A Manifold-Based ADMM Framework for Mixed-Integer Unit Commitment Problems

Yunhe Li, Wenchuan Wu, Chenhui Lin, and Qi Wang

***Abstract*—Unit commitment (UC) problems are fundamental in power system operations but computationally challenging due to their mixed-integer programming (MIP) nature. This paper proposes a novel manifold-based alternating direction method of multipliers (ADMM) framework that reformulates the discrete binary space into a continuous spherical manifold space, enabling efficient solution through continuous optimization techniques. The proposed method decomposes the UC problem into two subproblems: a manifold optimization over the sphere and a box-constrained quadratic program. Adaptive parameter tuning, early termination, and solution recovery strategies are integrated to enhance convergence and solution quality. Case studies on benchmark UC instances with different sizes demonstrate that the proposed method achieves convergent solutions with significant computational speedup, and the resulting optimality gaps are relatively small, making the framework suitable for practical power system applications.**

***Index Terms*—Unit commitment, manifold optimization, ADMM, mixed-integer programming, spherical manifold**

I. Introduction

Unit commitment (UC) is a critical optimization problem in power system operations that determines the on/off status and power output of generating units over a planning horizon while minimizing operational costs and satisfying various technical constraints. The UC problem is typically formulated as a large-scale mixed-integer programming (MIP) problem, where binary variables represent unit commitment decisions and continuous variables represent power generation levels. Generation costs are commonly modeled using piecewise linear approximations to maintain computational tractability while capturing essential cost characteristics.

A. Challenges in UC Problem Solving

The computational complexity of UC problems arises from several factors:

- **Discrete-continuous coupling**: Binary commitment variables are tightly coupled with continuous power dispatch variables through ramping constraints, minimum up/down time requirements, and startup/shutdown costs.
- **Scalability**: Modern power systems may involve hundreds to thousands of generating units with 24-168 hour planning horizons, resulting in tens of thousands of binary variables.
- **Computational demands**: Real-time or near-real-time decision-making requires efficient solution methods that can handle large-scale instances within strict time limits.

B. Related Work

Traditional approaches for solving UC problems include:

MIP-based methods: Commercial solvers like Gurobi and CPLEX employ branch-and-bound with sophisticated cutting plane techniques. Carrión and Arroyo [1] developed tight MIP formulations that significantly improve computational performance. Ostrowski et al. [2] proposed tighter formulations incorporating ramping constraints and startup/shutdown trajectories. Knueven et al. [3] introduced novel MIP formulations exploiting piecewise linear cost structures for enhanced computational efficiency. Bai et al. While effective for moderate-scale problems, these methods face scalability challenges for large systems with tight time requirements.

Decomposition methods: Alternating direction method of multipliers (ADMM) has gained attention for large-scale optimization. Guo et al. [4] developed ADMM-based algorithms for security-constrained unit commitment in multi-area systems. Dvorkin et al. [5] demonstrated ADMM's effectiveness for operational planning with distributed energy resources. Jacobson et al. [6] proposed a computationally efficient Benders decomposition for energy systems planning problems with detailed operations and time-coupling constraints. Recent works have shown promising results in distributed and parallel computing settings.

Continuous relaxation approaches: Several methods project discrete variables into continuous spaces to enable gradient-based optimization. Fuller and Chertkov [7] proposed distributed algorithms using continuous relaxations for approximate UC solutions. Zheng et al. [9] utilized chordal conversion techniques to solve semidefinite programming relaxations. Mhanna et al. [8] developed adaptive partition-based approaches that iteratively refine continuous relaxations. However, ensuring integer feasibility and solution quality remains an active research area.

C. Contributions

This paper proposes a novel manifold-based ADMM framework that addresses UC challenges through:

1) **Manifold reformulation**: Binary variables are mapped to a continuous spherical manifold, enabling smooth continuous optimization.

