

Algorithme quantique QAOA

Mathis Beaudoin

1 Introduction

QAOA (Quantum Approximate Optimization Algorithm) est un algorithme quantique permettant de résoudre des problèmes d'optimisation combinatoires. En d'autres mots, il permet de trouver pour un problème donné les meilleures solutions parmi un ensemble de solutions possibles.

Parmi tous les problèmes auxquels cet algorithme peut être appliqué, on retrouve un problème nommé MaxCut. L'exercice de MaxCut consiste à colorier les sommets d'un graphe de telle sorte que deux noeuds connectés par une arête soient le plus souvent possible d'une couleur différente. Nous allons ici considérer que seulement deux couleurs sont utilisées pour faire le coloriage du graphe (bleu ou rouge). Pour ce problème, il peut exister une solution «parfaite» où les sommets de chaque arête sont de couleurs différentes. Cela se produit lorsque le graphe est biparti, c'est-à-dire lorsqu'on peut séparer son ensemble de sommets en deux sous-ensembles A et B dont chaque arête a une extrémité dans A et l'autre dans B . Dans le cas où une solution parfaite existe, MaxCut est un problème trivial. Cependant, pour les graphes qui ne sont pas bipartis, il faudra faire des compromis et il y aura une ou plusieurs arêtes avec deux noeuds de même couleur. Dans ce cas, il faut chercher parmi l'ensemble des coloriages les solutions où cela se produit le moins possible. Pour des graphes complexes, cette tâche est fastidieuse et il est préférable d'utiliser un algorithme pour trouver ces solutions. Il existe des algorithmes classiques permettant de les trouver, mais ils demandent beaucoup de temps et de ressources afin d'avoir des réponses exactes. Alors, on préfère utiliser des méthodes classiques approximatives pour avoir des solutions approximatives dans un délai raisonnable. Ultimement, on souhaite utiliser QAOA afin de connaître ces meilleures solutions pour MaxCut. Petite remarque : dès qu'on trouve une solution, le fait d'inverser toutes ses couleurs donne aussi une solution valide. Donc, il y a toujours un nombre pair de solutions.

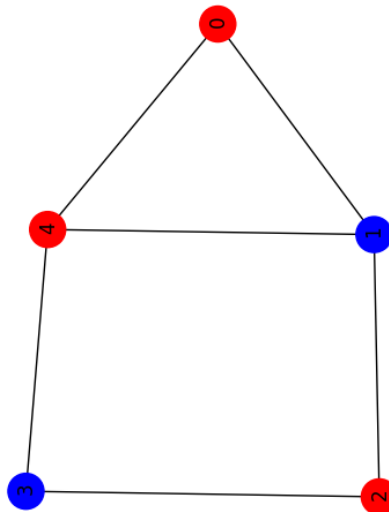


Figure 1 - Exemple d'une solution au problème MaxCut pour un graphe à 5 noeuds (graphe de la maison)

Essayons maintenant de comprendre les principes sur lesquels se base QAOA, puis tentons d'appliquer les notions vues afin de résoudre le problème MaxCut pour un graphe quelconque.

2 Théorie

2.1 Théorème adiabatique

Soit deux hamiltoniens \hat{H}_F et \hat{H}_D . \hat{H}_F est un hamiltonien dit *facile*, c'est-à-dire que sa forme et son état propre de plus faible énergie (appelé *l'état fondamental*) sont connus. Au contraire, \hat{H}_D est un hamiltonien dit *difficile* puisque bien qu'on connaît sa forme, son état fondamental est difficile à calculer. S'il nous est primordial de vouloir trouver l'état fondamental de \hat{H}_D , tout n'est pas perdu ! Effectivement, on peut employer un théorème physique intéressant nommé le théorème adiabatique.

Th. 1 (Théorème adiabatique) : *Un système quantique est gardé dans son état propre instantané si une perturbation agit sur lui suffisamment lentement et s'il y a un intervalle significatif entre la valeur propre et le reste du spectre de l'hamiltonien. Entre autres, si le système se trouve initialement dans un état propre d'un hamiltonien de départ, le système sera dans l'état propre correspondant de l'hamiltonien final après ces changements.*

Autrement dit, si un système quantique est initialement préparé dans l'état fondamental de \hat{H}_F et que ce même hamiltonien gouverne les énergies du système au départ, alors il suffit de faire évoluer l'hamiltonien de départ (\hat{H}_F) de telle sorte que l'hamiltonien du système corresponde à \hat{H}_D à la fin de l'évolution. Au cours de l'évolution, l'état du système restera dans l'état fondamental de l'hamiltonien instantané (à chaque instant) et l'état final sera donc l'état fondamental de \hat{H}_D . On appelle cette transformation une évolution adiabatique et elle est centrale dans le fonctionnement de QAOA. Cependant, il faut faire attention, car une évolution trop rapide peut provoquer une transition vers un état d'énergie supérieure (ce qui n'est pas voulu). Plus l'évolution sera lente, plus la probabilité qu'il y ait une transition d'états sera faible. Aussi, le spectre de l'hamiltonien instantané (à quel point les niveaux d'énergies sont séparés les uns des autres) joue un rôle dans la probabilité de transition. Plus la différence d'énergie entre l'état fondamental et l'état excité est grande, plus la probabilité de quitter l'état fondamental diminue.

2.2 Hamiltonien dépendant du temps

Pour utiliser le théorème adiabatique, il faudra un hamiltonien dépendant du temps. L'évolution d'un système quantique avec un hamiltonien dépendant du temps suit l'équation de Schrödinger

$$i \frac{d}{dt} |\psi(t)\rangle = \hat{H}(t) |\psi(t)\rangle$$

avec $|\psi(t)\rangle$ l'état quantique suivant cette évolution. Si un état initial $|\psi(t_0)\rangle$ est défini au départ de l'évolution, on peut réécrire

$$|\psi(t)\rangle = \hat{U}(t, t_0) |\psi(t_0)\rangle$$

où $\hat{U}(t, t_0)$ est un opérateur d'évolution faisant évoluer $|\psi(t_0)\rangle$ vers $|\psi(t)\rangle$. Donc, on peut réécrire l'équation de Schrödinger

$$i \frac{d}{dt} \hat{U}(t, t_0) |\psi(t_0)\rangle = \hat{H}(t) \hat{U}(t, t_0) |\psi(t_0)\rangle$$

où l'opérateur d'évolution $\hat{U}(t, t_0)$ doit satisfaire

$$\begin{aligned} i \frac{d}{dt} \hat{U}(t, t_0) &= \hat{H}(t) \hat{U}(t, t_0) \\ \frac{d}{dt} \hat{U}(t, t_0) &= -i \hat{H}(t) \hat{U}(t, t_0) \\ d\hat{U}(t, t_0) &= -i \hat{H}(t) \hat{U}(t, t_0) dt \\ \int_{\hat{U}(t_0, t_0)}^{\hat{U}(t, t_0)} d\hat{U}' &= -i \int_{t_0}^t \hat{H}(t') \hat{U}(t', t_0) dt' \\ \hat{U}(t, t_0) - \hat{U}(t_0, t_0) &= -i \int_{t_0}^t \hat{H}(t') \hat{U}(t', t_0) dt' \\ \hat{U}(t, t_0) &= \hat{I} - i \int_{t_0}^t \hat{H}(t_1) \hat{U}(t_1, t_0) dt_1 \end{aligned} \tag{1}$$

