Qudit synthesis

Luke Heyfron

13/02/2018

Contents

1 The Gate Synthesis Problem

We are interested in synthesizing qudit unitaries of the form,

$$U_f \left|\mathbf{x}\right\rangle = \omega^{f(\mathbf{x})} \left|\mathbf{x}\right\rangle, \tag{1}$$

where d is an odd prime, $\omega = e^{i\frac{2\pi}{d}}$ is the d^{th} primitive root of unity; $\mathbf{x} \in \mathbb{Z}_d^n$ are binary strings that enumerate n qudit computational basis states and $f : \mathbb{Z}_d^n \mapsto \mathbb{Z}_d$ is known as the *phase-polynomial* of U_f.

When f is a homogeneous cubic polynomial, it can be shown that U_f belongs in the third level of the Clifford hierarchy and can be generated by the single qudit gates $M^k : \left|x\right\rangle \to \omega^{kx^3} \left|x\right\rangle$ and $S : \left|x\right\rangle \to \left|ax\right\rangle$ where a is a primitive root mod d; and the two qudit entangling gate $SUM_{c,t} : \left|c,t\right\rangle \to \left|c, t+c\right\rangle$.

As f is a cubic form, it can be decomposed as a sum over a set of linear forms raised to the 3^{rd} power,

$$f(\mathbf{x}) = \sum_{t=1}^{r} \lambda_t (A_{1,t}x_1 + A_{2,t}x_2 + \cdots + A_{n,t}x_n)^3, \tag{2}$$

where $\lambda \in \mathbb{Z}_d^r$ and $A \in \mathbb{Z}_d^n \times \mathbb{Z}_d^r$.

Definition 1. *A decomposition for a phase polynomial f as in equation* (2) *is referred to as an* **implementation of f of order** r.

Lemma 1. *Given a U_f and an implementation of f or order r, it follows that one can generate a quantum circuit that implements U_f using no more than r M^k gates.*

The function f can also be written in the monomial basis,

$$f(\mathbf{x}) = \sum_{i,j,k=1}^{n} c_{i,j,k} x_i x_j x_k, \tag{3}$$

with coefficients $c_{i,j,k} \in \mathbb{Z}_d$. By setting equations (2) equal to (3) we write c in terms of λ and A,

$$\begin{aligned} f(\mathbf{x}) &= \sum_{t=1}^{r} \lambda_t (A_{1,t}x_1 + A_{2,t}x_2 + \cdots + A_{n,t}x_n)^3 & (4)\\ &= \sum_{t=1}^{r} \lambda_t \sum_{i,j,k=1}^{n} A_{i,t}x_i A_{j,t}x_j A_{k,t}x_k & (5)\\ &= \sum_{i,j,k=1}^{n} \left(\sum_{t=1}^{r} \lambda_t A_{i,t}A_{j,t}A_{k,t} \right) x_i x_j x_k, & (6) \end{aligned}$$

therefore,

$$c_{i,j,k} = \sum_{t=1}^{r} \lambda_t A_{i,t}A_{j,t}A_{k,t}, \tag{7}$$

From equation (7) we see that $c_{i,j,k}$ is symmetric with respect to index permutations and is in fact a symmetric tensor of order 3. Equation (2) corresponds to a *symmetric tensor decomposition* and the minimum r is the *symmetric tensor rank* of $c_{i,j,k}$. Therefore, the n qudit gate synthesis problem is equivalent to finding the symmetric tensor rank of an order 3 symmetric tensor on the field $\mathbb{Z}_d$ with n variables.

2 Solutions

2.1 Naive Decoder

Each monomial in equation (3) can be rewritten in phase polynomial form using the following

$$\begin{aligned} x_i^3 \rightarrow & x_i^3 \\ x_i x_j^2 \rightarrow & 6^{-1}(x_i + x_j)^3 + 6^{-1}(x_i - x_j)^3 - 3^{-1}x_i^3 \\ x_i x_j x_k \rightarrow & 6^{-1}x_i^3 + 6^{-1}x_j^3 + 6^{-1}x_k^3 + 6^{-1}(x_i + x_j + x_k)^3 \\ & - 6^{-1}(x_i + x_j)^3 - 6^{-1}(x_i + x_k)^3 - 6^{-1}(x_j + x_k)^3 \end{aligned} \tag{8}$$

As there are $\mathcal{O}(n^3)$ cubic monomials and each rewrite features a constant number of terms, this so called *naive* decoder executes in polynomial time, specifically scaling as $\mathcal{O}(n^3)$. This will lead to a gate synthesis matrix where every column has support on no more than 3 rows. The number of such columns is $\mathcal{O}(n^3)$. Let $i \neq j \neq k$ be the supported rows. There are d^3 columns of this form leading to a worst case total number of columns of $\mathcal{O}(n^3 d^3)$.

