

D²-UC: A Distributed–Distributed Quantum–Classical Framework for Unit Commitment

Milad Hasanzadeh *Graduate Student Member, IEEE* and Amin Kargarian, *Senior Member, IEEE*

Abstract—This paper introduces D²-UC, a quantum-ready framework for the unit commitment (UC) problem that prepares UC for near-term hybrid quantum–classical solvers by combining distributed classical decomposition with distributed quantum execution. We reformulate deterministic and stochastic UC into a three-block alternating direction method of multipliers (ADMM): (i) a convex quadratic subproblem for dispatch and reserves, (ii) a binary subproblem expressed as a quadratic unconstrained binary optimization (QUBO), and (iii) a proximal slack update for consensus. The core contributions are fivefold. First, we demonstrate how the full UC problem can be expressed as a single, monolithic QUBO, thereby establishing a direct interface to quantum solvers. Second, we decompose this large binary block into three type-specific QUBOs for commitment, startup, and shutdown, making the problem more tractable but revealing slower ADMM convergence. Third, we restore local logical couplings through per unit–time micro-QUBOs, which accelerate convergence. Fourth, we batch micro-QUBOs into K non-overlapping block-diagonal problems, reducing a large number of subproblems to a fixed number of solver-ready QUBOs per iteration, well matched to distributed variational quantum eigensolvers (DVQE). Fifth, we integrate an accept-if-better safeguard with DVQE to stabilize hybrid updates and prevent oscillations. Case studies on deterministic and stochastic UC confirm that the proposed methods deliver feasible schedules, faster convergence, and QUBO sizes aligned with current and near-term quantum hardware capabilities. All detailed data, codes, and parameter values for all cases studied are available at [GitHub](#).

Index Terms—Unit commitment, ADMM, QUBO, quantum computing, DVQE.

NOMENCLATURE

Indices, Sets, and Functions:

i	Index of generating units, $i \in \{1, \dots, N\}$.
t	Index of time periods, $t \in \{1, \dots, T\}$.
s	Index of uncertainty scenarios, $s \in \{1, \dots, S\}$.
k	Index for ADMM iterations.

Parameters:

L_t	System load demand at time t [MW].
$\tilde{L}_{t,s}$	Net load at time t under scenario s [MW].
$P_i^{\min}, P_i^{\max}$	Minimum/maximum generation capacity of unit i [MW].
$R_t^{\uparrow}, R_t^{\downarrow}$	Up/downward reserve requirements at time t [MW].
$R_{t,s}^{\uparrow}, R_{t,s}^{\downarrow}$	Scenario-dependent reserve requirements at time t [MW].

RU_i, RD_i	Ramp-up/-down limits of unit i [MW/h].
SU_i, SD_i	Startup and shutdown ramp limits [MW/h].
U_i, D_i	Minimum up and down time requirements [h].
A_i, B_i, C_i	Fixed, linear, and quadratic cost coefficients.
S_i, H_i	Startup and shutdown cost coefficients.
$\Delta\tau$	Reserve response time window [h].
π_s	Probability of scenario s .
ρ_y, ρ_u, ρ_v	ADMM penalty parameters for y, u, v .
$\beta_y, \beta_u, \beta_v$	Proximal penalty weights for y, u, v .
γ_c, γ_{ss}	Micro-QUBO penalty weights.
$\gamma_{u \rightarrow y}, \gamma_{v \rightarrow \bar{y}}$	Micro-QUBO penalty weights.
$\gamma_y, \gamma_u, \gamma_v$	Anchoring penalty weights for relaxed Block 1.
Variables:	
$y_{i,t}$	Commitment status of unit i at time t .
$u_{i,t}$	Startup indicator for unit i at time t .
$v_{i,t}$	Shutdown indicator for unit i at time t .
$p_{i,t,s}$	Generation output of unit i at time t under scenario s [MW].
$r_{i,t}^{\uparrow}, r_{i,t}^{\downarrow}$	Up/downward reserves of unit i at time t [MW].
$r_{i,t,s}^{\uparrow}, r_{i,t,s}^{\downarrow}$	Up/downward reserves under scenario s [MW].
$z_{i,t}^y, z_{i,t}^u, z_{i,t}^v$	Auxiliary binary proxies for y, u, v .
$\xi_{i,t}^y, \xi_{i,t}^u, \xi_{i,t}^v$	Proximal slack variables for y, u, v .
$\lambda_{i,t}^y, \lambda_{i,t}^u, \lambda_{i,t}^v$	Dual variables associated with consensus constraints.
$z_{i,t}$	Binary vector $[z_{i,t}^y, z_{i,t}^u, z_{i,t}^v]^T \in \{0, 1\}^3$.
$E_{i,t}(z)$	Local micro-QUBO energy function at (i, t) .

I. INTRODUCTION

UNIT commitment (UC) is a core optimization problem in power system operations, determining unit on/off status to minimize cost while satisfying demand, reserve, and technical constraints [1]. Formulated as a mixed integer linear programming/mixed integer quadratic programming (MILP/MIQP) and proven NP-hard [2], UC remains challenging despite advances in solvers and decomposition. Growing system scale and renewable integration further amplify its dimensionality, uncertainty, and computational burden [3]–[6]. The complexity of UC grows rapidly with system size, temporal detail, and uncertainty, making large-scale MILP/MIQP hard for classical solvers [7]. Quantum computing offers a promising alternative for such combinatorial tasks [8]–[11], with hybrid quantum–classical algorithms like variational quantum eigensolver (VQE) [12] and quantum approximate optimization algorithm (QAOA) [8] emerging as practical near-term approaches for noisy intermediate-scale quantum (NISQ) devices.

This work was supported by the National Science Foundation under Grant ECCS-1944752 and Grant ECCS-2312086.

The authors are with the Electrical and Computer Engineering Department, Louisiana State University, Baton Rouge, LA 70803 USA (email: mhasa42@lsu.edu, kargarian@lsu.edu).

Hybrid quantum–classical algorithms target quadratic unconstrained binary optimization (QUBO) problems, while UC is a mixed-integer program with both binary and continuous variables. Decomposition methods such as Benders and alternating direction method of multipliers (ADMM) [13]–[16] separate UC into convex continuous subproblems (dispatch, reserves) solvable classically and a discrete subproblem of binary commitments, which can be reformulated as a QUBO for quantum solvers [4], [17]. A three-block ADMM decomposition aligns UC with quantum solvers by separating continuous and discrete decisions [18]. Block 1 solves a convex QP for dispatch and reserves. Block 2 reformulates binary commitments, startups, and shutdowns as a QUBO with auxiliary proxies and slack variables. Block 3 updates the slack variables to enforce consensus. This structure preserves UC’s physical and operational constraints in a tractable form [14], [19].

Despite their promise, current quantum devices remain in the NISQ era, with limited numbers of qubits, restricted connectivity, and short coherence times [9]. As a result, solving large QUBO instances—such as those arising from full-scale UC formulations—remains a challenge, even when using hybrid quantum–classical algorithms. The gap between the problem sizes of practical interest and the hardware capabilities of existing quantum processors makes direct application infeasible [4], [5]. Two promising avenues to mitigate these limitations are QUBO decomposition methods, which partition a large QUBO into multiple smaller, tractable subproblems, and distributed quantum computing, which leverages parallel quantum resources or circuit partitioning to jointly address larger optimization tasks [20]–[22].

While the UC problem is formulated as a system-wide optimization, its binary structure exhibits a high degree of separability. In particular, the commitment, startup, and shutdown variables of each generating unit evolve independently of those of other units; for example, whether one unit starts up or remains committed has no direct effect on the binary status of another unit. The interdependence among units arises primarily through continuous constraints such as power balance, ramping limits, and reserve requirements, which can be handled in the convex subproblems [1], [2]. This structural property has also been recognized in emerging quantum-inspired formulations of UC [5], and it enables the decomposition of a large monolithic QUBO—containing all binary decisions across all units and periods—into multiple smaller QUBOs. Such a decomposition aligns naturally with the capabilities of current hybrid quantum–classical algorithms, allowing them to process these smaller QUBOs.

