Optimization of Quantum Circuits for Interaction Distance in Linear Nearest Neighbor Architectures

Alireza Shafaei  Mehdi Saeedi  Massoud Pedram

Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089
{shafaeib,msaeedi,pedram}@usc.edu

ABSTRACT
Optimization of the interaction distance between qubits to map a quantum circuit into one-dimensional quantum architectures is addressed. The problem is formulated as the Minimum Linear Arrangement (MinLA) problem. To achieve this, an interaction graph is constructed for a given circuit, and multiple instances of the MinLA problem for selected subcircuits of the initial circuit are formulated and solved. In addition, a lookahead technique is applied to improve the cost of the proposed solution which examines different subcircuit candidates. Experiments on quantum circuits for quantum Fourier transform and reversible benchmarks show the effectiveness of the approach.

Categories and Subject Descriptors
B.6.3 [Logic Design]: Design Aids—Automatic synthesis

General Terms
Algorithms, Design

Keywords
Logic synthesis, quantum circuits, interaction distance, quantum architectures.

1. INTRODUCTION
Current technologies for quantum computing often need gates that involve geometrically adjacent qubits. The architecture of a quantum computing system can be described by a simple connected graph \( G = (V, E) \) where vertices \( V \) represent qubits and edges \( E \) represent adjacent qubit pairs where gates can be applied on [1]. Accordingly, a complete graph expresses the absence of any constraints. Quantum algorithms usually consider no interaction constraint between qubits. However, physical implementation may impose additional geometrical constraints. Therefore, the developed quantum algorithms or quantum circuits should be modified to consider the effect of various technological limitations.

Quantum computation technologies arrange qubits of a physical layout in a one (1D), two (2D), or three (3D) dimensional architecture. The Linear Nearest Neighbor (LNN) architecture corresponds to a graph where an edge exists between only neighboring vertices in a line. Two-dimensional square lattices (2DSL) corresponds to a graph on a Manhattan grid with four neighboring qubits. The three-dimensional square lattices (3DSL) model is a set of stacked 2D lattices with six neighboring qubits. Generally, 3DSL is less restrictive. However, it can suffer from the difficulty of controlling 3D qubits. Several quantum computing systems of trapped ions [2] and liquid NMR [3] have been designed based on the interactions in a line. 2DSL proposals include arrays of trapped ions [2] and Josephson junctions [4]. The architecture in [5] is based on the 3DSL model.

Exploring an efficient realization of a given quantum algorithm or quantum circuit for a restricted architecture — the focus of this work — has been followed by different researchers during the recent years. Physical implementation of the quantum Fourier transformation (QFT) [6, 7], Shor’s factorization algorithm [8–10], quantum addition [11], quantum error correction [12], and general reversible circuits [13] for the LNN/2DSL architectures have been explored in the past. Worst-case synthesis cost of a general/Boolean unitary matrix under the nearest neighbor restriction has been discussed in [14–17]. In [18, 19] heuristic methods for converting an arbitrary quantum circuit to its equivalent circuit on the LNN architectures have been proposed.

In this work, we model the problem of improving locality, i.e., reducing interaction distance, of a given quantum circuit by graph theory. Precisely, we use the minimum linear arrangement (MinLA) problem in graph theory to find optimized local quantum computation, in terms of the total synthesis cost or latency, in architectures with qubits arranged in a line. The rest of this paper is organized as follows. In Section 2, basic concepts are introduced. Prior work is discussed in Section 3. Section 4 describes the proposed approaches for locality improvement of quantum circuits. Experimental results are given in Section 5 and finally Section 6 concludes the paper. We also discuss how the proposed (MinLA)-based techniques can be generalized for 2D quantum architectures.

2. BASIC CONCEPTS

\[\text{Quantum technologies proposed mainly for quantum communication, such as photon-based model, is not considered here.}\]
In the following two subsections, we briefly discuss related concepts in quantum circuits and quantum architectures.

