IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 September 2023, accepted 26 September 2023, date of publication 29 September 2023,
date of current version 11 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3320800

RESEARCH ARTICLE

Leveraging Knapsack QAOA Approach for Optimal Electric Vehicle Charging

KIMLEANG KEA, (Student Member, IEEE), CHANSREYNICH HUOT, AND YOUNGSUN HAN, (Member, IEEE)
Department of AI Convergence, Pukyong National University, Nam-gu, Busan 48513, South Korea

Corresponding author: Youngsun Han (youngsun@pknu.ac.kr)

This research was supported by Quantum Computing based on Quantum Advantage challenge research (RS-2023-00257994) through the National Research Foundation of Korea (NRF) funded by the Korean government (Ministry of Science and ICT (MSIT)).

ABSTRACT The electric vehicle (EV) industry is currently afflicted with inefficient charging systems. Considering the growing adoption of EVs, optimization strategies for efficient charging, and overcoming constraints such as a limited power supply and extended waiting times, are required. The knapsack algorithm, a classical technique that maximizes value and capacity, enables efficient utilization of the limited available power supply while minimizing waiting times in EV charging scenarios. However, the knapsack problem is notoriously NP-hard, making it difficult to find efficient solutions classically. In this paper, we propose an approach that leverages the quantum approximation optimization algorithm (QAOA) to resolve the EV charging problem using a knapsack-based formulation. By incorporating a knapsack problem constraint into the QAOA, we overcome the limitations of the original QAOA method and provide a potential solution to the knapsack problem. We extensively evaluate and analyze the effectiveness of our approach in finding optimal EV charging solutions in both noise-free simulations and noisy real quantum devices. The proposed method achieves impressive approximation ratios of up to 100% and 50% in noise-free and noisy environments, respectively. Even with a small circuit size, we confirm that our approach can find optimal solutions effectively.

INDEX TERMS Charging problem, electric vehicle, knapsack problem, limited power supply, QAOA, quantum computing.

I. INTRODUCTION

With the increasing global push for environmental sustainability and the urgent need to reduce greenhouse gas emissions, the adoption of electric vehicles (EVs) has increased remarkably [1]. This shift toward eco-friendly transportation has resulted in a surge in the number of EVs on the roads, presenting new challenges regarding the efficient management of electric power supply [2]. The limited power supply infrastructure poses a critical obstacle to the widespread integration of EVs, necessitating the implementation of effective optimization techniques to ensure a smooth transition to electrified transportation systems [3]. One of the primary challenges arising from the proliferation of EVs is the effective management of the energy demand at electric charging stations [4]. The existing power grid frequently faces difficulties in handling the simultaneous charging requirements of numerous EVs, leading to congestion and potential power shortages. In response to this issue, many studies have been conducted to develop optimization techniques for the efficient allocation of limited power resources among charging stations and vehicles. Among the techniques proposed, the knapsack problem has emerged as a compelling method [5].

The knapsack problem is a classical combinatorial optimization problem in computer science and operations research. In the context of the EV charging problem, it involves allocating an available power supply among charging stations and vehicles to maximize energy utilization while minimizing charging time delays. Each EV is represented

The associate editor coordinating the review of this manuscript and approving it for publication was Siddhartha Bhattacharyya.

as a knapsack ''item'' with associated power requirements and time requirements for achieving a full charge. The term ''capacity'' refers to the available power or energy at a charging station [6]. The knapsack-based approach for EV charging aims to provide a simple solution that efficiently utilizes the limited power supply while maximizing the number of charged EVs. This optimization technique holds significant potential for addressing the challenges associated with charging multiple EVs simultaneously. However, the knapsack problem is notorious for its computational complexity, particularly in real-world large-scale scenarios. For instance, 0/1 knapsack problem, each item can either be selected (0) or rejected (1), meaning that we cannot take a fraction of an item value. The complexity of solving the 0/1 knapsack problem is known to be NP-hard, which means that as the number of items increases, the time required to find the optimal solution can grow exponentially [7]. Thus, classical algorithms frequently struggle to find optimal solutions within a reasonable timeframe, hindering the efficient implementation of the knapsack approach for optimizing EV charging [8].

Therefore, quantum computing has emerged as a powerful solution for tackling problems that are inherently challenging to solve efficiently using classical methods. One promising technique for addressing complex optimization problems, including the knapsack problem, is the quantum approximate optimization algorithm (QAOA). This approach combines the strengths of both quantum computing and classical optimization to achieve impressive performance and scalability [9]. QAOA has proven to be effective in handling a wide range of combinatorial optimization problems, and researchers have explored its application to the challenging knapsack problem in various studies [10], [11]. These investigations have demonstrated the potential of QAOA for finding near-optimal solutions to the knapsack problem efficiently. Despite the promising applications of QAOA, there is still a gap in research concerning its application to EV charging problems in the context of the knapsack problem. Moreover, the initial research on QAOA does not address constrained optimization problems, such as the traveling salesman and knapsack problem.

In this paper, we present an approach that harnesses the power of QAOA to tackle the challenging of 0/1 knapsack problem, specifically for the EV charging optimization problem. Our contributions to this study are as follows:

- We utilize a QAOA for the EV charging optimization problem to improve the effectiveness of finding optimal charging solutions within a limited power supply.
- We present efficient quantum circuits capable of encoding knapsack problem-based EV charging into a format optimized for quantum computation.
- We rigorously evaluate our approach across numerous configurations, conducting a comprehensive analysis that showcases the superiority of our approach and highlights its potential for optimizing the EV charging problem.

The remainder of this paper is structured as follows. In Section II, we explore the background knowledge relevant to this study. Section III presents the existing research on EV charging optimization and QAOA regarding the knapsack problem. Our proposed approach and the application of the QAOA to the 0/1 knapsack problem are presented in Section IV. In Section V, we describe and discuss the experimental results of the proposed method. Section VI presents experimental findings, and the paper is concluded in Section VII.

