VQE et QAOA

Benoit Valiron

https://bit.ly/ro-2021-vqe

Introduction

Il s'agit d'algorithmes dits **variationnels**.

- Dans Grover, on construit UN circuit une fois pour toute
- On réitère éventuellement si on veut
- Mais c'est toujours le même circuit

VQE : Variational Quantum Eigensolver

- à la base un algorithme pour la physique
- pour "trouver un vecteur propre"
  - On va voir ce que cela veut dire
  - Et en quoi cela peut être utile pour nous

QAOA : Quantum Approximate Optimization Algorithm

- spécialisé pour des problèmes d'optimisation
- peut être regardé comme une variante de VQE
- simulation d'une évolution adiabatique
  - on va voir ce que cela veut dire, et pourquoi ça marche

Plan

- Un peu de maths
  - Notion de vecteur/valeur propre
  - Matrice unitaire, matrice hermitienne
  - Hermitiens et problèmes d'optimisation
  - Lien entre les deux
- VQE
  - Fonctionnement général
  - Cas particulier pour les matrices diagonale
- QAOA
  - Concept
  - Pourquoi ça marche
  - Instantiation concrète
  - Exemple : MAXCUT
  - Exemple : TSP

Notion de vecteur propre, de valeur propre.

On dit que $|v\rangle$ est un vecteur propre de A avec valeur propre λ si $A \cdot |v\rangle = \lambda |v\rangle$

Par exemple : un vecteur propre de $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$?

Comme X échange $|0\rangle$ et $|1\rangle$ il faut une combinaison des deux... De fait :

$$X \cdot \left(|0\rangle + |1\rangle\right) = |1\rangle + |0\rangle$$

Et donc... $X \cdot |+\rangle = |+\rangle$ → donc $|+\rangle$ a pour valeur propre 1

$$X \cdot \left(|0\rangle - |1\rangle\right) = |1\rangle - |0\rangle$$

Aussi : $X \cdot |-\rangle = -|-\rangle$ → donc $|-\rangle$ a pour valeur propre -1

Autre exemple : les vecteurs propres de $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$?

$Z|0\rangle = |0\rangle$
$Z|1\rangle = -|1\rangle$

Matrice Hermitienne

En physique, un **Hamiltonien**
Matrice hermitienne → égale à sa transposée complexe
Propriétés:

- Les valeurs propres sont réelles
- On peut trouver une base orthonormale de vecteurs propres.

→ Matrice qu'on peut écrire sous la forme de $\sum_i \lambda_i |v_i\rangle\langle v_i|$ (avec λ_i réels et les v_i orthogonaux, de norme 1)

Notatiion: $|v\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \rightarrow \langle v| = (\overline{\alpha} \ \overline{\beta})$

c'est quoi $|0\rangle\langle 0|$?

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} (1 \ \ 0) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

$$(|0\rangle\langle 0|)|0\rangle = |0\rangle \ (\langle 0||0\rangle) = |0\rangle$$

Pour ma matrice $A = \sum_i \lambda_i |v_i\rangle\langle v_i|$

$$A|v_0\rangle = \sum_i \lambda_i |v_i\rangle\langle v_i||v_0\rangle = \lambda_0 |v_0\rangle$$

Obtenons $\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = |0\rangle\langle 1|$

Dans le cas général :

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = a|0\rangle\langle 0| + b|0\rangle\langle 1| + c|1\rangle\langle 0| + d|1\rangle\langle 1|$$

Factorisation des matrices hermitiennes : diagonalisation sous la forme $U^* D U$ avec U unitaire et D diagonale à coefficients réels :

- U envoie dans la base des vecteurs propres
- D applique les coefficients λ_i qui vont bien
- U^* renvoie dans la base canonique

Exemple :

$$M := \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} = 2 \cdot Id + X$$ (combinaison de matrices de Pauli)

$$M \cdot (|0\rangle + |1\rangle) = 3 \cdot (|0\rangle + |1\rangle)$$
$$M \cdot (|0\rangle - |1\rangle) = |0\rangle - |1\rangle$$

Donc

$$M = 3|+\rangle\langle +| + |-\rangle\langle -|$$

$$M = Had \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} Had$$

Par exemple :

$$|+\rangle \xrightarrow{Had} |0\rangle \xrightarrow{diag} 3 \cdot |0\rangle \xrightarrow{Had} 3\,|+\rangle$$

Essai d'utilisation de la décomposition :

$$M|+\rangle = \left(3|+\rangle\langle+| + |-\rangle\langle-|\right)|+\rangle = 3|+\rangle\langle+||+\rangle + |-\rangle\langle-||+\rangle = 3|+\rangle$$

Propriété : une matrice hermitienne peut toujours s'écrire comme combinaison linéaire de (tenseurs de) matrices de Pauli (X, Y, Z, I) à coefficients réels

Pourquoi ? Essayons avec $A = \begin{pmatrix} a & b - c \cdot i \\ b + c \cdot i & d \end{pmatrix}$ agissant sur 1 qubit.

Supposons $a > d$ (ils doivent être réels)

On doit donc décomposer A avec

$X = |1\rangle\langle 0| + |0\rangle\langle 1|$
$Y = i|1\rangle\langle 0| - i|0\rangle\langle 1|$
$I = |0\rangle\langle 0| + |1\rangle\langle 1|$
$Z = |0\rangle\langle 0| - |1\rangle\langle 1|$

Posons $\lambda = a - d$

Alors $A = a|0\rangle\langle 0| + (b - c \cdot i)|0\rangle\langle 1| + (b + c \cdot i)\,|1\rangle\langle 0| + d|1\rangle\langle 1|$

$$A = c\,Y + b\,X + (d + \lambda/2)I + (\lambda/2)Z$$

Autre exemple:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes I + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes X = \frac{1}{2}(I + Z) \otimes I + \frac{1}{2}(I - Z) \otimes X$$

$$= \frac{1}{2}(I \otimes I + Z \otimes I + I \otimes X - Z \otimes X)$$

Problème d'optimisation vu comme un Hermitien

Un problème d'optimisation est typiquement sous la forme

Maximiser / minimiser $C(x)$ quand x appartient à l'ensemble S

avec C une fonction de coût (à valeurs réelles)

