Reordering and Partitioning of Distributed Quantum Circuits

DAVOOD DADKHAH¹, MARIAM ZOMORODI ²,³, SEYED E布拉HM HOSSEINI¹, PAWEL PLAWIAK³,⁴, AND XUJUAN ZHOU⁵(Member, IEEE)

¹Department of Electrical Engineering, Ferdowsi University of Mashhad
²Department of Computer Engineering, Ferdowsi University of Mashhad
³Department of Computer Science, Faculty of Computer Science and Telecommunications, Cracow University of Technology, Krakow, Poland
⁴Theoretical and Applied Informatics, Polish Academy of Sciences, Gliwice, Poland
⁵School of Business, University of Southern Queensland, Springfield Campus, Queensland, Australia

Corresponding authors: MARIAM ZOMORODI (e-mail: zomorodi@pk.edu.pl) and XUJUAN ZHOU (e-mail: xujuan.zhou@usq.edu.au).

ABSTRACT A new approach to reduce the teleportation cost and execution time in Distributed Quantum Circuits (DQCs) was proposed in the present paper. DQCs, a well-known solution, have been applied to solve the problem of maintaining a large number of qubits next to each other. In the distributed quantum system, the qubits are transferred to another subsystem by a quantum protocol like teleportation. Hence, a novel method was proposed to optimize the number of teleportation and to reduce the execution time for generating DQC. To this end, first, the quantum circuit was reordered according to the qubits placement to improve the computational execution time, and then the quantum circuit was modeled as a graph. Finally, we combined the genetic algorithm (GA) and the modified tabu search algorithm (MTS) to partition the graph model in order to obtain a distributed quantum circuit aimed at reducing the number of teleportation costs. A significant reduction in teleportation cost (TC) and execution time (ET) was obtained in benchmark circuits. In particular, we performed a more accurate optimization than the previous approaches, and the proposed approach yielded the best results for several benchmark circuits.

INDEX TERMS Quantum computing, Distributed quantum circuit, Optimization, Genetic algorithm, Teleportation.

I. INTRODUCTION

In the field of computation, quantum computing is a new field with a high capacity to perform classical computing more efficiently [1]. The quantum computing was first introduced by researchers in the 1980s. It has presented new algorithms for computer science to accelerate problem solving compared to classical algorithms [2], [3]. Undoubtedly, the use of quantum computation is indispensable, since it employs fundamental techniques, such as encryption and optimization, causing classical problems, to be solved. Nevertheless, there are significant new trends and approaches into quantum computability system [4], quantum algorithms [5], quantum communication protocols [6], [7] and quantum mechanics [8], particularly with the advent of the quantum information theory [1].

But, there are some problems with quantum computing. For example, when the number of qubits grows, protecting them from destroyed interactions and decoherence rapidly becomes uncontrollable [9]. Therefore, for a distributed quantum circuit, development of a quantum circuit with limited quantum subsystems communicating with each other through quantum teleportation is a logical method to use. To build the distributed quantum circuit, it is necessary to make multi quantum circuits with a limited capacity connected together through a classical or quantum channel, and they implement the functionality of a distributed quantum circuit. DQCs are a model for quantum circuits consisting of subsystems, which are far from each other and have many qubits.

Briefly, DQCs are comprised of several subsystems having qubits and gates, so that qubits are exchanged between subsystems by teleportation (a technique for transferring qubits). Then, the qubits are returned to the first subsystem after completing the operation.

In the present paper, we intended to survey the performance of the distributed quantum circuit by reordering the qubits placement and implementing the combination genetic and modified tabu search algorithm to reduce the execution time and the number of teleportation cost.
In this work, we developed an optimization approach to improve the execution time for generating distributed quantum circuits in order to reduce the number of teleportation cost significantly. Since the general optimization of distributed quantum circuits is NP-hard, we presented the genetic algorithm and the modified tabu search algorithm to partition the graph model of quantum circuit in order to create a distributed quantum circuit with the minimum number of teleportation. Afterward, we implemented our optimization approach in several distributed quantum circuits. In this paper, as a benchmark, quantum circuits were selected from the RevLib website [10] and Quipper’s library [11], [12] to test the proposed optimization. In all circuits, we focused on the number of teleportation and the execution time to optimize for the distributed quantum circuit.

Tabu search (TS) is one computational algorithm for optimization and is widely used in many sciences. Tabu search (TS) is a memory-based algorithm. However, TS cannot completely perform optimization problems. The modified version of TS (MTS) has been applied to improve its performance, for example, MTS has presented a significant performance when used to partition graph problems. Owing to motion mechanisms, MTS had rapid convergence to solution, and it performed a considerable search in the potential solution. Thus, we applied the modified tabu search, so that the problem of partitioning was best solved, and the optimal partitioning for the large distributed quantum circuits was obtained.

The long interaction distances between the qubits within the quantum circuit may increase the execution time for generating the distributed quantum circuit. Hence, the reordering leads to approach qubits for executing the quantum gates within a quantum circuit and ultimately reduces the execution time in DQC.

This work consists of four steps. In the first step, reordering operation is applied to the qubits placement in the initial quantum circuit; in the second step, reordered quantum circuit is modeled by a graph; in the third step, modified tabu search (MTS) partitions the graph model; and in the fourth step, genetic algorithm (GA) is presented to improve the graph partitioning.

The paper has been organized as follows: Section II explains some basic concepts of quantum computation. Section III presents the related work. Section IV offers the definitions required for the distributed quantum circuit. Sections V and VI discuss our proposed approach and results of simulation, respectively. Finally, Section VII concludes the paper.

II. BACKGROUND

Quantum computation is a novel computational field in which qubit is the basic unit of information. The state of a qubit is shown by a unit vector in a Hilbert space labeled as

\[ |\psi\rangle = \alpha|0\rangle + \beta|1\rangle \]  

(1)

where \( |0\rangle \) and \( |1\rangle \) are the basis of space, and \( \alpha \) as well as \( \beta \) are complex coefficients establishing \( |\alpha|^2 + |\beta|^2 = 1 \).

We used some single- and two-qubit gates, such as single-qubit Hadamard, T and S, and two-qubit CNOT gate in our proposed method.

Fig.1 shows the representation of the CNOT gate.

![FIGURE 1. Circuit of the CNOT gate](image)

\[ H_{adamard} = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix} \]  

(2)

\[ T = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \]  

(3)

\[ S = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \]  

(4)

Let a quantum system be formed by two components that are placed in Hilbert space \( H_1 \) and \( H_2 \), respectively. Therefore, the whole system is placed in a Hilbert space \( H = H_1 \otimes H_2 \), so that the state vector is shown as

\[ |\psi\rangle = \sum_{i,j} d_{i,j} |e_{1,i}\rangle |e_{2,j}\rangle \]  

(5)

where \( \{ e_{a,i}\} \) is considered the basis in the Hilbert space and \( \sum_{i,j} d_{i,j} = 1 \).

