Quantenalgorithmen für Graphen und Algebraprobleme

Sebastian Dörn

Institut für Theoretische Informatik
Universität Ulm
sebastian.doern@uni-ulm.de

Abstract: Die Entwicklung von Algorithmen für Quantencomputer hat sich in den letzten Jahren zu einem rasant wachsenden Forschungsgebiet in der Informatik und Physik entwickelt. Quantenalgorithmen können eine große Zahl von Problemen schneller lösen, als die bisher besten bekannten klassischen Verfahren. In unserer Arbeit konstruieren wir Quantenalgorithmen für grundlegende Probleme aus der Graphentheorie und Algebra, welche polynomial schneller sind, als die besten bekannten klassischen Verfahren. Für einige unserer Algorithmen können wir außerdem noch zeigen, dass diese optimal sind.

1 Einführung

Quantum Computing ist ein neues interdisziplinäres Forschungsgebiet zwischen Theoretischer Informatik und Quantenphysik. Ein Quantencomputer ist ein Computer, der die Gesetze der Quantenmechanik anstelle von elektrischen Schaltungen nutzt, wie sie bei den heutigen Rechnern üblich sind. Quantencomputer erlauben im Gegensatz zu klassischen Computern das Rechnen mit Quantenbits, die einer Superposition von Nullen und Einsen entsprechen. Mittels dieses Prinzips können Quantencomputer Quantensysteme simulieren, für die es auf klassischen Rechnern keine effizienten Verfahren gibt.
Die Idee, Gesetze der Quantenmechanik für den Bau von Computern zu verwenden, geht auf den Physiker und Nobelpreisträger Richard Feynman zurück. Er schlug 1982 für die aufwändigen Rechnungen mit Elementarteilchen-Modellen vor, eigens dafür konzipierte Quantenrechner zu verwenden.
Im Jahre 1994 fand Peter Shor einen grundlegenden Quantenalgorithmus zur Faktorisierung von großen Zahlen. Für die Primfaktorzerlegung einer 300-stelligen Zahl benötigt ein klassischer Algorithmus rund 5×10^{24} Rechenschritte, was 150000 Jahre entsprechen würde (10^{12} Schritte pro Sekunde). Der Quantenalgorithmus von Shor benötigt hingegen nur 5×10^{10} Schritte oder einen Sekundenbruchteil. Zwei Jahre später entwickelte Lov Grover einen Quantenalgorithmus, mit welchem Suchprobleme auf einem Quantencomputer quadratisch schneller gelöst werden können, als auf einem herkömmlichen Computer. Dieses Verfahren hat zentrale Bedeutung für die Konstruktion von Quantenalgorithmen für zahlreiche Optimierungsprobleme.

In dieser Arbeit entwickeln wir neue Quantenalgorithmen für Probleme aus der Graphentheorie und Algebra. Unsere Quantenalgorithmen sind polynomial schneller, als die bisher besten bekannten klassischen Verfahren. Es gibt verschiedene Gründe für die Untersuchung von Quantenalgorithmen für Graphen- und Algebraprobleme. Einerseits sind die von uns untersuchten Probleme grundlegender Natur. Insbesondere Graphenalgorithmen besitzen zahlreiche praktische Anwendungen in vielen wichtigen Optimierungsverfahren. Andererseits können wir auch an Hand der untersuchten Probleme die Mächtigkeit unserer Methoden für die Konstruktion von Schranken für die Laufzeit der Quantenalgorithmen untersuchen. Für zahlreiche algorithmische Probleme haben wir optimale Quantenalgorithmen aus einer Kombination von verschiedenen Quantensuchverfahren gefunden. Für einige dieser Fragestellung scheint dies nicht möglich zu sein. Vielleicht ist dies eine Motivation für die Entwicklung neuer Techniken im Quantum Computing.