Dans la précédente démonstration, on utilise le fait que $\hat{U}(t_0, t_0) = \hat{I}$ (aucune évolution n'a été effectuée). On peut remarquer qu'il y a un opérateur d'évolution à droite dans l'équation (1) et on utilise cette même équation pour trouver sa définition.

$$\hat{U}(t_1, t_0) = \hat{I} - i \int_{t_0}^{t_1} \hat{H}(t_2) \hat{U}(t_2, t_0) dt_2$$

Donc,

$$\begin{aligned} \hat{U}(t, t_0) &= \hat{I} - i \int_{t_0}^t dt_1 \hat{H}(t_1) \left(\hat{I} - i \int_{t_0}^{t_1} \hat{H}(t_2) \hat{U}(t_2, t_0) dt_2 \right) \\ \hat{U}(t, t_0) &= \hat{I} - i \int_{t_0}^t dt_1 \hat{H}(t_1) - \int_{t_0}^t dt_1 \hat{H}(t_1) \int_{t_0}^{t_1} \hat{H}(t_2) \hat{U}(t_2, t_0) dt_2 \end{aligned}$$

De manière récursive, on remplace l'équation dans elle-même une infinité de fois pour obtenir une relation générale

$$\hat{U}(t, t_0) = \hat{I} + \sum_{n=1}^{\infty} (-i)^n \int_{t_0}^t \hat{H}(t_1) dt_1 \int_{t_0}^{t_1} \hat{H}(t_2) dt_2 \cdots \int_{t_0}^{t_{n-1}} \hat{H}(t_n) dt_n$$

qu'on réorganise pour obtenir

$$\hat{U}(t, t_0) = \sum_{n=0}^{\infty} (-i)^n \int_{t_0}^t \hat{H}(t_1) dt_1 \int_{t_0}^{t_1} \hat{H}(t_2) dt_2 \cdots \int_{t_0}^{t_{n-1}} \hat{H}(t_n) dt_n \quad (2)$$

Par construction, $t_1 \geq t_2 \geq \cdots \geq t_n$.

2.3 Opérateur de décalage en temps

L'opérateur de décalage \hat{T} agit sur un second opérateur dépendant du temps $\hat{F}(t)$ en décalant par un temps τ le moment auquel $\hat{F}(t)$ est évalué.

$$e^{-i\tau\hat{T}} \hat{F}(t) e^{i\tau\hat{T}} = \hat{F}(t + \tau)$$

On en tire la relation suivante.

$$\hat{F}(t) e^{i\tau\hat{T}} = e^{i\tau\hat{T}} \hat{F}(t + \tau)$$

Par des chemins sombres, il est possible d'unir l'équation (2) et l'opérateur de décalage en temps.

$$\hat{U}(t_0 + \tau, t_0) = \sum_{n=0}^{\infty} (-i)^n \int_{t_0}^{t_0+\tau} \hat{H}(t_1) dt_1 \int_{t_0}^{t_1} \hat{H}(t_2) dt_2 \cdots \int_{t_0}^{t_{n-1}} \hat{H}(t_n) dt_n = e^{-i\tau\hat{T}} e^{i\tau(\hat{H}(t_0)+\hat{T})} \quad (3)$$

De plus, on peut montrer avec la définition de la dérivée (limite) une propriété intéressante de l'opérateur de décalage en temps.

$$i \frac{\partial}{\partial t} \hat{F}(t) = [\hat{T}, \hat{F}(t)]$$

2.4 Approximations d'un hamiltonien dépendant du temps

On cherche à approximer la relation (3) de sorte à se débarrasser de l'opérateur de décalage en temps. Plus précisément, on veut approximer l'exponentielle $e^{i\tau(\hat{H}(t_0)+\hat{T})}$ qui, comme on le verra, permettra d'utiliser les propriétés de l'opérateur de décalage en temps afin d'arriver à un résultat intéressant. Il existe plusieurs manières de faire cette approximation qui aura comme conséquence de rendre l'hamiltonien constant en l'évaluant à un temps entre t_0 et $t_0 + \tau$. Par de longs calculs, on peut trouver l'approximation permettant d'avoir l'hamiltonien constant évalué à un moment arbitraire dans l'intervalle.

$$e^{i\tau(\hat{H}(t_0)+\hat{T})} = e^{i\tau(\hat{T}+\hat{H}(t_0))} \approx e^{i\tau(1-\eta)\hat{T}} e^{i\tau\hat{H}(t_0)} e^{i\tau\eta\hat{T}} e^{\frac{\tau^2}{2}(1-2\eta)[\hat{T}, \hat{H}(t_0)]} e^{-i\frac{\tau^3}{6}((1-3\eta^2)[\hat{T}, [\hat{T}, \hat{H}(t_0)]] + (2-3\eta)[\hat{H}(t_0), [\hat{T}, \hat{H}(t_0)])}$$

Ainsi, on a par les propriétés de l'opérateur de décalage en temps

$$e^{i\tau(\hat{H}(t_0)+\hat{T})} = e^{i\tau(\hat{T}+\hat{H}(t_0))} \approx e^{i\tau\hat{T}} e^{i\tau\hat{H}(t_0+\eta\tau)} e^{\frac{\tau^2}{2}(1-2\eta)[\hat{T}, \hat{H}(t_0)]} e^{-i\frac{\tau^3}{6}((1-3\eta^2)[\hat{T}, [\hat{T}, \hat{H}(t_0)]] + (2-3\eta)[\hat{H}(t_0), [\hat{T}, \hat{H}(t_0)])}$$

qu'on remplace dans l'équation (3) pour obtenir

$$\begin{aligned}\hat{U}(t_0 + \tau, t_0) &\approx e^{-i\tau\hat{T}} e^{i\tau\hat{H}} e^{i\tau\hat{H}(t_0+\eta\tau)} e^{\frac{\tau^2}{2}(1-2\eta)[\hat{T}, \hat{H}(t_0)]} e^{-i\frac{\tau^3}{6}((1-3\eta^2)[\hat{T}, [\hat{T}, \hat{H}(t_0)]] + (2-3\eta)[\hat{H}(t_0), [\hat{T}, \hat{H}(t_0)])}) \\ \hat{U}(t_0 + \tau, t_0) &\approx e^{i\tau\hat{H}(t_0+\eta\tau)} e^{\frac{\tau^2}{2}(1-2\eta)[\hat{T}, \hat{H}(t_0)]} e^{-i\frac{\tau^3}{6}((1-3\eta^2)[\hat{T}, [\hat{T}, \hat{H}(t_0)]] + (2-3\eta)[\hat{H}(t_0), [\hat{T}, \hat{H}(t_0)])})\end{aligned}\quad (4)$$

avec $\eta \in [0, 1]$.

