

A Dynamic Multi-Stage Design Framework for Staged Deployment Optimization of Highly Stochastic Systems

Bayan Hamdan¹, Zheng Liu¹, Koki Ho², İ. Esra Büyüктаhtakın³, and Pingfeng Wang^{1, *}

¹Department of Industrial and Systems Engineering, University of Illinois at Urbana-Champaign

²Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology

³Grado Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University

*Corresponding author. Email: pingfeng@illinois.edu

Abstract

The need for staged design optimization for multidisciplinary systems with strong, cross-system links and complex systems has been acknowledged in various contexts. This is prominent in fields where decisions between subsystems are dependant, as well as in cases where tactical decisions need to be made in uncertain environments. The flexibility gained by incorporating evolutionary design options has been analyzed by discretizing the time-variant uncertainties into scenarios and considering the flexible decision variables in each scenario separately. However, these problems use existing information at the decision time step. This paper presents a Dynamic Multi-Staged Design (DMSD) framework to solve problems that dynamically incorporate updated system information and reformulate the problem to account for the updated parameters. The importance of considering staged decisions is studied, and the benefit of the model is evaluated in cases where the stochasticity of the parameters decreases with time. The impact of considering staged deployment for highly stochastic, large-scale systems is investigated through a numerical case study as well as a case study for the IEEE-30 bus system. The case studies presented in this paper investigate multi-disciplinary design problems for large-scale complex systems as well as operational planning for highly stochastic systems. The importance of considering staged deployment for multi-disciplinary systems that have decreasing variability of their parameters with time is highlighted and demonstrated through the results of a numerical and realistic case studies.

Keywords. Staged decisions, multi-stage, dynamic information, stochastic mixed-integer programming model.

1 Introduction

With the increasing complexity and scale of complex, engineered systems, there is a need for a robust framework for adapting to extraneous factors that cannot be accounted for in the planning stage. Currently, most systems anticipate the environmental conditions through stochastic methods that can predict and account for the uncertainty in the system. Then, a system can be designed with specifications that have a higher probability of meeting those conditions. However, a new avenue of research studies the optimal reaction to the realization of uncertain design parameters. Staged deployment is a means of establishing flexibility in the deployment of complex systems [1]. It refers to deploying a project gradually, where the initial deployment is conservative to avoid large initial costs [2]. This method allows policy and decision makers to increase their captured demand in stages depending on the actual realization. Staged deployment optimization is mostly popular in the aerospace industry [2, 3], however, studies are also common in the automotive industry [4], medical industry [5] and other fields.

The popularity of staged deployment optimization is due to it being an economical option for planning the development and enhancing the flexibility of the system as well as mitigating potential risks against unknown, future events. In addition to the long service life of such systems, these factors motivate the development of resilient and cost-effective complex engineered systems, especially those that consist of a growing number of interacting subsystems [6–8]. Given the dynamic nature of future operating and environmental changes on system performance over time, as well as the fast-paced nature of technology evolution, traditional methods for system design and deployment all at once could involve significant technical and financial risks [9]. Therefore, designing systems over time while considering the evolving nature of the future and subsequently deploying the system in progressive stages has been shown as an effective solution in mitigating the risks [10, 11]. For example, for the development of critical infrastructure systems for a developing region (e.g., hydroelectric power plants and water distribution systems), considering temporal demand variations and future community growth, the design and deployment cannot be done all at once. Instead, the system development with coupled subsystems that evolve together over time must be done through a series of design stages where the design and system deployments at previous stages constrain decisions at later stages. There is an imperative need for an effective complex system design framework and techniques that can take into account the design and staged deployment of interdependent subsystems while simultaneously considering time-variant uncertainty and future extreme events.

Long-term decision making is very difficult with stochastic problems. Not only are there random variables that need to be accounted for, but with non-stationary processes, accurate projections for changes in distributions can be very difficult to obtain and even impossible in most cases. Therefore, the integrity of long-term decisions cannot be guaranteed. In order to mitigate the

high uncertainty embedded in long-term decision making, staged deployment is used to make more frequent decisions for shorter planning horizons. Staged deployment allows more reliable decisions to be made; however, there is a higher cost associated with making more frequent decisions which also needs to be factored in. For example, the cost of building a factory with a capacity of 50,000 units will be less than the cost of building a factory with a 10,000-unit capacity and then expanding the capacity to 50,000 units. Additionally, the cost of changing the design variable might be higher per unit than for advanced decisions. Moreover, in some systems, the decision implementation lead time could lead to prolonged disturbances in system operations and, in some cases, lead to a complete stop of operations.

In the literature, increasing complexity and multi-disciplinary characteristics of complex systems design have become a challenge for engineers, and different tools have been proposed for optimization of decomposition using a design structure matrix (DSM) or its variants [12]. In the multi-disciplinary design optimization (MDO) context, effective system decomposition has led to efficient optimization of otherwise intractable large-scale design problems as covered in the following papers [13, 14]. Various partitioning techniques have been developed for optimal MDO problem decomposition [15–17], and different iterative and parallelizable multi-level optimization MDO coordination methods have been proposed to utilize the decomposed structure for an efficient system- and subsystem-level optimization. Examples of multi-level optimization include bi-level integrated system synthesis [18], collaborative optimization (CO) [19], and analytical target cascading (ATC) [20]. Computational experiments have been used to compare their performance [21, 22]. Uncertainties have been considered in MDO, leading to uncertainty-based MDO methods [23]. While considering the uncertainties in system design optimization, the reliability-based design optimization (RBDO) problem and further the reliability-based robust design optimization problem [24] have been investigated in the literature.

Different formulations have been studied using not only traditional stochastic optimization but also robust optimization and reliability-based optimization. The uncertainty propagation across the subsystems has often been handled using Monte Carlo simulation and Taylor series approximation, among many other methods [25]. For complex systems, where the number of uncertainty sources is more than manageable, screening methods have been used to narrow them down to a small number for computational reduction. Sensitivity analysis methods have also been used to evaluate the effects of uncertainties, or for design sequencing. However, existing uncertainty MDO methods are useful for identifying a single design that performs well with random variables or parameters, but their static nature makes them ineffective or inapplicable for identifying dynamic deployment plans in response to time-variant uncertainties. Also, the existing screening methods to narrow down the sources of uncertainties for computational efficiency will exclude sufficiently flexible deployment solutions that respond to a variety of uncertainties. There is a great need for strategic methods to co-optimize the dynamic decisions in response to the time-variant uncertainties while managing the computational complexity.

The needs of staged deployment, which share a similar concept with service-phase evolution, have been identified and studied extensively in different contexts [26]. In response to such needs, many studies have addressed modeling for future uncertainties. These studies improve the resilience

of systems by staging decisions over time to reduce the variability of the model. This variability is introduced modeling dynamic models over time and estimate long-term stochastic parameters [27]. These models typically discretize the uncertainty into a deterministic equivalent formulation and use the different possible scenario paths to influence staged decisions and first-stage decisions that can anticipate future trajectories of uncertain parameters. This approach does not only narrow the infinite number of possibilities down to a finite number of scenarios to analyze, but also aligns well with reality as deployment decisions are typically discrete. An example of such discretization is a binomial lattice model [28], a time-discrete representation of a geometric Brownian motion. For a general stochastic process besides a geometric Brown motion, there has been extensive literature in the economics and operations research fields for scenario discretization of stochastic processes as well as scenario reduction methods. Given the discretized scenarios, stochastic optimization with either direct formulation or decision-rule-based formulation can be used to find flexible designs. However, when applied to a large-scale system with numerous uncertainty sources, existing scenario discretization methods either require arbitrary decision rules or an explosive number of variables/constraints due to the scenario discretization methods. It is practically infeasible to identify a decision rule for a complex system.

To address this gap, this paper justifies the need for a novel stochastic optimization method based on time-dependent stochastic process models. This allows users to design stage-deployed large-scale complex systems with coupled and co-evolving subsystems while considering time-variant uncertainties and extreme events. In the case study, the importance of considering staged deployment for time-dependent stochastic processes in multi-disciplinary systems is highlighted. The rest of the paper is organized as follows: Section 2 provides an overview of the staged deployment design problem with respect to All-at-once solution strategies for solving the deterministic equivalent problem. Section 3 introduces a framework for solving multi-stage stochastic models using decomposition techniques and highlights the proposed approach. Section 4 presents a numerical example highlighting the need for a stage deployed optimization design for multi-disciplinary problems. Section 5 presents an IEEE-30 Bus System case study to validate the results on a real model. This study is concluded in Section 6.

2 Staged-Deployment Design Optimization Methodology

2.1 All-at-once Solution Strategy

Stochastic programming problems are an active research area due to their importance in accurately representing real-world systems to make close-to-optimal decisions. However, due to the complexity in solving stochastic problems, deterministic equivalent formulations exist to make decisions while accounting for the stochastic nature of systems. Such formulations allow decision makers to make decisions before the realization of stochastic events, given some prior knowledge on the behavior of the stochastic parameters. This prior information can be in the form of random distributions or projections, which allow decision makers to anticipate the behavior of the system

over time. Deterministic Equivalent formulations discretize the random distributions and represent them in the form of probabilistic scenarios. Although this approach accounts for the uncertain nature of stochastic variables, if the prior information is not accurate, or if it changes with time, then these models are unable to make reliable decisions. This section presents some of the most prominent all-at-once solution strategies for stochastic modeling.

2.1.1 Two-Stage Design

A common approach to optimizing stochastic problems involves utilizing Two-Stage Stochastic Programming to mitigate the impact of making decisions with highly stochastic parameters. Two-Stage Stochastic Programming utilizes existing data at the time of modeling to obtain a flexible decision plan based on discretized scenario approximations. Two-Stage Stochastic Programming is typically employed when some decisions need to be made before the realization of uncertain events, and other decisions, such as the recourse decisions, are made after their realization [28]. The scenario-based formulation of a two-stage stochastic program can be cast into a deterministic equivalent model to make it more amenable to numerical optimization [29]. The objective function of the deterministic equivalent model is shown in Eqn. (1).

$$\min_x \mathbb{E}(f(x, \omega)) = \min_x c^T x + \mathbb{E}(Q(x, \xi^\omega)) \quad (1)$$

Where x represents the vector of decision variables, $f(x, \omega)$ represents the scenario specific objective value for each scenario $\omega \in \Omega$ and the Two-Stage Stochastic Model aims to minimize the expectation of the objective function. c represents the cost associated with the decisions taken, and ξ^ω represents the set of stochastic parameters dependant on ω . The term represents the deterministic term that is based on the cost of the first stage decisions. Lastly, the $\mathbb{E}(Q(x, \xi^\omega))$ term represents the expectation of the cost of the second stage decisions in the model [30]. Equivalently, Eqn. (1) can be converted to the deterministic equivalent formulation by decomposing the stochastic realizations into a finite number of scenarios Ω that can accurately represent the initial probability space [29]. Given this, Eqn. (1) can be rewritten as in Eqn. (2).

$$\min_x \mathbb{E}(f(x, \omega)) = \min_x c^T x + \sum_{\omega} p_{\omega} Q(x, \xi^\omega) \quad (2)$$

In Eqn. (2), p_{ω} represents the probability of scenario ω occurring. In order to solve Eqn. (2), scenario discretization can be utilized. Scenario discretization creates duplicate of the decision variables for each scenario ω . This is shown in Eqn. (3a) below. Since the deterministic equivalent formulation introduces copies of the decision variables, non-anticipativity constraints need to be added (Eqn. 3b) in order to ensure that even after the value of the stochastic variable is known, the chosen value for the decision variable does not change [29]. Non-anticipativity constraints represent a set of equality constraints that prohibit future information (information from future stages) to be incorporated in current decisions by fixing the decision variable values at certain time periods. In other words, these constraints limit the decisions that have the same history to be equivalent.

$$\text{Minimize}_x \quad \mathbb{E}(f(x, \omega)) = \min_x (c^T x + \sum_{\omega} p_{\omega} Q(x^{\omega}, \xi^{\omega})) \quad (3a)$$

$$\text{s.t.} \quad \sum_{\omega \in \Omega} A_{\omega} x^{\omega} = h \quad (3b)$$

$$x \in \mathbb{R} \quad (3c)$$

Given the presented formulation, the non-anticipativity constraint can be relaxed using a Lagrangian penalty parameter, represented by λ . The following single-objective problem can be solved in Eqn. (4).

$$\text{Minimize}_{x, \lambda} \quad c^T x + \sum_{\omega} \{p_{\omega} Q(x^{\omega}, \xi^{\omega}) + \lambda^T A_{\omega} x^{\omega}\} - \lambda^T h \quad (4)$$

While Two-Stage Stochastic Modelling allows the user to account for possible future realizations of parameters in the model and adjusts the response of the second stage variables based on the realization of the parameters, it does not incorporate new information as it arrives in order to adjust the response of the model [31]. Two-Stage Stochastic Modeling works well when there is an accurate probability distribution of the unknown parameters; however, in reality, it is very difficult to accurately predict the probability distribution of long-term parameters. This introduces the need for dynamic methods that can incorporate new information to accurately update the model response as the updated predictions of the parameters arrive.