Si on est dans le cas discret et fini, on peut toujours choisir $S = \{0..2^n - 1\}$.

et donc C prend en entrée des chaines de bits de taille n

Une entrée de C est donc sous la forme $x_0 x_1 .... x_{n-1}$

On peut construire une matrice hermitienne diagonale comme suis:

$$H = \sum_x C(x)|x\rangle\langle x|$$

Notez que $H|x\rangle = C(x)|x\rangle$
Donc
- Minimiser $C(x)$ revient à trouver la valeur propre minimale de H
- Maximiser $C(x)$ revient à trouver la valeur propre maximale of H
  → donc la valeur propre minimale de $-H$

Relation entre hermitien et unitaire : L'exponentiation de matrice.

Prenons H hermitienne. Alors cette formule est bien définie:

$$U = e^{iH} = \sum_{n=0}^{\infty} \frac{(iH)^n}{n!}$$

(pour des raisons de norme)

Prenons la diagonalisation de H sous la forme $H = P^{-1}DP$

$$U = \sum_{n=0}^{\infty} \frac{(iP^{-1}DP)^n}{n!} = \sum_{n=0}^{\infty} \frac{i^n P^{-1} D^n P}{n!} = P^{-1}\left(\sum_{n=0}^{\infty} \frac{i^n D^n}{n!}\right)P = P^{-1}\begin{pmatrix} e^{i\theta_1} & & \\ & \ddots & \\ & & e^{i\theta_n} \end{pmatrix}P$$

si $D = \begin{pmatrix} \theta_1 & & \\ & \ddots & \\ & & \theta_n \end{pmatrix}$

et donc les valeurs propres de U sont e puissance les valeurs propres de H.

Comme H est hermitienne, on peut en déduire que U est unitaire car

$$U^* = \sum_{n=0}^{\infty} \left(\frac{(iH)^n}{n!}\right)^* = \sum_{n=0}^{\infty} \frac{(-iH^*)^n}{n!} = \sum_{n=0}^{\infty} \frac{(-iH)^n}{n!} = e^{-iH}$$

Donc on a :

$$UU^* = e^{iH}e^{-iH} = e^{iH-iH} = e^0 = Id$$

et pareil pour $U^*U = Id$

Inversement, toute matrice unitaire U peut s'écrire comme l'exponentiation d'une matrice hermitienne.

Par exemple X, Y et Z sont des matrices hermitiennes !
Si G = X, Y ou Z, on peut définir ce que l'on va appeler **une rotation autour de l'axe G** sous la forme

$$R_G(\theta) = e^{\frac{-i\theta \cdot G}{2}} = \cos(\theta/2) \cdot Id - i \sin(\theta/2) \cdot G$$

En effet, comme $G^2 = Id$, on a

$$R_G(\theta) = \sum_{n=0}^{\infty} \frac{(-i\theta G/2)^n}{n!}$$ qui peut être découpé en n pairs et n impairs, ce qui permets de récupérer un sinus (avec un coefficient $-i$) et un cosinus

avec X, Y et Z :

$$R_X(\theta) = \begin{pmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$$

$$R_Y(\theta) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$$

$$R_Z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}$$ (Notez la phase globale)

Notez que les matrices de ces 3 familles sont unitaires !

Question :
- étant donner les deux angles, comment générer $\cos(\theta/2) \cdot |0\rangle + e^{i\phi} \sin(\theta/2) \cdot |1\rangle$?

$$|0\rangle \xrightarrow{R_Y(\theta)} \cos(\theta/2) \cdot |0\rangle + \sin(\theta/2) \cdot |1\rangle$$
$$\xrightarrow{R_Z(\phi)} \cos(\theta/2) e^{-i\phi/2} \cdot |0\rangle + \sin(\theta/2) e^{i\phi/2} \cdot |1\rangle$$
$$= e^{-i\phi/2} \left(\cos(\theta/2) \cdot |0\rangle + \sin(\theta/2) e^{i\phi} \cdot |1\rangle \right)$$

Donc modulo une phase globale on obtient ce que l'on veut.

→ par ailleurs, cela indique que modulo une phase globale, tout unitaire sur 1 qubit peut être défini comme une rotation sur Z et une rotation sur Y. Trois angles sont donc suffisant pour atteindre tout unitaire sur 1 qubit.

Autre question : quel sense donner à "rotation autour de l'axe G" ?
→ littéralement une rotation autour de l'axe correspondant dans la sphère de Bloch

Bref. On a donc un lien entre la notion de matrice hermitienne et de matrice unitaire.

Quel intéret ?

D'abord, beaucoup de pbs physiques s'expriment avec des matrices hermitiennes (sous le terme "Hamiltoniens") (typiquement le calcul du niveau minimal d'énergie d'une molécule) D'où l'intéret pour le sujet. On y reviendra dans la discussion pour QAOA.

Mais de nombreux de problèmes d'optimisation classiques peuvent aussi s'"encoder" avec une matrice hermitienne. Le fait de pouvoir "passer" à une matrice unitaire donne un angle d'attaque pour utiliser un ordinateur quantique pour résoudre le problème.

VQE (Variational Quantum Eigensolver)

L'archétype d'un algorithme dit "hybride" (même si tout algorithme quantique est hybride). En tout cas, un algorithme où chaque circuit n'est utilisé qu'une seule fois.

Le type de problème que VQE peut résoudre pourrait être nommé **QUANTUM-MIN-EIGEN**:
Input : une matrice hermitienne H
Output : Un vecteur propre $|\psi\rangle$ de valeur propre minimale

L'algorithme est très simple : Il s'agit de résoudre un problème d'optimisation dans l'espace d'états. On veut minimiser la fonction

$$|\psi\rangle \quad \mapsto \quad \langle\psi|H|\psi\rangle$$

Il s'agit techniquement d'une fonction standard: on peut utiliser n'importe quelle procédure standard de déscente de gradient, ou autre.

