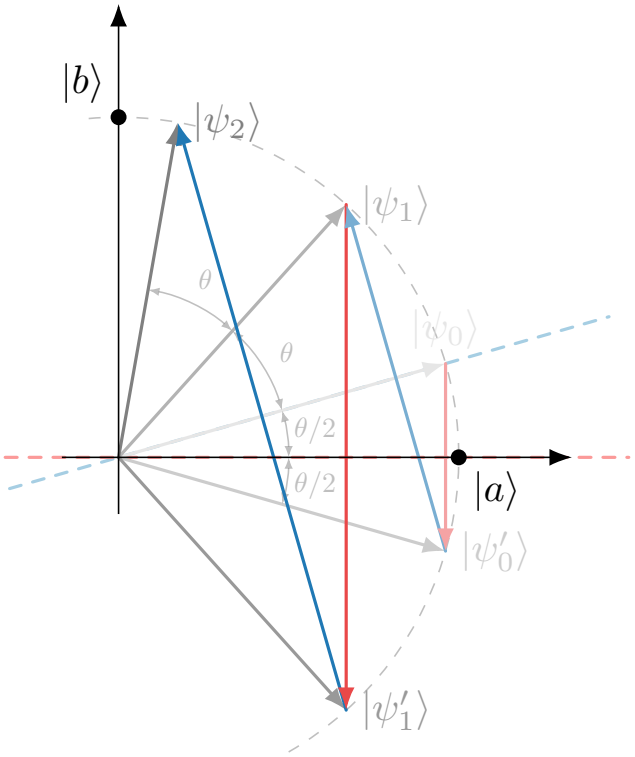
Algorithmes quantiques

Sciences de l'information quantique

par

Maxime Dion



Compilé le 14 janvier 2024
Version numérique régulièrement mise à jour.
Pensez-y avant d'imprimer.

AlgoLab Quantique

Institut Quantique

UdS Université de Sherbrooke

Chapitre 1

Algorithmes quantiques jouets

Les algorithmes jouets sont des algorithmes quantiques qui n'ont pas nécessairement d'applications pratiques. Cependant, ce sont les premiers à avoir été imaginés qui exploitent les phénomènes de superposition, d'intrication et d'interférence afin d'effectuer des opérations plus efficacement que pourrait le faire un ordinateur classique. Ces algorithmes ont contribué de manière importante à l'émergence du domaine de l'information et de la programmation quantique en suscitant l'intérêt des scientifiques.

En plus d'offrir une perspective historique sur les débuts de la programmation quantique, les algorithmes jouets permettent également de se familiariser avec certaines astuces qui peuvent être utilisées pour exploiter les phénomènes quantiques.

A Algorithme de Deutsch

L'algorithme de Deutsch est probablement l'exemple le plus élémentaire d'un algorithme qui utilise le traitement quantique de l'information pour résoudre un problème plus efficacement que la meilleure solution classique. Cet algorithme illustre également comment il est possible de sacrifier une partie de l'information acquise lors d'une solution classique pour répondre plus rapidement une question bien précise.

Nous allons d'abord présenter le problème ainsi que sa solution classique. On présentera ensuite comment le problème peut être posé de manière quantique et nous tenterons de construire de manière intuitive une solution quantique qui exploitera le parallélisme quantique.

L'algorithme de Deustch fait usage de fonctions binaires qui prennent en entrée un bit d'information x (0 ou 1) et qui retournent un bit d'information $f(x)$ (0 ou 1). Il n'existe que quatre fonctions de ce type. Leurs effets sont présentés au tableau 1.1.

	x	0	1
f_0	$f(x)$	0	0
f_1		0	1
f_2		1	0
f_3		1	1

TABLE 1.1 – Les quatre fonctions binaires.

On distingue deux types parmi ces fonctions : les fonctions *constantes* où le résultat est toujours le même ($f(0) = f(1)$) et les fonctions *balancées* dont le résultat est différent pour différentes entrées ($f(0) \neq f(1)$).

A.1 Problème

Le problème de Deutsch se formule ainsi. On vous donne accès à une fonction $f(x)$ sans vous dire de laquelle il s'agit. On vous demande d'identifier si cette fonction est une fonction *constante* ou une fonction

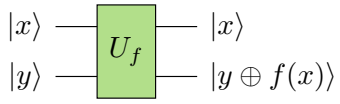


FIGURE 1.1 – Forme de l'oracle pour l'algorithme de Deutsch sous la forme d'une porte quantique à 2 qubits qui évalue la fonction $f(x)$.

balancée en utilisant celle-ci le moins souvent possible.

A.2 Solution classique

La solution classique à ce problème demande d'évaluer la fonction deux fois. On l'évalue d'abord pour $f(x = 0)$ et ensuite pour $f(x = 1)$ pour finalement comparer les deux résultats obtenus. Si les résultats sont identiques, la fonction est constante, sinon elle est balancée.

Remarque

Avec deux évaluations, on possède toute l'information pour identifier laquelle des quatre fonctions il s'agit ; pas seulement si elle est constante ou balancée. En quelque sorte, on a acquis trop d'information. La solution quantique au problème de Deustch que nous allons voir est un bon exemple où l'on sacrifie une certaine quantité d'information (qui nous n'est pas essentielle) pour répondre à une question plus rapidement.

A.3 Solution quantique

Dans la version quantique de ce problème, la fonction est fournie sous la forme d'une porte quantique à deux qubits qui agit de la manière suivante sur les états de base

$$\hat{U}_f \, |y\rangle \, |x\rangle = |y \oplus f(x)\rangle \, |x\rangle \tag{1.1}$$

où x et y peuvent prendre les valeurs 0 et 1. Celle-ci peut également être présentée sous forme du circuit quantique à la figure 1.1. Cette porte utilise l'état $|x\rangle$ pour calculer $f(x)$. Elle modifie ensuite l'état du second qubit en effectuant une addition modulo 2 entre sa valeur initiale y et le résultat de $f(x)$.

Info notation A.1 : L'oracle

La porte $\hat{U}_f$ qui définit le problème de Deustch est souvent appelée un oracle. Le concept d'oracle revient régulièrement pour de nombreux algorithmes quantiques. En règle générale, l'oracle est une porte quantique qui permet de distinguer différents états quantiques. Pour la plupart des algorithmes jouets, l'oracle est souvent présenté comme une boite noire dont on ignore le fonctionnement. Il apparait également nécessaire de déjà connaitre la solution au problème pour pouvoir construire l'oracle, rendant le concept totalement inutile.

Cela n'est (heureusement) qu'une conséquence de la simplicité inhérente des problèmes jouets. Pour des problèmes suffisamment complexes, il sera possible de construire un oracle sans nécessairement être capable de décrire son effet sur les différents états quantiques.

Exemple A.1 : Application de la porte unitaire U_{f_2}

Supposons que la fonction utilisée est f_2 (voir le tableau 1.1). Voici comment cette porte modifie les quatre états de base à deux qubits :

$$\begin{aligned}
\hat{U}_{f_2} \, |0\rangle \, |0\rangle &= |0 \oplus f(0)\rangle \, |0\rangle = |0 \oplus 1\rangle \, |0\rangle = |1\rangle \, |0\rangle \\
\hat{U}_{f_2} \, |0\rangle \, |1\rangle &= |0 \oplus f(1)\rangle \, |1\rangle = |0 \oplus 0\rangle \, |1\rangle = |0\rangle \, |1\rangle \\
\hat{U}_{f_2} \, |1\rangle \, |0\rangle &= |1 \oplus f(0)\rangle \, |0\rangle = |1 \oplus 1\rangle \, |0\rangle = |0\rangle \, |0\rangle \\
\hat{U}_{f_2} \, |1\rangle \, |1\rangle &= |1 \oplus f(1)\rangle \, |1\rangle = |1 \oplus 0\rangle \, |1\rangle = |1\rangle \, |1\rangle \, .
\end{aligned}$$

Exercice A.1 : Application de la porte unitaire U_{f_3}

Évaluez comment la porte unitaire $\hat{U}_{f_3}$ transforme les quatre états de base à deux qubits à l'aide de la fonction f_3. En vous basant sur vos résultats et sur l'exemple précédent, discuter si $\hat{U}_f$ est une transformation unitaire en général.

Remarque

La porte $\hat{U}_f$ ne fait pas simplement évaluer la fonction $f(x)$, mais fait intervenir son évaluation dans le cadre d'une transformation unitaire. En effet, il n'est pas possible de simplement construire une transformation unitaire qui aurait l'effet suivant

$$\hat{U}_f |x\rangle = |f(x)\rangle$$

car celle-ci ne serait pas inversible et donc non unitaire.

