Optimization in quantum circuits of c-Z and SWAP gates

Marc Bataille

PhD student, LITIS (University of Rouen, France)

Besançon 28/11/2019

"Quantum circuits of c-Z and SWAP gates : optimization and entanglement"
Marc Bataille and Jean-Gabriel Luque.

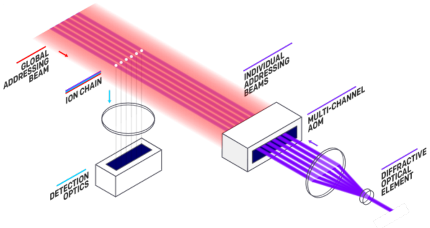
Published 15 July 2019.
Journal of Physics A : Mathematical and Theoretical, Volume 52, Number 32.

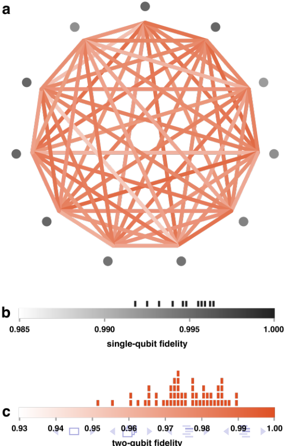
https://arxiv.org/abs/1810.01769

Example of quantum computer with a complete graph architecture

Trapped Ions quantum machine
(C. Monroe, University of Maryland)

Two qubits gates allowed on any pair of qubits.

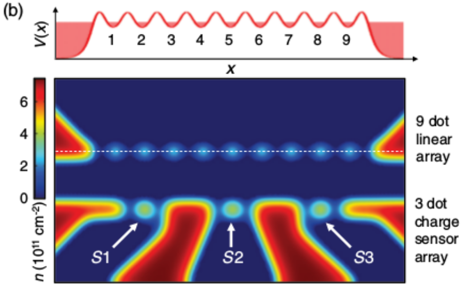




Example of quantum computer with the LNN graph architecture

LNN : Linear Nearest Neighbour

Only considers the interaction of adjacent qubits.



D. M. Zajac *et al.*, Scalable gate architecture for a one-dimensional array of semiconductor spin qubits. Phys. Rev. Applied, Nov 2016.

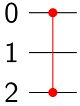
Plan

The $Z_{\{i,j\}}$ gates

$Z_{\{i,j\}}$: the cZ gate between qubit i and qubit j :

$$Z_{\{i,j\}}\,|x\rangle = Z_{\{i,j\}}\,|x_0x_1\ldots x_i\ldots x_j\ldots x_{n-1}\rangle = (-1)^{x_ix_j}\,|x\rangle$$

Example : $n = 3$ $\qquad Z_{\{0,2\}} \sim$



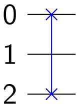
$$Z_{\{0,2\}}\begin{cases} |000\rangle \longrightarrow |000\rangle \\ |001\rangle \longrightarrow |001\rangle \\ |010\rangle \longrightarrow |010\rangle \\ |011\rangle \longrightarrow |011\rangle \\ |100\rangle \longrightarrow |100\rangle \\ |101\rangle \longrightarrow -|101\rangle \\ |110\rangle \longrightarrow |110\rangle \\ |111\rangle \longrightarrow -|111\rangle \end{cases} \qquad Z_{\{0,2\}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

The $S_{\{i,j\}}$ gates

$S_{\{i,j\}}$: the SWAP gate between qubit i and j.

$$\begin{aligned} S_{\{i,j\}} |x\rangle &= S_{\{i,j\}} |x_0 x_1 \dots {\color{red}x_i} \dots {\color{blue}x_j} \dots x_{n-1}\rangle \\ &= |x_0 x_1 \dots {\color{blue}x_j} \dots {\color{red}x_i} \dots x_{n-1}\rangle \end{aligned}$$

Example : $n = 3$ $\quad S_{\{0,2\}} \sim$



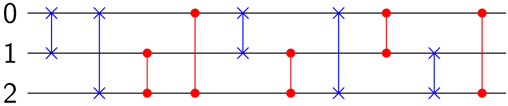
$$S_{\{0,2\}} \begin{cases} |000\rangle \longrightarrow |000\rangle \\ |001\rangle \longrightarrow |100\rangle \\ |010\rangle \longrightarrow |010\rangle \\ |011\rangle \longrightarrow |110\rangle \\ |100\rangle \longrightarrow |001\rangle \\ |101\rangle \longrightarrow |101\rangle \\ |110\rangle \longrightarrow |011\rangle \\ |111\rangle \longrightarrow |111\rangle \end{cases} \qquad S_{\{0,2\}} = \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix}$$

The group cZS_n

Definition

cZS_n is the group generated by the $Z_{\{i,j\}}$ gates and the $S_{\{i,j\}}$ gates for n qubits.

Example ($n = 3$) :



$C = Z_{\{0,2\}} S_{\{1,2\}} Z_{\{0,1\}} S_{\{0,2\}} Z_{\{1,2\}} S_{\{0,1\}} Z_{\{0,2\}} Z_{\{1,2\}} S_{\{0,2\}} S_{\{0,1\}}$

C is a circuit in cZS_3

The group $\mathcal{S}_n$

Definition

$\mathcal{S}_n$ is the subgroup of cZS_n generated by $S_{\{i,j\}}$ gates.

Proposition

$\mathcal{S}_n$ is isomorphic to the symetric group $\mathfrak{S}_n$.
Isomorphism : gate $S_{\{i,j\}} \longleftrightarrow$ transposition (i,j)

The group $\mathcal{S}_n$

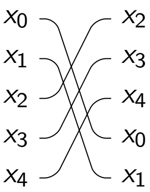
Definition

$\mathcal{S}_n$ is the subgroup of cZS_n generated by $S_{\{i,j\}}$ gates.

