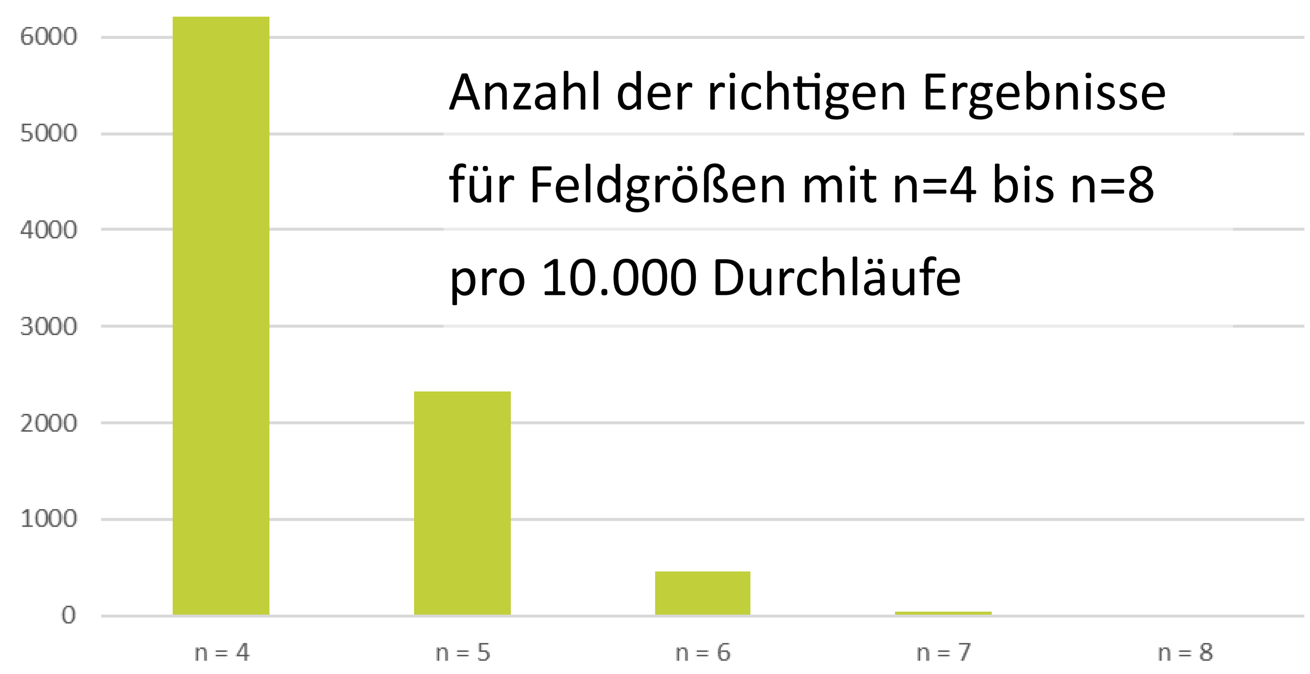
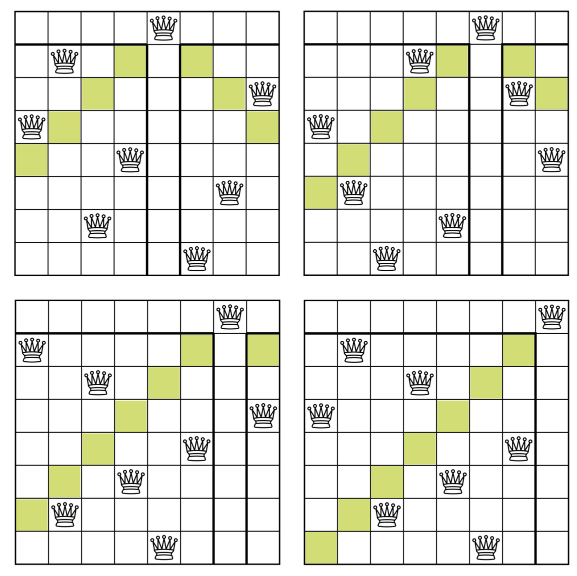
Ergebnisse für das n-Damenproblem