A further avenue to overcome the qubit limitations of current hardware is distributed quantum computing, where multiple small quantum processors cooperate to solve a larger optimization task. Recent advances, such as the TeleGate protocol, enable the execution of quantum gates across spatially separated quantum devices, creating the illusion of a single larger QPU by using entanglement and classical communication [23]–[25]. Building on such distributed primitives, hybrid algorithms can be extended to distributed settings. For example,

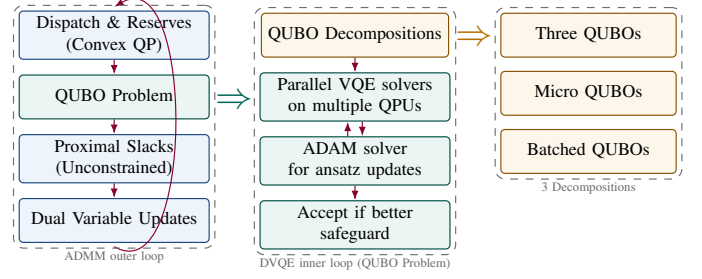


Fig. 1: Overview of the proposed framework

the distributed variational quantum eigensolver (DVQE) has been proposed as a distributed version of VQE, designed to solve QUBO problems by partitioning them across multiple quantum processors and aggregating results [22]. In DVQE, each QPU executes a subset of parameterized quantum circuits, and classical coordination iteratively updates the shared variational parameters. This architecture allows near-term QPUs to jointly tackle QUBOs of a size that would exceed the capacity of any single processor, making distributed quantum computing particularly promising for large-scale decompositions of the UC problem.

To address the challenges discussed above, we introduce D²-UC, a quantum-ready framework for UC that prepares the problem for near-term hybrid quantum–classical solvers by combining distributed classical decomposition with distributed quantum execution. We present three complementary decomposition strategies for isolating and solving the binary component of UC within a quantum-ready framework. The first approach decomposes the large QUBO into three independent blocks, separating the binary variables associated with unit commitments, startups, and shutdowns. The second approach restructures the problem into micro-QUBOs, each defined per unit–time pair, which restores local couplings and improves ADMM convergence. Building on this idea, a third strategy groups micro-QUBOs into batched QUBOs based on a hardness measure, a tool commonly used to characterize rugged energy landscapes in quantum optimization problems. To solve QUBOs of appropriate size, the framework leverages the DVQE, a distributed extension of VQE that enables QUBO problems to be partitioned and executed across multiple quantum processors. In addition, a safeguard mechanism is proposed to mitigate the “ping-pong” behavior reported in prior studies of hybrid quantum–classical methods, including observations made when applying DVQE. Fig. 1 illustrates the D²-UC framework, comprising the ADMM outer loop and the DVQE inner loop.

The paper is organized as follows. Section II formulates stochastic unit commitment and introduces the three-block ADMM decomposition. Section III establishes Lyapunov-based stability for the three-block scheme. Section IV develops the QUBO decompositions (three, micro, batched) and details the DVQE integration. Section V reports deterministic and stochastic case studies, and Section VI provides concluding remarks. **All detailed data, codes, and parameter values for**

all cases studied are available at [GitHub](#).

II. UC DECOMPOSITION WITH THREE-BLOCK ADMM

We formulate stochastic UC with renewable uncertainty and present a three-block ADMM decomposition that partitions the problem into continuous dispatch/reserves, binary QUBO, and proximal slack/dual updates.

A. Stochastic UC Formulation

Uncertainty in UC arises from variability in renewable generation and load forecasts. We adopt a scenario-based stochastic framework in which the net load at each time period is represented by a finite set of realizations with probabilities π_s . First-stage binary variables (y, u, v) are nonanticipative across scenarios, while second-stage continuous variables $(p, r^\uparrow, r^\downarrow)$ adapt to each scenario. The stochastic UC with N units is formulated as (1). The deterministic UC is recovered when $S = 1$.

$$\min \sum_{t=1}^T \sum_{i=1}^N \left(A_i y_{i,t} + S_i u_{i,t} + H_i v_{i,t} \right) + \sum_{s=1}^S \pi_s \sum_{t=1}^T \sum_{i=1}^N \left(B_i p_{i,t,s} + C_i p_{i,t,s}^2 \right) \quad (1a)$$

s.t.

$$\sum_{i=1}^N p_{i,t,s} = \tilde{L}_{t,s}, \quad \forall t, s \quad (1b)$$

$$P_i^{\min} y_{i,t} \leq p_{i,t,s} \leq P_i^{\max} y_{i,t}, \quad \forall i, t, s \quad (1c)$$

$$y_{i,t} - y_{i,t-1} = u_{i,t} - v_{i,t}, \quad \forall i, t \quad (1d)$$

$$u_{i,t} + v_{i,t} \leq 1, \quad \forall i, t \quad (1e)$$

$$\sum_{k=\max\{1, t-U_i+1\}}^t u_{i,k} \leq y_{i,t}, \quad \forall i, t \quad (1f)$$

$$\sum_{k=\max\{1, t-D_i+1\}}^t v_{i,k} \leq 1 - y_{i,t}, \quad \forall i, t \quad (1g)$$

$$p_{i,t,s} - p_{i,t-1,s} \leq RU_i y_{i,t-1} + SU_i u_{i,t}, \quad \forall i, t, s \quad (1h)$$

$$p_{i,t-1,s} - p_{i,t,s} \leq RD_i y_{i,t} + SD_i v_{i,t}, \quad \forall i, t, s \quad (1i)$$

$$0 \leq r_{i,t,s}^\uparrow \leq P_i^{\max} y_{i,t} - p_{i,t,s}, \quad \forall i, t, s \quad (1j)$$

$$0 \leq r_{i,t,s}^\downarrow \leq p_{i,t,s} - P_i^{\min} y_{i,t}, \quad \forall i, t, s \quad (1k)$$

$$r_{i,t,s}^\uparrow \leq RU_i \Delta \tau, \quad r_{i,t,s}^\downarrow \leq RD_i \Delta \tau, \quad \forall i, t, s \quad (1l)$$

$$\sum_{i=1}^N r_{i,t,s}^\uparrow \geq R_t^\uparrow, \quad \sum_{i=1}^N r_{i,t,s}^\downarrow \geq R_t^\downarrow, \quad \forall t, s \quad (1m)$$

$$y_{i,t}, u_{i,t}, v_{i,t} \in \{0, 1\}, \quad p_{i,t,s}, r_{i,t,s}^\uparrow, r_{i,t,s}^\downarrow \geq 0, \quad \forall i, t, s \quad (1n)$$

Power balance matches generation to net load as in (1b). Bounds, logic, minimum up/down times, and ramping are enforced in (1c)–(1i). Reserve deliverability and adequacy are enforced by (1j)–(1m). Domain constraints are given in (1n).

B. Three-Block ADMM Decomposition

To interface UC with binary QUBO solvers, we decouple integrality by introducing auxiliary binary proxies $Z = \{z^y, z^u, z^v\}$ and one-sided proximal slacks $\Xi = \{\xi^y, \xi^u, \xi^v\}$, replacing (1n) with consensus relations [18].

Consensus Constraints: For all i, t ,

$$0 \leq y_{i,t}, u_{i,t}, v_{i,t} \leq 1, \quad (2a)$$

$$y_{i,t} - z_{i,t}^y + \xi_{i,t}^y = 0, \quad (2b)$$

$$u_{i,t} - z_{i,t}^u + \xi_{i,t}^u = 0, \quad (2c)$$

$$v_{i,t} - z_{i,t}^v + \xi_{i,t}^v = 0, \quad (2d)$$

$$\xi_{i,t}^y, \xi_{i,t}^u, \xi_{i,t}^v \geq 0, \quad (2e)$$

$$z_{i,t}^y, z_{i,t}^u, z_{i,t}^v \in \{0, 1\}. \quad (2f)$$

Thus, (1n) is enforced indirectly via binary proxies and vanishing slacks at convergence. Z and Ξ are nonanticipative, while $(p_{i,t,s}, r_{i,t,s}^\uparrow, r_{i,t,s}^\downarrow)$ remain scenario dependent.

Augmented Lagrangian: Let $\Lambda = \{\lambda^y, \lambda^u, \lambda^v\}$ be the multipliers, $\rho_y, \rho_u, \rho_v > 0$ penalty parameters, and $\beta_y, \beta_u, \beta_v \geq 0$ proximal weights. Define the aggregate of all primal variables

$$\Delta := \{y, u, v, p, r^\uparrow, r^\downarrow\},$$

with (y, u, v) relaxed to $[0, 1]$ in Block 1. The augmented Lagrangian is

$$\begin{aligned} \mathcal{L}(\Delta, Z, \Xi, \Lambda) = & \sum_{t,i} (A_i y_{i,t} + S_i u_{i,t} + H_i v_{i,t}) \\ & + \sum_s \pi_s \sum_{t,i} (B_i p_{i,t,s} + C_i p_{i,t,s}^2) \\ & + \sum_{i,t} \left[\lambda_{i,t}^y (y_{i,t} - z_{i,t}^y + \xi_{i,t}^y) + \frac{\rho_y}{2} (y_{i,t} - z_{i,t}^y + \xi_{i,t}^y)^2 \right] \\ & + \sum_{i,t} \left[\lambda_{i,t}^u (u_{i,t} - z_{i,t}^u + \xi_{i,t}^u) + \frac{\rho_u}{2} (u_{i,t} - z_{i,t}^u + \xi_{i,t}^u)^2 \right] \\ & + \sum_{i,t} \left[\lambda_{i,t}^v (v_{i,t} - z_{i,t}^v + \xi_{i,t}^v) + \frac{\rho_v}{2} (v_{i,t} - z_{i,t}^v + \xi_{i,t}^v)^2 \right] \\ & + \sum_{i,t} \left[\frac{\beta_y}{2} (\xi_{i,t}^y)^2 + \frac{\beta_u}{2} (\xi_{i,t}^u)^2 + \frac{\beta_v}{2} (\xi_{i,t}^v)^2 \right], \end{aligned} \quad (3)$$

subject to (1b)–(1m).

