Université de Montréal

Informatique Quantique : Algorithmes et Complexité de la Communication

par

Alain Tapp

Département d'Informatique et de Recherche Opérationnelle
Faculté des Arts et des Sciences

Thèse présentée à la Faculté des études supérieures
en vue de l'obtention du grade de
Philosophiæ Doctor (Ph. D.)
en Informatique

Août 1999

Université de Montréal

Faculté des études supérieures

Cette thèse intitulée

Informatique quantique : Algorithmes et Complexitée de la Communication

présentée par :

Alain Tapp

a été évaluée par un jury composé des personnes suivantes :

_____

(président-rapporteur)

Michel Boyer

_____

(directeur de recherche)

Gilles Brassard

_____

(codirecteur de recherche)

Pierre Mckenzie

_____

(membre du jury)

Claude Crépeau

_____

(examinateur externe)

Miklos Santha

Thèse acceptée le : _____

Sommaire

L'informatique quantique est un domaine jeune qui, fondant le calcul sur des propriétés quantiques de la matière, tente d'apporter des solutions novatrices à différents types de problèmes informatiques. Dans cette thèse qui regroupe cinq publications, nous nous penchons sur deux facettes de ce domaine, soit les algorithmes et la complexité de la communication quantiques.

Dans une première partie, regroupant les chapitres 1 à 5, nous nous intéressons à la conception d'algorithmes pour l'ordinateur quantique. Dans le chapitre 1, nous faisons l'historique de la théorie des calculs réversibles pour aboutir à celle de l'ordinateur quantique à proprement parler. Dans le chapitre 2, nous présentons un modèle simple, mais suffisamment élaboré, qui servira de cadre à la description d'algorithmes quantiques. Le chapitre 3 est une introduction et un résumé des deux publications traitant d'algorithmes quantiques que nous avons choisi d'inclure. La première publication incluse dans cette thèse, « Tight Bounds on Quantum Searching » [25, chapitre 4 ici], présente une analyse de l'algorithme de recherche combinatoire dit algorithme de Grover [69]. Soit une fonction $F : X \to \{0, 1\}$ où $|X| = N$ pour laquelle le seul moyen d'en obtenir de l'information est de l'évaluer aux entrées de notre choix et soit $t = |\{x|F(x) = 1\}|$. En 1996, Lov K. Grover démontra l'existence d'un algorithme quantique de recherche capable de trouver x_0 tel que $F(x_0) = 1$ avec bonne probabilité. Ledit algorithme nécessite seulement $O(\sqrt{N})$ évaluations de F s'il est promis que $t = 1$. Notez que de façon classique, $N/2$ évaluations de F sont nécessaires ne serait-ce que pour obtenir une probabilité de succès de $1/2$. A l'aide d'une analyse fine de

sa technique, il nous a été possible de spécifier concrètement cet algorithme, mais surtout de le généraliser au cas où t est quelconque et inconnu.

Dans la deuxième publication, « Quantum Amplitude Amplification and Estimation » [34, chapitre 5 ici], une généralisation de l'algorithme de recherche en termes d'amplification d'amplitude est présentée. Il arrive parfois que la structure du problème de recherche suggère une heuristique permettant un algorithme de recherche beaucoup plus efficace que la recherche exhaustive. Nous démontrons donc que, pour une famille importante d'heuristiques, l'algorithme de recherche peut être adapté de façon à obtenir le même gain quadratique obtenu précédemment. Finalement, nous présentons une technique d'évaluation d'amplitude qui nous servira à obtenir plusieurs algorithmes permettant l'évaluation de t avec diverses précisions. Grossièrement, nous montrons comment obtenir un estimé $\tilde{t}$ de t tel que (1) $|t - \tilde{t}| \leq \sqrt{t}$ nécessitant $O(\sqrt{N})$ évaluations de F, (2) $|t - \tilde{t}| \leq \epsilon t$ nécessitant $O(\sqrt{\frac{N}{t}})$[1] évaluations de F (3), $\tilde{t} = t$ nécessitant $O(\sqrt{tN})$ évaluations de F. Inutile de dire que de telles performances ne sauraient être obtenues classiquement lorsque la structure de F est inaccessible.

Dans la deuxième partie de cette thèse, regroupant les chapitres 6 à 9, nous étudions les propriétés de l'intrication comme ressource en complexité de communication. Dans la première section du chapitre 6, nous présentons de façon historique la découverte de cette ressource extraordinaire. On dira que deux (ou plusieurs) systèmes sont intriqués si leur état ne peut être représenté par un produit de deux (ou plusieurs) états disjoints. Une manifestation importante de l'intrication est le fait qu'une mesure effectuée sur l'un de deux systèmes intriqués semble affecter l'autre système instantanément et ce, sans égard à la distance qui les sépare. On quantifie l'intrication entre deux systèmes en termes de bits d'intrication. La deuxième partie du chapitre 6 se penche sur un exemple concret d'intrication qui viole toute théorie locale de l'univers. Finalement, la dernière partie du chapitre 6 prépare à la lecture des trois publications de cette seconde

[1] Dans ce sommaire ainsi que dans les chapitres d'introduction, nous avons choisis d'ignorer le cas $t = 0$ dans le but de rendre plus lisibles les notations d'ordre.

partie de la présente thèse. La complexité de communication d'une fonction booléenne G sur k variables est la quantité minimale de communication nécessaire, dans le pire cas, pour que chacun de k participants, connaissant la valeur d'une variable, apprenne la valeur de G. Dans [48], une variante de cette mesure où l'on permet aux participants de partager de l'intrication est présentée. Dans la publication « Multiparty Quantum Communication Complexity » [41, chapitre 7 ici], nous présentons le premier exemple de fonction pour laquelle le partage d'intrication permet une diminution non constante de communication, à savoir un facteur $\log(k)$. Dans la publication « The Cost of Exactly Simulating Quantum Entanglement with Classical Communication » [30, chapitre 8 ici] nous démontrons qu'une quantité exponentielle de communication est nécessaire pour parfaitement simuler un nombre linéaire de bits d'intrication. Nous démontrons aussi que, dans le cas d'un seul bit d'intrication, 8 bits de communication sont suffisants pour obtenir une simulation parfaite de la corrélation obtenue par des mesures complètes. Finalement, nous avons étudié les limites de l'intrication. Dans la publication « Quantum Entanglement and the Communication Complexity of the Inner Product Function » [50, chapitre 9 ici] nous montrons qu'il existe une fonction pour laquelle l'intrication ne permet aucune réduction de la complexité de communication.

Mots clés : ordinateur quantique, algorithme de Grover, communication, complexité, intrication.

Remerciements

La rédaction d'une thèse n'est pas un travail sans embûches, l'entreprise n'a pas toujours été facile et c'est pourquoi je désire remercier tous ceux grâce à qui ce rêve est devenu réalité.

Je voudrais remercier Gilles Brassard pour sa science, son inébranlable confiance et ses encouragements constants, Pierre McKenzie pour son enseignement et pour m'avoir fait découvrir d'autres facettes de l'informatique ainsi que Claude Crépeau pour ses encouragements et pour m'avoir permis de garder contact avec la cryptographie autant à travers lui qu'à travers ses brillants étudiants. Merci à Michael Frank, Michael Nielsen, Harry Buhrman, Wim van Dam, Ronald De Wolf, Michele Mosca et Dominic Mayers qui à travers mes voyages, rencontres et collaborations m'ont beaucoup appris. En particulier, merci à Richard Cleve qui m'a plongé dans ce domaine nouveau et excitant qu'est la complexité de communication. Merci aux membres de mon jury pour leur lecture attentive de ma thèse et leurs commentaires pertinents.

Merci aux membres et visiteurs du LITQ et en particulier Julien Marcil, Jean-François Blanchette, Peter Høyer, Tal Mor et Paul Dumais. Finalement, un merci tout spécial à mon père et mon frère.

Table des matières

Chapitre 1

Histoire de l'ordinateur quantique

L'histoire des machines à calculer, ou si vous préférez, de l'ordinateur, longe depuis toujours la frontière des mathématiques et de la physique. Plusieurs des plus grandes découvertes dans ce domaine ont été faites par des mathématiciens se questionnant sur la vraie nature du calcul. Que l'on parle de Blaise Pascal, Alan Turing ou Johannes von Neumann, c'est de la rencontre entre logique et physique que l'ordinateur moderne est né.

Il est surprenant de constater que tous les ordinateurs existants sont construits suivant le même paradigme. La seule variation appréciable de ce modèle se trouve dans le parallélisme, mais les opérations élémentaires sont toujours des ET, OU et NON, encadrés par une architecture qui fondamentalement varie peu. Mais, pensez-vous, comment pourrait-il en être autrement ?

Dans les prochains chapitres, nous présenterons une alternative, un modèle de calcul où les opérations élémentaires sont fort différentes et qui pourtant est relié intimement à la réalité physique. Le modèle en question est celui du *calcul quantique*, modèle qui emprunte au *calcul réversible*. Le reste de ce chapitre est donc divisé en deux parties dans lesquelles nous discuterons de leurs historiques respectifs.

1.1 Calculs réversibles

Nous débuterons notre histoire du calcul réversible en 1871 par un passage de *Theory of Heat* écrit par James C. Maxwell : [91]

> One of the best established facts in thermodynamics is that it is impossible in a system enclosed in an envelope which permits neither change of volume nor passage of heat, and in which both the temperature and pressure are everywhere the same, to produce any inequality of temperature or pressure without the expenditure of work. This is the second law of thermodynamics, and it is undoubtedly true as long as we can deal with bodies only in mass, and have no power of perceiving or handling the separate molecules of which they are made up. But if we conceive a being whose faculties are so sharpened that he can follow every molecule in its course, such a being, whose attributes are still as essentially finite as our own, would be able to do what is at present impossible to us. For we have seen that the molecules in a vessel full of air at uniform temperature are moving with velocities by no means uniform, though the mean velocity of any great number of them arbitrarily selected is almost exactly uniform. Now let us suppose that such a vessel is divided into two portions, A and B, by a division in which there is a small hole, and that a being, who can see the individual molecules, opens and closes this hole, so as to allow only the swifter molecules to pass from A to B, and only the slower ones to pass from B to A. He will thus, without expenditure of work, raise the temperature of B and lower that of A, in contradiction to the second law of thermodynamics.

Il semblerait qu'un petit démon pourrait par sa perfidie, transgresser la deuxième loi de la thermodynamique. Il existe plusieurs formulations de cette deuxième loi. L'impossibilité de créer un mouvement perpétuel en est une éloquente. Ce para-

doxe du *démon de Maxwell* a préocupé les physiciens pendant près d'un siècle. Maxwell lui-même n'offra pas de solution satisfaisante au problème, se contentant d'affirmer qu'il nous est impossible de voir et manipuler les molécules individuellement. En 1912, Marian v. Smoluchowski [110] démontra que le démon ne pouvait être incarné par une machine simple, comme une porte à ressort, car son propre bruit thermal l'empêcherait, à la longue, de fonctionner convenablement.

En 1929, Leo Szilard croyait avoir solutionné le problème [107]. Il prétendait que toute mesure ou acquisition d'informations au sujet des molécules par le démon devait se traduire par une augmentation d'entropie proportionnelle à celle possiblement obtenue par l'utilisation future de cette information. Malheureusement, Szilard avait tort, ce n'est pas l'acte de mesure qui doit augmenter l'entropie mais bien les opérations irréversibles comme l'effacement de la mémoire du démon pour faire place à une nouvelle mesure. Dès les années 1950, on considérait le calcul comme un processus mécanique et déjà on se questionnait sur les liens entre informations et entropie. Von Neumann affirmait en 1949 que chaque opération élémentaire de calcul devait dissiper une énergie équivalente à $kT \log 2$ où k est la constante de Boltzmann et T la température.

Il fallut attendre 1961 pour que R. Landauer [84] montre clairement que toute opération irréversible doit être accompagnée d'une augmentation d'entropie dans l'environnement. Il en conclut lui aussi, de façon erronée, que tout calcul dissipe de l'énergie, puisque nécessairement constitué d'opérations irréversibles.

C'est Charles H. Bennett qui en 1982 fit le lien avec le paradoxe de Maxwell et le coût associé aux calculs irréversibles. Il donna donc la première solution satisfaisante au paradoxe de Maxwell. C'est-à-dire que le démon doit oublier, effacer sa mémoire régulièrement, pour pouvoir continuellement prendre des nouvelles mesures et ainsi séparer les molécules lentes des rapides. Bien entendu, tous ces *oublis*, opérations fondamentalement irréversibles, s'accompagneront d'une augmentation d'entropie supérieure à la diminution créée par le jeu du démon. Les résultats de Bennett furent le réel coup d'envoi du calcul réversible. En effet il

démontra [13] que tout calcul peut être effectué en utilisant uniquement des opérations élémentaires réversibles, ce qu'il fit en introduisant la machine de Turing réversible et en montrant son universalité. La principale conséquence de ce résultat est que tout calcul peut être effectué en dissipant une quantité arbitrairement faible d'énergie. Malheureusement, pour effectuer des calculs de façon réversible, il y a un coût. Dans la première construction de Bennett [13], le coût à payer est en terme d'espace mémoire. Il publia finalement en 1989 un article [15] où il énonçait un des algorithmes les plus importants du domaine. Il démontra comment on pouvait faire un compromis temps/espace pour obtenir une solution préservant par exemple le temps polynomial et n'ajoutant qu'un facteur logarithmique à l'espace. Nous reviendrons à ces résultats de façon plus précise dans le deuxième chapitre.

Les physiciens ayant solutionné ce problème, ils se penchèrent sur d'autres. En fait, ce n'est plus la thermodynamique qui les préoccupe mais la mécanique quantique ; laissons cela de côté jusqu'à la prochaine section. Du côté de la réversibilité, deux catégories de chercheurs se sont emparés du domaine. Premièrement, les ingénieurs qui travaillent à la construction de machines réversibles. N'oublions pas que la dissipation d'énergie se manifeste par un échauffement des circuits, ce qui peut éventuellement imposer une limite à leur taille et leurs performances. Nous nous trouvons plutôt dans la deuxième catégorie, celle des théoriciens qui s'intéressent à ce modèle de facon abstraite. Depuis les publications de Bennett, peu de résultats théoriques importants ont été obtenus et ils concernent en général le compromis temps/espace dont j'ai parlé plus tôt [89]. En particulier, il a été démontré par K. Lange, P. McKenzie et A. Tapp qu'il est possible de simuler une machine de Turing par une machine de Turing réversible sans aucune perte au niveau de l'espace si l'on est prêt à payer le prix en termes de temps [86, 85].

Pour un supplément d'informations, nous suggérons fortement la lecture de l'article *Notes on the history of reversible computation* [14] ou de celui paru dans le *Scientific American* [18]. Notons aussi le livre *Maxwell's demon : entropy, in-*

formation, computing [87] qui place sous une même couverture tous les articles importants concernant le démon de Maxwell ainsi qu'une introduction instructive et une bibliographie imposante.

1.2 Informatique quantique

> *Where a calculator on the Eniac is equipped with 18000 vacuum tubes and weighs 30 tons, computers in the future may have only 1000 tubes and weigh only 1 1/2 tons*
>
> Extrait de l'édition de mars 1949 du magazine *Popular Mechanics*

En 1900 Max Planck solutionne le problème de la radiation des corps noirs, en 1905 Albert Einstein explique l'effet photoélectrique et en 1913 Niels Bohr énonce le premier modèle de l'atome d'hydrogène expliquant ses raies spectrales caractéristiques. Ces trois découvertes majeures ont en commun l'introduction de valeurs discrètes, appellés aujourd'hui quanta, là où on aurait normalement utilisé des valeurs continues. C'était la naissance de la mécanique quantique, science qui est rapidement devenue la théorie fondamentale de l'étude des phénomènes atomiques et sub-atomiques.

L'histoire de l'informatique quantique est plus récente que celle du calcul réversible. Pouvons-nous vraiment séparer ces deux domaines ? En mécanique quantique, on décrit les *objets* par leur fonction d'onde, leur évolution dans le temps par un hamiltonien et les mesures pouvant être effectuées, par un opérateur hermitien. Il est intéressant de remarquer que l'évolution dans le temps décrite par un hamiltonien est intrinsèquement réversible (unitaire) d'où le lien étroit entre la dynamique des processus réversibles et celle des processus quantiques. Ce lien sera d'ailleurs plus apparent dans le prochain chapitre lorsque les circuits réversibles et quantiques seront présentés.

Notre monde est fondamentalement quantique ; si l'on désire un jour effectuer des calculs à une échelle atomique, il faudra tenir compte de ces règles, il faudra

tenir compte des effets quantiques. Notez que ces effets quantiques sont pour le moins *contre nature*, pensons au paradoxe du chat de Schrödinger. Pourquoi ne pas utiliser cette bizarrerie pour faire *mieux*, calculer plus vite ? Pourquoi ne pas utiliser des opérations élémentaires quantiques dans le cadre d'un processus qui aura pour but la solution de problèmes calculatoires ?

En 1982 parut un article de Richard Feynman qui s'interrogeait sur la simulation de systèmes quantiques par d'autres systèmes quantiques [63]. Il faut dire que la mécanique quantique est ainsi faite qu'il est difficile pour un ordinateur classique de la simuler. C'est en 1985 que l'informatique quantique est réellement née avec un article de David Deutsch [55] qui contenait la description d'un ordinateur quantique universel. À ce moment, le seul intérêt connu de cet ordinateur résidait dans sa capacité de simuler des systèmes quantiques efficacement, exponentiellement plus efficacement que les meilleurs algorithmes roulant sur un ordinateur classique. Il faut attendre en 1991 pour voir le premier résultat laissant croire que l'ordinateur quantique puisse être utilisé pour accélérer des calculs utiles. C'est David Deutsch et Richard Jozsa [58] qui énonceront ce premier algorithme quantique. Malheureusement, le problème étudié n'est pas naturel et de plus il peut être solutionné efficacement par un ordinateur classique de façon probabiliste.

L'année 1994 est sûrement la plus marquante dans ce domaine florissant car deux articles fondamentaux furent publiés, deux articles qui catapulteront la discipline à l'avant-plan de la recherche en physique et en informatique théorique. Le premier article est de Daniel Simon [102]. Il montre comment trouver des collisions dans une classe bien spécifique de fonctions. L'algorithme fonctionne en temps espéré polynomial sur un ordinateur quantique et on peut montrer que, formulé de cette façon, aucun ordinateur classique ne pourrait le résoudre efficacement, même de façon probabiliste. Bien que beaucoup plus intéressant et naturel que le problème de Deutsch, il n'en demeure pas moins dénué d'intérêt pratique. Le deuxième article, publié à la même conférence et fortement inspiré des résultats de Simon, nous vient de Peter Shor [100, 101]. Ce dernier montre comment fac-

toriser de grands entiers et extraire le logarithme discret. Inutile de rappeler que la factorisation de grands nombres est considérée comme un problème difficile ; cette croyance est suffisamment établie pour que la technique d'encryption la plus utilisée sur l'internet repose sur cette hypothèse. Les informaticiens quantiques jubilent ! Les applications pratiques de ce résultat sont indéniables puisque si un ordinateur quantique existait, cela remettrait en question toute la cryptographie telle que nous la connaissons [26]. Ce fut aussi le moment d'un radical changement d'attitude ; le scepticisme quant à l'intérêt de ces machines se transforma en scepticisme quant à leur faisabilité.

À partir de ce moment, les publications touchant l'ordinateur quantique se multiplient, mais après les résultats de Simon et de Shor, un seul autre algorithme important fut proposé. Cet algorithme ne permet pas comme les deux autres de résoudre en temps polynomial un problème qui jusqu'alors nécessitait un temps exponentiel mais son domaine d'application est extrêmement vaste. En effet, en 1996, Lov K. Grover publia un article fort important intitulé « A fast quantum mechanical algorithm for database search » [69]. Dans cet article il donne *l'esquisse* d'un algorithme permettant de résoudre des problèmes de recherche combinatoire, c'est à dire, trouver dans un très large espace de recherche une solution à des contraintes spécifiques. Les problèmes dans la classe de complexité **NP** sont directement touchés par cet algorithme. Bien des choses encore peuvent être écrites au sujet de l'article de Grover et une bonne partie de cette thèse y est consacrée.

Parallèlement à ce que nous appellerons l'algorithmique quantique s'est développée une science de l'information quantique. La téléportation [17, 36, 37] et la cryptographie [32] peuvent être classées dans cette catégorie. On étudie entre autres la relation entre l'information accessible au sujet d'un système quantique et la perturbation qu'on lui fait subir. Les études concernant la communication à travers un canal quantique auront énormément d'impact. En effet, il est possible d'encoder de l'information quantique de façon à pouvoir corriger d'éven-

tuelles erreurs [42, 104, 98], ce qui est surprenant étant donné la nature continue des erreurs et la nature discrète de leurs corrections. Les résultats concernant la correction d'erreur combinés avec ceux concernant les portes logiques quantiques [5, 56, 4, 59, 103] sont parmi les résultats théoriques les plus importants concernant la réalisation éventuelle d'un ordinateur quantique.

Qu'en est-il de la construction d'un prototype d'ordinateur quantique? La recherche pratique en est à ses balbutiements mais plusieurs laboratoires dans le monde y travaillent. Citons par example le California Institute of Technology [1], le National Institute of Standards and Technology [2], l'université d'Oxford [3], et le laboratoire de recherche de Los Alamos [4]. Soyons humbles, ces laboratoires font des expériences pour réaliser les opérations élémentaires servant à la construction d'un ordinateur. C'est un peu comme si l'on avait le plan complet d'une pièce fine d'horlogerie qui vraisemblablement décrit une montre très précise mais que, pour le moment, le premier obstacle technologique soit la construction d'un engrenage ayant bien la taille et la finesse requises par le plan. Une fois la technique des engrenages maîtrisée, encore faut-il construire la petite montre! Voilà ce à quoi ressemble la recherche pratique en informatique quantique. La mécanique quantique, si stable et tant de fois porteuse de succès, ne nous permet pas de conclure à l'impossibilité de la construction de ces machines mais elle ne nous promet pas non plus la réussite du projet. Les avis quant à la faisabilité de cet ordinateur sont partagés. Plusieurs chercheurs pensent que la réalisation d'une telle machine est impossible. Il est vrai que des difficultés majeures attendent ceux qui tentent sa construction. Les arguments contre sa faisabilité sont si semblables à ceux entendus lors des tentatives de réalisation des premiers ordinateurs électroniques qu'on ne peut que sourire. De toute façon, si des lois physiques inconnues empêchent la construction de cette machine merveilleuse, rendant probablement du même coup la mécanique quantique obsolète, leur découverte sera d'une telle im-

[1] http ://www.cco.caltech.edu/~qoptics/
[2] http ://www.bldrdoc.gov/timefreq/ion/index.htm
[3] http ://www.qubit.org/
[4] http ://qso.lanl.gov/qc/

portance pour notre compréhension du monde qu'elles justifieront plusieurs fois toute la recherche qui est faite dans ce domaine.

Chapitre 2

Circuits et algorithmes quantiques

Dans le chapitre précédent, nous avons esquissé l'histoire du calcul quantique en passant par celle du calcul réversible. De la même façon, l'étude des circuits quantiques ne peut se faire sans d'abord passer par celle des circuits réversibles. Le chapitre sera donc divisé en trois parties, les circuits réversibles, les circuits quantiques et finalement nous consacrerons une section aux principaux algorithmes quantiques.

2.1 Circuits réversibles

Dans tous les modèles de calcul, il existe une notion de graphe de configuration, en général les noeuds représentent l'état global de la machine et un arc relie deux configurations si une opération élémentaire permet de passer de la première à la deuxième. L'état global ainsi qu'une étape élémentaire doivent être précisément définis pour chaque modèle mais l'idée intuitive que nous nous en faisons est suffisante ici. Voici trois graphes de configurations (Figure 2.1) qui nous permettront d'imager la notion de calcul réversible.

Le graphe d'une machine non-déterministe est pratiquement sans restrictions, les degrés entrant et sortant des configurations sont quelconques. Dans celui de la machine déterministe tous les noeuds de son graphe de configuration ont un degré

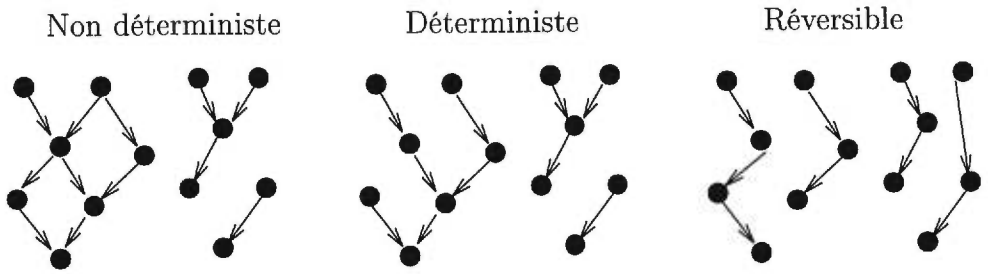

FIG. 2.1: Graphes de configurations

extérieur au plus un, ce qui signifie que le successeur de chaque configuration, s'il existe, est uniquement défini. Une configuration peut par contre avoir plusieurs prédécesseurs. Finalement, on dit d'une machine qu'elle est réversible si les degrés entrants et sortants sont au plus un, dans ce cas la machine est déterministe peu importe la direction de la flèche du temps. Le successeur et le prédécesseur de chaque configuration sont, s'ils existent, uniquement défini.

Il existe plusieurs modèles de calcul réversible, par exemple le modèle des balles de billard [65] où les opérations élémentaires sont des collisions parfaites entre des balles élastiques. Ces collisions sont réversibles et il n'est pas trop ardu de simuler des portes logiques par l'ajout d'obstacles fixes. Comme modèle théorique, nous avons, à l'image de la machine de Turing conventionnelle, la machine de Turing réversible [13]. Cette machine est très semblable à son homologue déterministe à l'exception de sa table de transition qui doit respecter certaines conditions de réversibilité. Un autre modèle qui nous intéresse particulièrement est celui du circuit réversible. C'est probablement le modèle de réversibilité le plus pratique.

Nous allons maintenant définir un modèle de circuit réversible et allons énoncer les principaux résultats connus à leur sujet. La majorité des faits présentés dans ce chapitre sont des conséquences simples de théorèmes déjà connus pour la machine de Turing réversible mais leur traduction dans le modèle des circuits réversibles est faite ici pour la première fois à notre connaissance.

Par souci de simplicité, nous nous concentrerons sur les circuits à valeur binaire même si tous les résultats qui suivent peuvent s'exprimer de façon plus géné-

rale dans un contexte où les fils transportent un symbole d'un alphabet plus large. Une porte réversible (r-porte) g d'ordre k calcule une fonction bijective $g : \{0,1\}^k \to \{0,1\}^k$. Un circuit réversible (r-circuit) est un graphe acyclique orienté où les noeuds internes sont des r-portes avec degré entrant et degré sortant égal à leur ordre. De plus, le graphe contient w noeuds initiaux et w noeuds finaux. Les w noeuds initiaux ont degré sortant 1 et degré entrant 0 (l'inverse pour les noeuds finaux). Dans l'évaluation d'un r-circuit, n noeuds de départ spéciaux sont initialisés avec la valeur de l'input tandis que les autres le sont à une valeur constante ne dépendant pas de l'input. De la même façon m noeuds finaux vont contenir la valeur de $F(x)$ de façon à ce que le circuit calcule la fonction $F : \{0,1\}^n \to \{0,1\}^m$. Définissons un autre ensemble de noeuds de sortie, soit tous les noeuds qui ne font pas partie de la valeur de $F(X)$ et qui ne sont pas constants par rapport à l'input ; nous appellerons ces bits des valeurs résiduelles. L'ensemble des bits résiduels peut lui aussi être considéré comme une fonction calculée par le circuit ; nous appellerons cette fonction calculée de façon fortuite la fonction parasite et la noterons $G : \{0,1\}^n \to \{0,1\}^g$. Dans le modèle réversible, la mémoire utilisée que l'on laisse dans une valeur inconnue (ici les valeurs résiduelles) est un problème puisqu'abandonner des valeurs inconnues est parfaitement équivalent à effacer ces valeurs. Schématiquement, nous avons

$$(x, c_1) \mapsto (G(x), F(x), c_2)$$

avec $|x| = n$, $|F(x)| = m$, $|G(x)| = g$ et $|x| + |c_1| = |G(x)| + |F(x)| + |c_2|$ où c_1 et c_2 sont des constantes indépendantes de x. L'ordre d'un r-circuit est le maximum de l'ordre de ses portes, sa taille est le nombre de portes qui le constituent, sa profondeur est définie par la longueur du chemin le plus long entre un noeud entrant et un noeud sortant et finalement sa largeur est définie de façon évidente par le nombre de fils.

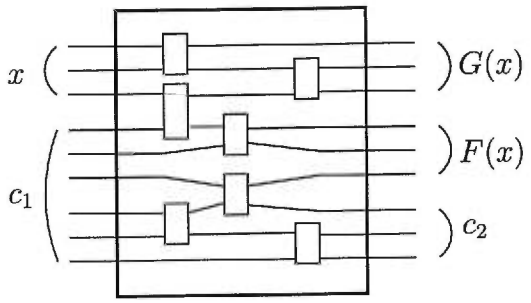

FIG. 2.2: Circuit Réversible

Pour les circuits classiques nous savons que le NAND $((x,y) \mapsto (1+(xy))$[1] est universel si l'on considère que le FAN-OUT et l'initialisation de constantes sont gratuits. En calculs réversibles, ce rôle est joué par la porte de Toffoli [109] qui est universelle si l'initialisation de constantes est permise. Cette r-porte d'ordre 3 effectue la transformation suivante :

$$(b_0, b_1, b_2) \mapsto (b_0, b_1, b_2 + (b_0 b_1)).$$

La porte de Toffoli peut être utilisée pour simuler le NOT, XOR, AND et le FAN–OUT en choisissant les bonnes constantes. En particulier, nous avons

$$\begin{array}{lll} \text{NOT :} & (b_0,1,1) & \mapsto (b_0,1,b_0+1) \\ \text{XOR :} & (b_0,1,b_1) & \mapsto (b_0,1,b_0+b_1) \\ \text{AND :} & (b_0,b_1,0) & \mapsto (b_0,b_1,b_0 b_1) \\ \text{FAN}-\text{OUT :} & (b_0,1,0) & \mapsto (b_0,1,b_0) \end{array}$$

Dans le reste de cette section, nous ne considérerons que les circuits classiques constitués uniquement de AND, NOT et de FAN-OUT de sortance 2. Notons qu'un FAN-OUT arbitraire peut être simulé par une cascade de FAN-OUT de sortance 2 et que même si cette restriction ne diminue pas en général le pouvoir d'expressivité des circuits, elle peut doubler leur taille et augmenter leur profondeur d'un facteur logarithmique par rapport à la largeur.

[1] On supposera par défaut que les opérations sur les bits sont effectuées modulo 2.

Théorème 2.1.1 (Bennett 73 [13]) *Toute fonction $F : \{0,1\}^n \to \{0,1\}^m$ calculable par un circuit de taille s, profondeur d et largeur w peut être calculée par un r-circuit d'ordre 3, de taille $s' = 2s + m$, largeur $w' \leq 2s + 2m$ et profondeur $2d + 1$ de telle sorte que $G(x) = x$.*

Idée de preuve. Il est facile de simuler un circuit classique en remplacant chaque porte logique par une porte de Toffoli branchée aux bonnes constantes. Malheureusement, cela produit en général une grande quantité de bits résiduels. La stratégie utilisée pour réduire leur nombre est simple, il suffit d'adapter la technique de Bennett [13] fondée sur l'utilisation d'un ruban qui conserve l'histoire des opérations effectuées par la machine. On calcule la fonction sans se soucier des bits résiduels, on copie la réponse en utilisant des portes de Toffoli comme FAN-OUT, finalement il suffit de décalculer la fonction, c'est à dire défaire toutes les opérations sauf la copie de la solution. ∎

Nous venons de voir que la réversibilité ne limite pas la puissance d'expressivité des circuits et qu'elle n'entraîne une augmentation de profondeur que d'un facteur deux. Le r-circuit ainsi obtenu est d'ordre 3, ce qui est optimal car on peut démontrer qu'avec un circuit d'ordre 2, seules des fonction linéaires peuvent être calculées. Il est intéressant de constater que cela est obtenu en produisant un nombre limité de bits résiduels avec la forme simple $(x, 0^m, c_1) \mapsto (x, F(x), c_1)$. L'augmentation de la largeur du r-circuit proportionnellement à sa taille est l'inconvénient majeur de cette technique simple. Heureusement la solution envisagée par Bennett pour régler le problème équivalent pour les machines de Turing réversibles s'applique encore une fois ici.

Théorème 2.1.2 (Bennett 89 [15], Levine & Sherman 90 [88]) *Toute fonction calculable par un circuit de taille s, largeur w et profondeur d peut être calculée par un r-circuit d'ordre 3, de taille $s' \in O(s^{1+\epsilon}/w^{\epsilon})$, de largeur $w' \in O(w(1 + \log(s/w))$ et de profondeur $d' \in O(d^{1+\epsilon}/w^{\epsilon})$ de telle sorte que $G(x) = x$ pour tout $\epsilon > 0$.*

Idée de preuve. La stratégie consiste à diviser le circuit en blocs de taille s et d'appliquer récursivement la technique du théorème 2.1.1 tel que décrit dans [15]. Des valeurs plus précises sont obtenues pour l'ordre en utilisant l'analyse plus fine de Levine & Sherman [88]. ∎

Le théorème 2.1.2 permet une réduction substantielle de la largeur du r-circuit qui calcule une fonction. Tous les résultats concernant les r-circuits présentés jusqu'à maintenant sont des corollaires de résultats équivalents concernant les machines de Turing réversibles ; ce ne sera pas le cas des théorèmes qui suivent.

Théorème 2.1.3 (Coppersmith & Grossman 75 [53]) *Toute fonction* $F : \{0,1\}^n \to \{0,1\}^m$ *peut être calculée par un r-circuit d'ordre 3 et de largeur* $n+m$ *de telle sorte que* $G(x) = x$.

Idée de preuve. Le théorème original de Coppersmith et Grossman annonce que *Toute permutation paire* $H : \{0,1\}^n \to \{0,1\}^n$ *peut être calculée par un r-circuit d'ordre 3 avec n fils.* Pour obtenir le théorème tel que formulé ici il suffit que $F : \{0,1\}^n \to \{0,1\}^m$ soit transformée dans une permutation paire de $2^{(n+m)}$ éléments calculant précisément $(x,0) \mapsto (x, f(x))$. ∎

Le théorème précédent est particulièrement intéressant quand $m = 1$ car autrement il est souvent possible de calculer la fonction avec moins de bits résiduels en étendant la fonction en une permutation paire de moins de $m + n$ bits.

Est-il possible de faire mieux ? Peut-on, en redéfinissant la largeur de façon appropriée, restreindre la largeur à une valeur moindre que la taille de l'input ? Avec les machines de Turing, il suffit d'utiliser un ruban accessible en lecture seulement, en plus d'un ruban de travail, pour pouvoir restreindre la mémoire à une valeur sous-linéaire. Avec les r-circuits on peut utiliser une astuce similaire. Nous exigerons des r-portes qui sont connectées aux fils reliés à l'entrée qu'elles ne modifient pas la valeur portée par ces fils. La largeur de tels circuits est définie par le nombre de fils qui ne sont pas reliés aux entrées et donc qui ne sont pas constantes. Nous allons maintenant nous concentrer sur les circuits qui reconnaissent des langages $(m = 1)$.

Théorème 2.1.4 (Barrington 86 [7]) *Tout langage peut être décidé par un r-circuit d'ordre 4 et de largeur* 3.

Idée de preuve. David A. Barrington a montré que tout langage peut être décidé par un programme de branchement réversible de largeur 3 (*width-3 Permutation Branching Program*) [7]. Une permutation de 3 éléments peut être représentée avec 3 bits. Il est facile de voir que des permutations conditionnelles dans un 3-PBP peuvent être simulées par une r-porte d'ordre 4 où chaque fil connecté à une entrée demeure inchangé. Le théorème s'ensuit directement. ■

Théorème 2.1.5 (Barrington [8, 7, 9]) *Tout langage dans* $\mathbf{NC}^1$ *peut être décidé par un r-circuit d'ordre 6, de taille polynomiale et de largeur* 5.

Idée de preuve. Rappelons que $\mathbf{NC}^1$ est la classe des langages reconnaissables par une famille de circuits de profondeur logarithmique. David A. Barrington a montré que tout langage dans $\mathbf{NC}^1$ peut être décidé par un programme de branchement réversible n'utilisant que des permutations cycliques d'ordre 5 (*width-5 PBP*) [9]. Une telle permutation peut être représentée avec 5 bits. Nous obtenons donc, comme précedemment, que des r-portes d'ordre 6 peuvent simuler ces permutations conditionnelles. Le théorème s'ensuit directement. ■

Nous sommes en droit de nous demander si le même genre de résultat est possible pour des circuits de taille polynomiale quelconque. La réponse est probablement non, mais cela risque de s'avérer difficile à démontrer puisqu'un tel résultat permettrait de séparer $\mathbf{P}$ de $\mathbf{NC}^1$.

2.2 Circuits quantiques

L'ordinateur quantique ayant été étudié depuis quelques années déjà, certaines avenues de recherche ont été éliminées pour faire place à d'autres. Pour le moment, la voie la plus prometteuse passe par les circuits quantiques. La partie de la mécanique quantique nécessaire à la compréhension de l'informatique quantique

peut être décrite en termes d'opérateur unitaire et de projection sur un espace de Hilbert de dimension finie. Nous n'avons nulle intention d'exposer ces principes dans toute leur généralité. Dans notre description du modèle, nous allons considérer que le lecteur n'est pas familier avec la mécanique quantique. Comme bien peu des concepts de cette science seront nécessaires pour la description du modèle, nous n'allons en définir qu'un petit sous-ensemble. Le lecteur qui s'intéresse à la mécanique quantique, soit de façon spécifique, soit pour comprendre la provenance de ce modèle de calcul, est invité à consulter [96, 77]. Pour des ouvrages plus orientés vers le calcul quantique nous conseillons [21, 27, 29]. Notez tout de même que le modèle choisi est robuste puisque toutes les tentatives d'implantation d'un ordinateur quantique s'y intègrent parfaitement. En particulier dans le cas des trappes à ions [78] ou de la résonance magnétique nucléaire [80, 79], il semble que la traduction entre les circuits que nous décrivons et l'implantation se fasse de façon naturelle.

Les circuits quantiques seront une extension à valeur *complexe* des circuits réversibles. Nos circuits ne traiteront plus de valeurs binaires mais plutôt de qubits et ils posséderont un type de portes supplémentaires entièrement quantiques, à savoir les portes unaires unitaires. Les circuits quantiques diffèrent des circuits réversibles principalement au niveau de la sémantique de leurs opérations. *La différence entre la réalité et la fiction est que la réalité n'a pas besoin d'avoir de sens.* Si le modèle quantique n'encapsulait pas une part de la réalité, il faudrait être pervers pour l'inventer.

Pour ceux qui sont à l'aise avec la mécanique quantique, nous allons essentiellement en exploiter deux principes fondamentaux soit la *superposition* et *l'interférence.* Le terme superposition traduit le fait qu'un objet quantique (une particule ou un registre) peut être à la fois dans plusieurs états et qu'à partir de ce moment toute opération effectuée sur lui s'effectue en parallèle sur chacun des états superposés. Il sera possible d'une certaine façon d'évaluer une fonction simultanément sur tous les points de son domaine. L'interférence est le fait que

si plusieurs *chemins* mènent au même état alors les *amplitudes* de ces différents chemins s'additionnent. Comme les amplitudes sont des nombres complexes elles pourront donc s'accumuler (interférence constructive) ou s'annihiler (interférence destructive). L'amplitude, terme que nous n'avons pas encore défini, peut être considérée comme une généralisation du concept de probabilité. Toute la beauté du calcul quantique sera de savoir combiner superposition et interférence pour favoriser les réponses correctes.

Tout au long de cette section, nous utiliserons en parallèle la notation matricielle et la notation dite de Dirac. Commençons par nous attarder à la notion de bits quantiques ainsi qu'aux registres quantiques qui en seront la généralisation. Contrairement aux bits classiques, les bits quantiques, appelés qubit, peuvent avoir à la fois la valeur 0 et 1. La description des qubits et des registres quantiques reflète parfaitement le concept de superposition. Un qubit est caractérisé par deux nombres complexes α et β représentant respectivement l'amplitude avec laquelle il est dans l'état 0 et l'amplitude avec laquelle il est dans l'état 1. Une restriction qui deviendra claire plus tard est que

$$|\alpha|^2 + |\beta|^2 = 1.$$

En notation de Dirac on représente le qubit

$$\alpha|0\rangle + \beta|1\rangle$$

où les drôles d'accolades entourant zéro et un sont appelées *kets* et servent à différencier les amplitudes α et β des états de base $|0\rangle$ et $|1\rangle$. On peut aussi utiliser la représentation matricielle

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

De façon générale, un registre quantique de k qubits est caractérisé par un vecteur de 2^n nombres complexes

$$\begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{2^n-1} \end{pmatrix} \quad \text{tel que} \quad \sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1.$$

On le note aussi

$$\sum_{i=0}^{2^n-1} \alpha_i |i\rangle$$

et chaque nombre complexe α_i représente l'amplitude de l'état de base $|i\rangle$. Lorsque l'on combine deux registres quantiques, on obtient un registre quantique où les amplitudes se combinent de la façon suivante :

$$\begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{2^n-1} \end{pmatrix} \otimes \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_{2^m-1} \end{pmatrix} = \begin{pmatrix} \alpha_0\beta_0 \\ \vdots \\ \alpha_0\beta_{2^m-1} \\ \alpha_1\beta_0 \\ \vdots \\ \alpha_{2^n-1}\beta_{2^m-1} \end{pmatrix}$$

et en notation de Dirac

$$\left(\sum_{x=0}^{2^n-1} \alpha_x |x\rangle\right)\left(\sum_{y=0}^{2^m-1} \beta_y |y\rangle\right) = \sum_{x=0,y=0}^{2^n-1,2^m-1} \alpha_x\beta_y |x : y\rangle$$

où « : » signifie la concaténation de chaînes de bits et $\otimes$ est le produit de Kronecker[2]. Notez que cette règle est valable uniquement pour la concaténation de registres et qu'elle ne saurait en général s'appliquer en sens inverse pour séparer un registre en deux. Plus spécifiquement, lorsque deux parties d'un registre ne

[2]Représentation matricielle du produit tensoriel.

peuvent s'écrire comme un produit on dira qu'elles sont *intriquées*. Nous reviendrons à la notion d'intrication dans la deuxième partie de cette thèse.

