TP 3 : Résolution de problèmes avec l'algorithme de Grover

UE Informatique quantique et Recherche opérationnelle
Module 2 : Algorithmes quantiques pour la recherche opérationnelle

Décembre 2023

ENSIIE, C. Grange

Tout au long du TP, les questions à faire sur papier sont précédées du symbole †.

Compétences acquises à la fin du TP :

- Traduire plusieurs problèmes d'optimisation combinatoire sous forme d'oracle
- Savoir coder l'algorithme de Grover
- Illustrer l'importance du nombre d'itérations de l'oracle + amplificateur
- Résoudre des problèmes d'optimisation en utilisant l'algorithme de Grover

Ce TP a pour but d'étudier l'algorithme de Grover afin de résoudre plusieurs problème de décision, comme le problème de satisfaisabilité ou le problème de recherche d'un élément dans un tableau.

Exercice 1 (Porte de Toffoli à n bits)**.**
On rappelle la définition de la porte de Toffoli à 3 qubits, autrement appelée CCNOT :
Pour $x, y, z \in \{0, 1\}$,

$$CCNOT\,|x\rangle \otimes |y\rangle \otimes |z\rangle = |x\rangle \otimes |y\rangle \otimes |(x \wedge y) \oplus z\rangle\ ,$$

où $\oplus$ *est l'addition modulo 2. Sa représentation en circuit est la suivante :*

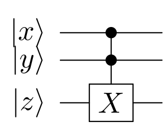

Figure 1: Porte de Toffoli

On généralise cette porte à n *qubits, appelée Porte de Toffoli à* n *qubits :*
Pour $x_1, \ldots, x_n \in \{0, 1\}$

$$|x_1\rangle \otimes \ldots |x_{n-1}\rangle \otimes |x_n\rangle \mapsto |x_1\rangle \otimes \ldots |x_{n-1}\rangle \otimes |(x_1 \wedge \ldots x_{n-2} \wedge x_{n-1}) \oplus x_n\rangle\ .$$

1. *† Écrire la porte de Toffoli à* n *qubits avec uniquement des portes de Toffoli.*

2. *L'implémenter en un circuit avec Qiskit.*

Exercice 2.

Nous allons appliquer l'algorithme de Grover a un cas très simple où l'on considère un système à 2 qubits, et où l'on cherche à trouver l'antécédent dont l'image vaut 1 *pour la fonction* f *suivante : pour* $x \in \{0,1\}^2$,

$$f(x) = \begin{cases} 1 & si\ x = 01 \\ 0 & sinon \end{cases}$$

1. *Construire le circuit à 3 qubits, notés* $|q_0\rangle, |q_1\rangle$ *et* $|q_{flag}\rangle$ *et initialisés chacun à* $|0\rangle$, *qui fait passer le qubit* $|q_{flag}\rangle$ *à* $|1\rangle$ *si et seulement si* $|q_0 q_1\rangle = |01\rangle$. *Il s'agit de l'oracle du problème de l'exercice.*

2. *Sachant que l'amplificateur de Grover est Figure 2 pour un système à 2 qubits, implémenter l'algorithme de Grover sur le simulateur IBM à 32 qubits pour ce problème, et comparer la probabilité obtenue avec la théorie.*

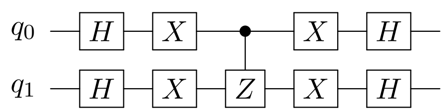

Figure 2: Amplificateur pour 2 qubits

Remarque : $Z = HXH$.

3. *Faire le même exercice avec la nouvelle fonction suivante : pour* $x \in \{0,1\}^2$,

$$f(x) = \begin{cases} 1 & si\ x = 11 \\ 0 & sinon \end{cases}$$

4. *Trouver l'expression du taux d'erreur sur chaque qubit pour un circuit de profondeur* p *et un taux d'erreur de porte* τ. *Calculer ce taux d'erreur pour* $p = 100$ *et* $\tau = 10^{-2}$, *puis* $\tau = 10^{-3}$. *Illustrer ce bruit sur les machines quantiques d'IBM.*

Exercice 3 (3-SAT)**.**

Soit la formule

$$F = (x_0 \vee x_1 \vee \neg x_2) \wedge (\neg x_0 \vee \neg x_1 \vee \neg x_2) \wedge (\neg x_0 \vee x_1 \vee x_2)\,.$$

1. *Construire l'oracle de Grover associé à* F.