This work was supported in part by the National Science Foundation of China (Grant. U24B6009) and in part by Beijing Natural Science Foundation(Grant.L243003)

979-8-3315-8138-1

2) **ADMM decomposition**: The problem is decomposed into a manifold projection subproblem and a box-constrained QP(Quadratic Programming) subproblem, both efficiently solvable.
3) **Enhanced convergence strategies**: Adaptive penalty parameters, progressive integerization, and strict early termination criteria ensure solution quality.
4) **Solution recovery mechanism**: A warm-start MIP strategy refines near-integer solutions to satisfy all constraints.

Comprehensive case studies on benchmark instances demonstrate the method's effectiveness and computational efficiency.

II. Problem Formulation

A. Unit Commitment Model

The UC problem is formulated as a mixed-integer programming problem with piecewise linear cost approximation:

$$\min_{u,p,c^{sc}} \quad \sum_{g\in\mathcal{G}}\sum_{t\in\mathcal{T}}\Big[C_g(p_{g,t}) + c^{sc}_{g,t}\Big] \tag{1a}$$

$$\text{s.t.} \quad \sum_{g\in\mathcal{G}} p_{g,t} = D_t, \quad \forall t\in\mathcal{T} \tag{1b}$$

$$P_g^{\min} u_{g,t} \le p_{g,t} \le P_g^{\max} u_{g,t}, \quad \forall g,t \tag{1c}$$

$$p_{g,t} - p_{g,t-1} \le S_g^{up} u_{g,t} + (R_g^{up} - S_g^{up}) u_{g,t-1}, \quad \forall g,t \tag{1d}$$

$$p_{g,t-1} - p_{g,t} \le S_g^{dn} u_{g,t-1} + (R_g^{dn} - S_g^{dn}) u_{g,t}, \quad \forall g,t \tag{1e}$$

$$c^{sc}_{g,t} \ge S_g^{up}(u_{g,t} - u_{g,t-1}), \quad \forall g,t \tag{1f}$$

$$c^{sc}_{g,t} \ge S_g^{dn}(u_{g,t-1} - u_{g,t}), \quad \forall g,t \tag{1g}$$

$$\sum_{\tau=t-T_g^{up}+1}^{t} (u_{g,\tau} - u_{g,\tau-1}) \le u_{g,t}, \quad \forall g,t \tag{1h}$$

$$\sum_{\tau=t-T_g^{dn}+1}^{t} (u_{g,\tau-1} - u_{g,\tau}) \le 1 - u_{g,t}, \quad \forall g,t \tag{1i}$$

$$\sum_{g\in\mathcal{G}} p_{g,t} \ge D_t + R_t^{req}, \quad \forall t \tag{1j}$$

$$u_{g,t}\in\{0,1\}, \quad p_{g,t}\ge 0, \quad c^{sc}_{g,t}\ge 0 \tag{1k}$$

where $C_g(p_{g,t})$ represents the generation cost, typically approximated as a piecewise linear function:

$$C_g(p_{g,t}) = a_g p_{g,t} + b_g u_{g,t} \tag{2}$$

with a_g and b_g being the marginal and no-load costs, respectively. $\mathcal{G}$ and $\mathcal{T}$ denote the sets of generators and time periods, respectively; $u_{g,t}\in\{0,1\}$ indicates unit commitment status, $p_{g,t}$ denotes active power dispatch, $c^{sc}_{g,t}$ represents the startup/shutdown cost at time t, $C_g(\cdot)$ represents generation costs, S_g^{up} and S_g^{dn} denote startup and shutdown costs, D_t is system demand, R_t^{req} the reserve requirement, and parameters $P_g^{\min/\max}$, $R_g^{up/dn}$, $S_g^{up/dn}$ and $T_g^{up/dn}$ encode capacity, ramping (including startup/shutdown ramps) and minimum up/down time constraints; this compact single-variable formulation captures the essential operational constraints and cost structure of UC without requiring explicit startup/shutdown indicator variables.

