Incentivizing Demand-Side Response Through Discount Scheduling Using Hybrid Quantum Optimization

DAVID BUCHER[1] ![ORCID], **JONAS NÜßLEIN**[2], **COREY O'MEARA**[3] ![ORCID],
IVAN ANGELOV[4] ![ORCID], **BENEDIKT WIMMER**[1] ![ORCID], **KUMAR GHOSH**[3] ![ORCID],
GIORGIO CORTIANA[3], **AND CLAUDIA LINNHOFF-POPIEN**[2]

[1] Aqarios GmbH, 81677 Munich, Germany
[2] Mobile and Distributed Systems Chair, LMU Munich, 80539 Munich, Germany
[3] E.ON Digital Technology GmbH, 30539 Hannover, Germany
[4] Comsysto Reply GmbH, 80992 Munich, Germany

Corresponding author: David Bucher (e-mail: david.bucher@aqarios.com).

ABSTRACT Demand-side response (DSR) is a strategy that enables consumers to actively participate in managing electricity demand. It aims to alleviate strain on the grid during high demand and promote a more balanced and efficient use of (renewable) electricity resources. We implement DSR through discount scheduling, which involves offering discrete price incentives to consumers to adjust their electricity consumption patterns to times when their local energy mix consists of more renewable energy. Since we tailor the discounts to individual customers' consumption, the discount scheduling problem (DSP) becomes a large combinatorial optimization task. Consequently, we adopt a hybrid quantum computing approach, using D-Wave's Leap Hybrid Cloud. We benchmark Leap against Gurobi, a classical mixed-integer optimizer, in terms of solution quality at fixed runtime and fairness in terms of discount allocation. Furthermore, we propose a large-scale decomposition algorithm/heuristic for the DSP, applied with either quantum or classical computers running the subroutines, which significantly reduces the problem size while maintaining solution quality. Using synthetic data generated from real-world data, we observe that the classical decomposition method obtains the best overall solution quality for problem sizes up to 3200 consumers; however, the hybrid quantum approach provides more evenly distributed discounts across consumers.

INDEX TERMS Demand-side response (DSR), problem decomposition, quadratic unconstrained binary optimization (QUBO), quantum annealing (QA), quantum computing (QC), smart grids.

I. INTRODUCTION

The rising demand for energy resources and the growing adoption of renewable electricity sources have prompted a search for innovative solutions to optimize energy consumption in order to reduce grid congestion and carbon emissions. Demand-side response (DSR) [1] has emerged as a promising strategy that focuses on actively managing and adjusting energy consumption patterns in response to grid conditions. Various studies in literature explore DSR, detailing its impact on smart grid technology [2], load scheduling [3], energy economics [4], and optimal control and pricing schemes [5].

Price adjustment serves as a straightforward method to influence consumer behavior. With the emergence of smart devices and the electrification of heating and transportation,

the response to price incentives can be progressively automated. Typically, DSR is achieved by handing out a dynamic price to all customers simultaneously. However, the diverse usage patterns among consumers may favor alternative dynamic pricing policies. Therefore, we aim to find individual price patterns on a per-customer basis to achieve an optimal load shift. We call the distribution of discounts or penalties to specific customers the discount scheduling problem (DSP). The number of customers to be considered in such a problem, i.e., an urban power grid, can become prohibitively large to be solved by classical resources.

In recent years, quantum computing (QC) has garnered significant attention as a potential game changer in various domains, including optimization. Leveraging the principles

of quantum mechanics, quantum optimization algorithms are hypothesized to solve complex optimization problems more efficiently than their classical counterparts. Besides gate-based universal QC, adiabatic quantum computing (AQC) has emerged, which can be shown in general to be equivalent to gate-based approaches [6]. Quantum annealing (QA) [7], [8], a subset of AQC, has been widely adopted for solving optimization problems [9], [10]. As the industry leader in QA hardware, D-Wave's quantum annealer is employed in this work to optimize the DSP. The limited size of current QC hardware forces us to utilize hybrid QC approaches, such as Leap, which is a Cloud service offered by D-Wave and is based on internal problem size reduction [11]. In this work, we additionally develop a customized hybrid approach that performs a problem-specific decomposition.

This article aims to investigate the applicability of QA to DSP optimization. The core contributions of this work are: 1) a QC suitable formulation of the DSP; 2) a problem-specific hybrid decomposition algorithm for the DSP with quadratic unconstrained binary optimization (QUBO) subproblems; and 3) an extensive benchmark of the performance of hybrid quantum–classical routines against purely classical counterparts. The rest of this article is organized as follows. After giving a concise literature review in Section II, we describe the problem formulation and mathematical modeling of the DSP as a quadratic integer programming (QIP) problem in Section III. Since the problem should be solvable for customer counts in realistic scenarios, Section IV motivates and develops the problem-specific decomposition algorithm for problem size reduction. This decomposition routine proves to be very effective, as the benchmarking of classical and quantum-enhanced solvers in Section V shows. Finally, in Section VI, we discuss the overall summary of the work and the implications of applied QC to large-scale optimization problems in industry, targeted to increase renewable energy usage.

II. LITERATURE REVIEW
A. RELATED WORK
Recently, QC applications in the power and energy sector [12], [13], [14], [15] have gained attention for the development of smart grid technology. Several important problems are addressed using QC, for example, power flow calculations [16], [17] or energy grid classification [18]. The traditional planning and scheduling tasks in power systems, such as the minimization of generation cost or the maximization of revenue from electricity generation, are generally formulated as combinatorial optimization problems, which are often NP-hard. Using quantum-inspired optimization algorithms is expected to outperform their classical counterparts [13], [19]. A wide range of optimization problems can be converted into QUBO problems [20], which can be efficiently solved with the quantum approximate optimization algorithm [21] using gate-based universal quantum computers or using D-Wave quantum annealers. In the literature, there exist multiple QC approaches toward unit

commitment [22], [23], [24], [25] and other mixed-integer problems [26], using quantum-inspired alternating direction method of multipliers [27] or Benders decomposition methods [28]. QA approaches are also used for community detection in electrical grids [29], peer-to-peer energy trading [30], or coalition structure optimization [31], [32]. Several research studies benchmark the performance of classical algorithms versus hybrid quantum–classical algorithms, such as Leap on large-scale instances. These include transport robot scheduling [33], job shop scheduling [34], power network partition [35], and SAT problems [36].

As one of this work's main contributions is developing a problem-specific decomposition method to solve large instances of the DSP on currently available hardware, we give a brief overview of combinatorial problem decomposition algorithms in the context of quantum optimization here. Divide-and-conquer approaches have been used for various problem instances, such as the MaxClique problem [37], [38], [39], [40], minimum vertex cover [40], [41], community detection [42], and MaxCut [42], [43]. They all combine the splitting of the problem into subproblems using problem-related methods. In special cases, such as [43], quantum optimization is further utilized in recombining the solution because of the special $\mathbb{Z}_2$ symmetry of MaxCut solutions. Quantum local search (QLS) [44] takes local subproblems of a graph-based problem and iteratively improves a global solution vector. Although applicable to any graph-based problem, QLS has been specifically tested for the maximum independent set problem. The recent emergence of distributed QC has led to the development of decomposition algorithms that still allow for a limited amount of quantum information exchange between the optimization of the subproblems [45], [46], which was successfully demonstrated for the maximum independent set problem. Apart from problem-specific methods, general QUBO decomposition methods have been devised, such as QBSolv [47]. Here, subsets of binary variables of the full QUBO are selected as subproblems, which are solved on a quantum annealer, while, in parallel, a classical heuristic optimizes the original problem. During the process, solutions to the subproblems will incrementally improve the current solution state of the heuristic.

B. INTRODUCTION TO QA
QA is a heuristic for solving combinatorial optimization problems, first proposed in 1998 by Kadowaki and Nishimori [9]. QA utilizes the adiabatic theorem to find the unknown ground state of an Ising Hamiltonian $\mathcal{H}_{\text{Ising}}$, whose minimal energy state corresponds to the solution of a target problem.

With $\mathcal{H}_{\text{Init}}$ being the initial Hamiltonian, the annealing process can be described by the following dynamic Hamiltonian:

$$\mathcal{H}(s) = A(s)\mathcal{H}_{\text{Init}} + B(s)\mathcal{H}_{\text{Ising}} \tag{1}$$

$$\mathcal{H}_{\text{Init}} = -\sum_i \sigma_x^i \tag{2}$$

$$\mathcal{H}_{\text{Ising}} = -\sum_i h_i \sigma_z^i - \sum_{i>j} J_{ij}\sigma_z^i\sigma_z^j \tag{3}$$

where $\sigma_{x,z}^{(i)}$ are Pauli matrices operating on qubit i, and h_i and $J_{i,j}$ are the qubit biases and coupling strengths, which encode the specific problem. $A(s)$ and $B(s)$ are known as the annealing schedule, with $s \in [0, 1]$. At $s = 0$, $A(s) \gg B(s)$, while $A(s) \ll B(s)$ for $s = 1$. As we increase s from 0 to 1, the system undergoes a gradual change from $\mathcal{H}_{\text{Init}}$ to $\mathcal{H}_{\text{Ising}}$. The adiabatic theorem of quantum mechanics states that if that evolution happens slowly enough and the system is initialized in the trivial ground state of $\mathcal{H}_{\text{init}}$, then the state will remain in the ground state of the momentary Hamiltonian [48]. Eventually, at $s = 1$, the state will be in the ground state of $\mathcal{H}_{\text{Ising}}$. Finding the ground state of the Ising model is isomorphic to QUBO [20]; therefore, measuring the final state will reveal the solution to an NP-hard optimization task.

In QA, this transition speed will typically be faster than required for the adiabatic theorem, due to practical considerations. Nevertheless, experimental evidence suggests that, depending on the spin glass model, faster evolution times still output the optimal solution with high probability [49]. Thus, measuring the output repeatedly will eventually find the correct solution.

III. DSP FORMULATION

Given a discrete time horizon of N_T steps t, a set of N_C customers c with projected consumption data $d_{c,t}$, and the local CO_2 grid intensity I_t [g/kWh], the goal of the DSP is to assign each customer individual discrete discounts $z_{c,t} \in Z$, such that the overall CO_2 emissions are minimized, but the overall consumption remains equal. Furthermore, grid constraints must be satisfied at any time step, and the overall consumption deviations of a single customer should be kept to a minimum. The discount categories Z are defined as a symmetric set $\max Z = -\min Z = z_{\max}$, where positive discounts are referred to as penalties, e.g., $Z = \{-0.5, -0.25, 0, 0.25, 0.5\}$.

