UDC 004.421
Original Scientific Paper

Quantum Algorithm for Finding a Maximum Clique in an Undirected Graph

Alan Bojić *alan.bojic@in2.hr*
IN2 Ltd.
Marohniceva 1/1, HR-10000 Zagreb, Croatia

Abstract

The maximum clique in an undirected graph is the largest subset of a set of graph's vertices where each pair of elements in the subset is connected. The problem of finding maximum clique in an arbitrary graph is NP-Hard problem with many applications in practice and it is very important to try to develop a new and a potentially faster algorithms that solve it. In this paper I would like to propose a new algorithm for quantum computers that finds the maximum clique in an arbitrary undirected graph which is based on a version of Grover's quantum algorithm for finding an element in an unsorted list in which there can be an unknown number of solutions. A new algorithm has $O(|V|\sqrt{2^{|V|}})$ worst case time complexity and $O(\sqrt{2^{|V|}})$ best case time complexity. Algorithm's space complexity for each case is $O(|V|)$.

Keywords: quantum computing, quantum algorithm, maximum clique

1. Introduction

The theory of quantum computing is a scientific field which researches how to use quantum mechanical effects such as entanglement and superposition in computing [1], [12]. A great deal of progress has been made and many researchers around the world attempt to build a large-scale quantum computer. Algorithms for quantum computers are not intuitive like those for classical computers because quantum mechanical effects must be considered during the construction of an algorithm [11]. In the theory of quantum algorithms there are several techniques how a quantum algorithm can be built. It can be built by using an amplitude amplification technique, or more specifically when an algorithm amplifies the amplitude of the base state which is the solution of the problem until it reaches very high probability [8]. Last measurement of the quantum system provides the outcome which is most likely the solution to the problem. The most known quantum algorithm which is based on amplitude amplification is Grover's algorithm [7] for finding an element in an unsorted list. Grover's algorithm runs in time $O(\sqrt{N})$ where N is the number of elements in the list. Grover's algorithm has a huge impact on the development of quantum algorithms and many proposed quantum algorithms are based on it. For instance, Carlile Lavor, Leo Liberti and Nelson Maculan proposed a quantum algorithm [9] that solves the molecular distance geometry problem, Dan Ventura and Tony Martinez proposed quantum associative memory [15], Paulo Mateus and Yasser Omar proposed a quantum algorithm for closest pattern matching [10], Mihai Udrescu, Lucian Prodan and Mircea Vladutiu proposed a methodology for running genetic algorithms on a quantum computer [14], Riccardo Franco proposed a model of human memory processes [6]. All mentioned achievements are based on Grover's algorithm. Another popular technique for building a quantum algorithm is the implementation of the quantum Fourier transform [8]. Quantum Fourier transform enables super-polynomial speed-ups compared to the algorithms for the same problem on a classical computer. The best known quantum algorithm which used the advantage of quantum Fourier transform is Shor's algorithm. Shor's algorithm [13] solves the problem of integer factorization in time $O((\log N)^3)$ which is exponentially faster than any known algorithm for integer factorization for a classical computer. Shor's algorithm represents one of the strongest proofs that a quantum computer could have significantly more computing power than a classical computer. The maximum clique in an undirected graph *G=(V,E)* is the largest subset of a set of graph's vertices where each pair of elements in the subset is connected [5]. The problem of finding the maximum clique is NP-Hard problem, so there doesn't exist polynomial time algorithm that solves it. The maximum clique problem has applications in many areas like data mining, network analysis and informatics, so it's of great interest to try to develop faster algorithms that solve it. Today, there exist several quantum algorithms that solve the maximum clique problem. Andrew M. Childs, Edward Farhi, Jeffrey Goldstone and Sam Gutmann proposed a quantum algorithm for finding maximum clique in their work "Finding cliques by quantum adiabatic evolution" [4]. They based their algorithm on quantum adiabatic evolution and it has $O(|V|^2)$ time complexity for graphs where number of vertices is $|V| \leq 18$. However, they didn't introduce the time complexity of the proposed algorithm for an arbitrary large graph. Weng-Long Chang, Ting-Ting Ren, Mang Feng, Jun Luo, Kawuu Weicheng Lin, Minyi Guo and Lai Chin Lu proposed a quantum algorithm that simulates DNA-based algorithm for finding maximum clique problem [3]. The time complexity of their algorithm is $O(|V|\sqrt{2^{|V|}})$. In this paper I would like to take a different approach than authors of existing quantum algorithms for maximum clique and propose a new quantum algorithm for maximum clique in an arbitrary graph which is based on Grover's algorithm. Complexity of a new quantum algorithm will be analyzed and compared to the complexity of existing quantum algorithms for the maximum clique problem.