2.1.2 Multi-Stage Design

The presented two-stage stochastic formulation can be extended to a multi-stage design. The multi-stage design considers several decision periods for random variables. The main difference between the two-stage stochastic model, is that the value of the decision variables can change at predetermined decision periods. The updated formulation is presented in Eqn. (5) below. In this case, the stages at which decisions are made are defined as t . In this case, t can be equivalent to the time periods, or can be a subset of the time periods involved. Only the decision variable values at time $t = 0$ (first-stage decisions) are fixed irregardless of the realizations of the stochastic parameters. Fig. 1 shows the main difference when considering multiple stages in terms of the parameters in the model. In the figure, three time periods are considered, with two periods that contain stochastic realizations of parameters. A single path, from root to tail, denotes a possible scenario occurring over the full time period. The probability of a scenario path is then obtained by multiplying the probability of the second stage realization with the third stage realization, as shown in the box on the left of Fig. 1. It shows that the probabilities are not aggregated for each scenario path for cases where the decisions are made for multiple time periods.

Given this updated formulation, the non-anticipativity constraints are updated to account for the pairing between stages, therefore, A_{ω_t} is updated to a matrix to account for the relationship

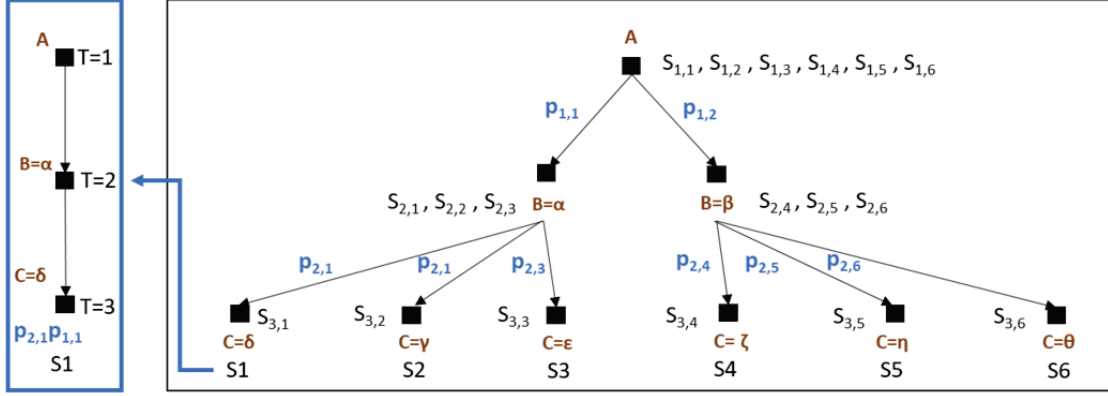


Figure 1: Example of scenario discretization with 3 decision points and possible evolution paths (i.e., scenarios)

between the stages. It ensures that the variables are decided based only on knowledge known up to that time, so that future knowledge (in subsequent stages) is not incorporated in earlier stages. Fig. 2 illustrates how non-anticipativity constraints are formulated for the same simple example presented in Fig. 1. The cost vector is defined by c_1 for the first time period and c_{t-1}^T for the rest of the time periods. Similarly, x_1 and x_{t-1}^T are divided into the first period decision variables and the decision variables in the subsequent time periods. This set of constraints updates the model so that λ is now a vector corresponding to each equation.

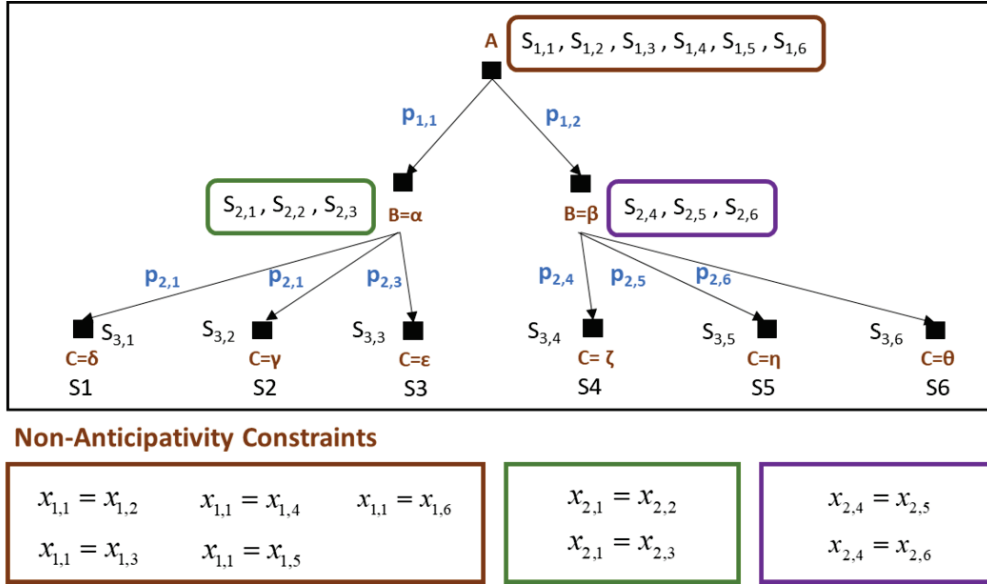


Figure 2: Non-anticipativity constraints for the stochastic decision variables are formulated based on the scenario tree.

$$\underset{x, \lambda}{\text{Minimize}} \quad c_1 x_1 + \sum_t \sum_{\omega_t} \{p_{\omega_t} c_{t-\{1\}}^T x_{t-\{1\}}^{\omega_t} + p_{\omega_t} Q(x^{\omega_t}, \xi^{\omega_t}) + \lambda^T A_{\omega_t} x_t^{\omega_t}\} - \sum_t \lambda^T h_t \quad (5)$$

Although this model can incorporate more flexibility by increasing the decision time steps, this approach is still reliant on prior information on the behavior of the stochastic parameters. Additionally, the number of decision variables increases drastically with the number of decision time steps. This is a result of changing the decision variable values more frequently, which means that additional indices are added to the decision variables and the size of the problem increases. This leads to a high computational cost to implement such models and can be prohibitive for realistic case studies where the scenario tree is large in size.

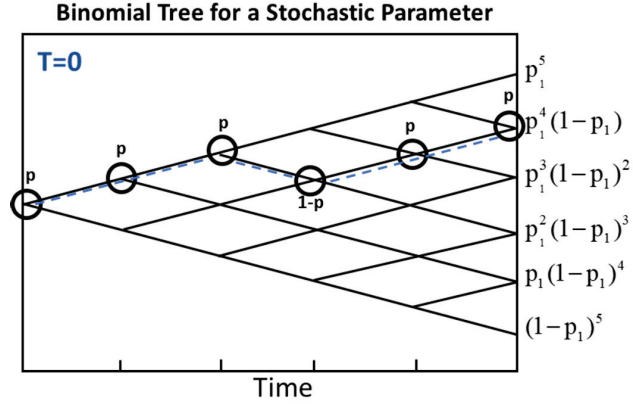
3 Dynamic Multi-Stage Design (DMSD)

3.1 The DMSD Formulation

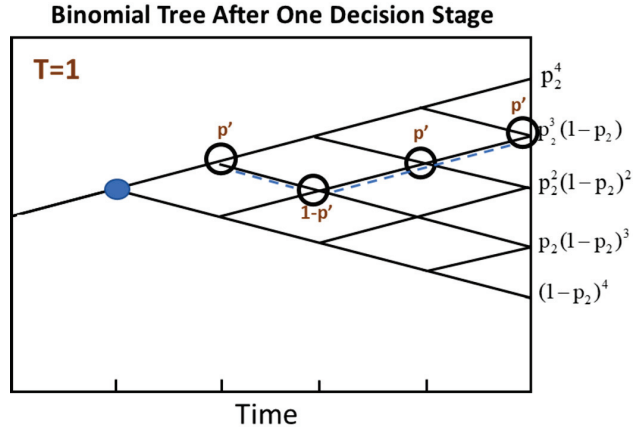
This section presents an overview of the proposed staged-deployment design problem that incorporates new information to update the model design. In the traditional methods that convert stochastic problems into a deterministic equivalent formulation, discretizing the scenarios for each subsystem and analyzing all combinations of scenarios for all decisions, as shown in Fig. 3, would be required in the staged-deployment optimization, resulting in a large number of variables/constraints. The size of the problem increases drastically when considering the staged uncertainty, which can be demotivating to decision makers. However, by dynamically updating the scenarios to consider more up-to-date probabilities, the model is able to consider smaller decision periods with results that better match the dynamic nature of the systems considered in the study. After each stage, the model is resolved with updated scenarios (both future and past realizations) to ensure that the system adapts to past realizations and anticipates the future variable realizations.

Multi-staged modeling takes into account the realization of events up to the time period that the decision maker is in, and after updating the state that the system is now in, solves a staged, stochastic model to determine if the previously determined values of the variables need to be modified given new predictions on the parameters in the model. Since the values of the scenario sets, $\xi_t, \dots, \xi_T$, can change in each time period, updating them over stages allows the feasibility sets, $\mathcal{U}_t, \dots, \mathcal{U}_T$, for each time period to change as well. In some cases, the updated feasibility sets can also be empty based on the previous realizations of the scenarios. However, in this study, only cases where the feasibility set is not empty are considered.

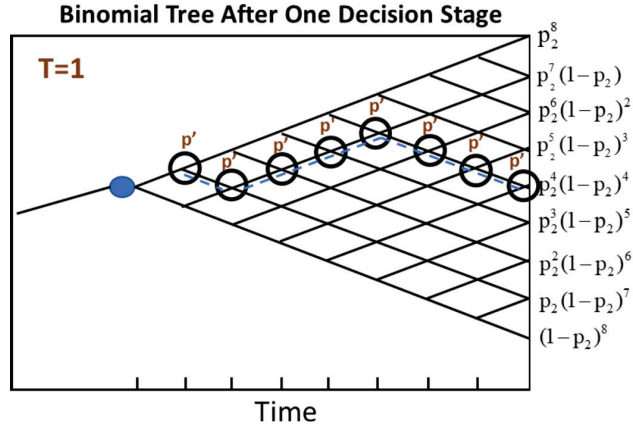
Fig. 3 shows the updated scenario tree for dynamic problems. In Fig. 3a, the complete scenario tree is shown at time step $t=0$ since no additional information is known at that decision time step. However, at the next decision time step, the scenario tree is updated according to new information. Fig. 3b shows a possible case where only the probability of the parameter realizations change. This change is due to the fact that after a subsequence of random variables, $\xi_1, \dots, \xi_{t-1}$, is known, the remaining sequence of random variables, $\xi_t, \dots, \xi_T$, is conditioned on the values of



(a) Scenario tree with the scenario information available at decision time step $t=0$ (adapted from [2]).



(b) The case where the expected value of p changes at $T=1$.



(c) The case in which the time discretization is updated given the new information as well as the value of p at $T=1$.

Figure 3: Possible updated scenario schemes at the next decision step

$\xi_1, \dots, \xi_{t-1}$. Alternatively, Fig. 3c shows the case where the number of scenarios is increased at that decision time step as well as the probability distribution. Intuitively, it is clear that considering a dynamic modeling scheme allows the most up-to-date information to be considered and can therefore improve the model accuracy. Selecting the number of time periods in each stage is one of the most important aspects of solving a dynamic, multi-stage model. While selecting a shorter decision time period allows for a higher result accuracy, intuitively, it is clear that an inherent tradeoff exists between improving the solution accuracy and increasing the cost incurred by the system when updating/modifying the value of the variables [32]. Moreover, the addition of decision time periods increases the size of the problem and can cause the model to become intractable.