Reducing CO_2 emissions has the advantage of increasing consumption during periods of abundant local renewable energy production. However, any other linear or quadratic function constructed from the discounts can be used as an objective for the DSP (e.g., minimizing the operational costs based on spot market prices).

A. PRELIMINARY CONSIDERATIONS

Since the distribution system operator (DSO) cannot yet automatically influence the consumption of the customer at a certain time, we have to go the detour over price incentives. We assume that customers are strictly economically motivated, i.e., they alter their consumption based on price. Of course, the convenience of having access to electricity at all times is more important than saving on the cost, such that, in reality, customers cannot vary their consumption arbitrarily at any given time. However, with the emergence of electric vehicles (EVs) with home charging and heat pumps,

automatically varying the load becomes possible. The given discounts then act as a protocol that communicates to a smart home appliance on the customer side when to use electricity and when not, e.g., start or stop charging the EV.

The central assumption of the DSP is that a given discount (or penalty) influences the customers to increase (or decrease) their consumption proportionally. The consumption changes as follows:

$$\tilde{d}_{c,t} = (1 - \chi_c z_{c,t})d_{c,t} \tag{4}$$

when given a discount $z_{c,t}$. The constant χ_c is the (negative) price elasticity of demand of customer c. That is, the higher χ_c is, the more customer c responds to price incentives (lower its demand if price increases and vice versa). In principle, the price elasticity takes positive values below 1, where $\chi_c = 1$ means a full reflection of the price change on the load change. In the literature, different estimations of the electricity demand price elasticity have been made, reaching values between 0.65 and 0.85 in residential U.S. customers [50] and 0.8 to 0.9 in Swiss households [51]. A metastudy [52] on dynamic pricing reveals that the short-term price elasticity has to be estimated lower than the long-term elasticity. Nevertheless, response to dynamic pricing may be increased by automation of the load of smart devices and other enabling technologies [52]. In reality, price elasticity will vary between customers, so we formulate it as a customer-specific value. The DSO can measure the response of individual customers and adjust the elasticities for a more accurate model.

Discrete discounts allow users to change their behavior more distinctly. For instance, providing a small discount to a thousand customers might not necessarily have the intended effect; then, supplying only a few customers with moderate discounts can have. Therefore, restricting to a discrete set of categories Z is sound.

B. MATHEMATICAL FORMULATION

Collecting the considerations, we can finally formulate the DSP optimization problem as QIP for minimizing CO_2 production through load shifting. All terms and constraints will be explained separately in the following sections:

minimize:

$$C(z) = \frac{1}{\mathcal{N}_0}\sum_{c,t} I_t(1 - \chi_c z_{c,t})d_{c,t} \tag{5}$$

$$+ \frac{\lambda_1}{\mathcal{N}_1}\sum_c \left(\frac{1}{D_c}\sum_t d_{c,t}z_{c,t}\right)^2 \tag{6}$$

$$+ \frac{\lambda_2}{\mathcal{N}_2}\sum_{c,t}(z_{c,t} - z_{c,t+1})^2 \tag{7}$$

$$+ \frac{\lambda_3}{\mathcal{N}_3}\sum_{c,t} z_{c,t}^2 \tag{8}$$

such that

TABLE 1 Normalization Constants for the Penalty Terms

Normalization Constant	Expression
$\mathcal{N}_0$	$E(0) - E_{\min}$
$\mathcal{N}_1$	$N_C z_{\max}^2$
$\mathcal{N}_2$	$4N_C(N_T - 1)z_{\max}^2$
$\mathcal{N}_3$	$N_C N_T z_{\max}^2$

$$\sum_{c,t} z_{c,t} d_{c,t} = 0 \tag{9}$$

$$-\Delta p_t \le \sum_c \chi_c z_{c,t} d_{c,t} \le \Delta p_t \quad \forall t \in \{1, \ldots, N_T\}. \tag{10}$$

Here, $C(z)$ is the cost function to be minimized, and D_c is the total power draw of a customer $D_c = \sum_t d_{c,t}$. Furthermore, λ_i refers to penalty factors and $\mathcal{N}_i$ to normalization constants employed to keep the impact of the penalty factors independent of problem size and data. The normalization constants are chosen such that the effect of each penalty term is 1 if all discounts are assigned in the worst possible way (see Table 1).

The formulation of the objective and the purpose of all penalty terms and constraints present in the problem statement (5)–(10) will be explained in the following sections.

1) CO$_2$ EMISSION MINIMIZATION

The combined CO_2 emission is proportional to the changed consumption (4)

$$E(z) = \sum_{c,t} I_t[1 - \chi_c z_{c,t}]d_{c,t} \tag{11}$$

and serves as the main objective of the minimization formulation. The normalization constant $\mathcal{N}_0$ is chosen to map the range of CO_2 emissions between 0 and 1. Therefore, we utilize the trivial origin configuration $P(z = 0)$ as the maximal value and compute a naive lower bound for the CO_2 emissions to set $\mathcal{N}_0 = E(0) - E_{\min}$

$$E_{\min} = \sum_{c,t} \left[1 - \chi_c \, \text{sign}(I_t - \langle I_t \rangle) z_{\max}\right] d_{c,t} \tag{12}$$

which gives all customers the full discount if I_t is smaller than the average and the full penalty if I_t is larger, respectively.

Note that this lower bound solution does not satisfy the constraints of the formulation (9) and (10). Therefore, it is substantially smaller than the actual best solution.

2) CONSUMPTION DEVIATION PENALTY

Customers should not change the total energy they consume over the optimization horizon, i.e.,

$$\sum_t d_{c,t} z_{c,t} \approx 0 \quad \forall c \in \{1, \ldots, N_C\}. \tag{13}$$

A perfect equality can generally not be achieved because of the discrete discounts, except for the trivial case $z_{c,t} = 0$. Therefore, it is represented by the penalty term (6) as a quadratic soft constraint with penalty factor λ_1.

3) DISCOUNT CHANGE PENALTY

As discussed in Section III-A, longer periods with similar discounts exhibit better customer response. We, therefore, employ a penalty function that tries to minimize consecutive discount changes in (7). The corresponding penalty factor λ_2 will be chosen small ($\lambda_2 < \lambda_1$).

4) DISCOUNT REGULARIZATION

We attempt to assign tariff discounts that affect the objective function C by a large enough amount. Suppose that a customer consumes an equal amount at two time steps with $I_{t_1} = I_{t_2}$. Assigning $z_{c,t_1} = -z_{c,t_2} = z_{\max}$ would not change the cost compared to $z_{c,t_1} = z_{c,t_2} = 0$, but can be given anyways. A small $L2$-regularization [see (8)] ensures that discounts are only given if they benefit the overall goal, with $\lambda_3 \le \lambda_2$. $L2$-regularization is chosen over $L1$-regularization since it naturally maps into QUBO.

5) GLOBAL CONSUMPTION DEVIATION CONSTRAINT

Even though the per-customer consumption deviation is soft-constrained (6), the consumption deviation of all customers together can be zero up to numeric precision. Globally, i.e., the combined view of all customers, we do not want any change in overall consumption. Hence, it is a hard constraint [see (9)].

6) POWER RESTRICTION CONSTRAINT

The momentary change in consumption (power restriction constraint) of all customers combined must be bounded due to grid voltage peaks, and therefore, the hard-constrained (10) has been introduced. In addition, for load shifting, we require a time window where consumption can be increased and decreased. The values for Δp_t can be determined using power flow computations and can, in principle, also be asymmetric. Of course, the presented power restriction is a simplification, but it suffices for an initial investigation of the problem.

C. DISCOUNT ENCODING

Discrete discounts $z_{c,t} \in Z$ offer another benefit, which is that they can relatively easily be encoded through binary variables [20]. This makes translating the problem formulation into QUBO easier, which is required for employing quantum optimization techniques.

We will focus on *integer encoding* of the discount set Z. Here, we discretize the range $[-z_{\max}, z_{\max}]$ into N_K linearly spaced categories. Generally, the range can also be asymmetric but is not considered in this work. Therefore, we have $Z = \{-z_{\max} + i\Delta z \mid i = 0, \ldots, N_K - 1\}$, with $\Delta z = \frac{2z_{\max}}{N_K - 1}$. This range can subsequently be expressed using $Q = \lfloor \log_2 N_K + 1 \rfloor$ binary variables $x_{c,t,k}$ for each discount $z_{c,t}$

$$z_{c,t} = \Delta z \sum_{k=0}^{Q-1} w_k x_{c,t,k} - z_{\max} \tag{14}$$

$$\text{with } w_k = \begin{cases} 2^k, & \text{if } k < Q - 1 \\ N_K - 2^{Q-1} + 1, & \text{else.} \end{cases} \quad (15)$$

Every bit combination $x_{c,t,k}$ results in a valid encoding, making an additional penalty term for encoding obsolete [20]. This encoding is very space efficient, allowing an exponential number of categories to be represented with a linearly growing number of qubits.

An alternative method for encoding discounts would be *one-hot encoding*, where N_K bits encode every item of Z by only setting one bit to 1 and the other ones to zero. This is a more general framework that allows any Z (not just linearly spaced) to be encoded through binary variables. However, it requires N_k binary variables per customer and an additional constraint. In the context of QUBO, that constraint has to be enforced as an additional penalty term. Preliminary experiments have shown advantageous results for integer encoding compared to one-hot encoding.

D. ON CUSTOMER SAVINGS

Given that the customers initially receive a flat tariff v_0 [C = /kWh], the discount or penalty ($v_0 \rightarrow (1 + z_{c,t})v_0$) only affects the consumption that deviates from the projected consumption $\tilde{d}_{c,t} - d_{c,t}$. Consequently, the customer pays

$$v_{c,t}\tilde{d}_{c,t} = v_0 d_{c,t} + (1 + z_{c,t})v_0(\tilde{d}_{c,t} - d_{c,t}) \quad (16)$$

for a specific time step.

The customer's cost change over the optimization horizon can be computed via the sum of momentary price differences through (16)

$$\Delta v_c = \sum_t (v_0 - v_{c,t})\tilde{d}_{c,t} = v_0 \sum_t z_{c,t}(\tilde{d}_{c,t} - d_{c,t}). \quad (17)$$

Note that we have used the sum over the changed consumption as the baseline for our comparison, since in any case $\sum_t d_{c,t} \approx \sum_t \tilde{d}_{c,t}$, and we only want to compare the cost for the same amount of purchased electricity.

Substituting in the altered consumption from (4), we obtain a change in cost given by

$$\Delta v_c = -v_0 \chi_c \sum_t z_{c,t}^2 d_{c,t}. \quad (18)$$

The absolute price change is dependent on the flat tariff and the total consumption of the customer. We will, therefore, look at the relative savings $s_c = -\Delta v_c / \sum_t v_0 \tilde{d}_{c,t} \geq 0$ in Section V.