En ayant accès à la porte $\hat{U}_f$ on peut exploiter la superposition d'états quantique afin de résoudre le problème de Deustch avec une seule évaluation et donc faire mieux que la solution classique qui en requiert deux.

Parallélisme quantique

Le parallélisme quantique consiste à appliquer une porte quantique sur un état en superposition afin d'effectuer plusieurs évaluations en parallèle. Comme on aimerait évaluer $f(x)$ pour $x = 0$ et $x = 1$ en même temps, on est tenté de placer le premier qubit dans une superposition d'états. Voyons voir comment $\hat{U}_f$ agit sur l'état $|y\rangle |+\rangle$ où le second qubit est laissé dans un état de base arbitraire. Comme l'effet de $\hat{U}_f$ est bien défini sur les états de base, écrivons d'abord,

$$|y\rangle |+\rangle = |y\rangle \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) = \frac{|y\rangle |0\rangle + |y\rangle |1\rangle}{\sqrt{2}}.$$

L'application de $\hat{U}_f$, à l'aide de l'équation 1.1, est alors triviale

$$\hat{U}_f |y\rangle |+\rangle = \frac{\hat{U}_f |y\rangle |0\rangle + \hat{U}_f |y\rangle |1\rangle}{\sqrt{2}} = \frac{|y \oplus f(0)\rangle |0\rangle + |y \oplus f(1)\rangle |1\rangle}{\sqrt{2}}. \tag{1.2}$$

L'état obtenu fait intervenir les deux évaluations $f(0)$ et $f(1)$ en même temps, alors qu'on a utilisé $\hat{U}_f$ qu'une seule fois ! Malheureusement, si l'on mesure les qubits alors que le système est dans cet état, on ne pourra connaitre qu'une seule de ces deux évaluations.

Exemple A.2 : Algorithme de Deustch incomplet

Illustrons ce que la mesure des qubits pour un système dans l'état quantique de l'équation 1.2 pourrait retourner. Dans le cas où $y = 0$, l'état quantique est

$$\hat{U}_f |0\rangle |+\rangle = \frac{|f(0)\rangle |0\rangle + |f(1)\rangle |1\rangle}{\sqrt{2}}.$$

Les résultats de mesure possibles sont

$$|f(0)\rangle |0\rangle \quad \text{et} \quad |f(1)\rangle |1\rangle$$

et les deux ont autant de chances de survenir. Si la mesure du premier qubit retourne 0 le second qubit est projeté dans l'état $f(0)$, et si on obtient 1 le second qubit est projeté dans l'état $f(1)$. Dans tous les cas, la mesure du second qubit ne nous permet d'obtenir d'une seule des deux évaluations de la fonction f. On arrive à une conclusion équivalente pour $y = 1$.

Jeu de phase

Placer le premier qubit dans une superposition d'états n'est pas suffisant pour résoudre le problème de Deustch. Il est nécessaire de faire intervenir l'interférence pour obtenir le résultat recherché en une seule évaluation. Et, qui dit interférence, dit phase. Notre objectif est alors de faire intervenir l'évaluation de $f(x)$ dans une phase quantique.

Il est possible de faire cela en initialisant le second qubit dans l'état $|-\rangle$. Voyons voir comment la porte $\hat{U}_f$ agit sur l'état $|-\rangle|x\rangle$ où c'est maintenant l'état du premier qubit est laissé arbitraire. D'abord, écrivons l'état

$$|-\rangle|x\rangle = \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)|x\rangle = \frac{|0\rangle|x\rangle - |1\rangle|x\rangle}{\sqrt{2}}.$$

Appliquons ensuite l'opérateur unitaire

$$\hat{U}_f|-\rangle|x\rangle = \frac{\hat{U}_f|0\rangle|x\rangle - \hat{U}_f|1\rangle|x\rangle}{\sqrt{2}} = \frac{|0\oplus f(x)\rangle|x\rangle - |1\oplus f(x)\rangle|x\rangle}{\sqrt{2}} = \left(\frac{|0\oplus f(x)\rangle - |1\oplus f(x)\rangle}{\sqrt{2}}\right)|x\rangle$$

où on a refactorisé l'état $|x\rangle$ du premier qubit. On distingue alors deux possibilités pour l'état du second qubit. Soit $f(x) = 0$ et son état reste inchangé ($|-\rangle$), soit $f(x) = 1$ et son état devient

$$\frac{|0\oplus f(x)\rangle - |1\oplus f(x)\rangle}{\sqrt{2}} = \frac{|0\oplus 1\rangle - |1\oplus 1\rangle}{\sqrt{2}} = \frac{|1\rangle - |0\rangle}{\sqrt{2}} = -|-\rangle$$

faisant apparaitre une phase -1. On peut alors résumer l'effet de $\hat{U}_f$ en général comme étant,

$$\hat{U}_f|-\rangle|x\rangle = (-1)^{f(x)}|-\rangle|x\rangle \tag{1.3}$$

où on utilise le fait que

$$(-1)^{f(x)} = \begin{cases} (-1)^0 = 1 & \text{pour } f(x) = 0, \text{ ou} \\ (-1)^1 = -1 & \text{pour } f(x) = 1. \end{cases}$$

Info notation A.2 : Des phases et des exposants

L'utilisation d'exposants appliqués à -1 (ou encore à i) revient régulièrement dans les développements d'algorithmes quantiques. On exploite simplement la structure des nombres complexes (voir l'annexe **??**). En particulier, n'importe quel nombre à l'exposant 0 retourne 1 et n'importe quel nombre à l'exposant 1 retourne ce même nombre

$$b^0 = 1 \qquad\qquad b^1 = b.$$

Jeu d'interférence

Combinons le parallélisme quantique avec le jeu de phase pour voir si cela nous permet de résoudre le problème de Deustch. En d'autres termes, on veut appliquer l'unitaire $\hat{U}_f$ sur l'état $|-\rangle|+\rangle$. Comme on sait écrire son effet lorsque le second qubit est dans l'état $|-\rangle$ (équation 1.3) il est suffisant d'écrire seulement l'état du premier qubit dans la base computationnelle

$$|-\rangle|+\rangle = \frac{|-\rangle|0\rangle + |-\rangle|1\rangle}{\sqrt{2}} \tag{1.4}$$

L'application de l'unitaire, à l'aide de l'équation 1.3, produit alors l'état

$$\hat{U}_f|-\rangle|+\rangle = \frac{(-1)^{f(0)}|-\rangle|0\rangle + (-1)^{f(1)}|-\rangle|1\rangle}{\sqrt{2}} = |-\rangle\left(\frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}}\right). \tag{1.5}$$

On constate que cet état est un état produit et que chaque qubit peut être étudié indépendamment de l'autre. Le second qubit restant dans l'état $|-\rangle$, il est inutile de le mesurer.

En contrepartie, l'état du premier qubit est alors très intéressant, car il fait intervenir les deux évaluations de f en même temps. En particulier, si $f(0) = f(1)$ les phases devant les deux états de base sont les mêmes, alors qu'une phase relative de -1 apparait si $f(0) \neq f(1)$. En d'autres termes, le premier qubit est (à une phase globale près) dans l'état

$$|+\rangle \text{ si } f(0) = f(1) \quad \text{ou} \quad |-\rangle \text{ si } f(0) \neq f(1).$$

Pour distinguer entre ces deux possibilités, on doit simplement introduire une porte Hadamard sur ce premier qubit pour transformer les états $|+\rangle$ et $|-\rangle$ en $|0\rangle$ et $|1\rangle$ respectivement. On peut alors mesurer ce qubit. Si l'on obtient la valeur 0 on sait que $f(0) = f(1)$ et que la fonction est constante, alors que si l'on obtient 1 on sait que $f(0) \neq f(1)$ et que la fonction est balancée.

On peut rendre cela plus explicite en modifiant l'écriture de l'état du premier qubit dans l'équation 1.5. En effet, isolons une phase globale pour écrire,

$$\frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}} \propto \frac{|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle}{\sqrt{2}}$$

ce qui rend plus explicite que cet état est soit $|+\rangle$ ou $|-\rangle$.

