



HAL
open science

Etude de l'intrication par les polynômes de Mermin : application aux algorithmes quantiques.

Grâce Dorcas Akpéné Amouzou

► To cite this version:

Grâce Dorcas Akpéné Amouzou. Etude de l'intrication par les polynômes de Mermin : application aux algorithmes quantiques.. Autre [cs.OH]. Université Bourgogne Franche-Comté; Université de Lomé (Togo). Faculté des sciences, 2024. Français. NNT : 2024UBFCK063 . tel-04914458

HAL Id: tel-04914458

<https://theses.hal.science/tel-04914458v1>

Submitted on 27 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**THESE DE DOCTORAT DE L'ETABLISSEMENT UNIVERSITE BOURGOGNE FRANCHE-COMTE
PREPAREE A L'UNIVERSITE DE TECHNOLOGIE DE BELFORT-MONTBELIARD**

Ecole doctorale n°553

CARNOT PASTEUR

Doctorat en Informatique

Par

AMOUZOU Grâce Dorcas Akpéné

Etude de l'intrication par les polynômes de Mermin : Application aux algorithmes quantiques

Thèse présentée et soutenue à Lomé, le 09 décembre 2024

Composition du Jury :

M. Abdeljalil ABBAS-TURKI	Professeur, Université de Technologie de Belfort Montbéliard	Président
M. Jean-Gabriel LUQUE	Professeur, Université de Rouen	Rapporteur
M. Amah Séna d'ALMEIDA	Professeur, Université de Lomé	Rapporteur
M. Hamza JAFFALI	Docteur, Université de Technologie de Belfort Montbéliard	Examineur
M. Frédéric HOLWECK	Maître de Conférences HDR, UTBM	Directeur de thèse
M. Kossi ATCHONOU GLO	Maître de Conférences CAMES, Université de Lomé	Codirecteur de thèse

*Je dédie ce manuscrit
À ma très chère famille.*

La vie n'est pas facile pour aucun de nous. Mais quel que soit notre sort, nous devons poursuivre notre chemin. Il y a quelque chose que nous devons faire.

M. Curie [23]

Remerciements

Ma gratitude s'adresse tout d'abord à ma famille : Pst Robert, Thérèse et Priscillia AMOUZOU, ainsi qu'à ma chère tante Jacqueline ABIASSI, pour leur soutien indéfectible et inconditionnel. Je tiens également à remercier chaleureusement mes directeurs de thèse, Dr Kossi ATCHONOU GLO (MC) et Dr Frédéric HOLWECK (MC-HDR), pour leurs précieux conseils, leurs encouragements et leur appui constant tout au long de ce projet. Leur expertise et leurs orientations éclairées ont été d'une aide inestimable.

Je remercie également mes amis proches, Christ, Israël A., Israël T., Paul, Francis, Patrice, Maxwell, Mario, Cindy, Minkitam, Blaise, Joël, David, Marcel, Martel, Pierre, Wing et Faïza, pour leurs encouragements et leur présence bienveillante tout au long de cette aventure académique.

Mes sincères remerciements vont au Prof. Yaogan MENSAH, directeur du Laboratoire d'Analyse, de Modélisation Mathématiques et Applications (LAMMA), à M. S. GUERIN, directeur du Laboratoire Interdisciplinaire Carnot de Bourgogne, au Prof. B. BAKOMA, directeur de la Formation Doctorale, ainsi qu'à M. H. JAUSLIN, directeur de l'ED Carnot-Pasteur, pour m'avoir accueillie dans leurs structures respectives durant ces années de thèse.

Je suis profondément reconnaissante envers les Professeurs Abdeljalil ABBAS-TURKI, Jean-Gabriel LUQUE, Amah Séna d'ALMEIDA, ainsi que le Dr Hamza JAFFALI, pour avoir accepté de faire partie de mon jury de soutenance.

Je tiens à exprimer ma gratitude envers Yannick, Dr. Tchamye T.-E. BOROZE (MC), à mon aîné Dr Henri de BOUTRAY, mon collègue de laboratoire C. KELLEHER, ainsi qu'à mes collègues de la DRSI/UL et EPL/UL, notamment Yaovi HOETOWOU, Komi Romain ABOLO-SEWOVI, M'bere KODJO, Dr Séna APEKE, Folly AKAKPO, Olivier AHIKPOR et Kinvi FOLLY-YISSOUH. Leurs discussions stimulantes et échanges fructueux ont enrichi de nombreux aspects de mes recherches et de ma vie. Je remercie mes étudiants de l'EPL/UL, pour qui j'ai une affection particulière. Leur dynamisme et leur énergie ont été une source de réconfort, notamment dans les moments difficiles.

Un merci particulier à Laure CASTES et ses trois merveilleux enfants Clarisse, Zoé et Anatole, Israël AGBOTSE et son épouse, Giovanni AYANOU, ainsi qu'à tous ceux qui m'ont accueillie à bras ouverts et ont rendu mes séjours en France agréables.

Je tiens également à exprimer ma reconnaissance à l'ambassade de France au Togo, et plus précisément au personnel du Service de Coopération et d'Action Culturelle (SCAAC), pour la bourse dont j'ai bénéficié, grâce à Catherine au projet PEA/IMPACT, pour le financement de mon séjour le plus récent.

Cette thèse n'aurait pas pu voir le jour sans le soutien précieux de MM. Joachim AMOUZOU, Adrien FEDY et Ben AGBAN qui ont joué un rôle-clé dans l'élaboration de mon projet d'études. Enfin, j'exprime ma gratitude à toutes les personnes dont l'apport, direct ou indirect, a contribué à la réussite de cette thèse.

À vous tous, j'adresse mes plus sincères remerciements.

Table des matières

Liste des Figures	viii
Liste des Tableaux	ix
Introduction générale	1
1 NOTIONS DE BASE DU CALCUL QUANTIQUE	4
1.1 Contexte	5
1.2 B.A.-BA de l'information quantique	5
1.3 Manipulation des qubits	6
1.4 Circuits Quantiques	9
1.5 Intrication quantique	12
1.6 Conclusion	14
2 ALGORITHMES QUANTIQUES	15
2.1 Contexte	16
2.2 Algorithmes quantiques	16
2.2.1 Algorithme de Deutsch-Jozsa	16
2.2.2 Algorithme de Grover	17
2.2.3 Algorithme d'estimation de phase	20
2.2.4 Algorithme de comptage quantique	21
2.3 Conclusion	23
3 OUTILS DE MESURE DE L'INTRICATION	24
3.1 Contexte	25
3.2 Algorithmes de recherche d'optimum	25
3.3 Mesure géométrique de l'intrication (MGI)	25
3.4 Polynômes de Mermin et Invariant μ	26
3.4.1 Polynômes de Mermin et Inégalités de Mermin	26
3.4.2 Invariant μ	28
3.4.3 Invariant $\tilde{\mu}$	29
3.5 Matrices de coefficients et E_{moy}	29
3.5.1 Définitions préalables	29
3.5.2 Invariant d'intrication moyen E_{moy}	30
3.6 Multirang	31
3.7 Implémentations sur un ordinateur quantique	32
3.7.1 Implémentation de MGI	32
3.7.2 Implémentation de μ	33
3.8 Conclusion	35

4	ETATS HYPERGRAPHIQUES	36
4.1	Contexte	37
4.2	Etats de graphes et d'hypergraphes	37
4.3	Intrication dans les états hypergraphiques	39
4.4	Implémentation sur un ordinateur quantique	42
4.5	Conclusion	44
5	INTRICATION DANS LES ALGORITHMES AEPQ ET ACQ	45
5.1	Contexte	46
5.2	Intrication de $ \psi_{\text{AEPQ}}(\phi)\rangle$	46
5.3	Intrication de $ \psi^{\text{ACQ}}(\phi)\rangle$	52
5.4	Conclusion	54
6	AGENTS RÉACTIFS	55
6.1	Contexte	56
6.2	Agents réactifs et Circuits quantiques	56
6.3	Implémentations et Généralisation	60
6.4	Autres résultats	61
6.5	Conclusion	66
	Conclusion générale	68
	Annexes	71
.1	Codes numériques en Python	71
.1.1	Algorithme de marche aléatoire	71
	Algorithme de marche aléatoire	71
.1.2	Algorithme d'aplatissement des tenseurs	72
	Algorithme d'aplatissement des tenseurs	72
.2	Codes sur machine quantique en Qiskit	73
.2.1	Algorithme de calcul des invariants μ et $\tilde{\mu}$ sur IBM-Q	73
	Algorithme de calcul des invariants μ et $\tilde{\mu}$ sur IBM-Q	73
.2.2	Algorithme du calcul de la mesure géométrique de l'intrication	73
	Algorithme du calcul de la mesure géométrique de l'intrication	73
.2.3	Hypergraphes sur IBMQ	74
	Hypergraphes sur IBMQ	74

Résumé

Cette thèse explore la mesure de l'intrication dans certains états hypergraphiques, dans certains algorithmes quantiques tels que les algorithmes quantiques d'estimation de phase et de comptage, ainsi que dans les circuits d'agents réactifs, à l'aide de la mesure géométrique de l'intrication, d'outils issus des polynômes de Mermin et des matrices de coefficients. L'intrication est un concept présent en physique quantique qui n'a pas d'équivalent connu à ce jour en physique classique. Le cœur de notre recherche repose sur la mise en place de dispositifs de détection et de mesure de l'intrication afin d'étudier des états quantiques du point de vue de l'intrication. Dans cette optique, des calculs sont d'abord effectués numériquement puis sur simulateur et ordinateur quantiques. Effectivement, trois des outils exploités sont implémentables sur machine quantique, ce qui permet de comparer les résultats théoriques et "réels".

Mots-clés : Intrication, mesure géométrique de l'intrication, matrices de coefficients, polynômes de Mermin, états d'hypergraphes, agents réactifs.

Abstract

This thesis explores the measurement of entanglement in certain hypergraph states, in certain quantum algorithms like the Quantum Phase estimation and Counting algorithms as well as in reactive agent circuits, using the geometric measurement of entanglement, tools from Mermin polynomials and coefficient matrices. Entanglement is a concept present in quantum physics that has no known equivalent to date in classical physics. The core of our research is based on the implementation of entanglement detection and measurement devices in order to study quantum states from the point of view of entanglement. With this in mind, calculations are first carried out numerically and then on a quantum simulator and computer. Indeed, three of the tools used can be implemented on a quantum machine, which allows us to compare theoretical and "real" results.

Key words : Entanglement, geometric measure of entanglement, coefficient matrices, Mermin polynomials, hypergraph states, reactive agents.

Table des figures

1.2.1 Représentation vectorielle d'un qubit	5
1.4.1 Circuit représentant un oracle	10
1.4.2 Interface de l'outil IBM Quantum Composer	11
2.2.1 Circuit illustrant l'algorithme de Deutsch-Jozsa	17
2.2.2 Circuit quantique de l'algorithme de Grover	18
2.2.3 Interprétation géométrique de l'algorithme de Grover	19
2.2.4 Évolution des amplitudes au cours de l'exécution de l'algorithme de Grover	19
2.2.5 Partie pré-TFQ de l'algorithme d'estimation de phase quantique (AEPQ) .	20
2.2.6 Schéma complet de l'algorithme d'estimation de phase (AEPQ)	21
2.2.7 Circuit qui illustre l'algorithme de comptage quantique (ACQ)	23
3.7.1 Représentation d'un circuit de recherche aléatoire et d'un optimiseur pour implémentation sur un ordinateur quantique	33
3.7.2 Circuit de l'implémentation du calcul de la MGI sur un ordinateur quantique	34
3.7.3 Circuits comportant le placement des portes U	35
4.2.1 Quelques exemples de graphes quantiques à 2, 3, 4 et 5 qubits	38
4.2.2 Circuits équivalents aux graphes de la Figure 4.2.1.	38
4.2.3 Quelques états hypergraphiques quantiques	39
4.2.4 Circuits équivalents aux hypergraphes de la Figure 4.2.3.	39
4.3.1 Classification LU des hypergraphes à quatre qubits connectés	40
4.4.1 Circuit générant l'hypergraphe 17	43
4.4.2 Version transpilée du circuit de l'hypergraphe 17	43
4.4.3 Exemple d'état hypergraphique et le circuit représentant son évaluation par un monôme du polynôme de Mermin	44
5.2.1 Évaluations numériques des états issus de AEPQ par les trois invariants . .	47
5.2.2 Évaluations numériques des trois invariants pour les états quantiques $ \psi^{\text{AEPQ}}(\phi)\rangle$ en fonction de ϕ lorsque le deuxième registre est constitué de deux qubits pour $n = 3$	52
5.3.1 Évaluations numériques des états issus de ACQ par les trois invariants . .	53
6.2.1 Circuit de l'architecture de subsomption agissant sur un circuit à 9 qubits .	58
6.2.2 Première partie des circuits d'agents réactifs	59
6.2.3 Exemple d'un circuit complet des agents réactifs sur l'outil IBM Composer	59
6.3.1 Circuit d'un agent réactif avec en entrée une porte H sur les perceptions .	60
6.3.2 Résultats de l'évaluation de l'état en sortie d'un circuit d'agent réactif par l'invariant E_{moy} avec une portes U	61
6.4.1 Courbes de l'évaluation des résultats de l'évaluation des 32 combinaisons de règles	66

Liste des tableaux

3.1	Tableau des coefficients numériques qui maximisent $\mu(GHZ\rangle_3)$	35
4.1	Tableau regroupant les valeurs calculées des invariants par familles d'états hypergraphiques quantiques	41
4.2	Comparaison du temps de calcul des invariants par algorithme de recherche	42
4.3	Estimation numérique de l'invariant μ pour des états hypergraphiques k -uniformes	42
6.1	Tableau des 44 combinaisons de règles retenues pour $n = 2$ perceptions. . .	63
6.2	Tableau regroupant les valeurs calculées des invariants pour toutes les combinaisons de règles énumérées.	65

Introduction générale

L'informatique quantique a émergé dans les années 80, à partir des idées de la physique quantique et de la théorie de l'information quantique. Son avènement a modifié notre compréhension de la théorie de l'information développée par C. Shannon à la fin des années 40. Elle se base sur les principes de la mécanique quantique pour traiter l'information d'une manière nouvelle. L'informatique quantique s'appuie sur deux piliers fondamentaux propres à la physique quantique qui sont : la superposition et l'intrication. Le premier suggère que les états quantiques peuvent exister dans une superposition de plusieurs bits quantiques (appelés qubits), permettant ainsi de représenter simultanément plusieurs valeurs et d'effectuer des calculs parallèles. L'intrication, quant à elle, fait référence à des corrélations non classiques entre des qubits connectés de sorte que la mesure de l'un affecte instantanément l'état des autres, même à grande distance. L'intrication quantique suscite de l'intérêt [1, 24, 41, 42, 43, 45, 46, 47, 50, 83], car elle semble jouer un rôle central notamment dans les algorithmes quantiques (tels que l'algorithme de Grover [39] et de Shor [77]) et dans la cryptographie quantique.

Ce concept a été introduit par A. Einstein, B. Podolsky et N. Rosen en 1935 ([29]) qui ont proposé une expérience de pensée pour critiquer le principe de non localité de la mécanique quantique. D'après [29], les résultats contre-intuitifs prévus par la mécanique quantique, lorsqu'on mesure des paires de particules intriquées, montrent indirectement l'incomplétude de la théorie. A. Einstein, B. Podolsky et N. Rosen en déduisent l'existence de variables cachées permettant de maintenir le principe de réalité locale. Néanmoins, les inégalités formulées par J. Bell. en 1960 [10] prouvent que les deux théories, la mécanique quantique d'une part, ou une théorie à variables cachées locales, ne prédisent pas les mêmes résultats. Ces inégalités ont par la suite été testées par A. Aspect et al. [6] et généralisées par D. Mermin [62]. Dans le sillage de ces débats, l'émergence de l'informatique quantique a ouvert de nouvelles voies pour exploiter les propriétés quantiques dans le domaine du calcul et du traitement de l'information. R. Feynman [31], dans les années 1980, a soulevé la question des limites des ordinateurs classiques dans la simulation de systèmes quantiques, encourageant ainsi le développement d'algorithmes et de protocoles exploitant les avantages quantiques. Les travaux de pionniers tels que D. Deutsch [26], de L. Grover [39] et P. Shor [77] ont approfondi cette vision, démontrant les performances supérieures de certains algorithmes quantiques par rapport à leurs équivalents classiques.

Grâce aux avancées continues en informatique quantique, ces algorithmes peuvent être aujourd'hui implémentés et exécutés sur des simulateurs et machines quantiques à l'aide notamment de langages de programmation. QCL [67] est l'un des premiers langages de programmation sur machine quantique à avoir été développé. Plusieurs autres langages de programmation quantique ont ensuite suivi. On peut citer par exemple Cirq, Q#, Pyquil, etc. L'un d'entre eux est Qiskit [51], un framework basé sur le langage Python. Il s'agit d'un ensemble d'outils et de bibliothèques utiles pour la programmation quantique,

développé par IBM [48] qui permet de concevoir, de simuler et d'exécuter des circuits quantiques sur des ordinateurs réels via le cloud ou des simulateurs quantiques. Ces progrès ont conduit à des réflexions sur les mécanismes fondamentaux sous-jacents, notamment la mesure de l'intrication.

Dans cette optique, les travaux de cette thèse se focalisent d'abord sur la définition d'outils de mesure de l'intrication, puis sur l'étude de l'intrication dans les états hypergraphiques, dans les algorithmes quantiques et dans les circuits d'agents réactifs. Ainsi, cinq outils de détection et/ou de mesure de l'intrication sont présentés et exploités dans ce manuscrit. Il s'agit de la mesure géométrique de l'intrication (*MGI*), l'invariant μ , sa variante $\tilde{\mu}$ calculés à partir des polynômes de Mermin, l'invariant d'intrication moyenne E_{moy} et le multirang tous les deux obtenus à partir des matrices de coefficients. La *MGI* a déjà été utilisée pour étudier les variations de l'intrication au cours de l'exécution de l'algorithme de Grover [71, 87] et de celui de Shor et de la Transformée de Fourier Quantique (TFQ) [75]. Les polynômes de Mermin ont également été utilisés pour évaluer l'intrication dans l'algorithme de Grover [9] et l'invariant μ a été utilisé pour détecter l'intrication dans les algorithmes de Grover et de TFQ [24].

Dans un premier temps, la capacité de ces outils à séparer les classes d'intrication des familles d'hypergraphes à 4 qubits connectés [42]. Dans un deuxième temps, nous étudions l'intrication générée par l'algorithme d'estimation de phase quantique (AEPQ) et l'algorithme de comptage quantique (ACQ) avant l'application de l'inverse de la Transformée de Fourier Quantique (TFQ) en suivant les travaux de [80]. Nous prouvons que ces algorithmes génèrent de l'intrication pour certaines valeurs de la phase de l'opérateur utilisé et ce lorsque les qubits du second registre sont en superposition. Ces travaux, présentés dans les chapitres 4 et 5 reposent sur deux articles que nous avons publiés dans des revues à comité de lecture (ACL) pendant cette thèse [3, 4]. Dans un dernier temps, nous présentons un travail exploratoire où nous utilisons à nouveau les invariants pour évaluer l'intrication issue de circuits représentant les perceptions et actions des agents réactifs suivant des règles données en partant d'un cas de la littérature [54]. Nous introduisons de l'intrication dans ces circuits en proposant de superposer les perceptions et nous tentons ensuite de généraliser notre approche.

De plus, à l'aide de Qiskit, nous avons testé la plupart des résultats numériques sur simulateur puis sur ordinateur quantiques. En effet, les états hypergraphiques à 4 qubits ont été implémentés et calculés avec l'invariant μ ainsi que les algorithmes d'estimation de phase et de comptage quantiques avec le même invariant en plus de la *MGI*.

Organisation du manuscrit

Notre manuscrit s'articule autour de six chapitres.

1. Chapitre 1 - Notions de base du calcul quantique : ce premier chapitre s'attachera à établir quelques fondements de l'informatique quantique qui sont notamment le qubit, le qudit, le circuit quantique et l'intrication quantique, jetant ainsi les bases nécessaires à la compréhension des développements ultérieurs.
2. Chapitre 2 - Algorithmes quantiques : le deuxième chapitre présentera différents algorithmes quantiques, tels que l'algorithme de Grover, l'algorithme d'estimation de phase quantique et l'algorithme de comptage quantique, dont les deux derniers seront étudiés dans un chapitre postérieur.
3. Chapitre 3 - Outils de mesure de l'intrication : ce chapitre se concentrera sur les outils de mesure de l'intrication, particulièrement la mesure géométrique de l'in-

trication, les invariants tirés des polynômes de Mermin et ceux tirés des matrices de coefficients, expliquant les méthodes de mesure de l'intrication exploitées par la suite.

4. Chapitre 4 - États d'hypergraphes : l'étude de l'intrication des états hypergraphiques sera l'objectif principal de ce chapitre, plus particulièrement, nous traiterons de la classification des états hypergraphiques à quatre qubits [42]. Nous montrons, entre autres, que les invariants μ , $\tilde{\mu}$ et MGI fonctionnent bien pour différencier ces familles.
5. Chapitre 5 - Algorithmes d'estimation de phase et de comptage quantiques : ce chapitre analysera l'intrication des algorithmes quantiques d'estimation de phase et de comptage avant l'application de l'inverse de la TFQ. Ici, nous avons prouvé que ces algorithmes génèrent de l'intrication pour certaines valeurs de la phase de la porte utilisée dans ces algorithmes, lorsque les qubits du second registre sont en superposition.
6. Chapitre 6 - Agents réactifs : enfin, dans le dernier chapitre nous explorons la détection de l'intrication dans les circuits générés à partir de divers perceptions et actions des agents réactifs. A partir d'un exemple issu de la littérature nous étudions l'intrication générée par un circuit quantique qui encode les règles de perceptions-action d'un robot. Pour générer de l'intrication, nous faisons l'hypothèse que les perceptions sont en superposition. Nous esquissons un travail de généralisation en proposant de construire tous les circuits à deux perceptions et en évaluant nos invariants sur les circuits ainsi construits.

Pour finir, une annexe fournit les pseudo-codes des algorithmes utilisés en Python pour les calculs numériques ainsi qu'en Qiskit pour les calculs sur les machines quantiques.

Chapitre 1

NOTIONS DE BASE DU CALCUL QUANTIQUE

Contenu du Chapitre

1.1	Contexte	5
1.2	B.A.-BA de l'information quantique	5
1.3	Manipulation des qubits	6
1.4	Circuits Quantiques	9
1.5	Intrication quantique	12
1.6	Conclusion	14

1.1 Contexte

Nous présentons le b.a.-ba de l'information quantique dont des notions telles que la superposition, la mesure, les circuits et l'intrication quantiques. La compréhension de ces concepts est fondamentale pour appréhender la suite de nos travaux.

1.2 B.A.-BA de l'information quantique

BIT QUANTIQUE La brique de base de l'information classique est le **bit** qui peut prendre deux valeurs : 0 ou 1. Le bit quantique, baptisé **qubit** est l'unité de base en information quantique.

Définition 1.2.1 (Qubit [5, 25]). *Un qubit est un état quantique obtenu par la superposition de deux états quantiques de base :*

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \quad (1.2.1)$$

où $\alpha, \beta \in \mathbb{C}$ sont appelés des **amplitudes de probabilité** avec la condition $|\alpha|^2 + |\beta|^2 = 1$.

Puisque $\alpha, \beta \in \mathbb{C}$ et que la condition $|\alpha|^2 + |\beta|^2 = 1$ est imposée, on peut écrire :

$$\begin{aligned} \alpha &= \cos(\theta/2) \\ \beta &= e^{i\phi} \sin(\theta/2) \end{aligned}$$

avec $\theta, \phi \in \mathbb{R}$, $\theta \in [0, \pi]$ et $\phi \in [0, 2\pi]$. Ces deux paramètres permettent de représenter l'état sur une sphère appelée la sphère de Bloch. L'espace vectoriel pour un qubit $\mathcal{H} = \mathbb{C}^2$, est appelé l'espace de Hilbert. Les états notés $|0\rangle$ et $|1\rangle$ (lire "ket" 0 et "ket" 1) sont des vecteurs qui forment une base orthonormée de l'espace \mathbb{C}^2 et s'écrivent :

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ et } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (1.2.2)$$

Cette base est appelée base standard des vecteurs de \mathbb{C}^2 . L'écriture vectorielle du qubit $|\psi\rangle$ dans cette base est $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$. La norme du vecteur $|\psi\rangle$ est 1.

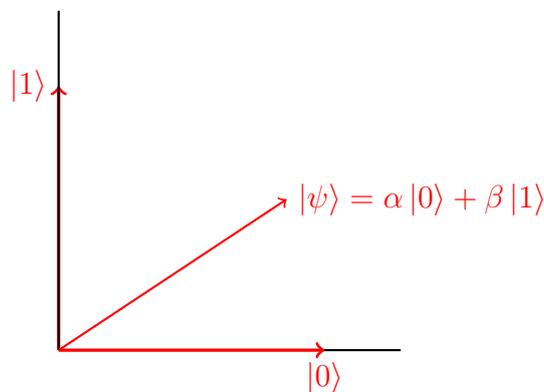


FIGURE 1.2.1 – Représentation vectorielle du qubit $|\psi\rangle$ en fonction des états de base $|0\rangle$ et $|1\rangle$.

Exemple 1.2.1. *Quelques exemples de qubits :*

$$\begin{aligned} & \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \\ & \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ & \frac{1}{\sqrt{3}}|0\rangle - i\frac{\sqrt{2}}{\sqrt{3}}|1\rangle \end{aligned}$$

Remarque 1.2.1. *L'espace de Hilbert \mathbb{C}^2 est muni du produit hermitien $\langle u|v\rangle = u^\dagger v$ où $u^\dagger = \langle u|$, "bra u ", est le vecteur ligne conjugué du vecteur colonne $|u\rangle$. Cette notation dite de Dirac [5, 35] représente les vecteurs par la notation "ket" et les formes linéaires avec la notation "bra". Le produit hermitien s'obtient alors par la concaténation "bra-ket". Cette notation se généralise lorsqu'on considère des produits tensoriels d'espaces de Hilbert (voir Section 1.3)*

QUDIT Une autre unité de base de l'information quantique est le **qudit**. Un qudit est une unité de calcul alternative de l'information quantique qui possède plus de deux états de base, contrairement au qubit.

Définition 1.2.2 (Qudit [86]). *Un qudit est basé sur un système quantique à plusieurs niveaux d tel que :*

$$|\psi_d\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle + \dots + \alpha_{d-1}|d-1\rangle, \quad (1.2.3)$$

où $d \in \mathbb{N}^*$ avec $d > 2$, $\alpha_i \in \mathbb{C}$ et $\sum_{i=0}^{d-1} |\alpha_i|^2 = 1$.

Par rapport aux qubits, les qudits offrent des avantages tels qu'une capacité de stockage d'information plus élevée et une robustesse améliorée au bruit [65]. L'utilisation de qudits permet aussi d'améliorer la correction d'erreur [20].

Exemple 1.2.2. *Quelques exemples de qudits :*

$$\begin{aligned} & \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle) \in \mathbb{C}^3 \\ & \frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle) \in \mathbb{C}^4 \end{aligned}$$

1.3 Manipulation des qubits

La composition des états quantiques s'obtient à l'aide du produit tensoriel.

Définition 1.3.1 (Produit tensoriel [5, 66]). *Soient $|\psi_1\rangle$ l'état d'un registre à n qubits et $|\psi_2\rangle$ celui d'un registre de m qubits, l'état du registre composé de $n + m$ qubits est :*

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle,$$

$$\forall x \in \{0, 1\}^n \text{ et } \forall y \in \{0, 1\}^m, |x\rangle \otimes |y\rangle = |xy\rangle.$$

Exemple 1.3.1. *Les différents produits tensoriels des états de base de \mathbb{C}^2 définis plus tôt dans la Section 1.2.2 produisent les résultats suivants :*

$$|0\rangle \otimes |0\rangle = |00\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Les états obtenus dans l'exemple (1.3.1) forment une base de l'espace $\mathbb{C}^2 \otimes \mathbb{C}^2$. Dans celle-ci, on peut définir l'état :

$$|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle,$$

où $\alpha_{ij} \in \mathbb{C}$ et $\sum |\alpha_{ij}|^2 = 1$.

Définition 1.3.2 (État quantique à n qubits [43]). *De manière générale, un système à n qubits est un état quantique défini par :*

$$|\psi\rangle = \alpha_{00\dots 0} |00\dots 0\rangle + \alpha_{00\dots 01} |00\dots 01\rangle + \dots + \alpha_{11\dots 11} |11\dots 11\rangle, \quad (1.3.1)$$

où $|00\dots 0\rangle, |00\dots 1\rangle, \dots, |11\dots 1\rangle$ sont les états formant la base standard de $\mathcal{H} = \mathbb{C}^2 \otimes \dots \otimes \mathbb{C}^2$ obtenus par produit tensoriel des vecteurs de base de chaque espace de Hilbert \mathbb{C}^2 . L'état $|\psi\rangle$ possède 2^n coefficients normalisés selon la condition $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$. Autrement dit, il est un **tenseur** normalisé de \mathcal{H} .

Exemple 1.3.2. *Exemples de deux états à 2 qubits et d'un état à 3 qubits.*

$$\begin{aligned} & \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ & \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle \\ & \frac{1}{2\sqrt{2}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle) \end{aligned}$$

Certains états à plusieurs qubits ne peuvent s'écrire comme le produit tensoriel des états de base. Par exemple, pour l'état $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, il n'existe pas d'états $|\psi_i\rangle$ tels que $|\psi_1\rangle \otimes |\psi_2\rangle = |\psi\rangle$. On dit qu'il est *intriqué*. Nous détaillons cette notion particulière dans la Section 1.5.

PRINCIPE DE MESURE La mesure quantique est un processus qui permet d'extraire des informations sur l'état d'un système quantique.

Définition 1.3.3 (Mesure [5, 66]). *Mesurer le qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ dans la base standard donne 0 avec probabilité $|\alpha|^2$ ou 1 avec probabilité $|\beta|^2$. La mesure projette le qubit dans l'état observé, autrement dit, si le résultat est 0, alors le qubit est dans l'état $|0\rangle$ et si le résultat est 1, alors le qubit est dans l'état $|1\rangle$. Cette action fait perdre au qubit sa superposition quantique.*

Exemple 1.3.3. *Soit l'état suivant :*

$$\frac{1}{\sqrt{3}}|0\rangle - i\frac{\sqrt{2}}{\sqrt{3}}|1\rangle.$$

En mesurant ce qubit, on obtient $|0\rangle$ dans $\left|\frac{1}{\sqrt{3}}\right|^2 = \frac{1}{3}$ des cas et $|1\rangle$ dans $\left|-i\frac{\sqrt{2}}{\sqrt{3}}\right|^2 = \frac{2}{3}$ des cas.

Remarque 1.3.1. *Un qubit peut être mesuré dans n'importe quelle base orthonormée de \mathbb{C}^2 . Le résultat de la mesure sera la projection du qubit sur un des deux états de base choisi avec des probabilités données par le carré du module des amplitudes du qubit dans cette base.*

ÉVOLUTION UNITAIRE En l'absence de mesure, le système quantique est dit "isolé" et évolue de façon unitaire : cette évolution est linéaire et préserve la condition de normalisation.

