PNNL-38619

CAMEO: A Co-design Architecture for Multi-objective Energy System Optimization

Project Report

October 2025

Sumit Purohit
Rounak Meyur
Sam Donald
Tonya Martin
Thiagarajan Ramachandran

U.S. DEPARTMENT of ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-38619

CAMEO: A Co-design Architecture for Multi-objective Energy System Optimization

Project Report

October 2025

Sumit Purohit
Rounak Meyur
Sam Donald
Tonya Martin
Thiagarajan Ramachandran

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Abstract

CAMEO (Codesign Architecture for Multi-objective Energy System Optimization) is a modular workflow management framework that abstracts co-design problems as Directed Acyclic Graphs (DAG). The framework employs JSON-based workflow specifications that enable systematic decomposition of complex optimization problems into reusable, interchangeable components including data loaders, scenario generators, optimization solvers, and result summarizers.

Summary

CAMEO project has developed a modular co-design architecture for multi-scale, multi-objective optimization to enable the optimized design and operation of energy systems with high PEL penetration. The project identifies key components, requirements, dependencies, and recommendations for the co-design. The architecture is instantiated as a containerized and configurable execution platform to integrate co-design capabilities developed at PNNL and externally.

Acknowledgments

The research was supported by the Energy System Co-Design with Multiple Objectives and Power Electronics (E-COMP) Initiative, under the Laboratory Directed Research and Development (LDRD) Program at Pacific Northwest National Laboratory (PNNL). PNNL is a multi-program national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract No. DE-AC05-76RL01830.

Acronyms and Abbreviations

CPS: Cyber Physical System

DER: Distributed Energy Resources

PEL: Power Electronic Load

CCD: Control Co-design

Contents

Figures

Tables

1.0 Introduction

Complex engineering system optimization increasingly requires sophisticated computational workflows that can orchestrate multiple interdependent components, handle diverse optimization paradigms, and scale across distributed computing environments. Modern power grid infrastructure exemplifies this challenge, as new technologies such as data centers with power demands exceeding 100 MW, utility-scale battery storage installations, and distributed energy resources are fundamentally reshaping energy generation, transmission, and distribution systems. The integration of these technologies introduces complex interdependencies that system planners must simultaneously optimize to ensure reliable power delivery, making them integral to contemporary grid planning. Effective grid planning must simultaneously address multiple objectives including infrastructure costs, reliability requirements, and economic performance across various operational scenarios such as peak demand periods, contingency conditions, and market participation strategies (Sharma et al., 2024). A control co-design approach to optimization is therefore crucial, as it simultaneously considers multiple objectives to maximize system performance and ensure efficient operation, while accounting for underlying control system dynamics and interdependencies between grid components.

Over the past few decades, Control co-design (CCD) has been extensively used in different science and technology domains such as underwater autonomous robotics (Yuh, Junku, 2000), networked control systems, chemical process plants (Bhattacharya et al. 2021), automotive suspensions, wind turbines (Sharma et al., 2024), and cyber-physical energy systems (Thiagarajan et al., 2023).

This project report presents CAMEO (Co-design Architecture for Multi-objective Energy System Optimization), a modular workflow management framework that addresses key software engineering challenges in CCD optimization. In contrast to existing use-case oriented solutions, CAMEO identifies common design patterns and computational components of multi-objective co-design problems and develops generalized, plug-and-play software architecture. Our proposed approach provides a robust modular workflow framework with standardized interfaces that collectively support dynamic and customizable co-design workflows across diverse optimization paradigms and computational environments. Architecture tackles key software challenges in *scalability*, enabling workflows to manage increased computational complexity without disproportionate resource expenditure, *heterogeneity* by supporting diverse computational components and optimization methods within unified framework, and *usability* through standardized workflow specifications and automated execution management. We demonstrate CAMEO's versatility through three comprehensive use cases spanning power grid expansion planning, battery size optimization, and large-scale distribution network generation, showcasing the framework's ability to handle varying computational scales and optimization paradigms.

