UNIVERSITÄT ZU LÜBECK
INSTITUTE OF MATHEMATICS
AND IMAGE COMPUTING

Nicht-binäre Quantenamplitudenverstärkung für Diskrete Optimierung

Non-Binary Quantum Amplitude Amplification For Discrete Optimization

Bachelorarbeit

verfasst am
Institute of Mathematics and Image Computing

im Rahmen des Studiengangs
Mathematik in Medizin und Lebenswissenschaften
der Universität zu Lübeck

vorgelegt von
Josephine Elisabeth Oettinger

ausgegeben und betreut von
Prof. Dr. Jan Lellmann

mit Unterstützung von
Natacha Kuete Meli

Lübeck, den 27. Oktober 2023

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

_____

Josephine Elisabeth Oettinger

Zusammenfassung

Ein klassisches Problem der diskreten Optimierung ist das Ising-Problem. Dieses beschreibt viele NP-vollständige Probleme und findet vielseitige praktische Anwendungen, zum Beispiel im Bereich der Bildverarbeitung und des Datenclusterings. Das Lösen des Ising-Problems ist also praxisrelevant und theoretisch höchst interessant, jedoch aufgrund des exponentiellen Wachstums des Suchraums mit der Problemgröße auch für klassische Rechner sehr herausfordernd.

Im Rahmen dieser Bachelorarbeit wird ein schaltkreis-basiertes Quantencomputing-Verfahren zum Lösen des Ising-Problems vorgestellt. Dieses basiert auf der sogenannten nicht-binären Amplitudenverstärkung, die wiederum eine Erweiterung des verallgemeinerten Grover-Algorithmus ist. Außerdem wird die nicht-binäre Amplitudenverstärkung angepasst, so dass die Grover-Phase-Matching-Bedingung näherungsweise erfüllt ist, um das Suchproblem mit einer höheren Wahrscheinlichkeit lösen zu können. Die beiden Verfahren werden miteinander und mit der bereits bestehenden Methode UQIsing verglichen.

Eine Durchführung der nicht-binären Amplitudenverstärkung liegt in $\mathcal{O}(\log(N)^2)$ bezüglich der Gatteranzahl. Durch das Erfüllen der Grover-Phase-Matching-Bedingung wird die Wahrscheinlichkeit, die Lösung zu messen, erhöht. Die Komplexität bezüglich der Gatteranzahl ändert sich aber zu $\mathcal{O}(\sqrt{N}\log(N)^2)$. Das neu entwickelte Verfahren kann noch nicht ganz mit dem approximation ratio existierender Verfahren, wie zum Beispiel UQIsing, mithalten. Es ist jedoch gewährleistet, dass die Basiszustände der Lösungen im Vergleich zu anderen Basiszuständen mit der höchsten Wahrscheinlichkeit gemessen werden.

Abstract

A classical problem of discrete optimisation is the Ising problem. This problem describes NP-complete problems and has many practical applications, for example in the field of image processing and data clustering. Solving the Ising problem is therefore practically relevant and theoretically highly interesting, but due to the exponential growth of the search space with the problem size, it is also very challenging for classical computers.

In this bachelor thesis, we present an circuit-based quantum computing method for solving the Ising problem. The method is based on the so-called non-binary amplitude amplification, which is an extension of the generalised Grover algorithm. In addition, the non-binary amplitude amplification is adjusted so that the Grover phase matching condition is approximately fulfilled in order to solve the search problem with a higher probability. The two methods are compared with each other and with the already existing method UQIsing.

One performance of the non-binary amplitude amplification is in $\mathcal{O}(\log(N)^2)$ with respect to the number of gates. Satisfying the Grover phase matching condition increases the probability of measuring the solution. However, the complexity with respect to the number of gates changes to $\mathcal{O}(\sqrt{N}\log(N)^2)$. The newly developed method cannot quite keep up with the approximation ratio of existing methods, such as UQIsing. However, it is guaranteed that the basis states of the solutions are measured with the highest probability compared to other basis states.

Danksagungen

Vielen vielen Dank an alle, die mich so fleißig beim Erstellen dieser Bachelorarbeit unterstützt haben. Ganz besonderer Dank geht natürlich an Natacha und Jan. Dankeschön, dass ihr wirklich jede noch so kleine Frage beantwortet habt, auch wenn die Antwort nicht 42 war.

Inhaltsverzeichnis

Liste benutzter Symbole

G	Grover-Operator, Verkettung der Operatoren S_{Ψ_0} und U_φ.
K	Anzahl der Iterationen eines Verfahrens.
M	Anzahl der Lösungen eines Suchproblems.
N	Anzahl der möglichen Eingaben eines Suchproblems.
S_{Ψ_0}	Diffusionsoperator, spiegelt entlang des Zustands $\|\Psi_0\rangle$.
U_φ	Quantenorakel auch 1-Register-Orakel, markiert jeden Basiszustand $\|x\rangle$ des Qubitsystems mit einer Phase $\varphi(x)$.
$\langle v\|$	Zustandsvekor in $\mathbb{C}^{1\times n}$, genannt Bra.
ϵ_x	Energiezustand im Ising-Modell, abhängig vom Zustand $x = (x_i)_{i=1}^n$, auch $\epsilon(x)$.
$\|\Psi_0\rangle$	Startzustand, Zustand in dem ein Algorithmus startet.
$\|\Psi_k\rangle$	Zustand des Systems nach k Iterationen innerhalb eines Algorithmus.
$\|\tilde{\Psi}\rangle$	Perfekt überlagerter Zustand, hat die Eigenschaft, dass jeder Basiszustand x in $\|\tilde{\Psi}\rangle$ die gleiche Amplitude hat.
$\|v\rangle$	Zustandsvekor in $\mathbb{C}^n$, genannt Ket.
$\overline{v_i}$	Die komplex konjugierte Zahl von v_i.
$\mathbf{U}_\varphi$	2-Register-Orakel, ein bedingter Orakelaufruf des Orakels U_φ.
v^H	Die Adjungierte des Vektors v, auch bei Matrizen verwendet.

1

Einleitung

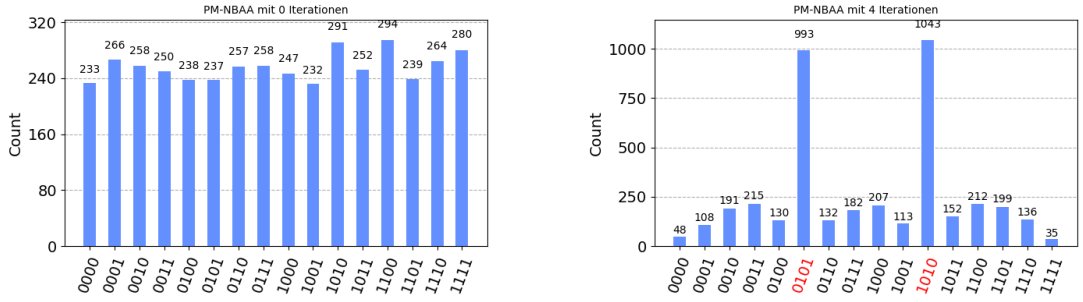
Das *Quantencomputing* ist eine neue Rechenmethode, die *Prinzipien der Quantenmechanik* nutzt, um leistungsfähig Rechnungen durchzuführen. In den letzten Jahren hat sich der Quantencomputer als eine vielversprechende Alternative zum klassischen Computer herausgestellt. Der Vorteil des Quantencomputings gegenüber des klassischen ist, dass beim Quantencomputing *Qubits* statt Bits verwendet werden. Ein Qubit kann mehr Zustände annehmen als ein klassisches Bit, das nur die möglichen Zustände „0" und „1" hat, und kann somit mehr Informationen speichern. Dadurch werden für viele Prozesse weniger Qubits benötigt als Bits beim klassischen Computing.

Das Quantencomputing lässt sich in zwei Kategorien unterteilen: das *adiabatische Quantencomputing* und das *schaltkreis-basierte Quantencomputing*. Das adiabatische Quantencomputing eignet sich zum Lösen von Optimierungsproblemen. Geräte wie *D-Wave Advantage* [11] können beispielsweise kombinatorische Probleme im Bereich der Computer Vision [29] und Datenbankensuche [34] effizient lösen. Schaltkreis-basiertes Quantencomputing ist flexibler einsetzbar, und es kann theoretisch jede klassische Operation implementiert werden [4].

1.1 Das Ising-Problem

Ein klassisches Problem der diskreten Optimierung ist das *Ising-Problem*, welches durch das *Ising-Modell* beschrieben wird [33]. Das Ising-Modell betrachtet ein quantenmechanisches System, das aus $n \in \mathbb{N}$ Teilchen besteht. Das magnetische Moment dieser Teilchen heißt *Spin*, und es wird angenommen, dass der Spin s_i jedes Teilchens i mit $i = 1, .., n$ nur die diskreten Zustände 1 und -1 annehmen kann. Die Teilchen interagieren mit einem externen Feld mit einer externen Energie der Stärke C_{ii} und untereinander. Die Stärke und Richtung der Interaktionen wird mit C_{ii} und C_{ij} beschrieben. Alle C_{ii} und C_{ij} sind reelle Zahlen und können sowohl positiv als auch negativ sein. Die Gesamtenergie ergibt sich als

$$\sum_{i=1}^{n} C_{ii} s_i + \sum_{1 \leq i < j \leq n} C_{ij} s_i s_j. \tag{1.1}$$



(a) Vor der Amplitudenverstärkung

(b) Nach der Amplitudenverstärkung

Abbildung 1.1: Veranschaulichung der Amplitudenverstärkung. Ziel der Amplitudenverstärkung ist es, die Wahrscheinlichkeit, die Lösung(en) zu messen, zu vergrößern und andererseits die Wahrscheinlichkeit, Nicht-Lösungen zu messen, zu verkleinern. Die Histogramme zeigen an, wie oft die Basiszustände gemessen wurden, die in diesem Beispiel den möglichen Zuständen eines Ising-Systems entsprechen. Bei (a) wurde vor der Amplitudenverstärkung gemessen. Alle Basiszustände sind etwa gleich häufig. In (b) sieht man die Ergebnisse der Messung nach der Amplitudenverstärkung. Hier wurden die Zustände „0101" und „1010" deutlich häufiger gemessen. Diese entsprechen genau den beiden Lösungen des betrachteten Ising-Problems.

Mit $s_i = (-1)^{x_1}$ und $x_i \in \{0, 1\}$ lässt sich (1.1) zu

$$\epsilon(x) := \sum_{i=1}^{n} C_{ii}(-1)^{x_i} + \sum_{1 \le i < j \le n} C_{ij}(-1)^{x_i + x_j} \tag{1.2}$$

umformen. Beim Ising-Problem wird der Zustand $x = (x_i)_{i=1}^n$ gesucht, bei dem der Energiezustand im Ising-Modell minimal ist:

$$\arg\min_{x \in \{0,1\}^n} \epsilon(x). \tag{1.3}$$

Die C_{ii} und C_{ij} beschreiben zusammen vollständig das Problem und werden zur *Kostenmatrix* $C \in \mathbb{R}^{n \times n}$ zusammengefasst. Das Ising-Problem beschreibt viele NP-vollständige Probleme und findet vielseitige praktische Anwendungen. So wird es zum Beispiel im Bereich der Bildverarbeitung [36] und des Datenclusterings [3] verwendet. Das Lösen des Ising-Problems ist also praxisrelevant und theoretisch höchst interessant. Wenn das Ising-Problem auf Quantencomputern in polynomieller Zeit lösbar ist, wäre Quantencomputing klassischem Computing weit überlegen.

In dieser Arbeit nutzen wir ein schaltkreis-basiertes Quantencomputing-Verfahren, die *nicht-binäre (Quanten-)Amplitudenverstärkung (NBAA)* [37], um das Ising-Problem zu lösen. Das Verfahren wurde 2021 von Shyamsundar vorgestellt und wurde nach unserem Kenntnisstand noch nicht auf das Ising-Problem angewendet. Bei NBAA wird jeder Eingabe eines nicht-binären Optimierungsproblems ein Basiszustand eines Quantensystems zugewiesen. Jeder Basiszustand hat eine *Amplitude*, die beschreibt, zu welchem Anteil sich das System in diesem Basiszustand befindet und dafür ausschlaggebend ist, wie wahrscheinlich der Basiszustand gemessen wird. Bei der nicht-binären Amplitudenverstärkung wird das Quantensystem Schritt für Schritt so modifiziert, dass sich einerseits die Amplituden der Basiszustände umso stärker vergrößern, je besser die dazugehörigen

Eingaben das Problem lösen. Auf der anderen Seite werden die Amplituden der zur Eingabe gehörenden Basiszustände umso stärker verkleinert, je schlechter die Eingabe das Optimierungsproblem löst. Am Ende wird der zur Lösung gehörende Basiszustand am häufigsten gemessen. Veranschaulicht ist dies in Abbildung 1.1.

In dieser Arbeit vorgestellte Neuerungen umfassen insbesondere:

- Das Verfahren NBAA wird angepasst, so dass dieses zum Lösen des Ising-Problems benutzt werden kann. Anschließend wurde experimentell ermittelt, welche Normierungsparameter die besten Ergebnisse liefern.
- Weiterführend wurde NBAA erweitert, damit in diesem die *Grover-Phase-Matching-Bedingung* erfüllt ist. Die Grover-Phase-Matching-Bedingung fordert, dass Phasen in bestimmten Operatoren einer Amplitudenverstärkung gleich sind. Das angepasste Verfahren nennen wir *PM-NBAA*. In diesem Rahmen wurde gezeigt, dass das erweiterte Verfahren gegen die Lösung des Suchproblems konvergiert. Das Optimieren der Anzahl der Iterationen von PM-NBAA steht noch aus.
- Außerdem wurden NBAA und PM-NBAA miteinander und mit der bereits bestehenden Methode UQIsing[28] experimentell verglichen.

1.2 Verwandte Arbeiten

Es gibt bereits einige Verfahren im Bereich des klassischen und des Quantencomputings, um das Ising-Problem zu lösen. Hierbei wird zwischen exakten und approximativen Verfahren unterschieden beziehungsweise zwischen Verfahren, die globale oder lokale Minimierer finden. Insbesondere im Bereich der kombinatorischen Optimierung wurden viele klassische Algorithmen entwickelt. Eine Übersicht dieser ist in [5, 24] zu finden. Im Folgenden betrachten wir kurz einige Verfahren aus verschiedenen Bereichen zum Lösen des Ising-Problems.

Klassisches Computing

Brute Force. Sucht man die Lösung des Ising-Problems nach dem *Brute Force Prinzip*, wird jede der 2^n Eingaben betrachtet und der Energiezustand berechnet. Anschließend werden die Energiezustände verglichen und der niedrigste Wert ermittelt und daraus dann der Minimierer. Da alle möglichen Eingaben angeschaut werden müssen, liegt die Anzahl der benötigten Funktionsauswertungen in $\mathcal{O}(2^n)$ und ist somit bereits für moderat große n nicht effizient umsetzbar.

Goemans and Williamson. Im Bereich des klassischen Computings stellten beispielsweise Goemans und Williamson [14] oder auch Jerrum und Sinclair [23] ein approximatives Verfahren mit besserer Laufzeit vor. Goemans und Williamsons Verfahren erweitert den Suchraum des Ising-Problems $\{0,1\}^n$ auf den n-Simplex, sucht im neuen Raum nach der Lösung und transformiert dann die gefundene Lösung in den ursprünglichen Suchraum zurück. Bei dem Prozess der Rücktransformation vom n-Simplex in $\{0,1\}^n$ gehen

Informationen verloren, wodurch die Lösung des Ising-Problems nicht immer gefunden wird.

Quantencomputing

Variationelle Schaltkreise. Approximative Verfahren im Umfeld des Quantencomputings lieferten zum Beispiel Fahri et al. [12] und Kuete Meli et al. [28]. Hierbei wird mithilfe variationeller Quantenschaltkreise der Minimierer gesucht. Die Winkel von Rotationen werden bei Kuete Meli et al. [28] mit dem *Gradientenabstiegsverfahren* optimiert. Da das Problem mittels einer nicht konvexen Funktion modelliert wird, kann nicht garantiert werden, dass das Gradientenabstiegsverfahren den globalen Minimierer findet. Daher wird die Lösung des Ising-Problems nicht immer gefunden.

Adiabatisches Quantencomputing (AQC). Im Bereich des adiabatischen Quantencomputings löst D-Wave [11, 13] das Ising-Problem. AQC beruht auf dem *Adiabatensatz der Quantenmechanik* [2]. Dieser besagt, dass – unter der Annahme einer hinreichend langsamen Evolutionsgeschwindigkeit – Quantensysteme mit hoher Wahrscheinlichkeit im Zustand des Energieniveaus bleiben, in dem sie gestartet sind [6]: Starten wir im *Grundzustand*, dem Zustand mit dem niedrigsten Energieniveau, bleiben das System auch im Grundzustand. Die Evolution eines aus n Quanten bestehenden Systems zu einem Zeitpunkt $t \in [0, T]$ kann durch einen *Hamiltonoperator* $H(t) \in \mathbb{C}^{2^n \times 2^n}$ beschrieben werden.

Die Idee des AQCs ist nun, den Hamiltonoperator und die Evolution so wählen, dass der Grundzustand des Systems zum Zeitpunkt T die Lösung des Ising-Problems ist. Das System wird nun in einen einfachen Grundzustand initialisiert und so langsam verändert, dass es mit hoher Wahrscheinlichkeit im Grundzustand bleibt. Am Ende wird der Zustand des Systems gemessen, der idealerweise der Minimierer des Ising-Problems ist. Je geringer der Unterschied zwischen den niedrigen Energieniveaus im Ising-Modell ist, desto wahrscheinlicher ist es, dass das System in einem höheren Energiezustand landet und den Grundzustand verlässt. Liegen die niedrigen Energieniveaus im Ising-Modell zu nahe beieinander, kann die Lösung demnach nicht mehr sicher gefunden werden. Dieses Problem wird als *Gap Problem* bezeichnet.

1.3 Aufbau der Arbeit

Die in dieser Arbeit vorgestellten Verfahren NBAA und PM-NBAA sind schaltkreis-basierte Quantencomputing-Verfahren zum Lösen des Ising-Problems. Um diese vorzustellen und zu untersuchen, ist diese Arbeit folgendermaßen aufgebaut:

In **Kapitel 2** werden die *vier Postulate des Quantencomputings* und das schaltkreisbasierte Quantencomputing eingeführt. Diese bilden die Grundlage für die darauffolgenden Verfahren. Anschließend wird in **Kapitel 3** die *binäre Amplitudenverstärkung* betrachtet, um das *Prinzip der Amplitudenverstärkung* vereinfacht anhand des *verallgemeinerten Grover-Algorithmus* [8] zu erklären. Hierbei wird auch die Grover-Phase-Matching-Bedingung eingeführt. Die danach in **Kapitel 4** vorgestellte nicht-binäre Amplitudenver-

stärkung ist eine Erweiterung der binären Amplitudenverstärkung. Hier wird NBAA von Shyamsundar et al. [37] näher betrachtet. Nachdem NBAA vorgestellt wurde, wird der Algorithmus in **Kapitel 5** angepasst, so dass dieser zum Lösen des Ising-Problems benutzt werden kann. Dazu wird insbesondere eine effiziente Implementierung des *Orakels* präsentiert. Anschließend wird NBAA in **Kapitel 6** angepasst, so dass die Grover-Phase-Matching-Bedingung erfüllt ist. Der angepasste Algorithmus PM-NBAA wird dann mit NBAA und UQIsing [28] verglichen. Zum Schluss wird die Arbeit zusammengefasst und ein Ausblick gegeben.

2

Grundlagen und Notation

Um die Algorithmen dieser Arbeit besser verstehen zu können, werden in diesem Abschnitt die Grundlagen des Quantencomputings eingeführt. Anders als im Fall klassischer Rechner, die mit Bits arbeiten, ist die grundlegende Informationseinheit des Quantencomputings das sogenannte *Quantenbit*. Ein Quantenbit hat mehr Freiheitsgrade als ein Bit und kann so mehr Informationen speichern. Eine genaue Definition eines Quantenbits sowie weitere mathematische Grundlagen des Quantencomputings werden in diesem Kapitel in Abschnitt 2.1 behandelt. Anschließend wird schaltkreis-basiertes Quantencomputing in Abschnitt 2.2 eingeführt. Die folgenden Grundlagen basieren auf [21], [32] und [35].

2.1 Die vier Postulate der Quantenmechanik

Die vier Postulate der Quantenmechanik bilden die mathematische Grundlage des Quantencomputings und verbinden die Quantenphysik mit der Mathematik.

Postulat 2.1 (Zustandsraum). Der Zustand eines physikalischen Systems wird vollständig durch einen Einheitsvektor – den *Zustandsvektor* – in einem *komplexen Hilbertraum* – dem *Zustandsraum* – beschrieben.

Definition 2.2 (Hilbertraum). Ein Hilbertraum $\mathcal{H}$ ist ein (*endlich-dimensionaler*) $\mathbb{K}$-*Vektorraum* mit einem *Skalarprodukt* $\langle \cdot, \cdot \rangle$, der im Hinblick auf die *vom Skalarprodukt induzierte Norm* $\| \cdot \|$ *vollständig* ist (siehe auch [19, 26]).

Definition 2.3 (vollständig). Sei $(V, \| \cdot \|)$ ein normierter $\mathbb{K}$-Vektorraum. $(V, \| \cdot \|)$ heißt vollständig, wenn für jede *Cauchy-Folge* $(v^n)_n$ in V ein $v \in V$ existiert, so dass v^n gegen v konvergiert für $n \to \infty$.

Im Quantencomputing wird als Zustandsraum $\mathbb{C}^n$ (mit $n \in \mathbb{N}$) mit dem Skalarprodukt $\langle \cdot, \cdot \rangle : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{C}$ gewählt, das für $v = (v_1, ..., v_n)^\top, w = (w_1, ..., w_n)^\top \in \mathbb{C}^n$ definiert ist als

$$\langle v, w \rangle := v^H w = \sum_{i=1}^{n} \overline{v_i} w_i. \tag{2.1}$$

Hierbei ist v^H die *Adjungierte* des Vektors v und $\overline{v_i}$ die *komplex konjugierte Zahl* von v_i. Die vom Skalarprodukt induzierte Norm $\|\cdot\| : \mathbb{C}^n \to \mathbb{C}$ ist für $v \in \mathbb{C}^n$

$$\|v\| := \sqrt{\langle v, v \rangle}. \tag{2.2}$$

Die *Dirac-Notation* (beziehungsweise *Bra-Ket-Notaion*) wird verwendet, um die Zustandsvektoren darzustellen. Bei dieser ist der Vektor $|v\rangle \in \mathbb{C}^n$ (genannt *Ket*) ein Spaltenvektor und $\langle v| \in \mathbb{C}^{1\times n}$ (genannt *Bra*) ein Zeilenvektor beziehungsweise das lineare Funktional auf $\mathcal{H}$, das durch $w \mapsto \langle v, w\rangle$ beschrieben ist.

Im Vergleich zu klassischen Rechnern, die mit klassischen Bits rechen, rechnen Quantenrechner mit Quantenbits, kurz *Qubits*. Klassische Bits können nur die Zustände „0" und „1" annehmen. Quantenbits unterscheiden sich zu diesen dadurch, dass sie aus einer *Überlagerung* der beiden Zustände bestehen können. Ein Quantenbit ist der Zustand eines Einzelpartikels eines physikalischen Quantensystems. Mathematisch ausgedrückt wird ein Qubit wie folgt definiert:

Definition 2.4 (Quantenbit, Qubit). Ein Qubit ist eine Linearkombination

$$|\psi\rangle = \alpha \cdot |0\rangle + \beta \cdot |1\rangle \tag{2.3}$$

von zwei Basiszuständen

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{und} \quad |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{2.4}$$

mit $\alpha, \beta \in \mathbb{C}$ und $|\alpha|^2 + |\beta|^2 = 1$.

Der Zustand eines Qubits kann durch einen Einheitsvektor im Zustandsraum $\mathbb{C}^2$ beschrieben werden. Die Amplituden α und β beschrieben hierbei den Anteil des Qubits in den jeweiligen Basiszuständen. Ist der Zustandsvektor nicht ein komplexes Vielfaches einer der beiden Basiszustände, wird er als *überlagert* bezeichnet.

Definition 2.5 (Überlagerter Zustand, *Superposition*). Ein Qubit $|\psi\rangle = \alpha \cdot |0\rangle + \beta \cdot |1\rangle$ mit $\alpha, \beta \in \mathbb{C}$ befindet sich in einem überlagerten Zustand, wenn $\alpha \neq 0$ und $\beta \neq 0$ gilt.

Während $|\psi\rangle$ ein Spaltenvektor ist, ist die Adjungierte von $|\psi\rangle$ ein Zeilenvektor. Das *adjungierte Qubit* $\langle\psi| \in \mathbb{C}^{1\times 2}$ des Qubits $|\psi\rangle = \alpha \cdot |0\rangle + \beta \cdot |1\rangle$ mit $\alpha, \beta \in \mathbb{C}$ ist:

$$\langle\psi| := \overline{\alpha} \cdot \langle 0| + \overline{\beta} \cdot \langle 1| \tag{2.5}$$

mit den Definitionen

$$\langle 0| := \begin{pmatrix} 1 & 0 \end{pmatrix} \quad \text{sowie} \quad \langle 1| := \begin{pmatrix} 0 & 1 \end{pmatrix}. \tag{2.6}$$

Kombinieren wir ein adjungiertes Qubit mit einem Qubit, erhalten wir das Skalarprodukt. Wir nutzen die Dirac-Notation auch für das Skalarprodukt (2.1) von $|v\rangle, |w\rangle \in \mathbb{C}^n$ und schreiben

$$\langle |v\rangle, |w\rangle \rangle = \langle v| \, |w\rangle =: \langle v|w\rangle. \tag{2.7}$$

Das Skalarprodukt wird auch *inneres Produkt* genannt. Tauschen wir die Reihenfolge der Vektoren, erhalten wir das *äußere Produkt*. Dieses ist für $|v\rangle = (v_1, ..., v_n)^\top$ und $|w\rangle = (w_1, ..., w_n)^\top \in \mathbb{C}^n$ folgendermaßen definiert:

$$|v\rangle \langle w| := \begin{pmatrix} v_1 \\ ... \\ v_n \end{pmatrix} \begin{pmatrix} \overline{w_1} & ... & \overline{w_n} \end{pmatrix} = \begin{pmatrix} v_1\overline{w_1} & ... & v_1\overline{w_n} \\ ... & & ... \\ v_n\overline{w_1} & ... & v_n\overline{w_n} \end{pmatrix}. \tag{2.8}$$

Das innere Produkt $\langle v|w\rangle$ erzeugt demnach eine Zahl und das äußere $|v\rangle \langle w|$ eine Matrix.

Um ein System von einem Zustand zu einem anderen zu überführen, wird eine *unitäre Transformation* benutzt.

Postulat 2.6 (Evolution). Die Evolution eines isolierten Systems wird durch eine unitäre Transformation beschrieben:

$$|\psi_{t_2}\rangle = U \cdot |\psi_{t_1}\rangle, \tag{2.9}$$

wobei $|\psi_{t_1}\rangle \in \mathbb{C}^n$ der Zustand des Systems zum Zeitpunkt t_1, $|\psi_{t_2}\rangle \in \mathbb{C}^n$ der Zustand des Systems zum Zeitpunkt t_2 und $U \in \mathbb{C}^{n\times n}$ eine *unitäre Matrix* ist. Die Matrix U hängt dabei nur von t_1 und t_2 ab und insbesondere nicht von den Zuständen $|\psi_{t_1}\rangle$ und $|\psi_{t_2}\rangle$.

Definition 2.7 (Unitäre Matrix). Sei $n \in \mathbb{N}$, $U \in \mathbb{C}^{n\times n}$ eine Matrix und U^H die adjungierte Matrix von U. Die Matrix U ist *unitär*, wenn sie invertierbar ist und

$$U^{-1} = U^H \tag{2.10}$$

gilt, das heißt

$$U^H U = UU^H = I_n, \tag{2.11}$$

wobei I_n die $n \times n$-Einheitsmatrix ist.