II. BACKGROUND

A. EV CHARGING PROBLEM

Regarding EV charging issues, three main classifications are considered: smart-grid-, aggregator-, and customer-oriented approaches. In the smart-grid-oriented type, researchers prioritize optimization algorithms for flattening the load and ensuring efficient electricity usage. The aggregator-oriented charging method uses control strategies to improve overall EV customer satisfaction. Conversely, the customer-oriented charging approach employs probabilistic methods to minimize charging costs for individual EV users [12]. Similarly, the considerable electricity demand of EVs can surpass the capacity of local power grids, resulting in voltage fluctuations and potential power-supply disruptions. To tackle these challenges, modeling these decisions as optimization problems becomes crucial. The objective is to strike an optimal balance, considering various objectives and constraints. For instance, ''first come, first served'' algorithms prioritize EVs based on their arrival time; however, this approach can result in prolonged waiting times for EV owners who arrive later [13]. The ''shortest time charge first'' is another scheduling approach commonly employed to minimize the total waiting time for EVs at charging stations [14]. However, this approach falls short in considering crucial factors, such as the charging station power capacity and prioritizing vehicles with urgent charging needs. To address this pressing issue, the knapsack-based algorithm has emerged as a promising solution [15].

B. KNAPSACK PROBLEM

The knapsack problem indeed stands as a classic optimization challenge in computer science and mathematics. It involves selecting items with different weights and values to fit in a container with a limited capacity, aiming to maximize the total value without exceeding the weight limit [16]. Notably, the knapsack problem exemplifies a classic example of an NP-hard problem, signifying the absence of an efficient algorithm capable of finding the optimal solution within polynomial time [17]. The non-linearity knapsack problem (NLK) is a generalized variant of the knapsack problem that has an additional nonlinear cost term in the objective function [18]. The non-linearity problem size scales because adding just one more item to the set of available items can significantly increase the number of possible combinations

that must be considered when solving the problem. As a solution, a groundbreaking paradigm called quantum computing, which leverages the unique properties of quantum mechanics, has been explored to execute operations and address NP-hard problems that present significant challenges for classical computers, including the knapsack problem [8]. Remarkably, quantum algorithms, such as the QAOA, have exhibited an impressive capability to handle the knapsack problem and various other NP-hard challenges [9].

C. QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM (QAOA)

QAOA is a hybrid quantum–classical algorithm employed to solve combinatorial problems with approximate solutions [9]. To prepare the state of QAOA, a parameterized circuit with p levels is employed, driven by $2p$ variational parameters. Notably, even at the lowest circuit depth ($p = 1$), QAOA has demonstrated remarkable performances, which cannot be effectively reproduced using classical computers [19]. Theoretical analysis reveals that the solution quality represented by an approximation ratio, denoted as r, reaches 100% when p approaches ∞. In the context of combinatorial optimization problems [20], we can define problems on N-bit binary strings, denoted as x, where $x = x_0, x_1, \ldots, x_{N-1}$, and the goal is to determine a string that maximizes a given objective function (f). Function f is expressed as follows:

$$f(x) : \{0, 1\}^N \longrightarrow R. \tag{1}$$

We define the phase Hamiltonian (H_C) to map to the objective function f. The form of H_C is denoted as follows:

$$H_C|x\rangle = f(x)|x\rangle. \tag{2}$$

H_C encodes f and acts on the computational basis states of 2^N dimensional Hilbert space. Thus, phase operator, $U(C, \gamma)$, is introduced with γ as a parameter, as follows:

$$U(C, \gamma) = e^{-i\gamma H_C} \tag{3}$$

However, H_C does not change the probability of obtaining a certain basis state. Thus, the mixing operator, $U(B, \beta)$, serves to transfer probability amplitudes between states. It is mathematically defined by the sum of Pauli-X operators σ_j^x over N qubits, where σ_j^x acts as a NOT operator $\sigma_j^x|1\rangle = |0\rangle$ and $\sigma_j^x|0\rangle = |1\rangle$. The mixing operator is expressed as:

$$U(B, \beta) = e^{-i\beta H_B}, \tag{4}$$

where the mixing Hamiltonian (H_B) is defined as follows:

$$H_B = \sum_{j=0}^{N-1} \sigma_j^x. \tag{5}$$

We define the state of the p-level QAOA by applying the phase operator $U(C, \gamma)$ and mixing operator $U(B, \beta)$, as follows:

$$|\gamma, \beta\rangle = U(B, \beta_p)U(C, \gamma_p) \ldots U(B, \beta_1)U(C, \gamma_1)|0\rangle, \tag{6}$$

where $p \geq 1$ and $2p$ angle parameters $\gamma_1 \ldots \gamma_p$ and $\beta_1 \ldots \beta_p$. We can perform the measurement repeatedly to obtain the expectation value of H_C determined as:

$$\langle H_C\rangle := \langle \gamma, \beta|H_C|\gamma, \beta\rangle = \langle f\rangle_{(\gamma,\beta)} \tag{7}$$

where $\langle f\rangle$ represents the expectation value of the objective function which can be obtained by using classical optimization algorithms [21], [22], [23].

For combinatorial optimization problems, finding optimal solutions in polynomial time is difficult. Approximation algorithms offer a fast alternative, providing approximate solutions with an approximation ratio defined as:

$$r = \frac{\langle f\rangle_{(\gamma_{op},\beta_{op})}}{f_{max}} \tag{8}$$

where γ_{op} and β_{op} are optimal parameters.

III. RELATED WORKS

We reviewed several relevant research works that explore the EV charging problem and its connection to the QAOA knapsack problem. In [24], the authors discuss the optimization of EV charging and discharging scheduling to minimize the total cost of charging EVs using both global and local optimal scheduling schemes. However, as the number of EVs and charging stations increases, the computation times become impractical for large-scale EV charging networks. Zhu et al. [25] applied mean-field game theory to optimize EV charging by treating all EVs as a collective unit, leading to consistent charging speeds and uniform charge distribution. However, practical implementation requires further improvements as this approach assumes simultaneous charging and incomplete battery charging. Conversely, a competitive algorithm is proposed for the online multiple knapsack problem, with particular emphasis on EV charging [15]. This algorithm aims to minimize the overall cost of charging EVs while ensuring that each EV has an adequate charge to complete its assigned tasks. The study focuses on modeling the problem as a fractional multiple knapsack problem, which may require more computational resources compared to solving the original model. To overcome this challenge, the QAOA has been proposed specifically for addressing the knapsack problem [26]. In this particular study, two techniques are discussed. The first technique enables the quantum optimization algorithm to explore possible solutions around the initial greedy solution. The second technique aims to guide the quantum exploration and avoid local minima around the greedy solutions. The results indicate that the adjusted quantum optimization heuristics typically outperform various classical heuristics. In [27], the authors employed the QAOA to tackle the knapsack problem, specifically for the battery revenue optimization problem. To the best of our knowledge, this was the first investigation regarding the application of QAOA to a non-Ising objective function, which is the knapsack problem. The authors created two variants of the QAOA to tackle the knapsack problem

by adding constraints to the problems. However, the solution quality of QAOA is characterized by a high degree of parameters chosen by the classical optimizer. To address this challenge, Roch et al. [28] applied the cross-entropy method, effectively shaping the parameter landscape. This adaptation enables the classical optimizer to discover better parameters, leading to a significantly improved performance of the QAOA. However, the authors did not explore larger problem instances involving more than just two knapsack items.