CAMEO is a framework to aid researchers and system planners in running optimization formulations developed a priori, over a large design parameter space. The contributions of CAMEO can be summarized as follows: (i) Scalable framework for design-space exploration: We develop a computational framework that explores wide design (hyper)-parameter spaces in high-performance computing environments for given optimization formulations. Using CAMEO, we systematically identify combinations of input parameter configurations and orchestrate parallel execution of optimization problems using available computing resources, enabling comprehensive design space exploration at scale. (ii) Modular approach for enhanced heterogeneity: We employ a modular software architecture that facilitates multiple workflow instances, allowing optimization formulations to be easily adapted and reconfigured at run-time for given co-design problems. Through CAMEO, we provide standardized interfaces and plug-and play capabilities that enable users to tailor the framework to specific needs and objectives, promoting broad applicability across different system optimization scenarios and computational paradigms. (iii) Containerized architecture for enhanced usability: We implement an automated workflow orchestration system using dedicated containerized environments, offering a lightweight and portable solution for deployment and execution across diverse computing infrastructures. We leverage containerized architecture to provide consistent performance and simplified dependency management across environments, with lower overhead than traditional virtual machine approaches. This enables quick setup, scalable operations, and efficient resource utilization, making our framework highly customizable and user-friendly for diverse co-design applications.

2.0 CAMEO Architecture and Methodology

CAMEO is a use-case agnostic co-design framework that provides scalable and modular ways to explore the design-space of an optimization formulation. The framework executes multiple instances of the formulation in parallel to provide insight into the accuracy, sensitivity, and computing performance of the formulation in a High-Performance Computing (HPC) environment.

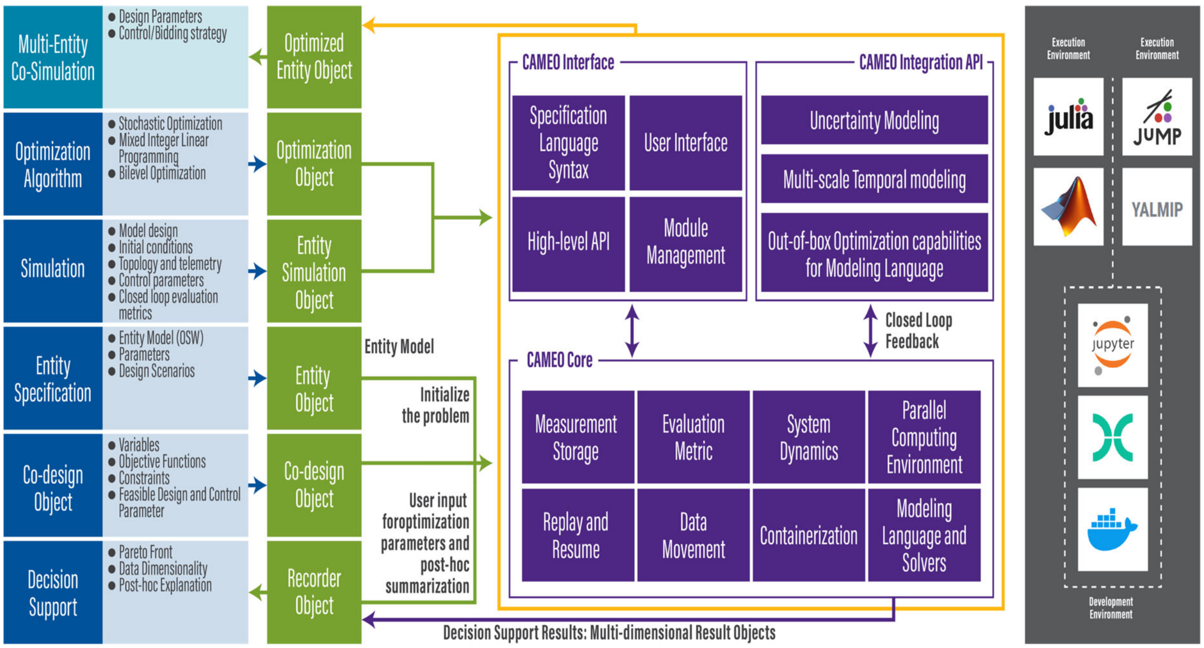

Figure 1 CAMEO Architecture: Modular Design to support multiple instances of co-design workflows

As shown in Fig. 1, CAMEO uses a modular approach to define various components and their behavior. For example, an entity object is used to specify design parameters and hyper-parameters. Similarly, the optimization object links the entity object to an algorithm and corresponding solver with seed parameters. Furthermore, a simulation object defines the underlying energy system topology, telemetry, control parameters, and evaluation metrics. CAMEO also monitors the execution environment and generates provenance summaries to share with downstream decision support applications such as pareto-front visualizations and dashboards.

