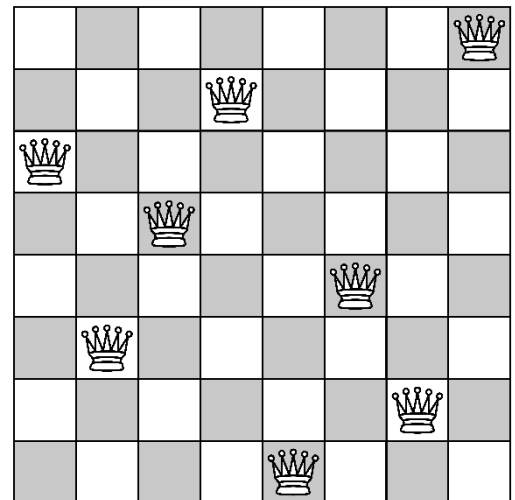
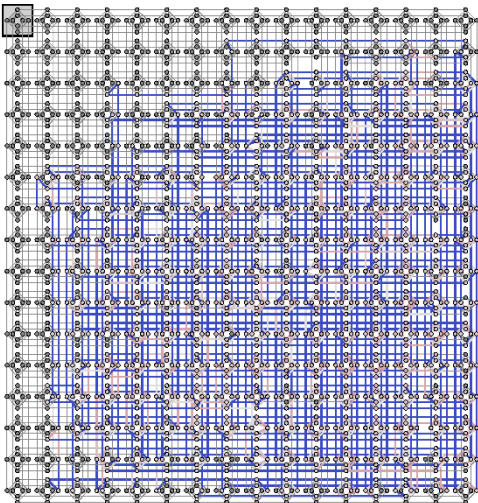


Lösung des n-Damenproblems auf einem adiabatischen Quantencomputer



Jakov D. Wallbrecher, Jonathan Treffler & Paul Schappert
GYMNASIUM DER REGENSBURGER DOMSPATZEN | REGENSBURG

Projektüberblick

Auf einem $n \times n$ großen Schachfeld sollen n Damen so aufgestellt werden, dass keine Dame eine andere bedroht. Dieses klassische Schachproblem haben wir auf einem adiabatischen Quantencomputer gelöst. Des Weiteren konnten wir unsere Lösungsstrategien auf andere Optimierungsprobleme, wie z.B. das Ergänzen von Sudokus und das Springerproblem, übertragen.

Bei unserem Projekt ging es aber nicht um das Finden von Lösungen für das n -Damenproblem, wofür es bereits gut geeignete Algorithmen auf herkömmlichen Computern gibt, sondern viel mehr um die Aufbereitung des Problems, die notwendig ist, damit es mit dem von uns verwendeten Quantencomputer, einem sogenannten **adiabatischen Quantum-Annealer**, gelöst werden kann. Das n -Damenproblem ist bisher von niemandem für einen Quantum-Annealer adaptiert worden.

Für unser Vorhaben mussten wir das Damenproblem in die Sprache der Mathematik übersetzen. Dabei galt es, eine Funktion zu definieren, die für jede Konstellation der Damen auf dem Schachbrett einen Wert zurückgibt, den man in der Optimierung als "**Energie**" der Verteilung bezeichnet. Diese Funktion muss für die korrekte Lösung des Problems minimal sein, da der Quantencomputer von Natur aus einen möglichst energiearmen Zustand anstrebt, sodass man, wenn er „gerechnet“ hat, die Lösung des Problems an den entsprechenden Werten der **Qubits** (äquivalent zu den herkömmlichen Bits auf einem Computer) ablesen kann, deren Zustände in unserem Fall als boolesche Variablen fungieren.

Um die erschlossene Energiefunktion auf Korrektheit zu testen, haben wir klassische Optimierungsalgorithmen wie **Simulated Annealing**, **Threshold Accepting** und den **Great Deluge Algorithmus** verwendet, mit denen man das Problem auf konventionellen Computern lösen kann. Diese konnten mit der getesteten Kostenfunktion alle eine Lösung des Problems finden, was grünes Licht für den ersten Test auf dem Quantencomputer bedeutete. Auf diesem haben wir unser „Programm“ dann ausgeführt. Die Rechenzeit auf dem für sechzehn Millionen Dollar erhältlichen Quantum-Annealer der Firma D-Wave Systems haben wir vom Forschungszentrum in Jülich (Prof. Dr. Kristel Michielsen) gestellt bekommen.

Wie genau ein **adiabatischer Quantumcomputer** funktioniert, welche Probleme wir lösen mussten, um das Damenproblem für den Quantum Annealer umzuformen, und zu welchen Ergebnissen wir beim Ausführen des Programms gekommen sind, möchten wir im Folgenden genauer erläutern.



Der D-Wave Quantum Annealer 2000Q, auf dem wir unsere Programme ausgeführt haben

Idee

Vom 5. Bis zum 7. November waren wir, im Anschluss an unsere Teilnahme am letztjährigen Schüler-experimentieren-Landeswettbewerb, zu einem Workshop am Fraunhofer Institut eingeladen. Ein Teil dieses Workshops bestand aus einem Seminar zur Findung neuer Ideen für Jugend-forscht-Projekte. Nach einigem Überlegen erinnerten wir uns an die Erzählungen unseres Betreuungslehrers Herrn Grünbauer von seinem Aufenthalt am Forschungszentrum in Jülich, das sich vor allem mit Quantencomputern beschäftigt. Wir waren schon beim ersten Kontakt mit diesem noch recht unbekanntem Forschungsgebiet fasziniert von der „Welt der Quanten“. Auf Nachfrage widersprach Herr Grünbauer unserer anfänglichen Vorstellung, dass es nahezu unmöglich sei, an Rechenzeit zu kommen oder überhaupt in dieser Richtung zu forschen. Wir erhielten erste Informationen über die Funktionsweise eines adiabatischen Quantencomputers und machten uns sogleich daran, ein lösbares Problem zu suchen.

