

BACHELORARBEIT

Evaluation von quantum-klassisch hybriden Lösungsverfahren für 3SAT-Probleme

Philip Hierhager



BACHELORARBEIT

Evaluation von quantum-klassisch hybriden Lösungsverfahren für 3SAT-Probleme

Philip Hierhager

Aufgabensteller: Prof. Dr. Claudia Linnhoff-Popien

Betreuer: Sebastian Zielinski
Jonas Stein

Abgabetermin: 10. Dezember 2021



Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 10. Dezember 2021

.....
(Unterschrift des Kandidaten)

Abstract

Das Ziel der vorliegenden Arbeit ist die Bewertung quantum-klassisch hybrider Lösungsverfahren für schwere 3SAT-Probleme (Klausel-zu-Variablen Verhältnis von ungefähr 4,2). 3SAT ist eine Art des Erfüllbarkeitsproblems der Aussagenlogik und NP-vollständig. Letzteres bezeichnet eine schwere Komplexitätsklasse. Es ist effizient reduzierbar auf viele andere NP-vollständige Probleme und stellt einen einfach zu formulierenden Prototyp dar. Das Verfahren besteht im Lösen einer mit der 3SAT-Formel assoziierten Instanz eines ganzzahligen linearen Programmes. Die erhaltene Zuordnung wird dann in eine Belegung eines Problems der maximalen stabilen Menge (MIS) umgewandelt und als Anfangszustand eines Reverse Annealing Schrittes verwendet. Um das Konzept zu evaluieren, werden Lösungszeiten des Verfahrens mit denen eines modernen SAT-Solvers verglichen. Des Weiteren wird versucht, einige Eigenschaften des quantum-klassisch hybriden Verfahrens und der Lösungslandschaft der 3SAT-Probleme anhand der Ergebnisse des Reverse Annealing zu ermitteln. Dafür wurde eine quantitative Studie mithilfe der Reverse Annealing Funktion des Quantum Computers *Advantage 4.1* der Firma *D-Wave Systems* durchgeführt. Die Auswertung von fast 2,5 Millionen Lösungen des Quantum Annealers zeigte interessante Ergebnisse. Der Skalierung des Verfahrens sind jedoch durch die Quantencomputer Hardware momentan noch Grenzen gesetzt und die Lösungszeit des klassischen Lösungsverfahrens kann weiter verbessert werden. Es werden zusätzlich Eigenschaften der Lösungslandschaft des Verfahrens gezeigt. So wird festgestellt, dass das globale Minimum und große lokale Minima des zu einem MIS umgewandelten 3SAT-Problem sehr breit sind und hohe Barrieren besitzen. Des Weiteren sind lokale Minima in der Nähe des globalen Minimums eher durch korrekte aber nicht optimale Zuordnungen des stabilen Mengenproblems (IS) zu beschreiben. Die großen lokalen Minima hingegen lassen sich durch inkorrekte IS beschreiben, die aber durch ihre hohe Kardinalität teilweise als 3SAT-Lösung interpretierbar sind.

Inhaltsverzeichnis

1. Einleitung	1
2. MIS und SAT	3
2.1. Das Erfüllbarkeitsproblem - SAT	3
2.2. Das Stabilitätsproblem - IS/MIS	4
2.3. Transformationsmethode zwischen 3-SAT und MIS	4
2.4. Transformationsmethode zwischen 3-SAT und MILP	6
2.5. Hamming-Distanz	9
3. Lineare Programmierung	11
3.1. Mathematische Beschreibung	11
3.1.1. Geometrische Interpretation	12
3.1.2. Tabellenform	13
3.2. Simplex	14
3.2.1. Simplextableau	14
3.2.2. Simplexverfahren	15
3.2.3. Überarbeitung des Simplexverfahrens	15
3.2.4. Überarbeiteter Simplexalgorithmus	17
3.3. Ganzzahlige Lineare Programmierung	18
3.3.1. Branch and Bound	19
3.3.2. Schnittebenenverfahren	21
3.3.3. Branch and Cut	22
4. Quantum Annealing	23
4.1. Beschreibung	23
4.2. Quanten Bits	24
4.3. Systeme mit mehreren Qubits	25
4.4. Das Ising-Modell	25
4.5. QUBO Darstellung	26
4.6. Minor Embedding	26
4.7. Simulated Annealing und der Tunneleffekt	27
4.8. Reverse Annealing	28
5. Methodik	31
5.1. Vergleich der Qualität der generierten Anfangszustände für das Reverse Annealing	31
5.2. Prüfung der Skalierbarkeit	32
5.3. Vergleich der Lösungszeiten eines 3SAT-Problems	32
5.4. Gurobi 9.1	33
5.5. Dwave Advantage 4.1	33

6. Ergebnisse	35
6.1. Vergleich der Qualität der generierten Anfangszustände für das Reverse Annealing	35
6.1.1. Beschreibung	35
6.1.2. Interpretation	37
6.2. Prüfung der Skalierbarkeit	46
6.2.1. Beschreibung	46
6.2.2. Interpretation	46
6.3. Vergleich der Lösungszeiten eines 3SAT-Problems	49
7. Diskussion	53
8. Fazit	55
A. Komplexitätstheorie	57
A.1. Das Entscheidungsproblem	57
A.2. Das Lambda Kalkül und die Turingmaschine	57
A.3. Komplexitätsklassen	58
A.4. Notation	58
B. Physikalische Grundlagen	61
B.1. Beschreibung Quantenmechanik	61
B.2. Dirac Notation und Operatoren	62
B.3. Messung eines quantenmechanischen Systems	64
B.4. Mathematische Herleitung der Schrödingergleichung	65
C. Beispiel zur Anwendung des Simplexverfahrens	69
Abbildungsverzeichnis	73
Tabellenverzeichnis	77
Literaturverzeichnis	79

1. Einleitung

Menschen sind bequem. Seitdem die Menschheit existiert, entwickelt sie Werkzeuge und Methoden, um einfacher, besser und bequemer leben zu können. Mit der Erfindung einer mit Strom betriebenen Rechenmaschine im 20. Jahrhundert konnte jedoch nicht nur der Körper entlastet werden, sondern auch der Verstand der Menschen. Mehr noch, der Mensch konnte plötzlich mit davor unerreichter Geschwindigkeit und Präzision rechnen. Ein immenser Fortschritt in allen Lebensbereichen ging damit einher, da das Leben von Ressourcenoptimierung geprägt ist. Sei es die Überlegung, wie möglichst viel Nahrung pro Landfläche produziert werden kann, oder was der schnellste Weg von A nach B ist, alle Berechnungen konnten durch die neue Technik weitaus schneller ausgeführt werden. Nachdem neue Methoden und Algorithmen entwickelt wurden, konnten auch vorher unerreichbare Genauigkeiten bei einigen Problemstellungen erzielt werden. Ein Beispiel solcher ist die Proteinfaltung. Durch das Team von *DeepMind* ist es mithilfe maschinellen Lernens gelungen, Proteinfaltung sehr viel genauer als jemals zuvor vorherzusagen. Dies ist beispielsweise hilfreich um Medikamente auch für aktuelle Krankheiten wie zum Beispiel Covid-19 zu entwickeln. (vgl. [EOP+21]) Auch im maschinellen Lernen spielt Optimierung eine große Rolle. Seit einigen Jahren gibt es jedoch eine neue Methode, solche Optimierungsaufgaben möglicherweise noch schneller ausführen zu können. Quantum Annealing ist ein Weg - basierend auf physikalischen Erkenntnissen des letzten Jahrhunderts - solche Probleme zu lösen. Die Methode konnte einige vielversprechende Ergebnisse produzieren. Beispielsweise zeigt eine Arbeitsgruppe von Google 2015 einen erheblichen Zeitvorteil von Quantum Annealing gegenüber seinem klassischen Partner Simulated Annealing ([DBI+15]). Die vorliegende Arbeit beschäftigt sich mit einem möglichen Weg mit Quantum Annealing schwere Probleme effizienter lösen zu können. Dabei wird eine leicht abgewandelte Form des Quantum Annealings - das Reverse Annealing - im Zusammenspiel mit einem klassischen Lösungsverfahren angewandt. Mit einer ähnlichen Frage beschäftigte sich Thomas Gabor, Sebastian Zielinski und weitere in ihrer Arbeit [GZF+19]. Dabei wurden 3SAT-Probleme (3SAT ist eine Art des Erfüllbarkeitsproblems der Aussagenlogik) mithilfe eines reinen Quantum Annealing Ansatzes gelöst. Das Ziel der vorliegenden Arbeit ist jedoch die Evaluation eines quanten-klassisch hybridem Lösungsverfahrens für 3SAT-Probleme. Dabei wird die 3SAT-Instanz in ein MILP (*Mixed Integer Linear Program*) umgewandelt, gelöst und dann als Startzustand für das Reverse Annealing genutzt. Es wird zusätzlich versucht, Rückschlüsse auf die Lösungslandschaft des 3SAT-Problems zu ziehen. 3SAT ist ein schweres (NP-vollständiges), aber trotzdem einfach zu formulierendes Problem. Da es auf viele andere NP-vollständige Problemstellungen effizient reduzierbar ist (vgl. [Kar72]), ist das 3SAT-Problem ein idealer Prototyp für komplexe praktische Anwendungen. Beispielsweise kann es zum *Problem des Handlungsreisenden* reduziert werden, bei dem die kürzeste Route zwischen mehreren Städten gesucht wird. Daher ist das 3SAT-Problem trotz seiner durchaus theoretischen Formulierung doch sehr direkt auf praktische Aufgabenstellungen anzuwenden. Die Methodik dieser Arbeit basiert auf einer quantitativen Studie mithilfe des Quantum Annealers *Ad-*

1. Einleitung

vantage 4.1 von *D-Wave Systems* und dem Optimierungsprogramm *Gurobi* der Firma *Gurobi*. Die vorliegende Arbeit gliedert sich in 7 Kapitel. Die ersten 3 Kapiteln stellen dabei die benötigte Theorie dar, die letzten 4 behandeln dann den praktischen Teil. Die 3 Kapitel des Anhangs präsentieren Themen, deren Umfang zu groß für die eigentliche Arbeit ist. In Kapitel 2 werden die drei zentralen Probleme MIS, SAT und MILP eingeführt. Dann folgt ein Paragraf 3 über lineare Programmierung, die in dem entwickelten hybriden Lösungsansatz eine wichtige Rolle einnimmt. In Sektion 4 folgt dann ein Kapitel über Quantum Annealing. Zu weiterführenden Informationen wird auf den Anhang verwiesen. Appendix A beschreibt die Komplexitätstheorie genauer, Anhang B gibt detaillierte für Quantum Annealing relevante Informationen der Physik und C stellt ein Beispiel für Kapitel 3 dar. Sektion 5 präsentiert den Versuchsaufbau und -beschreibung. Im Kapitel 6 werden dann die Ergebnisse dargestellt. Folgend schließt die Arbeit ab mit einer kurzen Diskussion in 7 und Fazit im Paragrafen 8.

2. MIS und SAT

In dieser Sektion wird das Erfüllbarkeitsproblem der Aussagenlogik (auch SAT) und das Problem der maximalen stabilen Menge (MIS) erläutert. Folgendes Kapitel orientiert sich an [ESB20].

2.1. Das Erfüllbarkeitsproblem - SAT

Das Erfüllbarkeitsproblem der Aussagenlogik fragt, ob eine Formel erfüllbar ist. Das dazugehörige Suchproblem sucht nach einer Belegung von n Variablen $x_1, x_2, \dots, x_n \in \{\text{wahr}, \text{falsch}\}$. Die zwei Werte *wahr* und *falsch* werden oft auch binär als 1 bzw. 0 ausgedrückt. Die Belegung soll so gewählt sein, dass eine Formel F erfüllt wird. Die verwendeten Literale $y_1, y_2, \dots, y_{2n}$ in einer Formel stellen die negierten und positiven Variablen dar. Die Literale sind mit Disjunktionen oder Konjunktionen verknüpft. Die Standardform des Problems heißt *Konjunktive Normalform* (oder CNF). Dabei besteht die Formel aus m Konjunktionen der Klauseln C_i .

$$F = \bigvee_{i \in 1, \dots, m} C_i$$

Jede Klausel besteht aus Disjunktionen von unterschiedlichen Literalen aus der Menge J_i .

$$C_i = \bigwedge_{j \in J_i} y_j$$

Anmerkung 2.1 *Sollten zwei Literale in einer Klausel identisch sein, so ist ein Literal überflüssig.*

Anmerkung 2.2 *Die Disjunktion einer Literal der negierten Variable v und einem Literal der Variable v bewirkt die Korrektheit der Disjunktion unabhängig von der Belegung der Variable.*

Das Erfüllbarkeitsproblem war das erste Problem, für das die NP-Vollständigkeit [1] bewiesen wurde [Coo71]. Es kann auf viele andere Probleme polynomiell reduziert werden. [Kar72]

Definition 2.3 (3SAT) *Das 3SAT-Problem ist eine spezielle Form des Erfüllbarkeitsproblems bei dem genau 3 Variablen pro Klausel in der konjunktiven Normalform enthalten sind. Das allgemeine Erfüllbarkeitsproblem lässt sich polynomiell auf 3SAT reduzieren. Es wird oft als ein einfach zu formulierendes NP-vollständiges Problem verwendet.*

¹Für eine genauere Erklärung der verwendeten Ausdrücke aus der Komplexitätstheorie wird auf den Anhang A verwiesen.

Anmerkung 2.4 (Erstellung von zufälligen 3SAT Instanzen) Für eine zufällige Erzeugung einer 3SAT Instanz mit n Variablen und k Klauseln werden Variablen zufällig für eine Klausel ausgewählt mit der Wahrscheinlichkeit $\frac{1}{2n}$. Es gibt dabei zwei verschiedene Vorgänge. Beim gezwungenen Vorgang wählt man eine Belegung der gewünschten Variablen und sucht dann dazu passende Klauseln. Bei der ungezwungenen Variante werden alle Klauseln in die Formel aufgenommen. Falls die Formel unloesbar ist, wird der Vorgang wiederholt. Wegen Anmerkung [2.1] und [2.2] verwerfen beide Vorgänge Klauseln, in denen sich dieselbe Variable befindet; unabhängig davon, ob deren Literale unterschiedliche Vorzeichen besitzen oder nicht. Da der erste Vorgang tendenziell leichter zu lösen ist als zweiter ([Hoo]), werden in dieser Arbeit zufällige 3SAT Instanzen ungezwungen (oder unforced) erzeugt.

2.2. Das Stabilitätsproblem - IS/MIS

Die Sektion ist angelehnt an [Cho10]. Ein weiteres NP-vollständiges Entscheidungsproblem ist das Stabilitätsproblem (oder IS für Independent Set). Es wird gefragt ob ein Graph G mindestens k Knoten enthält, die eine stabile Menge bilden. Eine stabile Menge eines Graphen enthält nur Knoten, die nicht mit Kanten verbunden sind. Das dazugehörige Optimierungsproblem sucht eine stabile Menge mit maximaler Kardinalität (auch Stabilitätszahl genannt). Es wird als MIS abgekürzt. Ein MIS kann mathematisch beschrieben werden als:

$$Y(x_1, \dots, x_n) = \sum_{i \in V(G)} c_i x_i + \sum_{ij \in E(G)} J_{ij} x_i x_j \quad (2.1)$$

Dabei drückt $V(G)$ die Knotenindexmenge und $E(G)$ die Kantenindexmenge des Graphen G aus.

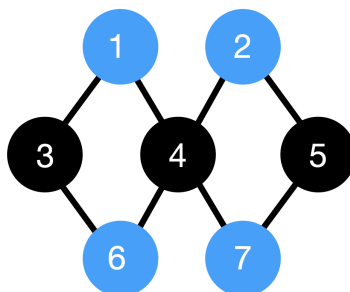


Abbildung 2.1.: Die maximale stabile Menge des Graphen ist $M = \{1, 2, 6, 7\}$. Die Stabilitätszahl des Graphen ist 4.

2.3. Transformationsmethode zwischen 3-SAT und MIS

Das in der Arbeit verwendete Verfahren verlangt nach einer Umwandlung zwischen dem 3SAT Problem und dem der stabilen Menge. In diesem Kapitel werden 3 Verfahren vorgestellt. Dies ist an [Cho10] und [Luc14] orientiert. Die Struktur der Grafiken ist von

[Cho10] inspiriert. Als Beispiel verwendet wird die aussagenlogische Formel

$$F = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3 \vee x_5) \wedge (x_1 \vee \neg x_5 \vee x_6) \quad (2.2)$$

3-SAT zu MIS

Nach [Cho10] können 3-SAT Instanzen leicht zu Graphen eines Stabilitätsproblems umgewandelt werden. Für jedes Literal wird ein Knoten erstellt. Die Knoten der Literale einer Klausel werden miteinander verbunden. Alle Knoten der Literale derselben Variable mit unterschiedlichem Vorzeichen werden durch Kanten verknüpft. Eine 3-SAT Instanz mit k Klauseln ist lösbar, wenn die maximale stabile Menge des assoziierten MIS Problems k unabhängige Knoten besitzt. Die Umwandlung wird anhand eines Beispiels in Abbildung 2.2 dargestellt.

MIS-Zuordnung zu 3-SAT-Zuordnung

Eine stabile Menge, deren Kardinalität der Anzahl der Klauseln der 3-SAT Instanz entspricht, kann in eine Lösung des 3-SAT Problems transformiert werden. Es ist aufgrund des Aufbaus des Graphen ausgeschlossen, dass zwei Literale derselben Variablen mit unterschiedlichen Vorzeichen beide in der stabilen Menge enthalten sind. Daher befindet sich in jeder Klausel mindestens ein Literal, dessen Zuordnung *wahr* ist. Jeder Variable der Literale der Knoten in der stabilen Menge wird der jeweilige Wert zugeordnet. Dann werden allen übrigen Variablen der Wert *falsch* oder *wahr* (bei positivem bzw. negiertem Vorkommen) zugewiesen. Die Transformation wird in Abbildung 2.3 dargestellt. Folgende Beschreibung ist orientiert an [GZF⁺19] von Thomas Gabor, Sebastian Zielinski und weiteren aus dem Jahr 2019. Selbst eine unvollständige MIS-Belegung kann jedoch als korrekte 3-SAT Lösung interpretiert werden. Dabei wird als erstes jeder Variable der Literale der Knoten in der stabilen Menge der jeweilige Wert zugeordnet. Dann werden allen übrigen Variablen der Wert *falsch* oder *wahr* (bei positivem bzw. negiertem Vorkommen) zugewiesen. Nun kann es vorkommen, dass die MIS-Zuordnung zwar nicht korrekt ist, aber die 3-SAT Belegung eine Lösung darstellt. Ein Beispiel wird in Abbildung 2.4 aufgezeigt. Mithilfe eines *Postprocessing* Schrittes können jedoch auch weitere MIS-Zuordnungen in eine korrekte 3-SAT Belegung umgewandelt werden. Dabei wird für die Variablen in den noch inkorrekten Klauseln geprüft, ob sie einen anderen Wert annehmen können, ohne dass andere Klausel inkorrekt werden. Wird ein solcher Fall gefunden, so erhält die Variable einen anderen Wert. Das Konzept wird in Abbildung 2.5 präsentiert. Sollte die MIS-Zuordnung nicht korrekt sein, so wird immer das erste im IS vorkommende Literal verwendet um der dazugehörigen Variable einen Wert zuzuweisen. Sollten mehrere Literale einer Klausel im IS sein, so werden diese einfach so umgewandelt. Es macht keinen Unterschied für die 3SAT Instanz, da auch mehrere Literale einer Klausel den Wert *wahr* annehmen können.

3-SAT-Zuordnung zu MIS-Zuordnung

Bei der Umwandlung in die entgegengesetzte Richtung wird das 3SAT Problem als erstes in ein MIS Problem umgewandelt. Ein Knoten des MIS-Graphen wird nur dann der stabilen Menge hinzugefügt, wenn das korrespondierende Literal der 3SAT Klausel den Wert 1 annimmt. Befinden sich in einer Klausel mehr als ein dem Wert *wahr* zugeordneten Literal, so wird zufaellig eines ausgewaehlt, dessen Knoten im

2. MIS und SAT

MIS Problem dann in die stabile Menge aufgenommen wird. Ein Beispiel wird in Abbildung 2.6 beschrieben.

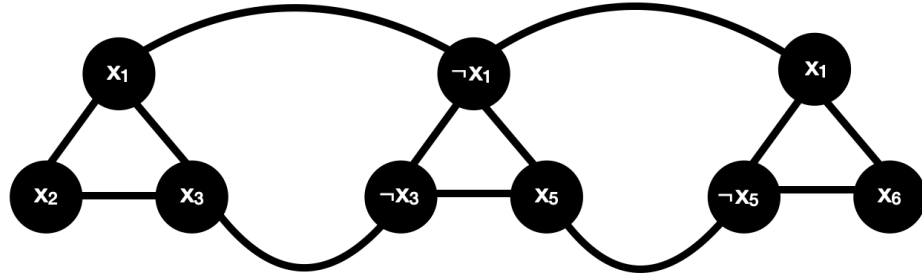


Abbildung 2.2.: Hier wird die Formel 2.2 in einen MIS-Graphen umgewandelt.

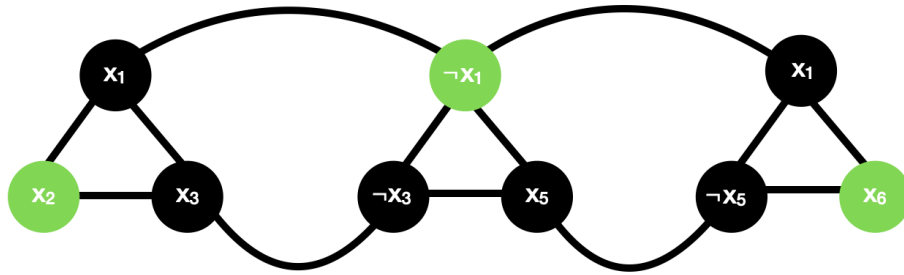


Abbildung 2.3.: Eine maximale stabile Menge des Graphen bilden die grün eingefärbten Knoten. Die Zuordnung kann direkt mit umgewandelt werden in Lösung L der Formel 2.2: $L = \{x_1 \leftarrow 0; x_2 \leftarrow 1; x_3 \leftarrow 0; x_5 \leftarrow 0; x_6 \leftarrow 1\}$

2.4. Transformationsmethode zwischen 3-SAT und MILP

Mixed Integer Linear Programs werden in 3.3 eingeführt. In diesem Kapitel wird eine intuitive Methode vorgestellt, ein 3-SAT Problem in ein MILP umzuwandeln. Eine Lösung eines 3-SAT Problems muss pro Klausel mindestens eine Variable besitzen, deren Wert *wahr* ist. Formuliert man dies in binärer Schreibweise so muss die Summe der Werte der drei Literale einer Klausel größer gleich 1 sein. Um nun ein lineares Programm zu formulieren, werden die Variablen x_i des 3-SAT Problems verwendet, um Nebenbedingungen aufzustellen. Sei C die Menge der Klauseln der 3-SAT Instanz, so muss für die Klausel C_i und $x_a, x_b, x_c \in C_i$ gelten:

$$x_a + x_b + x_c \geq 1$$

und

$$0 \leq x \leq 1$$

Werden alle Variablen auf Ganzzahlen beschränkt, so wird keine Zielfunktion zum Finden einer Lösung benötigt. Jede Zuordnung, die die Nebenbedingungen erfüllt, stellt eine

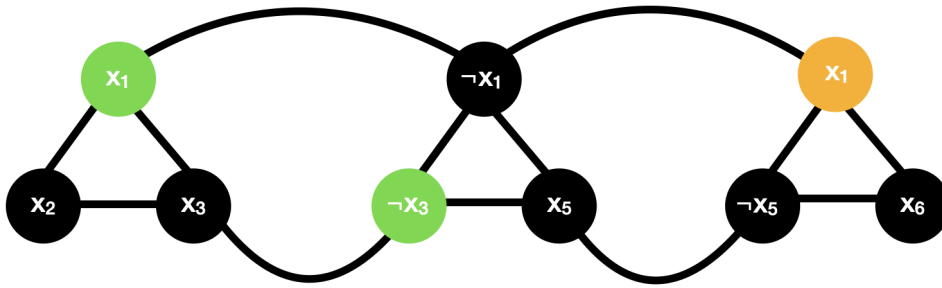


Abbildung 2.4.: Eine stabile Menge des Graphen bilden die grün eingefärbten Knoten. Durch die Interpretation wird auch der orange eingefärbte Knoten bei der Umwandlung in eine 3SAT Belegung als *wahr* betrachtet.

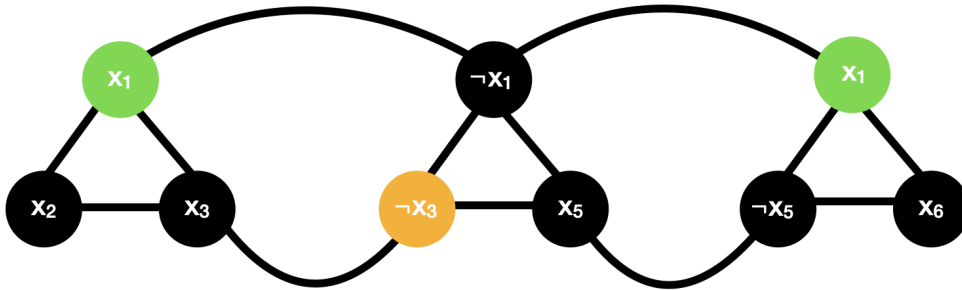


Abbildung 2.5.: Eine stabile Menge des Graphen bilden die grün eingefärbten Knoten. Durch das *Postprocessing* wird auch der orange eingefärbte Knoten bei der Umwandlung in eine 3SAT Belegung als *wahr* betrachtet.

