Received 6 March 2025, accepted 22 March 2025, date of publication 25 March 2025, date of current version 2 April 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3554690

RESEARCH ARTICLE

Evolutionary Algorithm for the Traveling Salesman Problem With Innovative Encoding on Hybrid Quantum-Classical Machines

RAVI SAINI[1], ASHISH MANI[2], M. S. PRASAD[3], SIDDHARTHA BHATTACHARYYA[4], (Senior Member, IEEE), AND JAN PLATOS[4], (Member, IEEE)

[1]Department of Computer Science and Engineering, ASET, Amity University, Noida, Uttar Pradesh 201301, India
[2]Amity Innovation and Design Centre, Amity University, Noida, Uttar Pradesh 201301, India
[3]Amity Institute of Space Science and Technology, Amity University, Noida, Uttar Pradesh 201301, India
[4]Faculty of Electrical Engineering and Computer Science, VŠB-Technical University of Ostrava, 708 00 Ostrava, Czech Republic

Corresponding author: Ravi Saini (ravi.saini1@student.amity.edu)

This work was supported in part by the European Union under the REFRESH-Research Excellence For REgion Sustainability and High-tech Industries Project via the Operational Programme Just Transition under Grant CZ.10.03.01/00/22_003/0000048.

ABSTRACT The Traveling Salesman Problem (TSP) is a widely studied NP-complete optimization challenge with significant theoretical and practical implications. This study proposes a hybrid quantum-classical framework using a Quantum-Inspired Evolutionary Algorithm (QEA) with Sort Gray Binary Encoding to solve the TSP. The proposed method guarantees the generation of valid TSP tours by eliminating invalid solutions. It employs quantum superposition with intrinsic randomness to enhance computational efficiency and scalability. The framework was implemented on cloud-based NISQ platforms, including IBM Quantum and AWS Braket, demonstrating its practicality and effectiveness. Experimental evaluations revealed that the proposed framework successfully solved TSP instances with up to 15 cities, achieving superior performance compared to classical methods and showcasing its ability to scale under NISQ constraints. The results also highlight the potential of hybrid quantum-classical approaches to overcome hardware limitations in current quantum systems. This study establishes a robust hybrid methodology for solving combinatorial optimization problems. It also sets a benchmark for leveraging the capabilities of NISQ-era quantum devices in real-world applications, thereby providing a foundation for future research in hybrid quantum-classical optimization techniques.

INDEX TERMS Evolutionary algorithm, gray encoding, hybrid algorithms, noisy intermediate-scale quantum, quantum computing, traveling salesman problem.

I. INTRODUCTION

The Traveling Salesman Problem (TSP) can be traced back to Euler as early as 1759 [1], where the knight tour problem is discussed to find a solution to ensure that the knight visits each of the 64 squares of a chessboard exactly once. A fully symmetric version of the TSP was considered to have been mentioned in the manual in 1832 with an example of 48 German cities. The problem is simple to explain, yet daunting to solve optimally. Mathematically, TSP has been proven to be NP-complete [2], thereby classifying it as a set of problems that are difficult to solve. For a detailed discussion of TSP evolution, readers can refer to [3]. TSP is central to a wide array of real-world applications, including logistics and supply chain management, where optimizing delivery routes can significantly reduce operational costs, improve resource utilization, and minimize energy consumption [4]. Additionally, TSP-based methodologies are widely used in robotics for motion planning, telecommunications for optimizing network routing, and advanced manufacturing

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

for efficient tool path optimization in CNC systems. These diverse applications highlight TSP's utility across industries.

In its simplest form, TSP aims to find the shortest route for visiting n cities once and returning to the starting point. Alternatively, it can be viewed as a problem in which a salesman visits each city only once, starting and ending in the same city, while covering the minimum possible distance. The distances between cities are known antecedently and are symmetric in the simplest form of the TSP. In this study, we focus on a fully connected TSP problem, in which city is directly connected to every other city, ensuring a complete graph representation of the problem. When the problem is presented on a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, with n vertices in $(\mathbf{V})$ representing cities and edges in $(\mathbf{E})$ indicating paths between cities, the following equations holistically describe the mathematical model.

$$\min \sum_{i,j=1}^{n} c_{i,j} x_{i,j} \tag{1}$$

$$\sum_{j=1}^{n} x_{i,j} = 1, \quad i \in \mathbf{V},\ i \neq j \tag{2}$$

$$\sum_{i=1}^{n} x_{i,j} = 1, \quad j \in \mathbf{V},\ i \neq j \tag{3}$$

$$\sum_{i,j \in S} x_{i,j} \leq |S| - 1, \quad S \subset V,\ 2 \leq |S| \leq n-2 \tag{4}$$

$$x_{i,j} = 0 \text{ or } 1, \quad \forall (i,j) \tag{5}$$

where,

$$c_{i,j} = \text{Traveling cost from city } i \text{ to } j$$

$$x_{i,j} = \begin{cases} 1, \text{ if path from city } i \text{ to } j \text{ is selected} \\ 0, \text{ otherwise} \end{cases}$$

Equation 1 represents the objective function that seeks to minimize the total travel cost in the Traveling Salesman Problem. The degree constraints to ensure that each city was visited only once were implemented using Equations 2 and 3. Finally, Equations 4 and 5 represent the sub-tour elimination and integrality constraints, respectively.

Given that TSP is an NP-complete problem and considering the implications of the No Free Lunch Theorems for optimization [5], extensive research has focused on solving the TSP using classical computers. These efforts span exact approaches [6], approximation techniques [7], and heuristic methods [8]. In heuristic approaches, many bio-inspired algorithms have been reported to efficiently solve TSP [4], [9]. The best known time-space complexity for solving specific categories of TSP is $\mathcal{O}(2^{\mathrm{n}}\mathrm{n}^{\mathcal{O}(1)})$ [10]. However, these approaches often depend on a probabilistic element, which is a part of the heuristic strategy, and do not guarantee an optimal solution [11], [12]. As problem instances grow in size and complexity, metaheuristics may struggle to scale efficiently, resulting in suboptimal solutions or excessively long computation times [13]. Therefore, it is necessary to explore alternative computing paradigms for solving NP-complete problems. Solving TSP efficiently is crucial not only for its direct applications but also for advancing approaches to other NP-complete problems, such as job-shop scheduling, resource allocation, and circuit design [14]. Each solution to TSP contributes to broader understanding and algorithmic developments for tackling computationally intractable problems.

The most revolutionary theory to explain subatomic physics was introduced by the ideas evolved by Max Planck, Erwin Schrödinger, Werner Heisenberg, and Paul Dirac [15]. Building upon these foundational ideas, Richard Feynman [16] and David Deutsch [17] introduced the idea of using quantum effects for computations. Before functional quantum devices were realized, theories such as Shor's algorithm for prime factorization [18] and Grover's algorithm for database searches [19] showed how quantum systems could outperform classical computers in certain tasks.

Despite this theoretical promise, real-world applications of quantum computing remain constrained owing to limited access to quantum hardware. The currently available quantum bits or *qubits* are prone to instability, making them vulnerable to interference, decoherence, and noise. As a result, present-day quantum systems fall under the category of Noisy Intermediate-Scale Quantum (NISQ) devices [20]. The term "intermediate scale" means having fewer than 100 noisy *qubits*. Their flaws make it difficult to control them accurately.

In this study, we examined the methods currently deployed to use NISQ machines to solve TSP. We present a new method based on our literature review and identified research gap. This method uses a population-based Evolutionary Algorithm (EA) with specially designed binary Gray encoding to be implemented on an NISQ machine. By integrating classical evolutionary strategies with quantum features like superposition and randomness, these methods can address the scalability challenges of current quantum systems [21]. Section II of the paper provides an overview of the current literature related to this problem. Section III highlights the research gaps and key contributions of this study. Section IV explains the proposed implementation of EA for the TSP on NISQ devices. Section V provides a comprehensive description of the experimental framework and the parameters used to solve the TSP using the NISQ devices. Section VI presents the experimental results, and Section VII provides a detailed analysis of these results. Section VIII concludes the paper by summarizing the key contributions and proposing directions for future research.

II. LITERATURE SURVEY

The TSP has been extensively studied using classical computing methods. Exact solvers such as Concorde have demonstrated state-of-the-art performance by leveraging combinatorial optimization techniques, including branch-and-bound strategies [22]. Additionally, advanced metaheuristics, such as Ant Colony Optimization (ACO) and Genetic Algorithms (GA), have been widely applied to efficiently approximate solutions for large-scale TSP instances [23]. Although classical solvers currently dominate TSP optimization, our study focuses on quantum and quantum-inspired approaches, particularly in the context of leveraging NISQ devices.

Hybrid quantum-classical methods have the potential to complement classical solvers by incorporating quantum properties such as superposition and entanglement, which can enhance search space exploration and provide novel optimization strategies. Although quantum hardware is still in its early stages, steady advancements in qubit stability, error mitigation, and quantum circuit design suggest that hybrid and quantum approaches may become viable for solving large-scale combinatorial problems [24]. Our literature review further explores how quantum-inspired evolutionary algorithms can leverage emerging quantum hardware to address TSP optimization challenges.

Early developments in theoretical quantum algorithms have proven the feasibility of large-scale improvements in computing capabilities by using this new paradigm. However, full-scale implementation of these algorithms is not possible owing to the low availability of quantum computing devices. This has led to innovative research methods that utilize quantum and classical paradigms in a hybrid mode to maximize the benefits of quantum techniques. Interested readers can refer to [25] and [26] for recent literature on the advances in hybrid quantum metaheuristics.

In this literature survey, we examined only quantum algorithms that have been applied to solve the TSP. Typically,, quantum algorithms for optimization are categorized into two paradigms: gate-based quantum machines and quantum annealing-based algorithms [27]. However, with the progressive blurring of boundaries for Quantum-inspired Evolutionary Algorithms (QEA) to be used on quantum machines in hybrid models [28], we propose three distinct categories for solving the TSP on quantum machines. Based on the type of quantum device used, we classified the current research into the categories mentioned below and present a brief description of each method for solving TSP.[1]

- Quantum-inspired Evolutionary Algorithms (QEA)
- Gate-Based Quantum Machines
- Quantum Annealing (QA) based Machines
- Digital Annealers (DA)

A. QUANTUM-INSPIRED EVOLUTIONARY ALGORITHMS (QEA)

Inspired by the quantum computing paradigm, these algorithms mimic quantum techniques for solving optimization problems on classical computers. They are designed to run on classical computers, and do not use quantum computers. QEA [29] represents a subset of population-based methods that utilize multimodal probability distribution estimation techniques [30], and has demonstrated significant success in addressing diverse search and optimization challenges. QEA employs a representation based on simulated quantum bits (*q-bits*) along with unitary rotation gates and measurement operators, all of which are simulated using classical computational devices [31]. NISQ devices, on the other hand, have achieved selective quantum supremacy by efficiently sampling complex probability distributions that are computationally hard for classical systems [32]. Such devices could enhance the QEA by facilitating the implementation of genotype *qubit* representations, rotation operations, and measurement tasks with greater efficiency than their classical counterparts. It is conjectured that NISQ systems can inherently improve the accuracy of multimodal probability distribution estimations in a QEA, potentially enabling accelerated search and optimization processes compared to traditional digital implementations.

