Proceedings of the 57th Hawaii International Conference on System Sciences | 2024

Data-Driven Unit Commitment Refinement - a Scalable Approach for Complex Modern Power Grids

Timothy Holt
Università della Svizzera italiana,
Oak Ridge National Laboratory
timothy.holt@usi.ch

Shrirang Abhyankar
Pacific Northwest
National Laboratory
shrirang.abhyankar@pnnl.gov

Teja Kuruganti
Oak Ridge National Laboratory
kurugantipv@ornl.gov

Olaf Schenk
Università della Svizzera italiana
olaf.schenk@usi.ch

Slaven Peles
Oak Ridge National Laboratory
peless@ornl.gov

Abstract

Integration of renewable generation, which is often intermittent and decentralized, substantially increases the stochasticity and complexity of power grid operations. Future power systems planning will require significant computational capability to evaluate balance between demand and supply under varying conditions, both temporally and spatially. The standard approach for generation unit commitment is to use mixed-integer linear programming to find the optimal generation schedule considering ramping and generator constraints. In the future grid this poses computational scalability challenges because generation and demand are not known with certainty due to stochasticity in weather and complexity of the grid. To address this challenge, we present a data-driven unit commitment approach that can efficiently include stochastic weather impacts and contingency considerations to improve unit commitment. Our approach uses graph-based data analytics techniques on solutions to the security constrained (and possibly stochastic) economic dispatch problem to identify potential improvements to a given unit commitment. Recent breakthroughs in fully-parallel stochastic economic dispatch software allow this approach to be scalably deployed. Simulations on synthetic South Carolina and Texas grids show this method can improve grid reliability with security constraints over a set of contingencies, while also meaningfully lowering total generation cost.

Keywords: Unit Commitment Scheduling, Economic Dispatch, Renewable Energy, Stochastic Grid Analysis

This research is part of the activities of the Innosuisse project no 34394.1 entitled “High-Performance Data Analytics Framework for Power Markets Simulation” and the Swiss Centre for Competence in Energy Research on the Future Swiss Electrical Infrastructure (SCCER-FURIES), which is financially supported by the Swiss Innovation Agency (Innosuisse - SCCER program).

1. Introduction

Transformational decarbonization of the electricity generation fleet is needed to achieve emissions reductions targets [25]. This will require a transformation from centralized dispatchable generation towards distributed renewable generation. Inherent variability in generation coupled with impacts of stochastic weather scenarios requires a new class of scalable engineering tools to maintain the reliability requirements of the grid while maximizing utilization of installed renewable generation. In particular, developing grid planning and management approaches that include stochastic factors and contingencies into generation dispatch decisions, is a key requirement for successful renewables integration [24]. These factors are normally considered downstream of the unit commitment (UC) computations in the economic dispatch process for computational reasons [14]. The standard procedure of conducting economic dispatch computations over a set of contingencies and weather scenarios generates data that is imbued with information about contingencies and weather. We propose a method to efficiently “recycle” this data to refine UC, thus including contingencies and weather information into the UC. This data-driven UC refinement is a minimal intervention approach, simply adding an extra step in standard workflows to update UC after economic dispatch.

Our novel data-driven unit commitment approach is enabled by ExaGO [2], a recent development in security constrained alternating current optimal power flow (SC ACOPF) software that introduces massively parallel computation of contingency and scenario analysis, greatly lowering the time-to-solution of such analysis. This means we can generate massive amounts of data in the economic dispatch step at low-cost that can be used to refine UC in an iterative manner. This integrated UC and economic dispatch approach leverages today’s vastly expanded computational capabilities in a scalable way, showing strong potential to improve both the security and cost of power grids.

UC is a challenging computational problem [30]. Its general formulation – a constrained non-linear mixed-integer optimization problem – is an NP-hard problem that is difficult to solve efficiently even when using parallel computations [13]. Typically, model approximations are made to make the UC problem easier to solve. Most commonly, the model is linearized so that mixed-integer *linear* programming can be used, which can be implemented more efficiently [18]. The weaknesses of this approach are: 1) it is difficult to integrate consideration of contingencies or stochastic scenarios into these solutions [4, 3, 12]; 2) the linear approximations of the UC model do not capture voltage or reactive power constraints, leading to solutions that are only approximations of the fully constrained problem [22]; and 3) the combinatorial nature of the problem formulation limits the scalability and solution accuracy achievable in a reasonable time-frame for large grids [6]. This standard UC approach could thus be improved by injecting information on contingencies and non-linear security constraints in a way that is scalable so that it can be applied to large grids.

Contingency analysis including all security constraints is typically done after UC through a series of alternating current power flow (ACPF) forward simulations [13]. Methods to include contingencies and stochastic information into UC have been studied and proposed, with [3, 4, 14] providing summaries of this research. This is a fundamentally challenging problem since it is large-scale, non-linear, non-convex, and combinatorial, while solutions are expected in short time-frames. Stochastic non-linear mixed-integer programming, that would provide a native way to solve such problems, is still an open research problem requiring more investigation.

On the other hand, there has been significant progress in the development of computational methods for security constrained and stochastic economic dispatch [21, 26, 16, 17]. Robust, parallelizable methods for stochastic SC ACOPF have been developed and successfully tested on different grid models [1, 20, 28]. These methods are stable with well understood complexity and convergence properties; allowing for scheduling analyses that run within strict time constraints. Furthermore, these analyses can model and *strictly enforce* all security constraints without making approximations. Finally, these methods run efficiently on inexpensive hardware, process a large number of contingencies in a relatively short time, and generate large amounts of high-quality data.