Proposition

$\mathcal{S}_n$ is isomorphic to the symetric group $\mathfrak{S}_n$.
Isomorphism : gate $S_{\{i,j\}} \longleftrightarrow$ transposition (i,j)

Example : cyclic permutation of 5 qubits

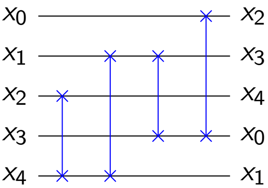


$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 4 & 0 & 1 & 2 \end{pmatrix}$$

$$\sigma = (0,3)(3,1)(1,4)(4,2)$$

Isomorphism : $\sigma \simeq S_\sigma$ with

$$S_\sigma = S_{\{0,3\}} S_{\{3,1\}} S_{\{1,4\}} S_{\{4,2\}}$$



Rule : $S_\sigma \left| x_0 x_1 \ldots x_{n-1} \right\rangle = \left| x_{\sigma^{-1}(0)} x_{\sigma^{-1}(1)} \cdots x_{\sigma^{-1}(n-1)} \right\rangle$

The group cZ_n

Definition

The group cZ_n is the subgroup of cZS_n generated by the $Z_{\{i,j\}}$ gates.

The group cZ_n

Definition

The group cZ_n is the subgroup of cZS_n generated by the $Z_{\{i,j\}}$ gates.

Proposition

The group cZ_n is a 2-elementary abelian group isomorphic to $\mathbb{Z}_2^{n(n-1)/2}$.

Example : $Z_{\{0,2\}}Z_{\{1,2\}}Z_{\{0,1\}}Z_{\{1,2\}}Z_{\{2,3\}}Z_{\{0,2\}} = Z_{\{0,1\}}Z_{\{2,3\}}$

The group cZ_n

Definition

The group cZ_n is the subgroup of cZS_n generated by the $Z_{\{i,j\}}$ gates.

Proposition

The group cZ_n is a 2-elementary abelian group isomorphic to $\mathbb{Z}_2^{n(n-1)/2}$.

Example : $Z_{\{0,2\}}Z_{\{1,2\}}Z_{\{0,1\}}Z_{\{1,2\}}Z_{\{2,3\}}Z_{\{0,2\}} = Z_{\{0,1\}}Z_{\{2,3\}}$

Proposition

Let $\mathcal{E}_n$ denote the set $\{\{i,j\} \mid 0 \leqslant i < j \leqslant n-1\}$.
cZ_n is isomorphic to the group $(P(\mathcal{E}_n), \oplus)$

Example :
$\{\{0,2\},\{1,2\},\{0,1\}\} \oplus \{\{1,2\},\{2,3\},\{0,2\}\} = \{\{0,1\},\{2,3\}\}$

Conjugation by SWAP gates

Example :

$$\begin{aligned} S_{\{0,2\}} Z_{\{2,3\}} S_{\{0,2\}} |x_0 x_1 x_2 x_3\rangle &= S_{\{0,2\}} Z_{\{2,3\}} |x_2 x_1 x_0 x_3\rangle \\ &= S_{\{0,2\}} (-1)^{x_2 x_3} |x_2 x_1 x_0 x_3\rangle \\ &= (-1)^{x_2 x_3} |x_0 x_1 x_2 x_3\rangle \end{aligned}$$

$$\implies S_{\{0,2\}} Z_{\{2,3\}} S_{\{0,2\}} = Z_{\{0,3\}}$$

Generalisation : conjugation by S_σ

$$S_\sigma Z_{\{i,j\}} S_\sigma^{-1} = Z_{\{\sigma(i),\sigma(j)\}}$$

Proposition

cZ_n is a normal subgroup of cZS_n.

Semi-direct product of two groups

Semi-direct product of two groups in general :

- Two groups N and H
- A morphism $\Phi : H \longrightarrow \mathrm{Aut}(N)$
- Underlying set : cartesian product $N \times H$
- Group opération : $(n_1, h_1) \times (n_2, h_2) = (n_1 \Phi(h_1)(n_2), h_1 h_2)$
- Semi-direct product denoted by $N \rtimes_\Phi H$

Semi-direct product of two groups

Semi-direct product of two groups in general :

- Two groups N and H
- A morphism $\Phi : H \longrightarrow \mathrm{Aut}(N)$
- Underlying set : cartesian product $N \times H$
- Group opération : $(n_1, h_1) \times (n_2, h_2) = (n_1 \Phi(h_1)(n_2), h_1 h_2)$
- Semi-direct product denoted by $N \rtimes_\Phi H$

cZS_n **as a semi-direct product of two groups :**

Semi-direct product of two groups

Semi-direct product of two groups in general :

- Two groups N and H
- A morphism $\Phi : H \longrightarrow \mathrm{Aut}(N)$
- Underlying set : cartesian product $N \times H$
- Group opération : $(n_1, h_1) \times (n_2, h_2) = (n_1 \Phi(h_1)(n_2), h_1 h_2)$
- Semi-direct product denoted by $N \rtimes_\Phi H$

cZS_n as a semi-direct product of two groups :

- Two groups $(P(\mathcal{E}_n), \oplus)$ and $\mathfrak{S}_n$

Semi-direct product of two groups

Semi-direct product of two groups in general :

- Two groups N and H
- A morphism $\Phi : H \longrightarrow \mathrm{Aut}(N)$
- Underlying set : cartesian product $N \times H$
- Group opération : $(n_1, h_1) \times (n_2, h_2) = (n_1 \Phi(h_1)(n_2), h_1 h_2)$
- Semi-direct product denoted by $N \rtimes_\Phi H$

cZS_n as a semi-direct product of two groups :

- Two groups $(P(\mathcal{E}_n), \oplus)$ and $\mathfrak{S}_n$
- A morphism $\Phi : \mathfrak{S}_n \longrightarrow \mathrm{Aut}((P(\mathcal{E}_n), \oplus))$
  $\Phi_\sigma(E) = \{\{\sigma(i), \sigma(j)\} \mid \{i, j\} \in E\}$ for any $E \in P(\mathcal{E}_n)$