In this paper, qubit transfer between subsystems was assumed to be accomplished through teleportation operation. In teleportation, the basic process is to transmit a quantum state of qubit, so that the destination receives the same state as the initial qubit state. Therefore, the initial state is removed, so that quantum teleportation matches the no-cloning theorem [13]. Moreover, the aim of quantum teleportation is to transfer a quantum state of qubit using two classical bits, so that the receiver generates the same state as the initial qubit state. Fig.2 presents the quantum circuit of qubit teleportation [14].

![FIGURE 2. Quantum circuit of qubit teleportation](image)

III. RELATED WORK

In this section, the previous methods for distributed quantum circuit have been reviewed. Grover [15], Cleve and Buhrman [16], and Cirac et al. [17] were the pioneers who started to survey the distributed quantum computing field. Grover divided a quantum circuit in which particles are far from each
other, and each particle completed its computation. Thus, he presented a distributed quantum system. In this system, requiring information is transferred from one particle and received in another particle. He also showed the use of this distribution system and presented a quantum algorithm to calculate the total time required to compute the number of distributed particles. Beals et al. also indicated that a quantum circuit could be modeled as a distributed quantum circuit by introducing the hypercube graph, so that each subsystem was placed on each vertex of a hypercube graph in [18]. Yepez et al. considered two communications structures for distributed quantum computing. Quantum subsystems are communicated through the quantum channel, while classical subsystems are communicated through classical channels. Quantum systems in which all qubits can be entangled to each other and subsystems are communicated to each other through classical channels in [19]. Streltsov et al. presented a way to distribute entanglement and indicated the minimum communication cost for sending entangled particles between subsystems. They also showed that the total entanglement between two particles sent should not be more than the total quantum communication required for moving each particle between two subsystems in [20].

Ying et al. determined an algebraic language to introduce the distributed quantum system. They also defined some notations for distributed quantum computing in [21]. In addition, the hyper graph-partitioning approach was presented for the quantum circuit. Authors modeled the quantum circuit as a hypergraph model. Finally, they employed the proposed model to be applied on five quantum circuits in [22]. However, their approach did not consider the execution time for partitioning graph.

In [23], the authors presented an algorithm optimization, so that reduced the teleportation cost of the distributed quantum circuit (DQC) in terms of the number of teleportation. They showed that DQCs were the potential solution for a multi subsystem and proposed an algorithm to optimize the cost of communication consisting of two subsystems. In the end, the final configuration with the minimum number of teleportation was reported. However, their optimization algorithm divided quantum circuit into only two partitions, whereas they could consider their proposed approach to create multiple partitions. Moreover, in [24], a GA was introduced to perform the optimization process on distributed quantum circuits more efficiently. The authors found the proposed approach had executed the genetic algorithm with high speed. Shor’s algorithm was introduced in [25]. In the proposed model, a distributed quantum circuit was implemented to execute the non-local gates of algorithm. Furthermore, the authors do not report the number of quantum teleportation for this distributed quantum circuit. Daei et al. presented a graph model to generate distributed quantum circuits from monolithic quantum circuits, so that communication between partitions of a distributed quantum circuit was minimized in [26]. However, their approach did not examine the execution time needed to generate graph-partitioning. Davarzani et al. proposed the algorithm consisting of two steps: The quantum circuit was converted into a bipartite graph model, and a dynamic programming approach was applied to generate distributed quantum circuits in [27].

IV. PROBLEM DEFINITION

A distributed quantum circuit (DQC) includes N number of quantum circuits that all of them construct a distributed quantum circuit. In a DQC, qubits are transferred between the subsystems by teleportation. In each subsystem, qubits are labeled from the top to the bottom line, where the ith line corresponds to the ith qubit. In the present work, we defined two kinds of quantum gates in a DQC as follows:

A local gate: Its control and target line are in the same partition applying on the local qubit. Each partition may have a finite number of local gates.

A global gate: Its control and target line are in different partitions. To run a global gate, a qubit in the current partition should be teleported to another partition to perform gate. Moreover, in a distributed quantum circuit, when a qubit is sent into another subsystem, it does not execute in its initial subsystem anymore. To accomplish local gates, qubits can be performed in their initial subsystem. Furthermore, to execute global CNOT gates, qubits should be executed in the same subsystem. Quantum teleportation is a good solution to transfer qubits between partitions. Suppose that there is a two-partition system with Subsystem A and Subsystem B. In this system, there are two ways to perform any global gate. The qubit in Subsystem A is transferred to Subsystem B through teleportation. This is also true the other way around. Single-qubit and local CNOT gates should be executed in their local subsystems.

Although at first glance, DQCs are similar to quantum circuits (QCs), the problem with a DQC is essentially different from that of a QC. In a DQC, the optimization process is focused on reducing the computational execution time and teleportation costs. In the current study, to optimize the distributed quantum circuit, an approach was proposed based on optimization algorithms, so that we attempted to find a desirable DQC with reduced teleportation costs. Our proposed approach calculated the minimum number of teleportation for each configuration, where global gates had an individual position between partitions for each configuration. Ultimately, the minimum number of teleportation was reported for all configurations. We also strived to begin with an initial quantum circuit like basic gates (i.e., CNOT and single-qubit gates). In the current work, we considered a quantum circuit, including basic gates (i.e., CNOT and single-qubit), which were applied on qubits.

We consider a graph as G(V, E), where V is a set of vertices, and E is a set of edges formed by pairs of vertices, respectively. The graph-partitioning problem divides graph into K partitions, so that we try to minimize the connection among all different partitions.

$$\sum_{i=1,2,\ldots,n} \sum_{j=i+1,\ldots,n} W(v_i, v_j)$$  (6)
Where $W(v_i, v_j)$ is the weight of between vertices $v_i$ and $v_j$ for all $v_i \in p_i, v_j \in p_j$.

**A. REORDERING OF QUBITS PLACEMENT**

We proposed reordering of the qubits placement in quantum circuits. This process can reduce the execution time to generate the distributed quantum circuit. In this section, our objective is to reorder the qubits placement based on the minimum nearest neighbor cost. Before partitioning of qubits into determined partitions as discussed in the next section, we apply the proposed approach in which a suitable qubit order is determined for each quantum circuit. For this goal, it is applied to each quantum circuit extensively. The reordering plays a crucial role in improving the execution time in DQC. So, we construct the matrix model for quantum circuit. In this matrix, the relationship between the qubits in the quantum circuit can be modeled as follows:

\[
\{ 1 \text{ if two qubits } (q_i, q_j) \text{ are communicated.} \\
\{ 0 \text{ otherwise} 
\]

Then, reordering of the qubits placement is carried out on the corresponding matrix according to the minimum nearest neighbor cost. The minimum nearest neighbor cost of the quantum circuit is determined as

\[
\text{NNCQC} = \frac{1}{2} \left( \sum_{i=1}^{n} \sum_{j=i+1}^{n} A_{ij}(|i-j|-1) \right) 
\]

The reordering operation was kept until the minimum NNCQC was attained. The optimization process depended on the number of qubits in the initial quantum circuit. Finally, we constructed a new quantum circuit whose qubits placement could be closed to each other and could improve the execution time when the distributed quantum circuit was generated.