Dit comme ça, cela pose donc 3 questions.

1) Pourquoi le minimum de cette fonction est bien un vecteur propre de valeur propre minimale ?

On peut faire une base orthonormale de vecteurs propres de H

Écrivons $|\psi_{min}\rangle$ un vecteur propre de valeur propre minimum, et λ_{min} cette valeur propre.

On peut alors écrire $|\psi\rangle = \alpha|\psi_{min}\rangle + \beta|\psi_{min}^{\perp}\rangle$
avec $|\psi_{min}^{\perp}\rangle$ orthogonal à $|\psi_{min}\rangle$ et $1 = |\alpha|^2 + |\beta|^2$

Mais comme $\left|\psi_{min}^{\perp}\right\rangle$ est une combinaison linéaire de vecteurs propres qui ne sont PAS $\left|\psi_{min}\right\rangle$, on
peut en dériver que $H\left|\psi_{min}^{\perp}\right\rangle$ est orthogonal à $\left|\psi_{min}\right\rangle$.

Donc

$$\langle\psi|H|\psi\rangle$$
$$=$$
$$\left(\overline{\alpha}\langle\psi_{min}|+\overline{\beta}\left\langle\psi_{min}^{\perp}\right|\right)H\left(\alpha\left|\psi_{min}\right\rangle+\beta\left|\psi_{min}^{\perp}\right\rangle\right)$$
$$=$$
$$|\alpha|^2\langle\psi_{min}|H|\psi_{min}\rangle+\alpha\overline{\beta}\left\langle\psi_{min}^{\perp}\right|H\left|\psi_{min}\right\rangle+\overline{\alpha}\beta\langle\psi_{min}|H\left|\psi_{min}^{\perp}\right\rangle+|\beta|^2\left\langle\psi_{min}^{\perp}\right|H\left|\psi_{min}^{\perp}\right\rangle$$
$$=$$
$$|\alpha|^2\lambda_{min}\langle\psi_{min}||\psi_{min}\rangle+\alpha\overline{\beta}\lambda_{min}\left\langle\psi_{min}^{\perp}\right||\psi_{min}\rangle+0+|\beta|^2\left\langle\psi_{min}^{\perp}\right|H\left|\psi_{min}^{\perp}\right\rangle$$
$$=$$
$$|\alpha|^2\lambda_{min}+|\beta|^2\left\langle\psi_{min}^{\perp}\right|H\left|\psi_{min}^{\perp}\right\rangle$$

Le vecteur $\left|\psi_{min}^{\perp}\right\rangle$ peut être écrit comme $\sum_j\gamma_j|\psi_j\rangle$ avec $|\psi_j\rangle$ les vecteurs propres de H qui

ne sont pas $\left|\psi_{min}\right\rangle$. Comme ce vecteur est normalisé, on a $\sum|\gamma_j|^2=1$

Soit λ_j les valeurs propres correspondant aux $\left|\psi_j\right\rangle$. On a que $\lambda_{min}\leq\lambda_j$ pour tout j.

On a $\left\langle\psi_{min}^{\perp}\right|H\left|\psi_{min}^{\perp}\right\rangle=\sum_j|\gamma_j|^2\left\langle\psi_j\right|H\left|\psi_j\right\rangle=\sum_j|\gamma_j|^2\lambda_j$.

donc puisque $\lambda_{min}\leq\lambda_j$, on peut dériver que

$$\lambda_{min}=1*\lambda_{min}=\left(\sum_j|\gamma_j|^2\right)\lambda_{min}=\sum_j|\gamma_j|^2\lambda_{min}\leqslant\sum_j|\gamma_j|^2\lambda_j=\left\langle\psi_{min}^{\perp}\right|H\left|\psi_{min}^{\perp}\right\rangle$$

et donc

$$\left\langle\psi_{min}^{\perp}\right|H\left|\psi_{min}^{\perp}\right\rangle\geqslant\lambda_{min}$$

Et donc en résumé

$$\begin{aligned}\langle\psi|H|\psi\rangle &= |\alpha|^2\lambda_{min} + |\beta|^2\langle\psi_{min}^{\perp}|H|\psi_{min}^{\perp}\rangle \\ &\geqslant |\alpha|^2\lambda_{min} + |\beta|^2\lambda_{min} \\ &= \left(|\alpha|^2 + |\beta|^2\right)\lambda_{min} \\ &= \lambda_{min}\end{aligned}$$

ce qui montre que $\langle\psi|H|\psi\rangle$ est toujours plus grand que λ_{min}.

Cette inégalité devient une égalité dans le cas où $|\psi\rangle$ est le vecteur propre $|\psi_{min}\rangle$

Donc si on récupère le minimum absolu de la fonction $|\psi\rangle \mapsto \langle\psi|H|\psi\rangle$, on a λ_{min}, et par extension $|\psi_{min}\rangle$

2) Comment calculer $\langle\psi|H|\psi\rangle$?

C'est là que l'on va faire usage d'un co-processeur quantique.
On décompose d'abord H en une somme de Pauli, comme suit

$$H = h_I I + \sum_{i,a} h_{i,a}\sigma_a^i + \sum_{i,j,a,b} h_{i,j,a,b}\sigma_a^i\sigma_b^j + \sum_{i,j,k,a,b,c} h_{i,j,k,a,b,c}\sigma_a^i\sigma_b^j\sigma_c^k + ...$$

où $a \in \{X, Y, Z\}$ et $\sigma_X^i = X$ sur le qubit i, etc...