As $z_{c,t}^2 \geq 0$ and $\chi_c \geq 0$, the customer's price change is guaranteed to be $\Delta v_c \leq 0$ negative, so a customer will always save money by complying with the incentives. The savings are exactly zero if the customer does not respond to incentives at all, i.e., $\chi_c = 0$.

E. GRID DATA

For the DSP, we require forecasted consumption data $d_{c,t} \geq 0$ [kWh] for each customer and predicted grid CO_2 intensity I_t of the power generation in the considered region. We use
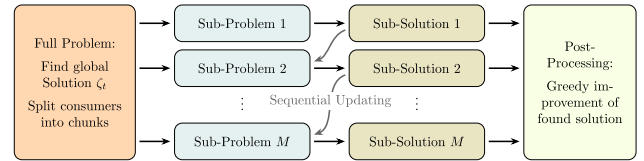

FIGURE 1. Overview of the decomposition routine. The problem is split into subproblems. Solutions can influence the following subproblems via sequential updating. Finally, subsolutions are gathered to a full solution and a postprocessing step is employed that improves the solution quality greedily while also making the power restriction constraint satisfied.

standard load profiles of residential and industrial customers, which we modify by adding noise and shifting in time. In addition, the load profiles get scaled to resemble various numbers of residents. Moreover, we include photovoltaic (PV) electricity generation by estimating the potential based on roof data of Munich residential areas and simulating the production from historical solar irradiance data. PV production reduces the customers' consumption. Grid infeed, i.e., if more PV is generated than consumed, is not especially considered. The grid CO_2 intensity is taken from the real-world data in Munich.[1] The data used throughout this text consists of roughly 16 000 customers and 76 time steps, spanning a 19-h period with 15-min intervals. The CO_2 and solar data are from January 13, 2023.

IV. PROBLEM DECOMPOSITION

The number of integer variables needed to construct the discount matrix is $N_C \times N_T$. Given a one-day optimization horizon with 15-min time steps, each customer requires 96 integer decision variables in the problem. However, as the number of customers will grow quite large,[2] the number of integers grows akin. Even worse, the number of qubits in the quantum formulation is scarce, and every integer must be encoded with Q qubits. Thus, the move to a hybrid quantum–classical optimization scheme seems inevitable.

In this section, we propose a hybrid approach that is based on problem decomposition. Despite the drawback that decomposition increases solution bias, we find that we can manage the hard constraints of the DSP classically in a preprocessing step. This eliminates the need for a costly reformulation of inequality constraints with slack variables. Fig. 1 shows an overview of the steps taken for the decomposition.

A. MOTIVATION

1) GLOBAL SOLUTION

Shifting the perspective from the individual customer level to a global scope, where all customers are regarded as a unified entity, we consider the overall consumption $D_t = \sum_c d_{c,t}$ and the mutable consumption, i.e., the consumption weighted by the individual customer susceptibilities $\tilde{D}_t = \sum_c \chi_c d_{c,t}$.

---

[1]The data are provided by E.ON's App for monitoring local CO_2 intensities. [Online]. Available: https://www.bayernwerk.de/de/fuer-zuhause/oekoheld.html

[2]Typically, we want to consider more than 1000 customers.

Furthermore, we can express the weighted average of all discounts given per customer—from now on called *effective discount*—as follows:

$$\zeta_t = \langle z_{c,t} \rangle_c = \frac{1}{\widetilde{D}_t} \sum_c \chi_c d_{c,t} z_{c,t} \in [-z_{\max}, z_{\max}]. \quad (19)$$

Utilizing the formulation of the effective discount, we can transform the CO_2 production from (11) into

$$E(\zeta) = \sum_t I_t \left(D_t - \widetilde{D}_t \zeta_t \right). \quad (20)$$

The global consumption deviation constraint (9) and the power restriction constraint (10) can be expressed solely in terms of the effective discount. Therefore, we represent the global version of the DSP as a linear program

$$\text{minimize:} \quad E(\zeta)$$

$$\text{such that:} \quad -\Delta p_t \leq \widetilde{D}_t \zeta_t \leq \Delta p_t \quad \forall t \in 1, \ldots, N_T$$

$$\sum_t \widetilde{D}_t \zeta_t = 0. \quad (21)$$

This formulation disregards any per-customer constraints that are still part of the DSP. Nevertheless, it is a useful tool to estimate how much CO_2 reduction is maximally possible with all the hard constraints (9) and (10) in place. In fact, the solution ζ_t^* is guaranteed to give an optimal lower bound $E(\zeta^*)$

$$E(\zeta^*) \leq E(z) \quad \forall z \in \mathcal{Z} \quad (22)$$

where $\mathcal{Z} = \{z \in Z^{N_C \times N_K} \text{ s.t. } (9), (10) \text{ hold}\}$ is the set of feasible discount matrix configurations. The global DSP consists of only N_T continuous variables. Thus, it can be quickly and efficiently solved using standard procedures such as the Simplex method [53].

Given an optimal effective discount, ζ_t^*, we can utilize (19) to optimize the integers $z_{c,t}$ for the individual customers per time step, i.e., $\min_z[\zeta_t(z) - \zeta_t^*]^2$. In addition, we can include the penalty terms from the DSP (6)–(8) in the subsequent optimization. However, doing so would yield an optimization problem the same size as the original problem.

Nonetheless, the following section reveals that we can achieve a satisfactory approximation of a continuous effective discount by considering only a limited number of customers. As a result, we can divide the customers into smaller groups or chunks and optimize each chunk separately.

2) REPRESENTATIONAL POWER

In this section, we motivate that (19) can be fulfilled for any arbitrary ζ_t with sufficient accuracy given a small constant number of customers. We will focus on a discount range $\zeta_t \in [-1/2, 1/2]$ and five discrete discounts $z_c \in \{-1/2, -1/4, 0, 1/4, 1/2\}$. From the generated consumption data (see Section V), we take a random set of customers
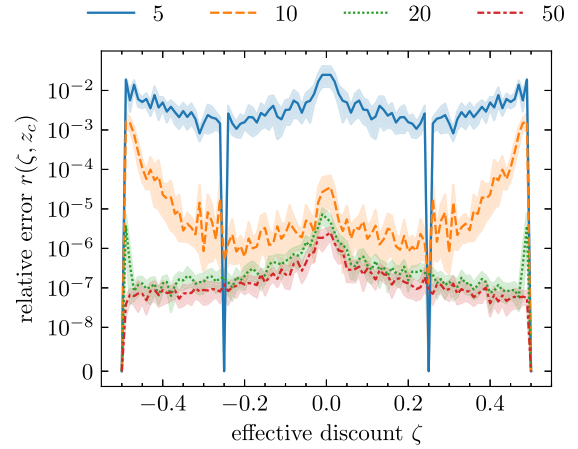

FIGURE 2. Relative approximation error for different values of ζ averaged over multiple time steps. The different colors show the approximation error for an increasing number of customers averaged over 20 time steps. The central peaks are due to absolute errors getting amplified at small effective discounts. Because the effective discounts $\zeta = \pm 0.25, \pm 0.5$ are exactly realizable by giving all customers $\pm 25\%, \pm 50\%$ discounts, we can observe spikes there. Even though zero error can be achieved in the $\zeta = \pm 0.25$ case, Gurobi finds different configurations with good enough error first (for larger than five customer chunks). Effective discount $\zeta = 0$ is not shown since the relative error is not defined.

and compute

$$\min_{z_c} r(\zeta, z_c) = \min_{z_c} \frac{1}{\zeta} \left| \frac{1}{D} \sum_c d_{c,t} z_c - \zeta \right| \quad (23)$$

for all available time steps. Fig. 2 shows the result with different numbers of customers. The average over all time steps is plotted, and the error bands indicate a 95% confidence interval. It is evident that even with only ten customers, the relative error remains consistently below 1%. As more customers are added, the error decreases significantly, reaching a negligible level. Therefore, we contend that by maintaining a small constant number of customers within a chunk (e.g., 20–50 customers), it is possible to obtain a reliable approximation of an effective discount while still considering the per-customer soft constraints of the DSP.

B. FULL DECOMPOSITION ROUTINE

We now assemble the pieces into a full hybrid routine for decomposition, as seen in Fig. 1. The process begins with solving the global DSP (21), followed by dividing customers into chunks. We sort the customers by total consumption and split them into M-sized groups, s.t. the largest customers are arranged in the first chunk, etc. We argue that it is better to have customers with comparable consumption in one chunk because they can counteract each other better than, e.g., one industrial customer and 20 single households. For each chunk, we can define subproblems in which special effective discounts per chunk are introduced in Section IV-B1. These subproblems are of QUBO form and aim to assign discounts to customers such that the overall effect matches

an effective discount while making sure that each customer does not deviate from its total consumption by much.

Since we can solve the subproblems sequentially, we can enhance the results by incorporating the errors from prior optimizations into the subsequent subproblems in Section IV-B2. Eventually, all the chunks are collected, and a final postprocessing step shown in Section IV-B3 is applied to ensure that no constraints are violated.

1) CHUNK PROBLEMS

The customers are partitioned into $M = N_c/m$ mutually exclusive chunks C_j, s.t. $\bigcup_j C_j = \{1, \ldots, N_C\}$ and $C_i \cap C_j = \emptyset \, \forall i \neq j$. Note that we require and expect the chunk size to be chosen, s.t. $N_C \mod m = 0$.

Most likely the consumption deviation per chunk

$$\sum_{c \in C_j} \sum_t \chi_c d_{c,t} \zeta_t^* \neq 0 \quad \forall j \tag{24}$$

is not zero, which, by default, introduces a bias in the consumption deviation soft constraint (6). Thus, the first goal is to define chunk effective discounts ξ_t^j with the following properties:

$$\sum_t \widetilde{D}_t^j \xi_t^j = 0 \quad \forall j \tag{25}$$

$$\frac{1}{\widetilde{D}_t} \sum_{j=1}^M \widetilde{D}_t^j \xi_t^j = \zeta_t^* \quad \forall t \tag{26}$$

where we define an alterable consumption for one chunk $\widetilde{D}_t^j = \sum_{c \in C_j} \chi_c d_{c,t}$, similar to the definition of the total mutable consumption.

We define the chunk-effective discount as follows:

$$\xi_t^j = \zeta_t^* - \frac{\alpha_t}{\widetilde{D}_t^j} \sum_{t'} \widetilde{D}_{t'}^j \zeta_{t'} \tag{27}$$

where α_t are arbitrarily chosen constants, s.t. $\sum_t \alpha_t = 1$. Conditions (25) and (26) are satisfied with this definition. The values α_t are chosen constant $\alpha_t = 1/N_T$, but we have to make sure that $\xi_t^j \in [-z_{\max}, z_{\max}] \, \forall t, j$. If this is not possible for one time step t, α_t has to be reduced, while the remaining αs have to be increased.