Grover's algorithm is described in section 2, whereas section 3 contains the logic of the proposed algorithm, time and space complexity analysis of the proposed algorithm is described in section 4, and section 5 provides the conclusion and reference to future research.

2. Grover's Algorithm

Grover's algorithm is a way of finding an element in an unsorted list with *N* elements using the quantum computer. It was discovered and proposed by Lov Grover in 1996, who described it in his famous work "A fast quantum mechanical algorithm for database search" [7]. An algorithm is based on an amplitude amplification of the base state which describes a position of a searched element in the list. The time complexity of Grover's algorithm is $O(\sqrt{N})$ compared with $O(\frac{N}{2})$ time complexity for an average case on a classical computer. It has also been shown that the time complexity of $O(\sqrt{N})$ is optimal for a quantum computer for the unsorted list searching problem [16]. Each iteration of Grover's algorithm amplifies the probability amplitude of the base state, which describes a position of an element in the list. Quantum circuit for a single iteration within Grover's algorithm is shown in Figure 1 [11].

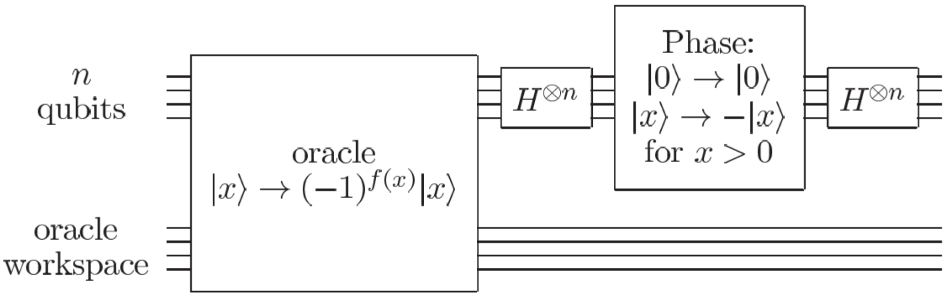
n
qubits
oracle
workspace
oracle
$|x\rangle \to (-1)^{f(x)}|x\rangle$
$H^{\otimes n}$
Phase:
$|0\rangle \to |0\rangle$
$|x\rangle \to -|x\rangle$
for $x > 0$
$H^{\otimes n}$

Figure 1. Quantum circuit for a single iteration of Grover's algorithm

Iteration of Grover's algorithm uses Oracle which identifies the position of the searched element, which is stored inside a quantum register of size $n = \log_2 N$ qubits. Oracle uses the function *f(x)* which returns 1 if state *x* is the one searched or 0 if state *x* isn't the one searched. Oracle also requires its own quantum register (workspace) for its operations. Overall space complexity of a single Grover iteration is $O(n)$. After Oracle has finished, Hadamard's gate on *n* qubits, Phase shift gate and then again Hadamard's gate on *n* qubits will amplify the probability of searched base state for $O(\frac{2}{\sqrt{N}})$. After $O(\sqrt{N})$ iterations, probability of the desirable base state will be 1 and we conduct a measurement of the quantum register (first *n* qubits). Figure 2 [11] shows the quantum circuit for Grover's algorithm.

