QMClique: A Proposal and Questions for Quantum Maximal Clique Algorithm

Gustavo C. G. van Erven*

August 25, 2016

Abstract

Finding cliques in graphs is well know computing problem. These notes present an algorithm for maximal cliques and a research question for its implementation in a quantum computer.

1 Definitions

A Graph $G(V, E)$ is a set of vertices $V = \{v_1, v_2, \ldots, v_n\}$ with set E of relations between them, $E = \{e_1, e_2, \ldots, e_m\}$ and $e_i = (u, v)|u, v \in V$.

A Clique is a subgraph of $G(V, E)$, which all vertices are connected to each other.

Theorem: Let W_i be the set of all adjacent vertices for vertex i, union with i, $W_i = V_{adj(i)} \cup \{v_i\}$, if C is a maximal clique of size k:

$$C = \{\{v_i\} \cup \{v_{i+1}\} \cup \cdots \cup \{v_k\}\} = \{W_i \cap W_{i+1} \cap \cdots \cap W_k\} \tag{1}$$

The right side of Equation 1 has all vertices in the clique C. Therefore, they must be present in the left side, because all vertex in a clique are connected each other. The intersection have only clique's vertex, because they are the elements in commune between all of them. If a some vertex is in the left side and not in the union, the vertex will be a common vertex, and the right side is not complete.

If there is a vertex in the union that are not in the maximal clique, it will not be present in the intersection, because it will not be connected to all set's elements.

*gustavo.erven@cgu.gov.br

2 Checking a maximal clique

Let $H_i = V_{adj(i)} \cup \{v_i\}$, we can represent it as a vector, such as shown in Equation 2, where e_i is equal to 1 if the vertex i is connected to the node in the index position.

$$W_i = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} \tag{2}$$

Therefore, the set with all vectors H_i is equivalent to the adjacent matrix in below.

$$W = \begin{bmatrix} 1 & e_{1,2} & \cdots & e_{1,n} \\ e_{2,1} & 1 & \cdots & e_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{n,1} & e_{n,2} & \cdots & 1 \end{bmatrix} \tag{3}$$

Let A be a vector represents the presence of the vertex v_i when a_i is set to 1 or 0 otherwise, such that:

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \tag{4}$$

And A_i a vector composed by n elements $a_i \in A$:

$$A_i = \begin{bmatrix} a_i \\ a_i \\ \vdots \\ a_i \end{bmatrix} \tag{5}$$

According with Equation 1, if a vector A is given, we can check if it is a maximal clique verifying if the intersection of vectors W_i are equal to A. The intersection can be calculated with AND ports.

$$\text{A is a maximal clique iff is equal to } W_i \wedge W_{i+1} \wedge \cdots \wedge W_k \tag{6}$$

Now, we can compute a true value which represents if the input vector A is a maximal clique. At first, if a_i is present, the vector W_i must be used. Otherwise, the presence of the vertex must be ignores. We can ignore

the vector change all the values to 1, then it will not influence the AND operations. The Equation 9 shows an example for a W_i vector.

$$\neg A_i \vee (A_i \wedge W_i) \tag{7}$$
$$(\neg A_i \vee A_i) \wedge (\neg A_i \vee W_i) \tag{8}$$
$$\neg A_i \vee W_i \tag{9}$$

Therefore, if a_1 is 0, all the values in W_i will be changed to 1, and it will not affect the AND operations. Now we can generalize to check a input A over all vectors in W. Let O be the output vector for the AND operations, thus:

$$O = (\neg A_1 \vee W_1) \wedge (\neg A_2 \vee W_2) \wedge \cdots \wedge (\neg A_n \vee W_n) \tag{10}$$

Therefore, O_i is the result for element a_i, where w_{ij} is the element j for the vector W_i:

$$O_i = (\neg a_1 \vee w_{11} \wedge (\neg a_2 \vee w_{21}) \wedge \cdots \wedge (\neg a_n \vee w_{n1}) \tag{11}$$

Now, A will be a clique if the output O is equal to A. We can summarize it through a single value. Let R be a boolean variable, such that:

$$U = (O \oplus \neg A) \tag{12}$$
$$R = u_1 \wedge u_2 \wedge \cdots \wedge u_n \tag{13}$$
$$R = (o_1 \oplus \neg a_1) \wedge (o_2 \oplus \neg a_2) \wedge \cdots \wedge (o_n \oplus \neg a_n) \tag{14}$$

Therefore, if R is true, the vector input A is a maximal Clique. Note that, if A is a subset of maximal Clique, R will be false because all vertices with $a_i = 0$ will have their values changes to 1. Therefore, if some node i is not present, but is adjacent to the selected vertices in A, the output value will be true for O_i, and the final verification will return false, such that the $O_i = 1$ and $a_i = 0$.