III. Methodology

A. Manifold-ADMM Framework

The key innovation of our approach is reformulating the binary constraint space into a continuous manifold space, enabling efficient optimization through differential geometry.

1) Spherical Manifold Reformulation: We substitute binary variables with the intersection of a hypercube and a sphere.

For binary variables $x_{\text{bin}}\in\{0,1\}^n$:

$$x\in\{0,1\}^n \Leftrightarrow x\in[0,1]^n \cap \left\{x : \left\|x - \frac{1}{2}1^n\right\|_2^2 = \frac{n}{4}\right\} \tag{3}$$

We introduce two sets of continuous variables: one in Euclidean space Ω and another on a spherical manifold $\mathcal{M}$. Euclidean variable satisfies box constraints:

$$\Omega = \{x\in\mathbb{R}^n | x\in[0,1]^n\} \tag{4}$$

Spherical variable lies on the unit sphere:

$$\mathcal{M} = \{z\in\mathbb{R}^n | \sum_{i=1}^n z_i^2 = 1\} \tag{5}$$

Define the coupling constraint between two spaces:

$$z - \frac{2}{\sqrt{n}}(x - \frac{1}{2}1^n) = 0, \quad z\in\mathcal{M}, \quad x\in\Omega \tag{6}$$

This coulping constraint is satisfied if and only if x is binary. The reformulation can be illustrated by the picture in Fig. 1.

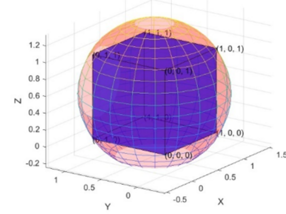

Fig. 1: Spherical manifold reformulation of binary constraints

2) ADMM Decomposition: We introduce an auxiliary variable o to replace the binary variables, enabling ADMM decomposition. The UC problem can be expressed in the following compact form:

$$\begin{aligned} \min_{o,p,y} \quad & f(o,p,y) \\ \text{s.t.} \quad & \text{Constraints } (1b)-(1j) \text{ for given } o, \\ & o\in[0,1]^d, \quad y\in\mathbb{S}^{d-1}, \\ & y = \frac{2}{\sqrt{d}}(o - \frac{1}{2}\mathbf{1}^d) \end{aligned} \tag{7}$$

979-8-3315-8138-1

where $f(o,p)$ represents the total cost function in (1a) expressed in terms of o and p by substituting u with o, $d = N_g \times N_t$.

The augmented Lagrangian function is:

$$\begin{aligned}\mathcal{L}(o,y,\lambda) =& f(o,p,y) + \lambda^T \left[o - \frac{2}{\sqrt{d}} \left(y - \frac{1}{2}\mathbf{1}^d \right) \right] \\ &+ \frac{\rho}{2} \left\| o - \frac{2}{\sqrt{d}} \left(y - \frac{1}{2}\mathbf{1}^d \right) \right\|_2^2 + Pen(o) \end{aligned} \quad (8)$$

The coupling constraint is seen as a consensus constraint in ADMM. The penalty term $Pen(o)$ encourages o to be binary.

ADMM iterations consist of:

Update 1 - Manifold projection (Update_Lp):

$$y^{k+1} = \arg \min_{y \in \mathbb{S}^{d-1}} \left\{ \langle \lambda^k, o^k - \tilde{o}(y) \rangle + \frac{\rho}{2} \| o^k - \tilde{o}(y) \|_2^2 \right\} \quad (9)$$

where $\tilde{o}(y) = \frac{\sqrt{d}}{2} y + \frac{1}{2}\mathbf{1}^d$.

This subproblem is solved using Riemannian steepest descent on the sphere manifold $\mathbb{S}^{d-1}$ with the Pymanopt library [12].