Semi-direct product of two groups

Semi-direct product of two groups in general :

- Two groups N and H
- A morphism $\Phi : H \longrightarrow \mathrm{Aut}(N)$
- Underlying set : cartesian product $N \times H$
- Group opération : $(n_1, h_1) \times (n_2, h_2) = (n_1\Phi(h_1)(n_2), h_1 h_2)$
- Semi-direct product denoted by $N \rtimes_\Phi H$

cZS_n as a semi-direct product of two groups :

- Two groups $(P(\mathcal{E}_n), \oplus)$ and $\mathfrak{S}_n$
- A morphism $\Phi : \mathfrak{S}_n \longrightarrow \mathrm{Aut}((P(\mathcal{E}_n), \oplus))$
  $\Phi_\sigma(E) = \{\{\sigma(i), \sigma(j)\} \mid \{i, j\} \in E\}$ for any $E \in P(\mathcal{E}_n)$
- Underlying set : $P(\mathcal{E}_n) \times \mathfrak{S}_n$

Semi-direct product of two groups

Semi-direct product of two groups in general :

- Two groups N and H
- A morphism $\Phi : H \longrightarrow \mathrm{Aut}(N)$
- Underlying set : cartesian product $N \times H$
- Group opération : $(n_1, h_1) \times (n_2, h_2) = (n_1\Phi(h_1)(n_2), h_1 h_2)$
- Semi-direct product denoted by $N \rtimes_\Phi H$

cZS_n as a semi-direct product of two groups :

- Two groups $(P(\mathcal{E}_n), \oplus)$ and $\mathfrak{S}_n$
- A morphism $\Phi : \mathfrak{S}_n \longrightarrow \mathrm{Aut}((P(\mathcal{E}_n), \oplus))$
  $\Phi_\sigma(E) = \{\{\sigma(i), \sigma(j)\} \mid \{i, j\} \in E\}$ for any $E \in P(\mathcal{E}_n)$
- Underlying set : $P(\mathcal{E}_n) \times \mathfrak{S}_n$
- Group opération : $(E_1, \sigma_1) \times (E_2, \sigma_2) = (E_1 \oplus \Phi_{\sigma_1}(E_2), \sigma_1\sigma_2)$.

Semi-direct product of two groups

Semi-direct product of two groups in general :

- Two groups N and H
- A morphism $\Phi : H \longrightarrow \mathrm{Aut}(N)$
- Underlying set : cartesian product $N \times H$
- Group opération : $(n_1, h_1) \times (n_2, h_2) = (n_1 \Phi(h_1)(n_2), h_1 h_2)$
- Semi-direct product denoted by $N \rtimes_\Phi H$

cZS_n as a semi-direct product of two groups :

- Two groups $(P(\mathcal{E}_n), \oplus)$ and $\mathfrak{S}_n$
- A morphism $\Phi : \mathfrak{S}_n \longrightarrow \mathrm{Aut}((P(\mathcal{E}_n), \oplus))$
  $\Phi_\sigma(E) = \{\{\sigma(i), \sigma(j)\} \mid \{i, j\} \in E\}$ for any $E \in P(\mathcal{E}_n)$
- Underlying set : $P(\mathcal{E}_n) \times \mathfrak{S}_n$
- Group opération : $(E_1, \sigma_1) \times (E_2, \sigma_2) = (E_1 \oplus \Phi_{\sigma_1}(E_2), \sigma_1 \sigma_2)$.
- **Theorem :** $cZS_n \simeq P(\mathcal{E}_n) \rtimes_\Phi \mathfrak{S}_n$

Isomorphism $cZS_n \simeq P(\mathcal{E}_n) \rtimes_\Phi \mathfrak{S}_n$: example with 4 qubits

$\rightarrow$ In cZS_4

$$C = Z_{\{0,1\}} Z_{\{1,3\}} S_{\{0,3\}} Z_{\{2,3\}} Z_{\{0,1\}} S_{\{1,3\}}$$

$\rightarrow$ In $P(\mathcal{E}_4) \rtimes_\Phi \mathfrak{S}_4$

$$C \simeq (\{\{0,1\},\{1,3\}\},(0,3)) \times (\{\{2,3\},\{0,1\}\},(1,3))$$

Isomorphism $cZS_n \simeq P(\mathcal{E}_n) \rtimes_\Phi \mathfrak{S}_n$: example with 4 qubits

→ In cZS_4

$$C = Z_{\{0,1\}} Z_{\{1,3\}} S_{\{0,3\}} Z_{\{2,3\}} Z_{\{0,1\}} S_{\{1,3\}}$$

$$C = Z_{\{0,1\}} Z_{\{1,3\}} \underbrace{S_{\{0,3\}} Z_{\{2,3\}} Z_{\{0,1\}} S_{\{0,3\}}} S_{\{0,3\}} S_{\{1,3\}}$$

→ In $P(\mathcal{E}_4) \rtimes_\Phi \mathfrak{S}_4$

$$C \simeq (\{\{0,1\},\{1,3\}\},(0,3)) \times (\{\{2,3\},\{0,1\}\},(1,3))$$

$$\Phi_{(0,3)}(\{\{2,3\},\{0,1\}\}) = \{\{2,0\},\{3,1\}\}$$

Isomorphism $cZS_n \simeq P(\mathcal{E}_n) \rtimes_\Phi \mathfrak{S}_n$: example with 4 qubits

→ In cZS_4

$$C = Z_{\{0,1\}} Z_{\{1,3\}} S_{\{0,3\}} Z_{\{2,3\}} Z_{\{0,1\}} S_{\{1,3\}}$$

$$C = Z_{\{0,1\}} Z_{\{1,3\}} \underbrace{S_{\{0,3\}} Z_{\{2,3\}} Z_{\{0,1\}} S_{\{0,3\}}} S_{\{0,3\}} S_{\{1,3\}}$$

$$C = Z_{\{0,1\}} Z_{\{1,3\}} \quad Z_{\{2,0\}} Z_{\{3,1\}} \quad S_{\{0,3\}} S_{\{1,3\}}$$