Dans notre modèle de circuits quantiques, nous permettons deux types de portes, les opérations unaires et le XOR quantique.

Les opérations unaires s'effectuent, comme leur nom l'indique, sur un seul qubit du registre. On peut caractériser ces opérations par une matrice unitaire 2×2, c'est-à-dire une matrice

$$U = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \text{ telle que } \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}^{\dagger} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

où α, β, γ et δ sont des nombres complexes et $\dagger$ représente l'opération de transposer et conjuguer une matrice. Lorsqu'on applique l'opération représentée par la matrice U à un qubit dans l'état $\begin{pmatrix} x_0 \\ x_1 \end{pmatrix}$, le registre passe simplement dans l'état

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} \alpha x_0 + \beta x_1 \\ \gamma x_0 + \delta x_1 \end{pmatrix}$$

Soit $\otimes$ le produit de Kronecker qui opère sur des matrices carrées de la façon suivante :

$$\begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1m} \\ \vdots & \ddots & \vdots \\ \alpha_{m1} & \cdots & \alpha_{mm} \end{pmatrix} \otimes \begin{pmatrix} \beta_{11} & \cdots & \beta_{1n} \\ \vdots & \ddots & \vdots \\ \beta_{n1} & \cdots & \beta_{nn} \end{pmatrix} =$$

$$\begin{pmatrix} \alpha_{11}\beta_{11} & \cdots & \alpha_{11}\beta_{1n} & & \alpha_{1m}\beta_{11} & \cdots & \alpha_{1m}\beta_{1n} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ \alpha_{11}\beta_{n1} & \cdots & \alpha_{11}\beta_{nn} & & \alpha_{1m}\beta_{n1} & \cdots & \alpha_{1m}\beta_{nn} \\ & \vdots & & \ddots & & \vdots & \\ \alpha_{m1}\beta_{11} & \cdots & \alpha_{m1}\beta_{1n} & & \alpha_{mm}\beta_{11} & \cdots & \alpha_{mm}\beta_{1n} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ \alpha_{m1}\beta_{n1} & \cdots & \alpha_{m1}\beta_{nn} & & \alpha_{mm}\beta_{n1} & \cdots & \alpha_{mm}\beta_{nn} \end{pmatrix}$$

L'application de U sur le i-ième qubit d'un registre de n qubits représenté par un vecteur v est v' tel que

$$v' = (I_{2^{i-1}} \otimes U \otimes I_{2^{n-i}})v$$

où I_k est la matrice identité dans l'espace à k dimensions. Notez que $\otimes$ est associatif. En notation de Dirac, l'effet sur des états de base est le suivant :

$$\begin{aligned} |0\rangle &\mapsto \alpha|0\rangle + \gamma|1\rangle \\ |1\rangle &\mapsto \beta|0\rangle + \delta|1\rangle. \end{aligned}$$

Une caractéristique importante des opérations unaires est leur linéarité : les amplitudes s'additionnent. C'est ce que l'on appelle l'interférence. L'application d'une opération unaire U sur le i-ième bit d'un registre dans l'état de base

$$|x\rangle = |x_1 x_2 \ldots x_n\rangle$$

donnera

$$|x_1 x_2 \ldots x_{i-1}\rangle(U|x_i\rangle)|x_{i+1} \ldots x_n\rangle$$

et donc son application sur le registre

$$\sum_{x=0}^{2^n-1} \alpha_x |x\rangle$$

donnera

$$\sum_{x=0}^{2^n-1} \alpha_x |x_1 x_2 \dots x_{i-1}\rangle (U|x_i\rangle)) |x_{i+1} \dots x_n\rangle$$

Par exemple l'application de

$$U = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

sur le deuxième qubit de l'état

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

donne

$$\left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \right) \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} =$$

$$\frac{1}{2} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Le même exemple en notation de Dirac est passablement plus lisible. L'opération unaire devient la transformation

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

L'état du registre peut maintenant s'écrire $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)$. L'application de l'opération unaire sur le registre nous donne donc

$$\begin{aligned} (I_2 \otimes U)|\psi\rangle &= (I_2 \otimes U)\left(\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)\right) \\ &= \frac{1}{\sqrt{2}}(|0\rangle(U|0\rangle) + |0\rangle(U|1\rangle)) \\ &= \frac{1}{\sqrt{2}}(|0\rangle\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + |0\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)) \\ &= \frac{1}{2}(|00\rangle + |01\rangle + |00\rangle - |01\rangle) \\ &= |00\rangle \end{aligned}$$

La deuxième opération élémentaire, le XOR quantique, agit de facon réversible sur deux qubits, la source et la cible. Un XOR quantique ayant comme source le qubit i et pour destination j effectue sur chaque état de base l'opération

$$(x_i, x_j) \mapsto (x_i, x_i + x_j)$$

ce qui a pour effet de permuter les coefficients des états de base d'un registre sans changer leurs amplitudes[3]. En notation matricielle, on obtient

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

[3]Encore ici, le + signifie la somme modulo 2.

et en notation de Dirac

$$\begin{aligned} |00\rangle &\mapsto |00\rangle \\ |01\rangle &\mapsto |01\rangle \\ |10\rangle &\mapsto |11\rangle \\ |11\rangle &\mapsto |10\rangle \end{aligned}$$

Nous pouvons maintenant donner la sémantique opérationnelle d'un circuit quantique ou q-circuit. L'évaluation d'un q-circuit se fait en trois étapes. Premièrement l'initialisation du registre à un vecteur binaire classique pouvant contenir l'input ainsi que des constantes. Deuxièmement l'évaluation porte à porte du circuit. La troisième opération est nécessaire pour obtenir un résultat classique, on l'appelle mesure ou projection. En fait la sortie classique du circuit dépend de façon probabiliste de l'état superposé obtenu après l'évaluation de toutes les portes. Si le registre est dans l'état

$$\sum_{i=0}^{2^n-1} \alpha_i |i\rangle$$

alors on obtiendra la valeur i avec probabilité $|\alpha_i|^2$ d'où la nécessité d'avoir

$$\sum_{i=0}^{2^n-1} |\alpha_i|^2.$$

Comme pour les circuits réversibles, lors de l'évaluation d'un circuit quantique le nombre de fils ne change pas. On choisira donc certains fils pour contenir le résultat du calcul. En général, on dira que sur entrée x la sortie d'un circuit est y si, ayant initialisé le circuit avec la valeur $(x, c_1, 0)$, évalué le circuit et projeté le registre, on obtient (x, g, y) où c_1 est une constante indépendante de x et g est quelconque. On dit qu'une fonction F est dans **QP** s'il existe une famille de

q-circuits Q de taille polynomiale telle que

$$\forall x, (Pr[Q(x, c, 0) \mapsto (x, c, y)] = 1) \Leftrightarrow F(x) = y.$$

On dit qu'une fonction est dans **BQP** s'il existe une famille de circuits de taille polynomiale telle que

$$\forall x, (Pr[Q(x, c, 0) \mapsto (x, c, y)] \geq \frac{2}{3}) \Leftrightarrow F(x) = y.$$

Pour plus d'information sur la complexité quantique et ses liens avec la complexité classique, le lecteur est invité à lire [47, 106, 22, 16].

Attention, la notion d'uniformité pour les q-circuits n'est pas encore bien définie mais pour les besoins de ce document, nous considérerons les familles de circuits efficacement calculables classiquement. Il est clair qu'il y a place au travail de recherche pour bien définir la notion d'uniformité dans les q-circuits. Cette tâche cache des difficultés dues au fait que la notion d'efficacement calculable est elle-même affectée par les algorithmes quantiques. Nous adopterons l'attitude sûre qui consiste à considérer uniquement les circuits qui sont efficacement constructibles par une machine de Turing.

2.2.1 Universalité

La description de q-circuit faite précédemment ne semble pas permettre d'effectuer des opérations utiles. Bien au contraire, les q-circuits sont universels à la fois classiquement et de façon quantique.

Théorème 2.2.1 (Deutsch & DiVincenzo [56, 59]) *Tout circuit réversible de largeur l, profondeur d et taille m peut être simulé par un circuit quantique de largeur l, profondeur $15d$ et taille $16m$.*

Pour simuler un circuit réversible, il suffit de pouvoir simuler la porte de Toffoli et l'entrée de constantes. Il a été démontré dans [5] que les opérations élémen-

taires que nous avons définies permettent la réalisation de la porte de Toffoli, c'est-à-dire qu'il existe un q-circuit qui, lorsqu'exécuté sur des entrées classiques (non superposées), se comporte comme la porte de Toffoli. Bien sûr, rien n'empêche d'appliquer ce circuit sur un registre en superposition ; le résultat sera la superposition de l'application du q-circuit sur chacune des valeurs de base.

Il est clair qu'en initialisant le registre avec l'input du circuit ainsi que des zéros, puis en simulant le r-circuit porte par porte, le registre contiendra finalement les valeurs que le circuit réversible aurait calculées. Ce même circuit composé de portes quantiques et calculant une bijection classique peut être évalué sur un registre en superposition ; la linéarité des opérations décrites plus tôt nous assure que cette opération laissera le registre en superposition. Nous ferons grand usage de ce fait dans les algorithmes qui suivent et nous pouvons donc simplifier le modèle en remplacant le XOR dans notre définition par *n'importe quel circuit réversible classique.* Bien sûr, la taille du q-circuit devra compter la taille des r-circuits qu'il simule.

Il y a aussi une autre facette à l'universalité des circuits quantiques. Nous avons vu qu'ils permettent de simuler n'importe quel circuit classique par la simulation de r-circuits mais existe-t-il des opérations que la mécanique quantique permet de réaliser sur un registre et que notre modèle ne permet pas ?

Théorème 2.2.2 (Barenco et cie [5]) *Toute opération unitaire peut être simulée par un circuit quantique.*

Jusqu'a maintenant nous n'avons défini que les opérations unitaires sur un espace de dimension deux (qubit). En général, une opération unitaire est une opération linéaire U telle que

$$UU^\dagger = U^\dagger U = I.$$

Dans l'état de la connaissance des règles qui régissent les particules élémentaires, on considère qu'une opération peut être effectuée sur un système donné si et

seulement si elle est unitaire. Le point important soulevé ici est que ce modèle est complet, robuste et réaliste. Il est complet pour la raison exprimée plus tôt. Sa robustesse tient du fait que d'autres ensembles de portes élémentaires complets existent et qu'ils peuvent s'exprimer simplement avec celui que nous avons décrit, et son réalisme tient du fait que toutes les implantations concrètes d'ordinateur quantique semblent plus ou moins utiliser les mêmes blocs de base [57]. Il est important que les opérations élémentaires ne soient pas trop puissantes car la complexité des problèmes pourrait être sous-évaluée. Elle ne doit pas non plus être trop simple, rendant le modèle inutilement lourd. L'ensemble de portes élémentaires choisies semble donc approprié.

2.2.2 Boîte à outils

Donnons-nous maintenant quelques outils ou procédures quantiques. Nous avons choisi de présenter ici trois transformations quantiques qui à elles seules forment le noyau de la presque totalité des algorithmes existant dans la littérature.

Premièrement, concentrons-nous sur la transformation de Walsh-Hadamard

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

On remarque d'abord que cette transformation est son propre inverse.

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

On remarque ensuite que son application sur chaque qubit d'un registre dans un état de base donne une égale superposition de tous les états de base au signe près

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

où $x \cdot y = \sum_{i=1}^{n} x_i y_i \ mod\ 2$ (on peut facilement vérifier cette équation par induction). En particulier, pour obtenir une superposition de tous les états de base avec une égale amplitude il suffit d'appliquer W sur tous les qubits d'un registre initialisé à l'état $|0^n\rangle$ c'est-à-dire

$$W^{\otimes n}|0^n\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} |y\rangle.$$

Une deuxième opération utilisée dans plusieurs algorithmes est le changement de phase conditionnel. Il s'agit, pour une fonction $F : \{0,1\}^n \to \{0,1\}$ donnée, d'inverser la phase des états de base x pour lesquels $F(x) = 1$. On désire donc implanter la transformation

$$|x\rangle \mapsto (-1)^{F(x)}|x\rangle.$$

Pour ce faire, il suffira d'avoir un circuit $\bar{F}$ qui calcule F de façon réversible, $(x, b) \mapsto (x, b \oplus F(x))$, (où $\oplus$ représente la somme bit à bit) ainsi qu'un qubit auxiliaire dans l'état $W\ |1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ puisque si $F(x) = 0$

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

mais si $F(x) = 1$ alors

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

La troisième transformation qui nous sera utile est la transformée de Fourier. Cette transformation définie pour les états de base d'un registre de n qubits par

$$\begin{aligned} \mathbf{F}_q \ : \ |k\rangle &\mapsto \tfrac{1}{\sqrt{q}} \textstyle\sum_{l=0}^{q-1} e^{2\pi i \frac{kl}{q}} |l\rangle \qquad (0 \le k < q) \\ |k\rangle &\mapsto |k\rangle \qquad\qquad\qquad\qquad\quad (q \le k < 2^n) \end{aligned}$$

effectue simplement la transformée de Fourier discrète des coefficients. Nous ne donnerons pas ici le circuit qui la calcule.

Théorème 2.2.3 ([100, 101, 52, 62]) *Il existe un circuit quantique de taille $O(n^2)$ effectuant la transformation $\mathbf{F}_{2^n}$.*

C'est dans [100] qu'on trouve le premier algorithme effectuant la transformée de Fourier quantique sur $\mathbb{Z}_q$. L'algorithme fonctionnait en temps polynomial pour autant que q soit un produit de petits nombres premier. Si q est une puissance de deux, il existe un circuit simple calculant cette transformation [101, 52, 62] qui est sans doute la plus importante transformation quantique que nous connaissions. Nous reviendrons dans les chapitre 3 et 5 sur cette transformation et son inverse. Contentons nous de dire que si on l'applique sur un registre dans un état

$$|\Psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$$

où les α_i sont périodiques de période k alors

$$\mathbf{F}_q|\Psi\rangle = \sum_{i=0}^{2^n-1} \beta_i |i\rangle$$

où une bonne partie de l'amplitude est concentrée aux états proches de l'état de base $2^n/k$. On peut donc utiliser cette transformation pour évaluer la période d'une fonction classique ou des amplitudes d'une superposition quantique.

2.3 Principaux algorithmes quantiques

Les principaux algorithmes sont Deutsch-Jozsa, Bernstein-Vazirani, Simon, factorisation et logarithme discret de Shor, Grover et comptage. Dans cette section, nous discutons des principaux algorithmes quantiques. Les premiers seront décrits dans les sous-sections suivantes. Quant aux deux derniers, l'algorithme de Grover et de comptage, nous leur consacrons les chapitres 3 et 4.

Notez que les quatre problèmes dans cette section sont présentés assez sommairement, aucune preuve de leur fonctionnement ne sera donnée et leurs algorithmes y sont décrits seulement de facon indicative. La littérature traitant de ces articles est d'ailleurs suffisamment abondante pour qu'un lecteur curieux puisse combler toute l'information manquante [58, 102, 101, 74, 21, 6, 20].

Certains des algorithmes quantiques connus résolvent un problème consistant à identifier une propriété d'une fonction. Ces algorithmes ont tous la caractéristique qu'ils n'utilisent pas la structure de la fonction. Ils se contentent d'évaluer ladite fonction sur une superposition adéquate. Dans ce cas on dira que la fonction est donnée sous la forme d'une *boîte noire*. On comparera la performance d'un algorithme quantique utilisant une fonction comme une boîte noire avec les algorithmes classiques qui font de même en comptant le nombre d'appels à la boîte noire. Il nous faudra d'ailleurs distinguer entre la fonction et son implantation en circuit. On notera donc $\bar{f}$ le circuit qui calcule de façon réversible la fonction $(x, b) \mapsto (x, b \oplus f(x))$

2.3.1 Deutsch-Jozsa

Le premier exemple sur lequel nous avons décidé de nous attarder est l'algorithme de Deutsch et Jozsa [58]. Malheureusement, le problème résolu n'est pas naturel et de plus un algorithme probabiliste classique peut le résoudre efficacement si l'on tolère une probabilité infinitésimale d'erreur. Par contre, son importance historique et sa simplicité nous empêchent de l'ignorer.

On dit d'une fonction F qu'elle est *constante* si $\exists y \forall x, F(x) = y$. On dira qu'elle est *équilibrée* si $|\{x|F(x) = 0\}| = |\{x|F(x) = 1\}|$.

Problème : Deutsch-Jozsa
Donnée : $F : \{0,1\}^n \to \{0,1\}$ une fonction,
Promesse : F est équilibrée ou constante.

Sortie : équilibrée ou **constante** selon le cas.

Si la fonction est donnée par une boîte noire, alors classiquement, dans le pire des cas, on doit s'attendre à évaluer F plus de 2^{n-1} fois pour connaître la réponse avec certitude. L'algorithme de Deutsch-Jozsa résout ce problème exactement en évaluant la fonction *une seule fois*. Il semble que nous ayons là une accélération fulgurante mais cela est trompeur car il existe un algorithme probabiliste pour résoudre ce problème, à savoir évaluer F sur k points aléatoires et répondre *constante* si et seulement si l'évaluation de F sur ces points donne toujours la même valeur. Néanmoins, on trouve dans cet algorithme le germe de ce qui sera utilisé dans des algorithmes plus utiles.

Deutsch-Jozsa(F)

1. $|\Psi\rangle \leftarrow |0^n, 1\rangle$
2. $|\Psi\rangle \leftarrow (I_{2^n} \otimes W)\ |\Psi\rangle$
3. $|\Psi\rangle \leftarrow (W^{\otimes n} \otimes I_2)\ |\Psi\rangle$
4. $|\Psi\rangle \leftarrow \bar{F}\ |\Psi\rangle$
5. $|\Psi\rangle \leftarrow (W^{\otimes n} \otimes I_2)\ |\Psi\rangle$
6. Projeter $|\Psi\rangle$ dans (x, b)
7. Si $x = 0$ retourner **Constant**, sinon **Balancé**

Théorème 2.3.1 (Deutsch & Jozsa 91 [58]) *L'algorithme* **Deutsch-Jozsa** *résout le problème de Deutsch-Jozsa de façon exacte avec un seul appel à F.*

2.3.2 Bernstein-Vazirani

Le deuxième algorithme que nous allons étudier est celui de Ethan Bernstein et Umesh Vazirani. Bien que résolvant, lui aussi, un problème peu naturel, son importance historique nous empêche de l'écarter.

Problème : Fourier-Sampling

Donnée : $F : \{0,1\}^n \to \{0,1\}$ une fonction

Promesse : Il existe $a \in \{0,1\}^n$ tel que pour tout x, $F(x) = a \cdot x$

Sortie : a.

Tout appel classique à F nous apprend un bit d'information au sujet de a. Il faut donc n appels à F pour résoudre le problème à l'aide d'un algorithme classique. Il existe pourtant un algorithme quantique résolvant le problème nécessitant un seul appel à F [20].

Bernstein-Vazirani(F)

1. $|\Psi\rangle \leftarrow |0^n, 1\rangle$
2. $|\Psi\rangle \leftarrow (I_{2^n} \otimes W) |\Psi\rangle$
3. $|\Psi\rangle \leftarrow (W^{\otimes n} \otimes I_2) |\Psi\rangle$
4. $|\Psi\rangle \leftarrow \bar{F} |\Psi\rangle$
5. $|\Psi\rangle \leftarrow (W^{\otimes n} \otimes I_2) |\Psi\rangle$
6. Projeter $|\Psi\rangle$ dans (x, b)
7. Retourner x.

Théorème 2.3.2 (Bernstein-Vazirani [20]) *L'algorithme Bernstein-Vazirani résout le problème Fourier-Sampling en un appel à F.*

L'importance de cet algorithme tient du fait qu'en considérant une version récursive du problème [20], Bernstein et Vazirani ont démontré qu'il existe un oracle A pour lequel $\mathbf{BQP}^A \not\subseteq \mathbf{BPP}^A$.

2.3.3 Simon

Le deuxième algorithme que nous allons étudier est celui de Simon. C'est le premier algorithme quantique à résoudre un problème concret plus rapidement qu'un ordinateur classique, même probabiliste.

Problème : Collision

Donnée : $F : \{0,1\}^{n+1} \to \{0,1\}^n$ une fonction

Promesse : $\exists s, \forall x, F(x) = F(y) \Leftrightarrow x = y \oplus s$

Sortie : s.

Comme l'algorithme que nous allons présenter est probabiliste, nous aurons besoin du théorème suivant qui montre qu'en fait, contrairement au problème de Deutsch-Jozsa, même un algorithme probabiliste ne peut résoudre ce problème efficacement.

Théorème 2.3.3 (Simon [102]) *Si F est donnée par une boîte noire, aucun algorithme probabiliste qui évalue F moins de $2^{n/4}$ fois ne peut résoudre le problème de* **Collision** *avec probabilité de succès supérieure à $1/2 + 2 \cdot 2^{-n/2}$.*

Nous allons maintenant décrire un algorithme quantique qui peut résoudre ce problème avec un nombre espéré d'évaluations de F linéaire.

Simon(F)

1. $S = \{\}$
2. $|\Psi\rangle \leftarrow |0^{n+1}, 0^n\rangle$
3. $|\Psi\rangle \leftarrow (W^{\otimes n+1} \otimes I_{2^n}) |\Psi\rangle$
4. $|\Psi\rangle \leftarrow \bar{F} |\Psi\rangle$
5. $|\Psi\rangle \leftarrow (W^{\otimes n+1} \otimes I_{2^n}) |\Psi\rangle$
6. Projeter $|\Psi\rangle$ dans (x, z)
7. Si x est indépendant de S alors ajouter x à S
8. Si $|S| = n + 1$ alors déduire s de S et retourner s, sinon recommencer à 2.

Théorème 2.3.4 (Simon [102]) *L'algorithme Simon résout le problème de* **Collision** *en un nombre espéré d'appels à F linéaire.*

Idée de preuve. L'algorithme **Simon** calculera à chaque tour de boucle une valeur aléatoire x tel que $x \cdot s = 0$ et accumulera ces valeurs dans un ensemble S. Lorsque n valeurs linéairement indépendantes auront été trouvées il sera facile d'en déduire s. Malheureusement plus S est grand, plus la probabilité que x soit indépendant de S est faible, c'est pourquoi cet algorithme fonctionne en temps espéré linéaire.

■

Il est à noter que récemment G. Brassard et P. Høyer [33] ont montré que le problème est dans **QP** en donnant un algorithme pour lequel le nombre d'appels à F est borné.

2.3.4 Shor

Le quatrième algorithme présenté est sans aucun doute le plus important en algorithmie quantique. En 1994, Peter Shor [100, 101] publia un article qui fit beaucoup de bruit. Il présenta deux algorithmes fondés sur une procédure quantique de son invention, la transformée de Fourier quantique ($\mathbf{F}_q$). Le premier permet de factoriser les grands nombres en temps espéré polynomial et le second d'extraire le logarithme discret. Nous avons choisi ici de décrire l'algorithme de factorisation, l'autre étant apparenté. Nous avons déjà, dans le premier chapitre, mentionné l'importance de ce problème. Rappelons simplement que bien qu'il ne soit pas prouvé qu'aucun algorithme classique ne peut résoudre ce problème efficacement[4], la confiance dans sa difficulté est suffisamment grande pour qu'une grande partie de la sécurité des protocoles cryptographiques utilisés en pratique repose sur cette hypothèse [26].

Problème : Factorisation
Donnée : $N \in \{0,1\}^n$
Retourne : y et z tel que $N = yz$
Promesse : N est composé.

[4] même de façon probabiliste

Le meilleur algorithme connu pour résoudre *factorisation* est le crible algébrique et nécessite un temps dans $O(e^{(1.92+o(1))n^{1/3}(\log n)^{2/3}})$ où n est la taille du nombre à factoriser. Il est conjecturé qu'aucun algorithme ne peut résoudre **Factorisation** en temps polynomial. L'algorithme de Shor utilise le fait que **Factorisation** se réduit de façon probabiliste classique à **Ordre**.

Problème : Ordre
Donnée : y et N
Retourne : le plus petit r tel que $y^r = 1 \ (mod \ N)$
Promesse : r existe.

Classiquement, on ne sait résoudre ce problème efficacement mais il existe un algorithme quantique utilisant la $\mathbf{F}_q$ qui le résout. La technique est très semblable à celle utilisée par Simon avec W remplacé par $\mathbf{F}_q$ et en choisissant $F(x) := y^x \ (mod \ N)$.

Pour plus de détails concernant l'algorithme de Shor, le lecteur pourra regarder [100, 101, 45, 45, 52, 62, 93, 49]. Notez qu'au même moment où Shor publiait son algorithme de factorisation, il donnait aussi une solution aux problèmes du logarithme discret [100, 101]. Subséquemment, A. Yu Kitaev [81] donna un algorithme pour résoudre le problème du stabilisateur abélien (*abelian stabilizer problem*) et montra que la factorisation et le logarithme discret en étaient des cas particuliers.

2.4 Conclusion

Les quatre exemples donnés précédemment couvrent les algorithmes quantiques les plus importants à l'exception de celui décrit dans le prochain chapitre. Il est intéressant de savoir qu'un lien existe entre ces algorithmes. Une analyse mathématique permet de mettre en relief la relation algébrique étroite entre ces

quatre problèmes puisqu'on peut en discuter en termes de transformée de Fourier appliquée sur un groupe abélien [74].

Les circuits quantiques peuvent servir à autre chose qu'à la résolution de problèmes calculatoires. On peut les utiliser pour modéliser des phénomènes très différents. Gilles Brassard a décrit le circuit de la téléportation quantique [36]. On peut aussi les utiliser pour discuter de correction d'erreur quantique, qui représente un domaine de recherche important [5, 42, 104, 98].

Les circuits peuvent aussi servir à décrire le comportement de participants dans un protocole quantique multi-parties, ce qui touche bien sûr à la cryptographie mais aussi et surtout à un domaine d'intérêt nouveau appelé complexité de communication quantique. Nous reviendrons à la complexité de communication dans les chapitres 6, 7, 8 et 9.

Chapitre 3

Algorithme de Grover, analyse et extensions

L'objectif de ce chapitre est de préparer le lecteur aux articles « Tight Bounds on Quantum Searching » et « Quantum Amplitude Amplification and Estimation » (chapitres 4 et 5), mais aussi d'en faire ressortir les éléments significatifs. Nous présenterons certains résultats différemment des articles pour faire ressortir l'intuition qui mena à leurs découvertes, ce qui, nous l'espérons, facilitera du même coup leur compréhension.

3.1 Algorithme de Grover

En 1996, Lov K. Grover publia un article intitulé « A Fast Quantum Mechanical Algorithm for Database Search » [69]. Cet article eut un impact énorme dans la communauté puisqu'il présentait une nouvelle procédure quantique tout à fait originale et d'un intérêt certain. Soit un tableau T de taille N, le problème étudié par Grover consiste a trouver pour un y donné, un i tel que $T[i] = y$. Si le tableau n'est pas trié, un algorithme classique devra en moyenne visiter la moitié des éléments du tableau ($O(N)$) avant de tomber sur le bon s'il existe et est unique. Dans son article, Grover montra l'existence d'un algorithme quantique

pour résoudre ce problème avec $O(\sqrt{N})$ accès au tableau. Bien que non exponentielle, cette accélération n'en demeure pas moins appréciable. Malheureusement, la procédure décrite par Grover n'est pas complète et c'est grâce à notre analyse formulée dans l'article « Tight Bounds on Quantum Searching » [25, chapitre 4 ici] qu'un algorithme complet existe pour résoudre ce problème dans toute sa généralité.

Commençons par reformuler le problème de recherche de façon plus adéquate. C'est cette formulation qui est utilisée dans le chapitre 4 bien qu'aucun nom explicite ne soit donné au problème.

Problème : F-SAT
Donnée : $F : \{0, \ldots, N-1\} \to \{0,1\}$ une fonction booléenne (boîte noire)
Retourne : i tel que $F(i) = 1$ ou NIL si $F^{-1}(1) = \{\}$.
Paramètres : $N = 2^n$ et $t = |\{i \mid F(i) = 1\}|$.

Lorsque F est une boîte noire, le théorème suivant est immédiat.

Theorem 3.1.1 *Tout algorithme classique résolvant* **F-SAT** *avec probabilité supérieure à* $2/3$ *nécessite* $\Omega(N/t)$ *évaluations de* F *; de plus,* $O(N/t)$ *appels à* F *sont suffisants.*

Le résultat originel de Grover s'énonce comme suit :

Theorem 3.1.2 (Grover 96 [69]) *Il existe un algorithme résolvant* **F-SAT** *nécessitant moins de* $\sqrt{2N}$ *évaluations de* F *dans le cas spécifique où* $T = 1$.

Il y a deux raisons qui nous amènent à choisir cette formulation en termes de fonction plutôt que celle de tableau utilisé par Grover. La première est que pour appliquer l'algorithme de Grover dans le contexte d'une base de données, une mémoire quantique spéciale, à accès direct et adressable en superposition, est nécessaire. Les q-circuits ne peuvent modéliser adéquatement ce genre de choses et il n'est même pas clair que ce soit réalisable en pratique. La deuxième raison est que le problème **F-SAT** s'apparente au problème de la satisfaisabilité de

circuit booléen qui est le problème **NP**–complet par excellence. Tout problème dans **NP** peut donc se ramener à **F-SAT** et ce, de façon naturelle [51, 66]. Inutile de rappeler la quantité phénoménale de problèmes d'importance pratique qui se trouvent dans cette classe. Pour une définition formelle de la classe ainsi qu'un survol des problèmes qui s'y trouvent, nous invitons le lecteur à consulter [66].

Comme nous l'avons vu (théorème 3.1.1), dans le cas classique, le concept de boîte noire rend l'analyse du problème **F-SAT** triviale. Il n'en est pas de même pour les algorithmes quantiques puisque ceux-ci peuvent évaluer une fonction (q-circuit) sur une superposition de valeurs. En fait, dans le cas quantique, une borne inférieure quant au nombre d'appels nécessaire était connue bien avant la publication par Grover de son fameux article. En 1994, C.H. Bennett, E. Bernstein, G. Brassard, U.V. Vazirani [16] ont démontré une borne inférieure de $\Omega(\sqrt{N})$ appels à la fonction. Dans le chapitre 4 (Théorème 4.8.5), une analyse plus approfondie nous permet d'affirmer qu'aucun algorithme quantique ne peut résoudre **F-SAT** avec probabilité supérieure à $1/2$ en moins de $\sin(\pi/8)\sqrt{N/t}$ évaluations de F.

L'algorithme quantique proposé par Grover est simple. On initialise un registre avec une superposition égale de tous les éléments du domaine puis on applique un certain nombre de fois quelques opérations que nous regrouperons sous l'appellation *d'itération de Grover* qui sera notée G_F. Plus précisément

$$G_F = -WS_0WS_F,$$

où W est la transformation de Walsh–Hadamard et S_0 et S_F sont des changements de phase conditionnels définis comme suit :

$$\begin{aligned} S_F|i\rangle &= (-1)^{F(i)}|i\rangle \\ S_0|i\rangle &= -|i\rangle \text{ si } i = 0 \text{ et } |i\rangle \text{ sinon} \end{aligned}$$

L'algorithme de Grover s'énonce comme suit :

Grover(F, j)

1. $|\Psi\rangle \leftarrow W^{\otimes n}|0^n\rangle$
2. Faire j fois
   - $|\Psi\rangle \leftarrow G_F|\Psi\rangle$
3. Projeter $|\Psi\rangle$ dans k et retourner k.

Dans son analyse, Grover avait démontré que pour $t = 1$ ($F(i) = 1$ pour un i unique) il existe $j < \sqrt{2N}$ pour lequel **Grover**(F, j) permet d'obtenir i tel que $F(i) = 1$ avec probabilité supérieure à $1/2$. Ce résultat est très intéressant, mais il ne décrit pas réellement un algorithme; il en montre plutôt l'existence puisque j n'est pas spécifié. Une analyse mathématique plus poussée de l'itération (G_F) nous a permis d'obtenir une description simple de l'état du registre après l'application de l'algorithme **Grover**(F, j). Soit θ tel que $\sin^2(\theta) = \frac{t}{N}$. Nous obtenons

$$G_F^j(W^{\otimes n}|0^n\rangle) = \sum_{i \in F^{-1}(1)} \frac{\sin((2j+1)\theta)}{\sqrt{t}}|i\rangle + \sum_{i \in F^{-1}(0)} \frac{\cos((2j+1)\theta)}{\sqrt{N-t}}|i\rangle. \qquad (3.1)$$

La probabilité d'observer i tel que $F(i) = 1$ si l'on mesure après la j-ième itération est donc de $\sin^2((2j+1)\theta)$. Le graphique 3.1 met en relief l'importance de bien spécifier j. On y remarque entre autres que le nombre optimal d'itérations quand $t = 1$ est un très mauvais choix quand $t = 4$. Il suffit maintenant de remarquer que $\sin^2((2j+1)\theta)$ est maximale quand $j = \lfloor \frac{\pi}{4\theta} \rfloor$ afin d'obtenir, lorsque t est connu, un algorithme bien spécifié faisant moins de $\frac{\pi}{4}\sqrt{\frac{N}{t}}$ appels à F et permettant de résoudre **F-SAT** avec probabilité d'erreur inférieure à $\frac{t}{N}$.

En général, la valeur de t nous est inconnue. Forts des résultats précédents, nous avons aussi montré (Chapitre 4 section 5) qu'il est possible de résoudre **F-SAT** même si t est inconnu, et ce avec un nombre espéré d'évaluations de F dans le même ordre que si t était connu. L'idée intuitive exploitée par l'algorithme est la

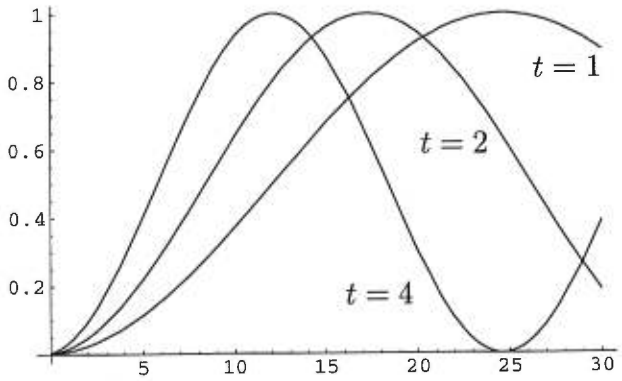

FIG. 3.1: Probabilité de succès de l'algorithme **Grover** en fonction de j quand $N = 1024$ pour $t = 1, 2$ et 4 respectivement.

suivante. Premièrement, si l'on choisit un nombre d'itérations j de façon aléatoire entre zéro et un multiple du choix optimal $(\Theta(\sqrt{N/t}))$, la probabilité d'obtenir une solution avec **Grover**(F, j) est supérieure à $1/4$. On appellera donc l'algorithme successivement avec un nombre d'itérations choisi aléatoirement entre zéro et une borne de plus en plus grande. Certains paramètres doivent toutefois être ajustés avec soin pour obtenir la convergence du temps espéré de calcul.

Notre analyse formelle de l'itération de Grover ainsi que notre solution au problème **F-SAT** ont ouvert la voie à d'autres résultats. Citons par exemple le problème de collision sans restriction [38], le min et le max [60] ainsi que la médiane [70].

Dans l'article « Quantum Amplitude Amplification and Estimation » [34, chapitre 5 ici], l'itération de Grover est généralisée à l'opération suivante :

$$\mathbf{Q}(\mathcal{A}, \chi) = -\mathcal{A}\, \mathbf{S}_0\, \mathcal{A}^{-1}\, \mathbf{S}_\chi.$$

où $\mathcal{A}$ est une opération unitaire quelconque et χ est une fonction booléenne quelconque. De façon spécifique, on a que

$$G_F = \mathbf{Q}(W, F).$$

Une partie de cet article sera donc consacrée à l'étude de cette itération généralisée dans une perspective *d'amplification d'amplitude.*

On peut voir $\mathcal{A}$ comme un algorithme probabiliste quantique qui, lorsqu'appliqué à une superposition égale des éléments du domaine de χ, retourne x tel que $\chi(x) = 1$ avec probabilité a. De façon classique, il faudrait répéter cette technique $1/a$ fois pour obtenir avec bonne probabilité une solution à χ. En utilisant adéquatement $\mathbf{Q}(\mathcal{A}, \chi)$ ainsi que des techniques similaires à celle utilisée pour obtenir un algorithme général pour **F-SAT**, il sera possible d'obtenir un algorithme quantique qui résout **F-SAT** avec heuristique quantique $\mathcal{A}$ en temps aproximativement $1/\sqrt{a}$.

La section 3 (Chapitre 5) traite d'une famille très générale d'heuristiques classiques et de leurs implantations en termes d'amplification d'amplitude. Il sera démontré (Théorème 5.4.1 chapitre 5) pour toute heuristique de cette famille que si l'algorithme classique utilisant cette heuristique trouve une solution en temps espéré T alors un algorithme quantique peut faire de même en temps espéré au plus $\sqrt{T}$.

3.2 Comptage

Comme le montre l'équation 3.1, l'amplitude des états solutions varie en fonction du nombre d'itérations j comme un sinus de période $\frac{\pi}{\theta}$. Comme $\sin^2(\theta) = \frac{t}{N}$, l'évaluation de cette période nous donne de façon indirecte de l'information sur t ($t = |\{i \mid F(i) = 1\}|$), de là l'idée d'utiliser l'itération de Grover pour construire un algorithme permettant d'estimer t. Comme nous l'avons vu dans la section précédente, la transformée de Fourier quantique est un outil extraordinaire : elle permet à partir d'un état $\sum_{i=0}^{M-1} A(i)|i\rangle$, où A est périodique de période p, de concentrer l'amplitude sur les états $|i\rangle$ où i est proche de $\frac{M}{p}$ (et $M + 1 - \frac{M}{p}$). Il ne reste plus qu'à construire, à partir de l'itération de Grover, une superposition adéquate.

Considérons donc l'opération unitaire $\Lambda_M(\mathbf{U})$, où $\mathbf{U}$ est unitaire quelconque, définie par

$$|j\rangle|y\rangle \longmapsto |j\rangle(\mathbf{U}^j|y\rangle) \qquad (0 \leq j < M). \tag{3.2}$$

Si l'on applique $\Lambda_M(G_F)$ à une paire de registres initialisée dans l'état suivant

$$\left(\frac{1}{\sqrt{M}}\sum_{j=0}^{M-1}|j\rangle\right)(W^{\otimes n}|0^n\rangle)$$

on obtiendra alors une superposition d'égale amplitude de zéro, une, deux, jusqu'à $M-1$ applications de l'itération de Grover. On peut donc réécrire cet état comme suit

$$\sum_{j=0}^{M-1}\frac{\sin((2j+1)\theta)}{\sqrt{tM}}|j\rangle\left(\sum_{i\in F^{-1}(1)}|i\rangle\right)+\sum_{j=0}^{M-1}\frac{\cos((2j+1)\theta)}{\sqrt{(N-t)M}}|j\rangle\left(\sum_{i\in F^{-1}(0)}|i\rangle\right). \tag{3.3}$$

Cela signifie donc que si l'on mesure le deuxième registre et que l'on néglige tout facteur constant de normalisation, l'état du premier registre sera soit

$$\sum_{j=0}^{M-1}\sin((2j+1)\theta)|j\rangle$$

ou bien

$$\sum_{j=0}^{M-1}\cos((2j+1)\theta)|j\rangle.$$

Il est maintenant clair que l'application de la transformée de Fourier quantique à ce registre nous donnera de l'information sur θ qui est, comme nous le savons, directement reliée à t. Appliquons donc $\mathbf{F}_M$, pour un M approprié, sur le premier registre et mesurons. Soit k le résultat de la mesure. On sait que k est un bon

estimateur de $\frac{M\theta}{\pi}$ (ou $\frac{M(1-\theta)}{\pi}$) et que $t = N\sin^2\theta$. On obtient donc l'algorithme suivant :

Count(F, M)

1. Initialiser $|\Psi\rangle \leftarrow \left(\frac{1}{\sqrt{M}}\sum_{j=0}^{M-1}|j\rangle\right)(W^{\otimes n}|0^n\rangle)$
2. $|\Psi\rangle \leftarrow \Lambda_M(G_F)|\Psi\rangle$
3. $|\Psi\rangle \leftarrow \mathbf{F}_M \otimes I_{2^n}|\Psi\rangle$
4. Projeter $(|\Psi\rangle)$ dans (k, x)
5. Retourner $N\sin^2(k\pi/M)$

La question difficile est maintenant de déterminer la qualité de l'estimation de t en fonction de M. Remarquez que nul n'est besoin de mesurer la seconde partie du registre puisque le résultat de cette mesure ne change en rien la sortie de notre algorithme. Cette mesure n'est qu'un artifice utile à l'analyse de l'algorithme. Il s'est avéré qu'une mesure dans une autre base facilite l'analyse de la qualité de l'estimation : il s'agit de la base formée par les vecteurs propres de l'opération G_F. En faisant l'analyse comme si le dernier registre était mesuré dans cette base, on obtient le résultat général suivant (chapitre 5, théorème 5.5.7). L'algorithme **Count**(F, M) retourne un estimé $\tilde{t}$ de t tel que

$$|\tilde{t} - t| \leq 2\pi\frac{\sqrt{t(N-t)}}{M} + \pi^2\frac{N}{M^2} + \frac{1}{2}$$

avec probabilité au moins $8/\pi^2$. En choisissant $M = \lceil\sqrt{N}\rceil$. On obtient directement le corollaire suivant (chapitre 5, corollaire 5.5.8). L'algorithme **Count**$(F, \lceil\sqrt{N}\rceil)$ retourne un estimé $\tilde{t}$ tel que

$$|\tilde{t} - t| < 2\pi\sqrt{\frac{t(N-t)}{N}} + 11 \tag{3.4}$$

avec probabilité $8/\pi^2$ et nécessite exactement $\lceil\sqrt{N}\rceil$ évaluations de F.

À partir des résultats précédents, il est possible d'obtenir, en modifiant l'algorithme de façon appropriée, deux corollaires intéressants. Premièrement (chapitre 5, corollaire 5.5.9), il est possible d'obtenir un estimé $\tilde{t}$ tel que $|\tilde{t} - t| < \epsilon t$ avec bonne probabilité et en ne faisant que $\Theta\left(\frac{1}{\epsilon}\sqrt{N/t}\right)$ évaluations de F. Il est intéressant de savoir que Nayak et Wu [94] ont démontré que pour un ϵ fixé, cette borne est optimale. Pour obtenir la même précision, un algorithme classique devrait faire $\Omega(\frac{N}{\epsilon^2 t})$ évaluations de F, ce qui nous donne encore une fois une accélération quadratique.