Implementation of population-based heuristics on quantum computers has recently been reported using the Hybrid Quantum Genetic Algorithm (HQGA) [28], [33], Quantum-Inspired Estimation of Distribution Algorithm (QIEDA) [34], and Tabu Search Algorithm with implementation for TSP [35]. A study of the applications of the above techniques to solve the TSP revealed the following.

- HQGA [28], [33] does not have a direct implementation for solving TSP.
- QIEDA [34] innovatively implemented an array representation of the TSP and a quantum simulator was used to sample valid routes using W-state quantum circuits. These algorithms have been reported to run directly on a real quantum computer using circuit-model programming, without modifications. However, results were obtained for TSP instances in up to 20 cities by simulating real IBM quantum computers.
- Similarly, owing to the restrictions of quantum machines, the Tabu Search Algorithm implemented for TSP [35] uses the QBSolv tool offered by the D-Wave System in its local variant for 14-23 cities.

Thus, to the best of our knowledge, no population-based systems have reported the results of a hybrid implementation in actual quantum machines to solve the TSP. However, there are reports on solving TSP instances for up to 20 cities on both gate-based machines and D-Wave simulators using the QEA.

The proposed method belongs to the QEA class and extends it by introducing hybrid quantum-classical strategies. The framework provides a bridge for extending the QEA class by utilizing gate-based circuits and employing superposition within the QEA on NISQ devices to solve the TSP.

B. GATE-BASED QUANTUM MACHINES

There are two common approaches for solving the TSP on gate-based quantum machines: Quantum Approximate Optimization Algorithm (QAOA) [36] and Quantum Phase

[1] A detailed recent survey of gate-based quantum machines and QA algorithms to solve routing problems and TSP is provided in [27].

Estimation Algorithm [37]. Both techniques encode the problem in a suitable method to be fed to a quantum computer (unitary/Hamiltonian operator), and then use separate procedures for manipulation with quantum gates to estimate the eigenstate to find the optimal solution. They have detailed tutorials on QISKIT, the Open-Source Quantum Development website [38], and in [39]. Recent results using QAOA have been reported for TSP problems in a maximum of four cities. The limitation in the number of cities is primarily owing to the non-availability of larger numbers of stable *qubits* [40]. The quantum phase estimation algorithm claims the theoretical gain of a quadratic speedup over the brute force search [41]; however, the practical implementation of all the algorithms in this category has been found to be restricted to trivial problems of TSP with up to 4-5 cities [42], [43] [44].

C. QUANTUM ANNEALING (QA) BASED MACHINES

QA machines accept TSP as a quantum Hamiltonian, and the minimum quantum energy of this Hamiltonian is estimated using an adiabatic annealing process. The general technique used in QA is adiabatic quantum optimization [45] with the following four steps:

- Convert TSP to an Ising model.
- Feed the Ising model to the Quantum Machine.
- Undertake QA.
- Interpret the results.

Quantum annealing outperforms gate-based machines in complex optimization problems. Furthermore, D-Wave launched Advantage Quantum Computers with over 5,000 *qubits* on its cloud in 2020 [46], whereas IBM offered 127 *qubit* gate-based devices on the cloud in 2021 [47]. However, owing to various implementation restrictions, D-Wave machines with more than 5000 *qubits* on quantum annealing devices translate approximately 73 logical *qubits* [48]. QA solutions for TSP are restricted to problems below the size of eight cities for practical implementation [48], [49], primarily because of the limited availability of stable and noise free *qubits*.

D. DIGITAL ANNEALERS (DA)

DAs have been developed by employing unique computing technologies inspired by quantum phenomena [50]. Results from these classical devices, such as Fujitsu's CMOS Digital Annealer, have demonstrated success in solving TSP instances with up to 100 vertices [51]. However, these approaches are fundamentally distinct from QEA, gate-based quantum machines, and quantum annealers. DA uses highly connected classical bits. Together, they run a stochastic search to minimize energy functions. However, they do not use *qubits* as computing elements [52]. Leading review papers on quantum systems and quantum-inspired evolutionary algorithms, typically do not classify digital annealers within these categories [27], [53], [54].

III. RESEARCH GAPS AND KEY CONTRIBUTIONS

Based on the literature survey, we do observe the following.

- There has been limited progress in evolving Quantum-Inspired Evolutionary Algorithms (QEA) for implementation on quantum simulators or gate-based quantum machines to solve the Traveling Salesman Problem (TSP). No study has reported results for the actual implementation of QEA on gate-based quantum machines.
- Existing QEA implementations are restricted to classical machines or simulators, meaning that key quantum phenomena such as superposition and entanglement have not been utilized in current approaches. This limits the potential advantages of quantum computing for solving optimization problems like the TSP.
- Traditional encoding schemes, such as binary or matrix-based representations, often produce invalid TSP solutions during evolutionary operations like crossover and mutation. These inefficiencies increase computational overhead and negatively impact the scalability of algorithms for larger problem instances.

To mitigate the above mentioned research gaps, we attempted to implement QEA on a gate-based model with a population-based technique using NISQ machines. The proposed novel approach ensures that all TSP solutions generated by the proposed EA in the hybrid mode using classical and quantum computers are valid and eliminates all invalid tours. This study provides a framework for solving TSP problems in up to 15 cities on present-day quantum machines in hybrid mode using EA. The specific contributions of this study are as follows.

- An innovatively designed Sort Gray Binary Encoding for TSP (SGBET) eliminates errors in TSP tours. This encoding ensures that all solutions generated during evolutionary operations are valid, overcoming the challenges associated with traditional encoding schemes. The method and its validity are demonstrated in Section IV-A with proofs and examples.
- A hybrid framework was presented for solving TSP instances of up to 15 cities, leveraging cloud-based access to IBM-Q [55] and AWS Braket [56]. The framework combines the strengths of classical and quantum paradigms to overcome the limitations of the NISQ hardware.
- The use of quantum superposition was demonstrated via controlled gate-based circuits on NISQ machines. This feature leverages quantum randomness to improve the solution diversity and convergence efficiency.

IV. PROPOSED IMPLEMENTATION OF EA FOR TSP ON NISQ

We propose a hybrid model to solve combinatorial optimization problems (COP) in gate-based quantum machines using population-based techniques. The motivation for the proposed model (as shown in Fig. 1) was drawn from the Quantum-inspired EA [29] and the implementation of EAs on IBM-Q [28]. A key advantage of quantum computing is its ability to generate true randomness by using quantum mechanics and measurements. In the proposed methodology, this randomness is leveraged to formulate the routes in the TSP. Unlike classical systems that rely on pseudo random number generators, quantum systems exhibit intrinsic randomness, which provides more authentic randomization.

In our approach, route formation is automated using quantum evolutionary algorithms that systematically explore potential routes for the TSP without requiring manual intervention at each decision point. The tasks of reading the TSP problem, initialization, interpretation, evaluation of solutions, and population management are performed using a classical machine. A quantum machine is used for key EA procedures such as crossover, mutation, and generation of offspring. The innovation in the implementation of TSP in this model is the design of a special binary Gray encoding, as explained in the following subsection.

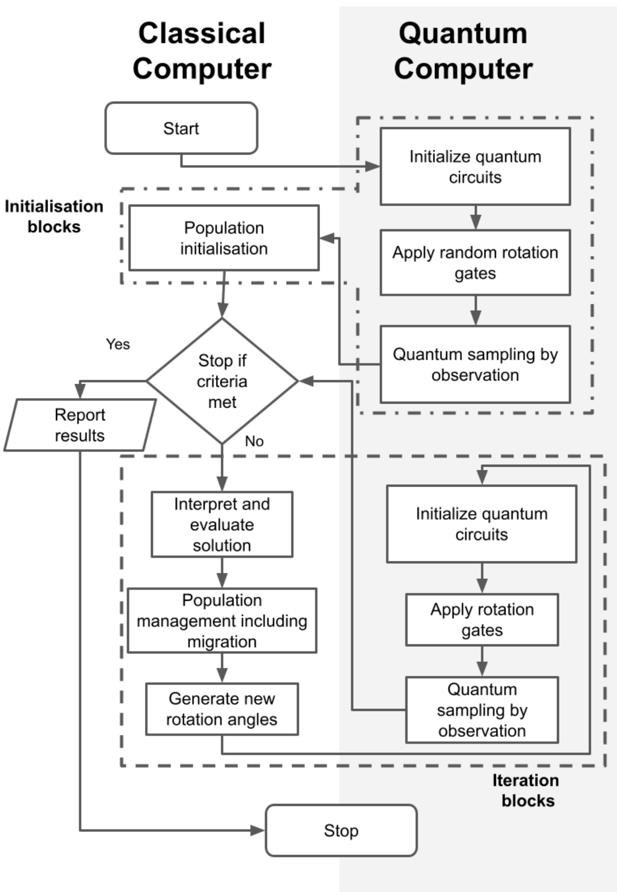


FIGURE 1. Block diagram.

A. GRAY BINARY ENCODING

Gray binary encoding is a system of representing binary numbers where two consecutive values differ in only one bit [57]. This helps to minimize changes between solutions during optimization. It reduces disruptions caused by mutations in GA and makes the search process more stable. Using Gray binary encoding in SGBET ensures smoother transitions between solutions. This improves the performance of solving problems such as TSP [58].

B. SORT GRAY BINARY ENCODING FOR TSP (SGBET)

In the Genetic Algorithm (GA) there are generally two representations of a TSP solution: binary representation and matrix representation. A binary representation was considered in the proposed methodology to minimize the number of *qubits* required. In this representation, each solution is called a chromosome, that contains variables called genes. The method used to interpret these genes and chromosomes is defined by SGBET.

As an example of a binary representation, a six-city TSP solution can be represented by a chromosome with genes 0-2-3-4-5-1 which is encoded as follows.

0	2	3	4	5	1	← *decimal*
000	010	011	100	101	001	← *binary*

It is trivial to interpret that the bits required in this case are $n\lceil \log_2 n \rceil$ and each ordered set of $\lceil \log_2 n \rceil$ bits represents the next city to be visited. This encoding is known as Lidd's encoding [59]. This representation is not suitable for the TSP. It creates many invalid tours for high-order TSP problems when GA operators (such as combination and mutation) are used [1]. To understand these issues, two simple mutations in the chromosome are presented below.

- A single-bit mutation in the second gene (bit 6) results in a binary sequence (000-011-011-100-101-001), which is interpreted as an invalid tour because City 3 is visited twice and City 2 is not visited at all, as depicted below.

000	01**1**	011	100	101	001
0	**3**	3	4	5	1

- Another problem can be demonstrated by considering the mutation in the third gene (bit 7), which leads to the inclusion of the seventh city (which does not exist as per our problem) in the tour. This situation can be described as follows.

