A Heuristic Qubit Placement Strategy for Nearest Neighbor Realization in 2D Architecture

Anirban Bhattacharjee, Chandan Bandyopadhyay(✉),
Laxmidhar Biswal, and Hafizur Rahaman

Indian Institute of Engineering Science and Technology, Shibpur,
Howrah 711103, West Bengal, India
anirbanbhattacharjee330@gmail.com,
chandanb.iiest@gmail.com, laxmidhar.cvrce@gmail.com,
rahaman_h@yahoo.co.in

Abstract. Recently, quantum computing has received massive attention of the researchers due to the advantages it offers in solving some problems efficiently compared to conventional computing. But there are several design challenges that need to be satisfied for various quantum technologies to perform reliable quantum operation. One such essential requirement demands to maintain the neighborhood organization of the operating qubits, referred to as the nearest neighbor (NN) constraint. This can be settled through insertion of SWAP gates which helps to synthesize a NN-compliant design by exchanging the positions of the qubits. Consequently, the cost overhead of the circuit enhances due to SWAP gate insertion as the number of gates in the circuit increases, thereby exploitation of various approaches to address this issue has turned out to be essential of late.

In this regard, here we have introduced an improved heuristic design approach for the synthesis of a two dimensional NN-compliant circuit with reduced cost overhead. The approach has been executed in three phases of qubit selection, qubit placement and SWAP gate insertion. Initially, the qubits have been chosen for placement based on some cost metric estimation and then organized them on the grid locations in a specific order in the second phase. Lastly, SWAP gate insertion phase has been exercised to make qubits adjacent. It has been observed that our method has performed comparatively better than the previous works after carrying out experimental evaluations over a wide range of benchmark specifications.

Keywords: Quantum circuit · Quantum gates · Nearest neighbor · SWAP gate · 2D architecture

1 Introduction

Quantum computing has made a tremendous landmark in the computing paradigm because of its capability of solving some specific problems in polynomial time. Some of the applications that have turned out to be beneficial for quantum computation are

S. Rajaram et al. (Eds.): VDAT 2018, CCIS 892, pp. 593–605, 2019.
https://doi.org/10.1007/978-981-13-5950-7_49

factorization in RSA cryptosystem [1], database search [2], discrete logarithm etc. over conventional computation. Subsequently, a massive interest has been generated among the researchers for the development of a small scale quantum computer in the last decade. In this conjecture, design of quantum algorithms for the practical realization of quantum computer in terms of quantum circuits has turned out to be significant. Quantum circuits manipulate quantum information represented in the form of qubits than as bits used in the conventional computing paradigm. Qubits are similar to bits except that it can also exist in multiple states that can be considered as the linear combination of the basis states. In quantum circuit, a sequence of quantum gates basically operates on the qubits and subsequently transforms it into a different state during quantum information processing. It has been witnessed through experimental observations that the qubits are susceptible to changes due to environmental disturbances or noises that might result in computational errors. In order to overcome this issue, the interactions between the interacting qubits should be made adjacent so as to ensure the computational accuracy. This has been considered as the necessary criteria for the synthesis of several quantum technologies like ion trap [3], nuclear magnetic resonance [4], quantum dots [5] and superconducting qubits [6]. In order to bring the interacting qubits close to each other, a standard way of applying SWAP gates in the original circuit has been followed to satisfy the design constraints for the quantum technologies. Basically, SWAP gate exchanges the states of the qubits rather than its magnitude to bring the non-adjacent qubits close to each other. Consequently, such an operation induces some circuit design issues in the form of increase in circuit depth and quantum cost of the resultant NN based circuit. As a result, optimizing the NN based design by using less number of SWAP gates has turned out an essential design challenge for the researchers. They have attempted to fulfil this design issue by exploiting several design methodologies as discussed next.