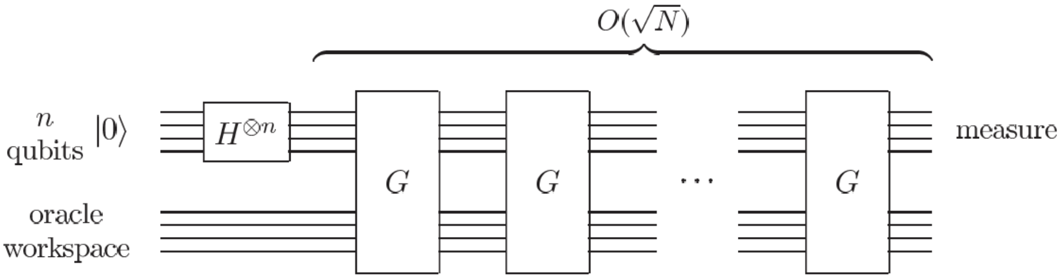

Figure 2. Quantum circuit for Grover's algorithm

The initial state for Grover's algorithm is $|\psi_0\rangle = |0\rangle$. Before the execution of the first Grover's iteration, Hadamard's gate puts the system in a uniform superposition of all possible base states, as shown in the following equation:

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle.$$

It can also be showed that if the element which we are searching for in an unsorted list appears *t* times, then algorithm can progressively increase the set of numbers from which we uniformly choose a number of iterations for Grover's algorithm. The time complexity of such a version of Grover's algorithm is $O(\sqrt{\frac{N}{t}})$. The following steps describe the version of Grover's algorithm for an unknown number of solutions proposed by Michel Boyer, Gilles Brassard, Peter Høyer and Alain Tapp in their well-known paper "Tight bounds on quantum searching" [2].

1. Let *T* be the unsorted list and *x* the searched value. Initialize *m* = 1 and set *λ* = 6/5. (Any value of *λ* strictly between 1 and 4/3 would do.)
2. Choose *j* uniformly at random among the nonnegative integers smaller than *m*.
3. Initialize quantum register of the size $n = \log_2 N$ in state $|\psi_0\rangle = |0\rangle$ and then use Hadamard's gate on every qubit to put the quantum register in the uniform superposition of base states. $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$.
4. Apply *j* iterations of Grover’s algorithm
5. Observe the register: let *i* be the outcome.
6. If *T[i]* = *x*, the problem is solved. Exit.
7. Otherwise, set *m* to $min(\lambda m, \sqrt{N})$ and go back to step 2.

3. Proposed Algorithm

The maximum clique in an undirected graph *G=(V,E)* is the largest subset of a set of graph's vertices where each pair of elements in the subset is connected [5]. Proposed quantum algorithm for finding a maximum clique in an arbitrary undirected graph is based on the version of Grover's algorithm which finds the searched element in an unsorted list in time $O(\sqrt{\frac{N}{t}})$, where *t* is the number of elements which satisfy the search criteria. Before describing the steps of the proposed algorithm, it is necessary to define two important things for the algorithm: the information that will be stored in the quantum register and the processing logic of the algorithm.

3.1. Information Stored Inside the Quantum Register

All possible subsets of a set of graph's vertices should be stored in a quantum register with the same probability amplitude. There are $2^{|V|}$ possible subsets so therefore $|V|$ quantum bits will be needed to store such information. The algorithm will achieve this quantum register's state by applying Hadamard's operator on every qubit in the quantum register which is initialized in the state $|\psi_0\rangle = |0\rangle$. It's important to emphasize that the quantum register will have $2^{|V|}$ possible base states and when we measure the quantum register, we will receive one of the $2^{|V|}$ base states. The received base state will be interpreted as an array of binary digits of length $|V|$. The binary digit 1 represents the vertex which is included in the resulting subset, and the binary digit 0 represent the vertex which is excluded from the resulting subset.