Exemple 1.3.4. *Les opérateurs unitaires suivants sont les matrices de Pauli [95]. Elles agissent sur un qubit :*

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (1.3.2)$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad (1.3.3)$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (1.3.4)$$

La porte X permet les transformations suivantes : $X|0\rangle = |1\rangle$ et $X|1\rangle = |0\rangle$; la porte Z celles-ci : $Z|0\rangle = |0\rangle$ et $Z|1\rangle = -|1\rangle$.

Un autre exemple d'opérateur unitaire est la porte d'Hadamard qui agit également sur un qubit :

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (1.3.5)$$

La transformation d'Hadamard est particulièrement utile pour créer de la superposition à partir des états de base :

$$\begin{aligned} H|0\rangle &= |+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\ H|1\rangle &= |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \end{aligned} \quad (1.3.6)$$

La porte Control-Not est une transformation unitaire qui agit sur deux qubits. Elle applique une porte X au second qubit (le qubit cible) uniquement lorsque le premier qubit (le qubit contrôle) est dans l'état $|1\rangle$:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (1.3.7)$$

L'application de cette porte sur les états d'une base de $\mathbb{C}^2 \otimes \mathbb{C}^2$ produit les résultats suivants : $CNOT(|00\rangle) = |00\rangle$, $CNOT(|01\rangle) = |01\rangle$, $CNOT(|10\rangle) = |11\rangle$ et $CNOT(|11\rangle) = |10\rangle$.

Définition 1.3.4 (Observable [38, 95]). *En mécanique quantique, un observable est une grandeur physique qui peut être mesurée et qui est représentée mathématiquement par un opérateur hermitien agissant sur un espace de Hilbert \mathcal{H} . Les valeurs propres de cet opérateur correspondent aux résultats possibles de la mesure de l'observable, et ses vecteurs propres déterminent les états quantiques sur lesquels le système se projette au moment de la mesure.*

Exemple 1.3.5. Les matrices de Pauli ((1.3.2), (1.3.3), (1.3.4)) sont des observables de valeurs propres 1 et -1 . Les vecteurs propres de la matrice X sont $|+\rangle$ et $|-\rangle$. Ceux de la matrice Y sont $|0_y\rangle$ et $|1_y\rangle$ et ceux de la matrice Z sont $|0\rangle$ et $|1\rangle$. On peut écrire :

$$\begin{aligned} X &= |+\rangle\langle +| - |-\rangle\langle -| \\ Y &= |0_y\rangle\langle 0_y| - |1_y\rangle\langle 1_y| \\ Z &= |0\rangle\langle 0| - |1\rangle\langle 1| \end{aligned}$$

où $|+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $|-\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$, $|0_y\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix}$ et $|1_y\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -i \end{pmatrix}$.

1.4 Circuits Quantiques

Afin de définir les circuits quantiques, il nous faut nécessairement passer par la définition d'une porte quantique.

Définition 1.4.1 (Porte quantique [66, 90]). *En informatique quantique, une porte quantique U_i est une transformation unitaire qui agit sur les qubits. Les portes quantiques sont toujours réversibles.*

Exemple 1.4.1. Quelques exemples de portes quantiques.

— La porte X de la Section 1.3.2 : équivalent quantique de la porte logique NOT.

$$\begin{aligned} |0\rangle &\text{---} \boxed{X} \text{---} |1\rangle \\ |1\rangle &\text{---} \boxed{X} \text{---} |0\rangle \end{aligned}$$

— La porte H de la Section 1.3.5 : utilisée pour créer des superpositions d'états quantiques.

$$\begin{aligned} |0\rangle &\text{---} \boxed{H} \text{---} |+\rangle \\ |1\rangle &\text{---} \boxed{H} \text{---} |-\rangle \end{aligned}$$

— La porte CNOT de la Section 1.3.7 : agissant sur deux qubits, le premier qubit (le qubit de contrôle) détermine si le second qubit (le qubit cible) est inversé ou non.

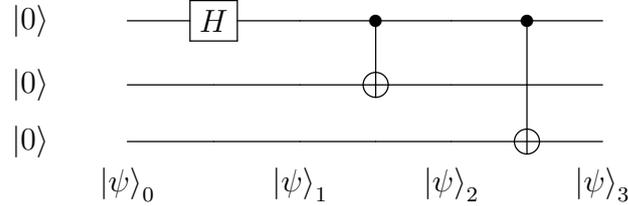
$$\begin{array}{cccccc} |0\rangle \text{---} \bullet \text{---} & |0\rangle & |0\rangle \text{---} \bullet \text{---} & |0\rangle & |1\rangle \text{---} \bullet \text{---} & |1\rangle & |1\rangle \text{---} \bullet \text{---} & |1\rangle \\ |0\rangle \text{---} \oplus \text{---} & |0\rangle & |1\rangle \text{---} \oplus \text{---} & |1\rangle & |0\rangle \text{---} \oplus \text{---} & |1\rangle & |1\rangle \text{---} \oplus \text{---} & |0\rangle \end{array}$$

Définition 1.4.2 (Circuit quantique [60]). *Un circuit quantique consiste en une séquence de portes quantiques. Pour un état initial $|\psi_0\rangle$ et un circuit quantique $Q = U_m \cdots U_0$ où U_i sont des portes unitaires, l'état final en sortie de circuit est donné par $|\psi\rangle = U_m \times \cdots \times U_0 |\psi_0\rangle$.*

Le principe d'un circuit quantique regroupe :

- en entrée, un système à n -qubits qui sont représentés par n fils parallèles.
- une succession de portes quantiques agissant sur un qubit ou plusieurs.
- des mesures qui renvoient des bits classiques (optionnel).

Exemple 1.4.2. Soit le circuit Q suivant :



L'état en entrée de Q est $|\psi_0\rangle = |0\rangle \otimes |0\rangle \otimes |0\rangle$ et $Q = CNOT\{0, 2\}CNOT\{0, 1\}H\{0\}$.
L'état en sortie obtenu est :

$$\begin{aligned} |\psi_0\rangle &= |000\rangle \\ |\psi_1\rangle &= \frac{1}{\sqrt{2}}(|000\rangle + |100\rangle) \\ |\psi_2\rangle &= \frac{1}{\sqrt{2}}(|000\rangle + |110\rangle) \\ |\psi_3\rangle &= \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \end{aligned}$$

Il s'agit de l'état $|GHZ\rangle_3$.

ORACLE Un oracle O_f [63], est une portion de circuit non exposée (une "boîte noire") qui est utilisée dans un autre algorithme.

Définition 1.4.3 (Fonction Oracle [44]). Une « fonction Oracle » est une représentation d'un circuit quantique généralement notée U_f qui implémente une fonction classique f . Certains oracles quantiques comprennent parfois des qubits auxiliaires qui sont nécessaires pour que l'opérateur soit réversible.

On montre que pour toute fonction classique $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$, on peut définir l'action d'un oracle U_f agissant sur $n + m$ qubits de la manière suivante :

$$\forall x \in \{0, 1\}^n \text{ et } y \in \{0, 1\}^m, U_f(|x\rangle \otimes |y\rangle) = |x\rangle \otimes |y \oplus f(x)\rangle \quad (1.4.1)$$

où \oplus est l'opération XOR classique.

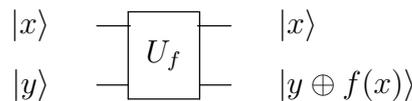


FIGURE 1.4.1 – Circuit représentant un oracle.

Exemple 1.4.3. Les oracles jouent un rôle central dans de nombreux algorithmes quantiques :

- dans l'algorithme de Deutsch-Jozsa, un oracle est utilisé pour représenter une fonction dont on veut savoir si elle est constante ou équilibrée [27];
- dans l'algorithme de Grover, un oracle est utilisé pour identifier un élément recherché dans une base de données non structurée;
- les oracles sont utilisés dans les algorithmes d'amplification d'amplitude et d'estimation de phase quantiques où ils aident à augmenter la probabilité que certains états apparaissent après des mesures [80].

Nous revenons sur ces algorithmes au Chapitre 2.

Les circuits quantiques notamment ceux issus d'algorithmes quantiques, peuvent être exécutés sur des simulateurs ou sur des ordinateurs quantiques. L'IBM Quantum Platform est un écosystème pour l'informatique quantique, comprenant des simulateurs et ordinateurs quantiques, des logiciels, des outils de développement et des ressources éducatives. Il a été développé par la société IBM. Dans les prochains paragraphes, nous explorerons l'univers de l'IBM Quantum Platform ainsi que le framework Qiskit de Python utilisé pour implémenter et tester des algorithmes quantiques.

IBM QUANTUM PLATFORM ET QISKIT L'IBM Quantum Platform [48] (auparavant IBM Quantum Experience) est une plateforme qui permet un accès public et premium aux services d'informatique quantique basés sur le cloud fournis par l'entreprise IBM. Cette plateforme permet, entre autres, d'accéder à des simulateurs et ordinateurs quantiques et de les utiliser pour créer et exécuter des circuits quantiques. La plateforme propose notamment une interface graphique conviviale appelée IBM Quantum Composer, ainsi qu'une interface de programmation en Qiskit, un framework de développement open source du langage Python. Il s'agit d'un ensemble d'outils, de bibliothèques et de diverses ressources permettant de concevoir, de simuler et d'exécuter des circuits quantiques.

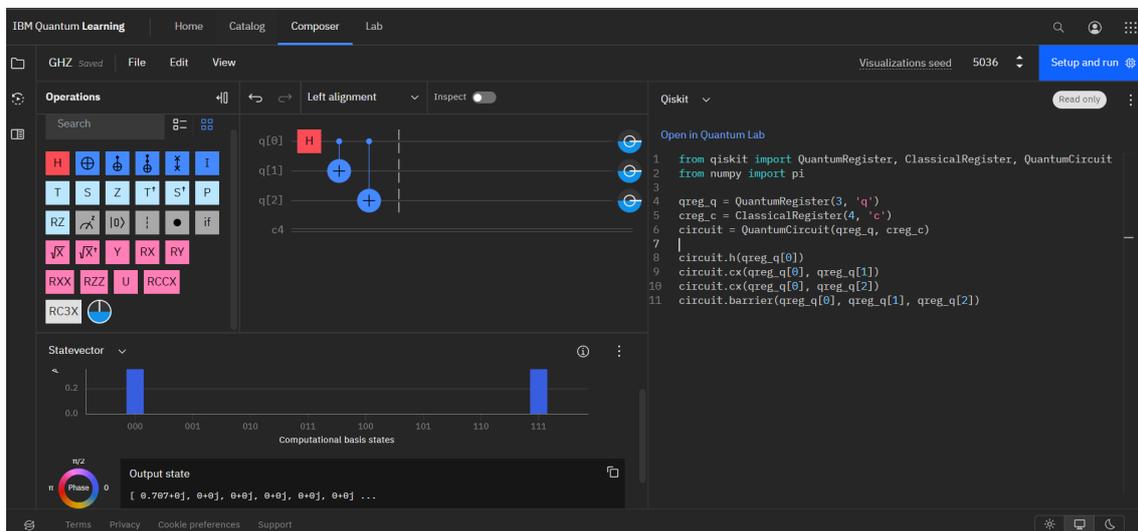


FIGURE 1.4.2 – Interface de l'outil IBM Quantum Composer.

La Figure 1.4.2 présente l'interface de IBM Quantum Composer avec trois parties principales. La première partie à gauche est constituée de portes et d'opérateurs classiques et quantiques qui permettent de créer des circuits quantiques directement visibles au centre. La seconde partie, située en dessous de la première, offre des options de visualisation des résultats de l'évaluation du circuit créé. La dernière partie présente le code Qiskit généré par le circuit.

Remarque 1.4.1. *D'autres entreprises développent aussi des plateformes d'informatique quantique qui comprennent des supports d'exécution pour les algorithmes quantiques. Voici un aperçu des plateformes et des langages associés à ces entreprises :*

- *Google (Google Quantum AI) : Google offre l'accès à son ordinateur quantique Sycamore via le service Google Quantum Computing. On peut y utiliser le langage de programmation Cirq développé par Google.*
- *Microsoft (Microsoft Quantum) : Microsoft propose le service Azure Quantum pour lequel il a spécialement développé le langage Q#.*
- *Rigetti (Rigetti Quantum Cloud Services) : Rigetti propose un service cloud et une interface de programmation Python pour interagir avec ses ordinateurs quantiques via la bibliothèque PyQuil.*

1.5 Intrication quantique

INTRICATION ET OPÉRATEUR DE BELL L'intrication quantique est l'un des concepts les plus fondamentaux et intrigants de la mécanique quantique. Elle se réfère à des corrélations non classiques entre les états quantiques de deux ou plusieurs particules où l'état de chaque particule ne peut pas être décrit indépendamment de l'autre, quelle que soit la distance qui les sépare. Elle fait l'objet de nombreuses études, expériences et applications [1, 31, 45, 47, 74]. Le concept d'intrication quantique a été popularisé pour la première fois dans un article de A. Einstein, B. Podolsky et N. Rosen publié en 1935 [29]. Cet article ne décrit pas explicitement l'état intriqué très connu du domaine $|\text{EPR}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$, mais il met en avant une situation où deux particules corrélées peuvent présenter des propriétés non locales, remettant en question l'idée que la mécanique quantique offre une description complète de la réalité physique. Einstein et ses collaborateurs ont ainsi suggéré que les résultats des mesures quantiques pourraient être déterminés par des variables préexistantes et cachées (inconnues), qui respecteraient le principe de localité, contrairement aux postulats fondamentaux de la mécanique quantique.

Dans les années 1960, J. Bell a démontré mathématiquement, par le biais des inégalités qui portent son nom, que si des variables locales cachées existaient, certaines prédictions de la mécanique quantique seraient incompatibles avec celles-ci [10]. La violation des inégalités de Bell par des expériences comme celles d'Alain Aspect et al dans les années 1980 [6] indique clairement que les corrélations observées entre particules intriquées ne peuvent pas être expliquées par des théories de variables cachées locales. Un exemple typique d'opérateur de Bell est l'opérateur CHSH, nommé d'après Clauser, Horne, Shimony et Holt [21] :

$$B = \left(Z \otimes \frac{(Z + X)}{\sqrt{2}} \right) + \left(X \otimes \frac{(Z + X)}{\sqrt{2}} \right) + \left(Z \otimes \frac{(Z - X)}{\sqrt{2}} \right) - \left(X \otimes \frac{(Z - X)}{\sqrt{2}} \right), \quad (1.5.1)$$

où X et Z sont des matrices de Pauli qui, ici mesurent deux particules dans des directions différentes. On note par $\langle B \rangle = |\langle \psi | B | \psi \rangle|$ la valeur moyenne des résultats de la mesure de l'état $|\psi\rangle$ par l'opérateur B . Les inégalités de Bell sont les suivantes :

$$\langle B \rangle \leq^{RL} 2 \text{ et } \langle B \rangle \leq^{MQ} 2\sqrt{2}, \quad (1.5.2)$$

où RL et MQ désignent les bornes de l'opérateur dans les hypothèses du Réalisme Local (RL) et de la Mécanique Quantique (MQ).

En 1980, A. Aspect et ses collaborateurs ont réalisé une série d'expériences pour tester les inégalités de Bell [6]. Celles-ci ont montré une violation claire de ces inégalités, confirmant ainsi que les corrélations entre les particules intriquées ne peuvent être expliquées par une théorie à variables cachées locales.

Dans cette thèse, nous n'emploierons que des états purs. Avant de donner une définition de l'intrication quantique, nous commençons par définir la séparabilité.

Définition 1.5.1 (Séparabilité [47] et Intrication [88]). *Soit $|\psi\rangle \in \mathcal{H} = \mathbb{C}^2 \otimes \dots \otimes \mathbb{C}^2$ un état à n qubits avec $n \geq 2$. L'état $|\psi\rangle$ est séparable si et seulement s'il existe des états $|\psi_i\rangle \in \mathbb{C}^2$ tels que :*

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \dots \otimes |\psi_n\rangle,$$

avec $i \in \mathbb{N}$ et $1 \leq i \leq n$. L'état $|\psi\rangle$ ainsi défini est séparable, c'est-à-dire non intriqué. Cependant, un état $|\Phi\rangle$ à n qubits est dit intriqué s'il ne peut s'écrire sous la forme d'une combinaison des états $|\psi_i\rangle \in \mathbb{C}^2$:

$$|\Phi\rangle \neq |\Phi_1\rangle \otimes |\Phi_2\rangle \otimes \dots \otimes |\Phi_n\rangle$$

Exemple 1.5.1. *Quelques exemples d'états séparables et d'états intriqués.*

$$|\psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

L'état $|\psi\rangle$ est séparable car $|\psi\rangle = |+\rangle \otimes |+\rangle$, $|+\rangle$ étant l'état dans l'Exemple 1.3.6.

$$|EPR\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

$$|GHZ\rangle_n = \frac{|00\dots 0\rangle + |11\dots 1\rangle}{\sqrt{2}}$$

Les états $|EPR\rangle$ et $|GHZ\rangle_n$ sont intriqués.

Un état quantique à n qubits est dit m -intriqué s'il existe au moins m qubits intriqués entre eux avec $n, m \in \mathbb{N}^*$ et $n \geq m$.

Exemple 1.5.2.

$$|GHZ\rangle_3 = \frac{|000\rangle + |111\rangle}{\sqrt{2}}$$

$$|W\rangle = \frac{|001\rangle + |010\rangle + |100\rangle}{\sqrt{3}}$$

Les états $|GHZ\rangle_3$ et $|W\rangle$ sont 3-intriqués.

On remarque que si la mesure du premier qubit de l'état $|GHZ\rangle_3$ renvoie $|0\rangle$, alors, l'état résultant de la mesure $|00\rangle$ est séparable. Cependant, si la mesure du premier qubit de l'état $|W\rangle$ renvoie $|0\rangle$, alors, l'état résultant de la mesure $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ est toujours intriqué. Ainsi, il existe différentes natures d'intrication. Pour étudier l'intrication quantique sous cet angle, nous introduisons les notions suivantes.

TRANSFORMATIONS LU [55, 58] L'intrication quantique reste invariante quelque soit le changement de base locale. En effet, le degré d'intrication d'un état reste inchangé sous l'application d'opérations unitaires locales (changement de base préservant les bases orthonormales), appelées transformations Local Unitary (LU). Une transformation LU qui agit sur un système quantique à n qubits est le produit de n matrices unitaires locales telles que :

$$U = U_1 \otimes U_2 \otimes \cdots \otimes U_n, \quad (1.5.3)$$

où U_i est une matrice unitaire de dimension 2×2 , c'est-à-dire $U_i \in U_2(\mathbb{C})$. Par conséquent, deux états $|\psi\rangle, |\phi\rangle \in \mathcal{H}$ sont dits LU-équivalents s'il existe une transformation unitaire locale $U \in LU = U_2(\mathbb{C}) \times \dots \times U_2(\mathbb{C})$ qui transforme $|\psi\rangle$ en $|\phi\rangle$ c'est-à-dire :

$$|\psi\rangle = U |\phi\rangle.$$

TRANSFORMATIONS SLOCC [11, 47] Les transformations Stochastic Local Operations and Classical Communication (SLOCC) sont une généralisation des transformations unitaires locales (dites LU) qui permettent de décrire une classe plus large de transformations possibles entre états quantiques. Une transformation $SLOCC$ qui agit sur un système quantique à n qubits est le produit de n matrices locales inversibles, pas nécessairement unitaires telles que :

$$A = A_1 \otimes A_2 \otimes \cdots \otimes A_n, \quad (1.5.4)$$

où A_i est une matrice inversible de dimension 2×2 , c'est-à-dire $A_i \in GL_2(\mathbb{C})$. Ainsi, deux états $|\psi\rangle, |\phi\rangle \in \mathcal{H}$ sont dits SLOCC-équivalents s'il existe une transformation inversible locale $A \in SLOCC = GL_2(\mathbb{C}) \times \dots \times GL_2(\mathbb{C})$ qui transforme $|\psi\rangle$ en $|\phi\rangle$ telle que

$$|\psi\rangle = A |\phi\rangle.$$

Les opérations unitaires locales ne modifient pas l'intrication d'un état. Comme cela sera défini au Chapitre 3, une mesure de l'intrication est caractérisée par son invariance par transformation LU. Par conséquent, si deux états sont LU-équivalents, alors ils ont quantitativement la même intrication. Ceci n'est plus vrai lorsqu'on considère les transformations SLOCC. En effet comme nous l'avons expliqué, le groupe de transformation SLOCC contient comme sous-groupe les transformations LU, c'est donc une classe de transformation plus large. Cependant, on peut montrer que toute tâche réalisable avec un état intriqué $|\psi\rangle$ l'est également avec un état intriqué $|\phi\rangle$ SLOCC équivalent à $|\psi\rangle$ mais avec potentiellement des probabilités différentes. Les transformations SLOCC préservent donc l'intrication qualitativement mais pas nécessairement quantitativement. Ainsi, si deux états sont SLOCC-équivalents, alors, ils appartiennent à un même type d'intrication mais dont le niveau d'intrication peut varier. Il est pertinent d'étudier les classes formées par ces transformations, ce qui a déjà été le cas de différents travaux [17, 40, 91, 94].

1.6 Conclusion

Ce premier chapitre a fourni une introduction aux principes de l'information quantique. Nous avons exploré des concepts tels que les qubits, les qudits, le produit tensoriel, les circuits quantiques et l'intrication quantique. Nous avons également discuté des ordinateurs quantiques utilisés pour créer, manipuler, simuler et exécuter des circuits quantiques. Les concepts abordés dans ce chapitre sont essentiels pour comprendre la suite de cette thèse. Ils établissent un cadre pour aborder les algorithmes quantiques qui constituent le sujet du chapitre suivant.

Chapitre 2

ALGORITHMES QUANTIQUES

Contenu du Chapitre

2.1	Contexte	16
2.2	Algorithmes quantiques	16
2.2.1	Algorithme de Deutsch-Jozsa	16
2.2.2	Algorithme de Grover	17
2.2.3	Algorithme d'estimation de phase	20
2.2.4	Algorithme de comptage quantique	21
2.3	Conclusion	23

2.1 Contexte

Après avoir fourni certaines bases de l'informatique quantique dans le chapitre précédent, nous nous tournons maintenant vers l'exploration des algorithmes quantiques qui utilisent ces bases, pour accomplir des tâches parfois plus rapides que les algorithmes classiques. D'abord, nous commençons par introduire l'algorithme de Deutsch-Jozsa, qui malgré sa simplicité, montre un certain avantage quantique théorique. Ensuite, nous présentons l'algorithme de Grover, célèbre pour sa capacité à trouver une occurrence dans un ensemble de données non ordonné plus rapidement qu'un algorithme homologue [39, 84, 93]. Enfin, en plus de ces algorithmes, nous étudions les algorithmes d'estimation de phase et de comptage quantiques. L'algorithme d'estimation de phase quantique permet de déterminer la phase du vecteur propre d'un opérateur unitaire tandis que l'algorithme de comptage quantique, quant à lui, se sert des techniques de Grover qu'il combine avec celui d'estimation de phase pour donner la possibilité de compter le nombre de solutions à un problème de recherche.

2.2 Algorithmes quantiques

Les algorithmes quantiques sont décrits par une séquence d'instructions logiques dont le but est de résoudre un problème donné, comme les algorithmes classiques. Cependant, il existe des différences significatives dans la manière dont ces deux types d'algorithmes manipulent l'information. En effet, les algorithmes classiques manipulent des bits classiques (0 ou 1) sur lesquels agissent des portes classiques (ET, OU, etc.) dans un ordre séquentiel et traduisibles dans des langages de programmation classiques (C, Java, etc.). Tandis que les algorithmes quantiques manipulent des qubits, les instructions sont des portes quantiques et ils peuvent exploiter la superposition quantique pour effectuer plusieurs calculs simultanément mais sont contraints par le phénomène de mesure, décrit au Chapitre 1. Pour finir, les algorithmes quantiques sont implémentés sur des ordinateurs quantiques. Par la suite, nous présentons des algorithmes quantiques en commençant par un des plus simples, celui de Deutsch-Jozsa.

2.2.1 Algorithme de Deutsch-Jozsa

D'abord découvert en 1985 par D. Deutsch [26], puis généralisé par lui-même ainsi que R. Jozsa en 1992 [27], l'algorithme de Deutsch-Jozsa fut enrichi par les améliorations de R. Cleve et al. en 1998 [22].

PRINCIPE Soit une fonction booléenne $f : \{0, 1\}^n \rightarrow \{0, 1\}$, avec $n \in \mathbb{N}^*$ qui est soit **constante** (la sortie $f(i)$ est 0 ou 1 pour toutes les entrées i) soit **équilibrée** (la sortie $f(i)$ est 0 dans la moitié des cas et 1 dans les autres). Le but de l'algorithme de Deutsch-Jozsa est de déterminer si la fonction f est constante ou équilibrée à l'aide d'un Oracle. Cet Oracle, noté U_f , permet la transformation

$$|x\rangle |y\rangle \longrightarrow |x\rangle |y \oplus f(x)\rangle$$

avec $x \in \{0, 1, \dots, 2^n - 1\}$.

Dans le cas où un algorithme classique déterministe est utilisé pour répondre à la problématique posée, dans le pire des cas il faut $2^{n-1} + 1$ **évaluations** de la fonction mathématique f pour être certain de répondre à la question. Cependant, l'algorithme

quantique de Deutsch-Jozsa permet de trouver une réponse toujours correcte avec **une seule application** de U_f . Le circuit qui représente cet algorithme est illustré à la Figure 2.2.1.

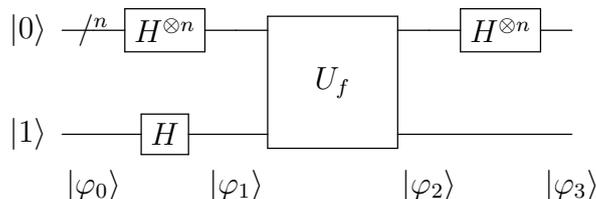


FIGURE 2.2.1 – Circuit illustrant l’algorithme de Deutsch-Jozsa.

Le circuit de l’algorithme de Deutsch-Jozsa de la Figure 2.2.1 prend en entrée l’état $|\varphi_0\rangle = |0\rangle^{\otimes n} \otimes |1\rangle$, et produit en sortie l’état $|\varphi_3\rangle$. Les états $|\varphi_i\rangle$ indiquent les différentes étapes de l’algorithme.

ETAPES Les étapes de l’algorithme de Deutsch-Jozsa sont les suivantes :

$$\begin{aligned}
 |\varphi_0\rangle &= |0\rangle^{\otimes n} |1\rangle \\
 |\varphi_1\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \\
 |\varphi_2\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \\
 |\varphi_3\rangle &= \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \sum_{z=0}^{2^n-1} (-1)^{x \bullet z} |z\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \\
 |\varphi_3\rangle &= \sum_{z=0}^{2^n-1} \sum_{x=0}^{2^n-1} \frac{(-1)^{x \bullet z + f(x)}}{2^n} |z\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right),
 \end{aligned}$$

où $x \bullet z = \sum_{i=1}^n x_i z_i \pmod{2}$. Dans l’état final $|\varphi_3\rangle$, la quantité $\alpha_0 = \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)}$ est l’amplitude de l’état $|0\rangle^{\otimes n}$ et $p_0 = |\alpha_0|^2$ est la probabilité de le mesurer. On déduit que :

- si f est équilibrée, alors, les valeurs 1 et -1 s’équilibrent. Ainsi, $\alpha_0 = 0$. L’état $|0\rangle^{\otimes n}$ ne peut être mesuré.
- si f est constante, alors, $\alpha_0 = 1$ ou $\alpha_0 = -1$ et $p_0 = 1$. L’état $|0\rangle^{\otimes n}$ sera le seul mesuré.

En conclusion, la mesure de l’état final en sortie de l’algorithme permet de répondre à la question initialement posée (constante ou équilibrée) en un seul appel à l’oracle U_f , exploitant ainsi un avantage quantique qui serait impossible à atteindre dans le cas classique.

2.2.2 Algorithme de Grover

L’algorithme de Grover [39] est un algorithme quantique qui amplifie la probabilité de trouver une solution dans un espace de recherche non structuré. Cet algorithme utilise

$O(\sqrt{N})$ appels à un oracle pour maximiser la probabilité de succès, tandis qu'une recherche exhaustive classique nécessite $O(N)$ appels. M. Boyer et al. [13] ont proposé une analyse détaillée de l'algorithme, incluant une formule pour estimer le nombre optimal d'itérations, qui est approximativement $\frac{\pi}{4}\sqrt{N}$, afin de maximiser la probabilité de succès. C. Zalka [93] a démontré que l'algorithme de Grover est asymptotiquement optimal dans le cadre des modèles d'oracle pour les problèmes de recherche non structurée.

PRINCIPE Supposons que nous avons $N = 2^n$ éléments dans un espace de recherche et que nous les indexons en affectant à chaque élément un entier compris entre 0 et $N - 1$. Soit la fonction f qui vaut 1 si x est l'élément recherché et 0 sinon. Elle est fournie sous la forme d'un oracle. L'algorithme de Grover comporte deux éléments principaux, à savoir : un Oracle O_f et un opérateur de diffusion.

ETAPES Les étapes de l'algorithme sur le premier registre sont :

1. commencer avec un registre de n qubits initialisés à l'état $|0^n\rangle$;
2. appliquer une porte Hadamard à chaque qubit du registre ;
3. appliquer les opérations suivantes au registre $\frac{\pi}{4}\sqrt{N}$ fois :
 - (a) l'oracle U_f qui signe la solution ;
 - (b) la porte de diffusion qui inverse l'amplitude de tous les états par rapport à leur moyenne ;
4. mesurer le registre pour obtenir l'index de la solution avec une très grande probabilité ;
5. vérifier qu'il s'agit d'une solution valide. Si ce n'est pas le cas, recommencer.

Le circuit complet qui permet de réaliser l'algorithme de Grover est placé sur la Figure 2.2.2.

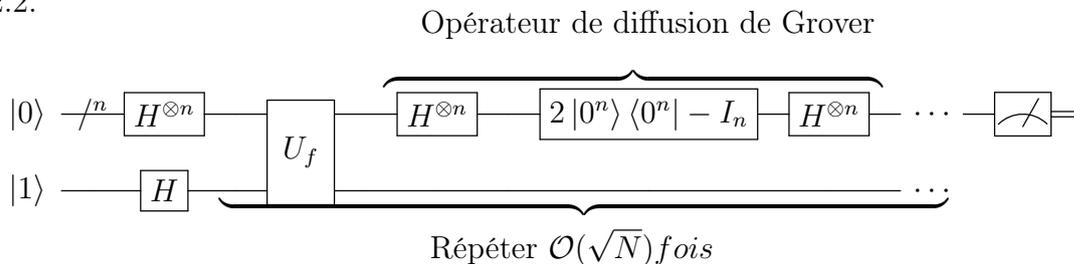


FIGURE 2.2.2 – Circuit représentatif de l'algorithme de Grover.