Several design strategies have been suggested for linear arrangement of qubits in 1D format, where they can communicate with at most two adjacent neighbours. In [7], the authors mapped the NN based design problem into a task assignment one and applied harmony based search technique to solve the equivalent problem. To obtain a better design, a unique technique based on 1D arrangement of qubits has been introduced in [8], which emphasized on using limited number of SWAP gates. By mapping the qubits on a lattice structure not only reduces the SWAP gate count but also the technique becomes feasible for large benchmarks that have been undertaken in the work [9]. In [10], an NN based design has been optimized by constructing an interaction graph for the corresponding problem and then subsequently optimized it by employing a look-ahead policy. A graph-partitioning based approach has been presented in [11] to realize an NN based circuit with reduced cost overhead. An improved NN based design has been synthesized by employing an efficient conversion technique is discussed in [12]. As heuristic approach does not provide an optimal solution, so an exact approach based on SAT solvers has been implemented by *Robert et al.* [13] to synthesize an optimal NN based design but consumed substantial amount of time to process large benchmark circuits. A much improved design compared to the optimal one can be achieved by transforming the circuit representation from 1D format to 2D structure. As the number of interacting neighbours rises from 2 to 4, has resulted in a better cost effective design. Several works pertaining to 2D representation has been declared of late.

To realize an improved NN based design compared to the linear based designs, a mixed integer programming approach for mapping the qubits on a grid has been reported in [14]. *Lye et al.* [15] exercised an exact search strategy to derive an optimal solution for small sized benchmarks as it was not applicable for larger benchmarks due to extensive computational cost. In [16], the authors presented a meta-heuristic approach for the realization of an efficient NN-compliant design. To obtain an improved and scalable design, a heuristic qubit placement scheme has been employed for mapping the qubits on a grid location based on the degree of interactions has been summarized in the work [17]. To optimize the NN design further, another heuristic approach based on look-ahead policy has been introduced by *Robert et al.* [18], where SWAP gates have been reduced extensively. For better placement of qubits in the grid structure, a priority based ordering strategy has been followed by the authors of [19]. In addition to 2D design synthesis, mapping of circuits in 3D configuration has been demonstrated in one of the recent works [19]. Although representation of circuits in 3D platform provides a better cost-effective design than in 2D format as the number of adjacent neighbours of qubits escalates to 6 but at the same time controlling of such qubit representation becomes a cumbersome problem compared to 2D orientation of qubits.

In this work, we have presented a design synthesis approach for 2D representation of any quantum circuit. The primary intention of the design strategy is to obtain a cost effective NN based design by reducing the SWAP gate count. In this conjecture, our design policy has been executed in three phases for the realization of a 2D NN-complaint circuit. We have conducted a comprehensive experimental analysis over a number of different benchmarks and the resulting observations are recorded in the result tables, which suggest that our strategy has performed reasonably well than the reported 2D works regarding NN based optimization of 2D quantum circuits.

The remaining portion of the paper has been organized as follows. Section 2 discusses about the preliminaries of quantum computation and nearest neighbour criteria. Discussion of our design methodology and experimental evaluations has been presented in Sects. 3 and 4 respectively. Finally, we conclude the paper in Sect. 5.

2 Background

A qubit represents the basic unit of information acted upon by a set of elementary quantum gates constituting a quantum circuit. These quantum information units can be found in either one of the basis states of $|0\rangle$, $|1\rangle$ or in other states which can be expressed as the combination of the basis states by means of a state vector as

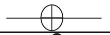
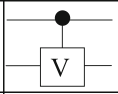
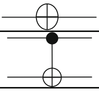
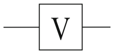
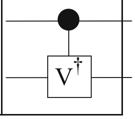
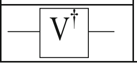
$$|\varepsilon\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

where α and β represents the probability amplitudes of the basis states $|0\rangle$ and $|1\rangle$ satisfying the condition $\alpha^2 + \beta^2 = 1$. To have a better interpretation about the concepts, we have introduced some fundamental terminologies related to quantum computing and nearest neighbor constraint.

Definition 1: *A quantum circuit consists of a set of elementary quantum gates that perform manipulation on qubits to implement a specific function.*

Generally, quantum gates from various quantum gate libraries are exercised to implement a quantum algorithm. Some of the most commonly used libraries are NCV, NCVW, Clifford +T group [20–23] for the logical representation of any specific function. However, in the present work we have restricted our investigation only on gate level representations from NCV library. Basically, the quantum gate operations can be represented as $2^n \times 2^n$ transformation matrices. These transformation matrices are all unitary in nature that operates on the states of the input qubits. Table 1 represents some of the basic quantum gates along with their corresponding symbolic representation as summarized below.