000	010	**1**11	100	101	001
0	3	**7**	4	5	1

To overcome the aforementioned problems of invalid solutions, we propose SGBET which draws inspiration from the proposal of using special sorted encoding to overcome the aforementioned shortcomings [60]. The Gray code was used for the initial interpretation of the genes [61] to minimize the Hamming distance between similar TSP solutions. The process of decoding chromosomes using SGBET involves the following steps.

1) **Score Sequence**: Decode individual genes in the chromosome to obtain a score using grayscale as a decimal value and call it the score sequence.

2) **Repaired Score Sequence**: In case of repetition in the score sequence, the score of the repetition gene is increased until no repetition exists.
3) **Sorted Sequence**: Sort the repaired score sequence in ascending order.
4) **City Tour**: For every element in the sorted sequence, the index of the same element in the Repaired Score Sequence was determined. The score index was considered as the city to be visited at each step.

Algorithm 1 SGBET: Sort Gray Binary Encoding for TSP

1: **procedure** SGBET (chromosome)
2: $n \leftarrow$ Number of cities; $gene\_len \leftarrow \lceil \log_2 n \rceil$
3: **for** i from 0 to $n-1$ **do**
4: $score\_seq_i \leftarrow$ gray_decode (chromosome[$i \times gene\_len$:$(i+1) \times gene\_len$])
5: **end for**
6: **create** repaired_score_seq:
7: each repetition in *score_seq* is increased by 1 until all repetitions are removed
8: **create** sorted_seq:
9: sort repaired_score_seq in ascending order
10: **for** i from 0 to $n-1$ **do**
11: $city\_tour_i \leftarrow$ index (repaired_score_seq [sorted_seq])
12: **end for**
13: **return** (city_tour, (length (city_tour)))
14: **end procedure**

TABLE 1. Example tour 1.

Example Tour 1: SGBET Decoding						
Chromosome	**010**	110	111	000	**010**	011
Score Sequence	**3**	4	5	0	**3**	2
Repaired Score Sequence	**3**	4	5	0	**6**	2
Sorted Score	0	2	3	4	5	**6**
City Tour	3	5	0	1	2	4

TABLE 2. Example tour 2.

Example Tour 2: SGBET Decoding						
Chromosome	**010**	**110**	111	**110**	**010**	011
Score Sequence	**3**	**4**	5	**4**	**3**	2
Repaired Score Sequence	**3**	**4**	5	**6**	**7**	2
Sorted Score	2	3	4	5	**6**	**7**
City Tour	5	0	1	2	3	4

The process is outlined in **Algorithm 1**. This algorithm resolves the key challenge in interpreting a random binary string as a valid TSP tour. The algorithm presents a step-by-step procedure for SGBET, and its implementation is presented in Tables 1 and 2. The tables show examples of interpreting two tours, which would have been considered invalid with a simple binary representation and are shown to represent a valid chromosome with SGBET. For better insight into the decoding operation, two additional examples are included in the **Appendix** for TSPs with 12 and 15 cities. Based on the distance matrix provided as an input for the TSP, Step 13 of the algorithm returns the tour length, which is used in Step 5 of **Algorithm 2**.

The following lemma holds for the decoding procedure.

Theorem 1: Every input chromosome (bitstring) of length $n\lceil \log_2 n \rceil$ provided as an input to the SGBET algorithm produces a valid TSP tour for n cities.

Proof: We define a valid TSP tour for n cities as a permutation of 0 to $(n-1)$ that satisfies the following conditions.

(a) Each city was visited exactly once.
(b) Only cities between 0 and $n-1$ were included in the tour.
(c) Every city from 0 to $n-1$ was included in the tour.

For the sake of contradiction, assume that there is a TSP tour produced by the SGBET algorithm that is invalid and violates at least one of the aforementioned conditions.

Algorithm 1 ensures the following:

1) In Steps 6 and 7, all repetitions of the tour are removed, guaranteeing that each city is visited exactly once, thus satisfying condition (a).
2) Steps 10 and 11 ensure that only city numbers from the `repaired_score_seq` index are included in the tour, which means that only cities from 0 to $n-1$ are included, which meets Condition (b).

Because (a) and (b) are satisfied, it follows that every city from 0 to $n-1$ is included in the tour, thereby satisfying condition (c).

Therefore, the assumption of an invalid TSP tour produced by the SGBET algorithm leads to contradictions.

Thus, every input chromosome of length $n\lceil \log_2 n \rceil$ given to the SGBET algorithm produces a valid TSP tour for n cities.

C. TIME COMPLEXITY OF SGBET

Here, we briefly describe the time-complexity analysis of the proposed SGBET, as explained in **Algorithm 1**. For this analysis, an asymptotic upper bound is used with $\mathcal{O}$-notation. The stepwise $\mathcal{O}$-notation analysis for each step in **Algorithm 1** is tabulated in Table 3 and is briefly explained below.

- **Step 3-5**. For a TSP of n cities, there are n Gray codes, each with length $\lceil \log_2 n \rceil$. Because we are calculating the upper bound, we assume that the Gray code length is $\log_2 n$. A Gray code of length m can be converted into a binary code in $\mathcal{O}(log\ m)$ time [62]. Thus, the total upper time bound for these steps is $\mathcal{O}(n\ log\ log_2 n)$.
- **Step 6-7**. These steps remove duplicates from the unsorted list. Numerous methods exist for removing duplicates, including hashing-based algorithms with a time complexity of $\mathcal{O}(n)$ for a list of n items [63].
- **Step 8-9**. These steps sort a given list of n items, without duplicates. One of the methods with optimal time complexity is merge sort with $\mathcal{O}(n\ log\ n)$.

- **Step 10-12**. For these steps, we considered two lists of n items each. The action performed in these steps is analogous to identifying the position of each element in the second list within the first list. Each of these search operations has worst-case complexity of $\mathcal{O}(n)$. Therefore, all n searches have a complexity of $\mathcal{O}(n^2)$.

TABLE 3. Stepwise $\mathcal{O}$-notation analysis of Algorithm 1.

Steps	$\mathcal{O}$-notation bound	Remarks
1-2	Constant	—
3-5	$\mathcal{O}(n\ log\ log_2 n)$	—
6-7	$\mathcal{O}(n)$	—
8-9	$\mathcal{O}(nlog\ n)$	—
10-12	$\mathcal{O}(n^2)$	Dominating Step for time complexity
13-14	Constant	—

Based on the dominant time complexity in the above steps, the overall worst-case time complexity of SGBET in **Algorithm 1** is $\mathcal{O}(n^2)$. It is noteworthy that the dominant time-complexity factor of this algorithm is related to the search for a key in an unsorted list, and quantum computing has the potential to provide quadratic improvement [19] in this function after the availability of quantum machines with sufficiently stable *qubits*.

D. EA ALGORITHM FOR HYBRID IMPLEMENTATION ON NISQ

This section outlines the key elements of implementing the proposed EA in gate-based quantum machines. It is based on the validity of SGBET and uses a hybrid framework, as illustrated in Fig. 1. As stochastic optimization techniques, EAs maintain a diverse population of candidate solutions. These solutions, referred to as individuals, undergo iterative improvement through the application of specialized operators to converge toward an optimal result. In QIEA, individuals in a population are represented as Q-bits, which inherently exhibit probabilistic characteristics similar to those of the *qubits* employed in quantum machines. Unlike the classical binary bit, which is limited to representing information in one of the two definite states, that is, 0 or 1, a *qubit* can encode information as a superposition of both states simultaneously. Mathematically, this property is expressed as.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{6}$$

$$\sqrt{|\alpha|^2 + |\beta|^2} = 1 \tag{7}$$

where,

$|\alpha|^2$: Probability of $|\psi\rangle$ collapsing to state 0

$|\beta|^2$: Probability of $|\psi\rangle$ collapsing to state 1

$\alpha, \beta \in \mathbb{C}$

The probabilistic framework inherent to QIEA offers an enhanced representational capacity. This fosters a greater solution diversity and reduces the likelihood of premature convergence.

In this study, we employed the QIEA framework for the TSP, as introduced in [29], and adapted it to align it with real-world quantum hardware. The modified algorithm was designed to run for *MAX_GEN*, generations and is presented in **Algorithm 2**. The steps outlined in this algorithm utilize a hybrid quantum-classical framework to achieve the objective of minimizing the TSP tour length, as mathematically defined in Equation 1. In this implementation, a quantum population, denoted as $\boldsymbol{Q}(t)$, is maintained, where each quantum entity is encoded with $n\lceil\log_2 n\rceil$ *qubits*, enabling the superposition of multiple candidate solutions. By performing measurements, this quantum population generates a corresponding classical population $\boldsymbol{P}(t)$, where each classical entity consists of $n\lceil\log_2 n\rceil$ bits. The classical population is evaluated using a fitness function, and the feedback from its performance guides the update process of the quantum population. Additionally, the algorithm retains a repository of top-performing solutions for each individual in $\boldsymbol{B}(t)$, and the global best solution is tracked as $\boldsymbol{b}$.

Algorithm 2 QEA on NISQ for TSP

procedure QEA TSP
    $t \leftarrow 0$
    $Q(t) \leftarrow$ Initialize ()
    $P(t) \leftarrow$ Observe $(Q(t))$
    $\{P(t)$, fitness $(P(t))\} \leftarrow$ SGBET $(P(t))$
    $B(t) \leftarrow \arg\min_{p \in P(t)}$ Fitness(p)
    while $t <$ MAX_GEN **do**
        $t \leftarrow t + 1$
        $P(t) \leftarrow$ Observe $(Q(t-1))$
        $P(t) \leftarrow$ SGBET $(P(t))$
        Update $(Q(t))$
        $B(t) \leftarrow$ Best individual in $P(t)$ and $B(t-1)$
        $b \leftarrow$ Best individual in $B(t)$
        if migration-period **then**
            migrate b_{tj} or b to $B(t)$ locally or globally respectively
        end if
    end while
end procedure

For an in-depth analysis of **Algorithm 2**, readers are encouraged to consult [29]. A concise overview of the key routines employed in the algorithm is provided below:

Initialize (): The quantum population is initialized by assigning a random probability to each *qubit* of every quantum individual to collapse into either State 0 or State 1 upon measurement. This was accomplished by applying a randomly selected rotation angle for each *qubit* ranging between 0 and 2π. This is equivalent to the random selection of the values of α and β for a *qubit*, as defined in Equations 6 and 7.

Observe ($\boldsymbol{Q}$): In this procedure, the *qubit* of each individual q within the quantum population (**Q**) is measured to produce the corresponding classical bit for each individual

p in the classical population (**P**). Detailed steps of this process are presented in **Algorithm 3**. The *QuantumCirc* procedure (described in the next section) implements the rotation and observation of *qubits* by using gate-based quantum circuits.