2.2 Duplicate and Merge

One approach is to try and 'merge' columns. A pair of columns can be merged if they are duplicates of one another. This is because we can collect terms in the phase polynomial where the coefficients combine linearly. An illustrative example is the following. Let f be a phase polynomial with two terms, hence has gate synthesis matrix A with two columns,

$$f(\mathbf{x}) = \lambda_1(A_{1,1}x_1 + A_{2,1}x_2 + \cdots + A_{n,1}x_n)^3 \tag{9}$$

$$+ \lambda_2(A_{1,2}x_1 + A_{2,2}x_2 + \cdots + A_{n,2}x_n)^3 \tag{10}$$

if the two columns of A are duplicates, then we have $A_{i,1} = A_{i,2}\ \forall\ i \in [1, n]$. And so

$$f(\mathbf{x}) = (\lambda_1 + \lambda_2)(A_{1,1}x_1 + A_{2,1}x_2 + \cdots + A_{n,1}x_n)^3, \tag{11}$$

which needs only a single column to represent it, and therefore only a single magic state to implement it.

Merging columns can only be done where duplicate columns exist. However, is it possible to transform the matrix in some way in order to increase the number of duplicates? If so, can we do so without altering the function f it implements? The answers to these questions is yes. Yes. Let us define the following transformation.

Definition 2. ***Duplication Transformation.*** *Let $A \in \mathbb{Z}_d^{(n,m)}$ be a gate synthesis matrix and $\mathbf{y} \in \mathbb{Z}_d^m$, $\mathbf{z} = \mathbf{c}_b - \mathbf{c}_a \in \mathbb{Z}_d^n$ where $\mathbf{c}_j$ is the j^{th} column of A. The duplication transformation is defined as the following.*

$$A \to A + \mathbf{z}\mathbf{y}^T. \tag{12}$$

We can use this transformation to 'create' duplicates as the following lemma shows.

Lemma 2. *Let $A' = A + \mathbf{z}\mathbf{y}^T$ and $y_a - y_b = 1$. It follows that $\mathbf{c}'_a = \mathbf{c}'_b$.*

Proof. From the definition of A',

$$A'_{i,j} = A_{i,j} + z_i y_j, \tag{13}$$

now substitute in $z_i \equiv A_{i,b} - A_{i,a}$,

$$A'_{i,j} = A_{i,j} + (A_{i,b} - A_{i,a})y_j. \tag{14}$$

Apply equation (14) to both $(\mathbf{c}_a)_i \equiv A'_{i,a}$ and $(\mathbf{c}_b)_i \equiv A'_{i,b}$,

$$A'_{i,b} = A_{i,b} + (A_{i,b} - A_{i,a})y_b, \tag{15}$$

$$A'_{i,a} = A_{i,a} + (A_{i,b} - A_{i,a})y_a. \tag{16}$$

Substitute $y_a = y_b + 1$ into equation (16) and rearrange,

$$\begin{aligned} A'_{i,a} &= A_{i,a} + (A_{i,b} - A_{i,a})(y_b + 1) && (17)\\ &= A_{i,a} + (A_{i,b} - A_{i,a})y_b + A_{i,b} - A_{i,a} && (18)\\ &= A_{i,b} + (A_{i,b} - A_{i,a})y_b = A'_{i,b}. && (19) \end{aligned}$$

If $A'_{i,a} = A'_{i,b} \;\forall\; i \in [1, n]$, it follows that $\mathbf{c}'_a = \mathbf{c}'_b$. □

We must make sure the duplication transformation does not alter f. This leads to the condition that $c'_{i,j,k} = c_{i,j,k}$.