Exemple :

$$CNOT = \frac{1}{2}(I\otimes I + Z\otimes I + I\otimes X - Z\otimes X)$$

Donc si j'ecris comme cela

$$CNOT = \frac{1}{2}I + \sum_{i,a} h_{i,a}\sigma_a^i + \sum_{i,j,a,b} h_{i,j,a,b}\sigma_a^i\sigma_b^j$$

- Ordre 2 : $Z\otimes X = \sigma_Z^1\sigma_X^2$ et $h_{1,2,Z,X} = 1/2$
- Ordre 1 : $I\otimes X = \sigma_X^2$ et $h_{2,X} = 1/2$

Donc en général

$$\langle\psi|H|\psi\rangle = \sum_{i,a} h_{i,a}\langle\psi|\sigma_a^i|\psi\rangle + \sum_{i,j,a,b} h_{i,j,a,b}\langle\psi|\sigma_a^i\sigma_b^j|\psi\rangle + ...$$

Il est suffisant de calculer séparemment chaque élément de la somme
$\langle\psi|\sigma_a^i|\psi\rangle$, $\langle\psi|\sigma_a^i\sigma_b^j|\psi\rangle$, ...
et de les sommer offline de manière classique.

Dans les cas qui nous intéressent, on aura pas nécessairement trop de termes à calculer,

donc ce ne sera pas nécessairement absolument inefficace.

Bref.

Si on veut pouvoir faire appel à un co-processeur quantique, il va falloir faire un lien entre le produit scalaire et la mesure de l'état d'une mémoire quantique.

- Prob. P(0) d'avoir $|0\rangle$ quand on mesure $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$
  $\rightarrow |\alpha|^2$
- on a $\alpha = \langle 0|\psi\rangle$

$\rightarrow$ donc $P(0) = |\langle 0|\psi\rangle|^2 = \overline{\langle 0|\psi\rangle}\langle 0|\psi\rangle = \langle\psi|0\rangle\langle 0|\psi\rangle$
donc $P(0) = \langle\psi|\ (|0\rangle\langle 0|)\ |\psi\rangle$

- Prob. P(1) d'avoir $|1\rangle$: $|\langle 1|\psi\rangle|^2 = \overline{\langle 1|\psi\rangle}\langle 1|\psi\rangle = \langle\psi|1\rangle\langle 1|\psi\rangle$

Peut-on dériver la valeur $\langle\psi|\sigma_Z^i|\psi\rangle$?
Notez que $Z = |0\rangle\langle 0| - |1\rangle\langle 1|$
Donc $\langle\psi|\sigma_Z^i|\psi\rangle = \langle\psi|0\rangle\langle 0|\psi\rangle - \langle\psi|1\rangle\langle 1|\psi\rangle = P_i(0) - P_i(1)$ (measurant le $i$ème qubit)

Pour $I = |0\rangle\langle 0| + |1\rangle\langle 1|$ so $\langle\psi|\sigma_I^i|\psi\rangle = P_i(0) + P_i(1)$ =... 1

Comment traiter les cas X et Y, puisque par exemple $X = |0\rangle\langle 1| + |1\rangle\langle 0|$?

On peut s'appuyer sur des relations algébriques, par exemple $X = Had\ Z\ Had$. donc $\langle\psi|X|\psi\rangle = \langle\psi|HZH|\psi\rangle = \big(\langle\psi|H\big)Z\big(H|\psi\rangle\big)$ et il est suffisant de travailler avec $H|\psi\rangle$ à la place de $|\psi\rangle$.

Pour Y on a la règle $Y = SHZHS^*$, donc on peut travailler avec $HS^*|\psi\rangle$.

Quand on travaille avec plus d'1 qubit, on peut quand même s'appuyer sur les résultats de mesure.
Par exemple, considère le cas de 3 qubits, et prenons $Z_0Z_1 = Z \otimes Z \otimes Id$

$\langle\psi|Z_0Z_1|\psi\rangle = \langle\psi|Z \otimes Z \otimes Id|\psi\rangle$
$=$
$\langle\psi|\big(|0\rangle\langle 0| - |1\rangle\langle 1|\big) \otimes \big(|0\rangle\langle 0| - |1\rangle\langle 1|\big) \otimes Id|\psi\rangle$
$=$
$\langle\psi||00\rangle\langle 00| \otimes Id|\psi\rangle - \langle\psi||01\rangle\langle 01| \otimes Id|\psi\rangle - \langle\psi||10\rangle\langle 10| \otimes Id|\psi\rangle + \langle\psi||11\rangle\langle 11| \otimes Id|\psi\rangle$
$=$

$$P(00x) - P(01x) - P(10x) + P(11x)$$

Où $P(00x)$ est la probabilité de mesurer 00 sur les 2 premiers qubits (oubliant le dernier).

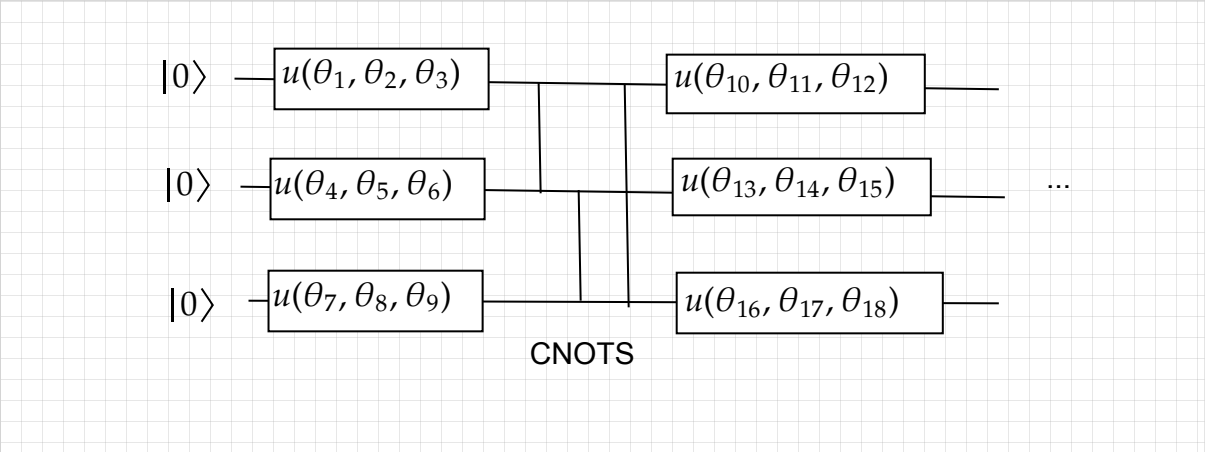
3) Que signifie "travailler dans l'espace d'états" ? et donc comment manipuler des états ?

