Project 5

Exhaustive search of Q-circuits

GOAL: Make a well structured report on the results obtained by an exhaustive search

In this project you have to use provided software to make a well structured search for all possibilities in the representation of a circuit. You will have a set of gates to use and a set of gates to find. It is on you to decide how to restrain the set of starting gates however the number of the used gates should not be smaller than 12. In this approach we consider few parameters. First you should decide on the maximum number of gates used in your search. We advise to use a maximum of 7 gates, more could result in a computation taking more than a week. However this depends strongly on the computer you will be using. The size of the circuit is also the maximum number of gates that can be used in a circuit. This is because we want to limit the exponential time required by this software.

The second part of this work is to make a structured report of the results. You will have to create a table or any other appropriate entity to represent the data and classify them according to these criterions:

- Classification by the same gate
- Classification on some internal criteria of the circuit such as complexity or cost (you can here assume that these two categories are the same)

The first type of the proposed classification implies creating of a branching tree where at the top you should put the cheapest gate of the type. The branches should be ordered either by the type of gates used or by another criterion that you define.

The second type of classification is based on a more linear representation. You should begin either with the most complex or expensive circuit and order the others in the gradient of decreasing price/complexity. You can branch in order to construct a tree where each branch corresponds to a different type of gate or where each branch is only a specified subset of gates. As an example, assume you want to classify all occurrences of Fredkin gate and you have found a lot of them. You want to classify them according to the different gates that occur in the gate. Begin with one selected gate and then make branches where each branch will either contain particular gates from the intital gate or where each branch has only one such gate. Proceed the next level similarly, until you arrive to a point where no more branching is possible.

The last type of classification requires also another setting of the algorithm. You will decide what gates you will be using for this search and what the maximum size of the circuit is. You will generate all possible gates and sort them according to some criteria you will choose. For example if you have three final gates such as Fredkin, Toffoli and Margolus you will take all generated gates (their number is very high) and group them together by groups. These groups will be created according to the hamming distance from the final gate. Example: imagine you have one generated gate and one final gate.

$$\text{Generated gate}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\quad\text{and final gate}\,\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

$$\text{the Hamming distance of these 2 gates is 4}$$

The figure above illustrates the methodology to evaluate two matrices. To have a Hamming distance we evaluate each element with each one but including the distance parameters. This means that all coefficients in the matrix above have distance parameter 1 because the maximal distance from one element to another one is 1. For bigger matrices this will change:

$$\text{Generated gate}\begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 0\end{bmatrix}\quad\text{and final gate}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$$

$$t\text{he Hamming distance of these 2 gates is vertically } 3+1+1+1=6$$

$$\text{or for this pair of matrices}$$

$$\text{Generated gate}\begin{bmatrix}0 & -0.5 & 0 & 0.5\\ 0.1 & 0.3 & -0.4 & 0.2\\ 0.5 & 0 & -0.5 & 0\\ 0.8 & 0 & -0.1 & 0.1\end{bmatrix}\quad\text{and final gate}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$$

$$t\text{he distance is horizontally } (1+3)+(2+1+1)+(3+1)+(2+1)=15$$

Two types of distances are presented here. First type is for permutation matrices and compares coefficients by columns, or called vertical Hamming distance. The second presented measure is called horizontal Hamming distance and calculates the distance of all different coefficients from the desired place.

Using here described methods and techniques you should be able to generate a big number of gates and then sort them. Most of these actions are done via the software program provided to you however you might be required to modify it slightly in order it satisfy your needs and demands.

Expected Results:

- Well structured report on gates you found
- Clear explanation of classification you used
- What were your ideas while selecting your parameters
- Extrapolation of your results in the form (how could the space look like if we add more segments to the circuit, and so on)