ADMM Updates: At iteration k , three primal blocks and a dual update are performed:

a) *Block 1: Continuous update.:*

$$\Delta^{(k)} = \arg \min_{\Delta} \mathcal{L}(\Delta, Z^{(k-1)}, \Xi^{(k-1)}, \Lambda^{(k-1)}), \quad (4)$$

subject to (1b)–(1m) with the relaxation $(y, u, v) \in [0, 1]$. This is a convex QP in the relaxed (y, u, v) and the scenario-dependent $(p, r^\uparrow, r^\downarrow)$.

b) *Block 2: Binary update (QUBO).:*

$$Z^{(k)} = \arg \min_{Z \in \{0,1\}^{3NT}} \mathcal{L}(\Delta^{(k)}, Z, \Xi^{(k-1)}, \Lambda^{(k-1)}). \quad (5)$$

This decouples into independent 1-bit QUBOs:

$$q_{i,t}^{(y)} = -\lambda_{i,t}^{y,(k-1)} - \rho_y (y_{i,t}^{(k)} + \xi_{i,t}^{y,(k-1)}) + \frac{\rho_y}{2}, \quad (6)$$

$$q_{i,t}^{(u)} = -\lambda_{i,t}^{u,(k-1)} - \rho_u (u_{i,t}^{(k)} + \xi_{i,t}^{u,(k-1)}) + \frac{\rho_u}{2}, \quad (7)$$

$$q_{i,t}^{(v)} = -\lambda_{i,t}^{v,(k-1)} - \rho_v (v_{i,t}^{(k)} + \xi_{i,t}^{v,(k-1)}) + \frac{\rho_v}{2}, \quad (8)$$

Algorithm 1 Three-Block ADMM for Stochastic UC

- 1: **Initialize:** $(Z^{(0)}, \Xi^{(0)}, \Lambda^{(0)})$, penalty parameters, proximal weights, and tolerances. Set $k \leftarrow 1$.
 - 2: **while** stopping criterion not met **do**
 - 3: **Block 1:** Solve $\Delta^{(k)}$ by minimizing (3) subject to (1b)–(1m) with $(y, u, v) \in [0, 1]$.
 - 4: **Block 2:** Update $Z^{(k)}$ by solving the QUBO subproblems defined in (5).
 - 5: **Block 3:** Update $\Xi^{(k)}$ using (9)–(11) and project $\Xi^{(k)} \leftarrow \max\{\Xi^{(k)}, 0\}$.
 - 6: **Dual:** Update $\Lambda^{(k)}$ via (12)–(14).
 - 7: $k \leftarrow k + 1$.
 - 8: **end while**
 - 9: **Return:** $(\Delta^{(k)}, Z^{(k)}, \Xi^{(k)}, \Lambda^{(k)})$.
-

with $z_{i,t}^\bullet \in \arg \min_{z \in \{0,1\}} q_{i,t}^{(\bullet)} z$. These 1-bit baselines are later extended to multi-bit micro-/batched QUBOs (Section IV).

c) *Block 3: Slack update.*

$$\xi_{i,t}^{y,(k)} = -\frac{\lambda_{i,t}^{y,(k-1)} + \rho_y(y_{i,t}^{(k)} - z_{i,t}^{y,(k)})}{\beta_y + \rho_y}, \quad (9)$$

$$\xi_{i,t}^{u,(k)} = -\frac{\lambda_{i,t}^{u,(k-1)} + \rho_u(u_{i,t}^{(k)} - z_{i,t}^{u,(k)})}{\beta_u + \rho_u}, \quad (10)$$

$$\xi_{i,t}^{v,(k)} = -\frac{\lambda_{i,t}^{v,(k-1)} + \rho_v(v_{i,t}^{(k)} - z_{i,t}^{v,(k)})}{\beta_v + \rho_v}, \quad (11)$$

followed by projection $\xi_{i,t}^\bullet \leftarrow \max\{\xi_{i,t}^\bullet, 0\}$.

d) *Dual update.*

$$\lambda_{i,t}^{y,(k)} = \lambda_{i,t}^{y,(k-1)} + \rho_y(y_{i,t}^{(k)} - z_{i,t}^{y,(k)} + \xi_{i,t}^{y,(k)}), \quad (12)$$

$$\lambda_{i,t}^{u,(k)} = \lambda_{i,t}^{u,(k-1)} + \rho_u(u_{i,t}^{(k)} - z_{i,t}^{u,(k)} + \xi_{i,t}^{u,(k)}), \quad (13)$$

$$\lambda_{i,t}^{v,(k)} = \lambda_{i,t}^{v,(k-1)} + \rho_v(v_{i,t}^{(k)} - z_{i,t}^{v,(k)} + \xi_{i,t}^{v,(k)}). \quad (14)$$

Stopping rule. We terminate when the primal residuals $\|y - z^y + \xi^y\|_2$, $\|u - z^u + \xi^u\|_2$, $\|v - z^v + \xi^v\|_2$ and the dual residual $\rho\|(Z^{(k)} - Z^{(k-1)}) - (\Xi^{(k)} - \Xi^{(k-1)})\|_2$ fall below thresholds $\varepsilon_{\text{pri}}, \varepsilon_{\text{dual}}$, or when a maximum iteration cap is reached.

III. THREE-BLOCK ADMM CONVERGENCE ANALYSIS

We analyze the convergence of the three-block ADMM. We first state the conditions under which the method converges to a stationary solution, and then study stability via a Lyapunov-based argument.

A. Convergence Analysis

The three-block ADMM reformulation introduces consensus constraints that link the relaxed first-stage variables $(y, u, v) \in [0, 1]^{N \times T}$ to binary proxies $(z^y, z^u, z^v) \in \{0, 1\}^{N \times T}$ through nonnegative proximal slacks $(\xi^y, \xi^u, \xi^v) \geq 0$. At convergence, feasibility is recovered if the primal residuals

$$\begin{aligned} \|y - z^y + \xi^y\|_2 &\rightarrow 0, & \|u - z^u + \xi^u\|_2 &\rightarrow 0, \\ \|v - z^v + \xi^v\|_2 &\rightarrow 0, \end{aligned} \quad (15)$$

and the slacks diminish elementwise, implying $y = z^y$, $u = z^u$, $v = z^v$ with binary values. Since the continuous variables $(p, r^\uparrow, r^\downarrow)$ remain within the feasible region defined by (1b)–(1m), the limit points correspond to valid UC schedules with no relaxation gap.

Sufficient conditions. The three-block ADMM iterations converge to a stationary solution of the UC problem under the following conditions [18]:

- 1) **Coercivity.** The stochastic cost (1a) is coercive, since it contains convex quadratic terms $(C_i p_{i,t,s}^2)$ and all decision variables are bounded by capacity, ramping, and reserve constraints. Thus, the augmented Lagrangian $\mathcal{L}$ is bounded.
- 2) **Feasibility (range condition).** The reformulated UC satisfies

$$\text{Im}(y, u, v) \subseteq \text{Im}(z^y, z^u, z^v),$$

because every relaxed variable has a corresponding proxy and slack. Therefore, primal feasibility is attainable.

- 3) **Lipschitz continuity.** The updates in Block 1 and Block 3 are Lipschitz continuous with a constant $M = 1$. Block 2 is a finite QUBO minimization and its value mapping over $\{0, 1\}$ is trivially Lipschitz.
- 4) **Prox-regularity and lower semicontinuity.** The UC cost is a sum of convex terms $(A_i y_{i,t}, B_i p_{i,t,s}, C_i p_{i,t,s}^2, S_i u_{i,t}, H_i v_{i,t})$ and quadratic penalties in $\mathcal{L}$, which are Lipschitz differentiable and prox-regular. Hence, $\mathcal{L}$ is lower semicontinuous.
- 5) **Kurdyka–Łojasiewicz (KL) property.** Since the UC problem is defined by linear and quadratic relations with binary variables, it is semialgebraic. Therefore, $\mathcal{L}$ satisfies the KL inequality, ensuring the convergence of ADMM iterates to a stationary point.