Il est aussi intéressant d'obtenir la valeur exacte de t (avec bonne probabilité). Dans ce cas, un algorithme nécessitant $\Theta(\sqrt{t(N-t)})$ évaluations de F est présenté (chapitre 5, corollaire 5.5.11). Contrairement à une utilisation répétée de l'algorithme de recherche, ce qui nécessiterait un espace proportionnel à t, notre algorithme ne requiert qu'un espace linéaire en $\log(N)$. Là aussi, l'algorithme présenté est optimal en termes du nombre d'évaluations de F. Bien sûr, tout algorithme classique effectuant cette tâche nécessitera $\Omega(N)$ évaluations de F.

Chapter 4

Tight bounds on quantum searching

This chapter reproduces an article presented at *the Fourth Workshop on Physics of Computation* [24] in Boston in November 1996. This is the final version that can be found in the journal *Fortschritte der Physik* [25]. It has been written with the collaboration of Gilles Brassard, Michel Boyer and Peter Høyer.

4.1 Abstract

We provide a tight analysis of Grover's algorithm for quantum database searching. We give a simple closed-form formula for the probability of success after any given number of iterations of the algorithm. This allows us to determine the number of iterations necessary to achieve almost certainty of finding the answer. Furthermore, we analyze the behaviour of the algorithm when the element to be found appears more than once in the table and we provide a new algorithm to find such an element even when the number of solutions is not known ahead of time. Finally, we provide a lower bound on the efficiency of any possible quantum database searching algorithm and we show that Grover's algorithm comes within 2.62% of being optimal.

4.2 Introduction

Let $X_N = \{0, 1, \ldots, N-1\}$ for some integer N and consider an arbitrary function $F : X_N \to \{0, 1\}$. The goal is to find some $i \in X_N$ such that $F(i) = 1$, provided such an i exists. If F is given as a black box—the only knowledge you can gain about F is in asking for its value on arbitrary points of its domain—and if there is a unique solution, no classical algorithm (deterministic or probabilistic) can expect to achieve a probability of success better than 50% without asking for the value of F on roughly $N/2$ points. Throughout this paper we assume for simplicity that each evaluation of F takes unit time. Grover [69] has discovered an algorithm for the *quantum* computer that can solve this problem in expected time in $O(\sqrt{N})$, provided there is a unique solution. He also remarked that a result in [16] implies that his algorithm is optimal, up to an unspecified multiplicative constant, among all possible quantum algorithms.

In this paper we provide a tight analysis of Grover's algorithm. In particular we give a simple closed-form formula for the probability of success after any given number of iterations. This allows us to determine the number of iterations necessary to achieve almost certainty of finding the answer, as well as an upper bound on the probability of failure. More significantly, we analyze the behaviour of the algorithm when there is an arbitrary number of solutions. An algorithm follows immediately to solve the problem in a time in $O(\sqrt{N/t})$ when it is known that there are exactly t solutions. Moreover we provide an algorithm capable of solving the problem in a time in $O(\sqrt{N/t})$ even if the number t of solutions is not known in advance. We also generalize Grover's algorithm to the case N is not a power of 2. Finally, we refine the argument of [16] to show that Grover's algorithm is within 2.62% of being optimal.

To motivate this work, here are three simple applications for Grover's algorithm. Assume you have a large table $T[0\,..\,N-1]$ in which you would like to find some element y. More precisely, you wish to find an integer i such that $0 \leq i < N$ and $T[i] = y$, provided such an i exists. This *database searching problem* can

obviously be solved in a time in $O(\log N)$ if the table is sorted, but no classical algorithm can succeed in the general case with probability better than 50%, say, without probing more than half the entries of T. Grover's algorithm solves this problem in a time in $O(\sqrt{N})$ on the quantum computer by using $F(i) = 1$ if and only if $T[i] = y$. An exciting cryptographic application is that Grover's algorithm can be used to crack the widely used Data Encryption Standard (DES) [95] under a known plaintext attack. Given a matching pair (m, c) of plaintext and ciphertext, consider function $F : \{0,1\}^{56} \to \{0,1\}$ defined by $F(k) = 1$ if and only if $\text{DES}_k(m) = c$. Provided there is a unique solution, the required key k can be found after roughly 185 million expected calls to a quantum DES device [28]. Thus quantum computing makes single-key DES totally insecure. For yet another application, consider a Boolean formula on n variables. You would like to determine if the formula is satisfiable. There may exist an efficient classical algorithm for this problem but none are known. (This is equivalent to the famous $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ open question in theoretical computer science [66]). In this case Grover's algorithm solves the problem in a time in $O(2^{n/2})$, which is better than the time in $O(2^n)$ required by the obvious classical algorithm, but not good enough to imply that $\mathbf{NP} \subseteq \mathbf{BQP}$ [16].

4.3 Overview of Grover's algorithm

Grover's algorithm consists of an initialization followed by a number of identical iterations, a final measurement, and a classical test. For every $F : X_N \to \{0,1\}$, let S_F be the conditional phase shift transform defined by

$$S_F|i\rangle = \begin{cases} -|i\rangle & \text{if } F(i) = 1 \\ |i\rangle & \text{otherwise.} \end{cases}$$

Let S_0 denote S_{F_0}, where $F_0(i) = 1$ if and only if $i = 0$.

Assume for the moment that $N = 2^n$ is a power of 2 and consider any integer $j \in X_N$ as a bit string of length n. Define $i \cdot j$ as the number of 1 in the bitwise AND of i and j. Let W be the Walsh–Hadamard transform defined by

$$W|j\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{i \cdot j} |i\rangle.$$

This is efficiently implemented [58] by applying the simple unitary transformation

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

independently to each qubit of $|j\rangle$. Now we can define one *Grover iteration* as the unitary transformation

$$G_F = -W S_0 W S_F. \tag{4.1}$$

Grover's algorithm first creates a state $|\Psi\rangle = W|0\rangle$. Then G_F is applied to $|\Psi\rangle$ some number m of times. (One primary purpose of this paper is to determine the optimal choice for m.) Finally, the state $|\Psi\rangle$ is measured, which yields some classical value i. The algorithm *succeeds* if and only if $F(i) = 1$.

Let us now assume we are given a quantum black box Q_F for computing F. This will usually come as a unitary transformation that sends state $|i, b\rangle$ to $|i, b \oplus F(i)\rangle$, where $|b\rangle$ is a single qubit and $\oplus$ denotes the exclusive–or. The obvious approach to implementing S_F as a unitary transformation requires two applications of Q_F: if P is the conditional phase-shift defined by $P|i, b\rangle = (-1)^b |i, b\rangle$ then $(S_F|i\rangle)|0\rangle$ can be computed as $Q_F P Q_F |i, 0\rangle$. However, it follows from Lemma 5.5 in [5] that S_F can be implemented using a single application of Q_F. For this, it suffices to note that

$$(S_F|i\rangle)|\Delta\rangle = Q_F(|i\rangle|\Delta\rangle)$$

where $|\Delta\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$.

The Walsh–Hadamard transform W is well-defined only if N is a power of 2. However, this assumption on N can be removed by observing that G_F is just one of many transforms that can be used as iteration in Grover's algorithm. Let W' be any unitary transform satisfying

$$W'|0\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle. \tag{4.2}$$

Then one may easily verify that the transform $G'_F = W' S_0 W'^{\dagger} S_F$ works just as well. (The minus sign in eq. (4.1) was clearly unnecessary although it makes the analysis easier.) Any transform W' satisfying eq. (4.2) can thus be used in Grover's algorithm. When N is a power of 2, the Walsh–Hadamard transform is indeed the simplest possible choice for W'. When N is not a power of 2, the approximate Fourier transform given by Kitaev [81] can be used.

4.4 Finding a unique solution

Assume for now that there is a unique i_0 such that $F(i_0) = 1$. For any real numbers k and ℓ such that $k^2 + (N-1)\ell^2 = 1$, define the state of a quantum register

$$|\Psi(k,\ell)\rangle = k|i_0\rangle + \sum_{i \neq i_0} \ell |i\rangle$$

where the sum is over all $i \neq i_0$ such that $0 \leq i < N$.

The heart of Grover's algorithm is the iteration described in the previous section. A simple calculation—see Grover's original article [69] for details—shows that each iteration efficiently transforms $|\Psi(k,\ell)\rangle$ into

$$\left|\Psi\left(\tfrac{N-2}{N}k + \tfrac{2(N-1)}{N}\ell, \tfrac{N-2}{N}\ell - \tfrac{2}{N}k\right)\right\rangle.$$

It follows that the j–th iteration produces state $|\Psi_j\rangle = |\Psi(k_j, \ell_j)\rangle$ where

$$k_{j+1} = \tfrac{N-2}{N} k_j + \tfrac{2(N-1)}{N} \ell_j \quad \text{and} \quad \ell_{j+1} = \tfrac{N-2}{N} \ell_j - \tfrac{2}{N} k_j \tag{4.3}$$

with initial conditions $k_0 = l_0 = 1/\sqrt{N}$.

In his paper, Grover proves that there exists a number m less than $\sqrt{2N}$ such that k_m^2, the probability of success after m iterations, is at least 50%. This is correct, but one must be careful in using his algorithm because the probability of success does not increase monotonically with the number of iterations. By the time you have performed $\sqrt{2N}$ iterations, the probability of success has dropped down to less than 9.5% and it becomes vanishingly small after about 11% more iterations before it picks up again. This shows that it is not sufficient to know the existence of m in order to apply the algorithm: its explicit value is needed.

The key to a tighter analysis of Grover's algorithm is an explicit closed-form formula for k_j and ℓ_j. This can be obtained by standard techniques—and a little sweat—from recurrence (4.3). Let angle θ be defined so that $\sin^2 \theta = 1/N$ and $0 < \theta \leq \pi/2$. It is straightforward to verify by mathematical induction that

$$k_j = \sin((2j+1)\theta) \quad \text{and} \quad \ell_j = \tfrac{1}{\sqrt{N-1}} \cos((2j+1)\theta). \tag{4.4}$$

It follows from eq. (4.4) that $k_m = 1$ when $(2m+1)\theta = \pi/2$, which happens when $m = (\pi - 2\theta)/4\theta$. Of course, we must perform an *integer* number of iterations but it will be shown in the next section that the probability of failure is no more than $1/N$ if we iterate $\lfloor \pi/4\theta \rfloor$ times. This is essentially $\frac{\pi}{4}\sqrt{N}$ iterations when N is large because $\theta \approx \sin\theta = 1/\sqrt{N}$ when θ is small. It is sufficient to perform half this number of iterations, approximately $\frac{\pi}{8}\sqrt{N}$, if we are satisfied with a 50% probability of success, as Grover considered in his original paper [69]. We shall prove in Section 4.8 that this is optimal within a few percent because any quantum algorithm that solves the search problem with a 50% probability of success must evaluate F at least $(\sin\frac{\pi}{8})\sqrt{N}$ times and $\frac{\pi}{8} \approx 1.026 \sin\frac{\pi}{8}$. One must know when to

stop, however: if we work twice as hard as we would need to succeed with almost certainty, that is we apply approximately $\frac{\pi}{2}\sqrt{N}$ iterations of Grover's algorithm, we fail with near certainty!

4.5 The case of multiple solutions

Let us now consider the case when there are t different values of i such that $F(i) = 1$. We are interested in finding an arbitrary solution. Grover briefly considers this problem [69], but he provides no details concerning the efficiency of his method.

We assume in this section that the number t of solutions is known and that it is not zero. Let $A = \{i \mid F(i) = 1\}$ and $B = \{i \mid F(i) = 0\}$. For any real numbers k and ℓ such that $tk^2 + (N-t)\ell^2 = 1$, redefine

$$|\Psi(k,\ell)\rangle \;=\; \sum_{i\in A} k|i\rangle + \sum_{i\in B} \ell|i\rangle.$$

A straightforward analysis of Grover's algorithm shows that one iteration transforms $|\Psi(k,\ell)\rangle$ into

$$\left|\Psi\left(\tfrac{N-2t}{N}k + \tfrac{2(N-t)}{N}\ell, \tfrac{N-2t}{N}\ell - \tfrac{2t}{N}k\right)\right\rangle.$$

This gives rise to a recurrence similar to (4.3), whose solution is that the state $|\Psi(k_j,\ell_j)\rangle$ after j iterations is given by

$$k_j = \tfrac{1}{\sqrt{t}}\sin((2j+1)\theta) \quad \text{and} \quad \ell_j = \tfrac{1}{\sqrt{N-t}}\cos((2j+1)\theta) \tag{4.5}$$

where the angle θ is so that $\sin^2\theta = t/N$ and $0 < \theta \leq \pi/2$.

The probability of obtaining a solution is maximized when ℓ_m is as close to 0 as possible. We would have $\ell_{\tilde{m}} = 0$ when $\tilde{m} = (\pi - 2\theta)/4\theta$ if that were an integer. Let $m = \lfloor \pi/4\theta \rfloor$. Note that $|m - \tilde{m}| \leq \frac{1}{2}$. It follows that

$|(2m+1)\theta - (2\tilde{m}+1)\theta| \leq \theta$. But $(2\tilde{m}+1)\theta = \pi/2$ by definition of $\tilde{m}$. Therefore $|\cos((2m+1)\theta)| \leq |\sin\theta|$. We conclude that the probability of failure after exactly m iterations is

$$(N-t)\ell_m^2 \;=\; \cos^2((2m+1)\theta) \;\leq\; \sin^2\theta \;=\; t/N.$$

This is negligible when $t \ll N$.

Note that this algorithm runs in a time in $O(\sqrt{N/t}\,)$ since $\theta \geq \sin\theta = \sqrt{t/N}$ and therefore

$$m \;\leq\; \frac{\pi}{4\theta} \;\leq\; \frac{\pi}{4}\sqrt{\frac{N}{t}}.$$

A slight improvement is possible in terms of the expected time if we stop short of m iterations, observe the register, and start all over again in case of failure. The expected number of iterations before success with this strategy is $E(j) = j/tk_j^2$ if we stop after j iterations since our probability of success at that point is tk_j^2. Setting the derivative of $E(j)$ to 0, we find that the optimal number of iterations is given by the j so that $4\theta j = \tan((2j+1)\theta)$. The solution to this equation is very close to $j = z/4\theta$ when $t \ll N$, where $z \approx 2.33112$ is such that $z = \tan(z/2)$. It follows that the optimal number of iterations is close to $0.58278\sqrt{N/t}$ when $t \ll N$ and the probability of success is close to $\sin^2(z/2) \approx 0.84458$. Therefore, the expected number of iterations before success if we restart the process in case of failure is roughly $(z/(4\sin^2(z/2)))\sqrt{N/t} \approx 0.69003\sqrt{N/t}$, which is about 88% of $\frac{\pi}{4}\sqrt{N/t}$, the number of iterations after which success is almost certain. For a numerical example, consider $N = 2^{20}$ and $t = 1$. In this case, we achieve almost certainty of success after 804 iterations. If, instead, we stop at 596 iterations, the probability of success is only 0.84420 but the expected number of iterations before success if we restart the process in case of failure is $596/0.8442 \approx 706$, which is indeed better than 804.

4.6 The case $t = N/4$

An interesting special case occurs when $t = N/4$. Of course, even a classical probabilistic computer can solve this problem efficiently, with high probability, but not quite as efficiently as a quantum computer. Here $\sin^2\theta = t/N = 1/4$ and therefore $\theta = \pi/6$. It follows that $\ell_1 = \frac{1}{\sqrt{N-t}}\cos(3\theta) = 0$. In other words, a solution is found *with certainty* after a single iteration. In terms of the number of times F has to be evaluated, this is essentially four times more efficient than the expected performance of the best possible classical probabilistic algorithm when N is large. Furthermore, the quantum algorithm becomes *exponentially* better than any possible classical algorithm if we compare worst-case performances, taking the worst possible coin flips in the case of a probabilistic algorithm. This is somewhat reminiscent of the Deutsch–Jozsa algorithm [58].

4.7 Unknown number of solutions

A more challenging situation occurs when the number of solutions is not known ahead of time. If we decide to iterate $\frac{\pi}{4}\sqrt{N}$ times, which would give almost certainty of finding a solution if there were only one, the probability of success would be vanishingly small should the number of solutions be in fact 4 times a small perfect square. For example we saw that we are almost certain to find a unique solution among 2^{20} possibilities if we iterate 804 times. The same number of iterations would yield a solution with probability less than one in a million should there be 4 solutions! To find a solution efficiently when their number is unknown, we need the following lemmas, the first of which is easily proved by mathematical induction using straightforward algebra.

Lemma 4.7.1 *For any positive integer m and real number α such that $\sin\alpha \neq 0$,*

$$\sum_{j=0}^{m-1} \cos((2j+1)\alpha) \;=\; \frac{\sin(2m\alpha)}{2\sin\alpha}.$$

Lemma 4.7.2 *Let t be the (unknown) number of solutions and assume that $0 < t < N$. Let angle θ be so that $\sin^2\theta = t/N$ and $0 < \theta < \pi/2$. Let m be an arbitrary positive integer. Let j be an integer chosen at random according to the uniform distribution between 0 and $m-1$. If we observe the register after applying j iterations of Grover's algorithm starting from the initial state, the probability P_m of obtaining a solution is given by*

$$P_m \;=\; \frac{1}{2} - \frac{\sin(4m\theta)}{4m\sin(2\theta)}.$$

In particular $P_m \geq 1/4$ when $m \geq 1/\sin(2\theta)$.

Proof. The probability of success if we perform j iterations of Grover's algorithm is $tk_j^2 = \sin^2((2j+1)\theta)$. It follows that the average success probability when $0 \leq j < m$ is chosen randomly is

$$\begin{aligned} P_m &= \sum_{j=0}^{m-1} \frac{1}{m}\sin^2((2j+1)\theta) \\ &= \frac{1}{2m}\sum_{j=0}^{m-1} 1 - \cos((2j+1)2\theta) \;=\; \frac{1}{2} - \frac{\sin(4m\theta)}{4m\sin(2\theta)}. \end{aligned}$$

If $m \geq 1/\sin(2\theta)$ then

$$\frac{\sin(4m\theta)}{4m\sin(2\theta)} \;\leq\; \frac{1}{4m\sin(2\theta)} \;\leq\; \frac{1}{4}.$$

The conclusion follows. ∎

We are now ready to describe the algorithm for finding a solution when the number t of solutions is unknown. For simplicity we assume at first that $1 \leq t \leq 3N/4$.

1. Initialize $m = 1$ and set $\lambda = 8/7$.
   (Any value of λ strictly between 1 and 4/3 would do.)

2. Choose an integer j uniformly at random such that $0 \leq j < m$.

3. Apply j iterations of Grover's algorithm starting from initial state

$$|\Psi_0\rangle \;=\; W|0\rangle \;=\; \frac{1}{\sqrt{N}} \sum_i |i\rangle.$$

4. Observe the register and let i be the outcome.

5. If $F(i) = 1$, the problem is solved: **exit**.

6. Otherwise, set m to $\min(\lambda m, \sqrt{N}\,)$ and go back to step (ii).

Theorem 4.7.3 *This algorithm finds a solution in expected time in* $O(\sqrt{N/t}\,)$.

Proof. Let angle θ be so that $\sin^2\theta = t/N$ and $0 < \theta < \pi/2$,

$$m_0 \;=\; 1/\sin(2\theta) \;=\; \frac{N}{2\sqrt{(N-t)t}}$$

and $s_0 = \lceil \log_\lambda m_0 \rceil$. Note that $m_0 \leq \sqrt{N/t}$ because $t \leq 3N/4$.

We shall estimate the expected number of times that a Grover iteration is performed before a solution is found: the total time needed is clearly in the order of that number since we assumed that F can be evaluated in unit time. On the s–th time round the main loop, the value of m is $\min(\sqrt{N}, \lambda^{s-1})$ and the expected number of Grover iterations is less than half that value since j is chosen randomly between 0 and $m-1$. Note that $m < m_0$ for the first s_0 times round the main loop, whereas $m \geq m_0$ afterwards. We say that the algorithm reaches the *critical stage* when $m \geq m_0$ for the first time, which may never happen of course if success comes earlier.

The expected total number of Grover iterations needed to reach the critical stage, if it is reached, is at most

$$\tfrac{1}{2}\sum_{s=1}^{s_0}\lambda^{s-1} \;<\; \frac{1}{2}\frac{\lambda}{\lambda-1}m_0 \;=\; 4m_0.$$

Thus, if the algorithm succeeds before reaching the critical stage, it does so in a time in $O(m_0)$, which is in $O(\sqrt{N/t}\,)$ as required.

If the critical stage is reached then every time round the main loop from this point on will succeed with probability at least $1/4$ by virtue of Lemma 4.7.2 since $m \geq 1/\sin(2\theta)$. Therefore, $\frac{1}{2}\lambda^{s_0}$ expected iterations will be performed at round $s = s_0 + 1$. This will succeed with probability at least $1/4$. With complementary probability at most $3/4$, at least one more trip round the loop will be necessary, requiring $\frac{1}{2}\lambda^{s_0+1}$ additional expected iterations. Again, this will succeed with probability at least $1/4$. With probability at most $(3/4)^2$, at least one more trip will be required, costing another $\frac{1}{2}\lambda^{s_0+2}$ expected iterations, and so on. Summing up, the expected number of Grover iterations needed to succeed once the critical stage has been reached is less than

$$\tfrac{1}{2}\sum_{u=0}^{\infty}\left(\tfrac{3}{4}\right)^u\lambda^{s_0+u} \;<\; \frac{2\lambda}{4-3\lambda}m_0 \;=\; 4m_0.$$

The total expected number of Grover iterations, whether or not the critical stage is reached, is therefore less than $8m_0$ and thus the total expected time is in $O(\sqrt{N/t}\,)$ provided $0 < t \leq 3N/4$. Note that $8m_0 \approx 4\sqrt{N/t}$ when $t \ll N$, which is less than six times the expected number of iterations that we would have needed had we known the value of t ahead of time. The case $t > 3N/4$ can be disposed of in constant expected time by classical sampling. The case $t = 0$ is handled by an appropriate time-out in the above algorithm, which allows us to claim in a time in $O(\sqrt{N}\,)$ that there are no solutions when this is the case, with an arbitrarily small probability of failure when in fact there is a solution. ∎

4.8 An improved lower bound

Drawing on general results from [16], Grover points out in [69] that any algorithm for quantum database searching must take a time at least proportional to $\sqrt{N}$ to succeed with nonnegligible probability when there is a unique solution. In this section we prove that if the function F having t solutions is used as a black box in any quantum algorithm Q that makes less than $(\sin\frac{\pi}{8})\sqrt{\lfloor N/t \rfloor} - 1$ calls to F then, averaging over all such possible F, the probability that Q succeeds cannot be better than 50%. Obviously, it follows that, for any $t < N$ and any quantum algorithm that makes less than $(\sin\frac{\pi}{8})\sqrt{\lfloor N/t \rfloor} - 1$ calls to F, there exists an F that has t solutions, yet the algorithm's probability of success does not exceed 50%. This proves that Grover's algorithm comes within 2.62% of being optimal when the number of solutions is known in advance since it follows from Section 4.5 that *it* needs to call F only about $\frac{\pi}{8}\sqrt{N/t}$ times to succeed with probability better than 50%.

After reading an early version of this paper, Grover noticed that our lower bound would *not* apply if we were interested in the *expected* (rather than worst-case) number of calls to F necessary to succeed with probability at least 50%. A better algorithm in terms of the expected number of calls to F consists in first tossing a biased coin. With probability 40%, do nothing—and fail for sure. With probability 60%, apply $0.58278\sqrt{N/t}$ iterations of Grover's algorithm before looking at the quantum state: this will succeed with probability roughly 84.458%, as we saw in Section 4.5. The total expected number of iterations—and thus of calls to F—is $60\% \times 0.58278\sqrt{N/t} < 0.35\sqrt{N/t}$, which is less than $(\sin\frac{\pi}{8})\sqrt{N/t}$, yet the expected success probability is $60\% \times 84.458\%$, which is better than 50%. Nevertheless this approach *never* yields success unless F is evaluated more than $(\sin\frac{\pi}{8})\sqrt{N/t}$ times, which is why our lower bound is not contradicted by this example.

To capture the notion that F is a black box, we consider that it is given as an *oracle*. All matrices and vectors in this section are finite and complex-valued. The norm of vector $\mathbf{a}$ is denoted $||\mathbf{a}||$. The norm of a complex number c is denoted $|c|$.

We restate a basic fact on complex-valued vectors.

Proposition 4.8.1 *For all normalized vectors* $\mathbf{a}$ *and* $\mathbf{b}$*, and all complex scalars* α *and* β*,*

$$||\alpha\mathbf{a} - \beta\mathbf{b}||^2 \geq |\alpha|^2 + |\beta|^2 - 2|\alpha||\beta|.$$

The following proposition is a consequence of Chebyshev's summation inequality.

Proposition 4.8.2 *For all set of complex numbers,* $\{x_i\}_{i=0}^{r-1}$*,*

$$\left(\sum_{i=0}^{r-1} |x_i|\right)^2 \leq r\sum_{i=0}^{r-1} |x_i|^2.$$

Lemma 4.8.3 *Let* S *be any set of* N *strings, and* $\mathcal{C}$ *be any configuration space. Let* $|\phi_0\rangle$ *be any superposition, and*

$$|\phi_r\rangle = U_r \ldots U_2 U_1 |\phi_0\rangle$$

any sequence of r *unitary transforms. Let* $\{f_i\}_{i=0}^{r}$ *be any set of partial functions from* $\mathcal{C}$ *into* S*. For any* $y \in S$*, let*

$$|\phi'_r\rangle = U'_r \ldots U'_2 U'_1 |\phi_0\rangle$$

be any sequence of r *unitary transforms where for all* $i = 1, \ldots, r$*,*

$$U'_i|c\rangle = U_i|c\rangle \quad \text{if} \quad f_{i-1}(|c\rangle) \neq y.$$

Set $|\phi'_0\rangle = |\phi_0\rangle$*, and for all* $i = 1, \ldots, r$*, set* $|\phi_i\rangle = U_i|\phi_{i-1}\rangle$ *and* $|\phi'_i\rangle = U'_i|\phi'_{i-1}\rangle$*. For all* $i = 0, 1, \ldots, r$*, set* $|\phi_i\rangle = \alpha_{i,y}|\phi_{i,y}\rangle + \alpha_{i,\overline{y}}|\phi_{i,\overline{y}}\rangle$*, where* $|\phi_{i,y}\rangle$ *(resp.* $|\phi_{i,\overline{y}}\rangle$*)*

is a normalized superposition of configurations where f_i equals (resp. does not equal) y. Denote $|\phi'_i\rangle$ similarly.

Then the following holds:

(1) $\||\phi'_r\rangle - |\phi_r\rangle\| \leq 2\sum_{i=0}^{r-1} |\alpha_{i,y}|$ *for all* $y \in S$.

(2) $2(1 - |\alpha_{r,y}| - |\alpha'_{r,\overline{y}}|) \leq \||\phi'_r\rangle - |\phi_r\rangle\|^2$ *for all* $y \in S$.

(3) $N - \sqrt{N} - \sum_{y\in S} |\alpha'_{r,\overline{y}}| \leq 2r^2$.

Proof. We divide the proof into three parts.

Proof of (1): For all $y \in S$ and all $i = 1, \ldots, r$ we have

$$\begin{aligned} U'_i|\phi_{i-1}\rangle &= U'_i\left(\alpha_{i-1,y}|\phi_{i-1,y}\rangle + \alpha_{i-1,\overline{y}}|\phi_{i-1,\overline{y}}\rangle\right) \\ &= U'_i\left(\alpha_{i-1,y}|\phi_{i-1,y}\rangle\right) + U_i\left(\alpha_{i-1,\overline{y}}|\phi_{i-1,\overline{y}}\rangle\right) \\ &= U'_i\left(\alpha_{i-1,y}|\phi_{i-1,y}\rangle\right) - U_i\left(\alpha_{i-1,y}|\phi_{i-1,y}\rangle\right) + U_i|\phi_{i-1}\rangle \\ &= |\phi_i\rangle + \left(U'_i - U_i\right)\left(\alpha_{i-1,y}|\phi_{i-1,y}\rangle\right). \end{aligned}$$

Hence, by mathematical induction on i,

$$|\phi'_i\rangle = U'_i \ldots U'_1|\phi_0\rangle = |\phi_i\rangle + \sum_{j=1}^{i}(U'_i \ldots U'_{j+1})(U'_j - U_j)\left(\alpha_{j-1,y}|\phi_{j-1,y}\rangle\right),$$

so,

$$\begin{aligned} \||\phi'_i\rangle - |\phi_i\rangle\| &= \|\sum_{j=1}^{i}(U'_i \ldots U'_{j+1})(U'_j - U_j)\left(\alpha_{j-1,y}|\phi_{j-1,y}\rangle\right)\| \\ &\leq 2\sum_{j=1}^{i}|\alpha_{j-1,y}|, \end{aligned}$$

and (1) follows.

Proof of (2): The inequality follows from:

$$\begin{aligned}
\||\phi'_r\rangle - |\phi_r\rangle\|^2 &= \|(\alpha'_{r,y}|\phi'_{r,y}\rangle + \alpha'_{r,\overline{y}}|\phi'_{r,\overline{y}}\rangle) - (\alpha_{r,y}|\phi_{r,y}\rangle + \alpha_{r,\overline{y}}|\phi_{r,\overline{y}}\rangle)\|^2 \\
&= \|(\alpha'_{r,y}|\phi'_{r,y}\rangle - \alpha_{r,y}|\phi_{r,y}\rangle) + (\alpha'_{r,\overline{y}}|\phi'_{r,\overline{y}}\rangle - \alpha_{r,\overline{y}}|\phi_{r,\overline{y}}\rangle)\|^2 \\
&= \|\alpha'_{r,y}|\phi'_{r,y}\rangle - \alpha_{r,y}|\phi_{r,y}\rangle\|^2 + \|\alpha'_{r,\overline{y}}|\phi'_{r,\overline{y}}\rangle - \alpha_{r,\overline{y}}|\phi_{r,\overline{y}}\rangle\|^2 \\
&\geq (\,|\alpha'_{r,y}|^2 + |\alpha_{r,y}|^2 - 2|\alpha'_{r,y}||\alpha_{r,y}|\,) \\
&\quad + (\,|\alpha'_{r,\overline{y}}|^2 + |\alpha_{r,\overline{y}}|^2 - 2|\alpha'_{r,\overline{y}}||\alpha_{r,\overline{y}}|\,) \\
&= 2(1 - |\alpha'_{r,y}||\alpha_{r,y}| - |\alpha'_{r,\overline{y}}||\alpha_{r,\overline{y}}|) \\
&\geq 2(1 - |\alpha_{r,y}| - |\alpha'_{r,\overline{y}}|),
\end{aligned}$$

where the two inequalities follow from proposition 4.8.1 and the fact that the norm of any scalar is at most 1.

Proof of (3): By (2), (1), and proposition 4.8.2,

$$1 - |\alpha_{r,y}| - |\alpha'_{r,\overline{y}}| \;\leq\; \frac{1}{2}\||\phi'_r\rangle - |\phi_r\rangle\|^2 \;\leq\; 2\left(\sum_{i=0}^{r-1}|\alpha_{i,y}|\right)^2 \;\leq\; 2r\sum_{i=0}^{r-1}|\alpha_{i,y}|^2.$$

Thus,

$$\begin{aligned}
\sum_{y\in S}\left(1 - |\alpha_{r,y}| - |\alpha'_{r,\overline{y}}|\right) &\leq \sum_{y\in S}\left(2r\sum_{i=0}^{r-1}|\alpha_{i,y}|^2\right) \\
&= 2r\sum_{i=0}^{r-1}\left(\sum_{y\in S}|\alpha_{i,y}|^2\right) \;\leq\; 2r^2.
\end{aligned}$$

Since

$$\begin{aligned}
\sum_{y\in S}\left(1 - |\alpha_{r,y}| - |\alpha'_{r,\overline{y}}|\right) &= N - \sum_{y\in S}|\alpha_{r,y}| - \sum_{y\in S}|\alpha'_{r,\overline{y}}| \\
&\geq N - \sqrt{N}\left(\sum_{y\in S}|\alpha_{r,y}|^2\right)^{1/2} - \sum_{y\in S}|\alpha'_{r,\overline{y}}| \\
&= N - \sqrt{N} - \sum_{y\in S}|\alpha'_{r,\overline{y}}|,
\end{aligned}$$

we have

$$N - \sqrt{N} - \sum_{y \in S} |\alpha'_{r,\bar{y}}| \;\leq\; \sum_{y \in S} (1 - |\alpha_{r,y}| - |\alpha'_{r,\bar{y}}|) \;\leq\; 2r^2,$$

and (3) follows. ∎

Theorem 4.8.4 *Let S be any set of N strings, and M be any oracle quantum machine with bounded error probability. Let $y \in S$ be a randomly and uniformly chosen element from S. Let F be the oracle such that $F(x) = 1$ if and only if $x = y$. Then the average number of times M must query F in order to determine y with probability at least 50% is at least $\left\lfloor (\sin \frac{\pi}{8})\sqrt{N} \right\rfloor$, where the average is taken over all possible y.*

Proof. Let S be any set of N strings and $\mathcal{C}$ be any configuration space. Let $|\psi_0\rangle$ be any superposition of configurations, and M any bounded-error oracle quantum machine. Given any oracle $F^\star$, assume that we run $M^{F^\star}$ for s steps, and assume that M queries r times its oracle $F^\star$ during the computation. Since we will only run M using oracle $F^\star$ with $F^\star(x) = 0$ if $x \notin S$, without loss of generality, assume that M never queries $F^\star$ on strings not in S.

First, consider the case that we run M using the trivial oracle: let F be the oracle such that $F(x) = 0$ for all $x \in S$, and let

$$|\psi_s\rangle = A_s \ldots A_1 |\psi_0\rangle \tag{4.6}$$

be the unitary transformation corresponding to the computation of M using oracle F.

For all $i = 1, \ldots, r$, let q_i be the time stamp for M's i–th query, and set $q_{r+1} = s + 1$. Then eq. (4.6) can also be written as

$$|\phi_r\rangle = U_r \ldots U_1 |\phi_0\rangle \tag{4.7}$$

where $|\phi_0\rangle = A_{q_1-1}\ldots A_1|\psi_0\rangle$, and for all $i = 1,\ldots,r$, $U_i = A_{q_{i+1}-1}\ldots A_{q_i}$ and $|\phi_i\rangle = U_i|\phi_{i-1}\rangle$. At the i–th query some configurations will query F, some will not. For all $i = 0,\ldots,r-1$, set $f_i(|c\rangle) = x$ if $|c\rangle$ queries F on x at the $(i+1)$–st query.

Now, consider what happens if we flip one of the oracle bits: Given any $y \in S$, let F' be the oracle such that $F'(x) = 1$ if and only if $x = y$. Then the computation of $M^{F'}$ corresponds to the unitary transformation

$$|\phi'_r\rangle = U'_r\ldots U'_1|\phi_0\rangle$$

where $U'_i|c\rangle = U_i|c\rangle$ if $f_{i-1}(|c\rangle) \neq y$.

At the end of the computation of $M^{F'}$, we measure the superposition $|\phi'_r\rangle$ in order to determine the unknown y. For each configuration $|c\rangle \in \mathcal{C}$, set $f_r(|c\rangle) = x$ if, by measuring $|c\rangle$, M answers that x is the unknown y.

Set $|\phi'_r\rangle = \alpha'_{r,y}|\phi'_{r,y}\rangle + \alpha'_{r,\overline{y}}|\phi'_{r,\overline{y}}\rangle$ where $|\phi'_{r,y}\rangle$ (resp. $|\phi'_{r,\overline{y}}\rangle$) is the normalized superposition of configurations where f_r equals (resp. does not equal) y. Then $|\alpha'_{r,y}|^2$ is the probability that $M^{F'}$ correctly determines y. Since, by assumption, this probability is at least 50%,

$$|\alpha'_{r,\overline{y}}| \leq \frac{1}{\sqrt{2}} \quad \text{for all} \quad y \in S. \tag{4.8}$$

Furthermore, by Lemma 4.8.3,

$$N - \sqrt{N} - \sum_{y\in S} |\alpha'_{r,\overline{y}}| \;\leq\; 2r^2.$$

Hence, by eq. (4.8)

$$2r^2 \;\geq\; N - \sqrt{N} - \frac{1}{\sqrt{2}}N \;=\; \left(1 - \frac{1}{\sqrt{2}}\right)N - \sqrt{N}.$$

It follows by straightforward algebra that

$$r \geq \frac{\sqrt{2-\sqrt{2}}}{2}\sqrt{N} - 1 = (\sin\tfrac{\pi}{8})\sqrt{N} - 1 \tag{4.9}$$

provided $N \geq 15$. But eq. 4.9 holds nevertheless for all N because the oracle must be queried at least once to succeed with probability at least 50% when $N > 2$, and therefore $r \geq 1 \geq (\sin\frac{\pi}{8})\sqrt{N} - 1$ holds as required for $2 < N < 15$. In addition, the equation holds vacuously when $N \leq 2$ since $r \geq 0 \geq (\sin\frac{\pi}{8})\sqrt{N} - 1$ in that case. The theorem follows directly from the generality of eq. 4.9. ∎

Theorem 4.8.4 gives a lower bound for finding a unique solution using a bounded-error quantum machine. However, in most applications we would expect that there will be more than one solution. Furthermore, we might even not know if there is a solution at all. Let t be the number of solutions. For the case $t \geq 1$, we have the following theorem.

Theorem 4.8.5 *Let S be any set of N strings, and M be any bounded-error oracle quantum machine. Let $A \subseteq S$ be a randomly and uniformly chosen subset of S of size t, $t \geq 1$. Let F be the oracle such that $F(x) = 1$ if and only if $x \in A$. Then the average number of times M must query F in order to determine some member $y \in A$ with probability at least 50% is at least $\left\lfloor (\sin\frac{\pi}{8})\sqrt{\lfloor N/t \rfloor} \right\rfloor$, where the average is taken over all possible A of size t.*

The proof of this theorem is almost identical to the proof of Lemma 4.8.3 and Theorem 4.8.4. In Lemma 4.8.3, eqs. (1) and (2) now hold for all subsets of t strings. Hence, by choosing a largest number of such disjoint subsets from S, say R of cardinality $N_t = \lfloor N/t \rfloor$, in the proof of (3), we obtain

$$N_t - \sqrt{N_t} - \sum_{X_i \in R} |\alpha'_{r,\overline{X_i}}| \leq 2r^2.$$

The remaining part of the proof is the same as the proof of Theorem 4.8.4, only with obvious and minor changes.

4.9 Conclusions and future directions

We have provided a tight analysis of Grover's quantum search algorithm and proved that it comes to within a few percent of being optimal in terms of the number of times the function must be evaluated when it is provided as a black box (or an oracle). Moreover, we showed how to apply the algorithm even when the number of solutions is unknown ahead of time. It would be interesting to determine if in fact Grover's algorithm is exactly optimal or whether it is possible to improve it slightly. Also, a lower bound on the *expected* number of function evaluations required to find the solution by any quantum algorithm would be useful. How would it compare with our upper bound $0.69003\sqrt{N}$?

Grover's algorithm and the ideas presented in this paper can be extended in several directions, which we are currently investigating and will be the topic of a subsequent paper. In particular, Grover's algorithm can be thought of in a more general setting than quantum searching. Each iteration of the algorithm can be used to amplify the amplitude of a desired state. From this perspective, Grover's algorithm is really an *amplitude amplification* process.

It would be silly to use Grover's algorithm directly to solve most **NP**–complete problems because there are classical heuristics that would go faster on almost all instances. We are currently investigating the extent by which these heuristics can be sped up on a quantum computer by way of amplitude amplification. In many cases, we can combine the classical heuristics with amplitude amplification to allow quadratic speed up compared to the best classical heuristics available, but we do not yet know how general this phenomenon is. Similarly, more efficient quantum algorithms might exist for specific **NP**–complete problems if the structure of the problem is exploited. Furthermore, we are investigating how to use ideas from Grover's algorithm to solve problems higher than **NP** in the polynomial-time hierarchy.

Assume $F : X_N \to \{0, 1\}$ is as in our paper but our goal is to determine the number t of $i \in X_N$ such that $F(i) = 1$ rather than finding a specific one. In light

of the theory of #**P**–completeness, this is thought to be a harder problem for classical computers. Combining Grover's algorithm with some ideas from Shor's quantum factoring algorithm [101], we have preliminary results that indicate the possibility of solving this *quantum counting* problem with high probability in a time in $O(t\sqrt{N})$ without need for a large supply of auxiliary quantum memory. If we are satisfied with an approximate answer, a time in $O(\sqrt{N})$ provides an answer whose absolute error is bounded by $\sqrt{t}$ with high probability, and a time in $O(\sqrt{N/t})$ suffices to count with small expected relative error.

We presented in the Section 4.2 an application of Grover's algorithm to the cryptanalysis of secret-key cryptosystems such as the DES. Can quantum computing be used in more subtle ways for cryptanalytical purposes, for instance when double or triple-key encipherment is used? What is the best way to use quantum searching for finding collisions in a cryptographic hash function?

4.10 Acknowledgements

We are grateful to Richard Cleve for telling us how to implement one iteration of Grover's algorithm with a single function evaluation, and to Lov Grover for pointing out that our lower bound would not apply to the *expected* number of function evaluations to succeed with a given probability. The third author would like to thank Edmund Christiansen for helpful discussions concerning recursion equations, and Joan Boyar for helpful discussions in general.

Chapter 5

Quantum amplitude amplification and estimation

This chapter reproduces the improved final version of the article *Quantum Counting* presented at the *25th Annual International Colloquium on Automata, Languages and Programming* [35] in Aalborg, Denmark in 1998. It has been written in collaboration of Gilles Brassard, Peter Høyer and Michele Mosca.