$$\begin{aligned} c'_{i,j,k} &= \sum_{t=1}^{r} \lambda_t A'_{i,t} A'_{j,t} A'_{k,t} && (20)\\ &= \sum_{t=1}^{r} \lambda_t (A_{i,t} + z_i y_t)(A_{j,t} + z_j y_t)(A_{k,t} + z_k y_t) && (21)\\ &= \sum_{t=1}^{r} \lambda_t A_{i,t} A_{j,t} A_{k,t} + \Delta_{i,j,k} = c_{i,j,k} + \Delta_{i,j,k}, && (22) \end{aligned}$$

where we define,

$$\begin{aligned} \Delta_{i,j,k} = \sum_{t=1}^{r} \lambda_t (&A_{i,t} A_{j,t} z_k y_t + A_{j,t} A_{k,t} z_i y_t + A_{k,t} A_{i,t} z_j y_t \\ &+ A_{i,t} z_j z_k y_t^2 + A_{j,t} z_k z_i y_t^2 + A_{k,t} z_i z_j y_t^2 + z_i z_j z_k y_t^3). \end{aligned} \tag{23}$$

In order for $c' = c$, we require that $\Delta_{i,j,k} = 0 \;\forall\; i, j, k \in [1, n]$. This leads to a system of $\sum_{a+b+c=3} \binom{n}{a,b,c}$ cubic polynomials on r variables, the $y_1, y_2, \ldots, y_r$, for which we wish to find the roots.

7^{th} March 2018
Lets rewrite equation(s) (23) as follows and include the duplication transformation condition from lemma (2):

$$\sum_{t=1}^{r} l_{\alpha,t} y_t + \sum_{t=1}^{r} q_{\alpha,t} y_t^2 + \sum_{t=1}^{r} c_{\alpha,t} y_t^3 = 0, \tag{24}$$

$$y_a - y_b - 1 = 0 \tag{25}$$

where the equation with indices (i,j,k) is labelled by single index α and a and b are fixed. The linear, cubic and quadratic coefficients for variable t for equation α are given by,

$$
\begin{aligned}
l_{\alpha,t} &= \lambda_t(A_{i,t}A_{j,t}z_k + A_{j,t}A_{k,t}z_i + A_{k,t}A_{i,t}z_j) && (26)\\
q_{\alpha,t} &= \lambda_t(A_{i,t}z_jz_k + A_{j,t}z_kz_i + A_{k,t}z_iz_j) && (27)\\
c_{\alpha,t} &= \lambda_t z_iz_jz_k. && (28)
\end{aligned}
$$

which can be rewritten solely in terms of λ and A:

$$
\begin{aligned}
l_{\alpha,t} &= \lambda_t(A_{i,t}A_{j,t}(A_{k,b}-A_{k,a})\\
&\quad + A_{j,t}A_{k,t}(A_{i,b}-A_{i,a}) + A_{k,t}A_{i,t}(A_{j,b}-A_{j,a}))\\
q_{\alpha,t} &= \lambda_t(A_{i,t}(A_{j,b}-A_{j,a})(A_{k,b}-A_{k,a}) && (29)\\
&\quad + A_{j,t}(A_{k,b}-A_{k,a})(A_{i,b}-A_{i,a}) + A_{k,t}(A_{i,b}-A_{i,a})(A_{j,b}-A_{j,a}))\\
c_{\alpha,t} &= \lambda_t(A_{i,b}-A_{i,a})(A_{j,b}-A_{j,a})(A_{k,b}-A_{k,a}).
\end{aligned}
$$

The affine variety of equations (24) and (25) is the set of $\mathbf{y}$ vectors that definitely lead to a reduction in magic states via the duplication transformation. The variety can be found in $\mathcal{O}(r^3d^r)$ evaluations of polynomials of the form (24) by an exhaustive search.
End 7^{th} March 2018

2.3 Codespace search

Consider the monomials on n variables,

$$m_{\mathbf{r}}(\mathbf{u}) = \prod_{i=1}^{n} u_i^{r_i}, \tag{30}$$

where $\mathbf{r} \in \mathbb{Z}^n$.

Definition 3. *A vector $a(\mathbf{u})$ is in $\mathcal{A}$ if*

$$f_a(\mathbf{x}) = \sum_{\mathbf{u}} a_{\mathbf{u}} \sum_{i=1}^{n} u_ix_i = 0 \tag{31}$$

for all $\mathbf{x} \in \mathbb{F}_d^n$.