Quantenalgorithmen für Graphenprobleme

Einige der ersten, die Quantenalgorithmen für Graphenprobleme untersucht haben, waren Dürr et al. [2] im Jahre 2004. Sie konstruierten nahezu optimale Quantenalgorithmen (bis auf log-Faktoren) für die Bestimmung von minimalen aufspannenden Bäumen, testen ob ein Graph zusammenhängend ist oder auch für die Berechnung von kürzesten Wegen in Graphen. Magniez et al. [10] fanden im Jahre 2005 einen interessanten Quantenalgorithmus, basierend auf Quanten Random Walks, für das Suchen von Dreiecken in Graphen. Aufbauend auf diesen bekannten Quantenalgorithmen untersuchen wir weitere wichtige Probleme der Graphentheorie. Wir entwickeln Quantenalgorithmen für Matching Probleme in ungewichteten und gewichteten Graphen. Wir zeigen, dass auf einem Quantencomputer ein maximales Matching in einem ungewichteten Graphen polynomial schneller berechnet werden kann. Dieses Ergebnis verbessert auch einen Quantenalgorithmus von Ambainis and Špalek [1]. Weiterhin konstruieren wir Quantenalgorithmen für die Bestimmung von unabhängigen Knotenmengen und für Rundreiseprobleme in gerichteten Graphen. Für diese Aufgabenstellungen zeigen wir untere und obere Schranken für die Laufzeit der Quantenalgorithmen. Wir beweisen beispielsweise, dass unsere Algorithmen für die Bestimmung einer maximalen unabhängigen Knotenmenge oder für das Entscheidungsproblem ob ein Graph einen Eulerkreis besitzt, optimal sind.

Quantenalgorithmen für Algebraprobleme

In der Algebra entwickeln wir Quantenalgorithmen für das Testen von algebraischen Eigenschaften. Magniez und Nayak [8] fanden 2005 einen optimalen Quantenalgorithmus zum Testen, ob eine Gruppe kommutativ ist. Für unsere untersuchten Probleme betrachten wir als Eingabe eine Menge von Elementen zusammen mit einer binären Operation, welche uns als Tabelle gegeben ist. Die Aufgabe ist es nun zu testen, ob diese Operationstabelle algebraische Eigenschaften, wie beispielsweise die Assoziativität, erfüllt. Für viele dieser algebraischen Probleme beweisen wir nichttriviale untere Schranken für die Quantenkomplexität. Weiterhin stellen wir die erste Anwendung der neuen Quanten Random Walk Technik von Magniez et al. [9] vor.

In unserer Arbeit betrachten wir auch zahlreiche relevante Entscheidungsprobleme der linearen Algebra. Im Matrix Potenz Problem haben wir zwei Matrizen A, B und eine positive Zahl m gegeben. Zu entscheiden ist, ob die m'th Potenz von A die Matrix B ist. Wir betrachten weiterhin das Entscheidungsproblem, ob die Inverse von A, die Matrix B ist. Ebenfalls interessieren wir uns, ob eine gegebene Matrix singulär ist, also die Determinante gleich null ist. Für alle diese bekannten Probleme aus der linearen Algebra zeigen wir, dass mittels Quantum Computing keine Beschleunigung gegenüber klassischen Algorithmen zu erreichen ist.

2 Wie Quantencomputer rechnen

Die Quanteninformatik beginnt mit der Verallgemeinerung der fundamentalen Träger klassischer Information zu so genannten Quantenbits oder kurz Qubits. Ein Qubit ist nicht notwendiger Weise in dem Zustand null oder eins, wie ein klassisches Bit. In der Quantenmechanik kann jedes Qubit durch Superpositionen von 0 und 1 weitere Zustände annehmen, welche als Quantenzustände bezeichnet werden. Ein allgemeiner Zustand $|\psi\rangle$ eines Qubits ist ein Einheitsvektor $\alpha_0|0\rangle + \alpha_1|1\rangle$ im zweidimensionalen Raum, wobei α_0, α_1 zwei komplexe Zahlen mit der Eigenschaft $|\alpha_0|^2+|\alpha_1|^2 = 1$ sind. Dieses Konzept kann zu mehreren Qubits verallgemeinert werden, welche als Quantenregister bezeichnet werden. Ein Zustand eines Quantenregisters der Größe n kann geschrieben werden als

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle \quad \text{mit} \quad \sum_{x\in\{0,1\}^n} |\alpha_x|^2 = 1.$$

Hierin erkennen wir, dass die Größe des Berechnungsraumes eines Quantenregisters exponential in der Anzahl der Qubits ist. Auf einem Quantenregister werden nun zwei grundlegende Operationen angewandt: unitäre Transformationen und Messungen. Die Quantenmechanik verlangt, dass die Entwicklung von Quantenzustände mittels linearer und unitärer Operationen beschrieben wird. Um dann ein Ergebnis einer solchen Berechnung zu bekommen, muss am Ende der Quantenzustand gemessen werden. Nach einer Messung des Quantenzustandes erhalten wir den Wert $|x\rangle$ mit Wahrscheinlichkeit $|\alpha_x|^2$. Die Messung zerstört den Superpositionszustand und man erhält genau einen der Basiszustände $|x\rangle$ als Ergebnis.