The optimization objective is to approximate the following equality with the chunk effective discount:

$$\widetilde{D}_t^j \xi_t^j = \sum_{c \in C_j} d_{c,t} \chi_c z_{c,t} \quad \forall t \in \{1, \ldots, N_T\}. \tag{28}$$

The objective can be reformulated as a least squares error problem to find an optimal $z_{c,t}^*$

$$\arg \min_{z_{c,t}} \frac{1}{N_T z_{\max}^2} \sum_t \left(\xi_t - \frac{1}{\widetilde{D}_t^j} \sum_{c \in C_j} d_{c,t} \chi_c z_{c,t} \right)^2 \tag{29}$$

and is directly in QUBO form after the binary representation of the discounts has been substituted into the formulation.

The previously discussed penalty terms and regularizations—consumption deviation (6), discount change penalty (7), and discount regularization (8)—can be carried over to this optimization problem.

2) SEQUENTIAL UPDATING

When the subproblems are solved in sequence, the error between the true achieved effective discount and the demanded one can be carried over into the next optimization to be corrected. For optimizing ξ_t^j, the procedure can be adapted as

$$\xi_t^j \leftarrow \xi_t^j + \frac{1}{\widetilde{D}_t^j} \sum_{i=1}^{j-1} \left(\widetilde{D}_t^i \xi_t^i - \sum_{c \in C_i} z_{c,t}^* d_{c,t} \right). \tag{30}$$

Doing so will significantly improve the overall accuracy of the method. Of course, one has to ensure that the altered ξs do not exceed the bounds $[-z_{\max}, z_{\max}]$.

3) POSTPROCESSING

Finally, we describe a postprocessing scheme that refines the result and ensures that the power restriction constraint (10) is held. Algorithm 1 describes the greedy improvement of the solution.

Conceptually, it is quite simple: for each time step, we extract those customers whose discounts can be increased or decreased while also improving the consumption deviation penalty (6). Then, we try all combinations between one increase and one decrease and investigate how the effective discount behaves. If ζ_t^* is negative, we want the real effective discount to be as close as possible but at least larger than ζ_t^*. If it is positive, the other way around. Doing so always satisfies constraint (10). We find the combination that matches the requirements the best and update the respective discounts if it achieves an improvement. Otherwise, the time step is skipped.

Since all the possible combinations of up and down moves have to be considered, the complexity of the algorithm scales at worst with $\mathcal{O}(N_T N_C^2/4)$. Nevertheless, limiting the possible moves to at most r provides sufficient accuracy, empirically. This then reduces the complexity to $\mathcal{O}(N_T N_C + N_T r^2)$.

V. EXPERIMENTS AND RESULTS

A. EXPERIMENTAL SETUP

To benchmark the performance of solving the DSP, we consider out-of-the-box solvers and our developed decomposition method and evaluate the results using a set of metrics that best represent the different goals described in the DSP formulation.

Algorithm 1: Postprocessing Algorithm.

Data: $d_{c,t}, r \in \mathbb{N}$
\# r is a parameter that dials the accuracy/runtime

Result: $z_{c,t}$

$\Delta z \leftarrow 2 z_{\max}/(N_K - 1);$ \# Discount step

$\Delta_{c,t} \leftarrow \chi_c d_{c,t} \Delta z;$ \# Possible deviations

$\delta_c \leftarrow \sum_t \chi_c d_{c,t} z_{c,t};$ \# Consumption deviation

for $t \in \{1, \dots, N_T\}$ **do**

    $p \leftarrow \sum \chi_c d_{c,t} z_{c,t};$ \# Power deviation

    $\varepsilon \leftarrow \zeta_t^* \tilde{D}_t - p;$ \# Error from demanded

    \# Increasable customers

    $C^{\uparrow} \leftarrow \{c = 1, \dots, N_C \mid z_{c,t} < z_{\max}, \delta_c < -\Delta_{c,t}/2\};$

    \# Decreasable customers

    $C^{\downarrow} \leftarrow \{c = 1, \dots, N_C \mid z_{c,t} > -z_{\max}, \delta_c > \Delta_{c,t}/2\};$

    $C^{\uparrow} \leftarrow \text{limit}(C^{\uparrow}, r);$ \# reduce size $|C^{\uparrow}| = r$

    $C^{\downarrow} \leftarrow \text{limit}(C^{\downarrow}, r);$

    \# Compute combinations of increasing and decreasing two customer discounts

    $X_{c,c'} \leftarrow \text{sign}(\zeta_t^*)(\varepsilon - [\Delta_{c,t} - \Delta_{c',t}]);$

    \# Find positive (feasible) ones

    $C_2 \leftarrow \{(c^{\uparrow}, c^{\downarrow}) \in C^{\uparrow} \times C^{\downarrow} \mid X_{c^{\uparrow},c^{\downarrow}} > 0\};$

    \# Get the best move

    $c^{\uparrow}, c^{\downarrow} \leftarrow \arg\min_{c^{\uparrow},c^{\downarrow} \in C_2} X_{c^{\uparrow},c^{\downarrow}};$

    if $X_{c^{\uparrow},c^{\downarrow}} > \text{sign}(\zeta_t^*)\varepsilon$ **then**

        | **continue**;   \# No improvement

    end

    \# Update solution and consumption deviation

    $z_{c^{\uparrow},t} \leftarrow z_{c^{\uparrow},t} + \Delta z;$

    $z_{c^{\downarrow},t} \leftarrow z_{c^{\downarrow},t} - \Delta z;$

    $\delta_{c^{\uparrow}} \leftarrow \delta_{c^{\uparrow}} + \Delta_{c^{\uparrow},t};$

    $\delta_{c^{\downarrow}} \leftarrow \delta_{c^{\downarrow}} - \Delta_{c^{\downarrow},t};$

end

TABLE 2 Overview of the Investigated Solvers

Name	Description
Gurobi	Classical Solver
Leap	Quantum classical hybrid solver
Decomp-Gurobi	Decomposition with Gurobi as subsolver
Decomp-Leap	Decomposition with Leap as subsolver
Decomp-QPU	Decomposition with D-Waves Advantage 4.1 QA processor as subsolver
Decomp-SA	Decomposition with simulated annealing as subsolver

1) INVESTIGATED SOLVERS

An overview of the considered solvers and settings can be found in Table 2. As a state-of-the-art purely classical baseline, we use Gurobi[3] [54]. This is compared to D-Wave's LeapHybridCQM solver [11] (called just Leap in the following), which is a quantum–classical hybrid algorithm that uses classical algorithms to optimize the problem while using quantum computers to solve suitable subtasks. This has the benefit of solving larger problems than possible directly on current quantum hardware while also supporting more sophisticated optimization models that include hard constraints. Like our decomposition routine, Leap partitions the

[3]All experiments with Gurobi were conducted on an M1 MacBook Pro (2020) with Gurobi Version 9.0

problem into subproblems via a proprietary algorithm. However, it follows a general ansatz compared to our problem-specific one. Leap is accessed through D-Wave's Cloud service. Both Leap and Gurobi solve the optimization problem presented in (5)–(10). These two out-of-the-box solvers are compared against our own problem-specific decomposition routine introduced in Section IV, subsequently called Decomp-Gurobi, Decomp-Leap, or Decomp-QPU, depending on the method considered for solving the chunk problems (29). QPU refers to direct access to the D-Wave's QA processor Advantage 4.1 [11]. Whenever a decomposition solver is followed by an integer, it refers to the chunk size m. The postprocessing algorithm is used with a cutoff value $r = 500$.

In preliminary experiments, we additionally investigated D-Wave's QBsolv hybrid decomposition algorithm [47], but the performance was not comparable to the approaches presented here. Furthermore, we have noticed that the solution quality did not depend on the subsolver chosen (e.g., D-Wave's QPU or a simulated annealing (SA) heuristic), indicating that the classical Tabu Search [55] is solely responsible for the optimization work done.

The hybrid Leap solver only has the time limit as a control parameter exposed to the user. The time limit is bound from below by the minimum runtime that is heuristically calculated from the input problem (i.e., from the number of variables and couplings involved). Because it only depends on the problem structure and not the solution quality found, the minimum time limit cannot be used as the scaling metric. Furthermore, Leap exploits the full time limit setting and does not abort when satisfactory energy has been reached. Thus, a time-to-solution metric is not achievable within a single run and, therefore, also not considered in our benchmarking.

To alleviate the issue and ensure a fair comparison, we give each solver a heuristically increasing time limit of $0.1\,\text{s} \times N_C$. We observed that Leap tends to overrun the set timeout, which is the reason why we first run Leap with the linear growing timeout and then run the remaining solvers with the timeout matching Leaps runtime. Since the decomposition solver consists of multiple subsolver calls, we set the timeout for each subsolver as the whole timeout divided by the number of chunks, i.e., a timeout of $0.1\,\text{s} \times m$.

2) METRICS

Because we consider an optimization task with multiple goals involved, it is not sufficient to consider only the objective value of our model as a performance metric. Instead, we simultaneously investigate multiple metrics.

1) *Cost:* The cost, or objective, of the optimization problem (5)–(8) is the main metric for comparing solver performance. To ensure an easier comparison between problem instances, we investigate the relative cost error with respect to the global solution from (22), defined

as follows:

$$\frac{|C(z) - C(\zeta^*)|}{|C(\zeta^*)|} \qquad (31)$$

with $C(\zeta^*) = E(\zeta^*)/\mathcal{N}_0$. This is a guaranteed lower bound to the cost since all penalty terms are bounded from below by zero.

2) *CO_2 reduction:* The CO_2 reduction is the central term in the DSP objective. Hence, it is also valuable to inspect it separately. We, therefore, compute the relative CO_2 reduction error through (22)

$$\frac{E(z) - E(\zeta^*)}{E(0) - E(\zeta^*)}. \qquad (32)$$

$E(0)$ is the CO_2 emission prior to discount scheduling.

3) *Consumption deviation standard deviation:* We expect the consumption deviations for each customer to be centered around zero since the problem is constrained to have a zero total consumption deviation. We, therefore, measure the standard deviation of the customer discount deviations as follows:

$$\sqrt{\frac{1}{N_C} \sum_c \left(\frac{1}{D_c} \sum_t d_{c,t} z_{c,t} \right)^2}. \qquad (33)$$

4) *Average discount changes:* Since we strive to reduce the changes between two discount categories as much as possible, we measure the average discount changes

$$\frac{1}{N_C(N_T - 1)} \sum_c \sum_{t=1}^{N_T - 1} (1 - \delta_{z_{c,t}, z_{c,t+1}}) \qquad (34)$$

where δ refers to the Kronecker delta.