Info notation A.3 : Ignorer une phase globale

Nous avons introduit le symbole *proportionnel à* ($\propto$) pour indiquer que la phase globale $(-1)^{f(0)}$ pouvait être ignorée. En effet,

$$\frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}} = (-1)^{f(0)}\left(\frac{|0\rangle + (-1)^{f(1)-f(0)}|1\rangle}{\sqrt{2}}\right) \propto \frac{|0\rangle + (-1)^{f(1)-f(0)}|1\rangle}{\sqrt{2}}.$$

Info notation A.4 : Des phases et des exposants pairs ou impairs

Nous avons également introduit l'addition modulo 2 ($\oplus$ ou XOR) dans l'exposant de la phase relative. En effet, le facteur de phase relatif

$$(-1)^{f(1)-f(0)}$$

ne peut prendre que les valeurs ± 1 dépendamment si l'exposant est pair ou impair. Or, la parité d'une soustraction est la même que celle d'une addition ou d'une addition modulo 2

$$f(1) - f(0) \pmod 2 = f(1) + f(0) \pmod 2 = f(0) \oplus f(1).$$

Finalement, l'application d'une porte Hadamard sur cet état permet d'obtenir l'état final,

$$\hat{H}\frac{|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle}{\sqrt{2}} = |f(0) \oplus f(1)\rangle$$

ou encore, pour l'ensemble de deux qubits,

$$|-\rangle\,|f(0) \oplus f(1)\rangle\,. \tag{1.6}$$

La mesure du premier qubit retourne donc directement

$$f(0) \oplus f(1) = \begin{cases} 0 & \text{si } f(0) = f(1) \text{ ou} \\ 1 & \text{si } f(0) \neq f(1). \end{cases}$$

Ainsi, si ce qubit est mesuré dans l'état 0, on apprend que la fonction est constante. S'il est mesuré dans l'état 1, cela veut dire que la fonction est balancée.

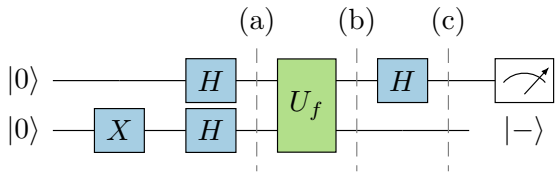


FIGURE 1.2 – Circuit quantique de l'algorithme de Deustch.

Circuit quantique

Le circuit quantique présenté à la figure 1.2 résume l'implémentation de l'algorithme de Deustch. Les états quantiques aux points (a), (b) et (c) sont respectivement donnés par les équations 1.4, 1.5 et 1.6.

Comme on le voit l'algorithme de Deustch permet de déterminer si une fonction $f(x)$ est constante ou balancée en exécutant une seule fois le circuit présenté à la figure 1.2 et, par conséquent, en appelant la fonction qu'une seule fois. Notez également que le résultat obtenu après l'exécution de ce circuit ne permet pas de distinguer de laquelle des 4 fonctions il s'agit. On a donc sacrifié l'information qui nous aurait permis de faire cette distinction pour répondre au problème de Deustch plus rapidement.

Exercice A.2 : Circuits quantiques pour les fonctions binaires

Construisez quatre circuits à deux qubits qui implementent l'évaluation des quatre fonctions $f(x)$ sous la forme donnée à l'équation 1.1 et à la figure 1.1. (*Indice :* Retournez voir la figure **??** pour vous aider.)

B Algorithme de Deutsch-Jozsa

L'algorithme de Deutsch-Jozsa est une tentative de généraliser l'algorithme de Deutsch afin d'impliquer un plus grand nombre de qubits. Cette version de l'algorithme fait intervenir des fonctions qui prennent en entrée n bits qu'on note x_0 à x_{n-1} et qui retourne un seul bit

$$y = f(x_0, x_1, \ldots, x_{n-1}).$$

Comme un ensemble de n bits peut être interprété comme un entier situé entre 0 et $2^n - 1$, on peut également ces fonctions comme prenant en entrée un entier x et retournant un bit

$$y = f(x).$$

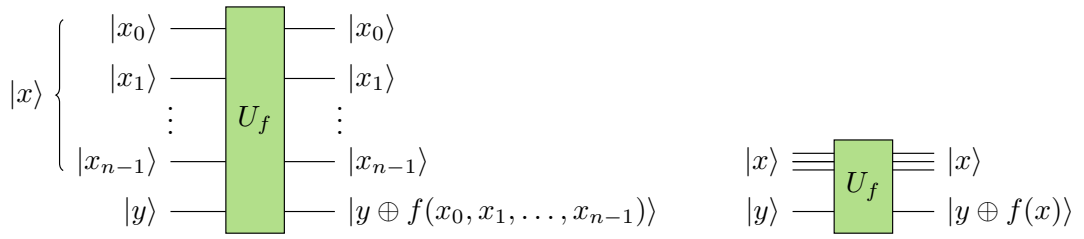
Pour cet algorithme on se limite aux mêmes types de fonction que pour l'algorithme de Deustch, c'est-à-dire des fonctions constantes qui retournent toujours la même chose, et des fonctions balancées qui elles sont garanties de retourner 0 pour la moitié des entrées possibles et 1 pour l'autre moitié. Comme il y a 2^n entrées possibles, une fonction balancée retourne 0 pour $2^n/2 = 2^{n-1}$ entrées possibles et retourne 1 pour les autres 2^{n-1} entrées possibles.

B.1 Problème

Le problème de Deutsch-Jozsa se formule exactement de la même manière que pour l'algorithme de Deustch. On vous donne accès à une fonction $f(x)$ sans vous dire de quel type il s'agit. On vous demande d'identifier si cette fonction est une fonction *constante* ou une fonction *balancée* en utilisant celle-ci le moins souvent possible.

B.2 Solution classique

Comme pour la version originale de l'algorithme de Deutsch, la solution classique de l'algorithme de Deutsch-Jozsa consiste à essayer un certain nombre d'entrées différentes pour vérifier si la fonction utilisée



(a) Représentation avec des bits (b) Représentation avec l'entier x

FIGURE 1.3 – Forme de l'oracle pour l'algorithme de Deutsch-Jozsa sous la forme d'une porte quantique à $n+1$ qubits qui évalue la fonction $f(x)$.

est constante ou balancée. Combien d'entrées différentes doit-on essayer avant d'être complètement certain que la fonction est constante ou balancée ?

D'un côté, du moment que la fonction retourne deux résultats différents, on sait que la fonction balancée (elle ne peut pas être constante). De l'autre côté, si on n'est vraiment pas chanceux, il se peut qu'on essaie les 2^{n-1} entrées qui produisent le même résultat. Il faut donc évaluer la fonction sur $2^{n-1}+1$ entrées possibles pour s'assurer qu'elle est constante.

B.3 Solution quantique

La solution quantique de ce problème est très similaire à celle du problème de Deustch à l'exception que la porte quantique de l'oracle doit maintenant évaluer une fonction de n bits

$$\hat{U}_f \left|y\right\rangle \left|x_{n-1} \cdots x_0\right\rangle = \left|y \oplus f(x_0, \ldots, x_{n-1})\right\rangle \left|x_{n-1} \cdots x_0\right\rangle . \tag{1.7}$$

Le circuit représentant cette porte est illustré à la figure 1.3a. En interprétant les bits comme un entier, le circuit de la porte quantique s'exprime de manière plus compacte (voir figure 1.3b) et l'effet de l'oracle se résume à

$$\hat{U}_f \left|y\right\rangle \left|x\right\rangle = \left|y \oplus f(x)\right\rangle \left|x\right\rangle \tag{1.8}$$

avec

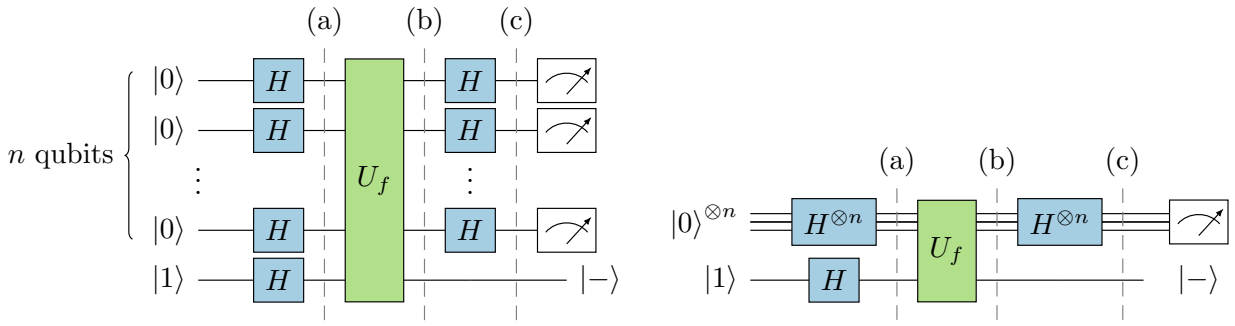
$$\left|x\right\rangle = \left|x_{n-1} \cdots x_0\right\rangle .$$

Encore une fois, nous allons placer le qubit $|y\rangle$ dans l'état $|-\rangle$ de sorte que l'application de cette porte évalue la fonction $f(x)$ dans un facteur de phase

$$\begin{aligned} \hat{U}_f \left|-\right\rangle \left|x\right\rangle &= \hat{U}_f \frac{1}{\sqrt{2}} (\left|0\right\rangle - \left|1\right\rangle) \left|x\right\rangle \\ &= \frac{1}{\sqrt{2}} (\left|0 \oplus f(x)\right\rangle - \left|1 \oplus f(x)\right\rangle) \left|x\right\rangle \\ &= (-1)^{f(x)} \left|-\right\rangle \left|x\right\rangle . \end{aligned} \tag{1.9}$$

Circuit quantique

Nous allons construire le circuit quantique pour l'algorithme de Deustch-Jozsa en nous inspirant directement du circuit quantique pour l'algorithme de Deustch à la figure 1.2. Dans ce cas-ci, pour exploiter le parallélisme quantique, nous allons appliquer des portes Hadamard sur les n premiers qubits qui encode l'entier x. Cela va nous permettre d'évaluer la fonction $f(x)$ pour les 2^n valeurs possibles de x en même temps. Le circuit quantique à utiliser pour l'algorithme de Deustch-Jozsa est illustré à la figure 1.4.