→ In $P(\mathcal{E}_4) \rtimes_\Phi \mathfrak{S}_4$

$$C \simeq (\{\{0,1\},\{1,3\}\},(0,3)) \times (\{\{2,3\},\{0,1\}\},(1,3))$$

$$\Phi_{(0,3)}(\{\{2,3\},\{0,1\}\}) = \{\{2,0\},\{3,1\}\}$$

$$C \simeq (\{\{0,1\},\{1,3\}\} \oplus \{\{2,0\},\{3,1\}\},(0,3)(1,3))$$

Isomorphism $cZS_n \simeq P(\mathcal{E}_n) \rtimes_\Phi \mathfrak{S}_n$: example with 4 qubits

→ In cZS_4

$$C = Z_{\{0,1\}} Z_{\{1,3\}} S_{\{0,3\}} Z_{\{2,3\}} Z_{\{0,1\}} S_{\{1,3\}}$$

$$C = Z_{\{0,1\}} Z_{\{1,3\}} \underbrace{S_{\{0,3\}} Z_{\{2,3\}} Z_{\{0,1\}} S_{\{0,3\}}} S_{\{0,3\}} S_{\{1,3\}}$$

$$C = Z_{\{0,1\}} Z_{\{1,3\}} \qquad Z_{\{2,0\}} Z_{\{3,1\}} \qquad S_{\{0,3\}} S_{\{1,3\}}$$

$$C = Z_{\{0,1\}} Z_{\{2,0\}} S_{\{0,3\}} S_{\{1,3\}}$$

→ In $P(\mathcal{E}_4) \rtimes_\Phi \mathfrak{S}_4$

$$C \simeq (\{\{0,1\},\{1,3\}\},(0,3)) \times (\{\{2,3\},\{0,1\}\},(1,3))$$

$$\Phi_{(0,3)}(\{\{2,3\},\{0,1\}\}) = \{\{2,0\},\{3,1\}\}$$

$$C \simeq (\{\{0,1\},\{1,3\}\} \oplus \{\{2,0\},\{3,1\}\},(0,3)(1,3))$$

$$C \simeq (\{\{0,1\},\{2,0\}\},(0,3)(1,3))$$

Optimization algorithm for the complete graph architecture

Consequences of the semi-direct product structure

1. Any element (circuit) of cZS_n may be written in a unique way as a product $Z_E S_\sigma$ with $Z_E \in cZ_n$ et $S_\sigma \in \mathcal{S}_n$
2. $|cZS_n| = 2^{n(n-1)/2} n!$

Optimization algorithm for the complete graph architecture

Consequences of the semi-direct product structure

1. Any element (circuit) of cZS_n may be written in a unique way as a product $Z_E S_\sigma$ with $Z_E \in cZ_n$ et $S_\sigma \in \mathcal{S}_n$
2. $|cZS_n| = 2^{n(n-1)/2} n!$

Algorithm CtoZS (cubic in the lenght of input)

Input : A circuit described as a sequence of gates $C = Z_{E_0}(S_{\sigma_1} Z_{E_1}) \cdots (S_{\sigma_{\ell-1}} Z_{E_{\ell-1}}) S_{\sigma_\ell}$ with $E_0, \ldots, E_\ell \subset \mathcal{E}_n$ and $\sigma_1, \ldots, \sigma_\ell \in \mathfrak{S}_n$.

Ouput : An equivalent description of the circuit under the form $Z_E S_\sigma$.

1. Compute $\sigma'_i = \sigma_1 \cdots \sigma_i$, for $i = 1 \ldots \ell$.
2. Compute $E'_i = E_0 \oplus \sigma'_1(E_1) \oplus \cdots \oplus \sigma'_i(E_i)$, for $i = 0 \ldots \ell - 1$.
3. Return $Z_{E'_{\ell-1}} S_{\sigma'_\ell}$.

1. The group cZS_n and the complete graph architecture
2. Conversion of a cZS_n circuit to the LNN architecture

LNN architecture

Changing the architecture $\implies$ changing the generators of the group

Complete graph architecture

Generators are the $Z_{\{i,j\}}$ and the $S_{\{i,j\}}$ gates

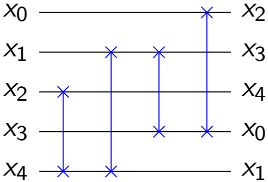
LNN architecture

Generators are the $Z_{\{i,i+1\}}$ and the $S_{\{i,i+1\}}$ gates

Notation :

$Z_i = Z_{\{i,i+1\}}$
$S_i = S_{\{i,i+1\}}$

Conversion of a $\mathcal{S}_n$ circuit to the LNN architecture

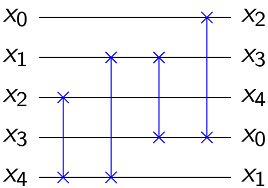


$$\sim \quad \sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 4 & 0 & 1 & 2 \end{pmatrix}$$

target architecture :

What's the minimal decomposition of σ in elementary transposition $s_i = (i, i+1)$?

Conversion of a $\mathcal{S}_n$ circuit to the LNN architecture



$$\sim \quad \sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 4 & 0 & 1 & 2 \end{pmatrix}$$

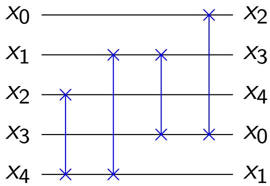
target architecture :

What's the minimal decomposition of σ in elementary transposition $s_i = (i, i+1)$?