Update 2 - Box projection (Update_Box):

$$\begin{aligned} o^{k+1} = \arg \min_{o,p} \quad & f(o,p,y) - \langle \lambda^k, o \rangle + \frac{\rho}{2} \| o - \tilde{o}(y^{k+1}) \|_2^2 \\ & + w \| o - [o]_{[0,1]} \|_2^2 \\ \text{s.t.} \quad & \text{Constraints } (1b) - (1j) \text{ for given } o \end{aligned} \quad (10)$$

where $[o]_{[0,1]}$ projects o onto $[0,1]$ element-wise, and w is the integerization penalty weight, $w\|o - [o]_{[0,1]}\|_2^2$ is $Pen(o)$. This is a continuous quadratic program solved using Gurobi [13] with continuous relaxation of o.

Dual update:

$$\lambda^{k+1} = \lambda^k + \rho(o^{k+1} - \tilde{o}(y^{k+1})) \quad (11)$$

B. Enhancement Strategies

1) Adaptive Parameter Tuning: The penalty parameters ρ and w are crucial for convergence. The consensus penalty ρ is progressively increased by a factor of α at each iteration to enforce the constraint $u = o$. The integerization weight w is adaptively adjusted to guide the solution towards binary values. We define a variable o_i as "fractional" if its value lies within the range $(0.01, 0.99)$. If the count of such fractional variables exceeds a threshold (e.g., 30% of total variables), the weight w is increased by a multiplicative factor β. This strategy intensifies the penalty on non-integer solutions only when the solution is far from being integral, thereby promoting integer feasibility more aggressively when needed. In our implementation, we use initial values $\rho_0 = 150$, $w_0 = 15$, and update factors $\alpha = 1.10$, $\beta = 1.5$.

2) Strict Early Termination: To ensure solution quality, a strict early termination strategy is employed. The algorithm terminates prematurely only if two conditions are simultaneously met over a "patience" window of p iterations: 1) the primal and dual residuals have stabilized and fallen below a predefined tolerance ϵ, indicating that the ADMM consensus has been reached; and 2) the number of fractional variables is below a small threshold γ, indicating the solution is sufficiently close to being integral. This dual-check mechanism prevents premature termination with a high-quality but non-integer solution or a binary but suboptimal one. We use $p = 8$, $\gamma = 0.01$, and $\epsilon = 3 \times 10^{-3}$.

TABLE I: Computational Results on Pglib-UC Benchmark Instances

Case	MIP Obj.	MIP Time	Manifold-ADMM Obj.	Manifold-ADMM Time	Gap (%)
2014-09-01_reserves_3	48159	303.58s	48166	214.52s	0.0150
2014-09-01_reserves_5	48283	307.91s	48296	244.63s	0.0279
2014-12-01_reserves_0	39127	62.80s	39128	41.52s	0.0025
2015-03-01_reserves_0	31707	87.83s	31715	54.01s	0.0264
2015-03-01_reserves_5	31845	158.10s	31857	53.26s	0.0364

3) Feasibility Recovery: If the ADMM algorithm converges to a solution that is not perfectly integer-valued, a feasibility recovery step is initiated. This is necessary because rounding a near-integer solution does not guarantee that all original problem constraints (e.g., ramping, minimum up/down times) will be satisfied. The recovery process first checks if the rounded solution is feasible by substituting it into the original UC model. If it is found to be infeasible, we then solve the full MIP problem using a MIP solver. The key to this step is providing the rounded, infeasible ADMM solution as a warm start. This gives the solver a high-quality starting point, significantly accelerating its search for a nearby, feasible integer solution within a strict time limit and a specified MIP gap tolerance.

C. Algorithm Summary

Algorithm 1 summarizes the complete procedure. The method seamlessly integrates manifold optimization, ADMM decomposition, and solution refinement.

IV. Case Study

A. Test Instances and Setup

We evaluate the proposed method on benchmark UC instances from two standard libraries:

- **Pglib-UC** [11]: A collection of realistic, large-scale unit commitment instances. The specific cases used in this study feature 610 generators over a 48-hour time horizon.
- **OR-Library** [10]: A widely used collection for UC algorithm benchmarking, with instances ranging from 10 to 200 units over a 24-hour horizon.

