Quantencomputing mit D-Wave

AM BEISPIEL VON FÄHRROUTENOPTIMIERUNG

Daniel Baumgärtner	daniel.baumgaertner@nordakademie.de
Anna Ehrenberg	anna.ehrenberg@nordakademie.de
Nick Stuke	nick.stucke@nordakademie.de
Ferdaus Zabihzadeh	ferdaus.zabihzadeh@nordakademie.de

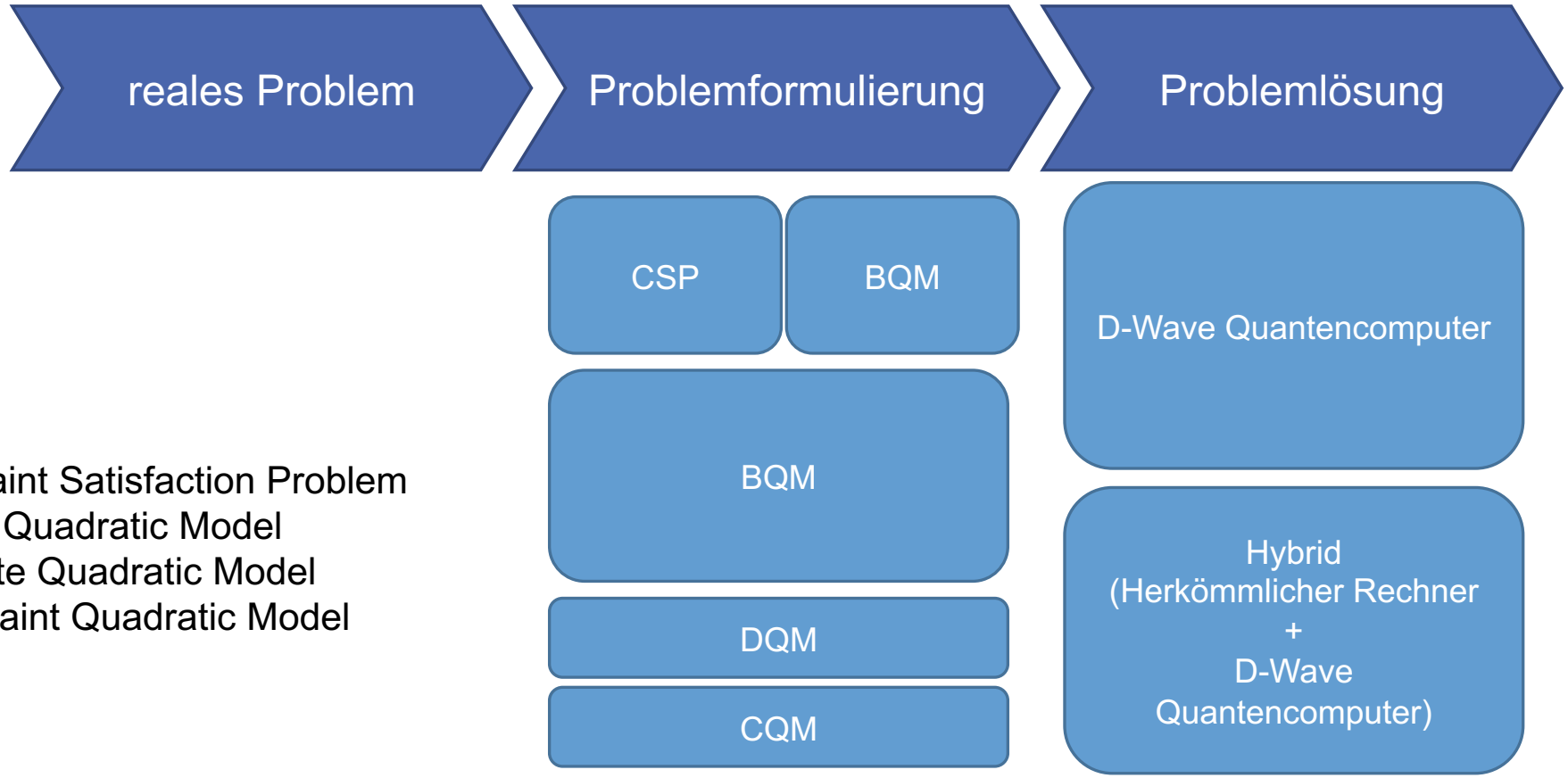
ODER

Agenda

- Konzeption
- Ergebnisse
- Bewertung der Ergebnisse

Konzeption

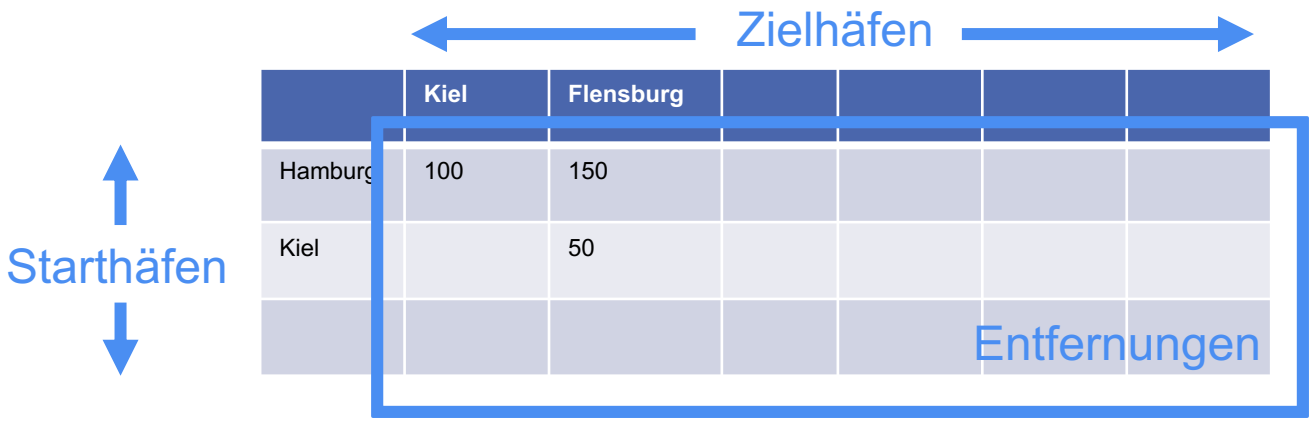
Prozessablauf D-Wave Quantencomputing



CSP = Constraint Satisfaction Problem
BQM = Binary Quadratic Model
DQM = Discrete Quadratic Model
CQM = Constraint Quadratic Model

Problembeschreibung

- Input:
  - Routen zwischen Fährhäfen mit Entfernungen
  - Userinput: Start- und Endhafen
- Output:
  - Kürzeste Route gemessen an Gesamtentfernung
- Datengrundlage:
  - Vorgegebenes Format vorhanden
  - Gewünschte Qualität vorhanden



Programmaufteilung

- Welche Objekte stellen Variablen dar?
  - Häfen
  - Wege

  →Wege in der Form VON-BIS-Distanz
- Wie kann Start und Ende abgebildet werden?
  - Künstlich erschaffene Variablen
- Welches Modell nutzen wir?
  - Test verschiedener Modelle, Erläuterung folgt
- Welchen Sampler nutzen wir?