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

Allgemeine Form unseres Hamiltonians H=(h_{ij}) für das n-Damenproblem

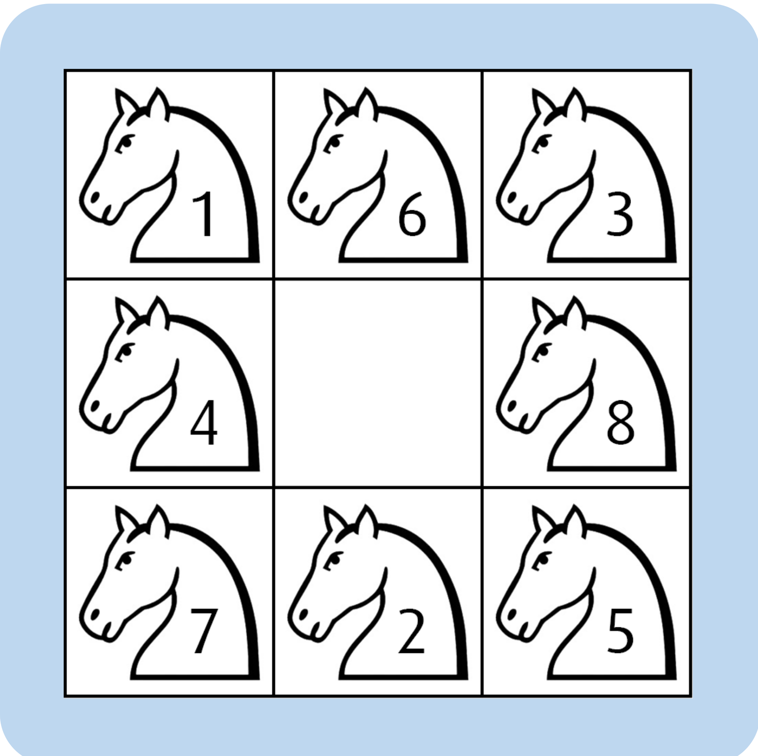
Energie der Konfiguration	Anzahl dieser Konfigurationen	Werte der Qubits (Konfig. der Damen)
-8.000000	**171**	**[0 0 1 0 1 0 0 0 0 0 0 1 0 1 0 0]**
-8.000000	**6794**	**[0 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0]**
-6.000000	2	[1 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0]

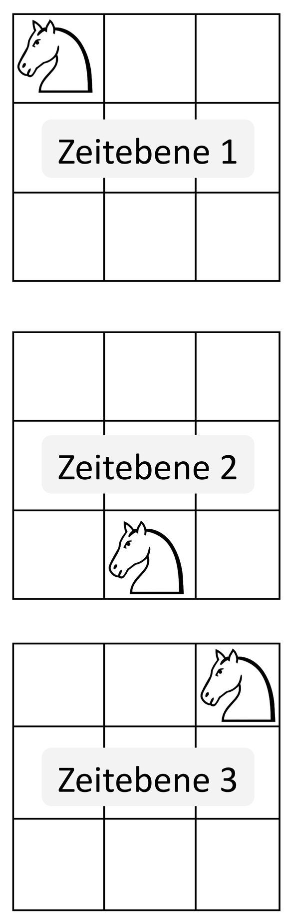


Da für das 8-Damenproblem **64** Quantenbits benötigt werden, der D-Wave Annealer aber derzeit nur mit **45** logischen Qubits arbeitet, kann für diese Feldgröße (noch) keine optimale Lösung gefunden werden. Unser Ziel war es aber, das **8-Damenproblem** auf einem Quantencomputer zu lösen! Dies gelang uns, indem wir aus dem Hamiltonian des **7-Damenproblems** vier modifizierte Hamiltonians ableiteten, bei denen jeweils bestimmte Diagonalen so unbesetzt bleiben, dass man eine weitere Dame auf das zum 8x8 Feld ergänzte Schachfeld setzen kann.

Das Springerproblem

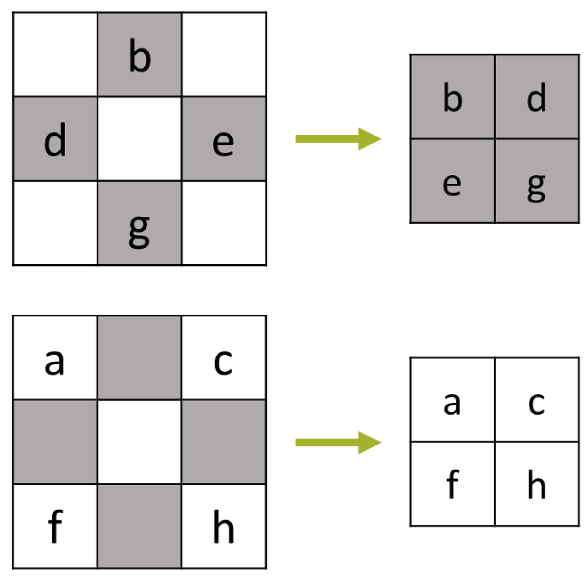
Beim Springerproblem geht es darum, dass ein Springer auf einem n x n Schachfeld eine Route findet, auf der er jedes Feld genau einmal betritt. In unserem Beispiel haben wir, wegen der einfacheren Lösbarkeit des Problems, das 3 x 3 Springerproblem bearbeitet. Um dieses Problem als Hamiltonmatrix darzustellen, 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 auf 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 genau einen „Springerabstand" voneinander entfernt stehen müssen.