2. † *Calculer, de façon classique, le nombre d'affectations des triplets* (x_0, x_1, x_2) *qui satisfont* F.

3. *Implémenter l'algorithme de Grover sur le simulateur IBM à 32 qubits, et vérifier les résultats.*

4. *Faire varier le nombre d'itérations du bloc* $\{$*oracle + amplificateur*$\}$ *et interpréter le résultat.*

5. † *Que renvoie l'algorithme de Grover si jamais aucune affectation ne satisfait* F *? Illustrer empiriquement le résultat sur la formule*

$$F' = (x_0 \vee x_1) \wedge (x_0 \vee \neg x_1) \wedge (\neg x_0 \vee x_1) \wedge (\neg x_0 \vee \neg x_1)\,.$$

Jusqu'ici, nous avons traité des problèmes de décision. A partir de maintenant, nous allons considérer des problèmes d'optimisation.

Exercice 4 (Recherche du minimum dans un tableau)**.**
Partie 1 : Encodage d'un tableau.
Dans cette partie, nous allons encoder un tableau T de taille n avec deux types de registres : le premier R_{ind} encodant les indices, le second R_{val} encodant les valeurs du tableau.

1. *† Combien de qubits faut-il pour chacun des registres, R_{ind} et R_{val} ?*

Puisqu'un tableau est une relation entre un indice et une valeur, nous allons représenter T en intriquant le registre codant un indice avec le registre codant sa valeur; et superposer le registre d'indices pour décrire entièrement le tableau.

2. *† Soit le tableau $T = [2, 3, 0, 1]$. Quelles sont les tailles de R_{ind} et R_{val} ?*

3. *Nous allons encoder la première valeur du tableau. Pour cela, écrire la proposition suivante : "Si le registre R_{ind} se trouve dans l'état qui encode l'indice 0, alors le registre R_{val} se trouve dans l'état qui encode la valeur $T[0] = 2$."*

4. *En déduire l'encodage de T, puis l'implémenter.*

Partie 2 : Recherche d'un élément dans un tableau.
Nous allons utiliser l'algorithme de Grover pour rechercher un élément e dans un tableau T. L'oracle de l'algorithme va à la fois encoder le tableau et ensuite encoder l'élément à rechercher.
Soit le même tableau $T = [2, 3, 0, 1]$, et l'élément $e = 1$.

5. *La première partie de l'oracle encode le tableau T comme vu précédemment. Quelle est la deuxième partie de l'oracle ? L'encoder.*

6. *Implémenter l'algorithme de Grover pour résoudre le problème de recherche dans T de l'élément $e = 1$ sur le simulateur IBM à 32 qubits. Faire de même pour $e = 3$.*

Partie 3 : Recherche du minimum dans un tableau.
Pour finir, nous allons écrire un algorithme dérivé de Grover qui, donné un tableau d'entiers T, trouve la valeur minimum et l'indice associé. Nous supposerons que la valeur minimum est atteinte une seule fois. Nous supposerons aussi avoir un majorant M des valeurs du tableau.
On appelle $\mathcal{O}_k$ l'oracle qui étiquette à 1 tous les états de R_{ind} tels que la valeur associée dans R_{val} est strictement inférieure à l'entier k.

7. *† Écrire le pseudo-code de l'algorithme qui renvoie le minimum à l'aide de la famille d'oracles $(\mathcal{O}_k)_{k \in [M]}$.*

8. *Implémenter l'oracle $\mathcal{O}_k$ comme fonction de k, pour $M = 3$. Vous vous aiderez du circuit Figure 3 qui applique la porte X sur le qubit q_4 si et seulement si*
$$q_0 + 2q_1 < q_2 + 2q_3 .$$
Notez que les qubits $q_5 \dots q_8$ sont des qubits auxiliaires.