Successful development of methods for security constrained economic dispatch creates new opportunities to exploit the big data contained in the economic dispatch solutions to improve UC. In this paper, we propose an integrated security constrained unit commitment and economic dispatch approach that examines data generated in SC ACOPF analysis to suggest refinements to UC. Simulations show that these refinements have potential to not only increase grid security across large sets of contingencies, but also reduce total generation cost. We call this approach data-driven unit commitment (DDUC). To keep the presentation streamlined, we consider only day-ahead reliability UC in a deregulated region.

The main contribution of this paper is the introduction of a scalable grid planning technique that exploits the data generated by economic dispatch analysis with contingencies to improve UC. The approach is formalized in an algorithm and presented alongside a statistically rigorous set of simulations that give a promising proof-of-concept. The advantages of this approach are:

1. Incorporates contingency analysis and stochastic weather scenarios into computation of UC.
2. Does not use mixed-integer programming for the refinements, instead relying on data analysis techniques that are more scalable for large complex grids.
3. Exploits the data generated by economic dispatch computations that must be done in any case.
4. Additional computations imposed by DDUC utilize graph-based algorithms of linear complexity, meaning they are extremely fast and efficient.

The organization of the remainder of paper is as follows: Section 2 briefly presents the novel ExaGO grid optimization software that has enabled the DDUC approach. Section 3 formalizes the DDUC algorithm. Section 4 shows the results of numerical experiments to evaluate the algorithm. Section 5 discusses potential uses cases, significance, and impact of the DDUC approach. Section 6 presents conclusions and directions for further research.

2. Exascale Grid Optimization (ExaGO) Toolkit

The Exascale Grid Optimization Toolkit (ExaGO) [1, 2] is a package for solving large-scale AC optimal power flow problems with stochastic (wind generation, load), security (generation and network contingencies), and scheduling (generator ramping) constraints. It implements scalable algorithms that allow it to run on hardware ranging form a laptop to a supercomputer. ExaGO is portable and can be deployed on traditional CPU and/or heterogeneous GPU-based architectures. It has interfaces to state-of-the-art optimization libraries HiOp [19] and Ipopt [27].

In this work, we use ExaGO's `scopflow` application to solve SC ACOPF application formulated as

$$\min \sum_{c \in \mathcal{C}} f(x_c) \tag{1}$$

$$\text{s.t. } g(x_c) = 0, \tag{2}$$

$$h(x_c) \leq 0, \tag{3}$$

$$x^- \leq x_c \leq x^+, \tag{4}$$

$$-\Delta_c x \leq x_c - x_0 \leq \Delta_c x,\ c \neq 0 \tag{5}$$

where, $\mathcal{C}$ represents the set of contingencies, including the base-case denoted by subscript 0. `scopflow` aims to minimize the objective $\sum_{c\in\mathcal{C}} f(x_c)$, while adhering to the equality $g(x_c)$, inequality $h(x_c)$, and the lower/upper bound (x^-,x^+) constraints. For notational ease we include the base-case in set $\mathcal{C}$, i.e., $\mathcal{C} \equiv \mathcal{C} \cup c_0$. Each subproblem c has the detailed formulation of an AC optimal power flow problem. Equation (5) represents the coupling between the base-case and each of the contingency states c_i. Equation (5) is the most typical form of coupling that limits the deviation of the contingency variables x_c from the base x_0 to within $\delta_c x$. An example of this constraint could be the allowed real power output deviation for the generators constrained by their ramp limit.

For the purpose of demonstrating and testing our DDUC approach, we relax the coupling constraints (5) between the base and contingency subproblems. This results in decoupling of AC optimal power flow subproblems for each contingency. Each AC optimal power subproblem is solved in parallel. In essence, this is similar to a parallel AC contingency analysis with the difference that instead of solving power flow for each contingency, we solve an AC optimal power flow. ExaGO's `scopflow` application has an in-built solver called *EMPAR* (short for "embarrassingly parallel") that can be used for such decoupled contingency analysis.

We also model load shedding for each AC optimal power subproblem where each load i can shed an up to γ_i % of its load at a given cost C_i. This load loss formulation allows setting priority or importance to loads (by setting higher costs C_i) and making provision for load that should not be curtailed $(1-\gamma_i)$, for example in the case of critical loads.

3. Data-Driven Unit Commitment Algorithm

3.1. Algorithm Objectives

Given an initial UC, the DDUC algorithm proposes updated UCs after observing the optimal results of the economic dispatch problem as solved by ExaGO. The algorithm has two objectives: 1) efficiently find a UC that reduces or eliminates the necessity for load shedding over a set of contingencies cases; and 2) exploit the solutions of economic dispatch over time, to evolve the UC to reduce the overall cost of the day-ahead UC. A UC that simultaneously realizes both of these goals would be an unambiguous improvement to grid operation. The DDUC algorithm achieves this by adding "important" generators to the UC and removing "unimportant" ones over a series of iterations.

The identification of "unimportant" generators is done using a data-driven approach that examines the solutions of the SC ACOPF problem finding those generators that have low capacity factors, that is generators whose available capacity is left mostly idle. The hypothesis of this heuristic is that if a generator provided either low-cost or critical power to the grid, it should have a high capacity factor in the optimal (lowest-cost) SC ACOPF solution. The identification of "important" generators is more nuanced. The basic idea is to find contingency cases that require load shedding to maintain security constraints, then find currently deactivated generators that are "close" to the load that was preferentially shed by the fully-constrained SC ACOPF optimization algorithm for that contingency case. The hypothesis of this heuristic is that those generators that are proximate to the areas of the grid that require load shedding will be most able to provide the missing power. Details on how proximity is measured and how proximate generators are algorithmically identified follow in Section 3.2.