2.1 Workflow Characterization and Specification

CAMEO characterizes co-design workflows as Directed Acyclic Graphs (DAGs) where stages represent distinct computational components and edges define data dependencies between them. This graph-based representation enables systematic decomposition of complex co-design problems into manageable, interconnected stages that execute in dependency order while maintaining data flow integrity throughout the process. Figure 2 illustrates a typical DAG structured co-design workflow.

The workflow specification encompasses several distinct categories of stages, each serving specific roles in the co-design process. *Data loading stages* handle the ingestion of scenario definitions, system parameters, design constraints, and operational data that characterize the co-design problem space. These stages establish the foundational inputs required for subsequent analysis and optimization. *Optimization stages* constitute the core computational components where co-design problems are formulated and solved using various paradigms including white-box mathematical optimization, heuristic-based approaches, or black-box simulation coupled with Bayesian optimization techniques. The modular nature of these stages enables direct comparison of different optimization methodologies and optimization parameters within the same workflow framework. Finally, *analysis and summarization stages* aggregate and synthesize results across multiple scenarios, performing comparative analysis and generating insights that inform design decisions.

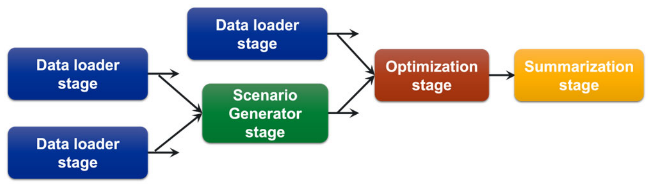

Figure 2 The DAG representation of a co-design workflow showing data dependencies between computational stages.

CAMEO's workflow characterization has several key architectural principles that facilitates its applicability to diverse co-design problems. *Standardized input-output interfaces* across all stages ensure seamless integration and compatibility, enabling components to exchange data regardless of their underlying implementation or the specific co-design domain. The *modular plug-and-play capability* allows stages to be dynamically reconfigured, replaced, or extended without disrupting the overall workflow structure, providing the flexibility needed to adapt to evolving problem requirements or incorporate new methodologies. *Multiple optimization paradigm support* enables the framework to accommodate various solution approaches within a unified computational

The DAG structure accommodates multi-modal and multi-temporal data flows between components, supporting diverse information types such as model parameters, control variables, design specifications, operational data, constraints, objective functions, and system topologies. Dependencies between stages are explicitly defined, ensuring that each component receives the appropriate inputs from its predecessors while maintaining computational efficiency through parallelization where possible. This dependency management also facilitates systematic exploration of design spaces across multiple scenarios and operating conditions.
environment, from traditional mathematical programming to state-of-the-art AI/ML-based techniques. This characterization promotes a structured yet flexible approach to co-design workflow creation, systematically addressing the complex and heterogeneous nature of multi-disciplinary design optimization while maintaining computational tractability and result reproducibility.

2.2 JSON-based declarative abstraction

CAMEO uses a JSON-based hierarchical serialization format for encoding and persisting various design variables, making it easier to manage and modify input data. This abstraction layer provides a clean separation between workflow definition and execution, allowing users to define complex computational pipelines without directly writing pipeline code. At its core, the framework organizes computational workflows as DAGs through two primary components: (i) *stages* are individual processing units that perform specific tasks, and (ii) *dependencies* are directed relationships between these stages that define their execution order. Together, these components form a DAG that represents the workflow.

During initialization, CAMEO validates the graph structure to ensure the absence of circular dependencies and confirms that all referenced stages exist, providing explicit error messages when validation fails.