Algorithm 3 Observe (Q)

```
procedure Observe (Q)
    for all qubits in individual q ∈ Q do
        C-bits of p in P ← QuantumCirc (qubits)    ▷ on NISQ
    end for
end procedure
procedure QuantumCirc (qubits, N = 5)
    for each set of N qubits do
        C-bits ← collapsed qubits value after rotation on NISQ
    end for
    return C-bits
end procedure
```

SGBET (*P*): The SGBET routine is executed for each solution p in the classical population (**P**) to guarantee the correctness of the TSP tour. The routine is guaranteed to produce a valid tour for all binary strings of length $\lceil \log_2 n \rceil$, where n is the number of cities.

Update (*Q*): The routine modifies the *qubits* of each individual q in the quantum population (**Q**) by considering the fitness of the corresponding classical individual p and that of the best solution b. Each *qubit* operates in a Bloch sphere with a rotation angle θ in the range $[-\pi/2, +\pi/2]$. The value of the rotation angle $\Delta\theta_i$ to be applied to each *qubit* q_i was selected as follows:

- Value of corresponding classical individual p_i versus fitness of the best solution b.
- Current quadrant of *qubit* q_i in the Bloch sphere. In this study, the Bloch sphere is divided into four quadrants. As mentioned below, these four quadrants are where the *qubit* can reside in the *x*-*z* plane of a Bloch sphere.

$$\begin{array}{ccc} |1\rangle & \text{to} & (1/\sqrt{2})(|0\rangle + |1\rangle) \\ (1/\sqrt{2})(|0\rangle + |1\rangle) & \text{to} & |0\rangle \\ |0\rangle & \text{to} & -(1/\sqrt{2})(|0\rangle + |1\rangle) \\ -(1/\sqrt{2})(|0\rangle + |1\rangle) & \text{to} & |1\rangle \end{array}$$

An illustration of the movement of a *qubit* in Hilbert space is presented using a Bloch sphere, as shown in Fig. 2.[2] Each *qubit* moves in the *x*-*z* plane (depicted with a dotted circle). Controlled rotations of *qubits* with parameterized quantum circuits are used to manipulate the probability of wave functions collapsing to an optimal solution.

In each quadrant, the *qubit* can be rotated clockwise or anticlockwise. This rotation of the *qubit* is required to control the probability of the *qubit* collapsing to $|0\rangle$ or $|1\rangle$ when the *qubit* is observed. With four quadrants and two rotation directions in each coordinate, eight angles can be applied to rotate any *qubit* during each iteration. In the current proposal, we selected the angle of rotation $\Delta\theta$ using the lookup in Table 4. The value of $\Delta\theta$ can be positive or negative based on clockwise or anticlockwise rotation. The details of the optimization for values of $\Delta\theta_1$ to $\Delta\theta_8$ are discussed in the next section under the optimization of parameters. The method used to select the value of $\Delta\theta$ for each *qubit* in Table 4 is explained below.

[2]Credit: The figure was generated using "QuTiP: Quantum Toolbox in Python".

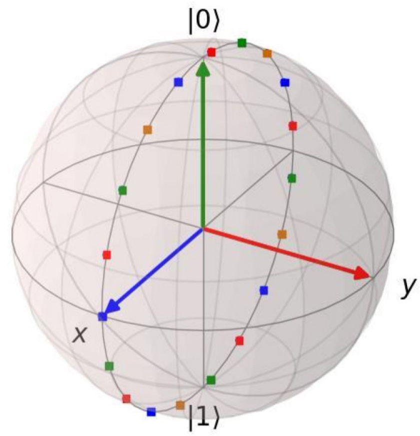


FIGURE 2. Movement of *qubit* during successive generation of EA in Bloch Sphere.

TABLE 4. Basis of selection of $\Delta\theta_i$.

p_i	b_i	$f(B)$ better than $f(P)$	$\Delta\theta_i$
0	0	True	$\Delta\theta_1$
0	0	False	$\Delta\theta_2$
0	1	True	$\Delta\theta_3$
0	1	False	$\Delta\theta_4$
1	0	True	$\Delta\theta_5$
1	0	False	$\Delta\theta_6$
1	1	True	$\Delta\theta_7$
1	1	False	$\Delta\theta_8$

- **Step 1**. Rows that represent the p_i value (0 or 1) corresponding to the value at which *qubit* q_i collapses in the current state are selected.
- **Step 2**. Within the rows selected in Step 1, those representing the value (0 or 1) of the current best solution for the corresponding bit b_i in Table 4 were selected.
- **Step 3**. Finally, within the rows selected in Steps 1 and 2, a row is selected based on whether the fitness of the current solution is better than that of the local best solution (true or false).

$\Delta\theta$ is set to $\Delta\theta_i$ as specified in the row selected in Step 3, and the rotation gate is applied, as shown in Step 4 of **Algorithm 4**.

For more detailed insight into similar update procedures in QEA, readers are advised to consult [29], [31], [33].

Algorithm 4 Update (Q, P, b)

procedure Update (Q, P, b)
  for each $(q, p) \in (Q, P)$ **do**
    determine $\Delta\theta$ with the lookup Table 4
    update $q \leftarrow |R_y(\Delta\theta)\rangle\langle q|$ ▷ Rotation
  end for
end procedure

Migration: The migration process introduces variability in the probability distributions of the quantum individuals. Local and global migration were applied to the population in the proposed design. The migration process is briefly explained below.[3]

- The EA population was divided into multiple groups for independent evolution. The size of the group is a design parameter, and the details regarding the selection of optimal values for group size are discussed in the next section under optimization of parameters. The number of groups was based on the size of each group and total number of agents employed.
- When the migration condition is satisfied, the optimal solution in b is copied locally or globally. In the proposed implementation, the conditions for local and global migration are based on a fixed number of EA generations. The parameters for initiating local and global migration were also optimized, as elucidated in the next section under the optimization of parameters.

E. EVOLUTION OPERATOR

For QIEA implemented on classical computers, evolution operators akin to mutation and crossover in GA are simulated using Q-gates, which modify qubit probability distributions. This enables a probabilistic and adaptive search mechanism that differs from classical heuristics. The simulation of quantum properties, including superposition and controlled qubit rotations on the Bloch sphere, enhances search space exploration and solution diversity.

In the proposed hybrid methodology, actual *qubits* on NISQ devices are placed in a superposition state using controlled rotation gates. This mechanism replaces classical mutation and crossover, dynamically inducing solution diversity through quantum state evolution rather than predefined heuristics. One of the key advantages of this approach is the ability of quantum superposition to simultaneously maintain diverse solution states, thereby broadening search space exploration [64], [65]. Unlike classical methods, which often suffer from premature convergence, quantum-inspired techniques leverage intrinsic randomness and quantum parallelism to explore multiple potential solutions simultaneously. This enhances both adaptability and optimization efficiency, making them well-suited for complex combinatorial problems.

[3]Consult [29] for detailed understanding of migration in QEA.

V. EXPERIMENTAL FRAMEWORK AND PARAMETER OPTIMIZATION

A. EA IMPLEMENTATION ON NISQ

As mentioned previously, a proposal for the implementation of EA in gate-based quantum machines using the hybrid mode is presented in Fig. 1. The steps of the proposed method were described in the previous section. This section presents a consolidated methodology for the implementation of proposed cloud-based access. As mentioned in *Theorem 1* and **Algorithm 1**, a chromosome (bitstring) of length $n\lceil\log_2 n\rceil$ is required to produce a valid TSP tour for n cities. Therefore, to implement a TSP with 15 cities, $60 = (15 \times \lceil 3.91 \rceil)$ *qubits* are required. However, owing to the restriction of the available *qubits* in present-day quantum devices, the implementation of **Observe** (Q) in **Algorithm 3** is executed using N *qubits* at a time, where N represents the number of *qubits* available on the NISQ device being used. Multiple quantum circuits were constructed, each containing N *qubits* configured with rotation gates and observed using a measurement operator.

B. QUANTUM CIRCUIT

Quantum computers are typically accessed by using a quantum circuit model that serves as an abstraction layer to mask the underlying physical architecture of a machine. Platforms such as IBM Quantum and AWS Braket enable users to interact with their quantum systems via cloud-based Application Programming Interfaces (APIs). The Python-based Qiskit framework [38] facilitates the development of quantum algorithms using circuits, and allows execution on quantum hardware.

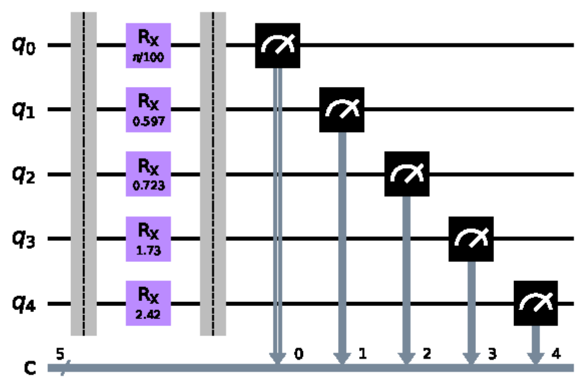


FIGURE 3. Sample circuit to observe 5 *qubits*.

The *QuantumCirc* (*qubits*, N) procedure in **Algorithm 3** involves the creation of multiple quantum circuits, each comprising N *qubits*. These circuits were designed with rotation gates determined by the state of each *qubit* and measurement gates, to record the resulting values as classical bits. An example circuit created using the QISKIT framework to measure the five *qubits* is shown in Fig. 3. The angles

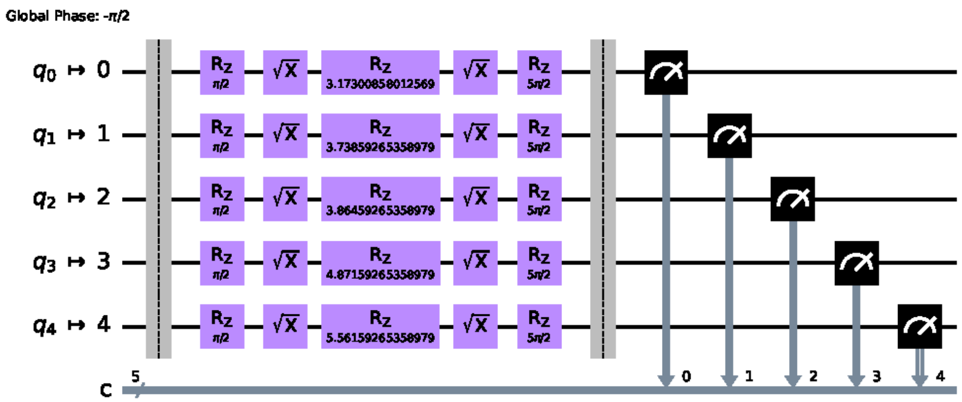


FIGURE 4. Transpiled circuit to observe 5 *qubits* on "ibmq_bogota".

associated with the *qubits* in this figure are $q_{00} = 0.01\pi$, $q_{01} = 0.19\pi$, $q_{02} = 0.23\pi$, $q_{03} = 0.55\pi$, and $q_{04} = 0.77\pi$. Before the circuit is executed, it must be transpiled to satisfy the constraints of the target quantum device. The transpiled version of the circuit optimized for the `ibmq_bogota` device is shown in Fig. 4. The quantum depth of the transpiled circuits was measured to be 6, which is considered a shallow quantum circuit minimizing the impact of noise. Furthermore, for EAs, noise-induced randomness can enhance population diversity.[4]

To estimate execution time, a job consisting of 75 circuits was submitted to the QPU over the Internet, yielding an average execution time of 5.3s per batch. This translates to approximately 70.67 ms per circuit, providing a practical measure of real-time execution performance.