IV. QAOA FOR THE KNAPSACK PROBLEM

A. KNAPSACK EV FORMULATION

Here, we present a comprehensive description of the knapsack approach for the EV charging problem using mathematical notation. In this problem, N EVs are given, with each EV requires a certain amount of power by a given time slot, p_j, and a charging time for a full charge, t_j, where $j = 0, \ldots, N-1$. The EVs to be charged must be chosen such that the total power required for the charging is less than or equal to the maximum power P provided by the station, represented by a binary variable, $x_j \in \{0, 1\}$. The objective is to find a feasible choice x that enables the prioritization of EVs with the shortest charging-time requirements to charge as many EVs as possible. The problem is defined as follows:

$$maximize \sum_{j=0}^{N-1} (1/t_j)x_j \tag{9}$$

$$subject\ to \sum_{j=0}^{N-1} p_j x_j \leq P \tag{10}$$

The knapsack algorithm primarily aims to maximize certain parameters; however, when considering the EV charging time, our objective shifts to minimization. To achieve this, we adopt a unique approach, where the charging time required for EV_j is represented as $(1/t_j)$, implying that a shorter charging time holds a greater value than a longer one. Consequently, this approach provides an optimization method that prioritizes EVs with the shortest charging-time requirement and EVs that can maximize the amount of power available at the charging station.

B. DESIGN OF QAOA FOR THE KNAPSACK EV CHARGING PROBLEM

Inspired by [27], we present an approach based on the constraint $p(x) \leq P$ of the knapsack problem by introducing a penalty scaling linearly with the amount by which the maximum power required P is exceeded.

1) QUBITS REQUIREMENT

We utilize the register R to store the choices within the EV charging problem, while the register A stores the EV power requirement calculation. A flag qubit F indicates whether the constraint is satisfied or not. We denote N_R, N_A, and N_F represent the number of qubits in R, A, and F, respectively. Thus, the number of qubits required is $N_R + N_A + N_F$.

2) OBJECTIVE FUNCTION

Each possible choice of any of the total N number of EVs is represented by a bitstring $x \in Z(N)$, where $Z(N) = \{0, 1\}^N$. The objective function can be formulated as follows:

$$f : Z(N) \to \{0, 1\},\ x \mapsto t(x) = \sum_{j=0}^{N-1} (1/t_j)x_j \tag{11}$$

Thus, the optimal solution (S) of the choices is given by:

$$S = \left\{ x \in Z(N) : p(x) = \sum_{j=0}^{N-1} p_j x_j \leq P \right\}. \tag{12}$$

where $x \in Z(N)$ represents a feasible choice of EVs to be charged. A choice is made feasible, $x \in S$, if and only if $p(x) \leq P$. The EV power required $p(x)$ can be calculated using an ancilla register A and variable x is set to 1 if the EV is selected to charge and 0 otherwise.

3) CONSTRAINT ENFORCEMENT

Here, we utilize an approach that enables the incorporation of the constraints directly into the objective function f. The central idea is to introduce a linearly scaling soft constraint that effectively enforces $p(x) \leq P$. We define the total time required function $T(x)$ and a $penalty(x)$ as follows:

$$T(x) = \sum_{j=0}^{N-1} (1/t_j(x)). \tag{13}$$

The objective function, $f(x)$, can be defined as

$$f(x) = T(x) + penalty(x), \tag{14}$$

with

$$penalty(x) = \begin{cases} 0, & \text{if } p(x) \leq P \\ -\alpha(p(x) - P), & \text{if } p(x) > P \end{cases} \tag{15}$$

where $\alpha > 0$ is a constant parameter for scaling the penalty factor. When choosing an EV to charge, any EV whose power requirement exceeds the available power or violates the constraint must be subject to a penalty.

4) EV TIME COMPUTATION

Here, we detail the realization of the circuit to calculate the EV time requirements. We define the unitary U_C corresponding to $f(x)$ as follows:

$$U(C, \gamma)|x\rangle = e^{-i\gamma f(x)}|x\rangle. \tag{16}$$

Incorporating Equation 14:

$$U(C, \gamma)|x\rangle = e^{-i\gamma T(x)} e^{-i\gamma penalty(x)}|x\rangle \tag{17}$$

The amount of time required for an EV to reach full charge can be directly applied using phase gate P:

$$P(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix} \tag{18}$$

We denote the $P_{R[j]}$ a phase gate P applied to $R[j]$ for register R to store each EV time requirements. Therefore, we can

compute the total time required $T(x)$ using the following equation:

$$e^{-i\gamma T(x)}|x\rangle = \prod_{j=0}^{N-1} P_{R[j]}(-\gamma(1/t(x))_j)|x\rangle. \quad (19)$$

The circuit corresponding to the implementation of time computation is depicted in Figure 1.

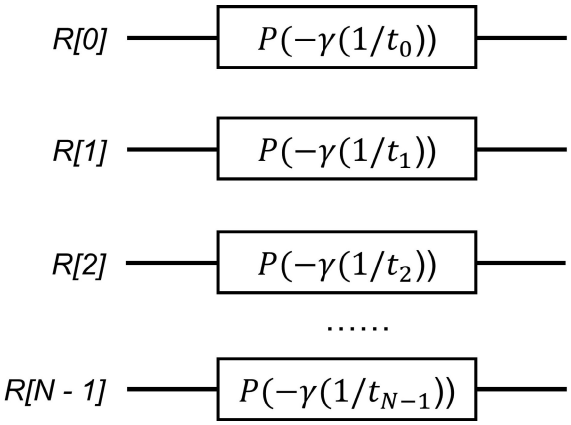


FIGURE 1. Phase gates based on the EV time required value on the register *R* to calculate the total time requirements.

5) PENALTY COMPUTATION

Here, we present the penalty computation implementation. Given the tight coupling of the penalty with the EV power requirement, we decomposed its computation into four subroutines: EV power calculation, constraint testing, penalty dephasing, and reinitialization.