Le point (3) des étapes de l'algorithme est connu sous le nom *d'itération Grover ou de matrice de Grover* et regroupe l'action de l'oracle et de la porte de diffusion. L'itération de Grover G peut être considérée comme une rotation dans le plan engendré par l'axe de la somme des états solutions $|s\rangle$ et de celui de la somme des états non solutions $|w\rangle$. Elle s'écrit :

$$G = \begin{pmatrix} \cos 2\theta_0 & -\sin 2\theta_0 \\ \sin 2\theta_0 & \cos 2\theta_0 \end{pmatrix} \quad (2.2.1)$$

où θ_0 est l'angle formé par l'axe $|w\rangle$ et l'état initial totalement superposé du premier registre $H^{\otimes n}|0\rangle^{\otimes n} = |+\rangle^{\otimes n} = |t\rangle$.

Les illustrations de la Figure 2.2.4 représentent les étapes de l'algorithme pour 3 qubits, la solution recherchée étant 6 : (a) Le rectangle rouge représente la valeur à déterminer

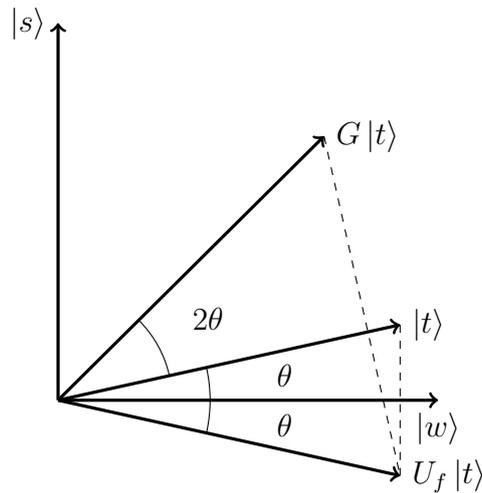


FIGURE 2.2.3 – Interprétation géométrique de l’algorithme de Grover.

dans le lot de $2^3 = 8$. Ces différentes valeurs correspondent aux amplitudes des vecteurs de base d’un état à 3 qubits. Les probabilités de mesure sont identiques pour tous les cas. (b) On applique l’oracle qui signe l’amplitude correspondante à l’élément recherché. Toutefois, cela ne modifie pas la probabilité de mesurer l’état en question (puisque’elle est égale à la valeur absolue de l’amplitude au carré). (c) On calcule la moyenne des coefficients. (d) La dernière étape de l’algorithme consiste à appliquer la porte de diffusion pour réaliser une symétrie par rapport à la moyenne des amplitudes. Cela a pour résultat d’augmenter la probabilité de mesurer l’amplitude de l’élément recherché et, par la même occasion, de diminuer celle des autres. Dans ce cas, on réalise les étapes (b) à (d) $\frac{\pi}{4}\sqrt{2^3} = 2,22 \approx 2$ fois pour augmenter la probabilité de mesurer l’état 6.

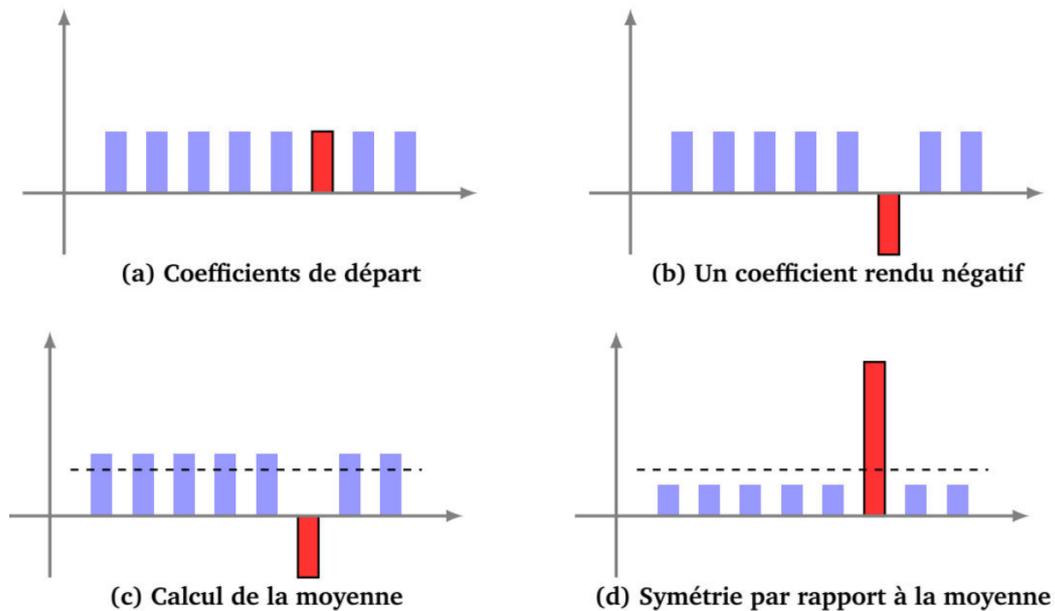


FIGURE 2.2.4 – Évolution des amplitudes du premier registre au cours de la première itération de l’algorithme de Grover pour 3 qubits et l’état recherché 6.

2.2.3 Algorithme d'estimation de phase

L'intrication de cet algorithme sera étudiée plus tard au Chapitre 5.

L'algorithme d'estimation de phase quantique AEPQ (en anglais *Quantum Phase Estimation Algorithm QPEA*) a pour but d'estimer la valeur propre (ou la phase quantique) d'un opérateur unitaire, de vecteur propre donné que ce dernier soit connu ou non. L'implémentation de cet algorithme implique l'utilisation d'un circuit quantique bien connu qui est l'inverse de la Transformée de Fourier Quantique TFQ (en anglais *Quantum Fourier Transform QFT*). On peut considérer que l'algorithme est composé de deux étapes principales. La première concerne toutes les opérations réalisées avant d'appliquer une porte TFQ[†]. Nous l'appellerons partie pré-TFQ. La seconde étape de l'algorithme consiste à appliquer la porte TFQ inverse sur le premier registre pour obtenir l'estimation de la phase recherchée.

PRINCIPE Soit U un opérateur unitaire de valeur propre $e^{2i\pi\phi}$ et de vecteur propre $|u\rangle$ tels que $U|u\rangle = e^{2i\pi\phi}|u\rangle$. Le but de cet algorithme est d'estimer la phase $\phi \in [0, 1]$.

Nous supposons que nous avons accès à la porte unitaire U comme une boîte noire et que l'on peut effectuer des opérations contrôle- U^j avec $j \in \mathbb{N}^*$. La partie pré-TFQ est composée de deux registres quantiques. Le premier registre est constitué de t qubits choisis en fonction de la précision de l'estimation de ϕ souhaitée. Le deuxième registre est initialisé à l'état $|u\rangle$ et une série de portes C- U^j sera appliquée à ce registre avec des bits de contrôle sur le premier registre. Dans le formalisme du circuit, l'étape pré-TFQ de l'algorithme d'estimation de phase est reproduite dans la Figure 2.2.5 et une version complète de l'algorithme est produite à la Figure 2.2.6.

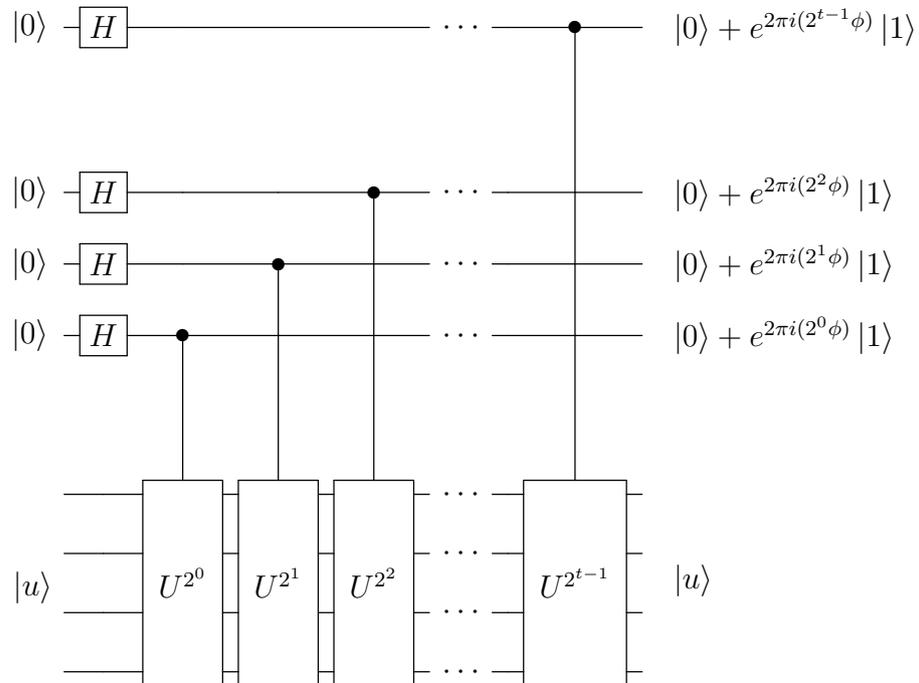


FIGURE 2.2.5 – Partie pré-TFQ de l'algorithme d'estimation de phase quantique (AEPQ).

La mesure de l'état final sur le premier registre rapporte une approximation de la phase ϕ désirée.

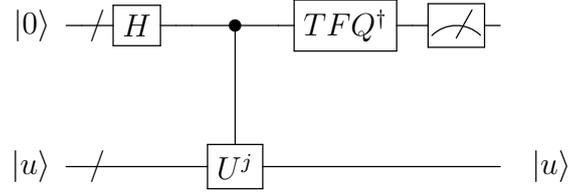


FIGURE 2.2.6 – Schéma complet de l’algorithme d’estimation de phase quantique (AEPQ).

ETAPES On peut montrer [66, p222] qu’à la fin de la partie pré-TFQ de l’AEPQ, l’état quantique généré en fonction de la phase ϕ s’écrit de la manière suivante.

$$\begin{aligned}
 |\psi_\phi\rangle &= \frac{1}{\sqrt{2^n}} (|0\rangle + e^{2i\pi 2^{n-1}\phi} |1\rangle) \otimes (|0\rangle + e^{2i\pi 2^{n-2}\phi} |1\rangle) \otimes \dots \otimes (|0\rangle + e^{2i\pi 2^0\phi} |1\rangle) \otimes |u\rangle \\
 &= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2i\pi k\phi} |k\rangle |u\rangle
 \end{aligned} \tag{2.2.2}$$

où $|k\rangle$ avec $0 \leq k \leq 2^n - 1$ désigne la base standard de l’espace de Hilbert.

La dernière étape consiste à appliquer la porte TFQ^\dagger à l’état $|\psi_\phi\rangle$ de l’équation 2.2.2. En effet, il est connu que [66] :

$$TFQ^\dagger \left(\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2i\pi\phi k} |k\rangle |u\rangle \right) = |\tilde{\phi}\rangle |u\rangle. \tag{2.2.3}$$

Si le vecteur propre $|u\rangle$ de la porte U n’est pas connu, il est possible de mettre au deuxième registre la superposition de tous les états de base. Cette superposition se décompose en base de vecteurs propres de U dont la dimension est $2^t \times 2^t$:

$$|+\rangle \otimes \dots \otimes |+\rangle = \sum_{j=0}^{2^t-1} a_j |u_j\rangle. \tag{2.2.4}$$

En conclusion, l’AEPQ permet donc d’estimer la phase de la valeur propre d’un opérateur unitaire U indépendamment du fait que le vecteur propre de l’opérateur soit connu ou non.

2.2.4 Algorithme de comptage quantique

L’intrication de cet algorithme sera également étudiée plus tard au Chapitre 5.

Imaginons que nous ayons un ensemble N de balles, dont un certain nombre M d’entre elles sont marquées d’une manière particulière. Le but est de trouver rapidement le nombre de balles marquées en effectuant le moins de recherches possibles. Ceci peut se faire classiquement par une recherche exhaustive, c’est-à-dire en $O(N)$ consultations de l’Oracle. Quantiquement, en combinant les techniques de l’algorithme de Grover et de celui d’estimation de phase, l’algorithme de comptage quantique ACQ (*Quantum Counting Algorithm QCA* en anglais) offre la possibilité d’estimer le ratio $\frac{M}{N}$ en faisant $\mathcal{O}(\sqrt{N})$ appels à l’Oracle [14].

PRINCIPE Soit $2^n = N$ le nombre d'éléments d'une liste qui sont identifiés aux états de base $|x\rangle$ avec $x \in \{0, \dots, N-1\}$. Notons $S = \{x'_1, x'_2, \dots, x'_M\} \subset \{0, \dots, N-1\}$ la sous-liste des éléments marqués M (les éléments reconnus par l'Oracle). Introduisons les états :

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{x \notin S} |x\rangle \quad \text{et} \quad |\beta\rangle = \frac{1}{\sqrt{M}} \sum_{x \in S} |x\rangle. \quad (2.2.5)$$

L'état $|\alpha\rangle$ correspond à la somme de tous les éléments non marqués (les "mauvais" états) tandis que l'état $|\beta\rangle$ représente celle de tous les éléments marqués (les "bons" états). La matrice de l'équation est la matrice de Grover étudiée plus tôt (2.2.1). Cependant, ici, l'angle de rotation θ est déterminé par l'itération de Grover pour les éléments marqués M , soit $\sin\left(\frac{\theta}{2}\right) = \sqrt{\frac{M}{N}}$. Et cette matrice s'écrit :

$$\mathcal{G} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

Soient $|\psi_+\rangle = \frac{1}{\sqrt{2}} |\alpha\rangle + \frac{i}{\sqrt{2}} |\beta\rangle$ et $|\psi_-\rangle = \frac{1}{\sqrt{2}} |\alpha\rangle - \frac{i}{\sqrt{2}} |\beta\rangle$ les vecteurs propres de la matrice \mathcal{G} de valeurs propres respectives $e^{-i\theta}$ et $e^{i\theta}$. En particulier :

$$\begin{aligned} \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle &= \sqrt{\frac{N-M}{N}} |\beta\rangle + \sqrt{\frac{M}{N}} |\alpha\rangle \\ &= \cos\left(\frac{\theta}{2}\right) |\beta\rangle + \sin\left(\frac{\theta}{2}\right) |\alpha\rangle \\ \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle &= e^{-i\frac{\theta}{2}} |\psi_+\rangle + e^{i\frac{\theta}{2}} |\psi_-\rangle \end{aligned} \quad (2.2.6)$$

On remarque que cette équation 2.2.6 est un cas particulier de l'équation 2.2.4 de l'algorithme d'estimation de phase quantique. Donc si on applique l'AEPQ avec $U = \mathcal{G}$ sur la superposition complète $\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$, on aura une estimation de l'angle θ , et donc de $M = N \sin^2\left(\frac{\theta}{2}\right)$ avec $\mathcal{O}(\sqrt{N})$ appels à \mathcal{G} . Le circuit qui illustre l'algorithme de comptage quantique est présenté à la Figure 2.2.7. Le premier et le second registre de la Figure commencent tous deux par la superposition apportées par les portes Hadamard. Sur le second registre, $|u\rangle$ est la superposition des deux vecteurs propres de la matrice d'itération de Grover \mathcal{G} .

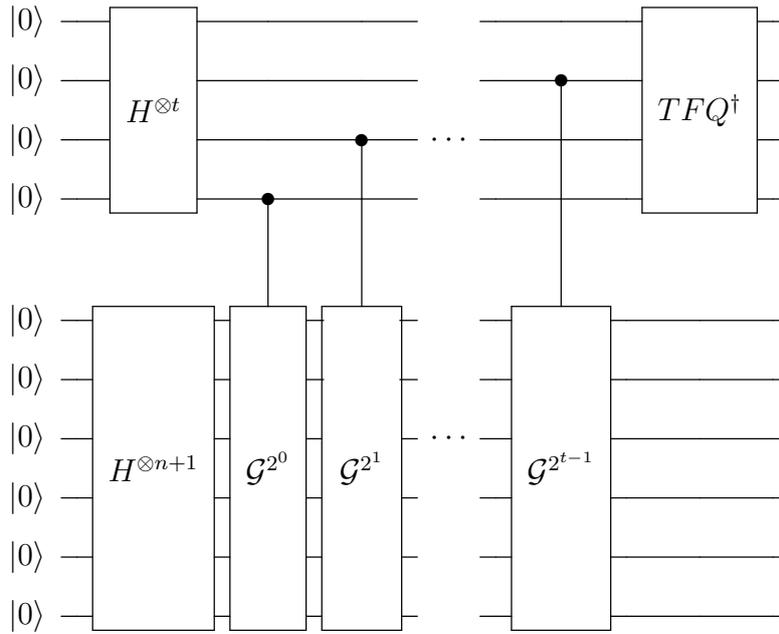


FIGURE 2.2.7 – Circuit qui illustre l’algorithme de comptage quantique (ACQ).

2.3 Conclusion

Ce chapitre a introduit plusieurs algorithmes quantiques emblématiques, tels que l’algorithme de Deutsch-Jozsa, l’algorithme de Grover, ainsi que les algorithmes d’estimation de phase et de comptage quantique. Ces algorithmes illustrent comment, dans des contextes spécifiques, l’informatique quantique peut offrir des solutions plus rapides ou plus efficaces que ce qui est réalisable classiquement. Néanmoins, les algorithmes quantiques ne surpassent pas toujours les algorithmes classiques, notamment en raison des défis liés à la construction des oracles quantiques, des contraintes liées au bruit et des ressources nécessaires pour leur implémentation.

L’exploration des algorithmes quantiques est importante pour la suite de notre étude car nous reviendrons sur deux d’entre eux, particulièrement les algorithmes d’estimation de phase et de comptage quantiques, pour les étudier du point de vue de l’intrication. Afin de réaliser cette étude, il est primordial de parler des outils que nous avons utilisés pour détecter l’intrication quantique. Ainsi, dans le prochain chapitre nous aborderons quatre outils de détection et/ou de mesure de l’intrication quantique.

Chapitre 3

OUTILS DE MESURE DE L'INTRICATION

Contenu du Chapitre

3.1	Contexte	25
3.2	Algorithmes de recherche d'optimum	25
3.3	Mesure géométrique de l'intrication (MGI)	25
3.4	Polynômes de Mermin et Invariant μ	26
3.4.1	Polynômes de Mermin et Inégalités de Mermin	26
3.4.2	Invariant μ	28
3.4.3	Invariant $\tilde{\mu}$	29
3.5	Matrices de coefficients et E_{moy}	29
3.5.1	Définitions préalables	29
3.5.2	Invariant d'intrication moyen E_{moy}	30
3.6	Multirang	31
3.7	Implémentations sur un ordinateur quantique	32
3.7.1	Implémentation de MGI	32
3.7.2	Implémentation de μ	33
3.8	Conclusion	35

3.1 Contexte

Faisant suite à l'étude de quelques algorithmes quantiques, nous nous intéressons aux apports de l'intrication aux performances des algorithmes quantiques par rapport aux algorithmes classiques. Pour se faire, nous passons par la quantification de l'intrication présente dans les algorithmes quantiques à l'aide d'outils de mesure. Pour qu'un outil de mesure de l'intrication soit classé comme tel, il faut notamment qu'il reste invariant sous l'action d'opérations locales unitaires (LU) [82]. Dans la suite de notre rédaction, nous appellerons un tel outil **un invariant**. Par la suite sont décrits les cinq invariants utilisés pour détecter et mesurer l'intrication. Il s'agit de la mesure géométrique de l'intrication (MGI), l'invariant de μ et sa variante $\tilde{\mu}$ calculés à partir des polynômes de Mermin, de l'invariant d'intrication moyenne E_{moy} et du multirang tous les deux obtenus à partir des matrices de coefficients. Afin de calculer trois de ces invariants, un algorithme de recherche d'un état séparable "optimum" (dans le cas de la MGI) et des paramètres de calcul des polynômes de Mermin (dans le cas de μ et sa variante $\tilde{\mu}$) a été utilisé. Dans ce chapitre, nous présentons d'abord les algorithmes de recherche d'optimum dont deux en particulier, qui sont l'algorithme de marche aléatoire et l'algorithme de recherche par tabou. Ensuite, nous présentons les invariants cités plus tôt.

3.2 Algorithmes de recherche d'optimum

Dans le calcul de nos invariants, nous employons deux types d'algorithmes de recherche qui sont l'algorithme de marche aléatoire et l'algorithme de recherche taboue.

Le concept de marche aléatoire est un modèle mathématique qui représente une série de pas aléatoires successifs le long d'un espace mathématique discret ou continu [64]. Les marches aléatoires sont couramment utilisées pour modéliser des phénomènes aléatoires, des processus stochastiques, ou pour résoudre des problèmes d'optimisation. Il s'agit un concept largement utilisé en mathématiques, en statistiques et en sciences de l'informatique [81]. Dans notre cas, l'implémentation de l'algorithme de marche aléatoire a consisté à effectuer une recherche dans un ensemble continu, sur un nombre fixe de pas, en partant d'un point choisi de manière arbitraire. À chaque pas, une direction aléatoire est choisie et permet de sélectionner un nouveau point dans le voisinage du point. Si la valeur de l'invariant à ce nouveau point est supérieure à celle du point précédent, car il s'agit de la recherche d'un maximum (voir les sections suivantes), alors ce point devient le nouveau point de départ de l'étape suivante. Sinon un nouveau point est choisi et le cercle de recherche rétréci, puis la recherche reprend à partir du nouveau point. L'algorithme correspondant à cette description est fourni à l'Annexe .1.1. Le pas de départ est $pas = 5$, le pas minimum à atteindre est $pas_{min} = 10^{-5}$ et le compteur vaut $cpt = 200$.

Une variante de cette recherche est la recherche [79] qui consiste à renseigner tous les points visités dans une liste. Les points déjà visités sont donc rajoutés à la liste dite taboue et exclus de la recherche.

3.3 Mesure géométrique de l'intrication (MGI)

Le premier invariant que nous avons utilisé est la mesure géométrique de l'intrication (MGI). La MGI quantifie l'intrication d'un état quantique $|\psi\rangle$ en mesurant sa distance par rapport à l'ensemble des états séparables. Cela consiste à trouver un état séparable $|\phi\rangle$ (pas forcément le seul) qui maximise la quantité $|\langle\psi|\phi\rangle|^2$.

Avant de définir l'invariant MGI , on introduit l'ensemble des états séparables Sep dans un espace de Hilbert \mathcal{H}_n associé à un système quantique à n qubits comme suit :

$$Sep = \{|\phi\rangle \in \mathcal{H}_n, |\phi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \otimes \dots \otimes |\phi_n\rangle, |\phi_i\rangle \in \mathbb{C}^2\} \quad (3.3.1)$$

L'ensemble des états séparables Sep décrit tous les états dans \mathcal{H}_n qui peuvent être décomposés comme un produit tensoriel de n états. Cela contraste avec les états intriqué.

Définition 3.3.1 (Mesure géométrique de l'intrication [87]). *On définit la mesure géométrique de l'intrication d'un état $|\psi\rangle$ comme suit :*

$$MGI(|\psi\rangle) = 1 - \text{Max}_{|\phi\rangle \in Sep} |\langle \psi | \phi \rangle|^2. \quad (3.3.2)$$

Si $|\psi\rangle$ est séparable alors, $\text{Max}_{|\phi\rangle \in Sep} |\langle \psi | \phi \rangle|^2 = 1$, donc $MGI(|\psi\rangle) = 0$.

Pour trouver un état $|\phi\rangle$ qui maximise la quantité $|\langle \psi | \phi \rangle|^2$, nous avons employé l'algorithme de marche aléatoire.

Exemple 3.3.1. *Quelques valeurs de MGI :*

- $MGI(|00\rangle) = 0$
- $MGI(|GHZ_3\rangle) = 0.5$
- $MGI(|W_3\rangle) = 0.56$

Dans l'analyse des algorithmes quantiques, la MGI a plusieurs fois été utilisée notamment pour mesurer la variation de l'intrication dans l'algorithme de Grover [18, 71] ainsi que dans celui de Shor [75]. Il a également été étudié par T. Wei et al [47, 76]. La mesure géométrique de l'intrication possède l'avantage d'être calculable sur un ordinateur quantique. Nous expliquons la démarche utilisée pour cela à la Section 3.7.

L'invariant que nous abordons par la suite, ainsi que la variante qui est issu de cet invariant, se basent sur les polynômes de Mermin.

3.4 Polynômes de Mermin et Invariant μ

L'inégalité de Bell [10] a été introduite pour répondre à l'article d'A. Einstein et al. [29] qui s'opposait au principe de non-localité de la mécanique quantique. Elle a été testée expérimentalement par A. Aspect et al. [6]. D. Mermin a proposé des polynômes qui généralisent les opérateurs de Bell pour un système à plusieurs qubits [62]. Ces polynômes permettent de révéler la non-localité d'un système à plusieurs qubits, ce qui implique la présence d'intrication [36].

3.4.1 Polynômes de Mermin et Inégalités de Mermin

Avant de définir l'invariant qui leur est associé, reprenons d'abord la définition des polynômes de Mermin suivie de celle des inégalités de Bell et enfin celles de Mermin.

Définition 3.4.1 (Polynômes de Mermin [2, 62]). *Les polynômes de Mermin sont des opérateurs définis de manière récursive.*

$$\begin{aligned} M_1 &= a_1, \\ M_n &= \frac{1}{2}(M_{n-1} \otimes (a_n + a'_n) + M'_{n-1} \otimes (a_n - a'_n)). \end{aligned} \quad (3.4.1)$$

où $\{a_1, a_2, \dots, a_m\}$ et $\{a'_1, a'_2, \dots, a'_m\}$ sont des familles d'observables à un qubit avec $a_i = \alpha_i X + \beta_i Y + \gamma_i Z$, $\alpha_i, \beta_i, \gamma_i \in \mathbb{R}$ et $\alpha_i^2 + \beta_i^2 + \gamma_i^2 = 1$. Ces conditions sont les mêmes pour a'_i . Les matrices X, Y, Z sont les matrices de Pauli présentées au premier chapitre ((1.3.2), (1.3.3), (1.3.4)). L'opérateur M'_n est le polynôme défini de la manière suivante.

$$\begin{aligned} M'_1 &= a'_1, \\ M'_n &= \frac{1}{2}(M'_{n-1} \otimes (a'_n + a_n) + M_{n-1} \otimes (a'_n - a_n)). \end{aligned} \quad (3.4.2)$$

Exemple 3.4.1. Expressions de M_n pour quelques valeurs de n :

$$\begin{aligned} M_1 &= a_1, \\ M_2 &= \frac{1}{2}(a_1 a_2 + a_1 a'_2 + a'_1 a_2 + a'_1 a'_2) \\ M_3 &= \frac{1}{2}(a'_1 a_2 a_3 + a_1 a'_2 a_3 + a_1 a_2 a'_3 - a'_1 a'_2 a'_3) \end{aligned}$$

INÉGALITÉS DE MERMIN [24] Soit l'opérateur de Bell suivant, comme défini au Chapitre 1 :

$$B = Z \otimes \frac{(Z+X)}{\sqrt{2}} + X \otimes \frac{(Z+X)}{\sqrt{2}} + Z \otimes \frac{(Z-X)}{\sqrt{2}} - X \otimes \frac{(Z-X)}{\sqrt{2}}.$$

En remplaçant les observables $a_1 = Z, a_2 = X, a'_1 = \frac{X+Z}{\sqrt{2}}$ et $a'_2 = \frac{X+Z}{\sqrt{2}}$ dans le polynôme M_2 issu de la définition (3.4.1), on obtient :

$$M_2 = \frac{1}{2} \left(Z \otimes \frac{(Z+X)}{\sqrt{2}} + X \otimes \frac{(Z+X)}{\sqrt{2}} + Z \otimes \frac{(Z-X)}{\sqrt{2}} - X \otimes \frac{(Z-X)}{\sqrt{2}} \right).$$

À un facteur près, M_2 correspond à l'opérateur B utilisé pour prouver le théorème de Bell (1.5.1). On rappelle que les inégalités de Bell (1.5.2) se traduisent par :

$$\langle B \rangle^{RL} \leq 2 \text{ et } \langle B \rangle^{MQ} \leq 2\sqrt{2},$$

où RL et MQ désignent respectivement les valeurs moyennes de l'opérateur dans les hypothèses du Réalisme Local (RL) et de la Mécanique Quantique (MQ). Le coefficient de normalisation des polynômes de Mermin modifie la valeur des deux bornes sans toutefois changer la nature des inégalités :

$$\langle M_2 \rangle^{RL} \leq 1 \text{ et } \langle M_2 \rangle^{MQ} \leq \sqrt{2}.$$

Plus généralement on peut prouver que [2, 62] :

$$\langle M_n \rangle^{RL} \leq 1 \text{ et } \langle M_n \rangle^{MQ} \leq 2^{\frac{n-1}{2}}, \quad \forall n \in \mathbb{N}^*. \quad (3.4.3)$$

En particulier, $\langle M_n \rangle > 1$ signifie que l'état $|\psi\rangle$ est non-local, donc intriqué. Sur la base des polynômes de Mermin et de la valeur moyenne d'un opérateur, nous définissons encore un invariant que nous nommerons μ .

3.4.2 Invariant μ

L'invariant μ a déjà été utilisé pour détecter l'intrication dans l'algorithme de Grover et la Transformée de Fourier Quantique (TFQ) [24, 56]. Il a également été utilisé pour étudier la propriété non locale des états issus d'hypergraphes à quatre qubits comme outil de classification [4]. Ce dernier point est détaillé dans le Chapitre 4.

Définition 3.4.2 (Invariant μ [24]). *Pour un état donné $|\psi\rangle \in \mathbb{C}^n$, on peut considérer la quantité :*

$$\mu(|\psi\rangle) = \text{Max}_{\alpha_1, \beta_1, \dots, \gamma'_n} \langle \psi | M_n(\alpha_1, \beta_1, \dots, \gamma'_n) | \psi \rangle, \quad (3.4.4)$$

où $M_n(\alpha_1, \beta_1, \dots, \gamma'_n)$ est le polynôme de Mermin représenté par une matrice de taille $2^n \times 2^n$ calculée à partir des $6n$ coefficients réels $\alpha_1, \beta_1, \dots, \gamma'_n$.

On note que :

- $\mu(|\psi\rangle) = 1$ signifie que $|\psi\rangle$ est un état séparable.
- $\mu(|\psi\rangle) > 1$ signifie que $|\psi\rangle$ est intriqué.

Pour trouver l'ensemble des valeurs qui maximisent $\langle \psi | M_n(\alpha_1, \beta_1, \dots, \gamma'_n) | \psi \rangle$, nous avons à nouveau utilisé l'algorithme de marche aléatoire.

Exemple 3.4.2. *Valeurs de μ pour quelques états :*

$$\begin{aligned} \mu(|EPR\rangle) &= \sqrt{2} \\ \mu(|GHZ_3\rangle) &= 2. \end{aligned}$$

Proposition 3.4.1. *La quantité $\mu(|\psi\rangle)$ définie à l'équation (3.4.4) est un invariant LU.*

Démonstration. On peut montrer qu'un observable a à un qubit ayant pour valeurs propres $\{-1, 1\}$ peut toujours s'écrire comme $a = \alpha X + \beta Y + \gamma Z$ avec $\alpha, \beta, \gamma \in \mathbb{R}$ et $\alpha^2 + \beta^2 + \gamma^2 = 1$.