1. Datenvorverarbeitung: Formatierung

→ VON-BIS-Distanz

→ Start-Hamburg-0
Kiel-Ende-0

2. Problemformulierung

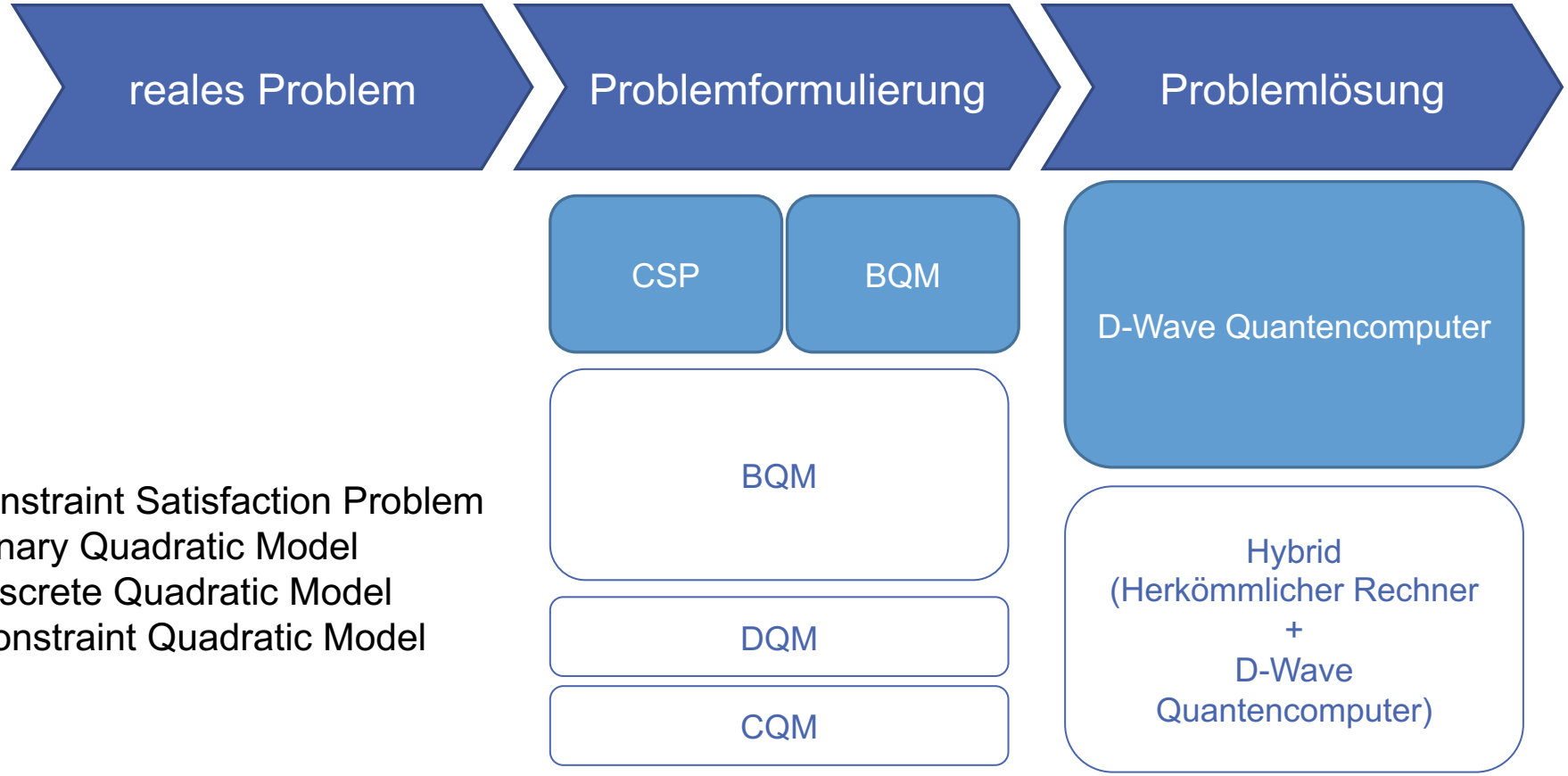
3. Problemlösung

Ergebnisse

Github Code Repository

https://github.com/annaehhh/Quantum-Computing

Variante 1: CSP + BQM + Quantum



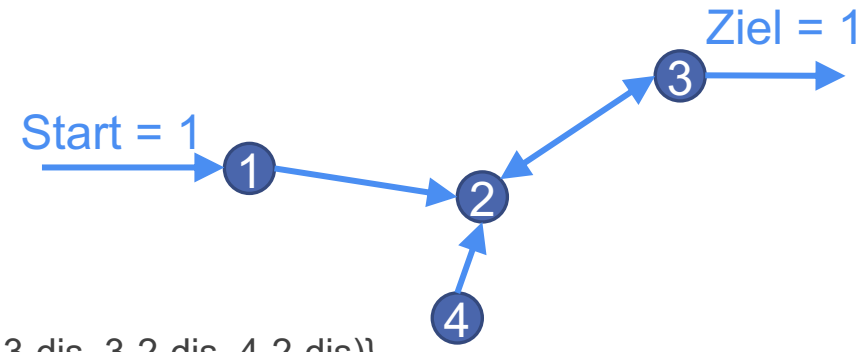
CSP = Constraint Satisfaction Problem
BQM = Binary Quadratic Model
DQM = Discrete Quadratic Model
CQM = Constraint Quadratic Model

Warum?

