Solving the subset sum problem on a quantum computer

by Sumeet Shirgure

USC, Fall 2022

1 Introduction

In this report, I will describe a quantum algorithm to solve the subset sum problem, and the result of running a small instance on a real quantum computer. I'll also provide references to some related ideas / papers that I found interesting.

Section 2 introduces the subset sum problem, and some classical algorithms for solving it. Section 3 discusses Fourier states and related topics. Section 4 describes the construction of an oracle that can be used within Grover's algorithm [4]. Section 5 presents the results of running the resulting circuit on a trapped ion - based quantum computing device.

2 The subset sum problem

The (promised) subset sum problem can be formulated as follows :

Given a list L of positive integers $[v_0, v_1, \ldots v_{n-1}]$, and a target sum t, the task is to construct a subset $S \subseteq \{0, 1, \ldots, n-1\}$ with $\sum_{i \in S} v_i = t$.

Equivalently, representing subsets of indices with $n-$bit strings $s = s_0 s_1 \ldots s_{n-1}$, we are searching for a string $s \in \{0, 1\}^n$ such that $\sum_i s_i v_i = t$. For simplicity, we assume the "promise" version of this problem where such a subset is guaranteed to exist. There also exist techniques to solve approximate versions of this problem. While approximate quantum algorithms are fun to think about, we won't be going into it for this problem.

The running time and space requirements of algorithms used to solve this problem depends on two parameters – the number of elements n and the bit precision k ($\sum_i x_i < 2^k$). This problem is known to be NP-complete [7]. Therefore it's unlikely that we can solve it in polyomial time, even with a quantum computer. Table 1 shows the computational complexities of a few classical algorithms.

Grover search is marked with an asterisk because the time complexity is measured with the circuit depth and the memory is measured in number of qubits, while those for the classical algorithms are in clock cycles and ordinary bits of a standard RAM-model computer.

To use Grover's algorithm, we need to construct a quantum circuit "recognizing" strings of subsets that sum to the required target. Constructing such an oracle effeciently required (to some extent) the use of special qubit states – *Fourier states*.

Algorithm	Time	Memory
Brute force	$O(nk2^n)$	$O(n+k)$
Dynamic programming[7]	$O(nk2^k)$	$O(nk2^k)$
“Meet in the middle”[5]	$O(nk2^{n/2})$	$O(nk2^{n/2})$
Optimized meet in the middle[10]	$O(nk2^{n/2})$	$O(nk2^{n/4})$
*Grover search	$O(nk2^{n/2})$	$O(n+k)$

Table 1. Time and space complexities of various algorithms

3 Fourier states

This section will borrow notations and terminology from this paper [1] titled “Fast parallel circuits for the quantum Fourier transform”. A $k-$bit integer $x\in\mathbb{Z}_{2^k}$, is said to have a binary representation $x_{k-1}x_{k-2}\ldots x_0\in\{0,1\}^k$. Similarly, the decimal value $0.x_0x_1\ldots$ is understood as equal to $x_0/2+x_1/4+\cdots+$.

The quantum state $|x_{k-1}x_{n-2}\ldots x_0\rangle$ is called a *computational basis state* (or a *Z-basis state*), and the following state is called the *Fourier basis state*.

$$|\psi_x\rangle\equiv\frac{1}{\sqrt{2^k}}\sum_{y=0}^{2^k-1}\left(e^{2\pi i\frac{xy}{2^k}}\right)|y\rangle$$
$$=\frac{1}{\sqrt{2^k}}(|0\rangle+e^{2\pi i(0.x_0)})(|0\rangle+e^{2\pi i(0.x_1x_0)})\ldots(|0\rangle+e^{2\pi i(0.x_{k-1}x_{k-2}\ldots x_0)})$$