1) **EV power calculation.** We need to perform computations on the entangled states $|x\rangle \otimes |p(x)\rangle$. The total power required $|p(x)\rangle$ is encoded and stored in the register A, where the EV power required addition is calculated using an algorithm based on the quantum Fourier transform (QFT) [29]. Figure 2 illustrates the implemented circuit for the addition of each EV power required using the QFT algorithm. The block *Add* scales linearly with the size of register A, which can be expressed as: $p(x) = \sum_{j=0}^{N-1} A[j]$.

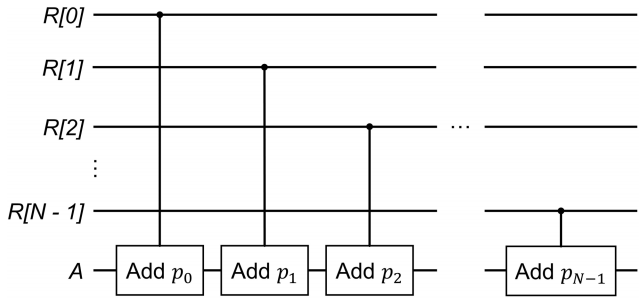


FIGURE 2. Addition of EV power required is done using the QFT algorithm on the register *A* controlled by register *R*.

2) **Constraint testing.** We study the constraint of $p(x) \leq P$ and toggle F to 0 if the condition is satified, 1 otherwise. First, we assume the max power available $P = 2^c$, where c is an integer. Then, multiple condition can be defined as $p(x) \leq P \iff \cap_{j=c}^{N-1}(A[j] = 0)$. The circuit presented in Figure 3 sets F to 1 if and only if $\cap_{j=c}^{N-1}(A[j] = 0) \iff \cap_{j=c}^{N-1}(\neg A[j] = 1)$. The multiple condition constraints can be tested by using a series of multiple C-NOT gates.

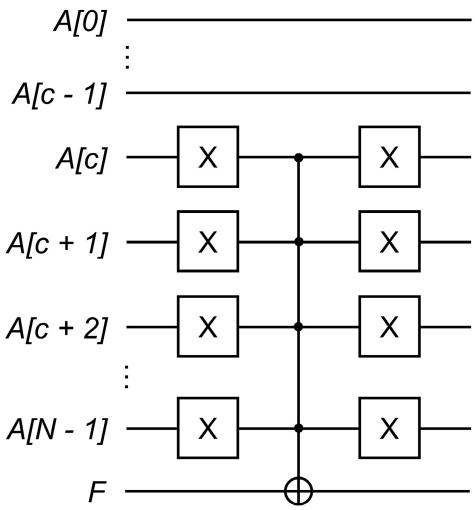


FIGURE 3. The multiple conditional C-NOT for implementing the multiple constraint testing.

3) **Penalty dephasing.** In case the condition constraints are violated, in which $F = 1$, it is necessary to apply a penalty phase $e^{-i\gamma penalty(x)}$ corresponding to the penalty value. Otherwise, $penalty(x) = 0$, and no phase is applied. The implementation is done using phase gates controlled by a flag qubit F. Consider a general case where max power P can be any integer, not just limited to the power of 2. We introduce a constant P_0, which adds to A after calculating $p(x)$, then $P+P_0 = 2^c$, which can be defined as follows:

$$p(x) \leq P \iff p(x) + P_0 \leq P + P_0 = 2^c \quad (20)$$

Given the penalty for $p(x) > P$ in Equation 15:

$$penalty(x) = -\alpha(p(x) - P). \quad (21)$$

With Equation 20, we have:

$$-\alpha(p(x) - P) = -\alpha((p(x) + P_0) - (P + P_0)) \quad (22)$$

The phase $-\alpha(p(x)+P_0)$ stored in register A is defined as follows:

$$-\alpha(p(x) + P_0) \iff \sum_{j=0}^{N-1} e^{i2^j\gamma\alpha A[j]} \quad (23)$$

and the phase $\alpha(P + P_0)$ applied using phase gate to flag qubit F is defined as:

$$\alpha(P + P_0) \Longleftrightarrow e^{-i\gamma\alpha(P+P_0)} = e^{-i\gamma\alpha 2^c} \tag{24}$$

The penalty value can be directly applied using phase gate P, which is controlled by flag qubit F. Therefore, the circuit for the penalty dephasing implementation is presented in Figure 4.

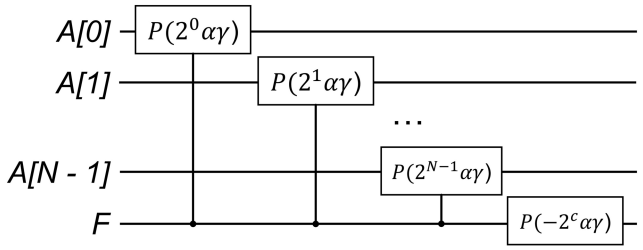


FIGURE 4. A circuit implementing for applying a phase gate corresponding to the penalty value.

4) **Reinitialization.** After the penalty computation, we restore the ancilla qubits by returning them to their original state, where all of them are set to $|0\rangle$. The EV power calculation and Constraint testing subroutines use only reversible gates such as NOT and C-NOT gates. Consequently, to reset the ancilla qubits, we compute the circuit in a reverse direction.

C. CIRCUIT IMPLEMENTATION AND COMPLEXITY

Here, we utilize the following notations: P_t represents the time required computation for the phase gates, as depicted in Figure 1. P_c stands for the EV power computation and Constraint testing subroutines, as illustrated in both Figure 2 and Figure 3. P_p corresponds to the penalty dephasing shown in Figure 4. Finally, $(P_c)^\dagger$ denotes the reinitialization subroutines, responsible for resetting the register A to the state $|0\rangle$. Figure 5 presents the complete circuit implementation, illustrating the main steps of designing QAOA for knapsack-based EV charging problem at circuit depth $p = 1$.

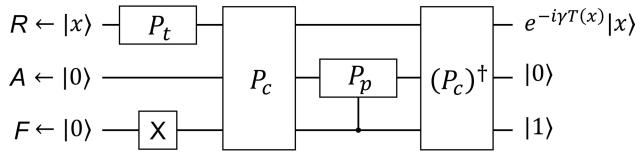


FIGURE 5. A complete overview circuit for QAOA knapsack in EV charging problem.