Die Evolution eines isolierten Systems ist somit reversibel. Beginnend in einem Startzustand beschreibt die Evolution des Systems die Reihenfolge von Operationen im Sinne eines Algorithmus. Um den Endzustand beziehungsweise das Ergebnis zu erhalten, muss eine Messung vorgenommen werden. Messungen im Quantencomputing sind irreversibel.

Postulat 2.8 (Messung). Sei $n \in \mathbb{N}$. Messungen an einem System werden durch eine *Familie von Messoperatoren* $\{M_m \in \mathbb{C}^{n\times n} | m \in \mathcal{M}\}$ mit

$$\sum_{m\in\mathcal{M}} M_m^H M_m = I_n \tag{2.12}$$

beschrieben. Dabei ist $\mathcal{M}$ die *Menge der Messergebnisse*.

Sei $|\psi\rangle$ der Zustand vor der Messung. Die Wahrscheinlichkeit, bei der Messung den Ausgang m zu beobachten, ist

$$p(m) := \langle\psi| M_m^H M_m |\psi\rangle. \tag{2.13}$$

Der Zustand nach der Beobachtung von m wird zu

$$|\psi\rangle_{M=m} := \frac{1}{\sqrt{\langle\psi| M_m^H M_m |\psi\rangle}} M_m |\psi\rangle. \tag{2.14}$$

Qubits werden oft in der sogenannten *Rechenbasis* gemessen. Im Quantencomputing gilt: Wenn ein Qubit in der Rechenbasis gemessen wird, kollabiert der Zustand entweder in den Zustand $1 \cdot |0\rangle$ oder $1 \cdot |1\rangle$. Die Amplituden des Qubits können nicht gemessen werden.

Beispiel 2.9 (Messung in der Rechenbasis). Die Messung eines Qubits in der Rechenbasis wird durch die Messoperatoren M_0 und M_1 beschrieben, wobei

$$M_0 := |0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \text{ und } M_1 := |1\rangle\langle 1| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2.15}$$

Wird zum Beispiel $|\psi\rangle = \alpha \cdot |0\rangle + \beta \cdot |1\rangle$ mit $\alpha, \beta \in \mathbb{C}$ in der Rechenbasis gemessen, gilt

$$p(0) = \langle\psi| M_0^H M_0 |\psi\rangle = |\alpha|^2 \quad \text{und} \tag{2.16}$$

$$p(1) = \langle\psi| M_1^H M_1 |\psi\rangle = |\beta|^2. \tag{2.17}$$

Die Messung liefert „0" mit einer Wahrscheinlichkeit $|\alpha|^2$ und „1" mit einer Wahrscheinlichkeit von $|\beta|^2$. Je nachdem, ob „0" oder „1" gemessen wurde, befindet sich das System danach im Zustand

$$|\psi\rangle_{M=0} := \frac{1}{\sqrt{\langle\psi| M_0^H M_0 |\psi\rangle}} M_0 |\psi\rangle = |0\rangle \quad \text{beziehungsweise} \tag{2.18}$$

$$|\psi\rangle_{M=1} := \frac{1}{\sqrt{\langle\psi| M_1^H M_1 |\psi\rangle}} M_1 |\psi\rangle = |1\rangle. \tag{2.19}$$

Diese Eigenschaften machen wir uns zunutze, um die Amplituden eines Qubits zu schätzen. Die Amplituden α und β können nur durch mehrfaches Erzeugen des gleichen Qubits und Messen dieses Qubits geschätzt werden. Das genaue Vorgehen wird anhand eines Beispiels erläutert.

Beispiel 2.10 (Schätzung der Amplituden von $H|0\rangle$). Es sollen die Amplituden des Qubits $|q\rangle = H|0\rangle$ geschätzt werden, wobei

$$H := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{2.20}$$

die sogenannte *Hadamard-Transformation* ist.

In der Theorie lässt sich

$$|q\rangle = H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{2.21}$$

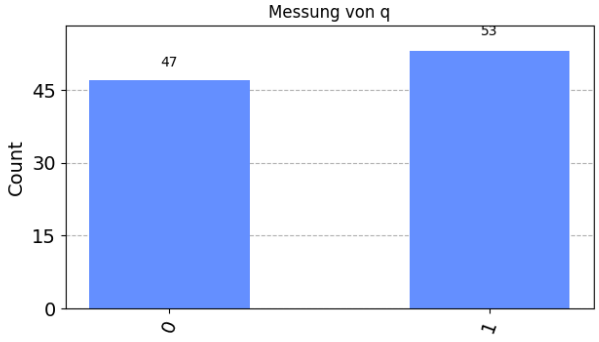
einfach berechnen und die Amplituden

$$\alpha = \frac{1}{\sqrt{2}} \text{ und } \beta = \frac{1}{\sqrt{2}}. \tag{2.22}$$

können direkt abgelesen werden. In der Praxis ist oft nicht bekannt, wie genau ein Qubit erzeugt und welche Transformationen auf das Qubit angewendet wurden. Um trotzdem

Messung	Messwert
1.	„0″
2.	„1″
3.	„1″
4.	„0″
...	
99.	„1″
100.	„1″

(a) Messwerte als Tabelle



(b) Histogramm der Messwerte

Abbildung 2.1: Die Amplituden α und β von $|q\rangle = H|0\rangle$ sollen geschätzt werden. Dazu wird das Qubit mehrmals erzeugt und gemessen, in diesem Beispiel 100-mal. Die Messwerte sind in (a) aufgelistet. Es wird nun gezählt, wie oft „0″ und wie oft „1″ gemessen wurde. Diese Werte wurden in (b) als Histogramm darstellt. Auf der x-Achse sind die möglichen Messergebnisse aufgeführt. Counts auf der y-Achse zeigt an, wie oft das jeweilige Ergebnis gemessen wurde. Es wurde also 47-mal die „0″ und 53-mal die „1″ gemessen. Das heißt, für die Amplituden α und β gilt: $|\alpha|^2 \approx |\beta|^2 \approx 0,5$.

α und β zu schätzen, wird das Qubit mehrmals erzeugt und gemessen. In unserem Beispiel wird das Erzeugen und Messen 100-mal wiederholt. Es wird nun gezählt, wie oft „0″ und wie oft „1″ gemessen wurde. Diese Werte lassen sich in einem Histogramm darstellen (siehe Abbildung 2.1). Hier wurde 47-mal „0″ und 53-mal „1″ gemessen. Aus den Messwerten kann jetzt

$$|\alpha|^2 \approx \frac{\text{Anzahl der Messungen von „0″}}{\text{Anzahl aller Messungen}} = \frac{47}{100} \tag{2.23}$$

und

$$|\beta|^2 \approx \frac{\text{Anzahl der Messungen von „1″}}{\text{Anzahl aller Messungen}} = \frac{53}{100} \tag{2.24}$$

geschätzt werden. Die Werte in (2.23) und (2.24) stimmen ungefähr mit $|\alpha|^2$ und $|\beta|^2$ der errechneten Werte aus (2.22) überein. Da die Messwahrscheinlichkeiten nur von $|\alpha|^2$ und $|\beta|^2$ abhängen, können grundsätzlich nur die Beträge (Magnituden) von α und β, nicht aber deren Phasen geschätzt werden.

Postulat 2.11 (*Zusammengesetztes System*). Der Zustandsraum eines zusammengesetzten Systems ist das *Tensorprodukt* der einzelnen Zustandsräume. Ein zusammengesetzter Zustand ist das Tensorprodukt der einzelnen Zustände.

Definition 2.12 (Tensorprodukt, *Kronecker-Produkt*). Seien $A = (a_{ij})_{\substack{i=1,..,n;\\ j=1,...,m}} \in \mathbb{C}^{n\times m}$, $B \in \mathbb{C}^{l\times k}$ mit $n, m, l, k \in \mathbb{N}$ Matrizen. Das Tensorprodukt von A und B ist

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1m}B \\ \dots & & \dots \\ a_{n1}B & \dots & a_{nm}B \end{pmatrix}. \tag{2.25}$$

Es können demnach mehrere Qubits kombiniert werden. Betrachtet man ein n-*Qubitsystem*, ein System, das aus $n \in \mathbb{N}$ Qubits zusammengesetzt ist, ist der Zustandsraum

$\mathbb{C}^{2^n}$. Seien $|q_1\rangle, ..., |q_n\rangle \in \mathbb{C}^2$ Qubits. Der aus $|q_1\rangle, ..., |q_n\rangle$ zusammengesetzte Zustand ist das Tensorprodukt (oder auch Kronecker-Produkt) der einzelnen Qubits:

$$|q\rangle = |q_1 ... q_n\rangle = |q_1\rangle \otimes ... \otimes |q_n\rangle \in \mathbb{C}^{2^n}. \tag{2.26}$$

Lässt sich ein Zustand nicht als Tensorprodukt von Zuständen der Einzelsysteme schreiben, heißt dieser *verschränkt*.

Ein aus zwei Qubits ($|q_1\rangle, |q_2\rangle \in \mathbb{C}^2$) zusammengesetzter Zustand ist zum Beispiel

$$|q\rangle = |q_1 q_2\rangle = |q_1\rangle \otimes |q_2\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle \in \mathbb{C}^4. \tag{2.27}$$

Wichtig ist auch hier, dass $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$ und $\alpha_{00}, \alpha_{01}, \alpha_{10}, \alpha_{11} \in \mathbb{C}$.

Eine übliche Notation ist, Basiszustände q als Binärzahl aufzufassen und mit der zugehörigen ganzen Zahl darzustellen, etwa

$$|0\rangle \otimes |1\rangle \otimes |0\rangle = |010\rangle = |2\rangle. \tag{2.28}$$

Die kanonische orthogonale Basis des Gesamtsystems mit n Qubits und $N := 2^n$ ist damit $\{|0\rangle, ..., |N-1\rangle\}$.

Werden mehrere Transformationen auf mehrere Qubits angewendet, werden diese mit $\otimes$ verknüpft. Die Hadamard-Transformation auf n Qubits angewendet wird mit $H^{\otimes n}$ bezeichnet. Wird diese auf $|0\rangle^{\otimes n}$ angewandt, ergibt sich der Zustand

$$|\tilde{\Psi}\rangle := H^{\otimes n} |0\rangle^{\otimes n} = (H|0\rangle) \otimes ... \otimes (H|0\rangle) = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle. \tag{2.29}$$

Dieser Zustand $|\tilde{\Psi}\rangle$ wird auch *perfekt überlagerter Zustand* genannt. Der perfekt überlagerte Zustand hat die Eigenschaft, dass jeder Basiszustand $|x\rangle \in \{|0\rangle, ..., |N-1\rangle\}$ in $|\tilde{\Psi}\rangle$ die gleiche Amplitude hat. Das heißt, beim Messen von $|\tilde{\Psi}\rangle$ erhält man $0, .., N-1$ mit genau gleicher Wahrscheinlichkeit.

2.2 Schaltkreis-basiertes Quantencomputing

Beim schaltkreis-basierten Quantencomputing werden unitäre Transformationen der Zustandsvektoren durch sogenannte *Gatter* (im Englischen: gates) durchgeführt. Eine Schaltkreis-Grafik wird zur Visualisierung der Quanten-Operationen verwendet. Der sogenannte *Quantenschaltkreis* ist eine Kombination von Gattern und beschreibt eine Hintereinanderausführung von Operationen auf Qubits.

Ein beispielhafter Schaltkreis ist in Abbildung 2.2 dargestellt. Der Quantenschaltkreis ist von links nach rechts zu lesen. Einzelne Qubits werden durch Linien dargestellt. Wenn keine Unterscheidung zwischen den einzelnen Qubits nötig ist, werden sie gegebenenfalls in einem sogenannten *Register* zusammengefasst, symbolisiert durch drei dichte, parallele Linien mit einer Zahl, die die Anzahl der Qubits im Register angibt. Gatter bzw. Transformationen einzelner oder mehrerer Qubits werden durch Rechtecke symbolisiert, Messungen mit  .

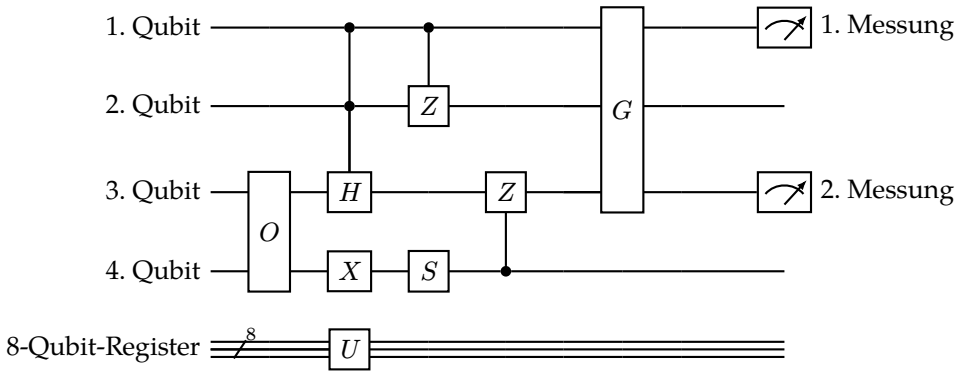


Abbildung 2.2: Quantenschaltkreis mit 12 Qubits. Die ersten vier Linien sind jeweils ein Qubit. Unten ist ein Register, das aus acht Qubits besteht, dargestellt. X und S sind Einzelgatter; O, U und G sind Multiqubitgatter. Z ist ein von einem Qubit kontrolliertes Gatter und H wird von zwei Qubits kontrolliert. Das erste und das dritte Qubit werden am Ende gemessen.

Einzelgatter sind Transformationen, die nur auf einem Qubit wirken. Gatter, die auf mehreren Qubits wirken, heißen *Multiqubitgatter*. Werden Transformationen auf Qubits unabhängig von anderen Qubits ausgeführt, handelt es sich um *nicht-kontrollierte Gatter*. Ist eine Transformation abhängig von anderen Qubits, auf die sie nicht wirken, wird die Transformation ein *Kontrollgatter* genannt. Diese zwei Gattertypen werden im Folgenden näher erläutert.

Einzelgatter

Einzelgatter wirken nur auf einem Qubit. Die zusammengesetzte Matrixdarstellung von n Einzelgattern wird mathematisch, genauso wie der zusammengesetzte Zustandsraum, als Tensorprodukt der n Einzelgatter berechnet. Einzelgatter, die im weiteren Verlauf wichtig sind, sind in Tabelle 2.3 zusammengefasst. Das „NOT"-Gatter hat zwei Darstellungen und kann sowohl als „X" als auch als durchkreuzter Kreis dargestellt werden. Es vertauscht die Basiszustände $|0\rangle$ und $|1\rangle$. Phasengatter verändern die Phase des Qubits. Das Hadamard-Gatter wurde bereits in (2.20) eingeführt. Auf $|0\rangle$ angewendet, erzeugt es den perfekt überlagerten Zustand. Das R_y-Gatter ist eine Rotation.

Ein Beispiel für die zusammengesetzte Darstellung von mehreren Einzelgattern ist $H \otimes H \otimes H$, das zu $H^{\otimes 3}$ zusammengefasst wird. Der dazugehörige Schaltkreis ist in Abbildung 2.4 zu finden.

Kontrollgatter

Eine spezielle Art der Multiqubitgatter sind Kontrollgatter. Bei diesen wird abhängig von *Kontrollqubits* entschieden, ob eine Operation auf anderen Qubits – den *Zielqubits* – angewendet werden soll.

Kontrollgatter können nicht als Kronecker-Produkt von Matrizen der Form $A \otimes B$ dargestellt werden. Daher werden diese der einfachheitshalber in dieser Arbeit mit Kronecker-Produkt und *Matrixpotenzen* ausgedrückt.

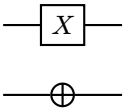
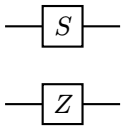
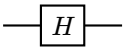
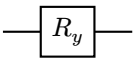
Name	Matrix	Gatter
„NOT"-Gatter	$X := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	
Phasengatter	$S := \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$, $Z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	
Hadamard-Gatter (Hadamard-Transformation als Gatter)	$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	
R_y*-Gatter* mit $\theta \in \mathbb{R}$	$R_y(2\theta) := \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$	

Tabelle 2.3: Wichtige Einzelgatter

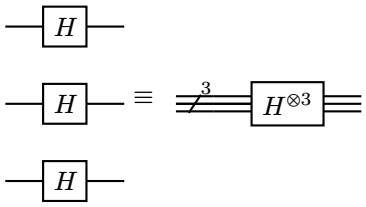


Abbildung 2.4: Das Hadamard-Gatter auf drei Qubits angewendet. Beide Schaltkreise stellen die gleiche Transformation dar. Rechts werden die drei Qubits zu einem Register zusammengefasst.

Definition 2.13 (Matrixpotenz). Für $A \in \mathbb{C}^{n \times n}$ mit $n \in \mathbb{N}$ wird die k-te Matrixpotenz A^k als

$$A^0 := I_n \text{ und } A^k := AA^{k-1} \text{ für } k \in \mathbb{N} \tag{2.30}$$

definiert.

Die formale Definition eines Kontrollgatters ist dann:

Definition 2.14 (Kontrollgatter). Bei einem Kontrollgatter kontrolliert eine Menge von Qubits $\{|q_1\rangle, .., |q_{m_1}\rangle\}$ mit $m_1 \in \mathbb{N}$, ob auf anderen Qubits $\{|p_1\rangle, .., |p_{m_2}\rangle\}$ mit $m_2 \in \mathbb{N}$ ein bestimmtes Gatter $U \in \mathbb{C}^{2^{m_2} \times 2^{m_2}}$ ausgeführt wird oder nicht. Seien alle Qubits $|q_1\rangle, .., |q_{m_1}\rangle$ sowie $|p_1\rangle, .., |p_{m_2}\rangle$ Basiszustände. Die Matrixdarstellung des Gatters ist dann:

$$I_2^{\otimes m_1} \otimes U^{q_1 \cdot \ldots \cdot q_{m_1}} \in \mathbb{C}^{2^{m_1+m_2} \times 2^{m_1+m_2}}. \tag{2.31}$$

Beispiel 2.15. Ein viel benutztes Kontrollgatter ist das kontrollierte „NOT"-Gatter, genannt *„CNOT"-Gatter*. Bei diesem kontrolliert das erste Qubit $|x_0\rangle$, ob auf dem zweiten $|x_1\rangle$ ein „NOT"-Gatter ausgeführt wird. Wenn $x_0 = 0$ gilt, verändern sich die Zustände

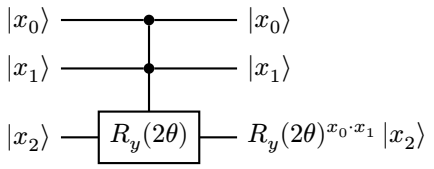


Abbildung 2.5: Ein von x_0 und x_1 kontrolliertes R_y-Gatter. Im Fall $x_0 = 1$ und $x_1 = 1$ wird auf x_2 das R_y-Gatter angewandt. Ist entweder $x_0 = 0$ oder $x_1 = 0$ oder sind beide 0, dann findet keine Rotation statt und x_0, x_1 und x_2 bleiben unverändert.

nicht. Ist $x_0 = 1$, wird $x_1 = 0$ zu $x_1 = 1$ und $x_1 = 1$ zu $x_1 = 0$. Sind $|x_0\rangle$ und $|x_1\rangle$ Basiszustände sieht Matrixschreibweise des „CNOT"-Gatters folgendermaßen aus:

$$I_2 \otimes X^{x_0} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^{x_0} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2.32}$$

Beispiel 2.16. Ein anderes Beispiel für ein Kontrollgatter ist ein von $|x_0\rangle$ und $|x_1\rangle$ kontrolliertes R_y-Gatter auf $|x_2\rangle$:

$$I_2 \otimes I_2 \otimes R_y(2\theta)^{x_0 \cdot x_1} = I_2 \otimes I_2 \otimes \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}^{x_0 \cdot x_1}, \tag{2.33}$$

wobei $|x_0\rangle$, $|x_1\rangle$ und $|x_2\rangle \in \mathbb{C}^2$ Basiszustände sind. Dieses Gatter ist als Schaltkreis in Abbildung 2.5 dargestellt.

Zusammenfassung

Wir haben die grundlegenden Prinzipien des Quantencomputings und schaltkreis-basierten Quantencomputing eingeführt. Bei Letzterem wurde insbesondere auf Einzelgatter und Kontrollgatter eingegangen. Insgesamt wurden in diesem Kapitel die Grundlagen geschaffen, so dass nun Algorithmen des Quantencomputings betrachtet werden können. Für das Lösen des Ising-Problems wollen wir das Prinzip der Amplitudenverstärkung benutzen. Dies wird im nächsten Kapitel vorgestellt.

3

Binäre Amplitudenverstärkung

Die Amplitudenverstärkung ist ein *iteratives Verfahren* zum Lösen von Optimierungsproblemen. Die Idee der Amplitudenverstärkung ist es, ausgehend von einer Superposition die Amplituden der gesuchten Zustände bei jeder Iteration zu vergrößern und die der nicht gesuchten Zustände zu verkleinern.

Um die Grundlagen der Amplitudenverstärkung besser zu verstehen, wird in diesem Kapitel die Amplitudenverstärkung erst zum Lösen eines *binären Suchproblems* verwendet. Hierfür wird der verallgemeinerte Grover-Algorithmus vorgestellt. Außerdem wird die Grover-Phase-Matching-Bedingung eingeführt. Im darauffolgenden Kapitel wird anschließend die nicht-binäre Amplitudenverstärkung betrachtet.

3.1 Binäres Suchproblem

Wir betrachten das folgende abstrakte binäre Suchproblem: Gegeben sei eine reellwertige Funktion $f : \{0, ..., 2^n - 1\} \to \{0, 1\}$ mit

$$f(x) := \begin{cases} 1, & x \text{ ist Lösung,} \\ 0, & \text{sonst.} \end{cases} \tag{3.1}$$

Gesucht ist eine Eingabe x aus der Menge $\{0, ..., 2^n - 1\} = \{0, ..., N-1\}$, so dass $f(x) = 1$ gilt.

3.2 Verallgemeinerter Grover-Algorithmus

Bei der binären Amplitudenverstärkung wird der Ansatz der Amplitudenverstärkung zum Lösen binärer Suchprobleme genutzt. Den Grundstein für die binäre Amplitudenverstärkung im Quantencomputing legte Grover 1996 mit dem *Grover-Algorithmus* [15]. Dieser wurde vielseitig erweitert und verbessert [7, 17, 16, 8]. Die Anzahl der Iterationen der binären Amplitudenverstärkung liegt in $\mathcal{O}(\sqrt{N})$ [8]. Ein wichtiger Faktor in Bezug auf die Effizienz des Algorithmus ist das Erfüllen der sogenannten Grover-Phase-Matching-Bedingung, auf die wir am Ende dieses Kapitels weiter eingehen werden.

Bei der Amplitudenverstärkung wird jede Eingabe als ein Zustand $|x\rangle$ in einem aus n Qubits zusammengesetzten System betrachtet. Diese Zustände $|0\rangle$, $|1\rangle$, ..., $|N-1\rangle$ mit $N = 2^n$ bilden eine Orthogonalbasis des n-Qubitsystems.

Der verallgemeinerte Grover-Algorithmus [8] zur Lösung des binären Suchproblems besteht im Wesentlichen aus dem iterativen Anwenden zweier Operationen – ein sogenannter *Orakelaufruf* und eine *Diffusion*. Diese verändern einen initialisierten Startzustand, so dass am Ende die Basiszustände der Lösungseingaben mit einer höheren Wahrscheinlichkeit als die anderen Basiszustände gemessen werden.

Startzustand. Das n-Qubitsystem wird in den perfekt überlagerten Zustand initialisiert. Hierfür wird in der Regel auf jedes Qubit ein Hadamard-Gatter angewendet. Der Startzustand ist dann

$$|\Psi_0\rangle = H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle . \tag{3.2}$$

Quantenorakel. Zusätzlich wird ein *Quantenorakel* definiert. Dies wird oft auch abgekürzt nur Orakel genannt.

Definition 3.1 (Quantenorakel). Sei $n \in \mathbb{N}$ und $N := 2^n$ sowie $\varphi : \{|0\rangle, ..., |N-1\rangle\} \to \mathbb{R}$ eine Funktion auf $\{|0\rangle, ..., |N-1\rangle\}$. Das Quantenorakel U_φ auf dem n-Qubitsystem ist die Transformation

$$U_\varphi := \sum_{x=0}^{N-1} \mathrm{e}^{\mathrm{i}\varphi(x)} |x\rangle \langle x| . \tag{3.3}$$

Das Quantenorakel markiert jeden Basiszustand $|x\rangle$ des n-Qubitsystems mit einer Phase abhängig von $\varphi(x)$: Es gilt

$$U_\varphi |x\rangle = \mathrm{e}^{\mathrm{i}\varphi(x)} |x\rangle = (\cos(\varphi(x)) + \mathrm{i}\sin(\varphi(x))) |x\rangle , \tag{3.4}$$

und aus (3.3) folgt für U_φ die Matrixdarstellung

$$U_\varphi = \begin{pmatrix} \mathrm{e}^{\mathrm{i}\varphi(0)} & 0 & ... & ... & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\varphi(1)} & 0 & ... & ... \\ ... & 0 & ... & 0 & ... \\ ... & ... & 0 & \mathrm{e}^{\mathrm{i}\varphi(N-2)} & 0 \\ 0 & ... & ... & 0 & \mathrm{e}^{\mathrm{i}\varphi(N-1)} \end{pmatrix} . \tag{3.5}$$

Das Orakel ist der Schlüsselbaustein des Algorithmus. Es wird problemspezifisch implementiert; eine effiziente Implementierung ist ausschlaggebend für die Effizienz des Gesamtverfahrens.

Für den verallgemeinerten Grover-Algorithmus [8] nimmt φ die Werte 0 oder π an. Damit markiert das Orakel die zu den Lösungen gehörenden Basiszustände mit einer Phasenverschiebung von 180°:

$$\begin{aligned} U_\varphi |x\rangle &= \begin{cases} \mathrm{e}^{\mathrm{i}\pi} |x\rangle , & x \text{ ist Lösung,} \\ |x\rangle , & \text{sonst,} \end{cases} \\ &= \begin{cases} -|x\rangle , & x \text{ ist Lösung,} \\ +|x\rangle , & \text{sonst.} \end{cases} \end{aligned} \tag{3.6}$$

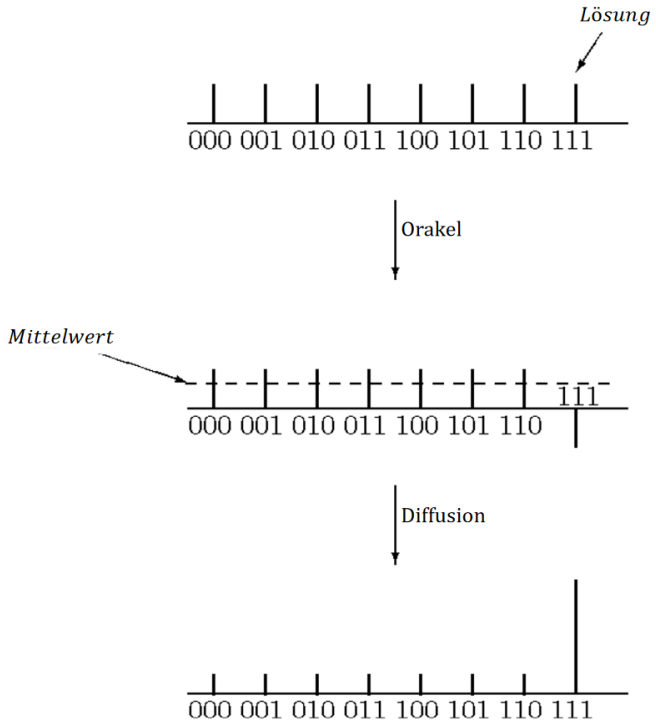


Abbildung 3.1: Vereinfachte Darstellung der Amplituden der Basiszustände beim Grover-Algorithmus, übernommen aus [39] und angepasst. Die Grafik zeigt die Änderung der Amplituden bei der ersten Anwendung des Grover-Operators. In diesem Beispiel hat das binäre Suchproblem die (einzige) Lösung 111. **Oben:** Das System ist im perfekt überlagerten Zustand initialisiert. **Mitte:** Das Orakel wurde auf den Zustand angewandt, wodurch die Amplitude des zur Lösung gehörenden Basiszustands negiert wurde. Dies entspricht einer Phasenverschiebung von 180°. Der Mittelwert hat sich durch die Anwendung des Orakels verringert. **Unten:** Im Diffusionsschritt werden die Amplituden am Mittelwert gespiegelt. Dadurch wird insgesamt die Amplitude der Lösung vergrößert und die restlichen Amplituden werden verringert.