Table 1. Some basic quantum gates and their symbolic representations

Gates	**Notation**	**Gates**	**Notation**
NOT		Controlled -V	
CNOT			
V		Controlled -V†	
V†			

Despite, quantum circuits are used to realize the Boolean functions in their gate level representation but there are some physical constraints that need to be fulfilled for the realization of such circuits. One such major criterion is to accomplish the neighborhood arrangement of qubits to enable the so-called nearest neighbor property. This is considered as one of the essential design criteria for the synthesis of efficient quantum circuit representation. To address this design challenge, a solution of SWAP gate insertion (diagrammatically represented in Fig. 1) is being followed to attain the nearest neighbor property.

Definition 2: *The interaction distance between the positions of the control (c) and target (t) qubits of any 2-qubit gates (G) represents the nearest neighbor cost (NNC) and it can be expressed mathematically as $NNC_G = |c - t| + 1$, where NNC_G denotes the NNC of any 2-qubit gate.*

To determine the NNC for the entire circuit ($\mathrm{NNC_C}$), an overall sum for all the 2-qubit gates in the given circuit (C) is computed. A quantum gate is said to satisfy the nearest neighbor constraint if it is either a 1-qubit gate or for 2-qubit gates the NNC, $\mathrm{NNC_G} = 0$.

Definition 3: *SWAP gate is a 2-qubit gate that acts on qubits q_a, q_b and interchanges the states than values of these qubits. For instance it acts on qubits q_7, q_9 and transforms them from the state (q_1, q_2,…,q_7, q_8, q_9,…,q_{20}) to (q_1, q_2,…,q_9, q_8, q_7,…,q_{20}).*

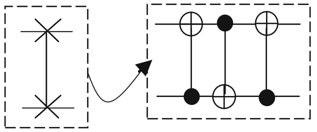


Fig. 1. SWAP gate representation

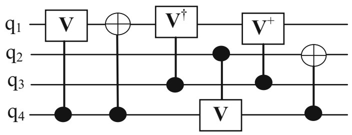


Fig. 2(a). Original circuit with NNC = 7

Example 1: *The circuit depicted in Fig. 2(a) does not satisfy the NN design criteria as the interaction distance of all the 2-qubit gates is having a positive numeral (interacting qubits are not placed adjacent). The corresponding NN based design of Fig. 2(a) has been obtained by adding SWAP gates before each of the non-adjacent gates as shown in Fig. 2(b). It is evident from Fig. 2(b) that 14 SWAP gates have been incorporated to accomplish the NN-compliant design of the circuit of Fig. 2(a).*

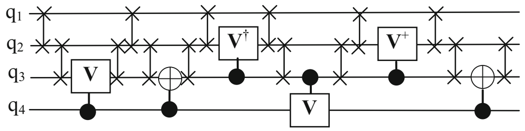
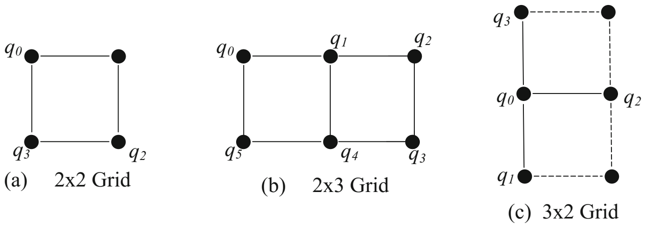


Fig. 2(b). NN based design of Fig. 2(a)

Though we have attained the equivalent NN design of the original circuit by means of SWAP insertion but still a room for improving the design further exists by projecting the entire circuit representation into 2D platform. Such higher dimensional representation facilitates to obtain a better design compared to 1D platform as less SWAP gates are required to derive the equivalent NN design of the given circuit.

A quantum circuit can be projected into a 2D format by organizing the qubits on a grid structure rather than organizing in a linear fashion. In way, the communication between the neighbors in such a representation enhances due to higher number of adjacent neighbors results in a lesser use of SWAP gates. Several possible 2D configurations exist in which the qubits can be mapped on the grid structure. Lets consider the circuit of Fig. 2(a) where the qubits are arranged linearly but several choices exists to represent the corresponding circuit in 2D format is represented in Fig. 3. Different configurations may influence the neighborhood organization of the qubits that in turn affects the nearest neighbor criteria.