À la place de manipuler directement $|\psi\rangle$, on va travailler avec un circuit qui **calcule** $|\psi\rangle$. On a juste besoin d'un circuit suffisamment intriquant pour s'approcher d'un état arbitraire, en commençant par exemple avec $|00...0\rangle$

Sur 1 qubit, souvenez-vous de la sphère de Bloch : un unitaire sur 1 qubit peut être décrit avec 3 angles.

Sur plus de qubits, il nous faut en plus une porte intriquante, typiquement la porte CNOT (souvenez-vous comment réaliser l'état de Bell...)

Sur 3 qubits, un tel circuit ressemblera typiquement à



(avec u un unitaire générique)

Le circuit calcule un état candidat pour être minimal, et il est paramétrisé par un tableau de θ's.

On a donc un ensemble purement classique de paramètres (des angles) à partir desquels on construit un circuit, ce qui nous permet de générer $|\psi\rangle$. Réalisant plusieurs runs, on peut du coup estimer les probabilités P(0) et P(1).

Bref, à la place de la fonction mathématique

$$|\psi\rangle \quad \mapsto \quad \langle\psi|H|\psi\rangle$$

on utilise plutôt une opération

$\theta_1 ... \theta_k \quad \mapsto$ estimation de $\langle\psi|H|\psi\rangle$ à partir des résultats de mesure

Celle-là est bien définissable en python par exemple, et on peut bien la donner à un opérateur de minimisation.

Cas particulier des matrices hermitiennes diagonales

Rappelez-vous, ce sont typiquement les matrices issus de problèmes d'optimisation !
Typiquement: minimiser $C(x)$ pour x une chaine de bits de longueur n

Cela revient à trouver le vecteur propre de valeur propre minimale de

$$H = \sum_x C(x)|x\rangle\langle x|$$

Bref.

Dans ce cas, on peut faire "plus simple" pour calculer $\langle\psi|H|\psi\rangle$. En effet, si

$$\psi = \sum_i \alpha_i |i\rangle$$

alors

$$\langle\psi|H|\psi\rangle = \sum_i |\alpha_i|^2 C(i)$$

et donc cela peut être calculé offline en estimant la distribution de probabilité issue de la mesure de $|\psi\rangle$.

Détaillons :

on fabrique une fonction $f : \vec{\theta} \quad \mapsto$ un réel.
Définition:

- Génère un circuit $CIRC_{\vec{\theta}}$: le nombre de couche est fonction de la taille de $\vec{\theta}$
- on évalue le circuit sur $|00...0\rangle$ suivi d'une mesure. un nombre N de fois
- Cela donne une estimation de la distribution de probabiités qui à chaque $x_1 ... x_n$ associe une estimation de proba $p_{\vec{x}} \in [0.1]$
- Le nombre réel produit par f est $\sum_{\vec{x}} p_{\vec{x}}\, C(\vec{x})$

QAOA: Quantum Approximate Optimization Algorithm

seminal paper : https://arxiv.org/abs/1411.4028

Prenons un peu de hauteur sur VQE : on part d'un hamiltonien dont on ne connait pas grand chose à part la décomposition en matrices de Pauli. On nous demande un vecteur propre de valeur propre minimale. L'algorithme fonctionne donc complètement à l'aveugle, en essayant d'inférer un circuit calculant ce vecteur à partir de la forme de circuit la plus générique possible.

On peut imaginer de faire un tout petit peu plus malin, et c'est le but de la proposition de Farhi et ses coauteurs avec QAOA.

On se sert de deux hermitiens particuliers, tous deux agissant sur nos n qubits

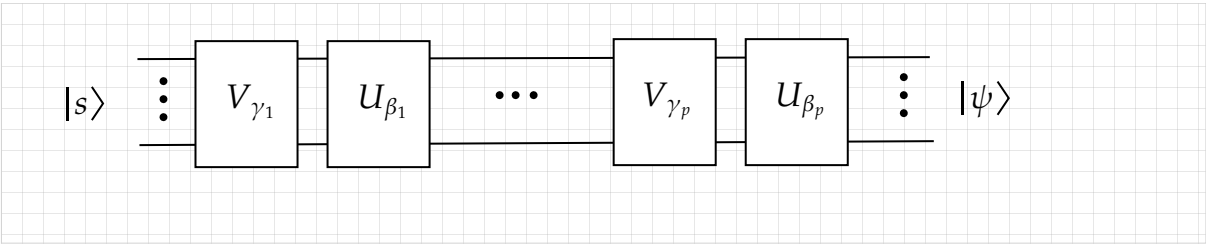
$$B = \sum_{i=1}^{n} \sigma_X^i$$

et l'hamiltonien codant la fonction de coût:

$$H_C = \sum_x C(x)|x\rangle\langle x|$$

Vu de loin, l'algorithme ressemble à VQE:

- On part de l'état $|s\rangle = \frac{1}{\sqrt{2}^n} \sum_{x_1 \ldots x_n} |x_1 \ldots x_n\rangle$
  → colonne de Hadamard à partir de $|000...00\rangle$
- On génère $|\psi\rangle$ à partir d'un circuit paramétré par deux tableaux $\vec{\beta}$ et $\vec{\gamma}$ de chacun p angles
  - $|\psi\rangle = U_{\beta_p} V_{\gamma_p} U_{\beta_{p-1}} V_{\gamma_{p-1}} \ldots U_{\beta_2} V_{\gamma_2} U_{\beta_1} V_{\gamma_1} |s\rangle$
  - avec
    * $U_\lambda = e^{-i\lambda B}$
    * $V_\lambda = e^{-i\lambda H_C}$