Daher ergibt sich die Wahl von $\varphi := \pi f(x)$ direkt aus der Problembeschreibung (3.1). Damit gilt:

$$U_\varphi |x\rangle = \mathrm{e}^{\mathrm{i}\varphi(x)} |x\rangle = \mathrm{e}^{\mathrm{i}\pi f(x)} |x\rangle = (-1)^{f(x)} |x\rangle = \begin{cases} -|x\rangle, & x \text{ ist Lösung,} \\ +|x\rangle, & \text{sonst.} \end{cases} \tag{3.7}$$

Diffusion. Der zweite zentrale Baustein des Grover-Algorithmus neben dem Quantenorakel ist der *Diffusionsoperator* S_{Ψ_0}, der die Amplituden der Basiszustände am Mittelwert der Amplituden spiegelt:

$$S_{\Psi_0} := 2 |\Psi_0\rangle \langle\Psi_0| - I_N. \tag{3.8}$$

Dabei ist $|\Psi_0\rangle$ der in (3.2) eingeführte Startzustand.

Definition 3.2 (*Grover-Operator*). Der *Grover-Operator* G ist definiert als die Verkettung der Operatoren S_{Ψ_0} und U_φ:

$$G := S_{\Psi_0} U_\varphi. \tag{3.9}$$

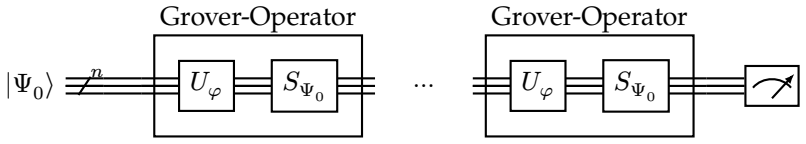


Abbildung 3.2: Quantenschaltkreis des verallgemeinerten Grover-Algorithmus (Algorithmus 1).

Welchen Einfluss der Grover-Operator auf die Amplituden der Basiszustände hat, ist in Abbildung 3.1 veranschaulicht. Der verallgemeinerte Grover-Algorithmus [8] ist folgendermaßen definiert:

Algorithmus 1 Verallgemeinerter Grover-Algorithmus

Eingabe: Anzahl der Iterationen $K \geq 1$
Initialisiere das n-Qubitsystem in den Startzustand $|\Psi_0\rangle$
for $i := 1, ...K$ **do**
$|\Psi_i\rangle \leftarrow G\,|\Psi_{i-1}\rangle$
end for
Messe $|\Psi_K\rangle$ in der Rechenbasis

Der Schaltkreis des verallgemeinerten Grover-Algorithmus ist in Abbildung 3.2 dargestellt. Die abschließende Messung liefert einen Messwert $x \in \{0, ..., 2^n - 1\}$, der idealerweise $f(x) = 1$ erfüllt und somit das Problem löst. Ist $f(x) = 0$, wird der Algorithmus so oft erneut ausgeführt, bis eine Lösung des binären Suchproblems gefunden wurde. Wie sicher der verallgemeinerte Grover-Algorithmus eine Lösung des Suchproblems findet, ist abhängig von der Anzahl der Lösungen [39]. Die Anzahl der Iterationen K hängt ebenfalls von der Anzahl der Lösungen ab. Dies wird in Abschnitt 3.4 betrachtet.

3.3 Darstellung des Grover-Operators als Rotation

Der Grover-Operator kann auch als Rotation im Unterraum der überlagerten Nicht-Lösungen und der überlagerten Lösungen interpretiert werden [1, 35]. Hierfür wird der Startzustand $|\Psi_0\rangle$ umgeformt. Sei M die Anzahl der Lösungen. Es gilt

$$\begin{aligned} |\Psi_0\rangle &= \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \\ &= \sqrt{\frac{N-M}{N}} \left(\frac{1}{\sqrt{N-M}} \sum_{x:f(x)=0} |x\rangle \right) + \sqrt{\frac{M}{N}} \left(\frac{1}{\sqrt{M}} \sum_{x:f(x)=1} |x\rangle \right) \end{aligned} \tag{3.10}$$

Seien nun $|\overline{m}\rangle$ die Überlagerung aller Nicht-Lösungen und $|m\rangle$ die Überlagerung aller Lösungen:

$$|\overline{m}\rangle := \frac{1}{\sqrt{N-M}} \sum_{x:f(x)=0} |x\rangle\,, \quad |m\rangle := \frac{1}{\sqrt{M}} \sum_{x:f(x)=1} |x\rangle\,. \tag{3.11}$$

Da

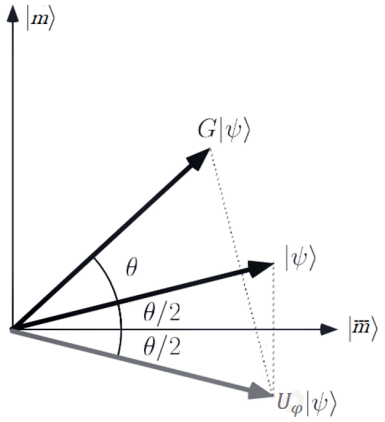


Abbildung 3.3: Der Grover-Operator kann als Rotation im Unterraum der überlagerten Nicht-Lösungen $|\overline{m}\rangle$ und der überlagerten Lösungen $|m\rangle$ interpretiert werden, übernommen aus [35]. Der Startzustand ist in dieser Abbildung $|\psi\rangle$. Jede Anwendung des Grover-Operators rotiert den Zustand des Systems um einen den Winkel θ von $|\overline{m}\rangle$ näher zu $|m\rangle$.

$$\left(\sqrt{\frac{N-M}{N}}\right)^2 + \left(\sqrt{\frac{M}{N}}\right)^2 = 1 \tag{3.12}$$

und die Ausdrücke in den Klammern beide nichtnegativ sind, existiert ein $\theta \in [0, \pi]$, so dass

$$\sqrt{\frac{N-M}{N}} = \cos(\tfrac{\theta}{2}) \text{ und } \sqrt{\frac{M}{N}} = \sin(\tfrac{\theta}{2}). \tag{3.13}$$

Setzen wir dies mit (3.11) in (3.10) ein, folgt für den Startzustand

$$|\Psi_0\rangle = \cos(\tfrac{\theta}{2})\,|\overline{m}\rangle + \sin(\tfrac{\theta}{2})\,|m\rangle\,. \tag{3.14}$$

Der Grover-Operator wirkt auf dem durch $|\overline{m}\rangle$ und $|m\rangle$ aufgespannten Unterraum wie eine Rotation und hat dort bezüglich der Basis $\{|\overline{m}\rangle\,, |m\rangle\}$ die Darstellung

$$G = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}. \tag{3.15}$$

Veranschaulicht ist diese Rotation in Abbildung 3.3. Da wir mit $|\Psi_0\rangle$ im von $|\overline{m}\rangle$ und $|m\rangle$ aufgespannten Unterraum starten, rotieren die ersten Anwendungen des Grover-Operators den Zustand des Systems um den Winkel θ von $|\overline{m}\rangle$ näher zu $|m\rangle$. Der Grover-Operator sollte daher genau so oft angewendet werden, bis der Zustand des Systems sich nicht mehr $|m\rangle$ annähert.

3.4 Komplexität des Algorithmus

Bei der Untersuchung des Aufwands des verallgemeinerten Grover-Algorithmus interessiert uns vor allem die Anzahl der Orakel-Aufrufe. Diese ist äquivalent zur Anzahl

der Iterationen K. Seien M und N die Anzahl der Lösungen beziehungsweise die Gesamtanzahl der Eingaben des Suchproblems wie in Abschnitt 3.3. Betrachten wir den Grover-Operator als Rotation im Unterraum der überlagerten Nicht-Lösungen und der überlagerten Lösungen, folgt, dass

$$K \leq \left\lceil \frac{\frac{\pi}{2}}{\theta} \right\rceil \tag{3.16}$$

eine Bedingung für die Anzahl der Iterationen ist. Ansonsten würde der Zustand sich nicht mehr $|m\rangle$ annähern. Ist $M \leq \frac{N}{2}$, kann θ mit $\frac{\theta}{2} \geq \sin(\frac{\theta}{2}) = \sqrt{\frac{M}{N}}$ abgeschätzt werden [39]. Damit ergibt sich

$$K \leq \left\lceil \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rceil \in \mathcal{O}\left(\sqrt{\frac{N}{M}}\right). \tag{3.17}$$

Da M oft nicht bekannt ist und es im schlimmsten Fall nur eine Lösung gibt, liegt die Anzahl der Iterationen K mindestens in $\mathcal{O}(\sqrt{N})$ [8]. Wenn $M > \frac{N}{2}$ ist, folgt $K = 0$. Das heißt, der Grover-Operator wird nicht angewendet und x wird zufällig gewählt.

Für die Gesamtkomplexität in Bezug auf die Orakel-Aufrufe reicht es aber nicht, die Komplexität von einem Algorithmusdurchlauf zu betrachten, weil der Algorithmus, nur mit einer gewissen Wahrscheinlichkeit die Lösung des Problems findet. Der Algorithmus wird demnach so oft erneut ausgeführt, bis eine Lösung, das heißt ein x mit $f(x) = 1$, gefunden wurde. Die Wahrscheinlichkeit, die Lösung des Suchproblems nach einmaligem Ausführen des verallgemeinerten Grover-Algorithmus zu messen, hat $1 - \frac{M}{N}$ als untere Schranke [39]. Das heißt, je weniger Lösungen es gibt, desto höher ist die Wahrscheinlichkeit, nach einmaligem Ausführen des Algorithmus die Lösung zu messen. Mit der unteren Schranke der Wahrscheinlichkeit lässt sich der Erwartungswert der Anzahl der Algorithmusdurchläufe berechnen. Dieser ist $\frac{N}{N-M}$. Das heißt, die Gesamtanzahl der Orakel-Aufrufe liegt in $\mathcal{O}(\sqrt{N}\frac{N}{N-M})$.

3.5 Phase-Matching

Der verallgemeinerte Grover-Algorithmus lässt sich weiter verallgemeinern [16, 31, 18]. Zum einen kann die Phasenverschiebung beliebig gewählt werden. Für das Orakel U_φ gilt dann mit $\varphi \in [0, \pi]$:

$$U_\varphi |x\rangle = \begin{cases} e^{i\varphi} |x\rangle, & x \text{ ist Lösung,} \\ |x\rangle, & \text{sonst.} \end{cases} \tag{3.18}$$

Außerdem kann die Diffusion S_{Ψ_0} aus (3.8) zu

$$S_{\Psi_0} = (1 - e^{i\gamma}) |\Psi_0\rangle \langle\Psi_0| - I_N \tag{3.19}$$

mit $\gamma \in [0, \pi]$ verallgemeinert werden. Damit ergeben sich zwei weitere Freiheitsgrade: $\varphi \in [0, \pi]$ und $\gamma \in [0, \pi]$. Es wurde vielseitig untersucht, welche Wahl von φ und γ sinnvoll ist und sogenannte Phase-Matching-Bedingungen vorgestellt [31, 18, 22, 9].

In dem in diesem Kapitel vorgestellten verallgemeinerten Grover-Algorithmus (Algorithmus 1) gilt $\varphi = \gamma = \pi$, zusammengefasst in der klassischen Grover-Phase-Matching-Bedingung:

Definition 3.3 (Grover-Phase-Matching-Bedingung). Die Grover-Phase-Matching-Bedingung ist erfüllt, wenn für die Winkel φ und γ des Orakels U_φ aus (3.18) und der Diffusion S_{Ψ_0} aus (3.19)

$$\varphi = \gamma = \pi \tag{3.20}$$

gilt.

Sind die Phasen ungleich, kann kein effizienter Quanten-Suchalgorithmus konstruiert werden, so Long et al. [31]. Ein Algorithmus wird als effizient angesehen, wenn der Suchalgorithmus die Lösung mit einer hohen Wahrscheinlichkeit findet. Wenn die Grover-Phase-Matching-Bedingung erfüllt ist, wächst die Wahrscheinlichkeit, die Lösung des Suchproblems zu messen, pro Iteration am stärksten im Vergleich zu Phasen ungleich π [18, 38]. Sind die Phasen ungleich π, kann die Verkettung der Operatoren S_{Ψ_0} und U_φ nicht mehr als eine Rotation in dem von $|\overline{m}\rangle$ und $|m\rangle$ aufgespannten Unterraum (siehe Abschnitt 3.3) interpretiert werden. Da $e^{i\varphi}$ für $\varphi \in (0, \pi)$ nicht reell ist, muss der Unterraum mit einer komplexen Dimension erweitert werden. Intuitiv gilt daher: Bei jeder Iteration nähert sich der Zustand nicht auf direktem Weg der Überlagerung der Lösungen an, sondern in Spiralen.

Wie nah der Zustand des Systems an die Überlagerung aller Lösungen durch den Grover-Operator im Fall $\varphi = \gamma = \pi$ rotiert werden kann, ist abhängig von dem Verhältnis der Anzahl der Lösungen zu der Gesamtanzahl der Eingaben $\frac{M}{N}$ (mit M und N aus Abschnitt 3.3). Existiert ein $k \in \mathbb{N}_0$, so dass $(2k+1)\arcsin(\sqrt{M/N}) = \pi/2$, kann direkt zur Überlagerung aller Lösungen rotiert werden. Existiert kein solches k, gilt: Je kleiner $\frac{M}{N}$ ist, desto näher kann der Zustand des Systems an die Überlagerung aller Lösungen rotiert werden. Ist $\frac{M}{N}$ zu groß, kann mit Phasen ungleich π der Zustand des Systems näher an die Überlagerung aller Lösungen rotiert werden. Dann sind aber mehr Iterationen nötig. Für diesen Fall wurden viele alternative Phase-Matching-Bedingungen vorgestellt [38, 30, 9, 22]. Eine höhere Wahrscheinlichkeit, die Lösung des Suchproblems zu messen, im Vergleich zur klassischen Grover-Phase-Matching-Bedingung liefern diese ab $\frac{M}{N} > 0.3$.

Die klassische Grover-Phase-Matching-Bedingung ist also eine sinnvolle Bedingung für Suchalgorithmen unter der Annahme, dass für das Suchproblem $\frac{M}{N} \leq 0.3$ gilt.

Zusammenfassung

Der verallgemeinerte Grover-Algorithmus kann zum Lösen binärer Suchprobleme benutzt werden. Die Anzahl der Orakel-Aufrufe pro Durchführung des Algorithmus liegt in $\mathcal{O}(\sqrt{N})$. Die Effektivität des Algorithmus ist unter anderem gegeben, weil die Grover-Phase-Matching-Bedingung erfüllt wurde. Da der Algorithmus ein binäres Suchproblem löst, kann der Grover-Operator auch als Rotation im Unterraum der überlagerten Nicht-Lösungen und der überlagerten Lösungen betrachtet werden. Das Ising-Problem ist aber ein nicht-binäres Suchproblem. Im folgenden Kapitel wird daher die nicht-binäre Amplitudenverstärkung betrachtet.

4

Nicht-binäre Amplitudenverstärkung

Der Ansatz der Amplitudenverstärkung kann auch zum Lösen *nicht-binärer Suchprobleme* benutzt werden. Hierbei wird der verallgemeinerte Grover-Algorithmus so erweitert, dass anstelle des binären Orakels ein nicht-binäres Orakel verwendet werden kann. Dieses Verfahren wird *nicht-binäre Amplitudenverstärkung* genannt; in der Literatur wird auch der Begriff *nicht-boolesche Amplitudenverstärkung* verwendet.

Koch et al. stellen in [27] eine Methode der nicht-binären Amplitudenverstärkung vor, um kombinatorische Optimierungsprobleme zu lösen. Allerdings erfordert das Verfahren eine Kombination aus Quantencomputing und klassischem Computing. Shyamsundar et al. stellen in [37] eine andere Methode der nicht-binären Amplitudenverstärkung vor, die allein auf Quantencomputing basiert.

Im folgenden Kapitel wird der Algorithmus von Shyamsundar et al. vorgestellt. Dieser wird im weiteren Verlauf als NBAA bezeichnet. Zuerst werden das Suchproblem sowie die benutzten Operatoren behandelt (Abschnitte 4.1 und 4.2). Anschließend wird das Vorgehen des Algorithmus erklärt und die Dynamik des Algorithmus sowie die optimale Anzahl an Iterationen untersucht (Abschnitte 4.4 und 4.5). Schließlich wird der Zusammenhang zwischen der binären und nicht-binären Amplitudenverstärkung hergestellt (Abschnitt 4.3). Das gesamte Kapitel orientiert sich an [37].

4.1 Nicht-binäres Suchproblem

Gegeben sei eine zu maximierende Funktion

$$\varphi : \{0, ..., N-1\} \to [0, \pi], \tag{4.1}$$

wobei N als Zweierpotenz $N = 2^n$ mit $n \in \mathbb{N}$ geschrieben werden kann, und die Möglichkeit, ein Orakel U_φ mit

$$U_\varphi \cdot |x\rangle = e^{i\varphi(x)} |x\rangle \tag{4.2}$$

als Quantenschaltkreis zu implementieren.

Das Ziel der nicht-binären Amplitudenverstärkung [37] ist es, ausgehend von einer Überlagerung von Basiszuständen sukzessive die Amplituden des Basiszustands $|x\rangle$ umso stärker zu vergrößern, je besser x das Problem löst, das heißt, je größer $\varphi(x)$ ist.

Die Funktion $\varphi(x)$ und das Orakel U_φ werden problemspezifisch konstruiert. Sei beispielsweise eine Funktion $f : \{0, ..., N-1\} \to \mathbb{R}$ gegeben, für die das $x \in \{0, .., N-1\}$ gesucht ist, bei dem der Wert $f(x)$ der 1 am nächsten kommt. Hier wäre eine Möglichkeit

$$\varphi(x) := \pi \mathrm{e}^{-(1-f(x))^2}. \tag{4.3}$$

Bei $f(x) = 1$ ist dann $\mathrm{e}^{-(1-f(x))^2} = \mathrm{e}^0 = 1$, also $\varphi(x) = \pi$. Je größer $(1 - f(x))^2$ wird, desto kleiner werden $\mathrm{e}^{-(1-f(x))^2}$ und folglich auch $\varphi(x)$.

Anders als im binären Fall, wird hier nicht nur zwischen Lösung und Nicht-Lösung unterschieden: Es kann graduell markiert werden, wie gut eine Lösung ist. In der Praxis ist eine scharfe Lösung/Nicht-Lösung Entscheidung oft nicht möglich, weshalb ein Verfahren, das Abstufungen machen kann, viele praxisrelevante Anwendungen hat [3, 36].

4.2 Aufbau des Algorithmus

Der Quantenschaltkreis des Algorithmus aus [37] besteht aus zwei Registern. Das erste Register besteht lediglich auch einem *Hilfsqubit* $|h\rangle$, das zweite Register umfasst n Qubits.

Die orthogonale Basis des zweiten Registers ist $\{|0\rangle, .., |N-1\rangle\}$ mit $N = 2^n$. Die Basis des Zwei-Registersystems wird mit $\{|0,0\rangle, .., |0,N-1\rangle, |1,0\rangle, .., |1,N-1\rangle\}$ bezeichnet. Das Hinzufügen eines Hilfsqubits verdoppelt die Dimension des Zustandsraumes.

Startzustand. Der Startzustand des zweiten Registers wird wieder mit $|\Psi_0\rangle$ bezeichnet und A_0 ist der unitäre Operator, der das System von $|0\rangle$ in den Startzustand überführt. Es gilt also

$$|\Psi_0\rangle = A_0 |0\rangle = \sum_{x=0}^{N-1} a_0(x) |x\rangle \quad \text{mit} \sum_{x=0}^{N-1} |a_0(x)|^2 = 1. \tag{4.4}$$

Dabei ist $a_0(x)$ die Startamplitude des Basiszustands $|x\rangle$. Dieser Startzustand kann beliebig gewählt werden. Gibt es kein Vorwissen zum Suchproblem, bietet sich der perfekt überlagerte Zustand als Startzustand an. Das Hilfsqubit $|h\rangle$ (auch ancilla genannt) wird mit $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ initialisiert. Insgesamt ist der Startzustand damit:

$$\begin{aligned} |\Psi_0\rangle &= [H \otimes A_0] |0,0\rangle \\ &= \frac{|0,\Psi_0\rangle + |1,\Psi_0\rangle}{\sqrt{2}} \\ &= \frac{1}{\sqrt{2}} \sum_{x=0}^{N-1} a_0(x) \left(|0,x\rangle + |1,x\rangle\right). \end{aligned} \tag{4.5}$$

Wie bei der binären Amplitudenverstärkung (Kapitel 3) werden auch bei der nicht-binären Amplitudenverstärkung eine Diffusion und ein Orakelaufruf verwendet, um die Amplituden der Basiszustände zu beeinflussen.

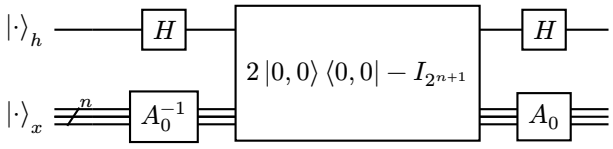
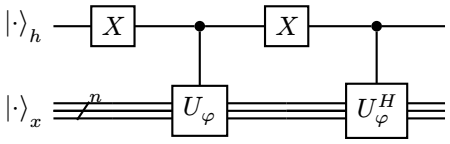
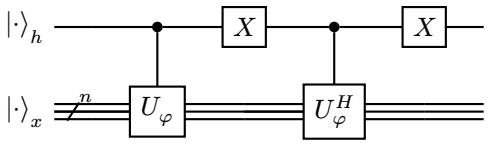


Abbildung 4.1: Schaltkreis des Diffusionsoperators $\mathbf{S}_{\Psi_0}$ für die nicht-binäre Amplitudenverstärkung.



(a) Schaltkreis des 2-Register Orakels $\mathbf{U}_\varphi$ im nicht-binären Fall. Das Hilfsqubit $|h\rangle$ fungiert als Kontrollqubit.



(b) Schaltkreis von $\mathbf{U}_\varphi^H$ – dem Inversen von $\mathbf{U}_\varphi$. Im Vergleich zu $\mathbf{U}_\varphi$ ändert sich nur die Reihenfolge der Gatter.

Abbildung 4.2: Schaltkreise von $\mathbf{U}_\varphi$ und $\mathbf{U}_\varphi^H$.

Diffusion. Die Diffusion wird wie folgt auf zwei Register erweitert:

$$\begin{aligned}\mathbf{S}_{\Psi_0} &= 2\,|\Psi_0\rangle\,\langle\Psi_0|\, - I_{2^{n+1}} \\ &= [H \otimes A_0^{-1}]\,(2\,|0,0\rangle\langle 0,0| - I_{2^{n+1}})\,[H \otimes A_0].\end{aligned} \tag{4.6}$$

Der zugehörige Schaltkreis ist in Abbildung 4.1 dargestellt.

2-Register-Orakel. Außerdem wird ein 2-Register-Orakel $\mathbf{U}_\varphi$ verwendet. Dies ist ein bedingter Orakelaufruf des Orakels $U_\varphi = \sum_{x=0}^{N-1} \mathrm{e}^{\mathrm{i}\varphi(x)}\,|x\rangle\langle x|$ (siehe auch Definition 3.1) und dem inversen Operator von U_φ. Der adjungierte Operator von U_φ ist U_φ^H. Es gilt $U_\varphi^H = \sum_{x=0}^{N-1} \mathrm{e}^{-\mathrm{i}\varphi(x)}\,|x\rangle\langle x|$ und $U_\varphi U_\varphi^H = U_\varphi^H U_\varphi = I_N$. Damit ist U_φ unitär und der inverse Operator von U_φ ist U_φ^H. Das Hilfsqubit $|h\rangle$ fungiert als Kontrollqubit:

$$\begin{aligned}\mathbf{U}_\varphi &:= |0\rangle\langle 0| \otimes U_\varphi + |1\rangle\langle 1| \otimes U_\varphi^H \\ &= \begin{pmatrix} U_\varphi & 0 \\ 0 & U_\varphi^H \end{pmatrix}.\end{aligned} \tag{4.7}$$

Es gilt also für alle $x \in \{0, \ldots, N-1\}$:

$$\mathbf{U}_\varphi\,|0,x\rangle = \mathrm{e}^{\mathrm{i}\varphi(x)}\,|0,x\rangle \text{ und} \tag{4.8}$$

$$\mathbf{U}_\varphi\,|1,x\rangle = \mathrm{e}^{-\mathrm{i}\varphi(x)}\,|1,x\rangle\,. \tag{4.9}$$

Der zugehörige inverse Operator ist

$$\begin{aligned}\mathbf{U}_\varphi^H &= |0\rangle\langle 0| \otimes U_\varphi^H + |1\rangle\langle 1| \otimes U_\varphi \\ &= \begin{pmatrix} U_\varphi^H & 0 \\ 0 & U_\varphi \end{pmatrix},\end{aligned} \tag{4.10}$$

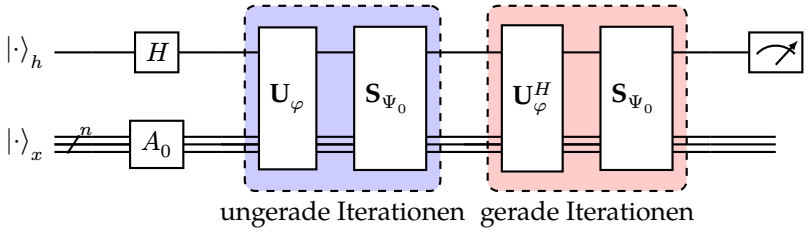


Abbildung 4.3: Quantenschaltkreis der nicht-binären Amplitudenverstärkung (Algorithmus 2), angelehnt an die Abbildung 4 aus [37]. Nach der Initialisierung werden alternierend $\mathbf{S}_{\Psi_0}\,\mathbf{U}_\varphi$ und $\mathbf{S}_{\Psi_0}\,\mathbf{U}_\varphi^H$ auf das System angewendet. Am Ende wird das Hilfsqubit gemessen.

und für alle $x \in \{0, ..., N-1\}$ gelten

$$\mathbf{U}_\varphi^H |0,x\rangle = \mathrm{e}^{-\mathrm{i}\varphi(x)} |0,x\rangle \text{ sowie} \tag{4.11}$$

$$\mathbf{U}_\varphi^H |1,x\rangle = \mathrm{e}^{\mathrm{i}\varphi(x)} |1,x\rangle\,. \tag{4.12}$$

Die Schaltkreise von $\mathbf{U}_\varphi$ und $\mathbf{U}_\varphi^H$ sind in Abbildung 4.2 dargestellt.

Mit diesen Definitionen ist der Gesamtalgorithmus folgendermaßen definiert [37]:

Algorithmus 2 Nicht-binäre Amplitudenverstärkung (NBAA)

Eingabe: Anzahl der Iterationen $K \geq 1$
Initialisiere das Zwei-Register-Qubitsystem in den Startzustand $|\Psi_0\rangle$
for $i := 1, ...K$ **do**
  if i ungerade **then**
    $|\Psi_i\rangle \leftarrow \mathbf{S}_{\Psi_0}\,\mathbf{U}_\varphi\,|\Psi_{i-1}\rangle$
  else
    $|\Psi_i\rangle \leftarrow \mathbf{S}_{\Psi_0}\,\mathbf{U}_\varphi^H\,|\Psi_{i-1}\rangle$
  end if
end for
Messe $|h\rangle$und $|x\rangle$

Abbildung 4.3 zeigt den Schaltkreis von NBAA. In Abschnitt 4.4 werden wir sehen, dass die Iterationen die Amplituden der Basiszustände $|0,x\rangle$ und $|1,x\rangle$ umso stärker verstärken, je besser x das Problem löst; also umso stärker, je kleiner $\cos(\varphi(x))$ ist. Die optimale Anzahl an Iterationen wird in Abschnitt 4.5 näher betrachtet. Das Messen des Hilfsqubits stellt sicher, dass die beiden Register nicht mehr verschränkt sind. Wie beim verallgemeinerten Grover-Algorithmus aus Kapitel 3 wird am Ende auch das zweite Register gemessen. Der Messwert x steht für die Eingabe x.