(a) 2x2 Grid

(b) 2x3 Grid

(c) 3x2 Grid

Fig. 3. Possible grid configurations for qubit mapping in 2D layout

The circuit in Fig. 2(a) has been transformed into a 2D structure by positioning the qubits in a 2 × 2 grid structure as depicted in Fig. 4(a) and the corresponding NN based design is shown in Fig. 4(b). We observe that only a single SWAP gate is required to obtain the NN design in 2D platform compared to 14 SWAP gates needed for the same circuit in 1D representation.

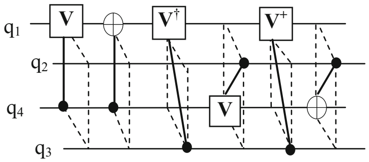


Fig. 4(a). Quantum circuit of Fig. 2 (a) is represented in 2D layout

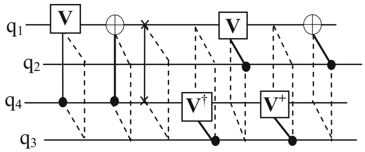


Fig. 4(b). 2D NN-compliant realization of Fig. 4(a)

3 Proposed Technique

Here, we have discussed about our proposed approach exercised for the purpose of designing of NN-compliant quantum circuit in 2D configuration. In order to attain the NN-compliant nature of any given circuit, the method initially arranges the qubits in the grid locations and then finally applies SWAP gates to bring any two non-neighboring interacting qubits operating over a quantum gate close to each other. In this context, we have implemented a qubit selection and placement strategies followed by SWAP gate insertion method to achieve the NN-compliant design of a given circuit are described next.

3.1 Qubit Selection Approach

In the present section, we have applied a heuristic technique that determines the sequence in which the qubits are preferred for placement in the grid location. In this regard, we have performed a number of calculations to derive the order in which the qubits are to be placed in a grid structure. Initially, we have estimated the time interval of interactions for each qubit by estimating the difference between the start and end time instants and recorded the same in a time span table (see Table 2) for the given circuit (see Fig. 5). From the time span table, we have computed the cost impact table (see Table 3) in which the cost factor of each qubit over their respective time intervals (obtained from the time span table) has been determined. In other words, the total nearest neighbor costs contributed by each qubit has been estimated. For instance, qubit q_1 interacts at the first, fourth, sixth and seventh time units of the given circuit (see Fig. 5) and contributes a cost of 2, 1, 1 and 2 (Total cost = 2 + 1 + 1 + 2 = 6) at the respective gate positions. Finally, the cost impact table has been sorted in a descending order based upon the cost factors and stored in a qubit order table (see Table 4) that determines the order in which the qubits are to be selected for placement in a 2D grid. It has been observed that the qubits q_1, q_4 have same cost factor but q_1 has been given higher priority for selection over q_4 (see Table 4) as q_1 has the least time interval compared to the one of q_4. Therefore, q_1 has been placed before q_4 in the qubit order table.