- On calcule $\langle\psi|H_C|\psi\rangle$, en capitalisant sur le fait que H_C est diagonale et générée par la fonction de coût : estimation de la distribution de probabilités issue de la mesure de $|\psi\rangle$, puis calcule avec $C(x)$.
- Cette série d'opérations peut être vue comme une fonction qui prend en entrée $(\vec{\beta}, \vec{\gamma})$ et qui rend un nombre réel : l'objectif sera ici de maximiser la fonction (donc d'arriver à un vecteur propre de valeur propre maximale)

Vu comme ça, l'algorithme QAOA est "juste" une variante de VQE, avec l'utilisation d'un circuit très particulier. La question qui se pose est : pourquoi est-ce qu'on a une bonne

chance de tomber sur le vecteur qui nous intéresse en faisant une minimisation?

En fait, il y a deux questions:

- Est-ce qu'il y a une preuve d'un quelconque speedup ?
  - Non
  - Quelques résultats théoriques pour p=1 et p=2, avec des bornes d'erreurs
  - Mais pas de speedup démontré jusqu'à présent, donc
  - Mais pas de preuve du contraire non plus
- Est-ce qu'on a à défaut une garantie de convergence ?
  - Oui, avec p "assez grand"
  - Pour comprendre comment ça marche, il faut faire un petit détour

Détour : Évolution adiabatique

(*Disclaimer:* Je ne suis pas physicien. Il y a des tas de petites lignes dans ce que je vais dire, que je ne maitrise pas nécessairement.)

Un système physique est soumis à un Hamiltonien H qui dans notre cas n'est rien d'autre qu'un opérateur hermitien, en dimension finie: donc une "matrice".

Cet Hamiltonien peut varier au cours du temps : $H(t)$. L'évolution du système $|\psi_t\rangle$ est décrite par l'équation de Schrödinger:

$$\frac{d|\psi_t\rangle}{dt} = -i \cdot H(t)|\psi_t\rangle$$

Note : si $H(t)$ est constante de valeur H, la solution à la condition initiale $|\psi_0\rangle$ est...

$$|\psi_t\rangle = e^{-itH}|\psi_0\rangle$$

(et on retrouve la relation entre Hermitien et unitaire)

Théorème adiabatique : "Si $H(t)$ varie suffisamment lentement, et si le système est à l'origine dans l'état propre de valeur propre minimale de $H(0)$, il passe à chaque instant t dans l'état propre minimale de $H(t)$"

Notes:

- La lenteur a à voir avec le "spectral gap", la distance minimale entre les deux valeurs propres les plus basse
- Idée : le vecteur propre de valeur propre minimal correspond à une énergie minimale pour le système.
- On peut faire le même jeu avec les vecteurs propres de valeurs propres maximales en travaillant avec... $-H(t)$.

Revenons à QAOA :

On part de $|\psi_0\rangle = |s\rangle = \frac{1}{\sqrt{2}^n}\sum_{x_1...x_n}|x_1...x_n\rangle = \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\otimes...\otimes\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$
On veut arriver au vecteur propre de valeur propre maximale pour H_C.

Question : $|s\rangle$ est-il vecteur propre de valeur propre maximale de quelqu'un ?

$\rightarrow B = \sum_{i=1}^{n}\sigma_X^i$
Et de fait, il a les bonnes propriétés pour les "petites lignes" du théorème adiabatique.

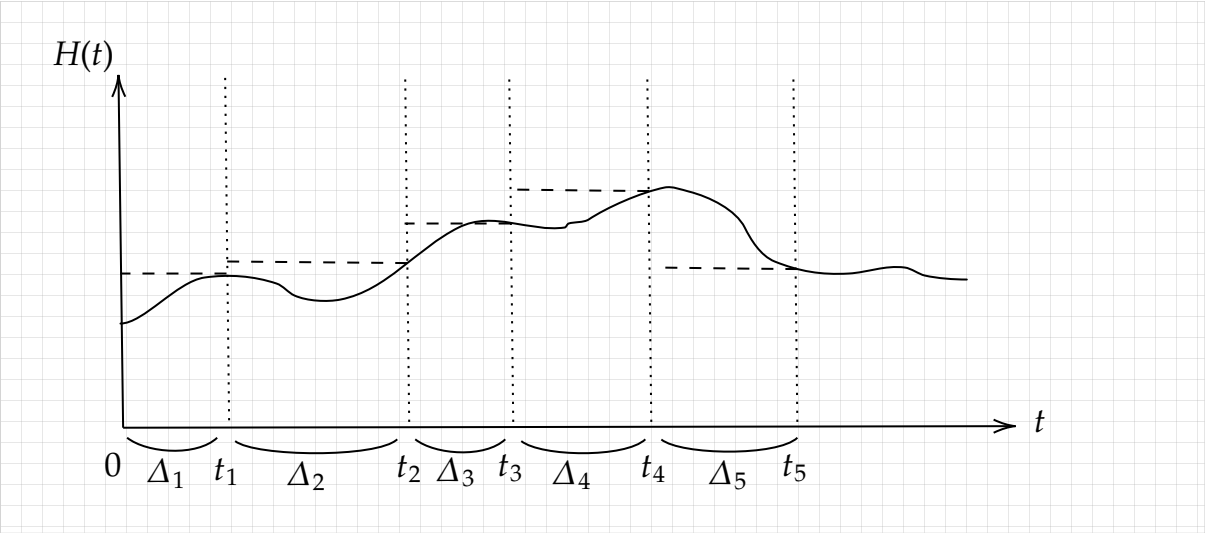
Du coup, d'après le théorème adiabatique, si on fabrique une interpolation:

$$H(t) = \frac{t}{T}H_C + \left(1-\frac{t}{T}\right)B$$

en prenant T "suffisamment grand" on passe de $|s\rangle$ au vecteur propre recherché.
$\rightarrow H(0) = B$ et $H(T) = H_C$

Reste à savoir comment réaliser cette opération.