Notons $\tilde{\alpha}_1, \tilde{\beta}_1, \dots, \tilde{\gamma}'_n$ un ensemble de paramètres qui maximisent μ , c'est-à-dire que

$$\mu(|\psi\rangle) = \langle \psi | M_n(\tilde{\alpha}_1, \tilde{\beta}_1, \dots, \tilde{\gamma}'_n) | \psi \rangle.$$

Soit $|\phi\rangle$ un état à n qubits LU-équivalent à l'état $|\psi\rangle$. Alors, il existe $g = (g_1, \dots, g_n) \in \text{LU}$ tel que $|\phi\rangle = |g \cdot \psi\rangle = G |\psi\rangle$ avec $G = g_1 \otimes \dots \otimes g_n$ et $G^\dagger G = I$. On peut écrire :

$$\begin{aligned} \langle \psi | M_n(\tilde{\alpha}_1, \tilde{\beta}_1, \dots, \tilde{\gamma}'_n) | \psi \rangle &= (\langle \psi | G^\dagger) G M_n(\tilde{\alpha}_1, \tilde{\beta}_1, \dots, \tilde{\gamma}'_n) G^\dagger (G |\psi\rangle) \\ &= \langle \phi | M_n(\tilde{\alpha}_1, \tilde{\beta}_1, \dots, \tilde{\gamma}'_n) | \phi \rangle \end{aligned}$$

avec des nouveaux paramètres $\tilde{\alpha}_1, \tilde{\beta}_1, \dots, \tilde{\gamma}'_n$.

Or $\langle \phi | M_n(\tilde{\alpha}_1, \tilde{\beta}_1, \dots, \tilde{\gamma}'_n) | \phi \rangle \leq \mu(|\psi\rangle)$. Alors $\mu(|\phi\rangle) < \mu(|\psi\rangle)$. Mais, puisque l'on peut également noter que $|\psi\rangle = G^\dagger |\phi\rangle$, donc un raisonnement similaire fournit l'inégalité $\mu(|\psi\rangle) < \mu(|\phi\rangle)$ et donc l'égalité $\mu(|\phi\rangle) = \mu(|\psi\rangle)$. \square

L'utilisation de l'invariant μ pour étudier la non-localité des états quantiques présente deux avantages principaux. L'un d'eux est que l'on peut en déduire un autre invariant $\tilde{\mu}$, inspiré des travaux de H. Yu et al. [92]. Le second est que l'évaluation de μ (et par extension de $\tilde{\mu}$) peut être exécutée sur un ordinateur quantique. Nous abordons cette partie un peu plus tard dans la Section 3.7 de ce chapitre.

3.4.3 Invariant $\tilde{\mu}$

L'invariant $\tilde{\mu}$ dérivé de l'invariant μ se base également sur les polynômes de Mermin. Il a été introduit et utilisé pour classifier des états issus d'hypergraphes quantiques.

Définition 3.4.3 (Invariant $\tilde{\mu}$ [30, 92]). *Pour un état donné $|\psi\rangle \in \mathbb{C}^n$, on peut considérer l'invariant LU suivant :*

$$\tilde{\mu}(|\psi\rangle) = \text{Max}_{\alpha_1, \beta_1, \dots, \gamma'_n} (\langle \psi | M_n(\alpha_1, \beta_1, \dots, \gamma'_n) | \psi \rangle^2 + \langle \psi | M'_n(\alpha_1, \beta_1, \dots, \gamma'_n) | \psi \rangle^2), \quad (3.4.5)$$

où $M_n(\alpha_1, \beta_1, \dots, \gamma'_n)$ et $M'_n(\alpha_1, \beta_1, \dots, \gamma'_n)$ sont des polynômes de Mermin.

Cette quantité s'est avérée utile pour détecter un type spécifique d'intrication faisant un lien intéressant entre la mesure de non-localité et l'intrication globale des systèmes quantiques. En effet, les conditions suivantes concernant l'intrication d'un état $|\psi\rangle$ à 3 qubits, peuvent être obtenues à partir d'un résultat plus général [30] :

- si $|\psi\rangle$ est 2-intriqué (produit d'une paire 2-intriquée avec un qubit simple), alors $\tilde{\mu} \leq 2$;
- si $|\psi\rangle$ est 3-intriqué (véritable état intriqué à trois qubits), alors, $\tilde{\mu}(|\psi\rangle) \leq 4$.

Celles qui s'appliquent dans le cas d'un état à 4 qubits sont :

- si $|\psi\rangle$ est 2-intriqué (produit de deux qubits 2-intriqués ou produit d'une paire 2-intriquée avec deux qubits simples), alors, $\tilde{\mu}(|\psi\rangle) \leq 2$;
- si $|\psi\rangle$ est 3-intriqué (produit d'un véritable état intriqué à trois qubits avec un seul qubit), alors, $\tilde{\mu}(|\psi\rangle) \leq 4$;
- si $|\psi\rangle$ est 4-intriqué (véritable état intriqué à quatre qubits), alors, $\tilde{\mu}(|\psi\rangle) \leq 8$.

Ces conditions sont exploitées dans le Chapitre 4 pour classer les états hypergraphiques à 4 qubits.

Exemple 3.4.3. *Valeurs de $\tilde{\mu}$ pour quelques états :*

$$\begin{aligned} \tilde{\mu}(|GHZ\rangle_3) &= 4 \\ \tilde{\mu}(|W\rangle_3) &= 2.37. \end{aligned}$$

Les deux derniers invariants que nous avons étudiés sont issus des matrices de coefficients. Nous commençons par définir ces dernières dans la section qui suit.

3.5 Matrices de coefficients et E_{moy}

Sur la base des matrices de coefficients, Chao Zhao et al. ont proposé une mesure d'intrication multipartite [96]. Nous l'appellerons invariant d'intrication moyenne noté E_{moy} . Cet invariant a notamment été utilisé pour caractériser l'intrication de l'algorithme de factorisation de Shor [59] et dans celle des algorithmes d'estimation de phase et de comptage [80].

3.5.1 Définitions préalables

Afin de procéder à la définition de l'invariant, il nous faut d'abord revenir sur la définition des concepts de matricisation ou aplatissage (*flattening* en anglais) et de matrices de coefficients.

Définition 3.5.1 (Aplatissement [96]). Soit $|\psi\rangle_{1..N} = \sum_{i_1, \dots, i_N=0}^{(d_1-1), \dots, (d_N-1)} c_{i_1, \dots, i_N} |i_1, \dots, i_N\rangle$ un état pur à N qudits. Ici, d_t est la dimension de chaque sous-système de l'espace d'états de Hilbert \mathcal{H} , $|i_1, \dots, i_N\rangle$ sont les états de la base standard et c_{i_1, \dots, i_N} sont des coefficients. $|\psi\rangle$ est un tenseur de \mathcal{H} . Considérons une partition en deux ensembles d'indices i_1, \dots, i_N , par exemple (i_1, \dots, i_l) et (i_{l+1}, \dots, i_N) . Pour cette partition $l \times N - l$ particulière, nous obtenons la matrice de coefficients suivante au format $(d_1 d_2 \dots d_l) \times (d_{l+1} \dots d_N)$:

$$C(|\psi\rangle)_{(i_1, \dots, i_l)} = \begin{pmatrix} \underbrace{c_{0, \dots, 0, 0, \dots, 0}}_{l \quad N-l} & \cdots & \underbrace{c_{0, \dots, 0, d-1, \dots, d_N-1}}_{l \quad N-l} \\ \underbrace{c_{0, \dots, 1, 0, \dots, 0}}_{l \quad N-l} & \cdots & \underbrace{c_{0, \dots, 1, d-1, \dots, d_N-1}}_{l \quad N-l} \\ \vdots & \vdots & \vdots \\ \underbrace{c_{d_1-1, \dots, d_l-1, 0, \dots, 0}}_{l \quad N-l} & \cdots & \underbrace{c_{d_1-1, \dots, d_l-1, d_l-1, \dots, d_N-1}}_{l \quad N-l} \end{pmatrix}, \quad (3.5.1)$$

où les bits de 1 à l et $l+1$ à N sont respectivement appelés bits de ligne et bits de colonne. On note qu'il suffit de définir les bits de ligne (ou de colonne) pour spécifier la partition. La matrice $C(|\psi\rangle)_{(i_1, \dots, i_l, (i_{l+1}))}$ est le résultat de l'aplatissement du tenseur $|\psi\rangle$ selon la partition (i_1, \dots, i_l) (bits de ligne) et (i_{l+1}, \dots, i_N) (bits de colonne). Les matrices obtenues de cette opération sont appelées **matrices de coefficients**.

On dira que la matrice $C(|\psi\rangle)_{(i_1, \dots, i_l)}$ de dimension $2^l \times 2^{n-l}$ est un l -aplatissement du tenseur $|\psi\rangle$. Sa transposée correspond à un $n - l$ aplatissement. Le nombre total de matrices de coefficients après aplatissement d'un tenseur est $2^n - 2$. Cependant, ne seront considérées que les $2^{n-1} - 1$ matrices indépendantes, par symétrie [96]. En effet, du fait de la transposition, il suffit de considérer $1 \leq l \leq \frac{n}{2}$. L'algorithme utilisé pour effectuer l'aplatissement de matrices est déroulé à l'Annexe .1.2.

Exemple 3.5.1. Soit le tenseur $|W\rangle_4 = \frac{1}{2}(|0001\rangle + |0010\rangle + |0100\rangle + |1000\rangle)$. Pour cet état, il est possible de faire du 2-aplatissement maximum pour un total de 7 matrices. On obtient les matrices suivantes par aplatissement :

$$C(|W\rangle_4)_{11} = C(|W\rangle_4)_{22} = C(|W\rangle_4)_{33} = C(|W\rangle_4)_{44} = \begin{pmatrix} 0 & 0.5 & 0.5 & 0 & 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$C(|W\rangle_4)_{12} = C(|W\rangle_4)_{13} = C(|W\rangle_4)_{14} = \begin{pmatrix} 0 & 0.5 & 0.5 & 0 \\ 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

3.5.2 Invariant d'intrication moyen E_{moy}

Considérons la quantité :

$$\kappa_{(i_1, \dots, i_l)}(|\psi\rangle) = \sqrt{\sum_{i < j} 4(|x_i|^2 |x_j|^2 - |(x_i \cdot x_j)|^2)}, \quad (3.5.2)$$

où x_i respectivement (x_j) représente le i -ème (respectivement le j -ième) vecteur ligne de la matrice de coefficients $C(|\psi\rangle)_{(i_1, \dots, i_l)}$ obtenue par aplatissement. L'opération $(x_i \cdot x_j)$ est le produit scalaire entre les vecteurs x_i et x_j . Si $|\psi\rangle$ est séparable, alors $\kappa_{(q_1, \dots, q_l)}(|\psi\rangle) = 0$.

Définition 3.5.2 (Invariant E_{moy} [96]). *La mesure moyenne de l'intrication de $|\psi\rangle$ est donnée par :*

$$E_{moy}(|\psi\rangle) = \sqrt{\frac{1}{M} \sum_{l=1}^{N-1} \sum_{(q_1, \dots, q_l)} \kappa_{(q_1, \dots, q_l)}(|\psi\rangle)^2}, \quad (3.5.3)$$

où $M = 2^{n-1} - 1$ est le nombre de l -aplatissements de $|\psi\rangle$.

Les auteurs dans [96] démontrent que E_{moy} est un invariant LU. $E_{moy} = 0$ signifie que l'état $|\psi\rangle$ est séparable.

Exemple 3.5.2. *Quelques valeurs de E_{moy} :*

$$\begin{aligned} E_{moy}(|00\rangle) &= 0 \\ E_{moy}(|GHZ\rangle_3) &= 1 \\ E_{moy}(|W\rangle_3) &= \frac{2\sqrt{2}}{3}. \end{aligned}$$

La définition du dernier invariant dans la section ultérieure résulte également des matrices de coefficients.

3.6 Multirang

La définition du multirang se base sur la définition du rang de tenseur qui généralise celle du rang des matrices. Le rang et par extension, le multirang, peuvent servir de mesure de l'intrication et sont pertinents à étudier. Nous commençons par définir les tenseurs de rang 1.

Définition 3.6.1 (Tenseur de rang 1 [49, 57]). *Soient $|\psi\rangle_1 \in \mathcal{H}_1, \dots, |\psi\rangle_n \in \mathcal{H}_n$ des vecteurs d'espaces de Hilbert respectifs. Un tenseur $|\psi\rangle \in \mathcal{H}_1 \otimes \dots \otimes \mathcal{H}_n$ est dit de rang 1 si et seulement si, il peut s'écrire sous la forme $|\psi\rangle = |\psi\rangle_1 \otimes \dots \otimes |\psi\rangle_n$.*

On définit alors le rang d'un tenseur de la manière suivante.

Définition 3.6.2 (Rang d'un tenseur [16, 57]). *Soit un tenseur $|\psi\rangle \in \mathcal{H}_1 \otimes \dots \otimes \mathcal{H}_n$. On définit le rang du tenseur $|\psi\rangle$ noté $rg(|\psi\rangle) = r$ l'entier naturel minimum tel que $|\psi\rangle = \sum_{i=1}^r |\Phi\rangle_i$, où $|\Phi\rangle_i \in \mathcal{H}$ est un tenseur de rang 1.*

Trouver le rang d'un tenseur $|\psi\rangle$ revient à déterminer le nombre minimal de tenseurs de rang 1 qu'il faut additionner pour retrouver le tenseur originel.

Exemple 3.6.1. *Rang de quelques tenseurs :*

$$\begin{aligned} rg(|GHZ\rangle_n) &= 2 \\ rg\left(\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)\right) &= 1 \end{aligned}$$

Définition 3.6.3 (Multirang [85]). *Soit $|\psi\rangle$ un tenseur de l'espace de Hilbert \mathcal{H}_n . Le multirang de $|\psi\rangle$ est le vecteur de nombres entiers dont les entrées sont égales au rang des matrices de coefficients des l -aplatissements du tenseur avec $k \in \mathbb{N}$ et $1 \leq l \leq \frac{n}{2}$.*

Exemple 3.6.2. Soit l'état : $|W\rangle_4 = \frac{1}{2}(|0001\rangle + |0010\rangle + |0100\rangle + |1000\rangle)$. Sur cet état peuvent être effectués les aplatissements suivants :

- du 1-aplatissage sur les qubits 1, 2, 3 et 4,
- du 2-aplatissage sur les qubits 12, 13, 14.

Le multirang de $|W\rangle_4$ est donné par le rang des matrices obtenus après les aplatissements vus à l'Exemple 3.5.1. Il est égal à :

$$\text{multirg}(|W\rangle_4) = [2, 2, 2, 2, 2, 2].$$

Le rang et le multirang sont des invariants de nature algébrique [33].

Les cinq invariants que nous avons présentés sont utiles à la détection et/ou à la mesure de l'intrication quantique. Cependant, trois d'entre eux, MGI , μ et $\tilde{\mu}$ sont particuliers car ils ont une interprétation physique claire. En effet, en ce qui concerne la MGI , on mesure la distance à l'état séparable le plus proche tandis que pour μ et $\tilde{\mu}$, on mesure l'état par rapport à différents observables (les monômes du polynôme de Mermin) et on combine les résultats de ces mesures pour évaluer l'opérateur de Mermin. Dans les deux cas, l'opération de base nécessaire au calcul de l'invariant consiste à effectuer une mesure (MGI) ou plusieurs (μ et $\tilde{\mu}$) dans des bases différentes (dans une base impliquant l'état le plus proche pour la MGI et dans la base de l'observable définie par un monôme donné pour μ et $\tilde{\mu}$). C'est la raison pour laquelle les trois invariants peuvent être mesurés sur un ordinateur quantique comme cela a été montré dans [4, 37, 78]. La prochaine section est consacrée à l'implémentation des invariants MGI et μ sur une machine quantique.

3.7 Implémentations sur un ordinateur quantique

Les invariants MGI et μ présentent un avantage non négligeable, celui d'être implémentables sur un simulateur ou une machine quantique. Lorsque les paramètres de l'état le plus proche (dans le cas du calcul de l'invariant MGI) ou ceux des polynômes de Mermin (dans le cas du calcul de l'invariant μ), sont connus, on peut facilement calculer ces invariants en effectuant simplement des mesures dans les bases correspondantes. Cependant, lorsque ces paramètres sont inconnus, une recherche (aléatoire, dans notre cas) est nécessaire pour calculer les invariants. La Figure 3.7.1 représente le schéma décrit. Dans cette section, nous expliquons comment nous avons procédé pour calculer ces invariants sur l'ordinateur quantique de IBM Quantum Experience.

3.7.1 Implémentation de MGI

La mesure géométrique de l'intrication (MGI) d'un état $|\psi\rangle$ a été définie plus tôt (3.3.2) comme :

$$MGI(|\psi\rangle) = 1 - \text{Max}_{|\phi\rangle \in \text{Sep}} |\langle \psi | \phi \rangle|^2. \quad (3.7.1)$$

Cet invariant peut être calculé sur un simulateur ou sur une machine quantique. La question est de savoir comment calculer la valeur $\text{Max}_{|\phi\rangle \in \text{Sep}} |\langle \psi | \phi \rangle|^2$ afin de calculer la MGI . L'état $|\phi\rangle$ étant séparable, on peut l'écrire sous la forme :

$$|\phi\rangle = \left(\cos \frac{\theta_1}{2} |0\rangle + e^{i\lambda_1} \sin \frac{\theta_1}{2} |1\rangle \right) \otimes \cdots \otimes \left(\cos \frac{\theta_n}{2} |0\rangle + e^{i\lambda_n} \sin \frac{\theta_n}{2} |1\rangle \right) \quad (3.7.2)$$

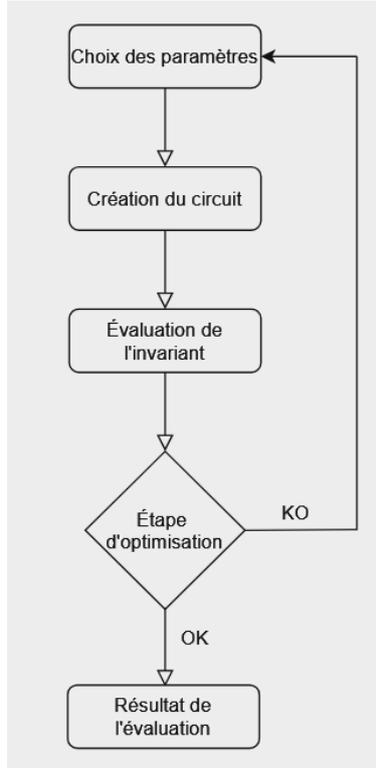


FIGURE 3.7.1 – Représentation d’un circuit de recherche aléatoire et d’un optimiseur pour implémentation sur un ordinateur quantique.

où les angles θ_i et λ_i proviennent des amplitudes α et β de l’état $|\phi\rangle$ tels que $\alpha = \cos \frac{\theta_i}{2}$ et $\beta = e^{i\lambda_i} \sin \frac{\theta_i}{2}$. L’état séparable $|\phi\rangle$ de l’équation 3.7.2 peut se réécrire $|\phi\rangle = U_1 \otimes \dots \otimes U_n |00\dots 0\rangle$ où U_i sont des matrices unitaires de dimension 2×2 correspondant aux rotations appliquées à $|00\dots 0\rangle$ pour obtenir $|\phi\rangle$. On en déduit :

$$|\langle\phi|\psi\rangle|^2 = \left| \langle 00\dots 0 | U_1^\dagger \otimes \dots \otimes U_n^\dagger |\psi\rangle \right|^2. \quad (3.7.3)$$

Cette quantité, qui correspond à la fréquence de l’état $|00\dots 0\rangle$ est obtenue. En résumé, les étapes pour calculer la MGI sur un simulateur ou une machine quantique sont :

- placer l’état $|\psi\rangle$ sur un circuit quantique ;
- placer des portes de rotation U_i^\dagger ;
- faire des mesures pour calculer la quantité $|\langle\psi|\phi\rangle|^2$.

Une étape d’optimisation est rajoutée pour trouver les angles de la matrice U_i^\dagger afin de trouver l’état séparable qui maximise la valeur $|\langle\psi|\phi\rangle|^2$. L’algorithme de recherche utilisé, parmi ceux présentés Section 3.2 est celui de marche aléatoire, car il converge plus rapidement que l’algorithme de recherche tabou. Cela s’explique par la rallonge de temps due au parcours de la liste taboue de chaque coefficient employé lors de l’optimisation de l’invariant.

L’algorithme utilisé pour implémenter ces calculs est fourni à l’Annexe .2.2.

3.7.2 Implémentation de μ

Calculer l’invariant μ d’un état $|\psi\rangle$ consiste à évaluer la quantité de la définition 3.4.1 :

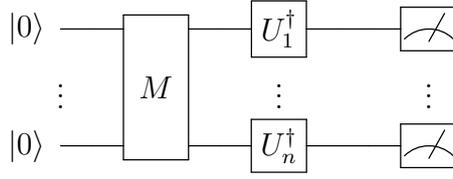


FIGURE 3.7.2 – Circuit de l’implémentation du calcul de la MGI sur un ordinateur quantique pour $|\phi\rangle = M |00 \dots 0\rangle$.

$$\mu(|\psi\rangle) = \text{Max}_{\alpha_1, \beta_1, \dots, \gamma'_n} \langle \psi | M_n(\alpha_1, \beta_1, \dots, \gamma'_n) | \psi \rangle .$$

Nous souhaitons faire ce calcul sur un simulateur quantique ou sur une machine quantique. Pour cela, il faut d’abord rechercher numériquement les valeurs des coefficients qui optimisent μ , puis, vérifier si cette valeur de μ peut être retrouvée en implémentant le circuit qui la calcule sur une machine quantique. Dans le but d’effectuer cette implémentation, plusieurs étapes sont réalisées.

- D’abord, calculer la quantité $\mu(|\phi\rangle)$, c’est mesurer cet état dans la base des observables qui composent M_n . Cependant, cela est directement impossible car l’unique opération de mesure existante est la mesure dans la base standard. Pour résoudre ce problème, une astuce consiste à effectuer un changement de base avant de faire la mesure. Cela est possible à l’aide de la porte U dont l’expression est :

$$U(\theta_j, \phi_j, \lambda_j) = \begin{pmatrix} \cos(\frac{\theta_j}{2}) & -e^{i\lambda_j} \sin(\frac{\theta_j}{2}) \\ e^{i\phi_j} \sin(\frac{\theta_j}{2}) & e^{i(\phi_j + \lambda_j)} \cos(\frac{\theta_j}{2}) \end{pmatrix} \quad (3.7.4)$$

En fixant :

- $\theta_j = \arccos(\gamma_j)$,
- $\phi_j = 0$ si $\sin(\theta_j) = 0$, sinon $\phi_j = \arccos\left(\frac{\alpha_j}{\sin(\theta_j)}\right)$ et si $\frac{\beta_j}{\sin(\theta_j)} \leq 0$, alors $\phi_j = -\phi_j$,
- $\lambda_j = -\phi - \pi$,

pour calculer une matrice $U(\theta_j, \phi_j, \lambda_j)$ en fonction des coefficients $\alpha_j, \beta_j, \gamma_j$ d’un observable a_j qui optimise $\mu(|\phi\rangle)$, on peut placer cette porte dans un circuit puis effectuer une mesure classique dans la base standard. Cela équivaut à faire une mesure dans la base a_j .

- Ensuite, construire le circuit comportant en entrée l’état à évaluer, ajouter les portes U correspondantes à chaque monôme de Mermin et effectuer les mesures sur un simulateur ou sur une machine quantique en précisant le nombre de fois que chaque circuit sera exécuté (1024, par défaut).
- Enfin, le polynôme de Mermin est reconstitué et les derniers calculs effectués pour obtenir le résultat de l’évaluation par μ .
- Pour finir, dans le cas d’un calcul effectué directement sur machine quantique, sans entrer les coefficients d’optimisation théoriques, un optimiseur est ajouté pour trouver les paramètres des matrices U .

L’algorithme utilisé pour implémenter ces calculs est fourni à l’Annexe .2.1.

Exemple 3.7.1. Calcul de μ pour l'état $|GHZ\rangle_3$ sur machine quantique.

On rappelle que $M_3 = \frac{1}{2}(a_1a_2a'_3 + a_1a'_2a_3 + a'_1a_2a_3 - a'_1a'_2a'_3)$ avec $a_i = \alpha_iX + \beta_iY + \gamma_iZ$. Les coefficients $\alpha_i, \beta_i, \gamma_i \in \mathbb{R}$ qui maximisent $\mu(|GHZ\rangle_3)$ sont calculés numériquement : On calcule alors les paramètres des matrices U qui permettent d'obtenir les circuits de la

a_i	α_i	β_i	γ_i
a_1	0,495	-0,868	-1.0765×10^{-06}
a'_1	0,868	0,495	1.422×10^{-05}
a_2	0,627	-0,779	1.722×10^{-06}
a'_2	0,779	0,627	1.132×10^{-05}
a_3	-0,931	-0,366	-1.962×10^{-06}
a'_3	0,366	-0,931	5.435×10^{-06}

TABLE 3.1 – Tableau des coefficients numériques qui maximisent $\mu(|GHZ\rangle_3)$.

Figure 3.7.3.

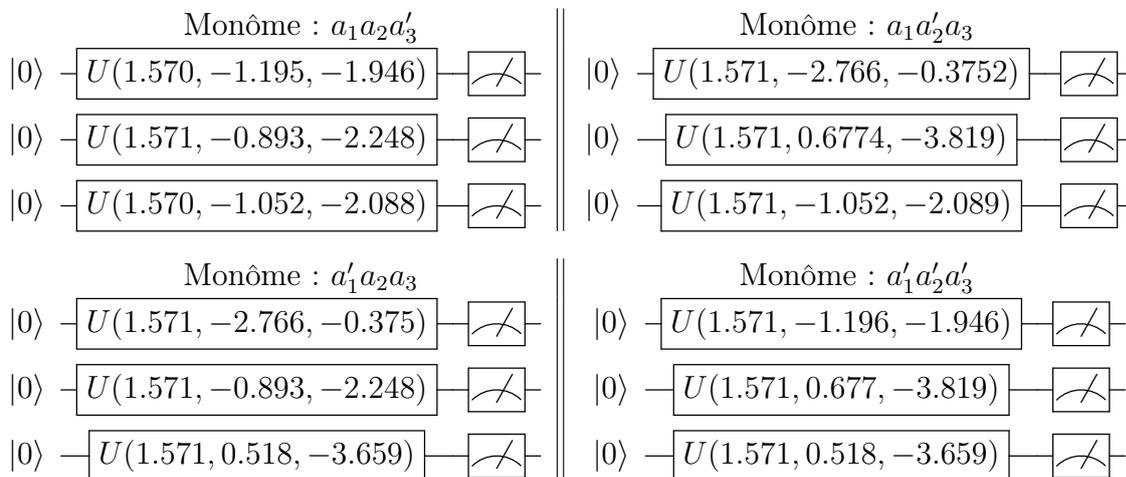


FIGURE 3.7.3 – Circuits comportant le placement des portes U correspondantes aux observables dont les paramètres ont été calculés numériquement.

Pour obtenir $\mu(|GHZ\rangle)$, on calcule les fréquences des matrices U des circuits. Au final, on obtient bien les mêmes valeurs calculées numériquement lorsque les circuits sont exécutés sur un simulateur quantique.

3.8 Conclusion

L'exploration des outils de mesure de l'intrication est fondamentale pour étudier ses apports à la performance des algorithmes quantiques. Ce chapitre a examiné cinq (05) outils de mesure qui sont : la mesure géométrique de l'intrication MGI , la valeur moyenne μ et sa variante $\tilde{\mu}$ basés sur les polynômes de Mermin, la mesure de l'intrication moyenne E_{moy} et le multirang basés sur les matrices de coefficients. Dans le prochain chapitre nous plongerons dans l'étude des états d'hypergraphes dont nous étudierons l'intrication à l'aide des outils de mesure présentés.

Chapitre 4

ETATS HYPERGRAPHIQUES

Contenu du Chapitre

4.1	Contexte	37
4.2	Etats de graphes et d'hypergraphes	37
4.3	Intrication dans les états hypergraphiques	39
4.4	Implémentation sur un ordinateur quantique	42
4.5	Conclusion	44

4.1 Contexte

Les états graphiques et hypergraphiques sont des états quantiques à plusieurs qubits associés à des graphes et hypergraphes mathématiques. Ils sont une manière simple et visuelle de représenter certains états quantiques. Les états graphiques sont des cas particuliers des états de superposition de tous les états de base avec des amplitudes réelles et des probabilités égales (en anglais Real Equally Weighted -REW- states). Ces états sont généralement retrouvés dans des algorithmes quantiques tels que ceux de Deutsch-Jozsa et de Grover. Les états hypergraphiques, quant à eux, sont une généralisation des états graphiques qui couvrent tous les REW states [72]. Les états graphiques et d'hypergraphes peuvent être utilisés pour concevoir des codes correcteurs d'erreurs quantiques [7, 73]. D'une manière générale, l'intrication et la non-localité des états hypergraphiques suscitent de l'intérêt de la part de la communauté scientifique [32, 34, 69, 89]. Dans ce chapitre, nous nous intéressons d'abord à la détection et à la classification de l'intrication des états hypergraphiques à quatre (04) qubits dont au moins trois (03) connectés. Ensuite, nous vérifions l'évolution de l'intrication dans les états hypergraphiques dits k-uniformes. Nous avons employé les outils de mesure de l'intrication présentés au Chapitre 3. Enfin, nous réalisons l'implémentation des états hypergraphiques et l'évaluation de l'intrication sur un ordinateur quantique. Les conclusions de cette étude ont fait l'objet d'un article [4] dans le journal *International Journal of Quantum Information*.

4.2 Etats de graphes et d'hypergraphes

Avant de définir les états hypergraphiques quantiques, nous définissons les états graphiques quantiques.

Définition 4.2.1 (Graphes et états graphiques quantiques [72]). *Soit le graphe mathématique $G = \{V, E\}$ où V est un ensemble de n sommets avec $n \in \mathbb{N}^*$ et E un ensemble d'arrêtes. On peut définir le graphe quantique associé en faisant correspondre chaque sommet au qubit $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ et en réalisant une succession d'opérations de la porte*

$$C_2Z_{i_1 i_2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \text{ successives entre deux qubits connectés } \{i_1, i_2\} \in E. \text{ Cette}$$

porte introduit un signe moins (-) devant l'état $|11\rangle_{i_1 i_2}$. L'état quantique associé au graphe G s'écrit :

$$|G\rangle = \prod_{i_1 i_2 \in E} C_2Z_{i_1 i_2} |+\rangle^{\otimes n}. \quad (4.2.1)$$

L'ordre des qubits i_1, i_2 n'a pas d'importance dans l'application de la porte C_2Z . Des exemples de graphes quantiques sont donnés à la Figure 4.2.1. Les circuits de la Figure 4.2.2 sont les circuits résultants des graphes de l'exemple 4.2.1.

On généralise la définition des états graphiques aux états hypergraphiques.