The stages in CAMEO represent diverse computational tasks and data operations, each with a specific type designation. Common stage types include ‘csvloader’ for importing tabular data from CSV files, ‘pathloader’ for accessing script files and data directories, and ‘process’ for executing computational tasks through shell commands or scripts. Each stage definition includes structured specifications for inputs (data required for processing) and outputs (results produced after execution). This input-output schema forms the basis for data flow throughout the workflow, with strict validation ensuring that inputs for any stage are available in the outputs of its parent stages in the dependency graph.

```
{
   "stages": [
      "stage":{...}, "dependency1":{...},
      "dependency2":{...}],
   "dependencies": {
      "stage": ["dependency1", "dependency2"]
   }
}
```

CAMEO uses Nextflow, a scientific workflow management system to demonstrate implementation and execute workflows defined by the JSON abstraction. After evaluating multiple workflow management systems (Table 1), Nextflow emerged as the optimal choice due to its core strengths that align with our requirements: multi-platform container support, diverse executor compatibility (SLURM, Moab, Kubernetes, cloud platforms), and efficient checkpoint-based process management. The framework is therefore instantiated as a containerized and configurable execution platform with relevant technology stack, standardized interfaces, optimal data formats, and validation schema. The framework’s language-agnostic architecture enables rapid prototyping while maintaining workflow integrity. Nextflow’s unified developmentto-production pipeline, coupled with comprehensive provenance tracking and execution reporting, effectively addresses our complex computational demands and deployment scenarios.

Table 1 Comparison of different scientific workflow systems

	Language	Parallelization	Flow Control	Containers Supported	Cloud Platforms
Nextflow	groovy, java	Configurable with automatic retries	Workflow definition files and variables, command line parameter settings.	docker, podman, singularity	AWS, Azure, Google Cloud, Kubernetes
Snakemake	python	Configurable with specific retires	File inputs and outputs, command line CPU settings	docker through singularity	Kubernetes
airflow	python	Configurable at task level scheduler	Directed Acyclic Graph (DAG) based pipeline defined in python script	docker, singularity	AWS, Azure, Google Cloud, Kubernetes

2.3 Computational Infrastructure to Support Scalable Co-design

Supporting modular and scalable co-design workflow systems requires robust computational infrastructure that can handle varying workloads and resource demands across distributed computing environments. CAMEO addresses these requirements through its abstraction layer that seamlessly interfaces with high-performance computing (HPC) clusters, enabling automatic scaling and resource management without requiring users to manage low-level infrastructure details.

We leverage Nextflow's configuration system to configure parallel SLURM jobs across multiple compute nodes, utilizing a declarative approach that separates workflow logic from infrastructure specifications. The configuration structure demonstrates how CAMEO abstracts computational requirements:

```
{
  "apptainer": {
    "autoMounts": true,
    "enabled": true,
    "runOptions": "<container runtime options>",
    "cacheDir": "<container cache directory path>"
  },
  "process": {
    "executor": "<cluster job scheduler type>",
    "queue": "<target job queue name>",
    "time": "<maximum execution time limit>",
    "cpus": "<number of CPUs>",
    "nodes": "<number of nodes>",
    "allocation": "<allocation name>",
  }
}
```

The containerization layer through Apptainer ensures reproducible execution environments across heterogeneous compute nodes, while the autoMounts capability provides seamless access to shared file systems and user directories. The SLURM integration enables dynamic resource allocation where clusterOptions specify computational requirements including CPU cores, memory, node count, and project accounting information. This infrastructure abstraction allows CAMEO workflows to automatically scale from single-node development environments to large-scale HPC deployments, accommodating the computational demands of complex co-design optimization problems involving thousands of parallel processes. The separation of workflow definition from infrastructure configuration enables portability across different computing environments while maintaining consistent execution semantics and performance characteristics.

3.0 Use Cases

The project selected three power grid planning use cases that demonstrate CAMEO's capability to implement co-design optimization workflows over extensive parameter spaces. We present three cases that encompass: (1) reliability-aware power grid expansion planning for data center integration, (2) optimal battery design for variable generation installations, and (3) power distribution network generation for the entire state of Virginia, USA. Each use case showcases different aspects of CAMEO's workflow orchestration capabilities, from scenario-based optimization and stochastic programming to large-scale parallel execution on high-performance computing platforms.