- Formulierung BQM manuell sehr aufwändig
- BQM als einziges Modell direkt auf QC ausführbar
- Nutzung des Inspectors möglich

Variante 1: Formulierung CSP

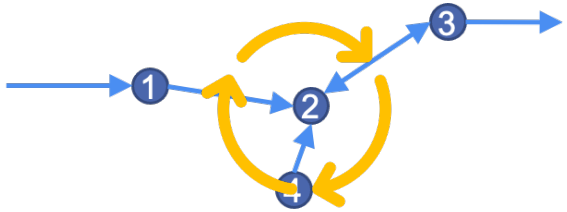
- Wiederholung: alle Wege stellen die binären Variablen dar
- Zunächst werden pro Hafen drei Constraints erstellt:
  - Summe aller Verbindungen = 2 ODER 0
  - Summe aller Hinwege = 1 ODER 0
  - Summe aller Rückwege = 1 ODER 0



- Beispiel für Hafen 2:
  - {(0,0,1,1)(0,1,0,1)(0,1,1,0)(1,0,1,0)(1,0,0,1)(0,1,1,0)(0,0,0,0);(1-2-dis, 2-3-dis, 3-2-dis, 4-2-dis)}
  - {(0,0,1)(0,1,0)(1,0,0)(0,0,0);(1-2-dis, 3-2-dis, 4-2-dis)}
  - {(0)(1);(2-3-dis)}
- Fixieren bekannter Variablen: Start und Ende = 1
  - Deshalb alle Wege = 0 keine Lösung

Variante 1: Umwandlung CSP → BQM

- d-wave Funktion stitch(): Erstellung vollständiges BQM
  - Iteration über vollständige Graphen, bis das Problem abgebildet werden kann



- Begrenzung: Parameter max_graph_size <= 8
  - Anzahl der Variablen pro Constraint begrenzt auf 8 Stück
  - ein Hafen darf nicht mehr als 8 Hin- und Rückwege (inkl. aux-Variablen), da 3 aux-Variablen
  - Aux_variables = Differenz der Knotenanzahl – Anzahl realer Variablen
  - Beispieldatensatz: Bremerhaven funktioniert nicht

- Überarbeitung der stitch()-Funktion: Erweiterung auf bis zu 32 Variablen
  - Ergebnis: Beispieldaten ergeben sinnvolle Routen

- BQM: {(lineare Terme),(quadratische Terme}

	Column1	Bremerhaven	Brunsbüttel	Emden	Hamburg	Kiel
0	Bremerhaven	0	NaN	137.0	50.0	NaN
1	Brunsbüttel	81	0.0	NaN	36.0	54.0
2	Emden	137	NaN	0.0	NaN	NaN
3	Hamburg	117	36.0	NaN	0.0	90.0
4	Kiel	135	NaN	NaN	NaN	0.0

Variante 1: Umwandlung CSP → BQM

- Erweiterung der Daten auf Originaldatensatz:
  - Kein Ergebnis innerhalb von rund 16h
- Weitere Analyse stitch()-Funktion:
  - Binäre Codierung aller Variablen mithilfe numpy-Funktion in Matrix
  - Anschließendes Durchlaufen einer Schleife
  - Bsp. Bremerhaven mit 6 Variablen: 2^8 = 256 Durchläufe
  - Bsp. Vergrößerung Funktion: 2^32= 4,29 Mrd. Durchläufe
  - Laufzeit $O(2^n)$ = exponentiell
- Maximal erreichte Graphengröße: 11 Knoten
  - Kein Ergebnis mit Originaldaten

Beispieldatensatz:

	Column1	Bremerhaven	Brunsbüttel	Emden	Hamburg	Kiel
0	Bremerhaven	0	NaN	137.0	50.0	NaN
1	Brunsbüttel	81	0.0	NaN	36.0	54.0
2	Emden	137	NaN	0.0	NaN	NaN
3	Hamburg	117	36.0	NaN	0.0	90.0
4	Kiel	135	NaN	NaN	NaN	0.0

Originaldatensatz:

	Column1	Bremerhaven	Brunsbüttel	Emden	Hamburg	Kiel	Lübeck	Rostock	Sassnitz	Stralsund	...	St. Petersburg	Gothenb
0	Bremerhaven	0.0	NaN	137.0	NaN	135.0	NaN	NaN	NaN	NaN	...	NaN	N
1	Brunsbüttel	81.0	0.0	NaN	36.0	54.0	NaN	NaN	NaN	NaN	...	NaN	N
2	Emden	137.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	N
3	Hamburg	117.0	36.0	NaN	0.0	90.0	187.0	174.0	145.0	NaN	...	NaN	N
4	Kiel	135.0	NaN	NaN	NaN	0.0	NaN	NaN	NaN	109.0	...	NaN	N
5	Lübeck	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN	93.0	...	NaN	N
6	Rostock	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN	...	NaN	N
7	Sassnitz	NaN	NaN	NaN	NaN	145.0	NaN	92.0	0.0	NaN	...	NaN	N
8	Stralsund	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	...	NaN	N
9	Wilhelmshaven	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	N
10	Wismar	585.0	140.0	NaN	176.0	86.0	NaN	NaN	NaN	NaN	...	NaN	N
11	Antwerpen	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	N
12	Rotterdam	255.0	NaN	NaN	NaN	323.0	NaN	NaN	NaN	NaN	...	NaN	N
13	Aarhus	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	N
14	Copenhagen	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	13
15	Bornholm	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	N
16	Gdansk	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	N
17	Klaipeda	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	N