Définition 4.2.2 (Hypergraphes et états hypergraphiques quantiques [72]). *Soit l'hypergraphe mathématique $H = \{V, E\}$ où V est un ensemble de n sommets avec $n \in \mathbb{N}^*$ et E un ensemble d'hyperarrêtes. Une hyperarrête est un ensemble d'au moins 2 sommets. On*

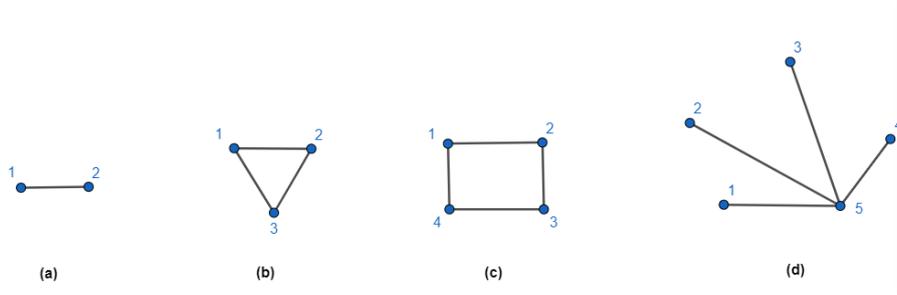


FIGURE 4.2.1 – Quelques exemples d'états graphiques quantiques à 2, 3, 4 et 5 qubits. (a) État à deux qubits connectés ayant une seule arrête : $|G\rangle_{(a)} = \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle + |10\rangle - |11\rangle)$. (b) et (c) États communément nommés RC_3 (respectivement RC_4) pour *Ring Cluster* en anglais, à 3 (resp. 4) qubits. On écrit $|G\rangle_{(b)} = \frac{1}{2\sqrt{2}}(|000\rangle + |001\rangle + |010\rangle - |011\rangle + |100\rangle - |101\rangle - |110\rangle - |111\rangle)$. (d) État appelé ST_5 pour *Star* en anglais.

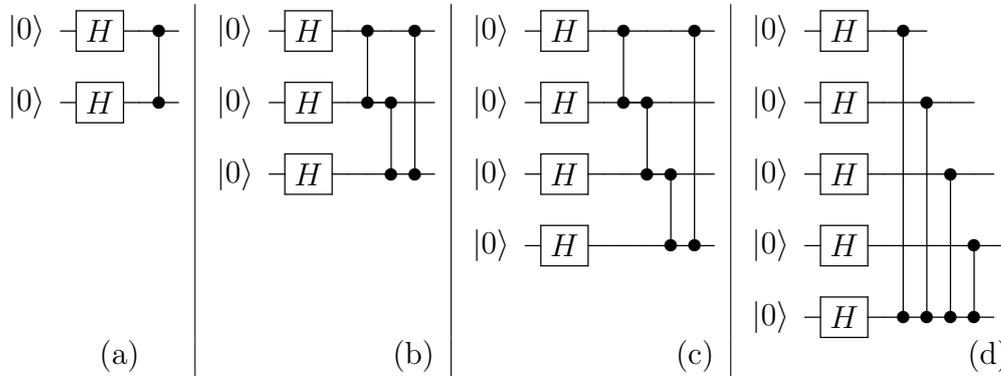


FIGURE 4.2.2 – Circuits équivalents aux graphes de la Figure 4.2.1. En premier lieu, chaque qubit est initialisé à l'état $|+\rangle$ grâce à l'application d'une porte Hadamard H . Puis, des portes C_iZ sont ajoutées entre deux qubits connectés.

peut définir l'état hypergraphique quantique suivant :

$$|H\rangle = \prod_{i_1 i_2 \dots i_k \in E} C_k Z_{i_1 i_2 \dots i_k} |+\rangle^{\otimes n}, \quad (4.2.2)$$

où $i_1 i_2 \dots i_k$ représentent k qubits connectés et $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. La porte $C_k Z_{i_1 i_2 \dots i_k}$ (appelée porte Z multi-contrôlée) introduit un signe moins ($-$) devant l'état $|11\dots 1\rangle_{i_1 i_2 \dots i_k}$. L'action de la porte $C_k - Z$ est invariante s'il y a une permutation des n qubits de la base de calcul [72].

Des exemples d'états hypergraphiques quantiques sont donnés à la Figure 4.2.3. Les circuits de la Figure 4.2.4 sont les circuits résultants des hypergraphes de l'exemple 4.2.3.

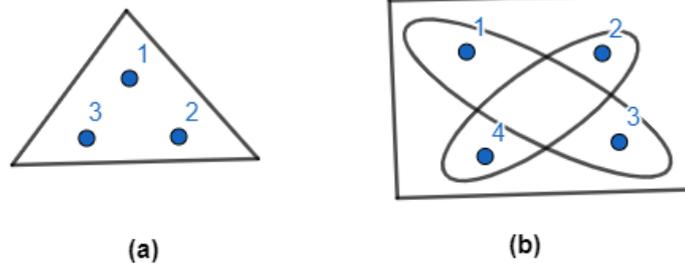


FIGURE 4.2.3 – Quelques exemples d'états hypergraphiques quantiques à 3 et 4 qubits. (a) Hypergraphe à 3 qubits avec une seule hyperarrête qui s'écrit : $|H\rangle_{(a)} = \frac{1}{2\sqrt{2}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle - |111\rangle)$. (b) Hypergraphe à 4 qubits entièrement connecté : $|H\rangle_{(b)} = \frac{1}{4}(|0000\rangle + |0001\rangle + |0010\rangle + |0011\rangle + |0100\rangle - |0101\rangle + |0110\rangle - |0111\rangle + |1000\rangle + |1001\rangle - |1010\rangle - |1011\rangle - |1100\rangle - |1101\rangle - |1110\rangle - |1111\rangle)$.

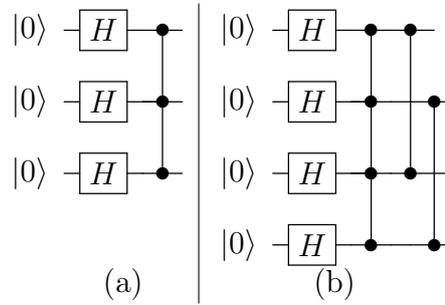


FIGURE 4.2.4 – Circuits équivalents aux hypergraphes de la Figure 4.2.3. En premier lieu, chaque qubit est initialisé à l'état $|+\rangle$ grâce à l'application d'une porte Hadamard H. Puis, des portes C_iZ sont ajoutées entre plusieurs qubits connectés.

4.3 Intrication dans les états hypergraphiques

Dans cette section, nous étudions d'abord l'intrication des états hypergraphiques à 4 qubits de la littérature puis dans des états hypergraphiques dits "k-uniformes". L'algorithme utilisé pour implémenter les hypergraphes sur un circuit quantique est fourni à l'Annexe .2.3.

INTRICATION DANS LES ÉTATS HYPERGRAPHIQUES DE LA LITTÉRATURE Les états hypergraphiques ont été le sujet de plusieurs études dans la communauté scientifique. Ils ont notamment été classés selon des opérations unitaires locales (notées LU) et des transformations opérations locales stochastiques et communication classique (notées SLOCC) [17, 40, 91, 94]. Notamment, basé sur les travaux de X. Chen et al. [19], O. Gühne et al. ont proposé une classification d'hypergraphes à 4 qubits comprenant au moins une hyperarrête de taille 3 [42].

Les hypergraphes 1 à 27 de la Figure 4.3.1 sont ceux de la littérature [19, 42]. Les graphes S_4 (*Star* en anglais) et LC_4 (*Local Cluster*), placés en dernière position, représentent les deux classes d'états graphiques à quatre qubits ajoutées à l'étude. L'état S_4 constitue une généralisation des états $|GHZ\rangle$ [32]. Pour chacune des familles d'hyper-

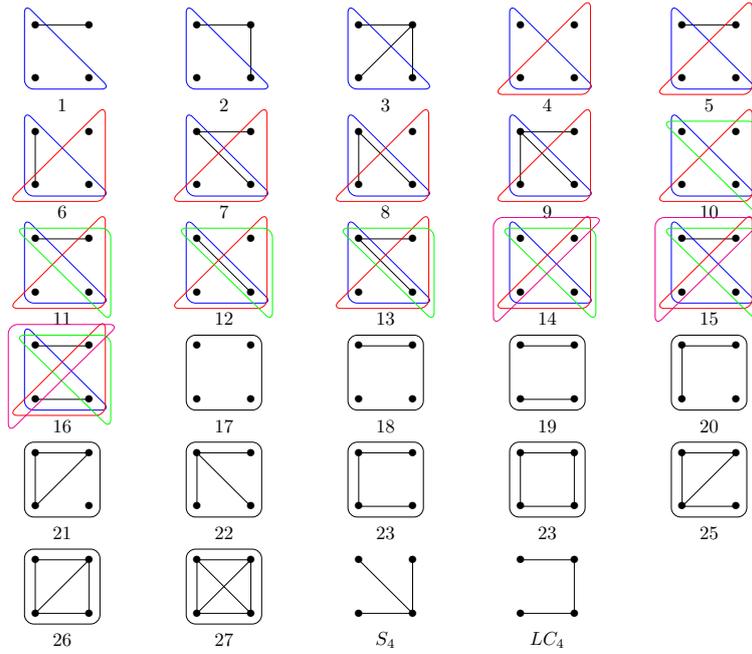


FIGURE 4.3.1 – Classification LU des hypergraphes à quatre qubits connectés [42].

graphes de la classification de la Figure 4.3.1, nous calculons une valeur de cinq invariants (μ , $\tilde{\mu}$, E_{moy} , multirang et MGI) étudiés au Chapitre 3. Dans le Tableau 4.1, nous avons compilé les résultats des calculs par invariant et par famille.

Le but de la démarche est d'abord de mesurer le degré d'intrication des familles d'états puis de vérifier s'il est possible d'obtenir pour chaque famille une valeur unique par invariant qui diffère d'une famille à l'autre. Ainsi, cela permettrait de différencier les familles les unes des autres du point de vue de l'intrication.

Dans le Tableau 4.1 qui contient les résultats de l'évaluation de chaque famille d'états hypergraphiques par chaque invariant, on constate que :

1. Pour toutes les familles d'états, $\mu > 1$, $E_{moy} > 0$, $MGI > 0$. Ils sont donc tous intriqués.
2. Concrètement, le nombre de familles distinguées par invariant est :
 - $\tilde{\mu} \rightarrow 29$ familles ;
 - $\mu \rightarrow 28$ familles ;
 - $MGI \rightarrow 22$ familles ;
 - $E_{moy} \rightarrow 15$ familles ;
 - $multirg \rightarrow 6$ familles.
3. L'invariant μ et son dérivé $\tilde{\mu}$ offrent une distinction plus fine des familles étudiées.
4. Pris individuellement, trois invariants ne permettent pas de distinguer toutes les familles d'états hypergraphiques. Cependant, l'ensemble des invariants permet de faire cette distinction.
5. Le calcul de $\tilde{\mu}$ détecte la véritable 4-intrication (car $\tilde{\mu} \leq 8$ cf Section 3.4.3) pour les états des familles suivantes :

$$\{9, 14, 22, 27, S_4\}$$

Famille N°	μ	$\tilde{\mu}$	E_{moy}	multirg	MGI
1	1.811	3.28	0.982	[3, 3, 2]	0.5
2	1.268	1.616	1.035	[3, 3, 3]	0.656
3	1.767	3.125	1.052	[3, 3, 3]	0.652
4	1.931	3.732	0.906	[3, 3, 2]	0.345
5	1.830	3.35	0.982	[4, 4, 2]	0.573
6	1.224	1.5	0.963	[3, 3, 3]	0.5
7	1.499	2.285	1.035	[4, 4, 3]	0.625
8	1.506	2.321	0.982	[3, 3, 3]	0.635
9	2.060	4.246	1.052	[4, 4, 3]	0.635
10	1.633	2.722	0.945	[3, 3, 3]	0.5
11	1.892	3.584	1.017	[4, 4, 3]	0.598
12	1.350	1.832	0.925	[3, 3, 3]	0.375
13	1.371	1.885	1.0	[4, 4, 3]	0.625
14	2.423	5.872	1.0	[3, 3, 3]	0.571
15	1.554	2.452	1.0	[4, 4, 3]	0.587
16	1.554	2.531	0.925	[3, 3, 3]	0.437
17	1.433	2.071	0.701	[2, 2, 2]	0.19
18	1.319	1.743	0.881	[3, 3, 2]	0.432
19	1.842	3.399	0.995	[4, 4, 2]	0.643
20	1.722	2.968	0.959	[3, 3, 3]	0.462
21	1.701	2.896	0.959	[3, 3, 3]	0.55
22	2.317	5.371	0.995	[3, 3, 3]	0.461
23	1.591	1.666	1.031	[4, 4, 3]	0.652
24	1.713	2.934	1.031	[4, 3, 4]	0.556
25	1.386	1.921	1.031	[4, 4, 3]	0.66
26	1.495	2.243	1.031	[4, 3, 4]	0.538
27	2.215	4.913	0.995	[3, 3, 3]	0.556
28	2.828	8	1.0	[2, 2, 2]	0.5
29	1.412	2	1.069	[4, 4, 2]	0.75

TABLE 4.1 – Tableau regroupant les valeurs calculées à 10^{-3} près, des invariants par familles d'états hypergraphiques quantiques.

et montre que les états des familles :

$$\{1, 3, 4, 5, 7, 8, 11, 15, 16, 17, 19, 20, 21, 24, 26\}$$

sont au moins 3-intriqués.

Dans le cas de l'évaluation des états hypergraphiques de la littérature, les invariants μ et MGI ont été calculés à l'aide de l'algorithme de marche aléatoire qui est celui qui converge le plus rapidement. Dans le tableau 4.2 sont fournis les temps de calcul des invariants à partir de la recherche de marche aléatoire et de celle de la recherche taboue.

Le paragraphe suivant évoque l'étude de l'intrication dans les états hypergraphiques k -uniformes.

INTRICATION DANS LES ÉTATS HYPERGRAPHIQUES k -UNIFORMES Les états hypergraphiques k -uniformes sont des états $H = \{V, E\}$ tels que $\forall e_i \in E, |e_i| = k$. Cela signifie que toute hyperarrête connecte exactement k sommets. La non-localité des états

Hypergraph N°	μ	μ_{taboue}	MGI	MGI_{taboue}
1	49 s	10 min 31 s	0.3 s	1 min
4	22 s	26 min 14 s	0.2 s	7 s
10	12 s	33 min 41 s	0.5 s	23 s
17	13 s	47 min	0.2 s	13 s

TABLE 4.2 – Comparaison du temps de calcul des invariants μ et MGI par algorithme de recherche pour quelques familles d’hypergraphes.

hypergraphiques 3-uniformes et 4-uniformes a été étudiée dans [32] pour des valeurs asymptotiques de n . En se limitant à une petite valeur de n , on peut estimer numériquement μ pour ces états. Les résultats obtenus jusqu’à $n = 9$ des calculs sont produits dans le Tableau 4.3.

	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$
$k = 2$	4	5.65685	8	11.31370	16
$k = 3$	2.45751	2.85947	4.34159	6.24393	8.2368
$k = 4$	2.02319	3.29038	4.51349	4.92526	6.0113
$k = 5$	1.29200	3.20848	5.93197	8.97846	11.6284
$k = 6$	-	1.14326	2.44886	3.69746	5.7151
$k = 7$	-	-	1.00307	3.17162	6.9736
$k = 8$	-	-	-	0.87610	2.4187
$k = 9$	-	-	-	-	0.7430

TABLE 4.3 – Estimation numérique de l’invariant μ pour les états hypergraphiques k -uniformes.

On remarque que les valeurs les plus basses de l’invariant μ sont atteintes lorsque $n = k$ et que, d’une manière générale, plus k est petit et plus μ est élevé.

4.4 Implémentation sur un ordinateur quantique

Comme mentionné précédemment, trois des invariants étudiés peuvent être évalués sur une machine quantique. Il s’agit de μ , $\tilde{\mu}$ et MGI . Nous nous sommes donc proposés de reproduire les états hypergraphiques sur un simulateur et sur une machine quantique dans le but de les évaluer à l’aide de ces invariants. L’objectif est de comparer ces nouveaux résultats aux résultats numériques déjà obtenus. L’évaluation des 29 familles d’hypergraphes sur un simulateur quantique a apporté les mêmes résultats que l’évaluation numérique à une précision de 10^{-2} près. Il en est de même dans le cas des états hypergraphiques k -uniformes. Cependant, lorsqu’il s’agit d’une machine quantique réelle, les résultats sont plus mauvais que sur le simulateur. Cela est dû à la **transpilation** qui est un processus de transformation d’un circuit quantique abstrait en un circuit concret qui peut être exécuté sur un matériel quantique spécifique. Cette étape est nécessaire pour qu’un circuit soit converti en accord avec l’architecture spécifique des machines quantiques. Par exemple, chaque porte de Toffoli est **transpilée** en un circuit impliquant 5 portes contrôle- X et plusieurs rotations. Par conséquent, la version transpilée de l’état hypergraphique à 4 qubits de la Figure 4.4.1 implique 32 portes CX et 18 portes de rotation 4.4.2.

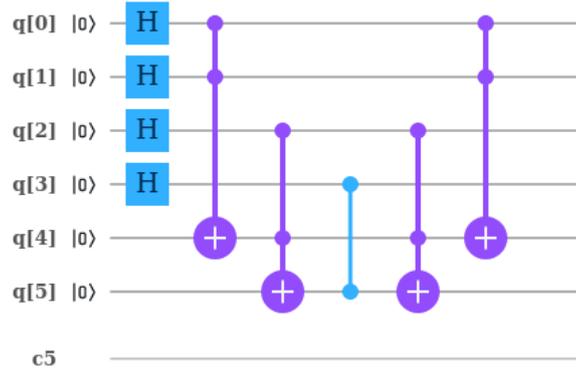


FIGURE 4.4.1 – Circuit générant l’hypergraphe 17 de la classification 4.3.1.

L’hypergraphe 17 de la Figure 4.3.1 est pourvu d’une seule hyperarrête de taille 4. Les quatre premiers qubits correspondent à ceux de l’hypergraphe tandis que les 2 qubits suivants sont des qubits auxiliaires qui aident à la création de l’hyperarrête de taille 4. Le circuit de la Figure 4.4.2 est la version transpilée du circuit 4.4.1 de la classe

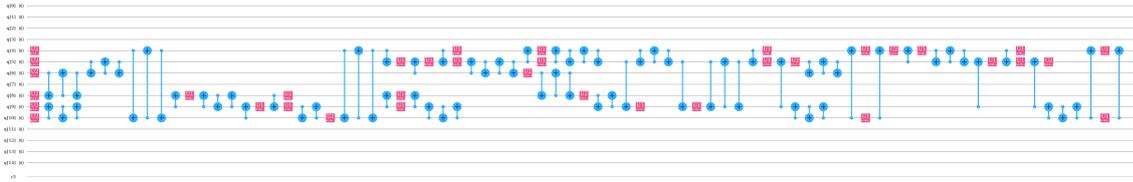


FIGURE 4.4.2 – Version transpilée du circuit représenté sur la Figure 4.4.1 lorsqu’il est implémenté sur la machine quantique **ibmq_16_melbourne**.

d’hypergraphes numéro 17. Sa grande taille, à la suite de la transpilation, explique en partie les mauvais résultats des calculs sur une machine quantique réelle. Néanmoins, l’optimisation des circuits est un problème central en informatique quantique [8]. Le seul exemple de violation du polynôme de Mermin sur une machine quantique que nous avons pu obtenir avec un hypergraphe est celui du cas à 3 qubits (Figure 4.4.3). Dans ce cas, le polynôme de Mermin est donné par :

$$M_3 = \frac{1}{2}(ABC' + AB'C + A'BC - A'B'C')$$

avec

$$\begin{array}{l|l} A = 0,58X + 0,44Y - 0,68Z & A' = 0,37X - 0,83Y - 0,41Z \\ B = -0,58X - 0,44Y + 0,68Z & B' = -0,37X + 0,83Y + 0,41Z \\ C = 0,58X + 0,44Y - 0,68Z & C' = 0,37X - 0,83Y - 0,41Z \end{array}$$

L’évaluation de M_3 sur l’état hypergraphique de la Figure 4.4.3 : $|G\rangle = C_2Z|+\rangle^{\otimes 3}$ donne le résultat suivant :

$$\mu(|G\rangle) = 1,52. \tag{4.4.1}$$

L’évaluation des quatre monômes sur la machine quantique d’IBM produit les résultats suivants :

$$\begin{array}{l|l} \langle A'BC \rangle = 0,52 & \langle AB'C \rangle = 0,62 \\ \langle ABC' \rangle = 0,64 & \langle A'B'C' \rangle = -0,48 \end{array}$$

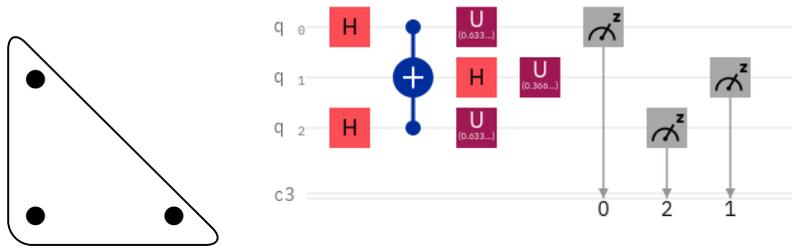


FIGURE 4.4.3 – Exemple d’état hypergraphique et le circuit représentant son évaluation par un monôme du polynôme de Mermin.

Ce qui fournit cette valeur de μ :

$$\mu_{\text{exp}} \approx 1,13. \quad (4.4.2)$$

Comme prévu, l’évaluation expérimentale de μ (équation 4.4.1), n’est pas aussi précise que celle de l’évaluation numérique (équation 4.4.2). Cependant, la valeur expérimentale viole les inégalités de Mermin car $\mu_{\text{exp}} \approx 1,13 > 1$ fournissant, à notre connaissance, un premier exemple de violation expérimentale du réalisme local avec un état hypergraphique sur une machine quantique d’IBM.

4.5 Conclusion

Dans ce chapitre, nous avons étudié, à l’aide de nos invariants, l’intrication d’une classification d’états hypergraphiques de la littérature. De plus, les invariants μ , $\tilde{\mu}$ et MGI fonctionnent bien pour différencier des familles d’hypergraphes à 4 qubits connectés [42]. Particulièrement, les invariants μ et $\tilde{\mu}$ permettent effectivement d’effectuer une classification des dites familles. Nous avons aussi réussi à implémenter les hypergraphes sur une machine quantique. Ils peuvent alors être directement mis sous la forme d’un circuit ou encore fournis sous la forme d’une boîte noire dans le but d’étudier leur intrication. Nos travaux ont été cités par [12] qui utilisent des ordinateurs quantiques pour générer une non-localité multipartite, servant de référence pour évaluer leur performance ; ainsi que par [28] qui étudient entre autres, comme nous, l’intrication des états hypergraphiques. Dans le chapitre ultérieur, nous plongeons dans l’étude de l’intrication dans les algorithmes d’estimation de phase et de comptage quantiques. Plus particulièrement, nous abordons les similitudes entre ces deux algorithmes et mettons en lumière l’existence de l’intrication.

Chapitre 5

INTRICATION DANS LES ALGORITHMES AEPQ ET ACQ

Contenu du Chapitre

5.1	Contexte	46
5.2	Intrication de $ \psi_{\text{AEPQ}}(\phi)\rangle$	46
5.3	Intrication de $ \psi^{\text{ACQ}}(\phi)\rangle$	52
5.4	Conclusion	54

5.1 Contexte

À l'aide de trois outils de détection (μ , E_{moy} , MGI), nous examinons l'intrication dans les étapes précédant l'application de la Transformée de Fourier quantique (pré-TFQ) des algorithmes d'estimation de phase quantique et de comptage quantique (AEPQ - ACQ) comme dans [80]. Ces algorithmes ont été introduits aux sections 2.2.3 et 2.2.4 du Chapitre 2. Spécifiquement, nous nous intéressons à la sensibilité de l'intrication par rapport à la valeur d'entrée (la phase ϕ et le rapport $\frac{M}{N}$ des éléments marqués et du total d'éléments). Dans l'AEPQ, le paramètre ϕ est la phase que l'on veut estimer alors que dans ACQ, c'est l'angle θ qui permet d'estimer $\frac{M}{N}$ sachant que $M = N \sin^2(\theta)$. L'angle θ peut également s'exprimer comme une phase entre 0 et 2π en posant $\phi = \frac{\theta}{2}$. L'étude de l'intrication est réalisée en faisant varier ces angles tels que $0 \leq \{\phi, \theta\} \leq 1$. Nous partons d'observations numériques et déduisons quelques résultats généraux en particulier concernant les classes d'intrication. Plus précisément, lorsque le second registre des deux algorithmes est initialisé dans la superposition non intriquée de deux vecteurs propres (séparables), on prouve que les courbes de l'évolution de l'intrication issues de AEPQ et ACQ sont les mêmes à une homothétie près du paramètre ϕ . Nous démontrons qu'un minimum local est obtenu et correspond à un état $|EPR\rangle$ et enfin nous prouvons que, à une opération SLOCC près, tous les états, à l'exception de quelques valeurs de ϕ , sont équivalents au produit d'un état séparable et d'un état GHZ généralisé. Les conclusions de notre étude ont fait l'objet d'un article [3] dans le journal *Physica Scripta*.

5.2 Intrication de $|\psi_{\text{AEPQ}}(\phi)\rangle$

Le cas le plus simple dans l'AEPQ à considérer est lorsque $U = \begin{pmatrix} 1 & 0 \\ 0 & e^{2i\pi\phi} \end{pmatrix}$ où $|0\rangle$ et $|1\rangle$ sont des vecteurs propres de valeurs propres respectives 1 et $e^{2i\pi\phi}$. Il est clair qu'il n'y aurait pas d'intrication générée par la partie pré-TFQ de l'algorithme si l'on initialise le second registre du circuit à la Figure 2.2.5 soit à $|u\rangle = |0\rangle$ ou à $|u\rangle = |1\rangle$. Cependant, si l'on choisit de superposer les vecteurs propres en mettant le second registre à $|u\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, alors l'état à la fin de la partie pré-TFQ pourrait être intriqué. En effet, si n est le nombre de qubits sur le premier registre, un calcul simple montre que l'état quantique, à la fin de la phase pré-TFQ, sera :

$$|\psi^{\text{AEPQ}}(\phi)\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle |0\rangle + e^{2i\pi x\phi} |x\rangle |1\rangle. \quad (5.2.1)$$

Les évaluations numériques des trois invariants μ , E_{moy} et MGI de l'état 5.2.1 en fonction de ϕ a été obtenue jusqu'à $n = 5$ (Figure 5.2.1). Sur cette figure, la courbe des valeurs de chaque invariant est symétrique par rapport à l'angle $\frac{1}{2}$ pour $n = 3 + 1, 4 + 1, 5 + 1$ qubits. Montrons maintenant que cette proposition est vraie pour tout n .

Proposition 5.2.1. *Pour $U = \begin{pmatrix} 1 & 0 \\ 0 & e^{2i\pi\phi} \end{pmatrix}$ et $\phi \in [0, 1]$ dans l'AEPQ, les courbes obtenues pour les invariants MGI , μ et E_{moy} en fonction de ϕ sont symétriques par rapport à $\phi = \frac{1}{2}$.*

Démonstration. D'après leurs définitions, les trois invariants μ , MGI et E_{moy} sont invariants par conjugué complexe, c'est-à-dire :

$$MGI(\psi) = MGI(\bar{\psi}), \mu(\psi) = \mu(\bar{\psi}), \text{ et } E_{moy}(\psi) = E_{moy}(\bar{\psi}). \quad (5.2.2)$$

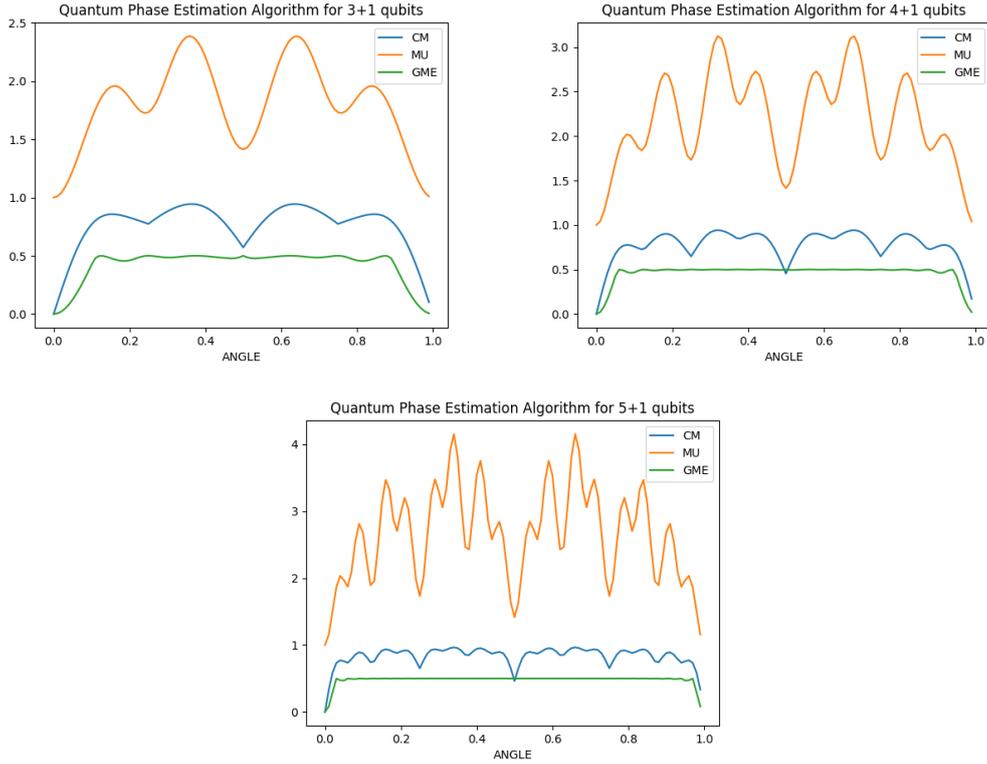


FIGURE 5.2.1 – Évaluations numériques des trois invariants μ , E_{moy} et MGI pour les états quantiques $|\psi_{\text{AEPQ}}(\phi)\rangle$ pour $n \in \{3, 4, 5\}$ qubits sur le premier registre.