3.2. Algorithm Processing Logic

The Processing logic of an algorithm is to use a version of Grover's algorithm for an unknown number of solutions in order to find a larger size than that of the value stored in some variable k, which is initially set to value 1. When an algorithm finds such a clique, it stores the clique size in the variable k and the clique in the vector $|r\rangle$, and runs again. Initially, the value in the vector $|r\rangle$ will be $|0\rangle$. The algorithm iterates through these steps until the size of the retrieved clique is not larger than the value stored in the variable k or the resulting subset is not a graph's clique. Finally, the algorithm returns the vector $|r\rangle$ and if it is different than $|0\rangle$, one of the maximum cliques of a given graph is found.

The steps of the algorithm are as follows:

1. Initialize variable $k = 1$ and vector $|r\rangle = |0\rangle$.
2. Define function *f(x)* for an oracle which will be used inside Grover's algorithm. Function *f(x)* returns 1 if base state *x* represent a clique of the graph and its size is larger than value stored in the variable *k*, otherwise function returns 0.
3. Initialize quantum register of size $n = \log_2 N$ in state $|\psi_0\rangle = |0\rangle$ and then use Hadamard's gate on every qubit to put the quantum register in the uniform superposition of base states. $|\psi\rangle = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle$.
4. Run Grover's algorithm for an unknown number of solutions.
5. Measure quantum register. If outcome is a clique and its size is larger than value in the variable k, then put outcome in the vector $|r\rangle$ and clique size in variable k and go to step 2, otherwise go to step 6.
6. Return vector $|r\rangle$.

4. Complexity Analysis

4.1. Space Complexity

An algorithm needs a $|V|$ qubit register in the uniform superposition state to store all subsets of a set of graph's vertices. The algorithm also needs space to store the size of currently found clique which is returned by algorithm's iteration. Because the size of the maximum clique of graph is maximally $|V|$, algorithm needs classical $\log_2 |V|$ bits for variable k to store such information. The algorithm will also need additional classical $|V|$ bits for vector $|r\rangle$ to store the last successfully found clique of a graph. To summarize, an algorithm needs one $|V|$ qubit register, one $|V|$ bit classic register and one $\log_2 |V|$ classical register, that is, the overall complexity of algorithm is O($|V|$). It is important to emphasize that the space complexity of algorithm step 5 is not relevant because we can determine the size of the subset and check if some subset of a set of graph's vertices is clique in polynomial space of |V| even on a classical computer.

4.2. Time Complexity

For the purposes of a time complexity analysis, please consider the following undirected graph.

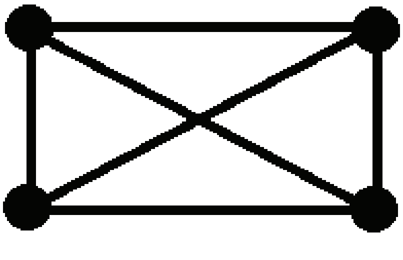

Figure 3. Graph G where ω(G) = |V|

Its maximum clique includes all graph's vertices and its clique number is *ω(G)*=4. *ω(G) is* the clique number of graph *G=(V,E)*, that is, the number of vertices in a maximum clique in graph *G* [5].

In an arbitrary graph *G=(V,E)* there are at least $2^{\omega(G)} - (\omega(G)+1)$ cliques. This is because every subset of a maximum clique, which has more than one element, is also a graph's clique. For the graph shown in Figure 3, *ω(G)* = 4, there are at least $2^4 - (4+1)$ = 11 cliques in the graph.

Since the logic of the algorithm is to iteratively increase the size of a searched graph's clique, there could be maximally *ω(G)* iterations of an algorithm. Algorithm's iterations in worst case scenario are:

1. Iteration - algorithm finds a graph's clique of size 2. In general, the worst case complexity of this iteration is $O\left(\sqrt{\frac{2^{|V|}}{\binom{\omega(G)}{2}+..+\binom{\omega(G)}{\omega(G)}}}\right)$, that is $O\left(\sqrt{\frac{2^{|V|}}{2^{\omega(G)}-(\omega(G)+1)}}\right)$. For graph shown in Figure 3, the complexity of this iteration is $O\left(\sqrt{\frac{2^{|V|}}{11}}\right)$.