Quantenparallelismus. Nun wollen wir die Frage beantworten, was die Leistungsfähigkeit der Quantenberechnung ausmacht. Der Hauptgrund ist der Quantenparallelismus, der von David Deutsch im Jahre 1985 entdeckt wurde und das fundamentale Prinzip aller Quantenalgorithmen ist. Mittels eines Quantenregister, bestehend aus zwei Qubits und einer einfachen unitäre Operation, können wir in einem Schritt den folgenden Quantenzustand erzeugen:

$$\frac{1}{\sqrt{2}}|0, f(0)\rangle + \frac{1}{\sqrt{2}}|1, f(1)\rangle\,, \text{ wobei } f : \{0,1\} \to \{0,1\}.$$

Aus diesem Quantenzustand erkennen wir, dass man die Funktion f für beide Eingabewerte $x = 0$ und $x = 1$ in einem Schritt berechnen kann.

Im Allgemeinen gilt, dass eine Funktion auf einen $n+1$-Qubit Quantencomputer alle 2^n Eingabewerte in einem Schritten parallel auswerten kann. Die Zahl der parallelen Funktionsberechnungen steigt also exponentiell mit der Zahl der Qubits.
Unglücklicherweise ist der Quantenparallelismus in dieser Form nicht verwendbar. Die Messung löscht die gesamte im Qubit enthaltene Information bis auf ein Bit, welches gemessen wird, aus. Nötig sind daher zusätzlich Algorithmen in Form von gültigen Transformationen, sodass wir nach einer Messung des Quantenzustandes die Lösung unseres algorithmischen Problems mit hoher Wahrscheinlichkeit erhalten.

Quanten Query Modell. Viele Quantenalgorithmen werden für das sogenannte Black Box oder Query Modell entwickelt. Im Query Modell ist die Eingabe $x_1, \ldots, x_N$ in einer Black Box enthalten und wir können diese mittels Fragen an die Black Box auslesen. Für ein Frage geben wir den Index i als Eingabe an die Black Box und erhalten x_i als Ausgabe. Das Ziel ist die Berechnung einer Booleschen Funktion f mit den Eingabebits $x = (x_1, \ldots, x_N)$ durch eine minimale Anzahl von Fragen an die Black Box.
Im Quanten Query Modell stellen wir auch Fragen an die Black Box, aber im Gegensatz zum klassischen Fall kann man mittels des Quantenparallelismus Fragen in Superposition stellen:

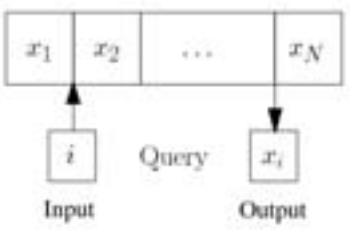

Abbildung 1: Black Box

$$\frac{1}{\sqrt{N}} \sum_{i=1}^{N} |i, 0\rangle \stackrel{O_x}{\to} \frac{1}{\sqrt{N}} \sum_{i=1}^{N} |i, x_i\rangle \text{ mit Query } O_x : |i, j\rangle \to |i, j \oplus x_i\rangle.$$

