Classiq

June 5, 2022

1 Preamble.

This document. On May-2022, Classiq, a pioneer quantum computing company, has lunched the first (at least at that scale) competition in quantum circuits programming. The purpose of this document is to describe a proposal solution for the Lithium simulation problem. Hence it important to notice that even that this paper has been written in LaTeX it is not an academic paper, that it, This paper might contains errors (and grammar mistakes for surly). Yet a lot of energy have been spent to bring that work to what it is, so the case that entire of it turned out to be completely nonsense probably won't happen.

The paper organized in the follow systematically manner, first in the preamble we will present the problem and talk about the general concepts which describe our specific Hamiltoinan, including the quantities regime (i.e number of qubits vs number of hamiltonians terms), the naive solution, and how mach improvement we could hope to make.

Then in next section we review all the techniques that we have used to improve a local gates assignment. Namely, this section deals with the question of how presenting each of the local term as subgate.

In contrast, the third and the forth sections reviews methods to determinate an order which clearly superior comparing to the naive approach. The forth section presenting an concept of analyzing the "product graph" a concept which could be thought as the second order analyses of the alternate path from the third section.

The problem. Generate a circuit, using no more than 10 qubits, that approximates the unitary e^{-iH} where H is the qubit Hamiltonian of a **LiH** (lithium hydride) molecule. The **LiH** Hamiltonian is composed of 276 Pauli strings, and can be found HERE. The approximation error is defined in the next section, and should be less than 0.1. The circuit should be composed of the CX and single qubit gates only.

Naive approach. Before diving into the technical details let's review our competitor first. Consider the straightforward assignment Lie-Trotter just handle each of Hamiltonian terms one by one. Assume the the hamming weight of the support of H_i equal to d_i. so we pay two steps to rotate each wire into the parity base and uncompute it in the end. then we could apply the CX from each qubit in the support to a chosen one which will sum the wires parity and finally will propgate the sum trough the $RZ(\Delta t)$ gate (rotation by the coefficient of the term and the step size). So in total we pay $(2 + 2d_i$ for each term and therefore the whole circuit will require:

$$D^{\text{naive}}(n, m) = \sum_i 2 + 2d_i = m \sum_i \frac{2 + 2d_i}{m} = 2m \left(1 + \mathbb{E}_{\sim i}[H]\right)$$

Where $D^{\text{naive}}(n, m)$ is the depth of the circuit which compute a single Δt step of Lie-Trotter formula. In order to getting a more solid feeling about what is good solution, let's assume for the moment that $\mathbb{E}_{\sim i}[H] > 5$, in that case we obtain that the naive approach obtain a circuit at depth $\geq 2 \cdot 276 \cdot 6 \sim 3100$. In the end we will see that our solution yields a 1759-depth circuit.

Characterize our LiH. A native question that one might ask is whether exists a more efficient way to presents the Hamiltonian. So even we can't a sign that there is not, let's try to estimate how mach improvement can be achived by applying the Kitaev transformation. In one sentence, Kitaev has showed that the ladder and

the annihilation operators has pauli representation such each operator touch at least $\log(n)$ of the qubits. By the fact that Hamiltonian in the energy base has "product" of four operators (i.e $a_i^\dagger b_j^\dagger a_n b_m$) we get that there are might be operators in the computing base which touch $4\log(n)$ qubits. In our case, $n = 10 \Rightarrow 4\log(n) > n$. Bottom line, It seems that the known techniques for reducing asymptotically the support of the terms don't work here. Hence, it's also an hint that 10-factor improvement is unlikely. Therefore we will aim to 2-factor of improvement over the naive.

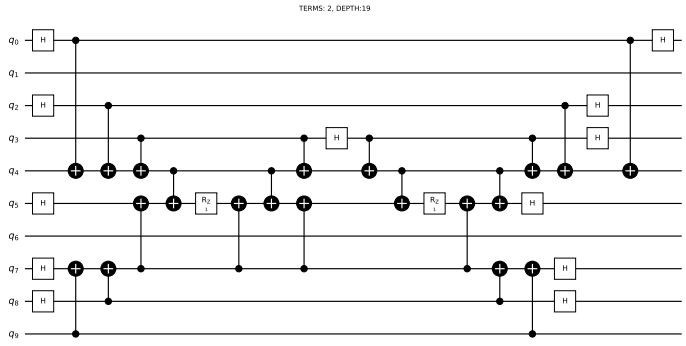

Figure 1: Demonstration of the above methods, applied over the following terms $XIXZZXIXXZ$ $XIXXZXIXXZ$. The fifth wire (qubit) is the main wire which sum the parity.

2 Single Term Heuristics.

Main wire principle. From now on we will call to the wire which sum the parity of the state the main wire. For given term we have chose the main wire to be the median of the wire in it's support. For example consider the following term:

$$XXXXIII$$

Then as the second wire is above (and beanth) half of the wires, it will be the main wire in that case. the separation into upper and lower sides will enable to us to guarantee that at least two CX will be sum-up in parallel (using the next heuristic).