Putting these together, if each block is solved and the penalty parameters ρ_y, ρ_u, ρ_v are chosen large, the three-block ADMM updates for stochastic UC converge to a stationary point of the original mixed-integer formulation.

B. Stability via Lyapunov Analysis

We assume throughout: (A1) the UC cost is coercive and all variables are bounded by (1b)–(1m); (A2) for fixed (Z, S, Λ) the Block 1 subproblem admits a minimizer and is strongly convex in its local variables; (A3) Block 2 is solved.

Proof 1: Define the consensus residuals at iteration k :

$$\begin{aligned} r^{y,k} &:= y^{(k)} - z^{y,(k)} + \xi^{y,(k)}, & r^{u,k} &:= u^{(k)} - z^{u,(k)} + \xi^{u,(k)}, \\ r^{v,k} &:= v^{(k)} - z^{v,(k)} + \xi^{v,(k)}, \end{aligned} \quad (16)$$

and stack them as $r^k := (r^{y,k}, r^{u,k}, r^{v,k})$. Let $\rho := \text{diag}(\rho_y I, \rho_u I, \rho_v I)$ and define the aggregated dual vector $\lambda^k := (\lambda^{y,(k)}, \lambda^{u,(k)}, \lambda^{v,(k)})$. The dual updates give

$$\lambda^{k+1} = \lambda^k + \rho r^{k+1} \implies r^{k+1} = \rho^{-1}(\lambda^{k+1} - \lambda^k). \quad (17)$$

Consider the Lyapunov candidate

$$V^k := \mathcal{L}(\Delta^{(k)}, Z^{(k)}, \Xi^{(k)}, \Lambda^{(k)}) + \frac{\kappa}{2} \|r^k\|_2^2, \quad (18)$$

where $\kappa \geq 0$. We show $V^{k+1} - V^k \leq 0$ and deduce convergence.

Step 1: Lyapunov difference expansion.:

$$V^{k+1} - V^k = \underbrace{(\mathcal{L}^{k+1}(\lambda^{k+1}) - \mathcal{L}^k(\lambda^k))}_{\mathcal{T}_{\mathcal{L}}} + \frac{\kappa}{2}(\|r^{k+1}\|^2 - \|r^k\|^2). \quad (19)$$

Step 2: Augmented Lagrangian change.: Split $\mathcal{T}_{\mathcal{L}}$ into a dual-shift part (changing Λ with fixed primals) and block updates (fixed Λ):

$$\begin{aligned} \mathcal{T}_{\mathcal{L}} &= [\mathcal{L}^{k+1}(\lambda^{k+1}) - \mathcal{L}^{k+1}(\lambda^k)] + [\mathcal{L}^{k+1}(\lambda^k) - \mathcal{L}^k(\lambda^k)] \\ &= \langle \lambda^{k+1} - \lambda^k, r^{k+1} \rangle + \left(\mathcal{L}(\Delta^{(k+1)}, Z^{(k+1)}, \Xi^{(k+1)}, \Lambda^{(k)}) \right. \\ &\quad \left. - \mathcal{L}(\Delta^{(k)}, Z^{(k)}, \Xi^{(k)}, \Lambda^{(k)}) \right). \end{aligned} \quad (20)$$

Using (17), the dual-shift term equals $\langle \rho r^{k+1}, r^{k+1} \rangle = \|\rho^{1/2} r^{k+1}\|^2$.

Step 3: Descent by blocks at fixed duals.: By (A2) and optimality of Block 1,

$$\begin{aligned} \mathcal{L}(\Delta^{(k+1)}, Z^{(k)}, \Xi^{(k)}, \Lambda^{(k)}) &\leq \mathcal{L}(\Delta^{(k)}, Z^{(k)}, \Xi^{(k)}, \Lambda^{(k)}) \\ &\quad - \sigma_1 \|\Delta^{(k+1)} - \Delta^{(k)}\|^2. \end{aligned} \quad (21)$$

By (A3) for Block 2,

$$\mathcal{L}(\Delta^{(k+1)}, Z^{(k+1)}, \Xi^{(k)}, \Lambda^{(k)}) \leq \mathcal{L}(\Delta^{(k+1)}, Z^{(k)}, \Xi^{(k)}, \Lambda^{(k)}). \quad (22)$$

For Block 3, the proximal quadratic update (with projection) yields

$$\begin{aligned} \mathcal{L}(\Delta^{(k+1)}, Z^{(k+1)}, \Xi^{(k+1)}, \Lambda^{(k)}) &\leq \\ \mathcal{L}(\Delta^{(k+1)}, Z^{(k+1)}, \Xi^{(k)}, \Lambda^{(k)}) &- \sigma_2 \|\Xi^{(k+1)} - \Xi^{(k)}\|^2. \end{aligned} \quad (23)$$

Step 4: Collect terms.: Substituting (20) and (23) into (19):

$$\begin{aligned} V^{k+1} - V^k &\leq \|\rho^{1/2} r^{k+1}\|^2 - \sigma_1 \|\Delta^{(k+1)} - \Delta^{(k)}\|^2 \\ &\quad - \sigma_2 \|\Xi^{(k+1)} - \Xi^{(k)}\|^2 + \frac{\kappa}{2}(\|r^{k+1}\|^2 - \|r^k\|^2) \\ &\leq \left(\lambda_{\max}(\rho) + \frac{\kappa}{2} \right) \|r^{k+1}\|^2 - \frac{\kappa}{2} \|r^k\|^2 \\ &\quad - \sigma_1 \|\Delta^{(k+1)} - \Delta^{(k)}\|^2 - \sigma_2 \|\Xi^{(k+1)} - \Xi^{(k)}\|^2. \end{aligned} \quad (24)$$

Step 5: Choice of κ and descent.: Choose $\kappa \geq 2\lambda_{\max}(\rho)$ so that $\lambda_{\max}(\rho) + \frac{\kappa}{2} \leq \kappa$. Then (24) becomes

$$\begin{aligned} V^{k+1} - V^k &\leq \kappa \|r^{k+1}\|^2 - \frac{\kappa}{2} \|r^k\|^2 - \sigma_1 \|\Delta^{(k+1)} - \Delta^{(k)}\|^2 \\ &\quad - \sigma_2 \|\Xi^{(k+1)} - \Xi^{(k)}\|^2. \end{aligned} \quad (25)$$

Summing over k and using that V^k is bounded below (coercivity, (A1)) yields

$$\|\Delta^{(k+1)} - \Delta^{(k)}\| \rightarrow 0, \quad \|\Xi^{(k+1)} - \Xi^{(k)}\| \rightarrow 0, \quad \|r^k\| \rightarrow 0,$$

by a standard telescoping argument. Hence V^k is non-increasing, and the iterates converge, with binary feasibility recovered via vanishing slacks.

IV. QUBO DECOMPOSITIONS AND DVQE INTEGRATION

We formulate the Block 2 binary update as a quadratic unconstrained binary optimization (QUBO) and develop three decomposition strategies—(i) type-specific (commitment/startup/shutdown), (ii) micro (per-unit, per-time), and (iii) batched block-diagonal—to create sub-QUBOs. We then introduce a distributed VQE solver (DVQE) with an accept-if-better safeguard to execute these sub-QUBOs efficiently across multiple quantum devices in a distributed quantum–classical workflow.

Within the three-block ADMM structure, the discrete subproblem (Block 2) isolates the auxiliary binary proxies $Z = \{z^y, z^u, z^v\}$, while all continuous and relaxed variables are handled in Block 1 and consensus is enforced in Block 3. At iteration k , the coefficients of Z are as (6)–(8) with $z_{i,t}^{\bullet} \in \arg \min_{z \in \{0,1\}} q_{i,t}^{\bullet} z$.

Although the QUBO formulation is straightforward, it presents two main challenges. The first is that the binary variables of each unit are not independent. Startup, shutdown, and commitment decisions are logically coupled through minimum up/down time conditions and other temporal constraints. These couplings are not directly visible in the linear objective above because they have been shifted into Block 1 and the consensus conditions. To ensure consistency of UC schedules, Block 2 must eventually include additional quadratic penalties that reintroduce these local dependencies. The second challenge is scalability. For large-scale UC problems, the monolithic QUBO corresponding to Block 2 can contain thousands of binary variables.