3.1 Reliability-Aware Power Grid Expansion Planning for Data Center Integration

In this use case, we address the challenge of determining optimal transmission grid expansion strategies to reliably support largescale data center installations with power demands ranging between 100 and 200 MW. We formulate a multi-objective co-design optimization that considers multiple location scenarios (urban edge, industrial park, rural tech hub) and data center operation profiles (standard, high-redundancy, AI/ML focused) to minimize total expansion costs while ensuring N-1 reliability standards. We utilize CAMEO to orchestrate the evaluation of transmission

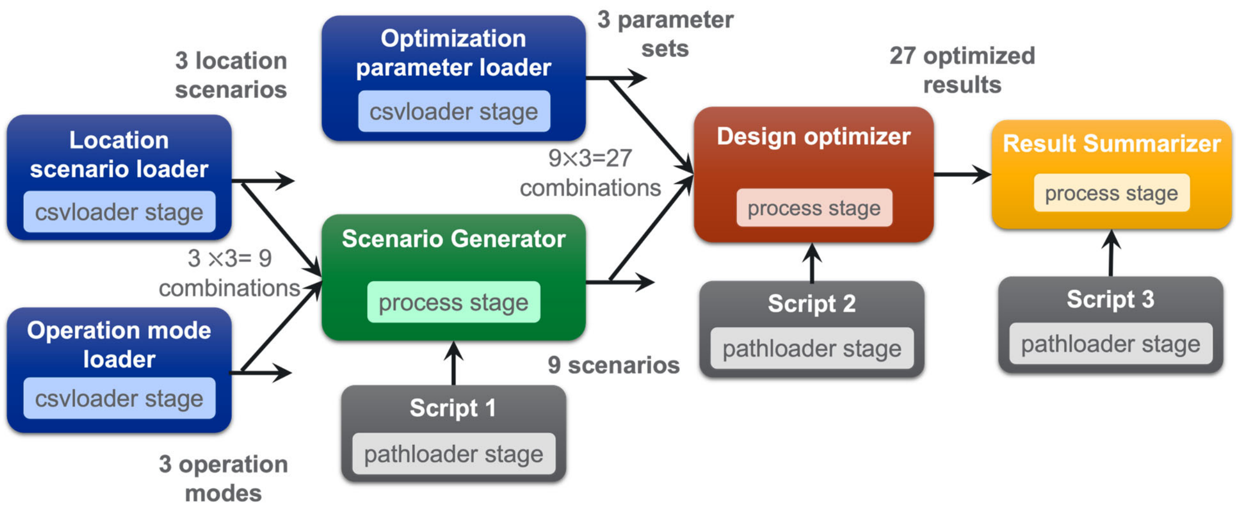

Figure 3 The DAG workflow used to perform the power grid expansion planning problem for data center integration.

line upgrades and capacity additions across these diverse scenarios, enabling grid planners to identify cost-effective expansion plans that maintain system reliability under various operating conditions and future uncertainties.

3.2 Optimal Battery Design for Variable Generation Installations

In this use case, we focus on determining the optimal battery storage size and MTDC cable capacity required to support variable generation installations while maximizing revenue from participation in real-time, day-ahead, and reserve electricity markets, accounting for installation costs. The optimization formulation is inspired by the scenario tree-based stochastic programming approach for revenue maximization proposed in [14]. We have extended the methodology to demonstrate CAMEO's capability in handling multiple formulations within the stochastic programming paradigm. We demonstrate two distinct optimization formulations: (i) a scenario based approach (illustrated through the DAG workflow in Figure 4) where we generate 20 random scenarios per generation location based on generation variability and market data, solving 32 battery configurations across 5 locations (totaling 3,200 optimization problems), and (ii) a