For convenience, define $|\mu_\theta\rangle=\frac{1}{\sqrt{2}}(|0\rangle+e^{2\pi i\theta}|1\rangle)$, which on the Bloch sphere lies on the $x-y$ plane with an angle of $2\pi\theta$ from the $x-$axis as viewed from the positive z direction. (The visualizations provided in the attached presentation correspond to this picture.) Using this notation, we get

$$|\psi_x\rangle=|\mu_{0.x_0}\rangle|\mu_{0.x_1x_0}\rangle\ldots|\mu_{0.x_{k-1}x_{k-2}\ldots x_0}\rangle$$

The quantum Fourier transform is the unitary operation $\sum_{x\in\mathbb{Z}_{2^k}}|\psi_x\rangle\langle x|$, i.e it maps $|x\rangle\rightarrow|\psi_x\rangle\forall x\in\mathbb{Z}_{2^k}$. There also exist related transformations for preparing a Fourier basis state $|\psi_x\rangle$ given $|x\rangle$ as $|x\rangle|0\rangle\longrightarrow|x\rangle|\psi_x\rangle$. (This is a simple modification of the circuit given in [8], where we first apply Hadamards on the second register, and then apply the respective rotations in parallel to get a depth of $O(k)$.)

The authors in [1] push this idea to the limit by studying the inverse problem of *quantum Fourier phase computation* – a unitary mapping $|\psi_x\rangle|0\rangle$ to $|\psi_x\rangle|x\rangle$, and using its inverse to compute the mapping

$$|x\rangle|0\rangle\rightarrow|x\rangle|\psi_x\rangle\rightarrow|0\rangle|\psi_x\rangle \text{ (as noted by [6])}$$

with a circuit depth smaller than $O(k^2)$.

After the problem in homework 4, I too started thinking about Fourier state computation, and whether it would result in better than $O(k)$ circuits if we discard small phase rotations. That's when I found the paper [1] with a claimed depth of $O(\log(k)+\log(\log(1/\varepsilon)))$ for an $\varepsilon-$approximate QFT.

Coincidentally, for the case of the exact QFT, the authors appeal to the algorithm of Schönhage and Strassen [9] being implemented reversibly, which was one of the things I was originally thinking of doing my term project on.

As we will see in the next section, we can use the concept of Fourier states and the quantum Fourier transform to perform controlled additions.

4 The subset sum oracle

For a fixed list $L = [v_0 \ldots v_{n-1}]$, let $f\colon \mathbb{Z}_{2^n} \rightarrow \mathbb{Z}_2,\ f(s) \equiv \begin{cases} 1, \sum_i s_i v_i = t \\ 0, \text{otherwise} \end{cases}$

Finding a subset with sum t is equivalent to searching a string $s \in \mathbb{Z}_{2^n}$ with $f(s) = 1$. If we can construct a quantum circuit for the unitary $U_f = \sum_{x \in \mathbb{Z}_{2^n}} (-1)^{f(x)} |x\rangle\langle x|$, it will be possible to use Grover's algorithm to amplify the amplitudes corresponding to the strings with $f(x) = 1$, and then measure in the computational basis to obtain those strings with high probability.

Define $h\colon \mathbb{Z}_{2^n} \rightarrow \mathbb{Z},\ h(s) \equiv \sum_{i=0}^{n-1} s_i v_i$. The quantum circuit to detect if $h(s) = t$ can be broken down into an adder, and a detector. The adder performs the mapping on two registers $|s\rangle_n$ and $|y\rangle_k$ as $|s\rangle|y\rangle \rightarrow |s\rangle|(y + h(s)) (\operatorname{mod} 2^k)\rangle \forall s \in \mathbb{Z}_{2^n}, y \in \mathbb{Z}_{2^k}$, using $n + k$ qubits. The detector acts on the second register by applying $X^{(1 - t_{n-1})} \otimes X^{(1 - t_{n-2})} \otimes \cdots \otimes X^{(1 - t_0)}$, followed by a multi-controlled-X gate on an ancilla qubit with the second register as control. Effectively we implement the mapping

$$|s\rangle|y\rangle|z\rangle \longrightarrow |s\rangle|(y + h(s)) (\operatorname{mod} 2^k)\rangle|z \oplus f(s)\rangle$$

for all valid s, y, z. Finally, to reuse the second register we need to uncompute the addition : $|s\rangle|y\rangle|z\rangle \longrightarrow |s\rangle|(y + h(s)) (\operatorname{mod} 2^k)\rangle|z \oplus f(s)\rangle \longrightarrow |s\rangle|y\rangle|z \oplus f(s)\rangle$. Initializing the ancilla qubit in $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$, the circuit will then implement the map that we require : $|s\rangle|y\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \rightarrow (-1)^{f(s)}|s\rangle|y\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$