Eine Quantenberechnung mit T Queries ist eine alternierende Folge von T Query Transformation O_x zusammen mit unitären Operationen, welche nicht von der Eingabe x abhängen. Das Ergebnis unserer Berechnung erhalten wir dann durch Messung des resultierenden Quantenzustandes.
Alle unsere Quantenalgorithmen in dieser Arbeit haben eine konstante Fehlerwahrscheinlichkeit. Wir unterscheiden hierbei zwei Komplexitätsmaße: die Query- und die Zeitkomplexität. Die Query Komplexität unseres Algorithmus ist die Anzahl der Fragen an die Black Box, also die Anzahl der Transformationen O_x. Die Zeitkomplexität wird in der Schaltkreisgröße der unitären Operationen angegeben. An der Query Komplexität können wir die Beschleunigung gegenüber klassischen Algorithmen bestimmen. Außerdem gibt es leistungsfähige Methoden, um untere Schranken hierfür zu berechnen. Da die Zeitkomplexität nach Definition mindestens so groß ist wie die Query Komplexität, gelten diese unteren Schranken auch für die Laufzeit unserer Quantenalgorithmen.

2.1 Grundlegende Quantensuchalgorithmen

Im Folgenden wollen wir nun einen kurzen überblick über die drei wichtigsten Methoden zur Konstruktion unserer Quantenalgorithmen geben.

Groversuche. Grover's Suchalgorithmus [7] ist einer der grundlegendsten Quantenalgorithmen. Nehmen wir an, dass wir ein Suchproblem mit N Elementen und k Lösungen haben. Grover hat gezeigt, dass die erwartete Quanten Query Komplexität für die Bestimmung einer Lösung $O(\sqrt{N/k})$ und für die Bestimmung aller Lösungen $O(\sqrt{k \cdot N})$ ist. Der Grover Algorithmus ist somit quadratisch schneller, als die klassische Suche in einer unsortierten Datenbasis.

Amplituden Amplifikation. Die klassische Amplituden Amplifikation ist ein bekanntes Prinzip in der Algorithmentheorie. Sei $\mathcal{A}$ ein Algorithmus der mit Wahrscheinlichkeit ϵ eine Lösung eines algorithmischen Problems findet. Dann müssen wir $\mathcal{A}$ genau $\Theta(1/\epsilon)$ wiederholen, um die Erfolgswahrscheinlichkeit auf beispielsweise 2/3 zu verbessern. Die Quanten Amplituden Amplifikation ist eine Verallgemeinerung von Grover's Suchalgorithmus. Hier benötigen wir nur $O(\frac{1}{\sqrt{\epsilon}})$ Wiederholungen von $\mathcal{A}$, um eine Lösung zu finden.

Quanten Walk. Quanten Walks sind das Analogon zu Random Walks. Ein Random Walk ist eine zufällige Irrfahrt durch einen Graphen, wobei wir den nächsten Knoten der Irrfahrt immer zufällig unter allen Nachbarknoten des aktuellen Knoten wählen. Im Quanten Walk können wir im Gegensatz zum Random Walk mittels Quantenparallelismus von einem Knoten gleichzeitig zu allen Nachbarknoten gehen. Für die Anwendung des Quanten Walk gibt es verschiedene Suchverfahren basierend auf Markov Ketten. Der Quanten Walk ist eine Quelle für viele neue Quantenalgorithmen, wie beispielsweise die Suche nach Dreiecken in Graphen [10].

3 Quantenalgorithmen für Graphenprobleme

Zahlreiche bekannte Graphenalgorithmen verwenden verschiedene Suchroutinen um Kanten in Graphen zu finden. In diesem Abschnitt wollen wir einige Graphenalgorithmen untersuchen, die wir mittels der im vorigen Abschnitt vorgestellten Quantensuchverfahren beschleunigen können. Zunächst definieren wir zwei Modelle, um Graphen zu repräsentieren. Zur Vereinfachung untersuchen wir hier nur ungerichtete und ungewichtete Graphen $G = (V, E)$ mit n Knoten und m Kanten.

- *Matrix Modell:* Gegeben ist die Adjazenzmatrix $A \in \{0,1\}^{n \times n}$ von G mit $A_{i,j} = 1$ gdw. $\{i, j\} \in E$.
- *Array Modell:* Gegeben sind die Knotengrade $d_1, \ldots, d_n$ von G und für jeden Knoten $i \in V$ ein Array mit seinen Nachbarn $f_i : \{1, \ldots, d_i\} \to \{1, \ldots, n\}$. Der Wert $f_i(j)$ ist der j-th Nachbar von Knoten i.