C. METHODOLOGY

The quantum routine outlined in **Algorithm 3** involves generating multiple circuits for execution on 5-*qubit* NISQ devices. `IBMQJobManager`, a function of Qiskit, was used to combine all the circuits for each generation. In the TSP encoding described above, solving a problem with n cities requires $n\lceil\log_2 n\rceil$ qubits, where n represents the number of cities. However, by applying the proposed methodology, the TSP with n cities can be solved using a NISQ device with N available qubits, where $N < n\lceil\log_2 n\rceil$. This approach ensures the efficient use of limited *qubits* in currently available quantum machines.

D. FRAMEWORK

The framework was designed to execute the EA concurrently on classical and quantum computing platforms for the same set of problems. The implementation of the EA discussed in Section II on a classical computing device is straightforward.

[4]While quantum noise is generally a limitation, in EAs, noise-induced randomness can enhance population diversity by introducing beneficial mutations. Thus, quantum error rates were not explicitly analyzed. However, future work could explore impact of quantum errors and mitigation techniques in detail.

However, executing the same EA on NISQ machines using Qiskit requires the use of specific techniques, which are outlined below:

- **Circuit Bunching:** NISQ machines operate in the fair-share mode, where submitted circuits are placed in a queue and executed sequentially in a first-in, first-out (FIFO) manner. To minimize the waiting time for circuit execution, the Qiskit's `IBMQJobManager` functionality was employed to group the circuits into batches. This approach enables the submission of a large set of circuits in a single job, thereby facilitating efficient simultaneous observation of hundreds of *qubits*.
- **NISQ Machine Selection:** The workload on each NISQ machine fluctuates depending on the volume of the submitted jobs. To optimize resource utilization, two approaches are employed to select a machine based on access availability.
  - **Fairshare Access:** This represents the default access provided by the IBMQ. Under this arrangement, the `least_busy` function is used to identify the machine that meets the required number of *qubits* and has the lowest load at any given moment.
  - **Dedicated Access:** IBMQ and AWS Braket were used to obtain preferential/dedicated access to quantum machines/simulators. During the reservation period, the designated machine was used for all submitted jobs.

E. PARAMETER OPTIMIZATION

The selection of optimal parameters is critical for efficient EA operation. The Taguchi method is widely used to select correct parameter values [66]. To select the parameters, the techniques explained in [67] were used with the Taguchi Orthogonal Arrays [68], [69]. The experiments for the selection of the values for the parameters listed in Table 5 were performed using classical computers. The design of experiments based on the Taguchi method for parameter selection was repetitively applied with a reduced number of variable parameters in each experiment to select the final value of each parameter, as shown in the last column of Table 5.

TABLE 5. EA parameters range and selected values.

Ser	Variable	Range of Values	Selected Value
1	No of agents	70, 100, 120, 150	120
2	Local group size	2, 5, 15, 25 agents	25
3	Local Migration condition	0, 10, 15, 30 generations	10
4	Global Migration condition	30, 70, 100, 200 generations	100
5	$\Delta\theta_1$	0.0001π, 0.0005π, 0.001π, 0.005π	0.001π
6	$\Delta\theta_2$		0.005π
7	$\Delta\theta_3$		0.001π
8	$\Delta\theta_4$		0.0005π
9	$\Delta\theta_5$		0.001π
10	$\Delta\theta_6$		0.001π
11	$\Delta\theta_7$		0.001π
12	$\Delta\theta_8$		0.001π

F. SENSITIVITY ANALYSIS FOR EA PARAMETERS

Sensitivity analysis of the proposed EA framework was conducted to examine the influence of these parameters on the performance of the algorithm [70]. The insights from this analysis based on the results obtained during the tuning process of the parameters are as follows.

- **Number of agents**. The selection of the number of agents in the proposed technique is similar to the population size in the classical EA. As expected, higher values for the number of agents lead to convergence to the optimal solution in fewer generations. However, the requirement for additional *qubits* and API calls to NISQ devices for a larger number of agents is the primary reason for restricting this value to 120.
- **Local/Global Migration**. The selection of values for the group size, local migration, and global migration parameters must be interdependent. It was observed that the selection of low values for migration frequently led to premature convergence of the EA, thus missing the global optimal solution. Alternatively, employing migration after a large number of generations leads to the island effect, where each subgroup evolves into separate results. Maintaining balance, we found that low local migration values and high global migration values were the optimal strategies for the problems in our study.
- $\imath\theta$. The rotation of the *qubits* is determined based on this parameter, and this rotation implements the primary functions of mutation and crossover in the proposed methodology. Our study revealed two contrasting effects of $\Delta\theta$ selection.
  - Low value of $\Delta\theta$. This leads to a slow improvement in the quality of the solution for the given problems. Thus, the number of generations required to reach the optimal value generally increases at lower values of $\Delta\theta$.
  - High value of $\Delta\theta$. This leads to early saturation of the *qubit*, which does not allow sufficient opportunities for the EA to explore the search space efficiently.

G. EXPERIMENTATION

Six TSPs are evaluated using the proposed algorithm. The problem size varied from 10 to 15 cities, and the problems were obtained from a set of benchmark TSPs used to solve drone routing, which are available on GitHub [71]. These problems have been considered in the classical TSP format, which is compatible with the TSPLIB [72]. The results discussed in the next section were derived from ten runs of the EA on each problem executed in hybrid mode on NISQ machines via the Internet cloud. The algorithms for the experiments were developed in Python using Qiskit and were executed on an open-source Linux OS.

Based on the parameters and framework discussed above, an overview of the quantum resources required for a 15-city TSP is presented below.

- Number of *qubits*/gene = $\lceil \log_2 n \rceil = \lceil \log_2 15 \rceil = 4$
- Number of *qubits*/chromosome = $n \times \lceil \log_2 n \rceil = 15 \times 4 = 60$
- Number of *qubits*/generation = chromosome $\times$ no of agents = $60 \times 120 = 7200$
- Number of *qubits* observed for 3000 generations = $7200 \times 3000 = 2, 16, 00, 000$ *qubits*
- Number of *qubits* observed on NISQ machines for 15-city TSP to run the experiment 10 times for 3000 generations = $21, 60, 00, 000$ *qubits*.

Finally, during the experiment for the six problems under consideration, each repeated 10 times, the total number of *qubits* observed on the IBMQ NISQ devices exceeded $1, 00, 00, 00, 000$ *qubits*. The results presented in the next section are based on ten runs of EA for each problem.

H. IMPLEMENTATION ON OTHER NISQ DEVICES AND SIMULATORS

We tested the efficacy and portability of the proposed technique by using various cloud-based NISQ devices and simulators. Owing to the limited availability of devices on the cloud, an additional test was conducted using a single problem generated using the coordinates of ten cities with a known minimum tour length of 7553. The results of running EA ten times on each device/simulator with a maximum generation of 600 for the above problem across various devices and simulators are presented in Table 6. These findings confirmed the feasibility of the proposed implementation across a spectrum of platforms and simulators. A brief description of the devices used in this experiment is provided below:

- **Ibm_nairobi**. This is a 7 *qubit* NISQ device available on the IBM Quantum Platform. The device is used over the cloud with API calls to run multiple quantum

TABLE 6. Implementation across various devices and simulators.

		Tour length					Min tour found	Generation to reach minimum length	
Method	**Runs**	**Minimum**	**Average**	**Median**	**Mode**	**Std Dev**	**Count**	**Minimum**	**Average**
Ibm_nairobi	10	7553	7,595.91	7561	7553	71.27	5	395	459.20
TN1	10	7553	7,635.44	7651	7553	86.72	2	416	472.40
QASM	10	7553	7,632.40	7561	7553	129.73	7	401	456.43
QEA	10	7553	7,620.13	7650	7553	70.15	5	411	466.20

circuits. A sample circuit for one of the rotation gates and observations is shown in Fig. 5.

- **TN1**. This is a high-performance tensor-network simulator. TN1 can simulate up to 50 *qubits* to implement a certain type of quantum circuits. The simulator was provided on the Amazon Braket platform.
- **QASM**. This simulator was provided by the Qiskit opensource ecosystem, which supports multiple simulation methods and configurable options for each simulation method. The simulator was installed in a Python environment on a Linux-based operating system. The experiments were performed using a local machine.
- **QEA**. These runs were performed using simulated Q-bits on a classical computer. For this purpose, trial observations were performed using the RANDOM function in Python library with a normal distribution.

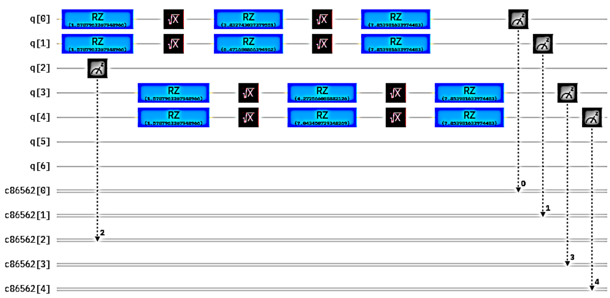


FIGURE 5. Sample circuit on ibm_nairobi.

VI. RESULTS

A. RECORD OF READINGS

The experimental results are presented in Tables 7 and 8. As mentioned above, ten experiments were conducted for each problem using NISQ machines with cloud-based access via the Internet. Table 7 provides the following values for each problem.

- **Optimal Reached**: It is the number of times the optimal tour length was achieved during the ten experiments.
- **Optimal**: Optimal tour length is established by applying the Bellman-Held-Karp Algorithm, which employs dynamic programming to determine the minimum cost tour [73].
- **Min/Max**: The minimum and maximum tour length achieved in the experiments.
- **Average** (μ): The average the tour length achieved during the ten experiments.
- **Standard Deviation** (σ): Standard Deviation of the tour length achieved during the ten experiments.

The number of generations required to reach the optimal value is a key parameter for evaluating EA quality. For the proposed EA in the hybrid mode, the values provided in Table 8 include only those experiments in which the optimal tour length was reached. The descriptions of the values presented in this table are as follows.

- **Min/Max**: Minimum and maximum number of generations that elapsed before reaching the optimal tour length.
- **Average** (μ): The average number of generations elapsed before reaching the optimal tour length.
- **Standard Deviation** (σ): The standard deviation of the number of generations elapsed before reaching the optimal tour length.

B. PLOTS

Plots of the results of the above experiments are shown in Figs. 6 and 7 for graphical appreciation of the performance of the proposed algorithm. Fig. 6 shows a plot of the tour lengths obtained for each TSP problem. For the problem size with cities 10, 11 and 13, the minimum, average, and maximum tour lengths obtained in ten runs overlapped. This indicates that the algorithm achieved an optimal tour in each run for these problems. Minor variations in the remaining problems indicate deviations in achieving the optimal tour length. Fig. 7 shows the number of generations required to reach the optimal tour length for each problem size. The graph shows that, as the size of the problem increases, the number of generations required to reach the optimal tour also increases. This is consistent with the logical view of problem solving.