The formulation for DMSD problem is updated for each stage. The original problem that is solved is identical to the multi-stage formulation presented in Eqn. (5). However, as the information prediction is updated, so are the scenario realizations. The formulation presented in Eqn. (6a) is the general formulation of the problem at stage t , where any decisions made prior to stage t are fixed decisions, and the scenario discretizations are updated based on the information available at t . Here, x_a represents the first stage decision variables and x_b represents the staged decisions. Then, the deterministic equivalent formulation is determined for the remaining staged time periods. In Eqn. (6b), the constraint represents the non-anticipativity constraints.

$$\text{Minimize}_{x} \quad \sum_{a=0}^{t-1} \sum_{\omega_t} c^T x_a + \sum_{b=t}^T \sum_{\omega_t} p_{\omega_t} Q(x_b^{\omega_t}, \xi_b^{\omega_t}) \quad (6a)$$

$$\text{s.t.} \quad \sum_{\omega \in \Omega} A_{\omega_t} x_b^{\omega_t} = h \quad (6b)$$

$$x \in \mathbb{R}$$

The non-anticipativity constraints can be relaxed by introducing a penalty term, λ , and adding the constraint violation term in the objective function such as in Eqn. (7).

$$\text{Minimize}_{x, \lambda} \quad \sum_{a=0}^{t-1} \sum_{\omega_t} c^T x_a + \sum_{b=t}^T \sum_{\omega_t} \{p_{\omega_t} Q(x_b^{\omega_t}, \xi_b^{\omega_t}) + \lambda^T A_{\omega_t} x_b^{\omega_t}\} - \lambda^T h \quad (7)$$

The main contributions of this paper consist of:

- providing a framework for solving stochastic multi-stage optimization problems for different scenario decomposition techniques and solution methods.
- dynamically incorporating new predictions of the stochastic parameters and system state and updating the multi-stage model based on their realizations.
- providing a structured way to combine scenario decomposition and solve the resulting deterministic model based on the ensuing decision tree. The decision tree is updated continuously and when it is updated, the dynamic multi-stage formulation is updated and solved.
- Implement the proposed methodology to a realistic, stochastic case study of the IEEE 30-bus system with and without the additional consideration of wind turbine power generation.

3.2 Problem Description

This section studies the benefit of implementing multi-stage modeling for several applications of the DMSD algorithm. In this section, staged capacity deployment problems and staged tactical planning, examples of problems to be solved using the DMSD framework, are studied. Some general formulations of target problems are described in Section 3.2.

3.2.1 Staged Tactical Planning

A large class of problems consist of making decisions that are both operational and tactical. Tactical decisions are usually made by middle management at regular intervals during the work year. Operational decisions, on the other hand, are made regularly during the work day and usually involve lower management or shop floor employees [33]. Therefore, since tactical decisions are made at a higher level, they can affect day to day operations and involve higher risk than operational decisions. Thus, they are made at regular intervals to ensure that the decisions are optimal based on the current environment. These problems are typically solved using multi-stage modeling, where the tactical decisions are made at the decision stages and the operational decisions are made in each time period [34]. Let us define a general Staged Tactical Planning problem as one with a cost reducing objective as seen in 8a. Where dt_τ^s refers to tactical decisions that are made at $\tau = 0, \dots, T_1$ decision stages under scenario $s = 1, \dots, S$. While operational decisions are made by do_t^s at time period tau under s . The tactical and operational decisions are then linked through a function depending on the problem domain as seen in 8b. Then, operational constraints are defined in 8c and tactical constraints are defined in 8d.

$$\min_{dt_\tau^s, do_t^s} = \sum_s p^s \left(\sum_\tau C_{tactical} dt_\tau^s + \sum_t C_{operational} do_t^s \right) \quad (8a)$$

$$f(dt_\tau^s, do_t^s) \leq 0 \quad \forall \tau, t, s \quad (8b)$$

$$g(do_t^s) \leq 0 \quad \forall t, s \quad (8c)$$

$$h(dt_\tau^s) \leq 0 \quad \forall \tau, s \quad (8d)$$

$$dt_\tau \geq 0 \quad \forall \tau$$

$$do_t \geq 0 \quad \forall t$$

Such problems are common to many industries and could be used to represent most commercial operations. The distinguishing feature for these models is that some decisions are made more frequently than staged decisions (in this case, the tactical decisions). The operational decisions are then updated based on the value of the tactical decisions since they impact the structure of the system.

3.2.2 Staged Capacity Deployment Problems

Traditionally, the deterministic formulation for expanding and rebuilding a single infrastructure system will have the form shown in Eqn. (9a). The variable x_t indicates the operating infrastructure size at a time step t , y_t indicates the infrastructure size that retires at the end of time t , and z_t indicates the infrastructure size that is added at the beginning of time t . The objective function includes the operational cost, $C_{operational}$, retire cost, C_{retire} , and expansion cost, C_{expand} . The constraints include the “mass balance” of the operational mass and a requirement on the interruption of the operation. The latter can be written that over the time horizon, the unsatisfied demand $\max(Demand(t) - Performance(x_t, t), 0)$ is within certain required demand shortage amount DS_{req} . Note that $Performance(x_t, t)$ is not only a function of the current operating infrastructure size, but also time, to take into account the performance degradation over time of aging infrastructure.

$$\text{Minimize} \quad \sum_t C_{operational}x_t + C_{retire}y_t + C_{expand}z_t \quad (9a)$$

$$\text{s.t.} \quad x_t = x_{t-1} - y_{t-1} + z_t \quad \forall t \quad (9b)$$

$$\sum_t \max(Demand(t) - Performance(x_t, t), 0) \leq DS_{req} \quad (9c)$$

The deterministic formulation above can be adapted to a multi-stage formulation. The updated formulation considers staged decisions, d , where the demand realization for previous time periods can be set and the future demand predictions can be updated based on new data. Additionally, decisions can be introduced to determine whether the capacity of the system should be increased (either due to performance degradation or increasing the overall capacity to handle higher production requirements to meet the updated demand expectation).

In order to accurately evaluate the benefit of staging decisions, the associated costs must also be considered in the evaluation of the methods [35]. These costs include the decision-making costs as well as the additional fixed costs incurred with modifying system architectures. To correctly assess the financial costs of the decisions, the Net Present Value (NPV) of the costs is usually used. This allows all the decisions to be evaluated considering the total cost at the time of the decision, even for multiple stages. The NPV of an investment in a certain time period can be calculated using Eqn. (10), and this allows the investments made at different time periods to be juxtaposed.

$$NPV = \sum_{t=1}^n \frac{R_t}{(1+i)^t} \quad (10)$$

where R_t corresponds to the value of the investment at time t and i corresponds to the discount rate [36]. Therefore, utilizing the NPV allows the value of future flexibility to be determined at the decision period.

3.3 Solution Framework

Dynamic Multi-Stage Modeling introduces flexibility when solving stochastic problems, yet they tend to increase the computational resources required to solve the problem. Solving multi-stage problems can be divided into two parts: scenario generation and solving the resulting problem.

3.3.1 Scenario Generation

Although stochastic programming is a tool that allows decisions to be made given uncertainties in the parameters of a system, most stochastic programming methods can be intractable or need advanced solution methods to address the specific structure of the model. In order to utilize deterministic solution schemes for realistic problems, a deterministic equivalent formulation can be used. This allows decision makers to mitigate the costs associated with advanced stochastic programming tools and implement existing deterministic solution methods. Scenario generation refers to the implementation of a finite discretization of a random vector [37]. The main difficulty with scenario generation lies in maintaining a balance between selecting a large enough scenario set size as to be representative of the true distribution of the random variable, as well as maintaining a problem size that ensures computational tractability. The specific scenario generation scheme can be selected based on the application and the granularity of the uncertain variables as well as their approximated underlying distribution. Some of the popular scenario generation methods include Monte Carlo simulations based on the expected distribution of the stochastic parameters, path-based methods, moment matching and optimization discretization [38].

Tail-risk measures are used to ensure that the risk of extreme events is minimized and that the number of scenarios far from the risk region are reduced. They focus on the upper tail region of random distributions to identify critical regions [37]. Some popular tail risk measures include the Value-at-Risk (VaR) and Conditional Value at Risk (CVaR) measures. These measures focus on identifying the risk of potential loss depending on the problem being considered. VaR was initially introduced to measure the potential loss of an investment. It determines the worst-case loss given an uncertain event. CVaR, on the other hand, calculates the average loss provided that the VaR value is exceeded [39]. Scenario generation schemes that are built based on tail-risk measures prioritize generating scenarios that lie within risk regions close to the tail of the distribution. These methods lie within the area of problem-driven scenario generation. One appropriate sampling method is the asymptotic sampling algorithm in [37]. This sampling scheme is based on aggregation sampling, where a risk region, $\mathcal{R}$, is identified and a target risk region sample size, $N_{\mathcal{R}}$, is specified. The algorithm then samples points from the distribution and if they are not located in the risk region, $\mathcal{R}^c$, then they are aggregated into a single point, which in this case is the mean of the sample points. Then, when assigning the probabilities of the points, they are assigned as in Eqn. (11) for the samples in the risk region and Eqn. (12) for the average value of the sample points not in the risk region. Here, p_i represents the probability of sample point i , where $i \in \{1, N_{\mathcal{R}}\}$. As for the points within the safe region, $\mathcal{R}^c$, as shown in Eqn. (12), they are given a probability based on the number of samples drawn from within that region.

$$p_i \leftarrow \frac{1}{n_{\mathfrak{R}^c} + N_{\mathfrak{R}}} \quad (11)$$

$$p_{N_{\mathfrak{R}+1}} \leftarrow \frac{n_{\mathfrak{R}^c}}{n_{\mathfrak{R}^c} + N_{\mathfrak{R}}} \quad (12)$$

Another popular scenario generation method is moment matching. Moment matching is used to ensure that the statistics of the generated scenarios match the original distribution [40]. Although scenarios are generated using Monte Carlo simulation (or equivalent methods) from the original distribution, depending on the number of scenarios and the generated samples. Therefore, moment matching allows the statistics of the sample to be compared with the true estimates and aims to minimize the deviation between them. Several methods in the literature provide methods to generate moment matching scenarios [41]. The main approach is to minimize the deviation between the sample and the original parameters of the distribution. For a normal distribution, the squared residuals of the main statistics can be minimized as in Eqn. (13) [42]. The main statistics that are compared are the mean, μ , standard deviation σ , skewness, $skew$, kurtosis, kur and correlation, $corr$.

$$\min (\mu * -\mu_s)^2 + (\sigma * -\sigma_s)^2 + (skew * -skew_s)^2 + (kur * -kur_s)^2 + (corr * -corr_s)^2 \quad (13)$$

3.3.2 Problem Decomposition Schemes

In order to accommodate the higher computational cost associated with multi-stage modeling, certain solution methods could be utilized to reduce the computational burden. Decomposition methods are among the most prevalent solution methods associated with multi-stage stochastic modeling. Decomposition methods, similar to their name, refer to methods that break down the problem into smaller subproblems to aid in solving them faster. Common decomposition methods are Bender's Decomposition, Lagrangian Relaxation, and Multidisciplinary Design Optimization (MDO) Methods like ATC and Response Surface Methodology [43], [44]. Traditionally, since scenario decomposition techniques can be ill-posed if a smaller number of scenarios is evaluated in the model, and increasing the number of scenarios considered can substantially elevate the computational time, parallel solution schemes can be utilized to exploit available resources [45].

For solving static multi-stage models, or models that have several staged decisions but have a single input time for the probability distribution of the random parameters in the model, these methods have been applied in the literature to obtain computationally inexpensive results. When the data is included dynamically, however, the same methods cannot be applied in the same way since the model itself is solved in stages. Each stage itself must be decomposed depending on the structure of the problem itself. The structure of the problem can help determine which decomposition method should be used. For example, Lagrangian Relaxation is typically used when there is a

complicating constraint or set of constraints [46]. These methods can be used to decouple problems based on their inherent structure.

In this study, we propose a multi-level framework that has a system-level optimization in the first level and subproblem optimization in the proceeding levels. However when each problem is solved it is solved as a multi-stage problem. The pseudo code as detailed in Fig. 4 shows the structure of the solution framework. Unlike the static multi-stage formulation predominant in the literature, the dynamic formulation solves the problem τ times. The framework highlighted in Fig. 4 considers the case where the stochastic problem is solved using the deterministic equivalent formulation based on scenario discretization. However, the same framework can apply to other solution methods as well as scenario decomposition techniques. Solving the Staged Problem can be done through the use of commercial solvers depending on the specific problem formulation. The second level allows for the recalculation of the scenarios considered (a new scenario tree is constructed and can have different branches with varying probabilities), which can allow for computational savings since the variance is reduced.