Wir interessieren uns nun für die minimale Anzahl der Fragen an das Matrix bzw. Array Modell um ein gegebenes Graphenproblem zu lösen. Im klassischen Fall benötigen wir dazu beispielsweise im Matrix Modell in den meisten Fällen $\Omega(n^2)$ Fragen an die Adjazenzmatrix.

3.1 Tiefen- und Breitensuche

Zunächst wollen wir ein wenig Intuition vermitteln, wie man mittels Grovers Quantensuchalgorithmus Kanten in Graphen schneller finden kann. Die Existenz einer Kante $\{u, v\}$ in einem Graphen kann im klassischen, wie im Quanten Fall, mit einer Frage an die Adjazenzmatrix festgestellt werden. Im Array Modell hingegen kann man diese Aufgabe von $O(\min\{d_u, d_v\})$ mittels einer einfachen Groversuche auf $O(\sqrt{\min\{d_u, d_v\}})$ verbessern. Falls wir uns für alle Nachbarn des Knotens v interessieren, so können wir diese in $O(\sqrt{n \cdot d_v})$ Fragen an die Adjazenzmatrix bestimmen. Alle Nachbarn von v mit einer speziellen Eigenschaft, können wir in $O(\sqrt{d_v \cdot a_v})$ Fragen an das Array Modell finden, wobei a_v die Anzahl der Nachbarknoten von v mit dieser Eigenschaft sind.

Eine weitere wichtige Anwendung der Groversuche ist die Tiefen- und Breitensuche, welche Bestandteil von vielen bekannte Graphenalgorithmen ist. Im klassischen Fall haben diese beiden Verfahren lineare Laufzeit in der Anzahl der Kanten des Graphen. Wir können nun unsere Quantensuche nach Kanten in Graphen verwenden, um diese Suchroutinen zu beschleunigen. Mit Hilfe des Grover Algorithmus kann man zeigen, dass die Tiefen- und Breitensuche auf $O(n^{1.5} \log n)$ im Matrix Modell und $O(\sqrt{nm} \log n)$ im Array Modell verbessert werden kann.

3.2 Unabhängige Knotenmengen

Wir betrachten nun folgendes Terminplanungsproblem, welche zahlreiche Anwendung in der Wirtschaft und Technik besitzt: Wir haben eine Menge von Aufträgen, die ausgeführt werden sollen. Jeder Auftrag benötigt für die Ausführung eine Menge von Ressourcen. Wir nehmen an, dass zwei Aufträge nur dann gleichzeitig ausgeführt werden können, wenn sie nicht die gleichen Ressourcen verwenden. Die Frage lautet nun: Wieviel Aufträge lassen sich gleichzeitig ausführen. Dieses Problem lässt sich einfach als Graph modellieren. Die Menge der Aufträge sind die Knoten. Zwei Knoten sind durch eine Kante verbunden, wenn sie gemeinsamen Ressourcen benötigen. Wir suchen nun eine maximale Menge von Knoten mit der Eigenschaft, dass keine zwei dieser Knoten durch eine Kante verbunden sind.

Am Beispiel der Berechnung von unabhängigen Knotenmengen, wollen wir nun exemplarisch einen relativ einfachen Quantenalgorithmus vorstellen. Zunächst wollen wir unser Problem nochmals formal definieren. Eine Knotenmenge U eines Graphen G heißt unabhängig, wenn keine zwei Knoten dieser Menge in G benachbart sind. Die unabhängige Knotenmenge U heißt gesättigt, wenn keine unabhängige Knotenmenge in G existiert, die U echt enthält. Eine maximale unabhängige Knotenmenge ist eine unabhängige Knotenmenge maximaler Mächtigkeit von G. Eine gesättigte unabhängige Knotenmenge ist nicht notwendiger Weise maximal, siehe Abbildung 2.