5.1 Abstract

Consider a Boolean function $\chi : X \to \{0,1\}$ that partitions set X between its *good* and *bad* elements, where x is good if $\chi(x) = 1$ and bad otherwise. Consider also a quantum algorithm $\mathcal{A}$ such that $\mathcal{A}|0\rangle = \sum_{x \in X} \alpha_x |x\rangle$ is a quantum superposition of the elements of X, and let a denote the probability that a good element is produced if $\mathcal{A}|0\rangle$ is measured. If we repeat the process of running $\mathcal{A}$, measuring the output, and using χ to check the validity of the result, we shall expect to repeat $1/a$ times on the average before a solution is found. *Quantum amplification* is a process that allows to find a good x after an expected number of applications of $\mathcal{A}$ and its inverse which is pro-

portional to $1/\sqrt{a}$, assuming algorithm $\mathcal{A}$ makes no measurements. This is a generalization of Grover's searching algorithm in which $\mathcal{A}$ was restricted to producing an equal superposition of all members of X and we had a promise that a single x existed such that $\chi(x) = 1$. Our algorithm works whether or not the value of a is known ahead of time. In case the value of a is known, we can find a good x after a number of applications of $\mathcal{A}$ and its inverse which is proportional to $1/\sqrt{a}$ even in the worst case. We show that this quadratic speedup can also be obtained for a large family of search problems for which good classical heuristics exist. Finally, as our main result, we combine ideas from Grover's and Shor's quantum algorithms to perform *amplitude estimation*, a process that allows to estimate the value of a. We apply amplitude estimation to the problem of *approximate counting*, in which we wish to estimate the number of $x \in X$ such that $\chi(x) = 1$. We obtain optimal quantum algorithms in a variety of settings.

5.2 Introduction

Quantum computing is a field at the junction of theoretical modern physics and theoretical computer science. Practical experiments involving a few quantum bits have been successfully performed, and much progress has been achieved in quantum information theory, quantum error correction and fault tolerant quantum computation. Although we are still far from having desktop quantum computers in our offices, the quantum computational paradigm could soon be more than mere theoretical exercise.

The discovery by Peter Shor [101] of a polynomial-time quantum algorithm for factoring and computing discrete logarithms was a major milestone in the history of quantum computing. Another significant result is Lov Grover's quantum search

algorithm [69, 71]. Grover's algorithm does not solve **NP**–complete problems in polynomial time, but the wide range of its applications more than compensates for this.

In this paper, we generalize Grover's algorithm in a variety of directions. Consider a problem that is characterized by a Boolean function $\chi(x, y)$ in the sense that y is a good solution to instance x if and only if $\chi(x, y) = 1$. (There could be more than one good solution to a given instance.) If we have a probabilistic algorithm $\mathcal{P}$ that outputs a guess $\mathcal{P}(x)$ on input x, we can call $\mathcal{P}$ and χ repeatedly until a solution to instance x is found. If $\chi(x, \mathcal{P}(x)) = 1$ with probability $p_x > 0$, we expect to repeat this process $1/p_x$ times on the average. Consider now the case when we have a quantum algorithm $\mathcal{A}$ instead of the probabilistic algorithm. Assume $\mathcal{A}$ makes no measurements: instead of a classical answer, it produces quantum superposition $|\Psi_x\rangle$ when run on input x. Let a_x denote the probability that $|\Psi_x\rangle$, *if measured*, would be a good solution. If we repeat the process of running $\mathcal{A}$ on x, measuring the output, and using χ to check the validity of the result, we shall expect to repeat $1/a_x$ times on the average before a solution is found. This is no better than the classical probabilistic paradigm.

In Section 5.3, we describe a more efficient approach to this problem, which we call amplitude amplification. Intuitively, the probabilistic paradigm increases the probability of success roughly by a constant on each iteration; by contrast, amplitude amplification increases the *amplitude* of success roughly by a constant on each iteration. Because amplitudes correspond to square roots of probabilities, it suffices to repeat the amplitude amplification process approximately $\sqrt{1/a_x}$ times to achieve success with overwhelming probability. For simplicity, we assume in the rest of this paper that there is a single instance for which we seek a good solution, which allows us to dispense from input x, but the generalization to the paradigm outlined above is straightforward. Grover's original database searching quantum algorithm is a special case of this process, in which χ is given by a function $f : \{0, 1, \ldots, N-1\} \to \{0, 1\}$ for which we are promised that there exists

a unique x_0 such that $f(x_0) = 1$. If we use the Fourier transform as quantum algorithm A—or more simply the Walsh–Hadamard transform in case N is a power of 2—an equal superposition of all possible x's is produced, whose success probability would be $1/N$ if measured. Classical repetition would succeed after an expected number N of evaluations of f. Amplitude amplification corresponds to Grover's algorithm: it succeeds after approximately $\sqrt{N}$ evaluations of the function.

We generalize this result further to the case when the probability of success a of algorithm $\mathcal{A}$ is not known ahead of time: it remains sufficient to evaluate $\mathcal{A}$ and χ an expected number of times that is proportional to $\sqrt{1/a}$. Moreover, in the case a is known ahead of time, we give two different techniques that are guaranteed to find a good solution after a number of iterations that is proportional to $\sqrt{1/a}$ in the worst case.

It can be proven that Grover's algorithm goes quadratically faster than any possible classical algorithm when function f is given as a black-box. However, it is usually the case in practice that information is known about f that allows us to solve the problem much more efficiently than by exhaustive search. The use of classical *heuristics*, in particular, will often yield a solution significantly more efficiently than straight quantum amplitude amplification would. In Section 5.4, we consider a broad class of classical heuristics and show how to apply amplitude amplification to obtain quadratic speedup compared to any such heuristic.

Finally, Section 5.5 addresses the question of estimating the success probability a of quantum algorithm $\mathcal{A}$. We call this process *amplitude estimation*. As a special case of our main result (Theorem 5.5.6), an estimate for a is obtained after any number M of iterations which is within $2\pi\sqrt{a(1-a)}/M + \pi^2/M^2$ of the correct value with probability at least $8/\pi^2$, where one iteration consists of running algorithm $\mathcal{A}$ once forwards and once backwards, and of computing function χ once. As application of this technique, we show how to approximately count the number of x such that $f(x) = 1$ given a function $f : \{0, 1, \ldots, N-1\} \rightarrow$

$\{0,1\}$. If the correct answer is $t > 0$, it suffices to compute the function $\sqrt{N}$ times to obtain an estimate roughly within $\sqrt{t}$ of the correct answer. A number of evaluations of f proportional to $\frac{1}{\epsilon}\sqrt{N/t}$ yields a result that is likely to be within ϵt of the correct answer. (We can do slightly better in case ϵ is not fixed.) If it is known ahead of time that the correct answer is either $t = 0$ or $t = t_0$ for some fixed t_0, we can determine which is the case with certainty using a number of evaluations of f proportional to $\sqrt{N/t_0}$. If we have no prior knowledge about t, the exact count can be obtained with high probability after a number of evaluations of f that is proportional to $\sqrt{t(N-t)}$ when $0 < t < N$ and $\sqrt{N}$ otherwise. Most of these results are optimal.

We assume in this paper that the reader is familiar with basic notions of quantum computing.

5.3 Quantum amplitude amplification

Suppose we have a classical randomized algorithm that succeeds with some probability a. If we repeat the algorithm, say, j times, then our probability of success increases to roughly ja (assuming $ja \ll 1$). Intuitively, we can think of this strategy as each additional run of the given algorithm boosting the probability of success by an additive amount of roughly a.

A quantum analogue of boosting the probability of success would be to boost the *amplitude* of being in a certain subspace of a Hilbert space. The general concept of amplifying the amplitude of a subspace was discovered by Brassard and Høyer [33] as a generalization of the boosting technique applied by Grover in his original quantum searching paper [69]. Following [33] and [24], we refer to their idea as *amplitude amplification* and detail the ingredients below.

Let $\mathcal{H}$ denote the Hilbert space representing the state space of a quantum system. Every Boolean function $\chi : \mathbb{Z} \to \{0,1\}$ induces a partition of $\mathcal{H}$ into a direct sum of two subspaces, a good subspace and a bad subspace. The *good subspace* is the

subspace spanned by the set of basis states $|x\rangle \in \mathcal{H}$ for which $\chi(x) = 1$, and the *bad subspace* is its orthogonal complement in $\mathcal{H}$. We say that the elements of the good subspace are *good*, and that the elements of the bad subspace are *bad*.

Every pure state $|\Upsilon\rangle$ in $\mathcal{H}$ has a unique decomposition as $|\Upsilon\rangle = |\Upsilon_1\rangle + |\Upsilon_0\rangle$, where $|\Upsilon_1\rangle$ denotes the projection onto the good subspace, and $|\Upsilon_0\rangle$ denotes the projection onto the bad subspace. Let $a_\Upsilon = \langle\Upsilon_1|\Upsilon_1\rangle$ denote the probability that measuring $|\Upsilon\rangle$ produces a good state, and similarly, let $b_\Upsilon = \langle\Upsilon_0|\Upsilon_0\rangle$. Since $|\Upsilon_1\rangle$ and $|\Upsilon_0\rangle$ are orthogonal, we have $a_\Upsilon + b_\Upsilon = 1$.

Let $\mathcal{A}$ be any quantum algorithm that acts on $\mathcal{H}$ and uses no measurements. Let $|\Psi\rangle = \mathcal{A}|0\rangle$ denote the state obtained by applying $\mathcal{A}$ to the initial zero state. The amplification process is realized by repeatedly applying the following unitary operator [33] on the state $|\Psi\rangle$,

$$\mathbf{Q} = \mathbf{Q}(\mathcal{A}, \chi) = -\mathcal{A}\mathbf{S}_0\mathcal{A}^{-1}\mathbf{S}_\chi. \tag{5.1}$$

Here, the operator $\mathbf{S}_\chi$ conditionally changes the sign of the amplitudes of the good states,

$$|x\rangle \longmapsto \begin{cases} -|x\rangle & \text{if } \chi(x) = 1 \\ |x\rangle & \text{if } \chi(x) = 0, \end{cases}$$

while the operator $\mathbf{S}_0$ changes the sign of the amplitude if and only if the state is the zero state $|0\rangle$. The operator $\mathbf{Q}$ is well-defined since we assume that $\mathcal{A}$ uses no measurements and, therefore, $\mathcal{A}$ has an inverse.

The usefulness of operator $\mathbf{Q}$ stems from its simple action on the subspace $\mathcal{H}_\Psi$ spanned by the vectors $|\Psi_1\rangle$ and $|\Psi_0\rangle$.

Lemma 5.3.1 *We have that*

$$\begin{aligned} \mathbf{Q}|\Psi_1\rangle &= (1-2a)|\Psi_1\rangle - 2a|\Psi_0\rangle \\ \mathbf{Q}|\Psi_0\rangle &= 2(1-a)|\Psi_1\rangle + (1-2a)|\Psi_0\rangle, \end{aligned}$$

where $a = \langle\Psi_1|\Psi_1\rangle$.

It follows that the subspace $\mathcal{H}_\Psi$ is stable under the action of $\mathbf{Q}$, a property that was first observed by Brassard and Høyer [33] and rediscovered by Grover [72].

Suppose $0 < a < 1$. Then $\mathcal{H}_\Psi$ is a subspace of dimension 2, and otherwise $\mathcal{H}_\Psi$ has dimension 1. The action of $\mathbf{Q}$ on $\mathcal{H}_\Psi$ is also realized by the operator

$$\mathbf{U}_\Psi \mathbf{U}_{\Psi_0}, \tag{5.2}$$

which is composed of 2 reflections. The first operator, $\mathbf{U}_{\Psi_0} = \mathbf{I} - \frac{2}{1-a}|\Psi_0\rangle\langle\Psi_0|$, implements a reflection through the ray spanned by the vector $|\Psi_0\rangle$, while the second operator $\mathbf{U}_\Psi = \mathbf{I} - 2|\Psi\rangle\langle\Psi|$ implements a reflection through the ray spanned by the vector $|\Psi\rangle$.

Consider the orthogonal complement $\mathcal{H}_\Psi^\perp$ of $\mathcal{H}_\Psi$ in $\mathcal{H}$. Since the operator $\mathcal{A}\mathbf{S}_0\mathcal{A}^{-1}$ acts as the identity on $\mathcal{H}_\Psi^\perp$, operator $\mathbf{Q}$ acts as $-\mathbf{S}_\chi$ on $\mathcal{H}_\Psi^\perp$. Thus, $\mathbf{Q}^2$ acts as the identity on $\mathcal{H}_\Psi^\perp$, and every eigenvector of $\mathbf{Q}$ in $\mathcal{H}_\Psi^\perp$ has eigenvalue $+1$ or -1. It follows that to understand the action of $\mathbf{Q}$ on an arbitrary initial vector $|\Upsilon\rangle$ in $\mathcal{H}$, it suffices to consider the action of $\mathbf{Q}$ on the projection of $|\Upsilon\rangle$ onto $\mathcal{H}_\Psi$.

Since operator $\mathbf{Q}$ is unitary, the subspace $\mathcal{H}_\Psi$ has an orthonormal basis consisting of two eigenvectors of $\mathbf{Q}$,

$$|\Psi_\pm\rangle = \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{a}}|\Psi_1\rangle \pm \frac{\imath}{\sqrt{1-a}}|\Psi_0\rangle\right), \tag{5.3}$$

where $\imath = \sqrt{-1}$ denotes the principal square root of -1. The corresponding eigenvalues are

$$\lambda_\pm = e^{\pm\imath 2\theta_a}, \tag{5.4}$$

where the angle θ_a is defined so that

$$\sin^2(\theta_a) = a = \langle\Psi_1|\Psi_1\rangle \tag{5.5}$$

and $0 \leq \theta_a \leq \pi/2$.

We use operator $\mathbf{Q}$ to boost the success probability a of the quantum algorithm $\mathcal{A}$. First, express $|\Psi\rangle = \mathcal{A}|0\rangle$ in the eigenvector basis,

$$\mathcal{A}|0\rangle = |\Psi\rangle = \frac{-\imath}{\sqrt{2}}\left(e^{\imath\theta_a}|\Psi_+\rangle - e^{-\imath\theta_a}|\Psi_-\rangle\right). \tag{5.6}$$

It is now immediate that after j applications of operator $\mathbf{Q}$, the state is

$$\mathbf{Q}^j|\Psi\rangle = \frac{-\imath}{\sqrt{2}}\left(e^{(2j+1)\imath\theta_a}|\Psi_+\rangle - e^{-(2j+1)\imath\theta_a}|\Psi_-\rangle\right) \tag{5.7}$$

$$= \frac{1}{\sqrt{a}}\sin((2j+1)\theta_a)\,|\Psi_1\rangle + \frac{1}{\sqrt{1-a}}\cos((2j+1)\theta_a)\,|\Psi_0\rangle. \tag{5.8}$$

It follows that if $0 < a < 1$ and if we compute $\mathbf{Q}^m|\Psi\rangle$ for some integer $m \geq 0$, then a final measurement will produce a good state with probability equal to $\sin^2((2m+1)\theta_a)$.

If the initial success probability a is either 0 or 1, then the subspace $\mathcal{H}_\Psi$ spanned by $|\Psi_1\rangle$ and $|\Psi_0\rangle$ has dimension 1 only, but the conclusion remains the same: If we measure the system after m rounds of amplitude amplification, then the outcome is good with probability $\sin^2((2m+1)\theta_a)$, where the angle θ_a is defined so that Equation 5.5 is satisfied and so that $0 \leq \theta_a \leq \pi/2$.

Therefore, assuming $a > 0$, to obtain a high probability of success, we want to choose integer m such that $\sin^2((2m+1)\theta_a)$ is close to 1. Unfortunately, our ability to choose m wisely depends on our knowledge about θ_a, which itself depends on a. The two extreme cases are when we know the exact value of a, and when we have no prior knowledge about a whatsoever.

Suppose the value of a is known. If $a > 0$, then by letting $m = \lfloor \pi/4\theta_a \rfloor$, we have that $\sin^2((2m+1)\theta_a) \geq 1 - a$, as shown in [24]. The next theorem is immediate.

Theorem 5.3.2 (Quadratic speedup) *Let $\mathcal{A}$ be any quantum algorithm that uses no measurements, and let $\chi : \mathbb{Z} \to \{0,1\}$ be any Boolean function. Let a be the initial success probability of $\mathcal{A}$. Suppose $a > 0$, and set $m = \lfloor \pi/4\theta_a \rfloor$,*

where θ_a *is defined so that* $\sin^2(\theta_a) = a$ *and* $0 < \theta_a \leq \pi/2$. *Then, if we compute* $\mathbf{Q}^m\mathcal{A}\,|0\rangle$ *and measure the system, the outcome is good with probability at least* $\max(1-a, a)$.

Note that any implementation of algorithm $\mathbf{Q}^m\mathcal{A}|0\rangle$ requires that the value of a is known so that the value of m can be computed. We refer to Theorem 5.3.2 as a quadratic speedup, or the square-root running-time result. The reason for this is that if an algorithm $\mathcal{A}$ has success probability $a > 0$, then after an expected number of $1/a$ applications of $\mathcal{A}$, we will find a good solution. Applying the above theorem reduces this to an expected number of at most $(2m+1)/\max(1-a, a) \in \Theta(\frac{1}{\sqrt{a}})$ applications of $\mathcal{A}$ and $\mathcal{A}^{-1}$.

As an application of Theorem 5.3.2, consider the search problem [71] in which we are given a Boolean function $f : \{0, 1, \ldots, N-1\} \to \{0, 1\}$ satisfying the promise that there exists a unique $x_0 \in \{0, 1, \ldots, N-1\}$ on which f takes value 1, and we are asked to find x_0. If f is given as a black box, then on a classical computer, we need to evaluate f on an expected number of roughly half the elements of the domain in order to determine x_0.

By contrast, Grover [71] discovered a quantum algorithm that only requires an expected number of evaluations of f in the order of $\sqrt{N}$. In terms of amplitude amplification, Grover's algorithm reads as follows: Let $\chi = f$, and let $\mathcal{A} = \mathbf{W}$ be the Walsh-Hadamard transform on n qubits that maps the initial zero state $|0\rangle$ to $\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle$, an equally-weighted superposition of all $N = 2^n$ elements in the domain of f. Then the operator $\mathbf{Q} = -\mathcal{A}\mathbf{S}_0\mathcal{A}^{-1}\mathbf{S}_\chi$ is equal to the iterate $-\mathbf{W}\mathbf{S}_0\mathbf{W}\mathbf{S}_f$ applied by Grover in his searching paper [71]. The initial success probability a of $\mathcal{A}$ is exactly $1/N$, and if we measure after $m = \lfloor \pi/4\theta_a \rfloor$ iterations of $\mathbf{Q}$, the probability of measuring x_0 is lower bounded by $1 - 1/N$ [24].

Now, suppose that the value of a is not known. In Section 5.5, we discuss techniques for finding an estimate of a, whereafter one then can apply a weakened version of Theorem 5.3.2 in which the exact value of a is replaced by an estimate of it. Another idea is to try to find a good solution without prior computation

of an estimate of a. Within that approach, by adapting the ideas in Section 6 in [24] we can still obtain a quadratic speedup.

Theorem 5.3.3 (Quadratic speedup without knowing a) *There exists a quantum algorithm* **QSearch** *with the following property. Let $\mathcal{A}$ be any quantum algorithm that uses no measurements, and let $\chi : \mathbb{Z} \to \{0,1\}$ be any Boolean function. Let a denote the initial success probability of $\mathcal{A}$. Algorithm* **QSearch** *finds a good solution using an expected number of applications of $\mathcal{A}$ and $\mathcal{A}^{-1}$ which are in $\Theta(\frac{1}{\sqrt{a}})$ if $a > 0$, and otherwise runs forever.*

The algorithm in the above theorem utilizes the given quantum algorithm $\mathcal{A}$ as a subroutine and the operator $\mathbf{Q}$. The complete algorithm is as follows:

Algorithm(QSearch$(\mathcal{A}, \chi)$)

1. Set $l = 0$ and let c be any constant such that $1 < c < 2$.
2. Increase l by 1 and set $M = \lceil c^l \rceil$.
3. Apply $\mathcal{A}$ on the initial state $|0\rangle$, and measure the system. If the outcome $|z\rangle$ is good, that is, if $\chi(z) = 1$, then output z and stop.
4. Initialize a register of appropriate size to the state $\mathcal{A}|0\rangle$.
5. Pick an integer j between 1 and M uniformly at random.
6. Apply $\mathbf{Q}^j$ to the register, where $\mathbf{Q} = \mathbf{Q}(\mathcal{A}, \chi)$.
7. Measure the register. If the outcome $|z\rangle$ is good, then output z and stop. Otherwise, go to step 2.

The intuition behind this algorithm is as follows. In a 2-dimensional real vector space, if we pick a unit vector $(x, y) = (\cos(\cdot), \sin(\cdot))$ uniformly at random then the expected value of y^2 is $1/2$. Consider Equation 5.8. If we pick j at random between 1 and M for some integer M such that $M\theta_a$ is larger than, say, 100π,

then we have a good approximation to a random unit vector, and we will succeed with probability close to $1/2$.

To turn this intuition into an algorithm, the only obstacle left is that we do not know the value of θ_a, and hence do not know an appropriate value for M. However, we can overcome this by using exponentially increasing values of M, an idea similar to the one used in "exponential searching" (which is a term that does not refer to the running time of the method, but rather to an exponentially increasing growth of the size of the search space).

The correctness of algorithm **QSearch** is immediate and thus to prove the theorem, it suffices to show that the expected number of applications of $\mathcal{A}$ and $\mathcal{A}^{-1}$ is in the order of $1/\sqrt{a}$. This can be proven by essentially the same techniques applied in the proof of Theorem 3 in [24] and we therefore only give a very brief sketch of the proof.

On the one hand, if the initial success probability a is at least $3/4$, then step 3 ensures that we soon will measure a good solution. On the other hand, if $0 < a < 3/4$ then, for any given value of M, the probability of measuring a good solution in step 7 is lower bounded by

$$\frac{1}{2}\left(1 - \frac{1}{2M\sqrt{a}}\right), \tag{5.9}$$

and thus, as soon as M becomes larger than $1/\sqrt{a}$, each measurement yields a good solution with probability approaching $1/2$ as M grows. In particular, if $M \geq 1/(2c'\sqrt{a})$ where $c' > 0$ is such that $c = 2(1-c')$, then the measurement in step 7 fails to yield a good solution with probability at most $p = \frac{1}{2}(1+c')$. Such a failure costs another factor of $c = 2(1-c')$, but since $pc < 1$, the expected number of applications of $\mathcal{A}$ is still in $O(1/(2c'\sqrt{a}))$, giving the upper bound of $O\left(\frac{1}{\sqrt{a}}\right)$. For the lower bound, if M is in $o\left(\frac{1}{\sqrt{a}}\right)$, then the probability that we measure a good solution in step 7 is vanishing small. This completes our brief sketch of the proof of Theorem 5.3.3.

5.3.1 Quantum de-randomization when the success probability is known

We now consider the situation where the success probability a of the quantum algorithm $\mathcal{A}$ is known. If $a = 0$ or $a = 1$, then amplitude amplification will not change the success probability, so in the rest of this section, we assume that $0 < a < 1$. Theorem 5.3.2 allows us to boost the probability of success to at least $\max(1-a, a)$. A natural question to ask is whether it is possible to improve this to certainty, still given the value of a. It turns out that the answer is positive. This is unlike classical computers, where no such general de-randomization technique is known. We now describe 2 optimal methods for obtaining this, but other approaches are possible.

The first method is by applying amplitude amplification, not on the original algorithm $\mathcal{A}$, but on a slightly modified version of it. By Equation 5.8, if we measure the state $\mathbf{Q}^m\mathcal{A}|0\rangle$, then the outcome is good with probability $\sin^2((2m+1)\theta_a)$. In particular, if $\tilde{m} = \pi/4\theta_a - 1/2$ happens to be an integer, then we would succeed with certainty after $\tilde{m}$ applications of $\mathbf{Q}$. In general, $\overline{m} = \lceil \tilde{m} \rceil$ iterations is a fraction of 1 iteration too many, but we can compensate for that by choosing $\overline{\theta_a} = \pi/(4\overline{m}+2)$, an angle slightly smaller than θ_a. Any quantum algorithm that succeeds with probability $\overline{a}$ such that $\sin^2(\overline{\theta_a}) = \overline{a}$, will succeed with certainty after $\overline{m}$ iterations of amplitude amplification. Given $\mathcal{A}$ and its initial success probability a, it is easy to construct a new quantum algorithm that succeeds with probability $\overline{a} \leq a$: Let $\mathcal{B}$ denote the quantum algorithm that takes a single qubit in the initial state $|0\rangle$ and rotates it to the superposition $\sqrt{1-\overline{a}/a}\,|0\rangle + \sqrt{\overline{a}/a}\,|1\rangle$. Apply both $\mathcal{A}$ and $\mathcal{B}$, and define a good solution as one in which $\mathcal{A}$ produces a good solution, and the outcome of $\mathcal{B}$ is the state $|1\rangle$. Theorem 5.3.4 follows.

Theorem 5.3.4 (Quadratic speedup with known a) *Let $\mathcal{A}$ be any quantum algorithm that uses no measurements, and let $\chi : \mathbb{Z} \to \{0,1\}$ be any Boolean function. There exists a quantum algorithm that given the initial success probabil-*

ity $a > 0$ of $\mathcal{A}$, finds a good solution with certainty using a number of applications of $\mathcal{A}$ and $\mathcal{A}^{-1}$ which is in $\Theta(\frac{1}{\sqrt{a}})$.

The second method to obtain success probability 1 requires a generalization of operator $\mathbf{Q}$. Given angles $0 \leq \phi, \varphi < 2\pi$, redefine $\mathbf{Q}$ as follows,

$$\mathbf{Q} = \mathbf{Q}(\mathcal{A}, \chi, \phi, \varphi) = -\mathcal{A}\mathbf{S}_0(\phi)\mathcal{A}^{-1}\mathbf{S}_\chi(\varphi). \tag{5.10}$$

Here, the operator $\mathbf{S}_\chi(\varphi)$ is the natural generalization of the $\mathbf{S}_\chi$ operator,

$$|x\rangle \longmapsto \begin{cases} e^{i\varphi}|x\rangle & \text{if } \chi(x) = 1 \\ |x\rangle & \text{if } \chi(x) = 0. \end{cases}$$

Similarly, the operator $\mathbf{S}_0(\phi)$ multiplies the amplitude by a factor of $e^{i\phi}$ if and only if the state is the zero state $|0\rangle$. The action of operator $\mathbf{Q}(\mathcal{A}, \chi, \phi, \varphi)$ is also realized by applying an operator that is composed of two pseudo-reflections: the operator $\mathcal{A}\mathbf{S}_0(\phi)\mathcal{A}^{-1}$ and the operator $-\mathbf{S}_\chi(\varphi)$.

The next lemma shows that the subspace $\mathcal{H}_\Psi$ spanned by $|\Psi_1\rangle$ and $|\Psi_0\rangle$ is stable under the action of $\mathbf{Q}$, just as in the special case $\mathbf{Q}(\mathcal{A}, \chi, \pi, \pi)$ studied above.

Lemma 5.3.5 *Let $\mathbf{Q} = \mathbf{Q}(\mathcal{A}, \chi, \phi, \varphi)$. Then*

$$\begin{aligned} \mathbf{Q}|\Psi_1\rangle &= e^{i\varphi}((1 - e^{i\phi})a - 1)|\Psi_1\rangle + \quad e^{i\varphi}(1 - e^{i\phi})a|\Psi_0\rangle \\ \mathbf{Q}|\Psi_0\rangle &= \quad (1 - e^{i\phi})(1 - a)|\Psi_1\rangle - ((1 - e^{i\phi})a + e^{i\phi})|\Psi_0\rangle, \end{aligned}$$

where $a = \langle\Psi_1|\Psi_1\rangle$.

Let $\tilde{m} = \pi/4\theta_a - 1/2$, and suppose that $\tilde{m}$ is not an integer. In the second method to obtain a good solution with certainty, we also apply $\lceil\tilde{m}\rceil$ iterations of amplitude amplification, but now we slow down the speed of the very last iteration only, as opposed to of all iterations as in the first method. For the case $\tilde{m} < 1$, this second method has also been suggested by Chi and Kim [43]. We start by applying the operator $\mathbf{Q}(\mathcal{A}, \chi, \phi, \varphi)$ with $\phi = \varphi = \pi$ a number of $\lfloor\tilde{m}\rfloor$ times to the initial state

$|\Psi\rangle = \mathcal{A}|0\rangle$. By Equation 5.8, this produces the superposition

$$\frac{1}{\sqrt{a}} \sin\left((2\lfloor \tilde{m} \rfloor + 1)\theta_a\right) |\Psi_1\rangle + \frac{1}{\sqrt{1-a}} \cos\left((2\lfloor \tilde{m} \rfloor + 1)\theta_a\right) |\Psi_0\rangle.$$

Then, we apply operator $\mathbf{Q}$ one more time, but now using angles ϕ and φ, both between 0 and 2π, satisfying

$$e^{\imath\varphi}(1 - e^{\imath\phi})\sqrt{a} \sin\left((2\lfloor \tilde{m} \rfloor + 1)\theta_a\right) = ((1 - e^{\imath\phi})a + e^{\imath\phi}) \frac{1}{\sqrt{1-a}} \cos\left((2\lfloor \tilde{m} \rfloor + 1)\theta_a\right). \quad (5.11)$$

By Lemma 5.3.5, this ensures that the resulting superposition has inner product zero with $|\Psi_0\rangle$, and thus a subsequent measurement will yield a good solution with certainty.

The problem of choosing $\phi, \varphi \in \mathbb{R}$ such that Equation 5.11 holds is equivalent to requiring that

$$\cot\left((2\lfloor \tilde{m} \rfloor + 1)\theta_a\right) = e^{\imath\varphi} \sin(2\theta_a)\big(- \cos(2\theta_a) + \imath \cot(\phi/2)\big)^{-1}. \quad (5.12)$$

By appropriate choices of ϕ and φ, the right hand side of Equation 5.12 can be made equal to any nonzero complex number of norm at most $\tan(2\theta_a)$. Thus, since the left hand side of this equation is equal to some real number smaller than $\tan(2\theta_a)$, there exist $\phi, \varphi \in \mathbb{R}$ such that Equation 5.12 is satisfied, and hence also such that the expression in Equation 5.11 vanishes. In conclusion, applying $\mathbf{Q}(\mathcal{A}, \chi, \phi, \varphi)$ with such $\phi, \varphi \in \mathbb{R}$ at the very last iteration allows us to measure a good solution with certainty.

5.4 Heuristics

As explained in the previous section, using the amplitude amplification technique to search for a solution to a search problem, one obtains a quadratic speedup

compared to a brute force search. For many problems, however, good heuristics are known for which the expected running time, when applied to a "real-life" problem, is even smaller than any function in $O(\sqrt{N})$, where N is the size of the search space. This fact would make amplitude amplification much less useful unless a quantum computer is somehow able to take advantage of these classical heuristics. In this section we concentrate on a large family of classical heuristics that can be applied to search problems. We show how these heuristics can be incorporated into the general amplitude amplification process.

By a heuristic, we mean a probabilistic algorithm, running in polynomial time, that outputs what one is searching for with some non-negligible probability.

Suppose we have a family $\mathcal{F}$ of functions such that each $f \in \mathcal{F}$ is of the form $f : X \to \{0,1\}$. For a given function f we seek an input $x \in X$ such that $f(x) = 1$. A *heuristic* is a function $G : \mathcal{F} \times R \to X$, for an appropriate finite set R. The heuristic G uses a random seed $r \in R$ to generate a guess for an x such that $f(x) = 1$. For every function $f \in \mathcal{F}$, let $t_f = |\{x \in X \mid f(x) = 1\}|$, the number of good inputs x, and let $h_f = |\{r \in R \mid f(G(f,r)) = 1\}|$, the number of good seeds. We say that the heuristic is *efficient* for a given f if $h_f/|R| > t_f/|X|$, that is, if using G and a random seed to generate inputs to f succeeds with a higher probability than directly guessing inputs to f uniformly at random. The heuristic is *good* in general if

$$\mathrm{E}_{\mathcal{F}}\left(\frac{h_f}{|R|}\right) > \mathrm{E}_{\mathcal{F}}\left(\frac{t_f}{|X|}\right).$$

Here $\mathrm{E}_{\mathcal{F}}$ denotes the expectation over all f according to some fixed distribution. Note that for some f, h_f might be small but repeated uses of the heuristic, with seeds uniformly chosen in R, will increase the probability of finding a solution.

Theorem 5.4.1 *Let $\mathcal{F} \subseteq \{f \mid f : X \to \{0,1\}\}$ be a family of Boolean functions and $\mathcal{D}$ be a probability distribution over $\mathcal{F}$. If on a classical computer, using heuristic $G : \mathcal{F} \times R \to X$, one finds $x_0 \in X$ such that $f(x_0) = 1$ for random f*

taken from distribution D in expected time T then using a quantum computer, a solution can be found in expected time in $O(\sqrt{T})$.

Proof. A simple solution to this problem is to embed the classical heuristic G into the function used in the algorithm **QSearch**. Let $\chi(r) = f(G(f,r))$ and $x = G(f, \mathbf{QSearch}(\mathbf{W}, \chi))$, so that $f(x) = 1$. By Theorem 5.3.3, for each function $f \in \mathcal{F}$, we have an expected running time in $\Theta(\sqrt{|R|/h_f})$. Let P_f denote the probability that f occurs. Then $\sum_{f\in\mathcal{F}} P_f = 1$, and we have that the expected running time is in the order of $\sum_{f\in\mathcal{F}} \sqrt{|R|/h_f}\, P_f$, which can be rewritten as

$$\sum_{f\in\mathcal{F}} \sqrt{\frac{|R|}{h_f} P_f} \sqrt{P_f} \leq \left(\sum_{f\in\mathcal{F}} \frac{|R|}{h_f} P_f\right)^{1/2} \left(\sum_{f\in\mathcal{F}} P_f\right)^{1/2} = \left(\sum_{f\in\mathcal{F}} \frac{|R|}{h_f} P_f\right)^{1/2}$$

by Cauchy–Schwarz's inequality. ∎

An alternative way to prove Theorem 5.4.1 is to incorporate the heuristic into the operator $\mathcal{A}$ and do a minor modification to f. Let $\mathcal{A}$ be the quantum implementation of G. It is required that the operator $\mathcal{A}$ be unitary, but clearly in general the classical heuristic does not need to be reversible. As usual in quantum algorithms one will need first to modify the heuristic $G : \mathcal{F} \times R \to X$ to make it reversible, which can be done efficiently using standard techniques [14]. We obtain a reversible function $G'_f : R \times \mathbf{0} \to R \times X$. Let $\mathcal{A}$ be the natural unitary operation implementing G'_f and let us modify χ (the good set membership function) to consider only the second part of the register, that is $\chi((r,x)) = 1$ if and only if $f(x) = 1$. We then have that $a = h_f/|R|$ and by Theorem 5.3.3, for each function $f \in \mathcal{F}$, we have an expected running time in $\Theta(\sqrt{|R|/h_f})$. The rest of the reasoning is similar. This alternative technique shows, using a simple example, the usefulness of the general scheme of amplitude amplification described in the preceding section, although it is clear that from a computational point of view this is strictly equivalent to the technique given in the earlier proof of the theorem.

5.5 Quantum amplitude estimation

Section 5.3 dealt in a very general way with combinatorial search problems, namely, given a Boolean function $f : X \to \{0,1\}$ find an $x \in X$ such that $f(x) = 1$. In this section, we deal with the related problem of estimating $t = |\{x \in X \mid f(x) = 1\}|$, the number of inputs on which f takes the value 1.

We can describe this counting problem in terms of amplitude estimation: Using the notation of Section 5.3, given a unitary transformation $\mathcal{A}$ and a Boolean function χ, let $|\Psi\rangle = \mathcal{A}|0\rangle$. Write $|\Psi\rangle = |\Psi_1\rangle + |\Psi_0\rangle$ as a superposition of the good and bad components of $|\Psi\rangle$. Then *amplitude estimation* is the problem of estimating $a = \langle\Psi_1|\Psi_1\rangle$, the probability that a measurement of $|\Psi\rangle$ yields a good state.

The problem of estimating $t = |\{x \in X \mid f(x) = 1\}|$ can be formulated in these terms as follows. For simplicity, we take $X = \{0, 1, \ldots, N-1\}$. If N is a power of 2, then we set $\chi = f$ and $\mathcal{A} = \mathbf{W}$. If N is not a power of 2, we set $\chi = f$ and $\mathcal{A} = \mathbf{F}_N$, the quantum Fourier transform which, for every integer $M \geq 1$, is defined by

$$\mathbf{F}_M \; : \; |x\rangle \longmapsto \frac{1}{\sqrt{M}} \sum_{y=0}^{M-1} e^{2\pi i x y/M} |y\rangle \qquad (0 \leq x < M). \tag{5.13}$$

Then in both cases we have $a = t/N$, and thus an estimate for a directly translates into an estimate for t.

To estimate a, we make good use of the properties of operator $\mathbf{Q} = -\mathcal{A}\mathbf{S}_0\mathcal{A}^{-1}\mathbf{S}_f$. By Equation 5.8 in Section 5.3, we have that the amplitudes of $|\Psi_1\rangle$ and $|\Psi_0\rangle$ as functions of the number of applications of $\mathbf{Q}$, are sinusoidal functions, both of period $\frac{\pi}{\theta_a}$. Recall that $0 \leq \theta_a \leq \pi/2$ and $a = \sin^2(\theta_a)$, and thus an estimate for θ_a also gives an estimate for a.

To estimate this period, it is a natural approach to apply Fourier analysis like Shor [101] does for a classical function in his factoring algorithm. This approach

can be viewed as an eigenvalue estimation [81, 49] and is best analysed in the basis of eigenvectors of the operator at hand. By Equation 5.4, the eigenvalues of $\mathbf{Q}$ on the subspace spanned by $|\Psi_1\rangle$ and $|\Psi_0\rangle$ are $\lambda_+ = e^{\imath 2\theta_a}$ and $\lambda_- = e^{-\imath 2\theta_a}$. Thus we can estimate a simply by estimating one of these two eigenvalues. Errors in our estimate $\tilde{\theta}_a$ for θ_a translate into errors in our estimate $\tilde{a} = \sin^2(\tilde{\theta}_a)$ for a, as described in the next lemma.

Lemma 5.5.1 *Let $a = \sin^2(\theta_a)$ and $\tilde{a} = \sin^2(\tilde{\theta}_a)$ with $0 \leq \theta_a, \tilde{\theta}_a \leq 2\pi$ then*

$$\left|\tilde{\theta}_a - \theta_a\right| \leq \epsilon \;\Rightarrow\; |\tilde{a} - a| \leq 2\epsilon\sqrt{a(1-a)} + \epsilon^2 .$$

Proof. For $\epsilon \geq 0$, using standard trigonometric identities, we obtain

$$\begin{aligned} \sin^2(\theta_a + \epsilon) - \sin^2(\theta_a) &= \sqrt{a(1-a)}\sin(2\epsilon) + (1-2a)\sin^2(\epsilon) \text{ and} \\ \sin^2(\theta_a) - \sin^2(\theta_a - \epsilon) &= \sqrt{a(1-a)}\sin(2\epsilon) + (2a-1)\sin^2(\epsilon). \end{aligned}$$

The inequality follows directly. ∎

We want to estimate one of the eigenvalues of $\mathbf{Q}$. For this purpose, we utilize the following operator Λ. For any positive integer M and any unitary operator $\mathbf{U}$, the operator $\Lambda_M(\mathbf{U})$ is defined by

$$|j\rangle|y\rangle \longmapsto |j\rangle(\mathbf{U}^j|y\rangle) \qquad (0 \leq j < M). \tag{5.14}$$

Note that if $|\Psi\rangle$ is an eigenvector of $\mathbf{U}$ with eigenvalue $e^{2\pi\imath\omega}$, then $\Lambda_M(\mathbf{U})$ maps $|j\rangle|\Psi\rangle$ to $e^{2\pi\imath\omega j}|j\rangle|\Psi\rangle$.

Definition. 5.5.2 *For any integer $M > 0$ and real number $0 \leq \omega < 1$, let*

$$|\mathcal{S}_M(\omega)\rangle = \frac{1}{\sqrt{M}} \sum_{y=0}^{M-1} e^{2\pi\imath\omega y}\,|y\rangle.$$

We then have, for all $0 \leq x \leq M-1$

$$\mathbf{F}_M |x\rangle = |\mathcal{S}_M(x/M)\rangle.$$

The state $|\mathcal{S}_M(\omega)\rangle$ encodes the angle $2\pi\omega$ $(0 \leq \omega < 1)$ in the phases of an equally weighted superposition of all basis states. Different angles have different encodings, and the overlap between $|\mathcal{S}_M(\omega_0)\rangle$ and $|\mathcal{S}_M(\omega_1)\rangle$ is a measure for the distance between the two angles ω_0 and ω_1.

Definition. 5.5.3 *For any two real numbers* $\omega_0, \omega_1 \in \mathbb{R}$, *let*

$$d(\omega_0, \omega_1) = \min_{z \in \mathbb{Z}} \{|z + \omega_1 - \omega_0|\}.$$

Thus $2\pi d(\omega_0, \omega_1)$ is the length of the shortest arc on the unit circle going from $e^{2\pi\imath\omega_0}$ to $e^{2\pi\imath\omega_1}$.

Lemma 5.5.4 *For* $0 \leq \omega_0 < 1$ *and* $0 \leq \omega_1 < 1$ *let* $\Delta = d(\omega_0, \omega_1)$. *If* $\Delta = 0$ *we have* $|\langle \mathcal{S}_M(\omega_0) | \mathcal{S}_M(\omega_1) \rangle|^2 = 1$. *Otherwise*

$$|\langle \mathcal{S}_M(\omega_0) | \mathcal{S}_M(\omega_1) \rangle|^2 = \frac{\sin^2(M\Delta\pi)}{M^2 \sin^2(\Delta\pi)}.$$

Proof.

$$\begin{aligned} |\langle \mathcal{S}_M(\omega_0) | \mathcal{S}_M(\omega_1) \rangle|^2 &= \left| \left(\frac{1}{\sqrt{M}} \sum_{y=0}^{M-1} e^{-2\pi\imath\omega_0 y} \langle y| \right) \left(\frac{1}{\sqrt{M}} \sum_{y=0}^{M-1} e^{2\pi\imath\omega_1 y} |y\rangle \right) \right|^2 \\ &= \frac{1}{M^2} \left| \sum_{y=0}^{M-1} e^{2\pi\imath\Delta y} \right|^2 = \frac{\sin^2(M\Delta\pi)}{M^2 \sin^2(\Delta\pi)}. \end{aligned}$$

■

Consider the problem of estimating ω where $0 \leq \omega < 1$, given the state $|\mathcal{S}_M(\omega)\rangle$. If $\omega = x/M$ for some integer $0 \leq x < M$, then $\mathbf{F}_M^{-1} |\mathcal{S}_M(x/M)\rangle = |x\rangle$ by definition, and thus we have a perfect phase estimator. If $M\omega$ is not an integer, then

observing $\mathbf{F}_M^{-1}|\mathcal{S}_M(\omega)\rangle$ still provides a good estimation of ω, as shown in the following theorem.