(a) Représentation avec des bits (b) Représentation avec un entier

FIGURE 1.4 – Circuit quantique de l'algorithme de Deustch-Jozsa.

Transformation unitaire

Voyons voir comment ce circuit modifie l'état des qubits. Procédons étape par étape pour bien comprendre comment fonctionne l'algorithme de Deustch-Jozsa. D'abord, les portes Hadamard appliquées sur les n premiers qubits préparent une superposition uniforme de tous les états de base [1]

$$\hat{H}^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle . \tag{1.10}$$

Une porte Hadamard est également appliquée sur le dernier qubit (initialement dans l'état $|1\rangle$) pour le placer dans l'état $|-\rangle$. L'état quantique du système au point (a) est donc

$$|\psi^{(a)}\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |-\rangle |x\rangle . \tag{1.11}$$

Ensuite, on applique la porte $\hat{U}_f$ à cet état. L'état quantique devient

$$|\psi^{(b)}\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} \hat{U}_f |-\rangle |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |-\rangle |x\rangle \tag{1.12}$$

où on a utilisé l'équation 1.9. Il ne reste qu'à appliquer une transformation d'Hadamard sur les n premiers qubits. Nous devons ici faire usage de l'équation suivante

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

qui décrit l'effet d'une transformation Hadamard appliquée à n'importe quel état de base. Consultez l'annexe A.1 pour une démonstration. En résumé, la transformation Hadamard d'un état de base produit une superposition uniforme de tous les états de base où chacun d'eux est affecté par une phase qui dépend du produit scalaire (modulo 2) entre les bits de l'entier z et les bits de l'entier x de l'état de base initial.

En appliquant cette transformation à l'état $|\psi^{(b)}\rangle$ on obtient l'état quantique final [2]

$$\begin{aligned} |\psi^{(c)}\rangle &= |-\rangle \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \hat{H}^{\otimes n} |x\rangle \\ &= |-\rangle \frac{1}{2^n} \sum_{z=0}^{2^n-1} \sum_{x=0}^{2^n-1} (-1)^{x \cdot z \oplus f(x)} |z\rangle . \end{aligned} \tag{1.13}$$

1. Voir l'équation A.3 à l'annexe A.1 pour vous en convaincre.
2. Consultez l'info notation A.4 si la présence du $\oplus$ dans le facteur de phase vous laisse perplexe.

Résultats de mesure

Voyons maintenant comment les résultats de mesure nous informent si la fonction $f(x)$ est balancée ou constante. Fait étonnant, il est nécessaire de considérer uniquement la probabilité de mesurer les n premiers qubits dans l'état $|0\rangle$. Avant la mesure, les n premiers qubits sont, à priori dans un état de superposition de tous les états de base

$$\sum_{z=0}^{2^n-1} \alpha_z |z\rangle = \frac{1}{2^n} \sum_{z=0}^{2^n-1} \sum_{x=0}^{2^n-1} (-1)^{x\cdot z \oplus f(x)} |z\rangle .$$

Pour calculer la probabilité de mesurer tous les qubits dans l'état 0, on doit d'abord obtenir l'amplitude de probabilité devant l'état $|z=0\rangle$. On l'obtient facilement comme étant

$$\alpha_0 = \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)}.$$

Dans l'éventualité où la fonction est constante, elle est indépendante de x. Notons sa valeur simplement par f. Dans ce cas,

$$\alpha_0 = \frac{(-1)^f}{2^n} \sum_{x=0}^{2^n-1} 1 = \frac{(-1)^f}{2^n} 2^n = (-1)^f \quad \text{et} \quad p_0 = |\alpha_0|^2 = 1.$$

Dans l'éventualité où la fonction est balancée, la somme sur x comporte autant de termes positifs que négatifs ; elle est donc nulle. Dans ce cas,

$$\alpha_0 = \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} = \frac{1}{2^n}(2^n/2 - 2^n/2) = 0 \quad \text{et} \quad p_0 = |\alpha_0|^2 = 0.$$

En résumé, l'algorithme de Deustch permet de distinguer en une seule exécution du circuit de la figure 1.4 si la fonction $f(x)$ est constante ou balancée. Si tous les qubits sont mesurés dans l'état $|0\rangle$, la fonction est constante, dans tous les autres cas elle est balancée.

L'algorithme de Deustch-Jozsa est plutôt contraignant au niveau du type de fonctions qu'il peut considérer. En particulier, il n'existe seulement que deux fonctions constantes, peu importe le nombre de bits d'entrée.

C Algorithme de Bernstein-Vazirani

Supposons que vous participiez à un jeu où vous devez deviner un nombre s choisi au hasard entre 0 et 15 inclusivement. Pour le découvrir, vous avez le droit de poser au meneur de jeu des questions qui se répondent par oui ou non. Quelle stratégie devriez-vous utiliser pour déterminer de quel nombre il s'agit en posant le moins de questions possible ?

D'une certaine manière, l'algorithme de Bernstein-Vazirani permet de déterminer ce nombre en une seule question ! Un autre aspect intéressant de cet algorithme est qu'il utilise exactement le même circuit que l'algorithme de Deustch-Jozsa.

C.1 Problème

Pour ce problème, le nombre à deviner est noté s. Pour cet exemple, on suppose qu'il est situé entre 0 et 15, mais dans un cas général il serait situé entre 0 et $2^n - 1$ où n est un nombre de bits donné. Ce nombre est encodé dans une fonction du type

$$f_s(x) = s \cdot x \pmod 2$$

qui prend en entrée un entier x. On l'écrit sous la forme binaire

$$f_s(x_0, x_1, \ldots, x_{n-1}) = s_0x_0 + s_1x_1 + \ldots + s_{n-1}x_{n-1} \pmod 2.$$

Celle-ci effectue un produit scalaire entre les bits des entiers x et s et applique une opération modulo 2 pour retourner 0 ou 1. En d'autres termes, ce type de fonction compte le nombre de 1 que x a en commun (au même endroit) avec s et retourne 0 si ce nombre est pair ou 1 s'il est impair.

C.2 Solution classique

Une approche systématique pour identifier l'entier s est d'utiliser des entiers qui sont caractérisés par une décomposition binaire qui ne comporte qu'un seul 1. Par exemple, en utilisant

$$x = 8 \quad (x_3x_2x_1x_0 = 1000)$$

on peut confirmer la valeur du bit le plus à gauche pour l'entier s. En effet, si la fonction retourne 0, c'est que ce bit est 0, sinon ce bit est 1. L'utilisation de $x = 8$ revient à demander « Est-ce que s est plus grand ou égal à 8 ?». En effet, tous les nombres plus grands ou égaux à 8 ont un 1 à cet endroit dans leur représentation binaire.

Pour un nombre s ayant une représentation binaire d'au maximum 4 bits, il faudra appeler la fonction 4 fois. Le tableau 1.2 dresse l'exemple des résultats obtenus dans le cas où $s = 6$ (0110) et dresse la liste des possibilités à chaque étape.

x	$s = 0110$	$f_s(x)$	Possibilités
8	1000	0	0, 1, 2, 3, 4, 5, 6 ou 7
4	0100	1	4, 5, 6, 7
2	0010	1	6 ou 7
1	0001	0	6

TABLE 1.2 – Exemple d'élimination pour le problème de Bernstein-Vazirani

De manière générale, pour un nombre situé entre 0 et $2^n - 1$, il faudra appeler la fonction n fois pour déterminer les n bits qui composent sa représentation binaire.