The DDUC algorithm works in two phases: a load shed recourse phase where "important" generators are identified and added to the UC, and a pruning phase where "unimportant" generators are identified and removed from the UC. Algorithm 1 defines how these phases are combined to produce the new UC.

3.2. Load Shed Recourse Phase

In the load shed recourse phase, the alternating current optimal power flow (ACOPF) solutions for the base-case and each contingency are analyzed to find situations in which load shedding is required to keep from violating security constraints. In each of the load shedding contingency cases, the network is analyzed to

Algorithm 1: Data-Driven Unit Commitment

Data:
- $\mathcal{G}$: power grid model,
- $\mathcal{U}$: unit commitment for that grid,
- $\mathcal{C}$: set of contingencies for that grid,
- n : number of iterations

Result: $\mathcal{U}'$: new unit commitment

Function `DDUC` ($\mathcal{G}$, $\mathcal{U}$, $\mathcal{C}$, n) **:**
    $\mathcal{U}' \leftarrow$ `recourse` ($\mathcal{G}$,$\mathcal{U}$,$\mathcal{C}$)
    for n **do**
        $\mathcal{U}' \leftarrow$ `prune` ($\mathcal{G}$,$\mathcal{U}'$,$\mathcal{C}$)
        $\mathcal{U}' \leftarrow$ `recourse` ($\mathcal{G}$,$\mathcal{U}'$, $\mathcal{C}$)
    end
return $\mathcal{U}'$

Algorithm 2: Load Shed Recourse

Data:
- $\{\mathcal{G},\mathcal{U},\mathcal{C}\}$,
- α : activation parameter,
- k : number of generators to return from BFS

Result: $\mathcal{U}'$: new unit commitment

Function `recourse` ($\mathcal{G}$, $\mathcal{U}$, $\mathcal{C}$, α, k) **:**
    if $\mathcal{C} \supset \{c_0\}$ **then**
        $\mathcal{U} =$ `recourse` ($\mathcal{G}$, $\mathcal{U}$, $\mathcal{C} = \{c_0\}$, α, k)
    end
    Run ExaGO `scopflow` on $\mathcal{G}$,$\mathcal{U}$,$\mathcal{C}$
    $\mathcal{S} \leftarrow$ set of cases with load shedding
    $\mathcal{U}' \leftarrow \mathcal{U}$
    for $s \in \mathcal{S}$ **do**
        $\mathcal{U}_s \leftarrow \mathcal{U}$
        $\mathcal{B} \leftarrow$ buses in s with load shedding
        $\pi \leftarrow$ total load shed in s
        $r \leftarrow 0$
        while $r < \alpha\pi$ **and** $\mathcal{B} \neq \varnothing$ **do**
            $b \leftarrow$ pop highest load shed in $\mathcal{B}$
            `breadthFirstSearch` (b, k)
            $\mathcal{N} \leftarrow k$ nearest deactivated gens to b
            for $g \in \mathcal{N}$ **do**
                $\rho \leftarrow$ path capacity from g to b
                $\rho' \leftarrow$ generating capacity of g
                $p_g \leftarrow \min(\rho, \rho')$
            end
            $g \leftarrow \text{argmax}(p)$
            $\mathcal{U}_s \leftarrow \mathcal{U}_s \cup g$
            $r \leftarrow r + \max(p)$
        end
        $\mathcal{U}' \leftarrow \mathcal{U}' \cup \mathcal{U}_s$
    end
return $\mathcal{U}'$

find generators that are currently inactive that would be good candidates to add to the unit commitment. A good candidate is a generator that: 1) is close, in the graphical sense, to the bus that shed the most load; and 2) has available transmission capacity on the shortest path connecting it with that bus.

Such generators are found using a breadth first search (BFS) from the bus that sheds the most load. The search terminates when the number of inactive generators found reaches k, a tuning parameter that effectively controls the relative importance of the graphical proximity and available capacity measures. A low k will emphasize graphical closeness, while a high k will emphasize available capacity. For each generator in this set, the available capacity on the shortest transmission path between that generator and the bus with load shedding is then measured. The generator with the greatest ability to provide generation to the bus, based on both generator capacity and transmission path capacity is then added to the set of generators to activate. This process repeats for the bus with the next highest amount of load shedding until the generation capacity activated multiplied by parameter α is greater than the total amount of load shedding in the scenario. Once this process has been repeated for each contingency that had load shedding, the union of all sets of generators identified for each case is added to the UC. Full details of this load shed recourse phase are outlined in Algorithm 2.

3.3. Pruning Phase

The sole purpose of the pruning phase is to further optimize UC. The ACOPF solution for the base-case and each of the contingency cases is analyzed to find the generators with the lowest capacity factors in the network, across all cases, and then remove some of those generators from the UC. The prune algorithm proceeds by taking a sum of the capacity factor for each generator in the UC over all of the cases / scenarios and then removing the z generators with the lowest total capacity factor across all scenarios (Algorithm 3).

Tuning the parameter z, which defines how many generators to prune at each pass of the DDUC algorithm, is critical to the effectiveness of the algorithm at reducing the cost of the UC. Too low a z, and inefficient generators will remain in the UC. Too high a z, and “important” generators may be mistakenly pruned. Methods for exploring the space of z values and algorithms for reaching optimal z values is something we would like to explore in further research. For this paper we used a percentage of active generators in the range of 2-8% that decreases with every iteration.