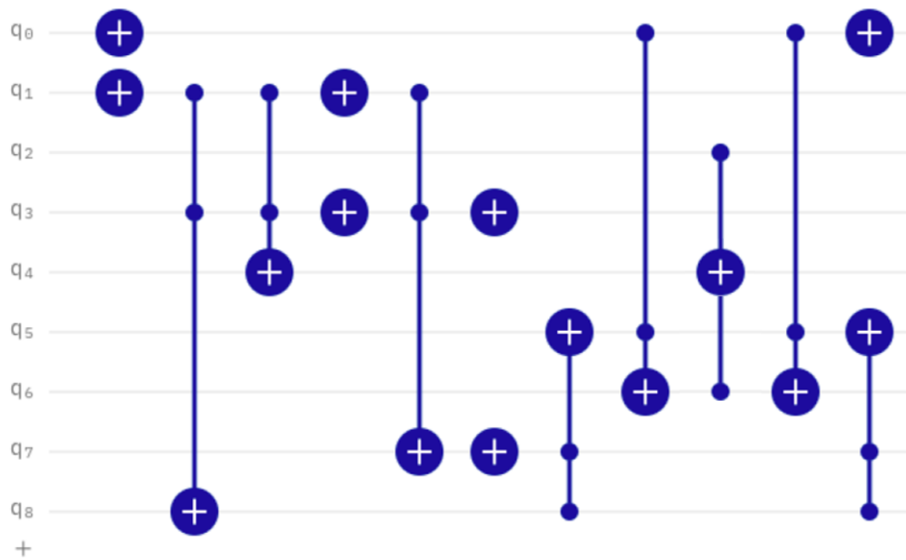

Figure 3: Circuit comparateur inférieur strict.

9. *Implémenter l'algorithme de recherche du minimum pour le tableau $T = [2, 3, 0, 1]$ entièrement.*

Exercice 5 (MAX-3-SAT)**.**
Le problème MAX-3-SAT est le problème d'optimisation associé au problème de décision 3-SAT. Il s'agit de trouver une affectation des variables qui satisfait le maximum de clauses, où chaque clause est la disjonction de 3 littéraux.

En se basant sur la technique employée lors de l'exercice précédent, construire un circuit qui résout MAX-3-SAT.

Exercice 6 (Vertex Cover)**.** *Nous considérons le problème de couverture de graphe par les sommets (Vertex Cover).*

Soit $G = (V, E)$ un graphe non-orienté. Le problème consiste à trouver une couverture minimale, c'est-à-dire un ensemble de sommets S dans V de cardinalité minimale qui couvre toutes les arêtes dans E. On rappelle qu'une arête $uv \in E$ est couverte ssi $u \in S$ ou $v \in S$.

1. *† Écrire la formulation mathématique du Vertex Cover.*

2. *† L'oracle se divise en 2 parties : l'une qui étiquette uniquement les solutions réalisables, l'autre qui sélectionne la meilleure. Trouver à quels problèmes se ramène chacune des parties, et encoder l'oracle.*

3. *Implémenter l'algorithme de Grover sur le graphe C_4, représenté Figure 4.*

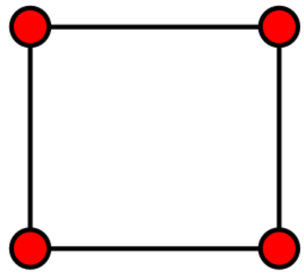

Figure 4: Graphe C_4.

4. *Comparer les résultats lorsque l'on exécute le code sur le simulateur, puis sur une machine quantique.*

Exercice 7 (Set Cover)**.** *Ecrire et implémenter la résolution du problème de couverture par ensemble pour l'instance Figure 5.*

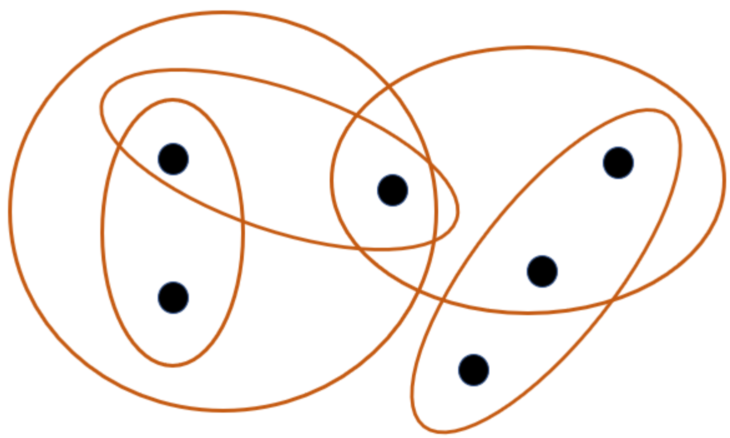

Figure 5: Instance de Set Cover