VII. DISCUSSION OF RESULTS

The results demonstrate the feasibility and effectiveness of the proposed algorithm in solving TSP problems using present-day NISQ machines. Compared to prior implementations, such as QIEDA on QASM [34], which relied on simulations, the proposed Hybrid NISQ-Classical EA achieves superior performance. This can be attributed to the use of quantum superposition in the algorithm, which enhances solution diversity through intrinsic randomness. The comparative analysis and tabulated results highlight significant improvements in both minimum and average tour

TABLE 7. Tour length achieved in hybrid NISQ-classical EA.

		Experiments		Tour Length Achieved				
Problem Name	Cities	Runs	Optimal Reached	Optimal	Min	Max	μ	σ
uniform-51-n10	10	10	10	302	302	302	302	0
uniform-1-n11	11	10	10	328	328	328	328	0
uniform-1-n12	12	10	9	359	359	361	359.2	0.63
uniform-1-n13	13	10	10	330	330	330	330	0
uniform-1-n14	14	10	9	339	339	348	339.90	2.85
uniform-1-n15	15	10	8	391	391	420	394.3	9.12

TABLE 8. Number of generations to achieve optimal in hybrid NISQ-classical EA.

		Experiments		No of Generations to Reach Optimal[5]			
Problem Name	Cities	Runs	Optimal Reached	Min	Max	μ	σ
uniform-51-n10	10	10	10	235	490	289.8	74.87
uniform-1-n11	11	10	10	222	494	386.1	82.47
uniform-1-n12	12	10	9	404	1949	1069.89	721.37
uniform-1-n13	13	10	10	360	2379	1032.7	654.92
uniform-1-n14	14	10	9	419	2106	906.11	686.04
uniform-1-n15	15	10	8	412	2933	1084	907.28

TABLE 9. Comparison between QIEDA on QASM and hybrid NISQ-classical EA.

				QIEDA on QASM		Hybrid NISQ-Classical EA	
TSP Name	Cities	Runs	Optimal Length	Minimum Length	Average Length	Minimum Length	Average Length
uniform-51-n10	10	10	302	308	344	302	302
uniform-1-n11	11	10	328	330	344	328	328
uniform-1-n12	12	10	359	418	445	359	361
uniform-1-n13	13	10	330	397	456	330	330
uniform-1-n14	14	10	339	399	459	339	348
uniform-1-n15	15	10	391	548	584	391	420

TABLE 10. Victory-draw-defeat WRT minimum & average tour length.

Method	→	QIEDA on QASM	
↓	Value	Minimum Length	Average Length
Hybrid NISQ-Classical EA	Minimum Length	5-1-0	-
	Average Length	-	6-0-0

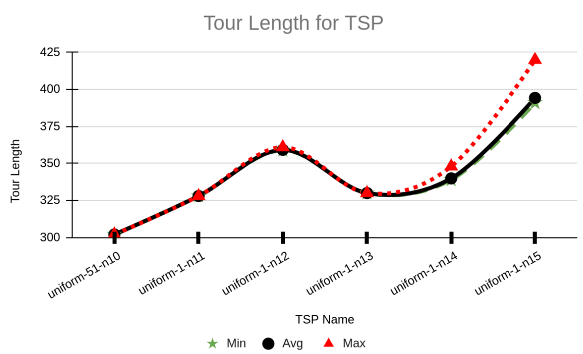


FIGURE 6. Tour length vs TSP name.

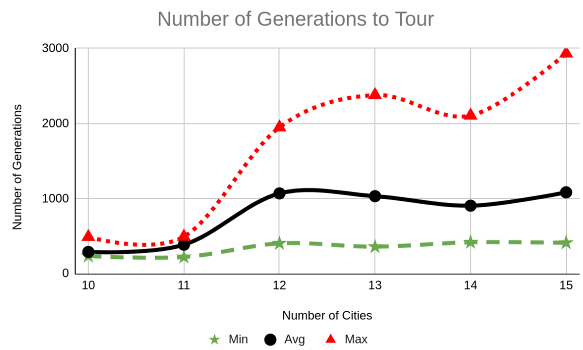


FIGURE 7. Number of generations to reach optimal tour.

lengths across various instances. These findings confirm the robustness of the proposed framework in leveraging quantum-classical hybrid techniques within the constraints of NISQ devices. Furthermore, for completeness, the results obtained with the proposed Hybrid NISQ-Classical EA were compared with QIEDA on QASM as proposed in [34]. For fair comparison, the problems under study were executed using the code provided in [74] for QIEDA [34]. The parameters for QIEDA have been retained as per the original values and QIDEA has been allowed to be executed for 2000 iterations in each run. Table 9 presents the values of QIEDA on QASM and Hybrid NISQ-Classical EA.

The comparative values in Table 9 illustrate the optimal tour length vis-à-vis the minimum and average tour lengths

TABLE 11. SGBET decoding for 12 city tour.

Example Tour: SGBET Decoding												Remarks
Chromosome	1010	0010	1010	1010	0010	0110	0101	0101	0101	1010	1101	Gray
Score Sequence	12	3	12	12	3	4	6	6	6	12	9	Gary Decode
Repair of Score Sequence	12	3	**13**	12	3	4	6	6	6	12	9	Repaired gene shown in bold
	12	3	13	**14**	3	4	6	6	6	12	9	
	12	3	13	14	**4**	4	6	6	6	12	9	
	12	3	13	14	4	**5**	6	6	6	12	9	
	12	3	13	14	4	4	6	**7**	6	12	9	
	12	3	13	14	4	4	6	7	**8**	12	9	
	12	3	13	14	4	4	6	7	8	**15**	9	
Repaired Score Sequence	12	3	13	14	4	4	6	7	8	15	9	
Sorted Score	3	4	5	6	7	8	9	10	12	13	14	
City Tour	1	4	5	6	7	8	10	11	0	2	3	Valid Tour

TABLE 12. SGBET decoding for 15 city tour.

Example Tour: SGBET Decoding																Remarks
Chromosome	0000	1111	1010	0010	1010	1010	0010	0110	0101	0101	0101	1010	1101	1111	0000	Gray
Score Sequence	0	10	12	3	12	12	3	4	6	6	6	12	9	10	0	Gary Decode
Repair of Score Sequence	0	10	12	3	**13**	12	3	4	6	6	6	12	9	10	0	Repaired gene shown in bold
	0	10	12	3	13	**14**	3	4	6	6	6	12	9	10	0	
	0	10	12	3	13	14	**4**	4	6	6	6	12	9	10	0	
	0	10	12	3	13	14	4	**5**	6	6	6	12	9	10	0	
	0	10	12	3	13	14	4	5	6	**7**	6	12	9	10	0	
	0	10	12	3	13	14	4	5	6	7	**8**	12	9	10	0	
	0	10	12	3	13	14	4	5	6	7	8	**15**	9	10	0	
	0	10	12	3	13	14	4	5	6	7	8	15	9	**11**	0	
	0	10	12	3	13	14	4	5	6	7	8	15	9	11	**1**	
Repaired Score Sequence	0	10	12	3	13	14	4	5	6	7	8	15	9	11	1	
Sorted Score	0	1	3	4	5	6	7	8	9	10	11	12	13	14	15	
City Tour	0	14	3	6	7	8	9	10	12	1	13	2	4	5	11	Valid Tour

obtained by the Hybrid NISQ-Classical EA and QIEDA on QASM. The "Pairwise Victory–Draw–Defeat" method was used for the statistical examination of these techniques. A summary of the Pairwise Victory–Draw–Defeat results for the TSP instances, comparing the two methodologies, is presented in Table 10. The dominance of Hybrid NISQ-Classical EA in producing better minimum and average tour lengths on the set of test case problems is clearly demonstrated by the number of wins in Table 10. Thus, we infer that the Hybrid NISQ-Classical EA produces better minimum and average tour lengths than QIEDA in the QASM for the TSP instances under study.

VIII. CONCLUSION

Present-day NISQ machines exhibit an array of complications, including restrictions on the available *qubits*, noise and error susceptibility, and *qubit* connectivity. Under these constraints, evolving methods for utilizing NISQ machines to solve real-world problems present both engineering and algorithmic challenges. This study successfully introduced a novel encoding scheme that allows for the implementation of EAs on NISQ machines with minimal overhead. This technique was successfully applied to current NISQ machines to solve TSP instances involving up to 15 cities, using a hybrid approach. To the best of our knowledge, this has not been reported previously.

Several avenues for future research have arisen based on these findings. The proposed methodology and encoding can be applied to explore its applicability to other combinatorial optimization problems and tackle larger instances of the TSP. The framework is effective in solving TSP instances. It has potential applications in real-world scenarios, such as logistics optimization, autonomous vehicle routing, and network design. Furthermore, this methodology can be extended to address other NP-complete problems, providing a foundation for future research into scalable quantum-inspired optimization techniques. In addition, it can be used to investigate the integration of advanced error-correction techniques to enhance the robustness of the algorithm against noise present in NISQ devices. The work presented can be expanded to combine this approach with machine learning methods to optimize the selection of hyperparameters in real time. Finally, a comparative analysis of the proposed method against other existing quantum algorithms is conducted to assess its performance across diverse problem sets. The methods discussed in this paper represent an ongoing effort to further explore the possibilities of improved optimization techniques that utilize contemporary quantum frameworks. The scalability of the proposed approach is primarily restricted by the number of *qubits* available on QPUs. As quantum devices evolve and qubit availability increases, this approach can be scaled to more complex problems. Future work will explore improved hardware, error impact, and mitigation techniques to enhance accuracy and efficiency.

APPENDIX
ADDITIONAL EXAMPLES OF SGBET

This appendix provides two examples of decoding a given bitstring by using Algorithm 1. These examples are presented in Tables 11 and 12 for TSPs with 12 and 15 cities, respectively.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of the Government of India, the Ministry of Electronics and Information Technology (MeitY), and the MEITY Quantum Computing Applications Laboratory (QCAL) in collaboration with AWS for providing access to AWS Braket credits as part of Cohort-2. These resources are instrumental for enabling quantum computing research. They also express their sincere gratitude to IBM for generously providing access to the Quantum Machines. Without such access, this study would not have been possible. They appreciate the support and assistance of IBM in enabling them to explore quantum computing.

REFERENCES

[1] P. Larrañaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, "Genetic algorithms for the traveling salesman problem: A review of representations and operators," *Artif. Intell. Rev.*, vol. 13, no. 2, pp. 129–170, 1999, doi: 10.1023/a:1006529012972.

[2] C. H. Papadimitriou, "The Euclidean travelling salesman problem is NP-complete," *Theor. Comput. Sci.*, vol. 4, no. 3, pp. 237–244, Jun. 1977, doi: 10.1016/0304-3975(77)90012-3.