Minimal length = number of inversion of $\sigma = 6$

Classical algorithm $\longrightarrow$ $\sigma = s_2 s_1 s_0 s_3 s_2 s_1$ then :

Conversion of a $\mathcal{S}_n$ circuit to the LNN architecture



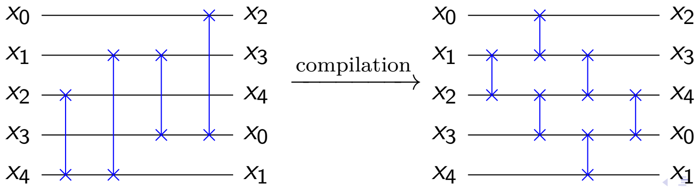
$$\sim \quad \sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 4 & 0 & 1 & 2 \end{pmatrix}$$

target architecture :

What's the minimal decomposition of σ in elementary transposition $s_i = (i, i+1)$?

Minimal length = number of inversion of $\sigma = 6$

Classical algorithm $\longrightarrow$ $\sigma = s_2 s_1 s_0 s_3 s_2 s_1$ then :



The (Coxeter + Dehn) heurisitic

The (Coxeter + Dehn) heurisitic

Main idea

Associating two classical algorithms of group theory to build a heuristic

- Reduction in the Coxeter groups
- Reduction using the Dehn algorithm

We need relations between symbols of the alphabet of generators

$\rightarrow$ find a presentation of the group cZS_n

A presentation of cZS_n

Theorem

Set of symbols : $\mathcal{G} = \{z_0, s_0, z_1, s_1 \ldots, z_{n-2}, s_{n-2}\}$

$cZS_n \simeq < \mathcal{G} \mid \mathcal{R} >$ with $\mathcal{R}$ containing the relations :

Coxeter relations : group $\mathcal{W}_n$:

$$\begin{cases} z_i^2 = s_i^2 = 1 \\ (s_i s_j)^2 = 1 \quad |i-j| \geqslant 2 \\ (s_i s_{i+1})^3 = 1 \\ (z_i z_j)^2 = 1 \\ (z_i s_i)^2 = 1 \\ (z_i s_j)^2 = 1 \quad |i-j| \geqslant 2 \\ (z_i s_j)^4 = 1 \quad |i-j| = 1 \end{cases}$$

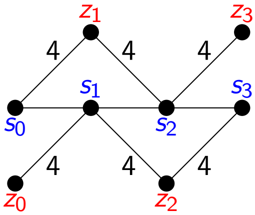
Non-Coxeter relations :

$$s_i s_{i+1} z_i s_{i+1} s_i z_{i+1} = 1$$

cZS_n is the quotient of the Coxeter group $\mathcal{W}_n$ by the relations $s_i s_{i+1} z_i s_{i+1} s_i z_{i+1} = 1$

Example : the Coxeter group $\mathcal{W}_5$

Coxeter Diagramm of $\mathcal{W}_5$



Coxeter Matrix of $\mathcal{W}_5$:
$M_5 = (m_{i,j})$ such that $(g_i g_j)^{m_{i,j}} = 1$

$$\begin{bmatrix} 1 & 2 & 2 & 4 & 2 & 2 & 2 & 2 \\ 2 & 1 & 4 & 3 & 2 & 2 & 2 & 2 \\ 2 & 4 & 1 & 2 & 2 & 4 & 2 & 2 \\ 4 & 3 & 2 & 1 & 4 & 3 & 2 & 2 \\ 2 & 2 & 2 & 4 & 1 & 2 & 2 & 4 \\ 2 & 2 & 4 & 3 & 2 & 1 & 4 & 3 \\ 2 & 2 & 2 & 2 & 2 & 4 & 1 & 2 \\ 2 & 2 & 2 & 2 & 4 & 3 & 2 & 1 \end{bmatrix}$$

The groupe cZS_5 is (isomorphic to) the quotient of the Coxeter group $\mathcal{W}_5$ by the relations :
$s_0 s_1 z_0 s_1 s_0 z_1 = s_1 s_2 z_1 s_2 s_1 z_2 = s_2 s_3 z_2 s_3 s_2 z_3 = 1.$

Reduction of a word in a Coxeter group

Input : a word w in the alphabet of generators.
Output : a word of minimal length that represents the same element of the group.

Reference :
Anders Björner and Francesco Brenti. Combinatorics of Coxeter Groups, Graduate Texts in Mathematics, 231. Springer, 2005.

Implemented in **SageMath** :
`http://www.sagemath.org/index.html` :

- Coxeter group given by his matrix.
- Method : `reduced_word()`

Reduction of a word using the Dehn Algorithm

Example : one step in Dehn algorithm.

$C = S_0 Z_1 S_0 S_1 Z_2 S_1 S_2 Z_3$ a circuit to reduce in cZS_5

Input word $w_0 = s_0 z_1 s_0 s_1 z_2 s_1 s_2 z_3$

Reduction of a word using the Dehn Algorithm

Example : one step in Dehn algorithm.

$C = S_0 Z_1 S_0 S_1 Z_2 S_1 S_2 Z_3$ a circuit to reduce in cZS_5

Input word $w_0 = s_0 z_1 s_0 s_1 z_2 s_1 s_2 z_3$

Relation $s_1 s_2 z_1 s_2 s_1 z_2 = 1 \xrightarrow{\text{cyclic permutation}} r = s_1 z_2 s_1 s_2 z_1 s_2 = 1$

Prefix of $r = s_1 z_2 s_1 s_2 z_1 s_2$ is factor of $w_0 = s_0 z_1 s_0 s_1 z_2 s_1 s_2 z_3$

Replace factor $s_1 z_2 s_1 s_2$ in w_0 by $(z_1 s_2)^{-1} = s_2 z_1$

$w_1 = s_0 z_1 s_0 s_2 z_1 z_3$: **end of the first step**

And so on, until stabilization

$C = S_0 Z_1 S_0 S_2 Z_1 Z_3$

Reduction of a word using the Dehn Algorithm

Algorithm

Input :

A set of relations $\mathcal{R}$ on the alphabet of generators.

A word w on the same alphabet.

Output : a word w' that represents the same element such that $|w'| \leqslant |w|$

Precomputation : Compute $\tilde{\mathcal{R}}$ the closure of $\mathcal{R}$ under cyclic permutation of the symbols and inverse.