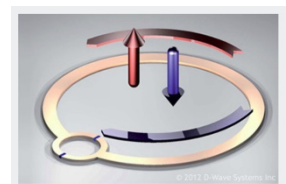
Wir mussten ein Optimierungsproblem finden, das sich möglichst einfach mit Einsen und Nullen darstellen lässt. Einer von uns (Paul) hatte sich zufällig ein halbes Jahr zuvor mit der Lösung von Optimierungsproblemen mittels herkömmlicher Algorithmen (v.a. simulated annealing) auf normalen Computern beschäftigt. Unter den damals bearbeiteten Problemen war unter anderem auch das n-Damenproblem, das alle vorhandenen Voraussetzungen zu erfüllen schien. Auf der Rückfahrt vom Workshop nach Regensburg waren wir dann schon damit beschäftigt, zu versuchen, die Regeln des n-Damenproblems mathematisch zu formulieren. Diese Zugfahrt war der Beginn unseres Projekts.

Funktionsweise eines adiabatischen Quantencomputers

Quantenbit auf einem Quantencomputer

Quantencomputer rechnen - anders als herkömmliche Computer - nicht mit normalen Bits, sondern mit Quantenbits, kurz „Qubits“. Im Gegensatz zu klassischen Bits können sich Qubits in einer Überlagerung der Werte 0 und 1 (quantenmechanisch: **Superposition**) befinden. Erst beim Auslesen des Zustands, das physikalisch einer Messung seines Werts entspricht, nimmt es einen der beiden Zustände mit einer bestimmten Wahrscheinlichkeit an. Diese Wahrscheinlichkeit kann durch **Coupler** beeinflusst werden. Ein Qubit befindet sich also quantenmechanisch im Zustand $a|0\rangle + b|1\rangle$, wobei a und b (komplexe) Zahlen sind, deren Betrag kleiner gleich 1 ist und für die gilt: $|a|^2 + |b|^2 = 1$. Beim Auslesen (Messen) des Qubits erhält man dann den Wert 0 mit einer Wahrscheinlichkeit von $|a|^2$ und 1 mit einer Wahrscheinlichkeit von $|b|^2$.

Qubits stellen einen Wert nicht wie Bits mit Transistoren durch Strom an oder aus dar, sondern durch den Spin (Drehrichtung) eines Elektrons, oder wie bei dem von uns verwendeten Rechner von D-Wave, durch die Flussrichtung des Stroms in einem supraleitenden Ring (siehe Abbildung rechts). Die Superposition entspricht also dem Fluss des Stroms in beide Richtungen.



Qubit auf einem D-Wave

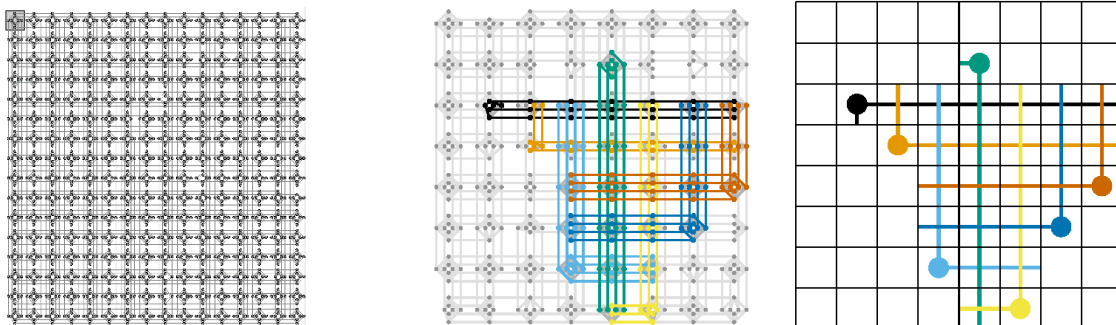
Man unterscheidet zwei Arten von Quantencomputern: Universelle **Quantengattercomputer**, bei denen man jedes Qubit mit jedem anderen koppeln kann und adiabatische Quantencomputer, auch Quantum Annealer genannt. Letztere funktionieren wie folgt:

"Gemäß den Gesetzen der Quantenmechanik bleibt ein quantenmechanisches System, das sich im Grundzustand (Zustand minimaler Energie) eines zeitunabhängigen Systems befindet, auch bei Veränderungen des Systems im Grundzustand, wenn die Veränderung nur hinreichend langsam (also adiabatisch) passiert. Die Idee des adiabatischen Quantencomputers ist es, ein System zu konstruieren, das einen zu dieser Zeit noch unbekanntem Grundzustand hat, der der Lösung eines bestimmten Problems entspricht, und ein anderes, dessen Grundzustand leicht experimentell zu präparieren ist. Anschließend wird das leicht zu präparierende System in das System überführt, an dessen Grundzustand man interessiert ist, und dessen Zustand dann gemessen. Wenn der Übergang langsam genug erfolgt ist, hat man so die Lösung des Problems." [Eintrag auf Wikipedia, aufgerufen am 18.1.2019]

Die Frage, ob Quantum-Annealer zu den „echten“ Quantencomputern zählen ist stark umstritten. Der Vorteil der Annealer ist, dass sie speziell für die Lösung von Optimierungsproblemen konzipiert sind. Außerdem arbeiten sie bei einer höheren Anzahl an Qubits aufgrund ihrer ausgereifteren Technik deutlich zuverlässiger. Trotzdem ist die Anzahl der verfügbaren Qubits nach wie vor gering, weshalb es wahrscheinlich noch einige Zeit dauern wird, bis (adiabatische) Quantencomputer auch größere „real-world“ Probleme lösen können und herkömmlichen Computern möglicherweise in bestimmten Bereichen überlegen sind.

Chimera-Graph

Der sogenannte **Chimera-Graph** beschreibt die Verbindungen aller physikalischen Qubits auf einem D-Wave-Quanten-Annealer. Er zeigt alle aktuell aktiven Qubits (Ausfälle einzelner Qubits sind möglich) und deren Verbindungsmöglichkeiten zu anderen Qubits.



Chimera Graph und Koppelung einzelner physikalischer Qubits zu logischen Qubits (jede Farbe stellt ein logisches Qubit dar)

Qubits können entweder mit dem Gleichheits- oder dem Ungleichheitscoupler gekoppelt werden. Auf einem adiabatischen Quantencomputer sind nicht alle physikalischen Qubits direkt miteinander verbunden (koppelbar). Um diesem Problem Abhilfe zu schaffen, werden mehrere physikalische Qubits zu logischen oder auch universellen Qubits zusammengefasst beziehungsweise gekoppelt (siehe Abbildung oben).