For instance, consider the quantum circuit in Fig.3. We reorder the qubits placement based on the minimum nearest neighbor cost from $\{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$ to $\{q_7, q_4, q_1, q_3, q_2, q_5, q_6, q_8\}$ as shown in Fig.4.

**B. GRAPH MODELING BASED ON LABELING OF QUANTUM CIRCUIT**

The basic method presented in this section is to employ the graph structure to model the quantum circuit. In our quantum representation model, we consider a set of qubits (V), a set of input qubits (I), a set of output qubits (O) and a set of gates like U on the qubits.

The set of U gates consists of gates like $U_1 U_2 \ldots$. Each of these gates may be dependent on the gates before them. It should be noted that these gates are applied from left to right, and to show the labeling model, we use the graph structure displayed as $G = (V, E)$ based on $\{I, O\}$ these components:

- **V**: Represents the set of nodes of graph G that each node of the graph represents a qubit.
- **I**: Represents the set of the input qubit of the U gate.
- **O**: Represents the set of the output qubit of the U gate.
- **E**: Represents the edges of graph G, which indicates the relationship between two nodes.

To determine the CNOT gate in the graph model, we use the following method to label the input and output in a quantum circuit and we model a quantum circuit into the graph based on differences labeling the target of the CNOT gate.

This labeling is such that if the two-part input and output of CNOT gates have the same label, these gates realize the same function. Hence, all CNOT gates having a different functionality are implemented in this model.

1. First, we label all quantum circuit input lines as 0 and consider an empty list $L_i$ for each quantum circuit. Each list $L_i$ keeps the input and output labeling of CNOT gates.
2. For each CNOT gate from the inputs toward the outputs of the quantum circuit, we apply the following step for CNOT gates:
   - If there are $k$ CNOT gates $C_1, \cdots, C_k$ that realize the quantum circuit, i.e., $C_1, \cdots, C_k = I$, then the label output side of the target for the latest CNOT gate equals the label on the input side of the target for the first CNOT gate. Therefore, we omit these CNOT gates from the graph model.
   - We propose creating an edge in graph model from the labeling by applying the following procedure:
     \[
     \text{comp}((c_i, t_j)\text{labeling,} (c'_i, t'_j)\text{labeling})
     \]

   where comp represents the comparison operation for the labeling of the inputs and outputs of each CNOT gate. Using the labeling method, we can find all changes in the input and output qubits of a CNOT gate.
In other words, by labeling the input qubits of a CNOT gate as \((c_i, t_j)\) and its output qubits as \((c_i', t_j')\), changes to the labeled inputs and outputs of CNOT gates can be saved in the \(L_i\) list, and with the help of this list, we can recognize the existence of the CNOT gate. After we find changes in labeling according to list \(L_i\), we must regard these changes for CNOT gates and begin to model a quantum circuit into a graph based on the order of qubits. The reason for this is that the qubits of CNOT gates depend on other possible qubits. To determine CNOT gates, we should consider the labeling method to represent the graph model of a quantum circuit. Because no edge is constructed, we do not consider the one-qubit in the graph model like \(\{H, T, T^4\}\).

However, for the CNOT gate, we assume two qubits as control and target as \(CNOT(q_1, q_2)\). Thus, a connection is needed for showing it with an edge between \(q_1\) and \(q_2\). The graph models all quantum qubits and gates in a quantum circuit to all interconnections needed to implement it. We consider a sample quantum circuit in Fig.5. In this figure, the sets of qubits are \(Q = \{q_1, q_2, q_3, q_4\}\) and CNOT gates are \(G = \{g_1, g_2, g_3, g_4, g_5, g_6, g_7\}\), respectively. To create a graph model as shown in Fig.6, first, we consider the labeling \((q_1, q_2, q_3, q_4) = (0, 0, 0, 0)\).

For the first CNOT gate, the target \((q_1, q_2)\) is assigned labeling 1. The line that is not empty (\(\oplus\)) should be labeled by 1 or 0. The qubits of the first CNOT gate are recorded in the list \(L_i\), so that the obtained labeling is \((0, 0, 0, 0)\) and \((0, 1, 0, 0)\) for the input and output, respectively.

The implementation of the proposed model for a quantum circuit will continue until all CNOT gates of a quantum circuit are modeled as shown in Table 1. Afterward, we find labeling changes to model the CNOT gate in a quantum circuit into the graph model based on the obtained list \(L_i\). Therefore, edges \(E_{12}(q_1, q_2)\) are added to the graph model. Other edges are modeled as

\[
E = \{E_{31}(q_3, q_1), E_{41}(q_4, q_1), E_{43}(q_4, q_3), E_{42}(q_4, q_2), E_{12}(q_1, q_2), E_{42}(q_4, q_2)\}
\]

**FIGURE 5.** Sample quantum circuit

**FIGURE 6.** Graph model for Figure 5

**FIGURE 7.** The representation of chromosome

**C. GENETIC ALGORITHM**

Genetic algorithm has been introduced as an optimization model for evolutionary computation. It is used to generate optimal solutions to optimize difficult problems. The partitioning of graph is an NP-hard problem, which can be solved by the genetic algorithm. Such an algorithm is commonly applied to generate the converged solutions by recombination, mutation operation, and production of new generations. Through a genetic model, a distributed quantum circuit was modeled to a graph, and then it was attempted to be partitioned with a minimum number of cuts (number of teleportation cost) by using the genetic algorithm. First, we presented a chromosome to indicate the way in which the proposed method was used in the genetic algorithm model. In this algorithm, a chromosome structure, whose elements represented partition of the vertices, was used. It assigns each gene a number 1 to \(k\) depending on which the gene belongs to the specified partition of vertices. For example, in the following chromosome, Vertex 1 is in Partition 1, and Vertex 2 is in Partition 3. The number of cuts (the teleportation cost of the distributed quantum circuit) is calculated when vertex \((v_i)\) in the partition \((p_i)\) is connected to the vertex \((v_j)\) in the partition \((p_j)\) by edges (global gates) as the considered chromosome. The points of cut are changed by using the genetic algorithm until the lowest quantum teleportation cost is achieved. In a genetic algorithm, a population of candidate solutions always evolves to obtain an optimized solution. Each chromosome represents a set of properties, which can be mutated and replaced. GA search is also an iterative process with a generation (a population in each iteration). To implement genetic algorithms, first we start to define the structure of the genetic algorithm used for finding best solutions. The following procedure scheme was used in this paper.