2.1 Quantum gates and circuits

A quantum bit, qubit, can be considered as a mathematical object which represents a quantum state with two basic states |0⟩ and |1⟩. In addition to the selected basis, a qubit can get any linear combination of its basic states. A quantum system which contains n qubits is often called a quantum register of size n. An n-qubit quantum gate performs a specific $2^n \times 2^n$ unitary operation on selected n qubits. The unitary matrix implemented by several gates acting on different qubits independently can be calculated by the tensor product of their matrices. Two or more quantum gates can be cascaded to construct a quantum circuit. For a set of k gates $g_1, g_2, \ldots, g_k$ cascaded in a quantum circuit C in sequence, the matrix of C can be calculated as $M_k M_{k-1} \cdots M_1$ where $M_i$ is the matrix of the i-th gate (1 ≤ i ≤ k). Given any unitary $U$ over m qubits $|x_1 x_2 \cdots x_m⟩$, a controlled-$U$ gate with k control qubits $|y_1 y_2 \cdots y_k⟩$ may be defined as an $(m + k)$-qubit gate that applies $U$ on $|x_1 x_2 \cdots x_m⟩$ iff $|y_1 y_2 \cdots y_k⟩ = |11 \cdots 1⟩$. For example, CNOT is the controlled-NOT with a single control, Toffoli is a NOT gate with two controls. A multiple-control Toffoli gate $C^k$ NOT is a NOT gate with k controls. In circuit diagrams, • is used for conditioning on the qubit being set to value one. A SWAP gate maps $|ab⟩$ into $|ba⟩$. We use $x$ on qubits of a SWAP gate in circuit diagrams. More information is in [20].

2.2 Physical layout

In a particular realistic physical layout, e.g., in an ion-trap quantum architecture [21], each qubit has a specific physical location at each time step. To apply a 2-qubit gate in a quantum computing system which uses mobile qubits, there is no need to move the involved qubits for a gate application. This reduces the circuit latency. The problem is that even after global reordering, some gates may remain (or be) non-local. In this case, one needs to add additional local operations, MOVE or SWAP, to move or permute qubits such that the qubits that are involved in the original non-local gates will be local afterward. This is called local reordering [18]. Note that after a local qubit reordering, qubit locations may be changed and working with the remaining gates may need additional reorderings. This is done by applying extra SWAP gates.

3. PRIOR WORK

For quantum architectures which support SWAP operation, a straightforward method to overcome the interaction constraints is to insert local SWAP gates in front of a non-local gate to permute lines (qubits) and move the involved lines toward each other. This should be followed by adding SWAP gates after the computation to recover the initial qubit ordering. For specific quantum circuits, one can explore more efficient implementations [6–9, 11, 12]. Additionally, one may try to use local gates “during” a general synthesis [13] instead of trying to reduce SWAP gates by a post-process approach. Although this seems interesting, considering locality besides other important metrics during the synthesis can complicate the overall process significantly. On another side, several researchers considered the overall impact of the interaction constraints on their developed circuits/constructions instead of working with actual circuits. In this case, they may “prove” that total cost may increase by a constant factor (e.g., 10 in [15] and < 2 in [14, 16]).

To work with arbitrary circuits, the authors in [18] developed exact and heuristic post-synthesis methods to reduce the number of SWAP gates. The exact method, which is limited to small circuits, was used in a peephole optimization approach. Along with 3 templates, the authors suggested two reordering strategies, global and local, where in global a qubit with the highest interaction impact is placed at the middle line continuously until no further improvement can be achieved. In local, the algorithm inserts SWAP gates only before a non-local gate, and the new ordering is used for the remaining gates.

In [19], the authors showed that a bubble sort generates the minimum number of SWAP gates required to construct an arbitrary permutation of qubits for each gate. They addition-

![Figure 1: (a) A sample circuit, and its interaction graph (b). (c) Circuit in (a) after applying global reordering.](image-url)
ally shown that in an n-qubit circuit, for two qubits of the
i-th gate positioned at locations $q'_i$ and $q'_j$ only qubits placed
between $q'_i$ and $q'_j$ should be considered instead of working
with all qubits (i.e., $|q'_i - q'_j|!$ permutations instead of $n!$
permutations). Note that finding the best local orderings for all
2-qubit gates needs considering all $|q'_i - q'_j|!$ permutations
for all gates at the same time (i.e., $|q'_i - q'_j|! \times |q'_k - q'_l|! \times \ldots$). To avoid this huge exponential search, authors worked with
at most $v$ consecutive gates.