4.3 Zusammenhang mit binärer Amplitudenverstärkung

Wie schon in der Einleitung des Kapitels angekündigt, ist NBAA eine Erweiterung des verallgemeinerten Grover-Algorithmus und sollte daher auch zum Lösen binärer Probleme genutzt werden können. In diesem Abschnitt werden die beiden Algorithmen kurz miteinander verglichen.

Um mit NBAA [37] binäre Probleme (siehe Kapitel 3) zu lösen, kann $\varphi(x) = \pi f(x)$ gewählt werden. Da für alle $f(x) \in \{0, 1\}$

$$e^{i\varphi(x)} = e^{i\pi f(x)} = e^{-i\pi f(x)} = e^{-i\varphi(x)} \tag{4.13}$$

gilt, folgt $U_\varphi = U_\varphi^H$ und somit auch $\mathbf{U}_\varphi = \mathbf{U}_\varphi^H$. Daher muss nicht mehr zwischen geraden und ungeraden Iterationen unterschieden werden. In Algorithmus 2 wird dies natürlich immer noch gemacht. Die Hauptunterschiede der NBAA mit einem binären Problem zum verallgemeinerte Grover-Algorithmus aus Kapitel 3 sind somit:

1. Es wird ein nicht-binäres Orakel im Gegensatz zu einem binären Orakel verwendet.
2. Es wird ein extra Hilfsqubit $|h\rangle$ verwendet.
3. Es wird theoretisch zwischen geraden und ungeraden Iterationen unterschieden, im binären Fall fallen diese aber zusammen.

4.4 Analyse des Algorithmus

Wir untersuchen nun das Iterationsverhalten des Algorithmus. Hierfür bezeichne $|\Psi_k\rangle$ den Zustand des 2-Registersystems nach k Iterationen. Für alle $k \in \{1, .., K\}$ gilt:

$$|\Psi_k\rangle = \begin{cases} \mathbf{S}_{\Psi_0}\, \mathbf{U}_\varphi\, |\Psi_{k-1}\rangle\,, & k \text{ gerade,} \\ \mathbf{S}_{\Psi_0}\, \mathbf{U}_\varphi^H\, |\Psi_{k-1}\rangle\,, & k \text{ ungerade.} \end{cases} \tag{4.14}$$

Seien $\tilde{a}_k(0, x)$ und $\tilde{a}_k(1, x)$ die Amplituden von $|0, x\rangle$ beziehungsweise $|1, x\rangle$ im überlagerten Zustand $|\Psi_k\rangle$. Auch bei diesen Amplituden gilt: $\sum_x |\tilde{a}_k(0, x)|^2 + |\tilde{a}_k(1, x)|^2 = 1$. Dann ist

$$|\Psi_k\rangle = \sum_{x=0}^{N-1} \tilde{a}_k(0, x)\, |0, x\rangle + \tilde{a}_k(1, x)\, |1, x\rangle\,. \tag{4.15}$$

Für $|\Psi_0\rangle$ gilt:

$$\tilde{a}_0(0, x) = \tilde{a}_0(1, x) = \frac{a_0(x)}{\sqrt{2}}. \tag{4.16}$$

Sei $\theta \in [0, \pi]$ der Winkel, so dass

$$\cos(\theta) := \sum_{x=0}^{N-1} |a_0(x)|^2\, \cos(\varphi(x)). \tag{4.17}$$

Die $\cos(\varphi(x))$ liegen im Intervall $[-1, 1]$ und $\cos(\theta)$ ist eine Konvexkombination dieser. Somit liegt auch $\cos(\theta)$ in $[-1, 1]$. Der Winkel $\theta \in [0, \pi]$ ist wohldefiniert, existiert und ist eindeutig, da die Kosinusfunktion $\cos : [0, \pi] \to [-1, 1]$ bijektiv ist. Der Erwartungswert des Ausdrucks $\varphi(x)$, wenn x im Startzustand $|\Psi_0\rangle$ gemessen würde, ist genau $\cos(\theta)$. Zusätzlich werden

$$|\alpha\rangle := \mathbf{U}_\varphi\, |\Psi_0\rangle = \frac{1}{\sqrt{2}} \sum_{x=0}^{N-1} a_0(x) \left[e^{i\varphi(x)}\, |0, x\rangle + e^{-i\varphi(x)}\, |1, x\rangle \right] \text{ und} \tag{4.18}$$

$$|\beta\rangle := \mathbf{U}_\varphi^H\, |\Psi_0\rangle = \frac{1}{\sqrt{2}} \sum_{x=0}^{N-1} a_0(x) \left[e^{-i\varphi(x)}\, |0, x\rangle + e^{i\varphi(x)}\, |1, x\rangle \right] \tag{4.19}$$

definiert. Sowohl θ als auch $|\alpha\rangle$ und $|\beta\rangle$ hängen von der Funktion φ und dem Startzustand $|\Psi_0\rangle$ ab.

Erste Iteration. Nach der ersten Iteration gilt:

$$\begin{aligned}
|\Psi_1\rangle &= \mathbf{S}_{\Psi_0}\, \mathbf{U}_\varphi\, |\Psi_0\rangle \\
&= \mathbf{S}_{\Psi_0}\ |\alpha\rangle \\
&= (2\,|\Psi_0\rangle\ \langle\Psi_0|\ - I_{2^{n+1}})\,|\alpha\rangle \\
&= 2\,|\Psi_0\rangle\ \langle\Psi_0|\alpha\rangle - |\alpha\rangle \\
&= 2\,\langle\Psi_0|\alpha\rangle\,|\Psi_0\rangle\ - |\alpha\rangle\ .
\end{aligned} \tag{4.20}$$

Das innere Produkt $\langle\Psi_0|\alpha\rangle$ ist reell, da

$$\begin{aligned}
\langle\Psi_0|\alpha\rangle &= \left(\frac{1}{\sqrt{2}}\sum_{x=0}^{N-1}\overline{a_0(x)}\ (\langle 0,x| + \langle 1,x|)\right) \\
&\qquad \cdot \left(\frac{1}{\sqrt{2}}\sum_{x=0}^{N-1} a_0(x)\left[\mathrm{e}^{\mathrm{i}\varphi(x)}\,|0,x\rangle + \mathrm{e}^{-\mathrm{i}\varphi(x)}\,|1,x\rangle\right]\right) \\
&\overset{(*)}{=} \frac{1}{2}\sum_{x=0}^{N-1}|a_0(x)|^2\Big[\mathrm{e}^{\mathrm{i}\varphi(x)}(\underbrace{\langle 0,x|0,x\rangle}_{=1} + \underbrace{\langle 1,x|0,x\rangle}_{=0}) \\
&\qquad\qquad + \mathrm{e}^{-\mathrm{i}\varphi(x)}(\underbrace{\langle 0,x|1,x\rangle}_{=0} + \underbrace{\langle 1,x|1,x\rangle}_{=1})\Big] \\
&= \frac{1}{2}\sum_{x=0}^{N-1}|a_0(x)|^2\left[\mathrm{e}^{\mathrm{i}\varphi(x)} + \mathrm{e}^{-\mathrm{i}\varphi(x)}\right] \\
&= \frac{1}{2}\sum_{x=0}^{N-1}|a_0(x)|^2\ 2\ \cos(\varphi(x)) \\
&= \cos(\theta)
\end{aligned} \tag{4.21}$$

gilt und $\cos(\theta)$ nach Voraussetzung reell ist. Die Orthogonalität der Basiszustände dazu, dass die Summen zu einer zusammengefasst werden können. Diese Eigenschaft wird bei (*) verwendet. Die Tatsache, dass $\langle\Psi_0|\alpha\rangle$ reell ist, ermöglicht es, geschlossene mathematische Aussagen über die Konvergenz des Algorithmus zu treffen. Das Hilfsqubit sorgt dafür, dass $\langle\Psi_0|\alpha\rangle = \cos(\theta) \in \mathbb{R}$ gilt.

Wäre $\langle\Psi_0|\alpha\rangle$ beziehungsweise $\cos(\theta)$ komplex und nicht reell, wäre es auch möglich, das Suchproblem zu lösen [37, Abschnitt 6]. Dafür wäre kein Hilfsqubit mehr nötig, es würden aber andere zusätzliche Kosten anfallen. In diesem Fall ergäbe sich $\langle\Psi_0|\alpha\rangle = \cos(\theta)\mathrm{e}^{\mathrm{i}\delta}$ mit der zusätzlichen Phase $\delta \in [0, 2\pi)$. Das Iterationsverhalten hinge dann zusätzlich von δ ab und nicht nur von $\varphi(x)$. Der Winkel δ müsste zusätzlich zu $\cos(\theta)$ ermittelt werden und $\varphi(x)$ müsste abhängig von δ gewählt werden. Die mathematischen Ausdrücke wären demnach etwas komplizierter.

Wir betrachten daher nur NBAA mit einem Hilfsqubit und reellem $\langle\Psi_0|\alpha\rangle$. Für $|\Psi_1\rangle$ ergibt sich nun:

$$|\Psi_1\rangle = 2\cos(\theta)\,|\Psi_0\rangle\ - |\alpha\rangle\ . \tag{4.22}$$

Die Amplituden von $|0,x\rangle$ beziehungsweise $|1,x\rangle$ in der Superposition $|\Psi_1\rangle$ können wie folgt geschrieben werden:

$$\tilde{a}_1(0,x) = \langle 0,x|\Psi_1\rangle = \frac{a_0(x)}{\sqrt{2}}[2\cos(\theta) - e^{i\varphi(x)}] = \tilde{a}_0(0,x)[2\cos(\theta) - e^{i\varphi(x)}] \quad (4.23)$$

$$\begin{aligned}\tilde{a}_1(1,x) = \langle 1,x|\Psi_1\rangle &= \frac{a_0(x)}{\sqrt{2}}[2\cos(\theta) - e^{-i\varphi(x)}] \\ &= \tilde{a}_0(1,x)[2\cos(\theta) - e^{-i\varphi(x)}].\end{aligned} \quad (4.24)$$

Aus (4.23) und (4.24) kann die Änderungsrate der Amplituden ermittelt werden:

$$\frac{|\tilde{a}_1(0,x)|^2}{|\tilde{a}_0(0,x)|^2} = \frac{|\tilde{a}_1(1,x)|^2}{|\tilde{a}_0(1,x)|^2} = 4\cos^2(\theta) + 1 - 4\cos(\theta)\cos(\varphi(x)). \quad (4.25)$$

Da $\cos(\theta)$ konstant ist, folgt somit, dass die Höhe der Änderungsrate abhängig von $\varphi(x)$ beziehungsweise $\cos(\varphi(x))$ ist. Das heißt, wenn $\cos(\theta)$ positiv ist, verstärkt der Algorithmus genau die Amplituden der Basiszustände, für die $\cos(\varphi(x)) < \cos(\theta)$ gilt. Je kleiner $\cos(\varphi(x))$ ist, desto stärker werden die zu x gehörenden Amplituden verstärkt. Da $\cos(\varphi(x))$ streng monoton fallend auf $[0,\pi]$ ist – also bei $\varphi(x) = \pi$ das Minimum annimmt, werden die Amplituden je stärker verstärkt, desto besser die zugehörige Eingabe x das nicht-binäre Suchproblem (Abschnitt 4.1) löst. Im weiteren Verlauf wird vorausgesetzt, dass $\cos(\theta)$ positiv ist.

K-te Iteration. Führen wir den Algorithmus weiter aus, erhalten wir nach K Iterationen den Zustand

$$|\Psi_K\rangle = \begin{cases} \frac{1}{\sqrt{2}\sin(\theta)}\,[\sin((K+1)\theta)\,|\Psi_0\rangle - \sin(K\theta)e^{-i\varphi(x)}\,|\alpha\rangle\,], & K \text{ ungerade,} \\ \frac{1}{\sqrt{2}\sin(\theta)}\,[\sin((K+1)\theta)\,|\Psi_0\rangle - \sin(K\theta)e^{-i\varphi(x)}\,|\beta\rangle\,], & K \text{ gerade.} \end{cases} \quad (4.26)$$

und es ergibt sich für die Amplituden $|0,x\rangle$ beziehungsweise $|1,x\rangle$ mit $b \in \{0,1\}$:

$$\tilde{a}_K(b,x) = \begin{cases} \frac{a_0(x)}{\sqrt{2}\sin(\theta)}\,[\sin((K+1)\theta) - \sin(K\theta)e^{-i\varphi(x)}], & K+b \text{ gerade,} \\ \frac{a_0(x)}{\sqrt{2}\sin(\theta)}\,[\sin((K+1)\theta) - \sin(K\theta)e^{i\varphi(x)}], & K+b \text{ ungerade.} \end{cases} \quad (4.27)$$

Das Alternieren des 2-Register-Orakels mit dem Inversen des 2-Register-Orakels sorgt genau dafür, dass der Zustand des Systems immer im von $|\Psi_0\rangle$, $|\alpha\rangle$ und $|\beta\rangle$ aufgespannten Unterraum bleibt. Eine detaillierte Herleitung des Zustands $|\Psi_K\rangle$ und der dazugehörigen Amplituden ist in [37] zu finden.

Nachdem K-mal iteriert wurde, ist der nächste Schritt im Algorithmus, das Hilfsqubit zu messen. Dies wird gemacht, um sicherzustellen, dass die zwei Register $|h\rangle$und $|x\rangle$ nicht verschränkt sind. Für alle $x \in \{0,\ldots,N-1\}$ gilt:

$$|\tilde{a}_K(0,x)| = |\tilde{a}_K(1,x)|. \quad (4.28)$$

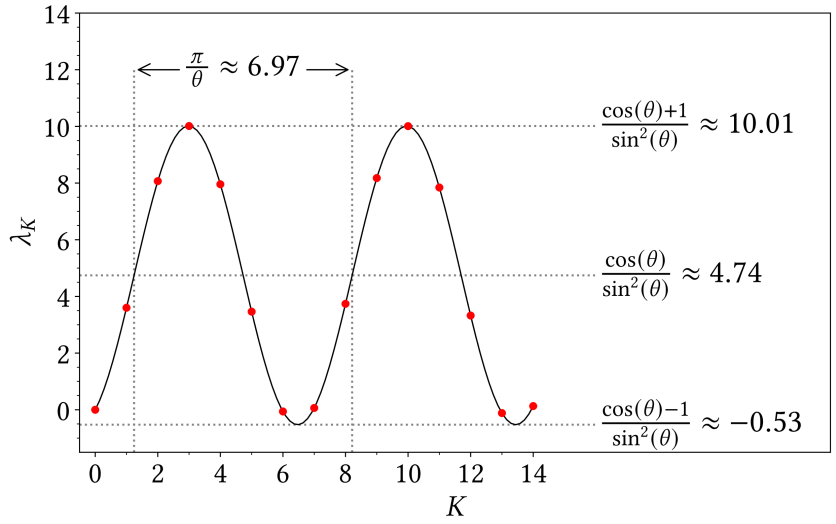


Abbildung 4.4: Der Verstärkungsfaktor λ_K aus (4.31) in Abhängigkeit von K, übernommen aus [37]. Der Verstärkungsfaktor λ_K ist eine oszillierende Funktion, die um $\cos(\theta)/\sin^2(\theta)$ zentriert ist. Außerdem ist λ_K π/θ-periodisch und hat eine Amplitude von $1/\sin^2(\theta)$.

Folglich werden beim Hilfsqubit „0" und „1" mit gleicher Wahrscheinlichkeit gemessen. Da die Basiszustände des zweiten Registers die Eingaben des Optimierungsproblems beschreiben, sind die Messwerte des Hilfsqubits nicht interessant für das Lösen des Problems, zumal diese die Amplituden der Basiszustände des zweiten Registers nicht beeinflussen.

Betrachten wir nun den Zustand nach der Messung. Sei $|\Psi_{K,b}\rangle$ der normalisierte Zustand des zweiten Registers nach der Messung des Hilfsqubits $|h\rangle$, bei der $b \in \{0,1\}$ erhalten wurde. Es gilt

$$|\Psi_{K,b}\rangle = \sum_{x=0}^{N-1} a_{K,b}(x)\,|x\rangle\,, \tag{4.29}$$

wobei $a_{K,b}(x) = \sqrt{2}\tilde{a}_K(b,x)$ die normalisierte Amplitude des Basiszustands $|x\rangle$ des zweiten Registers nach K Iterationen und Messen des Hilfsqubits ist.

Sei $p_K(x)$ die Wahrscheinlichkeit, nach K Iterationen bei einer Messung auf dem zweiten Register den Wert x zu erhalten. Für diese folgt dann aus (4.28):

$$p_K(x) = \left[|\tilde{a}_K(0,x)|^2 + |\tilde{a}_K(1,x)|^2\right] = |a_{K,0}(x)|^2 = |a_{K,1}(x)|^2. \tag{4.30}$$

Es macht für den Ausgang der Messung des zweiten Registers folglich keinen Unterschied, ob das Hilfsqubit gemessen wurde und was das Ergebnis der Messung war. Mit (4.27) gilt für $p_K(x)$:

$$p_K(x) = p_0(x)(1 - \lambda_K(\cos(\varphi(x)) - \cos(\theta))),$$

wobei

$$\lambda_K = \frac{2\sin(K\theta)\sin((K+1)}{\sin^2(\theta)} = \frac{\cos(\theta) - \cos((2K+1)\theta)}{\sin^2(\theta)}. \tag{4.31}$$

Somit ist $p_K(x)/p_0(x)$ affin in $\cos(\varphi(x))$ für alle $K \geq 0$. In Abbildung 4.4 wird der Parameter λ_K grafisch dargestellt. Betrachtet man λ_K als Funktion mit K als Parameter, erhält man, dass λ_K eine oszillierende Funktion ist, die um $\cos(\theta)/\sin^2(\theta)$ zentriert ist. Außerdem ist λ_K π/θ-periodisch und hat eine Amplitude von $1/\sin^2(\theta)$.

Insgesamt folgt:

1. K-mal Iterieren ändert die Wahrscheinlichkeit, einen Zustand x zu messen, um einen affinen Faktor, der von $\cos(\varphi(x))$ abhängt.
2. Wenn $\cos(\varphi(x)) = \cos(\theta)$ für ein x gilt, verändert sich die Wahrscheinlichkeit, im Laufe einer Iteration x zu messen, nicht.
3. Wenn λ_K positiv ist, werden die Zustände x mit $\cos(\varphi(x)) < \cos(\theta)$ verstärkt.
4. Wenn λ_K negativ ist, werden die Zustände x mit $\cos(\varphi(x)) > \cos(\theta)$ verstärkt.
5. Die Geschwindigkeit der Verstärkung ist abhängig von der Größe von λ_K.

Für den Erfolg des Algorithmus ist also ein positives λ_K notwendig. Dadurch wird die Amplitude eines Basiszustands $|x\rangle$ umso schneller verstärkt, je kleiner $\cos(\varphi(x))$ – und damit aufgrund der Einschränkung (4.1) je größer $\varphi(x)$ – ist.

4.5 Optimale Anzahl der Iterationen

Den Parameter λ_k können wir auch zum Ermitteln der optimalen Anzahl an Iterationen verwenden. Aus (4.31) ist ersichtlich, dass λ_K positiv ist für $K = 0, 1, 2, 3...$ mit $\theta \leq (2K+1)\theta \leq \pi + \theta$ beziehungsweise

$$0 \leq K \leq \left\lfloor \frac{\pi}{2\theta} \right\rfloor. \tag{4.32}$$

Daraus ergibt sich die optimale Anzahl an Iterationen

$$\tilde{K} = \left\lfloor \frac{\pi}{2\theta} \right\rfloor. \tag{4.33}$$

Mit dieser Strategie werden die Iterationen demnach, unmittelbar bevor λ_K nicht mehr positiv ist, gestoppt. Interessant ist, dass die optimale Anzahl der Iterationen $\tilde{K}$ nicht von der Größe des Suchraumes, sondern nur von θ abhängt. Mit dieser Wahl der Anzahl der Iterationen funktioniert die gewünschte Amplitudenverstärkung genau dann, wenn $0 < \cos(\theta) < 1$. Der Algorithmus kann also nur für Funktionen φ mit dem Startzustand $|\Psi_0\rangle$ mit positiven $\cos(\theta)$ benutzt werden. In der Regel wird $\cos(\theta)$ vor dem Ausführen des Algorithmus ermittelt. Ist $\cos(\theta)$ nicht positiv, muss die Funktion φ oder der Startzustand $|\Psi_0\rangle$ entsprechend angepasst werden.

Zusammenfassung

NBAA ist ein Algorithmus, der nicht-binäre Suchprobleme löst. Es müssen aber eine zu maximierende Funktion $\varphi : \{0, ..., N-1\} \to [0, \pi]$ und die Möglichkeit, ein Orakel U_φ als Quantenschaltkreis zu implementieren, gegeben sein. Außerdem muss der Startzustand des Algorithmus $|\Psi_0\rangle$ so gewählt sein, dass $\cos(\theta)$ positiv ist. Wie wir NBAA nun benutzen können, um das Ising-Problem zu lösen, betrachten wir im nächsten Kapitel.

5

NBAA zum Lösen des Ising-Problems

Wir wollen nun das Ising-Problem mit NBAA (Algorithmus 2) lösen. Dafür müssen wir einige Anpassungen am Algorithmus vornehmen. Zuerst stellen wir in Abschnitt 5.1 vor, wie wir $\varphi(x)$ wählen und wie das Orakel effizient implementiert werden kann. Anschließend legen wir in Abschnitt 5.2 den Startzustand fest. In Abschnitt 5.4 untersuchen wir, welche Skalierung der Zielfunktion $\varphi(x)$ die besten Ergebnisse liefert.

5.1 Implementierung des Orakels

Um NBAA verwenden zu können, muss eine zu maximierende Funktion $\varphi(x)$ und eine effiziente Implementierung des Orakels U_φ gegeben sein. Im Folgenden wird zunächst die Zielfunktion $\varphi(x)$ betrachtet und anschließend eine mögliche Implementierung von U_φ vorgestellt.

Skalierung der Zielfunktion

Da bei NBAA das Maximum der Funktion $\varphi(x)$ gesucht wird und $\varphi(x)$ auf das Intervall $[0, \pi]$ beschränkt ist, muss das Ising-Problem umgewandelt werden.

Das klassische Ising-Problem lässt sich in die Form

$$\arg\min_{x \in \{0,1\}^n} \epsilon(x) \tag{5.1}$$

mit

$$\epsilon(x) := \sum_{i=1}^{n} C_{ii}(-1)^{x_i} + \sum_{1 \le i < j \le n} C_{ij}(-1)^{x_i + x_j}. \tag{5.2}$$

umformen. Hierbei ist $x \in \{0,1\}^n = \{0, ..., N-1\}$ und die Kostenmatrix $C \in \mathbb{R}^{n \times n}$ ist gegeben. Im weiteren Verlauf kürzen wir $\epsilon(x)$ mit ϵ_x ab. Um das Minimierungsproblem in ein Maximierungsproblem zu überführen, genügt es, die Kosten ϵ_x zu negieren. Um NBAA benutzen zu können, müssen weiterhin die Werte von ϵ_x auf das Intervall $[0, \pi]$ beschränkt werden: Der Wert

$$D := \sum_{i=1}^{n} |C_{ii}| + \sum_{1 \le i < j \le n} |C_{ij}|. \tag{5.3}$$

liefert zunächst eine obere Schranke für den Betrag der Kosten ϵ_x. Mittels

$$\tilde{\epsilon}_x = -d_1\epsilon_x + d_2 \tag{5.4}$$

mit

$$d_1 := \frac{(b-a)}{2D} \quad \text{und } d_2 := \frac{a+b}{2} \tag{5.5}$$

können somit die Kosten auf ein gewähltes Intervall $[a, b] \subseteq [0, \pi]$ skaliert werden. Den Parameter $\varphi(x)$ setzen wir gleich $\tilde{\epsilon}_x$. Ist x^* die gesuchte Lösung des Suchproblems, dann ist $\tilde{\epsilon}_{x^*}$ das Maximum. Da NBAA Maximierer von $\varphi(x)$ sucht, verstärkt der Algorithmus genau die Amplitude des Zustands der Lösungen des Ising-Problems am stärksten. Für welches Intervall $[a, b]$ der NBAA besonders gute Ergebnisse liefert, wird in Abschnitt 5.4 untersucht.

Schaltkreis zum Orakel

Eine wichtige Voraussetzung für die Effizienz von NBAA ist eine effiziente Implementierung des Orakels. Zur Implementierung des Orakels für das Ising Problem verwenden wir den von Kuete Meli et al. vorgestellten Schaltkreis [28]. Dieser ermöglicht es, mittels Rotationen die Phasen $\varphi(x)$ der Basiszustände in den Amplituden zu kodieren. Hierfür wird ein zusätzliches *Cost-Qubit* $|c\rangle$ eingeführt. Das System wird demnach um ein Qubit erweitert:

$$|h, x, c\rangle \in \mathbb{C} \otimes \mathbb{C}^{\otimes n} \otimes \mathbb{C}. \tag{5.6}$$

Zur Implementierung des Orakels nutzen wir aus, dass die Phasen $e^{\pm i\theta}$ genau die Eigenwerte der Drehmatrix

$$R_y(2\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \tag{5.7}$$

zu den Eigenvektoren $|+_i\rangle = \frac{1}{\sqrt{2}}(1, -i)^\top$ und $|-_i\rangle = \frac{1}{\sqrt{2}}(1, i)^\top$ sind. Modifizieren wir nun das 1-Register-Orakel U_φ aus (3.3) zu der Blockstruktur

$$\tilde{U}_\varphi = \begin{pmatrix} R_y(2\varphi(0)) & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots \\ \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & R_y(2\varphi(N-1)) \end{pmatrix}, \tag{5.8}$$

so gilt

$$\tilde{U}_\varphi(|x\rangle \otimes |+_i\rangle) = e^{i\varphi(x)}(|x\rangle \otimes |+_i\rangle). \tag{5.9}$$

Das neue 1-Register-Orakel $\tilde{U}_\varphi$ ist eine lineare Abbildung von $\mathbb{C}^{2^{n+1}}$ nach $\mathbb{C}^{2^{n+1}}$. Das Cost-Qubit wird mit einer Hintereinanderausführung von H, S und Z in den Zustand $|+_i\rangle$

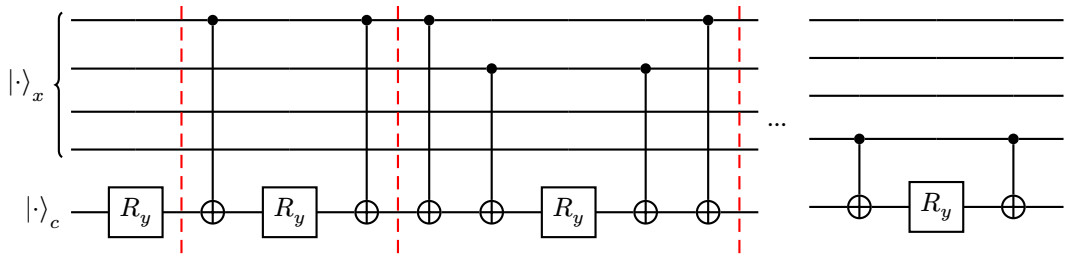


Abbildung 5.1: Die Implementierung des 1-Register-Orakels, angelehnt an Kuete Meli et al. [28], ist eine Hintereinanderausführung von „CNOT"-Gattern und R_y-Gattern.

initialisiert, wobei die Matrizen Z, S und H wie in Tabelle 2.3 definiert sind. Es gilt

$$\begin{aligned} ZSH|0\rangle &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} |0\rangle \\ &= \frac{1}{\sqrt{2}}(|0\rangle - \mathrm{i}\,|1\rangle) \\ &= |+_{\mathrm{i}}\rangle\,. \end{aligned} \tag{5.10}$$

Trotz des zusätzlichen Qubits behalten wir die Bezeichnung 1-Register- und 2-Register-Orakel bei. Für ein beliebiges $x \in \{0, 1, ..., N-1\}$ soll das Cost-Qubit nun um den Winkel $\tilde{\epsilon}_x$ gedreht werden. Dies wird wie folgt implementiert:

$$\begin{aligned} R_y(2\varphi(x)) &= R_y(2\tilde{\epsilon}_x) \\ &= R_y\Big(-2d_1\epsilon_x + 2d_2 \Big) \\ &= \prod_{i=1}^{n} X^{x_i} R_y\Big(-2d_1 C_{ii} \Big) X^{x_i} \cdot \\ &\quad \prod_{1\le i<j\le n} X^{x_i+x_j} R_y\Big(-2d_1 C_{ij} \Big) X^{x_i+x_j} \cdot R_y(2d_2). \end{aligned} \tag{5.11}$$

Hierbei nutzen wir aus, dass eine Rotation um eine Summe ein Produkt von Rotationen ist. Mehr Details hierzu sind im Anhang A.1 zu finden. Diese Implementierung des Orakels ist angelehnt an die von Kuete Meli et al. [28], bei der die Kodierung des Problems abweicht, und hat den Vorteil, dass für diese wenige Gatter benötigt werden. Zusätzlich kann $\cos(\theta)$ einfacher geschätzt werden. Auf Letzteres wird in Abschnitt 5.3 näher eingegangen. Der Schaltkreis des modifizierten Orakels ist in Abbildung 5.1 schematisch dargestellt.