Theorem 5.5.5 *Let X be the discrete random variable corresponding to the classical result of measuring $\mathbf{F}_M^{-1}|\mathcal{S}_M(\omega)\rangle$ in the computational basis. If $M\omega$ is an integer then* $\text{Prob}(X = M\omega) = 1$. *Otherwise, letting $\Delta = d(\omega, x/M)$,*

$$\text{Prob}(X = x) \;=\; \frac{\sin^2(M\Delta\pi)}{M^2\sin^2(\Delta\pi)} \;\leq\; \frac{1}{(2M\Delta)^2}\,.$$

For any $k > 1$ we also have

$$\text{Prob}\left(d(X/M,\omega) \leq k/M\right) \;\geq\; 1 - \frac{1}{2(k-1)}$$

and, in the case $k = 1$ and $M > 2$,

$$\text{Prob}\left(d(X/M,\omega) \leq 1/M\right) \;\geq\; \frac{8}{\pi^2}\,.$$

Proof. Clearly

$$\begin{aligned}
\text{Prob}(X = x) &= \left|\langle x|\,\mathbf{F}^{-1}|\mathcal{S}_M(\omega)\rangle\right|^2 \\
&= \left|(\mathbf{F}|x\rangle)^\dagger|\mathcal{S}_M(\omega)\rangle\right|^2 \\
&= \left|\langle\mathcal{S}_M(x/M)|\mathcal{S}_M(\omega)\rangle\right|^2
\end{aligned}$$

thus using Lemma 5.5.4 we directly obtain the first part of the theorem. We use this fact to prove the next part of the theorem.

$$\begin{aligned}
\text{Prob}\left(d(X/M,\omega) \leq k/M\right) &= 1 - \text{Prob}(d(X/M,\omega) > k/M) \\
&\geq 1 - 2\sum_{j=k}^{\infty}\frac{1}{4M^2(\frac{j}{M})^2} \\
&\geq 1 - \frac{1}{2(k-1)}.
\end{aligned}$$

For the last part, we use the fact that for $M > 2$, the given expression attains its minimum at $\Delta = 1/(2M)$ in the range $0 \leq \Delta \leq 1/M$.

$$\begin{aligned} \text{Prob}\left(d(X/M, \omega) \leq 1/M\right) &= \text{Prob}(X = \lfloor M\omega \rfloor) + \text{Prob}(X = \lceil M\omega \rceil) \\ &= \frac{\sin^2(M\Delta\pi)}{M^2 \sin^2(\Delta\pi)} + \frac{\sin^2(M(\frac{1}{M} - \Delta)\pi)}{M^2 \sin^2((\frac{1}{M} - \Delta)\pi)} \\ &\geq \frac{8}{\pi^2}. \end{aligned}$$

■

The following algorithm computes an estimate for a, via an estimate for θ_a.

Algorithm(Est_Amp$(\mathcal{A}, \chi, M)$ **)**

1. Initialize two registers of appropriate sizes to the state $|0\rangle\mathcal{A}|0\rangle$.
2. Apply $\mathbf{F}_M$ to the first register.
3. Apply $\Lambda_M(\mathbf{Q})$ where $\mathbf{Q} = -\mathcal{A}\mathbf{S}_0\mathcal{A}^{-1}\mathbf{S}_\chi$.
4. Apply $\mathbf{F}_M^{-1}$ to the first register.
5. Measure the first register and denote the outcome $|y\rangle$.
6. Output $\tilde{a} = \sin^2(\pi\frac{y}{M})$.

Steps 1 to 5 are illustrated on Figure 5.1. This algorithm can also be summarized, following the approach in [75], as the unitary transformation

$$\left((\mathbf{F}_M^{-1} \otimes \mathbf{I})\ \Lambda_M(\mathbf{Q})\ (\mathbf{F}_M \otimes \mathbf{I})\right)$$

applied on state $|0\rangle\mathcal{A}|0\rangle$, followed by a measurement of the first register and classical post-processing of the outcome.

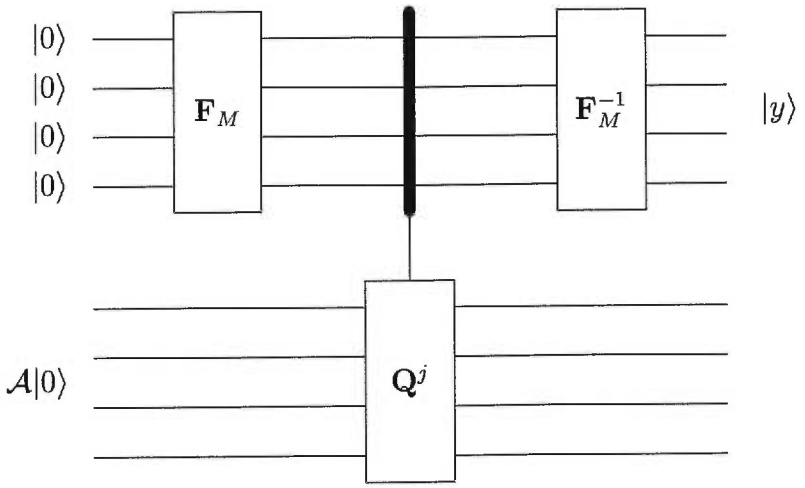

Figure 5.1: Quantum circuit for amplitude estimation.

Theorem 5.5.6 (Amplitude Estimation) *For any positive integer k, the algorithm* **Est_Amp**$(\mathcal{A}, \chi, M)$ *outputs $\tilde{a}$ $(0 \leq \tilde{a} \leq 1)$ such that*

$$|\tilde{a} - a| \leq 2\pi k \frac{\sqrt{a(1-a)}}{M} + k^2 \frac{\pi^2}{M^2}$$

with probability at least $\frac{8}{\pi^2}$ when $k = 1$ and with probability greater than $1 - \frac{1}{2(k-1)}$ for $k \geq 2$. It uses exactly M evaluations of f. If $a = 0$ then $\tilde{a} = 0$ with certainty, and if $a = 1$ and M is even, then $\tilde{a} = 1$ with certainty.

Proof. After step 1, by Equation 5.6, we have state

$$|0\rangle\mathcal{A}|0\rangle = \frac{1}{\sqrt{2}}|0\rangle \left(e^{\imath\theta_a}|\Psi_+\rangle - e^{-\imath\theta_a}|\Psi_-\rangle\right).$$

After step 2, we have

$$\frac{1}{\sqrt{2M}} \sum_{j=0}^{M-1} |j\rangle \left(e^{\imath\theta_a}|\Psi_+\rangle - e^{-\imath\theta_a}|\Psi_-\rangle\right)$$

and after applying $\Lambda_M(\mathbf{Q})$ we have

$$\begin{aligned}
&\frac{1}{\sqrt{2M}} \sum_{j=0}^{M-1} |j\rangle \left(e^{\imath\theta_a} e^{2\imath j\theta_a} |\Psi_+\rangle \; - \; e^{-\imath\theta_a} e^{-2\imath j\theta_a} |\Psi_-\rangle \right) \\
&= \frac{e^{\imath\theta_a}}{\sqrt{2M}} \sum_{j=0}^{M-1} e^{2\imath j\theta_a} |j\rangle |\Psi_+\rangle \; - \; \frac{e^{-\imath\theta_a}}{\sqrt{2M}} \sum_{j=0}^{M-1} e^{-2\imath j\theta_a} |j\rangle |\Psi_-\rangle \\
&= \frac{e^{\imath\theta_a}}{\sqrt{2}} |\mathcal{S}_M(\tfrac{\theta_a}{\pi})\rangle |\Psi_+\rangle \; - \; \frac{e^{-\imath\theta_a}}{\sqrt{2}} |\mathcal{S}_M(1 - \tfrac{\theta_a}{\pi})\rangle |\Psi_-\rangle .
\end{aligned}$$

We then apply $\mathbf{F}_M^{-1}$ to the first register and observe it in the computational basis. The rest of the proof follows from Theorem 5.5.5. Tracing out the second register in the eigenvector basis, we see that the first register is in an equally weighted mixture of $\mathbf{F}_M^{-1}|\mathcal{S}_M(\frac{\theta_a}{\pi})\rangle$ and $\mathbf{F}_M^{-1}|\mathcal{S}_M(1-\frac{\theta_a}{\pi})\rangle$. Thus the measured value $|y\rangle$ is the result of measuring either the state $\mathbf{F}_M^{-1}|\mathcal{S}_M(\frac{\theta_a}{\pi})\rangle$ or the state $\mathbf{F}_M^{-1}|\mathcal{S}_M(1-\frac{\theta_a}{\pi})\rangle$. The probability of measuring $|y\rangle$ given the state $\mathbf{F}_M^{-1}|\mathcal{S}_M(1-\frac{\theta_a}{\pi})\rangle$ is equal to the probability of measuring $|M-y\rangle$ given the state $\mathbf{F}_M^{-1}|\mathcal{S}_M(\frac{\theta_a}{\pi})\rangle$. Since $\sin^2\left(\pi\frac{(M-y)}{M}\right) = \sin^2\left(\pi\frac{y}{M}\right)$, we can assume we measured $|y\rangle$ given the state $\mathbf{F}_M^{-1}|\mathcal{S}_M(\frac{\theta_a}{\pi})\rangle$ and $\tilde{\theta}_a = \pi\frac{y}{M}$ estimates θ_a as described in Theorem 5.5.5. Thus we obtain bounds on $d(\tilde{\theta}_a, \theta_a)$ that translate, using Lemma 5.5.1, into the appropriate bounds on $|\tilde{a} - a|$. ∎

A straightforward application of this algorithm is to approximately count the number of solutions t to $f(x) = 1$. To do this we simply set $\mathcal{A} = \mathbf{W}$ if N is a power of 2, or in general $\mathcal{A} = \mathbf{F}_N$ or any other transformation that maps $|0\rangle$ to $\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle$. Setting $\chi = f$, we then have $a = \langle\Psi_1|\Psi_1\rangle = t/N$. The following algorithm comes to mind.

Algorithm(Count(f, M) **)**

1. Set $\tilde{t}' = N \times$ **Est_Amp**$(\mathbf{F}_N, f, M)$.
2. Output $\tilde{t} = \lfloor \tilde{t}' + \frac{1}{2} \rfloor$, a closest integer to $\tilde{t}'$.

By Theorem 5.5.6, we obtain the following.

Theorem 5.5.7 (Counting) *For any integer $M > 0$ and any Boolean function $f : \{0, 1, \ldots, N-1\} \to \{0, 1\}$, the algorithm* **Count**(f, M) *outputs an estimate $\tilde{t}$ to $t = |f^{-1}(1)|$ such that*

$$|\tilde{t} - t| \leq 2\pi k \frac{\sqrt{t(N-t)}}{M} + \pi^2 k^2 \frac{N}{M^2} + \frac{1}{2}$$

with probability at least $8/\pi^2$ when $k = 1$, and with probability greater than $1 - \frac{1}{2(k-1)}$ for $k \geq 2$. If $t = 0$ then $\tilde{t} = 0$ with certainty, and if $t = N$ and M is even, then $\tilde{t} = N$ with certainty.

If we want to estimate t within a few standard deviations, we can apply algorithm **Count** with $M = \lceil \sqrt{N} \rceil$.

Corollary 5.5.8 *Given a Boolean function $f : \{0, 1, \ldots, N-1\} \to \{0, 1\}$ with t defined as above,* **Count**$(f, \lceil \sqrt{N} \rceil)$ *outputs an estimate $\tilde{t}$ such that*

$$|\tilde{t} - t| < 2\pi \sqrt{\frac{t(N-t)}{N}} + 11 \tag{5.15}$$

with probability at least $8/\pi^2$ and requires exactly $\lceil \sqrt{N} \rceil$ evaluations of f.

We now look at the case of estimating t with some relative error, also referred to as *approximately counting t with accuracy ϵ*. For this we require the following crucial observation about the output $\tilde{t}$ of algorithm **Count**(f, L). Namely that $\tilde{t}$ is likely to be equal to zero if and only if $L \in o(\sqrt{N/t})$. Thus, we can find a rough estimate of $\sqrt{N/t}$ simply by running algorithm **Count**(f, L) with exponentially increasing values of L until we obtain a non-zero output. Having such a rough estimate L of $\sqrt{N/t}$ we can then apply Theorem 5.5.7 with M in the order of $\frac{1}{\epsilon}L$ to find an estimate $\tilde{t}$ of t with the required accuracy. The precise algorithm is as follows.

Algorithm(Basic_Approx_Count(f, ϵ) **)**

1. Start with $l = 0$.

2. Increase l by 1.

3. Set $t' = \mathbf{Count}(f, 2^l)$.

4. If $t' = 0$ and $2^l < 2\sqrt{N}$ then go to step 2.

5. Set $M = \lceil \frac{20\pi^2}{\epsilon} 2^l \rceil$.

6. Output $\mathbf{Count}(f, M)$.

Theorem 5.5.9 *Given a Boolean function f with N and t defined as above, and any $0 < \epsilon \leq 1$,* $\mathbf{Basic\_Approx\_Count}(f, \epsilon)$ *outputs an estimate $\tilde{t}$ such that*

$$|\tilde{t} - t| \leq \epsilon t$$

with probability at least $\frac{2}{3}$, using an expected number of evaluations of f which is in $\Theta\left(\frac{1}{\epsilon}\sqrt{N/t}\right)$. If $t = 0$, the algorithm outputs $\tilde{t} = t$ with certainty and f is evaluated a number of times in $\Theta\left(\sqrt{N}\right)$.

Proof. When $t = 0$, the analysis is straightforward. For $t > 0$, let θ denote $\theta_{t/N}$ and $m = \lfloor \log_2(\frac{1}{5\theta}) \rfloor$. From Theorem 5.5.5 we have that the probability that step 3 outputs $\mathbf{Count}(f, 2^l) = 0$ for $l = 1, 2, \ldots, m$ is

$$\prod_{l=1}^{m} \frac{\sin^2(2^l\theta)}{2^{2l}\sin^2(\theta)} \geq \prod_{l=1}^{m} \cos^2(2^l\theta) = \frac{\sin^2(2^{m+1}\theta)}{2^{2m}\sin^2(2\theta)} \geq \cos^2\left(\frac{2}{5}\right).$$

The previous inequalities are obtained by using the fact that $\sin(M\theta) \geq M\sin(\theta)\cos(M\theta)$ for any $M \geq 0$ and $0 \leq M\theta < \frac{\pi}{2}$, which can be readily seen by considering the Taylor expansion of $\tan(x)$ at $x = M\theta$.

Now assuming step 3 has outputted 0 at least m times (note that $2^m \leq \frac{1}{5\theta} \leq \frac{1}{5}\sqrt{\frac{N}{t}} < 2\sqrt{N}$), after step 5 we have $M \geq \frac{20\pi^2}{\epsilon} 2^{m+1} \geq \frac{4\pi^2}{\epsilon\theta}$ and by Theorem 5.5.7 the probability that $\mathbf{Count}(f, M)$ outputs an integer to $\tilde{t}$ satisfying $|\tilde{t} - t| \leq \frac{\epsilon}{4}t + \frac{\epsilon^2}{64}t + \frac{1}{2}$ is at least $8/\pi^2$. Let us suppose this is the case. If $\epsilon t < 1$, then $|\tilde{t} - t| < 1$ and, since $\tilde{t}$ and t are both integers, we must have $t = \tilde{t}$. If $\epsilon t \geq 1$, then $|\tilde{t} - t| \leq \frac{\epsilon}{4}t + \frac{\epsilon^2}{64}t + \frac{\epsilon}{2}t < \epsilon t$. Therefore the overall probability of outputting an estimate with error at most ϵt is at least $\cos^2\left(\frac{2}{5}\right) \times (8/\pi^2) > \frac{2}{3}$.

To upper bound the number of applications of f, note that by Theorem 5.5.7, for any integer $L \geq 18\pi\sqrt{N/t}$, the probability that **Count**(f, L) outputs 0 is less than 1/4. Thus the expected value of M at step 6 is in $\Theta(\frac{1}{\epsilon}\sqrt{N/t})$. ∎

We remark that in algorithm **Basic_Approx_Count**, we could alternatively to steps 1 to 4 use algorithm **QSearch** of Section 5.3, provided we have **QSearch** also output its final value of M. In this case, we would use (a multiple of) that value as our rough estimate of $\sqrt{N/t}$, instead of using the final value of 2^l found in step 4 of **Basic_Approx_Count**.

Algorithm **Basic_Approx_Count** is optimal for any fixed ϵ, but not in general. We give an optimal algorithm below, but first we present two simple optimal algorithms for counting the number of solutions exactly. That is, we now consider the problem of determining the exact value of $t = |f^{-1}(-1)|$. In the special case that we are given a nonzero integer t_0 and promised that either $t = 0$ or $t = t_0$, then we can determine which is the case with certainty using a number of evaluations of f in $O(\sqrt{N/t_0})$. This is an easy corollary of Theorem 5.3.4 and we state it without proof.

Theorem 5.5.10 *Let $f : \{0, 1, \ldots, N-1\} \to \{0, 1\}$ be a given Boolean function such that the cardinality of the preimage of 1 is either 0 or t_0. Then there exists a quantum algorithm that determines with certainty which is the case using a number of evaluations of f which is in $\Theta\left(\sqrt{N/t_0}\right)$, and in the latter case, also outputs a random element of $f^{-1}(1)$.*

For the general case in which we do not have any prior knowledge about t, we propose the following algorithm.

Algorithm(Exact_Count(f) **)**

1. Set $\tilde{t}_1 =$ **Count**$\left(f, \lceil 14\pi\sqrt{N}\,\rceil\right)$ and $\tilde{t}_2 =$ **Count**$\left(f, \lceil 14\pi\sqrt{N}\,\rceil\right)$.

2. Let $M_i = \left\lceil 20\sqrt{(\tilde{t}_i + 1)(N - \tilde{t}_i + 1)}\right\rceil$ for $i = 1, 2$.

3. Set $M = \min\{M_1, M_2\}$.

4. Output $\tilde{t} = \textbf{Count}(f, M)$.

The main idea of this algorithm is the same as that of algorithm **Basic_Approx_Count**. First we find a rough estimate $\tilde{t}_r$ of t, and then we run algorithm **Count**(f, M) with a value of M that depends on $\tilde{t}_r$. By Theorem 5.5.7, if we set M to be in the order of $\sqrt{\tilde{t}_r(N - \tilde{t}_r)}$, then the output $\tilde{t} = \textbf{Count}(f, M)$ is likely to satisfy that $|\tilde{t} - t| < 1$, in which case $\tilde{t} = t$.

Theorem 5.5.11 *Given a Boolean function f with N and t defined as above, algorithm* **Exact_Count** *requires an expected number of evaluations of f which is in $\Theta(\sqrt{(t+1)(N-t+1)})$ and outputs an estimate $\tilde{t}$ which equals t with probability at least $\frac{2}{3}$ using space only linear in* $\log(N)$.

Proof. Apply Theorem 5.5.7 with $k = 7$. For each $i = 1, 2$, with probability greater than 0.91, outcome $\tilde{t}_i$ satisfies that $|\tilde{t}_i - t| < \sqrt{\frac{t(N-t)}{N}} + 3/4$, in which case we also have that $\sqrt{t(N-t)} \leq \frac{\sqrt{2}}{20} M_i$. Thus, with probability greater than 0.91^2, we have

$$\frac{\sqrt{t(N-t)}}{M} \leq \frac{\sqrt{2}}{20}.$$

Suppose this is the case. Then by Theorem 5.5.7, with probability at least $8/\pi^2$,

$$|\tilde{t} - t| \leq \frac{2\pi\sqrt{2}}{20} + \frac{4\pi^2}{20^2} + \frac{1}{2} < 1.$$

Hence, with probability at least $0.91^2 \times 8/\pi^2 > 0.67$, we have $\tilde{t} = t$.

The number of applications of f is $2\lceil 14\pi\sqrt{N} \rceil + M$. Consider the expected value of M_i for $i = 1, 2$. Since

$$\sqrt{(\tilde{t}_i + 1)(N - \tilde{t}_i + 1)} \leq \sqrt{(t+1)(N-t+1)} + \sqrt{N|\tilde{t}_i - t|}$$

for any $0 \leq \tilde{t}_i, t \leq N$, we just need to upper bound the expected value of $\sqrt{N|\tilde{t}_i - t|}$. By Theorem 5.5.7, for any $k \geq 2$,

$$|\tilde{t}_i - t| > k\sqrt{\frac{t(N-t)}{N}} + \frac{k^2}{4} + \frac{1}{2}$$

with probability at most $\frac{1}{k}$. Hence M_i is greater than

$$(1+k)\left(\sqrt{(t+1)(N-t+1)} + \sqrt{N}\right) \tag{5.16}$$

with probability at most $\frac{1}{k}$.

In particular, the minimum of M_1 and M_2 is greater than the expression given by Equation 5.16 with probability at most $\frac{1}{k^2}$. Since any positive random variable Z satisfying $\text{Prob}(Z > k) \leq \frac{1}{k^2}$ has expectation upper bounded by a constant, the expected value of M is in $O\left(\sqrt{(t+1)(N-t+1)}\right)$. ∎

It follows from Theorem 4.10 of Beals *et al.* [10] that any quantum algorithm capable of deciding with high probability whether or not a function $f : \{0, 1, \ldots, N-1\} \to \{0, 1\}$ is such that $|f^{-1}(1)| \leq t$, given some $0 < t < N$, must query f a number of times which is at least in $\Omega\left(\sqrt{(t+1)(N-t+1)}\right)$ times. Therefore, our exact counting algorithm is optimal up to a constant factor.

Note also that successive applications of Grover's algorithm in which we strike out the solutions as they are found will also provide an algorithm to perform exact counting. In order to obtain a constant probability of success, if the algorithm fails to return a new element, one must do more than a constant number of trials. In particular, repeating until we get $\log(N)$ failures will provide an overall constant probability of success. Unfortunately, the number of applications of f is then in $O\left(\sqrt{tN} + \log(N)\sqrt{N/t}\right)$ and the cost in terms of additional quantum memory is prohibitive, that is in $\Theta(t)$.

We now combine the ideas of algorithms **Basic_Approx_Count** and **Exact_Count** to obtain an optimal algorithm for approximately counting. That

this algorithm is optimal follows readily from Corollary 1.2 and Theorem 1.13 of Nayak and Wu [94].

Theorem 5.5.12 *Given a Boolean function f with N and t defined as above, and any ϵ such that $\frac{1}{3N} < \epsilon \leq 1$, algorithm* **Approx_Count**(f, ϵ) *outputs an estimate $\tilde{t}$ such that*

$$\left|\tilde{t} - t\right| \ < \ \epsilon t + \frac{1}{2}$$

with probability at least $\frac{2}{3}$, using an expected number of evaluations of f in the order of

$$S = \sqrt{\frac{N}{\lfloor \epsilon t \rfloor + 1}} + \frac{\sqrt{t(N-t)}}{\lfloor \epsilon t \rfloor + 1}.$$

If $t = 0$ or $t = N$, the algorithm outputs $\tilde{t} = t$ with certainty.

We assume that $\epsilon N > 1/3$, since otherwise approximately counting with accuracy ϵ reduces to exact counting. Set

$$S' = \min\left\{ \frac{1}{\sqrt{\epsilon}} \sqrt{\frac{N}{t}} \left(1 + \sqrt{\frac{N-t}{\epsilon N}} \right), \ \sqrt{(t+1)(N-t+1)} \right\} \tag{5.17}$$

and note that $S' \in \Theta(S)$ where S is defined as in Theorem 5.5.12. The algorithm works by finding approximate values for each of the different terms in Equation 5.17. The general outline of the algorithm is as follows.

Algorithm(Approx_Count(f, ϵ) **)**

1. Find integer L_1 approximating $\sqrt{N/(t+1)}$.
2. Find integer L_2 approximating $\sqrt{(N-t)/(\epsilon N)}$.
3. Set $M_1 = \frac{1}{\sqrt{\epsilon}} L_1 (1 + L_2)$.
4. If $M_1 > \sqrt{N}$ then find integer M_2 approximating $\sqrt{(t+1)(N-t+1)}$. If $M_1 \leq \sqrt{N}$ then set $M_2 = \infty$.

5. Set $M = \min\{M_1, M_2\}$.

6. Output $\tilde{t} = \textbf{Count}(f, 14\lceil M \rceil)$.

Proof. To find L_1, we run steps 1 to 4 of algorithm **Basic_Approx_Count** and then set $L_1 = \lceil 9\pi \times 2^l \rceil$. A proof analogous to that of Theorem 5.5.9 gives that

- $L_1 > \sqrt{N/(t+1)}$ with probability at least 0.95, and
- the expected value of L_1 is in $\Theta\big(\sqrt{N/(t+1)}\big)$.

This requires a number of evaluations of f which is in $\Theta(L_1)$, and thus, the expected number of evaluations of f so far is in $O(S')$.

In step 2, for some constant c to be determined below, we use $2\lceil \frac{c}{\sqrt{\epsilon}} \rceil$ evaluations of f to find integer L_2 satisfying

- $L_2 > \sqrt{(N-t)/(\epsilon N)}$ with probability at least 0.95, and
- the expected value of L_2 is in $O\big(\sqrt{(N-t+1)/(\epsilon N)}\big)$.

Since $N - t = |f^{-1}(0)|$, finding such L_2 boils down to estimating, with accuracy in $\Theta(\sqrt{\epsilon})$, the square root of the probability that f takes the value 0 on a random point in its domain. Or equivalently, the probability that $\neg f$ takes the value 1, where $\neg f = 1 - f$. Suppose for some constant c, we run $\textbf{Count}(\neg f, \lceil \frac{c}{\sqrt{\epsilon}} \rceil)$ twice with outputs $\tilde{r}_1$ and $\tilde{r}_2$. By Theorem 5.5.7, each output $\tilde{r}_i$ $(i = 1, 2)$ satisfies that

$$\left| \sqrt{\frac{\tilde{r}_i}{\epsilon N}} - \sqrt{\frac{N-t}{\epsilon N}} \right| \leq \sqrt{\frac{2\pi k}{c}} \sqrt[4]{\frac{N-t}{\epsilon N}} + \frac{\pi k}{c} + \frac{1}{\sqrt{2\epsilon N}}$$

with probability at least $1 - \frac{1}{2(k-1)}$ for every $k \geq 2$. It follows that $\tilde{r} = \min\big\{\sqrt{\tilde{r}_1/(\epsilon N)}, \sqrt{\tilde{r}_2/(\epsilon N)}\big\}$ has expected value in $O\big(\sqrt{(N-t+1)/(\epsilon N)}\big)$. Setting $k = 21$, $c = 8\pi k$, and $L_2 = \lceil 2\tilde{r} \rceil + 4$, ensures that L_2 satisfies the two properties mentioned above. The number of evaluations of f in step 2 is in $\Theta(\frac{1}{\sqrt{\epsilon}})$ which is in $O(S')$.

In step 3, we set $M_1 = \frac{1}{\sqrt{\epsilon}} L_1(1 + L_2)$. Note that

- $M_1 > \frac{1}{\sqrt{\epsilon}} \sqrt{\frac{N}{t+1}} \left(1 + \sqrt{\frac{N-t}{\epsilon N}}\right)$ with probability at least 0.95^2, and
- the expected value of M_1 is in the order of $\frac{1}{\sqrt{\epsilon}} \sqrt{\frac{N}{t+1}} \left(1 + \sqrt{\frac{N-t+1}{\epsilon N}}\right)$.

In step 4, analogously to algorithm **Exact_Count**, a number of evaluation of f in $\Theta(\sqrt{N})$ suffices to find an integer M_2 such that

- $M_2 > \sqrt{(t+1)/(N-t+1)}$ with probability at least 0.95, and
- the expected value of M_2 is in $\Theta\left(\sqrt{(t+1)/(N-t+1)}\right)$.

Fortunately, since $\sqrt{(t+1)/(N-t+1)} \geq \sqrt{N}$, we shall only need M_2 if $M_1 > \sqrt{N}$. We obtain that, after step 5,

- M is greater than

$$\min\left\{\frac{1}{\sqrt{\epsilon}} \sqrt{\frac{N}{t+1}} \left(1 + \sqrt{\frac{N-t}{\epsilon N}}\right),\ \sqrt{(t+1)/(N-t+1)}\right\}$$

  with probability at least $0.95^3 > 0.85$, and

- the expected value of M is in $O(S')$.

To derive this latter statement, we use the fact that the expected value of the minimum of two random variables is at most the minimum of their expectation.

Finally, by Theorem 5.5.7, applying algorithm **Count**$(f, 14\lceil M \rceil)$ given such an M, produces an estimate $\tilde{t}$ of t such that $|\tilde{t} - t| \leq \epsilon t + 1/2$ with probability at least $8/\pi^2$. Hence our overall success probability is at least $0.85 \times 8/\pi^2 > 2/3$, and the expected number of evaluations of f is in $O(S')$. ∎

Acknowledgements

We are grateful to Joan Boyar, Harry Buhrman, Artur Ekert, Ashwin Nayak, Barbara Terhal and Ronald de Wolf for helpful discussions.

Chapitre 6

Complexité de la communication quantique

L'objectif de ce chapitre est de préparer le lecteur aux articles « Multiparty quantum communication complexity », « The cost of exactly simulating quantum entanglement with classical communication » et « Quantum entanglement and the communication complexity of the inner product function » (chapitres 7, 8 et 9) mais aussi d'en faire ressortir les éléments significatifs.

6.1 La fin des théories locales

tml

Une caractéristique fondamentale de la théorie quantique est sa nature probabiliste. Dans le cadre théorique de la mécanique quantique, les objets sont caractérisés par leurs fonctions d'onde et les quantités que l'on peut mesurer sont identifiées par des opérateurs hermitiens. Nous avons vu que généralement, même si la fonction d'onde (système à mesurer) et l'opérateur hermitien (mesure) sont parfaitement connus, la mécanique quantique ne permet pas de prévoir le résultat de l'expérience, elle nous permet seulement d'obtenir une distribution de probabilités sur les différents résultats possibles.

Albert Einstein est un des fondateurs de la mécanique quantique. Au début du siècle déjà, il était parfaitement conscient du pouvoir explicatif de cette théorie qui *colle* merveilleusement avec toutes les expériences effectuées par les physiciens. Cependant, il ne pouvait accepter l'idée que cette théorie fut complète. Cette fonction d'onde qui permettait des prédictions statistiques devait être régie par d'autres variables cachées qui expliqueraient ce comportement aléatoire face aux mesures. Un peu comme le mouvement brownien des grains de pollen qui paraît aléatoire jusqu'à ce que l'on considère le mouvement individuel des atomes d'air qui s'y cognent. « God doesn't play dice » clama-t-il durant son long débat à ce sujet avec le physicien Niels Bohr à la fin des années 1920. À cela Bohr répondit « Stop telling God what to do ». En effet, Bohr croyait que les quantités physiques mesurées prenaient existence lors de la mesure elle-même, interprétation connue sous le nom « Copenhague ». Pour sa part, Einstein croyait que la mécanique quantique était incomplète et donc que les quantités physiques existaient préalablement à l'observation.

Plus inquiétante encore que sa facette probabiliste, la mécanique quantique décrit l'existence de quantités qui sont dites complémentaires. Chacun connaît le principe d'incertitude d'Heisenberg qui grosso modo affirme qu'il est impossible de connaître à la fois avec une précision arbitraire la position et la quantité de mouvement d'une particule. Ce principe s'énonce en mécanique quantique par le fait que ces deux quantités, position et quantité de mouvement, sont complémentaires (c.-à-d. que leurs opérateurs ne commutent pas). Ce principe ne contredit aucune expérience puisque toute tentative de connaître à la fois la position et la quantité de mouvement d'un électron, par exemple, s'est soldée par un échec. Plus la connaissance de la position est précise, plus notre erreur au sujet de la quantité de mouvement est grande. La mécanique quantique affirme d'une certaine façon que, dans le cas de quantités complémentaires, notre limitation n'est pas due à nos appareils, mais plutôt au fait que ces deux quantités n'aient pas de sens simultanément.

En 1935, Albert Einstein, Boris Podolsky et Nathan Rosen ont proposé une expérience de pensée (comme l'expérience du démon de Maxwell), visant à démontrer l'incomplétude de la mécanique quantique. Cette expérience parut dans un article aujourd'hui célèbre sous le nom EPR [61]. Imaginons deux particules créées en un point central et voyageant en sens inverse. Elles ont la même quantité de mouvement et, à tout moment, sont à la même distance de leur origine. Bien que le principe d'incertitude s'applique sur chaque particule individuellement, on peut apprendre la position de la première particule en mesurant la position de la deuxième et ce sans du tout perturber la quantité de mouvement de la première. Il en va de même pour la quantité de mouvement. La seule supposition faite par Einstein, qui somme toute semble raisonnable, est qu'une mesure faite sur la première particule ne doit pas influencer la deuxième particule. Einstein croyait avoir démontré ainsi que la position et la quantité de mouvement, bien que ne pouvant pas être mesurées simultanément, ont néanmoins une réalité propre indépendante et locale.

L'argumentation des trois physiciens ne réussit pas à convaincre Bohr et cette expérience de pensée ne suscita pas un vif intérêt dans la communauté scientifique. La question restait ouverte.

L'importance fondamentale de l'expérience EPR deviendra claire en 1964, année où John S. Bell [11] démontra sa fameuse inégalité. Le résultat de Bell était basé sur une version simplifiée de l'expérience EPR proposée en 1951 par David Bohm [23]. Dans cet article, Bell démontre que la corrélation observée entre différents résultats de mesure, effectuée sur des paires de particules ayant interagi mais étant maintenant physiquement séparées, ne pouvait pas s'expliquer par une théorie où les quantités physiques sont locales. C'est-à-dire qu'on peut imaginer des expériences dans lesquelles le choix d'une mesure effectuée sur un système B semble influencer le résultat de mesure effectué sur un système A. Il est important de noter que cette interaction ne peut se faire par l'entremise de transmission de signal puisqu'elle est théoriquement perceptible instantanément sans aucun

égard à la distance. Bien qu'Einstein soit mort en 1955, plusieurs années avant la parution de l'article de Bell, on lui doit le terme *spukhafte Fernwirkungen* normalement traduit en anglais par « spooky action at a distance » ou si vous préférez, action surnaturelle à distance, pour désigner cette étrange interaction entre des particules éloignées. Aujourd'hui, on qualifie d'intriquée (*entangled*) une paire de systèmes dont l'état quantique n'est pas local (ne peut pas être factorisé).

Plusieurs expériences ont été effectuées dans le but de vérifier le caractère non-local de la mécanique quantique. En 1982 A. Aspect, P. Grangier et G. Rogeret [3] furent les premiers à effectuer une expérience où les deux systèmes, mesurés quasi simultanément, se trouvaient à une distance suffisamment grande pour qu'il soit possible d'exclure l'échange de signaux classique entre eux. Les corrélations prévues par la mécanique quantique et inaccessibles classiquement ont bien entendu été observées.

Plutôt que d'expliquer l'expérience physique ainsi que le raisonnement utilisé par Bell pour démontrer que la mécanique quantique n'est pas une théorie locale, nous avons choisi de présenter une version modifiée de la preuve de N. David Mermin[92] qui elle-même est une version simplifiée de l'argument utilisé par Daniel Greenberger, Michael Horne et Anton Zeilenger [67]. Cette version non probabiliste du théorème de Bell sera d'ailleurs la base d'une série de résultats en complexité de la communication.

6.2 GHZ

Imaginez une foule de petits appareils semblables venant en triplets où chaque appareil est identifié par une lettre (A,B,C) et un numéro qui permet de distinguer les triplets. Chaque appareil est coiffé d'une ampoule et possède un interrupteur se trouvant dans une position centrale. Lorsque l'on prend un de ces petits appareils et que l'on pousse l'interrupteur vers la gauche en position **G** ou vers la droite en

position **D**, il arrive que l'ampoule s'allume, auquel cas elle le fait *immédiatement.* Une fois déplacé, l'interrupteur gardera définitivement sa position.

Étudions ces appareils. La première constatation que l'on fait est statistique : lorsque l'on place l'appareil en position **G** ou **D**, il semble y avoir une chance sur deux pour que l'ampoule s'allume. Si l'on observe ces appareils de façon plus attentive, on remarque que les triplets ayant le même numéro affichent une certaine corrélation dans leurs réactions.

- **Règle 1 :** Si les trois interrupteurs sont placés en position **G**, alors un nombre impair d'ampoules est allumé. C'est-à-dire que toutes les trois, ou une seule, des ampoules sont allumées.
- **Règle 2 :** Si un seul interrupteur est positionné en **G** et les deux autres sont positionnés en **D**, alors il y a nécessairement un nombre pair d'ampoules allumées.

Clairement, avec ces règles, une fois que deux des appareils sont en position, le nombre d'ampoules allumées nous permet souvent de prévoir le comportement du troisième appareil. Par exemple, si l'appareil B est en position **G** avec son ampoule éteinte et l'appareil C est en position **D** avec son ampoule allumée on sait, par la règle 2, qu'en plaçant l'appareil A en position **D** son ampoule s'allumera.

Interrogeons-nous sur la technique utilisée pour construire ces appareils. Une première approche consisterait à programmer les appareils lors de leur construction afin qu'ils réagissent de façon préétablie à chacune des deux positions de l'interrupteur. Par exemple, un triplet d'appareil peut être construit de la façon suivante :

	G	D
A	1	1
B	1	0
C	1	1

Un 1 à la ligne B dans la colonne G signifie que si l'on choisit pour l'appareil B la position **G**, son ampoule s'allumera (un zéro aurait signifié que son ampoule ne

s'allumera pas). Dans l'exemple ci-dessus, il est clair que le triplet d'appareils ne contredit pas la règle 1 puisque si les trois appareils sont en position **G** les trois ampoules s'allumeront. Par contre, dans le cas où l'appareil A est en position **D**, le B en position **G** et le C en position **D**, un nombre impair de lumières s'allumera, ce qui contredit la règle 2.

Il existe $2^{3+3} = 64$ configurations possibles de triplets d'appareils. Or, *aucune* d'entre elles ne respecte les règles 1 et 2 simultanément pour chaque positionnement possible des interrupteurs. De plus, il est clair que de rendre probabiliste la réaction des appareils ne réglera rien. Pour obtenir de telles corrélations, il *semble* que les appareils doivent communiquer entre eux ; la réaction du dernier appareil dépend de la position de son interrupteur mais aussi du positionnement des deux autres appareils. Donc, si l'on interdit aux appareils de communiquer entre eux, soit en les plaçant dans une cage de Faraday ou simplement en les éloignant suffisamment, le comportement des appareils *devra* être déviant.

Le fait est que de tels appareils, respectant systématiquement les deux règles sans égard à la distance, peuvent en principe être construits. Cela constitue en fait l'exemple parfait de ce qu'Einstein appelait action surnaturelle à distance. Si l'univers était classique, cela serait impossible pour les raisons que nous avons évoquées plus haut. L'univers est quantique et, bien que même dans le cadre de la mécanique quantique il soit toujours impossible de communiquer sans envoyer de signal ou plus vite que la lumière, la mécanique quantique permet un type de corrélation entre des systèmes distants, corrélation *surnaturelle* qui permet, justement, la construction desdits appareils.

Voici comment, avec un système de trois qubits, il est possible de construire des appareils quantiques qui auront exactement le comportement indiqué plus haut. On construit premièrement l'état suivant :

$$|GHZ'\rangle = \frac{1}{2}(|001\rangle + |010\rangle + |100\rangle - |111\rangle) \tag{6.1}$$

puis on place un des trois qubit de $|GHZ'\rangle$ dans chaque appareil. Lorsque l'interrupteur est placé à gauche, le qubit est mesuré tel quel, par contre, si l'interrupteur est placé à droite, l'appareil effectue W (la transformation de Walsh-Hadamard) sur le qubit avant de le mesurer. Si le résultat est $|1\rangle$ alors l'ampoule de l'interrupteur sera allumée et sinon ($|0\rangle$) elle ne le sera pas. Clairement, si tous les interrupteurs sont en position **G**, un nombre impair d'ampoules s'allumera puisque les trois qubits sont dans leurs états originaux. Par contre si deux des interrupteurs sont en position **D** et le troisième en position **G**, ce qui revient à dire qu'on applique W sur deux des trois qubits, alors avant la mesure, le registre se retrouve nécessairement dans un état de superposition de toutes les combinaisons de vecteurs de trois bits ayant nécessairement un nombre pair de 1.

$$\begin{aligned}
(I \otimes W \otimes W)|GHZ'\rangle &= \frac{1}{2}(|000\rangle - |011\rangle + |101\rangle + |110\rangle) \\
(W \otimes I \otimes W)|GHZ'\rangle &= \frac{1}{2}(|000\rangle + |011\rangle - |101\rangle + |110\rangle) \\
(W \otimes W \otimes I)|GHZ'\rangle &= \frac{1}{2}(|000\rangle + |011\rangle + |101\rangle - |110\rangle)
\end{aligned}$$

Clairement, dans ce cas, un nombre pair de lumières s'allumera.

L'impact de cette expérience de pensée est immense. Le fait qu'il soit possible en principe de construire de tels appareils montre qu'il est impossible d'expliquer l'univers de façon locale, la mécanique quantique n'est pas une théorie locale.

De façon plus formelle, des systèmes quantiques ayant la capacité de démontrer des corrélations ne pouvant s'expliquer par une théorie locale sont appelés intriqués. On dira de deux ou plusieurs systèmes quantiques qu'ils sont intriqués s'ils ne peuvent s'écrire comme un produit. Par exemple, le premier qubit et le deuxième qubit de l'état

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

ne sont pas intriqués puisque cet état peut s'écrire de la façon suivante

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Un des états intriqués les plus connus est certainement l'état

$$|\psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

état introduit par Bohm dans sa version simplifiée de l'expérience EPR. C'est aussi l'état utilisé par Bell pour dériver la première preuve que la mécanique quantique n'est pas locale. L'état $|\psi^-\rangle$ est aussi utilisé dans la téléportation quantique [17] et dans le codage dense [19], deux techniques permettant la transmission d'information classique et quantique au sujet desquelles nous reviendrons.