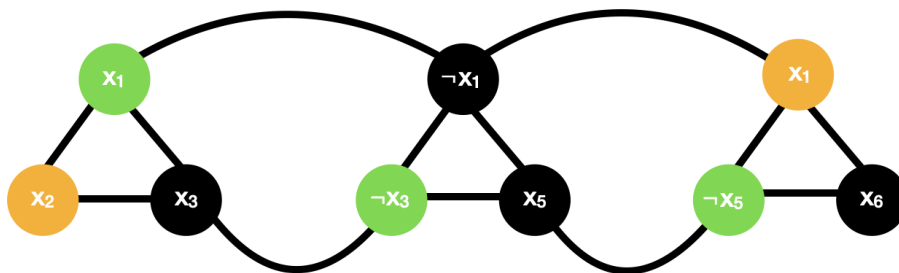


Abbildung 2.6.: Eine gefundene Lösung $L = \{x_1 \leftarrow 1; x_2 \leftarrow 1; x_3 \leftarrow 0; x_5 \leftarrow 0; x_6 \leftarrow 0\}$ wird in eine MIS-Lösung umgewandelt. Da nun in der ersten und dritten Klausel zwei Literale den Wert 1 besitzen wird jeweils ein dazugehöriger Knoten aus der stabilen Menge entfernt. (hier orange eingefärbt)

2. MIS und SAT

korrekte Belegung dar. Das dazugehörige MILP wird in [2.3](#) dargestellt.

$$\begin{array}{rcll}
 \cdots & +x_a+ & \cdots & +x_b+ & \cdots & +x_c & \cdots & \geq 1 \\
 \vdots & & \ddots & & \vdots & & & \\
 \cdots & +x_d+ & \cdots & +x_e+ & \cdots & +x_f & \cdots & \geq 1 \\
 & & & & 0 \leq & x_1 & \leq 1 \\
 & & & & \vdots & \vdots & \vdots \\
 & & & & 0 \leq & x_n & \leq 1 \\
 & & & & & x_1 & \text{integer} \\
 & & & & & \vdots & \\
 & & & & & x_n & \text{integer}
 \end{array} \tag{2.3}$$

Im praktischen Teil der Arbeit werden jedoch Ganzzahligkeitsbedingungen einiger Variablen vernachlässigt. Diese werden dann nach Finden der optimalen Lösung des MILP gerundet. Eine Belegung, die die Nebenbedingungen dieses MILP erfüllt, ist also nicht unbedingt eine Lösung des 3-SAT Problems. Deswegen muss eine Zielfunktion gefunden werden, mit der möglichst wahrscheinlich nach dem Lösen und Runden der Variablen ohne Ganzzahligkeitsbedingungen eine korrekte Belegung gefunden werden kann. Eine geeignete Zielfunktion zu finden ist also von zentraler Bedeutung in diesem Verfahren. Ein intuitiver Ansatz besteht darin, die Differenz der Anzahl der negierten und positiven Vorkommnisse einer Variable in einer Formel als Preis dieser zu verwenden. Damit lässt sich folgendes lineares Programm aufstellen:

$$\begin{array}{rcll}
 \min & f = & c_1x_1+ & \cdots & +c_nx_n \\
 \\
 \text{texts.t.} & \cdots & +x_a+ & \cdots & +x_b+ & \cdots & +x_c & \cdots & \geq 1 \\
 & \vdots & & \ddots & & \vdots & & \vdots & \\
 & \cdots & +x_d+ & \cdots & +x_e+ & \cdots & +x_f & \cdots & \geq 1 \\
 & & & & 1 \leq & x_1 & \leq 1 \\
 & & & & \vdots & \vdots & \vdots \\
 & & & & 1 \leq & x_n & \leq 1 \\
 & & & & & x_1 & \text{integer} \\
 & & & & & \vdots & \\
 & & & & & x_j & \text{integer}
 \end{array} \tag{2.4}$$

Kommt in einer Formel o_i positive und g_i negierte Literale derselben Variable x_i vor, so berechnet sich c_i mit $c_i = g_i - o_i$. Die Intuition hinter diesem Ansatz ist die Annahme, dass eine Variable in einer Lösung wahrscheinlicher den Wert 1 bzw. 0 annimmt, wenn sich mehr positive bzw. negierte Literale in einer Formel befinden. Ist die Differenz der

Anzahlen positiv, so existieren mehr negierte als positive Variablen. Der Preis ist positiv und der Zielwert somit besser, wenn die Variable den Wert 0 annimmt.

2.5. Hamming-Distanz

Die Hamming-Distanz ist eine Metrik für Bitstrings. Sie wurde von R. W. Hamming 1950 in [Ham50] präsentiert. Der Abstand zwischen zwei binären Zeichenketten ist dabei die Anzahl an unterschiedlichen Stellen der Kette. Abbildung 2.7 stellt das Konzept anhand einer Abbildung dar.

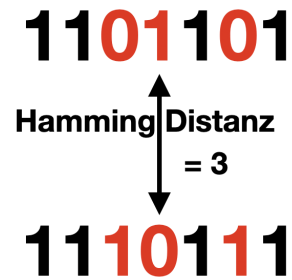


Abbildung 2.7.: Da drei Stellen in den beiden binären Zeichenketten unterschiedlich sind, beträgt die Hamming-Distanz 3.

3. Lineare Programmierung

Nur wenn 10^{50} Erden oder 10^{44} Sonnen ganz gefüllt sind mit Computern, wenn diese parallel und vom Urknall an bis die Sonne kalt wird rechnen, dann ist die Antwort vielleicht ja. - George B. Dantzig

Das schrieb George Dantzig im April 1981 in seinem Artikel [Dan83] zur Einführung der linearen Programmierung. Der Textausschnitt handelt vom benötigten Aufwand, die beste Zuordnung von 70 Jobs und 70 Arbeitern zu finden. Das direkte Vergleichen jeder Möglichkeit würde $70!$ Vergleiche nach sich ziehen und damit sehr viel Rechenzeit benötigen. Dennoch wurde Ende der 1940er Jahre eine Methode entwickelt, mit der sich dieses Problem in einer Sekunde auf einem damals modernen Computer lösen ließ. Das von George Dantzig entworfene und von vielen Wissenschaftlern weiterentwickelte Simplex-Verfahren wird in diesem Kapitel beschrieben. Es ist eine effiziente Lösungsmethode von linearen Programmen. Letzere beschreiben mathematische Modelle. Dabei wird das Maximum (oder Minimum) einer linearen Zielfunktion in einer Menge gesucht, die durch Nebenbedingungen eingeschränkt ist. Wenn nicht anders gekennzeichnet orientieren sich folgende Kapitel an [PAN14]. Auch die Abbildungen sind inspiriert von [PAN14].

3.1. Mathematische Beschreibung

Ein lineares Programm mit n Variablen und m Nebenbedingungen kann beschrieben werden durch eine Koeffizientenmatrix $A \in \mathbb{R}^{m \times n}$, einen Vektor der Variablen $x = x_1, \dots, x_n$ und einen Kostenvektor $c \in \mathbb{R}^n$. a_{ij} mit $i \in n, j \in m$ drückt dabei den Koeffizienten der Reihe i und der Spalte j der Matrix A aus. Die Nebenbedingungen können dabei sowohl Ungleichungen als auch Gleichungen sein. Um mit dem Problem einfacher arbeiten zu können, wird festgelegt, die Nebenbedingungen als Gleichungen zu formulieren. Es wird außerdem die Nichtnegativität der Variablen verlangt. Damit kann das Problem dargestellt werden als:

$$\begin{array}{llllll}
 \min & f = & c_1 x_1 + & \cdots & + c_n x_n & \\
 & & & & & \\
 \text{s.t.} & a_{11} x_1 + & \cdots & + a_{1n} x_n & = & b_1 \\
 & a_{21} x_1 + & \cdots & + a_{2n} x_n & = & b_2 \\
 & \vdots & \ddots & \vdots & & \vdots \\
 & a_{m1} x_1 + & \cdots & + a_{mn} x_n & = & b_m \\
 & & & & x_1 & \geq 0 \\
 & & & & \vdots & \vdots \\
 & & & & x_n & \geq 0
 \end{array} \tag{3.1}$$

3. Lineare Programmierung

Und kompakt mit den eingeführten Matrizen:

$$\begin{aligned}
 \min \quad & f = c^T x \\
 \text{s.t.} \quad & Ax = b \\
 & x_i \geq 0 \quad i \in 1, \dots, n \\
 & m < n
 \end{aligned} \tag{3.2}$$

Es gibt viele äquivalente Darstellungen desselben Problems. Alle können in die Standardform umgeformt werden. Eine zu maximierende Zielfunktion beispielsweise f' wird umgewandelt in eine zu minimierende Zielfunktion $f = -f'$. Bei Ungleichungen werden Slackvariablen eingeführt um sie in Gleichungen umzuwandeln.

- Beim Typ ' $\leq$ ' werden diese zu der Nebenbedingung addiert.
- Beim Typ ' $\geq$ ' werden diese von der Nebenbedingung subtrahiert.

Damit wird eine Ungleichung $\alpha_1 x_1 + \dots + \alpha_n x_n \leq \beta$ zu der Gleichung $\alpha_1 x_1 + \dots + \alpha_n x_n + x_{n+1} = \beta$ und $\alpha_1 x_1 + \dots + \alpha_n x_n \geq \beta$ zu $\alpha_1 x_1 + \dots + \alpha_n x_n - x_{n+1} = \beta$.

3.1.1. Geometrische Interpretation

Ein lineares Programm kann geometrisch interpretiert werden. Die Menge $\{(x, y) \in (\mathbb{R}, \mathbb{R}) | \alpha_1 x + \alpha_2 y \leq \beta\}$ kann im euklidischen Raum als Menge aller sich unter und auf der Ebene $\alpha_1 x + \alpha_2 y$ befindenden Punkte interpretiert werden. In mehreren Dimensionen wird die Menge $\{x \in \mathbb{R}^n | \alpha_1 x_1 + \dots + \alpha_n x_n = \beta\}$ eine Hyperebene genannt. Diese teilt den n -dimensionalen Raum in zwei Hälften. Deswegen wird die Menge $x \in \mathbb{R}^n$, die $x \in \mathbb{R}^n$ $\alpha_1 x_1 + \dots + \alpha_n x_n \leq \beta$ auch Halbraum genannt. Die zulässige Menge P des linearen Programms kann also direkt als die Schnittmenge vieler Halbräume interpretiert werden. Da jeder Halbraum konvex ist, ist auch P konvex. Damit bildet P ein n -dimensionales konvexes Polyeder. Die geometrische Interpretation der Suche nach einer optimalen Lösung des linearen Programms ist die maximale Verschiebung der Hyperebene $H = \{x \in \mathbb{R}^n | c^T x\}$ entlang ihres Normalenvektors c . Die Berührungspunkte der Hyperebene der Zielfunktion und des Polyeders P sind die Lösungen des linearen Programms. Lineare Programme kann man für $n < 3$ daher auch graphisch lösen. Abbildungen 3.1 und 3.2 präsentieren das Konzept. Folgendes lineare Programm aus [PAN14] wird als Beispiel verwendet:

$$\begin{aligned}
 \max \quad & f = 0x_1 + 1x_2 \\
 \text{s.t.} \quad & 1x_1 + 4x_2 + s_1 = 13 \\
 & 1x_1 + 1x_2 + s_2 = 5 \\
 & 2x_1 + 3x_2 + s_3 = 12 \\
 & x_1, x_2 \geq 0 \\
 & s_1, s_2, s_3 \geq 0
 \end{aligned} \tag{3.3}$$

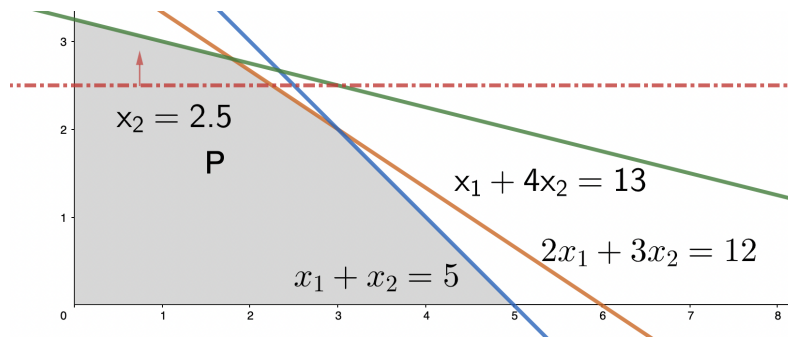


Abbildung 3.1.: Eine graphische Darstellung des linearen Programms [3.3](#). Die Hyperlebene $H = \{x \in \mathbb{R}^n | c^T x\}$ (hier rot gekennzeichnet) wird entlang ihres Normalenvektors c maximal verschoben, bis sie das Polyeder des linearen Programms nur mehr noch berührt.

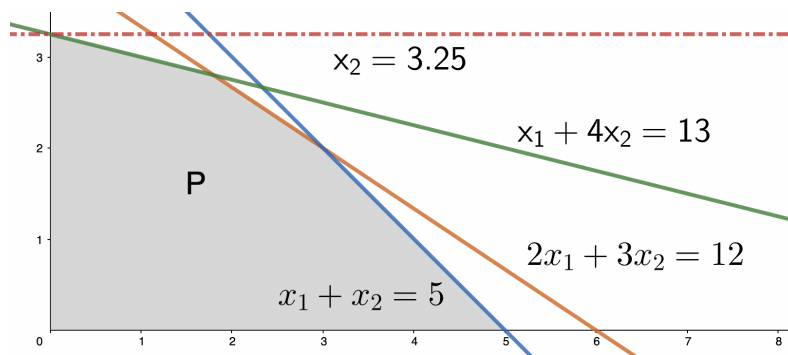


Abbildung 3.2.: Die Lösung des LP [3.3](#) sind dann die Berührungspunkte der verschobenen Zielfunktion und des Polyeders, also $x_1 = 0$ und $x_2 = 3.25$.

3.1.2. Tabellenform

Die Formulierung des linearen Programms [3.2](#) und das Beispiel [3.3](#) wird in eine Tabellenform (Tabelle [3.1](#), [3.3](#)) umgeformt.

Um nun eine Lösung zu finden, wird das Gauß'sche Eliminationsverfahren verwendet. (siehe Tabelle [3.3](#)) Durch elementare Zeilenumformungen wird die Tabelle so lange umgeformt bis eine $m \times m$ Einheitsmatrix enthalten ist. o_{xy} stellen die Koeffizienten dar, die aus den Zeilenumformungen erhalten werden. Es wird angenommen, dass $\text{rank} A = m$. Die Variablen, die in der Einheitsmatrix enthalten sind, nennt man Basisvariablen. Alle anderen heißen Nichtbasisvariablen. Die dazugehörigen Indexmengen werden B bzw N genannt. Es gilt $i \in B$ und $j \in N$. Alle Nichtbasisvariablen können nun als Parameter frei gewählt werden. Die Basisvariablen können dann mit

$$x_{i_d} = b_d - o_{dj_1}x_{j_1} - \dots - o_{dj_{n-m}}x_{j_{n-m}} \text{ mit } d \in \{0, \dots, m\}$$

berechnet werden. Setzt man alle Nichtbasisvariablen nun auf 0 erhält man die sogenannte Basislösung $x_{cd} = b_d$.

3. Lineare Programmierung

x_1	$\cdots$	x_n	RHS
a_{11}	$\cdots$	a_{1n}	b_1
$\vdots$	$\ddots$	$\vdots$	$\vdots$
a_{m1}	$\cdots$	a_{mn}	b_m

Tabelle 3.1.: RHS bedeutet *right-hand side* (aus dem Englischen für rechte Seite). Da $m < n$ gilt, ist dieses Gleichungssystem unterbestimmt. Das heißt es gibt entweder unendlich viele Lösungen oder keine Lösung.

x_1	x_2	s_1	s_2	s_3	RHS
1	4	1			13
1	1		1		5
2	3			1	12

Tabelle 3.2.: Das Beispiel 3.3 kann in einer Tabelle zusammengefasst werden.

3.2. Simplex

Es gibt zahlreiche Algorithmen zur Lösung von linearen Programmen. Zwei herausragende sind das Innere-Punkte-Verfahren und der Simplex-Algorithmus. Erstere Methode wurde 1984 von Karmakar ([Kar84]) entwickelt und hat eine Komplexität von $\mathcal{O}(x^{3.5} \log(1/\epsilon))$. ϵ drückt dabei die Genauigkeit der Lösung aus. Der von George B. Dantzig entwickelte Simplex-Algorithmus ([DOW55]) weist eine exponentielle Komplexität auf ([KM72]). Jedoch konnten Spielman und Teng mithilfe ihrer entwickelte *Smoothed Analysis* zeigen, dass der Simplexalgorithmus meistens nur eine polynomielle Laufzeit benötigt. ([ST01]) Durch die zahlreichen Verbesserungen in den letzten Jahrzehnten ist er meist zu dem effizientesten Lösungsverfahren geworden. Um eine Intuition für den Algorithmus zu entwickeln, wird wieder die geometrische Perspektive verwendet. Der Simplexalgorithmus bewegt sich dabei von einer Ecke des Polyeders zur nächsten. Die Ecken repräsentieren die Basislösungen des Systems. Dabei geht er in eine Richtung, die den Zielwert optimiert. Irgendwann wird damit der Optimalwert erreicht.

3.2.1. Simplextableau

Der Startpunkt des Algorithmus ist das Aufstellen des Simplextableaus. Dargestellt wird dieses in Tabelle 3.4

x_{i_1}	x_{i_2}	$\dots$	x_{i_m}	x_{j_1}	$\dots$	$x_{j_{n-m}}$	RHS
1	0	$\dots$	0	o_{1j_1}	$\dots$	$o_{1j_{n-m}}$	b_1
0	1	$\dots$	0	o_{2j_1}	$\dots$	$o_{2j_{n-m}}$	b_2
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
0	0	$\dots$	1	o_{mj_1}	$\dots$	$o_{mj_{n-m}}$	b_m

Tabelle 3.3.: Tabelle 3.1 wird hier nach Beenden des Gauß'schen Eliminationsverfahren dargestellt.

x_{b_1}	x_{b_2}	$\dots$	x_{b_m}	x_{c_1}	$\dots$	$x_{c_{n-m}}$	f	RHS
1	0	$\dots$	0	o_{1c_1}	$\dots$	$o_{1c_{n-m}}$		b_1
0	1	$\dots$	0	o_{2c_1}	$\dots$	$o_{2c_{n-m}}$		b_2
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$		$\vdots$
0	0	$\dots$	1	o_{mc_1}	$\dots$	$o_{mc_{n-m}}$		b_m
				z_1	$\dots$	z_n	-1	0

Tabelle 3.4.: Die Tabelle 3.3 wird um eine Zeile für den Kostenvektor und einer Spalte mit dem Zielwert erweitert. Hierbei kann die Spalte f entfernt werden, da sie sich während des Simplexalgorithmus nicht ändert.

3.2.2. Simplexverfahren

Der Algorithmus prüft Basislösungen, die die jeweils vorherigen verbessern. Das wird mit einem Pivotverfahren erreicht. Durch die Pivotauswahl wird ein negativer reduzierter Preisen ausgewählt. Die dazugehörige Spalte heißt Pivotspalte j . Das Minimum von $\frac{b_i}{a_{ij}}$ für alle $a_{ij} \geq 0$ in der Spalte j wird ermittelt um das Pivotelement a_{ij} zu finden. Mit elementaren Zeilenumformungen tritt nun die Variable x_i in die Basis ein. Variable x_{b_j} verlässt sie. Ein Beispiel des Verfahrens findet sich im Anhang C.

3.2.3. Überarbeitung des Simplexverfahrens

Es gibt zahlreiche Faktoren, die gegen die naive Implementierung des Simplex sprechen. Zwei davon sind die numerische Ungenauigkeit aufgrund Rundungsfehler und die Ineffizienz. Beide Problem werden mit einem überarbeiteten Simplexalgorithmus gelöst. Dabei wird die Tableauform durch eine Darstellung mit Matrizen ersetzt. (siehe Tabelle 3.5) Das ermöglicht auf der einen Seite mehr numerische Stabilität durch Neuberechnungen nach einigen Iterationen. Auf der anderen Seite können Fortschritte in der numerischen Analyse direkt zur Verbesserung der Effizienz genutzt werden. In Abbildung 3.6 wird Tabelle 3.5 mit reduzierten Preisen z geschrieben und ersetzt damit die Simplextabelle 3.4. Es gilt

$$B = \{j_1, \dots, j_m\} \text{ und } N = A \setminus B = \{j_{m+1}, \dots, j_n\}$$

3. Lineare Programmierung

B und N werden gleichzeitig als Basismatrix und Nichtbasismatrix verwendet und A als Koeffizientenmatrix:

$$B = \{A_{j_1}, \dots, A_{j_m}\} \text{ und } N = A \setminus B = \{A_{j_{m+1}}, \dots, A_{j_n}\}$$

Um die Werte der Basisvariablen auszurechnen, wird $Bx_B + Nx_N = b$ mit B^{-1} multipliziert:

$$x_B + B^{-1}Nx_N = B^{-1}b$$

Eine Basislösung ist mit $x_N = 0$ dann

$$x_B = B^{-1}b$$

Mit $x_B = B^{-1}b - B^{-1}Nx_N$ aus Tabelle 3.6 wird $c_B^T x_B + c_N^T x_N - f = 0$ zu

$$(c_N^T - c_B^T B^{-1}N)x_N - f = -c_B^T B^{-1}b$$

Damit kann eine neue überarbeitete Simplextabelle 3.7 eingeführt werden. In Tabelle finden sich dann Transformationen zwischen der normalen und überarbeiteten Simplextabelle.

x^T	RHS
A	b
c^T	

Tabelle 3.5.: A beschreibt hierbei die anfängliche Koeffizientenmatrix, c den Kostenvektor und b die rechte Seite der Gleichungen.

x_B^T	x_N^T	RHS
B	N	b
	z^T	

Tabelle 3.6.: B stellt in dieser Tabelle die Basis-Indexmenge und N die Nichtbasis-Indexmenge dar. A beinhaltet alle Spaltenindizes der Koeffizientenmatrix. z beschreibt die reduzierten Kosten.

x_B^T	x_N^T	RHS
I	$B^{-1}N$	$B^{-1}b$
	$c_N^T - c_B^T B^{-1}N$	$-c_B^T B^{-1}b$

Tabelle 3.7.: Die Abbildung stellt die neue überarbeitete Simplextabelle dar.

Erklärung	Simplex Tabelle	Überarbeitete Simplex Tabelle
Kostengleichung	z_N^T	$-c_B^T B^{-1}b$
Pivotspalte	a_q	$B^{-1}a_q$
RHS	b	$B^{-1}b$

Tabelle 3.8.: Für die Ausarbeitung des Algorithmus ist es sinnvoll noch einige weitere Relationen der Variablen der beiden Tabellen darzustellen.

3.2.4. Überarbeiteter Simplexalgorithmus

Zur Berechnung eines Pivotschrittes werden sehr wenig Elemente der Simplextabelle 3.4 wirklich gebraucht. Zur Auswahl der Pivotspalte genügt es die Zeile der Kosten zu betrachten. Zur Auswahl des Pivotelements beinhaltet die Pivotspalte und die rechte Seite genügend Informationen. Im Gegensatz zum naiven Simplex Verfahren wird berechnet nun also nur einen Teil der $(m+1)(n+1)$ Elemente des Problems. Um die drei Elemente der Tabelle 3.8 zu berechnen, ist eine Aktualisierung von B^{-1} notwendig. Wurde eine Pivotspalte q und eine Pivotreihe p gefunden, so wird die Spalte a_{j_p} von B durch die Pivotspalte a_q ersetzt.

$$B = (a_{j_1}, \dots, a_{j_{p-1}}, a_q, a_{j_{p+1}}, \dots, a_{j_m})$$

Es wird nun nach einer Matrix M gesucht, sodass gilt $MB^{-1}B = I$.

$$\begin{aligned}
MB^{-1}B &= M(B^{-1}a_{j_1}, \dots, B^{-1}a_{j_{p-1}}, B^{-1}a_q, B^{-1}a_{j_{p+1}}, \dots, B^{-1}a_{j_m}) \\
&= M(e_{j_1}, \dots, e_{j_{p-1}}, a_q, e_{j_{p+1}}, \dots, e_{j_m}) \\
&= (Me_{j_1}, \dots, Me_{j_{p-1}}, Ma_q, Me_{j_{p+1}}, \dots, Me_{j_m}) \\
&\stackrel{!}{=} I \\
&\implies Ma_q \stackrel{!}{=} e_p
\end{aligned}$$

Es folgt der Aufbau der Matrix M . Der Wert $-1/a_{p,q}$ befindet sich dabei in der Reihe und Spalte p .

$$\begin{pmatrix}
1 & & -a_{1,q}/a_{p,q} & & \\
& \ddots & \vdots & & \\
& & -a_{p-1,q}/a_{p,q} & & \\
& & -1/a_{p,q} & & \\
& & -a_{p+1,q}/a_{p,q} & & \\
& & \vdots & \ddots & \\
& & -a_{m,q}/a_{p,q} & & 1
\end{pmatrix}$$

Mit M kann B^{-1} nun aktualisiert werden. Ein überarbeiteter Simplexalgorithmus 3.1 wird dargestellt, der die obigen Ergebnisse nutzt. Er ist die Grundlage von modernen LP-Solvern und auch des kommerziellen Solvers Gurobi, der in dieser Arbeit verwendet wird. Es gibt zahlreiche weitere mögliche Verbesserungen der Effizienz des Algorithmus.

3. Lineare Programmierung

Das beinhaltet Fortschritte in der numerischen Analyse, aber auch beispielsweise bessere Pivotverfahren, die zu weniger Iterationen führen. Viele theoretische und praktische Weiterentwicklungen des Simplexalgorithmus präsentiert [PAN14] sehr ausführlich.

Algorithmus 3.1 (Simplexalgorithmus)

Startpunkt: (B, N) , B^{-1} , $x_B = B^{-1}b \geq 0$, $f = c_B^T x_B$

1. Berechne $z_N = c_N - N^T B^{-T} c_B$
2. Auswahl der Pivotspalte q mit $q \in \operatorname{argmin}_{j \in N} (z_j)$
3. Berechne neue Pivotspalte a_q mit $B^{-1} a_q$
4. Wenn $z_q \geq 0 \implies$ Lösung ist optimal
5. Wenn $a_{iq} \leq 0 \forall i \implies$ Problem ist ungebunden
6. Bestimme Schrittlänge α und Pivotreihe p mit $\alpha = \frac{x_{jp}}{a_{pq}} = \min(\frac{x_{ji}}{a_{iq}} | a_{iq} \geq 0; i \in \{1, \dots, m\})$
7. Setze $x_q = \alpha$
8. Wenn $\alpha \geq 0$, aktualisiere x_B mit $x_B - \alpha a_q$ und f mit $f + \alpha z_q$
9. Aktualisiere B^{-1} mit (...)
10. Austausch von j_p und q in (B, N)
11. Gehe zu Schritt 1

3.3. Ganzzahlige Lineare Programmierung

Diese Sektion behandelt das Lösen von linearen Programmen, dessen Variablen teilweise an Ganzzahlbedingungen gebunden sind (auch MILP für *Mixed Integer Linear Programming*). Um die mögliche Lösungsverfahren von MILPs zu erklären wird ein Beispiel (angelehnt an ein Beispiel aus [PAN14]) verwendet.

Umsatzmaximierung der Tischproduktion: Eine Firma produziert Tische. Der Umsatz soll maximiert werden. Maschine 1 kann maximal 12 Minuten arbeiten, Maschine 2 maximal 5 Minuten und Maschine 3 maximal 3 Minuten. Maschine 1 benötigt 2 Minuten zur Produktion eines Holztisches und 3 Minuten für einen Glastisch. Maschine 2 benötigt zur Herstellung beider Tische jeweils 1 Minute. Maschine 3 benötigt für einen Glastisch 1 Minute. Maschine 3 kann ausschließlich Glastische produzieren. Ein Holztisch kann die Firma für 20 Euro verkaufen. Für einen Glastisch erhält sie 50 Euro.

Ein lineares Programm der Situation lässt sich aufstellen.

$$\begin{aligned} \max f = & \quad 20x_1 \quad + 50x_2 \\ \text{s.t.} \quad & \quad 2x_1 \quad + 3x_2 \leq 12 \\ & \quad 1x_1 \quad + 1x_2 \leq 5 \\ & \quad \quad \quad x_2 \leq 3 \\ \text{integer} \quad & \quad x_1, x_2 \geq 0 \end{aligned} \tag{3.4}$$

x_1 und x_2 drücken dabei die Anzahl an Holz- bzw. Glastischen aus, die verkauft werden. Eine grafische Darstellung der Situation präsentiert Abbildung 3.3.