Fitness function \((f)\), which is the number of communication costs between the partitions, is evaluated by using an optimization function \(F(v)\). This function tries to attain the minimum number of teleportation, which is the main objective, and they often include multiple limitations. An optimization
function is defined as

$$O = F(v) = \min \left( f_1(v), f_2(v), \ldots, f_n(v) \right)$$

with $v = (v_1, v_2, \ldots, v_n) \in V$

$$o = (o_1 = f_1(v), o_2 = f_2(v), \ldots, o_n = f_n(v)) \in O$$

Subject to

$$W = \sum_{i \neq j} w(v_i, v_j) \leq \lambda$$

where $v$ is a vector of vertices variables, $n$ is the number of vertices, $V$ is the input space of the problem, $o$ is an output vector of vertices, and $O$ is the output space of the problem. Each fitness function $f(v)$ is an objective function, $W$ is the sum of weights of edges between partitions that is considered the feasible limitation for possible solutions, and $\lambda$ is a real number implying that $v_i$ is connected to $v_j$. The optimal solutions are the set of all possible solutions obtained by the optimization function within feasible limitations. The total cost of edges ($n_e$) for each vertex ($v_i$) connected to other vertices ($v_j$) for different partitions is obtained by

$$n_e = \min \left( \sum_{i=1}^{n} \sum_{j=1, i \neq j}^{n} w(v_i, v_j) \right)$$

The best individuals are selected from the current population by using the roulette wheel selection strategy, and genes of each chromosome are modified to form a new generation so that they are recombined and randomly mutated. GA applies a two-point crossover operation to randomly generate two new offsprings by swapping genes of the chromosomes. As Fig.8 depicts, Crossover operation starts by randomly selecting and moving parts of the parents with each other. It is applied to cut each of the selected parents from the determined points and to generate a new chromosome by recombining them with probability-pm. We implement the mutation operation to flip the chosen genes of a chromosome to generate new solutions with probability-pm. It randomly selects one or more partitions of the vertex and replaces them with other ones depending on the probability of the mutation operation. Fig.9 presents the mutation procedure. As Fig.10 shows, using the crossover, mutation, and recombination operation, we attain high searching performance while maintaining the population’s diversity. Genetic operators are applied to the selected chromosome to generate a new population. Ultimately, the genetic algorithm terminated is obtained for the population when the number of generations exceeded from acceptable number or minimum fitness function. We apply the modified tabu search algorithm [28], which is used in our optimization step to improve the quality of the obtained partitions, is based on the tabu search algorithm, in which the disruption mechanism is used to diversify the
Then, we randomly select a subset like $S_v$. In this case, we define two motion operators as follows:

- Two-move operator \( T_v \): We transfer the vertex \( v \) to bucket structure, which arranges the vertices according to the amount of gain move. This sorting is used to avoid unnecessary searching and reduces the time needed for finding vertices with a high gain move. Accordingly, we select the vertex that has the highest gain move from the bucket. After moving the vertex to the desired partition, we update the bucket structure. The complexity is removing a vertex and inserting it into bucket \( O(1) \). Thus, the complexity of moving vertex \( v \) from bucket \( S_v \) to \( S_m \) is equal to the number of adjacent vertices.

We explain the modified tabu search algorithm, and examine how the modified version of this algorithm is presented on the graph partitioning problem.

**D. MODIFIED TABU SEARCH ALGORITHM**

We define two motion operators \( T_1 \) and \( T_2 \). The application of these two motion operators causes to transfer the vertices to other subsets in order to reduce the connections. These operators play a crucial role in minimizing the sum of the connections of the edges between the partitions in the graph model. These operators move one vertex or two vertices between the subsets. Assuming that vertex \( v_m \) in partition \( S_m \), the gain move \( g(v, n) \) can be calculated when vertex \( v_m \) in partition \( S_m \) goes to any arbitrary \( S_n(n \neq m) \). The concept of gain move shows how motion operators cause to minimize the connection edges between partitions. Therefore, the high gain move decreases the connections in the graph model.

Suppose \( P = \{s_1, s_2, \ldots, s_k\} \) is a K-partition. For example, in Fig.7, \( \{v_1, v_6, v_8\} \) are considered in partition \( s_1 \), \( \{v_3, v_5, v_7\} \) are in partition \( s_2 \), and \( \{v_2, v_4, v_9, v_{10}\} \) are in partition \( s_3 \).

Additionally, \( S_{\text{max}} \) is the subset with the most edges so that:

\[
S_{\text{max}} = \max_{i=1,2,\ldots,k} W(s_i) \quad (11)
\]

In this case, we define two motion operators as follows:

- Single-move operator \( (T_1) \): We transfer a vertex like \( v_m \), which is the maximum gain move. We randomly select a partition like \( S_m(m \neq \text{max}) \), then the vertex like \( v_m \), which is in partition \( S_c \in \{S_i \in P|W(S_i) > W(S_m)\} \) is transferred to \( S_m \).

- Two-move operator \( (T_2) \): We select the two vertices \( v_x \) and \( v_y \), which have the highest gain move to transfer. We move vertex \( v_x \) to partition \( S_m \) based on the single-move operator. Then, we randomly select a subset like \( S_n \in P(n \neq \text{max}, n \neq m) \). Thus, we select the vertex \( v_y \) that has the highest gain move from the subset \( S_c \in \{S_i \in P|S_i \neq S_m, S_i \neq S_n\} \). Finally, we transfer \( v_x \) to \( S_m \) and \( v_y \) to \( S_n \).

To improve the assessment of the motion, we use the bucket structure, which arranges the vertices according to the amount of gain move. This sorting is used to avoid unnecessary searching and reduces the time needed for finding vertices with a high gain move. Accordingly, we select the vertex that has the highest gain move from the bucket. After moving the vertex to the desired partition, we update the bucket structure. The complexity is removing a vertex and inserting it into bucket \( O(1) \). Thus, the complexity of moving vertex \( v \) from bucket \( S_c \) to \( S_m \) is equal to the number of adjacent vertices.

Suppose \( V_{sel} \) is a set of candidate vertices with the highest gain to be transferred to subset \( S_m \). We select a vertex like \( v \in V_{sel} \), which is the current subset of \( S_c \), to transfer to \( S_m \) whenever it is not in the tabu list or the move from \( v \) to \( S_m \) causes the new partition to be better than the partition achieved thus far. If there is more than one vertex to move, we select the vertex to move to \( S_m \), which has the least connection with \( S_c \).