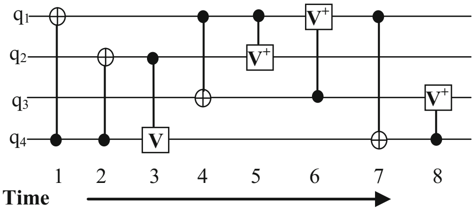


Fig. 5. Quantum circuit

Table 2. Time span table

Qubits	Start time	End time	Time interval
q_1	1	7	6
q_2	2	5	3
q_3	4	8	4
q_4	1	8	7

Table 3. Cost impact table

Qubits	Time interval	Cost factor
q_1	6	6
q_2	3	2
q_3	4	2
q_4	7	6

Table 4. Qubit order table

Qubits	Time interval	Cost factor
q_1	6	6
q_4	7	6
q_2	3	2
q_3	4	2

3.2 Qubit Placement Technique

In this phase, a qubit placement strategy has been executed after determining the order in which the qubits are to be arranged in a grid structure obtained in the previous phase. In way, the placement strategy organizes the qubits in a grid in the order in which they appear in the qubit order table. After obtaining the qubit order table, the placement strategy places the highest priority qubit in the centre of the grid. Then the rest of the qubits are selected from the order table and placed them around the location of the highest priority qubit by following the order viz. right, top, left, bottom depending upon the availability of space as explained in Algorithm 1. In our case, the qubit q_1 has been chosen for placement in a 2D grid as it is having the highest priority for selection compared to other qubits (see Table 4) and placed it in the centre of the grid (see Fig. 6 (a)). Then, the next qubit q_4 is chosen from Table 3 and placed it on the right of the location of q_1 since it is empty (see Fig. 6(b)). Now, we select the qubit appearing after q_4 viz. q_2 from the order table and checks whether the cell on the top of q_1 is available or not. Due to its availability, qubit q_2 has been allocated on the top of the location of q_1 (see Fig. 6(c)). Similarly, the last occurring qubit q_3 has been placed on the left of q_1 (see Fig. 6(d)).

After allocating all the qubits from the qubit order table on the grid locations, finally SWAP gates are incorporated to bring the positions of any two interacting qubits close to each other to enable the nearest neighborhood property for the given circuit is illustrated next.

	q_1	

Fig. 6(a). Placing qubit q_1 in the center of the grid

	q_1	q_4

Fig. 6(b). Placing qubit q_4 on the right of q_1

	q_2	
	q_1	q_4

Fig. 6(c). Placing qubit q_2 on the top of q_1

	q_2	
q_3	q_1	q_4

Fig. 6(d). Placing qubit q_3 on the left of q_1

Algorithm 1 describes about the process of arrangement of qubits in a 2D grid by selecting qubits from the qubit order table.

```
Algorithm 1: qubit placement strategy
Input: qubit order table, OT
Output: qubits arranged in a grid, $GD_{2D}$
begin
    for (($q_i$ € OT)) do
        if ($q_i$ is the highest priority qubit) then
            $q_{high\_order} = q_i$;
            row_centre = $GD_{2D}/2$;
            column_centre = $GD_{2D}/2$;
        else
            row_num = retrieve_row($q_{high\_order}$);
            column_num = retrieve_column($q_{high\_order}$);
            if ($GD_{2D}$(row_num, column_num+1) is empty) then
                place the next qubit in $GD_{2D}$(row_num, column_num+1);
            else if ($GD_{2D}$(row_num-1, column_num) is empty) then
                place the next qubit in $GD_{2D}$(row_num-1, column_num);
            else if ($GD_{2D}$(row_num, column_num-1) is empty) then
                place the next qubit in $GD_{2D}$(row_num, column_num-1);
            else if ($GD_{2D}$(row_num+1, column_num) is empty) then
                place the next qubit in $GD_{2D}$(row_num+1, column_num);
            else
                place the next qubit close to the location of the highest
                priority qubit $q_i$ in one of the empty locations;
            end if
        end if
    end for
    return $GD_{2D}$;
end
```

3.3 SWAP Gate Insertion

Here, SWAP gates are being utilized to obtain the NN-compliant design for the quantum circuit represented in Fig. 5.

For this purpose, we have examined each individual gate in the circuit to investigate whether its operating qubits are placed adjacent or not and subsequently inserted SWAP gates to alter the positions of the qubits of the corresponding non-adjacent gate in such a manner so that its interacting qubits approach towards each other and finally becomes adjacent. We have considered the grid obtained in the previous phase (see Fig. 6(d)) as input of this phase and applied SWAP gates wherever we find any gate whose qubits are not adjacent by traversing the circuit from the left. The entire process of SWAP gate insertion for the circuit of Fig. 5 is represented in Fig. 7.