Um ein logisches Qubit zu erhalten koppelt man mehrere physikalische Qubits mit dem Gleichheitscoupler, um mehr potentielle Verbindungen zu anderen, logischen Qubits zu ermöglichen. Diese kann man dann wiederum untereinander so verbinden, dass der **Hamiltonian** (zu Deutsch: Energiefunktion), der das Problem beschreibt, dargestellt wird. Es gibt leider kein „Erfolgsrezept“ um die physikalischen Qubits so zu koppeln, dass sich die entsprechenden logischen Qubits verknüpfen lassen. Daher hat die Firma D-Wave eine Library geschrieben, um das **Embedding** zu automatisieren, welche auch wir genutzt haben. Das Embedding ist für die Verbindung der physikalischen zu logischen Qubits verantwortlich.

Hamiltonian und Energieberechnung

Der Hamiltonian bildet die Basis für die Koppelung der Qubits auf dem Chimera-Graphen. Er bildet - in Form einer Matrix - ab, welche Kombination der Zustände zweier Qubits bestraft oder belohnt werden soll. Wenn man also nicht möchte, dass zwei Felder besetzt sind (z.B. die Felder a und b) trägt man an der Position (a|b) der Matrix den Wert 1 an. Wenn dann zwei Damen auf diese beiden Felder gesetzt werden, bestraft der Hamiltonian dies mit 1. Alle diese „Strafen“ oder auch „Fehler“ entsprechen zusammen der vorher beschriebenen Energie.

Diese Energie wird auf Basis der Hamiltonian-Matrix H und eines Vektors \vec{q} berechnet, in dem die Werte der Qubits angetragen sind. Um jeden Qubit-Wert mit jedem anderen abzugleichen, muss also der Energie-Term folgendermaßen aussehen:

$$E(\vec{q}) = q_a \cdot q_a \cdot h_{aa} + q_a \cdot q_b \cdot h_{ab} + q_a \cdot q_c \cdot h_{ac} + q_a \cdot q_d \cdot h_{ad} + \dots$$

Das lässt sich per Summenzeichen erneut zusammenfassen:

$$E(\vec{q}) = \sum_{i \leq j} h_{ij} \cdot q_i \cdot q_j = \vec{q}^t \cdot H_{\text{Problem}} \cdot \vec{q}$$

a	b
c	d

	a	b	c	d
a	0	1	0	0
b	0	0	0	0
c	0	0	0	0
d	0	0	0	0

Beispiel für eine Hamiltonian-Matrix

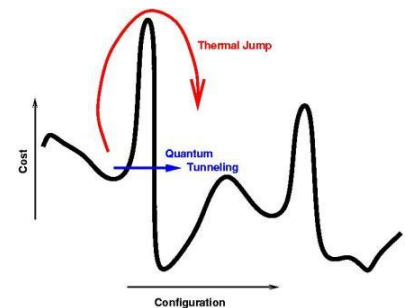
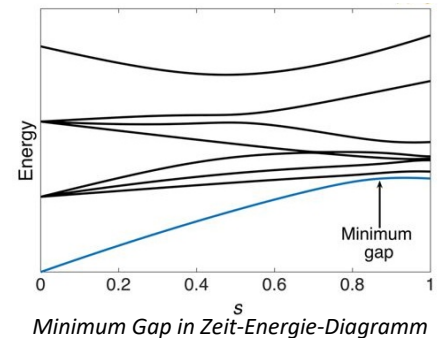
Quantum Annealing

Ein Durchlauf eines Quantum-Annealing-Programms beginnt damit, dass mittels des **Initial-Hamiltonian** alle Qubits in Superposition „gesetzt“ werden. Dann wird der eigentliche problemspezifische Hamiltonian langsam „eingblendet“. Dies geschieht durch eine Erhöhung von B bei gleichzeitiger Verringerung von A. Der Gesamt-Hamiltonian wird wie folgt berechnet:

$$H_{Gesamt} = A \cdot H_{initial} + B \cdot H_{Problem}$$

In der Grafik rechts sieht man (blau) den besten Energiezustand zu jedem Zeitpunkt des Annealing-Vorgangs. An der Stelle der **Minimum Gap** ist die Wahrscheinlichkeit, in einen schlechteren Zustand zu rutschen am größten. Je größer die Minimum Gap, desto leichter lässt sich das Problem mit einem Quantencomputer lösen.

Durch quantenmechanische Gesetzmäßigkeiten ist es außerdem möglich, beim unendlich langsamen Übergang vom einen in den anderen Hamiltonian alle Erhebungen in der Energielandschaft zu tunneln (siehe Grafik rechts). In der Theorie kann man durch unendlich langsamen Übergang zwischen beiden Hamiltonians immer das beste Ergebnis erzielen. In der Praxis reichen allerdings oft zweistellige Mikrosekundenbeträge, um zu einem Ergebnis zu kommen. Da Quantum-Annealer aber oft keine perfekte Lösung (globales Minimum), sondern nur eine gute Lösung (lokales Minimum) finden, lässt man das gleiche Problem üblicherweise ca. 10.000-mal nacheinander mit den gleichen Parametern berechnen. Abschließend soll erwähnt sein, dass die Anzahl richtiger Ergebnisse pro 10.000 Durchläufe sehr stark schwankt, da der Quantencomputer nicht immer im besten Energiezustand endet. Quantencomputer können allerdings bisher nur sehr kompakt formulierbare Probleme lösen.



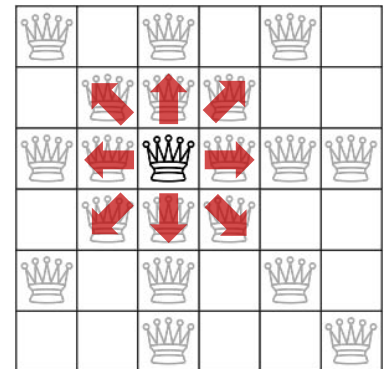
Tunneling in einer Energielandschaft

Von der Kostenfunktion bis zum Output

Nachdem ein Problem in eine Kostenfunktion verpackt wurde, wird diese an die Oceans-SDK von D-Wave weitergegeben, die das Embedding setzt. Das Embedding wird dann gemeinsam mit Parametern wie Durchlaufdauer und Anzahl der Durchläufe an den Server in Vancouver gesendet, der den Annealvorgang steuert. Nach Ende jeden Rechenvorgangs werden die Werte aller Qubits gemessen. Die Messwerte werden am Ende aller Anneals gesammelt zurückgeschickt und von unserem Programm problemspezifisch dekodiert.