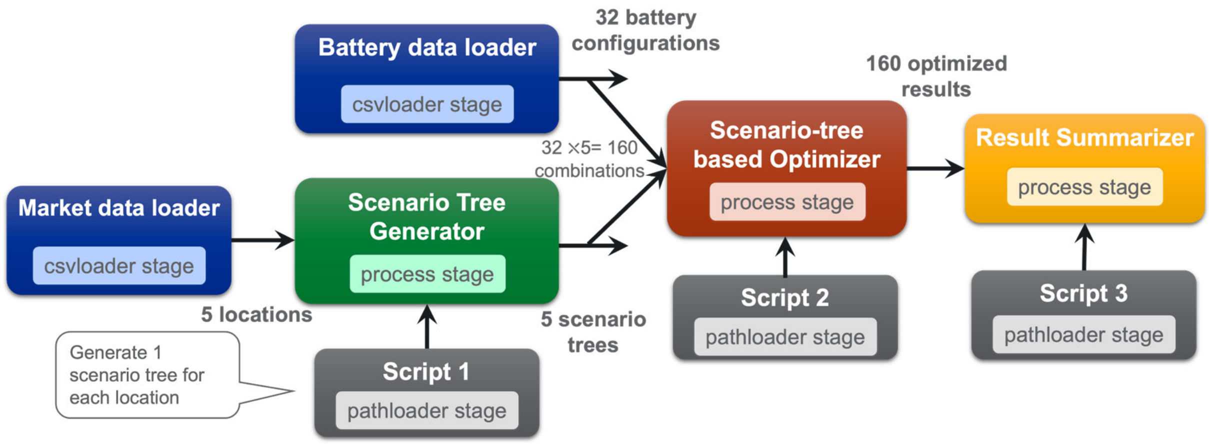

Figure 4 The DAG workflow for scenario tree-based optimization formulation for the battery design problem.

stochastic programming approach where we use scenario trees for each location with the same battery configurations (resulting in 160 stochastic optimization problems solved in parallel).

3.3 Power Distribution Network Generation for Virginia, USA

In this use case, we demonstrate CAMEO's scalability by showcasing secondary network generation for multiple counties across Virginia, highlighting the framework's ability to execute computationally intensive tasks in parallel across multiple cores on high-performance computing platforms. We execute simultaneous network generation processes for different counties, demonstrating CAMEO's capability to handle large-scale, geographically distributed optimization problems efficiently.

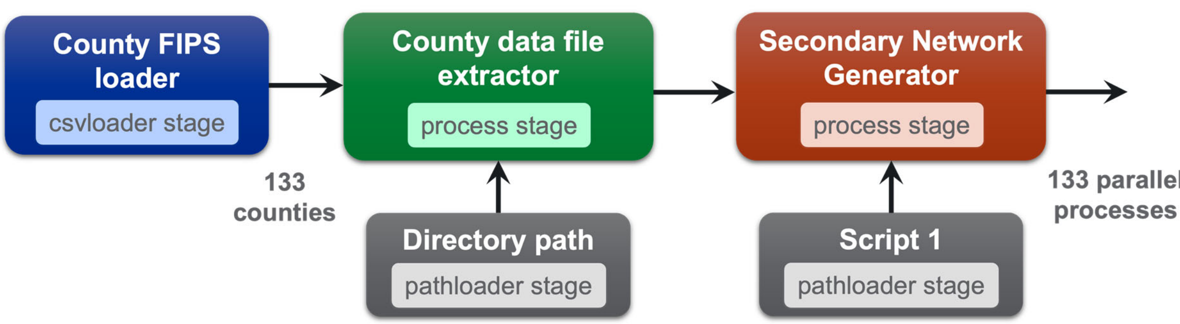

Figure 5 The DAG workflow for secondary distribution network generation for 133 counties in Virginia

4.0 Results and Discussion

To demonstrate the practical benefits of CAMEO's JSON abstraction, we evaluate three key aspects that directly impact productivity and workflow management in energy system co-design tasks: (i) modularity enabling seamless adaptation across multiple problem scenarios, (ii) deployability facilitating execution across diverse computing environments, and (iii) development efficiency streamlining workflow creation and maintenance processes.

These metrics provide evidence of the framework's usability advantages in handling varied optimization problems, supporting multi-environment execution, and reducing development overhead compared to traditional workflow implementation approaches. We analyze component reusability across our three use cases to demonstrate the modular design benefits of the JSON abstraction. Our analysis reveals significant component overlap: data loader and script loader stages achieve 100% reusability across all use cases. The optimization stages show no such reusability, since the nature of optimization formulations are different for our three use cases. The plug-and- play capability is exemplified in our second use case, where we seamlessly swap between scenario-based and stochastic optimization formulations by modifying only handful of lines in the JSON abstraction – specifically changing the optimization stage reference while defining the DAG workflow in the JSON abstraction.