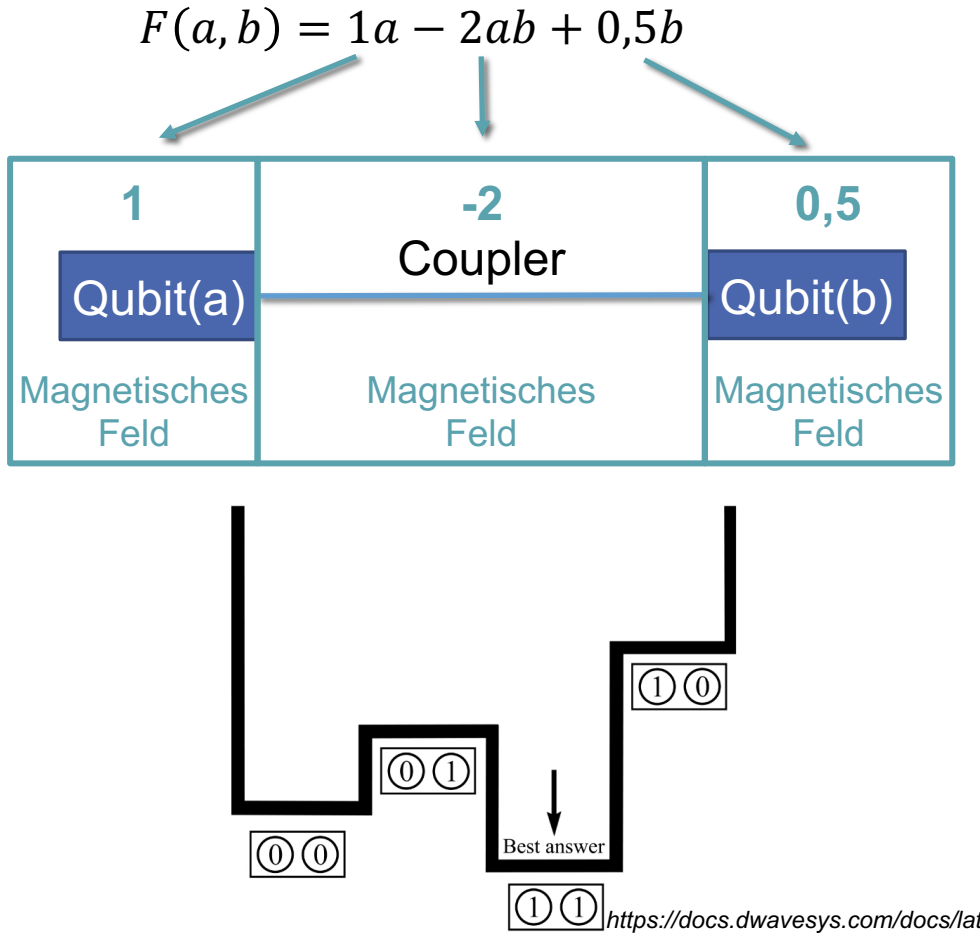
Quantencomputing

- Nutzung D’Wave Quantencomputer
  - Quantenannealing-Prinzip
  - Freier Zugang auf QC-Ressourcen
  - Bereitstellung von Python-Packages zur Nutzung der QC-Ressourcen
  - Gute Dokumentation für Starter
    - Beispiele/Tutorials vorhanden

Quantenannealing

- Das Herzstück des Quantencomputers sind die **Qubits**, die mit **Couplern** miteinander verbunden sind
- Die Qubits befinden sich in einem unbekannten Zustand ohne äußere Einwirkung
- Mit den Werten aus dem BQM wird das **Magnetfeld** für jeden Qubit und Coupler eingestellt
- Die Qubits versuchen den möglichst **niedrigsten Energiezustand** einzunehmen

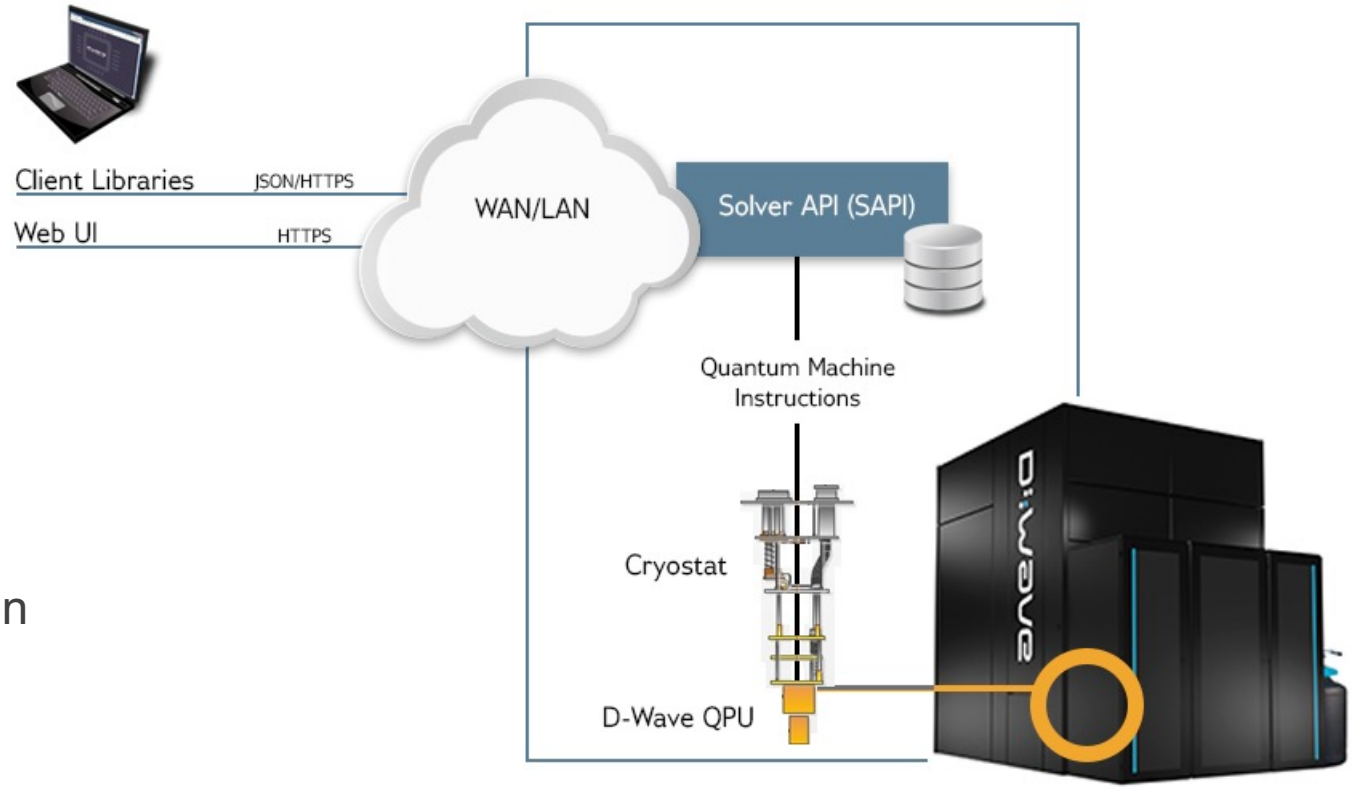
→ Quantenannealing-Prozess liefert den global niedrigsten Energiezustand der Eingabefunktion *(oder nahegelegende Zustände)*