5) *Average relative cost savings:* Not a quantity that is optimized for in the objective, but interesting for the DSO, is the relative cost savings per customer, as defined in Section III-D. To obtain a single indicator of the performance, we evaluate the mean $\langle s_c \rangle_c$ of the relative savings.

3) PARAMETERS

Solving the DSP for a given dataset, consisting of the consumption of N_C customers at N_T time steps, requires fixing a set of open variables and parameters. In a real-world scenario, the customer price elasticity on demand χ_c could be measured from the individual customer's behavior. However, as it only acts as a proportionality constant, we set $\chi_c = 1$ for this investigation. Next, we use five discount categories, with a 50% discount maximally. That, in turn, refers to the following valid discounts: $z_{c,t} \in \{-50\%, -25\%, 0\%, 25\%, 50\%\}$. As a consequence, a discount of, e.g., 50% would result in an increase in the customer's consumption by 50%.

The power deviation bounds Δp_t are set to a constant 10% of the average total consumption ($0.1 \times \langle D_t \rangle_t$). For the purpose of this novel problem formulation and benchmarking regarding scalability and solution quality, this is a pragmatic

TABLE 3 Parameter Setting for the Investigated Problems

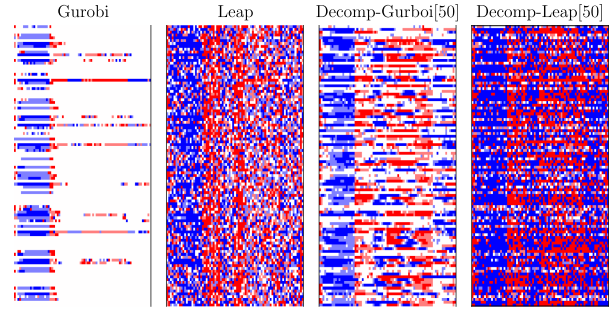| Parameter	Value	Description
χ_c	1	Customer price elasticity
z_{max}	50%	Maximum discount
N_K	5	Number of discounts
Δp_t	$0.1 \times \langle D_t \rangle_t$	Power restriction
λ_1	0.1	Consumption deviation penalty
λ_2	10^{-5}	Discount regularization
λ_3	10^{-4}	Discount change penalty

FIGURE 3. Discount matrices $z_{c,t}$ found by the investigated solvers for $N_C = 100$. Blue indicates a discount, and red corresponds to a penalty. White means no discount given at all. Despite their effects on the overall consumption (see Fig. 4) being the same, the discount matrices differ a lot from each other. It is apparent that `Gurobi` hands out the discounts more greedily than `Leap`, indicating a bigger impact of the regularization.

approach to approximation. In practice, however, those values may be derived from real-world grid constraints that can be inferred through power flow calculations.

Finally, the remaining penalty parameters are fixed by analyzing a small-scale example with `Gurobi` and dialing in the strengths of the penalties, such that they have a reasonable effect for the `Gurobi` result. It is important to note that a comprehensive examination of the solver's response to parameter settings is beyond the scope of the current investigation.

An overview of all parameter settings is given in Table 3.

B. EXAMPLE WITH 100 CUSTOMERS

We first examine the optimization result of the different solvers in detail for a 100-customer example qualitatively before focusing on the previously discussed metrics. For that, we analyze the solutions of four solvers, `Gurobi`, `Leap`, and two $m = 50$ decomposition methods with the same solvers as the subroutine. The results for the discount matrices $z_{c,t}$ can be seen in Fig. 3, while their overall effect on the consumption is displayed in Fig. 4. Visually, the individual discount matrices exhibit distinct patterns (cf. `Gurobi` and `Leap`), but the effective result stays comparable regarding the CO_2 reduction. The optimal CO_2 reduction is 12.45 kg, `Leap` differs by 0.3%, `Gurobi` by 0.6%, `Decomp-Leap` by 1.4%, and `Decomp-Gurobi` by only 0.005%.

Apart from the global optimization metrics, we are also interested in how the optimization performs per customer. In Fig. 5, one can see how the relative consumption changes are
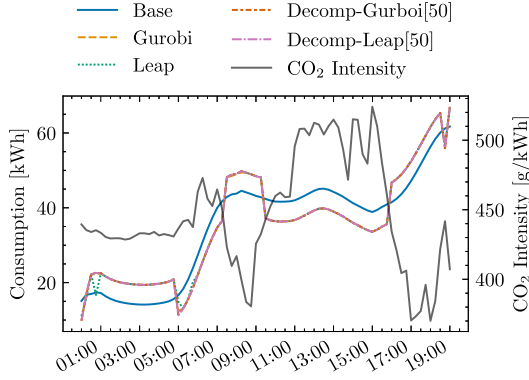
FIGURE 4. Effect of the DSP solution for problem size $N_C = 100$. The plot shows the aggregated consumption with and without ($z = 0$) discounts in place, as well as the grid CO_2 intensity. Visually, the solutions of all solvers produce a similar effective consumption change, as already predicted in Section IV. As expected, times with high CO_2 emissions produce an effective decrease in consumption and vice versa.
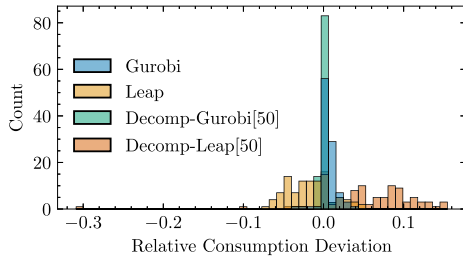

FIGURE 5. Histogram of the relative consumption deviation. One can see that both `Gurobi` solvers have relatively little spread and are well centered around zero. The `Leap` solvers, on the other hand, possess a large spread and are additionally shifted away from zero. The shift away from zero reduces in larger problem instances.
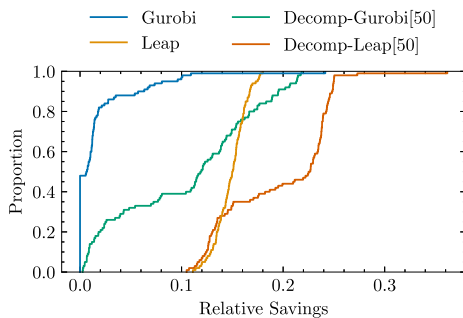
FIGURE 6. Cumulative distribution plot of the relative savings of the customers. The two chunks can be well distinguished in the `Decomp` solvers. `Gurobi` only distributes savings to relatively few customers. On the other hand, `Leap` distributes similar discounts to all customers.

distributed. Furthermore, Fig. 6 visualizes the distribution of savings to the customers.

Finally, it remains important to note that the results for the `Leap` solvers vary throughout multiple runs. Here, only a single run has been picked, which is characteristic of the
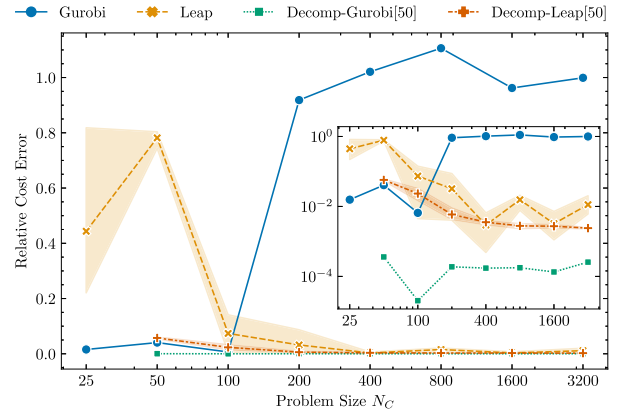
FIGURE 7. Relative cost error for different solvers with respect to problem size N_C. Cost is the optimization objective known from (5)–(8). The relative value is taken with respect to the bound known from $C(\zeta^*)$. The inset shows the relative cost error with logarithmic scaling. The error bands indicate the maximum and minimum of the three runs.

behavior of these solvers. Furthermore, no investigation toward direct QPU access has been made since the space requirements for a single customer are already 76 integer variables, i.e., 228 binary variables. The problem after gathering multiple customers in a chunk is, hence, not embeddable in the QPU since we are facing quite dense connectivity in the QUBO. For a reduced problem size, we perform investigations in Section V-F.

C. SCALING ANALYSIS

To test the performance of different solvers, we created test instances using generated data with N_C ranging from 25 to 3200 customers and considering the full 76 time steps. Our problem instances, therefore, consist of 1900 to 243 200 integer variables. To account for the stochasticity of the results from the quantum solvers, we run the quantum solvers three times.

The results in terms of the objective function depending on problem size are visualized in Fig. 7. It is evident that a crossover in performance between `Gurobi` and `Leap` happens between 100 and 200 customers. After that size, `Gurobi` is not able to finish the root relaxation within the given time bounds and falls back to a heuristic solution, which has inferior performance. Although not a directly fair comparison since `Gurobi` runs on a local machine while the `Leap` hybrid solver is run on a proprietary D-Wave cloud architecture, we argue that the pattern generalizes, i.e., the inflection point where `Gurobi` does not reach satisfactory results anymore shifts to larger instances but eventually happens. In the regime $N_C < 200$, Gurobi's MIP Gap roughly coincides with the relative cost error since the lower bound of Gurobi is almost equal to $C(\zeta^*)$. For low problem size, the `Leap` solver demonstrates high relative cost error; however, this error decreases rapidly up to low hundreds of consumers. Nonetheless, the decomposition routines outperform
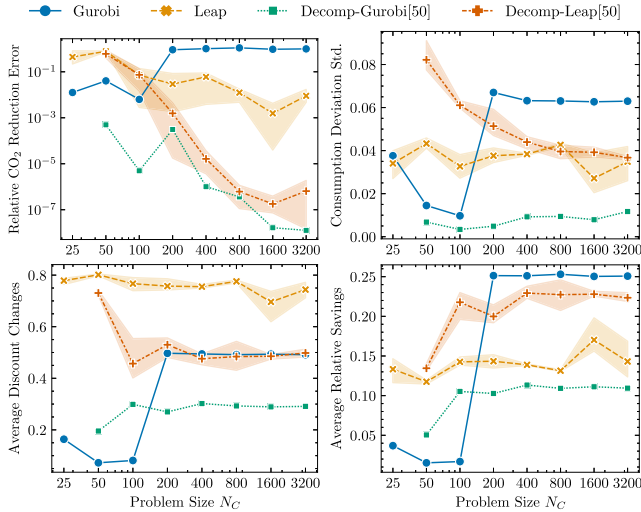
FIGURE 8. Auxiliary metrics for different problem sizes and different solvers. The plots show the CO_2 reduction error and the per-customer metrics: The standard deviation of all customer consumption deviations (6) and the average discount changes (7), which we want both to be small. The bottom right pane displays the average relative savings $\langle s_c \rangle$. The error bands indicate the maximum and minimum of the three runs.

the general-purpose solvers, especially the purely classical `Decomp-Gurobi` approach.