2-Register-Orakel. Das 2-Register-Orakel ist wie in Kapitel 4 als

$$\begin{aligned} \tilde{\mathbf{U}}_\varphi &:= |0\rangle\langle 0| \otimes \tilde{U}_\varphi + |1\rangle\langle 1| \otimes \tilde{U}_\varphi \\ &= \begin{pmatrix} \tilde{U}_\varphi & 0 \\ 0 & \tilde{U}_\varphi^H \end{pmatrix} \end{aligned} \tag{5.12}$$

definiert.

Komplexität des Orakels

Nachdem wir eine Implementierung des 1-Register-Orakels vorgestellt haben, untersuchen wir die Effizienz beziehungsweise Komplexität dieser. Der Schaltkreis des 1-Register-Orakels besteht aus maximal $n(n+1)$ kontrollierten „NOT"-Gattern und $n(n+1)/2+1$ Rotationen. Somit liegt die Gatteranzahl der Implementierung des 1-Register-Orakels in $\mathcal{O}(n^2) = \mathcal{O}(\log(N)^2)$ und der 2-Register-Orakel-Aufruf dann ebenfalls in $\mathcal{O}(\log(N)^2)$.

Wir haben nun eine Funktion φ für das Ising-Modell entwickelt und eine effiziente Implementierung des Orakels gefunden, für die wir aber ein zusätzliches Qubit hinzufügen mussten. Im nächsten Abschnitt legen wir den Startzustand fest.

5.2 Wahl des Startzustands

Für das Lösen des Ising-Problems haben wir in der Regel kein Vorwissen. Daher wählen wir als Initialisierung des x-Registers $A_0 = H^{\otimes n}$. Diese Initialisierung wurde auch von Shyamsundar et al. in ihrem Beispiel gewählt [37]. Somit ergibt sich als Startzustand des x-Registers $|\Psi_0\rangle$ der perfekt überlagerte Zustand

$$|\Psi_0\rangle = A_0 |0\rangle^{\otimes n} = H^{\otimes n} |0\rangle^{\otimes n} = \sum_{x=0}^{N-1} \frac{1}{\sqrt{N}} |x\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \tag{5.13}$$

und als Startzustand des 2-Registersystems

$$\begin{aligned} |\Psi_0\rangle &= [H \otimes A_0] |0,0\rangle = \frac{|0,\Psi_0\rangle + |1,\Psi_0\rangle}{\sqrt{2}} \\ &= \frac{1}{\sqrt{2N}} \sum_{x=0}^{N-1} (|0,x\rangle + |1,x\rangle). \end{aligned} \tag{5.14}$$

Da wir ein Qubit bei der Implementierung des Orakels hinzugefügt haben, müssen auch die anderen Transformationen des Algorithmus angepasst werden. Hierfür erweitern wir A_0 zu $\tilde{A}_0$:

$$\tilde{A}_0 := H^{\otimes n} \otimes (ZSH). \tag{5.15}$$

Das Inverse von $\tilde{A}_0$ ist

$$\tilde{A}_0^{-1} = H^{\otimes n} \otimes (HS^H Z). \tag{5.16}$$

Als Startzustand mit dem Cost-Qubit ergibt sich dann

$$\begin{aligned} |\tilde{\Psi}_0\rangle &= (H \otimes \tilde{A}_0) |0\rangle^{\otimes (n+2)} \\ &= \frac{1}{\sqrt{2N}} \sum_{x=0}^{N-1} (|0,x,+_{\mathrm{i}}\rangle + |1,x,+_{\mathrm{i}}\rangle). \end{aligned} \tag{5.17}$$

Diffusion. Der Operator $\tilde{A}_0$ wird auch für die erweiterte Diffusion verwendet:

$$\begin{aligned} S_{\tilde{\Psi}_0} &:= H \otimes \tilde{A}_0 (2 |0,0,0\rangle \langle 0,0,0| - I_{2^{n+2}}) H \otimes \tilde{A}_0^{-1} \\ &= 2|\tilde{\Psi}_0\rangle\langle\tilde{\Psi}_0| - I_{2^{n+2}}. \end{aligned} \tag{5.18}$$

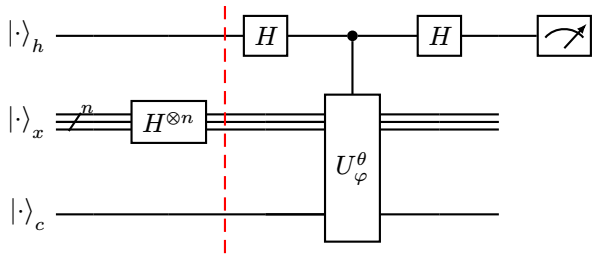


Abbildung 5.2: Schaltkreis zum Schätzen von $\cos(\theta)$. Das Hilfsqubit $|h\rangle$ wird im Zustand $|0\rangle$, das n-Qubit-Register wird mit $H^{\otimes n}$ in den perfekt überlagerten Zustand und das Cost-Qubit $|c\rangle$ in den Zustand $|0\rangle$ initialisiert. Anschließend wird auf $|h\rangle$ erst ein Hadamard-Gatter, dann das veränderte Orakel U_φ^θ von $|h\rangle$ kontrolliert auf $|x\rangle$ und $|c\rangle$ und dann noch ein Hadamard-Gatter auf $|h\rangle$ angewendet. Am Ende wird $|h\rangle$ in der Rechenbasis gemessen.

Somit sind alle Operationen, die wir für NBAA brauchen, definiert. Für den Algorithmus müssen wir nun noch die Anzahl der Iterationen festlegen. Um die optimale Anzahl an Iterationen zu berechnen, muss $\cos(\theta)$ beziehungsweise θ ermittelt werden (siehe Abschnitt 4.5). Die exakte Berechnung von $\cos(\theta)$ ist aufwendig und dabei muss jedes Energieniveau des Ising-Modells berechnet werden. Dabei würde das Ising-Problem schon gelöst werden. Eine Schätzung von $\cos(\theta)$ kann mithilfe des in Abschnitt 5.1 eingeführten Orakels $\tilde{U}_\varphi$ durchgeführt werden.

5.3 Approximieren von $\cos(\theta)$

Einer der Vorteile der Implementierung des 1-Register-Orakels als Rotation ist, dass $\cos(\theta)$ mit wenig Rechenaufwand geschätzt werden kann. Um $\cos(\theta)$ zu approximieren, wird $\tilde{U}_\varphi$ so modifiziert, dass der Operator hermitesch ist. Hierfür müssen die Drehmatrizen $R_y(2\varphi(x))$ zu

$$R_y^\theta(2\varphi(x)) = \begin{pmatrix} \cos(\varphi(x)) & \sin(\varphi(x)) \\ \sin(\varphi(x)) & -\cos(\varphi(x)) \end{pmatrix} \tag{5.19}$$

modifiziert werden. Das für die Schätzung von $\cos(\theta)$ angepasste 1-Register-Orakel ist dann

$$U_\varphi^\theta = \begin{pmatrix} R_y^\theta(2\varphi(0)) & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots \\ \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & R_y^\theta(2\varphi(N-1)) \end{pmatrix}. \tag{5.20}$$

U_φ^θ ist wie $\tilde{U}_\varphi$ in $\mathbb{C}^{2^{n+1}\times 2^{n+1}}$ und wird ähnlich implementiert. Der einzige Unterschied ist, dass nach der Initialisierung des Cost-Qubits $|c\rangle$ ein Z-Gatter auf $|c\rangle$ angewendet wird. Dann gilt:

$$\tilde{U}_\varphi(I_2^{\otimes n} \otimes Z) = U_\varphi^\theta. \tag{5.21}$$

Analog zum Suchalgorithmus werden drei Register initialisiert:

- Ein Hilfsqubit $|h\rangle$, das im Zustand $|0\rangle$ ist.
- Ein n-Qubit-Regster, das mit $H^{\otimes n}$ in den perfekt überlagerten Zustand initialisiert wird.
- Ein Cost-Qubit $|c\rangle$, das im Zustand $|0\rangle$ bleibt.

Wird nun der in Abbildung 5.2 dargestellte Schaltkreis ausgeführt und anschließend $|h\rangle$ in der Rechenbasis gemessen, kann $\cos(\theta)$ folgendermaßen approximiert werden:

$$\cos(\theta) = p(0) - p(1). \tag{5.22}$$

Dabei ist $p(0)$ die durch mehrfaches Messen empirisch ermittelte Wahrscheinlichkeit, „0" zu messen, und $p(1)$ die Wahrscheinlichkeit, „1" zu messen. Das Ermitteln dieser Wahrscheinlichkeiten wurde im Kapitel 2 detailliert behandelt. Ein detaillierter Beweis der Approximation ist im Anhang A.2 zu finden.

Da wir beim Ausführen des Schaltkreises der Schätzung nur die Initialisierung, zwei Hadamard-Gatter und einen 1-Register-Orakel-Aufruf durchführen müssen, hat dieses eine Komplexität von $\mathcal{O}(log(N)^2)$ bezüglich der Gatteranzahl. Allerdings muss der Schaltkreis mehrfach ausgeführt werden, um aus den Messdaten $\cos(\theta)$ zu approximieren.

Eine Alternative, um $\cos(\theta)$ zu schätzen, wäre den Quantenphasenschätzungsalgorithmus [25, 35] auszuführen. Dies wurde von Shyamsundar et al. [37, Kapitel 4] vorgeschlagen. Dabei wird ein angepasstes Orakel als Operator für die Phasenschätzung sowie eine Kombination des Startzustands (4.5) und des Zustands $|\alpha\rangle$ aus (4.18) als Eigenvektor benutzt. Die Phasenschätzung benötigt zusätzliche Kontrollgatter und Hilfsqubits. Solange die gesuchte Phase nicht in der Binärbasis darstellbar ist, muss auch bei dieser Methode mehrfach gemessen werden. Die Genauigkeit der Schätzung hängt hier von der Anzahl der hinzugefügten Hilfsqubits ab [37, Abbildung 10]. Dies beeinflusst wiederum die Anzahl der Gatter. Durch das Hinzufügen von Qubits kann $\cos(\theta)$ mit der Quantenphasenschätzung beliebig genau approximiert werden. Der Schaltkreis wird aber mit jedem hinzugefügten Qubit komplexer.

Das Approximieren von $\cos(\theta)$ ist demnach ein Abwägen von Ressourcen und Genauigkeit. Wir haben uns in dieser Arbeit für die Methode mit U_φ^θ entschieden; insbesondere auch weil diese mit wenigen zusätzlichen Schritten zu NBAA implementiert werden kann. Durch unsere Wahl der Implementierung des 1-Register-Orakels erhalten wir also eine effiziente Methode, $\cos(\theta)$ zu schätzen. Aus $\cos(\theta)$ lässt sich anschließend die optimale Anzahl an Iterationen schätzen (siehe Abschnitt 4.5).

5.4 Numerische Experimente

In Abschnitt 5.1 haben wir untersucht, wie wir $\varphi(x)$ auf ein Intervall $[a, b] \subseteq [0, \pi]$ beschränken. Jetzt soll untersucht werden, auf welchem Intervall NBAA das Ising-Problem am besten löst.

Wir wählen $a = 0$ und betrachten $b \in \{\frac{k}{8}\pi | k = 1, ..., 8\}$. Für jedes b wird der Parameter $\tilde{\epsilon}_x$ wie in (5.4) auf das Intervall $[0, b]$ beschränkt und NBAA mit $\varphi(x) = \tilde{\epsilon}_x$ ausgeführt.

Metriken

Zum Vergleichen der Effektivität des Algorithmus mit den verschiedenen Intervallen verwenden wir zwei Metriken: das *approximation ratio* und die *geschätzte Wahrscheinlichkeit*, den Zustand des Minimums zu messen.

Sei $|x^*\rangle$ ein zu einer der gesuchten Lösungen x^* gehörender Basiszustand. Sei ϵ_x wie in (5.2) definiert. Wir führen die Parameter ϵ_{min} und ϵ_{max} als Minimum beziehungsweise Maximum aller Energiezustände ϵ_x ein. Weiterhin sei

$$p(m = x) = \frac{\text{Anzahl der Messungen von „x''}}{\text{Anzahl aller Messungen}} \tag{5.23}$$

die empirisch bestimmte Wahrscheinlichkeit, nach Anwenden des Algorithmus 2 mit der jeweiligen oberen Intervallgrenze b den Wert $x \in \{0, ..., N-1\}$ zu messen (vgl. Kapitel 2). Die Metriken sind folgendermaßen definiert:

- Das approximation ratio r ist

  $$r := \frac{\epsilon_{max} - \mathbb{E}(\epsilon_x)}{\epsilon_{max} - \epsilon_{min}} \tag{5.24}$$

  mit

  $$\mathbb{E}(\epsilon_x) := \sum_{x=0}^{N-1} p(m = x)\epsilon_x \tag{5.25}$$

  als Erwartungswert der Energiezustände über x. Das approximation ratio liegt zwischen 0 und 1. Wird nur x^* gemessen, ist $r = 1$. Je größer r ist, umso effektiver ist der Algorithmus.
- Sei p_{Lsg} die Wahrscheinlichkeit, den Zustand des Minimierers zu messen. Es gilt also

  $$p_{\text{Lsg}} = \sum_{x^* \text{ ist Lösung}} p(m = x^*). \tag{5.26}$$

  Die Wahrscheinlichkeit p_{Lsg} liegt ebenfalls zwischen 0 und 1. Bei NBAA kann nicht garantiert werden, dass nur der Minimierer gemessen wird. Daher gilt, je größer p_{Lsg} ist, desto effektiver ist der Algorithmus, da dann die Lösung mit einer höheren Wahrscheinlichkeit gefunden wird.

Rechnerumgebung und Datenerhebung

Für jede Wahl der oberen Intervallgrenze b wurde das Experiment für dieselben, zufällig erzeugten 40 Matrizen C durchgeführt. Für das zufällige Erzeugen der Ising-Probleme wurde die Implementierung aus [28] übernommen. Die Matrizen wurden mit Graphen erzeugt, indem die Kanten und Kantengewichte des Graphen in eine Matrix überführt wurden. Genau diese Matrix ist dann die Matrix C, die das spezifische Ising-Problem definiert. Die zufälligen Graphen in den Experimenten wurden mit dem Python-Paket NetworkX [20] erzeugt. Alle Knoten wurden paarweise durch eine Kante verbunden.

Ob ein Knoten ein Knotengewicht hat, wurde binomialverteilt mit $p = 0.5$ entschieden. Die Kantengewichte C_{ij} beziehungsweise Knotengewichte C_{ii} wurden alle zufällig und gleichverteilt im Bereich $[-10, 10]$ gewählt. Die Experimente wurden in Python 3.9 implementiert und wurden mit der QisKit-Bibliothek und dem IBM-QSAM-Simulator [10] simuliert.

Zum Vergleichen des Algorithmus mit den verschiedenen Intervallen $[0, b]$ wurden die entsprechenden Werte über die 40 zufällig erzeugten Ising-Probleme gemittelt.

Ergebnisse

In Abbildung 5.3 werden für die verschiedenen Intervalle $[0, b]$ das approximation ratio und die geschätzte Wahrscheinlichkeit, den Zustand der Lösung zu messen, dargestellt. Um den Einfluss von θ hervorzuheben, werden zusätzlich auch $\cos(\theta)$ und die optimale Anzahl an Iterationen verglichen.

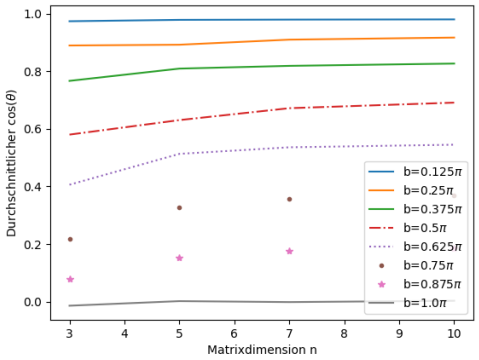
Wir können Folgendes beobachten:

- Je größer b ist, desto kleiner ist $\cos(\theta)$. Insbesondere ist $\cos(\theta)$ bei $b = \pi$ nicht immer positiv.
- Je kleiner b ist, desto häufiger muss iteriert werden.
- Das approximation ratio ist bei $b = \frac{1}{4}\pi$ und $b = \frac{1}{2}\pi$ für $n = 3$ maximal und nimmt einen Wert von ungefähr 0.9 an. Steigt die Problemgröße, nimmt das approximation ratio für $b \in \{\frac{1}{8}\pi, \frac{1}{4}\pi, \frac{3}{8}\pi, \frac{1}{2}\pi, \frac{5}{8}\pi, \frac{3}{4}\pi\}$ ab. Bei $b \in \{\frac{7}{8}\pi, \pi\}$ bleibt das approximation ratio bei 0.6. Allgemein ist das approximation ratio bei $b = \frac{1}{4}\pi$ und $b = \frac{1}{2}\pi$ höher als bei den anderen.
- Die Wahrscheinlichkeit p_{Lsg} nimmt bei wachsender Problemgröße exponentiell ab und ist bei $b = \frac{1}{4}\pi$ und $b = \frac{1}{2}\pi$ höher als bei den anderen Intervallen.

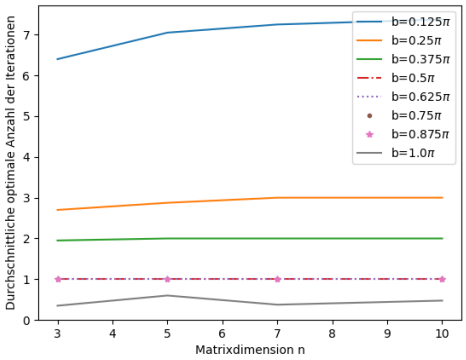
Es eignet sich beinahe jedes Intervall zum Lösen des Ising-Problems. Nur bei $[a, b] = [0, \pi]$ ist $\cos(\theta)$ nicht immer positiv, was in Kapitel 4 gefordert wurde. Dadurch kann NBAA für dieses Intervall nicht für jedes Ising-Problem verwendet werden.

Am besten schneiden $b = \frac{1}{2}\pi$ und $b = \frac{1}{4}\pi$ ab, da bei diesen das approximation ratio gut ist, nur eine bis drei Iterationen gebraucht werden und die Wahrscheinlichkeit, die Lösung zu messen, im Vergleich zu den anderen Intervallen hoch ist.

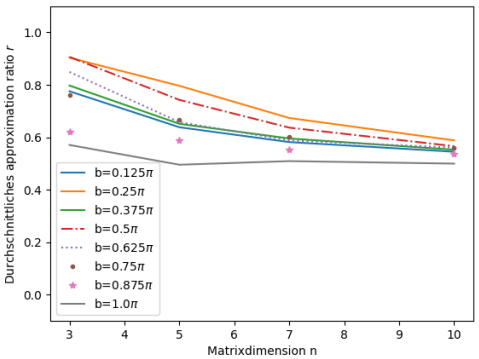
Es ist aber auffällig, dass das approximation ratio im Vergleich zu den von Kuete Meli et al. [28] (approximation ratio ≥ 0.9 nach 10 Iterationen) und von Goemans und Williamson [14] (approximation ratio ≈ 0.868) vorgestellten Algorithmen nicht besser ist. Dies versuchen wir nun noch zu verbessern, indem wir NBAA anpassen, so dass die Grover-Phase-Matching-Bedingung erfüllt wird.



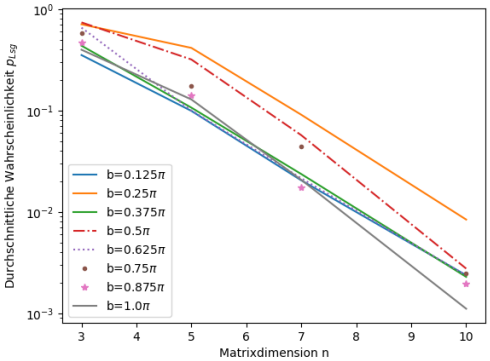
(a) Der durchschnittliche Wert von $\cos(\theta)$ in Abhängigkeit der Matrixdimension n der Kostenmatrix. Es wird NBAA mit verschiedenen Intervallen $[0, b]$, auf die $\varphi(x)$ beschränkt ist, verglichen.



(b) Die durchschnittliche optimale Anzahl an Iterationen in Abhängigkeit der Matrixdimension n der Kostenmatrix für die verschiedenen Intervalle $[0, b]$.



(c) Das durchschnittliche approximation ratio r in Abhängigkeit der Matrixdimension n der Kostenmatrix für die verschiedenen Intervalle $[0, b]$.



(d) Die durchschnittliche Wahrscheinlichkeit p_{Lsg} in Abhängigkeit der Matrixdimension n der Kostenmatrix für die verschiedenen Intervalle $[0, b]$.

Abbildung 5.3: Für die verschiedenen Intervalle $[0, b]$, auf die $\varphi(x)$ beschränkt ist, wird $\cos(\theta)$, die optimale Anzahl an Iterationen, das approximation ratio und die geschätzte Wahrscheinlichkeit, den Zustand des Minimums zu messen, in Abhängigkeit von der Matrixdimension n der Kostenmatrix des Ising-Problems untersucht. Die durchschnittlichen Werte werden aus den Werten von 40 zufällig erzeugten Ising-Problemen berechnet. Bei (a) fällt auf: Je größer b ist, desto kleiner ist $\cos(\theta)$. Insbesondere ist $\cos(\theta)$ bei $b = \pi$ nicht immer positiv. In (b) sieht man: Je kleiner b ist, desto häufiger muss iteriert werden. In (c) wird das approximation ratio verglichen. Dieses ist bei $b = \frac{1}{4}\pi$ und $b = \frac{1}{2}\pi$ für $n = 3$ maximal und nimmt ungefähr 0.9 an. Das approximation ratio nimmt bei steigendem n ab und beträgt bei $n = 10$ für jedes b ungefähr 0.6. Die in (d) betrachtete Wahrscheinlichkeit p_{Lsg} nimmt bei wachsender Problemgröße exponentiell ab und ist bei $b = \frac{1}{4}\pi$ und $b = \frac{1}{2}\pi$ höher als bei den anderen. Insgesamt liefern $b = \frac{1}{4}\pi$ und $b = \frac{1}{2}\pi$ die besten Ergebnisse. Es ist aber auffällig, dass das approximation ratio noch nicht mit bereits bestehenden Algorithmen [28, 14] mithalten kann. Ein Verbesserungsvorschlag ist, NBAA anzupassen, so dass auch die Grover-Phase-Matching-Bedingung erfüllt wird (siehe Kapitel 6).

6

Nicht-binäre Amplitudenverstärkung mit Phasenkorrektur

Die Ergebnisse bezüglich der Grover-Phase-Matching-Bedingung aus Abschnitt 3.5 legen nahe, dass die Phase der Diffusion γ mit der Phase der Lösungen $\varphi(x^\star)$ übereinstimmen muss, um mit dem verallgemeinerten Grover-Algorithmus das Suchproblem mit einer höheren Wahrscheinlichkeit lösen zu können. Außerdem zeigen [18, 38], dass die Suche umso schneller ist, je näher die Phasen bei π sind.

Allerdings wissen wir, dass eine wichtige Voraussetzung für die Anwendbarkeit von Algorithmus 2 ist, dass $\cos(\theta)$ positiv und nicht Null ist. Das hat uns im Abschnitt 5.4 gezwungen, die Kosten des Ising-Problems auf das Intervall $[0, \pi/2]$ beziehungsweise $[0, \pi/4]$ zu skalieren und insbesondere auch θ für $\cos(\theta)$ aus (4.17) in diesem Intervall zu beschränken. Betrachten wir nun eine Beispielkonfiguration, bei der wir Algorithmus 2 auf ein binäres Suchproblem mit $\varphi(x) = 0$ oder $\varphi(x) = \pi/2$ beschränken. Die Skalierung in $[0, \pi/2]$ führt dazu, dass die Phasen nicht mehr übereinstimmen und dadurch die Wahrscheinlichkeit, den Zustand der Lösung am Ende des Algorithmus zu messen, stark verringert wird [31].

In diesem Kapitel wird eine Korrektur eingeführt, mit der die Amplitude des Minimierers weiter verstärkt wird als bei dem in Kapitel 4 eingeführten Algorithmus 2 (NBAA). Hierfür wird in Abschnitt 6.1 der Aufbau des angepassten Algorithmus beschrieben. Anschließend wird der vorgestellte Algorithmus in Abschnitt 6.2 analysiert und insbesondere das Iterationsverhalten untersucht. Schließlich wird in Abschnitt 6.3 der abgewandelte Algorithmus mit NBAA aus dem Kapitel 4 verglichen.

6.1 Aufbau des Algorithmus

Damit die Grover-Phase-Matching-Bedingung (Definition 3.3) erfüllt ist, wird NBAA folgendermaßen abgewandelt:

- Nur in der ersten Iteration werden die Kosten in $[0, \pi/2]$ skaliert, ansonsten in $[0, \pi]$.
- Wir verwenden weiter 2-Register-Diffusion und -Orakel.
- Wir alternieren Diffusion und Orakel, unterscheiden aber nicht mehr zwischen geraden und ungeraden Iterationen wie bei Algorithmus 2. Dadurch verlassen wir den

durch $\{|\Psi_0\rangle, |\alpha\rangle, |\beta\rangle\}$ aufgespannten Raum.

Zusätzlich stellen wir folgende Voraussetzungen und Annahmen auf:

Voraussetzungen und Annahmen. Beschreibe $|\hat{\Psi}_0\rangle$ den Startzustand und $\mathbf{U}_{\varphi^0}$ das 2-Register-Orakel der ersten Iteration des neuen Algorithmus. Mit $\cos(\theta)$ aus (4.17) gilt $\langle\hat{\Psi}_0|\mathbf{U}_{\varphi^0}|\hat{\Psi}_0\rangle = \cos(\theta)$. Wie bei NBAA fordern wir, dass $\langle\hat{\Psi}_0|\mathbf{U}_{\varphi^0}|\hat{\Psi}_0\rangle$ positiv ist.