Das n-Damenproblem

Das n-Damenproblem verlangt, auf einem $n \times n$ Felder großen Schachbrett n Schachdamen so anzuordnen, dass keine Dame eine andere bedroht. Eine Dame darf vertikal, horizontal und diagonal in alle Richtungen ziehen. Erstmals formuliert wurde das Problem 1848 von Max Bezzel, allerdings in Zusammenhang mit der Anzahl möglicher Lösungen für die Grundform $n=8$. Das Problem lässt sich sogar mit **Bruteforce** (Durchprobieren aller Möglichkeiten) in maximal 8^8 Anläufen lösen. Mit etwas komplexeren heuristischen Optimierungsverfahren lässt sich selbst das 1000-Damen-Problem in angemessener Zeit lösen. Das besondere an unserem Projekt ist aber auch nicht das Lösen des Problems, sondern das „in-QUBO-Formbringen“, wobei **QUBO** für **Quadratic Unconstrained Binary Optimization** steht. Ansätze zur Lösung des n-Damenproblems auf einem gatterbasierten Quantencomputer wurden bereits in mehreren papers behandelt. Das Lösen auf einem Quantum-Annealer ist jedoch komplett neu.

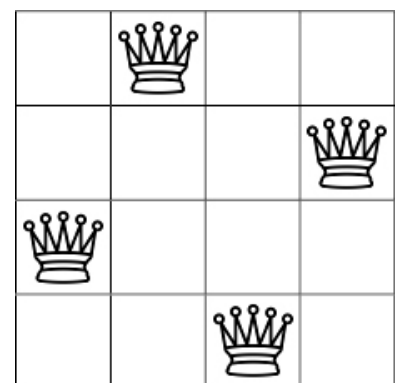


Mögliche Bewegungsrichtungen einer Schachdame

Hamiltonian für das n-Damenproblem

Wir haben unseren Hamiltonian (siehe „Hamiltonian und Energieberechnung“) über einen Term erschlossen. Wir mussten dazu eine Funktion kreieren, die für die Lösung des Problems minimal ist. Diese Kostenfunktion kann dann in eine Matrix umgeformt werden. Das im Folgenden beschriebene Vorgehen haben wir mit einem selbstgeschriebenen Programm für ein beliebiges n automatisiert.

Eine Schachdame darf vertikal, horizontal und in alle Richtungen diagonal ziehen. Das bedeutet, dass auf jeder Horizontale, Vertikale und Diagonale höchstens eine Dame stehen darf. Eine Diagonale darf aber auch frei bleiben.



Korrekte Lösung des 4-Damen-Problems

Wir zerlegen die Erstellung der Funktion in ihre Einzelbestandteile und beginnen dazu bei den Horizontalen.

Auf einem $n \times n$ großen Schachfeld gibt es n Horizontalen, auf denen jeweils genau eine Dame stehen muss. Wir suchen also eine Funktion, die dann den geringsten Wert zurückgibt, wenn genau eine Dame auf der Horizontalen steht. Dazu addiert man alle Felder der Horizontalen (0, wenn keine Dame; 1 wenn eine Dame), zieht 1 ab und quadriert dies, damit das Ergebnis bei einer Dame in der Horizontalen kleiner ist als bei 0 Damen oder mehr als eine Dame:

$$(a + b + c + d - 1)^2$$

Dieses Vorgehen wendet man auf jede Horizontale an und addiert die Einzelergebnisse auf.

Das Gleiche wiederholt man für die Vertikalen. Das Vorgehen bleibt dabei konstant.

$$(a + e + i + m - 1)^2$$

Auf den Diagonalen darf jeweils eine oder keine Dame stehen, da es immer $(n * 2) - 1$ Diagonalen pro Richtung (von links oben nach rechts unten oder von rechts oben nach links unten) gibt, von denen nur auf n eine Dame stehen kann.

$$(a + f + k + p - 0,5)^2$$

Für eine oder keine Dame auf der Diagonale resultiert bei diesem Term eine Energie von 0.25, bei mehr Damen höhere Werte. Dasselbe gilt für die andere Diagonalrichtung. Die Eckfelder (als Diagonalen mit genau einem Feld) werden nicht berücksichtigt, weil die Gesamtenergie auf diesen Diagonalen nur 0 oder 1 sein, dort also ohnehin kein Fehler auftreten kann.

Am Beispiel des 4-Damen-Problems hier die erste vollständige Hamiltonian-Formel:

$$E = (a + b + c + d - 1)^2 + (e + f + g + h - 1)^2 + (i + j + k + l - 1)^2 + (m + n + o + p - 1)^2 + \\ (a + e + i + m - 1)^2 + (b + f + j + n - 1)^2 + (c + g + k + o - 1)^2 + (d + h + l + p - 1)^2 + \\ (i + n - 0,5)^2 + (e + j + o - 0,5)^2 + (a + f + k + p - 0,5)^2 + (b + g + l - 0,5)^2 + (c + h - 0,5)^2 + \\ (b + e - 0,5)^2 + (c + f + i - 0,5)^2 + (d + g + j + m - 0,5)^2 + (h + k + n - 0,5)^2 + (l + o - 0,5)^2$$

Es kommen im gesamten Hamiltonian nur die beiden beschriebenen Terme vor. Diese kann man dann in eine Matrix umformen, die der Quantencomputer als Eingabe erwartet.

$$(a + b + c + d - 1)^2 \\ = a^2 + b^2 + c^2 + d^2 - 2a - 2b - 2c - 2d + 2ab + 2ac + 2ad + 2bc + 2bd + 2cd + 1$$