https://docs.dwavesys.com/docs/latest/c_gs_2.html

D‘Wave Sampler

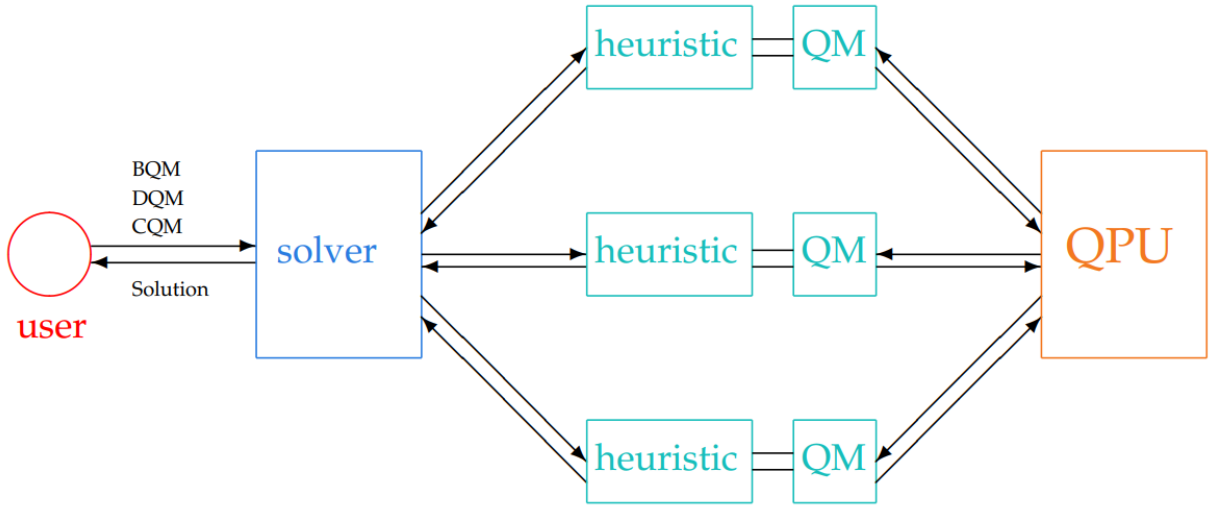
- Schnittstelle zwischen Code und QC
- Hat folgende Ausprägungen
  - Quantencomputing
  - Klassik
  - Hybrides Modell
- Ruft nach Vorverarbeitung die definierte QC-Ressource auf
  - Solver sind die Schnittstelle zu den eigentlichen Quantencomputer-Ressourcen
- Liefert eine aufbereitete Lösung zum Input-Problem



→ **Ermöglicht Nutzung von Quantencomputer ohne tiefere Kenntnisse in Quantenmechanik**

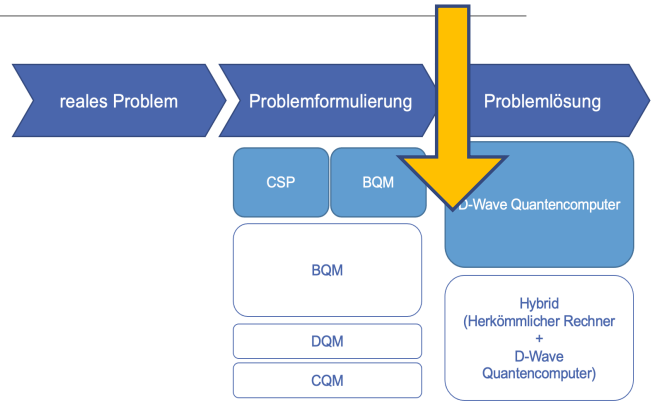
Hybrider Sampler

- Kombination aus klassischen und QC Ressourcen
  - Mehrere Anfragen oder keine an QC möglich
  - Aufteilung großer Probleme in kleine Teile
- Andere Modelle wie beim direkten Zugriff auf den QC möglich
  - CQM Modell kann Integer-Werte als Lösung ausgeben
  - DQM-Modell gibt diskrete-Werte als Lösung zurück