Mais on observe aussi que les états générés par la partie pré-TFQ de l'équation AEPQ (5.2.1) sont tels que :

$$\begin{aligned}
|\psi^{\text{AEPQ}}(1 - \phi)\rangle &= \sum_{x=0}^{2^n-1} (|x\rangle |0\rangle + e^{i(2\pi(1-\phi))x} |x\rangle |1\rangle) \\
&= \sum_{x=0}^{2^n-1} (|x\rangle |0\rangle + e^{-i2\pi\phi x} |x\rangle |1\rangle) \\
&= \overline{|\psi^{\text{AEPQ}}(\phi)\rangle}.
\end{aligned} \tag{5.2.3}$$

Ce qui prouve que

$$\begin{aligned}
MGI(|\psi^{\text{AEPQ}}(\phi)\rangle) &= MGI(|\psi^{\text{AEPQ}}(1 - \phi)\rangle), \\
\mu(|\psi^{\text{AEPQ}}(\phi)\rangle) &= \mu(|\psi^{\text{AEPQ}}(1 - \phi)\rangle), \\
E_{moy}(|\psi^{\text{AEPQ}}(\phi)\rangle) &= E_{moy}(|\psi^{\text{AEPQ}}(1 - \phi)\rangle).
\end{aligned} \tag{5.2.4}$$

et explique ainsi la symétrie des courbes de la Figure 5.2.1 d'intrication par rapport à la valeur $\phi = \frac{1}{2}$. \square

Nous observons également à partir de nos expériences numériques, que pour $\phi = \frac{1}{2}$, les valeurs des trois invariants correspondent aux trois mesures d'intrication différentes d'un état $|EPR\rangle = \frac{1}{2}(|00\rangle + |11\rangle)$. En effet, il est bien connu que $MGI(|EPR\rangle) = \frac{1}{2}$ [87] et que

les polynômes de Mermin atteignent la valeur maximale $\sqrt{2}$ pour les états $|EPR\rangle$, comme vu au Chapitre 3. Montrons que l'état $|\psi_{AEPQ}(\frac{1}{2})\rangle$ est en réalité toujours LU-équivalent à un état $|EPR\rangle$.

Proposition 5.2.2. *Pour $U = \begin{pmatrix} 1 & 0 \\ 0 & e^{2i\pi\phi} \end{pmatrix}$, $\phi \in [0, 1]$, dans la phase pré-TFQ de l'AEPQ, donne pour tout n ,*

$$|\psi\rangle_{AEPQ}(\frac{1}{2}) \sim_{LU} |+\rangle |+\rangle \dots |+\rangle |EPR\rangle.$$

Démonstration. Considérons

$$\begin{aligned} \left| \psi_{AEPQ}(\frac{1}{2}) \right\rangle &= \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle |0\rangle + e^{i\pi x} |x\rangle |1\rangle \\ &= \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle |0\rangle + (-1)^x |x\rangle |1\rangle \end{aligned}$$

Si x est impair, alors on peut écrire $|x\rangle = |k\rangle |1\rangle$ avec $k = \frac{x-1}{2}$ alors que si x est pair, nous avons $|x\rangle = |k\rangle |0\rangle$ avec $k = \frac{x}{2}$ donc,

$$\begin{aligned} \left| \psi_{AEPQ}(\frac{1}{2}) \right\rangle &= \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle |0\rangle + (-1)^x |x\rangle |1\rangle \\ &= \frac{1}{\sqrt{2^{n+1}}} \sum_{k=0}^{2^{n-1}-1} |k\rangle |0\rangle |0\rangle + |k\rangle |1\rangle |0\rangle + |k\rangle |0\rangle |1\rangle - |k\rangle |1\rangle |1\rangle \\ &= \frac{1}{\sqrt{2^{n-1}}} \sum_{k=0}^{2^{n-1}-1} |k\rangle \frac{1}{\sqrt{2}} (|0\rangle |+\rangle + |1\rangle |-\rangle) = |+\rangle |+\rangle \dots |+\rangle \left(\frac{|0\rangle |+\rangle + |1\rangle |-\rangle}{\sqrt{2}} \right) \\ &\sim_{LU} |+\rangle |+\rangle \dots |+\rangle |EPR\rangle. \end{aligned} \tag{5.2.5}$$

Ce qui prouve que pour tout n , $\left| \psi_{AEPQ}(\frac{1}{2}) \right\rangle$ est LU-équivalent au produit d'un état séparable $n-2$ et d'un état $|EPR\rangle$. \square

En effet, si l'on considère les transformations $SLOCC = SL_2(\mathbb{C}) \times \dots \times SL_2(\mathbb{C})$, c'est-à-dire l'opération locale inversible, on a la caractérisation suivante des états générés par l'AEPQ.

Proposition 5.2.3. *Pour $U = \begin{pmatrix} 1 & 0 \\ 0 & e^{2i\pi\phi} \end{pmatrix}$ et $\phi \in [0, 1]$, les états générés dans la phase pré-TFQ de l'AEPQ satisfont pour tout n ,*

1. $|\psi_{AEPQ}(0)\rangle, |\psi_{AEPQ}(1)\rangle$ sont séparables,
2. $\left| \psi_{AEPQ}(\frac{1}{2}) \right\rangle$ est SLOCC équivalent à un état $|EPR\rangle$ biséparable (c'est-à-dire qu'il peut être écrit comme un produit tensoriel entre deux sous-systèmes, dont l'un est séparable et l'autre est une paire EPR),
3. $|\psi_{AEPQ}(\phi)\rangle$ est SLOCC équivalent à un état généralisé $|GHZ\rangle$.

Démonstration. Démontrons les trois assertions de la proposition 5.2.3 :

1. est clair car U est la matrice identité pour $\phi = 0, 1$.
2. est une conséquence de la proposition 5.2.2, puisque l'équivalence LU implique l'équivalence SLOCC.
3. Démontrons le dernier point :

$$\begin{aligned} |\psi^{\text{AEPQ}}(\phi)\rangle &= \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} (|x\rangle |0\rangle + e^{2i\pi x\phi} |x\rangle |1\rangle) \\ &= \frac{1}{\sqrt{2}} \left(\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle |0\rangle + \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2i\pi x\phi} |x\rangle |1\rangle \right) \\ &= \frac{1}{\sqrt{2}} \left(|+\rangle \dots |+\rangle |0\rangle + \left(\frac{|0\rangle + e^{2i\pi 2^{n-1}\phi} |1\rangle}{\sqrt{2}} \right) (\dots) \left(\frac{|0\rangle + e^{2i\pi 2^0\phi} |1\rangle}{\sqrt{2}} \right) |1\rangle \right). \end{aligned}$$

Pour $\phi \neq 0, \frac{1}{2}, 1$, les états $|+\rangle$ et $\frac{|0\rangle + e^{2i\pi 2^k\phi} |1\rangle}{\sqrt{2}}$ sont linéairement indépendants pour $k = 0, \dots, n-1$. Il existe donc une matrice $A_k \in \text{SL}_2(\mathbb{C})$ telle que

$$A_k |+\rangle = |0\rangle \text{ et } A_k \left(\frac{|0\rangle + e^{2i\pi 2^{n-1}\phi} |1\rangle}{\sqrt{2}} \right) = |1\rangle,$$

c'est-à-dire que $|\psi^{\text{AEPQ}}(\phi)\rangle$ est SLOCC équivalent à $|GHZ\rangle_{n+1}$.

□

Ainsi, dans le cas le plus simple, lorsque le second registre est constitué d'un seul qubit et que la matrice U reste inchangée, nous venons de prouver que la plupart des états générés par la phase pré-TFQ de AEPQ sont SLOCC équivalents aux états $|GHZ\rangle$ généralisés. Bien sûr, les équivalences SLOCC ne préservent pas l'intrication quantitativement mais seulement qualitativement ce qui explique les variations des trois invariants.

Ce cas simplifié de la Proposition 5.2.3 peut être généralisé à n'importe quelle dimension si l'on considère une superposition non intriquée de deux vecteurs propres séparables de la matrice U . Cette superposition non intriquée est le premier cas intéressant car nous avons déjà souligné que si l'on initialise l'algorithme avec un seul vecteur propre séparable sur le deuxième registre, il n'y a pas d'intrication générée par la phase pré-TFQ (voir l'équation (2.2.2)). On obtient ainsi le résultat général qui suit.

Théorème 5.2.1. *Considérons U comme un opérateur unitaire à t qubits. Soient deux vecteurs propres séparables $|u_1\rangle$ et $|u_2\rangle$ de U ayant des valeurs propres respectives $e^{2i\pi\phi_1}$ et $e^{2i\pi\phi_2}$. Supposons d'abord que $|u\rangle = \frac{1}{\sqrt{2}}(|u_1\rangle + |u_2\rangle)$ soit un état séparable. Ensuite, si nous initialisons le deuxième registre de l'AEPQ avec $|u\rangle$, l'état quantique à $n+t$ qubits $|\psi^{\text{AEPQ}}\rangle$ généré par la partie pré-TFQ de l'AEPQ est SLOCC équivalent à l'une des trois possibilités suivantes :*

1. un état séparable si $\phi_2 - \phi_1 = 0$ ou $\phi_2 - \phi_1 = 1$;
2. le produit tensoriel d'un état séparable qubit ($n+t-2$) et d'un état $|EPR\rangle$ si $\phi_2 - \phi_1 = \frac{1}{2}$;
3. le produit tensoriel d'un état séparable à $(t-1)$ qubits et d'un état GHZ généralisé à $(n+1)$ qubits, $|GHZ_{n+1}\rangle = \frac{|00\dots 0\rangle + |11\dots 1\rangle}{\sqrt{2}}$ pour toutes les autres valeurs de $\phi_2 - \phi_1$.

Démonstration. Supposons que l'état en entrée de l'AEPQ soit maintenant $|0\dots 0\rangle \otimes \left(\frac{1}{\sqrt{2}}(|u_1\rangle + |u_2\rangle)\right)$ où $|u_1\rangle$ et $|u_2\rangle$ sont les vecteurs propres de U avec comme valeurs propres $e^{2i\pi\phi_1}$ et $e^{2i\pi\phi_2}$. Alors, d'après l'équation (2.2.2), à la fin de la partie pré-TFQ de l'algorithme l'état généré prend la forme suivante :

$$|\psi^{\text{aepq}}\rangle = \frac{1}{\sqrt{2}} \left(\frac{|0\rangle + e^{2i\pi 2^{n-1}\phi_1} |1\rangle}{\sqrt{2}} \right) \otimes \dots \otimes \left(\frac{|0\rangle + e^{2i\pi 2^0\phi_1} |1\rangle}{\sqrt{2}} \right) \otimes |u_1\rangle \\ + \frac{1}{\sqrt{2}} \left(\frac{|0\rangle + e^{2i\pi 2^{n-1}\phi_2} |1\rangle}{\sqrt{2}} \right) \otimes \dots \otimes \left(\frac{|0\rangle + e^{2i\pi 2^0\phi_2} |1\rangle}{\sqrt{2}} \right) \otimes |u_2\rangle. \quad (5.2.6)$$

Considérons les matrices unitaires $U_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{-2i\pi 2^{n-k}\phi_1} \end{pmatrix}$ pour $k = 1, \dots, n$, puis en agissant par $U_1 \otimes \dots \otimes U_n$ sur le premier registre, transforme l'équation (5.2.6) dans

$$U_1 \otimes \dots \otimes U_n \otimes I_{2^t} |\psi^{\text{AEPQ}}\rangle = \frac{1}{\sqrt{2}} \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \otimes \dots \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) |u_1\rangle \\ + \frac{1}{\sqrt{2}} \left(\frac{|0\rangle + e^{2i\pi 2^{n-1}(\phi_2 - \phi_1)} |1\rangle}{\sqrt{2}} \right) \otimes \\ \dots \otimes \left(\frac{|0\rangle + e^{2i\pi 2^0(\phi_2 - \phi_1)} |1\rangle}{\sqrt{2}} \right) |u_2\rangle \\ = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} (|x\rangle |u_1\rangle + e^{i2\pi x(\phi_2 - \phi_1)} |x\rangle |u_2\rangle). \quad (5.2.7)$$

En supposant que $|u_1\rangle$ et $|u_2\rangle$ sont deux vecteurs propres séparables de dimension 2^t , tels que $|u\rangle = \frac{1}{\sqrt{2}}(|u_1\rangle + |u_2\rangle)$ soit non intriqué, on en déduit qu'il existe une transformation unitaire locale $U'_1 \otimes \dots \otimes U'_t$ telle que $U'_1 \otimes \dots \otimes U'_t |u_1\rangle = |00\dots 0\rangle$ et $U'_1 \otimes \dots \otimes U'_t |u_2\rangle = |10\dots 0\rangle$ (si ce n'était pas vrai, $|u\rangle$ serait intriqué). En combinant ces transformations, on obtient l'équivalence suivante :

$$|\psi^{\text{AEPQ}}\rangle \equiv_{LU} \frac{1}{\sqrt{2^{n+1}}} \left(\sum_{x=0}^{2^n-1} |x\rangle |0\rangle + e^{i2\pi x(\phi_2 - \phi_1)} |x\rangle |1\rangle \right) \otimes |0\dots 0\rangle. \quad (5.2.8)$$

Si l'on néglige les $t-1$ derniers qubits qui sont dans l'état $|00\dots 0\rangle$, alors, l'état de l'équation (5.2.8) est l'état $|\psi^{\text{AEPQ}}(\phi)\rangle$ de la proposition 5.2.3 avec $\phi = \phi_2 - \phi_1$. L'application des conclusions de la proposition 5.2.3 termine la preuve du théorème :

- Si $\phi_2 - \phi_1 = 0$ ou $\phi_2 - \phi_1 = 1$ alors $|\psi^{\text{AEPQ}}\rangle$ est séparable selon la première conclusion de la proposition 5.2.3.
- Si $\phi_2 - \phi_1 = \frac{1}{2}$, alors, d'après la deuxième conclusion de la proposition 5.2.3, on a $|\psi^{\text{AEPQ}}\rangle \equiv_{SLOCC} |+\rangle \otimes |+\rangle \dots \otimes |+\rangle \otimes |EPR\rangle \otimes |00\dots 0\rangle$.
- Et si $\phi_2 - \phi_1 \neq 0, \frac{1}{2}, 1$, alors la dernière partie de la proposition 5.2.3 donne

$$|\psi^{\text{AEPQ}}\rangle \equiv_{SLOCC} |GHZ_{n+1}\rangle \otimes |00\dots 0\rangle.$$

□

Cette étude peut être complétée par une illustration de ce qui peut se produire si l'on prend en compte plus de deux vecteurs propres séparables de U dans une superposition non intriquée. Dans l'exemple suivant, la superposition de quatre vecteurs propres, introduisant l'intrication sur le deuxième registre, montre que la conclusion du théorème 5.2.1 n'est pas vérifiée en général pour plus de deux vecteurs propres. Dans ce cas, considérons l'opérateur unitaire U suivant :

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & e^{2i\pi\phi} & 0 \\ 0 & 0 & 0 & e^{-2i\pi\phi} \end{pmatrix}. \quad (5.2.9)$$

Nous appliquons la partie pré-TFQ de AEPQ, c'est-à-dire le circuit donné par la Figure 2.2.5 pour $|u\rangle = |+\rangle^{\otimes 2}$ (non intriqué) et la matrice U nouvellement définie. Notez que maintenant l'état d'entrée du deuxième registre, $|u\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$, est la superposition égale des quatre vecteurs propres séparables de U . Un calcul simple montre que dans ce cas :

$$\begin{aligned} |\psi^{\text{AEPQ}}(\phi)\rangle &= \frac{1}{\sqrt{2^{n+2}}} \sum_{x=0}^{2^n-1} (|x\rangle |00\rangle + (-1)^x |x\rangle |01\rangle + e^{-2i\pi\phi x} |x\rangle |10\rangle + e^{2i\pi\phi x} |x\rangle |11\rangle) \\ &= \frac{1}{2} (|+\rangle^{n-1} |00\rangle |+\rangle) + \frac{1}{2} (|+\rangle^{n-1} |10\rangle |-\rangle) \\ &\quad + \frac{1}{2} \left(\frac{|0\rangle + e^{-2i\pi 2^{n-1}\phi} |1\rangle}{\sqrt{2}} \right) (\dots) \left(\frac{|0\rangle + e^{-2i\pi 2^0\phi} |1\rangle}{\sqrt{2}} \right) |10\rangle \\ &\quad + \frac{1}{2} \left(\frac{|0\rangle + e^{2i\pi 2^{n-1}\phi} |1\rangle}{\sqrt{2}} \right) (\dots) \left(\frac{|0\rangle + e^{2i\pi 2^0\phi} |1\rangle}{\sqrt{2}} \right) |11\rangle. \end{aligned} \quad (5.2.10)$$

En particulier, on constate que dans ce cas, l'état généré par l'algorithme est déjà intriqué pour $\phi = 0$ (de même que pour $\phi = 1$) en raison de la valeur propre -1 de U . En effet, dans ce cas, on a :

$$\begin{aligned} |\psi^{\text{AEPQ}}(0)\rangle &= \frac{1}{2} |+\rangle^{n-1} |00\rangle |+\rangle + \frac{1}{2} |+\rangle^{n-1} |10\rangle |-\rangle \\ &\quad + \frac{1}{2} |+\rangle^{n-1} |+\rangle |10\rangle + \frac{1}{2} (|+\rangle^{n-1} |+\rangle |11\rangle). \end{aligned} \quad (5.2.11)$$

Ce qui mène à :

$$|\psi^{\text{AEPQ}}(0)\rangle = |+\rangle^{\otimes n-1} \otimes \left(\frac{|00+\rangle + |10-\rangle + \sqrt{2}|+1+\rangle}{2} \right). \quad (5.2.12)$$

Cette dernière équation montre que les trois derniers qubits sont intriqués.

Nous avons calculé numériquement l'évolution de MGI , μ , et E_{moy} pour $n = 3$ (Figure 5.2.2). La Figure 5.2.2 confirme que l'état $|\psi^{\text{AEPQ}}(\phi)\rangle$ est intriqué pour tout $\phi \in [0, 1]$. On peut également remarquer que les courbes de MGI et μ ne sont plus symétriques dans ce cadre.

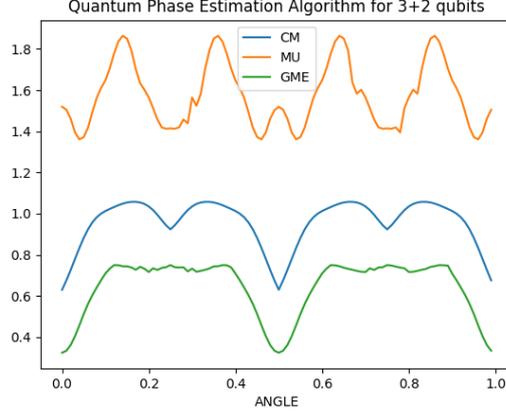


FIGURE 5.2.2 – Évaluations numériques des trois invariants MGI , μ et E_{moy} et pour les états quantiques $|\psi^{\text{AEPQ}}(\phi)\rangle$ en fonction de ϕ lorsque le deuxième registre est constitué de deux qubits pour $n = 3$.

5.3 Intrication de $|\psi^{\text{ACQ}}(\phi)\rangle$

On peut tracer des courbes similaire à celles de l'AEPQ pour ACQ en considérant

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

La Figure 5.3.1 fournit des résultats numériques concernant les trois mesures d'intrication à la fin de la partie pré-TFQ de l'ACQ en fonction de l'angle ϕ où $2\pi\phi = \theta$. On observe une similarité avec la courbe d'intrication de l'AEPQ. Il s'agit dorénavant, dans la Figure 5.3.1 des données de l'évaluation numérique des trois invariants, μ , E_{moy} et MGI pour les états quantiques $|\psi^{\text{ACQ}}(\phi)\rangle$ (équation 5.3.1) générés par ACQ en fonction de ϕ pour $n \in \{3, 4, 5\}$ qubits.

On peut montrer qu'à la fin de la phase pré-TFQ de l'ACQ décrite à la Section 2.2.4 l'état du système est :

$$|\psi^{\text{ACQ}}(\phi)\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} e^{-2\pi\phi x} |x\rangle |\psi_+\rangle + e^{2\pi\phi x} |x\rangle |\psi_-\rangle, \quad (5.3.1)$$

où $|\psi_+\rangle$ et $|\psi_-\rangle$ sont les valeurs propres de l'opérateur de Grover G (2.2.1). Basé sur l'équation 2.2.2, on peut réécrire l'état 5.3.1 tel que :

$$\begin{aligned} |\psi^{\text{ACQ}}(\phi)\rangle &= \frac{1}{\sqrt{2}} \left(\frac{|0\rangle + e^{-2i\pi 2^{n-1}\phi} |1\rangle}{\sqrt{2}} \right) \otimes \dots \otimes \left(\frac{|0\rangle + e^{-2i\pi 2^0\phi} |1\rangle}{\sqrt{2}} \right) \otimes |\psi_+\rangle \\ &+ \left(\frac{|0\rangle + e^{2i\pi 2^{n-1}\phi} |1\rangle}{\sqrt{2}} \right) \otimes \dots \otimes \left(\frac{|0\rangle + e^{2i\pi 2^0\phi} |1\rangle}{\sqrt{2}} \right) \otimes |\psi_-\rangle. \end{aligned} \quad (5.3.2)$$

Un argument similaire à la preuve du théorème 5.2.1 nous permet d'établir une relation entre $|\psi^{\text{ACQ}}(\phi)\rangle$ et $|\psi^{\text{AEPQ}}(2\phi)\rangle$. Considérons la matrice unitaire $U_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2i\pi 2^{n-k}\phi} \end{pmatrix}$

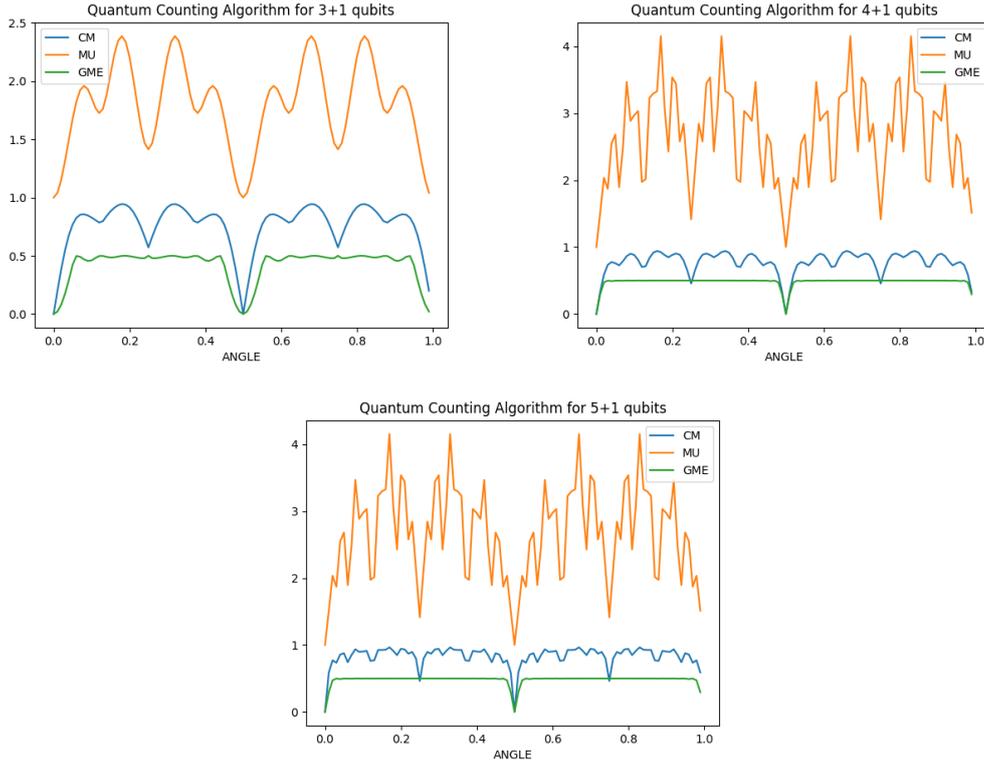


FIGURE 5.3.1 – Évaluations numériques des états issus de ACQ par les trois invariants.

pour $k \in \{1, \dots, n\}$, en agissant par $U_1 \otimes \dots \otimes U_n$ sur le premier registre, qui transforme l'équation 5.3.2 en :

$$\begin{aligned}
 U_1 \otimes \dots \otimes U_n \otimes I_{2^m} |\psi^{\text{ACQ}}(\phi)\rangle &= \frac{1}{\sqrt{2}} \left(\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \otimes \dots \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) |\psi_+\rangle \right) \\
 &+ \left(\left(\frac{|0\rangle + e^{2i\pi 2^{n-1} 2\phi} |1\rangle}{\sqrt{2}} \right) \otimes \dots \otimes \left(\frac{|0\rangle + e^{2i\pi 2^0 2\phi} |1\rangle}{\sqrt{2}} \right) |\psi_-\rangle \right) \\
 &= \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle |\psi_+\rangle + e^{i2\pi x 2\phi} |x\rangle |\psi_-\rangle \quad (5.3.3)
 \end{aligned}$$

L'intrication étant invariante par actions de transformations LU, on peut utiliser cette dernière expression équivalente à l'équation 5.3.3.

$$|\psi^{\text{ACQ}}(\phi)\rangle \equiv_{LU} \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle |\psi_+\rangle + e^{i2\pi x 2\phi} |x\rangle |\psi_-\rangle \quad (5.3.4)$$

Si on ne considère pas l'intrication restreinte au second registre, on peut choisir $|\psi_+\rangle = |0\rangle$ et $|\psi_-\rangle = |1\rangle$ comme vecteurs orthogonaux unitaires qui couvrent l'espace bidimensionnel sur le second registre. Alors, il est clair qu'avec cette simplification on a :

$$|\psi^{\text{ACQ}}(\phi)\rangle = |\psi^{\text{AEPQ}}(2\phi)\rangle. \quad (5.3.5)$$

L'équation 5.3.5 explique pourquoi la Figure 5.3.1 restreinte à $\phi \in [0, \frac{1}{2}]$ est similaire à la Figure 5.2.1.

Considérons quelques valeurs particulièrement intéressantes. Comme on peut s’y attendre, il n’y a pas d’intrication pour $\phi = 0$, $\phi = \frac{1}{2}$ et $\phi = 1$ qui correspondent respectivement à $\theta = 0, \pi$, et 2π , soit $\sin^2(\frac{\theta}{2}) = 0$ ou $\sin^2(\frac{\theta}{2}) = 1$. Cela correspond aux deux cas extrêmes où $M = 0$ (aucun élément marqué) et $M = N$ (tous les éléments marqués). Dans ces cas, l’algorithme ne génère jamais d’intrication, car il peut également être déduit de l’équation 5.3. La valeur $\phi = \frac{1}{4}$ (et par symétrie $\frac{3}{4}$) est également intéressante. Cela correspond au cas où $M = \frac{N}{2}$. Il est à noter que, d’après notre calcul sur l’AEPQ, dans ce cas, l’intrication est de type $|EPR\rangle$.

5.4 Conclusion

Dans ce chapitre, nous avons étudié l’intrication générée par l’AEPQ et l’ACQ avant l’application de l’inverse de la TFQ. Nous avons appelé cette partie, la phase pré-TFQ des algorithmes. En suivant les travaux de [80], nous avons prouvé que ces algorithmes génèrent de l’intrication pour $\phi \in [0, 1]$ lorsque les qubits du second registre sont en superposition. Pour AEPQ, nous avons étudié numériquement et analytiquement le cas le plus simple possible lorsque le second registre est constitué d’un seul qubit. Ce cas présente déjà des propriétés intéressantes. Les outils de mesure de l’intrication que nous avons utilisées sont symétriques par rapport à $\phi = \frac{1}{2}$ et nous avons prouvé que pour $\phi \neq 0, \frac{1}{2}, 1$, les états générés sont SLOCC équivalents aux états GHZ généralisés. Pour $\phi = \frac{1}{2}$, la quantité d’intrication atteint un minimum local correspondant à un état $|EPR\rangle$. Ce fait est également prouvé indépendamment du nombre de qubits sur le premier registre. Enfin, à partir de ces approches empiriques, nous avons pu prouver, que de manière générale, dans la partie pré-TFQ de l’AEPQ, les états générés par les algorithmes sont la plupart du temps SLOCC équivalents à un état $|GHZ_{n+1}\rangle$ si l’entrée du second registre est la superposition de deux vecteurs propres séparables de U . Nous avons également prouvé que ce résultat n’est plus vrai pour une superposition de quatre vecteurs propres. Pour l’ACQ, nous avons trouvé une relation entre les états générés par l’algorithme et ceux générés par l’AEPQ. Elle montre en particulier que la quantité d’intrication est sensible au rapport entre le nombre d’éléments marqués et le nombre total d’éléments. En particulier, on retrouve l’état $|EPR\rangle$ lorsque le nombre d’éléments marqués et non marqués est égal. Lorsque le nombre d’éléments marqués est nul (ou égal au nombre total d’éléments), l’algorithme ne génère pas d’intrication. Cette dernière observation est logique car dans ce cas l’algorithme ne fait rien de spécial.

Le dernier chapitre constitue l’exploration des agents réactifs, représentant une autre part importante de l’étude de l’intrication quantique. Nous explorerons d’abord le mécanisme de ces agents, puis les états qui peuvent en être issus et enfin, nous en étudierons l’intrication quantique à l’aide de nos outils habituels. Ce chapitre enrichira notre compréhension de l’évolution de l’intrication des états quantiques et particulièrement de l’interprétation des résultats de nos invariants.

Chapitre 6

AGENTS RÉACTIFS

Contenu du Chapitre

6.1	Contexte	56
6.2	Agents réactifs et Circuits quantiques	56
6.3	Implémentations et Généralisation	60
6.4	Autres résultats	61
6.5	Conclusion	66

6.1 Contexte

Les agents réactifs font référence à un modèle de programmation qui se concentre sur la réactivité et la réponse de robots aux événements et aux stimuli externes. La simplicité et le déterminisme de leurs actions font d'eux des cas d'étude et de modélisation qui suscitent de l'intérêt [52, 53, 54, 61, 70, 68]. En ce qui nous concerne, nous nous sommes intéressés à détecter et à étudier l'intrication produite par les états issus de circuits implémentant certains agents réactifs, et ce à l'aide des invariants que nous avons étudiés au Chapitre 3 et utilisés aux Chapitres 4 et 5. Du fait de la quantité de stimuli et de réactions, modéliser les circuits d'agents réactifs nécessite généralement un nombre de qubits élevé. Alors, afin de simplifier les calculs, nous avons fait usage de qudits pour calculer l'invariant E_{moy} qui a particulièrement été utile dans cette étude. À l'issue de nos travaux, nous parvenons à détecter la présence de l'intrication et ainsi que des cas particuliers parmi les circuits étudiés. Dans la suite, nous commençons par définir le formalisme de mise en circuit des agents réactifs, à partir d'un exemple de la littérature [54], puis nous proposons une généralisation des cas à étudier et enfin, nous présentons les résultats d'évaluation par les invariants.

6.2 Agents réactifs et Circuits quantiques

Les agents réactifs ont été étudiés par A. Koukam et al. [54], qui les ont modélisés sous forme de circuits quantiques. En effet, leurs travaux ont consisté en la création de circuits quantiques représentant les actions d'un robot en réaction à ses perceptions et suivant des règles définies au préalable. Afin de sélectionner la bonne action à effectuer par le robot lorsqu'il a plusieurs perceptions en même temps, l'architecture de subsomption a été utilisée pour permettre de faire une hiérarchisation correcte de la liste d'actions. Nous étant basés sur ce formalisme, dans cette section, nous commençons d'abord par expliquer le formalisme de création de ces circuits d'agents réactifs ; ensuite, nous expliquons l'architecture de subsomption et enfin, nous présentons le résultat final en un circuit quantique complet.