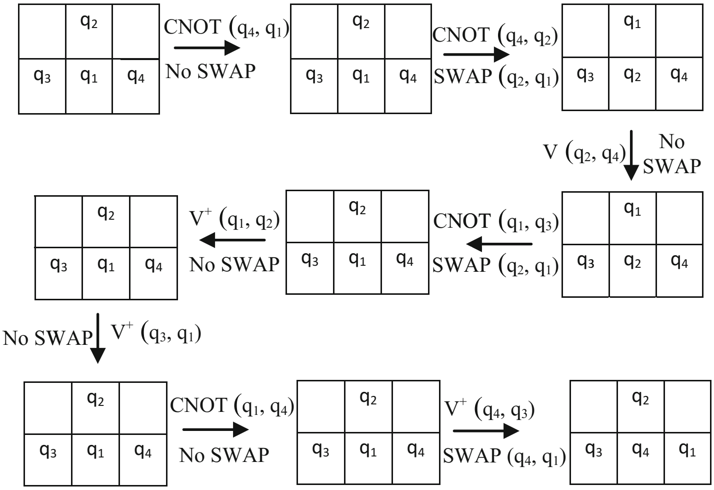


Fig. 7. SWAP gate insertion process

To understand the methodology more clearly, we have further illustrated the entire transformation process by considering a benchmark circuit as discussed next (Fig. 8(a), 8(b) and 8(c)).

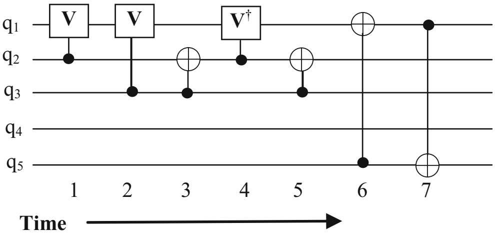


Fig. 8(a). Example benchmark (4gt11_84)

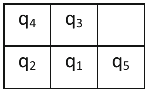


Fig. 8(b). Placement of qubits from qubit order table

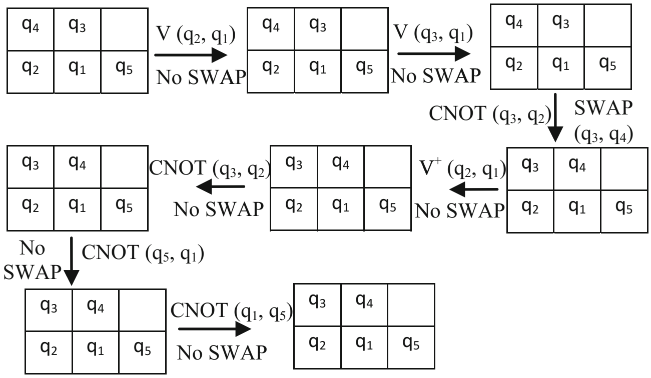


Fig. 8(c). SWAP gate insertion

Example 2: *Let's consider the NN-complaint transformation process of a benchmark circuit "4gt11_84" in this example. All the associated computational stages have also been summarized* (Tables 5, 6 and 7).

Table 5. Time span table

Qubits	Start time	End time	Time interval
q_1	1	7	6
q_2	1	5	4
q_3	2	5	3
q_4	-	-	-
q_5	6	7	1

Table 6. Cost impact table

Qubits	Time interval	Cost factor
q_1	6	7
q_2	4	0
q_3	3	1
q_4	-	-
q_5	1	6

Table 7. Qubit order table

Qubits	Time interval	Cost factor
q_1	6	7
q_5	1	6
q_3	3	1
q_2	4	0
q_4	-	-

4 Experimental Results

We have implemented the proposed method in C and executed on a system with configurations of intel5 processor, 3.30 GHz clock and 4 GB RAM. Experimental analysis has been carried out on various benchmark functions available on [24] as shown in Tables 8 and 9 respectively. Table 8 represents the experimental evaluations performed over small and medium size benchmarks (up to 16 qubits) while Table 9 represents the results carried out over large size benchmarks (more than 16 qubits).