Da alle Variablen nur 0 oder 1 sein können, gilt: $a = a^2$. Daraus folgt:

$$a^2 + b^2 + c^2 + d^2 - 2a^2 - 2b^2 - 2c^2 - 2d^2 + 2ab + 2ac + 2ad + 2bc + 2bd + 2cd + 1 \\ = -a^2 - b^2 - c^2 - d^2 + 2ab + 2ac + 2ad + 2bc + 2bd + 2cd + 1$$

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p

Nummerierung der Schachbrettfelder, die in der Formel gebraucht wird

Der andere Term für die Bedingung „Genau eine oder keine Dame“:

$$\begin{aligned}
 & (a + f + k + p - 0,5)^2 \\
 &= a^2 + f^2 + k^2 + p^2 - a - f - k - p + 2af + 2ak + 2ap + 2fk + 2fp + 2kp + 0,25 \\
 &= 2af + 2ak + 2ap + 2fk + 2fp + 2kp + 0,25
 \end{aligned}$$

Die Konstanten 1 und 0.25 werden nicht beachtet, da sie lediglich eine Verschiebung des Energiegraphs an der y-Achse bewirken.

Der daraus resultierende Term der Form $m_1ab + m_2cd + \dots$ lässt sich sehr einfach auf eine Matrix der Größe $n^2 * n^2$ übertragen. Dazu wird beispielsweise für $2ak$ bei Feld (a|k) der Wert 2 eingetragen. Aus diesen Überlegungen resultiert die nebenstehende Hamiltonian-Matrix für das 4-Damen-Problem.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
a	-2	2	2	2	2	2	0	0	2	0	2	0	2	0	0	2
b	0	-2	2	2	2	2	2	0	0	2	0	2	0	2	0	0
c	0	0	-2	2	0	2	2	2	2	0	2	0	0	0	2	0
d	0	0	0	-2	0	0	2	2	0	2	0	2	2	0	0	2
e	0	0	0	0	-2	2	2	2	2	2	0	0	2	0	2	0
f	0	0	0	0	0	-2	2	2	2	2	2	0	0	2	0	2
g	0	0	0	0	0	0	-2	2	0	2	2	2	2	0	2	0
h	0	0	0	0	0	0	0	-2	0	0	2	2	0	2	0	2
i	0	0	0	0	0	0	0	0	-2	2	2	2	2	2	0	0
j	0	0	0	0	0	0	0	0	0	-2	2	2	2	2	2	0
k	0	0	0	0	0	0	0	0	0	0	-2	2	0	2	2	2
l	0	0	0	0	0	0	0	0	0	0	0	-2	0	0	2	2
m	0	0	0	0	0	0	0	0	0	0	0	0	-2	2	2	2
n	0	0	0	0	0	0	0	0	0	0	0	0	0	-2	2	2
o	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-2	2
p	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-2

Außerdem haben wir in der Programmiersprache „Processing“, einem Java-Ableger, ein Programm geschrieben, das die Matrix für ein beliebiges n berechnet. Außerdem haben wir die Kostenfunktion mathematisch in allgemeiner Form aufgestellt. Die Variablen i und j werden beginnend bei eins bis n^2 hochgezählt, wobei sie jeweils für das i-te bzw. j-te Feld stehen.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Nummerierung der Felder, die in der Formel gebraucht wird

$$a_{ij} = \begin{cases} -2 & \text{für } i = j \\ 0 & \text{für } i > j \\ 2 & \text{für } i < j \wedge (i \bmod n = j \bmod n \\ & \vee i \bmod (n - 1) = j \bmod (n - 1) \\ & \vee i \bmod (n + 1) = j \bmod (n + 1) \\ & \vee (i - 1) - (i - 1) \bmod n = (j - 1) - (j - 1) \bmod n) \\ 0 & \text{sonst} \end{cases}$$

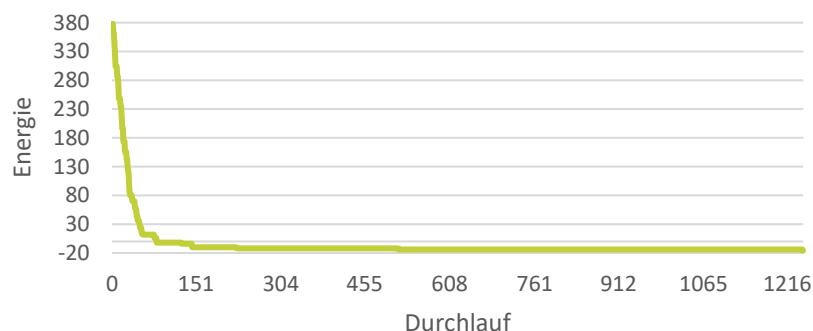
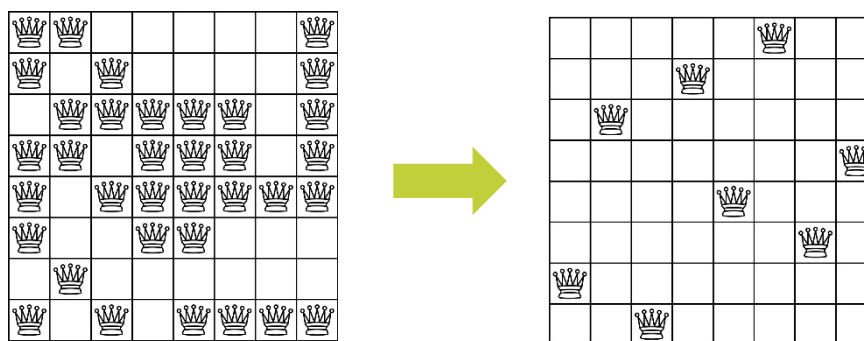
Lösung des n-Damenproblems mit Optimierungsverfahren auf klassischen Computern

Rechenzeit auf einem Quantencomputer ist teuer und knapp. Daher haben wir vor dem Ausführen unseres Programms auf dem echten Quanten-Annealer den berechneten Hamiltonian mit mehreren Optimierungsverfahren getestet und diese verglichen. Die verwendeten Optimierungsverfahren wollen wir im Folgenden vorstellen.