A further consideration is the target hardware environment. The purpose of isolating Block 2 is to cast UC in binary form and make it compatible with hybrid quantum–classical algorithms. To be effective, the QUBO must be large enough to demonstrate a potential advantage of quantum solvers over purely classical methods, but not so large that it overwhelms the limited qubits and circuit depth of current processors. Balancing these requirements motivates the decomposition strategies developed in the following subsections.

A. QUBO Formulation

1) Three QUBOs: The initial form of Block 2 shows that all binary proxies appear independently in the augmented Lagrangian. This independence motivates a first decomposition strategy in which QUBO is partitioned into three smaller subproblems, each corresponding to a distinct type of binary variable. Specifically, all auxiliary commitment binaries $\{z_{i,t}^y\}$ are grouped into one QUBO, all startup binaries $\{z_{i,t}^u\}$ form a second QUBO, and all shutdown binaries $\{z_{i,t}^v\}$ form a third QUBO. In this way, the burden of solving a very large QUBO in one attempt is avoided, while no physical or mathematical structure of the UC model is violated. The three sub-QUBOs remain fully consistent with the consensus framework, and together, they constitute the complete Block 2 update.

Formally, the decomposition yields

$$Z^{(k)} = \arg \min_{Z \in \{0,1\}^{3NT}} \mathcal{L}(\Delta^{(k)}, Z, \Xi^{(k-1)}, \Lambda^{(k-1)}) \quad (26)$$

$$= \arg \min_{\substack{z^y \in \{0,1\}^{NT} \\ z^u \in \{0,1\}^{NT} \\ z^v \in \{0,1\}^{NT}}} \left(\mathcal{L}_y(z^y) + \mathcal{L}_u(z^u) + \mathcal{L}_v(z^v) \right), \quad (27)$$

where $\mathcal{L}_y$, $\mathcal{L}_u$, and $\mathcal{L}_v$ denote the components of the augmented Lagrangian associated with the three classes of binary variables. Each subproblem is thus a self-contained QUBO with a linear objective over its respective binary set.

This decomposition provides two main benefits. First, the size of each QUBO is reduced by roughly a factor of three, which lowers the computational complexity of both classical and hybrid solvers. Second, the three QUBOs can be solved in parallel, thereby preserving the interpretation of Block 2 as a single update while reducing wall-clock time per ADMM iteration.

2) *Micro QUBOs*: The independent decomposition into three type-specific QUBOs reduces problem size but does not restore the logical dependencies that exist among commitment, startup, and shutdown variables of the same unit. To better capture these local couplings while maintaining the tractability of Block 2, we introduce a micro-level decomposition. The guiding idea is to reconsider, at the level of auxiliary proxies, the same relations that the relaxed variables (y, u, v) must satisfy in Block 1. Since these dependencies occur only within each unit individually, a natural approach is to construct one micro-QUBO per unit-time pair.

At each (i, t) we bundle the three proxies $z_{i,t}^y, z_{i,t}^u, z_{i,t}^v \in \{0, 1\}$ into a three-dimensional binary vector.

$$z_{i,t} := [z_{i,t}^y \quad z_{i,t}^u \quad z_{i,t}^v] \in \{0, 1\}^3, c_{i,t} := \begin{bmatrix} q_{i,t}^{(y)} & q_{i,t}^{(u)} & q_{i,t}^{(v)} \end{bmatrix},$$

where the entries of $c_{i,t}$ are the linear coefficients inherited from the augmented Lagrangian. We then define a local quadratic energy

$$E_{i,t}(z_{i,t}) = z_{i,t}^\top Q_{i,t} z_{i,t} + c_{i,t}^\top z_{i,t} + \text{const}, \quad (28)$$

with $Q_{i,t} \in \mathbb{R}^{3 \times 3}$ symmetric. In contrast to the type-specific decomposition, here we deliberately augment the base linear terms with small quadratic penalties that reintroduce period- t logic. These penalties are purely local and maintain independence across different units, allowing all microproblems to be solved in parallel.

To construct $Q_{i,t}$, we let the relaxed Block 1 outputs at (i, t) be $\hat{y} := y_{i,t}^{(k)}$, $\hat{u} := u_{i,t}^{(k)}$, and $\hat{v} := v_{i,t}^{(k)}$, with a reference $\eta := y_{i-1}^{\text{ref}}$ from the previous period. With nonnegative penalty weights $\gamma_c, \gamma_{ss}, \gamma_{u \rightarrow y}, \gamma_{v \rightarrow \bar{y}}, \gamma_y, \gamma_u, \gamma_v$, we define the penalty

$$\begin{aligned} \Pi_{i,t}(z_{i,t}) := & \gamma_c (z_{i,t}^y - z_{i,t}^u + z_{i,t}^v - \eta)^2 + \gamma_{ss} z_{i,t}^u z_{i,t}^v \\ & + \gamma_{u \rightarrow y} z_{i,t}^u (1 - z_{i,t}^y) + \gamma_{v \rightarrow \bar{y}} z_{i,t}^v z_{i,t}^y \\ & + \gamma_y (z_{i,t}^y - \hat{y})^2 + \gamma_u (z_{i,t}^u - \hat{u})^2 + \gamma_v (z_{i,t}^v - \hat{v})^2. \end{aligned} \quad (29)$$

The first term softly enforces the logical relation (1d). The second discourages simultaneous startup and shutdown. The

third and fourth encode start implies on and shutdown implies off. The last three anchor the proxies to the relaxed outputs from Block 1. These additions are incorporated by setting $E_{i,t} \leftarrow E_{i,t} + \Pi_{i,t}$.

Because binary variables satisfy $z^2 = z$, the penalty (29) contributes explicit updates to $(Q_{i,t}, c_{i,t})$. The linear terms are updated as

$$c_{i,t}^{(y)} \leftarrow c_{i,t}^{(y)} + \gamma_c(1 - 2\eta) + \gamma_y(1 - 2\hat{y}), \quad (30)$$

$$c_{i,t}^{(u)} \leftarrow c_{i,t}^{(u)} + \gamma_c(1 + 2\eta) + \gamma_u(1 - 2\hat{u}) + \gamma_{u \rightarrow y}, \quad (31)$$

$$c_{i,t}^{(v)} \leftarrow c_{i,t}^{(v)} + \gamma_c(1 - 2\eta) + \gamma_v(1 - 2\hat{v}), \quad (32)$$

while the pairwise couplings are

$$(Q_{i,t})_{y,u} \leftarrow (Q_{i,t})_{y,u} - 2\gamma_c - \gamma_{u \rightarrow y}, \quad (33)$$

$$(Q_{i,t})_{y,v} \leftarrow (Q_{i,t})_{y,v} + 2\gamma_c + \gamma_{v \rightarrow \bar{y}}, \quad (34)$$

$$(Q_{i,t})_{u,v} \leftarrow (Q_{i,t})_{u,v} - 2\gamma_c + \gamma_{ss}. \quad (35)$$

We initialize $Q_{i,t} = 0_{3 \times 3}$ and $c_{i,t} = [q_{i,t}^{(y)}, q_{i,t}^{(u)}, q_{i,t}^{(v)}]^\top$, then apply these updates. The Block 2 update thus decomposes into independent micro problems

$$z_{i,t}^* = \arg \min_{z_{i,t} \in \{0,1\}^3} z_{i,t}^\top Q_{i,t} z_{i,t} + c_{i,t}^\top z_{i,t}. \quad (36)$$

This approach restores local binary couplings and accelerates ADMM convergence.

3) *Batched QUBOs*: The micro-QUBO decomposition accelerates ADMM by introducing local couplings and reducing the size of the Block 2 subproblems compared to the three independent QUBOs. However, this decomposition alone does not exploit quantum efficiency. Each micro-QUBO involves only three binary variables, so even classical brute force can enumerate its eight possible assignments in negligible time. Moreover, in large-scale UC instances with N units, T periods, and multiple ADMM iterations, the total number of micro problems becomes $3NT \times (\text{iterations})$, which may reach millions. Solving such a volume of micro problems in parallel quickly exceeds computational resources, while solving them sequentially becomes prohibitively time-consuming.

The remedy lies in re-aggregating the micro problems into carefully chosen batches. Instead of solving all NT micro-QUBOs independently, we group them into k disjoint batches, with $3 < k < N$. Each batch is large enough to justify the use of quantum solvers but small enough to remain within the current qubit limitations of near-term quantum hardware. In addition, the grouping allows us to balance solver effort by assigning harder micro problems to different batches. A greedy distribution algorithm ensures that batches have roughly equal difficulty, allowing all workers to complete in comparable times. This approach thus interpolates between the two earlier decompositions: it yields more subproblems than the three-QUBO strategy, while preserving the local couplings absent in the fully independent decomposition, and it reduces the computational overhead of the micro-QUBO strategy by replacing millions of subproblems.