The authors of [24] considered circuits that perform “specif-
ied” operations spanning $n$ wires with focus on depth. They
showed that rotation of $n$ wires with local gates can be done
in depth $n + 5$, reversing $n$ wires with local gates is possible
with depth $2n + 2$, swapping across $n$ wires by local gates
can be done in depth $n + 7$ for even $n$ and in depth $n + 8$
for odd $n$ with size $6n - 9$. More information is in [24].

4. THE PROPOSED METHOD

Basically, a quantum computer technology, e.g., architec-
tures based on ions, may support the MOVE operation to
transform a qubit from one physical location to another. A
physical location may also be shared by several qubits at the
same time. In this case, to make a local two-qubit gate, one
needs to change the locations of far qubits, by applying a
sequence of single-qubit MOVE operations. Note that there
should be enough room to hold the moving qubit(s) in in-
termediate and final physical locations. On the other hand,
a quantum architecture may not provide the MOVE opera-
tion e.g., in architectures based on superconductors. For
this case, one needs to apply SWAP gates which physically
change the locations of both involved qubits. For quantum
architectures with 1D interaction distance, the MOVE op-
eration and a physical location for multiple qubits are not
usual, and the previous approaches discussed in Section 3
worked with SWAP gates. The same limitations exist for
some 2D quantum architectures too. For other 2D quantum
architectures, the MOVE operation and a multi-qubit phys-
ical location exist. In the following sections, we propose our
methods to make gates local in 1D quantum architectures.
Potential problems to extend our approach for 2D quantum
architectures are outlined in Section 6.

The MinLA problem is defined for a weighted graph $G =
(V, E)$. The goal is to arrange the vertices $V$ of $G$ on an
integer line by a one-to-one function $f : V \rightarrow [1 \ldots |V|]$ to
minimize $\sum_{(u,v) \in E} w(u,v)f(u) - f(v)$ where $w(u,v)$ is the
weight of the edge between nodes $u$ and $v$. This problem
can be considered as a label assignment of the given graph.
The MinLA problem is NP-hard in general. However, poly-
nomial time algorithms to compute exact solutions for some
particular graphs are known. In addition, some approxi-
mation algorithms have been proposed in the past. More
information can be found e.g., in [25]. In this paper, the
degree of a vertex $v$ in graph $G$ is represented as $\deg(v)$.
The maximum degree of a graph $G$, denoted by $\Delta(G)$,
is the maximum degree of its vertices.

4.1 Label assignment for qubit reordering

Consider a given circuit $C$ with $n$ qubits $q_1, q_2, \ldots, q_n$ and
$m$ 2-qubit gates $g_1, g_2, \ldots, g_m$. Working on $C$, we construct
4We ignore single-qubit gates for locality improvement since they
have no effect on circuit locality. Another reason is that single-
qubit gates can be absorbed into surrounding two-qubit gates.
The resulting circuit is called a “skeleton” circuit in [7]. Through-

4.2 Subcircuits with consecutive gates

Consider a set of $w$ consecutive 2-qubit gates $A = \{g_1, g_2, \ldots, g_w\}$ for an $n$-qubit LNN architecture. Assume that the
gate $g_i$ works on qubits $q'_i$ and $q'_j$ (and $q'_i \neq q'_j$). There
are many different qubit arrangements. For an interaction
graph $G$, we may have:

- $\Delta(G) = 0$. This is a trivial case with no gate.
- $\Delta(G) = 1$. In this case, all gates use distinct qubits.
  Accordingly, one can find a qubit (re)ordering when
  for each gate $g_i$, the qubits $q'_i$ and $q'_j$ are adjacent.
  To achieve this, group $q'_i$ and $q'_j$ as a new qubit for all
  gates in $A$ for a total of $n - w$ qubit groups — each
group can include either one qubit (if the qubit is not
  used by any gates in $A$) or two qubits (if exactly one
  gate in $A$ uses the two qubits). There are $2 \times (n - w)!$
  qubit orderings to make all gates in $A$ local. No SWAP
gate is required in this case.
- $\Delta(G) = 2$ and there is no cycle in $G$. For this case, there
  is a “staircase” construction where all gates are local.
  Assume there are $k_0$ vertices with $\deg(v) = 0$, $k_1$ vertices
  with $\deg(v) = 1$, and $k_2$ vertices with $\deg(v) = 2$.
  There is a “staircase” construction where all gates are local.
  No SWAP gate is required in this case.
To select sets, one can try gates is preferred. In case of a tie, one is selected randomly.

one that leads to the minimum number of inter-set SWAP MinLA to the

w for gates in each set (vs. the whole circuit) and applying lem. This is done by constructing an interaction graph

best possible ordering for each set, we use the MinLA

be found in a polynomial time.