C.3 Solution quantique

Pour solutionner le problème de Bernstein-Vazirani avec un algorithme quantique on utilise exactement le même circuit quantique que pour l'algorithme de Deustch-Jozsa, tel qu'illustré à la figure 1.4. La porte quantique de l'oracle doit alors avoir l'effet suivant

$$\hat{U}_f \left|y\right\rangle \left|x\right\rangle = \left|y \oplus f_s(x)\right\rangle \left|x\right\rangle . \tag{1.14}$$

Comme le circuit quantique et la forme de l'oracle sont identiques à ceux de l'algorithme de Deustch-Jozsa, on peut facilement comprendre le fonctionnement de l'algorithme de Bernstein-Vazirani en réutilisant l'équation 1.13. En y remplaçant simplement la fonction $f_s(x)$ on obtient l'état quantique suivant

$$\left|\psi^{(c)}\right\rangle = \left|-\right\rangle \frac{1}{2^n} \sum_{x=0}^{2^n-1} \sum_{z=0}^{2^n-1} (-1)^{x\cdot z \oplus x\cdot s} \left|z\right\rangle . \tag{1.15}$$

En effectuant une simple mise en évidence des produits scalaires dans le facteur de phase, l'état quantique des n premiers qubits peut s'écrire

$$\sum_{z=0}^{2^n-1} \alpha_z \left|z\right\rangle = \frac{1}{2^n} \sum_{z=0}^{2^n-1} \sum_{x=0}^{2^n-1} (-1)^{x\cdot(z\oplus s)} \left|z\right\rangle .$$

L'amplitude de probabilité devant l'état $|z\rangle$ est donc,

$$\alpha_z = \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{x\cdot(z\oplus s)}.$$

On peut montrer que toutes ces amplitudes sont nulles sauf celle pour $z = s$ qui est égale à 1. Pour démontrer cela, supposons que les représentations binaires de z et s diffèrent pour un seul bit. L'addition modulo 2 $z \oplus s$ possède donc un 1 à cet endroit. Parmi tous les entiers x qui font partie de la somme, il y en a autant qui ont un 0 qu'il y en a qui ont un 1 à cet endroit. Il y a donc autant de phases positives que négatives et la somme est nulle. Ce raisonnement tient pour toutes les valeurs de z sauf pour $z = s$. Pour celui-ci, on obtient

$$\alpha_s = \frac{1}{2^n} \sum_{x=0}^{2^n-1} 1 = 1.$$

et on a 100% des chances de mesurer l'état $|s\rangle$. Ce faisant, on peut déterminer le nombre s en exécutant ce circuit qu'une seule fois.

S Solutions aux exercices

Exercice A.1 : Application de la porte unitaire U_{f_3}

La porte unitaire $\hat{U}_{f_3}$ transforme les quatre états de base à deux qubits de la manière suivante

$$\begin{aligned}
\hat{U}_{f_3}|0\rangle|0\rangle &= |0 \oplus f(0)\rangle|0\rangle = |0 \oplus 1\rangle|0\rangle = |1\rangle|0\rangle \\
\hat{U}_{f_3}|0\rangle|1\rangle &= |0 \oplus f(1)\rangle|1\rangle = |0 \oplus 1\rangle|1\rangle = |1\rangle|1\rangle \\
\hat{U}_{f_3}|1\rangle|0\rangle &= |1 \oplus f(0)\rangle|0\rangle = |1 \oplus 1\rangle|0\rangle = |0\rangle|0\rangle \\
\hat{U}_{f_3}|1\rangle|1\rangle &= |1 \oplus f(1)\rangle|1\rangle = |1 \oplus 1\rangle|1\rangle = |0\rangle|1\rangle\,.
\end{aligned}$$

Comme chaque état de base apparait une seule fois dans les résultats obtenus à partir des quatre états de base ($\hat{U}_{f_3}$ agit comme une bijection), et que ceux-ci sont normalisés, $\hat{U}_{f_3}$ doit être unitaire.

Chapitre 2

Algorithme de Grover et amplification d'amplitude

Le parallélisme quantique émerge de la facilité avec laquelle on peut préparer des états de superposition uniforme de tous les états de base d'un ordinateur quantique. Dans ce chapitre, on considère des algorithmes qui exploitent cela en ajoutant une distinction entre deux familles d'états de sorte que l'état de superposition uniforme peut être écrit comme une combinaison linéaire des deux superpositions uniformes, une pour chaque famille. L'algorithme et de Grover et l'amplification d'amplitude reposent sur ce concept.

A Algorithme de Grover

L'algorithme de Grover est souvent présenté comme offrant un avantage quadratique pour effectuer une recherche dans une base de données non ordonnée. En pratique, l'algorithme de Grover permet d'amplifier les probabilités de mesurer un système quantique dans des états qui répondent à certains critères.

Pour bien comprendre le fonctionnement, mais aussi l'utilité de cet algorithme, nous allons d'abord formuler un problème générique qui pourrait traduire une recherche dans une base de données non ordonnée, mais aussi d'autre type de problèmes comme ceux dits de satisfiabilité (SAT).

Nous présenterons ensuite l'algorithme en introduisant l'état initial et ses deux sous-routines : l'oracle quantique et le diffuseur quantique. Nous décrirons comment ceux-ci modifient les états quantiques. Nous pourrons ensuite expliquer comment cet algorithme fonctionne et comment il peut être utilisé pour amplifier des amplitudes de probabilités et résoudre différents types de problèmes.

A.1 Problème générique

Les problèmes pouvant être résolus par l'algorithme de Grover peuvent tous être posés de la manière suivante. Supposons une fonction $f(x)$ qui peut prendre en entrée des valeurs discrètes, par exemple des entiers ($x \in \mathbb{N}$). Cette fonction retourne 1 pour quelques-unes des entrées possibles et retourne 0 pour toutes les autres. On veut trouver au moins une entrée valide $x^\star$ de sorte que

$$f(x^\star) = 1. \tag{2.1}$$

S'il y a N entrées possibles, et M entrées valides, la probabilité de trouver une entrée valide au hasard est

$$p_{\text{hazard}} = \frac{M}{N}.$$

Il faudra donc en moyenne environ $N/2M$ tentatives aléatoires pour trouver une solution. L'algorithme de Grover vise à amplifier cette probabilité au-dessus de p_{hazard} et d'ainsi réduire le nombre de tentatives nécessaires pour trouver une solution valide.

Exemple A.1 : Recherche dans une liste non-ordonnée

Supposons qu'on vous demande de trouver à qui appartient un numéro de téléphone t. Vous avez accès à un bottin téléphonique. Ce bottin agit comme une base de données ordonnée pour les noms : ils sont présentés en ordre alphabétique. Par contre, c'est une base de données non ordonnée pour les numéros de téléphone. Ce problème est sous la forme 2.1 avec une fonction

$$f_t(x)$$

qui retourne 1 si le numéro de téléphone t appartient à la personne x.

A.2 État de superposition uniforme

L'algorithme de Grover tourne autour de l'état de superposition uniforme qui consiste en un produit tensoriel de plusieurs qubits placé dans l'état $|+\rangle$. On note cet état

$$|s\rangle = |+\rangle \otimes |+\rangle \otimes \cdots \otimes |+\rangle = |+\rangle^{\otimes n} . \tag{2.2}$$

où la notation $^{\otimes n}$ indique qu'on répète un produit tensoriel n fois. L'état de superposition uniforme fait intervenir tous les états de base $|x\rangle$ du système de n qubits et peut être écrit (voir équation A.3)

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

Chaque état de base correspond alors à une entrée possible au problème.

A.3 État initial

Comme plusieurs circuits qui tentent de tirer avantage du parallélisme quantique, le circuit quantique de l'algorithme de Grover débute avec l'état de superposition uniforme. On peut préparer cet état en appliquant une série de portes Hadamard sur chacun des qubits

$$|\psi_0\rangle = |s\rangle = \hat{H}^{\otimes n} |\mathbf{0}\rangle$$

où $|\mathbf{0}\rangle$ est l'état dans lequel tous les qubits sont dans l'état $|0\rangle$.

À ce point, la mesure des qubits retourne un état $|x\rangle$ avec une probabilité $1/2^n = 1/N$ et les probabilités que cet état corresponde à une solution valide sont les mêmes qu'un choix au hasard. La suite de l'algorithme vise à augmenter les probabilités de mesure pour les états $|x\rangle$ qui correspondent à des solutions valides.

A.4 Deux ensembles d'états

Pour un problème donné, on distingue deux ensembles complémentaires d'états. D'abord, l'ensemble B qui contient les bons états, ceux qui sont valides, et ensuite l'ensemble A qui contient les autres, ceux qui ne sont pas valides

$$f(x) = 1 \text{ pour } x \in B \text{ et } f(x) = 0 \text{ pour } x \in A.$$

Le nombre d'états dans B est donc $|B| = M$ et donc le nombre d'états dans A est $|A| = N - M$.