All tests are conducted on a workstation with Intel Core i7 processor and 32GB RAM. We compare:

- **Direct MIP**: Gurobi 10.0 with 0.01% MIP gap tolerance
- **Proposed method**: Manifold-ADMM with default parameters ($\rho_0 = 150$, $w_0 = 15$, $\alpha = 1.10$)

B. Experimental Results

Table I and II presents results for selected instances with different scales and multiple time periods. The proposed method demonstrates competitive performance with near-optimal solution quality.

- **Solution Quality**: The proposed method consistently finds near-optimal solutions. For the PGLIB instances

979-8-3315-8138-1

Algorithm 1 Manifold-ADMM for Unit Commitment

Input: UC model, initial y^0 (from LP relaxation), penalty ρ_0, weight w_0, multiplier λ^0
Parameters: Increment factor α, weight factor β, patience p, threshold θ
Initialize: $k = 0$, $\rho = \rho_0$, $w = w_0$
while $k < k_{\max}$ and not converged **do**
  // Manifold update
  $y^{k+1} \leftarrow$ Update_Lp(o^k, λ^k, ρ) via Riemannian optimization
  $\tilde{o}^{k+1} \leftarrow \frac{\sqrt{d}}{2} y^{k+1} - \mathbf{1}$
  // Box update
  $o^{k+1}, p^{k+1} \leftarrow$ Update_Box$(\tilde{o}^{k+1}, \lambda^k, \rho, w)$ via MIQP
  // Dual update
  $\lambda^{k+1} \leftarrow \lambda^k + \rho(o^{k+1} - \tilde{o}^{k+1})$
  // Adaptive parameters
  $\rho \leftarrow \alpha \cdot \rho$
  if fractional variables $> \theta \cdot d$ **then**
    $w \leftarrow \beta \cdot w$
  end if
  // Check termination
  if strict early stopping criteria met **then**
    break
  end if
  $k \leftarrow k + 1$
end while
// Solution recovery
if o^k is not integer feasible **then**
  $u^* \leftarrow$ MIP_Recovery(round(o^k))
else
  $u^* \leftarrow o^k$
end if
Output: Commitment solution u^*, dispatch p^*

TABLE II: Computational Results on OR-Lib Benchmark Instances

Case	MIP Obj.	MIP Time	Manifold-ADMM Obj.	Manifold-ADMM Time	Gap (%)
50_0_4_w	8296296	159.67s	8296458	72.03s	0.4582
50_0_1_w	8452324	108.41s	8454277	84.09s	0.4823
75_0_3_w	12907499	287.70s	12911684	115.25s	0.4233
75_0_1_w	9834726	297.16s	9838114	108.06s	0.4019
75_0_2_w	12194718	209.18s	12198762	107.94s	0.4286
100_0_1_w	17079253	1401.51s	17081444	116.16s	0.4209
100_0_3_w	15941727	274.55s	15942231	123.09s	0.3236
100_0_2_w	15893745	498.55s	15897994	122.61s	0.3380
150_0_2_w	25258540	774.30s	25266002	251.58s	0.3480
150_0_1_w	27321403	189.76s	27328616	168.48s	0.3501
150_0_3_w	25257903	868.27s	25258859	218.00s	0.2894
200_0_10_w	38330379	531.51s	38338785	469.15s	0.3777
200_0_1_w	35227051	1138.30s	35233444	438.95s	0.4063
200_0_2_w	32331175	956.82s	32341447	386.52s	0.3945
200_0_4_w	30375505	445.23s	30383748	273.46s	0.3627
200_0_6_w	30936299	248.85s	30939461	247.43s	0.3151
200_0_7_w	37112289	468.34s	37123572	321.64s	0.4132
200_0_8_w	34068625	331.71s	34072700	254.01s	0.3478
200_0_3_w	30676730	584.03s	30685123	435.13s	0.3358

(Table I), the optimality gap is extremely small, typically below 0.04%. For the more challenging OR-Lib instances (Table II), the gap remains below 0.5%, which is acceptable for many operational purposes, demonstrating the robustness of the feasibility recovery step.