Plutôt que de considérer $H(t)$, on va considérer une approximation constante par morceaux. Pour ne pas diverger trop de la "vraie" fonction, il faut que les tranches soient suffisemment petites



Donc sur chaque tranche de temps Δ_i, l'Hamiltonien est approximé comme constant. On peut donc dire que

- $|\psi_{t_{i+1}}\rangle \simeq e^{-i\Delta_i H(t_i)}|\psi_{t_i}\rangle$

et donc que si on a découpé $[0, T]$ en p tranches,

$$|\psi_T\rangle = e^{-i\Delta_p H(t_p)} \dots e^{-i\Delta_2 H(t_2)} e^{-i\Delta_1 H(t_1)} |s\rangle$$

est donc une approximation du vecteur propre maximal de H_C.

On a presque la forme de QAOA !
Pour conclure, il faut utiliser la **formule de Trotter-Suzuki** qui implique que

$$e^{\delta(A+B)} = e^{\delta A} e^{\delta B} + O(\delta^2)$$

Donc $e^{-i\Delta_p H(t_p)} = e^{-i\Delta_p(\alpha B + \beta H_C)} \simeq e^{-i\Delta_p \alpha B} e^{-i\Delta_p \beta H_C}$

Résumons: pour QAOA, on construit

- $|\psi\rangle = U_{\beta_p} V_{\gamma_p} U_{\beta_{p-1}} V_{\gamma_{p-1}} \dots U_{\beta_2} V_{\gamma_2} U_{\beta_1} V_{\gamma_1} |s\rangle$

avec

- $U_\lambda = e^{-i\lambda B}$
- $V_\lambda = e^{-i\lambda H_C}$

Avec le théorème adiabatique, on a

$$|\psi_T\rangle = e^{-i\Delta_p H(t_p)} \dots e^{-i\Delta_2 H(t_2)} e^{-i\Delta_1 H(t_1)} |s\rangle$$

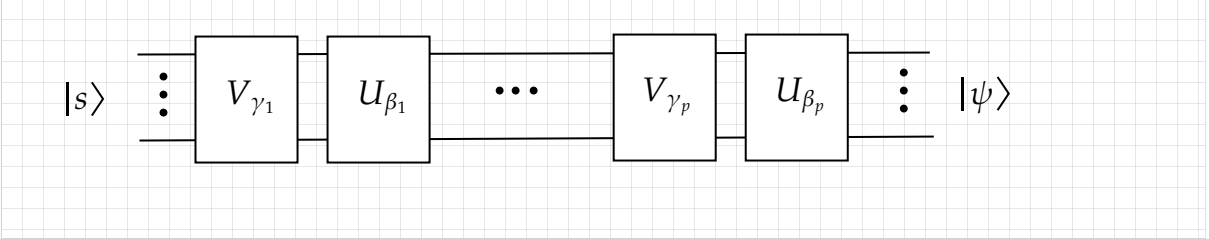
On peut donc remplacer chaque $e^{-i\Delta_p H(t_p)}$ par un produit $U_\alpha V_\beta$.

Comment choisir chacun de ces α, β, puisque on ne sait pas quels Δ_i et t_i sont optimaux ?

→ C'est l'intéret de regarder le problème comme un problème de minimisation.

Si on résume

Pour QAOA, il suffit de savoir générer un circuit



avec

- $U_\lambda = e^{-i\lambda B}$
- $V_\lambda = e^{-i\lambda H_C}$

puis on cherche à minimiser une fonction qui

- prend en entrée $\vec{\beta}, \vec{\gamma}$
- Réalise et fait tourner le circuit correspondant plusieurs fois
- obtient une distribution de probabilités
- calcule et rend $\langle\psi|H_C|\psi\rangle$

Et ***voila*** !

QAOA : instantiation en somme de matrices de Pauli

La fonction de coût peut toujours s'écrire sous la forme:

$$C(x) = \sum_{Q \cup Q' = \{0..n-1\}} w_{Q,Q'} \left(\prod_{i \in Q} x_i\right)\left(\prod_{j \in Q'} (1 - x_j)\right)$$

pour chaque x, il n'y a qu'un seul élément de la somme qui est non-nul : le cas où

- Q' = les indices où $x_i = 0$
- Q = les indices où $x_i = 1$

Et donc à x il y a exactement une seule paire (Q, Q') correspondant à cet élément non-nul de la somme, et on peut poser $w_{Q,Q'} = C(x)$ pour le x en question.

Note : $\left(\prod_{i \in Q} x_i\right)\left(\prod_{j \in Q'} (1 - x_j)\right)$ réalise explicitement la fonction caractéristique qui vaut 1 quand les x_i valent 1 sur Q et 0 sur Q'.

Nous, on s'intéresse à l'Hamiltonien diagonal:

$$H_C = \sum_x C(x)|x\rangle\langle x|$$

Le problème est de trouver la forme en somme de Pauli.

C'est réalisable en utilisant la formule de $C(x)$ avec les $\omega_{Q,Q'}$. On remplace x_i par $(I - Z_i)/2$
(Où Z_i est l'opération qui applique Z sur le qubit i et l'identité sur les autres ≡ σ_i^Z)

$$H_C$$
$$=$$
$$\sum_{(Q,Q') \in \{0..n-1\}} w_{Q,Q'} \left(\prod_{i \in Q} (I - Z_i)/2\right)\left(\prod_{j \in Q'} (1 - (I - Z_j)/2)\right)$$
$$=$$
$$\sum_{(Q,Q') \in \{0..n-1\}} \frac{w_{Q,Q'}}{2^{|Q|+|Q'|}} \left(\prod_{i \in Q} (I - Z_i)\right)\left(\prod_{j \in Q'} (I + Z_j)\right)$$

On peut vérifier que si on applique $|x\rangle$ à H_C on retrouve bien $C(x)|x\rangle$
Car :
- Dans le produit, les $i \in Q$ et les $j \in Q'$ sont tous 2 à 2 distincts -- donc pour qubit dans $|x\rangle$ on applique
  + soit $I - Z_i$: si x_i vaut 1
  + soit $I + Z_j$: si x_j vaut 0

QAOA pour MAXCUT

On part d'un graphe (non-orienté) $G = (V, E)$.
Une COUPE est une partition de V en $V_0 \uplus V_1$.
La fonction de coût à optimiser est le nombre d'arrêtes qui passent de V_0 à V_1.