2.4.1 Hamiltonien constant évalué à t_0 (début de l'intervalle)

On prend $\eta = 0$ et on complète l'équation (4).

$$\begin{aligned}\hat{U}(t_0 + \tau, t_0) &\approx e^{i\tau\hat{H}(t_0)} e^{\frac{\tau^2}{2}[\hat{T}, \hat{H}(t_0)]} e^{\frac{-i\tau^3}{6}([\hat{T}, [\hat{T}, \hat{H}(t_0)]] + 2[\hat{H}(t_0), [\hat{T}, \hat{H}(t_0)])}) \\ &\approx e^{i\tau\hat{H}(t_0)} e^{i\frac{\tau^2}{2}\frac{\partial}{\partial t}\hat{H}(t_0)} + \mathcal{O}(\tau^3) \\ &\approx e^{i\tau\hat{H}(t_0)} e^{i\tau\hat{R}} + \mathcal{O}(\tau^3)\end{aligned}$$

avec $\hat{R} = \frac{\tau}{2}\frac{\partial}{\partial t}\hat{H}(t_0)$ le premier terme correctif de cette approximation.

2.4.2 Hamiltonien constant évalué à $t_0 + \tau$ (fin de l'intervalle)

On prend $\eta = 1$ et on complète l'équation (4).

$$\begin{aligned}\hat{U}(t_0 + \tau, t_0) &\approx e^{i\tau\hat{H}(t_0+\tau)} e^{-\frac{\tau^2}{2}[\hat{T}, \hat{H}(t_0)]} e^{\frac{-i\tau^3}{6}(-2[\hat{T}, [\hat{T}, \hat{H}(t_0)]] - [\hat{H}(t_0), [\hat{T}, \hat{H}(t_0)])}) \\ &\approx e^{i\tau\hat{H}(t_0+\tau)} e^{-i\frac{\tau^2}{2}\frac{\partial}{\partial t}\hat{H}(t_0)} + \mathcal{O}(\tau^3) \\ &\approx e^{i\tau\hat{H}(t_0+\tau)} e^{i\tau\hat{R}} + \mathcal{O}(\tau^3)\end{aligned}$$

avec $\hat{R} = -\frac{\tau}{2}\frac{\partial}{\partial t}\hat{H}(t_0)$ le premier terme correctif de cette approximation.

2.4.3 Hamiltonien constant évalué à $t_0 + \frac{\tau}{2}$ (milieu de l'intervalle)

On prend $\eta = \frac{1}{2}$ et on complète l'équation (4).

$$\begin{aligned}\hat{U}(t_0 + \tau, t_0) &\approx e^{i\tau\hat{H}(t_0+\frac{\tau}{2})} e^{-i\frac{\tau^3}{24}[\hat{T}, [\hat{T}, \hat{H}(t_0)]] - i\frac{\tau^3}{12}[\hat{H}(t_0), [\hat{T}, \hat{H}(t_0)]])} \\ &\approx e^{i\tau\hat{H}(t_0+\frac{\tau}{2})} e^{i\frac{\tau^3}{24}\frac{\partial^2}{\partial t^2}\hat{H}(t_0) + \frac{\tau^3}{12}[\hat{H}(t_0), \frac{\partial}{\partial t}\hat{H}(t_0)]} \\ &\approx e^{i\tau\hat{H}(t_0+\frac{\tau}{2})} e^{i\tau\hat{R}}\end{aligned}$$

avec $\hat{R} = \frac{\tau^2}{24}\frac{\partial^2}{\partial t^2}\hat{H}(t_0) - i\frac{\tau^2}{12}[\hat{H}(t_0), \frac{\partial}{\partial t}\hat{H}(t_0)]$ le premier terme correctif de cette approximation. Logiquement, on peut se dire que d'évaluer l'hamiltonien au milieu de l'intervalle est le meilleur choix à faire, car c'est comme si on prenait l'hamiltonien "moyen" sur l'intervalle pour faire l'approximation.

2.5 Approximation d'une somme d'hamiltoniens

Soit un hamiltonien dépendant du temps de la forme suivante.

$$\hat{H}(t) = a(t)\hat{A} + b(t)\hat{B}$$

\hat{A} et \hat{B} sont deux hamiltoniens indépendants du temps avec des coefficients $a(t)$ et $b(t)$ dépendants du temps. De plus, \hat{A} et \hat{B} sont diagonalisables et ne commutent pas nécessairement. Puis, soit un opérateur d'évolution approximé selon la section 2.4 de la forme

$$\hat{U}(t_0 + \tau, t_0) \approx e^{i\tau\hat{H}(\bar{t})} e^{i\tau\hat{R}} \approx e^{i\tau[a(\bar{t})\hat{A} + b(\bar{t})\hat{B}]} e^{i\tau\hat{R}}$$

où on fixe l'hamiltonien à un temps \bar{t} entre t_0 et $t_0 + \tau$. Il n'est pas encore possible de construire un circuit quantique à partir de l'équation ci-haut du fait que $\hat{H}(\bar{t})$ n'est pas forcément diagonalisable. Pour résoudre ce problème, il faut séparer l'exponentielle en deux.

2.5.1 Approximation 1er ordre

Par la formule de Zassenhaus au 1er ordre (voir Annexe), on a

$$e^{i\tau\hat{H}(\bar{t})} = e^{i\tau[a(\bar{t})\hat{A}+b(\bar{t})\hat{B}]} \approx e^{i\tau a(\bar{t})\hat{A}} e^{i\tau b(\bar{t})\hat{B}} e^{i\tau\hat{S}}$$

$$\hat{U}(t_0 + \tau, t_0) \approx e^{i\tau\hat{H}(\bar{t})} e^{i\tau\hat{R}} \approx e^{i\tau a(\bar{t})\hat{A}} e^{i\tau b(\bar{t})\hat{B}} e^{i\tau\hat{S}} e^{i\tau\hat{R}} \quad (5)$$

avec $\hat{S} = -i\frac{\tau a(\bar{t})b(\bar{t})}{2}[\hat{A}, \hat{B}]$ le premier terme correctif de l'approximation.

2.5.2 Approximation 2e ordre

Par la formule de Zassenhaus au 2e ordre (voir Annexe), on a

$$e^{i\tau\hat{H}(\bar{t})} = e^{i\tau[a(\bar{t})\hat{A}+b(\bar{t})\hat{B}]} = e^{i\tau[b(\bar{t})\hat{B}+a(\bar{t})\hat{A}]} \approx e^{i\frac{\tau b(\bar{t})}{2}\hat{B}} e^{i\tau a(\bar{t})\hat{A}} e^{i\frac{\tau b(\bar{t})}{2}\hat{B}} e^{i\tau\hat{S}}$$

$$\hat{U}(t_0 + \tau, t_0) \approx e^{i\tau\hat{H}(\bar{t})} e^{i\tau\hat{R}} \approx e^{i\frac{\tau b(\bar{t})}{2}\hat{B}} e^{i\tau a(\bar{t})\hat{A}} e^{i\frac{\tau b(\bar{t})}{2}\hat{B}} e^{i\tau\hat{S}} e^{i\tau\hat{R}} \quad (6)$$

avec $\hat{S} = -\frac{\tau^2 a(\bar{t})b(\bar{t})}{24}(b(\bar{t})[\hat{B}, [\hat{B}, \hat{A}]] + 2a(\bar{t})[\hat{A}, [\hat{B}, \hat{A}]])$ le premier terme correctif de l'approximation.

3 Application à MaxCut

Essayons maintenant de résoudre MaxCut à l'aide d'un circuit quantique avec les notions de la section précédente.

3.1 Représentation des couleurs

Notre but étant de trouver des combinaisons de couleurs qui satisfont MaxCut pour un graphe donné, il nous faut d'abord un moyen de les transposer en des états quantiques qui nous seront retournés par l'ordinateur quantique. Puisque deux couleurs sont employées, on peut utiliser une représentation binaire où 0 est assigné à la couleur rouge et 1 à la couleur bleu. De plus, on peut assigner à chaque noeud un nombre entre 0 et $n - 1$ où n est le nombre total de noeuds du graphe. Ainsi, le qubit i représente le noeud i du graphe auquel on lui attribue la valeur binaire appropriée. Pour l'exemple de la figure 1, la combinaison de couleurs s'écrit donc $|01010\rangle$. Ainsi, on pourra facilement transposer les états quantiques mesurés en des colorations pour le graphe en cours de résolution.

3.2 Opérateur de couleur

MaxCut veut minimiser le nombre d'arêtes dans le graphe dont les sommets sont de la même couleur. Puisqu'on agit sur des états quantiques et que la couleur de chaque sommet est encodée dans ces états, il nous faut un opérateur qui soit capable d'extraire la couleur associée à chaque qubit. On cherche donc un opérateur \hat{q}_i appliqué sur un qubit i telle que $\hat{q}_i |\dots 0_i \dots\rangle = 0 |\dots 0_i \dots\rangle$ et $\hat{q}_i |\dots 1_i \dots\rangle = 1 |\dots 1_i \dots\rangle$. Il est trivial de trouver que l'opérateur de couleur est

$$\hat{q}_i = \frac{\hat{I} - \hat{Z}}{2} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \quad (7)$$

3.3 Opérateur de comparaison

Maintenant qu'il est possible d'obtenir la couleur de tous les sommets d'un graphe par l'opérateur de couleur, il est temps de comparer la couleur des sommets de toutes les arêtes. Pour deux sommets i et j unis par une arête dans le graphe, on peut définir l'opérateur de comparaison

$$\hat{c}_{ij} = \hat{I} - \hat{q}_i(\hat{I} - \hat{q}_j) - \hat{q}_j(\hat{I} - \hat{q}_i) = \frac{\hat{I} + \hat{Z}_i \hat{Z}_j}{2} \quad (8)$$

qui vaut 0 quand les sommets ne sont pas de la même couleur et 1 quand les sommets sont de la même couleur. Par exemple,

$$\hat{c}_{03} |\dots 0 \dots 1\rangle = \frac{1}{2}\hat{I} |\dots 0 \dots 1\rangle + \frac{1}{2}\hat{Z}_0 \hat{Z}_3 |\dots 0 \dots 1\rangle = \frac{1}{2} \cdot 1 \cdot 1 + \frac{1}{2} \cdot -1 \cdot 1 = 0$$

$$\hat{c}_{03} |\dots 1 \dots 1\rangle = \frac{1}{2}\hat{I} |\dots 1 \dots 1\rangle + \frac{1}{2}\hat{Z}_0 \hat{Z}_3 |\dots 1 \dots 1\rangle = \frac{1}{2} \cdot 1 \cdot 1 + \frac{1}{2} \cdot -1 \cdot -1 = 1$$

Afin de savoir si une solution est bonne ou mauvaise pour MaxCut, on doit appliquer un opérateur de comparaison sur chaque arête de la solution et sommer le résultat. Les meilleures solutions seront donc celles où la somme des opérateurs de comparaison sur chaque arête sera le plus proche de 0 (ou égale à 0 dans le cas d’une solution parfaite). Donc, on en déduit la fonction de coût \hat{C} pour le graphe en entier

$$\hat{C} = \sum_{i \neq j} \hat{c}_{ij} \quad (9)$$

où on cherche à trouver les combinaisons de couleur $|q\rangle$ qui la minimise le plus possible.

3.4 Hamiltoniens

On souhaite employer l’évolution adiabatique afin de résoudre MaxCut. Pour cela, il nous faut définir un hamiltonien *facile* et un hamiltonien *difficile*.

La construction de l’hamiltonien *facile* peut se faire de plusieurs manières, puisqu’il existe plusieurs hamiltoniens dont l’état fondamental est facilement calculable et transposable en un circuit quantique simple. On décide de prendre l’hamiltonien *facile* dont l’état fondamental est $|++ \dots +\rangle$. L’état peut être fabriqué facilement en appliquant une porte \hat{H} sur chaque qubit. Pour l’exemple à la figure 1, l’hamiltonien *facile* prend la forme

$$\hat{H}_F = -\hat{X}\hat{I}\hat{I}\hat{I}\hat{I} - \hat{I}\hat{X}\hat{I}\hat{I}\hat{I} - \hat{I}\hat{I}\hat{X}\hat{I}\hat{I} - \hat{I}\hat{I}\hat{I}\hat{X}\hat{I}$$

et est facilement généralisable pour n’importe quel graphe. Effectivement, l’hamiltonien possède n chaînes de Pauli de longueur n (n le nombre de noeuds) contenant une unique porte \hat{X} dont le placement est différent pour toutes les chaînes.

La construction de l’hamiltonien *difficile* s’appuie directement sur la fonction de coût calculée à la section précédente (équation (9)). En effet, on construit l’hamiltonien *difficile* comme une somme de chaînes de Pauli où chaque chaîne est un opérateur de comparaison appliqué à une arête du graphe. Ainsi, une porte \hat{Z} est appliquée sur les qubits appropriés de chaque arête. Pour l’exemple de la figure 1, la représentation en somme de chaînes de Pauli de l’hamiltonien *difficile* serait

$$\hat{H}_D = -3\hat{I}\hat{I}\hat{I}\hat{I} + \frac{1}{2}\hat{I}\hat{I}\hat{Z}\hat{Z} + \frac{1}{2}\hat{I}\hat{I}\hat{Z}\hat{Z}\hat{I} + \frac{1}{2}\hat{I}\hat{Z}\hat{Z}\hat{I}\hat{I} + \frac{1}{2}\hat{Z}\hat{Z}\hat{I}\hat{I} + \frac{1}{2}\hat{Z}\hat{I}\hat{I}\hat{Z} + \frac{1}{2}\hat{Z}\hat{I}\hat{Z}\hat{I}$$

On peut décider d’ignorer la chaîne de Pauli $\hat{I}\hat{I} \dots \hat{I}$ tout en obtenant les mêmes solutions au final. En effet, elle ne fait qu’altérer la valeur des énergies propres de \hat{H}_D sans en modifier la distribution (c’est une constante). Finalement, par construction, l’état fondamental de \hat{H}_D sera la/les solution(s) à MaxCut !

3.5 Trajectoire d’évolution

Afin de pouvoir représenter facilement l’évolution de \hat{H}_F vers \hat{H}_D qui prend un temps total T , on peut définir l’hamiltonien dépendant du temps

$$\hat{H}(t) = (1 - s(t))\hat{H}_F + s(t)\hat{H}_D \quad (10)$$

où $t \in [0, T]$. Le paramètre $s(t) \in [0, 1]$ gère la trajectoire d’évolution, c’est-à-dire la façon dont on passe d’un hamiltonien à l’autre. Cette trajectoire peut être choisie arbitrairement, mais il faut s’assurer que $s(t) = 0$ à $t = 0$ et que $s(t) = 1$ à $t = T$ afin d’effectuer l’évolution adiabatique. Quelques exemples de trajectoires sont montrés à la figure 2.

3.6 Évolution adiabatique

On cherche à obtenir l’état fondamental de l’hamiltonien *difficile* qu’on écrit $|\phi_0^h\rangle$. Cela s’effectue par l’opération suivante

$$|\phi_0^h\rangle = |\psi(T)\rangle = \hat{U}(T, 0) |++ \dots +\rangle$$

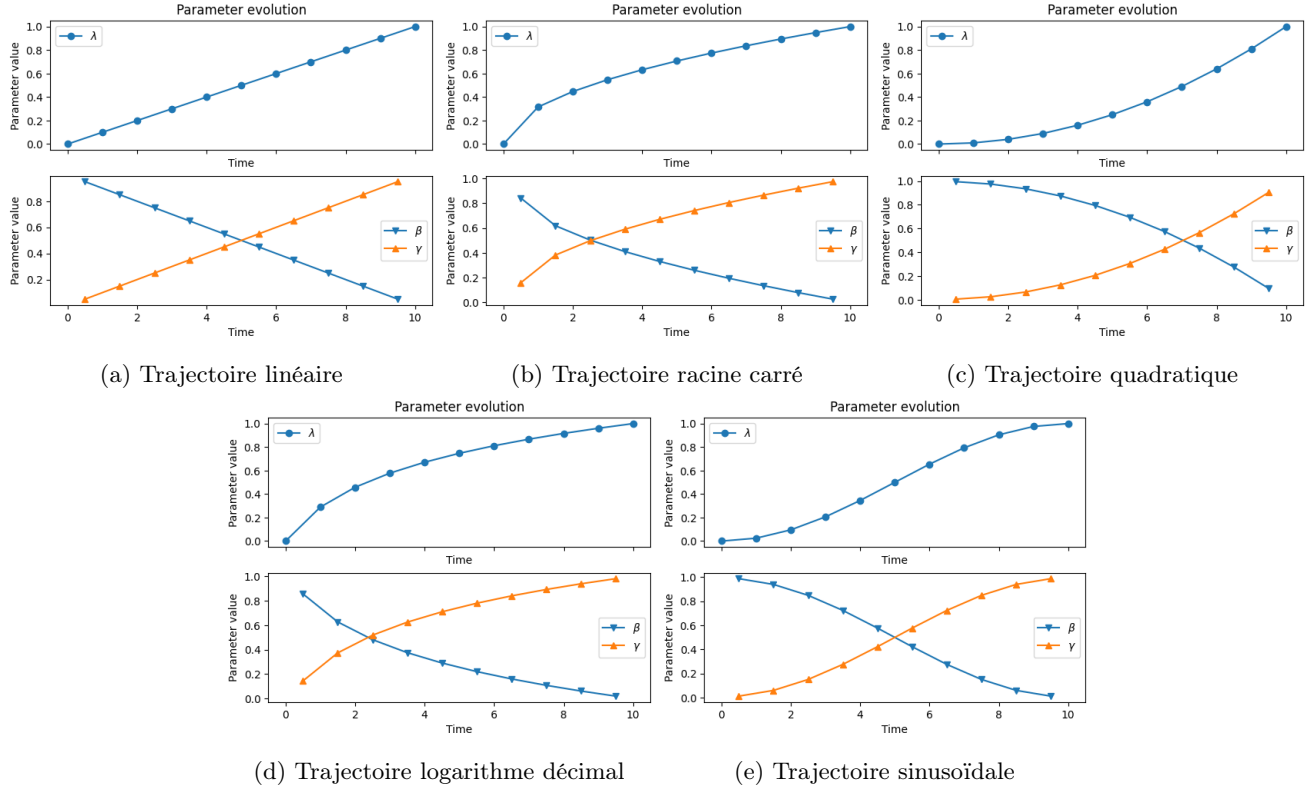


Figure 2 - Différentes trajectoires d'évolution pour 10 étapes et $\delta t = 1$

De manière équivalente, on peut effectuer plusieurs petites évolutions de sorte que T soit séparé en p étapes de longueur τ_p .

$$\begin{aligned}\hat{U}(T, 0) &= \prod_p \hat{U}(t_p + \tau_p, t_p) \\ |\phi_0^h\rangle &= |\psi(T)\rangle = \prod_p \hat{U}(t_p + \tau_p, t_p) |++\dots+\rangle\end{aligned}\quad (11)$$

Cependant, pour effectuer cette opération avec un circuit quantique, il faudra faire deux transformations.

3.6.1 Hamiltonien constant

La première transformation consiste à approximer l'opérateur d'évolution de l'équation (11) en s'inspirant de la section 2.4 pour avoir

$$\hat{U}(t_p + \tau_p, t_p) \approx e^{i\tau_p \hat{H}(\bar{t}_p)} e^{i\tau_p \hat{R}}$$

avec $t_p \leq \bar{t}_p \leq t_p + \tau_p$. Ainsi, pour chaque étape, l'hamiltonien sera évalué à \bar{t}_p afin de le rendre constant sur l'intervalle de temps. L'opérateur \hat{R} est le premier terme correctif, c'est-à-dire l'erreur provoquée par l'approximation rendant l'hamiltonien constant. Il peut être déduit grâce aux équations de la section 2.4 selon le temps \bar{t}_p choisi. Pour des raisons de simplicité, l'opération réellement effectuée par le circuit quantique sera

$$\hat{U}(T, 0) \approx \prod_p e^{i\tau_p \hat{H}(\bar{t}_p)} \approx \prod_p e^{i\tau_p [(1-s(\bar{t}_p))\hat{H}_F + s(\bar{t}_p)\hat{H}_D]}$$

sans le terme correctif de sorte que

$$|\phi_0^h\rangle = |\psi(T)\rangle \approx \prod_p e^{i\tau_p \hat{H}(\bar{t}_p)} |++\dots+\rangle \approx \prod_p e^{i\tau_p [(1-s(\bar{t}_p))\hat{H}_F + s(\bar{t}_p)\hat{H}_D]} |++\dots+\rangle\quad (12)$$

3.6.2 Séparation des hamiltoniens *facile* et *difficile* (1er ordre)

La deuxième transformation consiste à séparer \hat{H}_F et \hat{H}_D en deux exponentielles distinctes. Ainsi, on modifie l'exponentielle de l'équation (12) en s'aidant de la section 2.5.1.

$$e^{i\tau_p[(1-s(\bar{t}_p))\hat{H}_F + s(\bar{t}_p)\hat{H}_D]} \approx e^{i\tau_p(1-s(\bar{t}_p))\hat{H}_F} e^{i\tau_p s(\bar{t}_p)\hat{H}_D} e^{i\tau_p \hat{S}}$$

avec \hat{S} le premier terme correctif. Ainsi, on a

$$\hat{U}(T, 0) \approx \prod_p e^{i\tau_p(1-s(\bar{t}_p))\hat{H}_F} e^{i\tau_p s(\bar{t}_p)\hat{H}_D} e^{i\tau_p \hat{S}}$$

Pour faciliter le visuel, on écrit

$$\gamma_p = \tau_p s(\bar{t}_p), \beta_p = \tau_p(1 - s(\bar{t}_p))$$

de sorte que

$$\hat{U}(T, 0) \approx \prod_p e^{i\beta_p \hat{H}_F} e^{i\gamma_p \hat{H}_D} e^{i\tau_p \hat{S}}$$

Pour des raisons de simplicité, l'opération réellement effectuée par le circuit quantique sera

$$\hat{U}(T, 0) \approx \prod_p e^{i\beta_p \hat{H}_F} e^{i\gamma_p \hat{H}_D}$$

sans le terme correctif afin d'obtenir l'équation du circuit QAOA au 1er ordre

$$|\phi_0^h\rangle = |\psi(T)\rangle \approx \prod_p e^{i\beta_p \hat{H}_F} e^{i\gamma_p \hat{H}_D} |++ \dots +\rangle \quad (13)$$

3.6.3 Séparation des hamiltoniens *facile* et *difficile* (2e ordre)

La deuxième transformation peut aussi se faire en s'aidant de la section 2.5.2.

$$e^{i\tau_p[s(\bar{t}_p)\hat{H}_D + (1-s(\bar{t}_p))\hat{H}_F]} \approx e^{i\frac{\tau_p s(\bar{t}_p)}{2}\hat{H}_D} e^{i\tau_p(1-s(\bar{t}_p))\hat{H}_F} e^{i\frac{\tau_p s(\bar{t}_p)}{2}\hat{H}_D} e^{i\tau_p \hat{S}}$$

avec \hat{S} le premier terme correctif. Ainsi, on a

$$\hat{U}(T, 0) \approx \prod_p e^{i\frac{\tau_p s(\bar{t}_p)}{2}\hat{H}_D} e^{i\tau_p(1-s(\bar{t}_p))\hat{H}_F} e^{i\frac{\tau_p s(\bar{t}_p)}{2}\hat{H}_D} e^{i\tau_p \hat{S}}$$

Pour des raisons de simplicité, l'opération réellement effectuée par le circuit quantique sera

$$\hat{U}(T, 0) \approx \prod_p e^{i\frac{\tau_p s(\bar{t}_p)}{2}\hat{H}_D} e^{i\tau_p(1-s(\bar{t}_p))\hat{H}_F} e^{i\frac{\tau_p s(\bar{t}_p)}{2}\hat{H}_D}$$

sans le terme correctif. On remarque que l'évolution se termine sur l'opérateur d'évolution contenant \hat{H}_D . Ce dernier opérateur peut être omis car il ne fait qu'appliquer une phase sur l'état. Aussi, on peut voir que l'exponentielle à gauche va se combiner avec l'exponentielle à droite de l'étape suivante. Donc, on peut réorganiser l'équation de la façon suivante

$$\hat{U}(T, 0) \approx \prod_p e^{i\tau_p(1-s(\bar{t}_p))\hat{H}_F} e^{i\frac{1}{2}(\tau_p s(\bar{t}_p) + \tau_{p-1} s(\bar{t}_{p-1}))\hat{H}_D}$$

Pour faciliter le visuel, on écrit

$$\gamma_p = \frac{1}{2}(\tau_p s(\bar{t}_p) + \tau_{p-1} s(\bar{t}_{p-1})), \beta_p = \tau_p(1 - s(\bar{t}_p))$$

où on pose que $\tau_{p-1} s(\bar{t}_{p-1}) = 0$ pour le premier p . Ainsi, on obtient la formule pour le circuit quantique de QAOA au 2e ordre

$$|\phi_0^h\rangle = |\psi(T)\rangle \approx \prod_p e^{i\beta_p \hat{H}_F} e^{i\gamma_p \hat{H}_D} |++ \dots +\rangle \quad (14)$$

3.7 Construction du circuit de QAOA

Tout d'abord, il est nécessaire de définir le nombre d'étapes (p), la durée de chaque étape (τ), la trajectoire d'évolution pour $s(t)$ et le graphe que l'on veut résoudre (l'hamiltonien *difficile*). Une fois ces données en main, appliquons des portes H sur tous les qubits afin d'obtenir l'état $|+\dots+\rangle$. Ensuite, appliquons le circuit correspondant à l'évolution adiabatique (équation (13) ou (14)) avec les paramètres β_p et γ_p associés à chaque étape. Puis, finissons par mesurer le circuit. Les états les plus mesurés seront les meilleures solutions pour MaxCut.

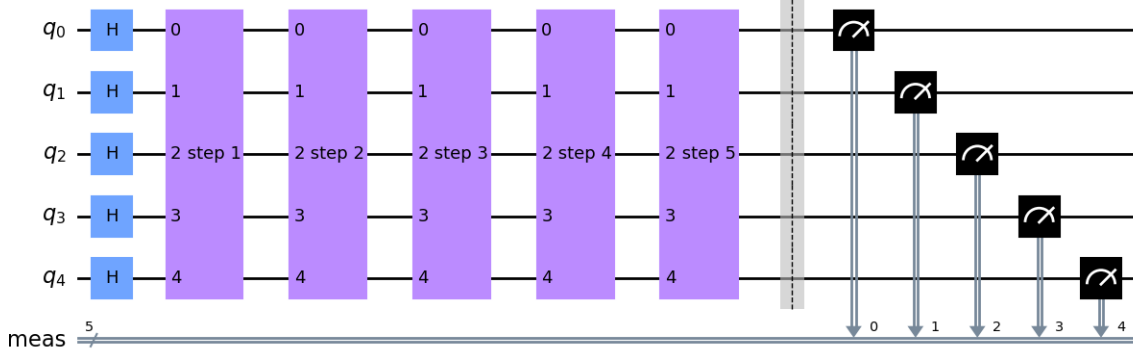


Figure 3 - Circuit QAOA pour le graphe de la maison (voir fig. 1) avec 5 étapes

3.8 Estimation des erreurs

Bien que les ordinateurs quantiques actuels accumulent les erreurs et donnent des résultats parfois douteux, il existe deux types d'erreurs intrinsèques à QAOA.

3.8.1 Erreur adiabatique

L'erreur adiabatique survient lorsque le temps d'évolution T est trop court, c'est-à-dire lorsqu'on effectue l'évolution rapidement. Alors, le théorème adiabatique n'est plus respecté (ce n'est plus *suffisamment lent*) et on quitte l'état fondamental au cours de l'évolution. Pour quantifier cette erreur, on peut regarder le pourcentage de bonnes réponses obtenues au fil du temps pour une évolution spécifique. Ce pourcentage est obtenu en calculant d'abord les réponses exactes en diagonalisant l'hamiltonien *difficile* puis en les comparant avec les états finaux obtenus après l'évolution adiabatique. Prenons le graphique suivant :

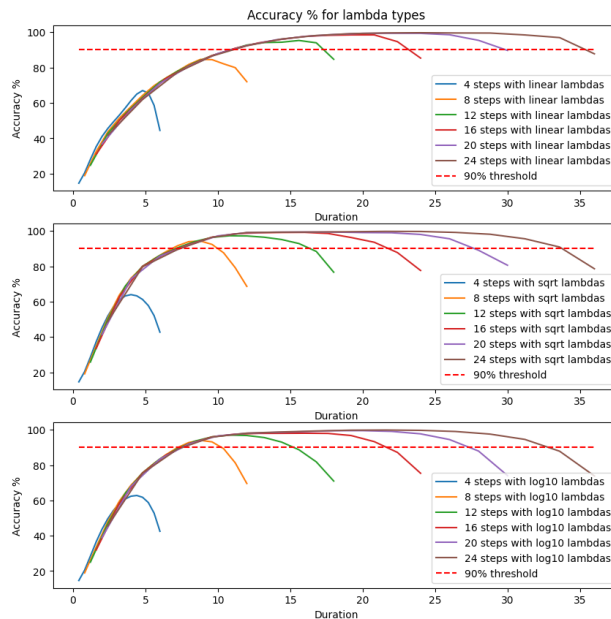


Figure 4 - Pourcentage de bonnes réponses pour différentes trajectoires d'évolution (graphe de la maison)

Ici, on change le nombre d'étapes, le temps de chaque étape ainsi que la façon dont on génère les paramètres β et γ . L'erreur adiabatique se trouve dans la "montée" de chaque courbe. En effet, plus la durée d'évolution et le nombre d'étape augmentent, plus on diminue les chances que le système quitte le fondamental. Puis, à partir d'une certaine durée d'évolution et d'un certain nombre d'étape, on voit le pourcentage plafonner très près de 100%. Il s'agit de la durée minimale d'évolution t_{min} pour être sûr de rester dans l'état fondamental. Ultimement, on aimerait calculer la probabilité de rester de l'état fondamental après chaque étape d'évolution. Cependant, cela s'avère difficile et aucune piste concrète n'a été trouvée. Malgré tout, la figure 4 montre que certains trajectoires d'évolution sont meilleures que d'autres pour un même nombre d'étapes et de temps pour chaque étape (le pourcentage monte plus vite).

3.8.2 Erreurs liées aux approximations

Deux approximations ont été effectuées afin d'obtenir le circuit pour QAOA. Premièrement, on a rendu constant un hamiltonien dépendant du temps. Deuxièmement, on a séparé l'hamiltonien facile et difficile en deux exponentielles distinctes. Pour chacune d'elles, il y avait un terme correctif associé ($e^{i\tau\hat{R}}$ et $e^{i\tau\hat{S}}$) qu'on a décidé d'ignorer. À chaque étape de l'évolution, ces approximations sans le terme correctif impactent la probabilité d'avoir obtenu le bon état après l'étape. Considérons un état obtenu avec les termes correctifs (une meilleure approximation)

$$|\psi(t_p + \tau_p)\rangle = e^{i\beta_p\hat{H}_F} e^{i\gamma_p\hat{H}_D} e^{i\tau_p\hat{S}_p} e^{i\tau_p\hat{R}_p} |\psi(t_p)\rangle$$

et un autre état obtenu sans les termes correctifs (une moins bonne approximation)

$$|\psi'(t_p + \tau_p)\rangle = e^{i\beta_p\hat{H}_F} e^{i\gamma_p\hat{H}_D} |\psi(t_p)\rangle$$

La probabilité que le système soit mesuré dans l'état quantique ayant suivi la meilleure approximation en utilisant la moins bonne pour une étape p sera donnée par

$$\begin{aligned} P_{\text{succès}}^{(p)} &= 1 - |\langle \psi'(t_p + \tau_p) | \psi(t_p + \tau_p) \rangle|^2 = |\langle \psi'(t_p) | e^{i\tau_p\hat{S}_p} e^{i\tau_p\hat{R}_p} |\psi(t_p)\rangle|^2 \\ &\approx 1 - \tau_p^2 (\langle \psi'(t_p) | (\hat{S}_p + \hat{R}_p)^2 | \psi(t_p) \rangle - \langle \psi'(t_p) | \hat{S}_p + \hat{R}_p | \psi(t_p) \rangle^2) \\ &\approx 1 - \tau_p^2 \text{Var}_{\psi(t_p)}[\hat{S}_p + \hat{R}_p] \end{aligned} \quad (15)$$

La probabilité d'avoir obtenu le bon état final est

$$P_{\text{succès}} = \prod_p P_{\text{succès}}^{(p)} \quad (16)$$

***En faisant des tests, je remarque que l'équation (16) donne parfois des résultats douteux. Peut-être que mon implémentation était mauvaise, mais dans le doute l'équation (16) n'est pas incontestable.

De manière plus concrète, ce type d'erreur apparaît à la figure 4 au moment où les courbes descendent. En effet, pour un nombre d'étape donné, plus la durée d'évolution T est grande, plus il faut que τ_p soit grand. En se fiant à l'équation (15), on voit que la probabilité d'obtenir le bon état diminue. Alors, le pourcentage de bonnes réponses obtenues chutent. De point de vue des approximations, cela semble faire du sens, car on approxime un hamiltonien dépendant du temps sur un intervalle de temps par un hamiltonien constant. Plus l'intervalle est grand, moins cette approximation est juste et représentative de la véritable évolution avec un hamiltonien dépendant du temps. Les erreurs liées aux approximations sont alors plus importantes et le pourcentage de bonnes réponses est forcément plus faible.

4 Pseudo-implémentation

Premièrement, il faut réussir à générer des graphes pour lesquels on voudra résoudre MaxCut. Pour ce faire, la librairie python "Networkx" sera d'une grande utilité. En effet, elle permet de générer une grande variété de graphes aléatoires ou même de concevoir son propre graphe.

```

1 import networkx as nx
2
3 graph = nx.house_graph()
4
5 number_of_nodes = graph.number_of_nodes()
6 colors = ['r' for node in graph.nodes()]
7 axes = plt.axes(frameon=False)
8 pos = nx.spring_layout(graph)
9 nx.draw_networkx(graph, node_color=colors, node_size=600, alpha=1, ax=axes, pos=pos)

```

Deuxièmement, il faut être capable de générer l'hamiltonien facile et l'hamiltonien difficile associés au graphe qu'on veut résoudre. La forme des hamiltoniens à la section 3.4 est facilement généralisable pour un graphe quelconque. On peut choisir de vouloir l'hamiltonien difficile avec ou sans la constante $II \dots II$. La représentation des hamiltoniens peut se faire en SparsePauliOp.

```

1 easy_hamiltonian = build_easy_hamiltonian(graph)
2 #hard_hamiltonian = build_hard_hamiltonian_w_constant(graph)
3 hard_hamiltonian = build_hard_hamiltonian_no_constant(graph)

```

Troisièmement, il peut être pertinent de vouloir calculer classiquement les solutions pour MaxCut. Il suffit de trouver le(s) vecteur(s) propre(s) (ils peuvent être dégénérés) de l'hamiltonien difficile associé à l'énergie la plus basse du système. *Astuce : les vecteurs propres de l'hamiltonien difficile possède un seul élément non-nul à l'index i ($i \in [0, 2^n - 1]$). Pour obtenir la représentation des solutions vue à la section 3.1, il suffit de convertir l'index i en sa représentation binaire. Encore une fois, ce procédé est généralisable pour un hamiltonien difficile quelconque.

```

1 exact_solutions = exact_QAOA_solver(hard_hamiltonian)

```

Ensuite, on doit définir la trajectoire d'évolution des paramètres, le nombre d'étapes ainsi que le temps de chaque étape. Avec ces informations, il faut calculer les β et γ associé à chaque étape. Les graphiques de la figure 2 sont des exemples de trajectoires qui peuvent être adoptées.

```

1 number_of_steps = 10
2 time_step = 1
3 function = lambda x : x
4
5 beta_parameters, gamma_parameters = generate_parameters(number_of_steps, time_step, function)

```

Puis, on définit le circuit quantique représentant le parcours d'évolution choisi. Cela peut être fait grâce à l'équation (13) ou (14). La classe PauliEvolutionGate de Qiskit permettra de faire des évolutions temporelles.

```

1 QAOA_circuit = build_QAOA_circuit(easy_hamiltonian, hard_hamiltonian, beta_parameters,
    gamma_parameters)

```

Finalement, on exécute ce circuit sur un backend quelconque, on regarde les états les plus mesurés et on transpose ces états en solutions pour MaxCut. Il est pertinent de tester différentes trajectoires d'évolution pour voir les différents résultats qu'on obtient. Une fois cela fait, l'implémentation de base pour QAOA afin de résoudre MaxCut est terminée! Bravo!

Explorations

1. Transposer les solutions obtenues en une liste de couleurs qu'on peut appliquer sur le graphe original afin de visualiser les différentes solutions.
2. Comparer les résultats obtenus par QAOA avec les solutions exactes calculées classiquement. En déduire le pourcentage de bonnes réponses (combien de "counts" sont dans le bon état par rapport au nombre total de "shots").
3. Sauvegarder les résultats dans des fichiers (simuler QAOA sur un ordinateur classique peut être long pour des graphes avec beaucoup de noeuds et d'arêtes)

4. Faire des tests sur une variété de graphes différents et comparer les résultats. Je recommande les graphes qui ont moins de 8 noeuds sinon c'est très long.
5. Tracer le pourcentage de bonnes réponses obtenues en fonction d'une variété d'évolution (semblable à la figure 4).
6. Comparer les résultats obtenus pour différentes propriétés des graphes (nombre de noeuds, nombre d'arêtes, graphe bipartite/eulérien, ...)
7. Tester QAOA sur un ordinateur quantique. Malheureusement, c'est très mauvais même pour des graphes très simples :(

Annexe

Formule de Zassenhaus (1er ordre) :

$$e^{x\hat{X}+y\hat{Y}} = e^{x\hat{X}}e^{y\hat{Y}}e^{-\frac{xy}{2}[\hat{X},\hat{Y}]} + \mathcal{O}(\tau^3)$$

Formule de Zassenhaus (2e ordre) :

$$e^{i\tau\hat{A}+i\tau\hat{B}} \approx e^{i\frac{\tau}{2}\hat{A}}e^{i\tau\hat{B}}e^{i\frac{\tau}{2}\hat{A}}e^{-i\frac{\tau^3}{24}[\hat{A},[\hat{A},\hat{B}]]-i\frac{\tau^3}{12}[\hat{B},[\hat{A},\hat{B}]]} + \mathcal{O}(\tau^4)$$

Formule de Baker-Campbell-Hausdorff :

$$e^{\hat{X}}e^{\hat{Y}} = e^{\hat{X}+\hat{Y}+\frac{1}{2}[\hat{X},\hat{Y}]+\frac{1}{12}([\hat{X},[\hat{X},\hat{Y}]]-[\hat{Y},[\hat{X},\hat{Y}]])+\dots}$$

Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude envers Maxime Dion qui m'a donné l'opportunité de travailler avec lui sur ce projet. Ta patience et ton soutien ont été des piliers essentiels dans mon apprentissage de nouvelles notions et dans le développement de mes compétences en tant que chercheur. De plus, je souhaite adresser mes remerciements à toute l'équipe de l'AlgoLab pour cette opportunité de stage exceptionnelle, une chance inestimable pour tout étudiant évoluant dans le domaine de l'informatique quantique. Merci également à Simon, Nolan et Florence, mes camarades stagiaires à l'IQ, pour leur compagnie tout au long de cet été enrichissant.

Références

- [1] Tameem Albash and Daniel A. Lidar. Adiabatic quantum computation. *Rev. Mod. Phys.*, 90 :015002, Jan 2018.
 - [2] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Leo Zhou. The Quantum Approximate Optimization Algorithm and the Sherrington-Kirkpatrick Model at Infinite Size. *Quantum*, 6 :759, July 2022.
 - [3] Luis Pedro García a-Pintos, Lucas T. Brady, Jacob Brington, and Yi-Kai Liu. Lower bounds on quantum annealing times. *Physical Review Letters*, 130(14), apr 2023.
 - [4] Rebekah Herrman, Lorna Treffert, Jim Ostrowski, Phil C. Lotshaw, Travis S. Humble, and George Siopsis. Impact of graph structures for qaoa on maxcut. *Quantum Information Processing*, 20(9), 9 2021.
 - [5] Jérémie Roland and Nicolas J. Cerf. Quantum search by local adiabatic evolution. *Physical Review A*, 65(4), mar 2002.
 - [6] Jonathan Wurtz and Peter J. Love. Counterdiabaticity and the quantum approximate optimization algorithm. *Quantum*, 6 :635, January 2022.
- [1] [6] [2] [4] [5] [3]

https://www.youtube.com/watch?v=m1FEgaddEds&ab_channel=QuantumVillage

https://en.wikipedia.org/wiki/Baker%E2%80%93Campbell%E2%80%93Hausdorff_formula