Fig. 8 shows the relative CO_2 reduction error and three per-customer metrics. The relative CO_2 reduction error shows a similar pattern as the inset in Fig. 7, which is due to the emission reduction being the main part of the optimization objective. It is apparent that the decomposition routines in the higher problem instances produce results with almost perfect CO_2 reduction (less than 10^{-5} error), which can be explained by the fact that they have access to the best emission reduction bound. As a consequence, the per-customer penalties [see (6)–(8)] are responsible for the cost error visible in Fig. 7.

Investigating the per-customer constraints, we notice that the `Gurobi`-based solvers outperform the quantum-enhanced routines ($N_C < 200$). This is likely due to `Gurobi` being better at handling smaller changes in the optimization objective. However, it is also important to note that, as apparent from the discount matrices in Fig. 3, `Gurobi` gives many customers not even a single discount. Hence, they do not receive any discount changes or consumption deviations, which reduces the average measure.

Examining the heuristic solutions of `Gurobi`, when solving its root relaxation aborts ($N_C \geq 200$), reveals that the discount matrix is almost completely filled with extremal values $z_{c,t} = \pm z_{\max}$. The constraints are satisfied, but the discounts are randomly distributed, which allows for computing the per-customer metrics analytically, supposing $z_{c,t} = \pm z_{\max}$ with equal probability. The consumption deviation metric from (33) simplifies to $z_{\max}\sqrt{\langle D_c^{-2} \sum_t d_{c,t}^2 \rangle_c}$, which is, e.g., 0.064 in the $N_C = 400$ case. The average discount change metric reduces to the probability of observing one discount
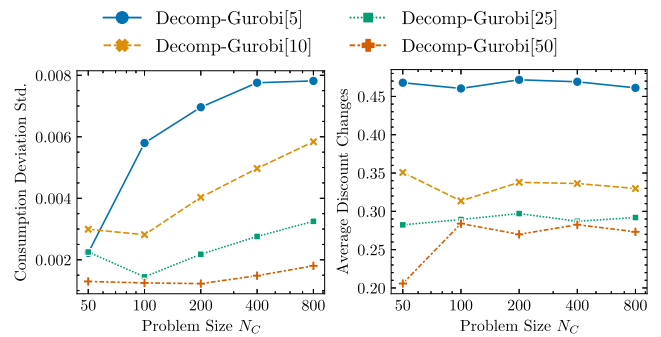
change (0.5). Finally, $s_c = z_{\max}^2 = 1/4$, since the customer savings are dependent on a weighted average over $z_{c,t}^2 = z_{\max}^2 = \text{const.}$ [see (18)].

To conclude this analysis, we remark that `Gurobi` struggles at large problem sizes since its root relaxation cannot be solved within the given time constraints, which indicates a potential advantage of the quantum-enhanced solver here. Yet, the domain-specific decomposition routine provides even better results, especially in conjunction with the classical solver. We argue that since the decomposition-based solvers work so well, the space of good solutions is large, which makes this problem a fitting choice for heuristic-based solvers more than mathematical solvers such as `Gurobi`.

D. CHUNK SIZE EFFECT

After we observed that the decomposition solver provides satisfying results both with `Gurobi` and `Leap` employed as subsolver, we are interested in what impact the chunk size has on the result. For that, we only inspect `Decomp-Gurobi` with different chunk sizes $m = 5, 10, 25, 50$ and focus on a reduced problem size frame up until $N_C = 800$. We have seen that the problem complexity does not grow linearly with the problem size. Thus, we give a more generous timeout of $0.5\,\text{s} \times m$ in this investigation in order to isolate the effects of the decomposition routine from the solver performance.[4] The global effect, i.e., how much CO_2 is reduced, does not differ between the chunk sizes (below 1% error). The constant sequential updating of the objective also helps a lot with finding the best CO_2 reduction, even with five customer chunks. Fig. 9 shows the consumption deviation and discount changes, where a clear tendency that larger chunks result in lower per-customer metrics can be observed, i.e., less consumption deviation per customer and fewer overall discount changes.

FIGURE 9. Per-customer metrics evaluated with different chunk sizes in the decomposition. As expected, the metrics improve (get smaller) as the chunk sizes get larger since more flexibility remains in the chunk.

---

[4]Preliminary experiments have shown a performance increase for the larger chunk sizes when increasing the timeout. This increase came to a slowdown at around $0.5\,\text{s} \times m$.
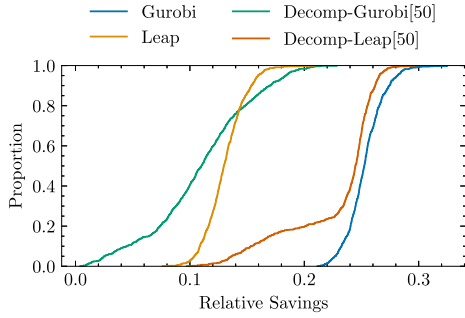
FIGURE 10. Cumulative distribution plot of the relative savings of the customers at $N_C = 800$. As discussed earlier, `Gurobi`'s root relaxation does not finish anymore, which causes savings of around 25%. `Leap` produces fair discounts, similar to Fig. 6. The other two solvers produce more complex unfair savings distributions.
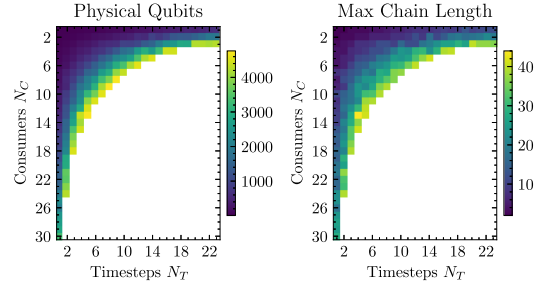

FIGURE 11. Embeddable subproblem size for the D-Wave Advantage 4.1 QPU. The left-hand matrix shows how many physical qubits are needed when a subproblem with N_C customers and N_T time steps are embedded. A white field indicates that no embedding has been found. The right-hand plot shows the maximal chain length for the found embedding, i.e., how many qubits are maximally connected to form one logical qubit. All embeddings were found using D-Wave's `MinorMiner` package.

E. FAIRNESS ANALYSIS

The goal of this section is to investigate how the solvers strategically distribute the discounts to the target customers. This is done by investigating how the relative savings s_c are distributed between individual customers. Figs. 6 and 10 show two cumulative distribution plots of the results from 100 and 800 customer problem sizes. The more vertical (zero slope) a given cumulative line is, the fairer the discounts are distributed among the consumers, thereby implying a better social welfare measure for the energy consumers. Except for `Gurobi`, the qualitative patterns of the solvers are similar. `Leap` produces a fair savings distribution, which means that all customers experience the same savings (10–15%).

In Fig. 6, the splitting in half of the decomposition can be observed quite remarkably. The resolution of the 16 individual chunks in Fig. 10 is no longer possible. However, a kink in `Decomp-Leap` can be observed, which means that about 70% of the customers save a similar and relatively large amount (22–25%), while fewer savings are distributed to a smaller group (10–22%). `Decomp-Gurboi` reveals a straight but shallow curve, which means that customers will receive savings between 0% and 20% almost equally likely.

F. DIRECT QPU ACCESS WITH DECOMPOSITION

A QA processor, such as D-Waves Advantage 4.1, suffers from limited connectivity between the physical qubits. However, for our QUBO subproblems (29), we can analytically compute the number of couplings for a single qubit as follows:

$$Q(N_T - 1) + Q(m - 1) + Q - 1 \tag{35}$$

where Q is the number of qubits required to encode the discount, N_T is the number of time steps, and m is the number of customers per chunk. This term is derived by inspecting the terms in the QUBO formula and observing that we either have couplings within all customers of a chunk at a single time step or couplings within all time steps of a single customer. For the first case, one qubit is connected to all Q qubits of the other $m - 1$ customers in the chunk and to $Q - 1$

qubits of the same customer. The same is for the second case, but the $Q - 1$ connections within the time step have already been covered in the first case.

The derived quantity grows with the problem size, but the coupling per qubit of the D-Waves Pegasus graph is a constant 15 [11]. Thus, physical qubits have to be chained together to logical qubits in order to allow for higher connectivity. Finding the best, so-called embedding, is itself an NP-hard optimization problem, for which we utilize D-Wave's heuristic `MinorMiner`.

Fig. 11 shows the computed embeddings for the subproblem QUBOs with different problem sizes. It is apparent that we are very limited to small problem sizes. Since we do not want too few customers in a chunk to preserve flexibility, we settle at a reasonable middle ground of chunk size six and 12 time steps. We interpolate the original data to 12 time steps and use various (multiples of 6) customer sizes to compare the performance of `Decomp-QPU` against the other solvers. For each subproblem, we take 100 readings from the QPU. We cannot directly steer the timeout in this case. Thus, we first run `Decomp-QPU` and then set the timeout of the remaining solvers to exactly that time. However, `Leap` has a minimum runtime of 5 s, which is the reason why we only include `Leap` in the cases where the `Decomp-QPU` time is more than 5 s, being the case from $N_c = 480$ onward. Embedding times are not considered since the embeddings are computed beforehand and remain constant for one time step, chunk size (N_T, m) combination, independent of grid data.

Again, we perform the analysis for different problem sizes, reaching from 60 to 1920 customers or 720 to 23 040 integer variables. The subproblems comprise 72 integer variables, resulting in 216 binary variables in the QUBO formulation. In contrast to the previous analysis, we additionally investigate SA as a subproblem QUBO solver in this instance. Due to the larger problem sizes of the previous subproblems, the SA routine could not return results within the runtime boundaries we had set. Fig. 12 displays the results of the
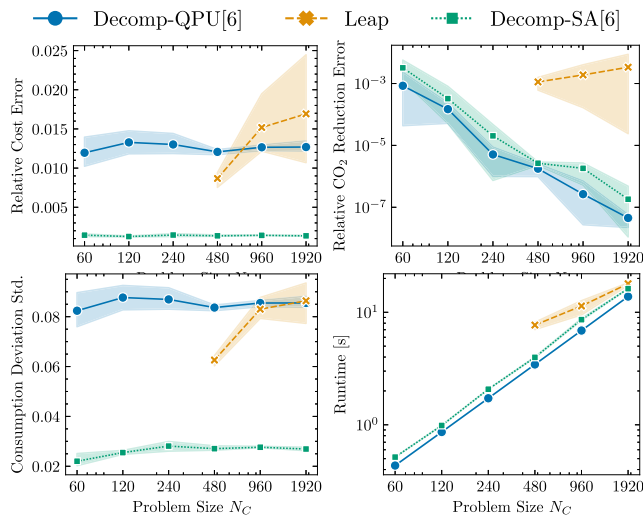
FIGURE 12. All metrics for the small problem sizes, also considering QUBO subproblems that can be solved using D-Wave's QPU. Comparing both Quantum routines, one can observe that `Decomp-QPU` returns results with slightly lower cost than `Leap`. Yet, the CO_2 reduction is vastly better in the decomposed case. Notably, the difference of the `Decomp` solvers in the relative cost error is mostly due to the higher consumption deviation from the QPU results. To match the runtime of the `Decomp-SA` to `Decomp-QPU`, we had to manually adjusted the number of samples from the SA routine to 30. We also observe that `Leap` outruns the set timeout here.

experiments. The previously discussed solvers (`Gurobi`, `Decomp-Gurobi`, `Decomp-Leap`) exhibited similar patterns to the investigation done for the larger problem sizes (see Fig. 7). Therefore, we only focus on the QPU and SA-based decomposition routines and `Leap`.