On peut stocker dans une variable booléenne x_i l'emplacement de $i \in V$:

$$i \in V_{x_i}$$

J'ai un vecteur $x_0 ... x_{n-1}$ (si j'ai $V = \{0 ... n-1\}$) : ce vecteur stocke là où se trouve chaque noeud.

On peut alors écrire $C(x) = \sum_{(i,j)\in E} x_i(1-x_j) + x_j(1-x_i)$ → cette fonction "compte" combien on a d'arêtes entre V_0 et V_1.

L'Hamiltonien correspond est

$$H_C = \sum_{(i,j)\in E} \left(\frac{1}{4}(1-Z_i)(1+Z_j) + \frac{1}{4}(1-Z_j)(1+Z_i) \right)$$

$$=$$

$$\frac{1}{4} \sum_{(i,j)\in E} (1 - Z_i + Z_j - Z_iZ_j + 1 - Z_j + Z_i - Z_jZ_i)$$

$=$ (note : on a $Z_iZ_j = Z_jZ_i$)

$$\frac{1}{4} \sum_{(i,j)\in E} (2 - 2Z_iZ_j)$$

$$=$$

$$\frac{1}{2} \sum_{(i,j)\in E} (1 - Z_iZ_j)$$

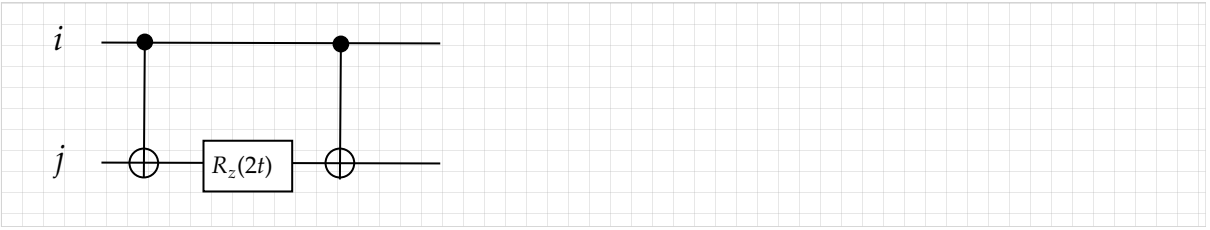
et hop ! On a une décomposition en matrices de Pauli.

$$e^{itZ_iZ_j} = \sum_{n=0}^{\infty} \frac{(it)^n}{n!}(Z_iZ_j)^n = \left(\sum_{n=0}^{\infty} \frac{(-t)^n}{(2n)!}\right) I + i\left(\sum_{n=0}^{\infty} \frac{(-t)^n}{(2n+1)!}\right) Z_iZ_j$$

$$= \cos(t) \cdot I + i \sin(t) \cdot Z_iZ_j$$

Que se passe-t-il sur les fils i, j :

- $|00\rangle \mapsto e^{it}|00\rangle$
- $|11\rangle \mapsto e^{it}|11\rangle$
- $|01\rangle \mapsto -e^{it}|01\rangle$
- $|10\rangle \mapsto -e^{it}|10\rangle$



QAOA pour le voyageur de commerce

C'est la même idée !

- On va trouver un encodage du problème sur une chaine de bits
- On écrit une fonction de coût à optimiser
- On fabrique un Hamiltonien
- Et voila !

Le problème à optimiser:
Soit un graphe pondéré $G = (V, E)$ et un réel L, existe-t-il un chemin Hamiltonien de longueur $\leqslant L$?

On va commencer par coder le problème dans un problème de programmation linéaire, avec un codage booléen comme suit:

$x_{i,t} = 1$ si le cycle solution passe par le noeud i au temps t, et 0 sinon

Donc on considère tableau de booléens: la première coordonnée parle du numéro de noeud, et la deuxième du temps de passage.

On peut en tout cas écrire la longueur totale du chemin de la façon suivante:

$$D(x) = \sum_{(i,j)\in E} w_{i,j} \sum_t x_{i,t}x_{j,t+1}$$

On pourrait alors essayer de trouver le minimum de cette fonction. Mais ce minimum n'a pas de garanti d'être correct ! Exemples de problèmes

- 2 villes visitées en meme temps
- ne pas visiter toutes les villes

On doit s'assurer de deux contraintes: on ajoute donc une pénalité à chaque fois.

- Chaque ville est visitée une et une seule fois. Pénalité si non vérifié:

$$P_1(x) = \sum_i \left(1 - \sum_t x_{i,t}\right)^2$$

- Chaque instant t ne doit correspondre qu'à une seule ville. Pénalité si non vérifié:

$$P_2(x) = \sum_t \left(1 - \sum_i x_{i,t}\right)^2$$

On construit donc la fonction de coût suivante:

$$C(x) = D(x) + A(P_1(x) + P_2(x))$$

avec un facteur multiplicatif A, choisi "assez grand".

Ensuite, on fait le même jeu que pour MAXCUT pour générer l'hamiltonien...

QAOA : discussion

La version originelle de QAOA utilise donc un opérateur hermitien **diagonal**.
Le corollaire est que TOUT se retrouve sur la seule dimension Z: le coût, et les pénalités.
Une proposition récente:

https://arxiv.org/abs/1709.03489

propose d'utiliser aussi des matrices de Pauli X et Y, ce qui permet de gagner en souplesse et potentiellement en expressivité.