Although each micro-QUBO has only three variables, its solver difficulty depends on the local energy landscape of the

problem. To balance batches, we compute a simple hardness score $\text{Hard}_{i,t}$ from $(Q_{i,t}, c_{i,t})$. Several complementary indicators are combined:

$$\text{Hard}_{i,t} = w_1 \frac{1}{\Delta_{i,t} + \eta} + w_2 \frac{g_{i,t}^{(\varepsilon)} - 1}{7} + w_3 \mathbb{I}_{\text{frust},i,t} + w_4 \frac{r_{i,t}}{1 + r_{i,t}} + w_5 \frac{\log_{10}(M_{i,t}/m_{i,t})}{4}, \quad (37)$$

where the terms are:

- $\Delta_{i,t}$: the energy gap between the best and second-best assignment (larger gap = easier);
- $g_{i,t}^{(\varepsilon)}$: number of assignments within ε of the optimum (larger = harder);
- $\mathbb{I}_{\text{frust},i,t}$: frustration indicator for the 3-variable coupling triangle;
- $r_{i,t}$: ratio of total coupling strength to total linear field strength;
- $\log_{10}(M_{i,t}/m_{i,t})$: dynamic range of coefficients, with $M_{i,t}$ and $m_{i,t}$ the maximum and minimum nonzero absolute entries of $Q_{i,t}$ and $c_{i,t}$.

The weights $w_1, \dots, w_5$ are user-tunable (initially equal), and $\eta > 0$ avoids division by zero. This score is inexpensive to compute and serves as a proxy for solver effort.

Given scores $\text{Hard}_{i,t}$, we partition the set $\{(i, t)\}$ into k disjoint batches $B_1, \dots, B_k$ with similar total hardness. A simple greedy bin-packing procedure suffices. Firstly, sort all pairs (i, t) by decreasing $\text{Hard}_{i,t}$ and secondly, assign each next element to the batch whose current hardness sum is smallest.

This balances the workload across batches. As a practical refinement, one may enforce that all time periods of the same unit $(i, 1), \dots, (i, T)$ are assigned to the same batch, preserving temporal coherence without overlap. Since batches are non-overlapping and each micro-QUBO couples only its own triplet $(z_{i,t}^y, z_{i,t}^u, z_{i,t}^v)$, batching preserves the independence of subproblems. Thus, ADMM convergence is unaffected.

B. DVQE Solver

As discussed earlier, the size of monolithic or even moderately decomposed QUBOs can exceed the qubit capacities of current NISQ devices. While UC admits natural decompositions across units and time, many other QUBO formulations are not inherently separable, which further exacerbates scalability challenges. As problem sizes grow, relying on a single QPU becomes infeasible. A promising direction to overcome this barrier is distributed quantum computing, where multiple smaller QPUs collaborate to solve a larger optimization task. The DVQE [22] is one such hybrid quantum-classical framework. DVQE extends the standard VQE algorithm by partitioning a QUBO across multiple quantum devices. Each QPU evaluates a subset of parameterized quantum circuits, while a classical coordination layer aggregates partial results, updates the global variational parameters, and drives the optimization forward. The TeleGate protocol enables entangling operations between qubits on different devices, creating the illusion of a single larger QPU.

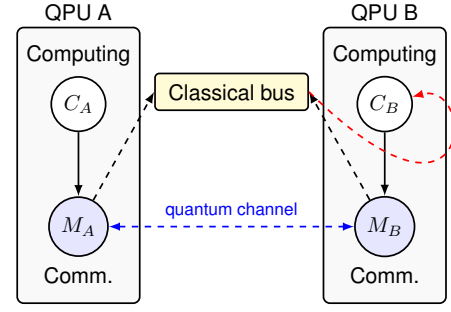


Fig. 2: Illustration of the TeleGate protocol.

When working with DVQE, the user only provides the quadratic and linear coefficients of the QUBO. Internally, these coefficients are mapped into an Ising Hamiltonian written in terms of Pauli-Z operators as

$$H = \sum_{i=1}^n h_i Z_i + \sum_{i < j} J_{ij} Z_i Z_j, \quad (38)$$

where Z_i is the Pauli-Z operator acting on qubit i , h_i is the linear field, and J_{ij} encodes pairwise couplings. The ground state of this Hamiltonian corresponds to the optimal QUBO solution. This automatic transformation allows DVQE to remain solver-agnostic with respect to the problem input.

After the Hamiltonian is formed, qubits are distributed across QPUs using a greedy allocation algorithm that balances computational load while respecting the coupling structure. Each QPU hosts computing qubits for local gates and reserves at least one communication qubit that is used to entangle with other devices. Local operations such as single-qubit rotations are executed directly within a QPU, while inter-QPU entanglement is enabled by the TeleGate protocol. TeleGate combines local entanglement, quantum communication, and classical coordination. A computing qubit is first entangled with its local communication qubit. Communication qubits of two different QPUs are then entangled through a shared channel. Finally, a classical message transmits the outcome, allowing the target QPU to apply the correct conditional operation. Through this process, several small QPUs and the classical bus collectively behave as if they were one larger processor, making it possible to run distributed ansatz circuits that preserve coherence across devices as shown in Fig. 2.

At the heart of VQE and DVQE lies the construction of an ansatz, a parameterized quantum circuit designed to approximate the ground state of the Hamiltonian. The ansatz prepares a quantum state $|\psi(\theta)\rangle$ using layers of single-qubit rotations, typically R_y and R_z , interleaved with CNOT gates that introduce entanglement. By stacking multiple layers, the circuit gains expressive power to capture increasingly complex correlations. Once the state is prepared, repeated measurements provide an estimate of the Hamiltonian expectation value, which serves as the cost function to be minimized.

The optimization loop is closed by a classical optimizer, which updates the variational parameters based on measurement outcomes. DVQE employs ADAM [26], which maintains

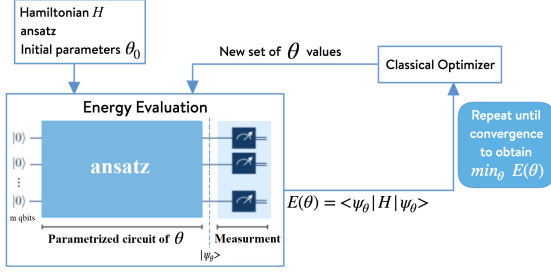


Fig. 3: Overview of the VQE workflow for solving the Hamiltonian.

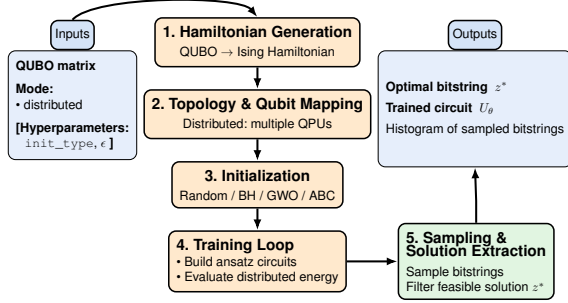


Fig. 4: Workflow of the DVQE algorithm.

exponentially weighted averages of gradients and squared gradients and uses bias correction to stabilize the updates. Compared with gradient-free optimizers, ADAM achieves faster convergence and greater robustness under noisy measurements, which is particularly valuable in distributed settings. The prepare–measure–update cycle continues until convergence or until a maximum number of iterations is reached. After convergence, the optimized ansatz is executed to produce a histogram of candidate bitstrings, from which the best-performing solution is selected as the binary vector.

In practice, DVQE has been implemented in the `raiselab` package, which supports distributed circuit execution and incorporates advanced initialization strategies such as black hole, gray wolf, and artificial bee colony metaheuristics to accelerate convergence. Fig. 3 illustrates the general workflow of the DVQE algorithm, and Fig. 4 summarizes the DVQE process, including Hamiltonian generation, qubit distribution, initialization, training, and solution extraction.

In principle, DVQE can solve the full Block 2 QUBO in one step; however, the implementation in [22] currently relies on a Qiskit simulator on classical CPUs, which limits scalability. On real distributed hardware, DVQE would naturally scale to larger UC instances. We validate DVQE on small cases and recommend its use on decomposed QUBOs for present-day feasibility. While micro-QUBOs are too small for quantum gain, DVQE is well-suited for batched QUBOs, where efficiency and scalability are both achieved.