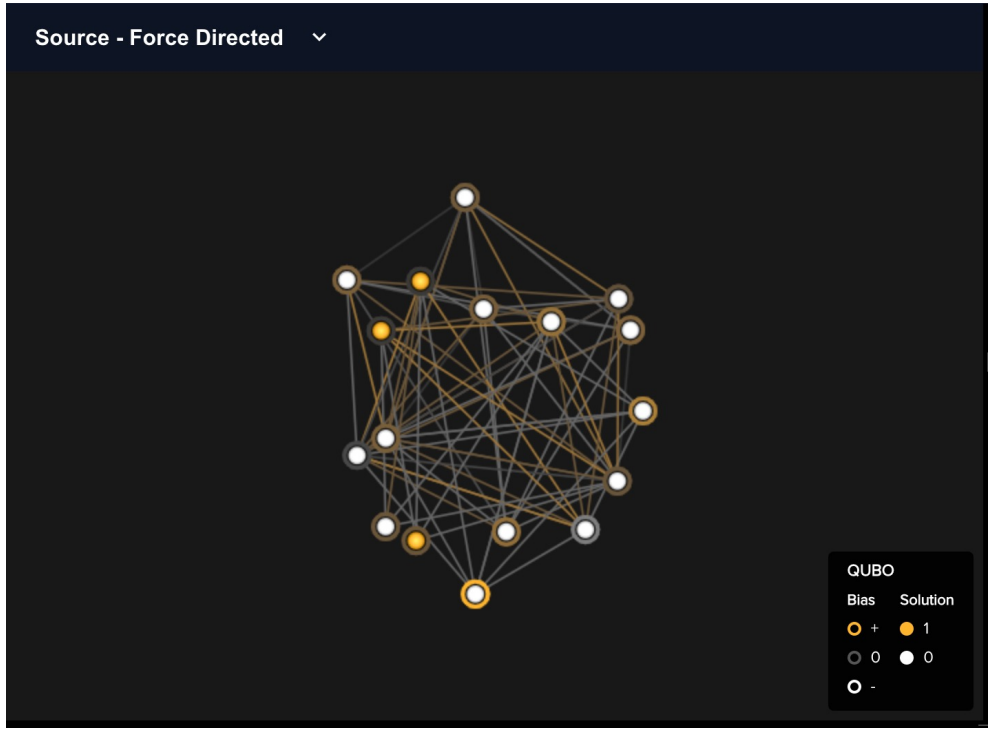


Variante 1: Ausführung QC

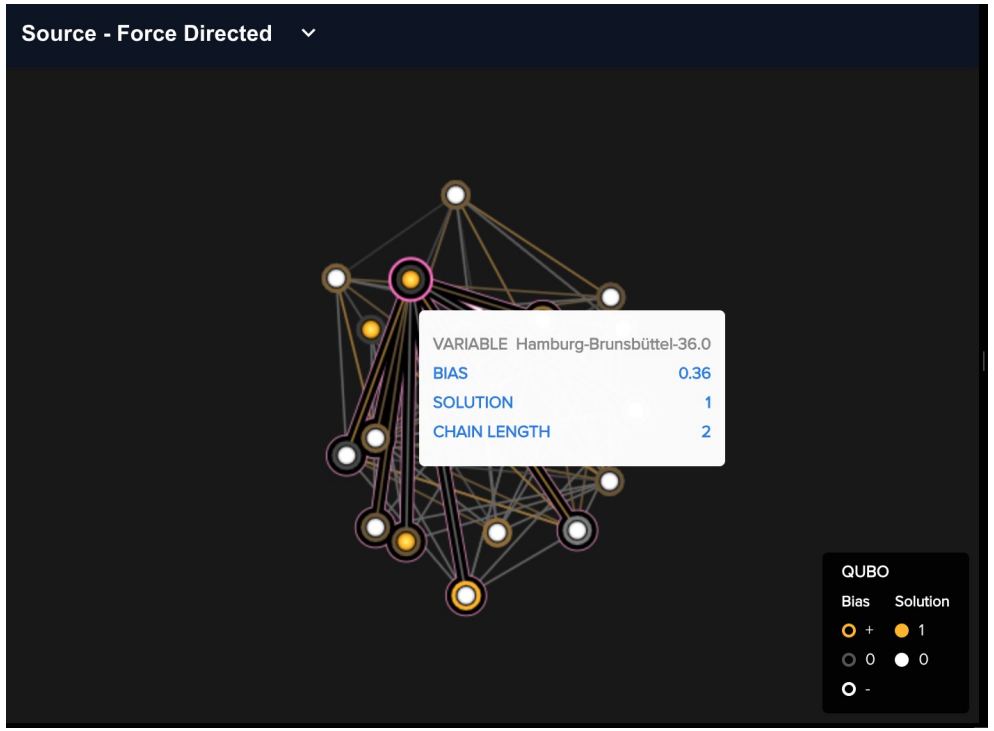
- Wiederholung: Ergebnis BQM erfolgreich für Beispieldatensatz
- Bisher gewichten alle Wege gleich → einfügen der Distanzen als Gewichtung in den Bias
- Weitergabe an den Sampler
- Ergebnis des Samplers im Inspector



Variante 1: Inspector Variables

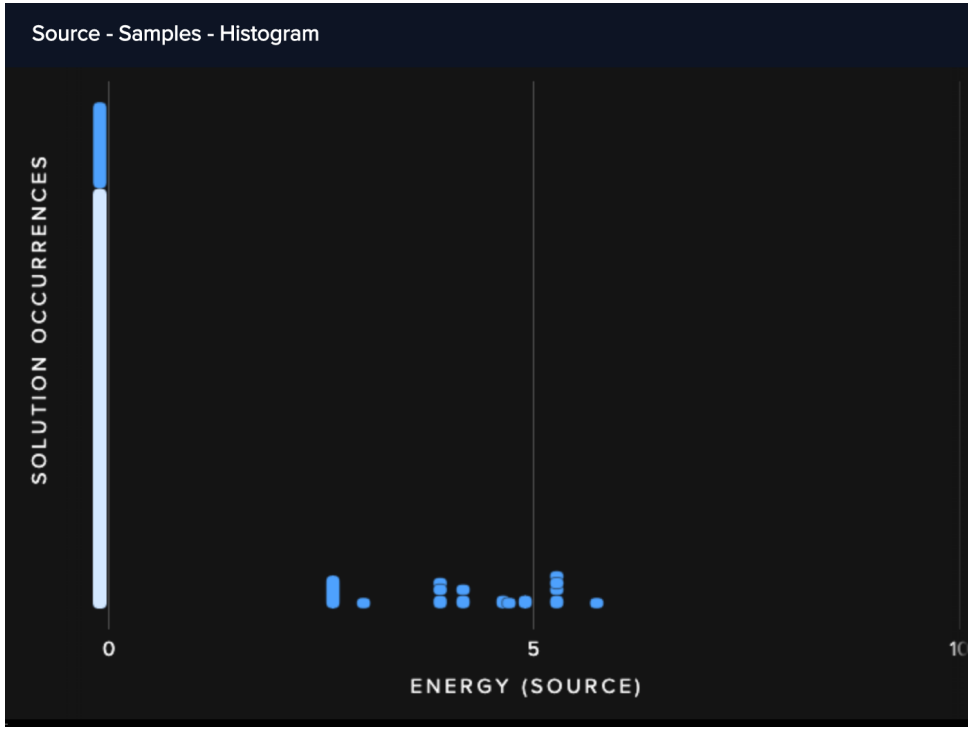


- Insgesamt 17 Variablen:
  - 10 reale Objekte (da insgesamt 10 Verbindungen)
  - 7 aux-Variablen