All that remains is to implement the adder. While it's easy to convert standard sum+carry based bitwise adders into reversible circuits, note that we also need to condition the addition of v_i to a target register controlled by the qubit $|s_i\rangle$. Reversible addition itself needs a reversible and-gate, which has to be implemented with Toffoli or Fredkin gates [11], [3]. However, multi-controlled gates are costly.

A better alternative is to use Fourier states (section 3) to implement addition. This idea is explained in [2]. We can use controlled phase gates

$$P(\theta) \equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{2\pi i\theta} \end{pmatrix}$$

which is a continuous analog of the controlled-Z gate (CZ=$P(1/2)$).

Note that $P(\theta)|z\rangle|\mu_{\varphi}\rangle = |z\rangle|\mu_{(\varphi + z\theta)}\rangle$. Therefore applying $P(x_1/2^j)$ between the control qubit and the target register's j^{th} qubit ($\forall j \in \{0, 1, \ldots k - 1\}$) is equivalent to mapping $|z\rangle|\psi_y\rangle \rightarrow |z\rangle|\psi_{y + z x_1}\rangle \forall z \in \mathbb{Z}_2, y \in \mathbb{Z}_{2^k}$. I.e, "controlled addition" in Z-basis is transformed to "controlled rotation" in Fourier basis.

However, since we need the second register to be in the Z-basis for the detector to work, we have to use the inverse quantum Fourier transform to map $|\psi_x\rangle \to |x\rangle$. Lastly, to bring $|00\ldots0\rangle \to |\psi_0\rangle = |\mu_0\rangle|\mu_0\rangle\ldots|\mu_0\rangle$, we simply apply Hadamard gates.

In summary, the adder circuit consists of Hadamards to convert $|s\rangle|0\rangle \to |s\rangle|\psi_0\rangle$, followed by controlled phase rotation with i^{th} wire of the first register as control, to add controlled phase rotations corresponding to v_i on the second register, for every $i \in \{0, 1, \ldots n-1\}$. (Notice that the circuit's structure depends on the instance (L, t), and the written code *constructs* the circuit.) Once the Fourier state corresponding to the subset sum is prepared, an inverse Fourier transform maps the sum to Z-basis for the detector

$$|s\rangle|0\rangle \to |s\rangle|\psi_0\rangle \to |s\rangle|\psi_{h(s)}\rangle \to |s\rangle|h(s)(\operatorname{mod} 2^k)\rangle$$

5 Results

I wrote the code (github.com/sumeetshirgure/qchack2022-microsoft-challenge) to generate the oracle and to perform Grover search, and used it to construct a circuit solving a small subset sum instance $L = [5, 7, 8, 9, 1]$ with a target $t = 16$. This instance needs $n = |L| = 5$, $k = 5$, and required $n + k + 1 = 11$ qubits. There are two possible solutions : 01101 and 01010. The number of Grover iterations to maximize the amplitude of the solution subspace is given by $\frac{\pi}{4}\sqrt{\frac{32}{2}} \approx 3$. Of course, in a real application, the number of solution will not be known. But it can be estimated from the eigenvalues of the oracle described in section 4.

This circuit was then executed on the 11 qubit trapped-ion quantum computer developed by IonQ, which was the largest available at the time. The histogram in figure 1 statistics show the measurement results of that execution.