Let $H \subset \mathbb{F}_d^n$ be the union of these three sets,

$$
\begin{aligned}
H_1 &= \{\sigma(d-4, d-1, d-1, \ldots, d-1)\} && (32)\\
H_2 &= \{\sigma(d-3, d-2, d-1, d-1, \ldots, d-1)\} && (33)\\
H_3 &= \{\sigma(d-2, d-2, d-2, d-1, d-1, \ldots, d-1)\} && (34)
\end{aligned}
$$

for all $\sigma \in S_n$ and let $G = \mathbb{F}_d^n \setminus H$.

Lemma 3. *$m_{\mathbf{r}} \in \mathcal{A}$ if $\mathbf{r} \in G$.*

It is trivial to see that $|H_1| = n$, $|H_2| = n(n-1)$ and $|H_3| = n(n-1)(n-2)/6$. Therefore, the number of monomials in $\mathcal{A}$ is at most

$$N_{\text{mon}} = d^n - n - n(n-1) - n(n-1)(n-2)/6 = \mathcal{O}(d^n) \tag{35}$$

and $|\mathcal{A}| = d^{N_{\text{mon}}}$. It follows that a brute force search over this space takes time $\mathcal{O}(d^{(d^n)})$.

3 Junk

The set of points $\{\mathbf{y}\}$ that satisfy all the above equations (and $y_a - y_b - 1 = 0$) is an affine variety. If this variety is non-empty, we can use any element of it to perform the duplication transformation.

Lets partition the terms in equation (23) like so,

$$\begin{aligned}\Delta_{i,j,k} = z_k \sum_{t=1}^{r} A_{i,t}A_{j,t}\lambda_t y_t + z_i \sum_{t=1}^{r} A_{j,t}A_{k,t}\lambda_t y_t + z_j \sum_{t=1}^{r} A_{k,t}A_{i,t}\lambda_t y_t \\ + z_j z_k \sum_{t=1}^{r} A_{i,t}\lambda_t y_t^2 + z_k z_i \sum_{t=1}^{r} A_{j,t}\lambda_t y_t^2 + +z_i z_j \sum_{t=1}^{r} A_{k,t}\lambda_t y_t^2 \\ + z_i z_j z_k \sum_{t=1}^{r} \lambda_t y_t^3,\end{aligned} \tag{36}$$

and define,

$$\mathbf{a} = \lambda \wedge \mathbf{y} \tag{37}$$

$$\mathbf{b} = \lambda \wedge \mathbf{y}^2 \tag{38}$$

$$\mathbf{c} = \lambda \wedge \mathbf{y}^3 \tag{39}$$

and the matrix formed by completing these rows,

$$B = \left(\ \mathbf{r}_i \wedge \mathbf{r}_j \ \right) \tag{40}$$

If we find vectors $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$ such that $B\mathbf{a} = \mathbf{0}$, and $A\mathbf{b} = \mathbf{0}$ and $|\mathbf{c}| = 0$ and $\mathbf{b} = \mathbf{a} \wedge \mathbf{y}$ and $\mathbf{c} = \mathbf{b} \wedge \mathbf{y}$, then $\Delta_{i,j,k} = 0 \ \forall \ i, j, k \in [1, n]$.

$$B\mathbf{a} = \left(\ \textstyle\sum_{t=1}^{r} A_{i,t}A_{j,t}\lambda_t y_t \ \right), \tag{41}$$

so if $B\mathbf{a}$ is the all-zero vector, then the first three terms in equation (23) are zero. Similar arguments hold for the other conditions.

Let say we have a set of vectors that form a linearly-independent basis for the solutions to $B\mathbf{a} = \mathbf{0}$. Let this set be $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{N_a}\}$. Similar for $A\mathbf{b} = \mathbf{0}$, we have $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{N_b}\}$. Any $\mathbf{a}$ vector can be decomposed as

$$\mathbf{a} = \sum_{i=1}^{N_a} \alpha_i \mathbf{a}_i, \tag{42}$$

similar for $\mathbf{b}$,

$$\mathbf{b} = \sum_{i=1}^{N_b} \beta_i \mathbf{b}_i, \tag{43}$$

so to find $\mathbf{b} = \mathbf{a} \wedge \mathbf{y}$, we substitute,

$$\sum_{j=1}^{N_b} \beta_j \mathbf{b}_j = \mathbf{a}_1 \wedge \mathbf{y}. \tag{44}$$