Une fois cette distinction faite entre ces deux ensembles d'états, on peut réécrire l'état initial, l'état de superposition uniforme comme deux sommes, une pour l'ensemble A et une pour l'ensemble B

$$|\psi_0\rangle = |s\rangle = \frac{1}{\sqrt{N}} \sum_{x\in A} |x\rangle + \frac{1}{\sqrt{N}} \sum_{x\in B} |x\rangle .$$

Cela est possible, car l'union des ensembles A et B couvre tous les états de base contenus dans $|s\rangle$. Il est alors possible d'écrire le même état comme une superposition de deux états

$$|\psi_0\rangle = \cos(\theta/2)\,|a\rangle + \sin(\theta/2)\,|b\rangle \tag{2.3}$$

où l'angle θ sera déterminé plus tard et où on a défini les états normalisés $|a\rangle$ et $|b\rangle$ de sorte que

$$\cos(\theta/2)\,|a\rangle = \frac{1}{\sqrt{N}}\sum_{x\in A}|x\rangle \quad \text{et} \quad \sin(\theta/2)\,|b\rangle = \frac{1}{\sqrt{N}}\sum_{x\in B}|x\rangle\,. \tag{2.4}$$

Les états $|a\rangle$ et $|b\rangle$ apparaissant donc comme des superpositions uniformes de tous les états de base contenus dans A et B respectivement.

A.5 Espace à deux dimensions

Par construction, les états $|a\rangle$ et $|b\rangle$ sont orthogonaux

$$\cos(\theta/2)\sin(\theta/2)\,\langle a|b\rangle = \frac{1}{N}\sum_{x\in A, y\in B}\langle x|y\rangle = 0.$$

En effet, tous les états de base $|x\rangle$ contenus dans A sont orthogonaux aux états de base $|y\rangle$ contenus dans B. Ensuite, pour assurer la normalisation des états $|a\rangle$ et $|b\rangle$ on doit fixer l'angle θ. En effet, pour $|a\rangle$ et $|b\rangle$ on a

$$\cos^2(\theta/2)\,\langle a|a\rangle = \frac{1}{N}\sum_{y\in A}\sum_{x\in A}\langle y|x\rangle \quad \text{et} \quad \sin^2(\theta/2)\,\langle b|b\rangle = \frac{1}{N}\sum_{y\in B}\sum_{x\in B}\langle y|x\rangle\,.$$

Comme $\langle x|y\rangle = \delta_{xy}$, les sommes sont égales au nombre d'éléments dans A et B respectivement, c'est-à-dire $M-N$ et M. En imposant $\langle a|a\rangle = \langle b|b\rangle = 1$, on déduit alors que

$$\cos^2(\theta/2) = \frac{N-M}{N} \quad \text{et} \quad \sin^2(\theta/2) = \frac{M}{N}.$$

Ces deux expressions sont équivalentes et permettent de fixer la valeur de l'angle θ. Notons également que $\sin^2(\theta/2)$ correspond au carré de l'amplitude devant l'état $|b\rangle$ ainsi que la probabilité initiale d'obtenir un bon état lors de la mesure des qubits.

Les états $|a\rangle$ et $|b\rangle$ forment donc une base d'un espace à deux dimensions et on peut placer l'état $|s\rangle$ dans cet espace, comme cela est fait à la figure 2.1. On voit que si le nombre de solutions M est petit par rapport à N l'angle θ sera également petit. Cela fait en sorte que l'état $|s\rangle$ est proche de l'état $|a\rangle$.

Remarque

Si l'état $|s\rangle$ est proche de l'état $|b\rangle$, cela implique qu'il y a plus d'éléments dans B que dans A. Il y a donc plus d'états qui correspondent à des solutions que d'états qui n'en sont pas. Le problème est alors facile à résoudre classiquement et il est inutile de considérer l'algorithme de Grover.

A.6 Amplification d'amplitude

L'idée fondamentale derrière l'algorithme de Grover est de préparer un état quantique où les états qui correspondent à des solutions valides au problème ont de fortes chances d'être obtenus lors de la mesure des qubits. En d'autres termes, on veut prendre l'état initial $|\psi_0\rangle = |s\rangle$ qui est proche de l'état $|a\rangle$ et le transformer de manière à le rapprocher de l'état $|b\rangle$. Cela aura pour effet d'amplifier les probabilités de mesurer un état de l'ensemble B, c'est-à-dire un état valide.

Pour y arriver, l'algorithme utilise une série de réflexions qui ont lieu dans l'espace à deux dimensions définies par les états $|a\rangle$ et $|b\rangle$. Ces réflexions seront effectuées par deux opérateurs, l'oracle et le diffuseur et leurs effets sont illustrés à la figure 2.2. Décrivons d'abord l'effet voulu pour ces deux opérateurs et nous verrons aux sections suivantes comment les traduire en autant de circuits quantiques.

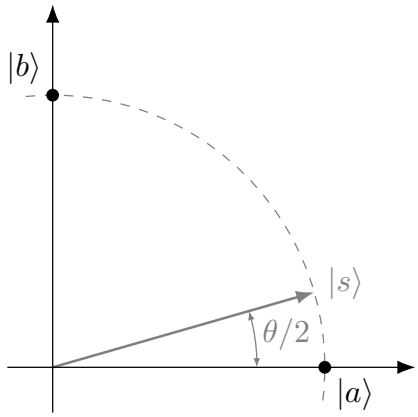


FIGURE 2.1 – Espace à deux dimensions où l'algorithme de Grover prend place. L'objectif de l'algorithme de Grover est de transformer un système quantique à partir de l'état $|s\rangle$ de manière à se rapprocher de l'état $|b\rangle$.

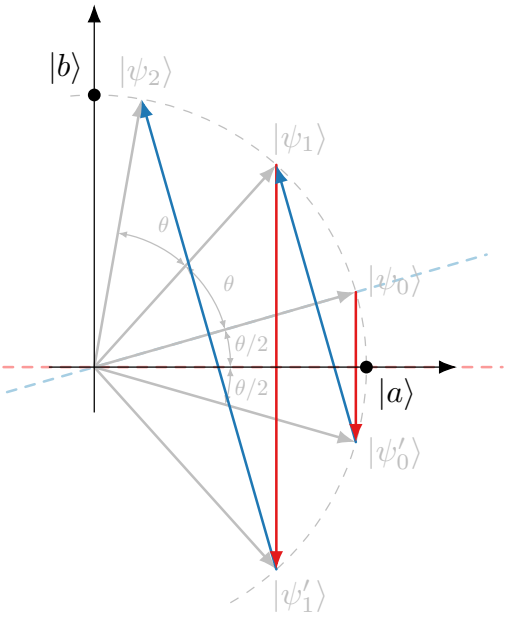


FIGURE 2.2 – Représentation géométrique des applications successives de l'oracle et du diffuseur sur l'état quantique. Les lignes pointillées de couleurs représentent les plans de réflexion pour chacun de ces opérateurs.

A.7 Oracle quantique

D'abord, l'oracle doit effectuer une réflexion par rapport au plan normal à $|b\rangle$. Ce plan est représenté par une ligne pointillée à la figure 2.2. L'état résultant de cette première transformation est donc

$$|\psi_0'\rangle = \cos(\theta/2)\,|a\rangle - \sin(\theta/2)\,|b\rangle \tag{2.5}$$

L'opération qui applique cette réflexion doit donc inverser la phase de tous les états de base qui sont présents dans $|b\rangle$. En particulier, on devra construire un circuit quantique qui, lorsqu'appliqué à un état de base $|x\rangle$, inverse sa phase si cet état est valide

$$\hat{U}_{\text{oracle}}\,|x\rangle = \begin{cases} -\,|x\rangle & x \text{ est valide } (x \in B). \\ |x\rangle & x \text{ n'est pas valide } (x \notin B)\,; \end{cases} \tag{2.6}$$

On peut résumer l'effet de l'oracle par l'équation suivante

$$\hat{U}_{\text{oracle}}\,|x\rangle = (-1)^{f(x)}\,|x\rangle\,.$$

C'est donc l'oracle qui permet de distinguer les *bons* des *mauvais* états et sépare l'espace des états en deux ensembles.

La forme que prend le circuit quantique qui applique l'oracle dépend du type de problème étudié. La création d'un tel circuit qui applique une transformation du type de l'équation 2.6 peut être assez complexe et nécessite généralement l'utilisation de qubits supplémentaires (ancillaires) afin de simplifier cette tâche. Un aspect important dans ce cas est que l'oracle doit laisser les qubits ancillaires inchangés, c'est-à-dire qu'ils doivent être dans le même état à l'entrée qu'à la sortie de l'oracle.