Alle Algorithmen arbeiten mit dem gleichen Ansatz. Sie nehmen eine Änderung an der Ausgangssituation vor (hier: auf einem beliebigen Feld eine Dame wegnehmen/hinzufügen). Diese Änderung wird als **Move** bezeichnet. Dann wird die daraus entstandene Situation von einer Kostenfunktion bewertet und mit der Energie der vorherigen Konstellation verglichen. Der Zug wird daraufhin entweder angenommen oder wieder rückgängig gemacht. Der einzige Unterschied besteht in den variierenden Kriterien, die festlegen, ob der Zug angenommen wird oder nicht.

Greedy-Algorithmus

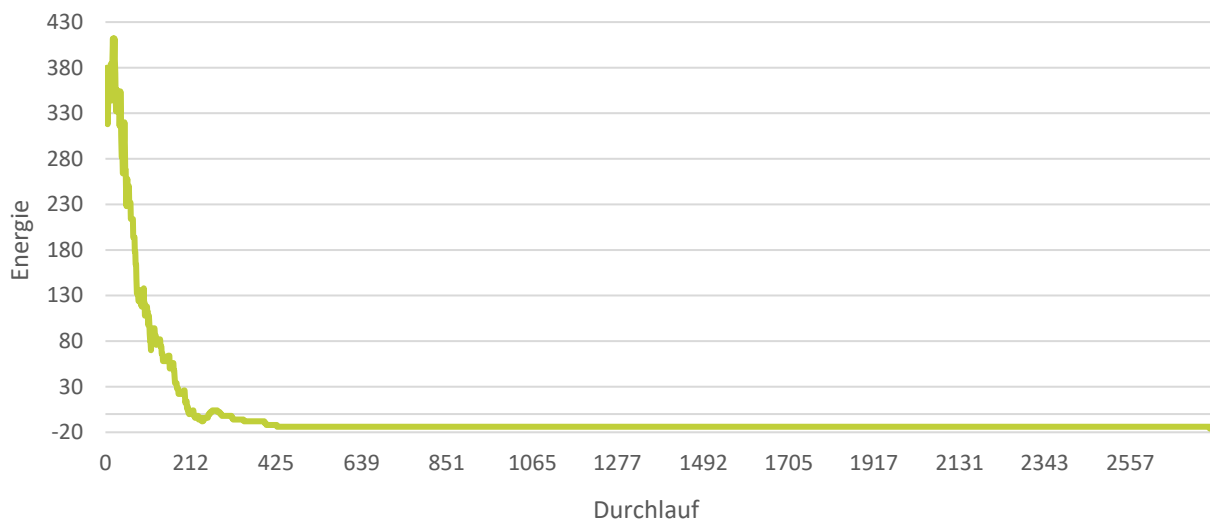
Der **Greedy-Algorithmus** (englisch für gierig) nimmt eine Änderung nur dann an, wenn sie die Situation verbessert hat. Das führt allerdings bei vielen np-harten Problemen zu einer unzureichenden Lösung, da man in einem lokalen Minimum stecken bleiben kann. Trotzdem hat der simpelste unserer Testalgorithmen überraschend gute Ergebnisse geliefert. Die Anzahl der notwendigen Durchläufe schwankt bei gleicher Ausgangsbesetzung des Schachfelds sehr stark.



Threshold Accepting

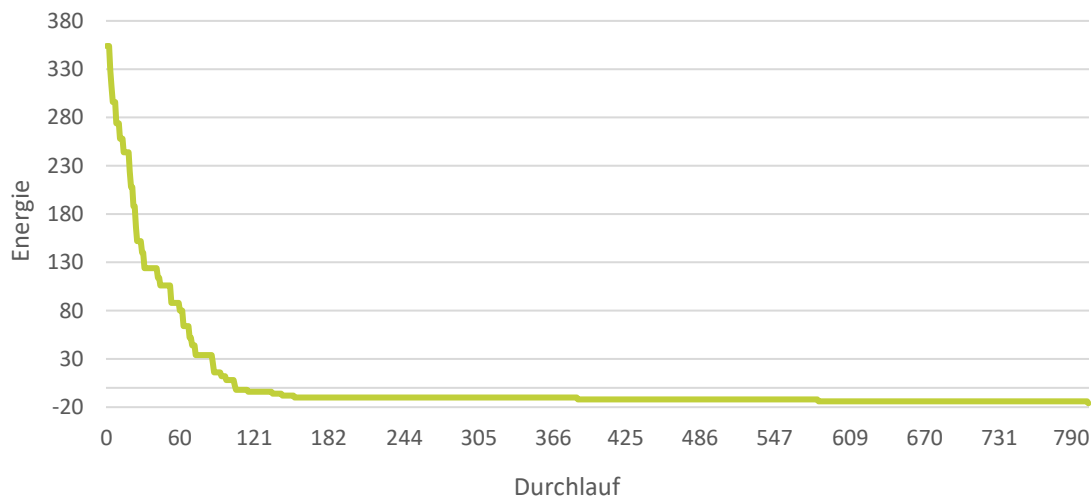
Beim **Threshold Accepting** (englisch für Schwellenwertannahme) wird die Änderung nur dann angenommen, wenn die Differenz von neuer und alter Energie unter einem bestimmten Wert liegt. Dieser Schwellenwert wird nach jedem Durchlauf etwas herabgesetzt. So dürfen am Anfang noch mehr Situationsverschlechterungen vorkommen, wohingegen später fast ausschließlich Verbesserungen angenommen werden. Das macht es möglich, lokale Minima in der Energielandschaft zu überwinden und dadurch bessere Ergebnisse zu erzielen. Das Threshold Accepting gehört - im Gegensatz zum Greedy-Algorithmus - zu den heuristischen Optimierungsverfahren.

Aus der Grafik lässt sich entnehmen, dass sich der Energiezustand zuerst verschlechtert, um dann zum globalen Minimum zu finden.