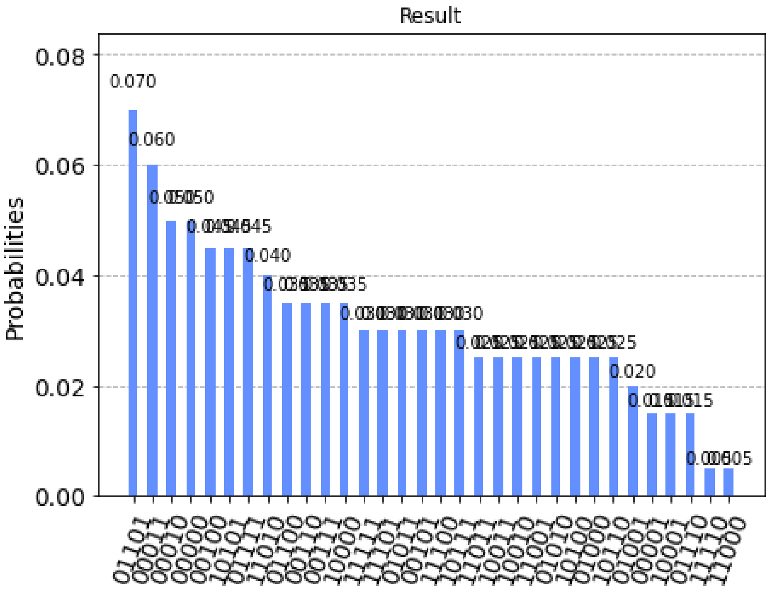

Figure 1. Measurement statistics from the quantum computer.

Note that the mode is a solution (01101), which is quite impressive for a circuit with depths of the order of a thousand or so one and two qubit gates, implemented without error correction. It's not entirely clear why the other solution (01010) isn't as frequently observed. Also note that the all zero string (00000) is observed a lot frequently, suggesting erasure errors in the device.

6 Conclusion

In this project, I wrote the code to construct quantum circuits solving the subset sum problem.

To use Grover's algorithm, a significant portion of the project was constructing an oracle that recognizes subsets that have the required sum. To do that, I designed a custom adder that computes the sum of a selected subset. To circumvent using reversible controlled-and gates, the adder performs rotations in the Fourier basis using controlled phase gates, and uses an effecient circuit for the quantum Fourier transform to switch between so called Z-basis and Fourier basis encodings for $k-$bit integers.

Finally, I executed a small instance of the problem on a quantum computer and analyzed the resulting measurement statistics. As an ambitious future exercise, I would like to do something similar, either other NP-hard problems, or perhaps smaller instances of subset sum, with error correction.

Thank you for reading.

Bibliography

[1] Richard Cleve and John Watrous. Fast parallel circuits for the quantum fourier transform. 2000.

[2] Thomas G. Draper. Addition on a quantum computer. 2000.

[3] Edward Fredkin and Tommaso Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21(3):219–253, Apr 1982.

[4] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219. New York, NY, USA, 1996. Association for Computing Machinery.

[5] Ellis Horowitz and Sartaj Sahni. Computing partitions with applications to the knapsack problem. *J. ACM*, 21(2):277–292, apr 1974.

[6] A. Yu. Kitaev. Quantum measurements and the abelian stabilizer problem. 1995.

[7] Jon Kleinberg and Éva Tardos. *Algorithm Design*. Addison Wesley, 2006.

[8] Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, England, dec 2010.

[9] A. Schönhage and V. Strassen. Schnelle multiplikation großer zahlen. *Computing*, 7(3):281–292, Sep 1971.

[10] Richard Schroeppel and Adi Shamir. A $T=O(2^{n/2}), S=O(2^{n/4})$ algorithm for certain np-complete problems. *SIAM Journal on Computing*, 10(3):456–464, 1981.

[11] Tommaso Toffoli. Reversible computing. In Jaco de Bakker and Jan van Leeuwen, editors, *Automata, Languages and Programming*, pages 632–644. Berlin, Heidelberg, 1980. Springer Berlin Heidelberg.