Außerdem nehmen wir Folgendes an: Wir betrachten die Energiezustände ϵ_x (5.4) des Ising-Problems und nehmen an, dass $\epsilon_{min} \approx -D$ mit ϵ_{min} als Minimum aller Energiezustände ϵ_x und D aus (5.3). Bei allen weiteren Iterationen wird $[a, b] = [0, \pi]$ gewählt. Skalieren wir ϵ_x wie in (5.4) auf das Intervall $[a, b] = [0, \pi]$ und wählen dieses skalierte ϵ_x als $\varphi(x)$ für das Orakel ab der zweiten Iteration, gilt $\varphi(x) \approx \pi$, wenn x der Minimierer des Ising-Problems ist. Ab der zweiten Iteration ist also die Grover-Phase-Matching-Bedingung näherungsweise erfüllt.

Viele der für Algorithmus 2 eingeführten Operatoren bleiben im korrigierten Algorithmus unverändert.

Startzustand. Der Startzustand ist

$$\begin{aligned} |\hat{\Psi}_0\rangle &= |\tilde{\Psi}_0\rangle \\ &= \frac{1}{\sqrt{2N}} \sum_{x=0}^{N-1} (|0, x, +_{\mathrm{i}}\rangle + |1, x, +_{\mathrm{i}}\rangle). \end{aligned} \tag{6.1}$$

Diffusion. Die Diffusion ist dann

$$\mathbf{S}_{\hat{\Psi}_0} := 2|\hat{\Psi}_0\rangle\langle\hat{\Psi}_0| - I_{2^{n+2}}. \tag{6.2}$$

Wahl von Phi. Um mit PM-NBAA das Ising-Problem zu lösen, werden analog zu Abschnitt 5.1 die Energieniveaus ϵ_x der einzelnen Zustände x negiert und auf ein Intervall $[a, b] \in [0, \pi]$ beschränkt. Der relevante Unterschied zu NBAA ist nun, dass sich $\varphi(x)$ abhängig von der Iteration verändert. In der ersten Iteration wird $[a, b] = [0, \pi/2]$ gewählt. Dann ist $\varphi^0(x) = \epsilon_x^0 \in [0, \pi/2]$ und ϵ_x^0 ergibt sich mit $a = 0$ und $b = \pi/2$ aus (5.4). Bei allen weiteren Iterationen k wird $[a, b] = [0, \pi]$ gewählt. Das skalierte Energieniveau ϵ_x^k wird mit $a = 0$ und $b = \pi$ (5.4) berechnet. Demnach gilt $\varphi^k(x) = \epsilon_x^k \in [0, \pi]$ für $k > 1$.

2-Register-Orakel. Das 2-Register-Orakel setzt sich wieder aus dem 1-Register-Orakel zusammen

$$\begin{aligned} \hat{\mathbf{U}}_{\varphi^k} &:= |0\rangle\langle 0| \otimes \hat{U}_{\varphi^k} + |1\rangle\langle 1| \otimes \hat{U}_{\varphi^k} \\ &= \begin{pmatrix} \hat{U}_{\varphi^k} & 0 \\ 0 & \hat{U}_{\varphi^k}^H \end{pmatrix}. \end{aligned} \tag{6.3}$$

Hierbei wird

$$\hat{U}_{\varphi^k} = \begin{pmatrix} R_y(2\varphi^k(0)) & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots \\ \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & R_y(2\varphi^k(N-1)) \end{pmatrix} \tag{6.4}$$

analog zum Orakel $\tilde{U}_\varphi$ (5.8) in Abschnitt 5.1 implementiert.

Den korrigierten Algorithmus bezeichnen wir auch mit PM-NBAA. Dieser ist dann folgendermaßen definiert:

Algorithmus 3 Nicht-binäre Amplitudenverstärkung mit Phasenanpassung (PM-NBAA)

Eingabe: n-dimensionales Ising-Problem (5.1)
$N := 2^n$
$K := \lfloor \sqrt{N} \rfloor$
Initialisiere das $n+2$-Qubitsystem in den Startzustand $|\hat{\Psi}_0\rangle$
$|\hat{\Psi}_1\rangle \leftarrow \mathbf{S}_{\hat{\Psi}_0} \hat{\mathbf{U}}_{\varphi^0} |\hat{\Psi}_0\rangle$
for $i := 2, \ldots K$ **do**
  $|\hat{\Psi}_i\rangle \leftarrow \mathbf{S}_{\hat{\Psi}_0} \hat{\mathbf{U}}_{\varphi^{i-1}} |\hat{\Psi}_{i-1}\rangle$
end for
Messe $|\hat{\Psi}_K\rangle$ in der Rechenbasis.

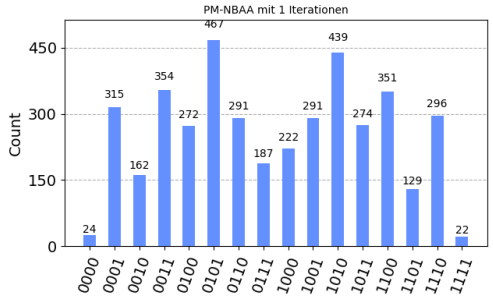
Analog zu Algorithmus 1 und Algorithmus 2 wird auch hier durch Messen des zweiten Registers die Lösung des Ising-Problems ermittelt. In PM-NBAA wird $\lfloor \sqrt{N} \rfloor$-mal iteriert. Auf die Anzahl der Iterationen wird in Abschnitt 6.2 näher eingegangen.

Beispiel 6.1. Wir wenden PM-NBAA beispielhaft auf das Ising-Problem mit $n = 4$, $N = 16$ und gegebener Kostenmatrix

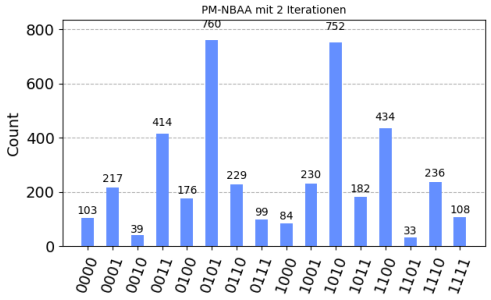
$$C = \begin{pmatrix} 0 & 6 & 7 & 9 \\ 0 & 0 & 7 & 2 \\ 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{6.5}$$

an und betrachten jede Iteration einzeln. Für die Anzahl der Iterationen zwischen eins und sechs erhalten wir nach 40896-mal messen die in Abbildung 6.1 gezeigten Histogramme. Die Minimierer des Ising-Problems sind „0101" und „1010". In den ersten vier Iterationen werden die Minimierer am häufigsten gemessen und die anderen Zustände deutlich weniger. Nach fünf beziehungsweise sechs Iterationen werden aber auch Zustände häufiger gemessen, die keine Lösung sind. Da der PM-NBAA nach $K = \lfloor \sqrt{2^n} \rfloor = 4$ Iterationen abbricht, würde dieses Phänomen nicht in dem von uns vorgeschlagenen Algorithmus auftreten. Insgesamt sehen wir hier: Der Algorithmus löst das Ising-Problem.

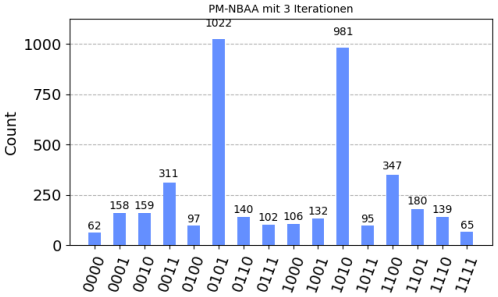
Diese Beobachtung soll nun für beliebige Ising-Probleme verallgemeinert werden. Im nächsten Abschnitt wird die Dynamik des Algorithmus und insbesondere das Iterationsverhalten genauer analysiert.



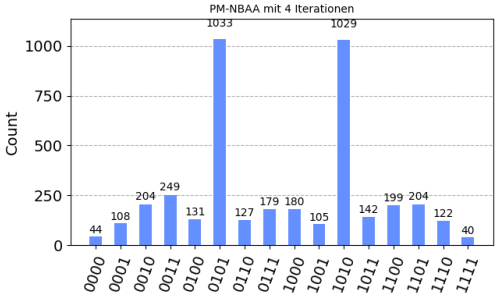
(a) Empirische Verteilung des Zustandes in PM-NBAA bei Messung nach einer Iteration.



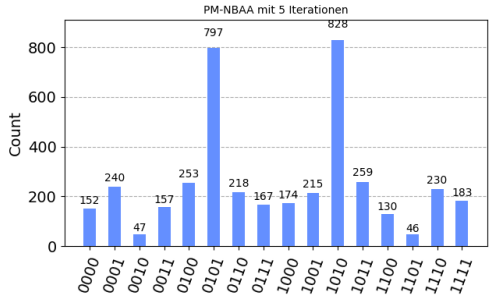
(b) Empirische Verteilung des Zustandes in PM-NBAA bei Messung nach zwei Iterationen.



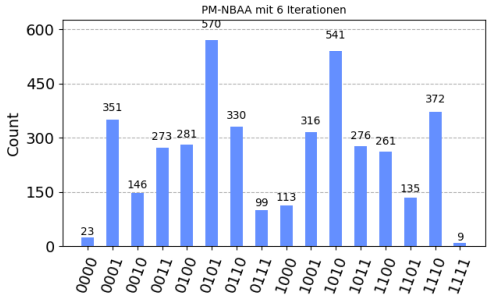
(c) Empirische Verteilung des Zustandes in PM-NBAA bei Messung nach drei Iterationen.



(d) Empirische Verteilung des Zustandes in PM-NBAA bei Messung nach vier Iterationen.



(e) Empirische Verteilung des Zustandes in PM-NBAA bei Messung nach fünf Iterationen.



(f) Empirische Verteilung des Zustandes in PM-NBAA bei Messung nach sechs Iterationen.

Abbildung 6.1: Entwicklung des Zustandes in PM-NBAA für das in Beispiel 6.1 vorgestellte Ising-Problem. Die Histogramme zeigen die Häufigkeiten der Zustände nach der Messung nach unterschiedlichen Anzahlen an Iterationen. Innerhalb der ersten vier Iterationen werden die Amplituden der Lösung zunehmend verstärkt gegenüber den Amplituden der restlichen Zustände. Nach fünf beziehungsweise sechs Iterationen würden auch Amplituden von Nicht-Lösungen deutlich vergrößert; aufgrund der maximalen Iterationszahl von $K = 4$ für dieses Problem werden diese in PM-NBAA wie in Algorithmus 3 definiert, jedoch nicht mehr ausgeführt.

6.2 Analyse des Algorithmus

Da PM-NBAA im Vergleich zu NBAA einige zentrale Änderungen enthält, ist eine neue Analyse notwendig. In diesem Abschnitt soll das Iterationsverhalten untersucht werden. Hierfür bezeichne

$$|\hat{\Psi}_k\rangle = \sum_{x=0}^{N-1} (\hat{a}_k(0,x)\,|0,x,+_\mathrm{i}\rangle + \hat{a}_k(1,x)\,|1,x,+_\mathrm{i}\rangle) \tag{6.6}$$

den Zustand nach $k \in \{0,\ldots,K\}$ Iterationen. Für den Startzustand $|\hat{\Psi}_0\rangle$ ist

$$\hat{a}_0(0,x) = \hat{a}_0(1,x) = \frac{1}{\sqrt{2N}}. \tag{6.7}$$

Betrachten wir eine Iteration mit $k > 0$: Aus $|\hat{\Psi}_k\rangle$ lässt sich

$$\begin{aligned}
|\hat{\Psi}_{k+1}\rangle &= \mathbf{S}_{\hat{\Psi}_0}\hat{\mathbf{U}}_{\varphi^k}\,|\hat{\Psi}_k\rangle \\
&= \mathbf{S}_{\hat{\Psi}_0}\hat{\mathbf{U}}_{\varphi^k} \sum_{x=0}^{N-1} (\hat{a}_k(0,x)\,|0,x,+_\mathrm{i}\rangle + \hat{a}_k(1,x)\,|1,x,+_\mathrm{i}\rangle) \\
&= 2|\hat{\Psi}_0\rangle\langle\hat{\Psi}_0| \sum_{x=0}^{N-1} (\hat{a}_k(0,x)\mathrm{e}^{\mathrm{i}\varphi^k(x)}\,|0,x,+_\mathrm{i}\rangle + \hat{a}_k(1,x)\mathrm{e}^{-\mathrm{i}\varphi^k(x)}\,|1,x,+_\mathrm{i}\rangle) \\
&\quad - \sum_{x=0}^{N-1} (\hat{a}_k(0,x)\mathrm{e}^{\mathrm{i}\varphi^k(x)}\,|0,x,+_\mathrm{i}\rangle + \hat{a}_k(1,x)\mathrm{e}^{-\mathrm{i}\varphi^k(x)}\,|1,x,+_\mathrm{i}\rangle) \\
&= 2\cos(\theta^k_{neu})|\hat{\Psi}_0\rangle \\
&\quad - \sum_{x=0}^{N-1} (\hat{a}_k(0,x)\mathrm{e}^{\mathrm{i}\varphi^k(x)}\,|0,x,+_\mathrm{i}\rangle + \hat{a}_k(1,x)\mathrm{e}^{-\mathrm{i}\varphi^k(x)}\,|1,x,+_\mathrm{i}\rangle) \\
&= \sum_{x=0}^{N-1} ((2\cos(\theta^k_{neu}) - \hat{a}_k(0,x)\mathrm{e}^{\mathrm{i}\varphi^k(x)})\,|0,x,+_\mathrm{i}\rangle \\
&\qquad + (2\cos(\theta^k_{neu}) - \hat{a}_k(1,x)\mathrm{e}^{-\mathrm{i}\varphi^k(x)})\,|1,x,+_\mathrm{i}\rangle)
\end{aligned} \tag{6.8}$$

berechnen. Dabei ergibt sich der neue gewichtete Mittelwert beziehungsweise die neue Konvexkombination der $\cos(\varphi^k(x))$ folgendermaßen:

$$\begin{aligned}
\cos(\theta^k_{neu}) &= \langle\hat{\Psi}_0| \sum_{x=0}^{N-1} (\hat{a}_k(0,x)\mathrm{e}^{\mathrm{i}\varphi^k(x)}\,|0,x,+_\mathrm{i}\rangle + \hat{a}_k(1,x)\mathrm{e}^{-\mathrm{i}\varphi^k(x)}\,|1,x,+_\mathrm{i}\rangle) \\
&= \frac{1}{\sqrt{2N}} \sum_{y=0}^{N-1} (\hat{a}_k(0,y)e^{\mathrm{i}\varphi^k(y)} + \hat{a}_k(1,y)e^{-\mathrm{i}\varphi^k(y)}).
\end{aligned} \tag{6.9}$$

Die Amplituden von $|\hat{\Psi}_{k+1}\rangle$ sind dann

$$\hat{a}_{k+1}(0,x) = 2\cos(\theta^k_{neu})\hat{a}_0(0,x) - \hat{a}_k(0,x)\mathrm{e}^{\mathrm{i}\varphi^k(x)}, \tag{6.10}$$

$$\hat{a}_{k+1}(1,x) = 2\cos(\theta^k_{neu})\hat{a}_0(1,x) - \hat{a}_k(1,x)\mathrm{e}^{-\mathrm{i}\varphi^k(x)}. \tag{6.11}$$

Erste Iteration. Für $k = 0$ wird $\varphi^0(x) = \epsilon^0_x$ gewählt. Außerdem gilt: $\cos(\theta^0_{neu})$ ist genau $\langle\hat{\Psi}_0|\mathbf{U}_{\varphi^0}|\hat{\Psi}_0\rangle = \cos(\theta)$ aus Algorithmus 2. Der Zustand $|\hat{\Psi}_1\rangle$ ist genau das Ergebnis der

nicht-binären Amplitudenverstärkung aus Kapitel 4 nach der ersten Iteration. Der Term $\cos(\theta)$ ist nach unserer Voraussetzung größer als 0. Für diesen Zustand $|\hat{\Psi}_1\rangle$ wurde in Abschnitt 4.4 bereits gezeigt, dass im Fall $\cos(\theta) > 0$ die Amplituden der Basiszustände $|x\rangle$ mit $\cos(\varphi^0(x)) < \cos(\theta)$ zulasten derer mit $\cos(\varphi^0(x)) > \cos(\theta)$ verstärkt werden. Dies folgt direkt aus (4.25).

Die Wahl $[a, b] = [0, \pi/2]$ hat in Abschnitt 5.4 gute Ergebnisse geliefert. Wenn x der Minimierer ist, ist $\varphi^0(x) = \epsilon_x^0 \approx \frac{\pi}{2}$ und somit $\cos(\varphi^0(x)) \approx 0 < \cos(\theta)$. Nach der ersten Iteration sind die Amplituden also schon richtig gewichtet. Je besser x das Problem löst, desto größer ist die Amplitude des dazugehörigen Basiszustands. Da die erste Iteration noch nicht von NBAA abweicht, können wir die Ergebnisse der Analyse von NBAA (Abschnitt 4.4) nutzen.

Weitere Iterationen. Für $k > 0$ führen wir nun eine Analyse der Wahrscheinlichkeit vor der Messung eines beliebigen Basiszustands $|x\rangle$ durch. Interessanterweise ist $\cos(\theta_{neu}^k)$ ein gewichteter Mittelwert. Um diesen analysieren zu können, müssen wir uns die komplexen Koeffizienten $\hat{a}_k(0, y)$ und $\hat{a}_k(1, y)$ anschauen. Jede komplexe Zahl $z = x + \mathrm{i}y$ hat eine Schreibweise $z = |z|\mathrm{e}^{\mathrm{i}\phi}$, wobei $|z|$ der Betrag und ϕ die Phase von z sind. Drücken wir die komplexen Amplituden in dieser Form aus, bekommen wir

$$\hat{a}_k(0, y) = |\hat{a}_k(0, y)|\mathrm{e}^{i\phi_k(0,y)} \quad \text{und} \quad \hat{a}_k(1, y) = |\hat{a}_k(1, y)|\mathrm{e}^{i\phi_k(1,y)}. \tag{6.12}$$

Aus (6.7), (6.10) und (6.11) folgt: Für alle $y \in \{0, ..., N-1\}$ und $k \geq 0$ gilt

$$\hat{a}_k(0, y) = \overline{\hat{a}_k(1, y)}. \tag{6.13}$$

Mit (6.13) und (6.9) erhalten wir

$$\cos(\theta_{neu}^k) = \frac{1}{\sqrt{2N}} \sum_{y=0}^{N-1} 2|\hat{a}_k(0, y)| \cos(\phi_k(0, y) + \varphi^k(y)). \tag{6.14}$$

Man sieht, dass je größer die vorherigen Amplituden $\hat{a}_k(0, y)$ waren, desto stärker tragen die entsprechenden Basiszustände zum Mittelwert bei.

Das Verstärken der Amplituden der Minimierer in der ersten Iteration führt dazu, dass sie mehr zu $\cos(\theta_{neu}^k)$ beitragen als die Maximierer, deren Amplituden in der ersten Iteration verkleinert wurden. Für die neue Wahrscheinlichkeit, einen Basiszustand $|0, x\rangle$ zu messen, folgt aus (6.10) und mit $\hat{a}_0(0, x) \in \mathbb{R}$

$$\begin{aligned} |\hat{a}_{k+1}(0, x)|^2 &= 4\cos^2(\theta_{neu}^k)\hat{a}_0(0, x)^2 \\ &\quad - 4\cos(\theta_{neu}^k)\hat{a}_0(0, x)|\hat{a}_k(0, x)| \cos(\phi_k(0, x) + \epsilon_x^k) \\ &\quad + |\hat{a}_k(0, x)|^2. \end{aligned} \tag{6.15}$$

Stellen wir diese Gleichung um und setzen $\hat{a}_0(0, x) = \frac{1}{\sqrt{2N}}$ aus (6.7) ein, ergibt sich

$$\begin{aligned} |\hat{a}_{k+1}(0, x)|^2 &= |\hat{a}_k(0, x)|^2 + 4\cos^2(\theta_{neu}^k)(\tfrac{1}{\sqrt{2N}})^2 \\ &\quad - 4\cos(\theta_{neu}^k)\tfrac{1}{\sqrt{2N}}|\hat{a}_k(0, x)| \cos(\phi_k(0, x) + \epsilon_x^k) \\ &= |\hat{a}_k(0, x)|^2 + \tfrac{4}{\sqrt{2N}} \cos(\theta_{neu}^k)\mu_k(x) \end{aligned} \tag{6.16}$$

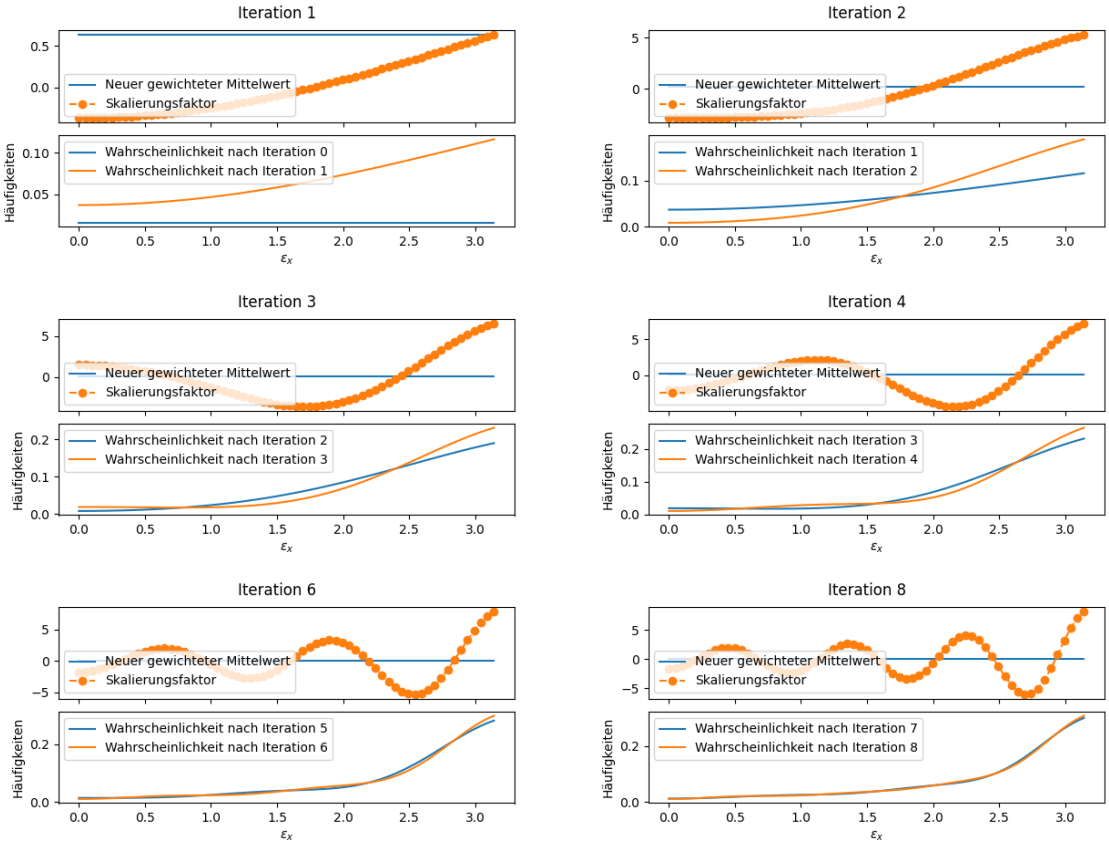


Abbildung 6.2: Beispielhaft klassisch berechnete Änderung der Amplituden. Es wurde $x \in [0, \pi]$ und $\epsilon_x = x$ gewählt. Für jede Iteration k ist $\cos(\theta^k_{neu})$ als neuer gewichteter Mittelwert, der Skalierungsfaktor $\mu_k(x)$ und die Wahrscheinlichkeit, den zu ϵ_x gehörenden Zustand nach $k-1$ und nach k Iterationen zu messen, dargestellt. Es fällt auf, dass der Skalierungsfaktor für $\epsilon_x \approx \pi$ immer positiv ist und deutlich größer als für andere Winkel. Wird also der Lösungszustand mit einer Phase nahe an π markiert, so wird dieser am stärksten verstärkt. Weiterhin ist zu sehen, dass die Wahrscheinlichkeit für die Zustände x mit $\epsilon_x \approx \pi$ mit jeder Iteration zunimmt. Allerdings werden auch manchmal Zustände, die keine Lösung sind, verstärkt.

mit

$$\mu_k(x) := \cos(\theta^k_{neu})\frac{4}{\sqrt{2N}} - |\hat{a}_k(0,x)|\cos(\phi_k(0,x) + \epsilon^k_x). \tag{6.17}$$

Ist $\cos(\theta^k_{neu}) > 0$, wird die Amplitude des zu x gehörenden Basiszustands umso größer, je größer $\mu_k(x)$ ist. Der Parameter $\mu_k(x)$ ist also ein Skalierungsfaktor für die Amplitudenänderung. Wenn für die neuen Kosten $|\hat{a}_k(0,x)|\cos(\phi_k(0,x) + \epsilon^k_x) = \cos(\theta^k_{neu})\frac{4}{\sqrt{2N}}$ gilt, dann ist $|a_{k+1}(0,x)|^2 = |a_k(0,x)|^2$. Wenn dies nicht der Fall ist, werden die Amplituden verändert.

Die Änderung der Amplituden wurde beispielhaft klassisch berechnet und in Abbildung 6.2 veranschaulicht. Dabei wurde $x \in [0, \pi]$ und $\epsilon_x = x$ gewählt. Für jede Iteration k sind der neue gewichtete Mittelwert $\cos(\theta^k_{neu})$, der Skalierungsfaktor $\mu_k(x)$ und die Wahrscheinlichkeit, den zu ϵ_x gehörenden Zustand nach $k-1$ und nach k Iterationen zu messen, dargestellt.

Es fällt auf, dass der Skalierungsfaktor für $\epsilon_x \approx \pi$ immer positiv ist und deutlich größer als für andere Winkel. Markieren wir also unseren Lösungszustand mit einer Phase

nahe an π, so wird dieser am stärksten verstärkt. Im unteren Teil der Grafiken, in dem die Amplituden dargestellt sind, sehen wir, dass die Wahrscheinlichkeit für die Messung der Zustände x mit $\epsilon_x \approx \pi$ mit jeder Iteration größer wird. Allerdings werden auch Zustände mit einer Phase, die nicht nahe π ist, verstärkt. Da die Amplituden dieser Zustände aber in der Iteration davor schon sehr klein sind, werden die dazugehörigen Eingaben immer noch deutlich seltener als die der Lösung gemessen. Wir wollen diese Beobachtungen nun mathematisch untermauern.

Der Winkel $\phi_k(0,x)$. Um die Veränderung der Amplituden besser zu verstehen, betrachten wir zunächst den Winkel $\phi_k(0,x)$.