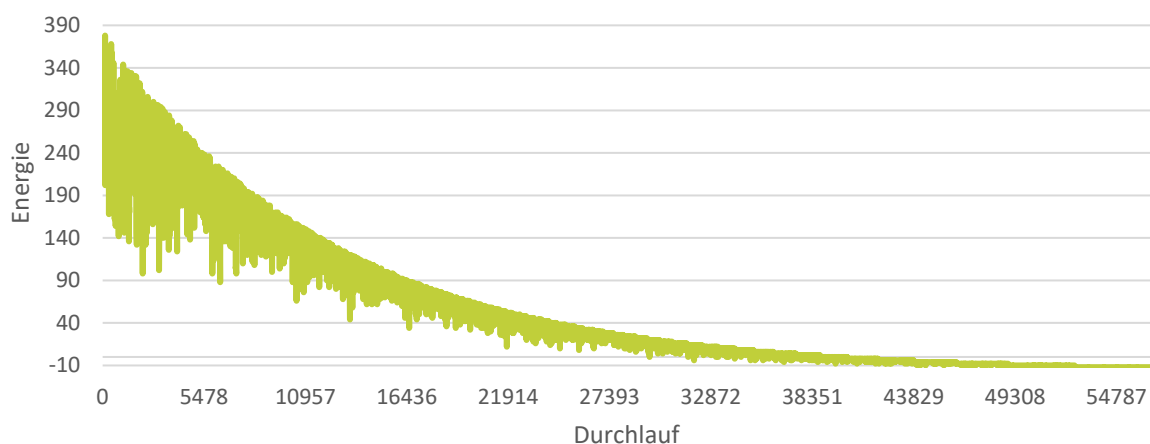
Simulated Annealing

Beim **Simulated Annealing** (englisch für simuliertes Abkühlen) werden alle gewinnbringenden Moves angenommen, wobei auch eine Verschlechterung zu einer Wahrscheinlichkeit von $e^{-\frac{\Delta E}{T}}$ angenommen werden darf. Der Parameter T ist die sogenannte Temperatur, welche nach jedem Durchlauf abgesenkt wird. Das erklärt auch den Namen Annealing (=Abkühlen). ΔE steht für die Energiedifferenz. Simulated Annealing überwindet lokale Minima in der Energielandschaft noch effizienter und mit höherer Wahrscheinlichkeit als Threshold Accepting, weshalb es für viele Probleme im Optimierungsbereich die erste Wahl ist.



Great Deluge Algorithmus

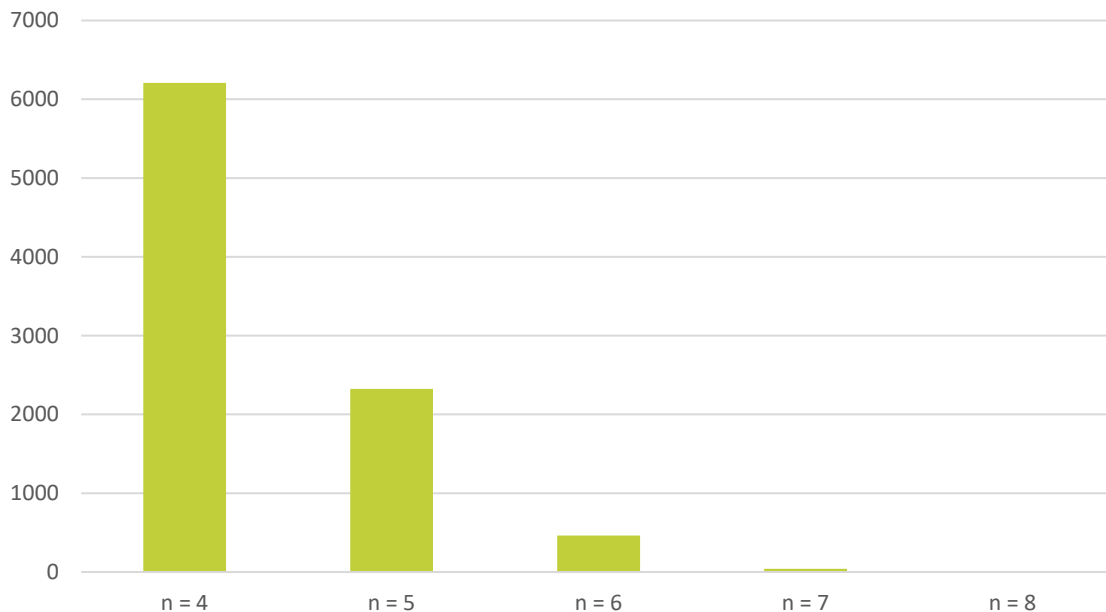
Der **Great Deluge Algorithmus** (englisch für Sintflut-Algorithmus) lässt eine Änderung dann zu, wenn die neue Energie unter einem bestimmten Wert liegt. Dieser wird nach jeder Änderung verringert. Der Sintflut-Algorithmus hat mit unserem Hamiltonian sehr schlechte Ergebnisse erreicht. Um eine Lösung zu finden benötigte er oft ein Vielfaches der Anzahl an Durchläufen im Vergleich zu den anderen Optimierungsverfahren.



Insgesamt zeigten alle diese Simulationen, dass unsere Energiefunktion richtig gewählt war.

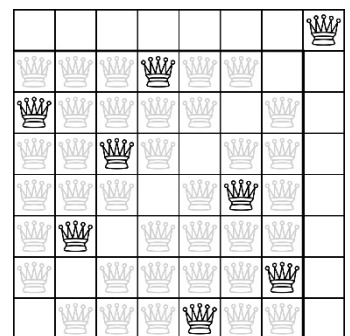
Auswertung der Ergebnisse des Quanten-Annealers

Wir haben auf dem Quantencomputer „DW_2000Q_2_1“ die Hamiltonians für alle n von 4 bis 8 ausgeführt. Als Eingabewerte erhält der Quantencomputer - neben der Besetzung des Chimera-Graphen - die Anzahl der Durchläufe und die Dauer eines Durchlaufs. Diese Dauer beträgt standardmäßig 20 Mikrosekunden. Hier die Anzahl der richtigen Ergebnisse pro 10.000 Durchläufe:



Für n=8 lieferte der Quantencomputer kein einziges richtiges Ergebnis. Das lässt sich auf die nicht ausreichende Anzahl an zur Verfügung stehenden Qubits zurückführen. Der Quanten-Annealer von D-Wave hat nämlich 2048 physikalische Qubits, womit sich nur $\sqrt{2048}$ also rund 45 logische Qubits simulieren lassen. Für das 8x8 Feld brauchen wir jedoch 64 logische Qubits.