Bien sûr, l'état $|GHZ'\rangle$ (équation 6.1) est intriqué. En fait cet état ainsi que l'état $|\psi^-\rangle$ sont deux exemples, à variation locale près, d'un état appelé *cat state* ou si vous préférez l'état de Schrödinger :

$$\frac{1}{\sqrt{2}}(|00\ldots0\rangle - |11\ldots1\rangle). \tag{6.2}$$

On voit facilement qu'en appliquant localement sur chaque qubit une opération unitaire appropriée on obtient le résultat désiré.

$$\begin{aligned}(I \otimes U_1)\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) &= |\psi^-\rangle \\ (U_2 \otimes U_2 \otimes U_2)\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle) &= |GHZ'\rangle\end{aligned}$$

où $U_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ et $U_2 = \frac{-i}{\sqrt{2}}\begin{pmatrix} i & i \\ 1 & -1 \end{pmatrix}$.

Cette expérience de pensée fut le coup d'envoi de la complexité de communication quantique.

6.3 Complexité de communication

Il existe plusieurs mesures de complexité caractérisant le calcul d'une fonction. On peut s'intéresser au temps, à l'espace nécessaire ou à la profondeur d'un circuit la calculant. En 1979, Andy Yao [111] introduisit une nouvelle mesure de complexité appelée complexité de communication. Soit un ensemble de k participants et une fonction booléenne F définie sur k variables représentées par des chaînes de n bits. Les participants communiquent entre eux à l'aide d'un canal de diffusion en suivant un protocole où le comportement de chacun ne dépend que de son entrée et de ce que les autres participants ont déjà diffusé. On note $C(F)$ le nombre minimal de bits communiqués par les participants pour que chacun apprenne la valeur de F et ce pour le pire k-tuplet d'entrées.[1]

Ce modèle a déjà reçu beaucoup d'attention et il en existe une multitude de variantes. Le lecteur est invité à consulter [83] qui traite du sujet en profondeur. Une des variantes intéressantes consiste à mettre à la disposition des participants une source commune de bits aléatoires. On exige alors qu'ils calculent la bonne valeur seulement avec une certaine probabilité. On notera $C_\alpha(F)$ le nombre minimal de bits communiqués pour que tous les participants apprennent $F(x_1, \ldots, x_k)$ avec probabilité supérieure ou égale à α, la probabilité étant prise sur l'ensemble des chaînes aléatoires possibles. Notez que l'on exige que tous les participants obtiennent la même valeur. Clairement

$$\forall F, C_1(F) = C(F) \quad \text{et} \quad \alpha \leq \alpha' \Rightarrow C_\alpha(F) \leq C_{\alpha'}(F).$$

En 1993, Andy Yao [112] introduisit l'informatique quantique dans le monde de la complexité de la communication. Il décrivit un modèle de complexité de communication à deux participants où la communication classique est remplacée par

[1] Il n'y a pas consensus sur le fait spécifique exigeant que tous les participants apprennent la valeur de la fonction. Parfois on exigera seulement qu'un participant prédésigné en apprenne la valeur. Comme nous nous intéressons uniquement à des fonctions booléennes, il est facile de se convaincre que la différence entre les deux mesures de complexité est toujours exactement un bit.

l'échange de qubits. Entre les rondes de communication, les participants peuvent effectuer n'importe quelle transformation unitaire sur leurs registres. On notera $Q(F)$ le nombre minimal de qubits que les participants doivent s'échanger pour que chacun apprenne la valeur de F avec probabilité 1. Clairement, ce modèle est aussi puissant que le modèle standard, c'est-à-dire que

$$Q(F) \leq C(F)$$

mais au moment de la parution de l'article, aucun exemple n'existait pour lequel cette inégalité était stricte. En fait, le modèle avait été introduit par A. Yao dans un autre but.

Les premiers résultats intéressants, montrant que l'utilisation de ressources quantiques peut diminuer la quantité de communication nécessaire à l'évaluation d'une fonction, virent le jour dans un modèle différent. Au lieu de permettre aux participants de communiquer des bits quantiques, on leur permettra de partager de l'intrication. Ceci est paradoxal puisqu'il est connu que l'intrication ne peut pas être utilisée pour se substituer à la communication, pas plus qu'elle peut être utilisée pour comprimer des messages [39, 50].

Il semble donc que l'intrication ne sera d'aucune utilité pour diminuer la quantité de communication nécessaire à l'évaluation d'une fonction. Cette corrélation quantique partagée entre les participants s'apparente aux modèles avec source aléatoire. Rappelons que l'ajout d'une source aléatoire commune ne rend pas le modèle déterministe plus puissant puisque

$$\forall F, C_1(F) = C(F).$$

Ici le lecteur devrait avoir en tête l'expérience décrite dans la section précédente qui nous avait permis de conclure que la mécanique quantique n'est pas une théorie locale. Bien que l'intrication ne permette pas la transmission d'information, elle permet quand même la réalisation de tâches qui classiquement en nécessitent.

C'est cette capacité de l'intrication qui permet à des systèmes quantiques de réagir de façon corrélée à des mesures spécifiques qui a été utilisée par Cleve et Buhrman[48] puis par Buhrman, Cleve et van Dam [39], pour obtenir les premiers résultats significatifs du domaine. On définira $C^*(F)$ le nombre minimal de bits que les k participants doivent s'échanger (canal de diffusion) pour que tous les participants apprennent F dans un contexte ou l'on permet aux participants une réserve arbitraire d'intrication.[2] Notez que l'intrication doit évidemment être *indépendante* des x_i et donc doit avoir été échangée *avant* que les participants reçoivent leurs entrées. Cet échange d'information quantique n'est pas comptabilisé, pas plus que les bits de la source aléatoire ne le sont dans le modèle probabiliste.

Dans l'article [48] Cleve et Buhrman proposent la première fonction F pour laquelle

$$C(F) = C^*(F) + 1.$$

Dans leur exemple $k = 3$ et le domaine de la fonction est restreint. Ici, nous devons faire une autre classification des problèmes de communication. Nous les diviserons en deux classes : les fonctions qui sont définies sur tous les éléments du domaine et les relations ou fonctions à domaine restreint. Dans plusieurs des exemples où il existe un gain entre le modèle quantique et le modèle classique, certains tuplets d'entrées sont proscrits. Si cette restriction du domaine ne nous plaît pas, on peut toujours transformer une fonction à domaine restreint en une relation de la façon suivante. Soit F définie sur D, alors la relation R est telle que $\forall x \in D, (x, F(x)) \in R$ et $\forall x \notin D, (x, 0)$ et $(x, 1) \in R$. Le problème à résoudre consistera dans ce cas à retourner en sortie une valeur telle que la paire entrée/sortie soit dans la relation.

Peu de temps après, Buhrman, Cleve et van Dam [39] proposent un autre exemple avec les mêmes caractéristiques. C'est dans cet article que l'on retrouve la défini-

[2] La définition que nous utilisons pour C^* provient de [39]

tion de C^*. Il est intéressant de noter que ces deux premiers résultats s'appuient directement sur l'expérience de Mermin que nous avons présentée dans la section précédente. Buhrman, Cleve et van Dam [39] proposent aussi une fonction g pour laquelle $C^*_{0.8}(g) = 2$ mais pour laquelle $C_{0.8}(g) \geq 3$. L'intérêt de cet exemple réside dans le fait qu'aucune restriction sur le domaine de la fonction n'est nécessaire pour obtenir ce gain. La mesure $C^*_{0.8}$ est l'extension probabiliste naturelle de la mesure C^*.

Bien que forts intéressants, ces gains d'un bit peuvent-t-ils être généralisés ? Est-il possible d'obtenir une séparation asymptotique ? C'est par l'affirmative que nous avons répondu à cette question importante dans l'article « Multiparty quantum communication complexity » [41, chapitre 7 ici]. Cet article contient deux résultats. Le premier donne une séparation asymptotique dans un modèle un peu différent où les communications sont réduites à une ronde. Dans ce modèle, nous exhibons une fonction G de trois variables (participants) pour laquelle $C^*(G) = n+1$ mais $C(G) = \frac{3}{2}n + 1$. Nous exhibons aussi une fonction F de k variables (participants) pour laquelle il est possible de démontrer que $C^*(F) = k + 1$ mais $C(F) \simeq k \log(k/2)$ et ce sans faire de restriction sur le nombre de rondes.

Avant d'aller plus loin, il sera utile de faire une remarque concernant le lien entre Q et C^*. Un des résultats les plus importants de l'informatique quantique est certainement la téléportation quantique [17]. Il a en effet été démontré par Bennett, Brassard, Crépeau, Jozsa, Peres et Wootters que si deux participants ont en commun l'état intriqué $|\psi^-\rangle$, et la capacité de transmettre de l'information classique, ils peuvent alors simuler un canal quantique. Chaque qubit transmis coûtera exactement une paire $|\psi^-\rangle$ et deux bits de communication classique. À partir de ce résultat, on déduit immédiatement pour le cas à deux participants ($k = 2$).

$$\forall F, C^*(F) \leq 2Q(F).$$

En 1998 Buhrman, Cleve et Wigderson [40] obtinrent des séparations plus importantes. En joignant le modèle de calcul quantique (boîte noire) à la complexité de communication de type qubits, ils sont parvenus à obtenir la première séparation exponentielle dans le modèle déterministe avec promesse ainsi qu'une séparation quadratique dans le modèle probabiliste sans promesse. Ces résultats leur ont aussi permis d'obtenir des bornes inférieures dans le modèle *boîte noire* de l'algorithme quantique. Plus spécifiquement, pour la fonction

$$DISJ(x,y) = \bigvee_{i\in\{1,\ldots,N\}} x_i y_i$$

ils ont démontré que

$$Q_{2/3}(DISJ) \in O(\sqrt{N}\log N)$$

mais il était déjà connu que

$$C_{2/3}(DISJ) \in \Omega(N)$$

ce qui donne une séparation quadratique et ce sans restriction au domaine de la fonction. Soit $D(x,y)$ la distance de Hamming entre x et y (i.e. le nombre de positions où les chaines x et y diffèrent). Pour la fonction

$$EQ'(x,y) = 1 \text{ si } (x=y) \text{ et } 0 \text{ si } D(x,y) = N/2,$$

considérée seulement sur le domaine où la fonction est définie, ils ont obtenu le résultat suivant

$$Q(EQ') \in \log(N).$$

Ils ont aussi démontré la borne inférieure suivante

$$C(EQ') \in \Omega(N)$$

cette borne inférieure nous sera d'ailleurs fort utile dans le chapitre 8. Malheureusement, cette séparation exponentielle ne tient plus dans le cas probabiliste puisque de façon évidente

$$Q_{2/3}(EQ') \text{ et } C_{2/3}(EQ') \in \Theta(1).$$

Ce n'est que très récemment qu'une séparation exponentielle entre ces deux mesures de complexité fut obtenue par Ran Raz [97] dans le cas d'une fonction à domaine restreint (avec promesse). En effet Raz définit une fonction $\mathcal{P}_1$ tel que

$$Q_{2/3}(\mathcal{P}_1) \in O(\log(n))$$

où n est la taille des entrées. Plus important, à l'aide de techniques sophistiquées, il démontre que

$$C_{2/3}(\mathcal{P}_1) \in \Omega(\sqrt{n}).$$

Dans certains des exemples vus précédemment, l'intrication a été utilisée pour diminuer le besoin de communication dans la réalisation de tâches d'évaluation de fonction sur données distribuées. Dans « The cost of exactly simulating quantum entanglement with classical communication » [30, chapitre 8 ici], nous avons poussé cette idée à sa limite. Nous présentons un problème, une tâche, que deux participants partageant n bits d'intrications peuvent accomplir *sans* communication ! Contraste intéressant, dans un monde classique, cette tâche nécessite $2^{\Omega(n)}$ bits de communication ! Ce résultat est très intéressant au point de vue de la théorie de l'information quantique puisqu'il donne une borne à la quantité d'information classique dont la communication est nécessaire pour simuler

un ensemble de n bits intriqués. C'est-à-dire que nous obtenons une borne à la quantité de communication qui doit être ajoutée au modèle classique (local) à variables cachées pour obtenir les mêmes résultats que la mécanique quantique prévoit lors d'expériences sur des systèmes bipartites. En effet, si l'on désire simuler parfaitement n paires intriquées, il faudra au moins pouvoir accomplir la tâche décrite précédemment sans erreur. Puisque cette tâche nécessite $2^{\Omega(n)}$ bits de communication, on obtient d'emblée une borne inférieure exponentielle.

Dans la même publication (chapitre 8), nous nous sommes posé la question inverse, c'est-à-dire combien de bits de communication sont suffisants pour parfaitement simuler le comportement de particules intriquées. Nous nous sommes concentrés sur le cas d'une paire maximalement intriquée

$$|\psi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Chaque participant a en sa possession une des deux particules puis reçoit une mesure complète[3] à effectuer. Il doit retourner le résultat de sa mesure qui consiste en un simple bit classique. La question qui nous intéresse ici est de savoir combien de bits classiques les participants doivent s'échanger pour obtenir des résultats corrélés *exactement* comme la mécanique quantique le prescrit s'ils *ne possèdent pas* de particule intriquée. Dans le chapitre 8, cette mesure est décrite par l'opérateur hermitien général pour une mesure sur un qubit. Cette description d'une mesure est strictement équivalente à l'application d'une opération unitaire arbitraire sur un seul qubit suivi d'une mesure dans la base standard. Pour toute mesure M_1 d'un participant, il existe une mesure[4] M_2 pour l'autre participant tel que les deux participants obtiennent le même résultat avec probabilité 1. Par contre, si la mesure du deuxième participant est plutôt M_2', une mesure dans une base très près de M_2, alors, avec une probabilité non nulle, ils obtiendront des résultats opposés. On est donc porté à croire qu'une quantité de communication arbitraire,

[3] de von Neumann

[4] On peut montrer qu'il en existe exactement deux.

dépendant de la précision avec laquelle les mesures sont spécifiées, est nécessaire si l'on n'est pas prêt à accepter une déviation, même minime, aux statistiques prévues par la mécanique quantique. Cette intuition est trompeuse puisque nous avons réussi à démontrer que, même si aucune restriction n'est supposée quant à la précision des mesures, les corrélations prescrites par la mécanique quantique sont obtenues exactement, sans que les participants n'aient d'intrication, si on leur permet d'échanger 8 bits classiques.

Nous avons obtenu plusieurs résultats intéressants montrant que l'intrication permet parfois de réduire la communication. Est-il toujours possible de réduire la communication en utilisant de l'intrication ? Existe-t-il une technique générale s'appliquant à tous les problèmes de communication nous assurant un gain systématique ? Nous pouvons tout de suite affirmer que la réponse à cette question est non. Dans l'article « Quantum entanglement and the communication complexity of the inner Product function » [50, chapitre 9 ici], nous avons étudié une fonction spécifique et nous avons montré que pour cette fonction, aucun gain ne pouvait être obtenu.

En particulier, pour la fonction

$$IP(x,y) = x \cdot y = x_1y_1 + x_2y_2 + \cdots x_ny_n \pmod 2$$

les résultats classiques suivants sont déjà connus [5]

$$\begin{aligned} C(IP) &= n+1 \\ C_{1/2+\epsilon} &= n - O(\log(1/\epsilon)) \end{aligned}$$

et Kremer[6] [82] avait démontré

$$Q_{1/2+\epsilon}(IP) \in \Omega(n)$$

[5] Dans le chapitre 9, toutes ces mesures sont inférieures d'un bit puisqu'on exige seulement qu'un participant apprenne la valeur de la fonction.

[6] Méthodologie attribuée à A. Yao

Nous avons obtenu les résultats suivants :

$$\begin{aligned} C^*(IP) &= n+1 \\ Q^*(IP) &= n/2+1 \quad (n \text{ pair}) \\ C^*_{(1/2+\epsilon)}(IP) &\geq \text{Max}(2\epsilon^2 n, 16\epsilon^4 n) \\ Q^*_{(1/2+\epsilon)}(IP) &\geq 2\epsilon^2 n \end{aligned}$$

Outre le fait que pour cette fonction l'intrication n'est d'aucune utilité (sauf peut-être dans le cas Q^* où le codage dense peut être utilisé), la méthodologie de la preuve utilisée est une réduction quantique qui n'a pas son équivalent classique. Nous avons en fait démontré qu'un protocole permettant le calcul de la fonction IP peut nécessairement être utilisé pour transmettre n bits d'information. Comme la transmission d'information au sens de Shannon est régie par des règles strictes, on obtient alors facilement les bornes inférieures citées plus haut.

Dans le même chapitre, nous démontrons aussi un curieux résultat dans le cas $n = 2$ où 1 bit peut être épargné puisque

$$\begin{aligned} C_{4/5}(IP(x_1x_2, y_1y_2)) &= 3 \quad \text{et} \\ C^*_{4/5}(IP(x_1x_2, y_1y_2)) &= 2. \end{aligned}$$

Chapter 7

Multiparty quantum communication complexity

This chapter reproduces an article to appear in the journal *Physical Review A* [41]. It has been written with the collaboration of Harry Buhrman, Wim van Dam and Peter Høyer.

7.1 Abstract

Quantum entanglement cannot be used to achieve direct communication between remote parties, but it can reduce the communication needed for some problems. Let each of k parties hold some partial input data to some fixed k-variable function f. The communication complexity of f is the minimum number of classical bits required to be broadcasted for every party to know the value of f on their inputs.

We construct a function G such that for the one-round communication model and three parties, G can be computed with $n+1$ bits of communication when the parties share prior entanglement. We then show that without entangled particles, the one-round communication complexity of G is $(3/2)n+1$. Next we generalize this function to

a function F. We show that if the parties share prior quantum entanglement, then the communication complexity of F is exactly k. We also show that if no entangled particles are provided, then the communication complexity of F is roughly $k \log_2 k$.

These two results prove for the first time communication complexity separations better than a constant number of bits.

7.2 Introduction

Suppose each of k parties holds some data that is unknown to the others, and they want to evaluate some fixed k-variable function on those data. If the function is non-trivial, then this cannot be done unless the parties communicate.

In [48], Cleve and Buhrman raised the question whether or not less communication is needed if the parties possess entangled particles. They demonstrated that, for a specific problem, prior quantum entanglement decreases the need for communication by 1 bit from 3 to 2 bits. A 1-bit saving was also obtained by Buhrman, Cleve, and van Dam in [39] for another problem where each party initially holds a 2-bit input-string. In both of these problems, there are 3 parties ($k = 3$). They left open the important question if a separation larger than 1 bit is possible. In particular, is a separation in an asymptotic setting possible? In this article we show that this is indeed the case.

Let f be a k-variable Boolean function whose inputs are n-bit binary strings (that is, $f : X^k \to \{0,1\}$ where $X = \{0,1\}^n$). There are k parties, denoted $P_1, \ldots, P_k$, where party P_i holds input data x_i $(i = 1, \ldots, k)$. Initially, party P_i only knows x_i, so, to evaluate f, the parties have to communicate among each other. The communication is done by broadcasting classical bits, where, each time, a party broadcasts one bit to everybody, on the total cost of one bit of communication.

We are interested in determining the minimum number of bits required to be broadcasted in the worst-case for every party to know the value of f. This

number is called the *communication complexity of* f and is denoted $C(f,k,n)$. We want to compare this number with $Q(f,k,n)$, the communication complexity of f *with* prior quantum entanglement. That is, the situation where we allow the parties to share a set of entangled particles before they learn their inputs [48, 39].

For example, with this terminology, the separation obtained in [39] reads: there exists a 3-variable ($k = 3$) Boolean function g whose inputs are 2-bit strings ($n = 2$), and for which $C(g,3,2) = 3$, but $Q(g,3,2) = 2$. For some functions, no separation at all is possible. For example, Cleve *et al.* [50] showed that prior quantum entanglement does not help in computing the so-called inner product function.

References [48, 39] left open the very interesting question if a separation in an asymptotic setting is possible. This question can be phrased more formal as: Does there exists a function f for which $C(f,k,n)$ grows in k or n, and for which the ratio between $C(f,k,n)$ and $Q(f,k,n)$ is bounded from below by some constant larger than 1?

In this paper, we first study the case where the number of parties is three ($k = 3$). In this setting we consider the *one-round* communication model where each party is allowed to communicate at most once. We construct a Boolean function G for which $C(G,3,n) = (3/2)n+1$ whereas $Q(G,3,n) = n+1$. This gives a separation by a factor of 3/2 in terms of the number of bits hold by each of the three parties.

Next we relax the requirement that only one round of communication is allowed and consider an arbitrary number of parties. To this end we generalize the communication function G to F. We demonstrate that the communication complexity of F with prior quantum entanglement is exactly k [that is, $Q(F,k,n) = k$], but that, if $n \geq \log_2 k$, then without quantum entanglement it is roughly $k \log_2 k$ [that is, $C(F,k,n) \approx k \log_2 k$]. We prove this by giving upper and lower bounds in both cases. This implies a separation by a logarithmic factor in k, the number of parties.

This paper thus presents a function with a separation by a constant factor in terms of the number of bits, and a function with a separation by a logarithmic factor in terms of the number of parties. Very recently, much more impressive separations have been obtained in terms of the number of bits. Buhrman, Cleve, and Wigderson [40], Ambainis *et al.* [1], and Raz [97] have all found two-party computational problems for which an exponential separation holds.

7.3 The modulo-4 sum problem

In this section, we fix the number of parties to three ($k = 3$). As common, we name the parties Alice, Bob, and Carol.

In [39], Buhrman, Cleve, and van Dam considered the *Modulo-4 Sum Problem* defined as follows. Alice, Bob, and Carol receive x, y, and z, respectively, where $x, y, z \in U = \{0, 1, 2, 3\}$, and they are promised that

$$(x + y + z) \bmod 2 = 0. \tag{7.1}$$

The common goal is for every party to learn the value of the function

$$f(x, y, z) = \frac{1}{2}\Big[(x + y + z) \bmod 4\Big]. \tag{7.2}$$

We say that $(x, y, z) \in U \times U \times U$ is a *valid* input if Eq. 7.1 holds. The function $f : U \times U \times U \to \{0, 1\}$ can be viewed as computing the second-least significant bit in the sum of x, y, and z. Note that for all inputs $y, z \in U$ to Bob and Carol, there exists a unique input $x \in U$ for Alice such that (x, y, z) is a valid input and $f(x, y, z) = 1$.

For every integer $m \geq 1$, we generalize f to $G_m : U^m \times U^m \times U^m \to \{0, 1\}$ by setting

$$G_m(\mathbf{x}, \mathbf{y}, \mathbf{z}) = 1 \quad \text{if and only if} \quad \text{for all } 1 \leq i \leq m \text{ we have } f(x_i, y_i, z_i) = 1,$$

where $\mathbf{x} = (x_1, \ldots, x_m)$, $\mathbf{y} = (y_1, \ldots, y_m)$, and $\mathbf{z} = (z_1, \ldots, z_m)$, and with the condition that,

$$(x_i + y_i + z_i) \bmod 2 = 0 \qquad (1 \leq i \leq m). \tag{7.3}$$

Thus, we give Alice, Bob, and Carol m valid instances of f, all at the same time, and ask if they all evaluate to 1. Again, we say that $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ is a *valid* input if Eq. 7.3 holds.

Buhrman *et al.* [39] showed that *with* prior entanglement, function f can be solved with one-round communication using 3 bits. In their protocol, Bob and Carol each broadcast one bit, where after Alice is capable of computing the value of f and then broadcasting the resulting bit. (See Section 7.4.1 for a direct generalization of their protocol.) Their protocol therefore immediately yields a $2m + 1$ bits protocol for G_m.

Theorem 7.3.1 *With prior quantum entanglement G_m can be solved with one-round communication using $2m + 1$ bits.*

In Subsection 7.3.2 below, we prove the following lower bound for the case that we do not allow quantum entanglement.

Theorem 7.3.2 *Without quantum entanglement, there is no one-round protocol for G_m that uses less than $3m + 1$ bits of communication.*

For one-round protocols we thus archive a separation of $2m+1$ bits against $3m+1$ bits. We do not know the classical communication complexity of computing G_m without any restriction on the number of rounds.

7.3.1 Classical upper bound

The lower bound in Theorem 7.3.2 is tight as there is a straightforward one-round protocol that computes G_m with $3m + 1$ bits of communication. It is instructive for understanding the proof of our lower bound, first to understand that protocol.

Consider an input $\mathbf{x} \in U^m$ to Alice. We can think of $\mathbf{x} = (x_1, \ldots, x_m)$ as consisting of two parts, the high bits and the low bits. That is, we identify $\mathbf{x}$ with the pair $(\mathbf{x}_{\text{high}}, \mathbf{x}_{\text{low}})$, where the i-th coordinate in $\mathbf{x}_{\text{high}} \in \{0,1\}^m$ is $(x_i \text{ div } 2)$, and where the i-th coordinate in $\mathbf{x}_{\text{low}} \in \{0,1\}^m$ is $(x_i \bmod 2)$. We think of Bob's input $\mathbf{y} = (y_1, \ldots, y_m)$ and Carol's input $\mathbf{z} = (z_1, \ldots, z_m)$ in a similar manner.

The $3m + 1$ one-round protocol works as follows: First Bob broadcasts all $2m$ bits of his input $(\mathbf{y}_{\text{high}}, \mathbf{y}_{\text{low}})$. Then Carol broadcasts the m high bits $\mathbf{z}_{\text{high}}$ of her input. Now Alice is capable of computing the value of f on all m instances, that is, she can compute $f(x_i, y_i, z_i)$ for all $1 \leq i \leq m$. Due to the promise that $(x_i + y_i + z_i) \bmod 2 = 0$, she does not need the low bits $\mathbf{z}_{\text{low}}$ of Carol's input. Finally Alice checks if $f(x_i, y_i, z_i) = 1$ for all $1 \leq i \leq m$. If so, $G_m(\mathbf{x}, \mathbf{y}, \mathbf{z}) = 1$ and Alice therefore broadcasts 1, otherwise she broadcasts 0.

Intuitively, Alice has to have all of Bob's m high bits, all of Carol's m high bits, but just m of the $2m$ low bits. Hence, *intuitively*, if there exists a protocol for G_m in which Bob broadcasts s_B bits and Carol broadcasts s_C bits, then s_B should be at least m, s_C at least m, and $s_B + s_C$ at least $3m$. It is the result of the following subsection that this intuition is valid.

7.3.2 Classical lower bound

We now prove our lower bound stated in Theorem 7.3.2. Since we only consider one-round protocols, we can without loss of generality assume that any protocol computing G_m is made up of the following three parts:

1. Bob (knowing only his input $\mathbf{y}$) broadcasts the message $\sigma_B = \sigma_B(\mathbf{y})$.

2. Carol (knowing her input $\mathbf{z}$ and Bob's message σ_B) broadcasts the message $\sigma_C = \sigma_C(\mathbf{z}, \sigma_B)$.

3. Alice (knowing $\mathbf{x}$, σ_B, and σ_C) computes the answer $\sigma_A(\mathbf{x}, \sigma_B, \sigma_C) \in \{0,1\}$ which she then broadcasts to Bob and Carol. Since this protocol computes

G_m, we can without loss of generality assume that $\sigma_A = G_m$ on all valid inputs.

In agreement with our intuition described above, the following key lemma explicitly specifies 2^{3m} different inputs on which Bob and/or Carol have to send different messages. Theorem 2 is immediate.

Lemma 7.3.3 *Consider the above one-round protocol for computing G_m. Let σ_B and σ_C denote Bob's and Carol's messages on inputs $\mathbf{y} = (\mathbf{y}_{high}, \mathbf{y}_{low})$ and $\mathbf{z} = (\mathbf{z}_{high}, \mathbf{y}_{low})$, respectively. Let σ'_B and σ'_C denote Bob's and Carol's messages on inputs $\mathbf{y}' = (\mathbf{y}'_{high}, \mathbf{y}'_{low})$ and $\mathbf{z}' = (\mathbf{z}'_{high}, \mathbf{y}'_{low})$, respectively. Then the following holds.*

(i) If $\mathbf{y}_{high} \neq \mathbf{y}'_{high}$ and $\mathbf{y}_{low} = \mathbf{y}'_{low}$, then $\sigma_B \neq \sigma'_B$.

(ii) If $\mathbf{z}_{high} \neq \mathbf{z}'_{high}$ and $\mathbf{y}_{low} = \mathbf{y}'_{low}$, then $\sigma_C \neq \sigma'_C$.

(iii) If $\mathbf{y}_{low} \neq \mathbf{y}'_{low}$, then $\sigma_B \neq \sigma'_B$ or $\sigma_C \neq \sigma'_C$.

Proof. We first prove (i) by contradiction. Assume $\mathbf{y}_{\text{high}} \neq \mathbf{y}'_{\text{high}}$, $\mathbf{y}_{\text{low}} = \mathbf{y}'_{\text{low}}$, and $\sigma_B = \sigma'_B$. Let $\mathbf{x}$ be the unique input to Alice such that $G_m(\mathbf{x}, \mathbf{y}, \mathbf{z}) = 1$. Then $(\mathbf{x}, \mathbf{y}', \mathbf{z}) = (\mathbf{x}, (\mathbf{y}'_{\text{high}}, \mathbf{y}_{\text{low}}), \mathbf{z})$ is a valid input on which G_m takes the value 0. But, since $\sigma_B = \sigma'_B$, we also have $\sigma_C(\mathbf{z}, \sigma_B) = \sigma'_C(\mathbf{z}, \sigma'_B)$, and hence Alice incorrectly outputs the same answer $\sigma_A(\mathbf{x}, \sigma_B, \sigma_C) = \sigma_A(\mathbf{x}, \sigma'_B, \sigma'_C)$ in both cases. Thus, the assumption is wrong and (i) holds.

The proof of (ii) is almost identical to the proof of (i), and we therefore omit it.

We also prove (iii) by contradiction. Assume $\mathbf{y}_{\text{low}} \neq \mathbf{y}'_{\text{low}}$, $\sigma_B = \sigma'_B$, and $\sigma_C = \sigma'_C$. Let $\mathbf{x} = (\mathbf{x}_{\text{high}}, \mathbf{0})$ be the unique input to Alice such that $G_m(\mathbf{x}, \mathbf{y}, \mathbf{z}) = 1$. Since the protocol correctly computes G_m, then Alice must answer 1 on the input $(\mathbf{x}, \mathbf{y}, \mathbf{z})$. But then $(\mathbf{x}, \mathbf{y}', \mathbf{z}')$ is also a valid input on which Alice answers 1. Further, let $\mathbf{x}' = (\mathbf{x}'_{\text{high}}, \mathbf{x}'_{\text{low}})$ be the unique input to Alice such that $G_m(\mathbf{x}', \mathbf{y}', \mathbf{z}) = 1$. Since the protocol correctly computes G_m, then Alice must answer 1 on the input $(\mathbf{x}', \mathbf{y}', \mathbf{z})$. But then $(\mathbf{x}', \mathbf{y}, \mathbf{z}')$ is also a valid input on which Alice answers 1.

Thus, Alice answers 1 on all of these 4 valid inputs: $(\mathbf{x}, \mathbf{y}, \mathbf{z})$, $(\mathbf{x}, \mathbf{y}', \mathbf{z}')$, $(\mathbf{x}', \mathbf{y}', \mathbf{z})$, and $(\mathbf{x}', \mathbf{y}, \mathbf{z}')$. But, since $\mathbf{y}_{\text{low}} \neq \mathbf{y}'_{\text{low}}$, then (as we show in the next paragraph) G_m takes the value 0 on at least one of the valid inputs $(\mathbf{x}, \mathbf{y}', \mathbf{z}')$ and $(\mathbf{x}', \mathbf{y}, \mathbf{z}')$, and thus the protocol incorrectly computes G_m. Hence, the assumption is wrong and (iii) follows.

To see that G_m has to take the value 0 on at least one of the valid inputs $(\mathbf{x}, \mathbf{y}', \mathbf{z}')$ and $(\mathbf{x}', \mathbf{y}, \mathbf{z}')$, assume otherwise. Let $1 \leq i \leq m$ be a coordinate where $\mathbf{y}_{\text{low}}$ and $\mathbf{y}'_{\text{low}}$ differ. For ease of notation, we let y_{low} denote the i-th coordinate (bit) of $\mathbf{y}_{\text{low}} \in \{0,1\}^m$, and we use similar notation for the i-th coordinate of the other vectors. Since $G_m(\mathbf{x}, \mathbf{y}, \mathbf{z}) = 1$, then

$$(x_{\text{high}} + y_{\text{high}} + z_{\text{high}} + y_{\text{low}}) \bmod 2 = 0.$$

Since $G_m(\mathbf{x}', \mathbf{y}', \mathbf{z}) = 1$, then

$$(x'_{\text{high}} + y'_{\text{high}} + z_{\text{high}} + 1) \bmod 2 = 0.$$

Since $G_m(\mathbf{x}, \mathbf{y}', \mathbf{z}') = 1$, then

$$(x_{\text{high}} + y'_{\text{high}} + z'_{\text{high}} + y'_{\text{low}}) \bmod 2 = 0.$$

Since $G_m(\mathbf{x}', \mathbf{y}, \mathbf{z}') = 1$, then

$$(x'_{\text{high}} + y_{\text{high}} + z'_{\text{high}} + 1) \bmod 2 = 0.$$

But all of these 4 equations cannot hold at the same time, and thus the assumption that G_m takes the value 1 on $(\mathbf{x}, \mathbf{y}', \mathbf{z}')$ and $(\mathbf{x}', \mathbf{y}, \mathbf{z}')$ is wrong. This completes our proof of Lemma 7.3.3, from which Theorem 7.3.2 immediately follows. ■

It is worthy noticing that, by Lemma 7.3.3, for Alice to correctly output the value of G_m, she has to be able to correctly compute f on every one of the m instances

of f. This is in general not so, and it is a deep open question in communication complexity to characterize the functions that possess this property.

7.4 Multirounds and multiparties

We now generalize f defined in Eq. 7.2 to a function F which we shall use to prove a separation in terms of the number of parties. There are k parties, where party P_i obtains input data $x_i \in V = \{0, \ldots, 2^n - 1\}$ $(i = 1, \ldots, k)$. We say that an input $\mathbf{x} = (x_1, \ldots, x_k)$ is *valid* if it satisfies that

$$\left(\sum_{i=1}^{k} x_i \right) \bmod 2^{n-1} = 0. \tag{7.4}$$

Let $F : V^k \to \{0, 1\}$ denote the Boolean function on the valid inputs defined by

$$F(\mathbf{x}) = \frac{1}{2^{n-1}} \left[\left(\sum_{i=1}^{k} x_i \right) \bmod 2^n \right]. \tag{7.5}$$

We say that a valid input $\mathbf{x}$ is *b-valid* if $F(\mathbf{x}) = b$ $(b = 0, 1)$. The function F can be viewed as computing the n-th least significant bit of the sum of the x_i's.

We first show that with prior quantum entanglement, k bits of communication are necessary and sufficient for every party to evaluate F. That is, for all $k \geq 2$ and $n \geq 1$,

$$Q(F, k, n) = k. \tag{7.6}$$

Then, we show how the parties can evaluate F with roughly $k \log_2 k$ bits of communication without using any entangled particles. Specifically, for all $k \geq 2$ and $n \geq 1$,

$$C(F, k, n) \leq (k - 1)\{\lceil \log_2(k - 1) \rceil + 1\} + 1. \tag{7.7}$$

Finally, we prove that this is optimal up to low order terms by showing that, for all $k \geq 2$ and $n \geq \log_2 k$,

$$C(F, k, n) > k \log_2(k) - k. \tag{7.8}$$

By comparing the bounds of Eqs. 7.6 and 7.8, we see that we have established a separation by a factor of $\log_2(k/2)$.

7.4.1 With entanglement

We first show that if the parties share entangled particles, then in a straightforward manner, the k parties can evaluate F using only one bit of communication each. This is obtained by a direct generalization of the idea used both in Sect. 2.1 of [39] (which itself is based on the work of Mermin [92]) and in [68]. The prior quantum entanglement shared by the k parties is the cat state $|q_1 \dots q_k\rangle = (|0 \dots 0\rangle + |1 \dots 1\rangle)/\sqrt{2}$, where party P_i holds qubit q_i $(i = 1, \dots, k)$.

Each party P_i uses the following procedure. First party P_i applies a phase-change operator $\phi(x_i)$ defined by $|0\rangle \mapsto |0\rangle$ and $|1\rangle \mapsto \exp(2\pi x_i \sqrt{-1}/2^n)|1\rangle$ on her qubit q_i. Thanks to the promise on the inputs, these phase rotations add up so that the resulting state is $(|0 \dots 0\rangle + (-1)^{F(\mathbf{x})}|1 \dots 1\rangle)/\sqrt{2}$. Then she applies the Walsh-Hadamard transform that maps $|0\rangle$ to $(|0\rangle + |1\rangle)/\sqrt{2}$, and $|1\rangle$ to $(|0\rangle - |1\rangle)/\sqrt{2}$. Finally, she measures her qubit q_i in the computational basis $\{|0\rangle, |1\rangle\}$ and broadcasts the outcoming bit.

Let b_i be the outcome of party P_i's measurement. Simple calculations show that $b_1 \oplus \cdots \oplus b_k$ equals $F(x_1, \dots, x_k)$, where $\oplus$ denotes addition in modulo-2 arithmetic. It follows that every party can compute the value of F from the k communicated bits. On the other hand, k bits of communication are necessary since if one of the parties does not broadcast any bits, then none of the others can determine the value of F. To see this, note that if we toggle the most significant bit of any one of the inputs, then the value of F changes. Equation 7.6 follows.

7.4.2 Without entanglement

The simplest way to evaluate the function F is for all but one of the parties to broadcast their inputs. The last party then evaluates $F(x_1, \ldots, x_k)$ and communicates the resulting bit to the others. Hence, the communication complexity (without entanglement) is at most $(k-1)n+1$.

Now, consider that all but one of the parties broadcast the d most significant bits of their inputs, for some integer $d \geq 1$. The last party, say P_k, then computes the sum $\left(\sum_{i=1}^{k} x_i\right) - \delta$ where

$$\delta = \sum_{i=1}^{k-1} (x_i \bmod 2^{n-d}).$$

Suppose $n \geq d$ where $d = 1 + \lceil \log_2(k-1) \rceil$. Then

$$0 \leq \delta \leq (k-1)(2^{n-d}-1) < 2^{n-1},$$

so party P_k knows the value of the sum $\sum_{i=1}^{k} x_i$ up to an additional non-negative term strictly smaller than 2^{n-1}. Since the sum is divisible by 2^{n-1} for all valid inputs, party P_k can determine it exactly and thus compute the value of F. It follows that $(k-1)d+1$ bits of communication suffice, as stated as Eq. 7.7.

A good method to prove lower bounds for the communication complexity of functions comes from a combinatorial view on the protocol for the communication. Consider the space V^k of all possible inputs, where $V = \{0, \ldots, 2^n - 1\}$. A *rectangle* in V^k is a subset $R \subseteq V^k$ such that $R = R_1 \times \cdots \times R_k$ for some $R_i \subseteq V$ $(i = 1, \ldots, k)$. If a rectangle contains no 0-valid inputs or no 1-valid inputs, then it is said to be *F-monochromatic.*

We now use the observation that every deterministic and errorless communication protocol corresponds to a covering of all the valid inputs in V^k by F-monochromatic rectangles (see [83]). Without increasing the communication complexity, such a protocol can always be transformed into a protocol that uses

a partitioning that covers all of V^k, and for which each monochromatic rectangle contains at least one valid input. By proving that every such partition requires at least t rectangles, we also prove that the communication complexity of F is at least $\log_2 t$ [83]. Hence, upper bounds on the cardinality of the possible F-monochromatic rectangles imply a lower bound on the communication complexity of F.

In the appendix, we prove that if a rectangle $R \subseteq V^k$ is F-monochromatic and if R contains a valid input, then its cardinality is upper bounded by a value r, for which

$$r = \left(\frac{2^n - 2}{k} + 1\right)^k. \tag{7.9}$$

Since there are 2^{nk} input values to be covered, this bound on the size of the rectangles shows that we need at least $t = 2^{nk}/r$ rectangles to partition V^k in the above described fashion.

If $n \geq \log_2 k$ and $k \geq 2$, then basic algebra gives that

$$\log_2 t = \log_2 \left(\frac{2^{nk}}{r}\right) > k \log_2(k) - k.$$

From this, the lower bound on the communication complexity of Eq. 7.8 follows.

We are grateful to Lucien Hardy for pointing out an error in an earlier version of this paper. We are grateful to Richard Cleve and Andrew Landahl for discussions. H. B. thanks Ricard Gavaldà for stimulating conversations. P. H. has been supported in part by the ESPRIT Long Term Research Programme of the EU under project number 20244 (ALCOM-IT). A. T. has been supported in part by a postgraduate fellowship from Canada's NSERC. Parts of this research were carried out while P. H. was at the Laboratoire d'informatique théorique et quantique at Université de Montréal, and during the 1997 Elsag-Bailey–I.S.I. Foundation research meeting on quantum computation.

7.5 Appendix: Upper bound on the cardinality of a monochromatic rectangle

Equip the set $V = \{0, \ldots, 2^n - 1\}$ with the natural addition operation, denoted $\oplus$ and given by $x \oplus y = (x + y) \bmod 2^n$. Then $V = \langle V, \oplus \rangle$ is a cyclic group of order 2^n.

Let $R \subseteq V^k$ be a fixed rectangle. By definition, $R = R_1 \times \cdots \times R_k$ for some subsets $R_i \subseteq V$, $i = 1, \ldots, k$. For any two subsets $A, B \subseteq V$, define $A \oplus B = \{a \oplus b \mid a \in A, b \in B\}$. We now define a family of subsets of V. Set $S_0 = \{0\} \subset V$ and $S_i = S_{i-1} \oplus R_i$ for $i = 1, \ldots, k$. Then for each element $(x_1, \ldots, x_k) \in R$, the value $\left(\sum_{i=1}^{k} x_i\right) \bmod 2^n$ is in S_k. We shall use Kneser's theorem [90] to give an upper bound on the cardinality of R.

Theorem 7.5.1 Kneser *Let $G = \langle G, \oplus \rangle$ be an Abelian group with finite subsets A and B. Then there exists a subgroup H of G such that*

$$A \oplus B \oplus H = A \oplus B$$

and

$$|A \oplus B| \geq |A \oplus H| + |B \oplus H| - |H|.$$

Let H_i be the largest subgroup of V for which $S_i = S_i \oplus H_i$, $(i = 0, \ldots, k)$. Since $\oplus$ is associative, then $H_{i-1} \subseteq H_i$ for all $1 \leq i \leq k$.