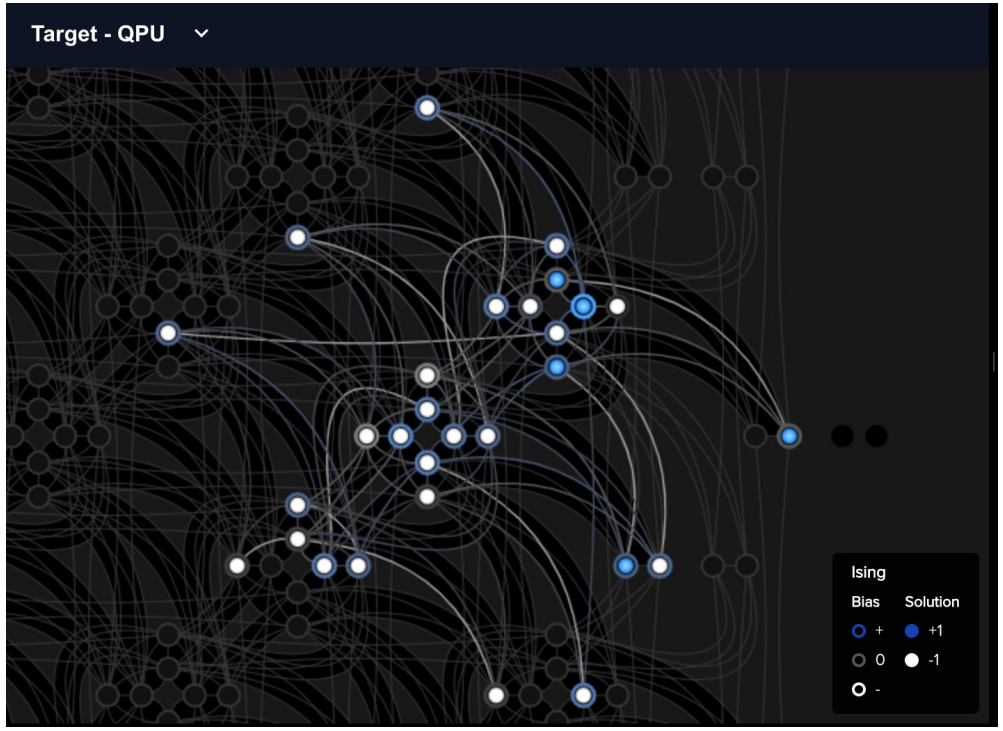


- Hier Lösung Hamburg-Kiel:
  - Über Brunsbüttel

Variante 1: Inspector Energy + QPU

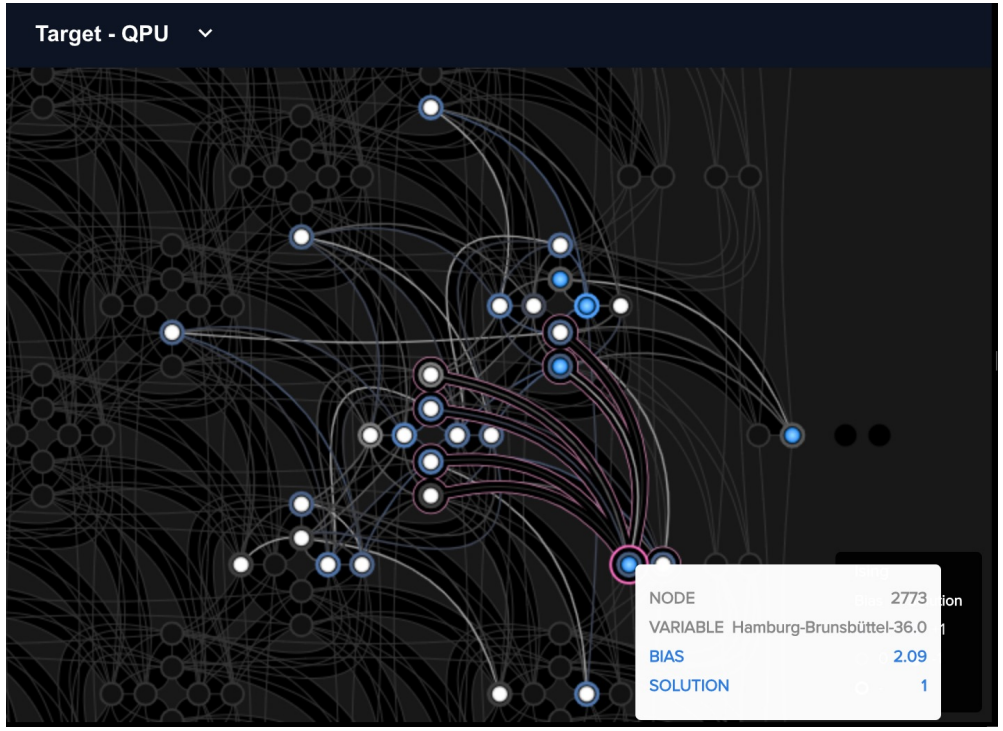


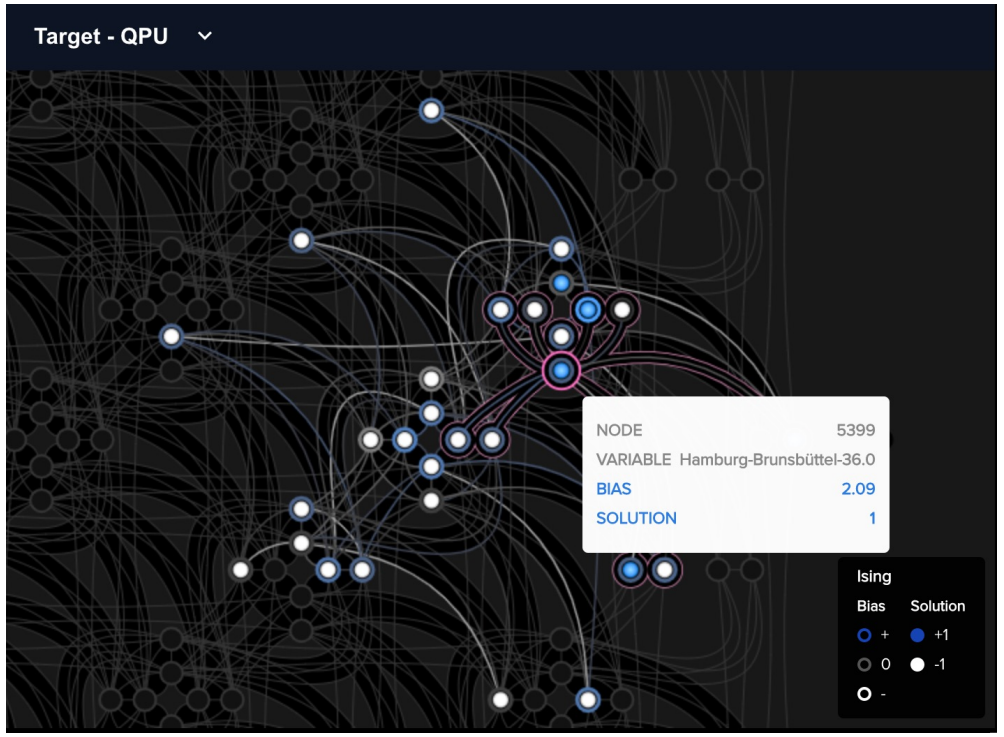
- alle Lösungen nach Energie aufsteigend:
  - Lösung mit niedrigster Energie auf vorheriger Folie