Algorithm 3: Prune Generators

Data:
- $\{\mathcal{G}, \mathcal{U}, \mathcal{C}\}$,
- z : number of generators to prune

Result: $\mathcal{U}'$: new unit commitment

Function `prune` ($\mathcal{G}$, $\mathcal{U}$, $\mathcal{C}$, z) **:**
    Run ExaGO `scopflow` on $\mathcal{G}, \mathcal{U}, \mathcal{C}$
    $\mathcal{S} \leftarrow$ set of all cases
    $f_g \leftarrow 0, \forall g \in \mathcal{U}$
    for $s \in \mathcal{S}$ **do**
        for $g \in \mathcal{U}$ **do**
            $f_g \leftarrow f_g +$ capacity factor of g in s
        end
    end
    $\mathcal{U}' \leftarrow \mathcal{U}$
    for z **do**
        $g \leftarrow$ pop $\operatorname{argmin}(f)$
        $\mathcal{U}' \leftarrow \mathcal{U}' \setminus g$
    end
return $\mathcal{U}'$

Table 1: Properties of tested grids.

	S.C. grid	Texas grid
Buses	500	2,000
Generators	90	544
Branches	597	3,206
Installed gen. (MW)	12,189	96,292
Total load (MW)	7,751	67,109
Contingencies considered	590	500

4. Numerical Experiments

4.1. Setup

Numerical experiments to test the effectiveness of the algorithm were conducted on two test grids from the ACTIVSg series [8, 29, 9] – the synthetic South Carolina (S.C.) and Texas grids. At 500 and 2000 buses respectively, these grids are sufficiently complex to provide challenging problems, while being small enough to provide SC ACOPF solutions with ExaGO in seconds rather than minutes. For each grid, a single scenario with static load conditions was analyzed, with variability on the supply and transmission side from the contingencies considered. Basic properties of these test grids are shown in Table 1.

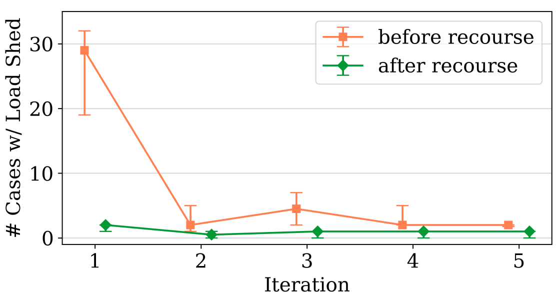
The experiments were conducted on a cluster with a 64-core AMD 3rd Gen EPYC CPU on each node. The experiments were run using the ExaGO `scopflow` EMPAR formulation using Ipopt [27] as the optimization engine and Pardiso [5, 11, 10] as the solver for linear systems. The `scopflow` EMPAR solver provides massively parallel ACOPF solutions of all contingencies in a fully de-coupled formulation. As such, one compute core is required for each contingency plus one for the base-case to achieve maximal parallel throughput. For the S.C. grid 590 contingencies were considered in all experiments, representing the full set of $n - 1$ contingencies for generators and one branch outage per bus. For the Texas grid a random sample of 500 $n - 1$ contingencies were considered including a mix of both branch and generator contingencies. The S.C. and Texas grids thus required 591 and 501 CPU cores, respectively, to solve fully parallel.

These are relatively small examples in terms of the number of contingencies and scenarios, however, this method is also applicable to extreme-scale analyses. ExaGO has been successfully run on Frontier, the world's most powerful and first exascale supercomputer [23], to perform analysis on the 10,000-bus synthetic Western Interconnection model, using 9,999 contingencies and 10 stochastic scenarios, resulting in nearly 100,000 sub-problems. This problem was run on 9,000 compute nodes with 72,000 MPI ranks and successfully completed in 16 minutes. To integrate such runs with the DDUC algorithm would require further research and development, but the linear algorithmic complexity of the data analysis methods make integration with extreme-scale problems feasible.

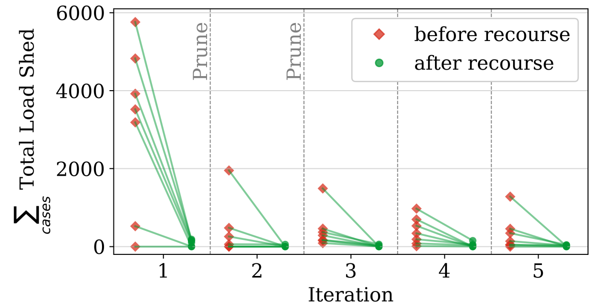
A statistical approach was used to evaluate the performance of the DDUC algorithm given many distinct starting UCs. Specifically, performance was evaluated for 100 stochastically generated starting UCs for each grid. The starting UCs were made by randomly deactivating x percent of the generators in the grid, then adding back generators until the base-case scenario was feasible without load-shedding; $x = \{40\%$ for S.C., 30% for Texas$\}$. This statistical approach provides assurance that the positive results are representative of the performance of the algorithm on a broad range of starting UCs and not a circumstance of a specific case. The DDUC algorithm was evaluated using 5 iterations (5 cycles of load shed recourse and pruning), as it was empirically most effective in the first 5 iterations.