4 Numerical Case Study

The first numerical example presented is a model based on the mass damper system and optimization model presented in [47]. The authors aim to maximize the energy storage capacity of the system by designing the spring thickness. Their model combines both the control and physical design aspects of the system in a co-design problem. Their model is dynamic through the introduction of differential equations to model the control part. However, their model considers a static, one-time decision of designing the spring thickness, when in some analogous systems, like a car suspension system, the decision that needs to be made is dynamic in nature [48]. This means that the value of the optimized variable can change at different points in time. Additionally, the model in [47] assumes that the control variable is known, when in reality, again, like in the case of the car suspension system, this parameter is stochastic. This introduces the need for a robust model that accounts for the inherent variability in the system. This is especially the case when the variability of the parameter decreases as the observation period approaches the time period for which the prediction of the input variable is made, such as in Fig. 5. Moreover, with time the expected distribution of the stochastic parameter approaches the true distribution. Therefore, updating the distribution of the stochastic parameter can drastically improve the response of the model.

Given the available distribution information, scenario discretization is performed as shown in Fig. 6. Given the most recent information at each decision stage, the distribution is updated and new scenarios are discretized. Since this allows the model to incorporate updated information with reduced variance, a smaller number of scenarios can be used to represent the stochastic nature of the parameter. This reduces the computational cost significantly.

This paper presents the numerical model in [47] updated so that the control input variable, u_i variable is stochastic and follows a similar behavior as that of Fig. 6. The original model developed

Algorithm 1: DMSD Algorithm

Data: Stationary Parameters
Stochastic Parameter Distributions (μ, σ)

Result: $(x_t, TotalExpectedCost) \leftarrow SP$

for $\tau \in T$ **do**

Scenario Generation:

Aggregate Sampling

Inputs: $N_{\mathfrak{R}}, \mathfrak{R}, \mathfrak{R}^c, \mu, \sigma$ *Outputs:* Scenario set ξ

$n_{\mathfrak{R}^c} \leftarrow 0, n_{\mathfrak{R}} \leftarrow 0, \xi_{\mathfrak{R}^c} = 0$

for $n_{\mathfrak{R}} \leq N_{\mathfrak{R}}$ **do**

if $\xi \in \mathfrak{R}$ **then**

$\xi_{N_{\mathfrak{R}}} \leftarrow \xi, n_{\mathfrak{R}} \leftarrow n_{\mathfrak{R}} + 1$

else

$\xi_{\mathfrak{R}^c} \leftarrow \frac{1}{n_{\mathfrak{R}^c} + 1} (n_{\mathfrak{R}^c} \xi_{\mathfrak{R}^c} + \xi)$

end

if $n_{\mathfrak{R}^c} = 0$ **then**

$\xi_{N_{\mathfrak{R}+1}} \leftarrow \xi_{\mathfrak{R}^c}$

else

Sample new point, ξ

$n_{\mathfrak{R}^c} \leftarrow 1, \xi_{N_{\mathfrak{R}+1}} \leftarrow \xi$

end

end

foreach $i \in \{1, \dots, N_{\mathfrak{R}}\}$ **do**

$p_i \leftarrow \frac{1}{n_{\mathfrak{R}^c} + N_{\mathfrak{R}}}$

end

$p_{N_{\mathfrak{R}+1}} \leftarrow \frac{n_{\mathfrak{R}^c}}{n_{\mathfrak{R}^c} + N_{\mathfrak{R}}}$

Staged Problem (SP):

$C_{\tau} \leftarrow C_{\tau-1}, \xi_{\tau} \leftarrow \xi_{\tau-1}$

Fix variables at $l \in [0, \tau - 1]$

Solve SP (ξ, C_{τ})

$(\mu, \sigma) \leftarrow \text{update}$

end

Figure 4: DMSD Solution Framework

in [47] is presented in Eqs. 14a to 14g. Four variations of the model are presented, and the main differences are highlighted in Table 1.

Nomenclature

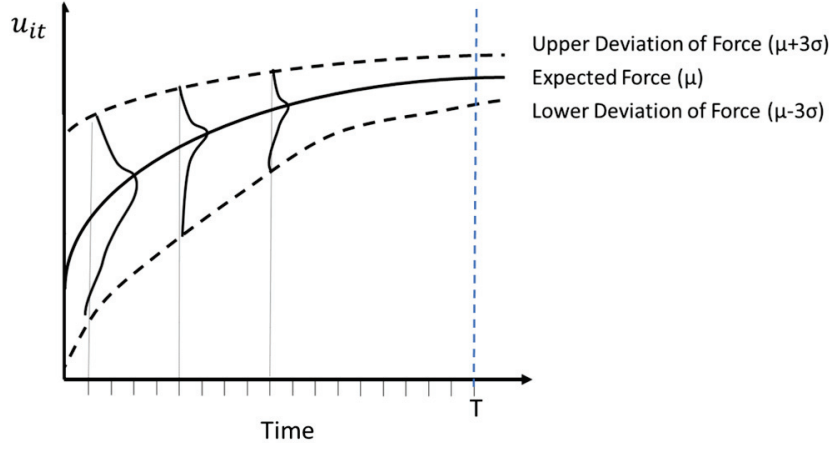


Figure 5: The figure shows the variability reduction and change in expected value over time when estimating the force at time T .

Indices

i Set of springs, indexed by $i \in \{1, 2, 3\}$.

s Set of scenarios, indexed by $s \in S$.

Parameters

κ_1 Minimum allowable inside diameter.

κ_2 lower bound on physical variables.

Π Probability of scenario occurrence.

A Amplitude of Oscillation.

C Spring index.

c Damping coefficients.

D_s Clearance constant.

F_u Maximum allowable force.

G Shear modulus.

k_i Spring constant.

m_i Mass for each spring subsystem i .

Q_i Diagonal weighting matrix associated with the velocity and position terms in the objective.

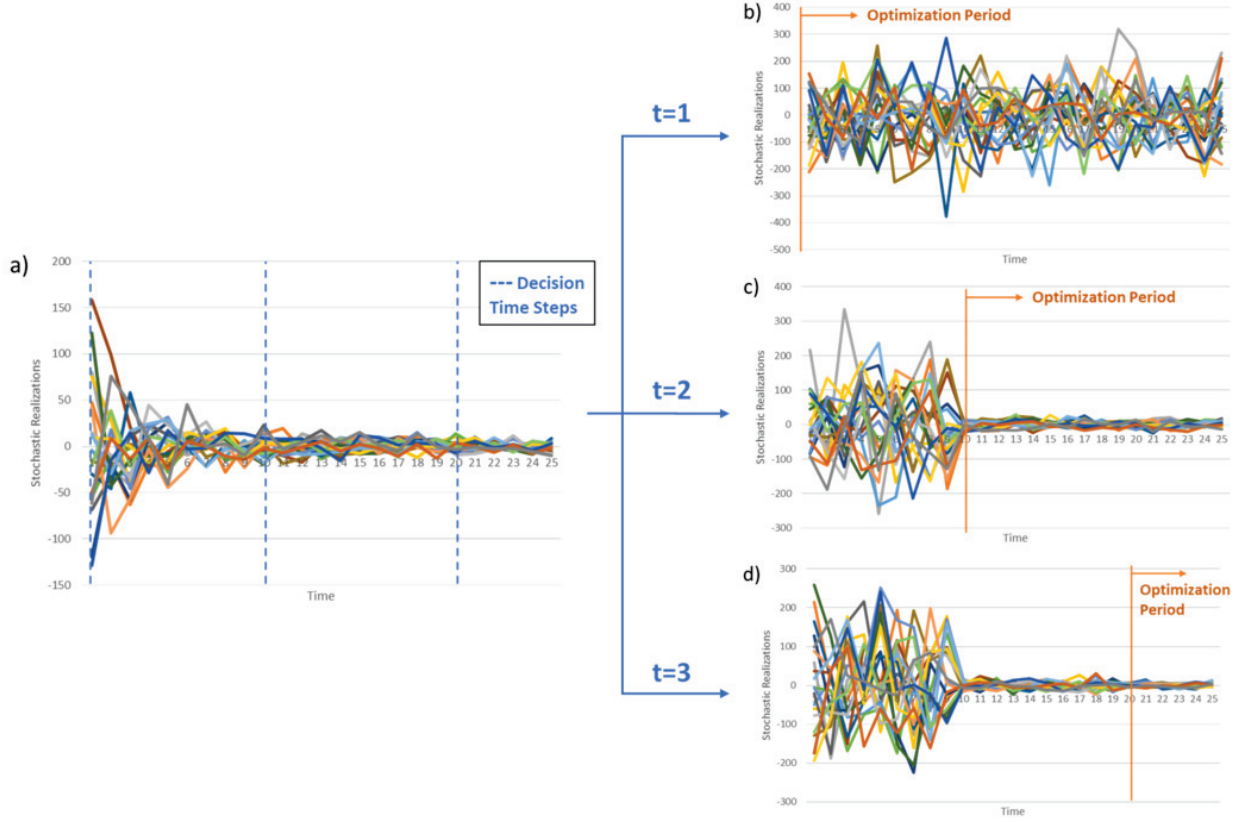


Figure 6: The figure highlights the benefit of a Dynamic model for stochastic variable scenario realizations. (a) shows the available knowledge of the stochastic parameter at each time period. (b) shows the information available for the multistage model at time $t=1$ and (c) and (d) show how the model is updated at $t=10$ and $t=20$ respectively.

- R_i Weighting parameter associated with the control term in the objective.
- w_{c_i} Weighting coefficient associated with the control objective function.
- w_{p_i} Weighting coefficient associated with the physical-design objective function.
- x_{i0} Initial conditions of state variables.

Decision Variables

- $u_i(t)$ Control variable corresponding to input applied to spring i at time t .
- $x_i(t)$ Position and velocity of the i^{th} mass at time t .
- y_i Local physical decision variables in i^{th} subsystem.

$y_s(i, j)$ Shared physical decision variables in i^{th} and j^{th} subsystems, i.e., combined, shared physical decision variables.

$$\min_{y_i, u_i(t)} \quad z = \sum_{i \in I} (10.24G) w_{p_i} \left(\frac{\pi}{12.8F_u} \right)^2 C^{-2} y_i^4 + \frac{1}{2} \sum_{i \in I} w_{c_i} \int_0^t (x_i^T Q_i x_i(t) + R_i u_i^2(t)) dt \quad (14a)$$

$$\text{s.t.} \quad (c^{-1} - 1)y_i - \kappa_1 C^{-1} \leq 0 \quad \forall i \quad (14b)$$

$$\kappa_2 - y_i \leq 1 \quad \forall i \quad (14c)$$

$$\frac{0.8F_u}{D_s G} C^3 y_i^{-2} \leq 1 \quad (14d)$$

$$\dot{x}_1(t) = \begin{bmatrix} 0 & 1 \\ -\frac{k_1+k_2}{m_1} & -\frac{c_1+c_2}{m_1} \end{bmatrix} x_1(t) + \begin{bmatrix} 0 & 0 \\ \frac{k_2}{m_1} & \frac{c_2}{m_1} \end{bmatrix} x_2(t) + \begin{bmatrix} 0 \\ \frac{1}{m_1} \end{bmatrix} u_1(t) \quad (14e)$$

$$\dot{x}_2(t) = \begin{bmatrix} 0 & 1 \\ -\frac{k_2+k_3}{m_2} & -\frac{c_2+c_3}{m_2} \end{bmatrix} x_2(t) + \begin{bmatrix} 0 & 0 \\ \frac{k_2}{m_2} & \frac{c_2}{m_2} \end{bmatrix} x_1(t) + \begin{bmatrix} 0 & 0 \\ \frac{k_3}{m_2} & \frac{c_3}{m_2} \end{bmatrix} x_3(t) + \begin{bmatrix} 0 \\ \frac{1}{m_2} \end{bmatrix} u_2(t) \quad (14f)$$

$$\dot{x}_3(t) = \begin{bmatrix} 0 & 1 \\ -\frac{k_3}{m_3} & -\frac{c_3}{m_3} \end{bmatrix} x_3(t) + \begin{bmatrix} 0 & 0 \\ \frac{k_3}{m_3} & \frac{c_3}{m_3} \end{bmatrix} x_2(t) + \begin{bmatrix} 0 \\ \frac{1}{m_3} \end{bmatrix} u_3(t) \quad (14g)$$

$$x_{i0} = [1, 1]^T, \quad \forall i$$

Table 1: Differences between the four models compared in this study

	Model 1	Model 2	Model 3	Model 4
Number of Y Variables	3	150	150	15
Number of K Variables	3	150	150	150
Number of Scenarios	15	15	15	15
Decision Time Steps	1	t	t	τ
Total Time Periods	T	T	T	T
Stochasticity of u_i	Random	Random	Decreasing	Decreasing

4.1 Model Descriptions

The four models that are compared in Table 1 are explicitly differentiated in this section. The differences between the model allow us to determine the importance of staging decisions when dealing with extraneous stochastic parameters.