MinLA solution [25].

ples of graphs that can be solved in polynomial time with optimal

k

indexed from 1 to

MinLA find optimal

w

SWAP gates for set

i

ordering of set

MinLA

work-

Inter-set SWAP gates. For a circuit

C

with

m

gates and

n

qubits, assume that one finds

w1, w2, \ldots, w3

gates

consecutive
gates, such that all

w1

gates in

i

t set need at most

s

SWAPs to be local. Assume all

wi

gates in the

i
-
th set work on

ni

\leq

n1

qubits. Each set

i

with

wi

gates needs a new qubit reordering for the involved

n1

qubits. Accordingly, to have a local circuit

C'

for

C

, one needs to add

SWAP gates between sets

i

and

i + 1

for

1 \leq i < w - 1

to change the qubit ordering in set

i

to the one in set

i + 1
.

These SWAP gates are called inter-set SWAP gates. Working with an unlimited

s

leads to no inter-set SWAP gates. Figure

2
(a)
illustrates the concept. Note that the methods in

[18, 19]
are special cases of the method discussed above in the sense that they assume

m

gates for a circuit with

m

onnull

and

n

SWAPs, and then apply inter-set

SWAPs to locally construct adjacent gates.

Label assignment for local reordering. To find the best possible ordering for each set, we use the MinLA problem. This is done by constructing an interaction graph

G

gates in each set (vs. the whole circuit) and applying the MinLA problem for each set accordingly. The solution to the MinLA problem may not be unique, in this case the one that leads to the number of inter-set SWAP gates is preferred. In case of a tie, one is selected randomly.

To select sets, one can try

w = 3

consecutive gates starting from gate

i

and then apply the MinLA problem. If a relabeling and at most

s

SWAP gates are sufficient to make the gates local, increment

w

and redo. Otherwise, use

w - 1

and restart with

i + w
.

Note that the exact solution of the MinLA problem for some particular graphs can be found in a polynomial time.\(^5\) Hence, one may be able to find optimal MinLA solutions in several sets.

Intra-set SWAP & inter-set SWAP minimization. To minimize the number of SWAP gates one should particularly consider the value of

s

for each set. Consider

k

sets indexed from

1

to

k

each of which with

wi

gates and working on

ni

qubits. Assign

si

as the maximum number of intra-set SWAP gates for set

i
.

The value of

si

affects the final qubit ordering of set

i

as well as its following sets, and also the total number of sets. Figure

2
illustrates this effect with one example. Accordingly,

s

values and the total number of sets should be carefully determined. On the other hand, after distinguishing sets and appropriate qubit orderings for each set, we apply

[19, Theorem 1]

to find the minimum number of inter-set SWAP gates. This method is based on simulating the bubble sort algorithm.

4.3 SWAP minimization with lookahead

Consider an

n
-qubit subcircuit

C

with

W

gates divided

\Delta(G) \geq 3

or

\Delta(G) = 2

and with cycle(s) in

G
, at least one SWAP gate is required. Assume that we divide all

2
-qubit gates in

C

into

k

sets with only local gates, each set with at most

s

SWAP gates. For

s = 0
, all sets belong to either case

1

or case

2
.

Otherwise, one needs to add SWAPs, called intra-set SWAP gates. If it is not possible to make the current

w

gates local with

s

SWAP gates, decrement

w

to

w - 1

and recheck. If not, decrement and proceed. It can be verified that at least for

w = 2
, we can find a local subcircuit with no SWAP gate, i.e.,

s = 0
. A larger

s

leads to a larger

w
.

\(^5\)Trees, rectangular and square meshes and hypercubes are examples of graphs that can be solved in polynomial time with optimal MinLA solution [25].