Durch Einsetzen von $k=0$ in (6.10) und (6.11) folgt für die Amplituden nach einer Iteration:

$$\hat{a}_1(0,x) = 2\cos(\theta)\hat{a}_0(0,x) - \hat{a}_0(0,x)\cos(\varphi^0(x)) - \mathrm{i}\hat{a}_0(0,x)\sin(\varphi^0(x)), \tag{6.18}$$
$$\hat{a}_1(1,x) = 2\cos(\theta)\hat{a}_0(0,x) - \hat{a}_0(1,x)\cos(\varphi^0(x)) + \mathrm{i}\hat{a}_0(1,x)\sin(\varphi^0(x)). \tag{6.19}$$

Wir wollen nun die Amplituden in der Form

$$\hat{a}_1(0,x) = |\hat{a}_1(0,x)|\mathrm{e}^{i\phi_1(0,x)} \quad \text{und} \tag{6.20}$$
$$\hat{a}_1(1,x) = |\hat{a}_1(1,x)|\mathrm{e}^{i\phi_1(1,x)} \tag{6.21}$$

schreiben. Es gilt mit (6.13)

$$\begin{aligned} |\hat{a}_1(0,x)|^2 &= |\hat{a}_1(1,x)|^2 \\ &= 4\cos^2(\theta)\hat{a}_0(0,x)^2 + \hat{a}_0(0,x)^2 - 4\cos(\theta)\hat{a}_0(0,x)\cos(\varphi^0(x)) \end{aligned} \tag{6.22}$$

mit $\cos(\theta)$ aus (4.17). Bezeichne $Im(z)$ den Imaginärteil einer komplexen Zahl z und $Re(z)$ den Realteil. Da nach (6.7) gilt, dass $\hat{a}_0(0,x) = \frac{1}{\sqrt{2N}} \in \mathbb{R}$ für alle x, folgt mit (6.18) für die Phase der Amplituden

$$\begin{aligned} \phi_1(0,x) &= \arctan\left(\frac{Im(\hat{a}_1(0,x))}{Re(\hat{a}_1(0,x))}\right) \\ &= \arctan\left(\frac{-\hat{a}_0(0,x)\sin(\varphi^0(x))}{2\cos(\theta)\hat{a}_0(0,x) - \hat{a}_0(0,x)\cos(\varphi^0(x))}\right). \end{aligned} \tag{6.23}$$

Die Amplituden $\hat{a}_1(0,x)$ (6.18) und $\hat{a}_1(1,x)$ (6.19) unterscheiden sich nur im Vorzeichen ihres Imaginärteils, weil nach (6.7) $\hat{a}_0(0,x) = \hat{a}_0(1,x)$ gilt. Somit gilt

$$\phi_1(1,x) = -\phi_1(0,x). \tag{6.24}$$

Dies lässt sich induktiv mit (6.10) und (6.11) sowie (6.13) weiterführen. Folglich gilt $\phi_k(1,x) = -\phi_k(0,x)$ für ein beliebiges k. Daher betrachten wir im weiteren Verlauf nur $\phi_k(0,x)$. Wir machen zur Analyse die folgenden vereinfachenden Annahmen:

- Da $\frac{1}{\sqrt{2N}}$ bei großen N annähernd verschwindet, gilt dann $\phi_1(0,x) \approx 0$ für alle x.
- Mit (6.18) und (6.19) folgt daraus $\hat{a}_1(0,x)^2 \approx |\hat{a}_1(0,x)|^2$ und $\hat{a}_1(1,x)^2 \approx |\hat{a}_1(1,x)|^2$.

Iterieren wir ein weiteres Mal, erhalten wir mit (6.10) und (6.11) die Amplituden

$$\hat{a}_2(0,x) \approx 2\cos(\theta^1_{neu})\hat{a}_0(0,x) - |\hat{a}_1(0,x)|\cos(\varphi^1(x)) - \mathrm{i}|\hat{a}_1(0,x)|\sin(\varphi^1(x)), \tag{6.25}$$

$$\hat{a}_2(1,x) \approx 2\cos(\theta^1_{neu})\hat{a}_0(0,x) - |\hat{a}_1(0,x)|\cos(\varphi^1(x)) + \mathrm{i}|\hat{a}_1(0,x)|\sin(\varphi^1(x)). \tag{6.26}$$

Für die Phase $\phi_2(0,x)$ gilt dann analog zu (6.23)

$$\phi_2(0,x) \approx \arctan\left(\frac{-|\hat{a}_1(0,x)|\sin(\varphi^1(x))}{2\cos(\theta^1_{neu})\hat{a}_0(0,x) - |\hat{a}_1(0,x)|\cos(\varphi^1(x))}\right). \tag{6.27}$$

Sei x^* eine Lösung. Wie schon davor nehmen wir an, dass $\epsilon_{min} = \epsilon_{x^*} \approx -D$ aus (5.3). Ab der zweiten Iteration wird $\varphi^1(x) = \epsilon^1_x$ mit $a = 0$ und $b = \pi$ aus (5.4) berechnet. Also gilt $\varphi^1(x^*) = \epsilon^1_{x^*} \approx \pi$. Setzen wir diese Approximation in (6.27) ein, folgt

$$\phi_2(0,x^*) \approx 0, \tag{6.28}$$

da $\sin(\pi) = 0$. Für alle weiteren Iterationen gilt ebenfalls $\varphi^k(x^*) \approx \pi$ und somit bleibt $\phi_k(0,x^*) \approx 0$. Setzen wir dies in (6.15) ein, gilt für die Amplituden der Minimierer

$$\begin{aligned} |\hat{a}_{k+1}(0,x^*)|^2 \approx\ & 4\cos^2(\theta^k_{neu})\hat{a}_0(0,x^*)^2 \\ & - 4\cos(\theta^k_{neu})\hat{a}_0(0,x^*)|\hat{a}_k(0,x^*)|\cos(0+\pi) \\ & + |\hat{a}_k(0,x^*)|^2. \end{aligned} \tag{6.29}$$

Da $\cos(0+\pi) = -1$, wächst $\hat{a}_{k+1}(0,x^*)$ im Vergleich zu $\hat{a}_k(0,x^*)$ solange $\cos(\theta^k_{neu})$ positiv ist. Für Nicht-Lösungszustände $\overline{x}$ gilt hingegen $\varphi(\overline{x}) \in [0,\pi)$; für diesen allgemeineren Fall ist keine Abschätzung für $\phi_k(0,\overline{x})$ bekannt.

Abschätzen der Amplituden. Wir nutzen die Erkenntnisse des letzten Abschnitts, um nun die Amplituden abzuschätzen. Mit der ersten Iteration haben wir sichergestellt, dass $|\hat{a}_1(0,x^*)| > |\hat{a}_1(0,\overline{x})|$, das heißt die Amplitude der Lösungszustände sind betraglich größer als die der Nicht-Lösungen. Wir führen dies induktiv weiter: Bezeichne x^* eine Lösung und $\overline{x}$ eine Nicht-Lösung. Wie im vorigen Abschnitt nehmen wir an, dass $\varphi^k(x^*) \approx \pi$ und somit $\phi_k(0,x^*) \approx 0$ für alle Lösungszustände x^* gilt. Sei $k > 0$ und

$$|\hat{a}_k(0,x^*)|^2 > |\hat{a}_k(0,\overline{x})|^2. \tag{6.30}$$

Wir zeigen nun

$$|\hat{a}_{k+1}(0,x^*)|^2 > |\hat{a}_{k+1}(0,\overline{x})|^2 \tag{6.31}$$

für alle Lösungen x^* und alle Nicht-Lösungen $\overline{x}$. Es gilt mit (6.30)

$$|\hat{a}_k(0,x^*)|\cos(0+\pi) = -|\hat{a}_k(0,x^*)| < |\hat{a}_k(0,\overline{x})|\underbrace{\cos(\phi_k(0,\overline{x}) + \varphi^k(\overline{x}))}_{\in[-1,1]}. \tag{6.32}$$

Ist $\cos(\theta_{neu}^k) \geq 0$, dann folgt aus (6.32) und mit $\hat{a}_0(0,\overline{x}) = \hat{a}_0(0,x^*) = \frac{1}{\sqrt{2N}}$

$$\begin{aligned} &-4\cos(\theta_{neu}^k)\hat{a}_0(0,x^*)|\hat{a}_k(0,x^*)|\cos(0+\pi) \\ &\qquad \geq -4\cos(\theta_{neu}^k)\hat{a}_0(0,\overline{x})|\hat{a}_k(0,\overline{x})|\cos(\phi_k(0,\overline{x})+\varphi^k(\overline{x})) \end{aligned} \tag{6.33}$$

Mit (6.16) und (6.29) folgt

$$|\hat{a}_{k+1}(0,x^*)|^2 > |\hat{a}_{k+1}(0,\overline{x})|^2. \tag{6.34}$$

Die Beträge der Amplituden der Basiszustände der Minimierer sind demnach immer größer als die der Nicht-Lösungen, solange $\cos(\theta_{neu}^k)$ nicht negativ ist.

Anzahl der Iterationen

Die Ergebnisse der letzten Abschnitte zeigen, dass in jeder Iteration k wichtig ist, dass $\cos(\theta_{neu}^k)$ nicht negativ ist. Es kann also so lange iteriert werden, bis $\cos(\theta_{neu}^k)$ negativ wird. Die optimale Anzahl der Iterationen ist demnach genau das K, für das $\cos(\theta_{neu}^K) > 0$ und $\cos(\theta_{neu}^{K+1}) < 0$ gilt.

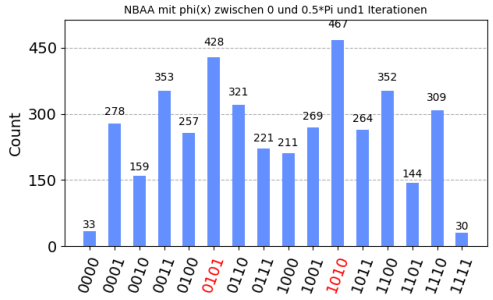
Für die optimale Anzahl der Iterationen K konnte im Rahmen dieser Arbeit leider keine geschlossene Formel gefunden werden. Daher orientieren wir uns an Analysen der Grover-Phase-Matching-Bedingung im binären Fall [31, 22], deren zufolge im binären Fall mit einer einzelnen Lösung, die mit der Phase π markiert wurde, $\sqrt{N}$ Iterationen notwendig sind, um die Amplitude des Minimierers zu maximieren. Wenn die Phase des Minimierers kleiner als π ist, dann sind im Allgemeinen mehr als $\sqrt{N}$ Iterationen erforderlich.

Da wir mit unserer Korrektur nur π annähern und es in der Regel sehr wenige globale Minimierer gibt, ist $\sqrt{N}$ daher eine untere Grenze der optimalen Anzahl von Iterationen. Wir wählen daher $K = \lfloor\sqrt{2^n}\rfloor = \lfloor\sqrt{N}\rfloor$. Diese Anzahl an Iterationen lieferte in unseren numerischen Experimenten auch in der Praxis gute Ergebnisse.

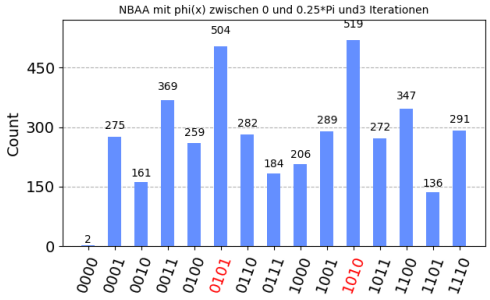
Es ist zu vermuten, dass die optimale Anzahl der Iterationen vom anfänglichen Wert von $\cos(\theta)$ abhängt und demnach von dem Anteil der markierten Elemente in der Gesamtzahl der Elemente, genau wie beim verallgemeinerten Grover-Algorithmus. Eine entsprechende mathematische Analyse wäre eine interessante Aufgabe für zukünftige Arbeiten.

Komplexität des Algorithmus

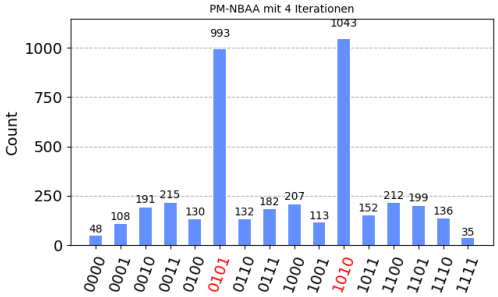
Da die Anzahl der Iterationen abhängig von N gewählt wird, ändert sich auch die Komplexität des Algorithmus. Wie bei NBAA können die Initialisierung sowie die Diffusion und der 2-Register-Orakel-Aufruf mit einer Gatteranzahl in $\mathcal{O}(\log(N)^2)$ implementiert werden. Die von uns gewählte Anzahl der Iterationen ist in $\mathcal{O}(\sqrt{N})$. Insgesamt folgt, dass die Komplexität einer Ausführung von PM-NBAA in $\mathcal{O}(\sqrt{N}\log(N)^2)$ liegt. Der gesamte Algorithmus muss mehrfach ausgeführt werden, da die Wahrscheinlichkeit, den Minimierer zu messen, am höchsten im Vergleich zur Messung andere Zustände ist, aber nicht 1.



(a) NBAA mit $b = \pi/2$, Anzahl der Iterationen $= 1$



(b) NBAA mit $b = \pi/4$, Anzahl der Iterationen $= 3$



(c) PM-NBAA, Anzahl der Iterationen $= 4$

Abbildung 6.3: NBAA und PM-NBAA auf das Ising Problem aus Beispiel 6.1 angewendet mit 4096 Messungen. Bei NBAA wurde φ auf das Intervall $[0, b]$ mit $b = \pi/4$ und $b = \pi/2$ skaliert (siehe dafür Abschnitt 5.1) und die Anzahl der Iterationen wurde wie in Abschnitt4.5 gewählt. Bei PM-NBAA wurde $\lfloor\sqrt{2^n}\rfloor$-mal, also viermal, iteriert. Die Minimierer des Ising-Problems sind „0101" uns „1010". In beiden Algorithmen werden die Minimierer am häufigsten gemessen. Vergleichen wir $b = \pi/4$ und $b = \pi/2$, wurden die Minimierer gegenüber der anderen Zustände bei $b = \pi/4$ häufiger gemessen. Im Histogramm des Algorithmus mit Phasenanpassung sind die Minimierer aber am deutlichsten zu erkennen. PM-NBAA braucht allerdings mehr Iterationen als NBAA.

Zusammenfassung

Der vorgestellte Algorithmus PM-NBAA verstärkt wie NBAA die Amplitude der zur Lösung gehörenden Basiszustände und ab der zweiten Iteration ist auch die Grover-Phase-Matching-Bedingung näherungsweise erfüllt. Wir konnten zeigen, dass die Beträge der Amplituden der Basiszustände der Minimierer immer größer sind als die der Nicht-Lösungen, solange $\cos(\theta^k_{neu})$ nicht negativ ist. Eine Ausführung von PM-NBAA zum Lösen des Ising-Problems hat eine Komplexität von $\mathcal{O}(\sqrt{N}\log(N)^2)$ bezüglich der Gatteranzahl.

6.3 Numerische Experimente für PM-NBAA

Bisher wurden die zwei Algorithmen NBAA und PM-NBAA zum Lösen des Ising-Problems vorgestellt und untersucht. Im Folgenden werden diese miteinander und mit dem UQIsing-

Algorithmus [28] verglichen. Uns interessiert vor allem, ob die Phasenkorrektur das approximation ratio und die Wahrscheinlichkeit, das Minimum zu messen, erhöht, und ob NBAA und PM-NBAA mit bereits existierenden Verfahren konkurrieren können. Wir verwenden Algorithmus 3, der in Abschnitt 6.1 vorgestellt wurde (PM-NBAA), Algorithmus 2 (NBAA) mit $b = \frac{1}{2}\pi$ und $b = \frac{1}{4}\pi$ sowie den UQIsing Algorithmus aus [28].

Vergleich von NBAA und PM-NBAA

Wir wenden NBAA und PM-NBAA beispielhaft auf ein Ising-Problem an. Für das Ising-Problem aus Beispiel 6.1 erhalten wir nach 4096 Messungen die in Abbildung 6.3 gezeigten Histogramme. Die Minimierer des Ising-Problems sind wieder „0101" und „1010". In beiden Algorithmen werden die Minimierer am häufigsten gemessen. Im Histogramm von PM-NBAA sind die Minimierer aber deutlicher zu erkennen.

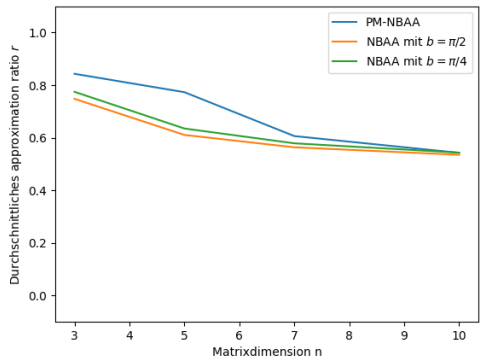
Um die Güte der Algorithmen experimentell zu vergleichen, verwenden wir nun das approximation ratio und die Wahrscheinlichkeit, das Minimum zu messen (siehe Abschnitt 5.4). Die Ergebnisse wurden wie in Abschnitt 5.4 ermittelt und sind in Abbildung 6.4 dargestellt.

In Abbildung 6.4 (a) werden die approximation ratios der beiden Algorithmen in Abhängigkeit von der Matrixdimension der Kostenmatrix n betrachtet. Die Phasenanpassung hat das approximation ratio in den niedrigen Dimensionen verbessert. Bei $n = 3$ ist das approximation ratio von PM-NBAA maximal mit dem ungefähren Wert 0.85. In den höheren Problemgrößen nimmt das approximation ratio beider Algorithmen ab. Für $n = 10$ nehmen NBAA mit $b = 1/2\pi$, NBAA mit $b = 1/4\pi$ und PM-NBAA ähnliche Werte an, die ungefähr bei 0.6 liegen. Betrachtet man die Histogramme, hätte man ein deutlich erhöhtes approximation ratio erwartet. Dies ist aber nicht der Fall, da bei PM-NBAA zwar die Amplituden der Lösungszustände deutlich erhöht werden, dafür aber die Amplituden der Nicht-Lösungszustände, also insbesondere auch des zum Maximum gehörenden Basiszustands nicht so stark verringert werden. Das Maximum fließt also noch stärker in den Erwartungswert ein und das approximation ratio wird somit nicht verbessert.

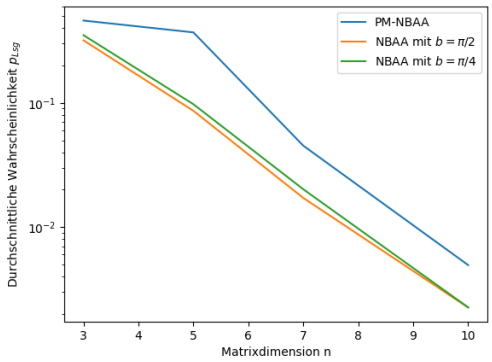
In Abbildung 6.4 (b) sind die Wahrscheinlichkeiten, die Lösung zu messen, in Abhängigkeit von n dargestellt. Die Wahrscheinlichkeiten des PM-NBAA sind immer größer als die von NBAA mit $b = 1/2\pi$ und mit $b = 1/4\pi$. Die Wahrscheinlichkeiten nehmen mit steigender Problemgröße exponentiell ab.

Betrachtet man das Verhältnis der Wahrscheinlichkeiten $p_{\text{Lsg}}^{\text{PM-NBAA}}/p_{\text{Lsg}}^{\text{NBAA}}$ in Abbildung 6.4 (c), fällt auf, dass gerade in höheren Problemgrößen sich die Wahrscheinlichkeit des Messens der Lösung beim Phase-Matching im Vergleich zur Wahrscheinlichkeit bei NBAA verdoppelt.

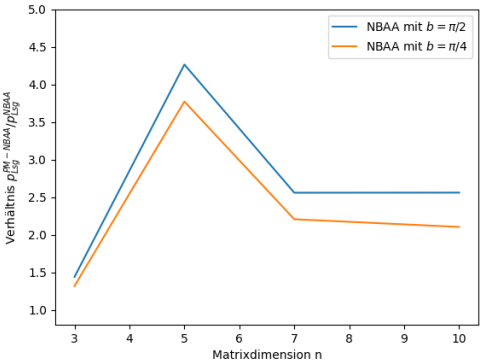
Um die Wahl von $K = \lfloor\sqrt{N}\rfloor$ auch experimentell zu begründen, wird in Abbildung 6.4 (d) auch das durchschnittliche Verhältnis von der Anzahl der Lösungen M zur Gesamtanzahl der Eingaben N der untersuchten Ising-Probleme dargestellt. In Abschnitt 3.5 wurde geschlussfolgert, dass für $M/N < 0.3$ die Grover-Phase-Matching-Bedingung im Vergleich zu anderen Phase-Matching-Bedingungen die höchste Wahrscheinlichkeit liefert, eine Lösung des Suchproblems zu messen. Wie aus der Grafik zu entnehmen ist, war in unseren Experimenten $M/N < 0.3$ und somit die Wahl von K berechtigt.



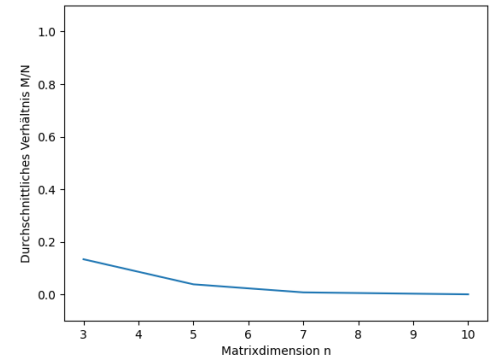
(a) Das durchschnittliche approximation ratio r in Abhängigkeit der Matrixdimension n der Kostenmatrix.



(b) Die durchschnittliche Wahrscheinlichkeit p_{Lsg} in Abhängigkeit der Matrixdimension n der Kostenmatrix.



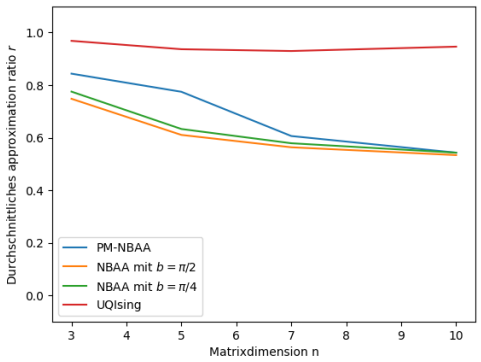
(c) Verhältnis der Wahrscheinlichkeiten $p_{\text{Lsg}}^{\text{PM-NBAA}}/p_{\text{Lsg}}^{\text{NBAA}}$ mit $b = 1/2\pi$ und $b = 1/4\pi$.



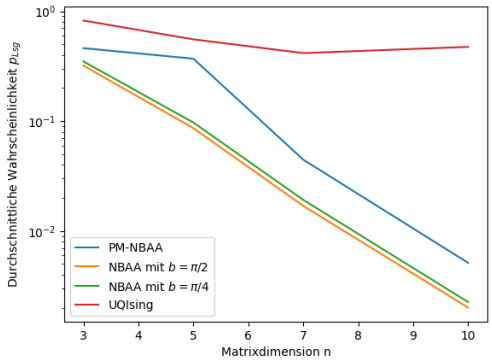
(d) Durchschnittliches Verhältnis von der Anzahl der Lösungen M zur Gesamtanzahl der Eingaben N.

Abbildung 6.4: Quantitativer Vergleich von NBAA mit $b = 1/2\pi$ sowie $b = 1/4\pi$ mit PM-NBAA. Die durchschnittlichen Werte wurden aus den Werten von 40 zufällig erzeugten Ising-Problemen berechnet. In (a) werden die approximation ratios der beiden Algorithmen in Abhängigkeit von der Matrixdimension der Kostenmatrix n betrachtet. Die Phasenanpassung hat das approximation ratio in den niedrigen Dimensionen verbessert. Bei $n = 3$ ist das approximation ratio von PM-NBAA maximal mit dem ungefähren Wert 0.85. In den höheren Problemgrößen nimmt das approximation ratio beider Algorithmen ab. Für $n = 10$ nehmen NBAA mit $b = 1/2\pi$, NBAA mit $b = 1/4\pi$ und PM-NBAA ähnliche Werte an, die ungefähr bei 0.6 liegen. In (b) sind die Wahrscheinlichkeiten, die Lösung zu messen, in Abhängigkeit von n dargestellt. Die Wahrscheinlichkeiten des PM-NBAA sind für alle getesteten Problemgrößen n deutlich größer als die von NBAA mit $b = 1/2\pi$ und mit $b = 1/4\pi$. Die Wahrscheinlichkeiten nehmen mit steigender Problemgröße exponentiell ab. Betrachtet man in (c) das Verhältnis der Wahrscheinlichkeiten $p_{\text{Lsg}}^{\text{PM-NBAA}}/p_{\text{Lsg}}^{\text{NBAA}}$, so fällt auf, sich bei Benutzung von PM-NBAA insbesondere in höheren Problemgrößen die Wahrscheinlichkeit des Messens der Lösung im Vergleich zu NBAA etwa verdoppelt. In (d) ist das durchschnittliche Verhältnis der Anzahl der Lösungen M zur Gesamtanzahl der Eingaben N der untersuchten Ising-Probleme dargestellt. Es fällt auf, dass dieses immer unter 0.3 ist. Mit den Ergebnissen aus Abschnitt 3.5 folgt daher, dass bei unseren Experimenten die Grover-Phase-Matching-Bedingung im Vergleich zu anderen Phase-Matching-Bedingungen die besten Ergebnisse liefert. Außerdem ist dadurch die Wahl von $K = \lfloor\sqrt{N}\rfloor$ auch experimentell begründet.

Vergleichen wir die Komplexität der Algorithmen, schneidet NBAA deutlich besser ab. In beiden Algorithmen wird pro Iteration ein Orakel angewendet und eine Diffusion durchgeführt. Diese lassen sich bei beiden Algorithmen mit ähnlicher Komplexität implementieren. Relevant ist daher die Anzahl der Iterationen. Während NBAA mit $b = \frac{1}{2}\pi$



(a) Das durchschnittliche approximation ratio r in Abhängigkeit der Matrixdimension n der Kostenmatrix.



(b) Die durchschnittliche Wahrscheinlichkeit p_{Lsg} in Abhängigkeit der Matrixdimension n der Kostenmatrix.

Abbildung 6.5: Vergleich von NBAA mit $b = 1/2\pi$ und $b = 1/4\pi$ und PM-NBAA mit UQIsing [28] mithilfe verschiedener Metriken. In (a) werden die approximation ratios der drei Algorithmen in Abhängigkeit von der Matrixdimension der Kostenmatrix n betrachtet. In (b) sind die Wahrscheinlichkeiten, die Lösung zu messen, in Abhängigkeit von n dargestellt. Für beide Metriken liefert UQIsing bessere Werte.

beziehungsweise $b = \frac{1}{4}\pi$ nur ein bis drei Iterationen benötigt (siehe Abschnitt 5.4), führen wir in PM-NBAA $\sqrt{N}$ Iterationen durch. Die Komplexität des PM-NBAA ist also deutlich höher. Allerdings ist die Wahrscheinlichkeit, die Lösung zu messen, bei PM-NBAA höher als bei NBAA. Im folgenden Abschnitt vergleichen wir beide Algorithmen nun mit UQIsing.

Vergleich von NBAA mit UQIsing

Wir vergleichen NBAA und PM-NBAA mit UQIsing [28], um zu untersuchen, ob die Amplitudenverstärkungsalgorithmen mit bereits bestehenden Verfahren mithalten können.

Hierfür werden wie im vorigen Abschnitt das approximation ratio und die Wahrscheinlichkeit, die Lösung zu messen, untersucht. Alle Werte wurden wieder wie in Abschnitt 5.4 erzeugt.

Es fällt auf, dass sowohl das approximation ratio, als auch die Wahrscheinlichkeit, die Lösung zu messen, bei UQIsing deutlich besser waren als bei PM-NBAA und NBAA. Allerdings ist bei PM-NBAA und NBAA die Wahrscheinlichkeit, die Zustände der globalen Minimierer zu messen, immer höher als die Wahrscheinlichkeit, einen anderen Basiszustand zu messen. Dies kann bei UQIsing nicht garantiert werden [28, Abbildung 7]. Außerdem wurde bei UQIsing öfter iteriert und der Algorithmus muss auch mehrfach ausgeführt werden. Die Eigenschaften der drei Algorithmen sind in Tabelle 6.6 zusammengefasst.

Zusammenfassung

Insgesamt können also PM-NBAA und NBAA in Bezug auf das approximation ratio noch nicht ganz mit dem bereits existierenden Verfahren UQIsing mithalten. Bei beiden Ver-

Verfahren	Anzahl der Iterationen	mehrfaches Ausführen des Verfahrens nötig	approximation ratio	$p(m = x^*)$	Basiszustände der Lösungen immer am häufigsten gemessen
NBAA	1-3	ja	0.6 - 0.8	abhängig von der Matrixdimension	ja
PM-NBAA	$\sqrt{N}$	ja	0.6 - 0.85, etwas höher als bei NBAA	etwas 1-4-fache Wahrscheinlichkeit im Vergleich zu NBAA	ja
UQIsing	30	ja	≈ 0.95	deutlich größer als die Wahrscheinlichkeit von NBAA und PM-NBAA	nein

Tabelle 6.6: Experimentelle Eigenschaften von NBAA, PM-NBAA und UQIsing im Überblick.

fahren ist es garantiert, dass die Basiszustände der Lösungen des Problems im Vergleich zu den anderen Basiszuständen am häufigsten gemessen werden, was bei vielen anderen Verfahren nicht der Fall ist (siehe zum Beispiel UQIsing [28]). Das in dieser Arbeit entwickelte Verfahren PM-NBAA hat eine höhere Komplexität als NBAA, weil öfter iteriert werden muss. Allerdings ist die Wahrscheinlichkeit, die Lösung zu messen, bei PM-NBAA größer als die von NBAA. Das approximation ratio wird beim PM-NBAA nur bei niedrigen Problemgrößen leicht verbessert und kann auch nicht mit [28] mithalten. Dennoch können sowohl NBAA als auch PM-NBAA zum Suchen der Lösung des Ising-Problems verwendet werden und die Wahrscheinlichkeit, die Zustände der globalen Minimierer zu messen, ist immer höher als das Messen anderer Basiszustände.