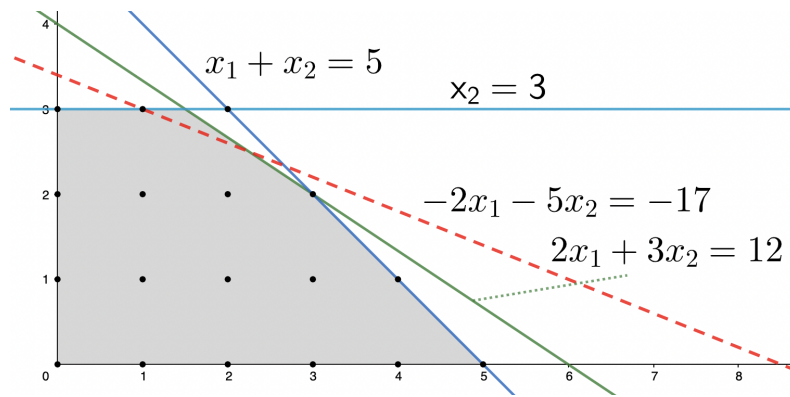


Abbildung 3.3.: Graphische Repräsentation des linearen Programmes 3.4.

3.3.1. Branch and Bound

Der Lösungsansatz des *Branch and Bound* (englisch für *Verzweigen und Einschränken*) versucht durch Verzweigung und Einschränkung des Problems die Lösung eines MILP zu finden. Erstmals zur Lösungsfindung eines MILP formuliert wurde es 1960 von Land und Doig ([LD60]). Das Verfahren beginnt mit der Lösung der LP-Relaxierung des Programms. In einem Verzweigungs-Schritt werden dann zwei neue Knoten erstellt, deren Lösungsmenge beschränkt ist. Die LP-Relaxierung eines neu ausgewählten Knoten wird wieder gelöst, und die Menge der Knoten eingeschränkt. Im Folgenden werden die wichtigsten Schritte genauer erläutert. Grafik 3.4 veranschaulicht das Prinzip und Algorithmus 3.2 beschreibt das Verfahren dann kompakt.

Knotenauswahl

In der Knotenauswahl gibt es zwei maßgebliche Verfahren: tiefenorientiert und breitenorientiert. Die Tiefenorientierung wird zum schnellen Finden einer Lösung genutzt. Wenn der aktuelle Knoten nicht verworfen wird, wählt der Algorithmus im nächsten Schritt einen der von ihm verzweigten Knoten. Daher geht der Algorithmus in jedem Ast in die Tiefe, bis er entweder eine Lösung findet oder der Ast verworfen wird. Die Breitenorientierung wird verwendet um sehr gute zulässige Lösungen zu finden. Dabei wählt der Algorithmus immer den Knoten aus, der den besten Zielwert hat. Damit sucht der Algorithmus bildlich gesprochen als erstes in der Breite. Im praktischen Teil verwenden wir die Standardeinstellung Gurobis, die Vorteile beider Varianten verknüpft.

Einschränkung / Bound

Nachdem die LP-Relaxierung des ausgewählten Knoten gelöst wurde, wird der Knoten aus der Knotenmenge entfernt. Sollte die Lösung der LP-Relaxierung gleichzeitig eine Lösung des MILP sein, so wird die Lösung die aktuelle des MILP, wenn der Zielwert besser ist als der bisherige. Ist das nicht der Fall, wird der Knoten aus der Knotenmenge entfernt. Gleiches passiert bei unzulässigen LP-Relaxierungen.

Verzweigung / Branch

Bei der Verzweigung eines Knotens wird die Lösung l dessen LP-Relaxierung zu-

3. Lineare Programmierung

grunde gelegt. Es wird ein Variable x_i ausgewählt, deren Ganzzahlbedingung unerfüllt ist. Zwei neue Knoten werden erstellt. Der eine erhält die zusätzliche Nebenbedingung $x_i \geq \lceil l_i \rceil$, der andere $x_i \leq \lfloor l_i \rfloor$ um die nicht-ganzzahlige Lösung der Variable in der nächsten LP-Relaxierung auszuschließen.

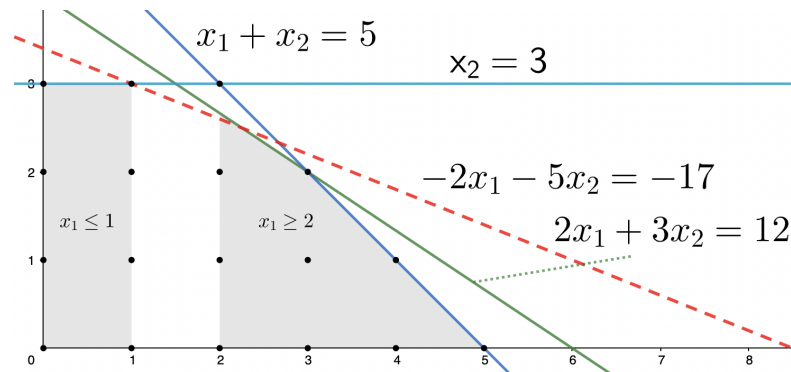


Abbildung 3.4.: Der Algorithmus teilt die Lösungsmenge der LP-Relaxierung bei jeder Verzweigung in drei Teile. Dies geschieht durch zwei Hyperebene die senkrecht zur Achse der ausgewählten Variable verlaufen. Die äußeren zwei Räume bilden die jeweiligen Lösungsräume der zwei neuen Knoten. Hier wird das Beispiel 3.4 dargestellt.

Algorithmus 3.2 (Branch and Bound)

Der Algorithmus berechnet die Lösung eines MILP.

1. (Bound) Relaxiere das Programm und löse das LP. Der Lösungsvektor wird mit l bezeichnet. Der Zielwert $f_{\min}$ gibt eine untere Schranke für optimalen Zielwert f_{opt} des MILP an. Setze $f_{\text{opt}} = \infty$. Das MILP kann keinen besseren Zielwert annehmen als seine LP-Relaxierung.
2. Wenn alle Ganzzahligkeitsbedingungen erfüllt sind, stoppe. Es liegt eine optimale Lösung l mit Zielwert $f_{\text{opt}} \leftarrow f_{\min}$ für das MILP vor.
3. (Branch) Wähle eine Variable x_i aus, deren Ganzzahligkeitsbedingung unerfüllt ist. Erstelle zwei Knoten, deren Nebenbedingungen um $x_i \geq \lceil l_i \rceil$ bzw. $x_i \leq \lfloor l_i \rfloor$ erweitert werden.
4. Wähle einen Knoten aus dem Suchbaum.
5. Optimierte die LP-Relaxierung des Knotens.
6. (Bound)
 - Wenn der Zielwert $f > f_{\text{opt}}$, lösche den Knoten.
 - Wenn der Zielwert $f < f_{\text{opt}}$, aktualisiere $f_{\text{opt}} \leftarrow f$ und lösche den Knoten.
 - Wenn LP Relaxation unlösbar, lösche den Knoten.
7. (Branch) Wenn der Knoten nicht verworfen wurde, verzweige. Wähle eine Variable x_i aus, deren Ganzzahligkeitsbedingung unerfüllt ist. Erstelle zwei Knoten deren MILP um jeweils eine Nebenbedingung erweitert ist. Einmal $x_i \geq \lceil l_i \rceil$ und das andere Mal $x_i \leq \lfloor l_i \rfloor$.
8. Wenn der Suchbaum nicht leer ist, gehe zu Schritt 4.

3.3.2. Schnittebenenverfahren

Wie beim *Branch and Bound* Algorithmus ist auch das Ziel beim Schnittebenenverfahren nicht-ganzzahlige Lösungen ohne ganzzahlige aus der Lösungsmenge eines MILP herauszuschneiden. Dabei wird versucht, die Ecken des Polyeders der LP-Relaxierung mit Hyperebenen abzuschneiden. Das Ziel dieses Verfahrens ist das Annähern an den Polyeder des MILP Problems, um die optimale Lösung dessen zu finden. Eine grafische Darstellung findet sich in der Abbildung. Dabei gibt es für jedes MILP gültige Schnittebenenverfahren und problemspezifische, die die spezielle Struktur des Problems ausnutzen. Beispiele für erstere sind: *Gomory mixed integer cuts* (auch *GMI*) ([Gom60]), *Mixed integer rounding cuts* (auch *MIR*) ([MW01]) und *Disjunctive cuts* ([Bal79]). Der Zusammenhang zwischen den letzten zwei im Kontext der 0-1-Programmierung (Variablen können ausschließlich 0 oder 1 annehmen)- die in dieser Arbeit relevant sein wird- kann in [BP03] nachgelesen werden.

Das *Problem des Handelsreisenden* (auch *TSP*), bei dem der kürzeste Weg zwischen Städten gesucht wird, konnte mithilfe von problemspezifischen Schnittebenen in einem *Branch and Cut* Verfahren (siehe Kapitel 3.3.3) Ende der 90er Jahre sehr viel schneller gelöst werden. ([PR89]) Nachdem 1987 zuvor die maximale Größe des TSPs 666 Städte betrug, konnten Padberg et al. ein Problem mit 2392 Städten lösen. 1994 konnten Applegate et al. die Lösung eines TSP mit 7397 finden und nur 4 Jahre später wurde bereits die kürzeste Strecke zwischen 13509 Städten in den USA gefunden. ([mis]) Für eine genauere und umfangreichere Beschreibung von Schnittebenen wird [MMWW02] und [PAN14] empfohlen. Das Schnittebenenverfahren wird in Abbildung 3.5 beispielhaft dargestellt. Die Methode ist zwar zur Beschränkung der Lösungsmenge (siehe Sektion 3.3.3) sehr sinnvoll, ist aber sehr ineffizient in der Suche nach einer Lösung eines MILP. Auch ist die Terminiertheit nicht bei jedem Verfahren gegeben.

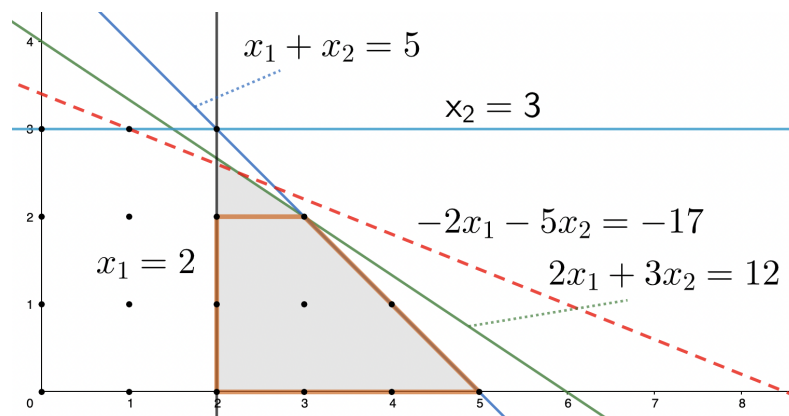


Abbildung 3.5.: Ein optimales Schnittebenenverfahren würde nach dem ersten Durchgang des *Branch and Bound* Algorithmus (siehe Abbildung 3.4) die zulässige Lösungsmenge der LP-Relaxierung des Knotens mit $x_1 \geq 2$ (grau eingefärbt) so beschränken, dass die Lösungsmenge der LP-Relaxierung der konvexen Hülle der Lösungsmenge des MILPs (in orange) des Knotens entspricht.

3.3.3. Branch and Cut

Um den *Branch and Bound* Algorithmus effizienter zu machen, wird in modernen Solvern bei jedem Lösen der LP-Relaxierungen ein Schnittebenenverfahren benutzt. Das verschärft die LP-Relaxierung. Das Verfahren wird *Branch and Cut* genannt. Die gewonnene Zeit um das MILP damit lösen zu können, muss dabei kleiner sein als die zur Erstellung der Schnittebene benötigte. Auch der MILP-Solver Gurobi arbeitet mit dieser Methode.

4. Quantum Annealing

Annealing im Namen *Quantum Annealing* ist an *Classical Annealing* angelehnt. Beim sogenannten Glühen werden Materialien mit sehr hoher Temperatur (also Energie) langsam abgekühlt um eine gewisse Eigenschaft (zum Beispiel erhöhte Widerstandsfähigkeit) zu erreichen. ([Whi99]) Beim *Quantum Annealing* besteht der Abkühlungsschritt des *Classical Annealing* in der Änderung des Energiespektrums des quantenmechanischen Systems. Das Kapitel orientiert sich an den Arbeiten [KN98], [FGGS00], [BAS⁺13] und [AL18]. Es startet mit einer Beschreibung des Konzeptes, dann mit den Bausteinen (Qubits) eines Quantum Annealers und deren Zusammenspiel. Ein Modell zur mathematischen Beschreibung wird in Sektion 4 eingeführt und eine neue Darstellung in Sektion 5. Fortgefahren wird mit einem Kapitel über Minor Embedding und einer Sektion über den Unterschied des klassischen und quantenmechanischen Algorithmus. Das Kapitel wird beendet mit dem für die Arbeit wichtigen Reverse Annealing (eine Art des Quantum Annealing mit Anfangszuständen).

4.1. Beschreibung

Ein wichtiges Theorem zur Beschreibung zeitabhängiger Systeme ist das adiabatische Theorem (Definition 4.1) der Quantenmechanik. Es wurde 1928 von Born und Fock bewiesen für Systeme, deren Hamiltonoperator nicht entartet ist. 1950 folgte dann die allgemeine Formulierung von Tosio Kato. ([Kat50])

Definition 4.1 (Adiabatisches Theorem der Quantenmechanik) Sei $H(t)$ ein zeitabhängiger Hamiltonoperator. Das System ist dabei im Grundzustand $E_{t_0}^0$ zur Zeit $t = t_0$. Ändert man nun den Hamiltonoperator genügend langsam, so befindet sich das System zur Zeit t_1 trotz des neuen Energieoperators im Grundzustand $E_{t_1}^0$.

Mithilfe des adiabatischen Theorems der Quantenmechanik ist es möglich, Optimierungsprobleme zu lösen. (adiabatisches Quantum Computing)

Definition 4.2 (Adiabatisches Quantum Computing / Quantum Annealing) Sei H_p der Energieoperator dessen Grundzustand der Lösung des zu optimierenden Problems entspricht und sei H_0 ein Hamiltonoperator eines einfachen physikalischen Systems. Die Zeit am Ende der Durchführung sei $t = T$. Es seien zusätzlich $A(s)$ und $B(s)$ kontinuierliche Funktionen, die Annealing Schedule genannt werden und es gelte $A(0) > 0, B(1) > 0, A(1) = 0, B(0) = 0$ und $s(t) = \frac{t}{T}$. Dann gilt wenn die Änderung von $\partial B(s)$ und $\partial A(s)$ genügend klein ist, dass das System mit dem Hamiltonoperator

$$H(s) = A(s)H_0 + B(s)H_p \quad (4.1)$$

bei $s = 1$ im Grundzustand ist. Da $H(1) = B(1)H_p$ gilt, kann der Grundzustand vom System direkt als Lösung des zu optimierenden Problems übersetzt werden.

4. Quantum Annealing

Zur Implementation des Konzeptes wird in Quantencomputern der Firma Dwave-Systems das Ising-Modell verwendet. (vgl. Kapitel 4.4 oder [dwa21]) H_p entspricht dem Hamiltonoperator des Isingproblems:

$$H_p = H_{Ising} = \sum_{i=1} h_i \sigma_i + \sum_{i < j} J_{i,j} \sigma_i \sigma_j \quad (4.2)$$

H_0 definiert man als Energieoperator eines physikalisch einfach zu initiiierenden Systems mithilfe eines transversalen magnetischen Feldes:

$$H_0 = H_{trans} = \sum_{i=1}^N \sigma_i \quad (4.3)$$

Bei sehr tiefen Temperaturen nimmt das System mit dem Hamiltonoperator H_{trans} den Grundzustand an, der einer Superposition aller 2^N möglicher Basiszustände entspricht. Misst man nun an der Zeit $t = T$ mit genügend großem T den Zustand des Systems - also die Richtung der Spins der Elemente - stellen die Spinwerte die Lösung des Problems dar. Für eine genauere Beschreibung der Implementierung einzelner Qubits (Kapitel 4.2) wird auf [JAG⁺11] verwiesen.

4.2. Quanten Bits

Dieses Kapitel und Kapitel 4.3 orientiert sich an [Rie11]. Ein Quantenbit $|b\rangle$ ist eine Superposition zweier Basiszustände β_1 und β_2 :

$$|b\rangle = \alpha_1 |\beta_1\rangle + \alpha_2 |\beta_2\rangle$$

Die Dynamik der Zustände werden durch die Schrödingergleichung beschrieben und befinden sich in einem Hilbertraum. Aufgrund Gleichung B.6 gilt

$$|\alpha_1|^2 + |\alpha_2|^2 = 1$$

Der Spin eines Elektron ist ein Beispiel für ein Quanten Bit. Er kann positiv oder negativ sein, hat also zwei Basiszustände. Diese werden beschrieben durch

$$|spin_+\rangle = |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

und

$$|spin_-\rangle = |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Ein Qubit ist damit ein quantenmechanisches Analog zum klassischen Bit und kann als Speichereinheit und somit Recheneinheit genutzt werden für Quantencomputer. Es existieren auch Zustände, die sich in einer Superposition aus n Basiszuständen befinden - sogenannte Qudits. Nach [Rie11] können diese jedoch immer mit mehreren Qubits modelliert werden. Daher werden meist nur Quantenbits verwendet.

4.3. Systeme mit mehreren Qubits

In Systemen der klassischen Physik können n Elemente eines 2-dimensionalen Vektorraums aufgrund der direkten Summe der Vektorräume in einem $2n$ -dimensionalen Vektorraum beschrieben werden. In quantenmechanischen System gilt dies jedoch nicht. Hier interagieren n Quantenbits mit einem Tensorprodukt über deren Hilberträume, daher muss das System in einem 2^n -dimensionalen Vektorraum V beschrieben werden. Wie in [Rie11] wird eine kompakte Schreibweise verwendet:

$$\begin{aligned}
 V = \begin{array}{ccccccc}
 \{ |0\rangle_{n-1} \otimes & \dots & \otimes |0\rangle_1 & \otimes |0\rangle_0, & & \{ |00\dots 00\rangle, \\
 |0\rangle_{n-1} \otimes & \dots & \otimes |0\rangle_1 & \otimes |1\rangle_0, & & |00\dots 01\rangle, \\
 |0\rangle_{n-1} \otimes & \dots & \otimes |1\rangle_1 & \otimes |0\rangle_0, & \doteq & |00\dots 10\rangle, \\
 \vdots & & \vdots & \vdots & & \vdots \\
 |1\rangle_{n-1} \otimes & \dots & \otimes |1\rangle_1 & \otimes |1\rangle_0 \} & & \{ |11\dots 11\rangle \}
 \end{array} \quad (4.4)
 \end{aligned}$$

Die Dimensionalität 2^n von V ist verantwortlich dafür, dass vor allem Richard Feynman und Yuri Manin ([Rie11]) in den 1980er Jahren begannen zu vermuten man könne mit einem Quantencomputer schnellere Operationen als mit klassischen Computern durchführen. Ein Beispiel dafür ist ein im Jahr 1985 erschienener Artikel Feynmans über Quantencomputer. ([Fey85])

4.4. Das Ising-Modell

Das Kapitel ist angelehnt an [Cip87]. Das Ising-Modell ist ein Konzept um den Magnetismus in Festkörpern besser beschreiben zu können. Ernst Ising beschrieb das Modell erstmals 1925 in [Isi25]. Ein 2-dimensionales Gitter wird als Vereinfachung eines Materials gewählt. Nun repräsentiert jeder Knoten dieses Gitters ein Element, dessen Wert ± 1 annehmen kann. Physikalisch beschreibt dieser Wert den Spin des Teilchens und damit das magnetische Moment ("magnetisch anziehend oder abstoßend"). Die Energie des Systems kann dann durch den Hamiltonoperator H_{Ising} wie folgt dargestellt werden:

$$H_{Ising} = \sum_{i=1} h_i \sigma_i + \sum_{i<j} J_{i,j} \sigma_i \sigma_j \quad (4.5)$$

σ_i und σ_j beschreiben die Spinoperatoren, dessen Eigenwerte s_i und s_j die Werte des Spins der Elemente i und j charakterisieren. h_i kennzeichnet die Stärke des externen Magnetfelds. $J_{i,j}$ stellt die Kopplungskonstante dar. Dieser Wert gibt an, inwieweit sich die Spins s_i, s_j zweier Elemente i, j beeinflussen. Beide Parameter h_i und $J_{i,j}$ können mit einer entsprechenden Vorrichtung von außen variiert werden. Dabei kann Gleichung 4.5 auch durch die Eigenwerte s_i der Operatoren σ_i als Hamiltonfunktion beschrieben werden:

$$H_{Ising}^f = \sum_{i=1} h_i s_i + \sum_{i<j} J_{i,j} s_i s_j \quad (4.6)$$

Die Spins der Elemente können aufgrund ihrer 2-Dimensionalität als Quanten Bits aufgefasst werden. Aufgrund thermischer Fluktuationen beschreibt das Modell die Energie eines Systems nur bei sehr tiefen Temperaturen. Daher operiert der Quantencomputer *Advantage* der Firma *D-Wave Systems* beispielsweise mit einer Temperatur von unter $15mK$.

4.5. QUBO Darstellung

Die folgenden drei Kapitel orientieren sich an [dwa21]. Um die Verwendung des Ising-Modells der klassischen Informatik anzupassen wird nach einer binären Formulierung von Gleichung 4.5 gesucht. Um das zu erreichen, werden die Spinvariablen zu neuen Variablen transformiert:

$$x_i = \frac{1 + s_i}{2} \text{ mit } \forall s_i \in -1, 1, x_i \in 0, 1 \quad (4.7)$$

Die Minimierung der Funktion $H_{QUBO}^f = x^T Q x$ mit $x \in 0, 1$ wird dann QUBO genannt. (QUBO steht für 'Quadratic unconstrained binary optimization')

Theorem 4.3 Die Hamiltonfunktion eines Ising-Modells $H_{Ising}^f = \sum_{i=1} h_i s_i + \sum_{i < j} J_{i,j} s_i s_j$ mit $s_i \in -1, 1$ kann ausgedrückt werden als QUBO-Funktion der Form $H_{QUBO}^f = x^T Q x$ mit $x_i \in 0, 1$:

$$\begin{aligned} H_{Ising}^f &= \sum_i h_i s_i + \sum_{i < j} J_{i,j} s_i s_j \\ &= \sum_i h_i (2x_i - 1) + \sum_{i < j} J_{i,j} (2x_i - 1)(2x_j - 1) \\ &= 2 \sum_i h_i x_i + \sum_{i < j} J_{i,j} (4x_i x_j - 2x_i - 2x_j + 1) - \sum_i h_i \\ &= \sum_{i \leq j} q_{ij} x_i x_j + C \\ H_{QUBO}^f &= \sum_{i \leq j} Q_{i,j} x_i x_j = x^T Q x \end{aligned}$$

Anmerkung 4.4 Nach [Cho10] entspricht die Formulierung 2.1 eines MIS Problems (vgl. 2.2) direkt seiner QUBO Funktion. Somit kann eine MIS Zuordnung direkt als Zuordnung der Variablen des QUBO des Problems aufgefasst werden.

4.6. Minor Embedding

Theoretisch kann ein Element des Ising-Modell mit allen vorkommenden Elementen gekoppelt sein. Praktisch jedoch weist die Komplexität der physischen Implementierung Grenzen auf. Der Quantencomputer *Advantage 4.1* der Firma *D-Wave Systems* - mit dem in dieser Arbeit gearbeitet wird - benutzt einen Pegasusgraphen als Struktur zur Kopplung. Jedes Qubit ist mit 15 weiteren gekoppelt. Da in unserem Problem jedoch

eventuell mehr als 16 Quanten Bits miteinander verbunden sein müssen (vgl. [2.3]), wird vor dem tatsächlichen Lösen des Problems mithilfe des Quantencomputers ein Prozess namens *Minor Embedding* durchgeführt. Es wird dabei versucht, die Qubits, deren Knoten mehr als 15 Kanten zu anderen in unserem MIS besitzen, auf mehrere Qubits zu verteilen. Dabei besitzen die Kanten hohe negative Gewichte. Da eine Energieminimierung stattfindet, ist es sehr wahrscheinlich, dass alle dieser Qubits denselben Wert annehmen. Eine Darstellung findet sich in Abbildung 4.1.

4.7. Simulated Annealing und der Tunneleffekt

Ein dem Quantum Annealing ähnlicher klassischer Algorithmus existiert unter dem Namen Simulated Annealing. Simulated Annealing ist ein heuristisches, an die statistische Mechanik der Physik angelehntes Optimierungsverfahren. (vgl. [KGV83]) Der Algorithmus kann folgendermaßen formuliert werden:

Algorithmus 4.5 (Simulated Annealing) *Beginne mit einer Temperatur T und einer Belegung des Problems.*

1. *Senke die Temperatur T um einen gewissen Wert. Wenn $T = 0$ beende den Algorithmus.*
2. *Verändere die Belegung des Problems um einen kleinen zufälligen Wert*
3. *Wenn für die Differenz der Energien gilt $\Delta E \leq 0$, gehe zu Schritt 1 und wiederhole den Vorgang mit der neuen Belegung*
4. *Wenn $\Delta E > 0$, übernehme die Belegung mit der Wahrscheinlichkeit $\exp(-\frac{\Delta E}{k_b T})$ und gehe zu Schritt 1*

Im Gegensatz zum quantenmechanischen Verfahren, kann jedoch jedes Bit nur einen Zustand annehmen. Das resultiert in einem weit längeren und unwahrscheinlicheren Weg aus lokalen zu globalen Minima für das Simulated Annealing, da dieses die Maxima des Problems passieren muss. Ein System aus Qubits kann jedoch durch Superposition mehrere Zustände gleichzeitig darstellen. (vgl. Kapitel 4.2) Da Quantum Annealing damit praktisch Barrieren ohne Aufwand überwinden kann, nennt man dieses Phänomen auch den Tunneleffekt. Grafisch wird der Unterschied zwischen Simulated Annealing und Quantum Annealing in Abbildung 4.2 dargestellt.