Cross-domain portability analysis shows that majority of workflow components remain identical when adapting workflows from power grid expansion to battery size optimization to distribution network generation. The core workflow pattern of data loading → scenario generation → optimization → summarization remains consistent, with only domain-specific parameters and script references requiring modification.

4.1 Deployable Across Multiple Compute Environments

CAMEO can be used to execute identical optimization workflows on different computational platforms without modification to the underlying workflow logic. We illustrated this cross-platform capability by showing execution duration and CPU usage performance when running identical data center integration optimization problems across three different computing environments: (i) multiple nodes on HPC resources, (ii) single node on HPC resources, and (iii) single node on a virtual machine. This seamless deployability enables users to leverage available computational resources optimally while maintaining workflow consistency and reproducibility across different infrastructure configurations.

The performance analysis reveals the compute platform-agnostic design of CAMEO, which provides actionable insights for strategic resource allocation across different computing environments. For example, in our data center integration use case, 'rural tech hub' scenarios consistently show the fastest execution times (60-100 seconds) across all computing platforms due to smaller grid sizes, while 'industrial park' scenarios exhibit longer execution times (400-500 seconds on virtual machines) due to more complex optimization problems. The ability of CAMEO framework to deploy identically across multiple nodes on HPC resources, single HPC nodes, and virtual machines demonstrates superior performance adaptability, with HPC resources achieving higher CPU utilization rates (100-120%) compared to virtual machines (80-90%).

5.0 AI-enabled optimization and workflow generation

CAMEO project also explored state-of-the-art artificial intelligence and large language model (LLM)-based approaches in optimization and complex workflow generation applications. The project set up LLM-based experiments to auto-select priors in a Bayesian optimization formulation. Traditional methods use random sampling or space-filling designs, which do not leverage problem-specific priors. We used LLMs as the initial sampling function and are tasked with generating initial sample points conditioned by varying levels of context related to the optimization problem. We defined a parameterized template that configures an LLM agent to iteratively converge on prior values for the Bayesian optimization. Such approach was also extended to perform surrogate modeling and acquisition function generation tasks.

> The following are examples of the performance of a {model} measured in {metric} and the corresponding model hyperparameter configurations. The model is evaluated on a tabular {task} task containing {number of classes} classes. The tabular dataset contains {number of samples} samples and {number of features} features ({number of categorical features} categorical, {number of continuous features} numerical). The allowable ranges for the hyperparameters are: {configuration and type}. Recommend a configuration that can achieve the target performance of {target score}. Do not recommend values at the minimum or maximum of allowable range, do not recommend rounded values. Recommend values with the highest possible precision, as requested by the allowed ranges. Your response must only contain the predicted configuration, in the format ## configuration ##.
> Performance: {performance 1}
> Hyperparameter configuration: configuration 1
> ...
> Performance: {performance n}
> Hyperparameter configuration: {configuration n}
> Performance: {performance used to sample configuration}
> Hyperparameter configuration:

Figure 6 Candidate Sample Prompt

We also developed agentic LLM-application to auto-generate abstractions and workflow definitions for the optimization formulations. The goal of this task was to analyze a code repository, in addition to user provided natural language description of the codebase. We developed a web-based prototype to identify optimization stages as defined in section 2 and recommend a nextflow workflow for user review and validation as shown in the figure below. This prototype also allows