2. Iteration - algorithm finds a graph's clique of size 3. In general, worst case complexity of this iteration is $O\left(\sqrt{\frac{2^{|V|}}{\binom{\omega(G)}{3}+..+\binom{\omega(G)}{\omega(G)}}}\right)$, that is $O\left(\sqrt{\frac{2^{|V|}}{2^{\omega(G)}-(\omega(G)+1)-\binom{\omega(G)}{2}}}\right)$. For the graph shown in Figure 3, the complexity of this iteration is $O\left(\sqrt{\frac{2^{|V|}}{4}}\right)$.

.

.

ω(G)-1. Iteration - algorithm finds only one of a graph's clique of size *ω(G)*. In general, the complexity of this iteration for the worst case scenario is $O\left(\sqrt{\frac{2^{|V|}}{\binom{\omega(G)}{\omega(G)}}}\right)$, that is $O(\sqrt{2^{|V|}})$.

ω(G). Iteration - algorithm could not find a clique the size of which is larger than *ω(G)*. In general, the complexity of this iteration is $O(\sqrt{2^{|V|}})$.

We can see that the worst case scenario is when the value of a searched clique's size, which is stored in the variable k, is incremented by only 1, after each algorithm's iteration. The most complex iteration is the last one because there isn't a graph's clique which is larger than clique number *ω(G)* and complexity of last iteration is $O(\sqrt{2^{|V|}})$. If there is a maximum of *ω(G)* iterations and *ω(G)* can maximally be |*V*|, and if the maximum complexity of one algorithm's iteration is $O(\sqrt{2^{|V|}})$, we can conclude that the complexity for the worst case scenario is $O(|V|\sqrt{2^{|V|}})$.

It is also important to emphasize that the probability of the worst case scenario is at least:

$$P_{wc} \geq \frac{\binom{\omega(G)}{2}}{\binom{\omega(G)}{2}+..+\binom{\omega(G)}{\omega(G)}} *..* \frac{\binom{\omega(G)}{\omega(G)}}{\binom{\omega(G)}{\omega(G)}} = \sum_{i=2}^{\omega(G)} \binom{\omega(G)}{i} \left(\sum_{j=i}^{\omega(G)} \binom{\omega(G)}{j} \right)^{-1}$$

For the graph shown in Figure 3, the probability of the worst case scenario is $P_{wc} \geq \frac{6}{11} * \frac{4}{5} * 1 = \frac{24}{55}$.

Best case scenario is when a graph does not have a clique. Then the algorithm will terminate after only one iteration of the complexity $O(\sqrt{2^{|V|}})$, but if a graph has at least one clique, then the probability for the best case scenario is:

$$P_{bc} \geq \frac{1}{\binom{\omega(G)}{2}+..+\binom{\omega(G)}{\omega(G)}} = \left(\sum_{i=2}^{\omega(G} \binom{\omega(G)}{i} \right)^{-1}$$

For the graph shown in Figure 3, the probability for the best case scenario is $P_{bc} \geq \frac{1}{6+4+1} = \frac{1}{11}$.

It should be emphasized that the time complexity of algorithm step 5 is not relevant for the worst and best case scenario analysis because we can determine the size of the subset and verify whether a subset of a set of graph's vertices is a clique in polynomial time of |V| even on a classical computer.