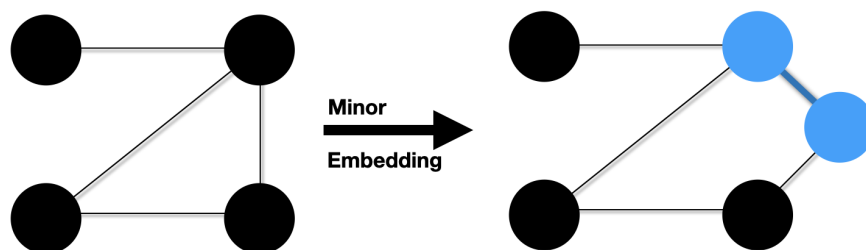


Abbildung 4.1.: Graphische Repräsentation eines *Minor Embedding*. Jeder kann in diesem Beispiel maximal 2 Verbindungen besitzen. Der Knoten in der oberen Ecke wird deshalb auf zwei Knoten verteilt. Die blaue Kante zwischen den neuen blauen Knoten besitzt ein stark negatives Gewicht. (inspiriert von [dwa21])

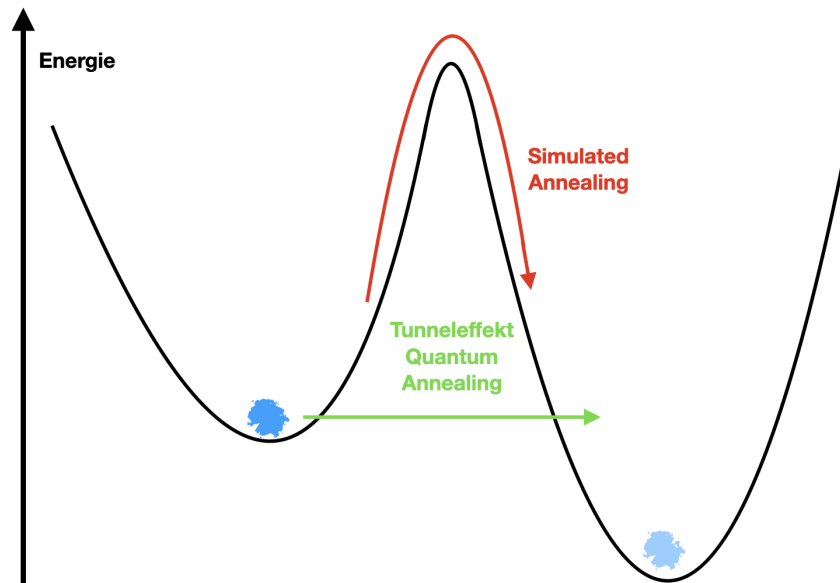


Abbildung 4.2.: Obwohl die Aufenthaltswahrscheinlichkeit (hier mit der Intensität des blauen Punktes gekennzeichnet) des Teilchens zu dem betrachteten Zeitpunkt in dem lokalen Minimum links höher ist, kann das Teilchen auch in dem globalen Minimum rechts gemessen werden. Daher kann ein quantenmechanisches System aufgrund der physikalischen Eigenschaften durch eine Potentialbarriere *tunneln*. (Inspiration von [dwa21])

4.8. Reverse Annealing

Reverse Annealing ist Quantum Annealing sehr ähnlich. Normalerweise wird s der Formel 4.1 jedoch von 0 zu 1 erhöht. Nun gilt bei $t = 0$ jedoch $s = 1$, um bis t_f auf einen bestimmten Annealing Bruchteil $s = f$ in Richtung 0 abzustiegen. Nach einer Pause steigt s dann wieder auf 1. Am Anfang gilt daher:

$$H(s) = B(s = 1)H_p \text{ mit } s(t = 0) = 1$$

Nach beispielsweise $t_f = 5\mu s$ gilt dann:

$$H(s) = A(s = f)H_0 + B(s = f)H_p \text{ mit } s(t = t_f) = f$$

und bei $t = T$ dann:

$$H(s) = B(s = 1)H_p \text{ mit } s(t = T) = 1$$

Der Vorteil des Reverse Annealing ist, dass klassisch gute Lösungen als Zustände der Qubits bei $t = 0$ initiiert werden können. Dann findet durch Reverse Annealing eine lokale Suche statt, deren Weite durch den Annealing Bruchteil f festgelegt wird. Initiiert man das System beispielsweise im Zustand des linken Minimum von Abbildung 4.2, so kann - wenn f groß genug gewählt wird- mithilfe des Reverse Annealing das rechte, globale Minimum gefunden werden. In Abbildung 4.3 wird ein typischer Reverse Annealing Sche-

dule dargestellt. Für detailliertere Informationen zu Reverse Annealing wird auf [\[dwa21\]](#) oder [\[ONL18\]](#) verwiesen.

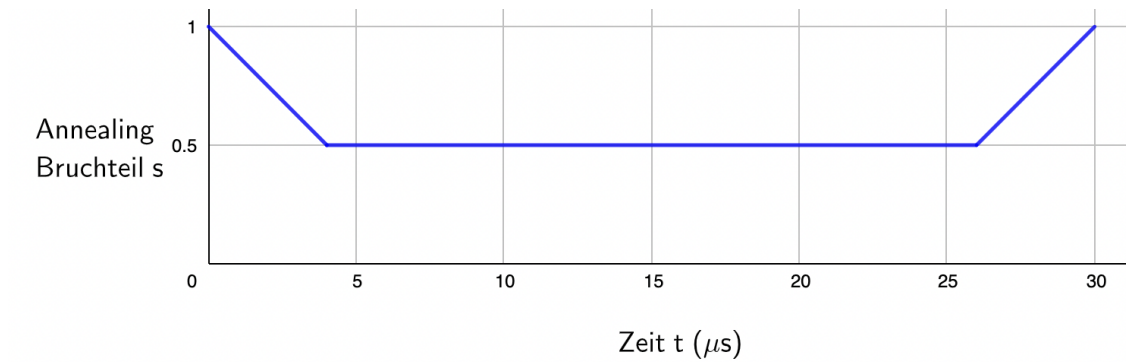


Abbildung 4.3.: Ein Annealing Schedule wird dargestellt bei dem der Annealing Bruchteil s innerhalb $4\mu s$ auf $s = f = 0.5$ absteigt. Dann findet eine Pause von $22\mu s$ statt, nach der s innerhalb von $4\mu s$ wieder auf $s = 1$ steigt. (Grafik inspiriert von [\[dwa21\]](#))

5. Methodik

In diesem Kapitel wird auf den eigentlichen Versuch der Arbeit eingegangen. Ziel der Arbeit ist ein Verfahren zu evaluieren, mit dem 3SAT Probleme künftig eventuell schneller gelöst werden können als mit klassischen Methoden bisher. Dabei wird Reverse Annealing verwendet. Der Versuch besteht aus grundsätzlich 3 Teilversuchen. Beim ersten wird versucht mit einem *Branch and Cut*-Verfahren gute Anfangszustände für die mit kleinen 3SAT Problem assoziierten MILP Probleme für das Reverse Annealing zu erhalten. Diese werden mit einem für das 3SAT Problem assoziierte MIS-Problem optimalem Anfangszustand und einem zufälligen Anfangszustand verglichen. Der zweite Versuch prüft die Skalierbarkeit des Verfahrens mit größeren 3SAT Instanzen. Der dritte Teil vergleicht die benötigte Zeit für dieses Verfahren mit der Lösungszeit eines 3SAT-Problems mit einem SAT-Solver. Darauf folgt je ein Kapitel über den verwendeten MILP-Solver und Quantencomputer.

5.1. Vergleich der Qualität der generierten Anfangszustände für das Reverse Annealing

Die 3SAT-Probleme werden in MILP Instanzen nach [2.4] umgewandelt. Da zahlreiche Kombinationen getestet werden, folgt eine Liste der durchgeführten Schritte.

Erstellung der 3SAT Instanzen

Es werden 50 schwere 3SAT-Instanzen mit 15 Variablen und 61 Klauseln erstellt. (Klausel-zu-Variablen Verhältnis ≈ 4.06) Die Konstruktion erfolgt 'unforced'. (vgl. Anmerkung [2.4]) Ausschließlich erfüllbare Instanzen werden verwendet. Diese Information wird dem Solver jedoch nicht mitgeteilt.

Umwandlung in MILP

Die Instanzen werden mit [2.4] in ein MILP umgewandelt, bei dem 15, 14, 13 und 12 zufällig ausgewählte Variablen ihre Ganzzahligkeitsbedingung verlieren. Dabei wird die intuitive, in Kapitel [2.4] erwähnte Zielfunktion gewählt. Es werden 10 Instanzen der Umwandlungen mit 14, 13 und 12 nicht-ganzzahligen Variablen und eine mit 15 erstellt. Die Variablen ohne Bedingung zur Ganzzahligkeit werden nach der Bestimmung der optimalen Lösung durch den Branch&Cut Solver Gurobi (vgl. Kapitel [5.4]) nach mathematischen Regeln gerundet.

Umwandlung in QUBO

Nachdem die $50 * (10 * 3 + 1) = 1550$ MILP Lösungen berechnet wurden, folgt die Umwandlung in eine MIS Zuordnung wie in Kapitel [2.3] dargestellt. Auch die optimale Lösungen der 50 3SAT Instanzen werden wie im Kapitel [2.3] beschrieben in ein MIS übersetzt. Es werden zusätzlich 50 zu den 3SAT-Instanzen assoziierte zufällige

MIS Zuordnungen erstellt. Durch Anmerkung 4.4 kann eine MIS Zuordnung direkt als eine QUBO Zuordnung aufgefasst werden.

Lösen mit Advantage 4.1

Die erhaltenen $(50 * (10 * 3 + 1)) + 50 + 50 = 1650$ QUBO Zuordnungen werden nun mithilfe des Reverse Annealings des Dwave Advantage 4.1 von Dwave Systems gelöst. Dabei muss zuvor ein Minor Embedding auf den Pegasusgraphen stattfinden (vgl. Kapitel 4.6), das jedoch von Dwave selbst übernommen wird. (vgl. dwa21) Es wird dann auf der einen Seite geprüft, welcher Anteil der Samples korrekten 3SAT Zuordnung entsprechen. Die Umwandlung erfolgt einmal mit und einmal ohne *Postprocessing* Schritt. (vgl. Kapitel 2.4) Auf der anderen Seite wird Korrektheit der MIS-Zuordnung betrachtet. Bei Letzterem, also ohne *Postprocessing* oder Interpretation wird auch der Hamming Abstand (vgl. Kapitel 2.5) zur nächsten Lösung des MIS berechnet. Pro Durchgang mit Reverse Annealing werden 100 Samples produziert. (vgl. Kapitel 4.8) Nach jedem Messvorgang eines Sample wird der Anfangszustand wieder auf den Anfangszustand gebracht. Der Annealing Schedule hat eine Steigung an der absteigende und aufsteigenden Flanke von 0.3 Mikrosekunden und eine Pause von 20 Mikrosekunden. Um die Lösungslandschaft des 3SAT Problems genauer zu erkunden, werden 10 Annealing Faktoren (0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.6) getestet.

Da zwischen 0.5 und 0.6 eine starke Veränderung stattfindet (vgl. Abbildung 6.3) werden auch hier noch einmal 8 weitere Annealing Faktoren (0.51, 0.52, 0.53, 0.54, 0.55, 0.56, 0.57, 0.58, 0.59) getestet. Aufgrund des Aufwands jedoch nur mit 14 und 13 relaxierten Variablen und der optimalen und zufälligen Zuordnung. Es wurden nur die ersten 5 zufälligen Umwandlungen von 3SAT zu MILP verwendet.

Für den ersten Versuch wurden somit $((50 * (10 * 3 + 1)) + 50 + 50) * 100 * 10 = 1650000$ Lösungen des Annealers ausgewertet. Für den zweiten wurden $((50 * (10 * 2)) + 50 + 50) * 100 * 5 = 550000$ Samples erstellt. Es wurden daher insgesamt 2200000 Samples ausgewertet.

5.2. Prüfung der Skalierbarkeit

Um die Skalierbarkeit zu testen, wird das Verfahren von Teilversuch 1 (Kapitel 5.1) mit weniger Durchgängen auf ein größeres 3SAT Problem angewendet (40 Variablen, 168 Klauseln). Es werden 50 3SAT-Instanzen *unforced* erstellt. Dann werden die durch Reverse Annealing erhaltenen Lösungen optimaler und zufälliger QUBO Zuordnungen mit denen durch die Umwandlung von 750 MILP Zuordnungen (jeweils 5 zufällige Umwandlungen zum MILP mit 38, 37 und 36 relaxierte Variablen) erhaltenen Zuordnungen verglichen. Der Annealing Schedule ist identisch mit dem von Teilversuch 1. Die Sample Anzahl beträgt wieder 100. Es wurden die Annealing Faktoren 0.2, 0.4 und 0.6 getestet. In diesem Teilversuch wurden $50(5 * 3) * 100 * 3 = 225000$ Lösungen ausgewertet.

5.3. Vergleich der Lösungszeiten eines 3SAT-Problems

In diesem Teilversuch wird die Effizienz der Methode getestet. Die Lösungszeit des SAT-Solvers Minisat wird mit dem in Teilversuch 1 beschriebenen statistischem Lösungsansatz

verglichen.

3SAT Testinstanzen

Es werden 2 große 3SAT-Instanzen (700 Variablen, 2800 Klauseln) aus einer Sat Competition von 2003 verwendet. Um die benötigte Zeit auch für kleinere Instanzen abschätzen zu können, werden 10 Instanzen mit 100 Variablen und 420 Klauseln und 10 Instanzen mit 300 Variablen und 1260 Klauseln 'unforced' erstellt.

Lösungszeit Minisat

Die Instanzen werden mit Minisat gelöst und die benötigten Zeiten notiert.

Lösungszeit des hybriden Ansatzes

Die gesamte Lösungszeit des neuen hybriden Lösungsansatzes berechnet sich aus der Summe der Lösungszeit des MILP, der benötigten Zeit zum Minor Embedding und der Lösungszeit eines Reverse Annealing Vorgangs. Hier wird die Dauer der beiden letzten vernachlässigt. Das ist valide, da angenommen wird, dass die Lösungszeit des MILPs weit größer ist als die Zeit zur Suche eines *Minor Embedding* oder der Samplingzeit des *Reverse Annealing*. Dazu kommen weitere Faktoren wie Zugriffszeiten, die jedoch auch vernachlässigt werden. Quantum Annealing findet mit dem verwendeten Annealing Schedule im Bereich $10^{-5}s$ statt. Auch wenn viele Samples ausgewertet werden, hat das vor allem bei großen 3SAT-Instanzen einen sehr kleinen Anteil an der benötigten Zeit des Algorithmus. Wie lange *Minor Embedding* bei großen 3SAT Instanzen dauert, muss noch quantitativ erforscht werden. Die 3SAT Instanzen werden mit Gurobi gelöst und die benötigte Zeit notiert. Es werden bei jeder Anzahl relaxierter Variablen 10 zufällige Umwandlungen in ein MILP vorgenommen. (vgl. Kapitel 2.4) Dabei werden bei der Instanz mit 700 Variablen 680, 670, 660 und 650 Variablen relaxiert und bei den 3SAT Problemen mit 300 Variablen werden 290, 280, 270, 260 und 250, bei den Problemen mit 100 Variablen 98, 96, 94, 92 und 90 Variablen relaxiert.

5.4. Gurobi 9.1

Die zum Zeitpunkt der Versuchsdurchführung aktuellste Version des Gurobi Solvers ist ein Programm, das sowohl LPs als auch MILP sehr schnell lösen kann. MILPs werden mithilfe eines *Branch and Cut*-Algorithmus (vgl. Kapitel 3.3.3) und anderen Erweiterungen bewältigt. Genauere Information zum Solver findet sich in [gur].

5.5. Dwave Advantage 4.1

Der Quantencomputer *Advantage* der Firma *D-Wave Systems* arbeitet mithilfe kleiner Metallschleifen, die jeweils Qubits oder deren Verbindungen (auch Couplers genannt) darstellen. Durch die geringe Temperatur bilden sie Strom und somit ein nach oben oder nach unten gerichtetes Magnetfeld aus. ^[1] Die zum Zeitpunkt der Versuchsdurchführung aktuellste Version des Quantencomputers *Advantage 4.1* besitzt mehr als 5000 Qubits

¹Für detaillierte Informationen wird [JAG⁺11] empfohlen. Der Artikel beschreibt die physikalischen Hintergründe genauer.

5. Methodik

und 35000 Couplers. Wie in Kapitel [4.4](#) erwähnt besitzt der Computer eine Betriebstemperatur von unter $15mK$. Als Verbindungsstruktur wird ein Pegasusgraph verwendet. Für eine genauere Beschreibung wird auf [\[MF20\]](#) verwiesen.

6. Ergebnisse

Nach der Ausführung der Teilversuche werden die Ergebnisse ausgewertet. Alle Daten sind der Arbeit angehängt. Standardabweichungen werden mithilfe der Formel [6.1](#) aus den jeweiligen Messwerten berechnet. (vgl. [DL12](#))

$$s = \sqrt{\frac{\sum_{i=1}^N x_i - \bar{x}}{N}} \quad (6.1)$$

6.1. Vergleich der Qualität der generierten Anfangszustände für das Reverse Annealing

6.1.1. Beschreibung

Abbildung [6.1](#) präsentiert die Ergebnisse des ersten Teilversuchs ohne Interpretation und *Postprocessing* der MIS-Zuordnung. (vgl. erster Versuchsteil des Kapitels [5.1](#)) Es ist zu erkennen, dass die optimale Zuordnung bis zum Annealing Bruchteil 0.25 sehr konstant fast 100% korrekte MIS Belegungen liefert. Ab Annealing Bruchteil 0.3 fällt jedoch die Rate der Lösungen ab bis sie bei Annealing Bruchteil 0.5 ungefähr 0% erreicht. Dies gilt auch für die Zuordnungen über das Lösen und Runden des MILP. Diese haben konstante Werte von 0.8%, 5.4% und 9.8% (für 14, 13 und 12 relaxierte Variablen), bis sie ab Annealing Bruchteil 0.3 beginnen abzufallen. Für diese und der optimalen Messung ab Annealing Bruchteil 0.3 lässt sich eine große Standardabweichung s bis Annealing Bruchteil 0.45 erkennen. Grafik [6.2](#) zeigt die Ergebnisse des ersten Teilversuchs (vgl. erster Versuchsteil des Kapitels [5.1](#)) mit der Interpretation in eine 3SAT-Belegung. Der Verlauf der optimalen Zuordnung fällt ab Annealing Bruchteil 0.3 ab, bis die Wahrscheinlichkeit der Anteil korrekter 3SAT Zuordnungen bei Annealing Bruchteil 0.5 ungefähr 0 beträgt. Der Anteil korrekter Lösungen nach dem Reverse Annealing mithilfe der zufälligen Zuordnung und der Lösung des entwickelten Verfahrens mit 15, 14, 13, 12 Variablen stagniert von Annealing Bruchteil 0.1 bis 0.5 auf ungefähr 0, 0, 0.8, 5.4 und 10%. Bei Annealing Bruchteil 0.6 verändert sich jedoch die Prozentzahl korrekter Lösungen bei allen Anzahlen relaxierter Variablen des entwickelten Verfahrens. Ungefähr 3% aller durch die vier Arten des Verfahrens (Reverse Annealing mit der Lösung eines MILP mit 15, 14, 13 oder 12 relaxierte Variablen) lassen sich als korrekte 3SAT-Belegungen interpretieren. 0% der Samples des Reverse Annealing mit der optimalen oder zufälligen Zuordnung lassen sich bei Annealing Bruchteil 0.6 als korrekte 3SAT-Belegungen interpretieren. Abbildung [6.3](#) präsentiert die Ergebnisse des ersten Teilversuchs - also des ersten Versuchsteils des Kapitels [5.1](#) - mit *Postprocessing* nach der Interpretation in eine 3SAT-Belegung. Der Anteil korrekter 3SAT-Belegungen der Samples des Reverse Annealing mit optimalem Anfangszustand bleibt von Annealing Bruchteil 0.1 bis 0.45 auf fast 100%. Bei der Fraktion 0.5 sinkt dieser auf ungefähr 95% und fällt dann beim Bruchteil 0.6 auf 5.9%. Die Anteile korrekter 3SAT-Zuordnungen des entwickelten Verfahrens mit 15, 14, 13 und 12 relaxier-

ten Variablen und die der zufälligen Zuordnung bleiben von Annealing Bruchteil 0.1 bis 0.5 stagnieren auf ungefähr 0, 1, 5.5, 10.4 und 0. Bei Annealing Bruchteil 0.6 ist der Anteil korrekter 3SAT-Belegungen aller im Teilversuch getesteten Verfahren zur Auswahl des Anfangszustandes des Reverse Annealing (zufällig oder Lösen des MILP mit 15, 14, 13 und 12 Variablen) ungefähr 3%. Um einen detaillierteren Einblick in die Lösungslandschaft zu erreichen, werden alle Diagramme auf den Annealing Bruchteil Bereich von 0.51 bis 0.59 erweitert. Grafik 6.4 beschreibt die Ergebnisse des zweiten Teils des ersten Teilversuchs, also die Anteile korrekter MIS-Zuordnungen der Reverse Annealing Samples zwischen Annealing Bruchteil 0.51 und 0.59. (vgl. Kapitel 5.1) Es ist zu erkennen, dass der Anteil korrekter MIS-Belegungen nach den Reverse Annealing Durchgängen mit denen in dem Versuch getesteten Varianten zum Finden eines guten Anfangszustands alle ungefähr 0% der Samples darstellt. Abbildung 6.5 präsentiert die Ergebnisse des zweiten Teils des ersten Teilversuchs mit der Interpretation der MIS-Zuordnung in eine 3SAT-Belegung. (vgl. Kapitel 5.1) Die Samples des Reverse Annealing mit einem optimalen bzw. zufälligen Anfangszustand sind zu ungefähr 0% korrekte 3SAT-Belegungen. Die entwickelte Methode mit 13 relaxierten Variablen besitzt einen Anteil korrekter 3SAT-Zuordnungen von ungefähr 6% bei Annealing Bruchteil 0.51 und steigt dann langsam ungefähr linear ab bis bei Annealing Fraktion 0.59 ungefähr 2.6% der Samples als 3SAT-Lösung interpretiert werden kann. 1.1% der Samples des Reverse Annealing mit dem Anfangszustand der gelösten MILP mit 14 relaxierten Variablen bei Annealing Bruchteil 0.51 lassen sich als korrekte 3SAT-Zuordnungen interpretieren. Der Anteil steigt an auf ungefähr 2.3% bei Annealing Bruchteil 0.59. Grafik 6.6 präsentiert die Ergebnisse des zweiten Teils des ersten Teilversuchs mit einem *Postprocessing* Schritt nach der Interpretation der MIS-Zuordnung in eine 3SAT-Belegung. (vgl. Kapitel 5.1) Es ist zu erkennen, dass der Anteil korrekt zu einer 3SAT-Belegung umgeformter Samples des Reverse Annealing des entwickelten Ansatzes mit 14 relaxierten Variablen und mit einem zufälligen Anfangszustand von 1.1% bzw. 0.2% bei Annealing Bruchteil 0.51 auf ungefähr 2.5% bei Annealing Bruchteil 0.59 ansteigt. Auch der Anteil korrekter 3SAT-Belegungen des entwickelten Verfahren mit 13 relaxierten Variablen beträgt bei Annealing Bruchteil 0.59 ungefähr 2.5%. Jedoch steigt dieser von Annealing Bruchteil 0.51 mit einem Anteil von 5.9% ab. Auch der Anteil korrekter 3SAT-Belegungen der erhaltenen Samples des Reverse Annealing mit einem optimalen Anfangszustand steigt von Annealing Bruchteil 0.51 bis 0.59 ab. Bei Annealing Fraktion 0.51 können ungefähr 95% der Samples einer korrekten 3SAT-Belegung zugeordnet werden. Bei Annealing Bruchteil 0.59 sind es nur mehr noch 12.7%. Grafiken 6.7 und 6.8 beschreiben die durchschnittlichen Hamming-Distanzen zwischen denen durch das Reverse Annealing erhaltenen MIS-Zuordnungen und den nächsten für das Problem optimalen MIS-Belegungen. Die durchschnittlichen Hamming-Distanzen stagnieren von Annealing Bruchteil 0.1 bis 0.5. Es ist zu erkennen, dass die Samples des Reverse Annealing mit dem zufälligen Anfangszustand eine Distanz von ungefähr 85.6, die der Samples des entwickelten Verfahrens mit 15, 14, 13 und 12 relaxierten Variablen ungefähr 84.3, 83.2, 80.5 und 77.1 und die des Reverse Annealing mit einem optimalen Anfangszustand ungefähr 40 besitzt. Die Hamming-Distanz der MIS-Zuordnungen des Reverse Annealing mit optimalen Anfangszuständen steigt von 40.6 bei Annealing Bruchteil 0.5 langsam auf 64.6 bei Annealing Bruchteil 0.6. Die Distanzen des entwickelten Verfahren mit 15, 14, 13 und 12 relaxierten Variablen steigen von Annealing Fraktion 0.5 bis 0.6 auf ungefähr 72.8, 74.2, 73.1 und 72.6 ab. Die durchschnittliche Hamming-Distanz der Samples des Reverse Annealing mit einem zufälligen Startzustand sinkt von Annealing Bruchteil 0.5 bis 0.6 auf

ungefähr 72.4 ab. Abbildungen 6.9, 6.10 und 6.11 stellen dar, wieviele 3SAT-Instanzen (bzw. MIS-Instanzen, falls die Zuordnung nicht als 3SAT-Belegung interpretiert wurde) das Verfahren tatsächlich löste. Abbildung 6.9 stellt dabei dar, wieviele MIS-Probleme gelöst wurden und 6.10 bzw. 6.11 wieviele 3SAT-Zuordnungen ohne bzw. mit *Postprocessing*. Als erstes wird Grafik 6.9 beschrieben. Hier wird erkannt, dass das Reverse Annealing mit optimalem Anfangszustand 100% der Instanzen löst von Annealing Bruchteil 0.1 bis 0.4. Ab Annealing Fraktion 0.45 fällt der Anteil ab, bis bei 0.6 keine der MIS-Probleme mehr gelöst werden. Die Verläufe des zufälligen Anfangszustandes, und der entwickelten MEthode mit 15, 14, 13 und 12 Variablen bleibt konstant auf den Prozentwerten 0, 0, 0.8, 5.3 und 10 von Annealing Bruchteil 0.1 bis 0.45. Dann sinken diese auf die Werte 0, 0, 0.4, 2 und 3.4 bei Annealing Fraktion 0.5 und bei 0.6 kann schließlich keines der Verfahren mehr MIS-Probleme lösen. In Abbildung 6.10 wird erkannt, dass das Reverse Annealing mit optimalem Anfangszustand denselben Verlauf, wie in dem letzten beschriebenen Diagramm besitzt. Die Verläufe der entwickelten Methode mit 15, 14, 13 und 12 relaxierten Variablen bleiben konstant von Annealing Bruchteil 0.1 bis ungefähr 0.35 auf 0, 1, 5.4 und 10 Prozent und steigen dann langsam an. Bei Annealing Fraktion 0.5 erreichen sie Werte von 2.0, 3.2, 7.2 und 12.8 Prozent und steigen bei Annealing Bruchteil 0.6 auf 52, 34.4, 37.6 und 40.4 Prozent. Das Reverse Annealing mit einem zufälligen Anfangszustand löst keine MIS-Instanz unabhängig des Annealing Bruchteils. In Grafik 6.11 ist der Verlauf der entwickelten Methode nahezu identisch mit dem der entwickelten Methode der Abbildung 6.10. Bis Annealing Bruchteil 0.45 bleibt er ungefähr konstant und steigt dann bis Bruchteil 0.6 an. Nun steigt jedoch die Lösungsrate des Verfahrens mit einem zufälligen Anfangszustand ab 0.5 auch an. Nachdem dieses im Bereich des Annealing Bruchteils 0.1 bis 0.45 keine 3SAT-Instanzen lösen kann, werden ungefähr 2% bei 0.5 und 48% der Probleme bei Annealing Fraktion 0.6 gelöst. Die Standardabweichung aller Daten der zuvor genannten Abbildungen ist meist sehr groß.