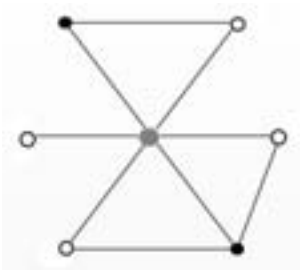

Abbildung 2: Gesättigte (grau) und maximale (weiß) unabhängige Knotenmenge

Theorem 3.1 *Die Quanten Query Komplexität zur Bestimmung einer unabhängigen gesättigten Knotenmenge ist $O(n^{1.5}\log n)$ im Matrix Modell und $O(\sqrt{nm}\log n)$ im Array Modell.*

Sei $G = (V, E)$ ein ungerichteter Graph, indem wir eine unabhängige gesättigte Knotenmenge V' bestimmen wollen. Am Anfang setzen wir $V' = \emptyset$ und wir markieren alle Knoten von G mit der Farbe weiß. Der Algorithmus testet nun in jedem Schritt, ob es noch weiße Knoten in G gibt. Falls ja, dann wählen wir einen solchen Knoten aus und fügen ihn zu der Menge V' hinzu. Dann verwenden wir den Grover Algorithmus, um die Menge aller weißen Nachbarknoten von v zu bestimmen. Diese Nachbarknoten und v werden nun mit der Farbe schwarz markiert. Aus diesem Verfahren sieht man sehr leicht, dass V' am Ende eine unabhängige gesättigte Knotenmenge ist.
Bestimmen wir nun die Anzahl der Fragen an das Matrix und Array Modell. Im Matrix Modell kann jeder Knoten in $O(\sqrt{n})$ Fragen gefunden werden. Somit erhalten wir eine Quanten Query Komplexität von $O(n^{1.5})$. Im Array Modell kostet die Bestimmung aller weißen Nachbarknoten von v genau $O(\sqrt{d_v \cdot a_v})$ Fragen, wobei a_v die Anzahl der weißen Nachbarknoten von v ist. Da jeder Knoten höchstens einmal mit der Farbe schwarz markiert wird, gilt $\sum_v a_v \leq n$. Mit Hilfe der Cauchy-Schwarz Ungleichung erhalten wir die Quanten Query Komplexität im Array Modell von

$$\sum_v \sqrt{d_v \cdot a_v} \leq \sqrt{\sum_v d_v} \cdot \sqrt{\sum_v a_v} = O(\sqrt{mn}).$$

Um eine konstant Erfolgswahrscheinlichkeit für unseren Algorithmus zu bekommen, müssen wir jede Grover Subroutine $O(\log n)$ wiederholen. Man kann zeigen, dass dieser einfacher Quantenalgorithmus bis auf den log-Faktor optimal ist.

Die Bestimmung einer maximalen unabhängigen Kontenmenge ist ein bekanntes NP-hartes Problem. Der schnellste heute bekannte Algorithmus hat eine Laufzeit von $O(1.1844^n)$ und basiert auf einer Computer generierten Analyse mit mehreren Tausend Fällen. Wir haben für dieses Problem einen Quantenalgorithmus konstruiert mit einer Laufzeit von $O(1.1488^n)$.

4 Quantenalgorithmen für Algebraprobleme

In unserer Arbeit untersuchen wir neben Graphenproblemen auch algebraische Algorithmen, die wir mittels verschiedener Quantentechniken beschleunigen können. Wir betrachten eine Menge S mit n Elementen und einer binären Operation auf S (Groupoid), repräsentiert als Operationstabelle. Wir interessieren uns dafür, ob die gegebene Tabelle bestimmte algebraische Eigenschaften besitzt, wie beispielsweise die Assoziativität (Semigruppe) oder ob sie eine Gruppe ist.

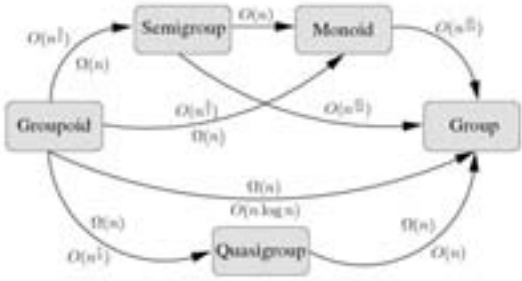

Abbildung 3: Quanten Query Komplexität zum Testen von algebraischen Eigenschaften

Die Abbildung 3 zeigt eine Zusammenfassung unserer Resultate. Beispielsweise haben wir nahezu optimale Algorithmen gefunden, zum testen, ob ein Groupoid oder eine Quasigruppe (jeder Zeile und Spalte der Tabelle ist eine Permutation von S) eine Gruppe ist. Für einige Fragestellungen wie z.B. ob ein Monoid (Semigruppe mit Einselement) eine Gruppe ist, haben wir gezeigt, dass man dieses Problem mittels eines Quantenalgorithmus in weniger als linearer Zeit lösen kann. Anwendungen finden solche Fragestellungen zum Beispiel in der Kryptographie, wo man sich insbesondere dafür interessiert, ob eine gegebene Operation eine Gruppe ist.