If the algorithm is implement in a program, the time complexity will be $O(n^2)$, since we need n operation by each n vertex to generate the O vector, and more n operations to calculate R.

3 Using Quantum ports

The algorithm can be implemented in a quantum computer using the Toffoli gate. The Equation 9 can be modified to make the implementation easier, as shown in Equation 15.

$$\neg A_i \vee W_i = 1 \oplus A_i W_i \tag{15}$$

The AND port can be implemented with a Toffoli gate as present in Equation 16

$$A \wedge B = 0 \oplus AB \tag{16}$$

These gates can replace the boolean operations in the maximal clique algorithm. A Hadamard gate can be used in the input vector A to create a superposition with all possible values.

The Figure 3 presents an implementation for a two vertices graph without edges. The qubits are:

- Q1 and Q2: A;
- Q3 and Q4: W_1;
- Q5 and Q6: W_2;
- Q7: $1 \oplus a_1 w_{11}$;
- Q8: $1 \oplus a_2 w_{21}$;
- Q9: $1 \oplus a_1 w_{21}$;
- Q10: $1 \oplus a_2 w_{22}$;
- Q11: o_1;
- Q12: o_2;
- Q13: R;
- Q14: Qubit set to $|1\rangle$. It is used to change the R state to $|1\rangle$ and verify if it force the collapse of A to a maximal clique state;

The W matrix for this example is $\left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ and must be true only for $\left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right)$ inputs.

Note that will be necessary n qubits for A, n^2 qubits for W, n^2 qubits to keep the $(1 \oplus A_i W_i)$ operations, n qubits for the O vectors and one to the result R.

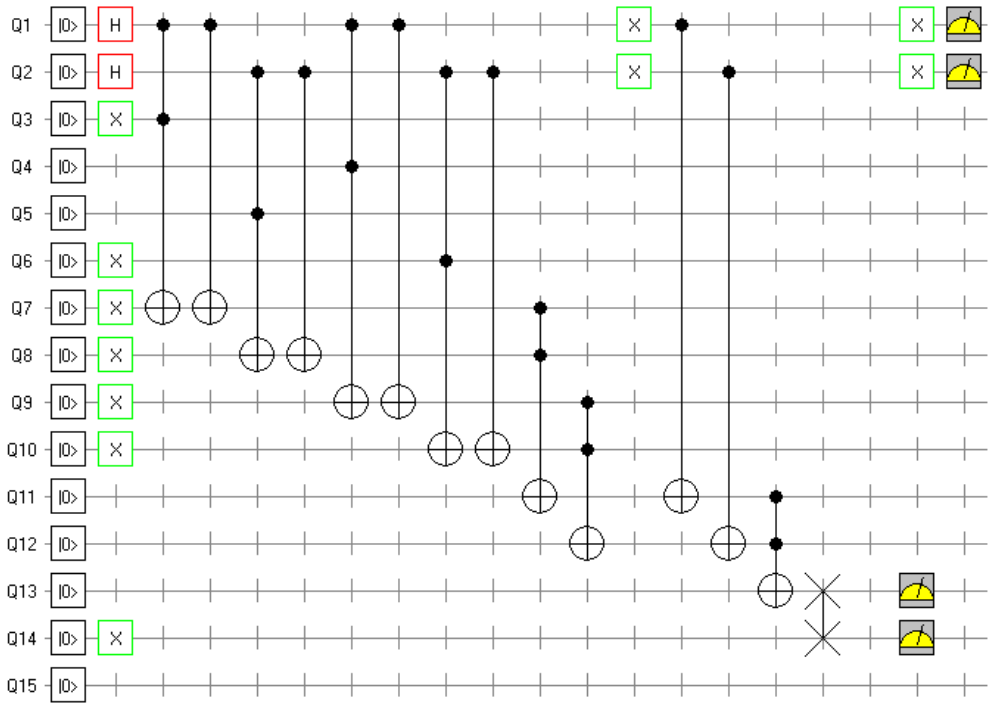

Figure 1: Quantum circuit for two vertices graph.

4 Conclusions and Question

The algorithm to verify a maximal clique can be implemented in a quantum computer using Toffoli gates. The Hadamard gate can be used to create a superposition in the input vector A. However, we are only interested in values that makes R true, which implies that in a maximal clique. **The question is: is there an entangle state that we can use to force the result R be true and makes A to collapse to a maximal clique? Could the swap gate X do this? And if it exists, is there a way to take only the maximum cliques superpositions?**