4.1.1 Model 1

The first model is a slight variation of the model presented by [47] in Eqns. 14a to 14g. The main difference is that the control input, $u_i(t)$, is made stochastic, $u_i^s(t)$, where s denotes a discrete scenario realization of the control input. So it is adapted to include different possible values at each time period given a fixed mean and standard deviation that we draw upon for scenario realizations. The updated objective function is shown in Eqn. (15a). Moreover, Eqns. (14e) to (14g) should be updated as in Eqns. (15b) to (15e).

$$\min_{y_i, u_i^s(t)} \sum_{i \in I} (10.24G) w_{p_i} \left(\frac{\pi}{12.8F_u} \right)^2 C^{-2} y_i^4 + \frac{1}{2} \sum_{i \in I} w_{c_i} \int_0^t (x_i^T Q_i x_i(t) + \sum_{s \in S} p^s R_i u_i^{s2}(t)) dt \quad (15a)$$

$$\text{s.t.} \quad \text{Eqns. 14b, 14c, 14d} \quad (15b)$$

$$\dot{x}_1(t) = \begin{bmatrix} 0 & 1 \\ -\frac{k_1+k_2}{m_1} & -\frac{c_1+c_2}{m_1} \end{bmatrix} x_1(t) + \begin{bmatrix} 0 & 0 \\ \frac{k_2}{m_1} & \frac{c_2}{m_1} \end{bmatrix} x_2(t) + \begin{bmatrix} 0 \\ \frac{1}{m_1} \end{bmatrix} u_1^s(t) \quad \forall s \quad (15c)$$

$$\dot{x}_2(t) = \begin{bmatrix} 0 & 1 \\ -\frac{k_2+k_3}{m_2} & -\frac{c_2+c_3}{m_2} \end{bmatrix} x_2(t) + \begin{bmatrix} 0 & 0 \\ \frac{k_2}{m_2} & \frac{c_2}{m_2} \end{bmatrix} x_1(t) + \begin{bmatrix} 0 & 0 \\ \frac{k_3}{m_2} & \frac{c_3}{m_2} \end{bmatrix} x_3(t) + \begin{bmatrix} 0 \\ \frac{1}{m_2} \end{bmatrix} u_2^s(t) \quad \forall s \quad (15d)$$

$$\dot{x}_3(t) = \begin{bmatrix} 0 & 1 \\ -\frac{k_3}{m_3} & -\frac{c_3}{m_3} \end{bmatrix} x_3(t) + \begin{bmatrix} 0 & 0 \\ \frac{k_3}{m_3} & \frac{c_3}{m_3} \end{bmatrix} x_2(t) + \begin{bmatrix} 0 \\ \frac{1}{m_3} \end{bmatrix} u_3^s(t) \quad \forall s \quad (15e)$$

$$x_{i0} = [1, 1]^T, \quad \forall i$$

4.1.2 Model 2

Model 2 also considers the control input to be stochastic in nature, however, it also changes the main decision variable, y_i , the spring diameter, to be a dynamic variable that changes with time. Although for this specific application, it is not a realistic constraint, it allows us to compare the benefit of staging decisions against an ideal case. The new decision variable, y_{it} , means that the value of the spring thickness can change with time and no restriction is placed on the change. To accommodate this modification, Model 2 follows the same model form as Model 1, however, the objective function is updated in Eqn. (16).

$$\min_{y_i, u_i^s(t)} \sum_{t \in T} \sum_{i \in I} (10.24G) w_{p_i} \left(\frac{\pi}{12.8F_u} \right)^2 C^{-2} y_{it}^4 + \frac{1}{2} \sum_{i \in I} w_{c_i} \int_0^t (x_i^T Q_i x_i(t) + \sum_{s \in S} p^s R_i u_i^{s2}(t)) dt \quad (16)$$

4.1.3 Model 3

Model 3 is similar to Model 2 in the sense that the main decision variable is dynamic in t and the control input is stochastic, yet, the variance of the input parameter, $u_i(t)$ decreases with time. In this case, it is assumed that the mean of the variable does not change. For this model, the formulation is the same as that of Model 2. The only difference lies in the value of the input.

4.1.4 Model 4

Model 4 differs from Model 3 with respect to the frequency that the main decision variable, y_{it} can change. In Model 2, the decision variable can change in every time period t , however, that is not a realistic scenario in many applications. In some applications, changing the value of the decision variable incurs a change cost in addition to the fixed cost of the variable being considered. In order to have a more realistic comparison, the change in the decision variable is assumed to only take place at staged intervals, τ . In this case, the prediction of the stochastic parameter is updated at these staged intervals as well. Therefore, the updated decision variables are $y_{i\tau}$ and the scenarios for the control input are updated at every τ stage (assuming new information is available). The updated formulation is presented in Eqns. 17a to 17e.

$$\begin{aligned} \min_{y_i, u_i^s(t)} \quad & \sum_{i \in I} \sum_{\tau \in T} (10.24G) w_{p_i} \left(\frac{\pi}{12.8F_u} \right)^2 C^{-2} y_{i\tau}^4 \\ & + \frac{1}{2} \sum_{i \in I} w_{c_i} \int_0^t (x_i^T Q_i x_i(t) + \sum_{s \in S} p^s R_i u_i^{s2}(t)) dt \end{aligned} \quad (17a)$$

$$\text{s.t.} \quad (c^{-1} - 1)y_{i\tau} - \kappa_1 C^{-1} \leq 0 \quad \forall i, \tau \quad (17b)$$

$$\kappa_2 - y_{i\tau} \leq 1 \quad \forall i, \tau \quad (17c)$$

$$\frac{0.8F_u}{D_s G} C^3 y_{i\tau}^{-2} \leq 1 \quad \forall \tau \quad (17d)$$

$$\text{Eqns. 14e to 14g} \quad (17e)$$

$$x_{i0} = [1, 1]^T, \quad \forall i$$

4.2 Results

Each of the four problems has been solved utilizing the same parameter values. The output of the models is presented in Table 2. The relative error for each model is also recorded due to slight inconsistencies with the finite difference method for the differential equations. The results show that Model 3 has the lowest objective value, which is desirable since the objective of the model is to be minimized. This is intuitive since the value of the decision variable is adapted at each time period to compliment the newest prediction of the stochastic control input, $u_i(t)$, where the variability is at its lowest since new information is known. However, as mentioned previously, this is not representative of reality in the sense that it is sometimes not feasible to constantly change the decision variable. For this purpose, the effect of staging the decision is studied at intervals of 1 second time periods. Additionally, the results show that Model 4 has a significant advantage in terms of objective value over the remaining two models. Model 2 also shows comparable results from Model 3, however, this model variation considers dynamically changing the decision variable, which is also unrealistic, given a randomly distributed control variable whose variation is not time dependent. These considerations are not representative of reality. As for model 1, not adapting to the changing input causes the model to remain highly sub-optimal even though the computational

time was lower higher than Model 4 since a single value for each spring thickness is obtained for the model and different values of the parameter k do not need to be evaluated at each time period. These results show the need for staging decisions when predictions about the value of the input variable improve as the observer approaches the prediction period. In terms of the computational cost results, although model 4 incurs the second highest computational cost, it is very close to the other models, especially considering that the staged time periods and the original time indices need to be coordinated. Model 2 has the highest computational cost as the decision variable values need to be determined at each period, and the variability of the stochastic parameter $u_i(t)$ is high and this makes the coordination of the model variables difficult with the differential equations.

Table 2: The output of the models is compared.

	Model 1	Model 2	Model 3	Model 4
Objective Value	9.73×10^{18}	6.29×10^{18}	4.45×10^7	3.05×10^{18}
Relative Error	1.54×10^{-1}	7.14×10^{-1}	9.85×10^{-2}	7.16×10^{-2}
Computation Time (s)	0.087	0.601	0.110	0.191
Order of Solutions	4	3	1	2%

5 Battery Cooling System Case Study

The proposed DSMD framework is applied to a battery management system with updated temperature measurements to determine the optimal layout of the battery pack and decide the coolant flow-rate over time. Section 5.1 gives a detailed overview of the problem and Section 5.2 presents the details of the optimization problem and the results.

5.1 Problem Description

With the booming of electric vehicles, immersion cooling has become a potential cooling method for high-performance vehicles. For an immersion cooling system of the battery pack, plant design along with control design need to be considered. The immersion cooling system needs to cool down eight battery cells effectively while simultaneously minimizing the energy consumption of the pump. To achieve the optimal design, we first tested 21700 battery cells and immersion cooling battery packs so that we were able to build the high-fidelity finite element (FE) model in COMSOL Multiphysics [49]. As shown in Fig. 7a, there are 8 battery cells in the battery pack, and 4C discharging of state of charge (SOC) from 100% to 20% was simulated. The discharge process takes 720 s and was divided into 4 decision stages (180 s each). The distance between battery cells can be fixed (d), to 0.5 mm, 1.0 mm, or 1.5 mm. We adopted open-loop control for the control of the coolant flow rate (v), with stages 1 and 2 having a flow rate of 5 or 10 kg/s, while stages

3 and 4 have a flow rate of 10 or 15 kg/s. Because stages 3 and 4 have a higher temperature of battery cells than stages 1 and 2, and thus need a higher flow rate to cool down the battery cells. Based on the FE model, we can obtain the temperature of battery cells (Fig. 7b) and the energy consumption of pumping the coolant in different scenarios. The pump energy consumption (P) [50] can be calculated as shown in Eqn. (18).

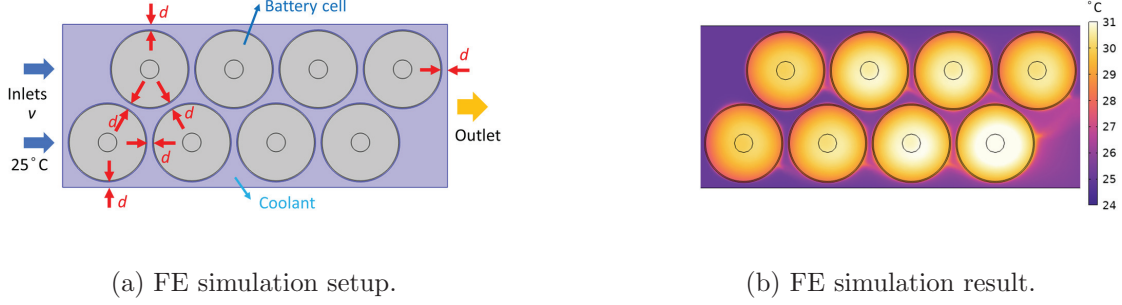


Figure 7: Battery pack immersion cooling system.

$$P = \int v \Delta p \, dt \quad (18)$$

where Δp is the pressure drop of the inlet to outlet. Since the maximum temperature (T_{max}) of battery cells also influences the performance of the battery pack, it is monitored when it goes above 30°C since it will decrease the battery pack's performance [51]. The objective function considers both the energy consumption of the pump and the maximum temperature of battery cells.

$$Cost = w_1 * P + w_2 * (T_{max} - 30) \quad (19)$$

where w_1 and w_2 are the weighting factors of the energy consumption of the pump and the maximum temperature of battery cells.

5.2 Case Study Results

The FE simulation was run for different designs of the cooling system for the battery pack. A full factorial design with 3 plant designs and 2 levels for each stage's flow rate control design was considered. The total time was 720 seconds, and 4 decision stages were studied, each consisting of 180 seconds. The design of the battery pack was considered only as a first stage decision and a static multi-stage model was implemented to determine the predicted optimal design. Then the design was fixed, and a temperature reading was used to update the FE simulation at each stage. The two objectives, reducing the energy consumption and ensuring that the temperature of the battery remains under 30°C are accounted for in a bi-objective model. The objectives are normalized by their max value for each of the different objectives, to ensure that a fair comparison between them is conducted and calculated as shown in Eqn. (19).

In order to conduct a DMSD study on the battery pack design, a three-prong solution process is conducted as seen in Fig. 8. First, the simulation results from the FE model are generated following a Design of Experiments for 3 design diameters and 2 flow rates at each stage. Then, a Gaussian Process Regression (GPR) model was implemented to extract more design options, where 4 levels of each of the 5 design variables was considered. This increased the dataset to include a total of 1,024 different designs. Then, the multi-objective multistage optimization model was conducted to find the optimal result over 720 seconds. At the second stage, a temperature reading of the battery is used to update the operating temperature of the battery in the simulation and it is re-simulated for the remaining time periods. The process is then repeated for the GPR and DMSD models in order to optimize the coolant flow rates for the remaining time periods.