Semigruppen Problem. Wir wollen im folgenden Abschnitt einen Quantenalgorithmus für das Semigruppen Problem skizzieren. Die Aufgabe ist hier zu testen, ob unsere gegebene Operationstabelle assoziative ist. Unser Algorithmus stellt die erste Anwendung eines neuen Quanten Walk Suchschema von Magniez et al. [9] dar. Der beste bekannte klassische randomisierten Algorithmus hat eine Laufzeit von $O(n^2 \log n)$. Als zusätzlichen Parameter betrachten wir hier nun die binäre Operation $\circ : S \times S \to S'$, wobei $S' \subseteq S$. Für dieses Problem konstruieren wir einen Quantenalgorithmus mit der Query Komplexität von $O(n^{5/4})$, falls die Anzahl der Elemente in S' konstant ist.

Um ein wenig Intuition für dieses Problem zu bekommen, betrachten wir zunächst einen sehr einfachen Quantenalgorithmus. Mittels des Grover Algorithmus suchen wir über alle Elemente in der Menge $S \times S \times S$ ein nichtassoziatives Tripel. Falls so ein Tripel existiert, finden wir es mit einer konstanten Wahrscheinlichkeit. Die Größe unseres Suchraumes ist somit n^3. Also ist die Quanten Query Komplexität dieses Verfahrens $O(n^{3/2})$.

Theorem 4.1 *Sei $k = n^{\alpha}$ die Anzahl der Elemente von S' mit $0 < \alpha \leq 1$. Die Quanten Query Komplexität des Semigruppen Problems ist*

$$\begin{cases} O(n^{\frac{5+\alpha}{4}}), & \textit{für } 0 \leq \alpha \leq \frac{1}{3}, \\ O(n^{\frac{6+2\alpha}{5}}), & \textit{für } \frac{1}{3} < \alpha \leq \frac{3}{4}, \\ O(n^{\frac{3}{2}}), & \textit{für } \frac{3}{4} < \alpha \leq 1. \end{cases}$$

Unser Algorithmus verwendet eine Quanten Walk Suche nach einem nichtassoziativen Tripel (falls es ein solches gibt). Der Quanten Walk findet in einem Johnson Graph statt. Die Knotenmenge V_J des Johnson Graphen $J(n, r)$ sind alle Teilmengen von S mit $r < n$ Elementen. Zwei Knoten sind durch eine Kante verbunden, falls sie sich durch genau ein Element unterscheiden (siehe Abbildung 4). Der Quanten Walk startet an einem Knoten A von $J(n, r)$. Die Aufgabe ist nun, nach einem Knoten $M \in V_J$ zu suchen, mit der Eigenschaft $(a \circ b) \circ c \neq a \circ (b \circ c)$ für $a, b \in M$ und $c \in S$. Die Knoten welche diese Eigenschaft erfüllen, bezeichnen wir als markierte Knoten. Falls wir einen markierten Knoten finden wissen wir, dass die Operationstabelle keine Semiguppe ist. Andernfalls können wir mit einer konstanten Wahrscheinlichkeit feststellen, dass die Tabelle assoziative ist. Unsere Quanten Walk Suchverfahren benötigt für das Testen ob ein Knoten markiert ist, eine Datenbasis D. Für $A \in V_J$ definieren wir die Menge $D(A) = \{ (a, b, a \circ b) \mid a, b \in A \cup S' \}$. In jedem Schritt des Quanten Walks können wir aufgrund des Quantenparallelismus vom aktuellen Knoten gleichzeitig zu allen seinen Nachbarknoten gehen. Nun müssen wir die Datenbasis der Knoten aktualisieren. Hierzu benötigen wir r Fragen an die Operationstabelle, da sich die Nachbarknoten im Johnson Graph in nur einem Element aus S unterscheiden. Wir können zeigen, dass wir nach $\sqrt{nrk}$ Fragen an die Operationstabelle entscheiden können, ob der Knoten markiert ist oder nicht. Das Aktualisieren der Datenbasis und Testen ob ein Knoten markiert ist, geschieht ebenfalls parallel. Weiterhin kann gezeigt werden, dass wir nach $\frac{n}{r}$ Schritten einen markierten Knoten finden. Mit Hilfe des Quanten Walk Theorem von [9] erhalten wir dann die obige Query Komplexität.