Suppose R is a monochromatic rectangle that contains a valid input. Without loss of generality, assume that it is a 0-valid input, that is, that $0 \in S_k$. Then H_i is the trivial subgroup $\{0\}$ for all i, since otherwise we have that $2^{n-1} \in H_i \subseteq H_k$ and hence R would not be monochromatic. This shows that if we identify $A = S_{i-1}$ and $B = R_i$ in Kneser's theorem, it follows that H is the trivial subgroup. We

therefore have that $|S_i| \geq |S_{i-1}| + |R_i| - 1$, so

$$|S_k| \geq \sum_{i=1}^{k} |R_i| - (k-1).$$

Since $2^{n-1} \notin S_k$, then $|S_k| \leq 2^n - 1$, so

$$\sum_{i=1}^{k} |R_i| \leq 2^n - 2 + k,$$

and therefore

$$|R| = \prod_{i=1}^{k} |R_i| \leq \left(\frac{2^n - 2}{k} + 1 \right)^k.$$

It follows that the right hand side of Eq. 7.9 is an upper bound on the cardinality of any F-monochromatic rectangle that contains a valid input.

Chapter 8

The cost of exactly simulating quantum entanglement with classical communication

This chapter is an extended version [31] of an article to appear in the journal *Physical Review Letter* [30]. It has been written with the collaboration of Gilles Brassard and Richard Cleve.

8.1 Abstract

We investigate the amount of communication that must augment classical local hidden variable models in order to simulate the behaviour of entangled quantum systems. We consider the scenario where a bipartite measurement is given from a set of possibilities and the goal is to obtain exactly the same correlations that arise when the actual quantum system is measured. We show that, in the case of a single pair of qubits in a Bell state, a constant number of bits of communication is always sufficient—regardless of the number of measurements

under consideration. We also show that, in the case of a system of n Bell states, a constant times 2^n bits of communication are necessary.

8.2 Introduction

Bell's celebrated theorem [11] shows that certain scenarios involving bipartite quantum measurements result in correlations that are impossible to simulate with a classical system if the measurement events are space-like separated. If the measurement events are time-like separated then classical simulation is possible, at the expense of some communication. Our goal is to quantify the required amount of communication.

The issue that we are addressing is part of the broader question of how quantum information affects various resources required to perform tasks in information processing. A two-way classical communication channel between two separated parties can be regarded as a *resource*, and a natural goal is for two parties to produce classical information satisfying a specific stochastic property. One question is, if the parties have an *a priori* supply of quantum entanglement, can they accomplish such goals with less classical communication than necessary in the case where their *a priori* information consists of only classical probabilistic information? And, if so, by how much? Our question is, to what extent does the fundamental behaviour of an entangled quantum system itself provide savings, in terms of communication, compared with classical systems?

Imagine a scenario involving two "particles" that may have been "together" (and interacted) at some previous point in time, but are "separated" (in a sense which implies that they can no longer interact) at the present time. Suppose that a measurement is then arbitrarily selected and performed on each particle (not necessarily the same measurement on both particles). If the underlying physics governing the behaviour of the system is "classical" then the behaviour of such a system could be based on correlated random variables (usually called "local

hidden variables"), reflecting the possible results of a previous interaction. If no communication can occur between the components at the time when the measurements take place then this imposes restrictions on the possible behaviour of such a system. In fact, if the underlying physics governing the behaviour of the system is "quantum" (in the sense that it can be based on entangled quantum states, rather than correlated random variables) then behaviour can occur that is impossible in the classical case. This is a natural way of interpreting Bell's theorem [11, 46]. To formalize—and later generalize—this, we shall define *quantum measurement scenarios* and *(classical) local hidden variable schemes.*

8.3 Definitions and preliminary results

Define a *quantum measurement scenario* as a triple of the form $(|\Psi\rangle_{AB}, M_A, M_B)$, where $|\Psi\rangle_{AB}$ is a bipartite quantum state, M_A is a set of measurements on the first component, and M_B is a set of measurements on the second component.

It is convenient to parametrize the simplest von Neumann measurements on individual qubits by points on the unit circle. (More general von Neumann measurements, which involve complex numbers, are considered later in this paper.) Let the parameter $x \in [0, 2\pi)$ denote a measurement with respect to the operator

$$R(x) = \begin{pmatrix} \cos x & \sin x \\ \sin x & -\cos x \end{pmatrix} \tag{8.1}$$

whose eigenvectors are $\cos(\frac{x}{2})|0\rangle + \sin(\frac{x}{2})|1\rangle$ and $\sin(\frac{x}{2})|0\rangle - \cos(\frac{x}{2})|1\rangle$.

Consider the case of a pair of qubits in the Bell state $|\Phi^+\rangle_{AB} = \frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|1\rangle$. [Our results are written for such states, but can be modified to apply to any of the other Bell states, including the Einstein-Podolsky-Rosen singlet state $|\Psi^-\rangle_{AB} = \frac{1}{\sqrt{2}}|0\rangle|1\rangle - \frac{1}{\sqrt{2}}|1\rangle|0\rangle$.] Let $x, y \in [0, 2\pi)$ be the respective measurement parameters of the two components and let $a, b \in \{0, 1\}$ be the respective outcomes. Then the joint probability distribution of these outcomes is given as:

	$\Pr[b=0]$	$\Pr[b=1]$
$\Pr[a=0]$	$\frac{1}{2}\cos^2(\frac{x-y}{2})$	$\frac{1}{2}\sin^2(\frac{x-y}{2})$
$\Pr[a=1]$	$\frac{1}{2}\sin^2(\frac{x-y}{2})$	$\frac{1}{2}\cos^2(\frac{x-y}{2})$

Two simple but noteworthy examples of bipartite quantum measurement scenarios with the Bell state $|\Phi^+\rangle_{AB}$ are:

Example 8.3.1: $(|\Phi^+\rangle_{AB}, M_A, M_B)$, where $M_A = M_B = \{0, \frac{\pi}{2}\}$.

Example 8.3.2: $(|\Phi^+\rangle_{AB}, M_A, M_B)$, where $M_A = \{-\frac{\pi}{8}, \frac{3\pi}{8}\}$ and $M_B = -M_A = \{\frac{\pi}{8}, -\frac{3\pi}{8}\}$.

In both examples, each individual outcome is a uniformly distributed bit regardless of the measurements. In Example 8.3.1, if the two measurements are the same then the outcomes are completely correlated; whereas, if the two measurements are different, the outcomes are completely independent. In Example 8.3.2, the two outcomes are equal with probability $\sin^2(\frac{\pi}{8})$ if $x = -y = +\frac{3\pi}{8}$; and with probability $\cos^2(\frac{\pi}{8})$ otherwise. These examples are interesting in the context of local hidden variable schemes, which are defined next.

Intuitively, we are interested in classical devices that simulate bipartite quantum measurement scenarios to varying degrees, and such devices are naturally explained as local hidden variable schemes. To define a *local hidden variable scheme*, it is convenient to view it as a two-party procedure whose execution occurs in two stages: a *preparation stage* and a *measurement stage*. For ease of reference, call the two parties Alice and Bob. During the preparation stage, *local hidden variables* u for Alice and v for Bob are determined by a classical random process. During this stage, arbitrary communication can occur between the two parties, so u and v may be arbitrarily correlated. During the measurement stage, measurements x and y are given to Alice and Bob (respectively), who produce outcomes $a = A(x, u)$ and $b = B(y, v)$ (respectively). During this stage, no com-

munication is permitted between the parties, which is reflected by the fact that the value of $A(x, u)$ is independent of the value of y (and vice versa).

A local hidden variable scheme *simulates* a measurement scenario $(|\Psi\rangle_{AB}, M_A, M_B)$ if, for any $x \in M_A$ and $y \in M_B$, the outputs produced by Alice and Bob, (namely, a and b respectively), have exactly the same bivariate distribution as the outcomes of the quantum measurement scenario as dictated by the laws of quantum physics.

The measurement scenario in Example 8.3.1 is easily simulatable by the following local hidden variable scheme. Let u and v each consist of a copy of the *same* uniformly distributed two-bit string. Then let Alice and Bob each output the first bit of this string if their measurement is 0 and the second bit if their measurement is $\frac{\pi}{2}$. On the other hand, for the measurement scenario of Example 8.3.2, it turns out that *there does not exist* a local hidden variable scheme that simulates it [46].

Now, we consider a more powerful classical instrument for simulating measurement scenarios. Define a local hidden variable scheme *augmented by k bits of communication*, as follows. Informally, it is a local hidden variable scheme, except that the prohibition of communication between the parties during the measurement stage is relaxed to a condition that allows up to k bits of communication. More formally, a local hidden variable scheme augmented by k bits of communication, has a preparation stage where random variables u and v for Alice and Bob are determined and during which arbitrary communication is permitted between the two parties. Then there is a measurement stage which begins by measurements x and y being given to Alice and Bob (respectively). Then one party computes a bit (as a function of his/her measurement and local hidden variables) which is sent to the other party. This constitutes one *round* of communication. Then again one party (the same one or a different one) computes a bit (as a function of his/her measurement, local hidden variables, and any data communicated from the other party at previous rounds) and sends it to the other party.

And this continues for k rounds, after which Alice and Bob output bits a and b (respectively).

For example, for the measurement scenario of Example 8.3.2, a local hidden variable scheme augmented with one single bit of communication can simulate it. This is a consequence of the following more general result, whose easy proof we include for completeness.

Theorem 8.3.1 *For any quantum measurement scenario* $(|\Psi\rangle_{AB}, M_A, M_B)$, *there exists a local hidden variable scheme augmented with* $\log_2(|M_A|)$ *bits of communication (from Alice to Bob) that exactly simulates it.*

Proof. First note that, if we allow $\log_2(|M_A|)$ bits of communication from Alice to Bob *and* $\log_2(|M_B|)$ bits of communication from Bob to Alice then it is trivial to simulate the quantum measurement scenario. With this much communication, Alice can obtain y and Bob can obtain x, which effectively defeats any "nonlocality" in the scenario. More precisely, during the preparation stage, Alice and Bob can construct $|M_A| \cdot |M_B|$ random variable pairs, $(a^{(x,y)}, b^{(x,y)})$, one for each value of $x \in M_A$ and $y \in M_B$. Each such random variable pair would specify the values of the outcomes of Alice and Bob for the given values of x and y, with the appropriate correlation. During the measurement stage, after the communication of x and y between them, Alice and Bob can simply output $a^{(x,y)}$ and $b^{(x,y)}$ (respectively).

To obtain a protocol in which only $\log_2(|M_A|)$ bits of communication from Alice to Bob occurs, note that the unconditional probability distribution of $a^{(x,y)}$ (the output of Alice when the measurements are x and y) is independent of the value of y. This is because the distribution of $a^{(x,y)}$ is completely determined by x and the reduced density matrix of $|\Psi\rangle_{AB}$ with the second component traced out $(\mathrm{Tr}_B(|\Psi\rangle_{AB}))$, and this quantity is independent of y. Therefore, the local hidden variables can be set up as follows. For each $x \in M_A$, $a^{(x)}$ is sampled according to the appropriate probability distribution, and then, for each $x \in M_A$ and $y \in M_B$, $b^{(x,y)}$ is sampled according the appropriate conditional probability distribution

(conditioned on the value of $a^{(x)}$) in order to produce the correct bivariate distribution for $(a^{(x)}, b^{(x,y)})$. Then, during the measurement stage it suffices for Alice to send x to Bob, and for Alice and Bob to output $a^{(x)}$ and $b^{(x,y)}$ (respectively).

■

We shall see that in some cases the upper bound of Theorem 8.3.1 is asymptotically tight while in other cases it is not. In the sections that follow, we focus on the case of a single Bell state and the case of n Bell states, and provide a new upper or lower bound in each case.

8.4 The case of a single Bell state

Consider the case of a single Bell state $|\Phi^+\rangle_{AB} = \frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|1\rangle$, but where the sizes of M_A and M_B may be arbitrarily large. By Theorem 8.3.1, we only obtain an upper bound of $\log_2(|M_A|)$ bits for the amount of communication necessary for an augmented local hidden variable scheme to simulate it. In the case where M_A and M_B are each the entire interval $[0, 2\pi)$, this communication upper bound would be infinite. If only a finite number, k, bits of communication are permitted then one alternative that might seem reasonable is for Alice to send x', a k-bit approximation of x, to Bob. The protocol for Alice and Bob would be along the lines of the one in Theorem 8.3.1, but using x' in place of x. This would clearly not produce an exact simulation for a general $x \in [0, 2\pi)$, but it would produce an *approximation* that improves as k increases. Is this the best that can be done with k bits of communication? The next theorem demonstrates that it is possible to obtain an *exact* simulation for any $x, y \in [0, 2\pi)$ with only a *constant* number of bits of communication.

Theorem 8.4.1 *For the quantum measurement scenario* $(|\Phi^+\rangle_{AB}, M_A, M_B)$ *with* $|\Phi^+\rangle_{AB} = \frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|1\rangle$ *and* $M_A = M_B = [0, 2\pi)$, *there exists a local hidden variable scheme augmented with four bits of communication (from Alice to Bob) that exactly simulates it.*

Proof. The local hidden variables are $c \in \{0,1\}$ and $\theta \in [0, \frac{3\pi}{5})$, and both are uniformly distributed.

For $j \in \{0,1,\ldots,9\}$, define $\alpha_j = \frac{j\pi}{5}$. It is useful to view $\alpha_0, \alpha_1, \ldots, \alpha_9$ as ten equally-spaced points on the unit circle. Define the j^{th} *α-slot* as the interval $[\alpha_j, \alpha_{(j+1) \bmod 10})$. Also, define $\beta_0 = \alpha_0 + \theta$, $\beta_1 = \alpha_3 + \theta$, and $\beta_2 = \alpha_6 + \theta$ and $\gamma_0 = \alpha_5 + \theta$, $\gamma_1 = \alpha_8 + \theta$, and $\gamma_2 = \alpha_1 + \theta$ (where the addition is understood to be modulo 2π). Define the j^{th} *β-slot* as the interval $[\beta_j, \beta_{(j+1) \bmod 3})$, and the j^{th} *γ-slot* as the interval $[\gamma_j, \gamma_{(j+1) \bmod 3})$.

The protocol starts by Alice sending Bob information specifying the α-slot, β-slot, and γ-slot in which x is located. Note that these slots partition the unit circle into sixteen intervals, so Alice can convey this information by sending four bits to Bob. Then Alice outputs the bit c.

The full procedure for Bob is summarized below, but, in order to explain the idea behind it, it is helpful to first consider the special case where y is in the 2^{nd} α-slot and the α-slot number of x is within two of that of y (in other words, the α-slot number of x is in $\{0,1,2,3,4\}$). Note that these conditions depend on the values of x and y only (and not on the values of the local hidden variables). Also, these conditions imply that $|x - y| \leq \frac{3\pi}{5}$. In this case, Bob does the following. If the β-slots of x and y are the same then Bob outputs c. If the β-slots of x and y are different then exactly one β_k is between x and y. Let $u = |y - \beta_k|$. Then Bob's procedure is to output c with probability $1 - \frac{3\pi}{10}\sin(u)$.

To analyse the stochastic behaviour of this procedure (still in the special case), let $r = |x - y|$ and note that the probability of x and y being in different β-slots is $\frac{5r}{3\pi}$. Also, conditional on x and y being in different β-slots, the probability distribution of the position of the β_k between x and y is uniform. Therefore,

$$\begin{aligned} \Pr[a = b] &= \left(1 - \tfrac{5r}{3\pi}\right) + \left(\tfrac{5r}{3\pi}\right)\left(\tfrac{1}{r}\right)\int_0^r \left(1 - \tfrac{3\pi}{10}\sin(u)\right)du \\ &= \tfrac{1}{2}(1 + \cos(r)) \\ &= \cos^2(\tfrac{r}{2}), \end{aligned} \tag{8.2}$$

which is exactly what is required.

The procedure for Bob in the above special case can be generalized to apply to the other possible cases by considering various similarities and symmetries among the cases. First note that the above procedure actually works in all cases where the α-slot number of y is in $\{2,3,4,5,6\}$ and the α-slot number of x is within two of that of y. This is because, in these cases, the interval between x and y (of length $\leq \frac{3\pi}{5}$) lies entirely within the interval $[0, \frac{9\pi}{5})$ and $\beta_0, \beta_1, \beta_2$ are uniformly distributed points spaced $\frac{3\pi}{5}$ apart in this interval.

Now, consider the cases where the α-slot number of y is in $\{7,8,9,0,1\}$ and the α-slot number of x is still within two of that of y. In these cases, the interval containing x and y may not lie entirely within $[0, \frac{9\pi}{5})$, and so the distribution of $\beta_0, \beta_1, \beta_2$ may no longer satisfy the relevant properties. To avoid this problem, Bob applies the above procedure with $\gamma_0, \gamma_1, \gamma_2$ substituted in place of $\beta_0, \beta_1, \beta_2$. This works because $\gamma_0, \gamma_1, \gamma_2$ are uniformly distributed points spaced $\frac{3\pi}{5}$ apart in the interval $[\pi, \frac{4}{5}\pi)$ (taken *clockwise*) and the interval containing x and y is within this interval.

The above covers all cases where the α-slot number of x is within two of that of y. To handle the remaining cases, Bob works with $y' = y + \pi$ (whose α-slot number will then be within two of that of x) instead of y. Let $r' = |x - y'|$. Then, since $\cos^2(\frac{r'}{2}) = \sin^2(\frac{r}{2})$, Bob will obtain the required distribution if he applies the above procedure but negates his output bit.

In summary, Bob's procedure after obtaining information specifying the α-slot, β-slot, and γ-slot of x from Alice is:

if the difference between the α-slot numbers of x and y is more than 2 then

    set y to $y + \pi$

    set c to $\neg c$

if the α-slot number of y is in $\{7,8,9,0,1\}$ then

    set $\beta_0, \beta_1, \beta_2$ to $\gamma_0, \gamma_1, \gamma_2$

if x **and** y **are in the same** β**-slot then**

  output c

else there exists a β_k **between** x **and** y

  set u **to** $|y - \beta_k|$

  output c **with probability** $1 - \frac{3\pi}{10}\sin(u)$ ∎

Theorem 8.4.1 applies to all measurements with respect to operators of the form given in Eq. (8.1). The most general possible von Neumann measurement on an individual qubit can be parametrized by $(x, x') \in [0, 2\pi) \times [0, 2\pi)$ and taken with respect to the operator

$$S(x, x') = \begin{pmatrix} \cos x & e^{-ix'} \sin x \\ e^{ix'} \sin x & -\cos x \end{pmatrix} \tag{8.3}$$

whose eigenvectors are $\cos(\frac{x}{2})|0\rangle + e^{ix'}\sin(\frac{x}{2})|1\rangle$ and $\sin(\frac{x}{2})|0\rangle - e^{ix'}\cos(\frac{x}{2})|1\rangle$. If Alice and Bob make such measurements with respective parameters (x, x') and (y, y') and a and b are the respective outcomes then $\Pr[a = 0] = \Pr[b = 0] = \frac{1}{2}$ and

$$\Pr[a = b] \;=\; \cos^2(\tfrac{x'+y'}{2})\cos^2(\tfrac{x-y}{2}) + \sin^2(\tfrac{x'+y'}{2})\cos^2(\tfrac{x+y}{2}). \tag{8.4}$$

Theorem 8.4.2 *For the quantum measurement scenario* $(|\Phi^+\rangle_{AB}, M_A, M_B)$ *with* $|\Phi^+\rangle_{AB} = \frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|1\rangle$ *and* $M_A = M_B = [0, 2\pi) \times [0, 2\pi)$*, there exists a local hidden variable scheme augmented with eight bits of communication (from Alice to Bob) that exactly simulates it.*

Proof. The local hidden variable scheme consists of two executions of the four-bit protocol of Theorem 8.4.1. In the first execution, Alice and Bob use measurement parameters x' and $-y'$ to obtain output bits a' and b' (respectively) such that

$$\Pr[a' = b'] = \cos^2(\tfrac{x'+y'}{2}). \tag{8.5}$$

In the second execution, Alice and Bob use measurement parameters $(-1)^{a'}x$ and $(-1)^{b'}y$ to obtain their final output bits a and b (respectively). Note that

$$\Pr[a = b] = \begin{cases} \cos^2(\frac{x-y}{2}) & \text{if } a' = b' \\ \cos^2(\frac{x+y}{2}) & \text{if } a' \neq b', \end{cases} \tag{8.6}$$

which, combined with Eq. (8.5), implies Eq. (8.4) as required. ∎

We do not know whether a similar result holds in the case of quantum measurements that are more general than von Neumann measurements (e.g. positive operator valued measures).

8.5 The case of n Bell states

Consider the case of n Bell states, i.e. the tensor product of $|\Phi^+\rangle_{AB}$ with itself n times. This state can be written as $|\Phi^+\rangle_{AB}^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} |i\rangle|i\rangle$. Theorem 8.4.2 implies that any n *independent* von Neumann measurements performed on the n Bell states can be simulated by a local hidden variable scheme augmented with $8n$ bits of communication. In the case of *coherent* measurements on such a state, the exact simulation cost can be much larger, as shown by the following theorem.

Theorem 8.5.1 *There exists a pair of sets of measurements, M_A and M_B (each of size 2^{2^n}) on n qubits, such that, for the quantum measurement scenario $(|\Phi^+\rangle_{AB}^{\otimes n}, M_A, M_B)$ with $|\Phi^+\rangle_{AB}^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} |i\rangle|i\rangle$, any local hidden variable scheme must be augmented with a constant times 2^n bits of communication in order to exactly simulate it.*

Proof. The proof is based on connections between a measurement scenario and a communication complexity problem examined in [40]. We begin by defining a set of 2^{2^n} measurements, which we call *Deutsch-Jozsa* measurements, due to their connection with the algorithm in [58]. The measurements are parametrized by the set $\{0,1\}^{2^n}$. For a parameter value $z \in \{0,1\}^{2^n}$, we index the bits of z

by the set $\{0,1\}^n$. That is, for $i \in \{0,1\}^n$, z_i denotes the "i^{th}" bit of z. The measurement on n qubits corresponding to $z \in \{0,1\}^{2^n}$ is easily described as two unitary transformations followed by a measurement in the computational basis. The first unitary transformation is a phase shift that maps $|i\rangle$ to $(-1)^{z_i}|i\rangle$ for each $i \in \{0,1\}^n$. The second unitary transformation is the n-qubit Hadamard transformation, which maps $|i\rangle$ to

$$\frac{1}{\sqrt{2^n}} \sum_{j \in \{0,1\}^n} (-1)^{i \cdot j} |j\rangle, \tag{8.7}$$

where $i \cdot j$ is the inner product of the two n-bit strings i and j (that is, $i \cdot j = i_0 j_0 + i_1 j_1 + \cdots + i_{n-1} j_{n-1}$). These two unitary transformations are followed by a measurement in the computational basis $\{|i\rangle : i \in \{0,1\}^n\}$, yielding an outcome in $\{0,1\}^n$.

Set $M_A = M_B = \{0,1\}^{2^n}$, the set of Deutsch-Jozsa measurements. We will now show that, for $x \in M_A$ and $y \in M_B$, the joint probability distribution of the outcomes a and b satisfies the following properties:

1. If $x = y$ then $\Pr[a = b] = 1$.

2. If the Hamming distance between x and y is 2^{n-1} then $\Pr[a = b] = 0$.

To show this, consider the quantum state after the phase flips and Hadamard transformations have been performed, but before the measurement. First, applying the phase flips to $|\Phi^+\rangle_{AB}^{\otimes n}$ yields the state

$$\frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} (-1)^{x_i + y_i} |i\rangle |i\rangle. \tag{8.8}$$

Next, after applying the Hadamard transformations, the state becomes

$$\frac{1}{\sqrt{2^{3n}}} \sum_{j,k,i \in \{0,1\}^n} (-1)^{x_i + y_i + i \cdot (j \oplus k)} |j\rangle |k\rangle \tag{8.9}$$

(where $j \oplus k$ is the bit-wise exclusive-or of j and k). To prove property 1, note that if $x = y$ then state (8.9) becomes

$$\frac{1}{\sqrt{2^{3n}}} \sum_{j,k,i \in \{0,1\}^n} (-1)^{i \cdot (j \oplus k)} |j\rangle |k\rangle = \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} |i\rangle |i\rangle,$$

so $\Pr[a = b] = 1$ when the measurement is performed. To prove property 2, note that if the Hamming distance between x and y is 2^{n-1} then $x_i + y_i$ is even for 2^{n-1} values of i and odd for 2^{n-1} values of i. Therefore, the amplitude of any ket of the form $|j\rangle|j\rangle$ in state (8.9) is

$$\frac{1}{\sqrt{2^{3n}}} \sum_{i \in \{0,1\}^n} (-1)^{x_i + y_i} = 0, \tag{8.10}$$

so $\Pr[a = b] = 0$.

Now we reduce a communication complexity problem in [40] to the problem of designing an augmented local hidden scheme that satisfies properties 1 and 2. The communication complexity problem (called EQ' in [40]) is a restricted version of the "equality" problem, and is defined as follows. Alice and Bob get inputs $x, y \in \{0,1\}^{2^n}$ (respectively), and one of them (say, Bob) must output 1 if $x = y$ and 0 if the Hamming distance between x and y is 2^{n-1}. The output of Bob can be arbitrary in all other cases. In [40], it is proven that any classical protocol that exactly solves this restricted equality problem requires $c2^n$ bits of communication for some constant $c > 0$ and all sufficiently large n. The proof is based on a combinatorial result in [64]. Suppose that there exists a local hidden variable scheme augmented with $f(n)$ bits of communication that simulates the measurement scenario $(|\Phi^+\rangle_{AB}^{\otimes n}, M_A, M_B)$. Then one can use this to construct a protocol for restricted equality with $f(n) + n$ bits of communication as follows. Alice and Bob first execute the protocol for $(|\Phi^+\rangle_{AB}^{\otimes n}, M_A, M_B)$ and then Alice sends her output a to Bob, who outputs 1 if $a = b$ and 0 if $a \neq b$. It follows that $f(n) + n \geq c2^n$, so $f(n) \geq c2^n - n \geq c'2^n$, for some $c' > 0$ and sufficiently large n. The theorem extends to all $n \geq 1$, possibly using a smaller constant

c'', because it follows from [46] that Example 8.3.2 cannot be simulated without communication. ■

In conclusion, we have shown how to *exactly* simulate the behaviour of a bipartite entangled quantum system consisting of two qubits prepared in a Bell state: 8 bits of classical communication suffice to reproduce exactly the correlations that arise when the qubits are independently subjected to arbitrary von Neumann measurements. In contrast, exact simulation of the behaviour of a bipartite quantum system consisting of n Bell states requires an amount of communication that is exponential in n.

Similar work was carried out independently by Michael Steiner [105]. Using a different technique, he showed how 1.48 classical bits of communication suffice *on the average* for the quantum measurement scenario $(|\Phi^+\rangle_{AB}, M_A, M_B)$ with $|\Phi^+\rangle_{AB} = \frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|1\rangle$ and $M_A = M_B = [0, 2\pi)$ that we consider in Theorem 8.4.1. It follows from the techniques used in our proof of Theorem 8.4.2 that 2.96 bits of classical communication suffice on the average to simulate the behaviour of arbitrary von Neumann measurements carried out independently on two qubits prepared in a Bell state. Although this is better than the 8 bits that we need in this letter, Steiner's technique imposes no upper limit on the number of bits that need be communicated in the worst case.

G.B. is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), Canada Council and Québec's Fonds Concerté pour l'Avancement de la Recherche. R.C. and A.T. are supported by NSERC.

Chapter 9

Quantum entanglement and the communication complexity of the inner product function

This chapter reproduces an article to appear in the proceeding *First NASA International Conference, QCQC'98* [50]. It has been written with the collaboration of Richard Cleve, Wim van Dam and Michael Nielsen.

9.1 Abstract

We consider the communication complexity of the binary inner product function in a variation of the two-party scenario where the parties have an *a priori* supply of particles in an entangled quantum state. We prove linear lower bounds for both exact protocols, as well as for protocols that determine the answer with bounded-error probability. Our proofs employ a novel kind of "quantum" reduction from a quantum information theory problem to the problem of computing the inner product. The communication required for the former problem can then be bounded by an application of Holevo's theorem. We also

give a specific example of a probabilistic scenario where entanglement reduces the communication complexity of the inner product function by one bit.

9.2 Introduction and summary of results

The *communication complexity* of a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is defined as the minimum amount of communication necessary among two parties, conventionally referred to as Alice and Bob, in order for, say, Bob to acquire the value of $f(x,y)$, where, initially, Alice is given x and Bob is given y. This scenario was introduced by Yao [111] and has been widely studied (see [83] for a survey). There are a number of technical choices in the model, such as: whether the communication cost is taken as the worst-case (x,y), or the average-case (x,y) with respect to some probability distribution; whether the protocols are deterministic or probabilistic (and, for probabilistic protocols, whether the parties have independent random sources or a shared random source); and, what correctness probability is required.

The communication complexity of the *inner product modulo two (IP)* function

$$IP(x,y) \;=\; x_1 \wedge y_1 \oplus x_2 \wedge y_2 \oplus \cdots \oplus x_n \wedge y_n \bmod 2 \tag{9.1}$$

is fairly well understood in the above "classical" models. For worst-case inputs and deterministic errorless protocols, the communication complexity is n and, for randomized protocols (with either an independent or a shared random source), uniformly distributed or worst-case inputs, and with error probability $\frac{1}{2} - \delta$ required, the communication complexity is $n - O(\log(1/\delta))$ [44] (see also [83]).

In 1993, Yao [112] introduced a variation of the above classical communication complexity scenarios, where the parties communicate with *qubits*, rather than with bits. Protocols in this model are at least as powerful as probabilistic protocols with independent random sources. Kremer [82] showed that, in this model,

the communication complexity of *IP* is $\Omega(n)$, whenever the required correctness probability is $1-\varepsilon$ for a constant $0 \leq \varepsilon < \frac{1}{2}$ (Kremer attributes the proof methodology to Yao).

Cleve and Buhrman [48] (see also [39]) introduced another variation of the classical communication complexity scenario that also involves quantum information, but in a different way. In this model, Alice and Bob have an initial supply of particles in an entangled quantum state, such as Einstein-Podolsky-Rosen (EPR) pairs, but the communication is still in terms of classical bits. They showed that the entanglement enables the communication for a specific problem to be reduced by one bit. Any protocol in Yao's qubit model can be simulated by a protocol in this entanglement model with at most a factor two increase in communication: each qubit can be "teleported" [17] by sending two classical bits in conjunction with an EPR pair of entanglement. On the other hand, we are aware of no similar simulation of protocols in the entanglement model by protocols in the qubit model, and, thus, the entanglement model is potentially stronger.

In this paper, we consider the communication complexity of *IP* in two scenarios: with prior entanglement and qubit communication; and with prior entanglement and classical bit communication. As far as we know, the proof methodology of the lower bound in the qubit communication model without prior entanglement [82] does not carry over to either of these two models. Nevertheless, we show $\Omega(n)$ lower bounds in these models.

To state our lower bounds more precisely, we introduce the following notation. Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a communication problem, and $0 \leq \epsilon < \frac{1}{2}$. Let $Q^*_\varepsilon(f)$ denote the communication complexity of f in terms of *qubits*, where quantum entanglement is available and the requirement is that Bob determines the correct answer with probability at least $1-\varepsilon$ (the $*$ superscript is intended to highlight the fact that prior entanglement is available). Also, let $C^*_\varepsilon(f)$ denote the corresponding communication complexity of f in the scenario where the communication is in terms of *bits* (again, quantum entanglement is available and

Bob is required to determine the correct answer with probability at least $1-\varepsilon$). When $\varepsilon = 0$, we refer to the protocols as *exact*, and, when $\varepsilon > 0$, we refer to them as *bounded-error* protocols. With this notation, our results are:

$$Q_0^*(IP) = \lceil n/2 \rceil \tag{9.2}$$

$$Q_\varepsilon^*(IP) \geq \tfrac{1}{2}(1-2\varepsilon)^2 n - \tfrac{1}{2} \tag{9.3}$$

$$C_0^*(IP) = n \tag{9.4}$$

$$C_\varepsilon^*(IP) \geq \max(\tfrac{1}{2}(1-2\varepsilon)^2, (1-2\varepsilon)^4)n - \tfrac{1}{2} \tag{9.5}$$

Note that all the lower bounds are $\Omega(n)$ whenever ε is held constant. Also, these results subsume the lower bounds in [82], since the qubit model defined by Yao [112] differs from the bounded-error qubit model defined above only in that it does not permit a prior entanglement.

Our lower bound proofs employ a novel kind of "quantum" reduction between protocols, which reduces the problem of communicating, say, n bits of information to the *IP* problem. It is noteworthy that, in classical terms, it can be shown that there is no such reduction between the two problems. The appropriate cost associated with communicating n bits is then lower-bounded by the following nonstandard consequence of Holevo's theorem.

Theorem 9.2.1 *In order for Alice to convey n bits of information to Bob, where quantum entanglement is available and qubit communication in either direction is permitted, Alice must send Bob at least $\lceil n/2 \rceil$ qubits. This holds regardless of the prior entanglement and the qubit communication from Bob to Alice. More generally, for Bob to obtain m bits of mutual information with respect to Alice's n bits, Alice must send at least $\lceil m/2 \rceil$ qubits.*

A slight generalization of Theorem 9.2.1 is described and proven in the Appendix.

It should be noted that, since quantum information subsumes classical information, our results also represent new proofs of nontrivial lower bounds on the

classical communication complexity of IP, and our methodology is fundamentally different from those previously used for classical lower bounds.

Finally, with respect to the question of whether quantum entanglement can *ever* be advantageous for protocols computing IP, we present a curious probabilistic scenario with $n = 2$ where prior entanglement enables one bit of communication to be saved.

9.3 Bounds for exact qubit protocols

In this section, we consider exact qubit protocols computing IP, and prove Eq. (9.2). Note that the upper bound follows from so-called "superdense coding" [19]: by sending $\lceil n/2 \rceil$ qubits in conjunction with $\lceil n/2 \rceil$ EPR pairs, Alice can transmit her n classical bits of input to Bob, enabling him to evaluate IP. For the lower bound, we consider an arbitrary exact qubit protocol that computes IP, and convert it (in two stages) to a protocol for which Theorem9.2.1 applies.

For convenience, we use the following notation. If an m-qubit protocol consists of m_1 qubits from Alice to Bob and m_2 qubits from Bob to Alice then we refer to the protocol as an (m_1, m_2)-qubit protocol.

9.3.1 Converting exact protocols into clean form

A *clean protocol* is a special kind of qubit protocol that follows the general spirit of the reversible programming paradigm in a quantum setting. Namely, one in which all qubits incur no net change, except for one, which contains the answer.

In general, the initial state of a qubit protocol is of the form

$$\underbrace{|y_1, \ldots, y_n\rangle \otimes |0, \ldots, 0\rangle}_{\text{Bob's qubits}} \otimes |\Phi_{BA}\rangle \otimes \underbrace{|x_1, \ldots, x_n\rangle \otimes |0, \ldots, 0\rangle}_{\text{Alice's qubits}}, \tag{9.6}$$

where $|\Phi_{BA}\rangle$ is the state of the entangled qubits shared by Alice and Bob, and the $|0,\ldots,0\rangle$ states can be regarded as "ancillas". At each turn, a player performs some transformation (which, without loss of generality, can be assumed to be unitary) on all the qubits in his/her possession and then sends a subset of these qubits to the other player. Note that, due to the communication, the qubits possessed by each player varies during the execution of the protocol.

We say that a protocol which exactly computes a function $f(x,y)$ is *clean* if, when executed on the initial state

$$|z\rangle \otimes |y_1,\ldots,y_n\rangle \otimes |0,\ldots,0\rangle \otimes |\Phi_{BA}\rangle \otimes |x_1,\ldots,x_n\rangle \otimes |0,\ldots,0\rangle, \tag{9.7}$$

results in the final state

$$|z \oplus f(x,y)\rangle \otimes |y_1,\ldots,y_n\rangle \otimes |0,\ldots,0\rangle \otimes |\Phi_{BA}\rangle \otimes |x_1,\ldots,x_n\rangle \otimes |0,\ldots,0\rangle \tag{9.8}$$

(where the addition is mod 2). The "input", the ancilla, and initial entangled qubits will typically change states during the execution of the protocol, but they are reset to their initial values at the end of the protocol.

It is straightforward to transform an exact (m_1,m_2)-qubit protocol into a clean (m_1+m_2,m_1+m_2)-qubit protocol that computes the same function. To reset the bits of the input, the ancilla, and the initial entanglement after the protocol is run once, the answer is recorded and then the protocol is run in the *backwards* direction to "undo the effects of the computation". The answer is recorded on a *new* qubit of Bob (with initial state $|z\rangle$) which is control-negated (with the qubit of Bob that is in the state $|f(x,y)\rangle$ as the control qubit). Note that, for each qubit that Bob sends to Alice when the protocol is run forwards, Alice sends the qubit to Bob when run in the backwards direction. Running the protocol backwards resets all the qubits—except Bob's new one—to their original states. The result is an (m_1+m_2,m_1+m_2)-qubit protocol that maps state (9.7) to state (9.8).

9.3.2 Reduction from communication problems

We now show how to transform a clean $(m_1 + m_2, m_1 + m_2)$-qubit protocol that exactly computes IP for inputs of size n, to an (m_1+m_2, m_1+m_2)-qubit protocol that transmits n bits of information from Alice to Bob. This is accomplished in four stages:

1. Bob initializes his qubits indicated in Eq. (9.7) with $z = 1$ and $y_1 = \cdots = y_n = 0$.
2. Bob performs a Hadamard transformation on each of his first $n+1$ qubits.
3. Alice and Bob execute the clean protocol for the inner product function.
4. Bob again performs a Hadamard transformation on each of his first $n+1$ qubits.

Let $|B_i\rangle$ denote the state of Bob's first $n+1$ qubits after the i^{th} stage. Then

$$|B_1\rangle = |1\rangle \otimes |0, \ldots, 0\rangle \tag{9.9}$$

$$|B_2\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{a,b_1,\ldots,b_n \in \{0,1\}} (-1)^a |a\rangle \otimes |b_1, \ldots, b_n\rangle \tag{9.10}$$

$$\begin{aligned} |B_3\rangle &= \frac{1}{\sqrt{2^{n+1}}} \sum_{a,b_1,\ldots,b_n \in \{0,1\}} (-1)^a |a + b_1 x_1 + \cdots + b_n x_n\rangle \otimes |b_1, \ldots, b_n\rangle \\ &= \frac{1}{\sqrt{2^{n+1}}} \sum_{c,b_1,\ldots,b_n \in \{0,1\}} (-1)^{c + b_1 \wedge x_1 \oplus \cdots \oplus b_n \wedge x_n} |c\rangle \otimes |b_1, \ldots, b_n\rangle \end{aligned} \tag{9.11}$$

$$|B_4\rangle = |1\rangle \otimes |x_1, \ldots, x_n\rangle, \tag{9.12}$$

where, in Eq. (9.11), the substitution $c = a + b_1 x_1 + \cdots + b_n x_n$ has been made (and arithmetic over bits is taken mod 2). The above transformation was inspired by the reading of [108] (see also [20]).

Since the above protocol conveys n bits of information (namely, $x_1, \ldots, x_n$) from Alice to Bob, by Theorem 9.2.1, we have $m_1 + m_2 \geq n/2$. Since this protocol can be constructed from an arbitrary exact (m_1, m_2)-qubit protocol for IP, this establishes the lower bound of Eq. (9.2).

Note that, classically, no such reduction is possible. For example, if a clean protocol for IP is executed in any classical context, it can never yield more than one bit of information to Bob (whereas, in this quantum context, it yields n bits of information to Bob).

9.4 Lower bounds for bounded-error qubit protocols

In this section we consider bounded-error qubit protocols for IP, and prove Eq. (9.3). Assume that some qubit protocol P computes IP correctly with probability $1-\varepsilon$, where $0 < \varepsilon < \frac{1}{2}$. Since P is not exact, the constructions from the previous section do not work exactly. We analyze the extent by which they err.

First, the construction of Section 2.1 will not produce a protocol in clean form; however, it will result in a protocol which *approximates* an exact clean protocol (this type of construction was previously carried out in a different context by Bennett *et al.* [16]).

Denote the initial state as

$$|y_1, \ldots, y_n\rangle \otimes |0, \ldots, 0\rangle \otimes |\Phi_{BA}\rangle \otimes |x_1, \ldots, x_n\rangle \otimes |0, \ldots, 0\rangle. \tag{9.13}$$

Also, assume that, in protocol P, Bob never changes the state of his input qubits $|y_1, \ldots, y_n\rangle$ (so the first n qubits never change). This is always possible, since he can copy $y_1, \ldots, y_n$ into his ancilla qubits at the beginning. After executing P until just before the measurement occurs, the state of the qubits must be of the form

$$\alpha|y_1, \ldots, y_n\rangle \otimes |x \cdot y\rangle \otimes |J\rangle + \beta|y_1, \ldots, y_n\rangle \otimes |\overline{x \cdot y}\rangle \otimes |K\rangle, \tag{9.14}$$

where $|\alpha|^2 \geq (1-\varepsilon)$ and $|\beta|^2 \leq \varepsilon$. In the above, the $n+1^{\text{st}}$ qubit is the *answer* qubit, $x \cdot y$ denotes the inner product of x and y, and $\overline{x \cdot y}$ denotes the negation of this inner product. In general, α, β, $|J\rangle$, and $|K\rangle$ may depend on x and y.