V. CASE STUDIES AND RESULTS

Across all D²-UC experiments, we fix DVQE hyperparameters for consistency: depth $d=2$, learning rate $\eta=0.1$, and at most 100 iterations. DVQE is run in distributed mode with disjoint 3-qubit registers ($[3, 3, 3, \dots, 3]$).

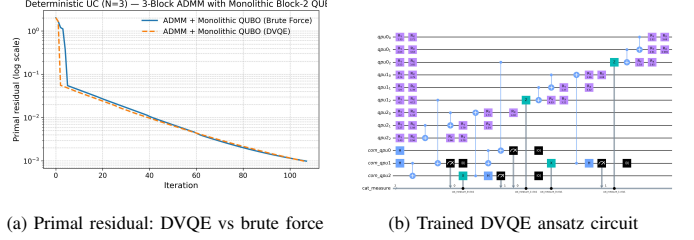


Fig. 5: Validation on three-unit UC ($T=6$). DVQE matches brute-force optimality while reducing simulation effort.

An accept-if-better safeguard ensures Block 2 updates monotonically reduce the local augmented Lagrangian, preventing oscillations. This configuration balances stability and runtime, and is used uniformly for the three-QUBO, micro-QUBO, and batched QUBO experiments. All detailed data, codes, and parameter values for all cases studied here with D²-UC framework are available at GitHub.

To ensure a fair comparison, we used identical ADMM settings and initialization in every case even for three-unit UC system: $\rho_y=\rho_u=\rho_v=9 \times 10^5$, $\beta_y=\beta_u=\beta_v=2 \times 10^6$, $\epsilon=10^{-3}$, $\text{max_iter}=4000$, micro-QUBO weights $\gamma_c=0.20\rho_y$, $\gamma_{ss}=0.10\rho_y$, $\gamma_{u \rightarrow y}=\gamma_{v \rightarrow y}=0.05\rho_y$, and anchors $\gamma_y=\gamma_u=\gamma_v=0.10\rho_y$. Initialization follows: $p_0=20$, $y_0=1$, relaxed (y, u, v) as $(0.5, 0, 0)$, p tiled from L/N , zero slacks/duals, and zero reserves.

A. Three-Unit UC

We first validate the framework on a three-unit UC system with a planning horizon of $T=6$ hours. For this illustrative case, the full Block 2 QUBO is solved directly using DVQE without any decomposition, to demonstrate DVQE’s ability to handle the binary optimization task as a whole. For benchmarking, the same problem is solved via exhaustive brute-force enumeration.

As shown in Fig. 5a, the primal residual trajectories of both brute force and DVQE converge to the same threshold, confirming that DVQE attains the globally optimal binary schedule. Importantly, DVQE requires substantially less simulation time than brute force, highlighting its potential as a scalable hybrid solver. The trained ansatz circuit from the final ADMM iteration is also depicted in Fig. 5b. The optimized parameters reproduce the optimal commitment bitstring with high sampling probability, verifying the effectiveness of the quantum–classical training loop.

B. Five-Unit UC

We use a five-unit UC system over $T=6$ hours, testing both deterministic and stochastic formulations. In each case, Block 2 is solved by brute force and DVQE for comparison. DVQE consistently reproduces the same binary schedules and objective values as brute force, but with lower runtimes.

For the deterministic UC, three decomposition strategies are compared. The three-QUBO decomposition (separating y, u, v) converges to feasible schedules but requires more ADMM iterations due to weakened logical couplings. The micro-QUBO

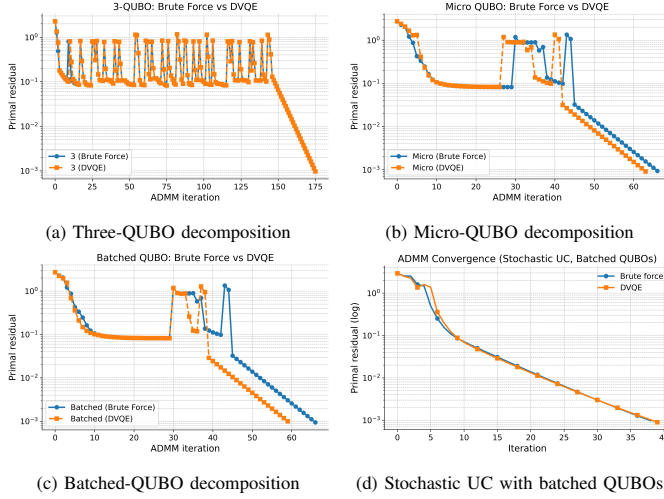


Fig. 6: Primal residual trajectories for five-unit UC under different QUBO decompositions, solved by brute force and DVQE.

decomposition restores local dependencies and accelerates convergence, but produces a large number of trivial QUBOs, which increases overall simulation burden. The batched-QUBO strategy (three batches) strikes the best balance: it improves convergence relative to the three-QUBO case and avoids the overhead of solving thousands of microproblems, yielding batch sizes compatible with near-term quantum devices.

Figs. 6a–6c show the primal residual trajectories for all decomposition strategies, solved by brute force and DVQE. In the micro- and batched-QUBO cases, DVQE converges faster than brute force, while in the three-QUBO case, the residual trajectories coincide exactly. Across all runs, feasibility is preserved: load balance, capacity, ramping, and reserves are satisfied; consensus is achieved; and all slack variables converge to zero. Identical ADMM hyperparameters and initialization are applied to ensure fair comparison. Table I reports the realized deterministic dispatch.

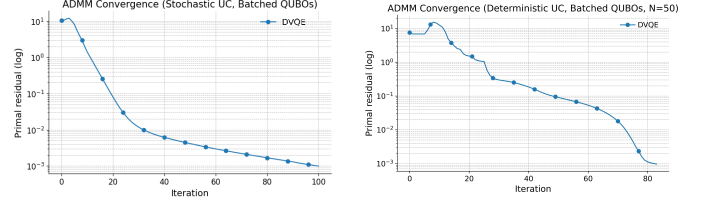
We next extend the study to the stochastic UC formulation under renewable uncertainty. We consider a scenario-based model with $S = 4$ scenarios, $N = 5$ generating units, and a planning horizon of $T = 6$ periods. For this stochastic UC, we employ the batched QUBO decomposition and solve the resulting QUBOs using both brute-force enumeration and DVQE. Fig. 6d shows the primal-residual trajectories for both brute force and DVQE under the batched QUBO decomposition. The residual curves coincide exactly, resulting in overlapping lines that confirm identical convergence behavior. Across both runs, we verified feasibility: power balance, capacity limits, ramping constraints, and reserve deliverability are all satisfied. Consensus is achieved, and all slack variables (s^y, s^u, s^v) converge to zero. We use the same ADMM hyperparameters as in the deterministic UC case. Table II reports the realized dispatch across all units and scenarios, showing that load is met exactly at each time period in each scenario.

C. Medium- and Large-Scale UC

We extend the study to larger deterministic UC problems. Two test cases are considered: a medium-scale system with

TABLE I: Deterministic UC ($N=5$, $T=6$): Dispatch vs. Demand

t	Unit Dispatch p_i (MW)					$\sum_i p_{i,t}$	L_t
	p_1	p_2	p_3	p_4	p_5		
1	0	44.24	52.76	0	53	150	150
2	0	20.00	70	0	80	170	170
3	0	35.02	60	0	84.98	180	180
4	0	40.02	35	0	84.98	160	160
5	0	55.02	0	0	84.98	140	140
6	0	45.02	0	0	84.98	130	130



(a) 20-unit UC ($T=24$): DVQE vs brute force (b) 50-unit UC ($T=24$): DVQE vs brute force

Fig. 7: Primal residual convergence for medium- and large-scale deterministic UC. DVQE and brute force coincide in residual accuracy, while DVQE runs faster in simulation time.

$N=20$ generating units and a large-scale system with $N=50$ units, both scheduled over a horizon of $T=24$ hours. For both cases, Block 2 is solved using the batched QUBO decomposition. Each batch is executed by DVQE.

Fig. 7 shows the primal-residual trajectories for the two systems. Tables III and IV report partial dispatch schedules for the medium- and large-scale experiments. The complete dispatch data for all units and time periods are available in the open-source repository GitHub.

VI. CONCLUSION

This paper proposed D²-UC, a quantum-ready reformulation of the UC problem using a three-block ADMM decomposition that isolates all binary decisions in a QUBO subproblem. To make this block compatible with near-term hardware, we advanced a progression of strategies: from a monolithic QUBO, to three type-specific QUBOs, to micro-QUBOs per unit–time pair, and finally to batched micro-QUBOs that reduce a large number of subproblems to a fixed number of solver-ready instances. A distributed quantum solver was presented to solve the decomposed QUBOs in a distributed quantum setting. An accept-if-better safeguard was also integrated with the DVQE to stabilize hybrid updates. Case studies on deterministic and stochastic UC demonstrated that the framework delivers feasible schedules, faster ADMM convergence, and QUBO sizes well suited to current hybrid and distributed quantum solvers.