To improve and complete the modified tabu search algorithm, we use the disruption mechanism for diversity in the search space. In this case, if the best partition \( P \) does not change after \( T \) repetitions, the disruption mechanism selects a vertex randomly. To be more precise, suppose that the current partition \( \{s_1, s_2, \ldots, s_k\} \). We use an independent step as follows:

**Algorithm: Disruption mechanism**

1. Select randomly \( S_m \in \{S_1, S_2, \ldots, S_k\} \)
2. Select randomly \( v \) from \( S_c \in \{S|W(S) > W(S_m)\} \)
3. Transfer \( v \) to \( S_m \)
4. Do steps 1 to 3 for \( T \) repetition

This mechanism is similar to \( (T_1) \). However, there is a significant difference, and this vertex, which is considered for partition \( S_m \), is not necessarily adjacent to \( S_m \). This causes us to go to a different and more diverse space in the process of finding the solution.

As Fig.11 depicts, to illustrate the functionality of MTS, for example, we use \( 4_{mod5} \) quantum circuit, which has 7-inputs and outputs. Fig.12 presents the graph model of \( 4_{mod5} \). The graph is partitioned into two parts by the modified tabu search partitioning algorithm, as follows: \( P_1 \{q_1, q_4\} \), \( P_2 \{q_2, q_3, q_5, q_6, q_7\} \). Fig.13 shows the \( 4_{mod5} \) distributed quantum circuit, and Table 2 presents the steps of executing quantum gates for the \( 4_{mod5} \) circuit. Therefore, \( 4_{mod5} \) is divided into two partitions with specified qubits.

**V. THE PROPOSED APPROACH**

We attempted to reorder the qubit placement in the initial quantum circuit to construct a new quantum circuit to improve the execution time and modeled the new quantum circuit into a graph, and finally partitioned the graph model.
TABLE 2. The steps of executing quantum gates in the MTS algorithm for the 4 mod 5 circuit

| # of gates | type of gate |
|------------|--------------|
| g₁         | global       |
| g₂         | local        |
| g₃         | global       |
| g₄         | global       |
| g₅         | global       |
| g₆         | local        |
| g₇         | local        |

FIGURE 11. 4 mod 5 quantum circuit

FIGURE 12. Graph model of the 4 mod 5 circuit

FIGURE 13. 4 mod 5 distributed quantum circuit

by the modified tabu search algorithm as local optimization, and then applied the genetic algorithm as global optimization to obtain a distributed quantum circuit with at least the number of teleportation cost. We implement the proposed approach in benchmark circuits by applying steps (reordering-graph modeling-modified tabu search algorithm-genetic algorithm).

Teleportation operation in DQCs is costly; therefore, we attempt to reduce it as far as possible. Our approach is based on reducing the number of teleportation cost and execution time in DQC. Hence, we model the quantum circuit with a graph and assume that the quantum circuit is composed of the quantum gates \{H, T, T^2\}+CNOT. It is also noteworthy that quantum gates with multiple qubits can be decomposed to the basic gates [29], [30]. To construct the graph model, we begin from the first quantum gate in the quantum circuit until the set of edges is completed. The weight of each edge is related to the number of connection between vertices. We model the quantum circuit as an undirected graph in which the weight of each edge represents the connection of two qubits through quantum gates.

Since connections between partitions in a distributed quantum circuit indicate the number of teleportation cost, one of the main goals is to reduce the connections between partitions by applying the modified tabu search algorithm and genetic algorithm. We can divide the graph model into K-partitions using the modified tabu search algorithm for graph partitioning, and then apply the genetic algorithm to improve the graph partitioning in order to obtain the minimum number of teleportation cost. It is important to note that the best partitioning of vertices attained in the modified tabu search algorithm is as an initial partitioning in the genetic algorithm. We implement these partitioning algorithms for benchmark circuits and finally present the best number of teleportation cost. After applying the modified tabu search to the graph model, we attain the partition of vertices so that we have K-partitions \{p₁, p₂, \ldots, pₖ\} in which each partition has a number of vertices, which may be connected to several other partitions through some edges. Then, the results of the modified tabu search algorithm transfer to the genetic algorithm to run the global optimization. Since our proposed approach attempts to partition a graph to obtain the least number of teleportation cost after applying the partitioning algorithms. All of the connected edges in different partitions determine the number of teleportation cost in DQC. Once the proposed approach is executed, DQC is achieved by reducing the number of teleportation cost. If the vertices are properly placed in partitions, the execution time will be reduced, and the least teleportation cost will be obtained for the distributed quantum circuit. The algorithm for the proposed approach is
Our proposed method changes the order of the qubits and causes quantum qubits to move closer to each other to run as
- First, the quantum circuit is reordered based on the proposed method in Section IV-A.
- The graph model is constructed according to the proposed approach in Section IV-B.
- The graph is partitioned by the modified tabu search and genetic algorithm into K-partitions and a partitioned graph model is created.
- If the partitioning result is obtained by the modified tabu search algorithm is better than the genetic algorithm, this result is regarded as the best partitioning. Otherwise, the partitioning obtained by the genetic algorithm will be the best result.
- In this paper, genetic algorithm terminates when the number of generations reaches the maximum value, and MTS terminates when the number of iterations reaches the maximum size.

As Fig.14 shows, we reordered the qubits placement in the initial quantum circuit, and then constructed a graph model from the new quantum circuit. Finally, partitioning algorithms were implemented by the modified tabu search and the genetic algorithm to create a DQC.

**VI. RESULTS**

The proposed approach leads to attain the best results for the most distributed quantum circuits according to the results of implementation. For many distributed quantum circuits, the algorithms of the proposed approach are better than the results of other algorithms. Therefore, the results illustrate that the proposed approach can obtain the minimum number of teleportation and reduce the execution time in the distributed quantum circuit effectively. Some of the results are new for benchmark circuits, which were not found previously. As the number of connections between the partitions of the DQC increases, the execution time increases exponentially. For this reason, we reorder the qubits placement to improve the execution time in DQC. The obtained teleportation cost of the proposed approach was also compared to that of other approaches, and it appeared that teleportation cost and execution time were lower than those of other approaches.

As Fig.15 depicts, our proposed method causes to execute distributed quantum circuits with less execution time compared to [24]. Therefore, the results prove that our proposed approach has superior performance and high efficiency for DQCs. The reasons are as follows: The proposed approach is the combination algorithm of MTS and GA. It combines the global search by using GA and local search by employing MTS to perform the proposed algorithms. The genetic algorithm has the ability for global searching, and the modified tabu search has the ability for local searching. Furthermore, genetic operators in GA and the motion structure in MTS are applied and implemented in the proposed approach. To indicate the performance and effectiveness of the proposed approach, we used RevLib circuits, Binary welded tree (BWT), and Ground state estimation (GSE) as benchmark circuits. Afterward, we applied our proposed method to benchmark circuits in order to determine teleportation cost. The partitioning of graph is applied, so that the least number of communications between partitions is required, and this causes to reduce the teleportation cost in distributed quantum circuits as shown in Fig.16. We implemented our proposed method to benchmark circuits for $K = 2$ and 3 partitions, respectively. The number of teleportation is related to the placement of gates in the partitions. Compared to the proposed approach in [24], our proposed method causes to execute distributed quantum circuits with less execution time.