Genau ein Springer pro Zeitebene im Springerabstand

Um die Matrix noch weiter zu vereinfachen, reduzierten wir das **3 x 3** auf ein **2 x 2 Feld**. 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 **vier** statt vorher **neun** Felder benötigt. Mit diesem Trick konnten wir schließlich auch das Springerproblem auf dem Quanten Annealer lösen!



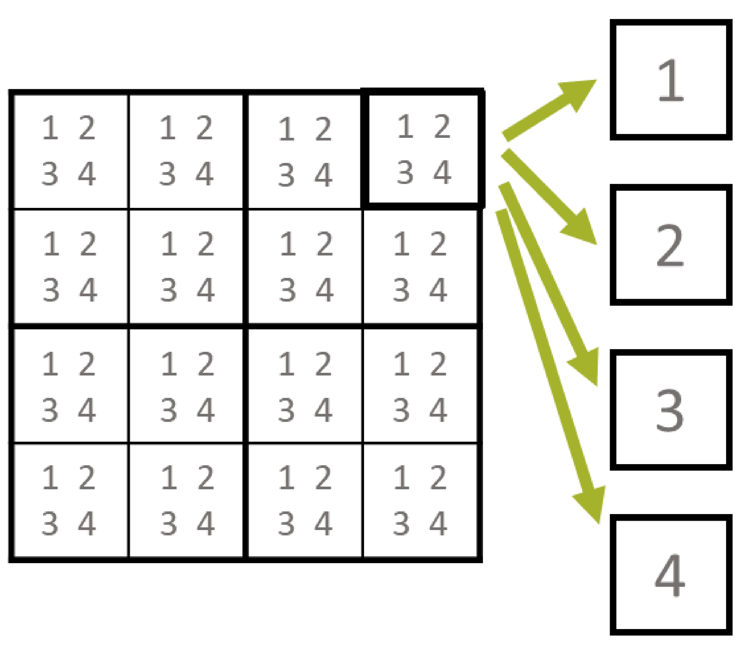
Vom Schach zum Sudoku

Ein Sudoku besteht aus 9 x 9 Feldern, welche in kleinere 3 x 3 Feld-Bereiche unterteilt sind. In jeder Spalte, jeder Zeile und jedem 3 x 3 Bereich müssen die Zahlen von 1-9 genau einmal vorkommen. Außerdem sind bei einem Sudoku immer einige Zahlen vorgegeben, so dass es für die fehlenden Werte nur eine einzige richtige Anordnung gibt.

Das Sudoku als Hamiltonian darzustellen war komplizierter als unsere bisherigen Problemstellungen. So gab es bei den Schachproblemen für jedes Feld nur zwei Möglichkeiten (besetzt oder nicht besetzt), was mit einem Qubit dargestellt werden kann. Nun aber gibt es **9 Möglichkeiten** pro Feld (die Zahlen von 1 bis 9 beim 9 x 9 Sudoku). Daher haben wir das Sudokufeld als **3-dimensionales 9 x 9 x 9 Feld** behandelt, bei welchem die einzelnen Zahlen jeweils von einem eigenen Qubit dargestellt werden. Nun ist aber das Problem, dass ein normales Sudoku auf Grund der Größe des Hamiltonians derzeit auf einem Quantencomputer noch nicht gelöst werden kann.

2		1	
			3
		4	
4	1	3	2

Vorlage für ein 4 x 4 Sudoku



Pro Feld für jede Zahl ein Qubit

Wir haben uns daher auf ein **4 x 4 Sudoku** beschränkt und weitere Vereinfachung des Problems vorgenommen: Da ein vorgegebener Wert in seiner Zeile, seiner Spalte und seinem Kästchen nicht mehr vorkommen darf, kann man alle Qubits, die diesen Wert repräsentieren, an den jeweiligen Stellen einfach **weglassen**.

Für diesen Prozess haben wir ein Programm geschrieben, was das Generieren des Hamiltonians auf Basis der vorgegebenen Werte und das „Zurückrechnen" der Ergebnisse aus dem Quantencomputer automatisiert. Somit konnten wir mit unserem Algorithmus auch noch 4 x 4 Sudokus auf dem Quanten Annealer lösen.

-2	0	2	2	0	0	0	0	0	0	0	0
0	-2	0	0	0	0	0	0	0	0	0	0
0	0	-2	0	0	0	0	2	0	0	0	0
0	0	0	-2	2	0	0	2	0	0	0	0
0	0	0	0	-2	2	0	0	0	2	0	0
0	0	0	0	0	-2	0	0	0	0	2	0
0	0	0	0	0	0	-2	2	0	0	0	0
0	0	0	0	0	0	0	-2	0	0	0	0
0	0	0	0	0	0	0	0	-2	0	0	0
0	0	0	0	0	0	0	0	0	-2	2	0
0	0	0	0	0	0	0	0	0	0	-2	0
0	0	0	0	0	0	0	0	0	0	0	-2

Hamilton-Matrix für ein 4 x 4 Sudoku mit einigen vorbesetzten Werten