Upper and lower summations. We summing the parity as follow, denote by $i_1, i_2, ...i_W, ...i_k \subset [n]$ the indices in the support of given term, such that i_W is the main wire. Then after rotate the wires into the phase base then we summing the parity of $i_1, i_2, ...i_{W-1}$ into i_{W-1}, That it, We apply a $CX\,(i_j, i_{W-1})$ for $j < W-1$.

As all the segements $[i_j, i_{W-1}]$ disjoint to any segments of the form $[i_{W+1}, i_j]$ for $j > W+1$ then we could sum the parity of $i_{W+1}, i_{W+2}..i_k$ wires into i_{W+1} in parallel. It's easy to see that this heuristic cut by almost half the depth of single term.

3 Greedy Order Heuristics.

Next, we will review our methods to impose the gates order. Clearly the order does matter, to see that consider an hamiltonian pair which differ by only single coordinate i.e $XXXX$ and $XXXZ$. In that case the contribute of the first tree qubits remain the same for both of them, and therefore there is no needed to uncompute them after the applying of the first gate.

Sample Greedy Diameters. Assume that we given a set of therms with a promise that it might has a lot of "closer" terms (relative to the Hamming distance of their supports). Our first simple heuristic goal is to chain them together in greedy manner such that the distance of each adjacent terms will be minimal.

Algorithm 1: Chain an Hamiltonian set

while $G \neq$ **do**
  $T \leftarrow$ Greedy Spanning Tree (G)
  $D_i \leftarrow$ Diameter (T)
  $G \leftarrow G/Di$
end
return $D_0...D_l$

It's worth to note that replacing the Greedy Spanning Tree by MST (and the diameter by just the DFS scanning order of the MST) could be proving as 2-OPT in the terms of chaining. We think about the uncompute stage as the climbing back at the tree, and therefore **OPT** $\leq 2w\,(\text{MST})$. But also it easy to see that every

path which contain all the vertices has weigth greater than $w\,(\text{MST})$, Hence $2w\,(\text{MST}) \leq 2\mathbf{OPT}$. Yet, By the fact that we have already improved the naive solution at factor which close to $\frac{1}{2}$ and by the intuition that the optimal depth is not very far from the naive (don't expect that $D^{\text{naive}} \geq 4\mathbf{OPT}$ we decided to stick to the greedy construction, which in the next chapter will generalized to the product graph.

Hyperplanes separation. The cost of chaining pair of Hamiltonians isn't a monotone function of their Hamming distance, for example consider the case of two terms which have not overlapping at all, i.e $XXXXIIII$ and $IIIIXXXX$ then it's clear that those two can be impose such they will be computed in parallel.

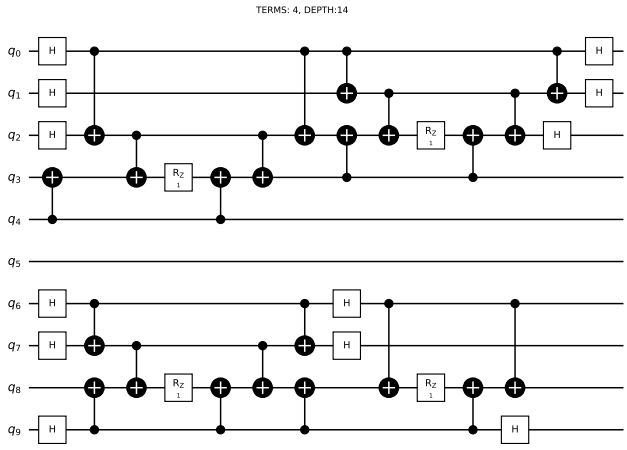

Figure 2: Example for a case in which chaining terms with high distance reduce the depth of the circuit. Here the terms are : $XIXZZIIIII$ $XXXZIIIIII$ $IIIIIIXXZX$ $IIIIIIZIZX$.

Technically we have separated those groups by follow, first we have sorted the term collection by the follow weight functions f, g such that:

$$f\,(H_i) = \min j : H_{j'} = I \text{ for } j' > j$$
$$g\,(H_i) = \max j : H_{j'} = I \text{ for } j' \leq j$$

And then we have matched pairs till we get into intersection.

4 The Product Graph.

The main disadvantage of the last separation is that it doesn't take into a count any relation between terms which classified in the same side of the plane. The suggested solution: We will say that the pair H_i, H_j share a "solid product" if $f\,(H_i) \leq g\,(H_j)$ where f, g defined above. We denote such relation by $(H_i, H_j) \in \mathbf{SP}$. Define the graph $G^2 = (V \times V, E')$ where V is the set of the terms, to be:

$$E' = \{\{(u,v),(w,z)\} : (u,v),(w,z),(u,z),(w,v) \in \mathbf{SP}\}$$
$$w(e) = w\,(\{(u,v),(w,z)\}) = \max w(u,w), w(v,z)$$

Note, that in the G^2 path which pass through all the vertices contain an $|V|$ copies of each term. Hence looking for spanning trees makes no sense. Therefore our diameters sampler will be more appropriate for that task (when we ensure that each sampled tree contains at most one copy of each term).

Another advantage of this method that it can be easily generalize to scanning triples or higher orders by taking higher power of the graph, the main disadvantage is (of course) processing time.

The final submission is a mixture of all the methods we have reviewed above.