4.2. Performance Evaluation Criteria

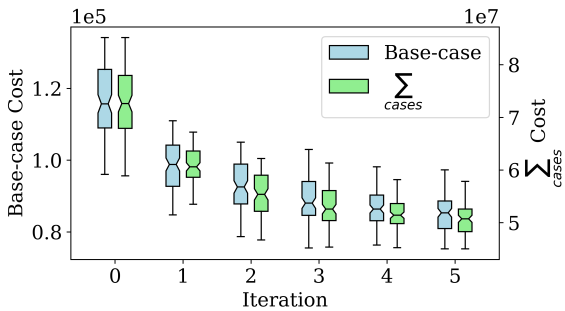
We evaluate the DDUC algorithm for its performance on two metrics: 1) reduction of the amount of load shedding from contingencies, and 2) improvement of the overall cost of running the grid with the suggested UC. The reduction of load shedding



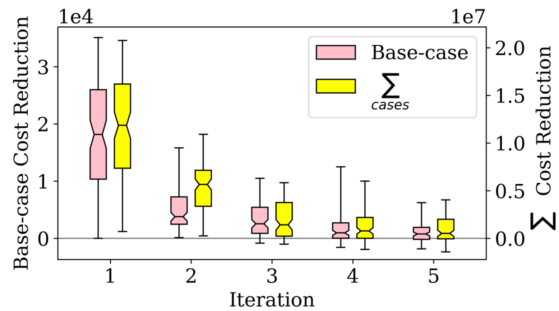
(a) S.C. grid number of scenarios with load shedding (n=100)



(b) S.C. grid total load shed reduction from recourse (n=7)



(c) S.C. grid absolute cost function value (n=100)



(d) S.C. grid iteration-over-iteration cost reduction (n=100)

Figure 1: DDUC algorithm performance on South Carolina grid. Figure 1a - recourse algorithm reduction in number of load shed cases across all contingencies; error bars show middle quartiles. Figure 1b - recourse algorithm reduction of load shed (MW) for a random sample of 7 runs. Figure 1c - value of the cost function ($) of the base-case and the sum of all cost function values across all contingency cases. Figure 1d - cost reduction ($) achieved on an iteration-over-iteration basis for each of 5 iterations. Box plot notches represent median, box represents middle quartiles, whiskers represent 5th to 95th percentiles.

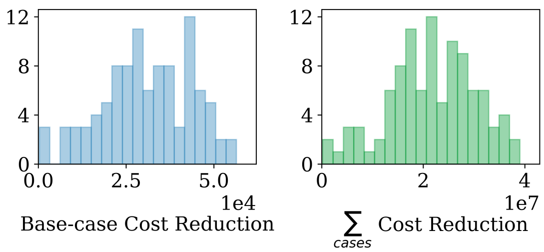
given a set of contingencies is the more important of these two metrics, since loss of power for consumers is a highly undesirable event. Any approach to find a UC to reduce load shedding under contingencies, however, should be evaluated not only for its benefits of reducing load shedding, but also for its impact on the cost of operating the grid. A UC that reduces load shedding in the contingencies may be undesirable if it significantly increases costs. The following results show that the DDUC algorithm tends to find UCs that reduce both load shedding and cost, giving us a win-win. However, the cost savings results are intrinsically sensitive to the particular cost model employed. Our model includes a cost attached to load shedding, so there is a benefit in the cost metric for preventing load shedding.

4.3. South Carolina Grid Results

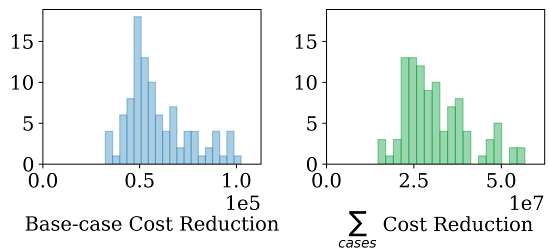
Upon testing the DDUC algorithm, we are pleased to see impressive performance on both metrics. Figure 1 visualizes the results for testing on the S.C. grid. The top panel shows performance against the objective of eliminating load shedding, while the lower panel shows performance at lowering overall cost.

Figure 1a shows the consistent ability of the recourse algorithm to find UCs that eliminate the necessity of load shedding from contingency scenarios. This figure shows that arbitrary starting UCs typically must shed load in 20-30 different contingency cases. After the recourse algorithm is run on this starting UC, however, the number of contingency cases requiring load shedding drops considerably to the range of 2-3. Subsequent iterations of the DDUC algorithm do not dramatically increase load shedding, and it is interesting to note that for this case, the recourse algorithm has difficulty eliminating the necessity for load shedding from all contingency scenarios, typically having 1 contingency that still requires it. This suggests that the resilience of S.C. grid (as described in the model) could be improved with suitable upgrades.

Figure 1b shows a slightly different view of the effectiveness of the recourse algorithm. On the y-axis, the sum of load shedding across all contingency cases ($\sum_{cases}$) for a particular UC is indicated by red



(a) South Carolina grid



(b) Texas grid

Figure 2: Histogram - cost function reduction from initial UC to UC assigned after 5 iterations of DDUC algorithm for n=100 runs, each seeded with different stochastic starting UC.

diamonds before the recourse algorithm and green dots after the recourse algorithm. Only 7 randomly selected starting UCs are shown so that the plot is not too crowded. This figure shows that while there may be significant load shedding required to maintain security constraints across the set of contingency scenarios before the recourse algorithm, after the algorithm, the amount of load shedding required across all contingencies is most often 0 or nearly 0.

Figure 1c shows performance of the DDUC algorithm at reducing the overall cost of the network. The different colors of boxes represent the cost function value for the base-case, and the sum of the cost functions of all contingency cases including the base-case. An impressive improvement in cost is shown, with the median cost of a starting UC at \$131.3k compared to the median cost of \$85.3k after 5 iterations of the UC algorithm; an impressive improvement of 26.2%.

Figure 1d shows The cost reduction on an iteration-over-iteration basis of the given UCs. What this means is that the metric measures the improvement for each individual UC over the five iterations of the algorithm, yielding a positive value if the cost decreased during the iteration and a negative value if the cost increased during the iteration. These measures are then aggregated in the box plot. We see that for each of the five iterations, the algorithm resulted in a cost reduction to the system in the vast majority of instances. Notably, in both the first and second iteration, at least 95% of the time the algorithm resulted in reductions for both the base-case cost and the sum of contingency cases cost.