As Fig.14 shows, we reordered the qubits placement in the initial quantum circuit, and then constructed a graph model from the new quantum circuit. Finally, partitioning algorithms were implemented by the modified tabu search and the genetic algorithm to create a DQC.

**VI. RESULTS**

The proposed approach leads to attain the best results for the most distributed quantum circuits according to the results of implementation. For many distributed quantum circuits, the algorithms of the proposed approach are better than the results of other algorithms. Therefore, the results illustrate that the proposed approach can obtain the minimum number of teleportation and reduce the execution time in the distributed quantum circuit effectively. Some of the results are new for benchmark circuits, which were not found previously. As the number of connections between the partitions of the DQC increases, the execution time increases exponentially. For this reason, we reorder the qubits placement to improve the execution time in DQC. The obtained teleportation cost of the proposed approach was also compared to that of other approaches, and it appeared that teleportation cost and execution time were lower than those of other approaches.

As Fig.15 depicts, our proposed method causes to execute distributed quantum circuits with less execution time compared to [24]. Therefore, the results prove that our proposed approach has superior performance and high efficiency for DQCs. The reasons are as follows: The proposed approach is the combination algorithm of MTS and GA. It combines the global search by using GA and local search by employing MTS to perform the proposed algorithms. The genetic algorithm has the ability for global searching, and the modified tabu search has the ability for local searching. Furthermore, genetic operators in GA and the motion structure in MTS are applied and implemented in the proposed approach. To indicate the performance and effectiveness of the proposed approach, we used RevLib circuits, Binary welded tree (BWT), and Ground state estimation (GSE) as benchmark circuits. Afterward, we applied our proposed method to benchmark circuits in order to determine teleportation cost. The partitioning of graph is applied, so that the least number of communications between partitions is required, and this causes to reduce the teleportation cost in distributed quantum circuits as shown in Fig.16. We implemented our proposed method to benchmark circuits for $K = 2$ and 3 partitions, respectively. The number of teleportation is related to the placement of gates in the partitions. Compared to the proposed approach in [24], our proposed method causes to execute distributed quantum circuits with less execution time.

Our proposed method changes the order of the qubits and causes quantum qubits to move closer to each other to run on quantum gates, thereby reducing the execution time of the partitioning algorithm. For each quantum circuit, we used a genetic algorithm and the modified tabu search to partition the graph model, so that the minimum number of teleportation costs was obtained. We performed the proposed approach over different quantum circuits to show the execution time of our proposed approach. The results indicate that the proposed approach can reduce the execution time compared to [24] by approximately 80% that is a considerable improvement. The execution time of the method presented in [24] for quantum circuit aluPrimitive and sym9_147 is $1182.51$ and $10930.47$ seconds, respectively. However, the proposed algorithm can considerably reduce the execution time needed for these distributed quantum circuits. By determining the various number of partitions for the benchmark quantum circuits, the proposed approach attained an average of 50% improvement in the number of teleportation compared to [26] and [27].

**FIGURE 14.** The flowchart of the proposed approach
Table 4 displays the execution time of proposed approach compared to [23], [24] and other quantum circuits. Table 5 compares the results of our proposed approach to those in [26], [27] and the RevLib website in terms of teleportation cost (TC) for 21 different quantum circuits. Thus, we proposed the novel approach to find the number of teleportation required for the communication between subsystems. Table 6 presents the results of the comparison of our proposed approach to [22] in terms of teleportation cost. Fig.17 depicts TC for various numbers of partitions (K) in comparison to [22] for two quantum circuits: BWT and GSE circuits. Fig.18 also shows the improved teleportation cost of the proposed method in comparison to the proposed methods in [22] and [27]. Table 3 presents the parameters for the proposed approach.

International Business Machines (IBM) has made the quantum computers which is available for researchers [31]. IBM Quantum (IBM Q) provided the quantum computing services based on the IBM Quantum Composer and the IBM Quantum Lab which builds a platform to create the quantum circuits and quantum models. Also, the configurations with several quantum gates are available for running the test circuits on an IBM Q computer. However, each configuration can be implemented in a quantum computer including the single-qubit and CNOT gates. For the quantum circuit in [23], by running IBM quantum computer on the distributed quantum circuit constructed by proposed approach in [23] and the distributed quantum circuit generated by our proposed approach, we verified that our proposed approach significantly minimized the number of teleportations based on an equal state vector obtained for this quantum circuit as shown in Fig.19. To demonstrate the crucial aspects of each quantum circuit, we developed certain quantum circuits in the quantum composer and run them on the IBM Q platform. In the IBM Q platform, transpiling time (TPT) and transpiled quantum circuit help us to assess the findings. TPT is the time which it takes for quantum circuit to be translated into a circuit that can be run on backend. This process includes converting quantum gates into standard basis gates and optimizes the quantum circuit in own method as condensing gates in the platform IBM Q. The time it takes to convert gates into basis gates is mostly independent of the backend system, and it relies on the number and complexity of quantum gates in the original quantum circuit. The IBM Q takes longer to make quantum circuits compatible and better efficient for the backend because of the lengthier transpiling time. Execution time (EXT) is the time which it takes to generate the distributed quantum circuit in our proposed approach based on transpiled quantum circuit. Regarding the transpiled quantum circuit, teleportation cost (TPC) is the number of teleportations that is obtained by our proposed approach. Fig.20 shows the transpiled quantum circuit for the quantum circuit in [23]. Table 7 shows the obtained status timeline and the number of teleportation cost to generate the distributed quantum circuits due to the quantum circuits, which are transpiled on the IBM Q.

Table 3. The parameters in the proposed approach

| Parameter                              | Value  |
|----------------------------------------|--------|
| The size of population                 | 400    |
| The number of generations              | 200    |
| Replacement probability, pr            | 0.01   |
| Crossover probability, pc              | 0.7    |
| Mutation probability, pm               | 0.2    |
| The maximum iteration of MTS           | 1000   |

VII. CONCLUSION

We introduced a new method to optimize the number of teleportation cost and to improve the execution time in a distributed quantum circuit. As we proved in the results, the proposed approach reduced the number of teleportation and execution time. We also implemented our optimization approach in different quantum circuits. We used the RevLib website as benchmark circuits [10]. Furthermore, all implementations were performed on an Intel Celeron Dual Core 3 GHz with 2 GB of main memory. We employed the genetic algorithm and modified tabu search for a graph model in which reduction of teleportation cost was possible. Moreover, we perform four steps in our proposed approach to construct a DQC. Finally, we demonstrated that the proposed approach could be easily extended to upgrade the performance of DQCs.

