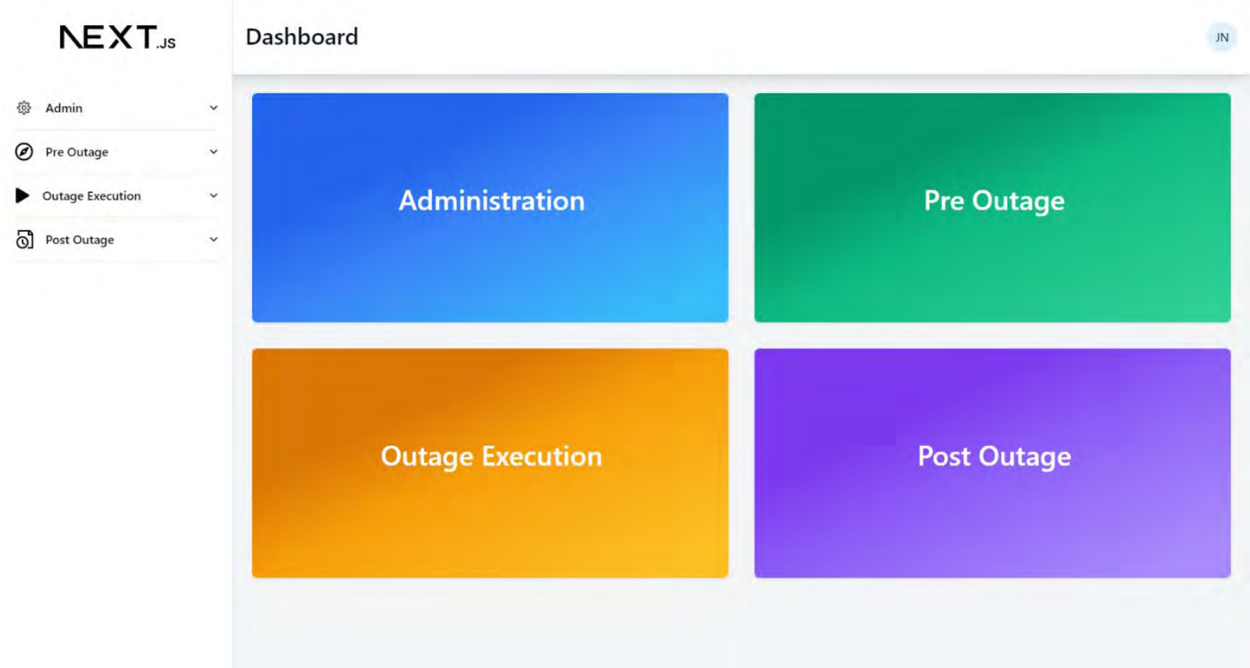


Figure 33. Plant Outage Optimization Tool, Next.js UI design.

Planned key features and capabilities include:

1. Schedule Analysis:
   - Analyzes planned outage schedules to assess activity duration variances
   - Evaluates schedule resilience and identifies CPs
   - Provides visualizations of the outage schedule
2. Optimization:
   - Helps optimize resource allocation and activity scheduling
   - Assists in handling emergent (unplanned) activities during outages
   - Aims to minimize outage durations while completing all necessary tasks
3. ML Integration:
   - Uses NLP and ML models to predict activity durations and analyze historical data
4. Risk Assessment:
   - Identifies high-risk activities and low-resilience paths in the schedule
   - Calculates importance measures for different activities
5. Data Integration:
   - Interfaces with existing plant systems such as P6 scheduling software and WO databases
   - Imports and processes historical outage data

6. Real-Time Execution Support:
    - Provides tools for monitoring outage progress
    - Offers capabilities for rescheduling and handling emergent work during the outage
7. Post-Outage Analysis:
    - Allows for comparing outage performance across different plants or outages

As development progresses, feedback and insights from potential users will be crucial to refine and enhance the functionality of this application. This project represents a significant advancement in NPP outage management. By combining cutting-edge technology with deep industry knowledge, we aim to deliver a tool that will substantially improve the safety, efficiency, and cost effectiveness of NPP operations.

10. CONCLUSIONS

This report presented a set of computational methods designed to assist NPP outage schedulers in analyzing planned outage schedules, the goal being to identify critical elements that might increase the risk of delay. Our work focused explicitly on the resources involved in a plant outage (i.e., time and crews). It measured schedule resilience based on how these resources are used throughout the outage, and it quantified the risk associated with schedule delays by quantifying activity duration variance via NLP methods and by propagating the resulting uncertainty throughout the schedule.

Regarding the assessment of activity duration variance, a few methods based on state-of-the-art NLP approaches were here developed and tested on a few plant datasets, with promising results.

Rather than utilize available CP-based tools, we moved away from an activity-based mindset and toward a sub-path-based one. Such an approach not only simplifies the outage resilience analysis (because in an outage, the number of sub-paths is considerably lower than the number of activities), it also provides more complete information on the impact of delays associated with activities that belong to the same sub-path, thus affording precise information on how available crews can be allocated to address such delays.

REFERENCES

Bishop, C. M. 2006. “Pattern recognition and machine learning.” New York: Springer.

Goodfellow, I., Y. Bengio, and A. Courville. 2016. “Deep learning.” MIT press.

IAEA - International Atomic Energy Agency. 2022. “Nuclear power plant outage optimization strategy.” IAEA Technical Report IAEA-TECDOC-1315.

Li, Y., et al. 2006. “Sentence similarity based on semantic nets and corpus statistics.” *IEEE Transactions on Knowledge and Data Engineering* 18(8): 1138–1150. http://dx.doi.org/10.1109/TKDE.2006.130.

Li, Y., Z. A. Bandar, and D. McLean. 2003. “An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources.” *IEEE Transactions on Knowledge and Data Engineering* 15(4): 871–882. http://dx.doi.org/10.1109/TKDE.2003.1209005.

Mandelli, D., A. Al Rashdan, S. St. Germain, S. Lawrence, N. Mapes, C. Wang, and J. Cogliati. 2023. “Tools and Methods for Optimization of Nuclear Plant Outages.” INL Technical Report INL/RPT-23-74439.

McDonald, G. C. 2009. “Ridge regression.” Wiley Interdisciplinary Reviews: Computational Statistics 1.1: 93-100.

Jeong, M., J. S. You. 2022. “Estimating the economic costs of nuclear power plant outages in a regulated market using a latent factor model.” *Renewable and Sustainable Energy Reviews* 166:112582. https://doi.org/10.1016/j.rser.2022.112582.

Navigli, R., and F. Martelli. 2019. “An overview of word and sense similarity.” *Natural Language Engineering* 25(6): 693–714. http://dx.doi.org/10.1017/S1351324919000305.

Ramos, J. 2003. “Using tf-idf to determine word relevance in document queries.” In proceedings of the First Instructional Conference on Machine Learning 242(1): 29–48.

St. Germain, S., R. Farris, H. Medema. 2013. “Development of Methodologies for Technology Deployment for Advanced Outage Control Centers that Improve Outage Coordination, Problem Resolution and Outage Risk Management.” INL Technical Report INL/EXT-13-29934.

St. Germain, S. 2015. “Use of Collaborative Software to Improve Nuclear Power Plant Outage Management.” Proceedings of 9th International Conference on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies.

Tian, Y., and Y. Zhang. 2022. “A comprehensive survey on regularization strategies in machine learning.” *Information Fusion* 80: 146–166.