The P_t is constantly computed in time $O(1)$, and it does not require any ancilla qubits. The P_p operates with a time complexity of $O(\log_2 n)$ and requires N_A number of ancilla qubits. The complexity of the entire algorithm is highly dependent on the depth of P_c. The *Add* function is computed in time $\log_2(d) * N_A = O(\log_2(n))$, as d is a constant [27]. We can obtain the N_A by computing all possible EV power required in register A; it is defined as:

$$N_A = \left\lceil \log_2 \sum_{j=0}^{N-1} p_j \right\rceil + 1 \tag{25}$$

As discussed previously, we define the number of qubits requirement by:

$$N_R + N_A + N_F \Longleftrightarrow N_R + \left\lceil \log_2 \sum_{j=0}^{N-1} p_j \right\rceil + 1 + 1 \tag{26}$$

where N_R is the number of qubits corresponding to the total number of EVs in the knapsack problem and N_F is the number of flag qubits, i.e., one.

TABLE 1. EV charging knapsack problem instances.

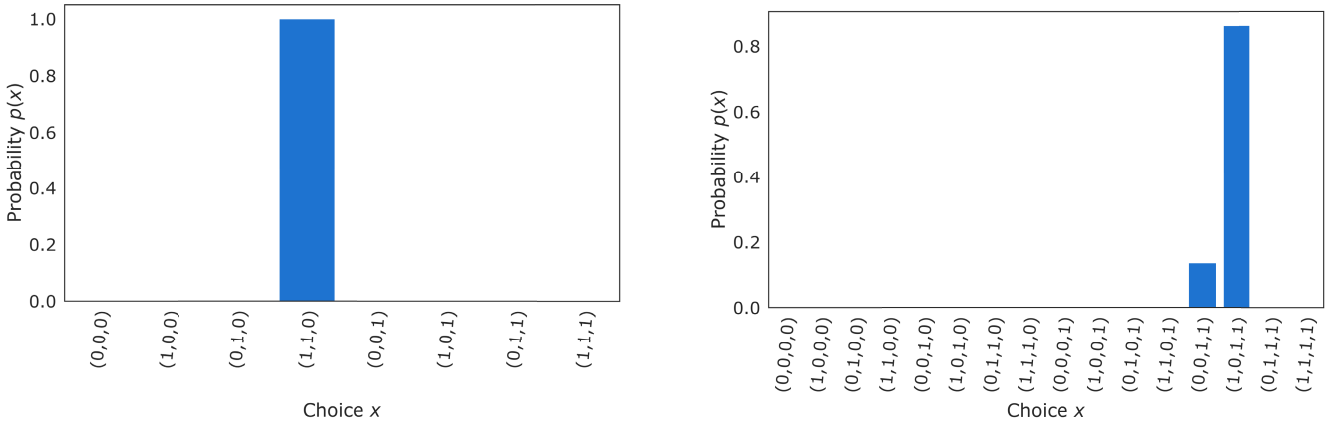
Problem	Time	Power	Max Power	Qubits	BKS
A	[1, 2]	[1, 1]	1	5	(1, 0)
B	[1, 2]	[1, 1]	2	6	(1, 1)
C	[1, 1, 2]	[1, 1, 1]	2	7	(1, 1, 0)
D	[1, 2, 3]	[1, 2, 3]	3	7	(1, 1, 0)
E	[2, 3, 1, 1]	[2, 2, 1, 1]	4	9	(1, 0, 1, 1)

V. SIMULATION AND RESULT

In this section, we present the implementation details of our proposed approach for the EV charging problem. Additionally, we explore its effectiveness across different problem instances and parameters.

A. IMPLEMENTATION SETUP

We implemented our proposed approach using the IBM Qiskit library and conducted experiments with the QASM simulator and IBM Nairobi real quantum computer [30]. Since quantum computing hardware is costly, simulating quantum algorithms on classical computers has proven effective for exploring and analyzing the quality of quantum algorithms. However, this approach has computational overhead, limiting it to a few qubits within a reasonable computing time. For this reason, we reduced the number of EVs per knapsack problem to manage the qubit resources. Nevertheless, our approach remains theoretically applicable to larger problems. For optimizing the $2p$ angle parameters (β, γ) in our approach, we choose the classical optimization algorithm SHGO [31]. To evaluate the algorithm performance, we employed a set of pre-defined knapsack problems listed in Table 1. Each entry represents an EV charging scenario, where the time-required indexes correspond to the power-required indexes. Our main objective was carefully selecting EVs with the lowest time requirements while maximizing the maximum power supply. In Table 1, the Best Known Solution (BKS) is the result using a classical algorithm [32] and representing the selected index of the EVs. A value of 1 indicates that the corresponding EV is selected, while 0 indicates that it is not selected.



(a) Problem C (b) Problem E

FIGURE 6. Probability distributions of our proposed approach on Problem *C* and *E*. In Problem *C* scenario, experiment with three EVs and have a choice (1, 1, 0) distributed up to 100%. In Problem *E*, the situation involves four EVs with choice preferences (1, 0, 1, 1) distributed up to 85%. To achieve higher probabilities, a larger circuit depth *p* is needed.

B. PERFORMANCE EVALUATION

The solution quality of the QAOA is profoundly influenced by the circuit depth p and its $2p$ angle parameters. In our approach, we also consider the penalty parameter α, which significantly impacts the solution quality. First, we optimize the circuit depth p parameter and examine the corresponding probability distributions. Next, we introduce the approximation ratio achieved by our approach and compare these results to those obtained using the QAOA cross entropy (QAOA-CE) method [28]. In their proposed, the cross entropy method is used for the classical optimizer to find optimal $2p$ angle parameters shortly. Additionally, we conduct an analysis highlighting the outcomes of noise-free simulation with those of a noisy quantum computer. Finally, we conduct an ablation study by systematically varying both p and α parameters, exploring a wide range of results.

1) PROBABILITY DISTRIBUTION

We present the probability distributions generated by our approach for Problems C and E from Table 1. For our experiments, we thoughtfully select a circuit depth of $p = 3$ and a penalty of $\alpha = 10$ to obtain the optimized $2p$ angle parameters. The results are depicted in Figure 6, where each subplot showcases the corresponding probability distributions for the parameter configuration. In Figure 6a, the probability distribution prominently peaks at the choice of $(1, 1, 0)$, indicating that the EVs with the time required at indexes 0 and 1 and power required at indexes 0 and 1 are the preferred choices. Similarly, in Figure 6b, the outcomes reveal a strong preference for the $(1, 0, 1, 1)$ choice, signifying the selection of EVs time and power requirements at indexes 0, 2, and 3. However, the outcome of Problem E indicates a probability associated with a non-optimal choice $(0, 0, 1, 1)$. This observation suggests that a higher p value is required to achieve a more optimal distribution. By comparing the distribution generated by our approach to the BKS value, we ensure that our method behaves as expected. Our approach prioritizes EVs with shorter charging time requirements and maximizes power supply, thus optimizing the charging strategy.