The contributions of this paper include:

- Reordering of the qubits placement can cause to change qubits placement in QC, leading to reduction of the execution time when partitioning the algorithm is implemented.
Davood Dadkhah. et al. Preparation of Papers for IEEE Access

### TABLE 4. Execution time comparison of the proposed approach (P) with the results in [23] and [24]

| circuits      | # of qubits | # of partitions | ET [23] (S) | ET [24] (S) | ET (P) (S) |
|--------------|-------------|-----------------|-------------|-------------|-------------|
| sym9_147     | 12          | 2               | N.A         | 10930.47    | 24.55       |
| flip-flop    | 8           | 3               | N.A         | N.A         | 18.21       |
| aluPrimitive | 6           | 2               | 154020      | 1182.51     | 24.48       |
| 4gt5_75      | 5           | 2               | N.A         | N.A         | 12.20       |
| 4gt5_76      | 5           | 2               | 275         | 37.74       |             |
| 4mod7        | 5           | 2               | N.A         | 292.26      | 15.97       |
| cm42a_207    | 14          | 2               | N.A         | N.A         | 40.54       |
| 4gt4-c0_79   | 6           | 2               | N.A         | N.A         | 38.50       |
| ham13_107    | 15          | 2               | N.A         | N.A         | 29.86       |
| 4_49_17      | 15          | 2               | N.A         | N.A         | 12.76       |
| mod5d1_63    | 15          | 2               | N.A         | N.A         | 15.40       |

![Execution Time Graph](chart.jpg)

**FIGURE 15.** Improvement in the execution time of the proposed approach (P) compared to [24]

### TABLE 5. Teleportation cost comparison of the proposed approach (P) with the results in [26] and [27]

| circuits      | # of qubits | # of gates | # of partitions | TC [26] | TC [27] | TC (P) |
|--------------|-------------|------------|-----------------|---------|---------|--------|
| sym9_147     | 12          | 28         | 2               | 28      | 8       | 2      |
| flip-flop    | 8           | 17         | 3               | 8       | 8       | 4      |
| aluPrimitive | 6           | 13         | 2               | 12      | 6       | 4      |
| 4gt5_75      | 5           | 5          | 2               | N.A     | N.A     | 6      |
| 4gt5_76      | 5           | 16         | 2               | N.A     | N.A     | 8      |
| 4mod7        | 5           | 11         | 2               | 10      | N.A     | 4      |
| cm42a_207    | 14          | 20         | 2               | N.A     | N.A     | 2      |
| 4gt4-c0_79   | 6           | 9          | 2               | N.A     | N.A     | 6      |
| 4_49_16      | 4           | 16         | 2               | N.A     | N.A     | 8      |
| 4_49_17      | 4           | 12         | 2               | N.A     | N.A     | 8      |
| alu_200      | 18          | 94         | 2               | N.A     | N.A     | 12     |
| clip_206     | 14          | 174        | 2               | N.A     | N.A     | 4      |
| ham15_107    | 15          | 132        | 3               | N.A     | N.A     | 44     |
| ham15_108    | 15          | 70         | 3               | N.A     | N.A     | 42     |
| mod5d1_63    | 5           | 7          | 2               | N.A     | N.A     | 6      |
| z4ml_269     | 11          | 48         | 2               | N.A     | N.A     | 10     |
| root_255     | 13          | 99         | 2               | N.A     | N.A     | 6      |
| radd_250     | 13          | 48         | 2               | N.A     | N.A     | 10     |
| pm1_249      | 14          | 35         | 2               | N.A     | N.A     | 4      |
| cycle10_22   | 12          | 19         | 2               | N.A     | N.A     | 4      |
| 24_208       | 11          | 48         | 2               | N.A     | N.A     | 10     |

N.A means the author did not provide the results.

- A combination of MTS and GA has been proposed to generate a distributed quantum circuit with the least number of teleportation cost.

The results indicate that the proposed approach has been
TABLE 6. The number of teleportation cost for $K = 2$ and $3$ for GSE and BWT circuits compared to [22]

| circuits | # of partitions | TC [22] | TC (P) |
|----------|-----------------|---------|--------|
| GSE      | 2               | 20.42   | 8      |
| GSE      | 3               | 48      | 16     |
| BWT      | 2               | 18      | 2      |
| BWT      | 3               | 22      | 4      |

TABLE 7. The status timeline and the number of teleportation cost for the transpiled quantum circuits

| circuits   | TPT(mS) | EXT(S) | TPC |
|------------|---------|--------|-----|
| mod5d1     | 745     | 20.42  | 6   |
| 4_49_16    | 920     | 10.65  | 4   |
| Figure of [23]| 790     | 10.04  | 12  |

FIGURE 16. Improvement in the teleportation cost of the proposed approach (P) compared to [26] and [27]

successfully used to generate DQCs.