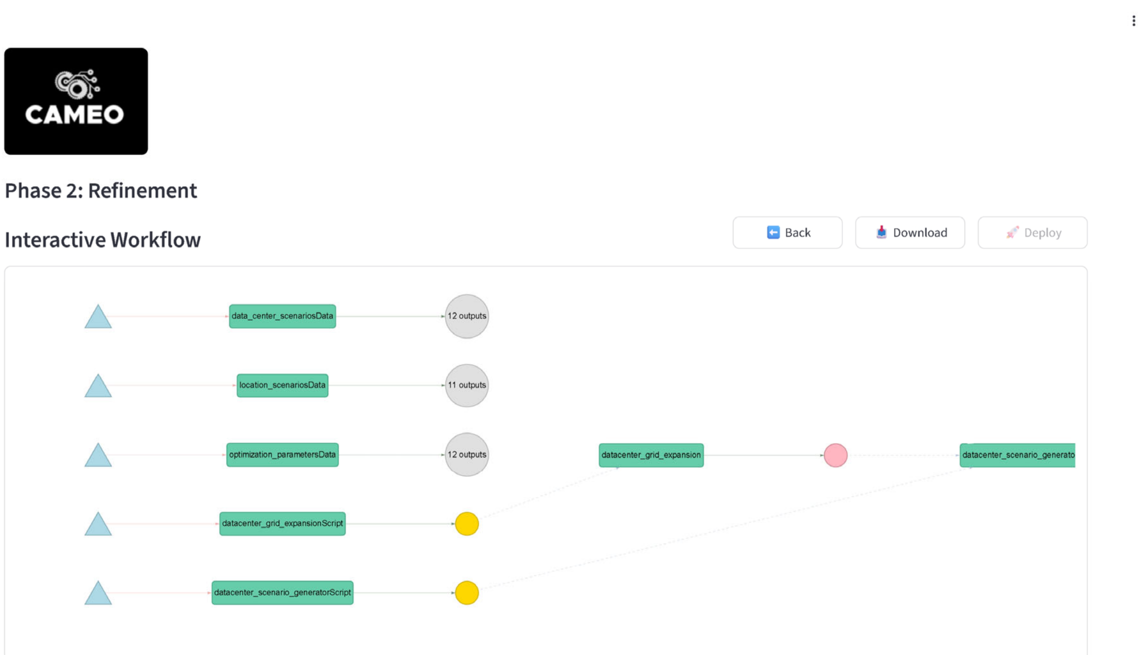

Figure 7 LLM-based workflow generation

you to edit the workflow by changing stage or parameter definition. The resulting workflow can be executed in a containerized environment.

6.0 Conclusion

CAMEO's use case agnostic design enables application across diverse optimization domains, as demonstrated through three distinct use cases spanning power grid expansion planning, battery size optimization, and large-scale distribution network generation, each requiring different computational scales and optimization paradigms.

CAMEO's JSON-based workflow abstraction and standardized component interfaces provide a foundation for systematic decomposition of complex optimization problems into reusable, interchangeable modules. The framework's containerized execution environment and seamless integration with HPC infrastructure demonstrate its capability to handle varying computational demands while maintaining workflow portability across different computing environments.

This work has several limitations that present opportunities for future research. We lack a comprehensive taxonomy relating to energy system workflow patterns to generic computational motifs, which would enhance cross-domain applicability. Our evaluation excludes systematic comparison with other declarative workflow systems such as CromWell (WDL), StreamFlow, and Toil (CWL), limiting context for CAMEO's relative positioning. Additionally, we demonstrate technical capabilities without comprehensive user surveys to assess real-world usability and adoption barriers.

Future development will address these limitations while expanding CAMEO's capabilities through an extensible library of optimization modules, formal workflow pattern taxonomies, and comparative benchmarking studies with existing workflow systems. We plan to implement graphical user interfaces for enhanced workflow specification, enable seamless interfacing with cloud platforms and specialized optimization solvers, and conduct comprehensive user studies to validate practical applicability. Furthermore, we will investigate generating JSON stage definitions through human-in-the loop Large Language Model methods to further abstract co-design formulation from low-level code development. These enhancements will strengthen CAMEO's position as a general-purpose computational framework for complex system optimization across diverse engineering disciplines.

7.0 References

Sharma, Himanshu, Wei Wang, Bowen Huang, Thiagarajan Ramachandran, and Veronica Adetola. "Multi-objective control co-design using graph based optimization for offshore wind farm grid integration." In 2024 American Control Conference (ACC), pp. 2580-2585. IEEE, 2024.

Yuh, Junku. "Design and control of autonomous underwater robots: A survey." *Autonomous Robots* 8, no. 1 (2000): 7-24.

Ramachandran, Thiagarajan, Soumya Vasisht, Aowabin Rahman, Arnab Bhattacharya, and Veronica Adetola. "A computational framework for control co-design of resilient cyber–physical systems with applications to microgrids." IEEE Transactions on Control Systems Technology 32, no. 3 (2023): 793-804.

Bhattacharya, Arnab, Soumya Vasisht, Veronica Adetola, Sen Huang, Himanshu Sharma, and Draguna L. Vrabie. "Control co-design of commercial building chiller plant using Bayesian o

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov