Chapitre 2 : Algorithmes quantiques à base d'oracles

Informatique quantique pour la recherche opérationnelle
Dimitri Watel - ENSIIE

2022

1 Introduction

Définition 1. Un oracle est une boîte noire dont on considère qu'il donne la réponse à un problème en temps constant.

Ces algorithmes peuvent servir dans plusieurs cas:

- On cherche à avoir des informations sur la boîte noire. Elle peut être donnée par un adversaire qui cherche à cacher des informations. On appelle un certain nombre de fois l'oracle pour en savoir plus sur son contenu, on présente ici deux exemples d'algorithmes de la sorte.
- On cherche à résoudre un autre problème à l'aide de la puissance de cette boîte noire. Une fois l'algorithme à base d'oracle conçu, on remplace la boîte noire par un vrai algorithme.

2 Algorithme de Deutsch-Jozsa

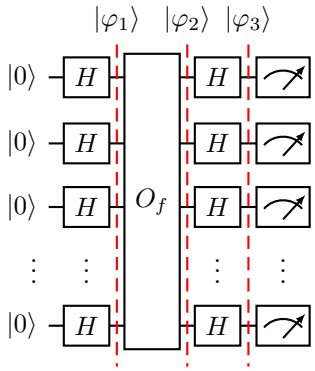
Problème résolu. Soit une fonction $f : \{0,1\}^n \to \{0,1\}$ inconnue dont on sait qu'elle est soit constante, soit équilibrée (la moitié des entrées donnent 0 et l'autre moitié donne 1), est-elle constante ?

Un algorithme classique n'a pas d'autre choix que de tester les entrées pour vérifier s'il y en a deux qui ont des sorties différentes, auquel cas la fonction est équilibrée. Si, pour plus de la moitié des entrées, f donne 0, ou si, pour toutes ces entrées, f donne 1, alors la fonction est constante. Puisqu'il y a 2^n entrées, alors dans le pire cas, on sélectionne $O(2^{n-1}) = O(2^n)$ entrées. L'algorithme de Deutsch-Jozsa est un algorithme quantique qui donne la réponse en appelant f une seule fois.

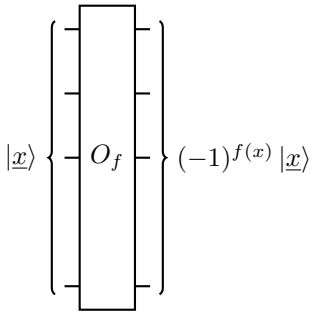
Cet algorithme créé dans les années 90 a pour intérêt de démontrer que l'informatique quantique a plus de puissances que l'informatique classique déterministe. Il n'a pas d'intérêt pratique connu.

2.1 Circuit de l'algorithme

Le circuit de l'algorithme est le suivant. Si on ne mesure que des 0, alors on répond que la fonction est constante.



La porte O_f a la propriété suivante :



La création de l'oracle à partir d'une porte calculant f est détaillée dans la dernière section.

2.2 Lemme préliminaire

Pour étudier cet algorithme et les suivants, on va avoir besoin des définitions et du lemme suivants.

Définition 2. On note $H^{\otimes n}$ la porte H parallélisée n fois, comme dans le circuit précédent.

Définition 3. On note $\underline{x}$ la réprésentation binaire d'un entier x, ou, de manière équivalente, un vecteur de 0 et de 1 qui correspond à cette représentation.

Définition 4. On note $\cdot$ le produit scalaire. $\underline{x} \cdot \underline{y}$ est le produit scalaire entre les vecteurs binaires représentant x et y.

Lemme 2.1. *Soit* $x \in [\![0; 2^n - 1]\!]$, *alors* $H^{\otimes n} |\underline{x}\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} (-1)^{\underline{x}\cdot\underline{y}} |\underline{y}\rangle$

Proof. Soit $\underline{x} = (x_1, x_2, \ldots, x_n)$, ce n-qbit est le produit tensoriel de tous les 1-qbits $|x_i\rangle$. Alors $H^{\otimes n} |\underline{x}\rangle$ revient à appliquer la porte H sur chacun de ces qbits et de faire le produit tensoriel de tous les résultats.

$$H^{\otimes n} |\underline{x}\rangle = \bigotimes_{i=1}^{n} H |x_i\rangle$$

$$H^{\otimes n} |\underline{x}\rangle = \bigotimes_{i=1}^{n} \frac{1}{\sqrt{2}} |0\rangle + (-1)^{x_i} |1\rangle$$

$$H^{\otimes n} |\underline{x}\rangle = \frac{1}{\sqrt{2^n}} \bigotimes_{i=1}^{n} |0\rangle + (-1)^{x_i} |1\rangle$$

Ces produits donnent toutes les combinaisons de produits tensoriels de $|0\rangle$ et de $|1\rangle$. Soit $\underline{y} = (y_1, y_2, \ldots, y_n)$. La composante de $H^{\otimes n} |\underline{x}\rangle$ en $|\underline{y}\rangle$ est un produit $\alpha_y = \prod_{i=1}^{n} \alpha_{y_i}$ de n scalaires égaux à 1 ou -1 selon que $y_i = 0$ ou 1.

$$H^{\otimes n} |\underline{x}\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} \alpha_y |\underline{y}\rangle$$

Si y_i vaut 0 alors le i-ieme scalaire α_{y_i} est 1. Si y_i vaut 1 alors cette valeur vaut $(-1)^{x_i}$. Dans les deux cas, on obtient $\alpha_{y_i} = (-1)^{x_i \cdot y_i}$. Ainsi $\alpha_y = \prod_{i=1}^{n} (-1)^{x_i \cdot y_i} = (-1)^{\sum_{i=1}^{n} x_i \cdot y_i} = (-1)^{\underline{x}\cdot\underline{y}}$.

$$H^{\otimes n} |\underline{x}\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} (-1)^{\underline{x}\cdot\underline{y}} |\underline{y}\rangle$$

□

2.3 Etude de l'algorithme

$$|\varphi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |\underline{x}\rangle$$

$$|\varphi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |\underline{x}\rangle$$

$$|\varphi_3\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} H^{\otimes n} |\underline{x}\rangle$$

$$|\varphi_3\rangle = \frac{1}{2^n} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^n-1} (-1)^{f(\underline{x})+\underline{x}\cdot\underline{y}} |\underline{y}\rangle$$

Cherchons à savoir quelle est la probabilité de ne mesurer que des 0, c'est-à-dire de mesurer $|\underline{y}\rangle = |\underline{0}\rangle$. Le facteur devant $|\underline{0}\rangle$ est

$$\frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(\underline{x})+\underline{x}\odot\underline{0}} = \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(\underline{x})}$$

Si la fonction est constante, cette valeur vaut 1 ou -1. La probabilité de mesurer $|\underline{0}\rangle$ est donc de 1. Sinon la fonction est équilibrée et cette somme vaut 0. La fonction est donc constante si et seulement si on mesure $|\underline{0}\rangle$.

2.4 Remarque

Certains qbits nécessaires au calcul de la porte O_f ont été masqués. Ils n'influencent pas le calcul.

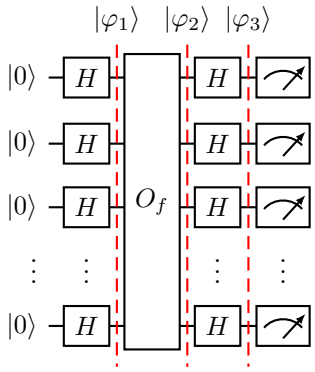
3 Algorithme de Bernstein-Vazirani

Problème résolu. Soit un entier s inconnu et une fonction $f : \{0,1\}^n \to \{0,1\}$ qui à x associe $x \cdot \underline{s}$. Que vaut s ?

Un algorithme classique peut trouver s en faisant n appels à f avec les vecteurs de la base de $\{0,1\}^n$ (contenant un seul 1). L'algorithme de Bernstein-Vazirani permet de trouver s en faisant un seul appel à f.

3.1 Circuit de l'algorithme

De manière assez surprenante, le circuit de l'algorithme est identique à celui de l'algorithme précédent. Ici la mesure renvoie les bits de s.



3.2 Etude de l'algorithme

$$|\varphi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |\underline{x}\rangle$$

$$|\varphi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |\underline{x}\rangle$$

$$|\varphi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{\underline{x}\cdot\underline{s}} |\underline{x}\rangle = H^{\otimes n} |s\rangle$$

$$|\varphi_3\rangle = H^{\otimes n} |\varphi_2\rangle = H^{\otimes n} H^{\otimes n} |s\rangle = |s\rangle$$

La dernière ligne s'obtient en remarquant que H est sa propre inverse. On mesure donc $|s\rangle$ avec une probabilité égale à 1.

4 Algorithme de Grover

Problème résolu. Soit une fonction $f : \{0,1\}^n \to \{0,1\}$. Trouver s tel que $f(s) = 1$?

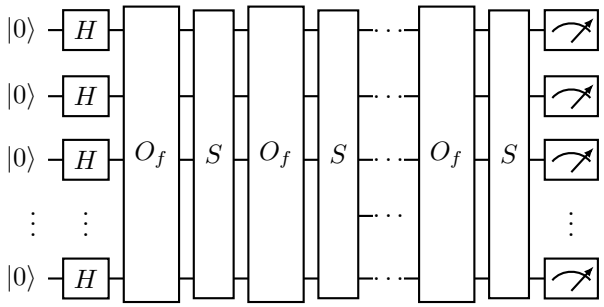
Une application immédiate est une recherche inversée dans une base de données. On pose $f(x) = 1$ si x vérifie une certaine propriété dans une base de données.

Résoudre ce problème rapidement aurait également de nombreux impacts puisqu'on peut réécrire de nombreux autres problèmes sous cette forme. Prenons par exemple le problème classique du sac à dos, on dispose d'un sac, de n objets ayant chacun une certaine valeur et on cherche un sous-ensemble qui rentre dans le sac et dont la valeur dépasse un objectif K fixé. On peut réécrire ce problème en posant n nombres binaires $x_1, x_2, \ldots, x_n$. Le nombre x_i vaut 1 si on choisit de mettre le i-ieme objet dans le sac et 0 sinon. Ainsi un vecteur binaire x représente un ensemble d'objets. On pose ensuite $f(x) = 1$ si les objets rentrent dans le sac et si leurs valeurs dépassent K. Donc trouver x tel que $f(x) = 1$ résout le problème. On peut appliquer cette technique à tous les problèmes d'optimisation.

Dans le cas général, un algorithme classique pour résoudre ce problème consiste à tester tous les x un par un. Il nécessite donc $O(2^n)$ appels à f. L'algorithme de Grover néccessite $O(\sqrt{2^n})$ appels à f pour en arriver au même résultat.

On suppose ici que s est unique mais il fonctionne si plusieurs vecteurs vérifient $f(s) = 1$.

4.1 Circuit de l'algorithme

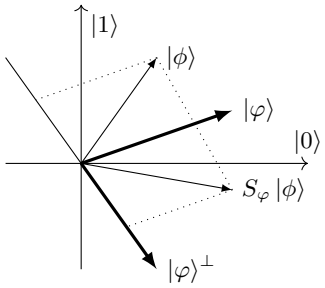


La porte O_f est la même que pour les deux précédents algorithmes. La porte S est une porte de symétrie par rapport au vecteur $H^{\otimes n} |\underline{0}\rangle$. Nous allons dans un premier temps étudier cette porte S puis comment l'algorithme fonctionne.

4.2 Porte de symétrie

S effectue une symétrie par rapport au vecteur $H^{\otimes n} |\underline{0}\rangle$. Nous allons voir dans cette partie comment construire une symétrie par rapport à un vecteur $|\varphi\rangle$ quelconque.

Soit S_φ cette porte. Elle doit avoir deux propriétés : ne pas changer $|\varphi\rangle$ et inverser tout vecteur orthogonal à $|\varphi\rangle$. Autrement dit $S_\varphi |\varphi\rangle = |\varphi\rangle$ et, si $|\phi\rangle \perp |\varphi\rangle$, alors $S_\varphi |\phi\rangle = -|\phi\rangle$. Le dessin ci-contre illustre la symétrie d'axe $|\varphi\rangle$.



On peut construire une telle porte avec cette matrice:

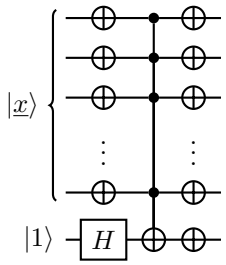
$$S |\varphi\rangle = 2 |\varphi\rangle \langle\varphi| - Id$$

On rappelle que $\langle\varphi|$ est le vecteur transposé conjugué de $|\varphi\rangle$ et que $\langle\varphi| |\varphi\rangle = \langle\varphi|\varphi\rangle = \|\varphi\| = 1$.

On peut montrer que cette matrice est unitaire en la construisant avec des petites portes.

4.2.1 Construire S_0

On va d'abord montrer qu'on peut construire la matrice S_0 de symétrie par rapport à $|\underline{0}\rangle$. Pour cela, on peut utiliser le circuit ci-contre.

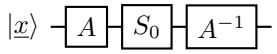


Ce circuit agit comme un oracle où $f(\underline{0}) = 0$ et $f(\underline{x}) = 1$ si $x \neq 0$. On a donc en résultat $|\underline{0}\rangle$ si $x = 0$ et $-|\underline{x}\rangle$ sinon.

La porte CCCC$\cdots$CNOT peut être construite avec des portes CCNOT (Toffoli) et des qbits de calculs supplémentaires en se rappellant qu'un CCCC$\cdots$CNOT est juste une grande porte ET. On peut donc réécrire cette porte avec plein de petites portes ET.

4.2.2 Construire S_φ

Pour construire S_φ, supposons qu'on dispose d'une porte quantique A qui envoie $|\varphi\rangle$ sur $|\underline{0}\rangle$. On rappelle qu'une porte unitaire conserve le produit scalaire. Donc si $|\phi\rangle \perp |\varphi\rangle$ alors $A |\phi\rangle \perp A |\varphi\rangle = |\underline{0}\rangle$. Donc le circuit suivant inverse bien $|\phi\rangle$ et préserve $|\varphi\rangle$.



4.2.3 Construire S

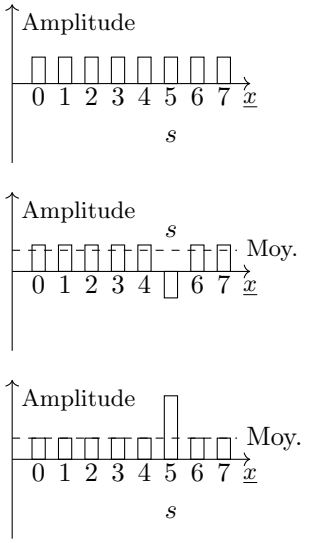
On veut construire la symétrie S_φ où $|\varphi\rangle = H^{\otimes n} |\underline{0}\rangle$. Pour construire cette porte, il suffit d'utiliser $H^{\otimes n}$ à la place de la porte A. En effet, puisque H est sa propre inverse, alors

$$H^{\otimes n}(H^{\otimes n} |\underline{0}\rangle) = |\underline{0}\rangle$$

4.3 Interprétations par amplitude et par rotation

On peut voir l'effet de la porte O_f et de la porte S de plusieurs manières. La première version est une interprétation par amplitude.

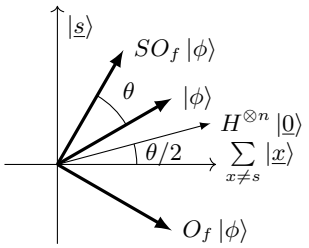
La porte O_f a pour effet d'inverser $|\underline{s}\rangle$ puisque c'est le seul vecteur pour lequel $f(x) = 1$. La porte S est une symétrie par rapport à $H^{\otimes n}|\underline{0}\rangle$ qui est la moyenne de tous les vecteurs. On inverse donc ensuite toutes les amplitudes par rapport à la moyenne. On peut voir sur l'image précédente que l'amplitude de $|\underline{s}\rangle$ augmente naturellement. Cette interprétation est correcte mais elle est difficile à valider par le calcul. Une seconde interprétation plus simple est celle par rotation.



La porte O_f est une symétrie par rapport à l'ensemble des vecteurs orthogonaux à $|\underline{s}\rangle$. La porte S est une symétrie par rapport $H^{\otimes n}|\underline{0}\rangle$. Dans un plan, deux symétries sont équivalentes à une rotation. L'angle de la rotation est le double de l'angle entre les deux axes de symétrie.

Théorème 4.1. *Tous les n-qbits successifs du circuit de Grover sont dans le plan généré par* $|\underline{s}\rangle$ *et* $\sum_{x\neq s} |\underline{x}\rangle$.

Avant la première porte O_f, on a $|\phi\rangle = H^{\otimes n}|\underline{0}\rangle$ et , à chaque itération, la rotation va rapprocher le vecteur $|\phi\rangle$ de $|\underline{s}\rangle$, augmentant ainsi la probabilité de mesurer $|s\rangle$.



4.4 Nombre d'itérations

Il ne faut pas trop faire d'itérations, sinon les rotations vont finir par éloigner $|\varphi\rangle$ de $|s\rangle$. Combien de rotation faut-il faire ? Il faut, pour cela calculer deux angles: l'angle $\theta/2$ entre $H^{\otimes n}|\underline{0}\rangle$ et $\sum_{x\neq s} |\underline{x}\rangle$, et l'angle α entre $H^{\otimes n}|\underline{0}\rangle$ et $|s\rangle$. On peut voir que $\alpha = \frac{\pi-\theta}{2}$.

Le nombre d'itérations est le nombre de rotations d'angle θ qu'on doit faire, soit environ $\lfloor \frac{\alpha}{\theta} \rfloor$.

$$\sin(\frac{\theta}{2}) = \cos(\alpha) = \frac{\langle H^{\otimes n}|\underline{0}\rangle|\underline{s}\rangle}{\|H^{\otimes n}|\underline{0}\rangle \cdot \||\underline{s}\rangle\|\|} = \frac{1}{\sqrt{2^n}}$$

$$\sin\left(\frac{\theta}{2}\right) \simeq \frac{\theta}{2} \Rightarrow \theta \simeq \frac{2}{\sqrt{2^n}} \quad (\theta \text{ est un angle très petit.})$$

$$\alpha \simeq \frac{\pi}{2} - \frac{1}{\sqrt{2^n}}$$

$$\left\lfloor \frac{\alpha}{\theta} \right\rfloor \simeq \frac{\pi}{4}\sqrt{2^n} - \frac{1}{2}$$

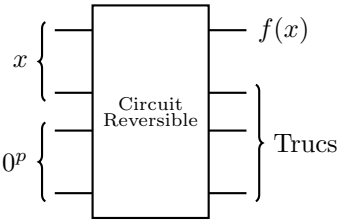
Il faut donc environ $\frac{\pi}{4}\sqrt{2^n}$ itérations pour maximiser les chances de mesurer s.

Théorème 4.2. *A l'issue de l'algorithme, la probabilité de mesurer s est supérieure à* $1 - \frac{4}{2^n}$.

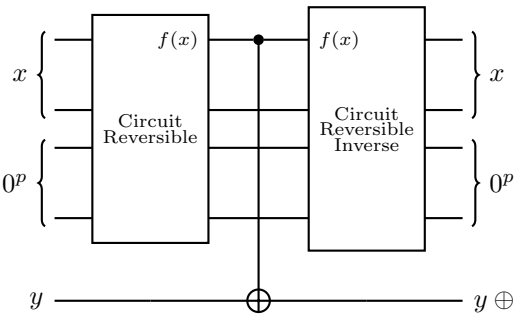
Il faut pour trouver cette probabilité calculer l'amplitude de $|\underline{s}\rangle$ après $\frac{\pi}{4}\sqrt{2^n}$ rotations. Il suffit de calculer le produit scalaire entre le vecteur $|\phi\rangle$ résultat et $|\underline{s}\rangle$ qui s'obtient grâce à l'angle entre ces deux vecteurs.

5 Créer une porte à oracle

Dans cette dernière partie, on montre comment créer la porte O_f utilisée dans ce cours. On prend comme hypothèse qu'on dispose d'un circuit logique qui calcule f. On peut alors montrer qu'il existe un circuit logique réversible qui calcule f.



On peut alors construire le circuit suivant:



Si on donne $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ en entrée y, on obtient en sortie $\frac{1}{\sqrt{2}}(|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle)$. Que $f(x)$ vale 0 ou 1, cette valeur est égale à $(-1)^{f(x)}H|1\rangle$.

Donc, en sortie de ce circuit on a

$$|x\rangle \otimes |0^p\rangle \otimes (-1)^{f(x)}H|1\rangle = (-1)^{f(x)}|x\rangle \otimes |0^p\rangle \otimes H|1\rangle$$

En ignorant les $p+1$ derniers qbits, on obtient bien la porte désirée.