2) APPROXIMATION RATIO

We conducted a comprehensive evaluation of our approach, focusing on the approximation ratio. In this evaluation, we varied the QAOA depth $p = 1, 2, 3$ and explored the impact of the variation. The outcomes obtained from our approach are illustrated in Figure 7, allowing us to compare the results across different circuit depth p values. To establish a benchmark for comparison with other approaches, we defined the problem instances based on the specifications provided in the QAOA-CE paper [28]. The results of our study clearly demonstrate the superiority of our approach over the QAOA-CE method at all p values. As p increased, our proposed approach consistently approaches an approximation ratio of 100%, showcasing its exceptional ability to optimize overall solution qualities more effectively than the classical knapsack problem, particularly as quantum computing continues to advance [26].

Furthermore, we performed experiments on the problems A to D under both noise-free and noisy conditions. The problem E was omitted because it demands a number of qubits that exceed the capacity supported by IBM Nairobi. The results in Figure 8 reveal that a 100% approximation ratio was achieved in the noise-free simulation, whereas in the noisy quantum device, the approximation ratio dropped to 50%. That is because the current real quantum devices are susceptible to noise interference, leading to errors in circuit outcomes. The level of errors depends on factors such as the number of qubits, circuit depth, and the specific quantum device used [33]. Current quantum devices are impacted by various errors, including gate errors and readout errors. Addressing those noise issues would require the implementation

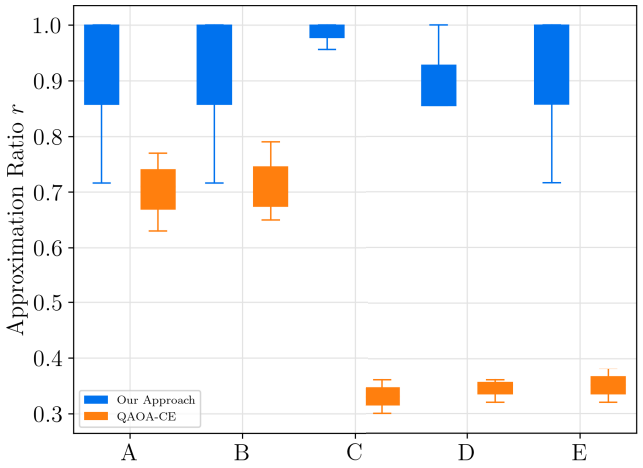


FIGURE 7. Approximation ratio of our approach with comparison for different $p = 1, 2, 3$ to QAOA-CE method on all problems.

of various error-mitigation techniques [34], [35], which is beyond the scope of this study.

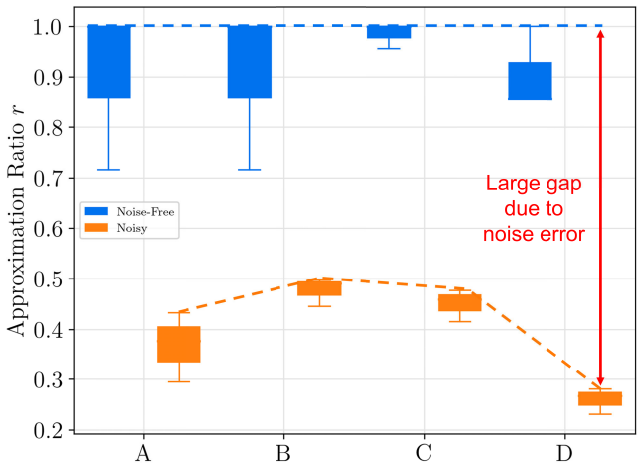


FIGURE 8. Comparing the approximation ratios of our approach on both a noise-free simulator and a noisy quantum device for problems *A* to *D*. While our method demonstrates high performance on a noise-free simulator, we also observe the impact of noise in the current quantum computer, which affects the overall solution quality of our approach.

3) ABLATION STUDY

We conducted experiments to analyze the importance of parameter values in our approach. We varied the p values and studied their impact on the approximation ratio. Additionally, we compared scenarios with different α values with those without penalties to understand their influence on solutions.

- **Circuit depth.** Theoretically, as the value of p increases, the obtained approximation ratios should also increase. Figure 9 illustrates that our approach achieves an approximation ratio of 100% for all problems when the depth $p \geq 4$. This finding demonstrates the remarkable reliability of our method in identifying optimal solutions, even at low circuit depths.

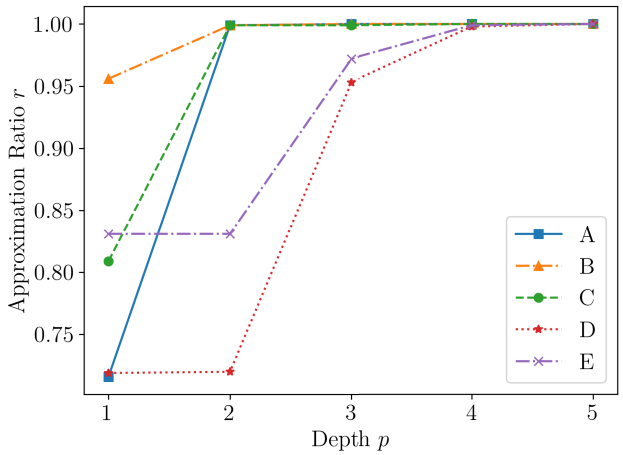


FIGURE 9. The approximation ratio of the circuit depth p ranges from 1 to 5 across all problems. Notably, when the depth $p \geq 3$, the approximation ratio reaches up to 100%. This finding suggests that achieving higher-quality solutions necessitates a greater circuit depth.

- **Penalty scaling factor.** To investigate the influence of penalties, we conducted experiments with a circuit depth $p = 3$ to optimize the β, γ parameters. Moreover, we varied the α value over a range of 0 to 10 in increments of 2. Subsequently, we computed the corresponding approximation ratios. The results are presented in Table 2. For all problems, the approximation ratio reaches 100% when the α value reaches its maximum value of 10. Conversely, when the α value is 0, the approximation ratio drops to 0%, highlighting the impact of the penalty value on the solution qualities.