Remarque

On pourrait échanger les *bons* et les *mauvais* états. Cela reviendrait à appliquer une inversion de phase à l'oracle $-\hat{U}_{\text{oracle}}$. Or, on sait que l'ajout d'une phase globale n'a pas de conséquences physiques. Comme on va le voir, ce qui détermine l'effet de l'algorithme de Grover est surtout les nombres de *bons* et de *mauvais* états. Typiquement, pour un problème d'intérêt, il y a beaucoup moins de *bons* états que de *mauvais* états.

A.8 Diffuseur quantique

Ensuite, le diffuseur doit effectuer une réflexion de part et d'autre de l'état $|s\rangle$. Dans le plan de la figure 2.2, cela correspond à une réflexion par rapport à la deuxième ligne pointillée. Cette réflexion dans l'espace des états implique que l'application de l'opérateur $\hat{U}_{\text{diffuseur}}$ sur n'importe quel état orthogonal à $|s\rangle$ devrait inverser sa phase

$$\hat{U}_{\text{diffuseur}}\,|\psi\rangle = \begin{cases} |\psi\rangle & \text{si } |\psi\rangle = |s\rangle\,; \\ -\,|\psi\rangle & \text{si } \langle\psi|s\rangle = 0. \end{cases}.$$

On peut exprimer un tel opérateur de la manière suivante

$$\hat{U}_{\text{diffuseur}} = 2\,|s\rangle\langle s| - \hat{I}. \tag{2.7}$$

Exemple A.2 : Opérateur de réflexion

L'équation 2.7 a une forme qui peut paraitre étrange la première fois qu'on la rencontre. Donnons un exemple d'une opération similaire qui peut être exprimé de manière similaire : la porte $C\hat{Z}$. Cette porte a pour effet d'inverser la phase de l'état $|11\rangle$ et de laisser les autres états de base inchangés. Sa représentation dans la base

computationnelle est

$$C\hat{Z} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

Cette matrice est très similaire à la matrice identité, à l'exception du dernier élément. On peut donc l'écrire comme la somme de deux matrices

$$C\hat{Z} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}.$$

En notation dyadique cela correspond donc à l'expression recherchée

$$C\hat{Z} = \hat{I} - 2\,|11\rangle\langle 11|\,.$$

Montrons que l'opérateur de l'équation 2.7 a bien le comportement attendu en l'appliquant sur l'état $|s\rangle$ et sur un état qui y est orthogonal $|s^{\intercal}\rangle$. D'abord,

$$\hat{U}_{\text{diffuseur}}\,|s\rangle = 2\,|s\rangle\langle s|s\rangle - \hat{I}\,|s\rangle = 2\,|s\rangle - |s\rangle = |s\rangle$$

et ensuite,

$$\hat{U}_{\text{diffuseur}}\,|s^{\intercal}\rangle = 2\,|s\rangle\langle s|s^{\intercal}\rangle - \hat{I}\,|s^{\intercal}\rangle = 0 - |s^{\intercal}\rangle = -\,|s^{\intercal}\rangle$$

où on a utilisé le fait que $\langle s|s^{\intercal}\rangle = 0$ car ces états sont orthogonaux.

Le circuit quantique qui implémente le diffuseur quantique est relativement facile à identifier du moment qu'on le transforme grâce aux opérations suivantes

$$\begin{aligned} \hat{U}_{\text{diffuseur}} &= \hat{H}^{\otimes n}\Big(2\,|\mathbf{0}\rangle\langle\mathbf{0}| - \hat{I}\Big)\hat{H}^{\otimes n} \\ &= \hat{H}^{\otimes n}\hat{X}^{\otimes n}\Big(2\,|\mathbf{1}\rangle\langle\mathbf{1}| - \hat{I}\Big)\hat{X}^{\otimes n}\hat{H}^{\otimes n} \end{aligned}$$

où $|\mathbf{1}\rangle$ est l'état où tous les qubits sont dans l'état $|1\rangle$. Or l'opérateur $2\,|\mathbf{1}\rangle\langle\mathbf{1}| - \hat{I}$ inverse la phase de l'état si tous les qubits sont dans l'état $|1\rangle$ ce qui correspond, à une phase globale près, à une porte multi-contrôle-$\hat{Z}$.

A.9 Protocole et circuit quantique

Une fois qu'on a toutes les pièces en main, la mise en place de l'algorithme de Grover est relativement simple. Le circuit quantique utilisé est présenté à la figure 2.3. Celui-ci comporte deux registres : un premier qui sert à représenter les configurations possibles du problème. Le second registre contient des qubits ancillaires qui peuvent être utiles à l'implémentation de l'oracle. Ce registre n'est pas toujours nécessaire.

La première étape de l'algorithme est de préparer une superposition uniforme de toutes les configurations possibles en appliquant des portes Hadamard sur les qubits du premier registre. On a vu que cet état initial peut être écrit comme à l'équation 2.3

$$|\psi_0\rangle = \cos(\theta/2)\,|a\rangle + \sin(\theta/2)\,|b\rangle$$

La mesure des qubits à ce point permet d'obtenir un bon état avec les mêmes chances qu'un choix aléatoire, c'est-à-dire

$$p_{\psi_0}(b) = \sin^2(\theta/2) = \frac{M}{N}.$$

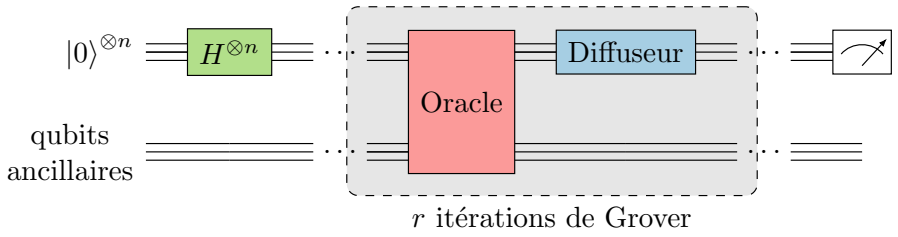


FIGURE 2.3 – Circuit quantique de l'algorithme de Grover.

On applique ensuite l'oracle pour inverser la phase des états qui répondent aux critères déterminés par le problème. On obtient alors l'état à l'équation 2.5

$$|\psi_0'\rangle = \cos(\theta/2)\,|a\rangle - \sin(\theta/2)\,|b\rangle\,.$$

Notons qu'à ce point la probabilité d'obtenir un bon état lors de la mesure des qubits est inchangée.

L'application du diffuseur effectue ensuite une réflexion pour tous les états orthogonaux à $|s\rangle$. Une simple construction géométrique à la figure 2.2 permet de constater qu'après ces étapes l'état quantique du système est

$$|\psi_1\rangle = \cos(3\theta/2)\,|a\rangle + \sin(3\theta/2)\,|b\rangle\,. \tag{2.8}$$

Avec cette transformation, la probabilité d'obtenir un bon état lors de la mesure des qubits est passée à

$$p_{\psi_1}(b) = \sin^2(3\theta/2)$$

ce qui est plus grand que les probabilités initiales. Ces deux étapes (oracle et diffuseur) constituent une itération de Grover. Si on continue ce processus, on voit qu'après r itérations de Grover, la probabilité d'obtenir un état valide passe à

$$p_{\psi_r}(b) = \sin^2(\theta(r+1/2)).$$

On maximise donc cette probabilité lorsque

$$\theta(r+1/2) = \pi/2$$

ce qui permet de déduire un nombre d'itérations optimal. Étant donné que l'angle θ dépend du nombre de solutions valides sur le nombre de possibilités, le nombre d'itérations optimal dépend également de ce ratio.

Annexe A

Démonstrations

Cette annexe contient des démonstrations qui trouvent des applications à plusieurs endroits dans les notes ou qui sont trop lourdes et auraient nui à la lecture.

1 Transformation Hadamard à plusieurs qubits

La transformation Hadamard à plusieurs qubits est impliquée dans de nombreux algorithmes. En particulier, on s'intéresse ici à la transformation des états de base à n qubits.