Now, suppose that the procedure described in Section 2.1 for producing a clean protocol in the exact case is carried out for P. Since, in general, the answer qubit is not in the state $|x \cdot y\rangle$—or even in a pure basis state—this does not produce the final state

$$|z + x \cdot y\rangle \otimes |y_1, \ldots, y_n\rangle \otimes |0, \ldots, 0\rangle \otimes |\Phi_{BA}\rangle \otimes |x_1, \ldots, x_n\rangle \otimes |0, \ldots, 0\rangle. \tag{9.15}$$

However, let us consider the state that is produced instead. After introducing the *new* qubit, initialized in basis state $|z\rangle$, and applying P, the state is

$$|z\rangle \otimes (\alpha|y_1, \ldots, y_n\rangle \otimes |x \cdot y\rangle \otimes |J\rangle + \beta|y_1, \ldots, y_n\rangle \otimes |\overline{x \cdot y}\rangle \otimes |K\rangle). \tag{9.16}$$

After applying the controlled-NOT gate, the state is

$$\begin{aligned}
&\alpha|z + x \cdot y\rangle \otimes |y_1, \ldots, y_n\rangle \otimes |x \cdot y\rangle \otimes |J\rangle \\
&\qquad + \beta|z + \overline{x \cdot y}\rangle \otimes |y_1, \ldots, y_n\rangle \otimes |\overline{x \cdot y}\rangle \otimes |K\rangle && (9.17)\\
&= \alpha|z + x \cdot y\rangle \otimes |y_1, \ldots, y_n\rangle \otimes |x \cdot y\rangle \otimes |J\rangle && (9.18)\\
&\qquad + \beta|z + x \cdot y\rangle \otimes |y_1, \ldots, y_n\rangle \otimes |\overline{x \cdot y}\rangle \otimes |K\rangle \\
&\qquad - \beta|z + x \cdot y\rangle \otimes |y_1, \ldots, y_n\rangle \otimes |\overline{x \cdot y}\rangle \otimes |K\rangle && (9.19)\\
&\qquad + \beta|z + \overline{x \cdot y}\rangle \otimes |y_1, \ldots, y_n\rangle \otimes |\overline{x \cdot y}\rangle \otimes |K\rangle \\
&= |z + x \cdot y\rangle \otimes (\alpha|y_1, \ldots, y_n\rangle \otimes |x \cdot y\rangle \otimes |J\rangle + \beta|y_1, \ldots, y_n\rangle \otimes |\overline{x \cdot y}\rangle \otimes |K\rangle) \\
&\qquad + \sqrt{2}\beta \left(\tfrac{1}{\sqrt{2}}|z + \overline{x \cdot y}\rangle - \tfrac{1}{\sqrt{2}}|z + x \cdot y\rangle \right) \otimes |y_1, \ldots, y_n\rangle \otimes |\overline{x \cdot y}\rangle \otimes |K\rangle. && (9.20)
\end{aligned}$$

Finally, after applying P in reverse to this state, the final state is

$$|z + x \cdot y\rangle \otimes |y_1, \ldots, y_n\rangle \otimes |0, \ldots, 0\rangle \otimes |\Phi_{BA}\rangle \otimes |x_1, \ldots, x_n\rangle \otimes |0, \ldots, 0\rangle + \sqrt{2}\beta |M_{x,y,z}\rangle, \tag{9.21}$$

where

$$|M_{x,y,z}\rangle = \left(\tfrac{1}{\sqrt{2}}|z + \overline{x \cdot y}\rangle - \tfrac{1}{\sqrt{2}}|z + x \cdot y\rangle\right) \otimes P^{\dagger}|y_1, \ldots, y_n\rangle \otimes |\overline{x \cdot y}\rangle \otimes |K\rangle. \tag{9.22}$$

Note that the vector $\sqrt{2}\beta|M_{x,y,z}\rangle$ is the difference between what an exact protocol would produce (state (9.15)) and what is obtained by using the inexact (probabilistic) protocol P (state (9.21)). There are some useful properties of the $|M_{x,y,z}\rangle$ states. First, as $y \in \{0,1\}^n$ varies, the states $|M_{x,y,z}\rangle$ are orthonormal, since $|y_1, \ldots, y_n\rangle$ is a factor in each such state (this is where the fact that Bob does not change his input qubits is used). Also, $|M_{x,y,0}\rangle = -|M_{x,y,1}\rangle$, since only the $(\frac{1}{\sqrt{2}}|z + \overline{x \cdot y}\rangle - \frac{1}{\sqrt{2}}|z + x \cdot y\rangle)$ factor in each such state depends on z.

Call the above protocol $\tilde{P}$. Now, apply the four stage reduction in Section 2.2, with $\tilde{P}$ in place of an exact clean protocol. The *difference* between the state produced by using $\tilde{P}$ and using an exact clean protocol first occurs after the third stage and is

$$\begin{aligned} &\tfrac{1}{\sqrt{2^{n+1}}} \sum_{y_1,\ldots,y_n,z \in \{0,1\}} (-1)^z \sqrt{2}\beta_y |M_{x,y,z}\rangle \\ &= \tfrac{1}{\sqrt{2^{n+1}}} \sum_{y_1,\ldots,y_n \in \{0,1\}} \sqrt{2}\beta_y \left(|M_{x,y,0}\rangle - |M_{x,y,1}\rangle\right) \\ &= \tfrac{2}{\sqrt{2^n}} \sum_{y_1,\ldots,y_n \in \{0,1\}} \beta_y |M_{x,y,0}\rangle, \end{aligned} \tag{9.23}$$

which has magnitude bounded above by $2\sqrt{\varepsilon}$, since, for each $y \in \{0,1\}^n$, $|\beta_y|^2 \leq \varepsilon$, and the $|M_{x,y,0}\rangle$ states are orthonormal. Also, the magnitude of this difference does not change when the Hadamard transform in the fourth stage is applied.

Thus, the final state is within Euclidean distance $2\sqrt{\varepsilon}$ from

$$|1\rangle \otimes |x_1, \ldots, x_n\rangle \otimes |0, \ldots, 0\rangle \otimes |\Phi_{BA}\rangle \otimes |x_1, \ldots, x_n\rangle \otimes |0, \ldots, 0\rangle. \qquad (9.24)$$

Consider the angle θ between this final state and (9.24). It satisfies $\sin^2\theta + (1 - \cos\theta)^2 \leq 4\varepsilon$, from which it follows that $\cos\theta \geq 1 - 2\varepsilon$. Therefore, if Bob measures his first $n+1$ qubits in the standard basis, the probability of obtaining $|1, x_1, \ldots, x_n\rangle$ is $\cos^2\theta \geq (1 - 2\varepsilon)^2$.

Now, suppose that $x_1, \ldots, x_n$ are uniformly distributed. Then Fano's inequality (see, for example, [54]) implies that Bob's measurement causes his uncertainty about $x_1, \ldots, x_n$ to drop from n bits to less than $(1-(1-2\varepsilon)^2)n+h((1-2\varepsilon)^2)$ bits, where $h(x) = -x\log x - (1-x)\log(1-x)$ is the binary entropy function. Thus, the mutual information between the result of Bob's measurement and $(x_1, \ldots, x_n)$ is at least $(1-2\varepsilon)^2 n - h((1-2\varepsilon)^2) \geq (1-2\varepsilon)^2 n - 1$ bits. By Theorem 9.2.1, the communication from Alice to Bob is at least $\frac{1}{2}(1-2\varepsilon)^2 n - \frac{1}{2}$ qubits, which establishes Eq. (9.3).

9.5 Lower bounds for bit protocols

In this section, we consider exact and bounded-error bit protocols for IP, and prove Eqs. (9.4) and (9.5).

Recall that any m-qubit protocol can be simulated by a $2m$-bit protocol using teleportation [17] (employing EPR pairs of entanglement). Also, if the communication pattern in an m-bit protocol is such that an even number of bits is always sent during each party's turn then it can be simulated by an $m/2$-qubit protocol by superdense coding [19] (which also employs EPR pairs). However, this latter simulation technique cannot, in general, be applied directly, especially for protocols where the parties take turns sending single bits.

We can nevertheless obtain a slightly weaker simulation of bit protocols by qubit protocols for IP that is sufficient for our purposes. The result is that, given any m-bit protocol for IP_n (that is, IP instances of size n), one can construct an m-qubit protocol for IP_{2n}. This is accomplished by interleaving two executions of the bit protocol for IP_n to compute two independent instances of inner products of size n. We make two observations. First, by taking the sum (mod 2) of the two results, one obtains an inner product of size $2n$. Second, due to the interleaving, an even number of bits is sent at each turn, so that the above superdense coding technique can be applied, yielding a $(2m)/2 = m$-qubit protocol for IP_{2n}. Now, Eq. (9.2) implies $m \geq n$, which establishes the lower bound of Eq. (9.4) (and the upper bound is trivial).

If the same technique is applied to any m-bit protocol computing IP_n with probability $1-\varepsilon$, one obtains an m-qubit protocol that computes IP_{2n} with probability $(1-\varepsilon)^2 + \varepsilon^2 = 1 - 2\varepsilon(1-\varepsilon)$. Applying Eq. (9.3) here, with $2n$ replacing n and $2\varepsilon(1-\varepsilon)$ replacing ε, yields $m \geq (1-2\varepsilon)^4 n - \frac{1}{2}$. For $\varepsilon > \frac{2-\sqrt{2}}{4} = 0.146\ldots$, a better bound is obtained by simply noting that $C^*_\varepsilon \geq Q^*_\varepsilon$ (since qubits can always be used in place of bits), and applying Eq. (9.3). This establishes Eq. (9.5).

9.6 An instance where prior entanglement is beneficial

Here we will show that in spite of the preceding results, it is still possible that a protocol which uses prior entanglement outperforms all possible classical protocols. This improvement is done in the probabilistic sense where we look at the number of communication bits required to reach a certain reliability threshold for the IP function. This is done in the following setting.

Both Alice and Bob have a 2 bit vector x_1x_2 and y_1y_2, for which they want to calculate the inner product modulo 2:

$$f(x,y) = x_1 \wedge y_1 \oplus x_2 \wedge y_2 \bmod 2 \tag{9.25}$$

with a correctness-probability of at least $\frac{4}{5}$. It will be shown that with entanglement Alice and Bob can reach this ratio with 2 bits of communication, whereas without entanglement 3 bits are necessary to obtain this success-ratio.

9.6.1 A two-bit protocol with prior entanglement

Initially Alice and Bob share a joint random coin and an EPR-like pair of qubits Q_A and Q_B:

$$\text{state}(Q_AQ_B) = \tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{9.26}$$

With these attributes the protocol goes as follows.

First Alice and Bob determine by a joint random coin flip[1] who is going to be the 'sender' and the 'receiver' in the protocol. (We continue the description of the protocol by assuming that Alice is the sender and that Bob is the receiver.) After this, Alice (the sender) applies the rotation $A_{x_1x_2}$ on her part of the entangled pair and measures this qubit Q_A in the standard basis. The result m_A of this measurement is then sent to Bob (the receiver) who continues the protocol.

If Bob has the input string '00', he knows with certainty that the outcome of the function $f(x,y)$ is zero and hence he concludes the protocol by sending the bit 0 to Alice. Otherwise, Bob performs the rotation $B_{y_1y_2}$ on his part of the entangled pair Q_B and measure it in the standard basis yielding the value m_B. Now Bob finishes the protocol by sending to Alice the bit $m_A \oplus m_B \bmod 2$.

[1]Because a joint random coin flip can be simulated with an EPR-pair, we can also assume that Alice and Bob start the protocol with two shared EPR-pairs and no random coins.

Using the rotations shown below and bearing in mind the randomization process in the beginning of the protocol with the joint coin flip, this will be a protocol that uses only 2 bits of classical communication and that gives the correct value of $f(x,y)$ with a probability of at least $\frac{4}{5}$ for every possible combination of x_1x_2 and y_1y_2.

The unitary transformations used by the sender in the protocol are:

$$A_{00} = \begin{pmatrix} \sqrt{\frac{2}{5}} & -\mathrm{i}\sqrt{\frac{3}{5}} \\ -\mathrm{i}\sqrt{\frac{3}{5}} & \sqrt{\frac{2}{5}} \end{pmatrix}$$

$$A_{01} = \begin{pmatrix} \sqrt{\frac{4}{5}} & \sqrt{\frac{3}{16}} + \mathrm{i}\sqrt{\frac{1}{80}} \\ -\sqrt{\frac{3}{16}} + \mathrm{i}\sqrt{\frac{1}{80}} & \sqrt{\frac{4}{5}} \end{pmatrix} \tag{9.27}$$

$$A_{10} = \begin{pmatrix} \sqrt{\frac{4}{5}} & -\sqrt{\frac{3}{16}} + \mathrm{i}\sqrt{\frac{1}{80}} \\ \sqrt{\frac{3}{16}} + \mathrm{i}\sqrt{\frac{1}{80}} & \sqrt{\frac{4}{5}} \end{pmatrix} \quad A_{11} = \begin{pmatrix} \sqrt{\frac{1}{5}} & \mathrm{i}\sqrt{\frac{4}{5}} \\ \mathrm{i}\sqrt{\frac{4}{5}} & \sqrt{\frac{1}{5}} \end{pmatrix},$$

whereas the receiver uses one of the three rotations:

$$B_{01} = \begin{pmatrix} \sqrt{\frac{3}{5}} & -\frac{1}{2} + \mathrm{i}\sqrt{\frac{3}{20}} \\ -\frac{1}{2} - \mathrm{i}\sqrt{\frac{3}{20}} & -\sqrt{\frac{3}{5}} \end{pmatrix} \quad B_{10} = \begin{pmatrix} \sqrt{\frac{3}{5}} & \frac{1}{2} + \mathrm{i}\sqrt{\frac{3}{20}} \\ -\frac{1}{2} + \mathrm{i}\sqrt{\frac{3}{20}} & \sqrt{\frac{3}{5}} \end{pmatrix} \tag{9.28}$$

$$B_{11} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

The matrices were found by using an optimization program that suggested certain numerical values. A closer examination of these values revealed the above analytical expressions.

9.6.2 No two-bit classical probabilistic protocol exists

Take the probability distribution π on the input strings x and y, defined by:

$$\pi(x,y) = \begin{cases} 0 & \text{iff } x = 00 \text{ or } y = 00 \\ \frac{1}{9} & \text{iff } x \neq 00 \text{ and } y \neq 00 \end{cases} \tag{9.29}$$

It is easily verified that for this distribution, every *deterministic* protocol with only two bits of communication will have a correctness ratio of at most $\frac{7}{9}$. Using Theorem 3.20 of [83], this shows that every possible randomized protocol with the same amount of communication will have a success ratio of at most $\frac{7}{9}$. (It can also be shown that this $\frac{7}{9}$ bound is tight but we will omit that proof here.) This implies that in order to reach the requested ration of $\frac{4}{5}$, at least three bits of communication are required if we are not allowed to use any prior entanglement.

9.6.3 Two qubits suffice without prior entanglement

A similar result also holds for qubit protocols without prior entanglement [112]. This can be seen by the fact that after Alice applied the rotation $A_{x_1x_2}$ and measured her qubit Q_A with the result $m_A = 0$, she knows the state of Bob's qubit Q_B exactly. It is therefore also possible to envision a protocol where the parties assume the measurement outcome $m_A = 0$ (this can be done without loss of generality), and for which Alice simply sends this qubit Q_B to Bob, after which Bob finishes the protocol in the same way as prescribed by the 'prior entanglement'-protocol. The protocol has thus become as follows.

First Alice and Bob decide by a random joint coin flip who is going to be the sender and the receiver in protocol. (Again we assume here that Alice is the sender.) Next, Alice (the sender) sends a qubit $|Q_{x_1x_2}\rangle$ (according to the input string x_1x_2 of Alice and the table 9.30) to the receiver Bob who continues the

protocol.

$$|Q_{00}\rangle = \sqrt{\tfrac{2}{5}}|0\rangle - \mathrm{i}\sqrt{\tfrac{3}{5}}|1\rangle \qquad |Q_{01}\rangle = \sqrt{\tfrac{4}{5}}|0\rangle + \left(\sqrt{\tfrac{3}{16}} + \mathrm{i}\sqrt{\tfrac{1}{80}}\right)|1\rangle$$
$$|Q_{10}\rangle = \sqrt{\tfrac{4}{5}}|0\rangle + \left(-\sqrt{\tfrac{3}{16}} + \mathrm{i}\sqrt{\tfrac{1}{80}}\right)|1\rangle \qquad |Q_{11}\rangle = \sqrt{\tfrac{1}{5}}|0\rangle - \mathrm{i}\sqrt{\tfrac{4}{5}}|1\rangle \tag{9.30}$$

If Bob has the input string $y_1y_2 = 00$, he concludes the protocol by sending a zero bit to Alice. In the other case, Bob applies the rotation $B_{y_1y_2}$ to the received qubit, measures the qubit in the standard basis, and sends this measurement outcome to Alice as the answer of the protocol. By doing so, the same correctness-probability of $\frac{4}{5}$ is reached for the *IP* function with two qubits of communication, whereas the classical setting requires 3 bits of communication as shown above.

9.7 Acknowledgments

We would like to thank Gilles Brassard, Harry Buhrman, Peter Høyer, and Tal Mor for their comments about this research. R.C. would like to thank the Laboratoire d'Informatique Théorique et Quantique, Université de Montréal for their gracious hospitality while this research was initiated. M.N. thanks the Office of Naval Research (Grant No. N00014-93-1-0116).

9.8 Appendix: capacity results for communication using qubits

In this appendix, we present results about the quantum resources required to transmit n classical bits between two parties when two-way communication is available. These results are used in the main text in the proof of the lower bound on the communication complexity of the inner product function, and may also be of independent interest.

Theorem 9.8.1 *Suppose that Alice possesses n bits of information, and wants to convey this information to Bob. Suppose that Alice and Bob possess no prior entanglement but qubit communication in either direction is allowed. Let n_{AB} be the number of qubits Alice sends to Bob, and n_{BA} the number of qubits Bob sends to Alice (n_{AB} and n_{BA} are natural numbers). Then, Bob can acquire the n bits if and only if the following inequalities are satisfied:*

$$n_{AB} \geq \lceil n/2 \rceil \tag{9.31}$$

$$n_{AB} + n_{BA} \geq n. \tag{9.32}$$

More generally, Bob can acquire m bits of mutual information with respect to Alice's n bits if and only if the above equations hold with m substituted for n.

Note that Theorem 9.2.1 follows from Theorem 9.8.1 because, if the communication from Bob to Alice is not counted then this can be used to set up an arbitrary entanglement at no cost.

Graphically, the capacity region for the above communication problem is shown in Fig. 9.1. Note the difference with the classical result for communication with bits, where the capacity region is given by the equation $n_{AB} \geq n$; that is, classically, communication from Bob to Alice does not help.

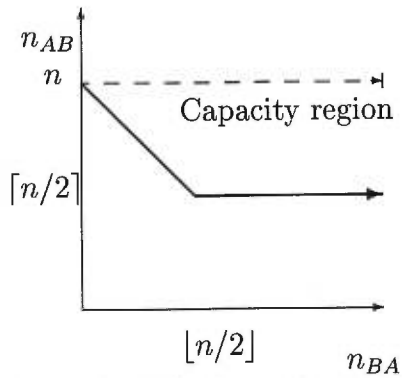

Figure 9.1: Capacity region to send n bits from Alice to Bob. n_{AB} is the number of qubits Alice sends to Bob, and n_{BA} is the number of qubits Bob sends to Alice. The dashed line indicates the bottom of the classical capacity region.

Proof of Theorem 9.8.1: The sufficiency of Eqns. (9.31) and (9.32) follows from the superdense coding technique [19]. The nontrivial case is where $n_{AB} < n$. Bob prepares $n - n_{AB} \leq n_{BA}$ EPR pairs and sends one qubit of each pair to Alice, who can use them in conjunction with sending $n - n_{AB} \leq n_{AB}$ qubits to Bob to transmit $2(n - n_{AB})$ bits to Bob. Alice uses her remaining allotment of $2n_{AB} - n$ qubits to transmit the remaining $2n_{AB} - n$ bits in the obvious way.

The proof that Eqns. (9.31) and (9.32) are necessary follows from an application of Holevo's Theorem [73], which we now review. Suppose that a classical information source produces a random variable X. Depending on the value, x, of X, a quantum state with density operator ρ_x is prepared. Suppose that a measurement is made on this quantum state in an effort to determine the value of X. This measurement results in an outcome Y. Holevo's theorem states that the mutual information $I(X : Y)$ between X and Y is bounded by the *Holevo bound* [73]

$$I(X : Y) \leq S(\rho) - \sum_x p_x S(\rho_x), \tag{9.33}$$

where p_x are the probabilities associated with the different values of X, $\rho = \sum_x p_x \rho_x$, and S is the von Neumann entropy function. The quantity on the right hand side of the Holevo bound is known as the *Holevo chi quantity*, $\chi(\rho_X) = S(\rho) - \sum_x p_x S(\rho_x)$.

Let X be Alice's n bits of information, which is uniformly distributed over $\{0,1\}^n$. Without loss of generality, it can be assumed that the protocol between Alice and Bob is of the following form. For any value $(x_1, \ldots, x_n)$ of X, Alice begins with a set of qubits in state $|x_1, \ldots, x_n\rangle|0, \ldots, 0\rangle$ and Bob begins with a set of qubits in state $|0, \ldots, 0\rangle$. The protocol first consists of a sequence of steps, where at each step one of the following processes takes place.

1. Alice performs a unitary operation on the qubits in her possession.

2. Bob performs a unitary operation on the qubits in his possession.

3. Alice sends a qubit to Bob.

4. Bob sends a qubit to Alice.

After these steps, Bob performs a measurement on the qubits in his possession, which has outcome Y. (Note that one might imagine that the initial states could be mixed and that measurements could be performed in addition to unitary operations; however, these processes can be simulated using standard techniques involving ancilla qubits.)

Let ρ_i^X be the density operator of the set of qubits that are in Bob's possession after i steps have been executed. Due to Holevo's Theorem, it suffices to upper bound the final value of $\chi(\rho_i^X)$—which is also bounded above by $S(\rho_i)$. We consider the evolution of $\chi(\rho_i^X)$ and $S(\rho_i)$. Initially, $\chi(\rho_0^X) = S(\rho_0) = 0$, since Bob begins in a state independent of X. Now, consider how $\chi(\rho_i^X)$ and $S(\rho_i)$ change for each of the four processes above.

1. This does not affect ρ_i^X and hence has no effect on $\chi(\rho_i^X)$ or $S(\rho_i)$.

2. It is easy to verify that χ and S are invariant under unitary transformations, so this does not affect $\chi(\rho_i^X)$ and $S(\rho_i)$ either.

3. Let B denote Bob's qubits after i steps and Q denote the qubit that Alice sends to Bob at the $i+1^{\text{st}}$ step. By the subadditivity inequality and the fact that, for a single qubit Q, $S(Q) \leq 1$, $S(BQ) \leq S(B) + S(Q) \leq S(B) + 1$. Also, by the Araki-Lieb inequality [2], $S(BQ) \geq S(B) - S(Q) \geq S(B) - 1$. It follows that $S(\rho_{i+1}) \leq S(\rho_i) + 1$ and

$$\begin{aligned} \chi(\rho_{i+1}^X) &= S(\rho_{i+1}) - \sum_{x \in \{0,1\}^n} p_x S(\rho_{i+1}^x) \\ &\leq (S(\rho_i) + 1) - \sum_{x \in \{0,1\}^n} p_x (S(\rho_x) - 1) \\ &= \chi(\rho_i^X) + 2. \end{aligned} \tag{9.34}$$

4. In this case, ρ_{i+1}^X is ρ_i^X with one qubit traced out. It is known that tracing out a subsystem of any quantum system does not increase χ [99], so

$\chi(\rho_{i+1}^X) \leq \chi(\rho_i^X)$. Note also that $S(\rho_{i+1}) \leq S(\rho_i) + 1$ for this process, by the Araki-Lieb inequality [2].

Now, since $\chi(\rho_i^X)$ can only increase when Alice sends a qubit to Bob and by at most 2, Eq. (9.31) follows. Also, since $S(\rho_i)$ can only increase when one party sends a qubit to the other and by at most 1, Eq. (9.32) follows. This completes the proof of Theorem 9.8.1. ■

Conclusion

En résumé, dans cette thèse constituée de cinq articles, nous avons obtenu les résultats suivants. Premièrement, notre analyse de l'algorithme de Grover nous a permis de le généraliser au cas où le nombre de solutions est différent de 1 et même au cas où le nombre de solutions est inconnu. Deuxièmement, nous avons découvert un nouvel algorithme quantique qui permet de compter le nombre de solutions à un problème[2] avec diverses précisions.

Les progrès en algorithmique quantique sont relativement lents ; il existe peu d'algorithmes quantiques. Plusieurs faits laissent croire que l'ordinateur quantique ne pourra pas résoudre des problèmes **NP-Dur** en temps polynomial. Ce qui laisse comme champ de manoeuvre les problèmes difficiles mais *non* **NP-Dur**. La situation est en réalité plus complexe puisque les classes **NP** et **BQP** sont pour le moment incomparables. Il est donc toujours concevable de résoudre à l'aide de l'ordinateur quantique des problèmes hors de **NP** ! En réalité, les espoirs se trouvent dans la classe (**NP****NP-Dur**)**P** des problèmes réputés difficiles mais pas **NP-Dur**. Dans cette catégorie, on retrouve en plus de factorisation le problème d'isomorphisme de graphe, le problème de treillis, et certains problèmes en théorie des codes. Les problèmes avec promesse comme le problème de Simon sont à notre avis un autre groupe de problèmes prometteurs.

Troisièmement, dans le domaine naissant de la complexité de communication quantique, nous avons démontré que l'intrication peut permettre une réduction

[2]Évaluer $|\{x|F(X) = 1\}$ pour un F donné.

non constante de communication. Quatrièmement, nous avons démontré qu'il existe une fonction pour laquelle l'intrication n'est d'aucune utilité.

Du côté de la complexité de communication aussi, plusieurs questions restent sans réponse. Une des questions qui nous semblent importantes consiste à caractériser parfaitement la relation entre les mesures C^* et Q. Clairement, du fait de l'existence de la téléportation quantique, $C^* \leq 2Q$. Pour le moment, dans tous les exemples utilisant de l'intrication, il y a autant de paires intriquées que d'information classique transmise. Ce qui revient à conjecturer $2C^* \geq Q$. Il serait intéressant de trouver un contre-exemple à cette relation ou sinon d'en démontrer la validité.

Finalement, nous avons démontré qu'une parfaite simulation de n paires intriquées nécessitait $2^{\Omega}(n)$ bits de communication classique et que dans le cas d'une paire intriqué 8 bits sont toujours suffisants. Il serait bien entendu intéressant de savoir combien de bits de communication sont suffisants pour simuler n paires intriquées.

Bibliography

[1] A. Ambainis, L. J. Schulman, A. Ta-Shma, U. Vazirani et A. Wigderson. The quantum communication complexity of sampling. *Proceeding of the 39th Annual Symposium on Foundations of Computer Science*, 1998.

[2] H. Araki et E. H. Lieb. Entropy inequalities. *Commun. Math. Phys.*, 18:160–170, 1970.

[3] A. Aspect, P. Grangier et G. Roger. Experimental realization of Einstein-Podolsky-Rosen-Bohm gedankenexperiment; a new violation of Bell's inequalities. *Physical Review Letters*, 49(2):91, 1982.

[4] A. Barenco. A universal two-bit gate for quantum computation. *Proceedings of the Royal Society of London, Series A*, volume 449, pages 679–683, 1995. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9505016`.

[5] A. Barenco, C H. Bennett, R. Cleve, D. P. DiVincenzo, N H. Margolus, P. W. Shor, T. Sleator, J. A. Smolin et H. Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52(5):3457–3467, 1995. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9503016`.

[6] A. Barenco et A. K. Ekert. Quantum computation. *Acta Physica Slovaca*, 45(3):205–216, 1995. Proceedings of the 3rd central-european workshop on Quantum Optics.

[7] D. A. Barrington. Bounded width branching programs. Rapport technique MIT-LCS//MIT/LCS/TR-361, Massachusetts Institute of Technology, Laboratory for Computer Science, juin 1986.

[8] D. A. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *Proceeding of the 18th ACM Symposium on the Theory of Computing*, pages 1–5, mai 1986.

[9] D. A. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *Journal of Computer et System Sciences*, 38(1):150–164, février 1989.

[10] R. Beals, H. Buhrman, R. Cleve, M. Mosca et R. de Wolf. Quantum lower bounds by polynomials. *Proceedings of 39thAnnual Symposium on Foundations of Computer Science*, pages 352–361, novembre 1998.

[11] J. Bell. On the Einstein Podolsky Rosen paradox. *Physics*, 1(3):195–200, 1964.

[12] M. Ben-Or et R. Cleve. Computing algebraic formulas with a constant nubmer of registers. *SIAM Journal on Computing*, 21:54–58, février 1992.

[13] C. H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 6:525–532, 1973.

[14] C. H. Bennett. Notes on the history of reversible computation. *IBM Journal of Research and Development*, 32(1):281–288, 1988.

[15] C. H. Bennett. Time/space trade-offs for reversible computation. *SIAM Journal on Computing*, 18(4):766–776, 1989.

[16] C. H. Bennett, E. Bernstein, G. Brassard et U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.

[17] C. H. Bennett, G. Brassard, C. Crépeau, A. Peres R. Jozsa et W. Wootters. Teleporting an unknown quantum state by dual classical and Einstein Podolsky Rosen channels. *Physical Review Letter*, 70:1895–1898, 1993.

[18] C. H. Bennett et R. Landauer. The fundamental physical limits of computation. *Scientific American*, 253:48–56, 1985.

[19] C. H. Bennett et S. J. Wiesner. Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states. *Physical Review Letter*, 69(20):2881–2884, 1992.

[20] E. Bernstein et U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.

[21] A. Berthiaume. *Complexity Theory Retrospective II*, chapitre, Quantum Computation. Springer-Verlag, 1996.

[22] A. Berthiaume et G. Brassard. Oracle quantum computing. *Journal of Modern Optics*, 41(12):2521–2535, 1994.

[23] D. Bohm. *Quantum Theory*, page 614. Prentice-Hall, 1951.

[24] M. Boyer, G. Brassard, P. Høyer et A. Tapp. Tight bounds on quantum searching. *Proceedings of the Fourth Workshop on Physics of Computation*, pages 36–43, Boston, novembre 1996. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9605034`.

[25] Michel Boyer, Gilles Brassard, Peter Høyer et Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46:493–505, 1998. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9605034`.

[26] G. Brassard. Cryptology column – quantum computing: The end of classical cryptography ? *ACM SIGACT News*, 25(4):15–21, décembre 1994.

[27] G. Brassard. New trends in quantum computing. *Proceedings of the 13th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*,

volume 1046 de *Lecture Notes in Computer Science*, pages 3–10, Grenoble, France, février 1996. Springer.

[28] G. Brassard. Searching a quantum phone book. *Science*, 275:627–628, Janvier 1997.

[29] G. Brassard. *Quantum Information Processing for Computer Scientists.* MIT Press, 2000. À paraître.

[30] G. Brassard, R. Cleve et A. Tapp. Cost of exactly simulating quantum entanglement with classical communication. *Physical Review Letter*, 83(9):1874–1878, août 1999. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9901035` .

[31] G. Brassard, R. Cleve et A. Tapp. The cost of exactly simulating quantum entanglement with classical communication. Rapport technique 1160, Université de Montréal, 1999. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9901035` .

[32] G. Brassard et C. Crépeau. 25 years of quantum cryptography. *ACM SIGACT News*, 27(3):13–24, septembre 1996.

[33] G. Brassard et P. Høyer. An exact quantum polynomial-time algorithm for Simon's problem. *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems (ISTCS'97)*, pages 12–23. IEEE Computer Society Press, juin 1997.

[34] G. Brassard, P. Høyer, M. Mosca et A. Tapp. Quantum amplitude amplification and estimation. En préparation.

[35] G. Brassard, P. Høyer et A. Tapp. Quantum counting. *25th Annual International Colloquium on Automata, Languages and Programming (ICALP'98)*, pages 820–831, Aalborg, Danemark, 1998.

[36] Gilles Brassard. Teleportation as a quantum computation. *Proceedings of the Fourth Workshop on Physics of Computation*, pages 48–50, 1996.

[37] Gilles Brassard. Teleportation as a quantum computation. *Physica*, D120:43, 1998. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9605035`.

[38] Gilles Brassard, Peter Høyer et Alain Tapp. Cryptology column — quantum cryptanalysis of hash and claw-free functions. *ACM SIGACT News*, 28:14–19, 1997. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9705002`.

[39] H. Buhrman, R. Cleve et W. van Dam. Quantum entanglement and communication complexity, 1997. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9705033`.

[40] H. Buhrman, R. Cleve et A. Wigderson. Quantum vs. classical communication and computation. *Proceedings of the 30th ACM Symposium on the Theory of Computing*, pages 63–68, mai 1998.

[41] H. Buhrman, W. van Dam, P. Høyer et A. Tapp. Multiparty quantum communication complexity. *Physical Review A*, 60(3), septembre 1999. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9710054`.

[42] A. R. Calderbank et P. W. Shor. Good quantum error-correcting codes exist. Manuscrit, décembre 1995. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9512032`.

[43] D.-P. Chi et J. Kim. Quantum database searching by a single query. *Quantum computing and quantum communications: First NASA International Conference, QCQC'98:selected papers*, volume 1509 de *Lecture Notes in Computer Science*. Springer-Verlag, février 1999.

[44] B. Chor et O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.

[45] I. L. Chuang, R. Laflamme, P. W. Shor et W. H. Zurek. Quantum computers, factoring and decoherence. *Science*, 270:1635–1637, 1995. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9503007`.

[46] J. F. Clauser, M. A. Horne, A. Shimony et R. A. Holt. Proposed experiment to test local hidden-variable theories. *Physical Review Letter*, 23:880–884, 1969.

[47] R. Cleve. An introduction to quantum complexity theory, 1999. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9705033`.

[48] R. Cleve et H. Buhrman. Substituting quantum entanglement for communication. *Physical Review A*, 56(2):1201–1204, 1997. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9704026`.

[49] R. Cleve, A. Ekert, C. Macchiavello et M. Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society of London*, volume A454, pages 339–354, 1998. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9708016`.

[50] R. Cleve, van Dam W, M. Nielsen et A. Tapp. Quantum entanglement and the communication complexity of the inner product function. *Quantum computing and quantum communications: First NASA International Conference, QCQC'98:selected papers*, volume 1509 de *Lecture Notes in Computer Science*. Springer-Verlag, février 1999. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9708019`.

[51] S. A. Cook. The complexity of theorem-proving procedures. *3th ACM Symposium on the Theory of Computing*, pages 151–158, 1971.

[52] D. Coppersmith. An approximate Fourier transform useful in quantum factoring. Rapport technique RC 19642, IBM, 1994.

[53] D. Coppersmith et E. Grossman. Generators for certain alternating groups with applications to cryptography. *SIAM Journal on Applied Mathematic*, 29(4):624–627, décembre 1975.

[54] T. M. Cover et J. A. Thomas. *Elements of Information Theory.* Wiley Series in Telecommunications. John Wiley & Sons, New York, NY, USA, 1991.

[55] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London, Series A*, A400:97–117, 1985.

[56] D. Deutsch. Quantum computational networks. *Proceedings of the Royal Society of London, Series A*, A425:73–90, 1989.

[57] D. Deutsch, A. Barenco et A. K. Ekert. Universality in quantum computation. *Proceedings of the Royal Society of London, Series A*, 449:669–677, 1995. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9508012`.

[58] D. Deutsch et R. Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London, Series A*, A439:553–558, 1992.

[59] D. P. DiVincenzo. Two-bit gates are universal for quantum computation. *Physical Review A*, 51(2):1015–1022, février 1995. Disponible à `http://xxx.lanl.gov/abs/cond-mat/9407022`.

[60] C. Dürr et P. Høyer. A Quantum Algorithm for Finding the Minimum. Manuscrit. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9607014`.

[61] A. Einstein, B. Podolsky et N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical Review*, 41:777–780, 1935.

[62] A. K. Ekert et R. Jozsa. Shor's quantum algorithm for factorising numbers. *Review of Modern Physics*, 68(3):733–753, juillet 1996.

[63] R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6):467–488, 1982.

[64] P. Frankl et V. Rödl. Forbidden intersections. *Transactions of the American Mathematical Society*, 300(1):259–286, 1987.

[65] E. F. Fredkin et T. Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21(3/4):219–253, 1982.

[66] M. R. Garey et D. S. Johnson. *Computers and Intractibility, a Guide to the Theory of NP-Completeness.* W.H. Freeman and Co., San Francisco, 1979.

[67] D. M. Greenberger, M. Horne et A. Zeilinger. Going beyond bell's theorem. M. Kafatos, *Bell's Theorem, Quantum Theory et Conception of the Universe*, pages 69–72. Kluwer Academic, 1989.

[68] L. K. Grover. Quantum telecomputation. Disponible à `http://xxx.lanl.gov/abs/cond-mat/9704012`.

[69] L. K. Grover. A fast quantum mechanical algorithm for database search. *Proceedings of the 28th ACM Symposium on the Theory of Computing*, pages 212–219, mai 1996.

[70] L. K. Grover. A fast quantum mechanical algorithm for estimating the median. Rapport technique ITD-96-30115J, Bell Labs, 1997. Disponible à `http://xxx.lanl.gov/abs/cond-mat/9607024`.

[71] L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, 79(2):325–328, juillet 1997.

[72] L. K. Grover. Quantum computers can search rapidly by using almost any transformation. *Physical Review Letters*, 80:4329–4332, mai 1998.

[73] A. S. Holevo. Bounds for the quantity of information transmitted by a quantum communication channel. *Problems of Information Transmission*, 9:177–183, 1973. Traduction anglaise.

[74] P. Høyer. Efficient quantum transforms. Manuscrit. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9702028`.

[75] P. Høyer. Conjugated operators in quantum algorithms. *Physical Review A*, 59:3280–3289, mai 1999.

[76] P. Høyer et C. Dürr. A Quantum Algorithm for Finding the Minimum. Manuscrit, juillet 1996. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9607014`.

[77] R. I. G. Hughes. *The Structure and Interpretation of Quantum Mechanics.* Harvard University Press, 1992.

[78] R. J. Hughes, D. F. V. James, J. J. Gomez, M. S. Gulley, M. H. Holzscheiter, P. G. Kwiat, S. K. Lamoreaux, C. G. Peterson, V. D. Sandberg, M. M. Schauer, C. M. Simmons, C. E. Thorbum, D. Tupa, P. Z. Wang et A. G. White. The los alamos trapped ion quantum computer experiment. *Fortschritte der Physik*, 46(4-5):329–361, 1998.

[79] J. A. Jones et M. Mosca. Approximate quantum counting on an NMR ensemble quantum computer. *Physical Review Letter*, 83:1050, 1999. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9808056`.

[80] J. A. Jones, M. Moscaand et R. H. Hansen. Implementation of a quantum search algorithm on a quantum computer. *Nature*, 393:344–346, 1998. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9805069`.

[81] A. Yu. Kitaev. Quantum measurements and the abelian stabilizer problem, 1995. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9511026`.

[82] I. Kremer. Quantum communication. Master's thesis, The Hebrew University of Jerusalem, 1995.

[83] E. Kushilevitz et N. Nisan. *Communication Complexity.* Cambridge University Press, 1997.

[84] R. Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5:183–191, 1961.

[85] K.-J. Lange, P. McKenzie et A. Tapp. Reversible space equals deterministic space. *Journal of Computer and System Sciences*, 1999. À paraître.

[86] Klaus-Jörn Lange, Pierre McKenzie et Alain Tapp. Reversible space equals deterministic space. *12th Annual IEEE Conference on Computational Complexity (CCC'97)*, pages 45–50, Ulm, Allemagne, 1997.

[87] H. S. Leff et A. F. Rex. *Maxwell's Demon: Entropy, Information, Computing.* Princeton University Press, Princeton, N. J., 1990.

[88] R Y. Levine et A T. Sherman. A note on Bennett's time-space tradeoff for reversible computation. *SIAM Journal on Computing*, 19(4):673–677, 1990.

[89] M. Li et P. M. B. Vitányi. Reversibility and adiabatic computation: trading time and space for energy. *Proceedings of the Royal Society of London, Series A*, 452:1–21, 1996.

[90] H.B. Mann. *Addition theorems: The addition theorems of group theory and number theory*, page 6. R.E. Krieger Pub. Co., New York, édition John Wiley & Sons, 1965.

[91] J. C. Maxwell. *Theory of Heat*, chapter 12. Longmans, Green and co, London, 1871.

[92] N. D. Mermin. Quantum misteries revisited. *American Journal of Physics*, 58(8):731–734, 1990.

[93] C. Miquel, J. P. Paz et R. Perazzo. Factoring in a dissipative quantum computer. Manuscrit, janvier 1996. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9601021`.

[94] A. Nayak et F. Wu. The quantum query complexity of approximating the median and related statistics, 1998. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9804066`.

[95] National Bureau of Standards. Data encryption standard. Rapport technique, fips pub 46, Federal Information Processing Standard, U. S. Department of Commerce, Washington, DC, 1977.

[96] A. Peres. *Quantum Theory : Concepts and methods.* Kluwer Academic Publishers, 1993.

[97] R. Raz. Exponential separation of quantum and classical communication complexity. *31th ACM Symposium on the Theory of Computing*, 1999.

[98] B. Schumacher. Quantum coding. *Physical Review A*, 51, 1995.

[99] B. Schumacher, M. Westmoreland et W. K. Wootters. Limitation on the amount of accessible information in a quantum channel. *Physical Review Letter*, 76:3453–3456, 1996.

[100] P. W. Shor. Algorithms for quantum computation: Discrete log and factoring. *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.

[101] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, octobre 1997.

[102] D. R. Simon. On the power of quantum computation. *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 116–123, 1994.

[103] T. Sleator et H. Weinfurter. Realizable universal quantum logic gates. *Physical Review Letters*, 74:4087–4090, 1995.

[104] A. Steane. Multiple particle interference and quantum error correction. *Proceedings of the Royal Society of London Series A*, 1996. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9601029`.

[105] M. Steiner. Towards quantifying non-local information transfer: finite-bit non-locality, 1999. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9902014`.

[106] K. Svozil. Quantum computation and complexity theory, décembre 1994. Disponible à `http://xxx.lanl.gov/abs/hep-th/9412047`.

[107] L. Szilard. On the decrease of entropy in a thermodynamic system by the intervention of intelligent beings. *Z. Physik*, 53:840–856, 1929.

[108] B.M. Terhal et J.A. Smolin. Superfast quantum algorithms for coin weighing and binary search problems, 1997. Disponible à `http://xxx.lanl.gov/abs/quant-ph/9705041`.

[109] T. Toffoli. Reversible computing. Rapport technique MIT/LCS/TM-151, MIT LCS, février 1980.

[110] M. v. Smoluchowski. Experimentell nachweisbare der üblichen thermodynamik widersprechende molekularphänomene. *Z. Physik*, 13:1069–1080, 1912.

[111] A.C. Yao. Some complexity questions related to distributed computing. *Proceedings of the 11th ACM Symposium on the Theory of Computing*, pages 209–213, 1979.

[112] A.C. Yao. Quantum circuit complexity. *Proceedings of the 34th ACM Symposium on the Theory of Computing*, pages 352–361, 1993.