6.1.2. Interpretation

Das Ergebnis des Reverse Annealing mit optimaler Zuordnung wird im Bereich der Annealing Bruchteile 0.35 und 0.6 stark durch die Anwendung des *Postprocessing* Schrittes verbessert. Das wird deutlich im Vergleich der Abbildung 6.2 mit Grafik 6.3 erkannt. Die Ergebnisse werden hingegen überhaupt nicht durch die Umwandlung in eine 3SAT Belegung ohne *Postprocessing* verändert. (vgl. Grafik 6.1 und 6.2) Desweiteren bleiben die Hamming-Distanzen der optimalen Lösung in 6.7 von Annealing-Bruchteil 0.1 bis 0.5 gleich. Trotzdem sinkt der Bruchteil korrekter MIS Belegungen des Reverse Annealing mit optimaler Zuordnung schon ab Annealing Bruchteil 0.3. Daher kann festgehalten werden, dass die Ergebnisse des Reverse Annealing mit der optimalen MIS Belegung zwar zwischen Annealing Bruchteil 0.3 und 0.5 durchschnittlich gleich nah an der nächsten Lösung bleiben, sie aber immer weniger als korrekte 3SAT Zuordnungen interpretiert werden können. Es gibt mehrere Ursachen, die das hervorrufen könnten. Eine Interpretation der Ergebnisse wäre, dass das Reverse Annealing mit optimaler Zuordnung dazu tendiert, IS zu produzieren, die nicht maximal, aber korrekt sind. Dadurch würden wie zum Beispiel in Abbildung 2.5 durch einen *Postprocessing* Schritt MIS Zuordnungen zu korrekten 3SAT Lösungen umgewandelt werden können. Nur mit der in Kapitel 2.3 beschriebenen Interpretation würde die 3SAT Formel jedoch unerfüllt bleiben. Das Ergebnis der zufälligen Zuordnung wird nur durch die Interpretation zu einer 3SAT Belegung stark verbessert.

Auch das Resultat der entwickelten Methode wird nur durch die Interpretation als eine 3SAT-Zuordnung maßgeblich verbessert. Ein *Postprocessing* Schritt vergrößert die durchschnittlichen korrekten 3SAT Belegungen nur minimal. Es kommt deswegen bei der reinen Interpretation dazu, dass das Verfahren mit dem optimalen Anfangszustand nach dem Reverse Annealing zwischen Annealing Bruchteil 0.5 und 0.6 sehr viel weniger korrekte 3SAT Zuordnungen besitzt als das Reverse Annealing Verfahren der entwickelten Methode. Wie in der vorangegangenen Interpretation, gibt es Möglichkeiten die Beobachtung zu erklären. Da das Ergebnis der zufälligen Zuordnung und der der entwickelten Methode stark durch die reine Interpretation verbessert wird und kaum durch den darauffolgenden *Postprocessing* Schritt, lässt das die Interpretation zu, dass die MIS Zuordnung zu viele Knoten besitzt. Das heißt in vielen Klauseln befinden sich mehr als ein Literal, dass in das IS aufgenommen wurde. Daher ist das MIS zwar inkorrekt, bei der Interpretation wird diese dann aber in eine korrekte 3SAT Belegung transformiert. (vgl. Kapitel 2.3) Daher kann es entgegen der Intuition passieren, dass die Ergebnisse des Reverse Annealing mit der optimalen Zuordnung zwischen den Annealing Bruchteilen 0.5 und 0.6 sehr viel schlechter sind als die der entwickelten Methode. Es wird erkannt, dass je weniger Variablen des entwickelten Verfahren relaxiert werden, desto besser das Ergebnis nach dem Reverse Annealing ist. Das kann intuitiv erklärt werden. Wenn mehr Variablen unserer Methode eine Ganzzahligkeitsbedingung besitzen, so wird auch der durchschnittliche Abstand zur korrekten Lösung weniger. Das kann sehr gut in Abbildung 6.7 erkannt werden. Dadurch steigt auch die Wahrscheinlichkeit eine korrekte Lösung des MIS oder der 3SAT Instanz zu erhalten. (vgl. Diagramme 6.1, 6.2, 6.3) Da das Problem schwer zu lösen und komplex ist, gibt es manche Kombinationen aus relaxierten Variablen und 3SAT Instanzen, mit denen sehr hohe korrekte Werte nach dem Reverse Annealing erzielt werden und manche, deren Ergebnisse sehr niedrig sind. Gleichzeitig gibt es aufgrund der Schwere des Problems und damit der Komplexität der Lösungslandschaft auch eine starke Abweichung beim Reverse Annealing selber. Da es viele lokale niedrige Minima gibt, ist die Wahrscheinlichkeit hoch, dass das heuristische Lösungsverfahren in einem dieser stecken bleibt. Daher sind auch die Abweichungen der Ergebnisse der Methode mit der optimalen Zuordnung groß. Da die Raten der korrekten Zuordnungen von Annealing Bruchteil 0.1 bis 0.35 bzw. 0.5 (ohne bzw. mit Interpretation und *Postprocessing*) sind bei allen Varianten nahezu konstant sind, kann angenommen werden, dass die Minima sehr weit auseinander liegen, und das Verfahren des Reverse Annealing deswegen erst bei einem großen Annealing Bruchteil aus dem Minima heraus- bzw. hineinkommen kann. Einige Interpretationen werden nun zu den Diagrammen der Anteile gelöster MIS-/3SAT-Formeln (vgl. Diagramme 6.9, 6.10, 6.11) getroffen. Es wird erkannt, dass die Dynamik der verschiedenen Verläufe ungefähr mit dem der normalen Diagramme übereinstimmt. Die optimale Zuordnung wird nur durch den *Postprocessing* Schritt wesentlich beeinflusst, alle anderen fast nur durch den Interpretationsschritt. Es wird beobachtet, dass der Verlauf der optimalen Zuordnung jedoch weit später bei 0.45 ohne *Postprocessing* abfällt. Der Abfall mit dem *Postprocessing* Schritt ist sehr verlangsamt. Es kann daher angenommen werden, dass die Ergebnisse des Reverse Annealing der optimalen Zuordnung sehr viel mehr durch die Fluktuation des Reverse Annealing selber als durch die Auswahl der 3SAT Instanzen geprägt ist. So können auch die schon erwähnten gleichbleibenden Hamming Distanzen erklärt werden. Desweiteren kann festgestellt werden, dass die Prozentzahl gelöster 3SAT Instanzen nach der Interpretation beim Annealing Bruchteil 0.6 bei ungefähr 40% liegt. Dies ist sehr viel höher als die eigentliche Anzahl korrekter Zuordnungen. Dies deutet darauf hin, dass ab

6.1. Vergleich der Qualität der generierten Anfangszustände für das Reverse Annealing

0.5 ein Verhaltenswechsel des 3SAT Problem stattfindet. Während der Verlauf bis 0.5 ungefähr mit dem der Gesamtzahl korrekter Belegungen (vgl. Grafik 6.3) übereinstimmt, so stehen hohe Prozentzahlen der gelösten 3SAT-Instanzen einer niedrigen Gesamtzahl korrekter Zuordnungen entgegen. (vgl. Beobachtung 9 oder Abb. 6.11) Das bedeutet, dass der Verlauf ab 0.5 nicht mehr durch die passende Auswahl von 3SAT Instanzen, sondern mehr durch die statistischen Fehler des Reverse Annealing regiert wird. Darüberhinaus lässt diese Beobachtung die Interpretation zu, dass die Ergebnisse des Reverse Annealing mit der zufälligen und entwickelten Zuordnung stark von der Auswahl der 3SAT-Probleme - bzw. auch der relaxierten Variablen - abhängig sind. Identische Prozentwerte beider Diagramme bedeuten, dass ein korrektes Sample eines 3SAT-Problems, impliziert, dass fast alle 100 Samples der Insatnz korekt sind.

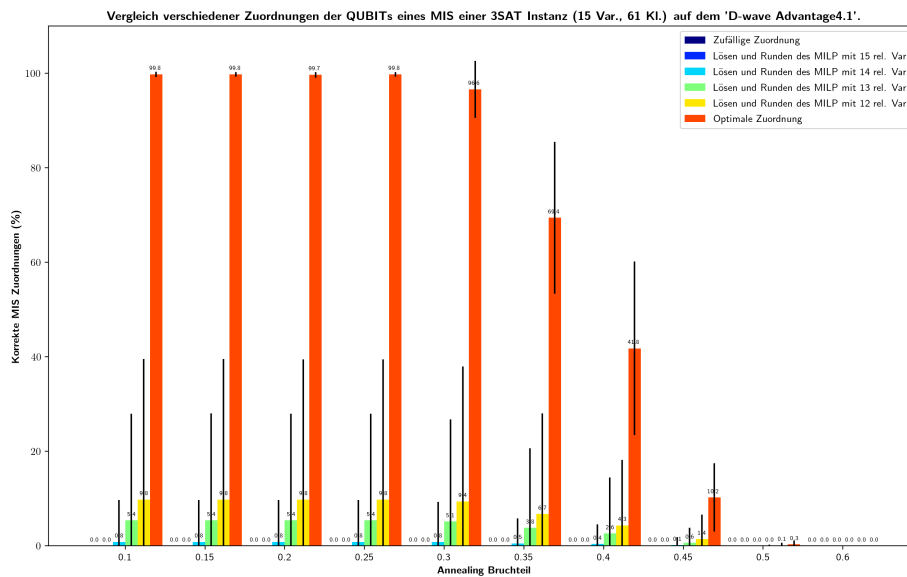


Abbildung 6.1.: Es werden die Anzahl korrekter MIS-Zuordnungen der Samples des Reverse Annealing angezeigt. Der schwarze Balken stellt die Standardabweichung dar.

6. Ergebnisse

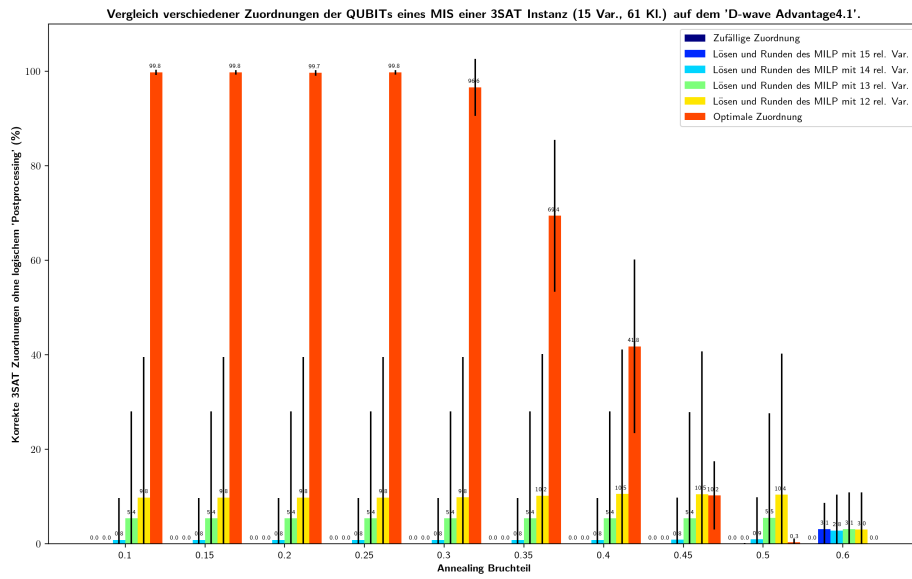


Abbildung 6.2.: Die Grafik zeigt die durchschnittlichen Bruchteile korrekter 3SAT Zuordnungen (ohne *Postprocessing* Schritt) der Samples des Reverse Annealing mit verschiedenen Startzuständen.

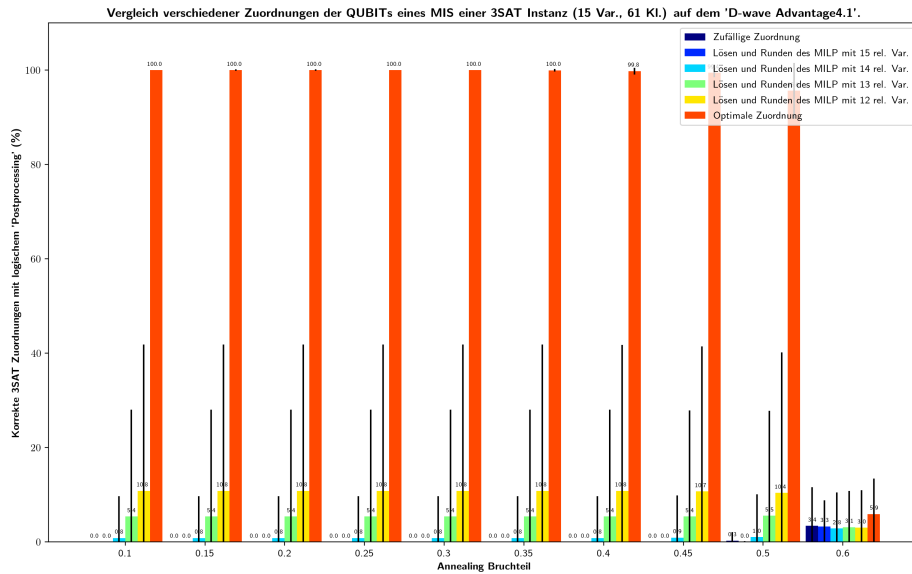


Abbildung 6.3.: Die Ergebnisse von Annealing Bruchteilen 0.1 bis 0.6 werden dargestellt. Dabei wird nach dem Reverse Annealing ein *Postprocessing* Schritt zur Umwandlung in eine 3SAT Belegung verwendet.

6.1. Vergleich der Qualität der generierten Anfangszustände für das Reverse Annealing

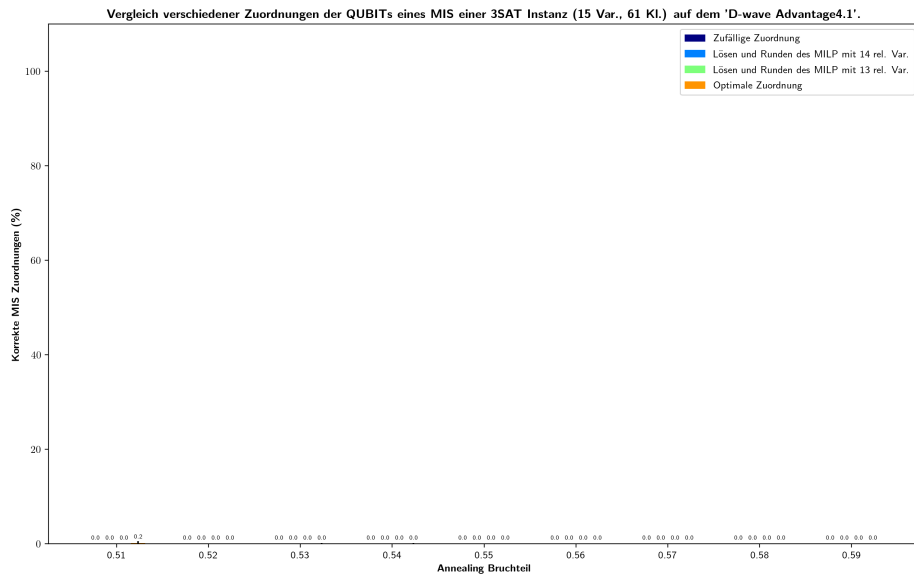


Abbildung 6.4.: In der Grafik erkennt man die Ergebnisse des Reverse Annealing ohne Interpretation als 3SAT Belegung.

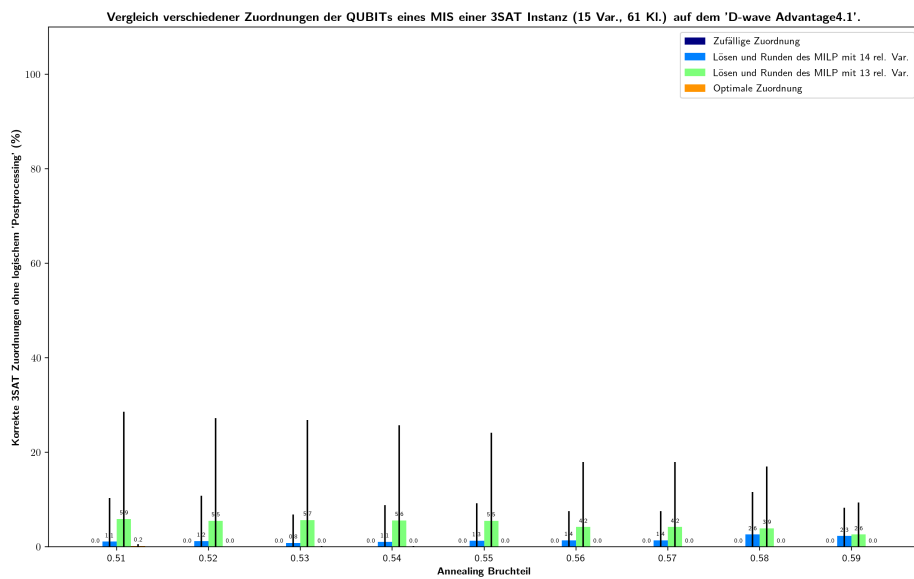


Abbildung 6.5.: Es werden die Ergebnisse des Reverse Annealing der in MIS umgeformte 3SAT Instanzen mit Annealing Bruchteilen 0.51 bis 0.59 dargestellt. Bei der Umwandlung wurde die MIS Belegung zwar als 3SAT Zuordnung interpretiert, jedoch kein *Postprocessing* verwendet.

6. Ergebnisse

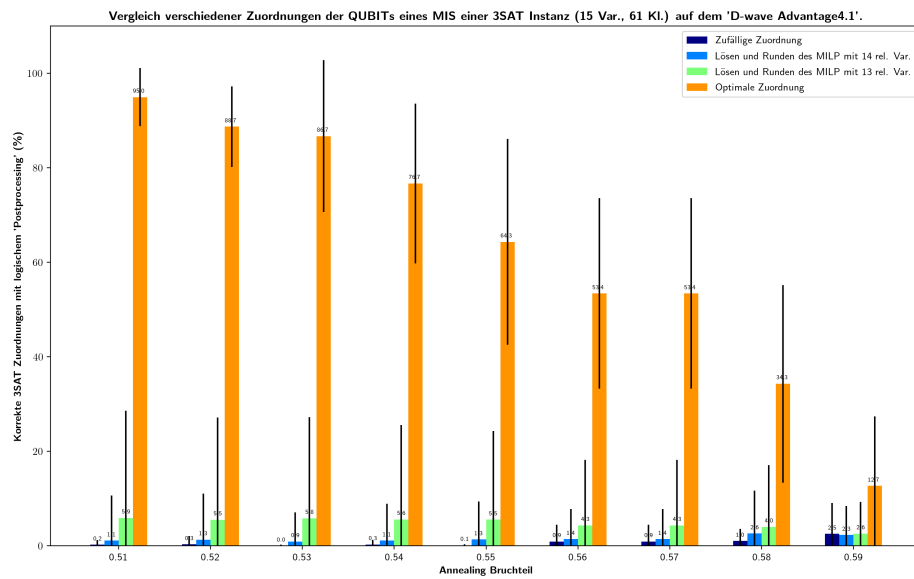


Abbildung 6.6.: Ähnlich wie Abbildung 6.3 stellt auch diese Grafik die Ergebnisse der Umwandlung der Lösungen des Reverse Annealing mit einem *Postprocessing* Schritt dar. Hier werden jedoch die Annealing Bruchteile 0.51 bis 0.59 betrachtet. Die 3SAT Instanz besitzt 15 Variablen und 61 Klauseln.

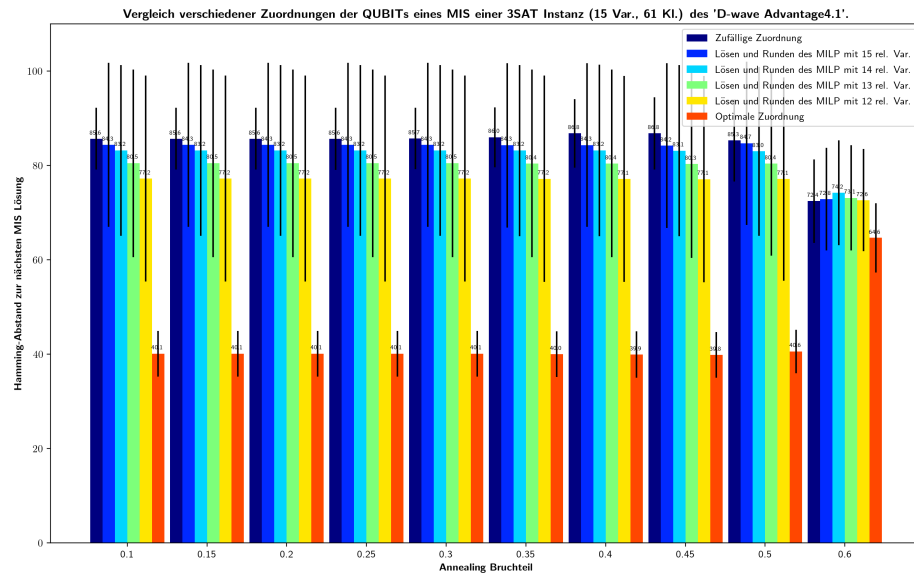


Abbildung 6.7.: In diesem Diagramm werden die durchschnittlichen Hamming- Distanzen der Samples des Reverse Annealing zur nächsten optimalen MIS Belegung dargestellt. Die schwarzen Striche bezeichnen die Standardabweichung.

6.1. Vergleich der Qualität der generierten Anfangszustände für das Reverse Annealing

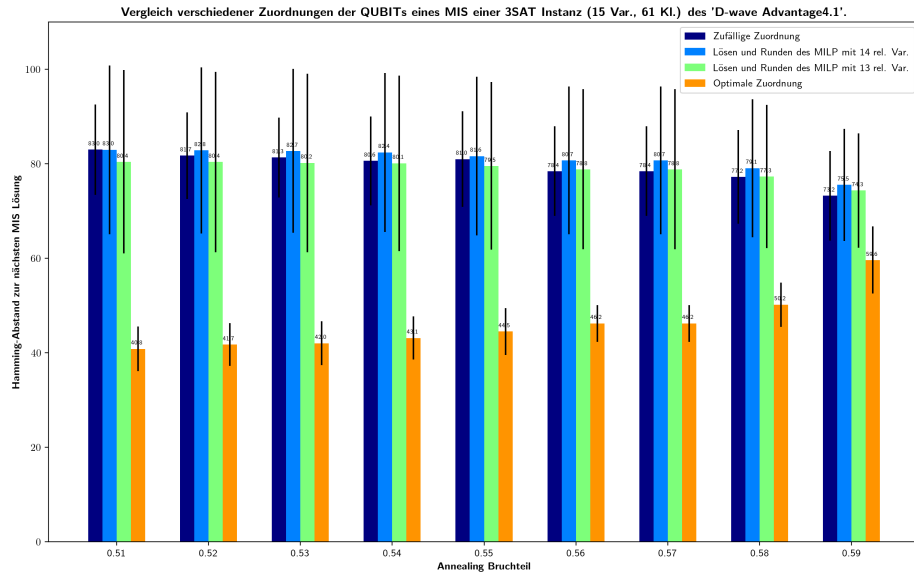


Abbildung 6.8.: Wie Abbildung 6.7 zeigt diese Grafik die durchschnittliche Hamming-Distanz an. Diesmal jedoch mit anderen Annealing Bruchteilen. (0.51 bis 0.59)

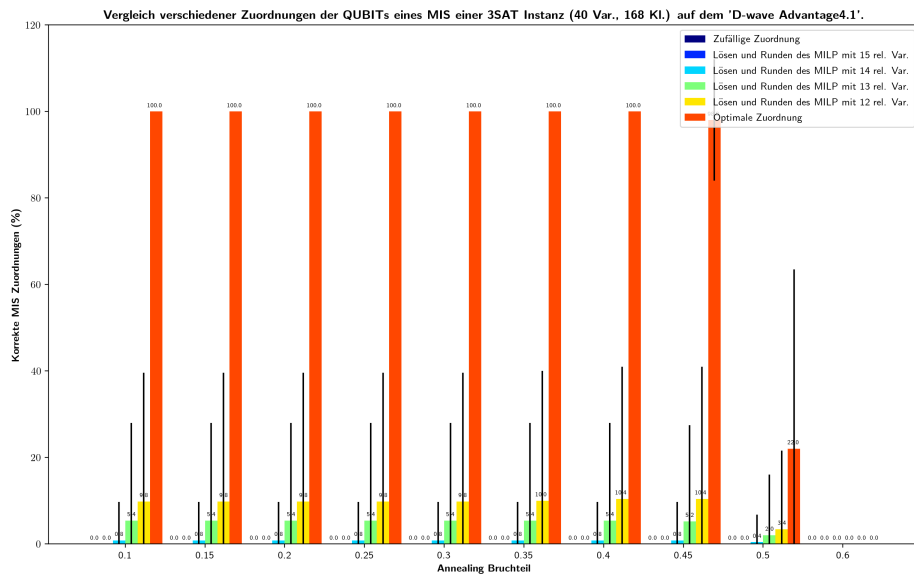


Abbildung 6.9.: Die Anzahl gelöster MIS Instanzen mithilfe des Reverse Annealing werden angezeigt.

6. Ergebnisse

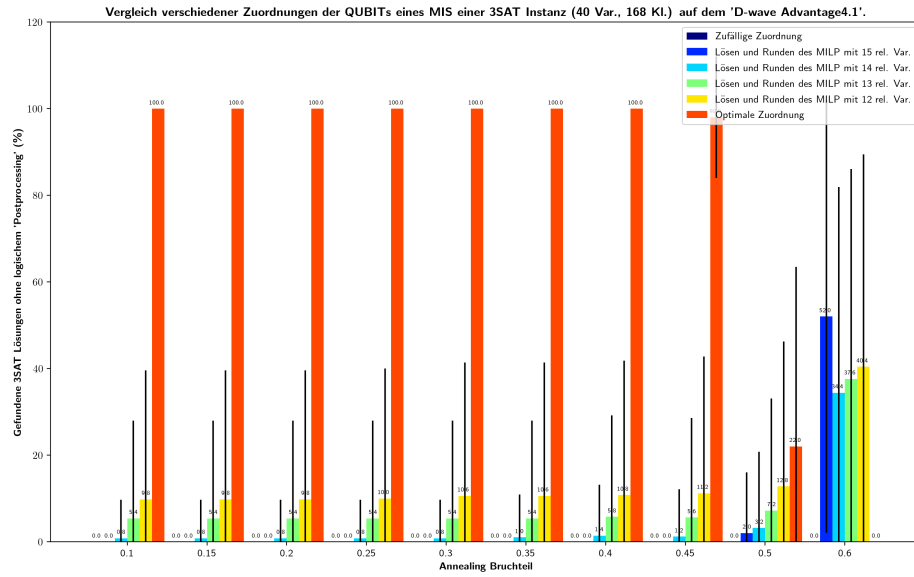


Abbildung 6.10.: Das Diagramm spiegelt wieder, wieviele 3SAT-Probleme gelöst wurden mit der reinen Interpretation der MIS-Belegung.