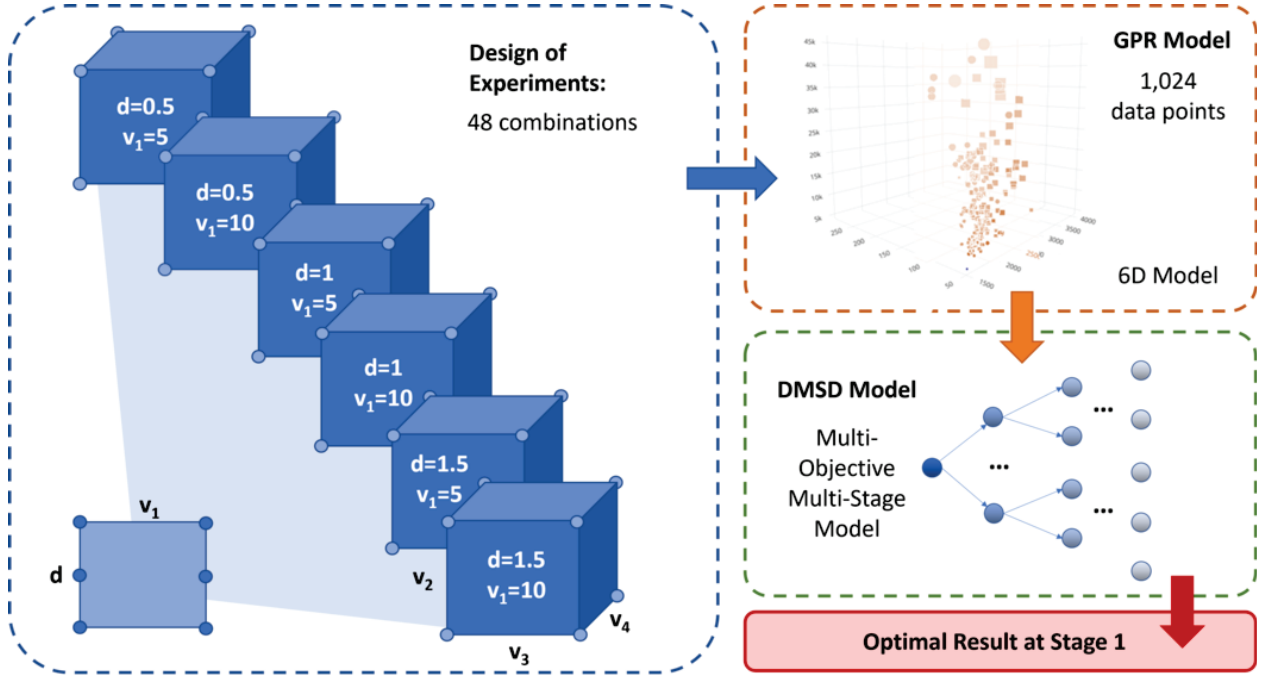


Figure 8: The framework for the first stage solution process.

The results of this design are presented in Fig. 9, where the optimal design was a diameter distance of 1.5 mm, and the flow rate design is updated at each of the stages given the new battery temperature measurements. As seen, the flow rate changes given the new information, which allows for a more effective control of the cooling system. The starting temperatures for stages 2, 3, and 4 are 28.35°C, 29.85°C, and 26.85°C, respectively. The costs of the designs are also shown in the figure and they refer to the energy consumption of the design. The shaded region shows the possible design values for the coolant flow rate over time.

The case study provides an implementation for the DMSD framework for problems where the analysis is not strictly numerical. The results shown, that by incorporating real-time information, the design of the system can be updated to improve both the performance as well as the cost of

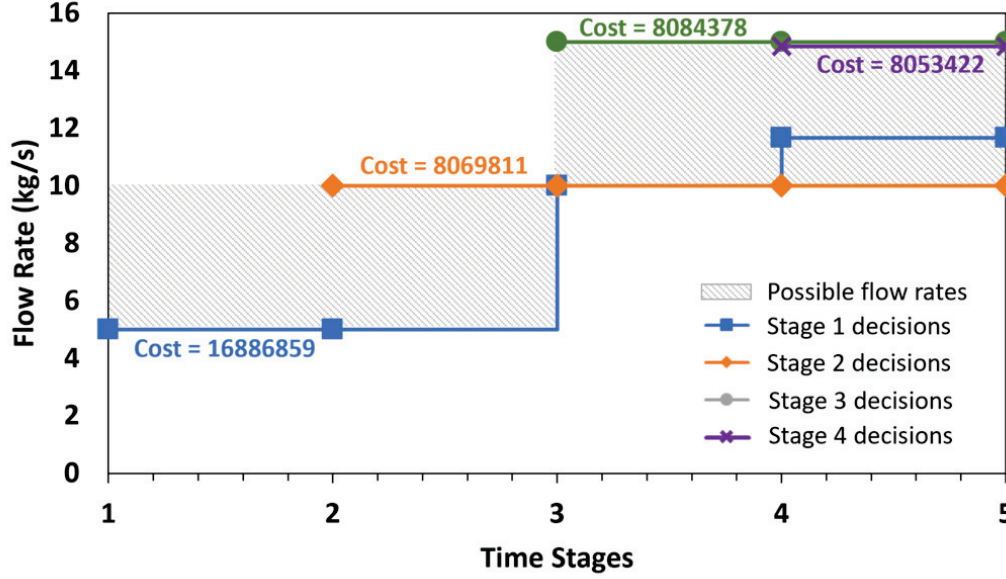


Figure 9: The optimized coolant flow rate over 720 seconds with decisions made over four stages. The shaded area represents the possible flow rates and the points represent the optimization results. The energy consumption cost of each of the stages is also presented.

running the system.

6 IEEE-30 Bus Case Study

The importance of staged deployment for short-term operational decisions is also investigated for volatile stochastic variables. The IEEE-30 bus system is used as a case study to implement staged unit commitment for a day ahead system. This case study illustrates the benefit of providing updated demand data to improve short-term demand forecasting. An illustration of the power system considered in this case study is shown in Fig. 10 below. The system consists of 30 buses and 6 generators placed, as seen in Fig. 10. The generator and transmission line data is collected from [52]. The staged formulation considers updated demand and wind generation distributional data every 6 hours (24 hours discretized into 4 time periods). This model considers 20 scenarios at each staged time period where the total number of scenarios is $20^3 = 8000$. The model decides the generator schedule at each staged time period as well as the forecasted unit commitment. Due to the stochasticity of the load demand and wind generation, given a realization of a stage, the next stage considers the previously made decisions as fixed values and updates the generator operating schedule for the upcoming stages. While the model consists of 4 stages, the system considers and hourly time discretization.

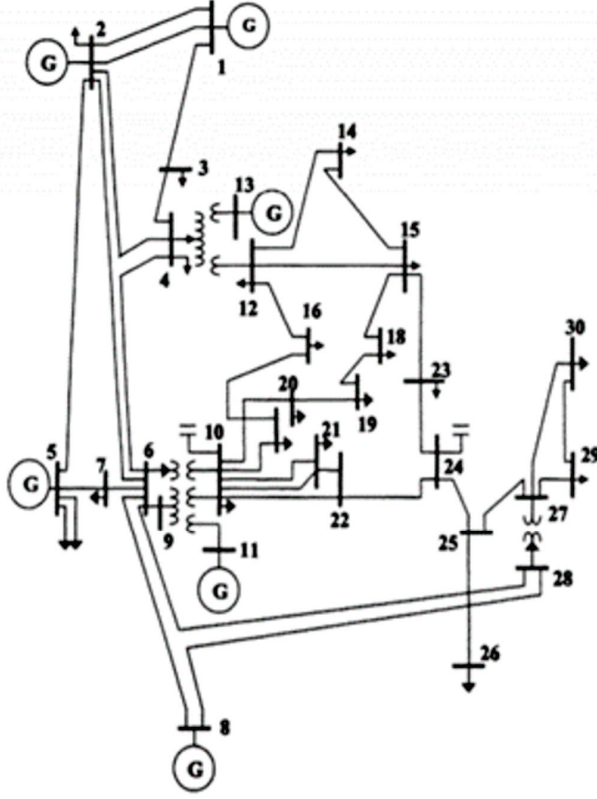


Figure 10: IEEE 30-Bus System as presented in [52]

6.1 Model Formulation

The model formulation is based on the formulation in [53] but extended to a multi-stage stochastic formulation. The updated formulation is presented in Eqns. 20a to 20k.

Nomenclature

Indices

t	Set of time periods indexed by $t, k \in T$.
st	Set of staged decision time periods indexed by $st \in T$.
b	Set of buses indexed by $b, i, j \in B$.
g	Set of generators indexed by $g, q \in G$.
s	Set of scenarios indexed by $s \in S$.

Parameters

p^s	Probability of scenario s occurring.
-------	--

C_{gb}	Power generation cost using generator g at bus b in MBtu/MW.
SU_{gb}	Startup cost for generator g at bus b .
SD_{gb}	Shutdown cost for generator g at bus b .
$d_{b,st}$	Unit commitment prediction realizations at bus b at time t under scenario s .
MI_{gb}	Minimum generating capacity of generator g at bus b .
MA_{gb}	Maximum generating capacity of generator g at bus b .
RU_{gb}	Ramp-up rate limit of generator g at bus b .
RD_{gb}	Ramp-down rate limit of generator g at bus b .
F_{bij}	Line flow distribution factor based on the net injection of bus b at the transmission line linking buses i and j .
U_{ij}	Transmission line capacity linking buses i and j .
w_{bt}^s	Wind generation realization at bus b at time t under scenario s .

Decision Variables

x_{gbt}^s	Power generation from generator g at bus b at time t under scenario s .
y_{gbt}^s	Binary variable: 1 if generator g at bus b at time t under scenario s is on, 0 otherwise.
u_{gbst}^s	Binary variable: 1 if generator g at bus b at staged time period st under scenario s is turned on, 0 otherwise.
v_{gbst}^s	Binary variable: 1 if generator g at bus b at staged time period st under scenario s is turned off, 0 otherwise.

$$\min_z \sum_g \sum_b \sum_s p^s \left(\sum_t C_{gb} x_{gbt}^s + \sum_{st} SU_{gb} u_{gbst}^s + SD_{gb} v_{gbst}^s \right) \quad (20a)$$

$$\text{s.t.} \quad y_{gbt-1}^s - y_{gbt}^s + y_{gbk}^s \leq 0 \quad \forall s, g, b, t : 1 \leq k - (t - 1) \leq MU_{gb} \quad (20b)$$

$$-y_{gbt-1}^s + y_{gbt}^s - y_{gbk}^s \leq 1 \quad \forall s, g, b, t : 1 \leq k - (t - 1) \leq MD_{gb} \quad (20c)$$

$$-y_{gbt-1}^s + y_{gbt}^s - u_{gbst}^s \leq 0 \quad \forall s, g, b, t : st - 1 < t \leq st \quad (20d)$$

$$y_{gbt-1}^s - y_{gbt}^s - v_{gbst}^s \leq 0 \quad \forall s, g, b, t : st - 1 < t \leq st \quad (20e)$$

$$MI_{gb} y_{gbt}^s \leq x_{gbt}^s \quad \forall s, g, b, t \quad (20f)$$

$$x_{gbt}^s \leq MA_{gb} y_{gbt}^s \quad \forall s, g, b, t \quad (20g)$$

$$x_{gbt}^s - x_{gbt-1}^s \leq y_{gbt-1}^s RU_{gb} + (1 - y_{gbt-1}^s) MA_{gb} \quad \forall s, g, b, t \quad (20h)$$

$$x_{gbt-1}^s - x_{gbt}^s \leq y_{gbt}^s RD_{gb} + (1 - y_{gbt}^s) MA_{gb} \quad \forall s, g, b, t \quad (20i)$$

$$\sum_b \sum_g (x_{gbt}^s + w_{bt}^s) \geq \sum_b d_{b,st} \quad \forall s, t \quad (20j)$$

$$-U_{ij} \leq \sum_b F_{bij} \left(\sum_g (x_{gbt}^s + w_{bt}^s) - d_{bt}^s \right) \leq U_{ij} \quad \forall s, i, j, t \quad (20k)$$

The formulation above denotes a unit commitment model where the decisions of when to turn the generators on or off can be made at staged intervals, st . The objective aims to minimize the total cost which includes the production cost and the cost of turning the generators off and on. Constraints 20b and 20c ensure that the generators are operated within the minimum and maximum uptime. Constraints 20d and 20e guarantee that the operating state of the generators does not change unless the generators are turned off or on. Constraints 20f and 20g maintain that the total generated capacity at a generator, if on, remains within the maximum and minimum generating capacities. The next set of constraints, constraints 20h and 20i, limit the change in generated capacity to be within predetermined generator thresholds. Constraint 20j represents the unit commitment constraint to ensure that the demand is met. Lastly, constraint 20k represents the capacity limits of the transmission lines.