REFERENCES

[1] M. A. Nielsen and I. Chuang, "Quantum computation and quantum information," ed: American Association of Physics Teachers, 2002.
[2] L. K. Grover, "A fast quantum mechanical algorithm for database search," in Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, 1996, pp. 212-219.
[3] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM review, vol. 41, no. 2, pp. 303-332, 1999.
[4] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences, vol. 439, no. 1907, pp. 553-558, 1992.
[5] L. K. Grover, "A fast quantum mechanical algorithm for database search," arXiv preprint quant-ph/9605043, 1996.
[6] C. H. Bennett and S. J. Wiesner, "Communication via one-and two-particle operators on Einstein-Podolsky-Rosen states," Physical review letters, vol. 69, no. 20, p. 2881, 1992.
[7] C. H. Bennett et al., "Reduction of quantum entropy by reversible extraction of classical information," Journal of Modern Optics, vol. 41, no. 12, pp. 2307-2314, 1994.
[8] A. Peres, "Quantum cryptography with orthogonal states?," arXiv preprint quant-ph/9503003, 1995.
[9] J. Stolze and D. Suter, "Quantum computing," Wiley-VCH, Weinheim, vol. 29, pp. 30-31, 2004.
[10] R. Wille, D. Grolle, L. Teubner, G. W. Dueck, and R. Drechsler, "RevLib: An online resource for reversible functions and reversible circuits," in 38th International Symposium on Multiple Valued Logic (ismvl 2008), 2008: IEEE, pp. 220-225.
[11] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik, "Simulation of electronic structure Hamiltonians using quantum computers," Molecular Physics, vol. 109, no. 5, pp. 735-750, 2011.
[12] M. Ghosh, A. Chakrabarti, and N. K. Jha, "Automated quantum circuit synthesis and cost estimation for the binary welded tree oracle," ACM Journal on Emerging Technologies in Computing Systems (JETC), vol. 13, no. 4, pp. 1-14, 2017.
[13] M. A. Nielsen and I. L. Chuang, "Quantum computation and quantum information," Phys. Today, vol. 54, pp. 60-62, 2001.
[14] M. Nakahara and T. Ohmi, Quantum computing: from linear algebra to physical realizations. CRC press, 2008.
[15] L. K. Grover, "Quantum teleportation," arXiv preprint quant-ph/9704012, 1997.
[16] R. Cleve and H. Buhrman, "Substituting quantum entanglement for communication," Physical Review A, vol. 56, no. 2, p. 1201, 1997.
[17] J. Cirac, A. Ekert, S. Huelga, and C. Macchiavello, "Distributed quantum computation over noisy channels," Physical Review A, vol. 59, no. 6, p. 4249, 1999.
[18] R. Beals et al., "Efficient distributed quantum computing," Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 469, no. 2153, p. 20120686, 2013.
[19] J. Yepez, "Type-II quantum computers," International Journal of Modern Physics C, vol. 12, no. 09, pp. 1273-1284, 2001.
[20] A. Streltsov, H. Kampermann, and D. Bruti, "Quantum cost for sending entanglement," Physical review letters, vol. 108, no. 25, p. 250501, 2012.
[21] M. Ying and Y. Feng, "An algebraic language for distributed quantum computing," IEEE Transactions on Computers, vol. 58, no. 6, pp. 728-743, 2009.
[22] P. Andrés-Martínez and C. Heunen, "Automated distribution of quantum circuits by bipartite graphs," Quantum Information Processing, vol. 19, no. 12, pp. 3804-3820, 2020.
[23] M. Zomorodi-Moghadam, M. Houshmand, and M. Houshmand, "Optimizing teleportation cost in distributed quantum circuits," International Journal on Emerging Technologies in Computing Systems (JETC), vol. 10, no. 4, pp. 1-14, 2017.
[24] A. Yimsrirawattana and S. J. Lomonaco Jr, "Distributed quantum computing: A distributed Shor algorithm," in Quantum Information and Computation II, 2004, vol. 5436: International Society for Optics and Photonics, pp. 360-372.
[25] O. Daei, K. Navi, and M. Zomorodi-Moghadam, "Optimized Quantum Circuit Partitioning," International Journal of Theoretical Physics, vol. 59, no. 12, pp. 3804-3820, 2020.
[26] Z. Duvarzani, M. Zomorodi-Moghadam, M. Houshmand, and M. Nouri-baygi, "A dynamic programming approach for distributing quantum circuits by bipartite graphs," Quantum Information Processing, vol. 19, no. 10, pp. 1-18, 2020.
[27] D. Maslov, C. Young, D. M. Miller, and G. W. Dueck, "Quantum circuit simplification using templates," in Design, Automation and Test in Europe, 2005: IEEE, pp. 1208-1213.
FIGURE 17. Number of the teleportation cost for GSE and BWT circuits in comparison to [22]

FIGURE 18. Percentage of the improved teleportation cost of the proposed approach compared to [22] and [27]

FIGURE 19. State vector for the quantum circuit in [23]

FIGURE 20. The representation of the transpiled quantum circuit for the quantum circuit in [23]

[30] D. Maslov, G. W. Dueck, and D. M. Miller, “Toffoli network synthesis with templates,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 24, no. 6, pp. 807-817, 2005.

[31] A. Cross, ”The IBM Q experience and QISKit open-source quantum computing software,” Bulletin of the American Physical Society, vol. 63, 2018.

FIRST A. AUTHOR Davood Dadkhah received the B.S. degree in electrical engineering from Imam Reza University, Mashhad, Iran, in 2012 and the M.S. degree in electrical engineering from Ferdowsi University of Mashhad, Mashhad, Iran, in 2014. He is currently pur-
suing the Ph.D. degree electrical engineering at Ferdowsi University of Mashhad, Mashhad, Iran.

SECOND B. AUTHOR Mariam Zomorodi received the B.Sc. degree in computer engineering from Iran University of Science and Technology, in 2002, the M.Sc. degree in computer engineering from Amirkabir University of Technology (Tehran Polytechnic), in 2007, and the Ph.D. degree in computer engineering from Shahid Beheshti University, in 2014. She is currently with the Department of Information and Communications Technology, Cracow University of Technology, and Ferdowsi University of Mashhad. Her current research interests include quantum computing, evolutionary computation, and machine learning.

THIRD C. AUTHOR Seyed Ebrahim Hosseini is currently a full Professor of electrical engineering at Ferdowsi University of Mashhad, Iran. His research interests include quantum information and Semiconductor Science and Technology.

FOURTH D. AUTHOR Pawel Pławiak, B.Eng., M.Sc., Ph.D., D.Sc., was born in Ostrowiec, Poland, in 1984. He obtained his B.Eng. and M.Sc. degree in Electronic and Telecommunications, his Ph.D. degree with honors in Biocybernetics and Biomedical Engineering at the AGH University of Science and Technology, Cracow, Poland, and his D.Sc. degree in Technical Computer Science and Telecommunications at Silesian Technical University, Gliwice, Poland in 2021, 2016 and 2020, respectively. He is the Dean of the faculty of Computer Science and Telecommunications and Associate Professor in Cracow University of Technology, Cracow, and Deputy Director for research in National Institute of Telecommunications, Warsaw, and Associate Professor in Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Gliwice Poland. He has published more than 30 papers in refereed international SCI-IF journals. He is the editor of the PLoS ONE journal and a reviewer of many prestigious and reputed journals. His research interests include machine learning and computational intelligence (e.g., artificial neural networks, genetic algorithm, fuzzy systems, support vector machines, k-nearest neighbors, and hybrid systems), ensemble learning, deep learning, evolutionary computation, classification, pattern recognition, signal processing and analysis, data analysis and data mining, sensor techniques, medicine, Biocybernetics, biomedical engineering and telecommunications.

FIFTH F. AUTHOR Dr. Xujuan Zhou is Associate Professor in University of Southern Queensland (USQ), Australia. She received her PhD from Queensland University of Technology, Australia in 2009. Her research interests include machine learning and Deep learning, Health informatics, Information retrieval/filtering, Recommender systems, Sentiment analysis, Natural language processing. In 2021, she received a three-year (2022-2024) Advance Queensland Industry Research Fellowship grant for the project titled “Early detection of chronic health conditions using AI prediction model”. She has completed the Commonwealth Innovation Connections Grant research project titled “Prediction of Cancer Recurrence Using Innovative Machine Learning Approaches” (2017 – 2019). She published papers in prestige Journals (e.g., Artificial Intelligence Review, Decision Support Systems, Pattern Recognition Letter, Artificial Intelligence in Medicine, Journal of Medical Intent Research, Knowledge-Based Systems, IEEE Access, etc.). She is an editorial board member for Web Intelligence – an international journal, a reviewer for many journals and conferences related to her research fields.

* * *

This article has been accepted for publication in IEEE Access. This is the author’s version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3186485

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/