5 Experimental Results

We implemented the proposed optimization method in C++ and all experiments were done on an Intel Core i7-3770 machine with 16GB memory. To achieve this, the program initially extracts an interaction graph from a given circuit. Then, it determines the number of sets and set borders in such a way that each set can be implemented locally with no SWAP gate. This step is followed by running several instances of the MinLA algorithm for each set. Finally, the algorithm inserts SWAP gates between adjacent sets as discussed. To reduce the number of SWAPs, for each set the ordering which leads to the minimum number of inter-set SWAP gates is selected.

To evaluate the proposed interaction distance optimization method, we compared our results with those obtained by applying the method in

[18]
for reversible benchmarks in

[5]
as well as for the quantum Fourier transform circuits. For all cases, the number of SWAP gates added by each method to construct a local circuit was compared. In

[18]
, quantum costs before and after the optimization were reported where SWAP gate was considered as a unit-cost gate (see [18, Table 3]). Accordingly, the number of SWAP gates is the difference of quantum cost before and after the optimization. We limited the runtime to 30 minutes and reported those cases that our algorithm leads to a solution. Table

1
reports the results.

For each circuit in Table

1
, besides the number of SWAP
gates we reported the number of sets and the minimum, maximum and average numbers of SWAP gates in each set. Note that we limited the algorithm to use $s=0$ in each set. Accordingly, no intra-set SWAP gate is used and all SWAPs are the result of applying inter-set SWAP insertion method. We also limited the algorithm to use a lookahead of depth $=1$ to reduce runtime. Increasing the lookahead depth improves the results with the penalty of runtime. As can be seen in Table 1, the average number of SWAP gates required to transform a local ordering from one set to another set is small, and our algorithm leads to a considerable reduction in the number of SWAP gates — 28%, on average and up to 60%. Figure 3 and Figure 4 illustrate the results of applying the proposed method on two benchmarks. In these circuits, all SWAP gates are inserted between sets.

6. CONCLUSION AND FUTURE WORK

In this paper, the interaction distance constraint in quantum architectures with 1D interactions was addressed. We modeled the interactions between gate qubits in a given circuit by an interaction graph, and used the well-known Minimum Linear Arrangement (MinLA) problem to find qubit reordering to improve circuit locality. The proposed approach divides a given circuit into several subcircuits and uses the MinLA instances within each subcircuit to find qubit locations for qubits involved in each subcircuit. Next, local SWAP gates are inserted inside each subcircuits to make the remaining non-local gates local. Finally, additional SWAP gates are inserted between subcircuits to transform one qubit ordering to another to keep circuit functionally unchanged. The proposed approach applies a lookahead to determine subcircuit borders and to minimize the total number of SWAP gates. Given that the MinLA is NP-hard, the problem of minimizing the number of necessary SWAP gates to run an arbitrary quantum circuits on LNN architectures is NP-hard. This addresses the conjecture in [19, page 25].

In addition to considering circuit depth and improving runtime to handle larger circuits, a major step in future direction is to consider the interaction distance constraint in 2D quantum architectures. This path has been followed for specific circuits in the past e.g., [10]. However, the case of general circuits needs attention.

- For architectures which do not support MOVE and use one physical location per qubit, we can extend MinLA to the 2-dimensional grid arrangement problem [27]. New methods are required to insert intra-set SWAPs and to determine initial qubit locations.
- For quantum architectures which support MOVE and may hold several qubits in one physical location our ideas should be revised. In this case, one can construct a graph for physical locations (vs. qubits) and use one node for qubits with the same location. This can be followed by a 2-dimensional grid arrangement problem to determine a qubit ordering. Besides the challenges stated above, the algorithm should handle the maximum size of intermediate and final physical locations when one qubit is passing from one physical location. Method in [28] is a related approach.

7. REFERENCES

Table 1: The synthesis results (# of SWAPs) for benchmarks in [26] as well as for quantum Fourier transform circuits after applying the method in [18] and ours. Runtime results (all in second) for [18] vary from \(\approx 0\) for small circuits to 1300 for large circuits. On average, the results in [18] are improved by 28%.

![Figure 4: The result of applying the proposed method on the 4gt11_84 benchmark. (a) Non-local circuit, (b) local circuit.](image)


**Acknowledgments**

This research was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20165. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.