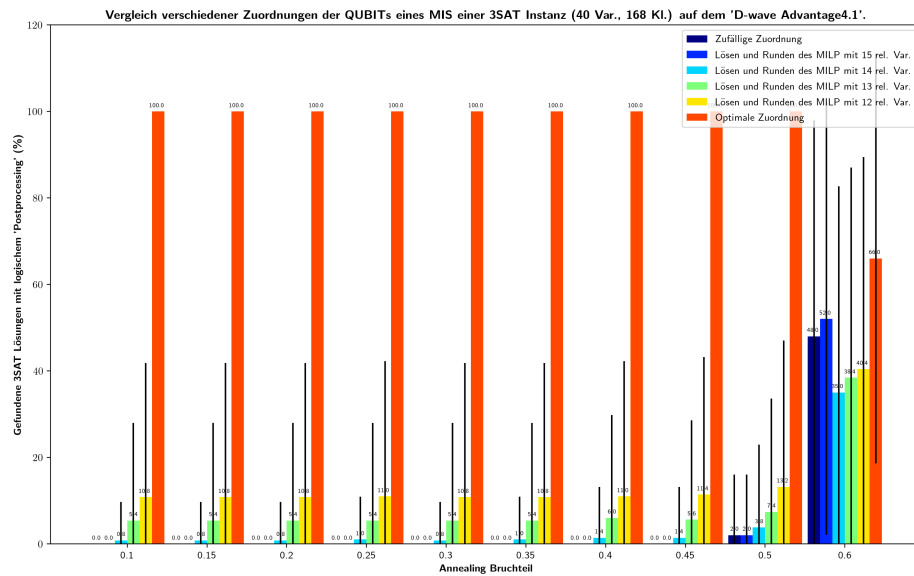


Abbildung 6.11.: Die Abbildung zeigt auf, welcher Prozentsatz der 3SAT Instanzen gelöst wurde.

6.1. Vergleich der Qualität der generierten Anfangszustände für das Reverse Annealing

6.2. Prüfung der Skalierbarkeit

6.2.1. Beschreibung

Gleiches wie in Teilversuch 1 (Kapitel 6.1) wird reduziert auch bei einer größeren 3-SAT Instanz durchgeführt. Aufgrund der Limitationen, die bisherige QC-Hardware aufweist, war es nicht möglich schwere 3SAT-Probleme zu verwenden, die signifikant größer als 40 Variablen waren. Durch das Minor Embedding (vgl. Kapitel 4.6) werden weit mehr Qubits verwendet werden, als die Transformation (beschrieben in Kapitel 2.3) erkennen lässt. In Abbildung 6.12 und 6.13 werden die Verläufe der korrekten MIS-Belegung bzw. der korrekten 3SAT-Zuordnungen ohne *Postprocessing* dargestellt. Beide besitzen sehr ähnliche Ergebnisse. Es wird wieder das Abfallen der zur optimalen Zuordnung gehörenden Balken erkannt. (vgl. Versuch 1, Abbildung 6.1 oder 6.2) Das Ergebnis bei Annealing Bruchteil 0.4 ist 20.8% korrekter Zuordnungen und besitzt eine hohe Standardabweichung. Auch in Abbildung 6.3, wo die Ergebnisse des Verfahrens mit *Postprocessing* dargestellt werden, wird die Ähnlichkeit zum Diagramm mit *Postprocessing* aus Teilversuch 1 erkannt. (6.3) Der Bruchteil korrekter 3SAT Lösungen stagniert auf fast 100% bei Annealing Bruchteil 0.2 und 0.4. Dann fällt er auf 0% bei Annealing Bruchteil 0.6. Das Reverse Annealing mit der zufälligen Zuordnung und der Belegung mithilfe des entwickelten Verfahren ergab hingegen in allen drei Abbildungen fast keine 3-SAT Lösungen. Es gibt einzig einen kleinen Peak von ungefähr 0.7% bei der Zuordnung mithilfe der Lösung eines MILP mit 37 relaxierten Variablen bei allen drei Abbildungen bei Annealing Bruchteil 0.2 und 0.4. (bei der Überprüfung der MIS-Zuordnung ist der Peak bei Annealing Bruchteil 0.4 lediglich 0.3%)

6.2.2. Interpretation

Die Interpretation des Verlaufs der korrekten Lösungen des Reverse Annealing mit optimaler Zuordnung gleicht dem des Kapitels 6.1. Durch die größere Anzahl an Knoten des MIS der 3SAT Instanzen mit 168 Klauseln (504 Knoten versus 183 Knoten) und die damit verbundene noch viel größere durch Minor-Embedding entstehende Anzahl an Qubits wird das Problem schwerer zu lösen. Daher gibt es fast keine korrekte Lösungen des Reverse Annealing mit der entwickelten Methode oder mit einer zufälligen Zuordnung. Trotzdem erkennt man, dass die Ergebnisse des Reverse Annealing mit einer optimalen Belegung bei Annealing Bruchteil 0.2 (vgl. Abbildung 6.1) nicht von dem Verlauf der korrekten Ergebnisse (vgl. Abbildung 6.1) abweicht. Man erkennt jedoch dass die relative Häufigkeit der korrekten Ergebnisse bei Annealing Bruchteil 0.4 halb so klein ist, wie bei den kleineren 3SAT Instanzen.

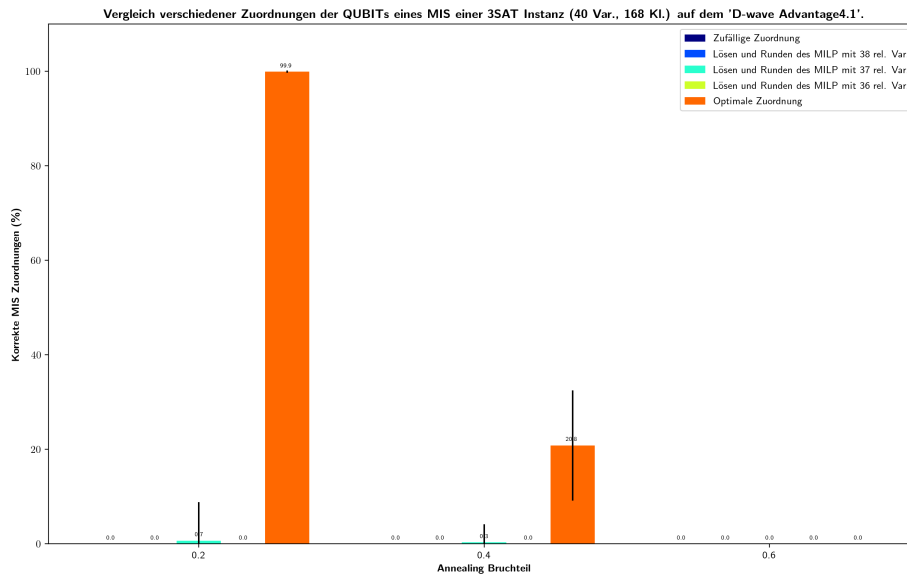


Abbildung 6.12.: Wie Abbildung 6.1 oder 6.4 zeigt dieses Diagramm die Ergebnisse der korrekten MIS Belegungen nach Reverse Annealing. Die Testinstanzen besitzen diese Mal jedoch 40 Variablen und 168 Klauseln.

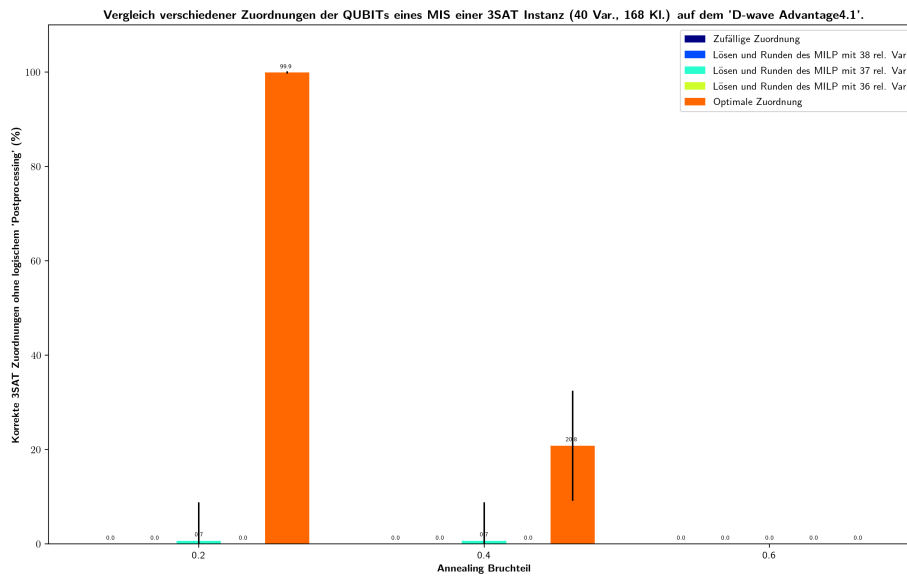


Abbildung 6.13.: Wie die Grafiken 6.2 oder 6.5 zeigt diese Abbildung die korrekten 3SAT Zuordnungen ohne *Postprocessing* Schritt. Dieses Mal werden 3SAT Probleme mit 40 Variablen und 168 Klauseln verwendet.

6. Ergebnisse

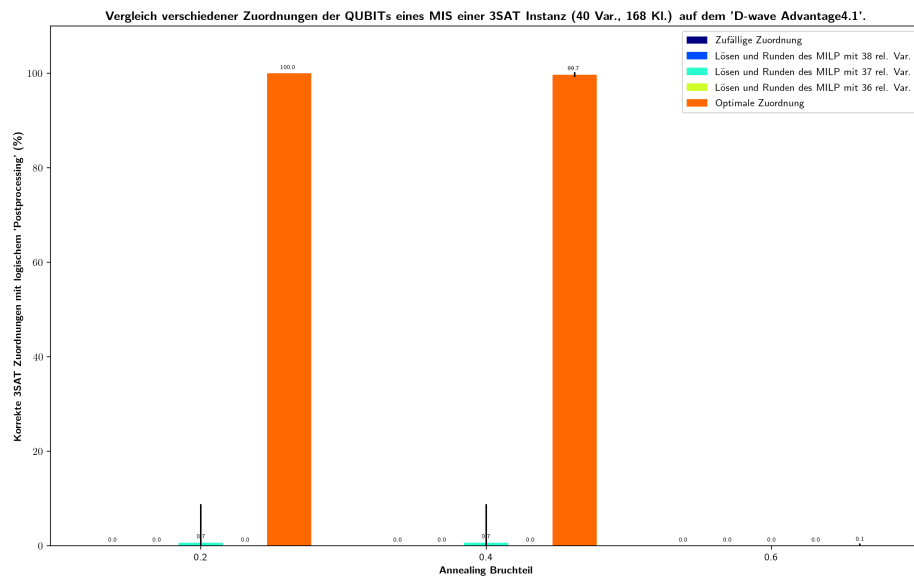


Abbildung 6.14.: Es werden die durchschnittlichen Bruchteile korrekter 3SAT Zuordnungen nach einem Reverse Annealing und *Postprocessing* aufgezeigt.

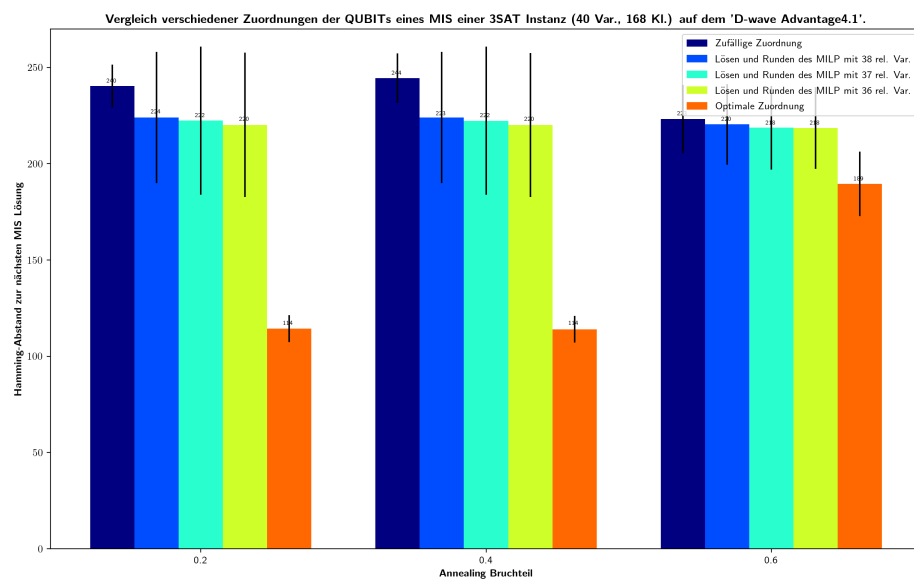


Abbildung 6.15.: Wie Grafik [6.7](#) zeigt auch diese Abbildung die durchschnittliche Entfernung der MIS Belegung nach dem Reverse Annealing zur nächsten optimalen MIS Zuordnung. Dieses Mal wird jedoch 3SAT Instanzen mit 40 Variablen und 168 Klauseln verwendet.

6.3. Vergleich der Lösungszeiten eines 3SAT-Problems

In Abbildung 6.16 wird erkannt, dass selbst das Lösen und Runden eines assoziierten MILPs mit 98% relaxierten Variablen bei 3SAT Instanzen mit 100 Variablen noch viel mehr Zeit benötigt, als das direkte Lösen mithilfe des SAT-Solvers Minisat. Diagramm 6.17 zeigt, dass bei Relaxierung von ungefähr 90% der Variablen das Lösen und Runden des MILP schneller sein kann als das direkte Lösen mithilfe von Minisat bei 3-SAT Instanzen mit 300 Variablen. Abbildung 6.18 präsentiert die Ergebnisse von 3-SAT Instanzen mit 700 Variablen. Es kann daraus geschlossen werden, dass ungefähr 93% der Variablen im MILP relaxiert werden können, um schneller zu sein als die Lösung mithilfe des 3-SAT-Solvers Minisat. Bei allen drei Teilversuchen besitzen die gemessenen Zeiten der entwickelten Methode eine sehr starke Standardabweichung. Die benötigten Zeiten des Sat-Solvers hingegen besitzen eine kleine Varianz.

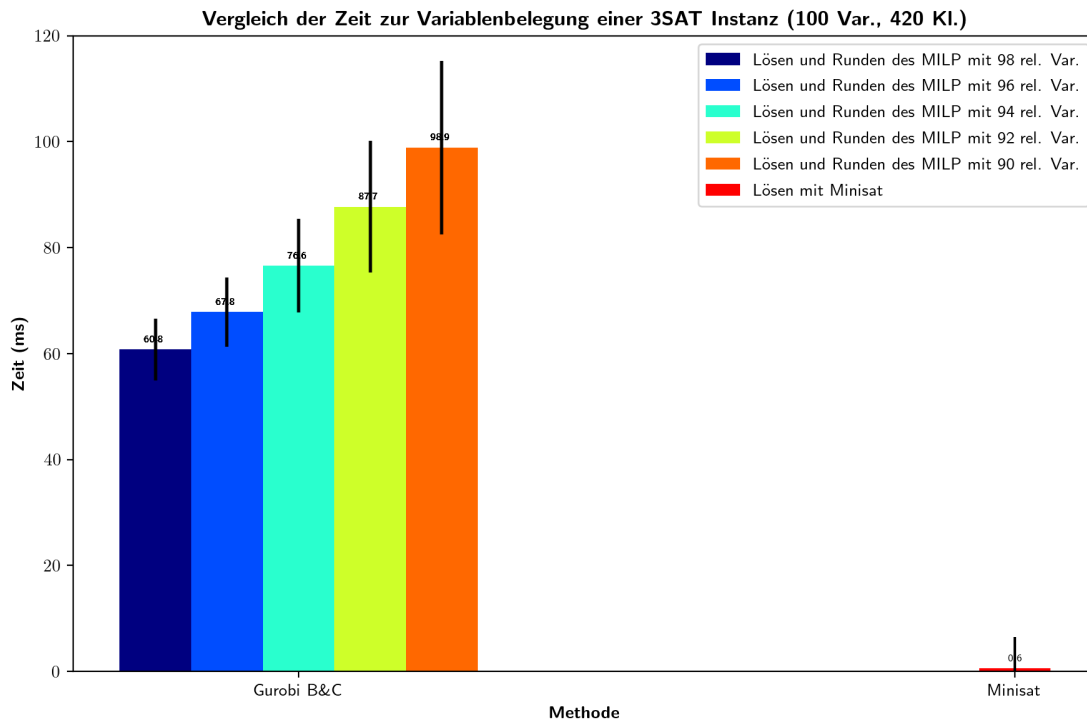


Abbildung 6.16.: Die Grafik vergleicht die benötigte durchschnittliche Lösungszeit verschiedener zu MILPs umgewandelter 3SAT Instanzen mit der des Minisat Solvers.

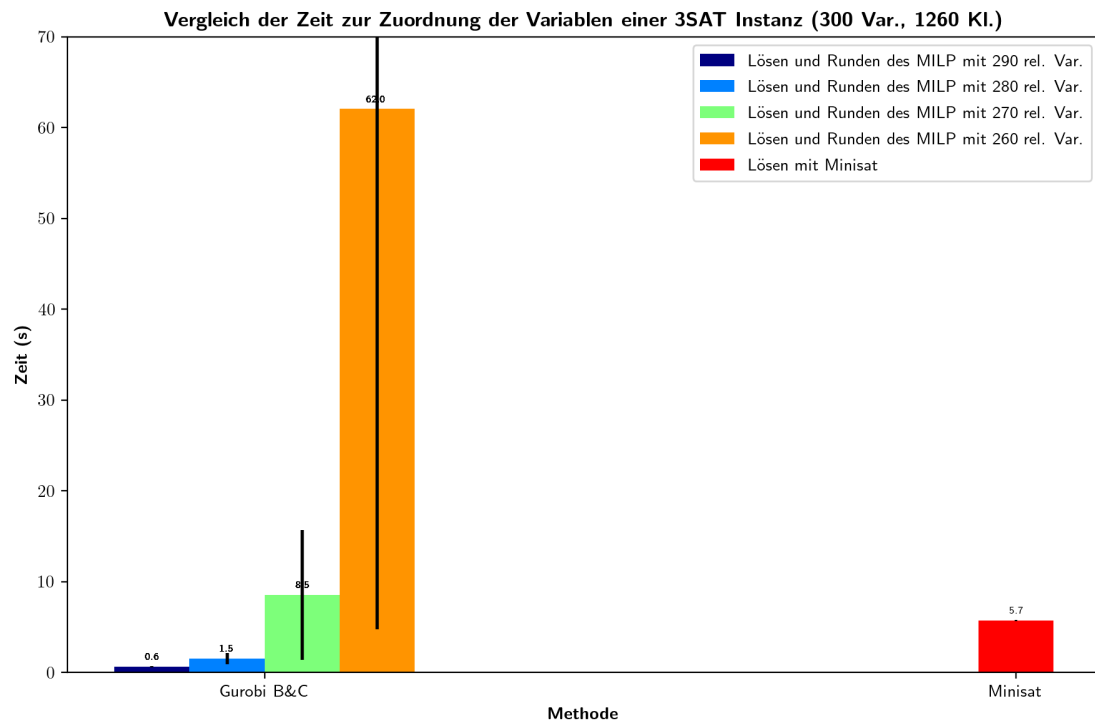


Abbildung 6.17.: Wie Diagramm 6.16 beschreibt auch diese Abbildung den Vergleich zwischen den durchschnittlichen Lösungszeiten verschiedener MILPs mithilfe dem *Branch and Cut* Solver Gurobi und denen des SAT-Solvers Minisat.

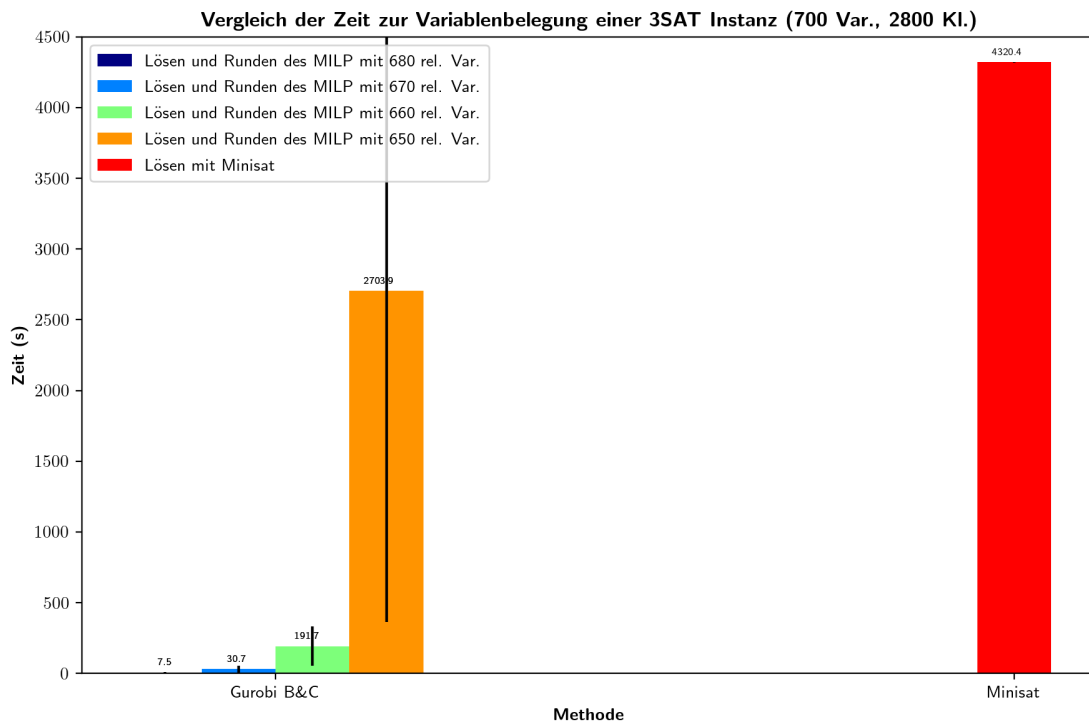


Abbildung 6.18.: Es wird der Vergleich der Lösungszeit des *Branch and Cut* Solvers Gurobi und dem SAT-Solver Minisat aufgezeigt.

7. Diskussion

In den vorangegangenen Beobachtungen und Interpretationen konnten die Eigenschaften des hybriden Verfahrens beleuchtet werden. In diesem Kapitel werden die wichtigsten Erkenntnisse noch einmal zusammengefasst und diskutiert. Es wird zuerst auf die Praktikabilität des Verfahrens und dann auf die spezifischen Eigenschaften der Lösungslandschaft eingegangen. Des Weiteren werden Vorschläge für die weitere Forschung gegeben.

Die Ergebnisse der Arbeit zeigen, dass für das Reverse Annealing bessere Anfangszustände als eine zufällige MIS-Zuordnung für das gegebene Problem produziert werden können. Es kann überdies festgestellt werden, dass bei 3SAT Instanzen ab 300 Variablen ungefähr 10 % der Variablen als Ganzzahlen fixiert werden können, sodass die Lösung des dazugehörigen MILP schneller ist als die Lösung des Minisat Solvers. In Diagramm 6.1 ist zu sehen, dass die ersten korrekten Lösungen bereits mit $\frac{1}{15}$ fixierten Variablen zu sehen sind. Da das Verfahren aber nicht einfach skalierbar ist (vgl. Diagramm 6.12), kann daraus nicht geschlussfolgert werden, dass das Verfahren schneller ist, als die klassische Lösung mithilfe des SAT-Solvers Minisat. Mithilfe längerer Annealing Schedules könnte dies jedoch geändert werden. Auch die verwendete Zielfunktion des MILPs kann sicherlich verbessert werden. (vgl. 2.4) Darüber hinaus ist die Lösungsmethode des MILPs noch sehr grob und nicht an die eigentliche Struktur des 3SAT Problems angepasst. Auch hier kann noch Zeit eingespart werden.

Durch die Beobachtungen und Interpretation aus Kapitel 5.1 kann geschlossen werden, dass die Minima der Lösungslandschaft der zu MIS Problemen umgewandelten 3SAT Problemen sehr breit sind. Erst ab einem Annealing Faktor von ungefähr 0.5-0.6 kann gehofft werden, dass der Quantum Annealer möglicherweise nicht mehr in einem großen lokalen Minimum feststeckt, sondern das globale Minimum findet. Aufgrund des starken Unterschieds zwischen gelösten 3SAT-Instanzen und korrekten Zuordnungen (vgl. Interpretation 8 in Section 5.1) kann davon ausgegangen werden, dass die Tunnelwahrscheinlichkeit (vgl. Kapitel 4.7) gering und somit der Wall zwischen globalem Minimum und anderen Minima hoch ist. Da durch den *Postprocessing* Schritt nur die Ergebnisse der optimalen Lösung verbessert werden, kann man des Weiteren schließen, dass die lokalen Minima im Bereich des globalen Minimums korrekte IS Belegungen, aber keine maximalen präsentieren. Die großen Minima hingegen präsentieren hingegen IS Belegungen, die nicht korrekt sind, aber eine hohe Kardinalität besitzen. Dadurch kann es vorkommen, dass die IS-Eigenschaften zwischen der Klauseln erfüllt sind, und die IS-Belegung korrekt als 3SAT-Lösung interpretiert werden kann. (vgl. Section 5.1) Grafisch können die Erkenntnisse in zwei Dimensionen wie in Abbildung 7.1 dargestellt werden.

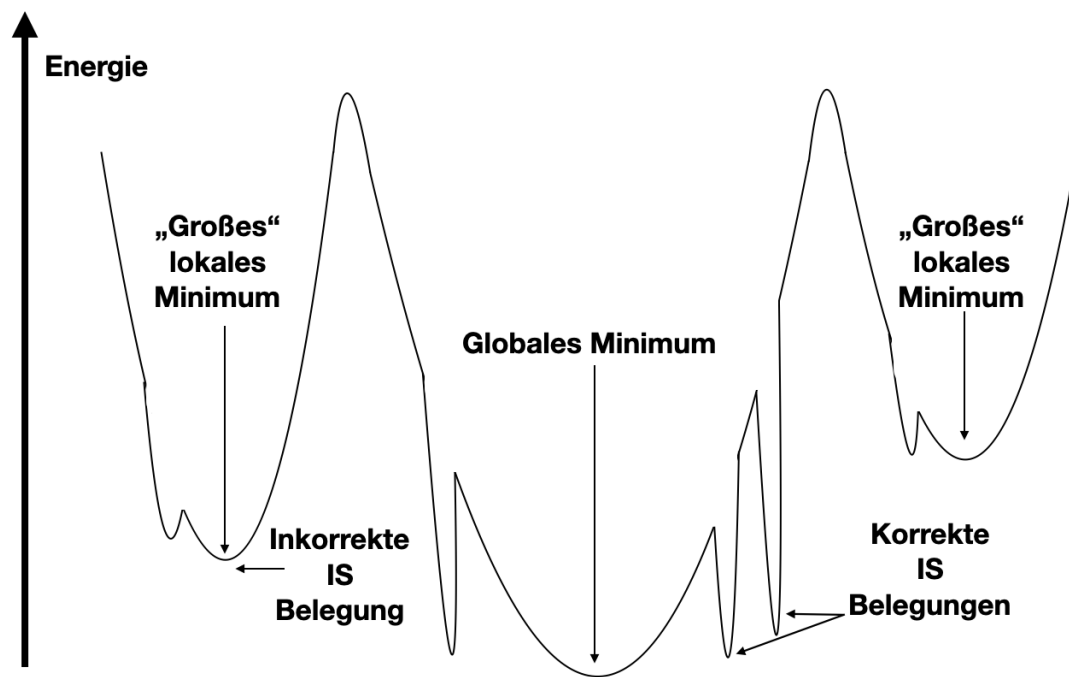


Abbildung 7.1.: Die Grafik präsentiert die Ergebnisse zur Lösungslandschaft des 3SAT-Problems in dem entwickelten Verfahren.