`Decomp-SA` exhibits the lowest cost error compared to the two quantum-enhanced methods. The two `Decomp` solvers (one using classical and the other using quantum compute) demonstrate similar performance in terms of CO_2 reduction. Interestingly, the consumption deviation and, therefore, also the dominating factor in the cost are measured at a very constant level between the problem sizes. Curiously, SA, as the subsolver, performs better ($\approx 2\%$) concerning consumption deviation than the QPU does ($\approx 8\%$), leading to a gap in the cost. `Leap` exhibits similar performance as in our previous experiments. Most notably, `Decomp-QPU` seems to perform better than `Leap` regarding the optimization objective (1.3% error versus 1.7% error) and concerning the CO_2 reduction (4.5×10^{-8} error versus 3.4×10^{-3} error). That indicates that our developed hybrid quantum routine does seem to outperform the general-purpose `Leap` for this particular task.

VI. CONCLUSION
We explored the feasibility of current QC techniques for DSR by developing a mathematical formulation that utilizes discount scheduling to shift grid load to more appropriate times. Our formulation involves providing discretized discounts to multiple customers at different times to incentivize a load

shift while ensuring the total consumption stays fixed. We chose CO_2 emission reduction as the main objective for our DSP implementation of DSR. With secondary objectives, such as maintaining grid stability and ensuring customer well-being, we formulated a QIP problem.

Upon close inspection of the problem, we developed a custom decomposition algorithm that compartmentalizes the problem into customer chunks. These subproblems involve unconstrained integer optimization and can be effectively addressed on quantum computers if encoded correctly. Moreover, since the problems are solved sequentially, we incorporated the accumulated errors into the subsequent optimization problems. Finally, we developed a postprocessing algorithm that further refines the solution.

In the end, we benchmarked the performance of a classical general-purpose solver against D-Wave's Leap hybrid quantum–classical solver and our customized decomposition method with various (quantum or classical) subsolvers employed. We observed that the classical solver fails to produce acceptable results after a specific problem size when using a linearly increasing timeout for the problem size. In contrast, the quantum-enhanced `Leap` continues to provide satisfactory results. This indicates a potential advantage of solving this particular problem using `Leap` over the purely classical counterpart, `Gurobi`. Nonetheless, the decomposition method with the classical solver as subsolver developed the best-achieved results over the range of problem sizes we investigated. Furthermore, using quantum annealing or SA for the QUBO problems has resulted in good performance. We found that decomposition paired with QA returned comparable energies to `Leap`.

We remark that the pairing of the decomposition method with `Leap` with large chunk sizes might be a promising pathway for utilizing the quantum-enhanced method for huge instances of this problem. This statement requires further experiments, but we argue that solving large subproblems within time constraints may pose challenges for `Gurobi`, whereas `Leap` could yield acceptable results. Further, future work includes the response of the solvers to different problem parameter settings and making the grid constraints more physically realistic rather than our realistic yet pragmatic chosen constant band.

Finally, determining precise energy requirements of QC hardware and the tradeoff between QC algorithms runtime and used energy is an active and interesting area of research [56], [57], [58]. If the community can demonstrate practical quantum advantage (e.g., quantum runtime of seconds or hours as opposed to weeks or years for classical high-performance computing runtime) for use cases that themselves reduce CO_2 emissions, perhaps, this may offset the potential CO_2 cost resulting from the manufacturing or operating future quantum computers. We believe that this work on how quantum computers may solve optimization problems related to the sustainable energy transition by embedding sustainability notions into the use case itself is a progressive step toward using QC for important global issues.

TABLE 4 Symbols and Parameters in the DSP Formulation

Label	Range	Description	Units
c	$\{1, \ldots, N_C\}$	Customer index	
t	$\{1, \ldots, N_T\}$	Time step index	
k	$\{1, \ldots, N_K\}$	Discount type index	
$d_{c,t}$	$\mathbb{R}^+$	Customer power draw	kWh
χ_c	$\mathbb{R}^+$	Customer price elasticity	
I_t	$\mathbb{R}^+$	Grid CO_2 intensity	$\frac{g}{kWh}$
$z_{c,t}$	Z	Discounts	
$z_{\max}$	$[0, 1]$	Maximum discount	
Z	$\{-z_{\max} \ldots z_{\max}\}$	Discount categories	
Q	$\mathbb{N}$	Number of bits in encoding	
$x_{c,t,k}$	$\{0, 1\}$	Binary decision variables	
Δp_t	$\mathbb{R}^+$	Momentary consumption deviation bound	kWh
λ_1	$\mathbb{R}^+$	Consumption deviation penalty	
λ_2	$\mathbb{R}^+$	Discount $L2$-regularization	
λ_3	$\mathbb{R}^+$	Discount change penalty	
ζ_t	$[-z_{\max}, z_{\max}]$	Effective discount	
D_c	$\mathbb{R}^+$	Total consumption per customer	kWh
D_t	$\mathbb{R}^+$	Total consumption per time step	kWh
$\widetilde{D}_t$	$\mathbb{R}^+$	Mutable consumption	kWh
M	$\mathbb{N}$	Number of chunks	
m	$\mathbb{N}$	Chunk size	
j	$\{1, \ldots, M\}$	Chunk index	
ξ_t^j	$[-z_{\max}, z_{\max}]$	Effective chunk discount	
r	$\mathbb{N}$	Postprocessing cutoff	

APPENDIX
GLOSSARY

For a better overview of the used symbols and parameters in the formulation of the DSP, we provide an overview in Table 4.

REFERENCES

[1] J. Torriti, *Peak Energy Demand and Demand Side Response* (ser. Routledge Explorations in Environmental Studies). New York, NY, USA: Routledge, 2015, doi: 10.4324/9781315781099.

[2] P. Siano, "Demand response and smart grids—A survey," *Renewable Sustain. Energy Rev.*, vol. 30, pp. 461–478, Feb. 2014, doi: 10.1016/j.rser.2013.10.022.

[3] H. T. Haider, O. H. See, and W. Elmenreich, "A review of residential demand response of smart grid," *Renewable Sustain. Energy Rev.*, vol. 59, pp. 166–178, Jun. 2016, doi: 10.1016/j.rser.2016.01.016.

[4] N. O'Connell, P. Pinson,, H. Madsen, and M. O'Malley, "Benefits and challenges of electrical demand response: A critical review," *Renewable Sustain. Energy Rev.*, vol. 39, pp. 686–699, Nov. 2014, doi: 10.1016/j.rser.2014.07.098.

[5] J. S. Vardakas, N. Zorba, and C. V. Verikoukis, "A survey on demand response programs in smart grids: Pricing methods and optimization algorithms," *IEEE Commun. Surv. Tut.*, vol. 17, no. 1, pp. 152–178, First Quarter 2015, doi: 10.1109/COMST.2014.2341586.

[6] A. Mizel, D. A. Lidar, and M. Mitchell, "Simple proof of equivalence between adiabatic quantum computation and the circuit model," *Phys. Rev. Lett.*, vol. 99, Aug. 2007, Art. no. 070502, doi: 10.1103/PhysRevLett.99.070502.

[7] A. Das and B. K. Chakrabarti, "Colloquium: Quantum annealing and analog quantum computation," *Rev. Modern Phys.*, vol. 80, pp. 1061–1081, Sep. 2008, doi: 10.1103/RevModPhys.80.1061.

[8] T. Albash and D. A. Lidar, "Adiabatic quantum computation," *Rev. Modern Phys.*, vol. 90, Jan. 2018, Art. no. 015002, doi: 10.1103/RevModPhys.90.015002.

[9] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model," *Phys. Rev. E*, vol. 58, pp. 5355–5363, Nov. 1998, doi: 10.1103/PhysRevE.58.5355.

[10] A. B. Finnila, M. A. Gomez,, C. Sebenik, C. Stenson, and J. D. Doll, "Quantum annealing: A new method for minimizing multidimensional functions," *Chem. Phys. Lett.*, vol. 219, no. 5, pp. 343–348, Mar. 1994, doi: 10.1016/0009-2614(94)00117-0.

[11] C. McGeoch, P. Farre, and W. Bernoudy, "D-wave hybrid solver service + advantage: Technology update," D-Wave, Burnaby, BC, Canada, Tech. Rep. Ser. 14-1048A-A, 2020.

[12] R. Eskandarpour, K. J. B. Ghosh, A. Khodaei, A. Paaso, and L. Zhang, "Quantum-enhanced grid of the future: A primer," *IEEE Access*, vol. 8, pp. 188993–189002, 2020, doi: 10.1109/ACCESS.2020.3031595.

[13] Y. Zhou et al., "Quantum computing in power systems," *iEnergy*, vol. 1, no. 2, pp. 170–187, Jun. 2022, doi: 10.23919/IEN.2022.0021.

[14] D. Tylavsky and G. Heydt, "Quantum computing in power system simulation," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, 2003, pp. 950–956, doi: 10.1109/PES.2003.1270438.

[15] R. Eskandarpour, A. Khodaei, L. Zhang, E. A. Paaso, and S. Bahramirad, "Quantum computing applications in power systems," in *Proc. CIGRE US Nat. Committee Grid Future Symp.*, 2019. [Online]. Available: https://cigre-usnc.org/wp-content/uploads/2019/10/5A_3.pdf

[16] R. Eskandarpour, K. Ghosh, A. Khodaei, and A. Paaso, "Experimental quantum computing to solve network DC power flow problem," 2021, *arXiv:2106.12032*, doi: 10.48550/arXiv.2106.12032.