Wir hatten uns aber vorgenommen, das klassische 8-Damenproblem auf einem Quantencomputer zu lösen. Um dies trotz der Begrenzung durch die Anzahl logischer Qubits möglich zu machen, haben wir uns überlegt, dass man zu einer Lösung des 7-Damen Problems mit freier Gesamtdiagonale, falls diese existiert, einfach eine weitere Dame dazustellen könnte (siehe Grafik). Dazu mussten wir unseren Hamiltonian um diese Bedingung erweitern. Von 10.000 Durchläufen auf dem Quantencomputer konnten wir tatsächlich 80-mal ein richtiges Ergebnis für diesen Spezialfall finden. Mit diesem „Trick“ lässt sich allerdings nur eine bestimmte Lösung des 8-Damenproblems finden. Bis zum Bundeswettbewerb wollen wir alle weiteren Lösungen für das 8-Damenproblem durch Dazustellen der achten Dame an einer anderen Position in der ersten Reihe finden.

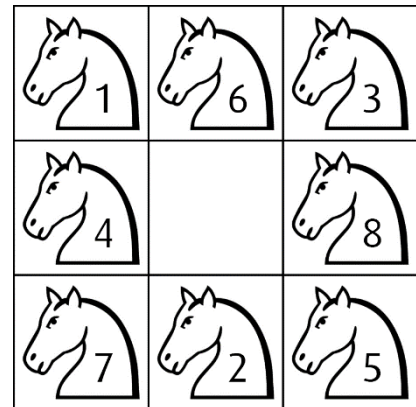


Ergänzung des 7-Damen Problems zum 8-Damen Problem

Lösung des Springerproblems auf einem adiabatischen Quantencomputer

Um unser Projekt noch zu erweitern haben wir uns nach weiteren, potenziell mit einem Quantencomputer lösbaren Problemen umgesehen. Fündig geworden sind wir bei einem weiteren Schachproblem: dem „knights tour problem“ (zu Deutsch: Springerproblem). Bei diesem Problem es darum, auf einem Schachfeld eine Route finden, bei der ein Springer jedes Feld genau einmal besucht.

Wegen der momentan noch eingeschränkten Anzahl an verfügbaren Qubits haben wir uns auf das 3 x 3 Springerproblem beschränkt. Um dieses Problem als Hamiltonmatrix darzustellen, haben wir uns die Züge des Springers als Zeitebenen vorgestellt, die untereinander angeordnet sind. So „hüpft“ der Springer von einer in die nächste Zeitebene. Daraus ergeben sich mehrere Regeln, die später im Hamiltonian dargestellt werden müssen: Zum einen darf auf einem Feld in unterschiedlichen Zeitebenen insgesamt nur ein Springer stehen (jedes Feld darf nur einmal betreten werden). Außerdem darf auf einer Zeitebene nur ein Springer stehen (der Springer kann nicht auf zwei Feldern gleichzeitig stehen). Die dritte Einschränkung ist, dass die Springer in übereinanderliegenden Zeitebenen immer einen „Springerabstand“ voneinander entfernt stehen müssen.



Um die Matrix noch weiter zu vereinfachen, konnten wir das 3 x 3 auf ein 2 x 2 Feld reduzieren. Das ist möglich, da der Springer, bedingt durch den Springerzug, bei jeder Zeitebene die Feldfarbe wechselt. Da das mittlere Feld nicht erreicht werden kann, werden in jeder Zeitebene somit nur noch 4 statt vorher 9 Felder benötigt. Mit diesem Trick konnten wir schließlich das Springerproblem auf dem Quanten Annealer lösen.

Für dieses Problem haben wir den Hamiltonian nicht über eine Formel, sondern direkt hergeleitet. Dazu haben wir ein Programm geschrieben, welches jedes Feld mit jedem Anderen betrachtet. Sobald es zwei Felder findet, die nach den oben beschriebenen Regeln nicht gleichzeitig besetzt sein dürfen, trägt es an der entsprechenden Stelle in der Matrix eine Bestrafung an. Damit die Energie nicht auch dann minimal ist, wenn keine oder zu wenige Felder besetzt sind, wird jedes besetzte Feld, wie auch beim n-Damenproblem, belohnt. Daher ist die ganze Diagonale der Hamilton-Matrix mit -2 besetzt.

Danksagung und Ausblick

Wir bedanken uns bei unserem Projektbetreuer René Grünbauer für die Betreuung und den Kontakt zum Forschungszentrum Jülich. Außerdem möchten wir uns bei Prof. Dr. Kristel Michielsen vom Forschungszentrum Jülich für ihre Unterstützung und die Rechenzeit am D-Wave Quantencomputer bedanken. Nicht zuletzt wollen wir Herrn Dennis Willsch aus der Forschungsgruppe von Prof. Michielsen für die Erklärungen und Hilfestellungen rund um die Benutzung und Steuerung des Quantencomputers danken.

Kurz vor dem Landeswettbewerb ist es uns gelungen, Sudokus auf dem Quantum-Annealer zu lösen. Unseren Lösungsweg hierzu wollen wir im Bundeswettbewerb präsentieren.

Bildquellen

- Titelbild (Chimera-Graph mit Kopplungen):
https://cloud.dwavesys.com/qubist/submit_problem/
- D-Wave Quantencomputer 2000Q:
<https://www.dwavesys.com/sites/default/files/2000Q%20Extracted%20Image%20website.jpeg>
- Darstellung des Spins eines Qubits:
<https://www.dwavesys.com/sites/default/files/tut-hardware-qubit-schematic.jpg>
- Chimera-Graph (grau):
https://cloud.dwavesys.com/qubist/submit_problem/
- Chimera-Graph Beispielkoppelung:
<https://ai2-s2-public.s3.amazonaws.com/figures/2017-08-08/39130a8ff551498da753acb0d9a3787f3f403b11/4-Figure2-1.png>
- Zeit-Energie-Diagramm eines Anneals:
https://media.springernature.com/lw900/springer-static/image/art%3A10.1038%2Fs41534-018-0060-8/MediaObjects/41534_2018_60_Fig1_HTML.jpg
- Energielandschaft (Tunneling):
<https://upload.wikimedia.org/wikipedia/commons/1/12/Quant-annl.jpg>
- Alle nicht genannten Grafiken sind selbst erstellt.