PERCEPTIONS ET ACTIONS Le comportement des agents réactifs s'apparente à un réflexe : le robot réagit d'une manière pré-déterminée aux informations qu'il perçoit. Néanmoins, malgré cette apparente simplicité de comportement, il est plus complexe de représenter le modèle perception-action. A. Koukam et al. [54] ont étudié un robot qui se déplace dans une aire où sont placées des balles et des obstacles. L'objectif du robot est de se déplacer dans cette aire et de ranger les balles dans une zone de dépôt tout en évitant les obstacles. Ce robot possède **quatre** (04) perceptions à savoir :

- repérer une balle (B pour *Ball* en anglais) ;
- se trouver dans la zone de dépôt (D pour *Deposit area* en anglais) ;
- repérer un obstacle (O pour *Obstacle* en anglais) ;
- tenir une balle (C pour *Carrying a ball* en anglais).

et réalise **cinq** actions selon les règles suivantes :

1. (a_1) : si O , alors, changer de direction ;
2. (a_2) : si D et C , alors, déposer la balle ;
3. (a_3) : si D et non C , alors, suivre la trajectoire ;

4. (a_4) : si B , alors, saisir la balle ;
5. (a_5) : si aucune des perceptions précédentes n'est observée, alors, se déplacer aléatoirement.

Pour réaliser le circuit quantique, on fait correspondre chaque perception à un qubit : $|B\rangle, |D\rangle, |O\rangle$ et $|C\rangle$. On obtient ainsi l'état formé de toutes les perceptions du robot qui s'écrit :

$$|Perception\rangle = |BDOC\rangle \quad (6.2.1)$$

dans le même ordre que la liste des perceptions.

Exemple 6.2.1. *L'état $|1100\rangle$ c'est-à-dire $B = |1\rangle, D = |1\rangle, O = |0\rangle$ et $C = |0\rangle$, traduit un scénario dans lequel le robot a détecté à la fois un obstacle et une balle, n'est pas dans une zone de dépôt et ne tient aucune balle.*

De la même manière, chaque action est représentée par un qubit. Toutes les actions du robot peuvent alors être transcrites par l'état ci-après :

$$|Action\rangle = |a_1a_2a_3a_4a_5\rangle. \quad (6.2.2)$$

Exemple 6.2.2. *L'action $|1000\rangle$ correspondant à $(a_1) = 1, (a_2) = 0, (a_3) = 0, (a_4) = 0$ et $(a_5) = 0$, désigne uniquement l'activation de l'action (a_1) qui fait changer de direction au robot.*

Comme mentionné plus haut, un robot peut avoir plusieurs perceptions à la fois. Par exemple, l'état $|1100\rangle$ correspond au scénario suivant : le robot détecte une balle et se trouve dans la zone de dépôt. Dans ce cas, plusieurs actions sont enclenchées et le robot doit en "sélectionner" une. La partie abordée par la suite est celle de l'architecture de décision du robot. Cette architecture permet au robot de "décider" de l'action à effectuer en priorité.

ARCHITECTURE DE SUBSOMPTION L'architecture de subsomption est un modèle d'architecture logicielle utilisé principalement en robotique et en intelligence artificielle. Proposée par R. Brooks dans les années 1980 [15], cette architecture est destinée à concevoir des robots capables de se comporter intelligemment dans des environnements complexes, sans avoir besoin de modèles explicites du monde ou de planifications centralisées. Les principes de l'architecture de subsomption sont :

1. L'architecture de subsomption est organisée en plusieurs couches, où chaque couche représente un niveau de comportement. Les couches les plus basses gèrent des comportements simples (comme éviter les obstacles), tandis que les couches supérieures ajoutent des comportements plus complexes (comme explorer ou planifier des trajets).
2. Chaque couche fonctionne indépendamment et n'a pas de contrôle direct sur les autres couches.
3. Les comportements des couches inférieures (plus basiques, comme l'évitement d'obstacles) qui peuvent "subsommer" (c'est-à-dire remplacer ou inhiber) les couches supérieures (comme le déplacement aléatoire) si nécessaire.
4. L'architecture de subsomption est conçue pour être réactive, c'est-à-dire que le robot peut répondre en temps réel à des stimuli de son environnement sans avoir à calculer ou planifier à long terme.

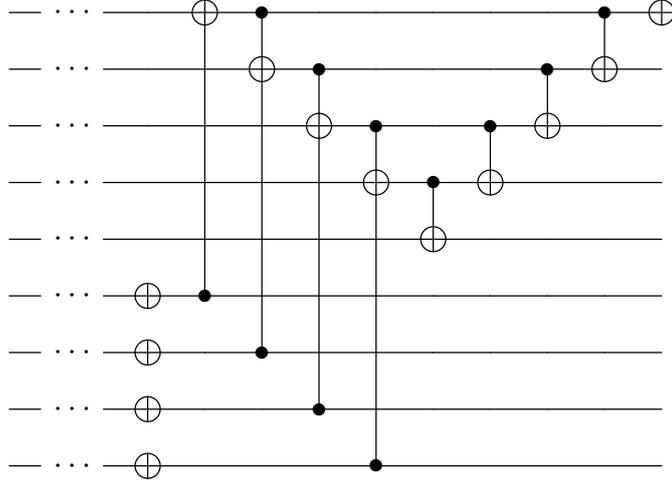


FIGURE 6.2.1 – Circuit de l’architecture de subsomption agissant sur un circuit à 9 qubits. Les 5 premiers qubits correspondent aux qubits de sortie qui encodent les actions de l’agent. Les 4 qubits suivants sont les qubits d’entrée qui encodent les perceptions.

L’architecture de subsomption présente plusieurs avantages. Par exemple, le robot peut répondre aux stimuli sans avoir besoin de calculs complexes. De plus, les robots peuvent réagir de manière flexible et s’adapter aux changements de l’environnement en temps réel.

La représentation mathématique de l’architecture de subsomption est donnée à l’Equation 6.2.3 telles qu’énoncées par A. Koukam et al. [54].

$$|a_i\rangle = \begin{cases} |\neg(\neg x_i)\rangle & \text{si } i = 1 \\ \left| \left(\bigwedge_{j=1}^i \neg x_j \right) \oplus \left(\bigwedge_{j=1}^{i-1} \neg x_j \right) \right\rangle & \text{si } 1 < i < 5 \\ \left| \left(\bigwedge_{j=1}^{i-1} \neg x_j \right) \right\rangle & \text{si } i = 5 \end{cases} \quad (6.2.3)$$

Les états $|a_i\rangle$ représentent alors les comportements ou décisions du robot, issue de l’architecture. Ainsi, lorsque l’état en entrée est $|1010\rangle$, l’état obtenu en sortie est $|10000\rangle$, ce qui est correct car dans ce cas, l’action $|a_1\rangle$ est prioritaire.

Dans le paragraphe suivant, nous expliquons comment ont été codées sous forme de circuit les perceptions et actions des agents réactifs d’après [54].

CIRCUIT GLOBAL Nous définissons l’état d’entrée du circuit d’un agent réactif comme le produit tensoriel du vecteur perception et du vecteur action.

$$|Entree\rangle = |Perception\rangle \otimes |Action\rangle$$

On désigne par $n, p \in \mathbb{N}$ respectivement le nombre de perceptions de d’actions. Pour une meilleure lisibilité de l’état des actions, des qubits auxiliaires $m \in \mathbb{N}$ sont rajoutés. Le circuit est alors divisé en deux parties. La première consiste en l’activation des qubits auxiliaires à partir des qubits de perception et des règles appliquées et la seconde se rapporte à aux placement des portes de l’architecture de subsomption.

Cette première partie de circuit de la Figure 6.2.2 représente d’une part, l’application de la succession des règles décrites plus haut en fonction des perceptions des robots et d’autre part l’activation des qubits auxiliaires. La seconde et la dernière partie du circuit comporte l’architecture de subsomption qui relie les qubits auxiliaires aux qubits d’action 6.2.1.

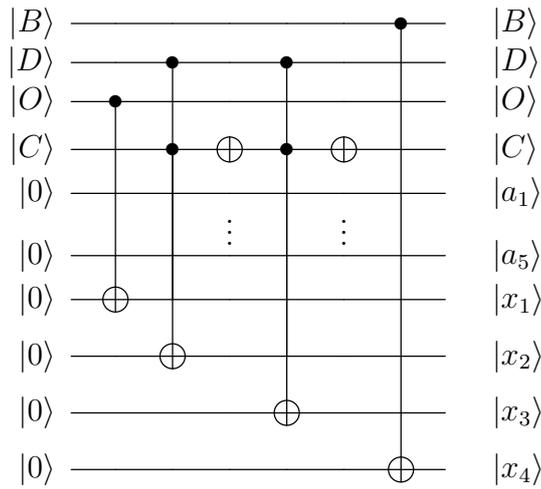


FIGURE 6.2.2 – Première partie des circuits d’agents réactifs.

Le circuit 6.2.1 de l’architecture de subsomption telle que décrite plus tôt. Dans ce circuit, l’architecture permet de sélectionner une action parmi les cinq (05) possibles. La deuxième étape du circuit permet d’aboutir à :

$$|a_i\rangle = \left| x_i \wedge \left(\bigwedge_{j=1}^{i-1} \neg x_j \right) \right\rangle.$$

Exemple 6.2.3. *Voici un exemple de circuit dans lequel un agent réactif détecte une balle, se trouve dans une zone de dépôt et tient une balle ce qui équivaut à |1101>. Dans*

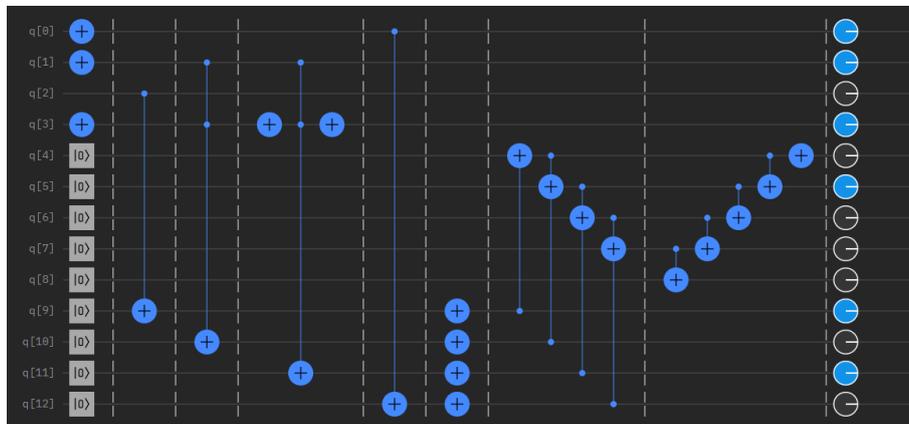


FIGURE 6.2.3 – Exemple d’un circuit complet des agents réactifs sur l’outil IBM Composer.

cet exemple, on peut citer dans l’ordre, les quatre premiers qubits correspondent aux perceptions, les cinq qubits suivants aux actions et les quatre derniers qubits sont des auxiliaires. L’état en sortie, uniquement les actions est |01000>. L’état |01000> revient à sélectionner l’action a_2 en priorité, c’est-à-dire, pour le robot de déposer la balle.

6.3 Implémentations et Généralisation

En faisant varier les règles agissant sur les perceptions en entrée d'un circuit, on peut construire divers circuits perceptions-actions-auxiliaires pour obtenir des états différents en sortie dont on peut étudier l'intrication. Pour évaluer cette intrication, nous avons utilisé quatre invariants qui sont : la MGI , μ , E_{moy} et le multirang. Dans le cas de l'invariant E_{moy} , un changement de base a été nécessaire. En effet, on rappelle qu'il y a n qubits de perception, m qubits auxiliaires et p qubits. Dans notre situation, le circuit de base comporte $n = 4$ qubits de perceptions, $m = 5$ qubits auxiliaires et $p = 5$ qubits actions soit un total de 13 qubits. Cependant, au lieu de considérer un système à $n + m + p$ qubits, nous avons abordé le problème comme un système à **3 qudits** de dimension respective 2^n , 2^m et 2^p . Cela est dû au fait qu'avec un état à 13 qubits, le nombre total de matrices d'aplatissement, nécessaires pour calculer l'invariant serait $2^{t-1} - 1 = 4095$ matrices, t étant le nombre total de qubits. Ce nombre est très élevé et augmente la durée de calcul de E_{moy} . En conclusion, pour faciliter les calculs, E_{moy} n'est plus calculé pour un système à 13 qubits mais bien pour un systèmes à 3 qudits. Pour finir, une porte Hadamard est placée en début de circuit sur les qubits qui correspondent au perceptions, dans le but d'apporter de l'intrication dans l'état final. On obtient alors le circuit de la Figure 6.3.1.

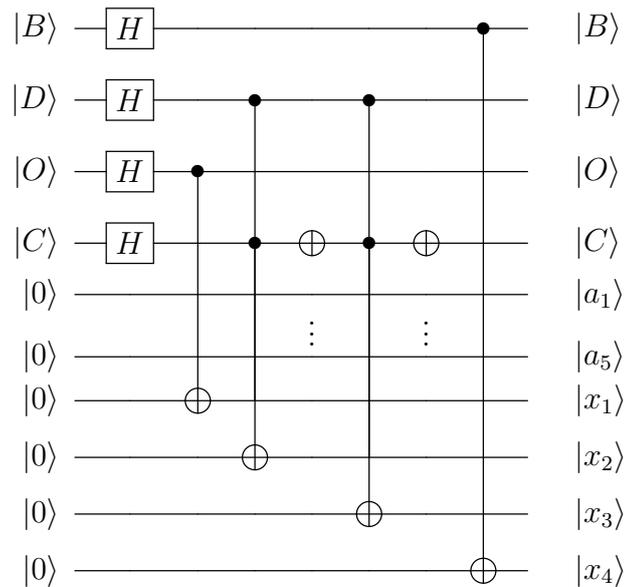


FIGURE 6.3.1 – Circuit d'un agent réactif avec en entrée une porte H sur les perceptions.

Avec cette disposition, le résultat de l'évaluation par l'invariant E_{moy} est supérieure à 0, signifiant que l'état est intriqué. Alors, pour aller plus loin et vérifier la présence d'intrication, nous avons placé des portes quantiques $U(\theta_j, \phi_j, \lambda_j)$. Il s'agit des portes déjà exploitées au Chapitre 4 à l'Equation 3.7.4. La première étape a consisté à placer une porte $U(\theta, 0, 0)$ sur un, deux, trois puis quatre qubits de perceptions avec $\theta \in [0, \pi]$ qui varie avec un pas de 0.1. Cette porte s'écrit :

$$U(\theta, 0, 0) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

Les résultats de l'évaluation par l'invariant E_{moy} du circuit créé à partir de chaque

porte U calculée pour une occurrence de θ de U produisent les courbes de la Figure 6.3.2.

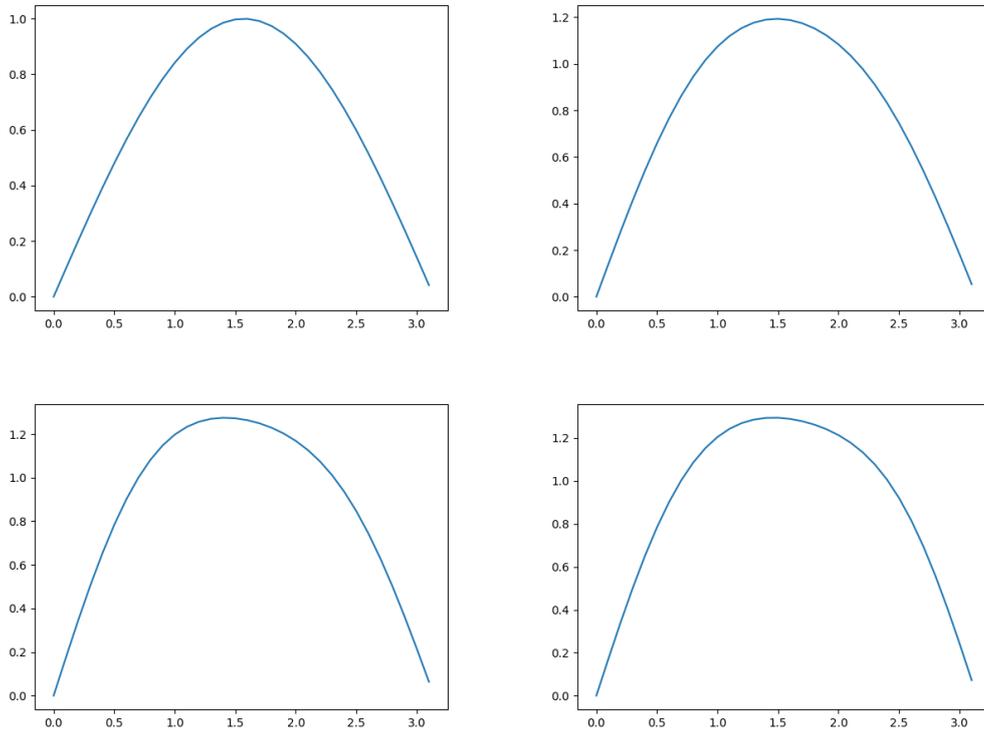


FIGURE 6.3.2 – Résultats de l'évaluation de l'état en sortie d'un circuit d'agent réactif par l'invariant E_{moy} avec une porte U sur un, deux, trois puis quatre qubits de perception. On remarque que l'intrication est inexistante lorsque $\theta = 0$. Elle atteint son maximum lorsque $\theta = \frac{\pi}{2}$ (ce qui équivaut aux portes Hadamard). Elle redescend à 0 lorsque $\theta = \pi$.

6.4 Autres résultats

A la lumière de ces premiers résultats, nous nous sommes proposés d'étudier tous les circuits générés par toutes les combinaisons possibles de règles. Ainsi, pour $n = 1$ perception, on obtient les trois combinaisons de règles ci-après :

- $[1]$ qui correspond à une action à entreprendre lorsque la perception est perçue ;
- $[-1]$ qui correspond à une action à entreprendre lorsque la perception n'est pas perçue ;
- $[[-1], [1]]$ qui correspond à deux actions à entreprendre lorsqu'une perception est perçue et si elle ne l'est pas.

Sur les qubits qui encodent les perceptions, une porte Hadamard est forcément placée en début de circuit. On remarquera que les tailles des listes de règles varient de 1 à 2 et qu'une combinaison est intuitivement impossible : $[-1, 1]$. En effet, cette règle signifierait une action du robot pour une perception et son inverse en même temps.

On calcule que pour $n = 2$ perceptions, on peut effectuer un total de $2^8 = 256$ combinaisons de règles. On commence d'abord par trier ces combinaisons en éliminant celles qui sont logiquement impossibles à concevoir, comme expliqué un peu plus haut.

On retire alors toute liste comprenant à la fois une perception et son opposée. Puis, on retire les règles qui répondent aux critères suivants :

- L'implémentation en circuit d'une règle "négative" implique de placer une porte X sur la perception ciblée. Cependant, on rappelle qu'une porte Hadamard est mise en entrée de chaque perception et que $XH = H$. On conclut alors que les règles $[i]$ et $[-i]$, $i \in \mathbb{N}^*$ produisent des circuits qui s'équivalent, du point de vue de l'intrication.
- Á une permutation de qubit près, certaines règles produisent également un même circuit à ce qui ne produit pas de changement significatif sur l'intrication. Exemple des règles $[[1], [1, 2]]$ et $[[2], [1, 2]]$.

En suivant cette démarche, on obtient au final, un total de 44 combinaisons de règles implémentables. Ces règles sont répertoriées dans le Tableau 6.1 qui contient également le nombre total de qubits nécessaires pour les implémenter. Ce nombre est la somme du nombre de qubits de perceptions (qui est toujours 2), de qubits auxiliaires (qui est égal au nombre de règles) et de qubits d'action (nombre de qubits auxiliaires +1). Car, le nombre de qubits d'action est égal au nombre de règles plus une dernière correspondant à un déplacement aléatoire du robot.

N°	Règles	Nombre de qubits
1	[[1]]	4
2	[[2]]	5
3	[[1, 2]]	5
4	[[1], [1, 2]]	7
5	[[1, 2], [1]]	7
6	[[1], [2]]	7
7	[[1], [-1]]	6
8	[[1, 2], [1, -2]]	7
9	[[1], [-1], [2]]	9
10	[[1], [-1], [1, 2]]	9
11	[[1, 2], [-1], [1]]	9
12	[[1], [2], [1, 2]]	9
13	[[1, 2], [2], [1]]	9
14	[[1], [1, 2], [-1, -2]]	9
15	[[1, -2], [1, 2], [1]]	9
16	[[1, 2], [-1, 2], [-1, -2]]	9
17	[[1], [-1], [1, 2], [-1, -2]]	11
18	[[1, -2], [1, 2], [-1], [1]]	11
19	[[1], [2], [1, 2], [-1, -2]]	11
20	[[1, -2], [1, 2], [2], [1]]	11
21	[[1], [-1], [2], [-2]]	11
22	[[1], [1, 2], [-1, 2], [-1, -2]]	11
23	[[1, -2], [-1, 2], [1, 2], [1]]	11
24	[[1, 2], [-1, 2], [1, -2], [-1, -2]]	11
25	[[1], [-1], [2], [-2], [1, 2]]	13
26	[[1, 2], [-2], [2], [-1], [1]]	13
27	[[1], [-1], [2], [1, 2], [-1, 2]]	13
28	[[1, -2], [1, 2], [2], [-1], [1]]	13
29	[[1], [-1], [1, 2], [-1, 2], [1, -2]]	13
30	[[1, -2], [-1, 2], [1, 2], [-1], [1]]	13
31	[[1], [1, 2], [-1, 2], [1, -2], [-1, -2]]	13
32	[[1, -2], [1, -2], [-1, 2], [1, 2], [1]]	13
33	[[1], [-1], [2], [-2], [1, 2], [-1, 2]]	15
34	[[1, -2], [1, 2], [-2], [2], [-1], [1]]	15
35	[[1], [-1], [2], [1, 2], [-1, 2], [1, -2]]	15
36	[[1, -2], [-1, 2], [1, 2], [2], [-1], [1]]	15
37	[[1], [-1], [1, 2], [-1, 2], [1, -2], [-1, -2]]	15
38	[[1, -2], [1, -2], [-1, 2], [1, 2], [-1], [1]]	15
39	[[1], [-1], [2], [-2], [1, 2], [1, -2], [1, -2]]	17
40	[[1, -2], [-1, 2], [1, 2], [-2], [2], [-1], [1]]	17
41	[[1], [-1], [2], [1, 2], [-1, 2], [1, -2], [-1, -2]]	17
42	[[1, -2], [1, -2], [-1, 2], [1, 2], [2], [-1], [1]]	17
43	[[1], [-1], [2], [-2], [1, 2], [-1, 2], [1, -2], [-1, -2]]	19
44	[[1, -2], [1, -2], [-1, 2], [1, 2], [-2], [2], [-1], [1]]	19

TABLE 6.1 – Tableau des 44 combinaisons de règles retenues pour $n = 2$ perceptions.

Néanmoins, toutes ces 44 combinaisons n'ont pas pu être étudiées à cause de l'espace mémoire requis pour évaluer l'intrication à l'aide de nos invariants. En effet, le calcul de l'invariant μ par exemple, requiert l'utilisation de matrices de taille $2^n \times 2^n$. Pour $n = 15$, une matrice de taille $32\,768 \times 32\,768$ contient $327\,682^2 = 1\,073\,741\,824$ éléments. Sachant qu'il s'agit de nombres complexes et qu'un nombre complexe occupe 16 octets en mémoire, la mémoire nécessaire pour stocker la matrice entière est $1\,073\,741\,824 \times 16$ octets soit 17 179 869 184 octets est égal à environ 17.18 Go de RAM. Le vecteur issu du circuit étudié est de taille $2^n = 32\,768$ ce qui fait $32\,768 \times 16 = 524\,288$ octets, égal à environ 0.50 Go de RAM. Somme toute, la mémoire totale nécessaire pour stocker à la fois la matrice et le vecteur est : 17.18 Go + 0.5 Go = 17.68 Go de RAM. Même avec une grande valeur de RAM, plusieurs facteurs peuvent empêcher notre code de fonctionner lorsqu'il manipule des matrices et des vecteurs aussi grands comme :

- la fragmentation de la mémoire : Les grands blocs contigus de mémoire nécessaires pour les matrices ne sont pas forcément disponibles ;
- l'utilisation de la mémoire par d'autres processus : le système d'exploitation lui-même consomme une part de la RAM disponible, réduisant la mémoire utilisable pour notre programme ;
- l'utilisation du ramasse-miettes (Garbage Collector, GC) : en Python, le GC peut entraîner une fragmentation de la mémoire ; son propre fonctionnement peut augmenter temporairement l'utilisation de la mémoire ;
- un problème dans le code : la manière dont les données sont gérées n'est pas adaptée à la manipulation de grandes données et peuvent entraîner une consommation de mémoire excessive.

Ces problèmes peuvent être surmontés en adoptant les méthodes suivantes pour optimiser le code :

- utiliser d'autres structures de données pour stocker les matrices, comme les listes chaînées, qui n'utilisent pas d'espace contiguë en mémoire ;
- intégrer les concepts de la Programmation Orientée Objet (POO) pour créer des structures de données adaptées, comme des objets encapsulés pour économiser de la mémoire lorsqu'une grande partie des éléments sont nuls ;
- utiliser un autre langage de programmation, comme C++ pour avoir un contrôle direct sur l'allocation et la libération de la mémoire, pour ses bibliothèques qui pourraient être plus adaptées et pour ses performances en général.

Au final, en tenant compte de la contrainte liée au nombre de qubits maximum, nous avons effectué les calculs jusqu'à $n = 13$ qubits soit jusqu'à la règle N°32 du Tableau 6.1. Dans le Tableau 6.2 sont rapportées les résultats de l'évaluation des vecteurs issus de circuits correspondant aux 32 règles sélectionnées à l'aide des invariants E_{moy} , μ et MGI que nous présentés au Chapitre 3.

N°	Règles	μ	E_{moy}	MGI
1	[[1]]	2.83	0.99	0.5
2	[[2]]	2.83	0.99	0.5
3	[[1, 2]]	2.73	0.87	0.35
4	[[1], [1, 2]]	3.62	1.08	0.75
5	[[1, 2], [1]]	3.46	1.12	0.75
6	[[1], [2]]	2.83	1.19	0.75
7	[[1], [-1]]	4.0	0.99	0.5
8	[[1, 2], [1, -2]]	2.45	1.12	0.5
9	[[1], [-1], [2]]	4.0	1.15	0.75
10	[[1], [-1], [1, 2]]	5.12	1.08	0.5
11	[[1, 2], [-1], [1]]	3.46	1.08	0.75
12	[[1], [2], [1, 2]]	4.0	1.15	0.75
13	[[1, 2], [2], [1]]	4.0	1.22	0.75
14	[[1], [1, 2], [-1, -2]]	4.0	1.15	0.75
15	[[1, -2], [1, 2], [1]]	4.0	1.19	0.75
16	[[1, 2], [-1, 2], [-1, -2]]	2.83	1.19	0.75
17	[[1], [-1], [1, 2], [-1, -2]]	5.66	1.12	0.75
18	[[1, -2], [1, 2], [-1], [1]]	5.66	1.15	0.75
19	[[1], [2], [1, 2], [-1, -2]]	5.66	1.19	0.75
20	[[1, -2], [1, 2], [2], [1]]	5.66	1.22	0.75
21	[[1], [-1], [2], [-2]]	5.66	1.08	0.75
22	[[1], [1, 2], [-1, 2], [-1, -2]]	2.83	1.19	0.75
23	[[1, -2], [-1, 2], [1, 2], [1]]	4.0	1.19	0.75
24	[[1, 2], [-1, 2], [1, -2], [-1, -2]]	2.83	1.19	0.75
25	[[1], [-1], [2], [-2], [1, 2]]	8.0	1.19	0.75
26	[[1, 2], [-2], [2], [-1], [1]]	5.66	1.12	0.75
27	[[1], [-1], [2], [1, 2], [-1, 2]]	5.66	1.19	0.75
28	[[1, 2], [1, 2], [2], [-1], [1]]	5.66	1.15	0.75
29	[[1], [-1], [1, 2], [-1, 2], [1, -2]]	5.66	1.15	0.75
30	[[1, -2], [-1, 2], [1, 2], [-1], [1]]	5.66	1.15	0.75
31	[[1], [1, 2], [-1, 2], [1, -2], [-1, -2]]	4.0	1.19	0.75
32	[[1, -2], [1, -2], [-1, 2], [1, 2], [1]]	4.0	1.19	0.75

TABLE 6.2 – Tableau regroupant les valeurs calculées des invariants pour toutes les combinaisons de règles énumérées.

Dans le Tableau 6.2 qui contient les résultats de l'évaluation de chaque circuit par invariant, on constate que :

1. Pour toutes les familles d'états, $\mu > 1, E_{moy} > 0, MGI > 0$. Ils sont donc tous intriqués.
2. Le nombre de circuits distingués par invariant est :
 - $\mu \rightarrow 9$ groupes de règles ;
 - $E_{moy} \rightarrow 7$ groupes de règles ;
 - $MGI \rightarrow 3$ groupes de règles ;
3. Les courbes d'évolution des résultats de l'évaluation des règles par les invariants selon les règles sont fournis par les Figures 6.4.1. L'allure de la courbe correspondant

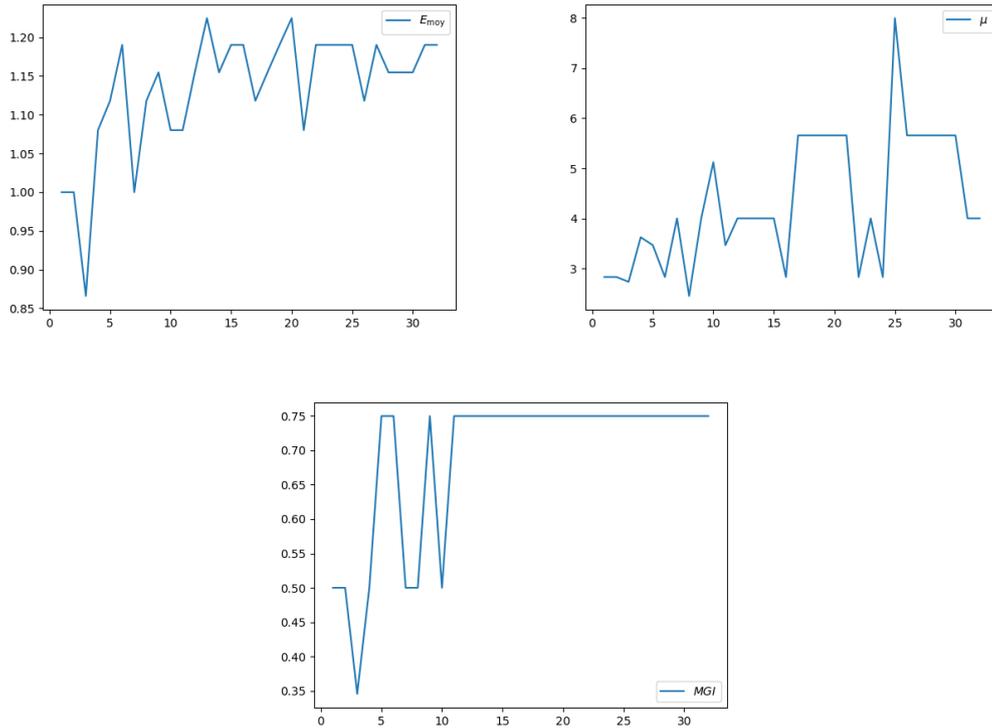


FIGURE 6.4.1 – Courbes de l'évaluation des résultats de l'évaluation des 32 combinaisons de règles par les invariants du Tableau 6.2.