Table 8. Improvements in SWAP gates over existing 2D works

Benchmarks	No. of qubits	Prop. work		Prev. work [17]	Joint scheme [18]	Iterative scheme [18]	Prev. work [14]	% Improv. over			
		Grid	SWAP count	SWAP count	SWAP count	SWAP count	SWAP count	[17]	[18]	[18]	[14]
4gt10-v1_81	5	2 × 3	12	15	15	22	16	20	**20***	**83.33***	25
4gt11_81	5	2 × 3	1	2	-	-	2	**50***	-	-	**50***
QFT7	7	3 × 3	13	14	13	22	18	7.14	0.0	40.90	27.77
QFT8	8	3 × 3	16	23	17	25	18	30.43	5.88	36	11.11
QFT9	9	3 × 3	24	36	22	27	34	33.33	−9.09	11.11	29.41
QFT10	10	4 × 3	36	51	37	43	53	29.41	2.70	16.27	32.07
aj-e11_165	5	3 × 3	14	22	16	37	24	36.36	12.5	62.16	41.66
3_17_13	3	2 × 2	5	3	5	8	6	−66.6	0.0	37.5	16.66
4gt5_75	5	3 × 3	8	10	-	-	8	20	-	-	0.0
mod5adder_128	6	3 × 3	31	36	33	45	41	13.88	6.06	31.11	24.39
mod8-10_177	6	3 × 3	34	43	-	-	45	20.93	-	-	24.44
hwb4_52	4	2 × 3	7	9	-	-	9	22.22	-	-	22.22
hwb5_55	5	3 × 3	30	49	37	64	45	38.77	18.91	53.12	33.33
hwb6_58	6	3 × 3	53	76	59	85	79	30.26	10.16	37.64	32.91

(*continued*)

Table 8. (*continued*)

Benchmarks	No. of qubits	Prop. work		Prev. work [17]	Joint scheme [18]	Iterative scheme [18]	Prev. work [14]	% Improv. over			
		Grid	SWAP count	SWAP count	SWAP count	SWAP count	SWAP count	[17]	[18]	[18]	[14]
alu-v4_36	5	2 × 3	9	11	-	-	10	18.18	-	-	10
4mod7-v0_95	5	3 × 3	10	14	-	-	13	28.57	-	-	23.07
rd53_135	7	3 × 3	26	40	30	47	39	35	13.33	44.68	33.33
ham7_104	7	3 × 3	32	45	37	53	48	28.88	13.51	39.62	33.33
rd73_140	10	4 × 3	31	43	-	-	37	27.90	-	-	16.21
rd84_142	15	5 × 3	56	62	-	-	54	9.67	-	-	−3.70
cnt3-5_180	16	3 × 6	63	84	-	-	69	25	-	-	8.69
Average improvement over SWAP gate count								*21.87*	*9.39*	*41.12*	*24.59*

Note: The values marked with * represents the improvement in best case over the related works.

Table 9. Results for large size benchmarks

Benchmarks	No. of qubits	Gate count	Grid	SWAP
ac_21_1	21	129	4 × 6	90
ac_21_2	21	67	6 × 4	52
ac_21_3	22	42	6 × 4	48
QFT_17	17	136	5 × 4	199
QFT_18`	18	153	5 × 4	238
QFT_19	19	171	5 × 4	258
hm_20	20	73	5 × 4	77
hm_21	21	79	4 × 6	70
hm_22	22	85	6 × 4	82

We have performed a detailed evaluation over SWAP gate count and compared the results with some of the related 2D works as represented in the result table. After analyzing the result table, we observe that our methodology organizes the qubits in an efficient manner such that less number of SWAP gates is used than the ones used in existing works.

5 Conclusion

Here, we have implemented a heuristic design strategy for arranging qubits in a 2D structure so as to obtain the NN based design of a given circuit. To this end, we have segmented the design strategy into three phases of qubit selection followed by qubit placement and lastly applying SWAP insertion technique. Such application of heuristic policy for determining a suitable qubit placement leads to faster execution. We have evaluated the SWAP gate count for different benchmark functions and witnessed significant improvements over the related works.

Though, we have achieved better results over the reported works, but these are far from optimal due to the application of heuristic based design policy thereby we would investigate better design policies so as to obtain optimal or near optimal results in the future.