While : There exists a factor u of w such that u is the prefix of a relation $r = uv$ in $\tilde{\mathcal{R}}$ with $|u| > |v|$

Replace u by v^{-1} in w

Reduce w in the free group.

End While

Return w

The (Coxeter + Dehn) heuristic

Algorithm (Coxeter + Dehn)

Input :

A set of relations $\mathcal{R}$ on the alphabet of generators.

A word w on the same alphabet.

Ouput : A word w' that represents the same element such that $|w'| \leqslant |w|$

Do :

$w \longleftarrow$ reductionCoxeter(w)

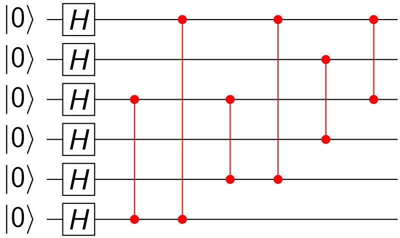
$w \longleftarrow$ reductionDehn(w)

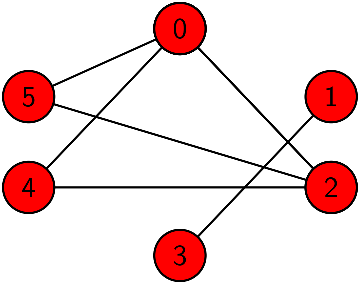
Until : Stabilisation of w

Return : w

Example : producing a graph state for a LNN architecture

$$|G\rangle = \underbrace{Z_{\{0,2\}}Z_{\{1,3\}}Z_{\{0,4\}}Z_{\{2,4\}}Z_{\{0,5\}}Z_{\{2,5\}}}_{\text{Circuit of } cZS_6}|+\rangle^{\otimes 6}$$





Producing a graph state for a LNN architecture

Circuit to compile : $C = Z_{\{0,2\}} Z_{\{1,3\}} Z_{\{0,4\}} Z_{\{2,4\}} Z_{\{0,5\}} Z_{\{2,5\}}$

Producing a graph state for a LNN architecture

Circuit to compile : $C = Z_{\{0,2\}}Z_{\{1,3\}}Z_{\{0,4\}}Z_{\{2,4\}}Z_{\{0,5\}}Z_{\{2,5\}}$

Phase 1 : conversion

Applying conjugation by SWAP gates to use only Z_i and S_i gates :
$Z_{\{2,5\}} = S_{\{2,3\}}S_{\{3,4\}}Z_{\{4,5\}}S_{\{3,4\}}S_{\{2,3\}} = S_2S_3Z_4S_3S_2$

And so on for each $Z_{\{i,j\}}$ in C

Producing a graph state for a LNN architecture

Circuit to compile : $C = Z_{\{0,2\}} Z_{\{1,3\}} Z_{\{0,4\}} Z_{\{2,4\}} Z_{\{0,5\}} Z_{\{2,5\}}$

Phase 1 : conversion

Applying conjugation by SWAP gates to use only Z_i and S_i gates :
$Z_{\{2,5\}} = S_{\{2,3\}} S_{\{3,4\}} Z_{\{4,5\}} S_{\{3,4\}} S_{\{2,3\}} = S_2 S_3 Z_4 S_3 S_2$

And so on for each $Z_{\{i,j\}}$ in C

$S_0 Z_1 S_0 S_1 Z_2 S_1 S_0 S_1 S_2 Z_3 S_2 S_1 S_0 S_2 Z_3 S_2 S_0 S_1 S_2 S_3 Z_4 S_3 S_2 S_1 S_0 S_2 S_3 Z_4 S_3 S_2$
(30 gates)

Producing a graph state for a LNN architecture

Circuit to compile : $C = Z_{\{0,2\}} Z_{\{1,3\}} Z_{\{0,4\}} Z_{\{2,4\}} Z_{\{0,5\}} Z_{\{2,5\}}$

Phase 1 : conversion

Applying conjugation by SWAP gates to use only Z_i and S_i gates :
$Z_{\{2,5\}} = S_{\{2,3\}} S_{\{3,4\}} Z_{\{4,5\}} S_{\{3,4\}} S_{\{2,3\}} = S_2 S_3 Z_4 S_3 S_2$

And so on for each $Z_{\{i,j\}}$ in C

$S_0 Z_1 S_0 S_1 Z_2 S_1 S_0 S_1 S_2 Z_3 S_2 S_1 S_0 S_2 Z_3 S_2 S_0 S_1 S_2 S_3 Z_4 S_3 S_2 S_1 S_0 S_2 S_3 Z_4 S_3 S_2$
(30 gates)

Phase 2 : reduction using the (Coxeter + Dehn) heuristic

- Input word :

  $w_0 = s_0 z_1 s_0 s_1 z_2 s_1 s_0 s_1 s_2 z_3 s_2 s_1 s_0 s_2 z_3 s_2 s_0 s_1 s_2 s_3 z_4 s_3 s_2 s_1 s_0 s_2 s_3 z_4 s_3 s_2$

Producing a graph state for a LNN architecture

Circuit to compile : $C = Z_{\{0,2\}} Z_{\{1,3\}} Z_{\{0,4\}} Z_{\{2,4\}} Z_{\{0,5\}} Z_{\{2,5\}}$

Phase 1 : conversion

Applying conjugation by SWAP gates to use only Z_i and S_i gates :
$Z_{\{2,5\}} = S_{\{2,3\}} S_{\{3,4\}} Z_{\{4,5\}} S_{\{3,4\}} S_{\{2,3\}} = S_2 S_3 Z_4 S_3 S_2$

And so on for each $Z_{\{i,j\}}$ in C

$S_0 Z_1 S_0 S_1 Z_2 S_1 S_0 S_1 S_2 Z_3 S_2 S_1 S_0 S_2 Z_3 S_2 S_0 S_1 S_2 S_3 Z_4 S_3 S_2 S_1 S_0 S_2 S_3 Z_4 S_3 S_2$
(30 gates)