Figure 2a shows a histogram of the reduction in the value of the cost function for 100 runs of the DDUC algorithm. This shows us the distribution of the cost reduction to be expected from an arbitrary starting UC to a corrected UC for the S.C. grid. Considering the base-case cost, over the 100 run sample: the minimum cost reduction observed was \$313, the mean cost reduction observed was \$30,986, and the maximum cost reduction observed was \$56,432. Importantly, in none of the 100 runs was a cost increase observed.

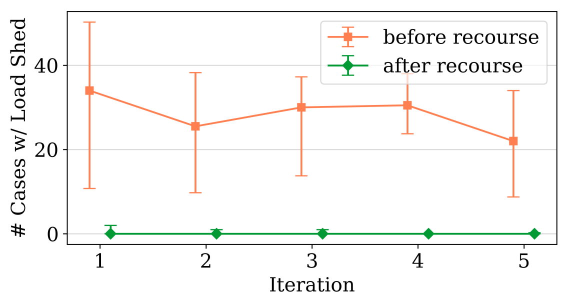
4.4. Texas Grid Results

Repeating the same tests on the Texas grid, we see similarly positive results. Figures 2b and 3 visualize these results on the Texas grid test case.

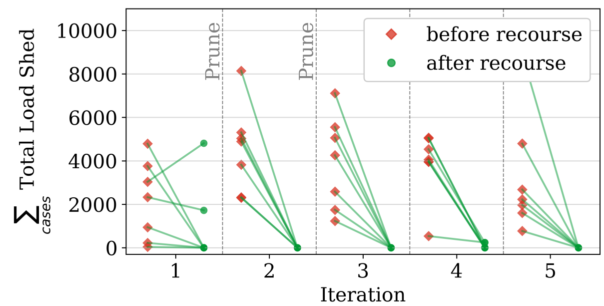
In Figure 2b, we see that considering the base-case cost, over the 100 run sample: the minimum cost reduction observed was \$32,341, the mean cost reduction observed was \$59,587, and the maximum cost reduction observed was \$102,508. While theses reductions are lower in relative terms than they were for the S.C. grid, they are significant in absolute terms. The distribution is also more impressive on the Texas grid in that even the minimum observed cost reductions are significantly positive. Again, in none of the 100 runs, was a cost increase observed.

Figure 3a shows the impressive results of the DDUC algorithm for eliminating load shed also persist on the Texas grid case. This figure shows that arbitrary starting UCs typically must shed load in 15-50 different contingency cases. After the recourse algorithm is run on this starting UC, however, the number of cases requiring load shedding drops considerably to the range of 0-2. Subsequent iterations of the DDUC algorithm perform similarly well, eventually eliminating load shedding altogether seeing no load shedding cases for iterations 4 and 5 in the middle quartiles of data. This is a better result than on the S.C. grid, and is perhaps attributable to the greater size and diversity of the Texas grid making it less vulnerable to a particular outage.

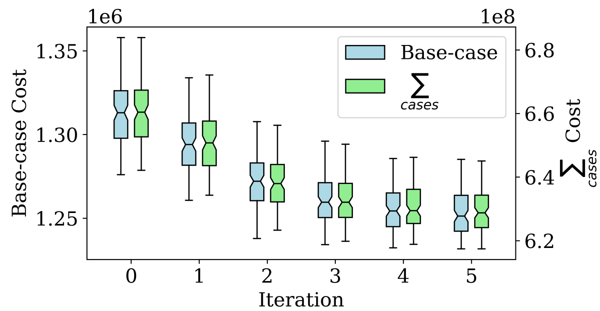
In Figure 3b, we see that during the first iteration of the DDUC algorithm there is one run where total load shedding over all cases increases after the recourse algorithm. This is not persistent however, as the recourse algorithm consistently eliminates load shedding on subsequent iterations. The high level of load shedding after the pruning phase in contrast to the S.C. grid is attributable to the fact that the larger



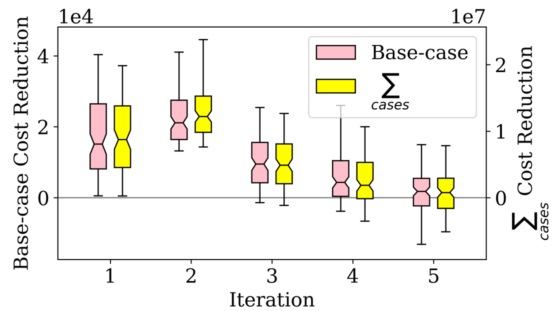
(a) Texas grid number of scenarios with load shedding (n=100)



(b) Texas grid total load shed reduction from recourse (n=7)



(c) Texas grid absolute cost function value (n=100)



(d) Texas grid iteration-over-iteration cost reduction (n=100)

Figure 3: DDUC algorithm performance on Texas grid. See Figure 1 for detailed description of plots.

Texas grid undergoes more aggressive pruning since the parameter of how many generators to eliminate is set as a percentage of the active generators, and was set higher in the Texas grid.

Figure 3c shows an impressive improvement in cost, with the median cost of a starting UC at \$1.31m compared to the median cost of \$1.25m after 5 iterations of the UC algorithm; an improvement of 4.7%.