1.1 Transformation du premier état de base

Comme tous les qubits du premier état de base sont dans l'état $|0\rangle$, l'application d'une porte Hadamard sur chacun d'eux va produire un état $|+\rangle$. On peut expliciter cela en écrivant.

$$\hat{H}^{\otimes n} |0\rangle^{\otimes n} = \bigotimes_{q=0}^{n-1} \hat{H} |0\rangle = \bigotimes_{q=0}^{n-1} |+\rangle = |+\rangle^{\otimes n}.$$

On va montrer ici, qu'un tel état est en fait une superposition uniforme de tous les états de base à n qubits. D'abord, on écrit chacun des états $|+\rangle$ en fonction des états de base à un qubit

$$|+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \bigotimes_{q=0}^{n-1} (|0\rangle + |1\rangle).$$

On explicite ensuite le produit tensoriel pour obtenir n paires de termes

$$\frac{1}{\sqrt{2^n}} \bigotimes_{q=0}^{n-1} (|0\rangle + |1\rangle) = \frac{1}{\sqrt{2^n}} \underbrace{(|0\rangle + |1\rangle) \otimes \ldots \otimes (|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle)}_{n \text{ fois}}.$$

On peut facilement se convaincre que le produit de ces n paires de termes va produire une somme 2^n termes

$$\underbrace{(|0\rangle + |1\rangle) \otimes \ldots \otimes (|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle)}_{n \text{ fois}} = \underbrace{|0\ldots00\rangle + |0\ldots01\rangle + |0\ldots10\rangle + \ldots + |1\ldots11\rangle}_{2^n \text{ termes}} \tag{A.1}$$

qui ne sont rien d'autre que tous les états de base à n qubits. Ainsi, on peut résumer que le produit tensoriel de n qubits dans l'état $|+\rangle$ retourne une superposition de tous les états de base

$$\frac{1}{\sqrt{2^n}} \bigotimes_{q=0}^{n-1} (|0\rangle + |1\rangle) = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \tag{A.2}$$

On conclut donc que la transformation Hadamard à n qubits appliqués au premier état de base produit une superposition uniforme de tous les états de base à n qubits

$$\hat{H}^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \,. \tag{A.3}$$

1.2 Relations utiles

L'état obtenu à la section précédente peut être exprimé de plusieurs manières différentes. En les décrivant et en sachant qu'elles sont toutes équivalentes, on pourra établir des relations qui nous seront fort utiles pour la suite. La première expression est déjà donnée à l'équation A.3

$$|+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \,. \tag{A.4}$$

La seconde s'obtient en écrivant chacun des états $|+\rangle$ comme une somme sur les deux états de base du qubit q on peut écrire

$$|+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \bigotimes_{q=0}^{n-1} \sum_{x_q=0}^{1} |x_q\rangle \,. \tag{A.5}$$

Pour la troisième, on constate que le côté droit de l'équation A.1 peut également être exprimé comme le résultat de n sommes sur les deux états de base de chacun des qubits

$$\underbrace{\sum_{x_{n-1}=0}^{1} \dots \sum_{x_0=0}^{1}}_{n \text{ sommes}} |x_{n-1} \cdots x_0\rangle = \underbrace{|0 \dots 00\rangle + |0 \dots 01\rangle + |0 \dots 10\rangle + \dots + |1 \dots 11\rangle}_{2^n \text{ termes}} .$$

En écrivant l'état $|x_{n-1} \cdots x_0\rangle$ comme un produit tensoriel, on obtient la dernière expression

$$|+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \underbrace{\sum_{x_{n-1}=0}^{1} \dots \sum_{x_0=0}^{1}}_{n \text{ sommes}} \bigotimes_{q=0}^{n-1} |x_q\rangle \,. \tag{A.6}$$

En mettant en relation les équations A.4 et A.6, on déduit qu'une somme sur les 2^n état de base à n qubits est équivalente à n sommes sur les deux états de base de chacun des qubits

$$\sum_{x=0}^{2^n-1} (\dots) = \underbrace{\sum_{x_{n-1}=0}^{1} \dots \sum_{x_0=0}^{1}}_{n \text{ sommes}} (\dots). \tag{A.7}$$

En mettant en relation les équations A.5 et A.6, constate qu'on peut sortir une somme sur les deux états de base à un qubit d'un produit tensoriel en la remplaçant par les n sommes sur chacun des n qubits

$$\bigotimes_{q=0}^{n-1} \sum_{x_q=0}^{1} (\dots) = \underbrace{\sum_{x_{n-1}=0}^{1} \dots \sum_{x_0=0}^{1}}_{n \text{ sommes}} \bigotimes_{q=0}^{n-1} (\dots). \tag{A.8}$$

1.3 Transformation générale d'un état de base

On peut maintenant se pencher sur la transformation Hadamard des autres états de base à n qubits. D'abord, un état de base est un état produit

$$|x\rangle = |x_{n-1} \cdots x_0\rangle = \bigotimes_{q=0}^{n-1} |x_q\rangle$$

où chacun des bits x_q peut être dans les états $|0\rangle$ ou $|1\rangle$. L'application d'une porte Hadamard sur un qubit dans chacun de ces états va respectivement produire les états $|+\rangle$ et $|-\rangle$. On peut résumer cela avec l'équation suivante

$$\hat{H}\,|x_q\rangle = \frac{|0\rangle + (-1)^{x_q}\,|1\rangle}{\sqrt{2}}$$

où on obtient bien $|+\rangle$ si $x_q = 0$ et $|-\rangle$ si $x_q = 1$. La transformation Hadamard d'un état de base peut donc s'exprimer comme

$$\hat{H}^{\otimes n}\,|x\rangle = \frac{1}{\sqrt{2^n}} \bigotimes_{q=0}^{n-1} (|0\rangle + (-1)^{x_q}\,|1\rangle). \tag{A.9}$$

Comme à la section 1.1, le produit tensoriel de n paires de termes va produire une superposition des 2^n états de base. Cependant, dans ce cas-ci, chacun des différents états de base sera affecté d'une phase. Pour identifier la phase de chaque état de base, notons qu'on peut écrire l'état de chaque qubit après la transformation Hadamard comme étant une somme sur ses deux états de base

$$\frac{|0\rangle + (-1)^{x_q}\,|1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \sum_{z_q=0}^{1} (-1)^{x_q z_q}\,|z_q\rangle\,.$$

Le facteur de phase devient $(-1)^{x_q z_q}$ pour s'assurer que la phase -1 est appliquée uniquement quand $|z_q\rangle = |1\rangle$. L'expression A.9 devient alors

$$\hat{H}^{\otimes n}\,|x\rangle = \frac{1}{\sqrt{2^n}} \bigotimes_{q=0}^{n-1} \sum_{z_q=0}^{1} (-1)^{x_q z_q}\,|z_q\rangle\,.$$

Insistons ici sur le fait que les bits x_q décrivent l'état initial, alors que les bits z_q servent à décrire l'état transformé.

On fait ensuite usage de la relation A.8 pour sortir la somme sur z_q du produit tensoriel

$$\hat{H}^{\otimes n}\,|x\rangle = \frac{1}{\sqrt{2^n}} \underbrace{\sum_{z_{n-1}=0}^{1} \cdots \sum_{z_0=0}^{1}}_{n \text{ sommes}} \bigotimes_{q=0}^{n-1} (-1)^{x_q z_q}\,|z_q\rangle\,. \tag{A.10}$$

On voit que l'état $\hat{H}^{\otimes n}\,|x\rangle$ est une superposition d'états qui ne sont rien d'autre que les états de base accompagnés d'un facteur de phase. En effet,

$$\bigotimes_{q=0}^{n-1} (-1)^{x_q z_q}\,|z_q\rangle = (-1)^{x_0 z_0 + \ldots + x_{n-1} z_{n-1}} \bigotimes_{q=0}^{n-1} |z_q\rangle$$

où on a sorti les n facteurs de phase du produit tensoriel pour les combiner en un seul. Pour simplifier l'expression de l'état $\hat{H}^{\otimes n}|x\rangle$ notons les états de base qui décrivent l'état transformé à l'aide d'entiers z tel que

$$|z\rangle = \bigotimes_{q=0}^{n-1} |z_q\rangle \,.$$

On peut alors écrire l'état

$$\bigotimes_{q=0}^{n-1} (-1)^{x_q z_q} |z_q\rangle = (-1)^{x\cdot z} |z\rangle$$

où

$$x \cdot z = \sum_{q=0}^{n_1} x_q z_q = x_0 z_0 + \ldots + x_{n-1} z_{n-1} \tag{A.11}$$

est un produit scalaire entre les bits qui composent l'entier x et ceux qui composent l'entier z. En réécrivant l'équation A.10 à l'aide de cette expression et en utilisant l'équivalence A.7 pour remplacer les sommes sur les z_q par une somme sur z, on obtient

$$\hat{H}^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} (-1)^{x\cdot z} |z\rangle \,. \tag{A.12}$$

Ainsi, la transformation Hadamard d'un état de base retourne une superposition uniforme de tous les états de base où chacun d'eux est affecté par une phase qui dépend du produit scalaire entre les bits de l'entier z et les bits de l'entier x de l'état de base initial.