In order to consider stochastic and dynamic demand patterns, a typical consumer load profile is used to predict the system behavior. Fig. 11 below presents a typical load profile from which the demand pattern is adapted in order to maintain realistic load behavior. Since the IEEE-30 bus system provides average load values at each bus, these were utilized to ensure that the demand pattern for each bus maintained a consistent load with the dataset. In order to model the different demand behaviors expected at each stage, a Weibull distribution is used. Weibull distributions are useful for demand predictions as they can represent a multitude of different stochastic parameter behaviors. For each stage, the pattern of demand changes given updated information, and each of the scenarios used for the deterministic equivalent formulation has a specific value for the Weibull hyperparameters, θ and β , which are used to represent the distribution as seen in Eqn. (21).

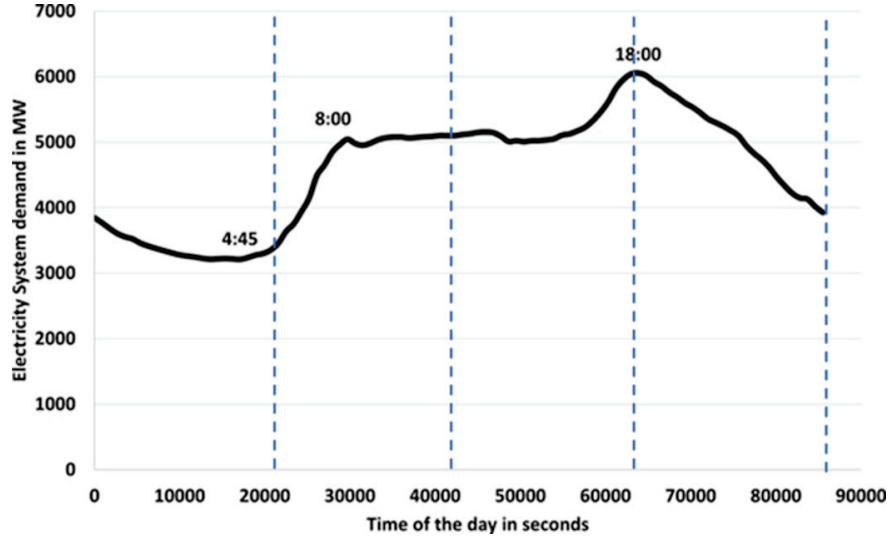


Figure 11: Hourly load profile behavior divided into stages. Adapted from [54].

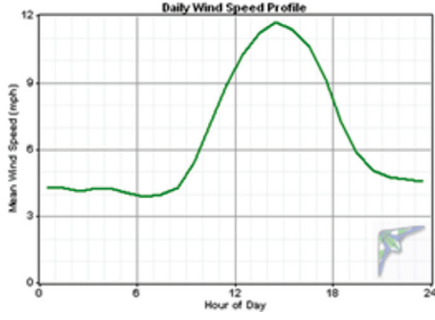


Figure 12: Average daily wind speed [55]

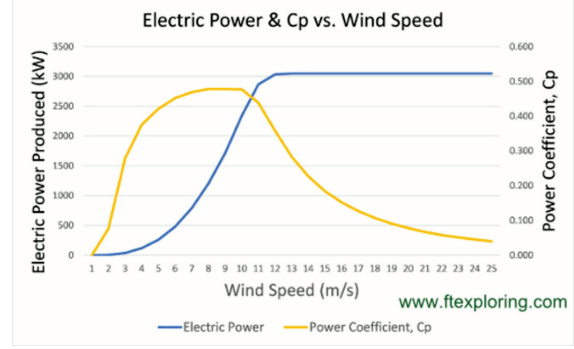


Figure 13: Electric power produced based on wind speed [56]

$$f(t) = \frac{\beta t^{\beta-1}}{\theta^\beta} e^{-(\frac{t}{\theta})^\beta} \quad (21)$$

In this case, the parameters of the Weibull distribution are varied in each stage to represent the same behavior with different parameter values. However, the behavior itself differs between stages based on Fig. 5. In terms of the wind generation profile, it is based on the average hourly wind speed profile from [55] as seen in Fig. 12. The wind generation is assumed to be directly proportional to the wind speed and is scaled as in Fig. 13.

In order to generate realistic wind data, Table 4 summarizes the wind speed peaks that are achieved at each of the buses with wind turbines. In all cases, we assume that the wind speed follows a characteristic curve as shown in Fig. 12, scaled to account for area variations. The peak power is approximated roughly from Fig. 13.

The model was modeled in GAMS and solved using the BARON MINLP solver embedded in GAMS IDE.

Table 4: Generated Wind Profiles

Bus Location	Peak Wind Speed (m/s)	Peak Power (kW)
9	10	2400
22	7	750
26	12	3000%

6.2 Results

The model was compared to the two-stage stochastic model, which is referred to as the static model (due to its use of static demand information available at the first time stage), and the results

show the benefit of staging decisions based on new approximations of stochastic variables. This case study allows for a realistic comparison of the two methods as it accounts for different demand realizations that could occur as well as different wind availability. The generator operation is the main first stage decision variable in the two-stage model and it is the staged decision in the multi-stage model. The production of each generator is a dynamic variable that depends on the demand realization in both models.

Table 5: Comparing results for the Dynamic and Static models of the IEEE-30 Bus System

Scenario	Dynamic	Static	% Improvement	Scenario	Dynamic	Static	% Improvement
1	17005.56	17654.97	3.6783	11	17468.13	17909.72	2.4656
2	17433.59	17672.27	1.3505	12	17402.18	17519.55	0.6699
3	16896.36	17511.1	3.5105	13	17381.29	17734.85	1.9935
4	17089.54	17770.71	3.8331	14	17120.56	17034.66	-0.5042
5	17497.84	17514	0.0922	15	17127.39	17082.26	-0.2641
6	17197.75	17184.31	-0.0782	16	17136.29	17431.55	1.6938
7	17412.12	17375.04	-0.5606	17	17309.38	18117.14	4.4585
8	16892.22	17105.91	1.2492	18	17752.22	17977.17	1.2513
9	16574.4	17396.37	4.7249	19	17274.67	17871.15	3.3376
10	17447.7	17514.38	0.3807	20	17456.58	17597.19	0.7990
Average				1.7041			

The model compares the static two-stage stochastic model with the dynamic multi-stage model with different scenario realizations in Table 5 without accounting for wind turbines in the model. The same scenarios are used to compare the total system costs between the two models. Using the 20 scenarios that the model was trained on; The generator operation results are used as an input to update the power generation given the demand realizations. The results show that there are some instances where the static model outperforms the dynamic, multistage model, however, on average, the use of the DMSD model has an improvement of approximately 1.70% in objective function value. Moreover, the average dynamic model cost, 17228.97, can be compared with the average static model cost, 17470.71, to get an improvement of 241.74. In both cases, the the dynamic, muti-stage model achieves the best results. These results are expected since the stochasticity in the model is only contained in the realization of the demand, therefore, if the expectation of the demand values are close to the realization, then the dynamic model might be comparable to the static model.

The model is then extended to the case in which three turbines are added to buses 9, 22 and 26, as in Table 4. Fig. 14 shows the results of running the dynamic model with the addition of three wind turbines as previously described. The results show the wind turbine schedule for the

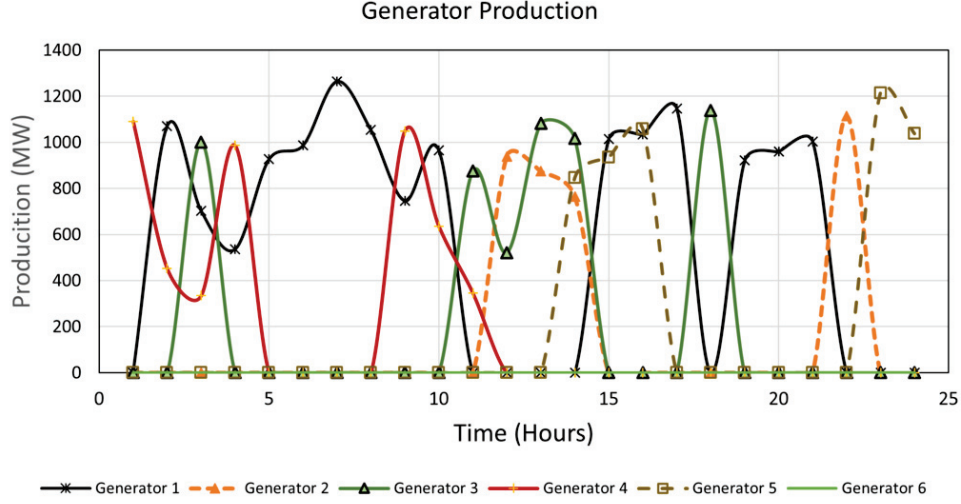


Figure 14: The results of the production of the 6 generators per hour for the dynamic model

multi-stage model. As seen, they show an almost linear production level by the active generators (generator 6 is not used). However, we can see that during the peak hours extracted from Fig. 11, the generators tend to have a higher production level. This production amount, in addition to the wind profile, allow the model to meet the hourly demand. The model chose not to utilize generator 6 given the additional wind turbines introduced to the system and since the associated costs with that generator were higher. Additionally, generator 6 had a small interval for varying production.

Table 6: Table showing the results of the Multi-stage model and the Two-Stage Stochastic model considering the addition of three wind turbines in the system.

Scenario	Dynamic	Static	Difference	Scenario	Dynamic	Static	Difference
1	12649.13	18070.19	5421.05	11	12812.16	18303.08	5490.92
2	12672.83	18104.04	5431.21	12	12524.29	17891.85	5367.55
3	12505.78	17865.4	5359.61	13	12661.04	18087.2	5426.16
4	12709.06	18155.8	5446.73	14	12192.75	17418.21	5225.46
5	12536.35	17909.07	5372.72	15	12247.78	17496.83	5249.04
6	12319.85	17599.78	5279.93	16	12454.4	17791.99	5337.59
7	12397.92	17711.32	5313.39	17	12922.75	18461.06	5538.31
8	12210.44	17443.49	5233.05	18	12311.41	17587.72	5276.31
9	12453.54	17790.77	5337.23	19	12281.82	17545.45	5263.63
10	12535.16	17907.37	5372.21	20	12585.7	17979.57	5393.87

The updated model, referred to as the wind dynamic model, is tested against the static model to test how well they work for 20 random scenarios in Table 6. As in Table 5, the operating schedule of the dynamic and static models are fixed based on the results of the model. However, based on the realization of the stochastic parameters (demand and wind generation), the production of each generator is updated. The results in Table 6 show that the dynamic multi-stage model always outperforms the static model by over 5000 dollars every time (with a cost-minimizing objective). While the difference in the objective values of the models seems constant, this is due to the fact that the main cost difference arises from changing the generator operation and the deviation in production quantities does not drastically change the objective value. There is a significant improvement in the results of the wind dynamic model over the dynamic model that does not consider wind generation. This is mostly due to the fact that the wind dynamic model incorporates more stochasticity that can greatly alter the results of the model. The dynamic model can then consider robust solutions to handle variations in the stochastic realizations.

7 Conclusions and Future Directions

This paper highlights the need to implement staged decision analysis in the modeling of real systems in order to improve system costs, minimize computation time, and make use of the property of diminishing stochasticity with time. This property is a result of stochastic processes that follow a Markov-based stochastic pattern, where with the passing of time, the prediction of the uncertain parameter is improved. In such systems, utilizing staged decisions becomes advantageous in exploiting the stochastic nature of the system. The paper presents the DMSD framework for solving multi-staged models by taking advantage of the modular structure of the system. In order to understand the true benefit of staging decisions, the impact of studying different decision periods, τ , should also be studied. Additionally, the costs associated with changing the value of the decision variable at each stage should also be studied to better understand the added benefit of the model. With this added cost, a tradeoff between the number of decisions to be made and the overall system costs is introduced in the system, where the number of staged decisions becomes a variable. Furthermore, the paper identifies a need for solution procedures to mitigate the increased solution time of discretizing the random variables. Moreover, the benefit of implementing multi-stage models is tested on a numerical case study as well as a real case study of the IEEE-30 bus system, in which the overall cost benefits are investigated. The paper also accounts for the added stochasticity of having a wind source as well. The results clearly show a benefit to implementing multi-staged models, especially for problems with high stochasticity. Alternatives to discretizing stochastic processes should also be investigated to find optimization approaches that can account for stochastic variables and apply MDO theory to exploit the subsystem design of the problem while simultaneously accounting for the dependencies between them.