Phase 2 : reduction using the (Coxeter + Dehn) heuristic

- Input word :

  $w_0 = s_0 z_1 s_0 s_1 z_2 s_1 s_0 s_1 s_2 z_3 s_2 s_1 s_0 s_2 z_3 s_2 s_0 s_1 s_2 s_3 z_4 s_3 s_2 s_1 s_0 s_2 s_3 z_4 s_3 s_2$

- reductionCoxeter $\rightarrow$

  $w_1 = s_0 z_1 s_0 s_1 z_2 s_1 s_2 z_3 s_0 s_1 s_2 s_3 z_4 z_3 s_2 s_3 z_4 s_2 s_3 s_1 s_2 s_0$ (length=22)

Producing a graph state for a LNN architecture

Circuit to compile : $C = Z_{\{0,2\}} Z_{\{1,3\}} Z_{\{0,4\}} Z_{\{2,4\}} Z_{\{0,5\}} Z_{\{2,5\}}$

Phase 1 : conversion

Applying conjugation by SWAP gates to use only Z_i and S_i gates :
$Z_{\{2,5\}} = S_{\{2,3\}} S_{\{3,4\}} Z_{\{4,5\}} S_{\{3,4\}} S_{\{2,3\}} = S_2 S_3 Z_4 S_3 S_2$

And so on for each $Z_{\{i,j\}}$ in C

$S_0 Z_1 S_0 S_1 Z_2 S_1 S_0 S_1 S_2 Z_3 S_2 S_1 S_0 S_2 Z_3 S_2 S_0 S_1 S_2 S_3 Z_4 S_3 S_2 S_1 S_0 S_2 S_3 Z_4 S_3 S_2$ (30 gates)

Phase 2 : reduction using the (Coxeter + Dehn) heuristic

- Input word :

  $w_0 = S_0 Z_1 S_0 S_1 Z_2 S_1 S_0 S_1 S_2 Z_3 S_2 S_1 S_0 S_2 Z_3 S_2 S_0 S_1 S_2 S_3 Z_4 S_3 S_2 S_1 S_0 S_2 S_3 Z_4 S_3 S_2$

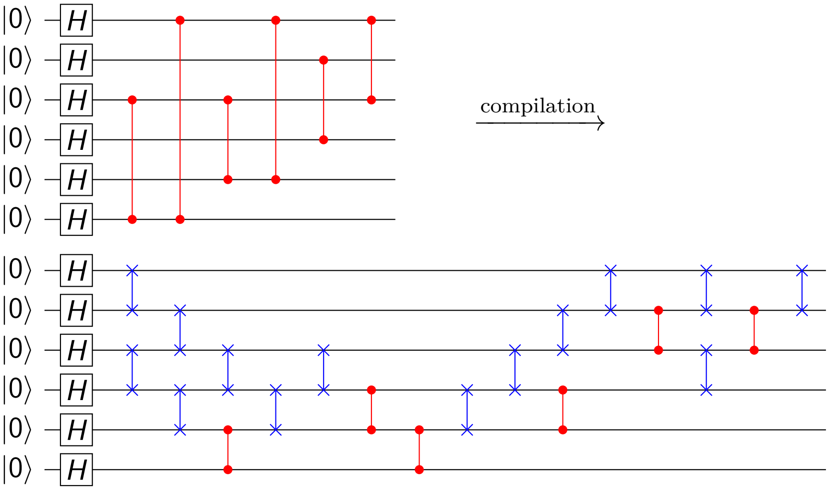
- reductionCoxeter $\rightarrow$

  $w_1 = S_0 Z_1 S_0 S_1 Z_2 S_1 S_2 Z_3 S_0 S_1 S_2 S_3 Z_4 Z_3 S_2 S_3 Z_4 S_2 S_3 S_1 S_2 S_0$ (length=22)

- reductionDehn $\rightarrow$

  $w_2 = S_0 Z_1 S_0 S_2 Z_1 Z_3 S_0 S_1 S_2 S_3 Z_4 Z_3 S_2 S_3 Z_4 S_2 S_3 S_1 S_2 S_0$ (length=20)

Producing a graph state for a LNN architecture



The Minimal Weight heuristic

PROS of the (Coxeter + Dehn) heuristic :

- Polynomial
- Can be applied to others quantum circuits

CONS :

- Even for small cases ($n = 3, 4, 5$) generally doesn't give an optimal result
- So for circuits of many qubits ?

The Minimal Weight heuristic

PROS of the (Coxeter + Dehn) heuristic :

- Polynomial
- Can be applied to others quantum circuits

CONS :

- Even for small cases ($n = 3, 4, 5$) generally doesn't give an optimal result
- So for circuits of many qubits ?

The Minimal Weight heuristic :

$\longrightarrow$ Specialy designed for the group cZS_n.

$\longrightarrow$ Fast : time complexity $\mathrm{O}(kn^2)$ where k is input length

$\longrightarrow$ Gives most of the time an optimal result for small cases.

Weight of a circuit of cZS_n

Definition

Let $Z_E S_\sigma$ be a circuit of the group cZS_n.
The weight w of the Z part of the circuit is :
$w(E) = \sum_{\{i,j\} \in E} (|i-j| - 1)$

Example :

$Z_E = Z_{\{0,2\}} Z_{\{1,3\}} Z_{\{0,4\}} Z_{\{2,4\}} Z_{\{0,5\}} Z_{\{2,5\}}$
$E = \{\{0,2\},\{1,3\},\{0,4\},\{2,4\},\{0,5\},\{2,5\}\}$
$W(E) = 1 + 1 + 3 + 1 + 4 + 2 = 12$

Weight of a circuit of cZS_n

Definition

Let $Z_E S_\sigma$ be a circuit of the group cZS_n.
The weight w of the Z part of the circuit is :
$w(E) = \sum_{\{i,j\} \in E} (|i-j|-1)$

Example :
$Z_E = Z_{\{0,2\}} Z_{\{1,3\}} Z_{\{0,4\}} Z_{\{2,4\}} Z_{\{0,5\}} Z_{\{2,5\}}$
$E = \{\{0,2\},\{1,3\},\{0,4\},\{2,4\},\{0,5\},\{2,5\}\}$
$W(E) = 1+1+3+1+4+2 = 12$

Two simple ideas behind the algorithm :

Idea 1 : At each step of the algorithm the weight decreases by at least 1. (theorem)
Idea 2 : Algorithm stops when $w(E) = 0 \implies$
All gates in the Z part are already allowed by the LNN architecture.