7

Fazit

In dieser Arbeit wurden Algorithmen der binären und nicht-binären Amplitudenverstärkung beschrieben und untersucht. Die Ergebnisse der Arbeit werden im nächsten Abschnitt zusammengefasst. Anschließend wird ein Ausblick über weiterführende Fragestellungen und Verbesserungsmöglichkeiten gegeben.

7.1 Zusammenfassung der Arbeit

In dieser Arbeit wurde ein schaltkreis-basiertes Quantencomputing-Verfahren vorgestellt und auf das Ising-Problem angewendet. Das betrachtete Verfahren NBAA [37] ist eine nicht-binäre (Quanten-)Amplitudenverstärkung und ist ein iteratives Verfahren. Für die Implementierung des Orakels wurde aus [28] übernommen und die Kodierung des Problems angepasst. Die Komplexität des Orakels liegt in $\mathcal{O}(log(N)^2)$ bezüglich der Gatteranzahl. Bei NBAA ist die Wahrscheinlichkeit, die Lösung des Ising-Problems zu messen, immer am höchsten im Vergleich zur Wahrscheinlichkeit, eine Nicht-Lösung zu messen.

Um die Wahrscheinlichkeit, die Lösung zu messen, und das approximation ratio zu erhöhen, wurde NBAA angepasst, so dass die Grover-Phase-Matching-Bedingung ab der zweiten Iteration näherungsweise erfüllt ist. Die Ausführung dieses Algorithmus (PM-NBAA) liegt in $\mathcal{O}(\sqrt{N}\log(N)^2)$ bezüglich der Gatteranzahl. Zum Lösen des Ising-Problems muss der Algorithmus aber mehrfach ausgeführt werden.

Experimentell wurde durch die Anwendung von PM-NBAA im Vergleich zu NBAA die Wahrscheinlichkeit, die Lösung zu messen, erhöht, das approximation ratio aber nicht. Außerdem wurden beide Algorithmen mit UQIsing [28] verglichen. Auch hierbei wurden die Wahrscheinlichkeit, die Lösung zu messen, und das approximation ratio benutzt. Sowohl NBAA als auch PM-NBAA können in Bezug auf die beiden Metriken nicht mit UQIsing mithalten. Das approximation ratio beschreibt, wie nahe der Erwartungswert der Energieniveaus der gemessenen Zustände an dem minimalen Energieniveau des Ising-Problems, liegt. Wenn uns also nur die Wahrscheinlichkeit interessiert, den Minimierer zu messen, und die Wahrscheinlichkeiten, die Nicht-Lösungen zu messen, nur in dem Sinne, dass sie geringer sind als die der Lösung, dann ist das approximation ratio nicht geeignet, um die Effektivität des Algorithmus zu bewerten.

Beide in dieser Arbeit vorgestellten Algorithmen, um das Ising-Problem global zu

lösen, liegen im Bereich des Quantencomputings. Leider sind die Ressourcen und Laufzeiten im Quantencomputing momentan aber noch beschränkt und es können somit noch nicht beliebig große Probleme in der Praxis gelöst werden.

7.2 Ausblick

Neben der hier gezeigten Erfüllung der Grover-Phase-Matching-Bedingung gibt es im Hinblick auf zukünftige Entwicklungen einige weitere interessante Ansätze, um NBAA zu verbessern.

Eine mögliche Forschungsrichtung wären alternative Implementierungen des 1-Register-Orakels, etwa [12]. Eine interessante Fragestellung ist, ob diese eine effizientere Implementierung ermöglichen als in dieser Arbeit.

Weiterhin können sowohl der Startzustand als auch $\varphi(x)$ anders gewählt werden. Insbesondere kann Vorwissen, wenn es vorhanden ist, bei der Wahl des Startzustands einbezogen werden. Bei der Wahl von $\varphi(x)$ wurde nur eine begrenzte Anzahl an Intervallen $[a, b]$ betrachtet. Gerade das zusätzliche Variieren von a kann weiterführend zu dieser Arbeit untersucht werden. Außerdem kann für den Winkel $\varphi(x)$ auch eine nicht-affine Funktion benutzt werden. Ein vielversprechender Ansatz wäre, die natürliche Exponentialfunktion oder den natürlichen Logarithmus zu benutzen. Bei geschickter Wahl könnte dann die Grover-Phase-Matching-Bedingung auch schon bei NBAA erfüllt sein. Wird aber $\varphi(x)$ eine nicht-affine Funktion, muss auch die Implementierung des 1-Register-Orakels angepasst werden. Diese basiert in dieser Arbeit darauf, dass $\varphi(x)$ eine affine Funktion ist.

Der nächste Schritt, um PM-NBAA zu verbessern, ist, die Anzahl der Iterationen zu optimieren: In der aktuellen Implementierung wird $\sqrt{N}$-mal iteriert. Um die optimale Anzahl an Iterationen zu finden, könnte $\cos(\theta^k_{neu})$ analytisch untersucht und die Iteration abgebrochen werden, sobald $\cos(\theta^k_{neu})$ nicht mehr positiv ist.

Literatur

[1] Aharonov, D. Quantum computation. In: *Annual Reviews of Computational Physics VI*:259–346, 1999.

[2] Albash, T. und Lidar, D. A. Adiabatic quantum computation. In: *Reviews of Modern Physics* 90(1):015002, 2018.

[3] Bauckhage, C., Brito, E., Cvejoski, K., Ojeda, C., Sifa, R. und Wrobel, S. Ising Models for Binary Clustering via Adiabatic Quantum Computing. In: Jan. 2018, S. 3–17. ISBN: 978-3-319-78198-3. DOI: 10.1007/978-3-319-78199-0_1.

[4] Bennett, C. H. Logical Reversibility of Computation. In: *IBM Journal of Research and Development* 17(6):525–532, 1973. DOI: 10.1147/rd.176.0525.

[5] Bertsekas, D. *Network optimization: continuous and discrete models*. Bd. 8. Athena Scientific, 1998.

[6] Born, M. und Fock, V. Beweis des Adiabatensatzes. In: *Zeitschrift für Physik* 51(3-4):165–180, 1928.

[7] Brassard, G. und Høyer, P. An exact quantum polynomial-time algorithm for Simon's problem. In: *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems*:12–23, 1997.

[8] Brassard, G., Høyer, P., Mosca, M., Montreal, A., Aarhus, B. U. of und Waterloo, C. U. of Quantum Amplitude Amplification and Estimation. In: *arXiv: Quantum Physics*, 2000.

[9] Chowdhury, S. R., Baruah, S. und Dikshit, B. Phase matching in quantum search algorithm. In: *Europhysics Letters* 141(5):58001, 2023. DOI: 10.1209/0295-5075/acba41. URL: https://dx.doi.org/10.1209/0295-5075/acba41.

[10] Cross, A. W. The IBM Q experience and QISKit open-source quantum computing software. In: *Bulletin of the American Physical Society* 2018, 2018. URL: https://api.semanticscholar.org/CorpusID:67150463.

[11] D-Wave Systems: QPU Solver Datasheet. In: 2022. URL: https://docs.dwavesys.com/docs/latest/docqpu.html.

[12] Farhi, E., Goldstone, J. und Gutmann, S. A quantum approximate optimization algorithm. In: *arXiv preprint arXiv:1411.4028*, 2014.

[13] FCatherine McGeoch, P. F. Advantage Processor Overview. In: 2014. URL: https://www.dwavesys.com/media/3xvdipcn/14-1058a-a_advantage_processor_overview.pdf.

[14] Goemans, M. X. und Williamson, D. P. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. In: *J. ACM* 42:1115–1145, 1995.

[15] Grover, L. K. A fast quantum mechanical algorithm for database search. In: *Symposium on the Theory of Computing*. 1996.

[16] Grover, L. K. Quantum Computers Can Search Rapidly by Using Almost Any Transformation. In: *Physical Review Letters* 80:4329–4332, 1997.

[17] Grover, L. K. Quantum Mechanics Helps in Searching for a Needle in a Haystack. In: *Physical Review Letters* 79:325–328, 1997.

[18] GuiLu, L., WeiLin, Z., YanSong, L. und Li, N. Arbitrary phase rotation of the marked state cannot be used for Grover's quantum search algorithm. In: *Communications in Theoretical Physics* 32(3):335, 1999.

[19] Göllmann, L. *Lineare Algebra: im algebraischen Kontext*. Jan. 2020. ISBN: 978-3-662-61737-3. DOI: 10.1007/978-3-662-61738-0.

[20] Hagberg, A., Swart, P. und Chult, D. Exploring Network Structure, Dynamics, and Function Using NetworkX. In: Jan. 2008.

[21] Homeister, M. *Quantum Computing verstehen: Grundlagen – Anwendungen – Perspektiven*. Wiesbaden: Springer Fachmedien Wiesbaden, 2022. ISBN: 978-3-658-36434-2. DOI: 10.1007/978-3-658-36434-2. URL: https://doi.org/10.1007/978-3-658-36434-2.

[22] Høyer, P. Arbitrary phases in quantum amplitude amplification. In: *Physical Review A* 62(5):052304, 2000.

[23] Jerrum, M. und Sinclair, A. Polynomial-time approximation algorithms for the Ising model. In: *SIAM Journal on computing* 22(5):1087–1116, 1993.

[24] Jordan, M. I. Graphical models. In: 2004.

[25] Kitaev, A. Y. Quantum measurements and the Abelian stabilizer problem. In: *arXiv preprint quant-ph/9511026*, 1995.

[26] Knabner, P. und Barth, W. *Lineare Algebra: Grundlagen und Anwendungen*. Jan. 2013. ISBN: 3-642-32185-2. DOI: 10.1007/978-3-662-55600-9.

[27] Koch, D., Cutugno, M., Patel, S., Wessing, L. und Alsing, P. M. Variational Amplitude Amplification for Solving QUBO Problems. In: *arXiv preprint arXiv:2301.13665*, 2023.

[28] Kuete Meli, N., Mannel, F. und Lellmann, J. A universal quantum algorithm for weighted maximum cut and Ising problems. In: *Quantum Information Processing* 22(7):279, 2023.

[29] Kuete Meli, N., Mannel, F. und Lellmann, J. An Iterative Quantum Approach for Transformation Estimation from Point Sets. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, S. 519–527. DOI: 10.1109/CVPR52688.2022.00061.

[30] Li, P. und Li, S. Phase matching in Grover's algorithm. In: *Physics Letters A* 366(1-2):42–46, 2007.

[31] Long, G. L., Li, Y. S., Zhang, W. L. und Niu, L. Phase matching in quantum searching. In: *Physics Letters A* 262(1):27–34, 1999.

[32] Luongo, A. *Quantum algorithms for data analysis*. 2020. URL: https://quantumalgorithms.org.

[33] McCoy, B. M. und Wu, T. T. The Two-Dimensional Ising Model. In: 1973. URL: https://api.semanticscholar.org/CorpusID:124565259.

[34] Mesbahi, M. Quantum Computing Impact on Databases M30208, Database Management. In: Nov. 2020.

[35] Nielsen, M. A. und Chuang, I. L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511976667.

[36] Qin, P. und Zhao, J. A polynomial-time algorithm for image segmentation using Ising models. In: *2011 Seventh International Conference on Natural Computation*. Bd. 2. 2011, S. 932–935. DOI: 10.1109/ICNC.2011.6022198.

[37] Shyamsundar, P. Non-Boolean Quantum Amplitude Amplification and Quantum Mean Estimation. In: 2021.

[38] Toyama, F., Van Dijk, W, Nogami, Y, Tabuchi, M und Kimura, Y Multiphase matching in the Grover algorithm. In: *Physical Review A* 77(4):042324, 2008.

[39] Younes, A. Strength and Weakness in Grover's Quantum Search Algorithm. In: *arXiv: Quantum Physics*, 2008. URL: https://api.semanticscholar.org/CorpusID:15416275.

A

Anhang

A.1 Detaillierte Herleitung des 1-Register-Orakels als Rotation

Für die Implementierung des 1-Register-Orakels als Rotation (Abschnitt 5.1) wurde ein extra Cost-Qubit eingeführt. Nun soll für ein beliebiges x das Cost-Qubit um $\tilde{\epsilon}_x$ (5.4) gedreht werden. Sei D der betraglich maximale Wert aller ϵ_x (5.2):

$$D := \sum_{i=1}^{n} |C_{ii}| + \sum_{1 \le i < j \le n} |C_{ij}|. \tag{A.1}$$

Weiterführend seien der Skalierungs- und Verschiebungsfaktor wie folgt definiert:

$$d_1 := \frac{(b-a)}{2D} \quad \text{und } d_2 := \frac{a+b}{2}. \tag{A.2}$$

Die Rotation R_y (5.7) um den Winkel

$$2\tilde{\epsilon}_x = -2d_1\epsilon_x + 2d_2 \tag{A.3}$$

mit ϵ_x aus (5.2) lässt sich folgendermaßen umformen:

$$\begin{aligned}
R_y(2\tilde{\epsilon}_x) &= R_y\Big(-2d_1\epsilon_x + 2d_2 \Big) && \text{(A.4)}\\
&= R_y\Big(-2d_1\epsilon_x \Big)\, R_y(2d_2) && \text{(A.5)}\\
&= R_y\Big(-2d_1 \Big(\sum_{i=1}^{n} C_{ii}(-1)^{x_i} + \sum_{1 \le i < j \le n} C_{ij}(-1)^{x_i + x_j} \Big) \Big)\, R_y(2d_2) && \text{(A.6)}\\
&= R_y\Big(-2d_1 \sum_{i=1}^{n} C_{ii}(-1)^{x_i} \Big) && \text{(A.7)}\\
&\quad R_y\Big(-2d_1 \sum_{1 \le i < j \le n} C_{ij}(-1)^{x_i + x_j} \Big)\, R_y(2d_2).
\end{aligned}$$

Außerdem gilt: Für alle $x_i \in \{0, 1\}$ und alle $\theta \in \mathbb{R}$ ist

$$R_y(2\theta(-1)^{x_i}) = X^{x_i} R_y(2\theta) X^{x_i}. \tag{A.8}$$

Beweis. Sei $x_i = 0$, dann gilt:

$$R_y(2\theta\,(-1)^{x_i}) = R_y(2\theta) = I_2 R_y(2\theta) I_2 = X^{x_i} R_y(2\theta) X^{x_i} \tag{A.9}$$

Sei $x_i = 1$, dann gilt:

$$R_y(2\theta\,(-1)^{x_i}) = R_y(-2\theta) = \begin{pmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{pmatrix} \tag{A.10}$$

$$= \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \tag{A.11}$$

und es gilt

$$X^{x_i} R_y(2\theta) X^{x_i} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{A.12}$$

$$= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} -\sin(\theta) & \cos(\theta) \\ \cos(\theta) & \sin(\theta) \end{pmatrix} \tag{A.13}$$

$$= \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}. \tag{A.14}$$

Insgesamt folgt dann (A.8) für alle $x_i \in \{0, 1\}$ und alle $\theta \in \mathbb{R}$. □

Setzen wir (A.8) in (A.7) ein folgt

$$R_y(2\tilde{\epsilon}_x) = R_y\Big(-2d_1 \sum_{i=1}^{n} C_{ii}(-1)^{x_i} \Big) \tag{A.15}$$

$$R_y\Big(-2d_1 \sum_{1\le i<j\le n} C_{ij}(-1)^{x_i+x_j} \Big)\, R_y(2d_2)$$

$$= \prod_{i=1}^{n} X^{x_i} R_y\Big(-2d_1 C_{ii} \Big) X^{x_i} \tag{A.16}$$

$$\prod_{1\le i<j\le n} X^{x_i+x_j} R_y\Big(-2d_1 C_{ij} \Big) X^{x_i+x_j} \cdot R_y(2d_2).$$

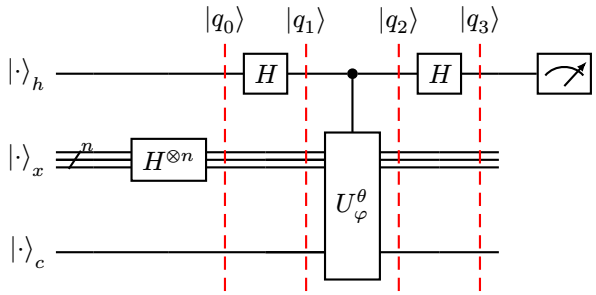


Abbildung A.1: Schaltkreis zum Schätzen von $\cos(\theta)$. Das Hilfsqubit $|h\rangle$ wird im Zustand $|0\rangle$, das n-Qubit-Regster wird mit $H^{\otimes n}$ in den perfekt überlagerten Zustand und das Cost-Qubit $|c\rangle$ in den Zustand $|0\rangle$ initialisiert. Anschließend wird auf $|h\rangle$ erst ein Hadamard-Gatter, dann das veränderte Orakel U_φ^θ von $|h\rangle$ kontrolliert auf $|x\rangle$ und $|c\rangle$ und dann noch ein Hadamard-Gatter auf $|h\rangle$ angewendet. Am Ende wird $|h\rangle$ in der Rechenbasis gemessen.

A.2 Beweis: Approximation von $\cos(\theta)$

Einer der Vorteile der Implementierung des 1-Register-Orakels als Rotation ist es, dass $\cos(\theta)$ (4.17) mit wenig Rechenaufwand geschätzt werden kann (siehe auch Abschnitt 5.3). Um $\cos(\theta)$ zu approximieren, wird $\tilde{U}_\varphi$ (5.12) so modifiziert, dass der Operator hermitesch ist:

$$U_\varphi^\theta = \begin{pmatrix} R_y^\theta(2\varphi(0)) & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots \\ \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & R_y^\theta(2\varphi(N-1)) \end{pmatrix}. \tag{A.17}$$

Die Matrix $R_y^\theta(2\varphi(x))$ ist für alle x folgendermaßen definiert:

$$R_y^\theta(2\varphi(x)) = \begin{pmatrix} \cos(\varphi(x)) & \sin(\varphi(x)) \\ \sin(\varphi(x)) & -\cos(\varphi(x)) \end{pmatrix}. \tag{A.18}$$

Wir wollen nun zeigen, dass die geschätzten Wahrscheinlichkeiten $p(0)$ und $p(1)$, die nach Ausführen des Schaltkreises in Abbildung A.1 erhalten wurden, mit

$$p(0) - p(1) = \cos(\theta) \tag{A.19}$$

zum Approximieren von $\cos(\theta)$ genutzt werden können.

Beweis. Der Startzustand ist

$$|q_0\rangle = |0\rangle \otimes \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \otimes |0\rangle . \tag{A.20}$$

Nachdem H auf $|h\rangle$ angewendet wurde, ist das System im Zustand

$$|q_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \otimes |0\rangle \tag{A.21}$$

$$= \frac{1}{\sqrt{2}} \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} (|0\rangle + |1\rangle) \otimes |x\rangle \otimes |0\rangle . \tag{A.22}$$

Der Zustand nach dem Anwenden von U_φ^θ kontrolliert durch $|h\rangle$ ist:

$$|q_2\rangle = \frac{1}{\sqrt{2}}\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}\Big[\,|0\rangle\otimes|x\rangle\otimes|0\rangle + |1\rangle\otimes U_\varphi^\theta|x\rangle\otimes|0\rangle\,\Big] \tag{A.23}$$

$$= \frac{1}{\sqrt{2}}\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}\Big[\,|0\rangle\otimes|x\rangle\otimes|0\rangle \tag{A.24}$$

$$+ |1\rangle\otimes|x\rangle\otimes(\cos(\varphi(x))\,|0\rangle + \sin(\varphi(x))\,|1\rangle)\Big]$$

$$= \frac{1}{\sqrt{2}}|0\rangle\otimes\left(\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle\right)\otimes|0\rangle \tag{A.25}$$

$$+ \frac{1}{\sqrt{2}}|1\rangle\otimes\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}\Big[\,|x\rangle\otimes(\varphi(x))\,|0\rangle + \sin(\varphi(x))\,|1\rangle)\Big].$$

Nachdem H erneut auf $|h\rangle$ angewendet wurde, ist das System im Zustand

$$|q_3\rangle = \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\otimes\left(\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle\right)\otimes|0\rangle \tag{A.26}$$

$$+ \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)\otimes\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}\Big[\,|x\rangle\otimes(\cos(\varphi(x))\,|0\rangle + \sin(\varphi(x))\,|1\rangle)\Big]$$

$$= \frac{1}{2}\frac{1}{\sqrt{N}}(|0\rangle+|1\rangle)\otimes\left(\sum_{x=0}^{N-1}|x\rangle\right)\otimes|0\rangle \tag{A.27}$$

$$+ \frac{1}{2}\frac{1}{\sqrt{N}}(|0\rangle-|1\rangle)\otimes\sum_{x=0}^{N-1}\Big[\,\cos(\varphi(x))\,|x\rangle\otimes|0\rangle + \sin(\varphi(x))\,|x\rangle\otimes|1\rangle\,\Big]$$

$$= \frac{1}{2}\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}\Big[\,|0,0,x\rangle + |1,0,x\rangle + \cos(\varphi(x))\,|0,0,x\rangle \tag{A.28}$$

$$+ \sin(\varphi(x))\,|0,1,x\rangle - \cos(\varphi(x))\,|1,0,x\rangle - \sin(\varphi(x))\,|1,1,x\rangle\Big]$$

$$= \frac{1}{2}\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}\Big[\,|0,0,x\rangle + \cos(\varphi(x))\,|0,0,x\rangle + \sin(\varphi(x))\,|0,1,x\rangle \tag{A.29}$$

$$+ |1,0,x\rangle - \cos\varphi(x))\,|1,0,x\rangle - \sin(\varphi(x))\,|1,1,x\rangle\Big]$$

$$= \frac{1}{2}\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}\Big[(1+\cos(\varphi(x)))\,|0,0,x\rangle + \sin(\varphi(x))\,|0,1,x\rangle \tag{A.30}$$

$$+ (1-\cos(\varphi(x)))\,|1,0,x\rangle - \sin(\varphi(x))\,|1,1,x\rangle\Big].$$

Für die Wahrscheinlichkeiten „0" beziehungsweise „1" zu erhalten, wenn $|h\rangle$ gemessen wird, ergibt sich dann

$$p(0) = \frac{1}{4}\frac{1}{N}\sum_{x=0}^{N-1}\Big[(1+\cos(\varphi(x)))^2 + \sin(\varphi(x))^2\Big] \tag{A.31}$$

$$= \frac{1}{4}\frac{1}{N}\sum_{x=0}^{N-1}\Big[1+\cos(\varphi(x))^2 + 2\cos(\varphi(x)) + \sin(\varphi(x))^2\Big] \tag{A.32}$$

und

$$p(1) = \frac{1}{4}\frac{1}{N}\sum_{x=0}^{N-1}\left[(1-\cos(\varphi(x)))^2 + \sin(\varphi(x))^2\right] \tag{A.33}$$

$$= \frac{1}{4}\frac{1}{N}\sum_{x=0}^{N-1}\left[1+\cos(\varphi(x))^2 - 2\cos(\varphi(x)) + \sin(\varphi(x))^2\right] \tag{A.34}$$

und somit gilt

$$p(0) - p(1) = \frac{1}{4}\frac{1}{N}\sum_{x=0}^{N-1}\left[2\cos(\varphi(x)) - (-2\cos(\varphi(x)))\right] \tag{A.35}$$

$$= \frac{1}{N}\sum_{x=0}^{N-1}\cos(\varphi(x)) \tag{A.36}$$

$$= \cos(\theta). \tag{A.37}$$

□

A.3 Detaillierte Herleitung des Iterationsverhaltens von PM-NBAA

In Abschnitt 6.2 wird unter anderem das Iterationsverhalten von PM-NBAA untersucht. Dies wird hier mit mehr Details hergeleitet: Beschreibe

$$|\hat{\Psi}_k\rangle = \sum_{x=0}^{N-1}(\hat{a}_k(0,x)\,|0,x,+_\mathrm{i}\rangle + \hat{a}_k(1,x)\,|1,x,+_\mathrm{i}\rangle) \tag{A.38}$$

den Zustand nach $k \in \{0,...,K\}$ Iterationen. Betrachten wir nun eine Iteration mit $k > 0$, dann lässt sich aus $|\hat{\Psi}_k\rangle$

$$|\hat{\Psi}_{k+1}\rangle = \mathbf{S}_{\hat{\Psi}_0}\hat{\mathbf{U}}_{\varphi^k}\,|\hat{\Psi}_k\rangle \tag{A.39}$$

$$= \mathbf{S}_{\hat{\Psi}_0}\hat{\mathbf{U}}_{\varphi^k}\sum_{x=0}^{N-1}(\hat{a}_k(0,x)\,|0,x,+_\mathrm{i}\rangle + \hat{a}_k(1,x)\,|1,x,+_\mathrm{i}\rangle) \tag{A.40}$$

$$= 2|\hat{\Psi}_0\rangle\langle\hat{\Psi}_0|\sum_{x=0}^{N-1}(\hat{a}_k(0,x)\mathrm{e}^{\mathrm{i}\varphi^k(x)}\,|0,x,+_\mathrm{i}\rangle + \hat{a}_k(1,x)\mathrm{e}^{-\mathrm{i}\varphi^k(x)}\,|1,x,+_\mathrm{i}\rangle) \tag{A.41}$$

$$- \sum_{x=0}^{N-1}(\hat{a}_k(0,x)\mathrm{e}^{\mathrm{i}\varphi^k(x)}\,|0,x,+_\mathrm{i}\rangle + \hat{a}_k(1,x)\mathrm{e}^{-\mathrm{i}\varphi^k(x)}\,|1,x,+_\mathrm{i}\rangle)$$

$$= 2\cos(\theta^k_{neu})|\hat{\Psi}_0\rangle \tag{A.42}$$

$$- \sum_{x=0}^{N-1}(\hat{a}_k(0,x)\mathrm{e}^{\mathrm{i}\varphi^k(x)}\,|0,x,+_\mathrm{i}\rangle + \hat{a}_k(1,x)\mathrm{e}^{-\mathrm{i}\varphi^k(x)}\,|1,x,+_\mathrm{i}\rangle)$$

$$= \sum_{x=0}^{N-1}((2\cos(\theta^k_{neu}) - \hat{a}_k(0,x)\mathrm{e}^{\mathrm{i}\varphi^k(x)})\,|0,x,+_\mathrm{i}\rangle \tag{A.43}$$

$$+ (2\cos(\theta^k_{neu}) - \hat{a}_k(1,x)\mathrm{e}^{-\mathrm{i}\varphi^k(x)})\,|1,x,+_\mathrm{i}\rangle)$$

berechnen. Der neue gewichtete Mittelwert der $\cos(\varphi^k(x))$ ergibt sich folgendermaßen:

$$\cos(\theta^k_{neu}) = \langle\hat{\Psi}_0| \sum_{x=0}^{N-1} (\hat{a}_k(0,x)\mathrm{e}^{\mathrm{i}\varphi^k(x)} |0,x,+_\mathrm{i}\rangle + \hat{a}_k(1,x)\mathrm{e}^{-\mathrm{i}\varphi^k(x)} |1,x,+_\mathrm{i}\rangle) \tag{A.44}$$

$$= \frac{1}{\sqrt{2N}} \sum_{y=0}^{N-1} \hat{a}_k(0,y)e^{i\varphi^k(y)} + \hat{a}_k(1,y)e^{-i\varphi^k(y)}. \tag{A.45}$$