[3] D. L. Applegate, R. E. Bixby, V. Chvat, and W. J. Cook, "History of TSP computation," in *The Traveling Salesman Problem: A Computational Study*. Princeton, NJ, USA: Princeton Univ. Press, 2006, pp. 93–128. [Online]. Available: http://www.jstor.org/stable/j.ctt7s8xg.7

[4] B. Toaza and D. Esztergár-Kiss, "A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems," *Appl. Soft Comput.*, vol. 148, Nov. 2023, Art. no. 110908, doi: 10.1016/j.asoc.2023.110908.

[5] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997. [Online]. Available: https://www.cs.ubc.ca/

[6] S. Xu, X. Hu, W. He, and K. Li, "Analysis of various methodologies for the TSP problem," *Theor. Natural Sci.*, vol. 28, no. 1, pp. 103–115, Dec. 2023, doi: 10.54254/2753-8818/28/20230393.

[7] R. van Bevern and V. A. Slugina, "A historical note on the 3/2-approximation algorithm for the metric traveling salesman problem," *Historia Mathematica*, vol. 53, pp. 118–127, Nov. 2020.

[8] A. H. Halim and I. Ismail, "Combinatorial optimization: Comparison of heuristic algorithms in travelling salesman problem," *Arch. Comput. Methods Eng.*, vol. 26, no. 2, pp. 367–380, Apr. 2019, doi: 10.1007/s11831-017-9247-y.

[9] E. Osaba, X. S. Yang, and J. D. Ser, "Traveling salesman problem: A perspective review of recent research and new results with bio-inspired metaheuristics," *Nature-Inspired Computation and Swarm Intelligence*. Amsterdam, The Netherlands: Elsevier, 2020, pp. 135–164, doi: 10.1016/B978-0-12-819714-1.00020-8.

[10] D. Eppstein, "The traveling salesman problem for cubic graphs," in *Algorithms and Data Structures. WADS 2003* (Lecture Notes in Computer Science), vol. 2748. Cham, Switzerland: Springer, 2003, doi: 10.1007/978-3-540-45078-8_27.

[11] E. G. Talbi, *Metaheuristics: From Design to Implementation*. Hoboken, NJ, USA: Wiley, 2009. [Online]. Available at: https://zeus.inf.ucv.cl/~bcrawford/DiplomadoIA_2021/Cap1_Metaheuristi cs_Talbi.pdf

[12] A. H. Halim, I. Ismail, and S. Das, "Performance assessment of the metaheuristic optimization algorithms: An exhaustive review," *Artif. Intell. Rev.*, vol. 54, no. 3, pp. 2323–2409, Mar. 2021, doi: 10.1007/s10462-020-09906-6.

[13] M. Gendreau and J. Y. Potvin, *Handbook of Metaheuristics*. New York, NY, USA: Springer, 2010, doi: 10.1007/978-1-4419-1665-5.

[14] G. Gutin and A. P. Punnen, *The Traveling Salesman Problem and Its Variations*, vol. 12. Cham, Switzerland: Springer, 2006. [Online]. Available: https://www.cs.rhul.ac.uk/home/gutin/paperstsp/chGYZ.pdf

[15] M. S. Zubairy, *Quantum Mechanics for Beginners: With Applications to Quantum Communication and Quantum Computing*. London, U.K.: Oxford Univ. Press, 2020. [Online]. Available: https://academic.oup.com/book/36992

[16] R. P. Feynman, "Quantum mechanical computers," *Found. Phys.*, vol. 16, no. 6, pp. 507–531, Jun. 1986, doi: 10.1007/bf01886518.

[17] D. Deutsch, "Quantum theory, the church-turing principle and the universal quantum computer," *Proc. Roy. Soc. London. A. Math. Phys. Sci.*, vol. A400, no. 1818, pp. 97–117, 1985, doi: 10.1098/rspa.1985.0070.

[18] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, 1994, pp. 124–134, doi: 10.1109/sfcs.1994.365700.

[19] L. K. Grover, "Quantum computers can search rapidly by using almost any transformation," *Phys. Rev. Lett.*, vol. 80, no. 19, pp. 4329–4332, 1998.

[20] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018, doi: 10.22331/q-2018-08-06-79.

[21] R. Campos, "Hybrid quantum-classical algorithms," 2024, *arXiv:2406.12371*.

[22] D. L. Applegate, R. E. Bixby, V. Chvátal, W. Cook, D. G. Espinoza, M. Goycoolea, and K. Helsgaun, "Certification of an optimal TSP tour through 85,900 cities," *Oper. Res. Lett.*, vol. 37, no. 1, pp. 11–15, Jan. 2009, doi: 10.1016/j.orl.2008.09.006.

[23] P. C. Pop, O. Cosma, C. Sabo, and C. P. Sitar, "A comprehensive survey on the generalized traveling salesman problem," *Eur. J. Oper. Res.*, vol. 314, no. 3, pp. 819–835, May 2024, doi: 10.1016/j.ejor.2023.07.022.

[24] A. Abbas, A. Ambainis, and B. Augustino, "Challenges and opportunities in quantum optimization," *Nature Rev. Phys.*, vol. 6, no. 12, pp. 718–735, Oct. 2024, doi: 10.1038/s42254-024-00770-9.

[25] S. Bhattacharyya, M. Köppen, E. Behrman, and I. Cruz-Aceves, *Hybrid Quantum Metaheuristics: Theory and Applications*. Boca Raton, FL, USA: CRC Press, 2022, doi: 10.1201/9781003283294.

[26] S. Hakemi, M. Houshmand, E. Kheirkhah, and S. A. Hosseini, "A review of recent advances in quantum-inspired metaheuristics," *Evol. Intell.*, vol. 17, no. 2, pp. 627–642, Apr. 2024, doi: 10.1007/s12065-022-00783-2.

[27] E. Osaba, E. Villar-Rodriguez, and I. Oregi, "A systematic literature review of quantum computing for routing problems," *IEEE Access*, vol. 10, pp. 55805–55817, 2022, doi: 10.1109/ACCESS.2022.3177790.

[28] G. Acampora and A. Vitiello, "Implementing evolutionary optimization on actual quantum processors," *Inf. Sci.*, vol. 575, pp. 542–562, Oct. 2021, doi: 10.1016/j.ins.2021.06.049.

[29] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 6, no. 6, pp. 580–593, Dec. 2002, doi: 10.1109/TEVC.2002.804320.

[30] M. D. Platel, S. Schliebs, and N. Kasabov, "Quantum-inspired evolutionary algorithm: A multimodel EDA," *IEEE Trans. Evol. Comput.*, vol. 13, no. 6, pp. 1218–1232, Dec. 2009, doi: 10.1109/TEVC.2008.2003010.

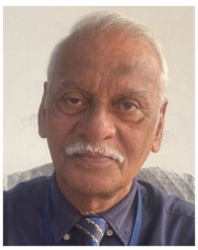
[31] N. Mani, G. Srivastava, A. K. Sinha, and A. Mani, "Effect of population structures on quantum-inspired evolutionary algorithm," in *Applied Computational Intelligence and Soft Computing*. London, U.K.: Hindawi Publishing Corporation, 2014, doi: 10.1155/2014/976202.