- **Computational Speedup**: Significant computational speedup is observed, particularly for the OR-Lib instances. As shown in Table II and illustrated in Figure 3, the ADMM method can be over 10 times faster than a direct MIP solve for certain cases (e.g., '100_0_1_w'). For the selected PGLIB instances, the speedup is more moderate but still consistent, ranging from 1.2× to 2.9×.
- **Scalability**: The method demonstrates strong scalability. The time comparison in Figure 3 shows that the performance advantage of the ADMM-based approach is maintained across various problem sizes.
- **Convergence**: The algorithm exhibits stable convergence. As shown in Figure 2 for a 200-generator OR-Lib instance, the objective function value steadily improves while the ADMM residual decreases towards the tolerance threshold, typically within 90 iterations. The number of fractional variables also diminishes, indicating the effectiveness of the integerization strategy.

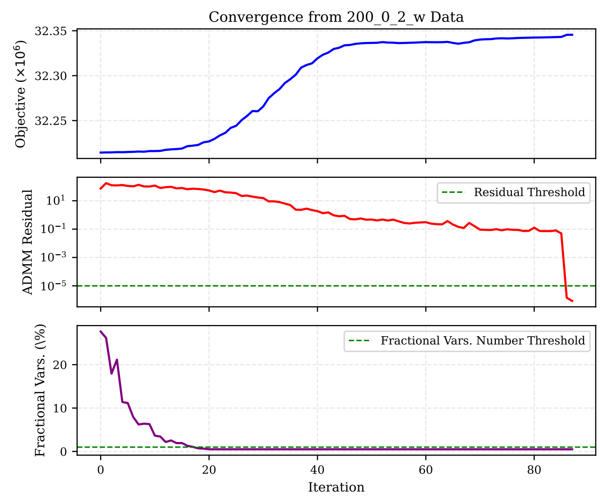

Fig. 2: Convergence behavior of manifold-ADMM on a real data instance (200 generators, OR-Lib), showing objective value, ADMM residual, and percentage of fractional variables over iterations.

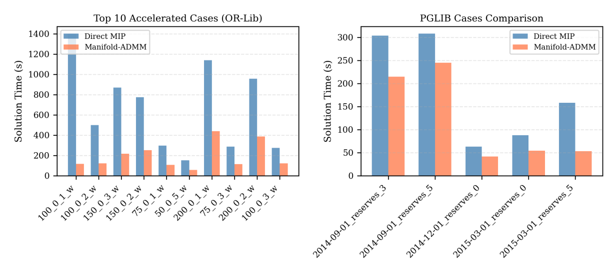

Fig. 3: Computational time comparison (OR-Lib cases): absolute solution times for direct MIP vs. manifold-ADMM across selected OR-Lib instances where the proposed method achieved significant speedup.

C. Discussion

The experimental results validate several key aspects:

979-8-3315-8138-1

Effectiveness of manifold reformulation: The continuous manifold space enables smooth optimization trajectories, avoiding the combinatorial explosion of discrete branching. The consistently low optimality gaps—below 0.04% for PGLIB cases—underscore the method's ability to find high-quality solutions.

ADMM decomposition benefits: Separating the manifold projection and the box-constrained projection allows for the use of specialized solvers for each subproblem. The convergence progress shows that the ADMM framework effectively coordinates these updates to reach consensus, which demonstrates this manifold-ADMM has the potential to efficiently solve large-scale problems.

Solution recovery strategy: This paper proposes several strategies to obtain high-quality feasible solutions of MIP problems based on the continuous manifold solutions. For the PGLIB-UC instances, this step successfully produced feasible solutions with gaps under 0.04%, demonstrating its reliability in converting high-quality continuous solutions into practical, integer-feasible schedules.