Acknowledgments

This work was partially supported by the National Science Foundation (NSF) through the supplement project to the Engineering Research Center for Power Optimization of Electro-Thermal Systems (POETS) with cooperative agreement EEC-1449548, and through the Faculty Early Career Development awards CMMI-1942559 and CBET-1554018.

8 Declarations

8.1 Conflict of Interest

The authors declare that they have no conflict of interest.

8.2 Replication Of Results

The full formulation for the case studies considered is provided in the paper. The scenarios were generated stochastically and changed in every run based on the method detailed in the paper. GAMS modeling software was used with BARON optimization solver used for the independent runs and the DSMD solution framework provided. The IEEE-30 Bus Case Study has been referenced to refer the reader to the full data available online. The codes are available upon request by contacting the corresponding author.

References

- [1] H. W. Lee, P. C. Jakob, K. Ho, S. Shimizu, and S. Yoshikawa, “Optimization of satellite constellation deployment strategy considering uncertain areas of interest,” *Acta Astronautica*, vol. 153, pp. 213–228, 2018.
- [2] O. L. De Weck, R. De Neufville, and M. Chaize, “Staged deployment of communications satellite constellations in low earth orbit,” *Journal of Aerospace Computing, Information, and Communication*, vol. 1, no. 3, pp. 119–136, 2004.
- [3] B. Hamdan, K. Ho, and P. Wang, “Staged-deployment optimization for expansion planning of large scale complex systems,” in *AIAA Scitech Forum, 2019*, American Institute of Aeronautics and Astronautics Inc, AIAA, 2019.
- [4] S. Li, Y. Huang, and S. J. Mason, “A multi-period optimization model for the deployment of public electric vehicle charging stations on network,” *Transportation Research Part C: Emerging Technologies*, vol. 65, pp. 128–143, 2016.

- [5] R. A. White, C. Donayre, I. Walot, J. Lee, and G. E. Kopchok, "Regression of a descending thoracoabdominal aortic dissection following staged deployment of thoracic and abdominal aortic endografts," *Journal of Endovascular Therapy*, vol. 9, no. 2_suppl, pp. II–92, 2002.
- [6] N. Yodo and P. Wang, "Engineering resilience quantification and system design implications: A literature survey," *Journal of Mechanical Design*, vol. 138, no. 11, p. 111408, 2016.
- [7] N. Yodo, P. Wang, and M. Rafi, "Enabling resilience of complex engineered systems using control theory," *IEEE Transactions on Reliability*, vol. 67, no. 1, pp. 53–65, 2017.
- [8] N. Yodo and P. Wang, "Resilience allocation for early stage design of complex engineered systems," *Journal of Mechanical Design*, vol. 138, no. 9, p. 091402, 2016.
- [9] P. Wang, Z. Wang, and A. T. Almaktoom, "Dynamic reliability-based robust design optimization with time-variant probabilistic constraints," *Engineering Optimization*, vol. 46, no. 6, pp. 784–809, 2014.
- [10] X. Yin and İ. E. Büyüktaktakın, "Risk-averse multi-stage stochastic programming to optimizing vaccine allocation and treatment logistics for effective epidemic response," *IIEE Transactions on Healthcare Systems Engineering*, pp. 1–23, 2021.
- [11] S. Bushaj, İ. E. Büyüktaktakın, and R. G. Haight, "Risk-averse multi-stage stochastic optimization for surveillance and operations planning of a forest insect infestation," *European Journal of Operational Research*, 2021.
- [12] D. V. Steward, "The design structure system: A method for managing the design of complex systems," *IEEE transactions on Engineering Management*, no. 3, pp. 71–74, 1981.
- [13] J. Agte, O. De Weck, J. Sobieszczanski-Sobieski, P. Arendsen, A. Morris, and M. Spieck, "Mdo: assessment and direction for advancement—an opinion of one international group," *Structural and Multidisciplinary Optimization*, vol. 40, no. 1, pp. 17–33, 2010.
- [14] S. Tosserams, L. P. Etman, and J. Rooda, "A classification of methods for distributed system optimization based on formulation structure," *Structural and Multidisciplinary Optimization*, vol. 39, no. 5, pp. 503–517, 2009.
- [15] C. Bloebaum, "Coupling strength-based system reduction for complex engineering design," *Structural optimization*, vol. 10, no. 2, pp. 113–121, 1995.
- [16] T. C. Wagner, *A general decomposition methodology for optimal system design*. PhD thesis, University of Michigan, 1993.
- [17] G. Feng, "A decomposition algorithm for nonlinear inseparable hierarchical optimization problems," *Structural optimization*, vol. 5, no. 3, pp. 184–189, 1993.

- [18] J. Sobieszczanski-Sobieski, T. D. Altus, M. Phillips, and R. Sandusky, “Bilevel integrated system synthesis for concurrent and distributed processing,” *AIAA Journal*, vol. 41, no. 10, pp. 1996–2003, 2003.
- [19] R. Braun and I. Kroo, “Development and application of the collaborative optimization architecture in a multidisciplinary design environment,” *Multidisciplinary design optimization: state of the art*, vol. 80, p. 98, 1997.
- [20] H. M. Kim, N. F. Michelena, P. Y. Papalambros, and T. Jiang, “Target cascading in optimal system design,” *J. Mech. Des.*, vol. 125, no. 3, pp. 474–480, 2003.
- [21] S.-I. Yi, J.-K. Shin, and G. Park, “Comparison of mdo methods with mathematical examples,” *Structural and Multidisciplinary Optimization*, vol. 35, no. 5, pp. 391–402, 2008.
- [22] A. de Wit and F. van Keulen, “Numerical comparison of multi-level optimization techniques,” in *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, p. 1895, 2007.
- [23] K. Lewis and F. Mistree, “The other side of multidisciplinary design optimization: Accommodating a multiobjective, uncertain and non-deterministic world,” *Engineering Optimization*, vol. 31, no. 2, pp. 161–189, 1998.
- [24] Z. Wang and P. Wang, “A nested extreme response surface approach for time-dependent reliability-based design optimization,” *Journal of Mechanical Design*, vol. 134, no. 12, 2012.
- [25] W. B. Powell, “Sequential decision analytics and modeling: Modeling with python,” *Foundations and Trends in Technology, Information and Operations Management*, 2022.
- [26] M. W. Tackett, C. A. Mattson, and S. M. Ferguson, “A model for quantifying system evolvability based on excess and capacity,” *Journal of Mechanical Design*, vol. 136, no. 5, p. 051002, 2014.
- [27] A. Engel and T. R. Browning, “Designing systems for adaptability by means of architecture options,” *Systems Engineering*, vol. 11, no. 2, pp. 125–146, 2008.
- [28] N. Noyan, “Risk-averse two-stage stochastic programming with an application to disaster management,” *Computers & Operations Research*, vol. 39, no. 3, pp. 541–559, 2012.
- [29] S. Ahmed, “A scenario decomposition algorithm for 0–1 stochastic programs,” *Operations Research Letters*, vol. 41, no. 6, pp. 565–569, 2013.
- [30] Z. Zhou, J. Zhang, P. Liu, Z. Li, M. C. Georgiadis, and E. N. Pistikopoulos, “A two-stage stochastic programming model for the optimal design of distributed energy systems,” *Applied Energy*, vol. 103, pp. 135–144, 2013.

- [31] G. Barbarosoğlu and Y. Arda, “A two-stage stochastic programming framework for transportation planning in disaster response,” *Journal of the operational research society*, vol. 55, no. 1, pp. 43–53, 2004.
- [32] J. Bengtsson, “Manufacturing flexibility and real options: A review,” *International Journal of Production Economics*, vol. 74, no. 1-3, pp. 213–224, 2001.
- [33] R. J. Harrington and M. C. Ottenbacher, “Decision-making tactics and contextual features: Strategic, tactical and operational implications,” *International Journal of Hospitality & Tourism Administration*, vol. 10, no. 1, pp. 25–43, 2009.
- [34] A. Alonso-Ayuso, L. F. Escudero, M. Guignard, and A. Weintraub, “On dealing with strategic and tactical decision levels in forestry planning under uncertainty,” *Computers & Operations Research*, vol. 115, p. 104836, 2020.
- [35] E. H. Bowman and G. T. Moskowitcz, “Real options analysis and strategic decision making,” *Organization science*, vol. 12, no. 6, pp. 772–777, 2001.
- [36] W. Kenton, “Net Present Value (NPV).”
- [37] J. Fairbrother, A. Turner, and S. W. Wallace, “Problem-driven scenario generation: an analytical approach for stochastic programs with tail risk measure,” *Mathematical Programming*, pp. 1–42, 2019.
- [38] M. Kaut and W. Stein, *Evaluation of scenario-generation methods for stochastic programming*. Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät . . . , 2003.
- [39] A. I. Kibzun and E. A. Kuznetsov, “Analysis of criteria var and cvar,” *Journal of Banking & Finance*, vol. 30, no. 2, pp. 779–796, 2006.
- [40] K. Høyland, M. Kaut, and S. W. Wallace, “A heuristic for moment-matching scenario generation,” *Computational optimization and applications*, vol. 24, no. 2, pp. 169–185, 2003.
- [41] S. Mehrotra and D. Papp, “Generating moment matching scenarios using optimization techniques,” *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 963–999, 2013.
- [42] T. Bjerring, “Scenario generation and moment matching,” Dec 2019.
- [43] J. R. Birge, “Decomposition and partitioning methods for multistage stochastic linear programs,” *Operations research*, vol. 33, no. 5, pp. 989–1007, 1985.
- [44] S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley, “Notes on decomposition methods,” *Notes for EE364B, Stanford University*, vol. 635, pp. 1–36, 2007.
- [45] A. Ruszczyński, “Parallel decomposition of multistage stochastic programming problems,” *Mathematical programming*, vol. 58, no. 1, pp. 201–228, 1993.

- [46] M. L. Fisher, “An applications oriented guide to lagrangian relaxation,” *Interfaces*, vol. 15, no. 2, pp. 10–21, 1985.
- [47] T. Liu, S. Azarm, and N. Chopra, “On decentralized optimization for a class of multisubsystem codesign problems,” *Journal of Mechanical Design*, vol. 139, no. 12, p. 121404, 2017.
- [48] J. T. Allison, T. Guo, and Z. Han, “Co-design of an active suspension using simultaneous dynamic optimization,” *Journal of Mechanical Design*, vol. 136, no. 8, 2014.
- [49] Z. Liu, J. Wu, W. Fu, P. Kabirazadeh, S. Kohtz, N. Miljkovic, Y. Li, and P. Wang, “Generative design and optimization of battery packs with active immersion cooling,” in *2023 IEEE Transportation Electrification Conference & Expo (ITEC)*, p. in press, IEEE, 2023.
- [50] M. Akbarzadeh, T. Kalogiannis, J. Jaguemont, L. Jin, H. Behi, D. Karimi, H. Beheshti, J. Van Mierlo, and M. Bercibar, “A comparative study between air cooling and liquid cooling thermal management systems for a high-energy lithium-ion battery module,” *Applied Thermal Engineering*, vol. 198, p. 117503, 2021.
- [51] B. Carter, J. Matsumoto, A. Prater, and D. Smith, “Lithium ion battery performance and charge control,” in *IECEC 96. Proceedings of the 31st Intersociety Energy Conversion Engineering Conference*, vol. 1, pp. 363–368, IEEE, 1996.
- [52] “Data For IEEE-30 Bus Test System.”
- [53] C. Zhao, J. Wang, J.-P. Watson, and Y. Guan, “Multi-stage robust unit commitment considering wind and demand response uncertainties,” *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2708–2717, 2013.
- [54] A. Monterrat, R. Christie, J. Kelly, and D. B. Murray, “Grid integration for marine renewable energy devices in a real time application,” in *12th European Wave and Tidal Energy Conference*, pp. 1–7, 2017.
- [55] “ALP No. 77 - Granby,” June 2009.
- [56] “Wind turbine power coefficient - definition and how it’s used,” 2015.