References

1. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**(5), 1484–1509 (1997)
2. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Symposium on the Theory of Computing, pp. 212–219 (1996)
3. Kielpinski, D., Monroe, C., Wineland, D.J.: Architecture for a largescale ion-trap quantum computer. Nature **417**(6890), 709–711 (2002)
4. Criger, B., Passante, G., Park, D., Laflamme, R.: Recent advances in nuclear magnetic resonance quantum information processing. Philos. Trans. R. Soc. Lond. A: Math. Phys. Eng. Sci. **370**(1976), 4620–4635 (2012)
5. Taylor, J., Petta, J., Johnson, A., Yacoby, A., Marcus, C., Lukin, M.: Relaxation, dephasing, and quantum control of electron spins in double quantum dots. Phys. Rev. B **76**(3), 035315 (2007)
6. Blais, A., et al.: Quantum information processing with circuit quantum electrodynamics. Phys. Rev. A **75**(3), 032329 (2007)
7. Alfailakawi, M., Alterkawi, L., Ahmad, I., Hamdan, S.: Line ordering of reversible circuits for linear nearest neighbor realization. Quant. Info. Proc. **12**(10), 3319–3339 (2013)
8. Saeedi, M., Wille, R., Drechsler, R.: Synthesis of quantum circuits for linear nearest neighbor architectures. Quant. Info. Proc. **10**(3), 355–377 (2011)
9. Perkowski, M., Lukac, M., Shah, D., Kameyama, M.: Synthesis of quantum circuits in linear nearest neighbor model using positive Davio lattices (2011)
10. Shafaei, A., Saeedi, M., Pedram, M.: Optimization of quantum circuits for interaction distance in linear nearest neighbor architectures. In: Design Automation Conference (2013)
11. Chakrabarti, A., Sur-Kolay, S., Chaudhury, A.: Linear nearest neighbour synthesis of reversible circuits by graph partitioning. arXiv preprint arXiv:1112.0564 (2011)
12. Hirata, Y., Nakanishi, M., Yamashita, S., Nakashima, Y.: An efficient conversion of quantum circuits to a linear nearest neighbor architecture. Quantum Info. Comput. **11**(1), 142–166 (2011)
13. Wille, R., Lye, A., Drechsler, R.: Exact reordering of circuit lines for nearest neighbour quantum architectures. IEEE Trans. CAD **33**(12), 1818–1831 (2014)
14. Shafaei, A., Saeedi, M., Pedram, M.: Qubit placement to minimize communication overhead in 2D quantum architectures. In: Proceedings of ASP Design Automation Conference, pp. 495–500, January 2014
15. Lye, A., Wille, R., Drechsler, R.: Determining the minimal number of swap gates for multi-dimensional nearest neighbor quantum circuits. In: Proceedings of ASP Design Automation Conference, pp. 178–183, January 2015
16. Alfailakawi, M.G., Ahmad, I., Hamdan, S.: Harmony-search algorithm for 2D nearest neighbor quantum circuits realization. Expert Syst. Appl. **61**, 16–27 (2016)
17. Shrivastwa, R., Datta, K., Sengupta, I.: Fast qubit placement in 2D architecture using nearest neighbour realization. In: IEEE International Symposium on Nanoelectronic and Information Systems, pp. 95–100, December 2015

18. Wille, R., Keszocze, O., Walter, M., Rohrs, P., Chattopadhyay, A., Drechsler, R.: Look-ahead schemes for nearest neighbor optimization of 1D and 2D quantum circuits. In: Proceedings of ASP Design Automation Conference, pp. 292–297, January 2016
19. Kole, A., Datta, K., Sengupta, I.: A new heuristic for N-dimensional nearest neighbour realization of a quantum circuit. In: IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **12** (2017). https://doi.org/10.1109/tcad.2017.2693284
20. Barenco, A., et al.: Elementary gates for quantum computation. APS Phys. Rev. **52**, 3457–3467 (1995)
21. Sasanian, Z., Miller, D.Michael: Transforming MCT circuits to NCVW circuits. In: De Vos, A., Wille, R. (eds.) RC 2011. LNCS, vol. 7165, pp. 77–88. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29517-1_7
22. Miller, D., Wille, R., Sasanian, Z.: Elementary quantum gate realizations for multiple-control Toffolli gates. In: Proceedings of International Symposium on Multiple-valued Logic, pp. 217–222 (2011)
23. Sasanian, Z., Wille, R., Miller, D.M.: Realizing reversible circuits using a new class of quantum gates. In: Proceedings of Design Automation Conference, pp. 36–41 (2012)
24. Wille, R., Große, D., Teuber, L., Dueck, G.W., Drechsler, R.: RevLib: an online resource for reversible functions and reversible circuits. In: International Symposium on Multi-Valued Logic, pp. 220–225 (2008). RevLib is available at http://www.revlib.org