5. Conclusion and Future Work

The new quantum algorithm for finding a maximum clique in an arbitrary undirected graph has been introduced. It is based on a version of Grover's algorithm for finding an element in unsorted list when the exact number of possible solutions is unknown. The proposed quantum algorithm has the same time complexity in worst case as quantum algorithm proposed by Long Chang, Ting-Ting Ren, Mang Feng, Jun Luo, Kawuu Weicheng Lin, Minyi Guo and Lai Chin Lu, that is $O(|V|\sqrt{2^{|V|}})$. Proposed quantum algorithm has worse time complexity in worst case than algorithm proposed by Andrew M. Childs, Edward Farhi, Jeffrey Goldstone and Sam Gutmann, but only for the case where n <=18. It is obvious that a proposed algorithm cannot be improved significantly (i.e. to achieve the polynomial time complexity in worst case) because the general idea of an algorithm is to find the subset of a set of graph's vertices using Grover's algorithm which has $O(\sqrt{N})$ time complexity where N is the number of the subset of a set of graph vertices, that is $N = 2^{|V|}$. Despite this fact, there are some possible improvements. One of the ideas for algorithm improvement is to guess the searched clique size in algorithm iteration by using binary search logic. The mentioned trick could improve the time complexity of the worst case scenario to $O\left((\log_2 |V|)\sqrt{2^{|V|}}\right)$. Also, one of the topics for future research is the calculus for the average case, and time and space analysis based on simulation of the proposed algorithm using the quantum computer simulator on the classical computer for randomly generated graphs. Finally, I conclude that the proposed algorithm isn't the best possible way to solve the maximum clique problem, but it describes a new approach and use of Grover's algorithm in finding the maximum clique in an arbitrary graph.

References

[1] Bennett, Charles H.; Bernstein, E.; Brassard, G.; Vazirani U. Strengths and Weaknesses of Quantum Computing. *SIAM Journal on Computing*, volume 26(5):1510-1523, 1997.

[2] Boyer, M.; Brassard, G.; Høyer, P.; Tapp, A. Tight bounds on quantum searching. *Fortschritte der Physik*, volume 46(4-5):493-505, 1998.

[3] Chang, W-L.;Ren, T-T.; Feng, M.; Luo, J.; Weicheng Lin, K.; Guo, M.; Chin Lu, L. Quantum Algorithms of Bio-molecular Solutions for the Clique Problem on a Quantum Computer, 2009, URL http://arxiv.org/ftp/arxiv/papers/0905/0905.2028.pdf

[4] Childs, A. M.; Farhi, E.; Goldstone, J.; Gutmann, S. Finding cliques by quantum adiabatic evolution, *Quantum Information and Computation 2*, pages 181 – 191, 2002

[5] Divjak, B.; Lovrencic, A. *Diskretna matematika s teorijom grafova*, TIVA Tiskara Varaždin, 2005.

[6] Franco, R. Grover's algorithm and human memory, 2008, URL http://arxiv.org/pdf/0804.3294v1.pdf

[7] Grover, L.K. A fast quantum mechanical algorithm for database search. In *28th Annual ACM Symposium on the Theory of Computing*, pages 212 – 219, Philadelphia, USA, 1996.

[8] Kaye, P.; Laflamme, R.; Mosca, M. *An Introduction to Quantum Computing*, Oxford University Press, USA, 2007.

[9] Lavor, C.; Liberti, L.; Maculan, N. Grover's algorithm applied to the molecular distance geometry problem, In *Proceedings of VII Brazilian Congress of Neural Networks,* 2005.

[10] Mateus, P.; Omar, Y. Quantum Algorithm for Closest Pattern Matching, *Quantum Information Processing*, IOS Press, 2006.

[11] Nielsen, Michael A.; Chuang, Isaac L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*, Cambridge University Press, New York, 2011.

[12] Pittenger, Arthur O. *An Introduction to Quantum Computing Algorithms*, Birkhäuser Boston, 1999.

[13] Shor, P. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Journal on Computing*, Volume 26(5), pages 1484-1509, 1997.

[14] Udrescu, M.; Prodan, L.; Vladutiu, M. Implementing Quantum Genetic Algorithms: A Solution Based on Grover's Algorithm. In *Proceedings of the 3rd Conference on Computing Frontiers*, pages 71 – 82, 2006

[15] Ventura, D.; Martinez, T. A quantum associative memory based on Grover's algorithm, In *Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms*, pages 22 – 27, 1999.

[16] Zalka, C. Grover’s quantum searching algorithm is optimal. *Phys. Rev. A*, volume 60(4):2746–2751, 1999.