[32] F. Arute, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, Oct. 2019. [Online]. Available: https://www.nature.com/articles/s41586-019-1666-5
[33] Y. Rubio, C. Olvera, and O. Montiel, "Quantum-inspired evolutionary algorithms on IBM quantum experience," *Eng. Lett.*, vol. 29, no. 4, p. 1573, 2021. [Online]. Available: https://www.engineeringletters.com/issues_v29/issue_4/EL_29_4_30.pdf
[34] V. P. Soloviev, C. Bielza, and P. Larranaga, "Quantum-inspired estimation of distribution algorithm to solve the travelling salesman problem," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2021, pp. 416–425. [Online]. Available: https://oa.upm.es/74088/1/LARRANAGA_CONG_03.pdf
[35] E. Osaba, E. Villar-Rodriguez, I. Oregi, and A. Moreno-Fernandez-de-Leceta, "Hybrid quantum computing–Tabu search algorithm for partitioning problems: Preliminary study on the traveling salesman problem," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2021, pp. 351–358.
[36] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," 2014, *arXiv:1411.4028*.
[37] K. Srinivasan, S. Satyajit, B. K. Behera, and P. K. Panigrahi, "Efficient quantum algorithm for solving travelling salesman problem: An IBM quantum experience," 2018, *arXiv:1805.10928*.
[38] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, "Quantum computing with qiskit," 2024, *arXiv:2405.08810*.
[39] S. Chander, N. Sathishkumar, A. Hussain, and K. Blekos, "Solving the traveling salesperson problem with quantum approximate optimization algorithms," *Preprint*, Feb. 2024, doi: 10.13140/RG.2.2.17030.34882.
[40] B. Mete, "A novel approach for solving constrained optimization problems with QAOA using encoders," M.S. thesis, Dept. Inform., Technical Univ. Munich, Munich, Germany, 2022. [Online]. Available: https://mediatum.ub.tum.de/doc/1661426/document.pdf
[41] Y. Tang, J. Yan, and H. Edwin, "From quantum graph computing to quantum graph learning: A survey," 2022, *arXiv:2202.09506*.
[42] V. Pattabi. (2019). *A Quantum Phase Estimation Approach to the Traveling Salesman Problem*. [Online]. Available: https://cs269q.stanford.edu/projects2019/Pattabi_Y.pdf
[43] W. Qian, R. A. M. Basili, M. M. Eshaghian-Wilner, A. Khokhar, G. Luecke, and J. P. Vary, "Comparative study of variations in quantum approximate optimization algorithms for the traveling salesman problem," *Entropy*, vol. 25, no. 8, p. 1238, Aug. 2023.
[44] M. T. Khumalo, H. A. Chieza, K. Prag, and M. Woolway, "An investigation of IBM quantum computing device performance on combinatorial optimisation problems," *Neural Comput. Appl.*, vol. 37, no. 2, pp. 611–626, Jan. 2025, doi: 10.1007/s00521-022-07438-4.
[45] A. Lucas, "Ising formulations of many NP problems," *Frontiers Phys.*, vol. 2, p. 5, Feb. 2014, doi: 10.3389/fphy.2014.00005.
[46] C. McGeoch and P. Farré, "The D-wave advantage system: An overview," D-Wave Syst. Inc., Burnaby, BC, USA, Tech. Rep. 14-1049A-A, 2020. [Online]. Available: https://www.dwavequantum.com/media/s3qbjp3s/14-1049a-a_the_d-wave_advantage_system_an_overview.pdf
[47] H. Collins and K. Easterly. (Nov. 16, 2021). *IBM Unveils Breakthrough 127-Qubit Quantum Processor*. IBM Res. Commun. [Online]. Available: https://newsroom.ibm.com/2021-11-16-IBM-Unveils-Breakthrough-127-Qubit-Quantum-Processor
[48] S. Jain, "Solving the traveling salesman problem on the D-wave quantum computer," *Frontiers Phys.*, vol. 9, Nov. 2021, Art. no. 760783, doi: 10.3389/fphy.2021.760783.
[49] E. Villar-Rodriguez, E. Osaba, and I. Oregi, "Analyzing the behaviour of D'WAVE quantum annealer: Fine-tuning parameterization and tests with restrictive Hamiltonian formulations," 2022, *arXiv:2207.00253*.
[50] H. Kameyama, Y. Takita, M. Parizy, S. Sazawa, and P. Caballero, "Benchmarks for digital annealer with quadratic constrained binary optimization problems," in *Proc. Genetic Evol. Comput. Conf. Companion*, Jul. 2024, pp. 1954–1957. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3638530.3664114
[51] J.-R. Jiang, Y.-C. Shu, and Q.-Y. Lin, "Benchmarks and recommendations for quantum, digital, and GPU annealers in combinatorial optimization," *IEEE Access*, vol. 12, pp. 125014–125031, 2024, doi: 10.1109/ACCESS.2024.3455436.
[52] J. Boyd, "Silicon chip delivers quantum speeds [news]," *IEEE Spectr.*, vol. 55, no. 7, pp. 10–11, Jul. 2018, doi: 10.1109/MSPEC.2018.8389173.
[53] C. Gomes, J. P. Fernandes, G. Falcao, S. Kar, and S. Tayur, "A systematic mapping study on quantum and quantum-inspired algorithms in operations research," *ACM Comput. Surveys*, vol. 57, no. 3, pp. 1–35, Mar. 2025, doi: 10.1145/3700874.
[54] A. Rajak, S. Suzuki, A. Dutta, and B. K. Chakrabarti, "Quantum annealing: An overview," *Phil. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 381, no. 2241, Jan. 2023, Art. no. 20210417, doi: 10.1098/rsta.2021.0417.
[55] IBM Quantum. (2024). *IBM Quantum Experience*. [Online]. Available: https://quantum.ibm.com/
[56] Amazon Web Services. (2020). *Amazon Braket: Explore and Experiment With Quantum Computing*. [Online]. Available: https://aws.amazon.com/braket/
[57] F. Gray, "Pulse code communication," U.S. Patent 2 632 058, Mar. 17, 1953. [Online]. Available: https://cir.nii.ac.jp/crid/1571698600974840832
[58] K. E. Mathias and L. D. Whitley, "Transforming the search space with gray coding," in *Proc. 1st IEEE Conf. Evol. Comput. World Congr. Comput. Intell.*, vol. 1, Jun. 1994, pp. 513–518, doi: 10.1109/ICEC.1994.349897.
[59] M. L. Llid, "The traveling salesman problem domain application of a fundamentally new approach to utilizing genetic algorithms," MITRE Corp., Bedford, MA, USA, Tech. Rep., 1991.
[60] M. A. Aytekin and T. E. Kalayc, "A new approach using binary encoding to solve traveling salesman problem with genetic algorithms," *Algorithms*, vol. 20, pp. 1–6, Jun. 2010. [Online]. Available: https://www.researchgate.net/profile/T-Kalayci
[61] P. Gregor and T. Mütze, "Trimming and gluing gray codes," *Theor. Comput. Sci.*, vol. 714, pp. 74–95, Mar. 2018.
[62] R. W. Doran, "The gray code," *Dept. Comput. Sci., J. Universal Comput. Sci., Univ. Auckland*, vol. 13, pp. 1573–1597, Jan. 2007. [Online]. Available: https://api.semanticscholar.org/CorpusID
[63] S. Suman and S. Shukla, *Advanced Data Structures: Theory and Applications*. Boca Raton, FL, USA: CRC Press, 2019, doi: 10.1201/9780429488757.
[64] A. Dey, S. Dey, S. Bhattacharyya, J. Platos, and V. Snasel, "Quantum inspired meta-heuristic approaches for automatic clustering of colour images," *Int. J. Intell. Syst.*, vol. 36, no. 9, pp. 4852–4901, Sep. 2021, doi: 10.1002/int.22494.
[65] A. R. Mansori and S. K. Nguyeni, "Quantum-inspired genetic algorithms for combinatorial optimization problems," *Algorithm Asynchronous*, vol. 1, no. 1, pp. 16–23, Aug. 2023. [Online]. Available: https://hasmed.org/index.php/jourasy/article/download/47/66/465
[66] M. G. P. Lacerda, L. F. A. Pessoa, F. B. L. Neto, T. B. Ludermir, and H. Kuchen, "A systematic literature review on general parameter control for evolutionary and swarm-based algorithms," *Swarm Evol. Comput.*, vol. 60, Feb. 2021, Art. no. 100777, doi: 10.1016/j.swevo.2020.100777.
[67] N. Mani and A. Mani, "Solving combinatorial optimization problems with quantum inspired evolutionary algorithm tuned using a novel heuristic method," 2016, *arXiv:1612.08109*.
[68] A. Bendell, J. Disney, and W. Pridmore, *Taguchi Methods: Applications in World Industry*. Hoboken, NJ, USA: Wiley, 1989. [Online]. Available: https://api.semanticscholar.org/CorpusID:106732039
[69] R. K. Roy, *Design of Experiments Using the Taguchi Approach*. Hoboken, NJ, USA: Wiley, 2022.
[70] C. Srinivas, B. R. Reddy, K. Ramji, and R. Naveen, "Sensitivity analysis to determine the parameters of genetic algorithm for machine layout," *Proc. Mater. Sci.*, vol. 6, pp. 866–876, Jan. 2014, doi: 10.1016/j.mspro.2014.07.104.
[71] N. Agatz, P. Bouman, and M. Schmidt, "Optimization approaches for the traveling salesman problem with drone," *Transp. Sci.*, vol. 52, no. 4, pp. 965–981, Aug. 2018, doi: 10.1287/trsc.2017.0791.
[72] G. Reinelt, "TSPLIB—A traveling salesman problem library," *ORSA J. Comput.*, vol. 3, no. 4, pp. 376–384, Nov. 1991, doi: 10.1287/ijoc.3.4.376.
[73] R. Bellman, "Dynamic programming treatment of the travelling salesman problem," *J. ACM*, vol. 9, no. 1, pp. 61–63, Jan. 1962, doi: 10.1145/321105.321111.
[74] V. P. Soloviev. (2021). *QIEDA: Quantum-Inspired Estimation of Distribution Algorithm*. Accessed: Feb. 27, 2024. [Online]. Available: https://github.com/VicentePerezSoloviev/QIEDA

RAVI SAINI received the Bachelor of Technology degree in electrical and electronics engineering from the Naval College of Engineering, Lonavala, and the Master of Technology degree in computer science from the Indian Institute of Technology, Madras (IIT Madras). He is currently pursuing the Ph.D. degree in quantum evolutionary algorithms for the optimization of autonomous routing in UAVs.

He has cleared the UGC NET as an eligibility for an Assistant Professor. He was a recipient of several awards, including the H. N. Mahabala Medal from IIT Madras, in 2007.

ASHISH MANI received the M.Tech. degree in engineering systems and the Ph.D. degree in quantum computing from Dayalbagh Educational Institute (Deemed University), India. He is currently a Professor with the Amity Innovation and Design Centre, Amity University, Noida, Uttar Pradesh, India. With over 23 years of research and teaching experience and four years of experience in the industry, his expertise spans artificial intelligence, quantum computing, machine learning, embedded systems, and the IoT. He has published over 100 papers in reputed journals and conferences, co-edited two books, and holds several patents in advanced technologies. His interdisciplinary research includes applications in healthcare, energy systems, and autonomous systems. A passionate advocate of innovation, he actively mentors deep-tech startups and serves as the Director for the Acuminous Research Foundation, promoting science and technology through research and innovation.



M. S. PRASAD is an Ex-DRDO Scientist with over 20 years of experience in designing complex defense systems and space system simulations and designs. He has been the Program Director for notable projects, such as Meghna, MOR, ASTROIDS, Shatranj, and RESINS (ISRO), and has contributed to the development of unique multi-target tracking algorithms and guidance seekers based on radiometric imaging. He is currently an Alumnus of IIT Kharagpur and IIT Kanpur, and a Research Scholar with Texas A&M University, where he developed nonlinear 3D asymmetric body motion simulations for the space station. At DRDO, he was instrumental in creating navigation and guidance systems for missiles and redundant navigation systems for satellites and contributed to several classified defense projects. He was a Data Network Security Auditor with the UN's Organization for the Prohibition of Chemical Weapons (OPCW), The Hague, for two years. He is currently a Professor with the Amity Institute of Space Science and Technology, Amity University, Noida, Uttar Pradesh, India. He has published 45 papers in national and international journals, and conferences. He is a recognized expert in aerospace and defense and has been on advisory boards for significant initiatives in drone regulations, low-altitude air traffic management, and AI ethics and bias.

Dr. Prasad has received numerous accolades, including the DRDO Best Scientist Award (twice) and Texas Instruments Best Innovative DSP Design Award, in 1978.

SIDDHARTHA BHATTACHARYYA (Senior Member, IEEE) received the bachelor's degree in physics and the bachelor's and master's degrees in optics and optoelectronics from the University of Calcutta, Kolkata, India, in 1995, 1998, and 2000, respectively, the Ph.D. degree in computer science and engineering from Jadavpur University, Kolkata, in 2008, and the Ph.D. (Habilitation) degree from VŠB-Technical University of Ostrava, Czech Republic, in 2023.

He is currently a Senior Researcher with the Faculty of Electrical Engineering and Computer Science, VŠB-Technical University of Ostrava. He is also the Scientific Advisor at the Algebra Bernays University, Zagreb, Croatia. He was the Principal of Rajnagar Mahavidyalaya, Birbhum, India. He was a Professor with the Department of Computer Science and Engineering, CHRIST (Deemed to be University), Bengaluru, India. He was the Principal of the RCC Institute of Information Technology, Kolkata. He was also a Senior Research Scientist with the Faculty of Electrical Engineering and Computer Science, VŠB-Technical University of Ostrava. He has authored and co-authored over 480 research publications in international journals and conference proceedings. He co-authored six books and co-edited 112 books. He holds more than 20 patents. His research interests include hybrid intelligence, pattern recognition, multimedia data processing, social networks, and quantum computing.

JAN PLATOS (Member, IEEE) received the Ph.D. degree in computer science from the VŠB-Technical University of Ostrava, in 2006.

He was an Associate Professor of computer science, in 2014. Since 2017, he has been the Head of the Department of Computer Science, Faculty of Electrical Engineering and Computer Science, VŠB-Technical University of Ostrava. He has been the co-author of over 180 scientific articles published in proceedings and journals. His citation report consists of 302 citations and an H-index of nine on the Web of Science, 685 citations, and an H-index of 13 on Scopus, and 2660 citations and an H-index of 22 on Google Scholar. His research interests include computer science, text processing, data compression, bio-inspired algorithms, information retrieval, data mining, data structures, data prediction, the application of different computer science algorithms and methods, and innovative approaches to severe practical problems in any area.