8. Fazit

Die Arbeit stellt dar, wie 3SAT-Probleme mithilfe quantum-klassisch hybrider Lösungsverfahren gelöst werden können. Mithilfe des MILP-Solvers *Gurobi* und der Reverse Annealing Funktion des Quantencomputer *Advantage 4.1* der Firma *D-Wave Systems* wurde eine Versuchsreihe durchgeführt. Die Auswertung über 2 Millionen Samples des Quantum Annealers lieferte interessante Ergebnisse und erlaubte einen Einblick in die Struktur des zum MIS umgeformten 3SAT-Problems. Dabei ragen vor allem die Eigenschaften der lokalen Minima der MIS-Formulierung hervor. Nahe des globalen Minimums sind dabei diese, die korrekte, aber nicht maximale IS-Zuordnungen darstellen. Entfernt, durch eine hohe Barriere getrennt vom globalen Minimum, lassen sich lokale Minima verorten, die inkorrekte IS-Zuordnungen repräsentieren, die jedoch durch eine hohe Kardinalität eventuell zu korrekten 3SAT-Lösungen umgewandelt werden können. Das Konzept zeigt Potenzial bei kleinen Instanzen. Dort können nur wenige Variablen fixiert werden, um einen signifikanten Effekt auf die Lösungsqualität des Reverse Annealing zu sehen. Jedoch verteilen sich die korrekten MIS-Belegungen vor allem bei kleinen Annealing Bruchteilen auf wenige 3SAT-Instanzen. Das heißt es müssten nicht nur mehrere Reverse Annealing Samples produziert werden, sondern auch mehrere Lösungen des MILP mit verschiedenen relaxierten Variablen um ein schweres 3SAT-Problem zu lösen. Das Verfahren kann auch nicht ohne Weiteres skaliert werden. Um gute Ergebnisse auch bei größeren 3SAT-Instanzen zu erreichen kann versucht werden, den Annealing Schedule zu verbessern und zu verlängern. Wird die Skalierung erreicht, so kann das jedoch zu einer schnellen Lösungszeit von einzelnen 3SAT-Instanzen führen, da bei Instanzen mit 300 Variablen 10% der Variablen des MILP fixiert werden können, um dieses durchschnittlich schneller zu lösen als ein moderner SAT-Solver die eigentliche 3SAT-Instanz. 10% fixierte Variablen reichen nach den Ergebnissen des Versuchs aus, um kleine Instanzen Instanzen mit einer kleinen Wahrscheinlichkeit lösen zu können. Der Versuch zeigte jedoch auch, dass die Abweichungen der Zeit und auch der Lösungswahrscheinlichkeit sehr hoch sind. Daher sind Schlussfolgerungen nur vorsichtig zu formulieren. Im Moment zeigen sich dem quantum-klassisch hybriden Lösungsverfahren für 3SAT-Probleme also vor allem Grenzen in der Skalierungsmöglichkeit auf.

A. Komplexitätstheorie

In diesem Kapitel werden Teile der Komplexitätstheorie beleuchtet. Es orientiert sich an [Hro14]. Die Komplexität eines Algorithmus kann dabei anhand der Änderung seiner Laufzeit pro weiteren Inputparameter bestimmt werden.

A.1. Das Entscheidungsproblem

1928 stellten David Hilbert and William Ackermann in ihrer Arbeit [WA28] ein Logikkalkül vor. Dabei stellten sie die Frage, ob es einen Algorithmus gibt, der ein logisches Statement verarbeitet und eine Ja/Nein Antwort zur universalen Korrektheit liefert.

A.2. Das Lambda Kalkül und die Turingmaschine

Alonzo Church veröffentlichte 1936 einen Beweis in seiner Arbeit [Chu36], der das Problem [A.1] als effizient unlösbar einstuft. Dabei verwendete er mit dem Lambda-Kalkül eine neue Methode zum Aufzeigen von Berechenbarkeit. 1937 zeigte Alan Turing in seinem Artikel [Tur37] dasselbe mithilfe der sogenannten Turing-Maschine. Das Lambda-Kalkül ([A.1]), als auch die Turing-Maschine ([A.2]) sind essentielle Konzepte der modernen Komplexitätstheorie und theoretischen Informatik. Durch beide Ansätze kann die Berechenbarkeit eines Algorithmus formal beschrieben werden. Vor 1936 konnte nur die Intuition verwendet werden, um zu beurteilen, ob ein Problemn effizient berechenbar ist.

Definition A.1 (Lambda-Kalkül) *Das Lambda-Kalkül ist eine Möglichkeit Funktionen und Algorithmen formal zu beschreiben. Dabei wird ein Lambda-Term rekursiv mit folgender Grammatik gebildet:*

$$t := C \mid x \mid t_1 t_2 \mid \lambda x. t$$

C präsentiert hier eine Konstante, x eine Variable, t_1 eine Funktion die auf das Argument t_2 angewendet wird und $\lambda x. t$ eine Abstraktion mit Parameter x und Term t . In letzterem ist die Variable x damit gebunden.

Definition A.2 (Turingmaschine) *Die Turingmaschine ist ein mathematisches Modell zur Repräsentation von Algorithmen. Die Maschine kann mit gewissen Regeln Zeichen auf ein Speicherband schreiben und manipulieren und davon lesen. Interpretiert man die Zeichen als Zahlen, so kann das Modell Funktionen nachbilden. Eine Turingmaschine wird beschrieben mit dem Tupel*

$$T = (Q, \Sigma, \Gamma, \delta, q_0, \square, F).$$

*Q beschreibt die endliche Zustandsmenge,
 Σ das endliche Eingabealphabet,*

A. Komplexitätstheorie

Γ das endliche Bandalphabet.

Es gilt $\Sigma \subset \Gamma$ und

$\delta: (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \{L, N, R\}$ ist die Überföhrungsfunktion.

$q_0 \in Q$ ist der Anfangszustand.

$\square \in \Gamma \setminus \Sigma$ steht für das leere Feld (Blank).

$F \subseteq Q$ beschreibt die Menge Endzustände.

Anmerkung A.3 Die Äquivalenz beider Methoden bewiesen in ... (hier noch Beweis suchen)

Mithilfe des formalen Rahmens der Turingmaschine konnte Cook 1971 in [Coo71] zeigen, dass das Entscheidungsproblem (SAT) NP-vollständig ist.

A.3. Komplexitätsklassen

Es folgt ein Überblick über verschiedene Komplexitätsklassen. Abbildung A.1 verdeutlicht deren Zusammenhang.

Komplexitätsklasse P

Alle sich in dieser Klasse befindenden Probleme sind in Polynomialzeit mit einer deterministischen Turingmaschine lösbar.

Beispiele: Linear Programming, 2SAT

Komplexitätsklasse NP

Die Lösungen der zu den Problemen in der Klasse NP müssen in Polynomialzeit verifizierbar sein. Damit gilt auch $P \subseteq NP$. Ob $P = NP$ gilt ist ein ungelöstes Problem der theoretischen Informatik. Es wird vermutet, dass $P \neq NP$. (siehe artikel)

NP-schwere Probleme

Diese Probleme sind mindestens so schwer wie alle anderen Probleme in NP.

NP-vollständige Probleme

Die Probleme sind NP-schwer und in NP selber enthalten.

Beispiele: Integer Linear Programming, 3SAT, Das Problem des Handelsreisenden, Rucksackproblem

A.4. Notation

Hier folgt eine kurze Sektion über die Notation der Komplexitätstheorie. Um die Komplexität eines Algorithmus zu beschreiben benutzt man die $\mathcal{O}$ -Notation, Θ -Notation und Ω -Notation. Die Konvention der Symbole folgt der von Donald Knuth in seinem Artikel [Knu76] beschriebenen.

Definition A.4 ($\mathcal{O}$ -Notation) $\mathcal{O}(f(x))$ beschreibt alle $g(x)$ für die zwei positive Konstanten C und x_0 existieren, mit denen gilt:

$$|g(x)| \leq C f(x) \text{ für alle } x \geq x_0$$

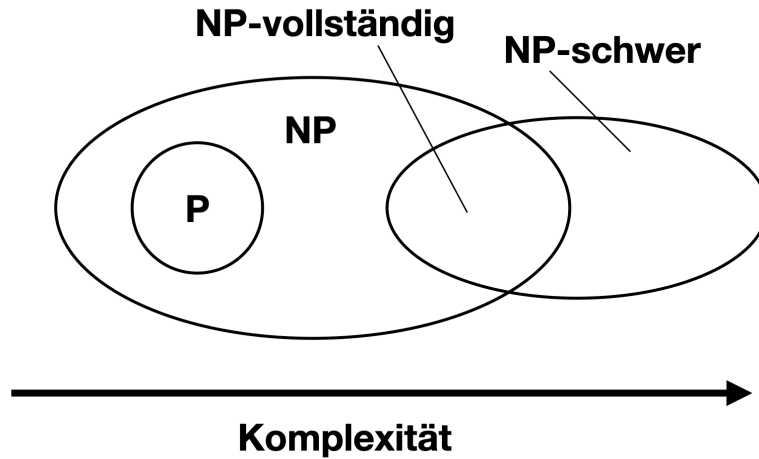


Abbildung A.1.: Die Komplexitätsklassen NP, NP-schwer, NP-vollständig und P werden im Vergleich dargestellt.

$g(x)$ wächst asymptotisch nicht schneller als $f(x)$.

Definition A.5 (Ω -Notation) $\Omega(f(x))$ beschreibt alle $g(x)$ für die zwei positive Konstanten C und x_0 existieren, mit denen gilt:

$$g(x) \geq C f(x) \text{ für alle } x \geq x_0$$

$g(x)$ wächst asymptotisch mindestens so schnell wie $f(x)$.

Definition A.6 (Θ -Notation) $\Theta(f(x))$ beschreibt alle $g(x)$ für die zwei positive Konstanten C_1 , C_2 und x_0 existieren, mit denen gilt:

$$C_1 f(x) \geq g(x) \geq C_2 f(x) \text{ für alle } x \geq x_0$$

$g(x)$ wächst asymptotisch so schnell wie $f(x)$.

B. Physikalische Grundlagen

Um die Funktionsweise eines Quantencomputers erklären zu können, sind grundlegende Kenntnisse der Physik notwendig. Die Lehre der Physik schafft einen mathematischen Rahmen zum Verständnis der Natur. Sie baut auf wenigen Axiomen auf und versucht, physikalische Phänomene damit erklären zu können. Ein solcher Rahmen ist die Newton'sche Mechanik. Wenn Teilchen genügend groß sind (und sich nicht mit zu schneller Geschwindigkeit bewegen), erklärt die Theorie Newtons die Natur hinreichend genau. Bei sehr kleinen Teilchen treten hingegen Effekte auf, die mit dieser nicht vereinbar sind. Hier wird die Quantenmechanik verwendet. Folgende Kapitel orientieren sich an [Gri05] und [SN17].

B.1. Beschreibung Quantenmechanik

Bei kleinen Teilchen kann die klassische Mechanik nicht mehr angewendet werden. Hier treten Effekte auf, die damit nicht erklärt werden können. Als Beispiel kann man das Doppelspaltexperiment nennen. Hierbei werden kleinste Teilchen auf eine Wand mit zwei schmalen, parallelen Spalten geschossen. Dahinter befindet sich in größerem Abstand ein Schirm, der den genauen Ort aufzeichnet, an der die Teilchen auftreffen. Man erkennt ein Interferenzmuster. Sobald die Teilchen aber dabei beobachtet werden, welchen Weg sie zurücklegen, treffen Sie wieder wie *normale* Teilchen auf den Schirm. Dies kann nicht mit der klassischen Newton'schen Physik erklärt werden. Die Quantenmechanik ist ein mathematischer Rahmen, der dieses Phänomen beschreiben kann. Dabei sind Teilchen nicht mehr Punkte im euklidischen Raum, sondern werden als dreidimensionale Wellen dargestellt. Damit können physikalische Experimente sehr gut beschrieben werden. Es gibt jedoch verschiedene Interpretationen der mathematischen Theorie. Die populärste und in dieser Arbeit durchgehend verwendete ist die Kopenhagener Deutung. Dabei drückt das Betragsquadrat der Wellenfunktion eines Teilchens die Wahrscheinlichkeit aus, es an einem bestimmten Ort finden zu können. (siehe Gleichung [B.7]) Diese Standardinterpretation führt die intrinsisch statistische Ortsbestimmung auf einen fundamentalen Indeterminismus zurück. Das heißt, dass die Natur den Ort des Teilchens erst bei Messung mit irreduziblem Zufall bestimmt. Da dies mit der menschlichen Intuition nicht wirklich fassbar ist, sagte Richard Feynman 1964 an der Cornell University:

I think I can safely say that nobody understands quantum physics

Richard Feynman war der Meinung, dass niemand Quantenphysik verstehen würde. ([Fey64]) Ein Jahr später erhielt er den Nobelpreis für Physik für seine Arbeit in der Quantenphysik.

B.2. Dirac Notation und Operatoren

Im weiteren Verlauf der Arbeit wird die von Paul Dirac eingeführte Bra-Ket Notation verwendet. Dabei wird ein Vektor v im Vektorraum V - auch Ket-Raum genannt - mit einem Ket $|v\rangle$ ausgedrückt. Quantenmechanische Zustände werden mit komplexen Zustandsvektoren ausgedrückt und befinden sich in einem Hilbertraum. Ein Hilbertraum ist ein vollständig normierter Raum, der zulässig für Skalarprodukte ist. Zwei Kets können addiert und mit Konstanten multipliziert werden:

$$|a\rangle + |b\rangle = |c\rangle$$

$$c|a\rangle = |a\rangle c = |d\rangle$$

Ein hermitescher Operator wird auch Observable genannt. Er wirkt von links auf einen Ket:

$$A|a\rangle = |e\rangle$$

Ein Eigenket $|e\rangle$ der Observable A - oder in physikalischen Systemen Eigenzustand genannt - mit einem Eigenwert α kann beschrieben werden mit:

$$A|e\rangle = \alpha|e\rangle$$

Der Kovektor w aus dem Dualraum V^* - oder Bra-Raum - wird geschrieben als Bra $\langle w|$. Zwischen Ket- und Bra-Raum existiert eine duale Korrespondenz:

$$|a\rangle \Longleftrightarrow dual \langle a|$$

$$|a\rangle + |b\rangle \Longleftrightarrow dual \langle a| + \langle b|$$

$$c|a\rangle \Longleftrightarrow dual c^* \langle a|$$

Der hochgestellte Stern bedeutet das komplex Konjugierte der Variable. Das innere Produkt eines Bras und eines Kets wird geschrieben als:

$$\langle w|v\rangle = \langle v|w\rangle^*$$

Der Vektorraum ist daher semidefinit:

$$\langle a|a\rangle \geq 0 \tag{B.1}$$

Wenn $\langle a|a\rangle = 0$ gilt ist $|a\rangle$ ein Null-Ket Zwei Vektoren $|a\rangle$ und $|b\rangle$ sind orthogonal wenn gilt:

$$\langle a|b\rangle = 0 \implies \langle b|a\rangle = 0 \tag{B.2}$$

Daher kann ein Ket $|a\rangle$ - der kein Null-Ket ist - normalisiert werden mit:

$$|\bar{a}\rangle = \frac{1}{\sqrt{\langle a|a\rangle}} |a\rangle \tag{B.3}$$

$$\tag{B.4}$$

Dabei ist $\langle \bar{a} | \bar{a} \rangle = 1$. $\sqrt{\langle a | a \rangle}$ wird die Norm von $|a\rangle$ genannt. Da $c|\bar{a}\rangle$ und $|a\rangle$ dieselben physikalischen Zustände darstellen werden die Kets dieser Zustände immer normiert sein. Für lineare Operatoren, die auf Kets wirken - nicht nur hermitesche Operatoren (oder Observablen), gelten folgende Rechenregeln.

$$A(\alpha |a\rangle + \beta |b\rangle) = \alpha A |a\rangle + \beta A |b\rangle \quad (\text{B.5})$$

Ein Operator B wirkt auf einen Bra immer von rechts.

$$\langle a | B = \langle b |$$

Die duale Korrespondenz von Operatoren ist:

$$\langle a | B \iff B^\dagger |a\rangle$$

$B^\dagger$ drückt das Adjungierte des Operators B aus. Operatoren können multipliziert werden. Im Allgemeinen kommutieren sie aber nicht,

$$XY \neq YX$$

sind jedoch assoziativ:

$$(XY)Z |a\rangle = XY(Z |a\rangle) = XYZ |a\rangle$$

Wegen

$$XY |a\rangle = X(Y |a\rangle) \iff (\langle a | Y^\dagger) X^\dagger = \langle a | X^\dagger Y^\dagger$$

gilt

$$(XY)^\dagger = X^\dagger Y^\dagger$$

Das äußere Produkt von $|a\rangle$ und $\langle b|$ wird geschrieben als

$$\langle b | a \rangle$$

Es gilt mit $\langle b | c \rangle = c_0$

$$(|a\rangle \langle b|) |c\rangle = |a\rangle (\langle b | c \rangle) = c_0 |a\rangle$$

Wenn also ein äußeres Produkt auf einen Ket wirkt entsteht ein weiterer Ket. Daher kann $|a\rangle \langle b|$ als ein Operator aufgefasst werden. Der Vektor $|c\rangle$ auf den das Produkt angewendet wird, wird in die Richtung von $|a\rangle$ rotiert und erhält die Länge der Projektion auf $\langle b|$.

Das äußere Produkt

$$M(a) = |a\rangle \langle a|$$

nennt man auch Projektionsoperator $M(a)$ entlang des Basiszustandes $|a\rangle$. Wird ein beliebiger Zustand darauf angewendet, so projiziert der Operator diesen auf $|a\rangle$

Ein Operator O , der zwischen einem Bra und einem Ket steht, kann wegen der Assoziativität auf beide Seiten angewandt werden:

$$\langle a | (O |b\rangle) = (\langle a | O) |b\rangle = \langle a | O |b\rangle \langle a | O |b\rangle = (\langle b | O^\dagger |a\rangle)^*$$

Für hermitesche Operatoren gilt $O^\dagger = O$ und daher:

$$\langle a | O | b \rangle = (\langle b | O | a \rangle)^*$$

Theorem B.1 *Hermitesche Operatoren besitzen ausschließlich reelle Eigenwerte und orthogonale Eigenzustände.*

Beweis: Ein hermitescher Operator H besitzt zwei Eigenwerte α, β , und die dazugehörigen Eigenzustände a, b . Es gilt mit $H = H^\dagger$, dass $\alpha \langle b | a \rangle = \langle b | (H | a \rangle) = (\langle b | H) | a \rangle = \beta^ \langle b | a \rangle$ und damit $(\alpha - \beta^*) \langle b | a \rangle = 0$ ist. Wenn $a = b$ gilt $a = a^*$ und damit, dass $a \in \mathbb{R}$. Ist $a \neq b$ gilt $\langle b | a \rangle = 0$. Daher müssen alle Eigenzustände, die nicht denselben Eigenwert besitzen, orthogonal zueinander stehen.*

Um hermitesche Operatoren physikalisch als Observablen auffassen zu können, muss der observierte Raum mit den Eigenfunktionen der Operatoren aufgespannt werden können.

Axiom B.2 *Die Eigenzustände einer Observablen müssen eine lineare Hülle über dem observierten Hilbertraum bilden.*

B.3. Messung eines quantenmechanischen Systems

Eine Observable O besitzt die Eigenzustände $|\omega_i\rangle$ und Eigenwerte o_i und kann als Linearkombination derer geschrieben werden: $O = \sum_{\omega_i} o_i |\omega_i\rangle$. Der physikalische Zustand eines Systems $|\Psi\rangle$ kann mit dessen Eigenzuständen $|\alpha_i\rangle$ und Eigenwerten a_i beschrieben werden: $|\Psi\rangle = \sum_i a_i |\alpha_i\rangle$. Da die Eigenzustände von O den gesamten Hilbertraum in dem $|\Psi\rangle$ liegt aufspannen, kann der Zustand des Systems $|\Psi\rangle$ auch beschrieben werden durch:

$$|\Psi\rangle = \sum_i c_i |\omega_i\rangle \quad \text{mit } c_i = \langle \omega_i | \Psi \rangle$$

Die statistische Interpretation der Quantenmechanik besagt nun, dass $|c_i|^2$ die Wahrscheinlichkeit darstellt, den Eigenzustand $|\omega_i\rangle$ bei der Messung von $|\Psi\rangle$ mit dem Operator O zu erhalten. Die Eigenschaft von Wahrscheinlichkeiten

$$P(\Omega) = \sum_i |c_i|^2 = 1 \tag{B.6}$$

kann aufgrund der Normierung von $|\Psi\rangle$ bestätigt werden:

$$\begin{aligned} 1 = \langle \Psi | \Psi \rangle &= \left\langle \sum_{i'} c_{i'} \omega_{i'} \middle| \sum_i c_i \omega_i \right\rangle = \sum_{i'} \sum_i c_{i'}^* c_i \langle \omega_{i'} | \omega_i \rangle \\ &= \sum_{i'} \sum_i c_{i'}^* c_i \delta_{i'i} = \sum_i c_i^* c_i \\ 1 &= \sum_i |c_i|^2 \end{aligned}$$

Bei der Messung des Systems kollabiert der Zustand in einen Basiszustand. Wird das System direkt danach mit demselben Operator noch einmal gemessen, so wird der gleiche

Basiszustand als Zustand des Systems gemessen. Für eine genauere Beschreibung des Messvorgangs wird auf die Bücher [SN17] und [Gri05] verwiesen.

B.4. Mathematische Herleitung der Schrödingergleichung

Um ein besseres Bild davon zu bekommen, was in realen physischen Systemen passiert, wird in diesem Kapitel die Schrödingergleichung hergeleitet. Diese ist die Bewegungsgleichung für quantenmechanische Systeme. Sie ähnelt der der klassischen Mechanik, jedoch wird nun nicht ein Ort $x(t)$ des Teilchens bestimmt, sondern die Wellenfunktion $|\Psi(x, t)\rangle$ eines Teilchens. Die Zustände des Systems und Eigenzustände der Observablen sind nun nicht mehr diskret, wie in vorherigen Kapiteln angenommen, sondern kontinuierlich. Das Betragsquadrat der gefundenen komplexen Wellenfunktion gibt die Wahrscheinlichkeit an, das Teilchen am Ort x oder einem sehr kleinen Abstand dx zu finden:

$$||\Psi_t(x)\rangle|^2 dx \quad (\text{B.7})$$

Da sich das Teilchen im Raum befinden muss, gilt $P(\Omega) = 1$ und somit auch für alle t :

$$P_{-\infty}^{\infty}(t) = \int_{-\infty}^{\infty} ||\Psi_t(x)\rangle|^2 dx = 1 \quad (\text{B.8})$$

Verändert sich die initiale Wellenfunktion Ψ_0 mit der Zeit mit $|\Psi_t\rangle = U_t |\Psi_0\rangle$, so muss gelten:

$$P_{-\infty}^{\infty}(t) = \int_{-\infty}^{\infty} \langle \Psi_0(x) | U_t^\dagger U_t | \Psi_0(x) \rangle dx = 1$$

und damit auch $U_t^\dagger U_t = I$. U_t muss also ein unitärer Operator sein und kann immer geschrieben werden als $U_t = e^{\frac{itH}{\hbar}}$ mit H als ein hermischem Operator. $\hbar$ drückt hierbei eine physikalische Konstante aus. * Hermitische Operatoren im quantenmechanischen Sinne können als Messungen interpretiert werden. Sie besitzen ausschließlich reelle Werte auf ihrer Diagonalen und somit auch für die quantenmechanische Erwartungswert des Operators eines Systems im Zustand $\Psi(x, t)$. (siehe Gleichung...) Die Schrödingergleichung, mit deren Hilfe die Wellenfunktionen von Teilchen berechnet werden, beschreibt denselben Zusammenhang:

$$i \frac{\partial}{\partial t} \Psi(x, t) = H \Psi(x, t) \quad (\text{B.9})$$

Da $\Psi(x, t)$ eine Wellenfunktion darstellt, kann sie sowohl mit als auch ohne der Notation Diracs beschrieben werden. Im weiteren Verlauf dieses Kapitels werden wir der Einfachheit halber auf die Notation verzichten. H stellt den Hamiltonoperator (oder Energieoperator) des Systems und somit dessen Energie dar. Um den Operator zu beschreiben, wird zuerst die Hamiltonfunktion der klassischen Mechanik betrachtet:

$$H = \frac{p^2}{2m} + V$$

B. Physikalische Grundlagen

p drückt den Impuls aus und V stellt die von der Zeit unabhängige potenzielle Energie des Teilchens dar. Um nun den Energieoperator zu erhalten wird der kanonisch transformierte quantenmechanische Impulsoperator eingesetzt. Im Ortsraum ist er definiert als

$$p = -i\hbar \frac{\partial^2}{\partial x^2}$$

Der Hamiltonoperator kann im Ortsraum beschrieben werden als

$$H = -\frac{p^2}{2m} + V = -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V$$

H wirkt auf einen Zustand Ψ im Hilbertraum $\mathcal{H}$. Gleichung 6.6 lässt sich damit schreiben als

$$i\hbar \frac{\partial}{\partial t} \Psi(x, t) = -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} \Psi(x, t) + V \Psi(x, t)$$

Um die Wellenfunktion eines Teilchens zur Zeit t zu berechnen löst man die Differentialgleichung mithilfe einer **Variablenseparation**. Man nimmt an, dass sich die Wellenfunktion $\Psi(x, t)$ als ein Produkt von einer zeitabhängigen und einer zeitunabhängigen Funktion $\psi(x)\phi(t)$ schreiben lässt. Mit

$$\begin{aligned} \frac{\partial \psi(x)\phi(t)}{\partial t} &= \psi(x) \frac{d\phi(t)}{dt} \\ \frac{\partial^2 \psi(x)\phi(t)}{\partial x^2} &= \phi(t) \frac{d^2 \psi(x)}{dx^2} \end{aligned}$$

und

$$\begin{aligned} i\hbar \frac{\partial \psi(x)\phi(t)}{\partial t} &= -\frac{\hbar^2}{2m} \frac{\partial^2 \psi(x)\phi(t)}{\partial x^2} + V \psi(x)\phi(t) \\ i\hbar \psi(x) \frac{d\phi(t)}{dt} &= -\frac{\hbar^2}{2m} \phi(t) \frac{d^2 \psi(x)}{dx^2} + V \psi(x)\phi(t) \end{aligned}$$

kann die Schrödingergleichung dann umgeschrieben werden zu

$$i\hbar \frac{1}{\phi(t)} \frac{d\phi(t)}{dt} = -\frac{\hbar^2}{2m} \frac{1}{\psi(x)} \frac{d^2 \psi(x)}{dx^2} + V \quad (\text{B.10})$$

Nun ist der linke Term ausschließlich abhängig von der Zeit t . Der rechte Teil ändert sich dabei nur mit dem Ort x . Beide müssen demnach konstant sein um die Gleichung zu erfüllen. Die daraus resultierende Separationskonstante wird E benannt. Es gilt für den zeitabhängigen Term

$$i\hbar \frac{1}{\phi(t)} \frac{d\phi(t)}{dt} = E$$

und für den ortsabhängigen Term

$$-\frac{\hbar^2}{2m} \frac{1}{\psi(x)} \frac{d^2 \psi(x)}{dx^2} + V = E$$

folgt die stationäre Schrödingergleichung

$$-\frac{\hbar^2}{2m} \frac{d^2\psi(x)}{dx^2} + V\psi(x) = E\psi(x) \quad (\text{B.11})$$

oder mit Verwendung des Hamiltonoperators

$$E\psi(x) = H\psi(x) \quad (\text{B.12})$$

Die Separationskonstante E bezeichnet den Erwartungswert der totalen Energie des Systems. Der zum kleinsten Erwartungswert E_0 von H gehörende Zustand ψ^0 wird Grundzustand genannt.