Scalability considerations: The framework demonstrates strong performance on instances with up to 610 generators and a 48-hour horizon in efficiency. The results suggest that the approach is scalable and provides a solid foundation for further extensions, such as parallelization and distributed computing.

V. Conclusion

This paper presents a novel manifold-based ADMM framework for solving large-scale unit commitment problems by reformulating binary variables onto a continuous spherical manifold. The proposed method combines Riemannian optimization for manifold projection with ADMM decomposition, adaptive penalty parameters, and strict early stopping criteria to achieve near-optimal solutions efficiently. Experimental results on benchmark instances demonstrate both high solution quality and computational efficiency. For PGLIB instances with 610 generators, the method achieves optimality gaps below 0.04% with a speedup of 1.2-2.9$\times$ over a direct MIP solve. For OR-Lib instances, speedups of up to 12$\times$ are recorded while maintaining gaps below 0.5%.

Future research directions include parallel implementation exploiting generator-level decomposition, adaptive parameter tuning using machine learning, and extensions to stochastic UC with renewable uncertainty and network constraints. The manifold-ADMM framework provides a promising alternative approach that complements traditional MIP methods by leveraging continuous optimization techniques for large-scale mixed-integer power system optimization.

References

[1] M. Carrión and J. M. Arroyo, "A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem," *IEEE Transactions on Power Systems*, vol. 21, no. 3, pp. 1371–1378, 2006.

[2] J. Ostrowski, M. F. Anjos, and A. Vannelli, "Tight mixed integer linear programming formulations for the unit commitment problem," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 39–46, 2012.

[3] B. Knueven, J. Ostrowski, and J.-P. Watson, "A novel matching formulation for startup costs in unit commitment," *Mathematical Programming Computation*, vol. 12, no. 2, pp. 205–248, 2020.

[4] Y. Guo, L. Wu, and T. Ding, "ADMM-based decentralised unit commitment considering transmission constraints," *IET Generation, Transmission & Distribution*, vol. 11, no. 17, pp. 4341–4350, 2017.

[5] Y. Dvorkin, R. Fernandez-Blanco, D. S. Kirschen, H. Pandzic, J.-P. Watson, and C. A. Silva-Monroy, "Ensuring profitability of energy storage," *IEEE Transactions on Power Systems*, vol. 32, no. 1, pp. 611–623, 2017.

[6] S. H. Jacobson, F. N. Pecci, J. Sepulveda, Y. Xu, and J. D. Jenkins, "A computationally efficient Benders decomposition for energy systems planning problems with detailed operations and time-coupling constraints," *Computers & Chemical Engineering*, vol. 170, p. 108107, 2023.

[7] J. D. Fuller and M. Chertkov, "Distributed control of distributed generation: The consensus approach," in *American Control Conference*, 2014, pp. 2461–2466.

[8] S. Mhanna, G. Chapman, and G. Verbic, "Adaptive ADMM for distributed AC optimal power flow," *IEEE Transactions on Power Systems*, vol. 34, no. 3, pp. 2025–2035, 2019.

[9] Y. Zheng, S. Fantacci, and A. Papachristodoulou, "Fast ADMM for semidefinite programs with chordal sparsity," in *American Control Conference*, 2016, pp. 3335–3340.

[10] A. Frangioni, C. Gentile, and F. Lacalandra, "Tighter approximated MILP formulations for unit commitment problems," *IEEE Transactions on Power Systems*, vol. 24, no. 1, pp. 105–113, 2009.

[11] R. D. Zimmerman and C. E. Murillo-Sánchez, "Pglib-uc: A library of unit commitment problems," in *2020 IEEE Power & Energy Society General Meeting (PESGM)*, 2020, pp. 1–5.

[12] J. Townsend, N. Koep, and S. Weich, "Pymanopt: A python toolbox for optimization on manifolds," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1–5, 2016.

[13] Gurobi Optimization, LLC, "Gurobi optimizer reference manual," 2023.

979-8-3315-8138-1