à la MGI n'est pas surprenante, au vu des valeurs du tableau. La valeur maximale de cet invariant est 0.75 pour 37 familles. Quant aux invariants μ et E_{moy} , bien qu'il y ait plus de détails, les courbes ne varient pas de la même manière. Cela ne permet donc pas d'effectuer une comparaison directe entre les deux invariants comme cela était le cas dans le Chapitre 5 par exemple. On peut quand même noter que le maximum pour l'invariant μ est atteint pour la règle 1 et que celui de E_{moy} est atteinte pour deux règles : 13, 20. En effet, il faut noter que, dans le cas de μ , la valeur maximale est à estimer selon le nombre de qubits.

4. Les règles ayant les mêmes valeurs pour le trio d'invariants sont :
 - 1 et 2 ;
 - 6, 16, 22 et 24 ;
 - 15 et 23 ;
 - 9, 12 et 14.

6.5 Conclusion

Dans ce dernier chapitre, nous avons évalué l'intrication dans les circuits issus d'agents réactifs à l'aide des invariants MGI , μ , et E_{moy} . En partant d'un cas de la littérature [54], nous nous sommes intéressés à l'évaluation de l'intrication dans les circuits générés par les perceptions et actions des agents réactifs pour un jeu de règles et de celles qui peuvent être implémentées pour deux qubits de perception. Par la suite, on pourrait s'intéresser à vérifier l'évolution de l'intrication avant l'application de l'architecture de subsomption. Plus encore, on pourrait tester d'autres architectures de décision.

Conclusion générale

L'intrication est un phénomène propre à la physique quantique où l'état de deux ou de plusieurs particules corrélées ne peut s'écrire comme le produit tensoriel des états de base, même si elles sont séparées par une grande distance. Ce phénomène peut être mis en évidence par les inégalités de Bell [10] (et les polynômes de Mermin [62]). Ces inégalités ont été testées par A. Aspect et al. [6] dans les années 80.

Dans cette thèse, nous avons proposé cinq (05) outils de détection et de mesure de l'intrication que nous nommons "invariants". Il s'agit de la mesure géométrique de l'intrication (MGI), l'invariant μ , sa variante $\tilde{\mu}$ calculés à partir des polynômes de Mermin, l'invariant d'intrication moyenne E_{moy} et le multirang tous les deux obtenus à partir des matrices de coefficients. Trois (03) d'entre eux, c'est-à-dire la MGI , μ et $\tilde{\mu}$, présentent l'avantage d'être implémentables sur une machine quantique .

Nous avons utilisé les invariants pour évaluer l'intrication d'une classification d'états hypergraphiques à 4 qubits connectés [40]. Ainsi, nous avons pu établir que, ensemble, les invariants MGI , μ et $\tilde{\mu}$ fonctionnent bien pour différencier ces états d'hypergraphes. Plus particulièrement, les invariants μ et $\tilde{\mu}$ permettent de distinguer les-dites familles. Après avoir effectué des tests numériques, nous avons implémenté les états hypergraphiques sur un simulateur quantique avec l'invariant μ . Les résultats des tests sur les simulateurs quantiques étaient les mêmes que les résultats numériques. Cependant, les résultats de l'évaluation des états hypergraphiques par μ sur un ordinateur quantique n'ont pas été aussi bons à cause de la transpilation.

Nous avons également étudié l'intrication générée par les algorithmes d'estimation de phase quantique et de comptage quantique avant l'application de l'inverse de la Transformée de Fourier Quantique. Nous avons prouvé que ces algorithmes génèrent de l'intrication pour $\phi \in]0, 1[$ lorsque les qubits du second registre sont en superposition. Nous avons également pu prouver, que de manière générale, dans la partie pré-TFQ de l'AEPQ, les états générés par les algorithmes sont la plupart du temps SLOCC équivalents à un état $|GHZ_{n+1}\rangle$ si l'entrée du second registre est la superposition de deux vecteurs propres séparables de U , la matrice utilisée dans cet algorithme. Pour l'ACQ, nous avons trouvé une relation entre les états générés par l'algorithme et ceux générés par l'AEPQ.

Pour finir, nous nous sommes intéressés à l'évaluation de l'intrication dans les circuits générés par les perceptions et actions des agents réactifs pour un jeu de règles issu de la littérature [54] et de celles qui peuvent être implémentées pour un nombre donné de qubits de perception.

Dans nos travaux, nous avons remarqué des particularités sur les invariants utilisés. Tout d'abord, la MGI , μ et $\tilde{\mu}$ ont une signification physique claire. Pour la MGI , on mesure la distance vers l'état séparable le plus proche tandis que pour μ et $\tilde{\mu}$, on mesure l'état par rapport à différents observables (les monômes du polynôme de Mermin) et on combine les résultats de ces mesures pour évaluer l'opérateur de Mermin. Dans les deux cas, l'opération de base nécessaire pour calculer l'invariant est d'effectuer une (cas de la MGI) ou plusieurs mesures (cas de μ et $\tilde{\mu}$) dans des bases différentes (dans une

base impliquant l'état produit le plus proche pour la *MGI* et dans la base de l'observable définie par un monôme donné pour μ et $\tilde{\mu}$. C'est la raison pour laquelle les deux invariants peuvent être évalués sur un ordinateur quantique comme cela a été montré dans [37, 78], pour la *MGI* et dans [4] pour μ et $\tilde{\mu}$. La mesure d'intrication moyenne est davantage un invariant basé sur la nature algébrique de l'état quantique que l'on évalue. En effet, d'un point de vue tensoriel, un état séparable (pur) $|\psi\rangle$ est un tenseur de rang un, c'est-à-dire que pour tout aplatissement possible, les matrices de coefficients correspondantes seront de rang un, donc les invariants $\kappa_{(q_1, \dots, q_t)}(|\psi\rangle)$ seront nuls et le multirang est $\text{multirg}(|\psi\rangle) = [1, \dots, 1]$. La mesure d'intrication moyenne reflète la quantité générale d'intrication en vérifiant, dans chaque aplatissement, la possibilité que l'état soit un tenseur de rang un. Contrairement aux trois premières mesures, il n'y a pas de signification physique claire en termes de projection et il n'y a donc pas de moyen simple d'évaluer E_{moy} et le multirang directement sur un ordinateur quantique. Une autre différence majeure entre d'un côté la *MGI*, μ , $\tilde{\mu}$ et de l'autre côté E_{moy} et le multirang est le temps nécessaire pour calculer ces invariants. Pour la *MGI*, μ et $\tilde{\mu}$, il faut optimiser le choix des paramètres qui définissent soit l'état séparable le plus proche (pour la *MGI*) soit les observables utilisés dans le polynôme de Mermin (pour μ et $\tilde{\mu}$). Cette partie demande un temps de calcul plus élevé et c'est la raison pour laquelle nous sommes limités dans le nombre de qubits dans nos expériences numériques (par exemple pour les agents réactifs au Chapitre 6). Cependant, pour E_{moy} et le multirang, l'évaluation est plus rapide, car on peut calculer directement les invariants en se basant sur la connaissance de l'état $|\psi\rangle$.

Annexes

Tous les codes utilisés dans cette thèse sont disponibles à l'adresse suivante :
https://gitlab.com/grace.dorcas.amouzou/codes_these.

.1 Codes numériques en Python

Cette section regroupe les algorithmes qui ont été utilisés pour effectuer les calculs numériques.

.1.1 Algorithme de marche aléatoire

Afin d'optimiser les fonctions correspondants aux invariants MGI , μ et $\tilde{\mu}$ vus au Chapitre 3, l'algorithme de marche aléatoire a été utilisé pour trouver des valeurs qui, dans notre cas, les maximisent. La fonction f est celle de l'invariant à optimiser, p correspond aux coefficients à partir desquels sont calculés l'invariant, c est un compteur, c_{max} est la valeur maximale du compteur, α est le pas de descente et α_{min} est la valeur minimale du pas de descente.

```

Données:  $n, c, c_{max}, \alpha, \alpha_{min}$ 
Générer des paramètres initiaux  $p_{init}$  ;
 $f_{init} = f(p_{init})$  ;
 $p_{ok} = p_{init}$  ;
 $f_{ok} = f_{init}$  ;
 $c = 0$  ;
while  $\alpha > \alpha_{min}$  do
  while  $c < c_{max}$  do
    Générer de nouveaux coefficients  $p_{nouv}$  ;
    Calculer  $f_{nouv} = f(p_{nouv})$  ;
    if  $f_{nouv} > f_{ok}$  then
       $p_{ok} = p_{nouv}$  ;
       $f_{ok} = f_{nouv}$  ;
       $c = 1$  ;
    else
       $c = c + 1$  ;
    end
  end
   $c = 1$  ;
   $\alpha = \frac{\alpha}{2}$  ;
end
Résultat:  $f_{ok}$ 

```

Algorithme 1: Algorithme de Marche aléatoire

.1.2 Algorithme d'aplatissement des tenseurs

Cet algorithme est utilisé pour réaliser l'aplatissement des matrices nécessaire au calcul des invariants E_{moy} et multirang.

```

Données:  $n, T$ 
for  $1 \leq i \leq \frac{n}{2}$  do
  for  $l \in C_n^i$  do
    for  $0 \leq j < 2^n$  do
       $bin_j = \text{binaire}(j)$  sur  $n$  bits ;
       $ind = 0$  ;
      for  $0 \leq k < i$  do
        Calculer l'indice :  $ind = ind + bin_j[l[k] - 1]$  ;
        Placer  $T[j]$  à l'indice  $int(ind)$  de la matrice  $m$  ;
      end
    end
  end
  Rajouter  $m$  à la liste des matrices de coefficients  $M$  ;
end
Résultat:  $M$ 

```

Algorithme 2: Aplatissement du tenseur T

.2 Codes sur machine quantique en Qiskit

Dans cette section, sont regroupés les algorithmes qui ont été utilisés pour effectuer les calculs sur simulateur quantique et sur machine quantique.

.2.1 Algorithme de calcul des invariants μ et $\tilde{\mu}$ sur IBM-Q

Comme énoncé au Chapitre 3, les invariants μ et $\tilde{\mu}$ possèdent l'avantage d'être calculables sur une machine quantique. Nous avons réussi à les implémenter sur les machines quantiques d'IBM Quantum Experience suivant l'algorithme ci-dessous. Nous fournissons les codes pour le calcul de $\mu(|\psi\rangle)$. Ces codes sont également valables pour le calcul de $\tilde{\mu}(|\psi\rangle)$.

```
Données:  $n$ , coeffs
Calculer  $M_n$  ;
 $\mu = 0$  ;
for  $0 \leq i \leq 2^n$  do
    if  $M_n[i] \neq 0$  then
        Initialiser un circuit avec  $|\psi\rangle$  ;
        Calculer  $\theta, \phi, \lambda$  à partir de coeffs ;
        Placer les portes  $U(\theta, \phi, \lambda)$  ;
        Effectuer les mesures ;
        Calculer les fréquences  $freq$  ;
         $total = freq \times M_n[i]$  ;
         $\mu = \mu + total$  ;
    end
end
Résultat:  $\mu$ 
Algorithme 3: Evaluation des invariants  $\mu$  (et  $\tilde{\mu}$ ) sur IBM-Q
```

.2.2 Algorithme du calcul de la mesure géométrique de l'intrication

Le pseudo-code suivant explique comment est réalisé le calcul de la *MGI* d'un état quantique $|\psi\rangle$ sur une machine quantique.

```

Données:  $n, |\psi\rangle$ 
Générer aléatoirement des coefficients de départ coeffs ;
Initialiser un circuit avec  $|\psi\rangle$  ;
for  $0 \leq i \leq n$  do
    | circ.U(coeffs[0], coeffs[1], 0,  $q_i$ ) ;
    | circ.measure( $qr_i$ ,  $cr_i$ ) ;
end
Exécuter le circuit en un nombre  $nb_{essais}$  essais ;
Obtenir le dictionnaire dic qui en résulte ;
 $m = 0$  ;
Écrire  $0_n = 0$  sur  $n$  bits ;
if  $0_n \in dic$  then
    |  $m = dic[0_n]$  ;
else
    |  $m = 0$  ;
end
 $mgi = 1 - \frac{m}{nb_{essais}}$  ;
Résultat:  $mgi$ 
Algorithme 4: Pseudo-code Qiskit pour calculer la MGI sur IBM-Q

```

.2.3 Hypergraphes sur IBMQ

Les hyperarrêtes de taille 2 et 3 peuvent être créées à partir des portes $C-Z$ et C_2-Z (obtenues à partir de la porte de Toffoli). Pour les hyperarrêtes de taille supérieure ou égale à 4, il faut introduire des qubits auxiliaires. La dernière boucle **while** est nécessaire pour supprimer l'intrication introduite par les qubits auxiliaires du circuit principal. E est la liste des hyperarrêtes.

```

Données:  $n, E$ 
for  $0 \leq e \leq |E|$  do
   $N = n$  ;
  if  $|e| \geq 3$  then
     $N = 2n - 2$ 
  end
  Créer le circuit à partir de  $N$  qubits ;
end
for  $0 \leq i \leq n - 1$  do
   $\text{circ.h}[i]$ 
end
for  $0 \leq e \leq |E|$  do
  if  $|e| = 1$  then
     $\text{circ.CZ}(\text{h}[0])$ 
  end
  if  $|e| = 2$  then
     $\text{circ.CZ}(\text{h}[0], \text{h}[1])$ 
  end
  if  $|e| = 3$  then
     $\text{circ.h}[2]$  ;
     $\text{circ.toffoli}(\text{h}[0], \text{h}[1], \text{h}[2])$  ;
     $\text{circ.h}[2]$  ;
  end
  if  $|e| > 3$  then
     $\text{cpt} = 0$  ;
     $\text{max} = |e| - 1$ ;
     $\text{qubit\_aux} = n$ ;
     $\text{circ.toffoli}(\text{e}[\text{cpt}], \text{e}[\text{cpt} + 1], \text{qubit\_aux})$ ;
     $\text{cpt} = \text{cpt} + 2$ ;
    while  $\text{cpt} < \text{max}$  do
       $\text{circ.toffoli}(\text{e}[\text{cpt}], \text{qubit\_aux}, \text{qubit\_aux} + 1)$ ;
       $\text{qubit\_aux} = \text{qubit\_aux} + 1$ ;
       $\text{cpt} = \text{cpt} + 1$ ;
       $\text{circ.CZ}(\text{qubit\_aux}, \text{e}[\text{cpt}])$ ;
       $\text{qubit\_aux} = \text{qubit\_aux} - 1$ ;
       $\text{cpt} = \text{cpt} - 1$ ;
    end
    while  $\text{max} > 2$  do
       $\text{circ.toffoli}(\text{qubit\_aux} - 1, \text{e}[\text{cpt}], \text{qubit\_aux})$ ;
       $\text{qubit\_aux} = \text{qubit\_aux} - 1$ ;
       $\text{cpt} = \text{cpt} - 1$ ;
       $\text{max} = \text{max} - 1$ ;
    end
     $\text{circ.toffoli}(\text{e}[\text{cpt}], \text{e}[\text{cpt} - 1], \text{qubit\_aux})$ ;
  end
end

```

Résultat: *circ*

Algorithme 5: Pseudo-code Qiskit pour générer des hypergraphes

Articles

- Entanglement and Non-locality of four-qubit connected hypergraph states, G. Amouzou et al. Publié [4].
- Phase sensitivity of entanglement in the Quantum Phase Estimation Algorithm, G. Amouzou et al. Publié [3].

Bibliographie

- [1] A. D. Aczel. *Entanglement : the greatest mystery in physics*. Raincoast Books, 2002.
- [2] D. Alsina and J. I. Latorre. Experimental test of mermin inequalities on a five-qubit quantum computer. *Physical Review A*, 94(1) :012314, 2016.
- [3] G. Amouzou, K. Atchonouglo, and F. Holweck. Phase sensitivity of entanglement in the quantum phase estimation algorithm. *Physica Scripta*, 99(9) :095122, 2024.
- [4] G. Amouzou, J. Boffelli, H. Jaffali, K. Atchonouglo, and F. Holweck. Entanglement and nonlocality of four-qubit connected hypergraph states. *International Journal of Quantum Information*, 20(03) :2250001, 2022.
- [5] P. Arrighi and S. Perdrix. Modèles de calcul quantique. *Informatique mathématique*, 2016.
- [6] A. Aspect, P. Grangier, and G. Roger. Experimental realization of einstein-podolsky-rosen-bohm gedankenexperiment : A new violation of bell’s inequalities. *Physical review letters*, 49(2) :91, 1982.
- [7] S. Balakuntala and G. Paul. Quantum error correction using hypergraph states. *arXiv preprint arXiv :1708.03756*, 2017.
- [8] M. Bataille. Quantum circuits of cnot gates. *arXiv preprint arXiv :2009.13247*, 2020.
- [9] J. Batle, C. R. Ooi, A. Farouk, M. Alkhambashi, and S. Abdalla. Global versus local quantum correlations in the grover search algorithm. *Quantum Information Processing*, 15 :833–849, 2016.
- [10] J. S. Bell. On the einstein podolsky rosen paradox. *Physics Physique Fizika*, 1(3) :195, 1964.
- [11] I. Bengtsson and K. Życzkowski. *Geometry of quantum states : an introduction to quantum entanglement*. Cambridge university press, 2017.
- [12] J. L. Bönsel, O. Gühne, and A. Cabello. Generating multipartite nonlocality to benchmark quantum computers. *arXiv preprint arXiv :2406.07659*, 2024.
- [13] M. Boyer, G. Brassard, P. Høyer, and A. Tapp. Tight bounds on quantum searching. *Fortschritte der Physik : Progress of Physics*, 46(4-5) :493–505, 1998.
- [14] G. Brassard, I. Chuang, S. Lloyd, and C. Monroe. Quantum computing. *Proceedings of the National Academy of Sciences*, 95(19) :11032–11033, 1998.
- [15] R. Brooks. A robust layered control system for a mobile robot. *IEEE journal on robotics and automation*, 2(1) :14–23, 1986.
- [16] J.-L. Brylinski. Algebraic measures of entanglement. In *Mathematics of quantum computation*, pages 19–40. Chapman and Hall/CRC, 2002.
- [17] Y. Cao and A. M. Wang. Discussion of the entanglement classification of a 4-qubit pure state. *The European Physical Journal D*, 44 :159–166, 2007.

- [18] S. Chakraborty, S. Banerjee, S. Adhikari, and A. Kumar. Entanglement in the grover’s search algorithm. *arXiv preprint arXiv :1305.4454*, 2013.
- [19] X.-y. Chen and L. Wang. Locally inequivalent four-qubit hypergraph states. *Journal of Physics A : Mathematical and Theoretical*, 47(41) :415304, 2014.
- [20] A. Chiesa, F. Petiziol, M. Chizzini, P. Santini, and S. Carretta. Theoretical design of optimal molecular qudits for quantum error correction. *The Journal of Physical Chemistry Letters*, 13(28) :6468–6474, 2022.
- [21] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23 :880–884, Oct 1969.
- [22] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society of London. Series A : Mathematical, Physical and Engineering Sciences*, 454(1969) :339–354, 1998.
- [23] E. Curie. *Madame Curie*. Gallimard Paris, 1938.
- [24] H. de Boutray, H. Jaffali, F. Holweck, A. Giorgetti, and P.-A. Masson. Mermin polynomials for non-locality and entanglement detection in grover’s algorithm and quantum fourier transform. *Quantum Information Processing*, 20 :1–29, 2021.
- [25] E. Desurvire. *Classical and quantum information theory : an introduction for the telecom scientist*. Cambridge university press, 2009.
- [26] D. Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818) :97–117, 1985.
- [27] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A : Mathematical and Physical Sciences*, 439(1907) :553–558, 1992.
- [28] S. Dutta. A boolean functions theoretic approach to quantum hypergraph states and entanglement. *arXiv preprint arXiv :1811.00308*, 2018.
- [29] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical review*, 47(10) :777, 1935.
- [30] J. Endrejat and H. Büttner. Characterization of entanglement of more than two qubits with bell inequalities and global entanglement. *Physical Review A*, 71(1) :012305, 2005.
- [31] R. P. Feynman, A. R. Hibbs, and D. F. Styer. *Quantum mechanics and path integrals*. Courier Corporation, 2010.
- [32] M. Gachechiladze, C. Budroni, and O. Gühne. Extreme violation of local realism in quantum hypergraph states. *Physical review letters*, 116(7) :070401, 2016.
- [33] M. Gharahi and S. Mancini. Algebraic-geometric characterization of tripartite entanglement. *Physical Review A*, 104(4) :042402, 2021.
- [34] M. Ghio, D. Malpetti, M. Rossi, D. Bruß, and C. Macchiavello. Multipartite entanglement detection for hypergraph states. *Journal of Physics A : Mathematical and Theoretical*, 51(4) :045302, 2017.
- [35] F. Gieres. Formalisme de dirac et surprises mathématiques en mécanique quantique. *arXiv preprint quant-ph/9907070*, 1999.
- [36] N. Gisin. Bell’s inequality holds for all non-product states. *Physics Letters A*, 154(5-6) :201–202, 1991.

- [37] K. P. Gnatenko and N. Susulovska. Geometric measure of entanglement of multi-qubit graph states and its detection on a quantum computer. *Europhysics Letters*, 136(4) :40003, 2022.
- [38] D. J. Griffiths. *Introduction to Quantum Mechanics*. Cambridge University Press, 2nd edition, 2005.
- [39] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.
- [40] O. Gühne, M. Cuquet, F. E. Steinhoff, T. Moroder, M. Rossi, D. Bruß, B. Kraus, and C. Macchiavello. Entanglement and nonclassical properties of hypergraph states. *Journal of Physics A : Mathematical and Theoretical*, 47(33) :335303, 2014.
- [41] O. Gühne and G. Tóth. Entanglement detection. *Physics Reports*, 474(1-6) :1–75, 2009.
- [42] O. Gühne, M. Cuquet, F. E. S. Steinhof, T. Moroder, M. Rossi, D. Bru{\ss}, B. Kraus, and C. Macchiavello. Entanglement and nonclassical properties of hypergraph states. *Journal of Physics A : Mathematical and Theoretical*, 2014.
- [43] A. Hamad, S. F. Al-tikrity, and A. H. Mohammed. Quantum teleportation and entanglement of the 5-qubit state over a ghz-like channel. *Tikrit Journal of Pure Science*, 28(5) :65–73, 2023.
- [44] J. M. Henderson, E. R. Henderson, A. Sinha, M. A. Thornton, and D. M. Miller. Automated quantum oracle synthesis with a minimal number of qubits. In *Quantum Information Science, Sensing, and Computation XV*, volume 12517, pages 50–67. SPIE, 2023.
- [45] F. Holweck and H. Jaffali. Three-qutrit entanglement and simple singularities. *Journal of Physics A : Mathematical and Theoretical*, 49(46) :465301, 2016.
- [46] M. Horodecki. Measures of entanglement. *Quantum Information and Computation*, 1(1) :3–26, 2001.
- [47] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki. Quantum entanglement. *Reviews of modern physics*, 81(2) :865, 2009.
- [48] IBM. Ibm quantum platform, 2024.
- [49] H. Jaffali. *Étude de l’Intrication dans les Algorithmes Quantiques : Approche Géométrique et Outils Dérivés*. PhD thesis, Bourgogne Franche-Comté, 2020.
- [50] H. Jaffali and F. Holweck. Quantum entanglement involved in grover’s and shor’s algorithms : the four-qubit case. *Quantum Information Processing*, 18 :1–41, 2019.
- [51] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, et al. Quantum computing with qiskit. *arXiv preprint arXiv :2405.08810*, 2024.
- [52] N. R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous agents and multi-agent systems*, 1 :7–38, 1998.
- [53] F. Kabanza, M. Barbeau, and R. St-Denis. Planning control rules for reactive agents. *Artificial Intelligence*, 95(1) :67–113, 1997.
- [54] A. Koukam, A. Abbas-Turki, V. Hilaire, and Y. Ruichek. Towards a quantum modeling approach to reactive agents. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 130–136. IEEE, 2021.

- [55] B. Kraus. Local unitary equivalence and entanglement of multipartite pure states. *Physical Review A*, 82(3) :032121, 2010.
- [56] C. Kruszynska and B. Kraus. Local entanglability and multipartite entanglement. *Physical Review A*, May 2009.
- [57] J. M. Landsberg. *Tensors : geometry and applications : geometry and applications*, volume 128. American Mathematical Soc., 2011.
- [58] M. S. Leifer, N. Linden, and A. Winter. Measuring polynomial invariants of multi-party quantum states. *Physical Review A*, 69(5) :052304, 2004.
- [59] J. T. Z. Liu and H. Chen. Entanglement in shor’s factoring algorithm. *academia*, 2020.
- [60] X.-Y. Liu and Z. Zhang. Classical simulation of quantum circuits : Parallel environments and benchmark. In *Advances in Neural Information Processing Systems*, 2023.
- [61] S. Mahanti, S. Das, B. K. Behera, and P. K. Panigrahi. Quantum robots can fly ; play games : an ibm quantum experience. *Quantum Information Processing*, 18 :1–10, 2019.
- [62] N. D. Mermin. Extreme quantum entanglement in a superposition of macroscopically distinct states. *Physical Review Letters*, 65(15) :1838, 1990.
- [63] Microsoft. Quantum oracles. <https://learn.microsoft.com/en-us/azure/quantum/concepts-oracles>, 2024. Accessed : 2024-05-07.
- [64] R. Motwani. *Randomized Algorithms*. Cambridge University Press, 1995.
- [65] A. Muthukrishnan and C. R. Stroud. Qudits and high-dimensional quantum computing. *Frontiers in Physics*, 9 :108, 2021.
- [66] M. A. Nielsen and I. Chuang. *Quantum computation and quantum information*, 2002.
- [67] B. Ömer. Classical concepts in quantum programming. *International Journal of Theoretical Physics*, 44 :943–955, 2005.
- [68] R. Peter Bonasso, R. James Firby, E. Gat, D. Kortenkamp, D. P. Miller, and M. G. Slack. Experiences with an architecture for intelligent, reactive agents. *Journal of Experimental & Theoretical Artificial Intelligence*, 9(2-3) :237–256, 1997.
- [69] R. Qu, Z.-s. Li, J. Wang, and Y.-r. Bao. Multipartite entanglement and hypergraph states of three qubits. *Physical Review A*, 87(3) :032329, 2013.
- [70] A. Raghuvanshi, Y. Fan, M. Woyke, and M. Perkowski. Quantum robots for teenagers. In *37th International Symposium on Multiple-Valued Logic (ISMVL’07)*, pages 18–18. IEEE, 2007.
- [71] M. Rossi, D. Bruß, and C. Macchiavello. Scale invariance of entanglement dynamics in grover’s quantum search algorithm. *Physical Review A*, 87(2) :022331, 2013.
- [72] M. Rossi, M. Huber, D. Bruß, and C. Macchiavello. Quantum hypergraph states. *New Journal of Physics*, Nov. 2013.
- [73] D. Schlingemann and R. F. Werner. Quantum error-correcting codes associated with graphs. *Physical Review A*, 65(1) :012308, 2001.
- [74] R. Shankar. *Principles of quantum mechanics*. Springer Science & Business Media, 2012.
- [75] Y. Shimoni, D. Shapira, and O. Biham. Entangled quantum states generated by shor’s factoring algorithm. *Physical Review A*, 72(6) :062308, 2005.

- [76] A. Shimony. Degree of entanglement a. *Annals of the New York Academy of Sciences*, 755(1) :675–679, 1995.
- [77] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2) :303–332, 1999.
- [78] N. Susulovska and K. P. Gnatenko. Quantifying geometric measure of entanglement of multi-qubit graph states on the ibm’s quantum computer. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 465–466. IEEE, 2021.
- [79] E. D. Taillard. Parallel taboo search techniques for the job shop scheduling problem. *ORSA journal on Computing*, 6(2) :108–117, 1994.
- [80] J. Tan, Z. Liu, and H. Chen. Entanglement in phase estimation algorithm and quantum counting algorithm. *International Journal of Theoretical Physics*, 59 :1372–1381, 2020.
- [81] I. Tsogbetse. *Etude de codages et voisinages d’un espace de recherche. Application à l’ordonnancement de tâches dans des cas contraints*. PhD thesis, Université Bourgogne Franche-Comté, 2024.
- [82] V. Vedral and M. B. Plenio. Entanglement measures and purification procedures. *Physical Review A*, 57(3) :1619, 1998.
- [83] V. Vedral, M. B. Plenio, M. A. Rippin, and P. L. Knight. Quantifying entanglement. *Physical Review Letters*, 78(12) :2275–2279, 1997.
- [84] R. Wang. Comparing grover’s quantum search algorithm with classical algorithm on solving satisfiability problem. In *2021 IEEE Integrated STEM Education Conference (ISEC)*, pages 204–204. IEEE, 2021.
- [85] X. Wang et al. *Quantum algorithms for tensor decompositions and machine learning*. PhD thesis, Hong Kong Polytechnic University, 2021.
- [86] Y. Wang, Z. Hu, B. C. Sanders, and S. Kais. Qudits and high-dimensional quantum computing. *Frontiers in Physics*, 8 :589504, 2020.
- [87] T.-C. Wei and P. M. Goldbart. Geometric measure of entanglement and applications to bipartite and multipartite quantum states. *Physical Review A*, 68(4) :042307, 2003.
- [88] R. F. Werner. Quantum states with einstein-podolsky-rosen correlations admitting a hidden-variable model. *Physical Review A*, 40(8) :4277, 1989.
- [89] F.-L. Xiong, Y.-Z. Zhen, W.-F. Cao, K. Chen, and Z.-B. Chen. Qudit hypergraph states and their properties. *Physical Review A*, Jan. 2018.
- [90] N. S. Yanofsky and M. A. Mannucci. *Quantum computing for computer scientists*. Cambridge University Press, 2008.
- [91] D. Yao-Min and C. Ya. Lu invariants and canonical forms and slocc classification of pure 3-qubit states. *Communications in Theoretical Physics*, 45(4) :596, 2006.
- [92] S. Yu, Z.-B. Chen, J.-W. Pan, and Y.-D. Zhang. Classifying n-qubit entanglement via bell’s inequalities. *Physical review letters*, 90(8) :080401, 2003.
- [93] C. Zalka. Grover’s quantum searching algorithm is optimal. *Physical Review A*, 60(4) :2746, 1999.
- [94] S. Zangi, J.-L. Li, and C.-F. Qiao. Entanglement classification of four-partite states under the slocc. *Journal of Physics A : Mathematical and Theoretical*, 50(32) :325301, 2017.

- [95] N. Zettili. *Quantum Mechanics : Concepts and Applications*. Wiley, 2009.
- [96] C. Zhao, G.-w. Yang, W. N. Hung, and X.-y. Li. A multipartite entanglement measure based on coefficient matrices. *Quantum Information Processing*, 14 :2861–2881, 2015.