TABLE 2. Approximation ratio of penalty factor α from 0 to 10. The choice of penalty value influences the approximation ratio, showing the importance of careful selection.

Problem	$\alpha = 0$	$\alpha = 2$	$\alpha = 4$	$\alpha = 6$	$\alpha = 8$	$\alpha = 10$
A	0	0.99	0.99	0.99	1	1
B	0	0.98	0.99	0.99	1	1
C	0	0.97	0.99	0.99	1	1
D	0	0.93	0.94	0.96	0.99	1
E	0	0.94	0.95	0.97	0.99	1

4) EXECUTION TIME

The quantum circuit execution time in quantum devices varies with device technology. However, queuing can lead to hours-long waits for circuits that execute in mere seconds. The execution time corresponds to the duration required for generating the probability distribution of the quantum circuit [36]. Table 3 illustrates the execution time to generate probability distribution with circuit depth $p = 3$, which run on the real quantum device.

VI. DISCUSSION

On examining the problems in Table 1 and the results from the previous section, there are a few notable findings. First,

TABLE 3. Execution time of problems *A* to *D*.

Problem	Execution Time (sec)
A	5
B	5
C	6
D	7

the limited number of available qubits in this study restricts the scale of the EV charging problem that can be considered. This limitation emphasizes the need for an increased number of qubits to handle relatively large problem sizes effectively. Second, in Figure 6b, we observe that for Problem E, non-optimal solutions are chosen because of the low circuit depth $p = 3$. This indicates a strong dependence of the optimal solution on a larger circuit depth. However, as the size of the gate parameters increases polynomially with the circuit depth p, classical optimization methods face challenges in efficiently finding optimal $2p$ angle parameters within a short timeframe. Third, as depicted in Figure 8, a 100% approximation ratio is achieved in the noise-free simulation. While it is likely that complicated classical algorithms may surpass our quantum algorithms in performance, quantum computing holds promising potential for solving complex problems more efficiently in the future. However, when running on a noisy quantum computer, the results are significantly affected, with the approximation ratio reaching only 50%. This observation strongly indicates that the presence of noise in current quantum computers has a substantial impact on the overall solution quality. Fourth, Table 2 demonstrates the variation of the approximation ratio with the α value from 0 to 10 for different problems. The optimization of the α value should be performed individually for each problem instance rather than applying a specific penalty value universally across all problems. Finally, our quantum circuit executes within seconds as shown in Table 3, whereas the queuing time can range from minutes to hours, or even days.

VII. CONCLUSION

The existing EV charging systems are characterized by extended waiting times and constrained power supply, necessitating optimization. Therefore, we introduced a promising solution in the form of the knapsack algorithm, which effectively minimizes waiting times while optimizing the utilization of available power supply. In this research, we proposed an extended QAOA approach by introducing constraints, which solved the limitation of the original QAOA. By exploiting quantum computing, we applied the QAOA to the knapsack problem, enabling the efficient identification of optimal solutions. We thoroughly evaluated the performance of our approach using five different EV charging problem scenarios. This evaluation encompassed various aspects, including the QAOA circuit depth, comparisons with the QAOA-CE method, and the variation in the penalty values. Metrics such as the probability distribution and approximation ratio were employed to measure the effectiveness of the approach. The results demonstrated the high efficiency of our proposed approach, which outperformed the QAOA-CE method in all the test scenarios. Remarkably, even with small circuit sizes, our approach achieved impressive approximation ratios of up to 100% in noise-free simulations, underscoring its superiority. However, it is important to acknowledge that in noisy quantum devices, the approximation ratio was reduced to 50%. Overall, these findings highlight the potential of our approach in enhancing solution qualities. In future works, we will focus on scaling our approach to handle larger problem scenarios. Moreover, we intend to employ advanced error-mitigation techniques to enhance the reliability and robustness of our approach.

CODE AVAILABILITY

The code that supports the findings of this study is openly available in the Github repository, https://github.com/QCL-PKNU/QAOA-Knapsack-EV-Charging.

REFERENCES

[1] A. Gulzari, Y. Wang, and V. Prybutok, "A green experience with eco-friendly cars: A young consumer electric vehicle rental behavioral model," *J. Retailing Consum. Services*, vol. 65, Mar. 2022, Art. no. 102877.
[2] A. Dubey and S. Santoso, "Electric vehicle charging on residential distribution systems: Impacts and mitigations," *IEEE Access*, vol. 3, pp. 1871–1893, 2015.
[3] S. F. Tie and C. W. Tan, "A review of energy sources and energy management system in electric vehicles," *Renew. Sustain. Energy Rev.*, vol. 20, pp. 82–102, Apr. 2013.
[4] T. S. Bryden, G. Hilton, A. Cruden, and T. Holton, "Electric vehicle fast charging station usage and power requirements," *Energy*, vol. 152, pp. 322–332, Jun. 2018.
[5] R. Gopalakrishnan, A. Biswas, A. Lightwala, S. Vasudevan, P. Dutta, and A. Tripathi, "Demand prediction and placement optimization for electric vehicle charging stations," 2016, *arXiv:1604.05472*.
[6] W. Wang, L. Liu, J. Liu, and Z. Chen, "Energy management and optimization of vehicle-to-grid systems for wind power integration," *CSEE J. Power Energy Syst.*, vol. 7, no. 1, pp. 172–180, Jan. 2021.
[7] A. Fréville, "The multidimensional 0–1 knapsack problem: An overview," *Eur. J. Oper. Res.*, vol. 155, no. 1, pp. 1–21, May 2004.
[8] A. Awasthi, F. Bär, J. Doetsch, H. Ehm, M. Erdmann, M. Hess, J. Klepsch, P. A. Limacher, A. Luckow, C. Niedermeier, L. Palackal, R. Pfeiffer, P. Ross, H. Safi, J. Schönmeier-Kromer, O. von Sicard, Y. Wenger, K. Wintersperger, and S. Yarkoni, "Quantum computing techniques for multi-knapsack problems," 2023, *arXiv:2301.05750*.
[9] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," 2014, *arXiv:1411.4028*.
[10] S. Hadfield, Z. Wang, B. O'Gorman, E. Rieffel, D. Venturelli, and R. Biswas, "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz," *Algorithms*, vol. 12, no. 2, p. 34, Feb. 2019.
[11] S. Bravyi, A. Kliesch, R. Koenig, and E. Tang, "Obstacles to variational quantum optimization from symmetry protection," *Phys. Rev. Lett.*, vol. 125, no. 26, Dec. 2020, Art. no. 260505.
[12] Q. Wang, X. Liu, J. Du, and F. Kong, "Smart charging for electric vehicles: A survey from the algorithmic perspective," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1500–1517, 2nd Quart., 2016.
[13] X. Fernandes, J. Rebelo, J. Gouveia, R. Maia, and N. B. Silva, "On-off scheduling schemes for power-constrained electric vehicle charging," *4OR*, vol. 15, no. 2, pp. 163–181, Jun. 2017.
[14] B. Alinia, M. H. Hajiesmaili, and N. Crespi, "Online EV charging scheduling with on-arrival commitment," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4524–4537, Dec. 2019.