Example : one step of the Minimal Weight heuristic

s_i : the elementary transposition $(i, i+1)$

Input : $C = Z_{\{0,2\}} Z_{\{1,3\}} Z_{\{0,4\}} Z_{\{2,4\}} Z_{\{0,5\}} Z_{\{2,5\}}$

Step 1 :

$Z_E \leftarrow Z_{\{0,2\}} Z_{\{1,3\}} Z_{\{0,4\}} Z_{\{2,4\}} Z_{\{0,5\}} Z_{\{2,5\}}$

$E \leftarrow \{\{0,2\},\{1,3\},\{0,4\},\{2,4\},\{0,5\},\{2,5\}\}$ $w(E) = 12$

$\Phi s_0(E) = \{\{1,2\},\{0,3\},\{1,4\},\{2,4\},\{1,5\},\{2,5\}\}$	$w(s_0(E)) = 10$
$\Phi s_1(E) = \{\{0,1\},\{2,3\},\{0,4\},\{1,4\},\{0,5\},\{1,5\}\}$	$w(s_1(E)) = 12$
$\Phi s_2(E) = \{\{0,3\},\{1,2\},\{0,4\},\{3,4\},\{0,5\},\{3,5\}\}$	$w(s_2(E)) = 10$
$\Phi s_3(E) = \{\{0,2\},\{1,4\},\{0,3\},\{2,3\},\{0,5\},\{2,5\}\}$	$w(s_3(E)) = 11$
$\Phi s_4(E) = \{\{0,2\},\{1,3\},\{0,5\},\{2,5\},\{0,4\},\{2,4\}\}$	$w(s_4(E)) = 12$

Example : one step of the Minimal Weight heuristic

s_i : the elementary transposition $(i, i+1)$

Input : $C = Z_{\{0,2\}} Z_{\{1,3\}} Z_{\{0,4\}} Z_{\{2,4\}} Z_{\{0,5\}} Z_{\{2,5\}}$

Step 1 :

$Z_E \leftarrow Z_{\{0,2\}} Z_{\{1,3\}} Z_{\{0,4\}} Z_{\{2,4\}} Z_{\{0,5\}} Z_{\{2,5\}}$

$E \leftarrow \{\{0,2\},\{1,3\},\{0,4\},\{2,4\},\{0,5\},\{2,5\}\}$ $w(E) = 12$

$\Phi s_0(E) = \{\{1,2\},\{0,3\},\{1,4\},\{2,4\},\{1,5\},\{2,5\}\}$	$w(s_0(E)) = 10$
$\Phi s_1(E) = \{\{0,1\},\{2,3\},\{0,4\},\{1,4\},\{0,5\},\{1,5\}\}$	$w(s_1(E)) = 12$
$\Phi s_2(E) = \{\{0,3\},\{1,2\},\{0,4\},\{3,4\},\{0,5\},\{3,5\}\}$	$w(s_2(E)) = 10$
$\Phi s_3(E) = \{\{0,2\},\{1,4\},\{0,3\},\{2,3\},\{0,5\},\{2,5\}\}$	$w(s_3(E)) = 11$
$\Phi s_4(E) = \{\{0,2\},\{1,3\},\{0,5\},\{2,5\},\{0,4\},\{2,4\}\}$	$w(s_4(E)) = 12$

Greedy : choose the best (smallest) weight at each step

Non deterministic : random choice between S_0 and S_2.

choice $\leftarrow S_0$.

$Z_E = S_0 Z_{\{1,2\}} Z_{\{0,3\}} Z_{\{1,4\}} Z_{\{2,4\}} Z_{\{0,5\}} Z_{\{2,5\}} S_0$

$C = S_0 Z_1 \underbrace{Z_{\{0,3\}} Z_{\{1,4\}} Z_{\{2,4\}} Z_{\{0,5\}} Z_{\{2,5\}}}_{\text{New value of } Z_E \text{ for step 2}} S_0$

Testing the Minimal Weight heuristic for small cases

Best n : choose the shortest word found after n repetitions of the heuristic.

Group	1 rep	Best 10	Best 100	Best 1000
$\lvert cZS_3 \rvert = 48$	min : 67%	min : 96%	min : 100%	min : 100%
$\lvert cZS_4 \rvert = 1\,536$	min : 41%	min : 74%	min :86%	min : 100%
$\lvert cZS_5 \rvert = 122\,880$	min : 19%	min : 46%	min : 71%	min : 72%

Time to compute min for all circuits of cZS_5 is about 4'30".

Average time for Best 1000 on a circuit of cZS_5 is about 9 ms.

Testing the Minimal Weight heuristic for small cases

Best n : choose the shortest word found after n repetitions of the heuristic.

Group	1 rep	Best 10	Best 100	Best 1000
$\lvert cZS_3 \rvert = 48$	min : 67%	min : 96%	min : 100%	min : 100%
$\lvert cZS_4 \rvert = 1\,536$	min : 41%	min : 74%	min :86%	min : 100%
$\lvert cZS_5 \rvert = 122\,880$	min : 19%	min : 46%	min : 71%	min : 72%

Time to compute min for all circuits of cZS_5 is about 4'30".

Average time for Best 1000 on a circuit of cZS_5 is about 9 ms.

In 99,9% of the cases Best 1000 gives a length between min and 1.5min.

THANK YOU FOR YOUR ATTENTION !