C. Beispiel zur Anwendung des Simplexverfahrens

Um den Simplex Algorithmus besser verstehen zu können, wird hier ein Beispiel zur Lösung eines Problems dargestellt. Es ist angelehnt an ein Beispiel aus [IpE]. Man nehme an, eine Fabrik stellt Tische her und hat 3 Maschinen die diese produzieren. Ein Glastisch kann für 20 Euro verkauft werden, ein Holztisch für 30 Euro.

Maschine 1

benötigt für die Herstellung vom Holztisch 2 Minuten und dem Glastisch 1 Minute.
Sie kann maximal 100 Minuten in Betrieb sein.

Maschine 2

benötigt für die Herstellung vom Holztisch 1 Minuten und dem Glastisch 1 Minute.
Sie kann maximal 60 Minuten in Betrieb sein.

Maschine 3

benötigt für die Herstellung vom Holztisch 1 Minuten und kann keinen Glastisch produzieren. Sie kann maximal 40 Minuten in Betrieb sein.

Zur Vorbereitung des Simplexalgorithmus werden einige Schritte angewandt:

1. Modellierung des Problems mithilfe eines linearen Programms

$$\begin{aligned} \max f = & 30x_1 + 20x_2 \\ \text{s.t.} \quad & 2x_1 + 1x_2 \leq 100 \\ & 1x_1 + 1x_2 \leq 60 \\ & 1x_1 \leq 40 \\ & x_1, x_2 \geq 0 \end{aligned} \tag{C.1}$$

2. Transformation zum linearen Programm in Standardform

$$\begin{aligned}
\min f = & -30x_1 - 20x_2 \\
\text{s.t.} \quad & 2x_1 + 1x_2 + x_3 = 100 \\
& 1x_1 + 1x_2 + x_4 = 60 \\
& 1x_1 + x_5 = 60 \\
& x_i \geq 0 \\
& \text{mit } i \in \{1, 2, 3, 4, 5\}
\end{aligned} \tag{C.2}$$

3. Aufstellen des Simplextableaus

x_1	x_2	x_3	x_4	x_5	RHS
1			2	1	100
	1		1	1	70
		1	1		40
			-30	-20	

Tabelle C.1.: Das Simplextableau des Beispiels.

Nun kann das Simplexverfahren (vgl. [Sektion 3.2](#)) angewandt werden. Hierbei wiederholen sich die drei Schritte bis das Problem als unlösbar befunden wird oder eine optimale Lösung gefunden wird.

1. Iteration

- Auswahl einer Pivotspalte j
Beide Nichtbasisvariablen besitzen negative reduzierte Preise. Den kleinsten besitzt x_4 , daher wird die 4. Spalte ausgewählt.
- Ermittlung der Pivotreihe i
Die Verhältnisse $V_i = \frac{b_i}{a_{i4}}$ werden ermittelt. Aus der Menge $V = \{50, 70, 40\}$ wird nun das kleinste Element ausgewählt. Also ist $i = 3$.
- Zeilenumformungen zum Verändern der Basis
Das Pivotelement ist a_{34} . Die Tabelle wird umgewandelt, um die Variable der Pivotspalte in die Basis zu bringen. Dazu wird der Wert des Pivotelement zu eins transformiert. Dann werden alle anderen Elemente der Pivotspalte auf null gebracht. Die resultierende Tabelle ist:

2. Iteration

- Auswahl der Pivotspalte $j = 5$
- Ermittlung der Pivotreihe $i = 1$
- Zeilenumformungen zum Verändern der Basis

x_1	x_2	x_3	x_4	x_5	RHS
1		-2		1	20
	1	-1		1	30
		1	1		40
		30		-20	1200

Tabelle C.2.: Tabelle der 1. Iteration.

Die resultierende Tabelle ist:

x_1	x_2	x_3	x_4	x_5	RHS
1		-2		1	20
-1	1	1			10
		1	1		40
20		-10			1600

Tabelle C.3.: Tabelle der 2. Iteration.

3. Iteration

- Auswahl der Pivotspalte $j = 3$
- Ermittlung der Pivotreihe $i = 3$
- Zeilenumformungen zum Verändern der Basis

Die resultierende Tabelle ist:

x_1	x_2	x_3	x_4	x_5	RHS
-1	2			1	40
-1	1	1			10
1	-1		1		30
10	10				1700

Tabelle C.4.: Tabelle der 3. Iteration.

Es gibt keine negativen reduzierten Preise, daher ist die Lösung optimal. Damit ist die optimale Basislösung $x_1 = 0$ $x_2 = 0$ $x_3 = 10$ $x_4 = 30$ $x_5 = 40$. Die Firma maximiert also ihren Gewinn indem Sie Maschine 1 40 Minuten, Maschine 2 10 Minuten und Maschine 3 30 Minuten lang betreibt.

Abbildungsverzeichnis

2.1. Die maximale stabile Menge des Graphen ist $M = \{1, 2, 6, 7\}$. Die Stabilitätszahl des Graphen ist 4.	4
2.2. Hier wird die Formel 2.2 in einen MIS-Graphen umgewandelt.	6
2.3. Eine maximale stabile Menge des Graphen bilden die grün eingefärbten Knoten. Die Zuordnung kann direkt mit umgewandelt werden in Lösung L der Formel 2.2: $L = \{x_1 \leftarrow 0; x_2 \leftarrow 1; x_3 \leftarrow 0; x_5 \leftarrow 0; x_6 \leftarrow 1\}$	6
2.4. Eine stabile Menge des Graphen bilden die grün eingefärbten Knoten. Durch die Interpretation wird auch der orange eingefärbte Knoten bei der Umwandlung in eine 3SAT Belegung als <i>wahr</i> betrachtet.	7
2.5. Eine stabile Menge des Graphen bilden die grün eingefärbten Knoten. Durch das <i>Postprocessing</i> wird auch der orange eingefärbte Knoten bei der Umwandlung in eine 3SAT Belegung als <i>wahr</i> betrachtet.	7
2.6. Eine gefundene Lösung $L = \{x_1 \leftarrow 1; x_2 \leftarrow 1; x_3 \leftarrow 0; x_5 \leftarrow 0; x_6 \leftarrow 0\}$ wird in eine MIS-Lösung umgewandelt. Da nun in der ersten und dritten Klausel zwei Literale den Wert 1 besitzen wird jeweils ein dazugehöriger Knoten aus der stabilen Menge entfernt. (hier orange eingefärbt)	7
2.7. Da drei Stellen in den beiden binären Zeichenketten unterschiedlich sind, beträgt die Hamming-Distanz 3.	9
3.1. Eine graphische Darstellung des linearen Programms 3.3. Die Hyperebene $H = \{x \in \mathbb{R}^n c^T x\}$ (hier rot gekennzeichnet) wird entlang ihres Normalenvektors c maximal verschoben, bis sie das Polyeder des linearen Programms nur mehr noch berührt.	13
3.2. Die Lösung des LP 3.3 sind dann die Berührungspunkte der verschobenen Zielfunktion und des Polyeders, also $x_1 = 0$ und $x_2 = 3.25$	13
3.3. Graphische Repräsentation des linearen Programmes 3.4.	19
3.4. Der Algorithmus teilt die Lösungsmenge der LP-Relaxierung bei jeder Verzweigung in drei Teile. Dies geschieht durch zwei Hyperebene die senkrecht zur Achse der ausgewählten Variable verlaufen. Die äußeren zwei Räume bilden die jeweiligen Lösungsräume der zwei neuen Knoten. Hier wird das Beispiel 3.4 dargestellt.	20
3.5. Ein optimales Schnittebenenverfahren würde nach dem ersten Durchgang des <i>Branch and Bound</i> Algorithmus (siehe Abbildung 3.4) die zulässige Lösungsmenge der LP-Relaxierung des Knotens mit $x_1 \geq 2$ (grau eingefärbt) so beschränken, dass die Lösungsmenge der LP-Relaxierung der konvexen Hülle der Lösungsmenge des MILPs (in orange) des Knotens entspricht. .	21

4.1. Graphische Repräsentation eines <i>Minor Embedding</i> . Jeder kann in diesem Beispiel maximal 2 Verbindungen besitzen. Der Knoten in der oberen Ecke wird deshalb auf zwei Knoten verteilt. Die blaue Kante zwischen den neuen blauen Knoten besitzt ein stark negatives Gewicht. (inspiriert von [dwa21])	27
4.2. Obwohl die Aufenthaltswahrscheinlichkeit (hier mit der Intensität des blauen Punktes gekennzeichnet) des Teilchens zu dem betrachteten Zeitpunkt in dem lokalen Minimum links höher ist, kann das Teilchen auch in dem globalen Minimum rechts gemessen werden. Daher kann ein quantenmechanisches System aufgrund der physikalischen Eigenschaften durch eine Potentialbarriere <i>tunneln</i> . (Inspiration von [dwa21])	28
4.3. Ein Annealing Schedule wird dargestellt bei dem der Annealing Bruchteil s innerhalb $4\mu s$ auf $s = f = 0.5$ absteigt. Dann findet eine Pause von $22\mu s$ statt, nach der s innerhalb von $4\mu s$ wieder auf $s = 1$ steigt. (Grafik inspiriert von [dwa21])	29
6.1. Es werden die Anzahl korrekter MIS-Zuordnungen der Samples des Reverse Annealing angezeigt. Der schwarze Balken stellt die Standardabweichung dar.	39
6.2. Die Grafik zeigt die durchschnittlichen Bruchteile korrekter 3SAT Zuordnungen (ohne <i>Postprocessing</i> Schritt) der Samples des Reverse Annealing mit verschiedenen Startzuständen.	40
6.3. Die Ergebnisse von Annealing Bruchteilen 0.1 bis 0.6 werden dargestellt. Dabei wird nach dem Reverse Annealing ein <i>Postprocessing</i> Schritt zur Umwandlung in eine 3SAT Belegung verwendet.	40
6.4. In der Grafik erkennt man die Ergebnisse des Reverse Annealing ohne Interpretation als 3SAT Belegung.	41
6.5. Es werden die Ergebnisse des Reverse Annealing der in MIS umgeformte 3SAT Instanzen mit Annealing Bruchteilen 0.51 bis 0.59 dargestellt. Bei der Umwandlung wurde die MIS Belegung zwar als 3SAT Zuordnung interpretiert, jedoch kein <i>Postprocessing</i> verwendet.	41
6.6. Ähnlich wie Abbildung 6.3 stellt auch diese Grafik die Ergebnisse der Umwandlung der Lösungen des Reverse Annealing mit einem <i>Postprocessing</i> Schritt dar. Hier werden jedoch die Annealing Bruchteile 0.51 bis 0.59 betrachtet. Die 3SAT Instanz besitzt 15 Variablen und 61 Klauseln.	42
6.7. In diesem Diagramm werden die durchschnittlichen Hamming- Distanzen der Samples des Reverse Annealing zur nächsten optimalen MIS Belegung dargestellt. Die schwarzen Striche bezeichnen die Standardabweichung.	42
6.8. Wie Abbildung 6.7 zeigt diese Grafik die durchschnittliche Hamming-Distanz an. Diesmal jedoch mit anderen Annealing Bruchteilen. (0.51 bis 0.59)	43
6.9. Die Anzahl gelöster MIS Instanzen mithilfe des Reverse Annealing werden angezeigt.	43
6.10. Das Diagramm spiegelt wieder, wieviele 3SAT-Probleme gelöst wurden mit der reinen Interpretation der MIS-Belegung.	44
6.11. Die Abbildung zeigt auf, welcher Prozentsatz der 3SAT Instanzen gelöst wurde.	44

6.12. Wie Abbildung 6.1 oder 6.4 zeigt dieses Diagramm die Ergebnisse der korrekten MIS Belegungen nach Reverse Annealing. Die Testinstanzen besitzen diese Mal jedoch 40 Variablen und 168 Klauseln.	47
6.13. Wie die Grafiken 6.2 oder 6.5 zeigt diese Abbildung die korrekten 3SAT Zuordnungen ohne <i>Postprocessing</i> Schritt. Dieses Mal werden 3SAT Probleme mit 40 Variablen und 168 Klauseln verwendet.	47
6.14. Es werden die durchschnittlichen Bruchteile korrekter 3SAT Zuordnungen nahe einem Reverse Annealing und <i>Postprocessing</i> aufgezeigt.	48
6.15. Wie Grafik 6.7 zeigt auch diese Abbildung die durchschnittliche Entfernung der MIS Belegung nach dem Reverse Annealing zur nächsten optimalen MIS Zuordnung. Dieses Mal wird jedoch 3SAT Instanzen mit 40 Variablen und 168 Klauseln verwendet.	48
6.16. Die Grafik vergleicht die benötigte durchschnittliche Lösungszeit verschiedener zu MILPs umgewandelter 3SAT Instanzen mit der des Minisat Solvers.	49
6.17. Wie Diagramm 6.16 beschreibt auch diese Abbildung den Vergleich zwischen den durchschnittlichen Lösungszeiten verschiedener MILPs mithilfe dem <i>Branch and Cut</i> Solver Gurobi und denen des SAT-Solvers Minisat.	50
6.18. Es wird der Vergleich der Lösungszeit des <i>Branch and Cut</i> Solvers Gurobi und dem SAT-Solver Minisat aufgezeigt.	51
7.1. Die Grafik präsentiert die Ergebnisse zur Lösungslandschaft des 3SAT-Problems in dem entwickelten Verfahren.	54
A.1. Die Komplexitätsklassen NP, NP-schwer, NP-vollständig und P werden im Vergleich dargestellt.	59

Tabellenverzeichnis

3.1. RHS bedeutet <i>right-hand side</i> (aus dem Englischen für rechte Seite). Da $m < n$ gilt, ist dieses Gleichungssystem unterbestimmt. Das heißt es gibt entweder unendlich viele Lösungen oder keine Lösung.	14
3.2. Das Beispiel 3.3 kann in einer Tabelle zusammengefasst werden.	14
3.3. Tabelle 3.1 wird hier nach Beenden des Gauß'schen Eliminationsverfahren dargestellt.	15
3.4. Die Tabelle 3.3 wird um eine Zeile für den Kostenvektor und einer Spalte mit dem Zielwert erweitert. Hierbei kann die Spalte f entfernt werden, da sie sich während des Simplexalgorithmus nicht ändert.	15
3.5. A beschreibt hierbei die anfängliche Koeffizientenmatrix, c den Kostenvektor und b die rechte Seite der Gleichungen.	16
3.6. B stellt in dieser Tabelle die Basis-Indexmenge und N die Nichtbasis-Indexmenge dar. A beinhaltet alle Spaltenindizes der Koeffizientenmatrix. z beschreibt die reduzierten Kosten.	16
3.7. Die Abbildung stellt die neue überarbeitete Simplextabelle dar.	16
3.8. Für die Ausarbeitung des Algorithmus ist es sinnvoll noch einige weitere Relationen der Variablen der beiden Tabellen darzustellen.	17
C.1. Das Simplextableau des Beispiels.	70
C.2. Tabelle der 1. Iteration.	71
C.3. Tabelle der 2. Iteration.	71
C.4. Tabelle der 3. Iteration.	71

Literaturverzeichnis

- [AL18] ALBASH, TAMEEM und DANIEL A. LIDAR: *Adiabatic quantum computation*. REVIEWS OF MODERN PHYSICS, 90(1):015002, jan 2018.
- [Bal79] BALAS, EGON: *Disjunctive Programming*. In: *Annals of Discrete Mathematics*, Seiten 3–51. Elsevier, 1979.
- [BAS⁺13] BOIXO, SERGIO, TAMEEM ALBASH, FEDERICO M. SPEDALIERI, NICHOLAS CHANCELLOR und DANIEL A. LIDAR: *Experimental signature of programmable quantum annealing*. Nature Communications, 4(1), jun 2013.
- [BP03] BALAS, EGON und MICHAEL PERREGAARD: *A precise correspondence between lift-and-project cuts, simple disjunctive cuts, and mixed integer gomory cuts for 0-1 programming*. Mathematical Programming, 94(2-3):221–245, jan 2003.
- [Cho10] CHOI, VICKY: *Different Adiabatic Quantum Optimization Algorithms for the NP-Complete Exact Cover and 3SAT Problems*. Quantum Information and Computation, Vol. 11, No. 7&8 (2011) 0638-0648, Oktober 2010.
- [Chu36] CHURCH, ALONZO: *An Unsolvable Problem of Elementary Number Theory*. American Journal of Mathematics, 58(2):345, apr 1936.
- [Cip87] CIPRA, BARRY A.: *An Introduction to the Ising Model*. The American Mathematical Monthly, 94(10):937–959, dec 1987.
- [Coo71] COOK, STEPHEN A.: *The complexity of theorem-proving procedures*. In: *Proceedings of the third annual ACM symposium on Theory of computing - STOC '71*. ACM Press, 1971.
- [Dan83] DANTZIG, G. B.: *Reminiscences About the Origins of Linear Programming*. In: *Mathematical Programming The State of the Art*, Seiten 78–86. Springer Berlin Heidelberg, 1983.
- [DBI⁺15] DENCHEV, VASIL S., SERGIO BOIXO, SERGEI V. ISAKOV, NAN DING, RYAN BABBUSH, VADIM SMELYANSKIY, JOHN MARTINIS und HARTMUT NEVEN: *What is the Computational Value of Finite Range Tunneling?* Phys. Rev. X 6, 031015 (2016), Dezember 2015.
- [DL12] DEGEN, HORST und PETER LORSCHIED: *Statistik-Lehrbuch*. Oldenbourg Wissenschaftsverlag, 2012.
- [DOW55] DANTZIG, GEORGE, ALEXANDER ORDEN und PHILIP WOLFE: *The generalized simplex method for minimizing a linear form under linear inequality restraints*. Pacific Journal of Mathematics, 5(2):183–195, jun 1955.
- [dwa21] *Documentation of D-Wave products*, November 2021.

- [EOP⁺21] EVANS, RICHARD, MICHAEL O’NEILL, ALEXANDER PRITZEL, NATASHA ANTROPOVA, ANDREW SENIOR, TIM GREEN, AUGUSTIN ŽÍDEK, RUSS BATES, SAM BLACKWELL, JASON YIM, OLAF RONNEBERGER, SEBASTIAN BODENSTEIN, MICHAL ZIELINSKI, ALEX BRIDGLAND, ANNA POTAPENKO, ANDREW COWIE, KATHRYN TUNYASUVUNAKOOL, RISHUB JAIN, ELLEN CLANCY, PUSHMEET KOHLI, JOHN JUMPER und DEMIS HASSABIS: *Protein complex prediction with AlphaFold-Multimer*. no journal, oct 2021.
- [ESB20] ERNST, HARTMUT, JOCHEN SCHMIDT und GERD BENEKEN: *Grundkurs Informatik*. Springer-Verlag GmbH, Juli 2020.
- [Fey64] FEYNMAN, RICHARD: *Speech on Quantum Physics*, 1964.
- [Fey85] FEYNMAN, RICHARD P.: *Quantum Mechanical Computers*. Optics News, 11(2):11, feb 1985.
- [FGGS00] FARHI, EDWARD, JEFFREY GOLDSTONE, SAM GUTMANN und MICHAEL SIPSER: *Quantum Computation by Adiabatic Evolution*. no journal, Januar 2000.
- [Gom60] GOMORY, RALPH E.: *AN ALGORITHM FOR THE MIXED INTEGER PROBLEM*. In: *Research Memoranda*, 1960.
- [Gri05] GRIFFITHS, DAVID: *Introduction to quantum mechanics*. Pearson Prentice Hall, Upper Saddle River, NJ, 2005.
- [gur] *Beschreibung des MILP-Solvers Gurobi der Firma Gurobi*.
- [GZF⁺19] GABOR, THOMAS, SEBASTIAN ZIELINSKI, SEBASTIAN FELD, CHRISTOPH ROCH, CHRISTIAN SEIDEL, FLORIAN NEUKART, ISABELLA GALTER, WOLFGANG MAUERER und CLAUDIA LINNHOFF-POPIEN: *Assessing Solution Quality of 3SAT on a Quantum Annealing Platform*. In: *Quantum Technology and Optimization Problems*, Seiten 23–35. Springer International Publishing, 2019.
- [Ham50] HAMMING, R. W.: *Error Detecting and Error Correcting Codes*. Bell System Technical Journal, 29(2):147–160, apr 1950.
- [Hoo] HOOS, HOLGER H.: *Erklärung zur Erstellung von Random 3SAT Instanzen der University of British Colombia*.
- [Hro14] HROMKOVIC, JURAJ: *Theoretische Informatik*. Springer Fachmedien Wiesbaden, September 2014.
- [Isi25] ISING, ERNST: *Beitrag zur Theorie des Ferromagnetismus*. Zeitschrift für Physik, 31(1):253–258, feb 1925.
- [JAG⁺11] JOHNSON, M. W., M. H. S. AMIN, S. GILDERT, T. LANTING, F. HAMZE, N. DICKSON, R. HARRIS, A. J. BERKLEY, J. JOHANSSON, P. BUNYK, E. M. CHAPPLE, C. ENDERUD, J. P. HILTON, K. KARIMI, E. LADIZINSKY, N. LADIZINSKY, T. OH, I. PERMINOV, C. RICH, M. C. THOM, E. TOLKACHEVA, C. J. S. TRUNCIK, S. UCHAIKIN, J. WANG, B. WILSON und G. ROSE: *Quantum annealing with manufactured spins*. Nature, 473(7346):194–198, may 2011.

- [Kar72] KARP, RICHARD M.: *Reducibility among Combinatorial Problems*. In: *Complexity of Computer Computations*, Seiten 85–103. Springer US, 1972.
- [Kar84] KARMARKAR, N.: *A new polynomial-time algorithm for linear programming*. In: *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, Seiten 373–395. Springer Science and Business Media LLC, dec 1984.
- [Kat50] KATO, TOSIO: *On the Adiabatic Theorem of Quantum Mechanics*. Journal of the Physical Society of Japan, 5(6):435–439, nov 1950.
- [KGV83] KIRKPATRICK, S., C. D. GELATT und M. P. VECCHI: *Optimization by Simulated Annealing*. Science, 220(4598):671–680, may 1983.
- [KM72] KLEE, VICTOR und GEORGE J. MINTY: *How good is the simplex algorithm?* O. Shisha, ed. Inequalities – III, Seiten 159–175, 1972.
- [KN98] KADOWAKI, TADASHI und HIDETOSHI NISHIMORI: *Quantum annealing in the transverse Ising model*. Physical Review E, 58(5):5355–5363, nov 1998.
- [Knu76] KNUTH, DONALD E.: *Big Omicron and big Omega and big Theta*. ACM SIGACT News, 8(2):18–24, apr 1976.
- [LD60] LAND, A. H. und A. G. DOIG: *An Automatic Method of Solving Discrete Programming Problems*. Econometrica, 28(3):497, jul 1960.
- [lpE] *Beschreibung des Simplexverfahrens anhand eines Beispiels*.
- [Luc14] LUCAS, ANDREW: *Ising formulations of many NP problems*. Frontiers in Physics, 2, 2014.
- [MF20] MCGEOCH, CATHERINE und PAU FARRE: *The D-Wave Advantage System: An Overview*. Technischer Bericht, D-Wave Systems, September 2020.
- [mis] *Darstellung der Fortschritte im Problem des Handelsreisenden der Universität Waterloo*.
- [MMWW02] MARCHAND, HUGUES, ALEXANDER MARTIN, ROBERT WEISMANTEL und LAURENCE WOLSEY: *Cutting planes in integer and mixed integer programming*. Discrete Applied Mathematics, 123(1-3):397–446, nov 2002.
- [MW01] MARCHAND, HUGUES und LAURENCE A. WOLSEY: *Aggregation and Mixed Integer Rounding to Solve MIPs*. Operations Research, 49(3):363–371, jun 2001.
- [ONL18] OHKUWA, MASAKI, HIDETOSHI NISHIMORI und DANIEL A. LIDAR: *Reverse annealing for the fully connected p-spin model*. Physical Review A, 98(2):022314, aug 2018.
- [PAN14] PAN, PING-QI: *Linear Programming Computation*. Springer Berlin Heidelberg, 2014.
- [PR89] PADBERG, M. und G. RINALDI: *A Branch-and-Cut Approach to a Traveling Salesman Problem with Side Constraints*. Management Science, 35(11):1393–1412, 1989.

- [Rie11] RIEFFEL, ELEANOR: *Quantum computing : a gentle introduction*. MIT Press, Cambridge, Mass, 2011.
- [SN17] SAKURAI, J. J. und JIM NAPOLITANO: *Modern Quantum Mechanics*. Cambridge University Press, sep 2017.
- [ST01] SPIELMAN, DANIEL A. und SHANG-HUA TENG: *Smoothed Analysis of Algorithms: Why the Simplex Algorithm Usually Takes Polynomial Time*. no journal, November 2001.
- [Tur37] TURING, A. M.: *On Computable Numbers, with an Application to the Entscheidungsproblem*. Proceedings of the London Mathematical Society, s2-42(1):230–265, 1937.
- [WA28] WILHELM ACKERMANN, DAVID HILBERT: *Grundzüge der theoretischen Logik*. Springer-Verlag GmbH, November 1928.
- [Whi99] WHITE, KENT: *The Family of Aircraft Aluminums*. Sport Aviation, November 1999.