In Figure 3d, again we see that for each of the five iterations, the algorithm resulted in a cost reduction to the system in the majority of instances. Notably, in both the first and second iteration, at least 95% of the time the algorithm resulted in reductions for both the base case cost and the sum of contingency scenario costs. Interestingly, in the Texas grid, the DDUC algorithm does a better job of reducing cost in the second iteration than the first one.

5. Discussion

The data-driven approach to finding UCs that are both secure and low-cost has some fundamental merits that make it worthwhile exploring in more depth. Perhaps the most important of these is that the input data for the approach is coming from a stochastic SC ACOPF process that considers both contingencies and weather scenarios. Using this data as a feed-stock for UC refinement therefore provides the possibility to find UCs that are best with respect to not only some base-case scenario, but also to an aggregation of possible contingencies and weather conditions. The ability to include contingencies and stochastic factors into the UC optimization process is going to be critical to increasing the penetration of renewables without sacrificing grid reliability. In addition, inducting knowledge of contingencies and weather into the UC optimization process could give grid planners confidence to tighten the safety margins used in UC decisions. This would allow for more cost-effective grid operation in the same way that superior calculation techniques in civil engineering allowed for building designs using fewer materials.

Another important benefit of this DDUC approach is that the incremental computations to find the UC refinements are extremely efficient and low-cost. The underlying algorithms are linear complexity with respect to both the grid topology, and the number of contingencies and scenario cases considered. In contrast to the standard mixed-integer programming approach to UC, this means that the method is highly scalable as network complexity increases.

A potential deployment of the DDUC approach could be: an initial seed UC is found using the traditional mixed-integer approach. Economic dispatch with respect to contingencies and weather scenarios is then run using this seed UC. The data from the economic dispatch is then recycled into the DDUC algorithm to refine the seed UC with the knowledge of contingencies and weather scenarios that is contained in the economic dispatch solution, and this refinement process is repeated as new information on the state of the grid and the weather becomes available.

Another interesting possible application of the DDUC approach is for black-start operations where it is known a priori which units to start and due to the weak grid conditions, the chances of load shedding due to improper unit commitment is significant. In such a case, the DDUC approach could provide feasible and optimal grid operational solutions with no or least amount of load shed.

The primary goal of the DDUC algorithm is to reduce load shedding across a range of contingencies by finding UCs that are less vulnerable to the *ensemble* of contingencies. This is a critical consideration since more than 13 million people in the United States were affected by power outages each year from 2008 to 2015, and the annual number of outages grew from 2,169 to 3,571 in this same time period [15]. Another interesting area of research to explore would be "demand-response" which amounts to voluntary load shedding. Demand-response programs are common and rapidly growing in the US, and usually involve some incentive that is provided by the grid operator to induce large consumers to lower their consumption when conditions are tight [7]. The DDUC approach would allow for demand-response to be explicitly modeled using the cost of the incentive as the cost for load reduction variables in ExaGO. This is an exciting possibility that could allow operators to better understand the relationship between UCs and demand-response requirements.

While the results of the numerical experiments presented herein are preliminary, they give reason to justify optimism in the possibilities of the DDUC approach as it is refined through further research. These experiments have shown a validation of the heuristics, showing their ability to identify both the "important" and "unimportant" generators in the grid. While benefits in terms of total generation cost will likely diminish with more optimal seed UCs generated from a mixed-integer programming approach, the DDUC method should be recognized for its ability to efficiently include contingency and weather information into the UC. While the experiments were only run with contingencies in this work, the inclusion of stochastic weather scenarios would be a simple extension since this functionality is already fully supported by ExaGO.

6. Conclusion

To achieve decarbonization goals [25], we need to develop tools to efficiently manage highly dynamic and stochastic power grids. Stochasticity of weather and operations and their impact on the grid can be modeled effectively in the economic dispatch process using standard models and proven approaches. New computational tools such as ExaGO allow us to vastly scale-up stochasticity modeling efforts by leveraging parallel computing and modern computational resources to give high-resolution information in short time-frames. The outcome is a tool set and data to support the decision-making process on resilience investments, improving operational efficiency and grid reliability. These economic dispatch computations with contingencies and stochastic scenarios generate huge amounts of high-quality data that can be analyzed and effectively "recycled" to imbue contingency and stochastic information into unit commitment through heuristic refinements. This process is a minimal intervention approach that is intrinsically scalable given the linear complexity of the graph-based algorithms used to analyze the economic dispatch output data. The preliminary results presented in this work show that this data-driven unit commitment approach has merit. It performed effectively on non-trivial grids of 500 and 2000 buses, consistently providing unit commitments that strictly enforced all security constraints (including nonlinear) and minimized or fully eliminated the need for load shedding across contingencies with power imbalance.

With further research, this data-driven unit commitment refinement approach could be tested across a wide array of grids, with larger contingency sets, with stochastic weather scenarios, and with different seed unit commitments. With further development and a robust implementation, this could prove to be a computationally efficient and practical method to include contingency and stochastic weather information into the unit commitment optimization process; providing more reliable and efficient grids with high penetrations of renewable energy.

doe-public-access-plan).

References

[1] S. Abhyankar et al. "Evaluation of AC optimal power flow problems on Graphical Processing Units". In: *Proceedings of IEEE PES General Meeting*. 2021.

[2] S. Abhyankar et al. *ExaGO Git repository*. https://gitlab.pnnl.gov/exasgd/frameworks/exago.

[3] Saleh Y. Abujarad, M.W. Mustafa, and J.J. Jamian. "Recent approaches of unit commitment in the presence of intermittent renewable energy resources: A review". In: *Renewable and Sustainable Energy Reviews* 70 (2017), pp. 215–223. ISSN: 1364-0321.