[15] B. Sun, A. Zeynali, T. Li, M. Hajiesmaili, A. Wierman, and D. H. K. Tsang, "Competitive algorithms for the online multiple knapsack problem with application to electric vehicle charging," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 4, no. 3, pp. 1–32, Nov. 2020.
[16] H. Kellerer, U. Pferschy, D. Pisinger, H. Kellerer, U. Pferschy, and D. Pisinger, *Multidimensional Knapsack Problems*. Cham, Switzerland: Springer, 2004.
[17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, 2022.
[18] S. Elhedhli, "Exact solution of a class of nonlinear knapsack problems," *Oper. Res. Lett.*, vol. 33, no. 6, pp. 615–624, Nov. 2005.
[19] E. Farhi and A. W. Harrow, "Quantum supremacy through the quantum approximate optimization algorithm," 2016, *arXiv:1602.07674*.
[20] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Cham, Switzerland: Springer, 2012.
[21] J. Choi and J. Kim, "A tutorial on quantum approximate optimization algorithm (QAOA): Fundamentals and applications," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, 2019, pp. 138–142.
[22] G. G. Guerreschi and A. Y. Matsuura, "QAOA for max-cut requires hundreds of qubits for quantum speed-up," *Sci. Rep.*, vol. 9, no. 1, p. 6903, May 2019.
[23] R. Shaydulin, H. Ushijima-Mwesigwa, C. F. A. Negre, I. Safro, S. M. Mniszewski, and Y. Alexeev, "A hybrid approach for solving optimization problems on small quantum computers," *Computer*, vol. 52, no. 6, pp. 18–26, Jun. 2019.
[24] Y. He, B. Venkatesh, and L. Guan, "Optimal scheduling for charging and discharging of electric vehicles," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1095–1105, Sep. 2012.
[25] Z. Zhu, S. Lambotharan, W. H. Chin, and Z. Fan, "A mean field game theoretic approach to electric vehicles charging," *IEEE Access*, vol. 4, pp. 3501–3510, 2016.
[26] W. V. Dam, K. Eldefrawy, N. Genise, and N. Parham, "Quantum optimization heuristics with an application to knapsack problems," in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, Oct. 2021, pp. 160–170.
[27] P. D. de la Grand'Rive and J.-F. Hullo, "Knapsack problem variants of QAOA for battery revenue optimisation," 2019, *arXiv:1908.02210*.
[28] C. Roch, A. Impertro, T. Phan, T. Gabor, S. Feld, and C. Linnhoff-Popien, "Cross entropy hyperparameter optimization for constrained problem Hamiltonians applied to QAOA," in *Proc. Int. Conf. Rebooting Comput. (ICRC)*, Dec. 2020, pp. 50–57.
[29] T. G. Draper, "Addition on a quantum computer," 2000, *arXiv:quant-ph/0008033*.
[30] Qiskit Contributors. (2023). *Qiskit: An Open-Source Framework for Quantum Computing*. [Online]. Available: https://github.com/Qiskit/qiskit-terra
[31] S. C. Endres, C. Sandrock, and W. W. Focke, "A simplicial homology algorithm for Lipschitz optimisation," *J. Global Optim.*, vol. 72, no. 2, pp. 181–217, Oct. 2018.
[32] S. Martello, D. Pisinger, and P. Toth, "Dynamic programming and strong bounds for the 0–1 knapsack problem," *Manage. Sci.*, vol. 45, no. 3, pp. 414–424, Mar. 1999.
[33] F. Leymann and J. Barzen, "The bitter truth about gate-based quantum algorithms in the NISQ era," *Quantum Sci. Technol.*, vol. 5, no. 4, Oct. 2020, Art. no. 044007.
[34] H. Wang, J. Gu, Y. Ding, Z. Li, F. T. Chong, D. Z. Pan, and S. Han, "QuantumNAT: Quantum noise-aware training with noise injection, quantization and normalization," in *Proc. 59th ACM/IEEE Design Autom. Conf.*, Jul. 2022, pp. 1–6.
[35] D. Qin, Y. Chen, and Y. Li, "Error statistics and scalability of quantum error mitigation formulas," *NPJ Quantum Inf.*, vol. 9, no. 1, p. 35, Apr. 2023.
[36] D. Dadkhah, M. Zomorodi, S. E. Hosseini, P. Plawiak, and X. Zhou, "Reordering and partitioning of distributed quantum circuits," *IEEE Access*, vol. 10, pp. 70329–70341, 2022.

KIMLEANG KEA (Student Member, IEEE) received the bachelor's degree in computer science and engineering from the Royal University of Phnom Penh, Phnom Penh, Cambodia, in 2020. He is currently pursuing the M.S. degree with the Department of AI Convergence, Pukyong National University, Busan, South Korea. His research interests include deep learning, embedded systems, and quantum computing.

CHANSREYNICH HUOT received the bachelor's degree in informatic economics from the Royal University of Law and Economics, Cambodia, in 2020. She is currently pursuing the M.S. degree with the Department of AI Convergence, Pukyong National University, Busan, South Korea. Her research interests include deep learning, embedded systems, and quantum computing.

YOUNGSUN HAN (Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Korea University, Seoul, South Korea, in 2003 and 2009, respectively. He was a Senior Engineer with the System LSI, Samsung Electronics, Suwon, South Korea, from 2009 to 2011. He was an Assistant/Associate Professor with the Department of Electronic Engineering, Kyungil University, Gyeongsan-si, South Korea, from 2011 to 2019. He is currently an Associate Professor with the Department of Computer Engineering, Pukyong National University, Busan, South Korea. His research interests include quantum computing, compiler construction, microarchitecture, and high-performance computing.