REFERENCES

- [1] N. P. Padhy, “Unit commitment—a bibliographical survey,” *IEEE Transactions on power systems*, vol. 19, no. 2, pp. 1196–1205, 2004.
- [2] 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.
- [3] M. Asensio and J. Contreras, “Stochastic unit commitment in isolated systems with renewable penetration under cvar assessment,” *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1356–1367, 2015.

TABLE II: Stochastic UC ($N=5$, $T=6$, $S=4$): Dispatch vs. Demand Across Scenarios

Scenario 1							
t	Unit Dispatch p_i (MW)					$\sum_i p_{i,t}$	L_t
	p_1	p_2	p_3	p_4	p_5		
1	50	20	45.00	0	25	140	140
2	50.02	20	69.99	0	20	160	160
3	60	20	69.99	0	20.02	170	170
4	40	20	69.99	0	20.02	150	150
5	0	20	69.99	0	40.02	130	130
6	0	20	69.99	0	30.02	120	120

Scenario 2							
t	Unit Dispatch p_i (MW)					$\sum_i p_{i,t}$	L_t
	p_1	p_2	p_3	p_4	p_5		
1	50	20	45	0	35	150	150
2	60	20	69.99	0	20.02	170	170
3	60	20	69.99	0	30.02	180	180
4	40	20	69.99	0	30.02	160	160
5	0	20	69.99	0	50.02	140	140
6	0	20	69.99	0	40.02	130	130

Scenario 3							
t	Unit Dispatch p_i (MW)					$\sum_i p_{i,t}$	L_t
	p_1	p_2	p_3	p_4	p_5		
1	50	20	45	0	45	160	160
2	60	20	69.99	0	30.02	180	180
3	60	20	69.99	0	40.02	190	190
4	40	20	69.99	0	40.02	170	170
5	0	20	69.99	0	60.02	150	150
6	0	20	69.99	0	50.02	140	140

Scenario 4							
t	Unit Dispatch p_i (MW)					$\sum_i p_{i,t}$	L_t
	p_1	p_2	p_3	p_4	p_5		
1	50	25	45	0	50	170	170
2	60	20	69.99	0	40.02	190	190
3	60	20	69.99	0	50.02	200	200
4	40	20	69.99	0	50.02	180	180
5	0	20	69.99	0	70.02	160	160
6	0	20	69.99	0	60.02	150	150

TABLE III: Partial Dispatch for Medium-Scale UC ($N=20$, $T=24$). Ten representative units over first six hours.

t	Unit Dispatch p_i (MW)										$\sum_i p_{i,t}$	L_t
	p_1	p_3	p_4	p_6	p_7	p_{10}	p_{12}	p_{15}	p_{18}	p_{20}		
1	0	70	0	70	85	60	0	85	70	80	900	900
2	0	100	0	40	40	30	0	40	100	110	950	950
3	33.3	70	0	30	35	30	50	35	130	70	1000	1000
4	76.7	40	20	30	35	30	90	35	130	40	1050	1050
5	100	30	20	30	35	30	100	35	140	40	1100	1100
6	100	43.1	20	30	35	30	100	35	140	40	1150	1150

TABLE IV: Partial Dispatch for UC ($N=50$, $T=24$). Ten representative units over first six hours.

t	Unit Dispatch p_i (MW)										$\sum_i p_{i,t}$	L_t
	p_2	p_5	p_9	p_{11}	p_{14}	p_{20}	p_{25}	p_{33}	p_{42}	p_{50}		
1	0	66.7	17.9	0	0	51.5	66.8	0	0	52.8	900	900
2	0	66.7	41	0	0	44.1	66.8	0	0	52.8	950	950
3	24.8	42.9	17.9	0	27.9	19.6	66.8	0	13.8	68.4	1000	1000
4	49.7	19.1	19	0	51.4	19.6	66.8	15.8	16.9	53.4	1050	1050
5	52	19.1	19	0	46	19.6	66.8	17.7	16.9	47.9	1100	1100
6	39.1	19.1	19	0	34.6	19.6	66.8	17.7	16.9	36	1150	1150

- [4] R. Mahroo and A. Kargarian, "Learning infused quantum-classical distributed optimization technique for power generation scheduling," *IEEE Transactions on Quantum Engineering*, vol. 4, pp. 1–14, 2023.
- [5] N. Nikmehr, P. Zhang, and M. A. Bragin, "Quantum distributed unit commitment: An application in microgrids," *IEEE transactions on power systems*, vol. 37, no. 5, pp. 3592–3603, 2022.
- [6] H. Dong, L. Wu, J. Zhu, S. Li, Z. Liang, H. Yang, and C.-y. Chung, "A data-driven cost budget satisficing model for unit commitment under solar power uncertainty," *IEEE Transactions on Power Systems*, 2025.
- [7] A. B. Javadi, A. Kargarian, and M. Naraghi-Pour, "Learning constraint surrogate model for two-stage stochastic unit commitment," *arXiv preprint arXiv:2509.10246*, 2025.
- [8] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.
- [9] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, 2018.
- [10] P. Pareek, A. Jayakumar, C. Coffrin, and S. Misra, "Limitations of fault-tolerant quantum linear system solvers for quantum power flow," *IEEE Transactions on Power Systems*, 2025.
- [11] T. Morstyn, "Annealing-based quantum computing for combinatorial optimal power flow," *IEEE Transactions on Smart Grid*, vol. 14, no. 2, pp. 1093–1102, 2022.
- [12] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J.

- Love, A. Aspuru-Guzik, and J. L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature communications*, vol. 5, no. 1, p. 4213, 2014.
- [13] A. J. Conejo, E. Castillo, R. Minguez, and R. Garcia-Bertrand, *Decomposition techniques in mathematical programming: engineering and science applications*. Springer, 2006.
- [14] C. Zhang, L. Yang, and J. Jian, "Two-stage fully distributed approach for unit commitment with consensus ADMM," *Electric Power Systems Research*, vol. 181, p. 106180, 2020.
- [15] M. Hasanzadeh and A. Kargarian, "ADMM enhancement techniques for distributed optimal power flow," *IEEE Transactions on Power Systems*, 2025.
- [16] Q. Shi, Y. Zhu, K. Fan, R. Cheng, J. Shen, and W. Liu, "Two-stage linearization of frequency nadir constraint for unit commitment," *IEEE Transactions on Power Systems*, 2025.
- [17] X. Han, Z. Li, W. Gu, and M. Shahidepour, "Quantum computing for stochastic economic dispatch in renewables-rich power systems," *IEEE Transactions on Smart Grid*, 2025.
- [18] C. Gambella and A. Simonetto, "Multiblock admm heuristics for mixed-binary optimization on classical and quantum computers," *IEEE Transactions on Quantum Engineering*, vol. 1, pp. 1–22, 2020.
- [19] J. Stein, J. Jojo, A. Farea, D. Bucher, P. Altmann, M. Çelebi, and C. Linnhoff-Popien, "Combining the qaoa and hhl algorithm to achieve a substantial quantum speedup for the unit commitment problem (2023)," *arXiv preprint arXiv:2305.08482*.
- [20] F. Glover, G. Kochenberger, and Y. Du, "A tutorial on formulating and using QUBO models," *arXiv preprint arXiv:1811.11538*, 2018.
- [21] M. Caleffi, M. Amoretti, D. Ferrari, J. Illiano, A. Manzalini, and A. S. Cacciapuoti, "Distributed quantum computing: a survey," *Computer Networks*, vol. 254, p. 110672, 2024.
- [22] M. Hasanzadeh and A. Kargarian, "Distributed implementation of variational quantum eigensolver to solve QUBO problems," *arXiv preprint arXiv:2508.17471*, 2025.
- [23] Y. Du, Y. Qian, X. Wu, and D. Tao, "A distributed learning scheme for variational quantum algorithms," *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–16, 2022.
- [24] S. DiAdamo, M. Ghibaudi, and J. Cruise, "Distributed quantum computing and network control for accelerated VQE," *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–21, 2021.
- [25] A. Yimsiriwattana and S. J. Lomonaco Jr, "Distributed quantum computing: A distributed shor algorithm," in *Quantum information and computation II*, vol. 5436. SPIE, 2004, pp. 360–372.
- [26] D. P. Kingma, "ADAM: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.