[4] W. van Ackooij et al. "Large-scale unit commitment under uncertainty: an updated literature survey". In: *Annals of Operations Research* 271.1 (Dec. 2018), pp. 11–85. ISSN: 1572-9338.

[5] Christie Alappat et al. "A Recursive Algebraic Coloring Technique for Hardware-Efficient Symmetric Sparse Matrix-Vector Multiplication". In: *ACM Trans. Parallel Comput.* 7.3 (June 2020). ISSN: 2329-4949.

[6] Ignacio Aravena et al. *Recent Developments in Security-Constrained AC Optimal Power Flow: Overview of Challenge 1 in the ARPA-E Grid Optimization Competition*. 2022. arXiv: 2206.07843 [math.OC].

[7] Luis Alejandro Arias et al. "A Review and Analysis of Trends Related to Demand Response". In: *Energies* 11.7 (2018). ISSN: 1996-1073.

[8] Adam B. Birchfield et al. "Grid Structural Characteristics as Validation Criteria for Synthetic Networks". In: *IEEE Transactions on Power Systems* 32.4 (2017), pp. 3258–3265.

[9] Adam Barlow Birchfield et al. "Building Synthetic Power Transmission Networks of Many Voltage Levels, Spanning Multiple Areas". In: *51st Hawaii International Conference on System Sciences*. 2018.

[10] Matthias Bollhöfer et al. "Large-scale Sparse Inverse Covariance Matrix Estimation". In: *SIAM Journal on Scientific Computing* 41.1 (2019), A380–A401.

[11] Matthias Bollhöfer et al. "State-of-the-Art Sparse Direct Solvers". In: (2020). Ed. by Ananth Grama and Ahmed H. Sameh, pp. 3–33.

[12] Kwok Cheung et al. "Toward scalable stochastic unit commitment". In: *Energy Systems* 6.3 (Sept. 2015), pp. 417–438. ISSN: 1868-3975.

[13] I.A. Farhat and M.E. El-Hawary. "Optimization methods applied for solving the short-term hydrothermal coordination problem". In: *Electric Power Systems Research* 79.9 (2009), pp. 1308–1320. ISSN: 0378-7796.

[14] Martin Håberg. "Fundamentals and recent developments in stochastic unit commitment". In: *International Journal of Electrical Power & Energy Systems* 109 (2019), pp. 38–48. ISSN: 0142-0615.

[15] Hassan Haes Alhelou et al. "A Survey on Power System Blackout and Cascading Events: Research Motivations and Challenges". In: *Energies* 12.4 (2019). ISSN: 1996-1073.

[16] Juraj Kardoš et al. "High-Performance Data Analytics Techniques for Power Markets Simulation". In: *2021 International Conference on Smart Energy Systems and Technologies (SEST)*. 2021, pp. 1–6.

[17] Juraj Kardoš et al. "Massively parallel data analytics for smart grid applications". In: *Sustainable Energy, Grids and Networks* 31 (2022), p. 100789. ISSN: 2352-4677.

[18] Bernard Knueven, James Ostrowski, and Jean-Paul Watson. "On Mixed-Integer Programming Formulations for the Unit Commitment Problem". In: *INFORMS Journal on Computing* 32.4 (2020), pp. 857–876.

[19] Cosmin Petra. *HiOP User Guide version 0.3*. https://github.com/LLNL/hiop/blob/master/doc/hiop_usermanual.pdf. 2017.

[20] Cosmin G Petra and Ignacio Aravena. "Solving realistic security-constrained optimal power flow problems". In: *Operations Research* (2023). to appear, preprint available at https://arxiv.org/pdf/2110.01669.pdf.

[21] Logan Rakai and William Rosehart. "GPU-Accelerated Solutions to Optimal Power Flow Problems". In: *47th Hawaii International Conference on System Sciences*. 2014.

[22] B. Saravanan et al. "A solution to the unit commitment problem—a review". In: *Frontiers in Energy* 7.2 (June 2013), pp. 223–236. ISSN: 2095-1698.

[23] David Schneider. "The Exascale Era is Upon Us: The Frontier supercomputer may be the first to reach 1,000,000,000,000,000,000 operations per second". In: *IEEE Spectrum* 59.1 (2022), pp. 34–35.

[24] J. Charles Smith et al. "R&D Requirements for Integration of Wind Generation". In: *45th Hawaii International Conference on System Sciences*. 2012.

[25] United States Department of State. *The Long-Term Strategy of the United States: Pathways to Net-Zero Greenhouse Gas Emissions by 2050*. Tech. rep. United States Department of State and the United States Executive Office of the President, 2021.

[26] Xueneng Su et al. "Full Parallel Power Flow Solution: A GPU-CPU-Based Vectorization Parallelization and Sparse Techniques for Newton–Raphson Implementation". In: *IEEE Transactions on Smart Grid* 11.3 (2020), pp. 1833–1844.

[27] Andreas Wächter and Lorenz T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical Programming* 106.1 (Mar. 2006), pp. 25–57. ISSN: 1436-4646.

[28] Jingyi Wang and Cosmin G. Petra. "An adaptive sampling sequential quadratic programming method for nonsmooth stochastic optimization with upper-C^2 objective". In: *SIAM Journal On Optimization* (2023). to appear, preprint available at https://arxiv.org/abs/2204.09631.

[29] Ti Xu et al. "Application of Large-Scale Synthetic Power System Models for Energy Economic Studies". In: *50th Hawaii International Conference on System Sciences*. 2017.

[30] Qipeng P. Zheng, Jianhui Wang, and Andrew L. Liu. "Stochastic Optimization for Unit Commitment—A Review". In: *IEEE Transactions on Power Systems* 30.4 (2015).