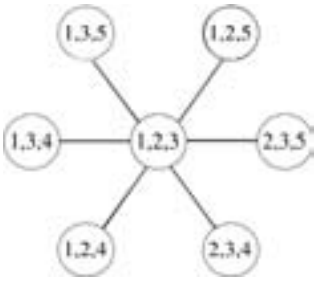

Abbildung 4: Nachbarknoten des Johnson Graph $J(5, 3)$

5 Fazit

In unserer Arbeit haben wir grundlegende Quantenalgorithmen für bekannte Probleme aus der Graphentheorie und Algebra gefunden, welche polynomial schneller sind, als die besten bekannten klassischen Verfahren. Es ist zur Zeit noch völlig unklar, ob Quantencomputer alle Probleme in der Komplexitätsklasse NP in polynomialer Zeit lösen können. Für das Faktorisierungsproblem ist dies der Fall, aber von diesem Problem wurde noch nicht nachgewiesen, dass es NP-vollständig ist. Die Entwicklung der Quantencomputer ist

heute noch im Bereich der Grundlagenforschung angesiedelt. Trotz vieler Schwierigkeiten eröffnet die Quantenmechanik faszinierende Perspektiven für die Kommunikation und Informationsverarbeitung. Es ist daher sehr wichtig, die Entwicklung von Quantencomputern und deren Algorithmen weiter voranzutreiben, um den notwendigen Rechenkapazität für Wettervorhersagen, Simulations- oder Optimierungsprozesse gewährleisten zu können.

Literatur

[1] A. Ambainis, R. Špalek, *Quantum Algorithms for Matching and Network Flows*, Proceedings of STACS'06: pages 172-183, 2006.

[2] C. Dürr, M. Heiligman, P. Høyer, M. Mhalla, *Quantum query complexity of some graph problems*, Proceedings of ICALP'04: pages 481-493, 2004.

[3] S. Dörn, T. Thierauf, *The Quantum Query Complexity of Algebraic Properties*, Proceedings of FCT'07: pages 250-260, 2007.

[4] S. Dörn, *Quantum Complexity of Graph and Algebraic Problems*, Dissertation, Universität Ulm, 2007.

[5] S. Dörn, T. Thierauf, *The Quantum Query Complexity of the Determinat*, Information Processing Letters, pages: 325-328, 2008.

[6] S. Dörn, *Quantum Algorithms for Matching Problems*, to appear at Theory of Computing Systems, 2008.

[7] L. Grover, *A fast mechanical algorithm for database search*, Proceedings of STOC'96: pages 212-219, 1996.

[8] F. Magniez, A. Nayak, *Quantum complexity of testing group commutativity*, Proceedings of ICALP'05: pages 1312-1324, 2005.

[9] F. Magniez, A. Nayak, J. Roland, M. Santha, *Search via Quantum Walk*, Proceedings of STOC'07: pages: 575-584, 2007.

[10] F. Magniez, M. Santha, M. Szegedy, *Quantum Algorithms for the Triangle Problem*, Proceedings of SODA'05: pages 1109-1117, 2005.

Sebastian Dörn wurde am 29. März 1981 in Mittweida geboren. Er hat von Oktober 2000 bis Februar 2005 Angewandte Mathematik an der Hochschule Mittweida und der Bergakademie Freiberg studiert. Nach seinem Abschluss war er seit April 2005 Stipendiat des Graduiertenkolleg „Mathematical Analysis of Evolution, Information and Complexity“ am Institut für Theoretische Informatik an der Universität Ulm. Seine Promotion hat er im Februar 2008 mit Auszeichnung abgeschlossen. Seit April 2008 ist er in der Forschungs- und Entwicklungsabteilung bei der Carl Zeiss SMT AG im Halbleiterbereich beschäftigt.