[17] R. Eskandarpour, K. Ghosh, A. Khodaei, L. Zhang, A. Paaso, and S. Bahramirad, "Quantum computing solution of DC power flow," 2020, *arXiv:2010.02442*, doi: 10.48550/arXiv.2010.02442.

[18] S. DiAdamo, C. O'Meara, G. Cortiana, and J. Bernabé-Moreno, "Practical quantum K-means clustering: Performance analysis and applications in energy grid classification," *IEEE Trans. Quantum Eng.*, vol. 3, 2022, Art. no. 3102316, doi: 10.1109/TQE.2022.3185505.

[19] V. S. Denchev et al., "What is the computational value of finite-range tunneling?," *Phys. Rev. X*, vol. 6, no. 3, Aug. 2016, Art. no. 031015, doi: 10.1103/PhysRevX.6.031015.

[20] A. Lucas, "Ising formulations of many NP problems," *Front. Phys.*, vol. 2, 2014, Art. no. 5, doi: 10.3389/fphy.2014.00005.

[21] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," 2014, *arXiv:1411.4028*, doi: 10.48550/arXiv.1411.4028.

[22] N. Nikmehr, P. Zhang, and M. A. Bragin, "Quantum distributed unit commitment: An application in microgrids," *IEEE Trans. Power Syst.*, vol. 37, no. 5, pp. 3592–3603, Sep. 2022, doi: 10.1109/TPWRS.2022.3141794.

[23] N. Nikmehr, P. Zhang, and M. A. Bragin, "Quantum-enabled distributed unit commitment," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2022, pp. 1–5, doi: 10.1109/PESGM48719.2022.9917029.

[24] S. Koretskyet al., "Adapting quantum approximation optimization algorithm (QAOA) for unit commitment," in *Proc. IEEE Int. Conf. Quantum Comput. Eng.*, 2021, pp. 181–187, doi: 10.1109/QCE52317.2021.00035.

[25] F. Feng, P. Zhang, M. A. Bragin, and Y. Zhou, "Novel resolution of unit commitment problems through quantum surrogate Lagrangian relaxation," *IEEE Trans. Power Syst.*, vol. 38, no. 3, pp. 2460–2471, May 2023, doi: 10.1109/TPWRS.2022.3181221.

[26] A. Ajagekar and F. You, "Quantum computing for energy systems optimization: Challenges and opportunities," *Energy*, vol. 179, pp. 76–89, Jul. 2019, doi: 10.1016/j.energy.2019.04.186.

[27] C. Gambella and A. Simonetto, "Multiblock ADMM heuristics for mixed-binary optimization on classical and quantum computers," *IEEE Trans. Quantum Eng.*, vol. 1, 2020, Art. no. 3102022, doi: 10.1109/tqe.2020.3033139.

[28] Z. Zhao, L. Fan, and Z. Han, "Hybrid quantum benders decomposition for mixed-integer linear programming," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Austin, TX, USA, 2022, pp. 2536–2540, doi: 10.1109/WCNC51071.2022.9771632.

[29] C. O'Meara, M. Fernández-Campoamor, G. Cortiana, and J. Bernabé-Moreno, "Quantum software architecture blueprints for the cloud: Overview and application to peer-2-peer energy trading," in *Proc. IEEE Conf. Technol. Sustain.*, 2023, pp. 191–198, doi: 10.1109/SusTech57309.2023.10129617.

[30] M. Fernández-Campoamor, C. O'Meara, G. Cortiana, V. Peric, and J. Bernabé-Moreno, "Community detection in electrical grids using quantum annealing," 2021, *arXiv:2112.08300*, doi: 10.48550/arXiv.2112.08300.

[31] S. M. Venkatesh, A. Macaluso, and M. Klusch, "BILP-Q: Quantum coalition structure generation," in *Proc. 19th ACM Int. Conf. Comput. Front.*, New York, NY, USA, 2022, pp. 189–192, doi: 10.1145/3528416.3530235.

[32] S. M. Venkatesh, A. Macaluso, and M. Klusch, "GCS-Q: Quantum graph coalition structure generation," in *Proc. 23rd Int. Conf. Comput. Sci.*, 2023, pp. 138–152, doi: 10.1007/978-3-031-36030-5_11.

[33] D. Leib et al., "An optimization case study for solving a transport robot scheduling problem on quantum-hybrid and quantum-inspired hardware," *Sci. Rep.*, vol. 13, no. 1, Oct. 2023, Art. no. 18743, doi: 10.1038/s41598-023-45668-1.

[34] M. Geitz, C. Grozea, W. Steigerwald, R. Stöhr, and A. Wolf, "Solving the extended job shop scheduling problem with AGVs— Classical and quantum approaches," in *Integration of Constraint Programming Artificial Intelligence, and Operations Research*, P. Schaus, Ed., Cham, Switzerland: Springer, 2022, pp. 120–137, doi: 10.1007/978-3-031-08011-1_10.

[35] G. Colucci, S. V. D. Linde, and F. Phillipson, "Power network optimization: A quantum approach," *IEEE Access*, vol. 11, pp. 98926–98938, 2023, doi: 10.1109/ACCESS.2023.3312997.

[36] H. Oshiyama and M. Ohzeki, "Benchmark of quantum-inspired heuristic solvers for quadratic unconstrained binary optimization," *Sci. Rep.*, vol. 12, no. 1, Feb. 2022, Art. no. 2146, doi: 10.1038/s41598-022-06070-5.

[37] G. Chapuis, H. Djidjev, G. Hahn, and G. Rizk, "Finding maximum cliques on a quantum annealer," in *Proc. Comput. Front. Conf.*, New York, NY, USA, 2017, pp. 63–70, doi: 10.1145/3075564.3075575.

[38] G. Hahn and H. Djidjev, "Reducing binary quadratic forms for more scalable quantum annealing," in *Proc. IEEE Int. Conf. Rebooting Comput.*, 2017, pp. 1–8, doi: 10.1109/ICRC.2017.8123654.

[39] E. Pelofske, G. Hahn, and H. N. Djidjev, "Solving large maximum clique problems on a quantum annealer," in *Quantum Technology and Optimization Problems* (ser. Lecture Notes in Computer Science), S. Feld and C. Linnhoff-Popien, Eds., Cham, Switzerland: Springer, 2019, pp. 123–135, doi: 10.1007/978-3-030-14082-3_11.

[40] E. Pelofske, G. Hahn, and H. Djidjev, "Decomposition algorithms for solving NP-hard problems on a quantum annealer," *J. Signal Process. Syst.*, vol. 93, no. 4, pp. 405–420, Apr. 2021, doi: 10.1007/s11265-020-01550-1.

[41] E. Pelofske, G. Hahn, and H. Djidjev, "Solving large minimum vertex cover problems on a quantum annealer," in *Proc. 16th ACM Int. Conf. Comput. Front.*, New York, NY, USA, 2019, pp. 76–84, doi: 10.1145/3310273.3321562.

[42] G. G. Guerreschi, "Solving quadratic unconstrained binary optimization with divide-and-conquer and quantum algorithms," 2021, *arXiv:2101.07813*, doi: 10.48550/arXiv.2101.07813.

[43] Z. Zhou, Y. Du, X. Tian, and D. Tao, "QAOA-in-QAOA: Solving large-scale MaxCut problems on small quantum machines," *Phys. Rev. Appl.*, vol. 19, no. 2, Feb. 2023, Art. no. 024027, doi: 10.1103/PhysRevApplied.19.024027.

[44] T. Tomesh, Z. H. Saleem, and M. Suchara, "Quantum local search with the quantum alternating operator ansatz," *Quantum*, vol. 6, Aug. 2022, Art. no. 781, doi: 10.22331/q-2022-08-22-781.

[45] Z. H. Saleem, T. Tomesh, M. A. Perlin, P. Gokhale, M. Suchara, and M. Martonosi, "Divide and conquer for combinatorial optimization and distributed quantum computation," in *Proc. IEEE Int. Conf. Quantum Comput. Eng.*, Bellevue, WA, USA, 2023, pp. 1–12, doi: 10.1109/QCE57702.2023.00009.

[46] K. Fujii, K. Mizuta, H. Ueda, K. Mitarai, W. Mizukami, and Y. O. Nakagawa, "Deep variational quantum eigensolver: A divide-and-conquer method for solving a larger problem with smaller size quantum computers," *PRX Quantum*, vol. 3, no. 1, Mar. 2022, Art. no. 010346, doi: 10.1103/PRXQuantum.3.010346.

[47] M. Booth, S. P. Reinhardt, and A. Roy, "Partitioning optimization problems for hybrid classical/quantum execution," D-Wave, Burnaby, BC, Canada, Tech. Rep. Ser. 14-1006A-A, 2017.

[48] M. Born and V. Fock, "Beweis des Adiabatensatzes," *Zeitschrift Physik*, vol. 51, no. 3, pp. 165–180, Mar. 1928, doi: 10.1007/BF01343193.

[49] M. W. Johnson et al., "Quantum annealing with manufactured spins," *Nature*, vol. 473, no. 7346, pp. 194–198, May 2011, doi: 10.1038/nature10012.

[50] A. Alberini, W. Gans, and D. Velez-Lopez, "Residential consumption of gas and electricity in the U.S.: The role of prices and income," *Energy Econ.*, vol. 33, no. 5, pp. 870–881, Sep. 2011, doi: 10.1016/j.eneco.2011.01.015.

[51] M. Filippini, "Short- and long-run time-of-use price elasticities in Swiss residential electricity demand," *Energy Policy*, vol. 39, no. 10, pp. 5811–5817, Oct. 2011, doi: 10.1016/j.enpol.2011.06.002.

[52] G. Dutta and K. Mitra, "A literature review on dynamic pricing of electricity," *J. Oper. Res. Soc.*, vol. 68, no. 10, pp. 1131–1145, Oct. 2017, doi: 10.1057/s41274-016-0149-4.

[53] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, Jan. 1965, doi: 10.1093/comjnl/7.4.308.

[54] Gurobi, *Gurobi Optimizer Reference Manual*, Gurobi Optimization, Beaverton, OR, Canada, 2023. [Online]. Available: https://www.gurobi.com/documentation/current/refman/index.html

[55] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, Jan. 1986, doi: 10.1016/0305-0548(86)90048-1.

[56] C. Berger et al., "Quantum technologies for climate change: Preliminary assessment," 2021, *arXiv:2107.05362*, doi: 10.48550/arXiv.2107.05362.

[57] S. Chen, "Are quantum computers really energy efficient?," *Nature Comput. Sci.*, vol. 3, no. 6, pp. 457–460, Jun. 2023, doi: 10.1038/s43588-023-00459-6.

[58] "QEI 2023—Quantum Energy Initiative," 2023. [Online]. Available: https://quantum-energy-initiative.org/qei2023/