- Mapping und Verschränkung von Qubits:
  - 29 Qubits

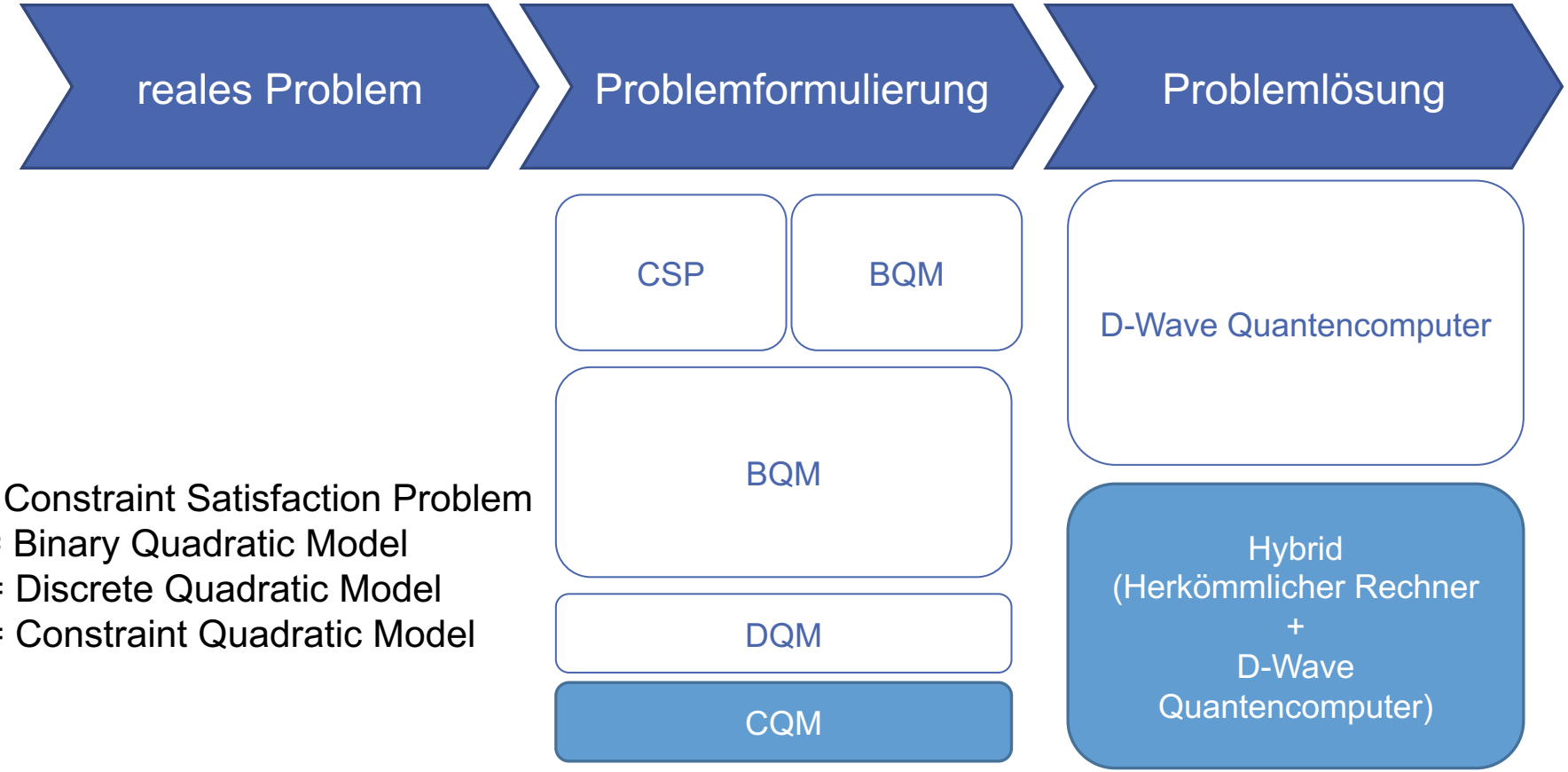
Variante 1: Inspector QPU





- Begründung:
  - einzelne Variablen werden mit mehr als einem Qubit abgebildet

Variante 2: CQM + Hybrid



CSP = Constraint Satisfaction Problem
BQM = Binary Quadratic Model
DQM = Discrete Quadratic Model
CQM = Constraint Quadratic Model

Warum?

- andere Problemformulierung
- Umfangreichstes Modell
- Nutzung des hybriden Workflows

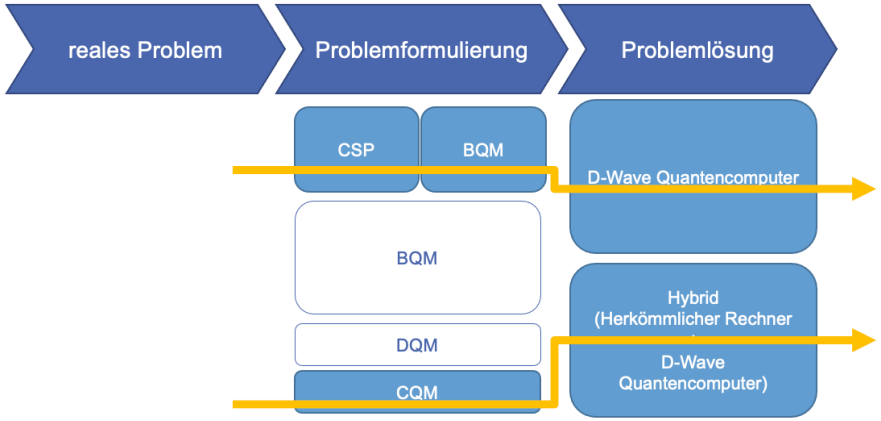
Variante 2: Formulierung CQM

- Problem wird in zwei Teilen angegeben:
  - Optimierungsfunktion
    - Was soll minimiert/maximiert werden?
    - Höchster Preis, kürzeste Strecke
  - Constraints
    - Welche Abhängigkeiten sollen eingehalten werden?
    - Neben harten Abängigkeiten gibt es auch weiche, die in Abhängigkeit zur Optimierungsfunktion eingebracht werden.

Modellvergleich

- Warum haben wir diese zwei Wege gewählt
  - Ausgabe: Welche Route wird genutzt? → binäres Ergebnis
  - DQM wäre entweder als binäres Problem definiert(M={0,1}) oder es würde eine Variable mit allen Routenmöglichkeiten geben(schlechte Skalierung)
  - BQM(über CSP) und CQM stellen zwei unterschiedliche Definitionswege dar
- Nutzung von „direkten“ Quantencomputing und hybrider Workflows von D‘Wave

→ BQM ermöglicht die volle Kontrolle und maximale individuelle Verarbeitung. Bei CQM passiert viel im nicht einsehbaren Backend.



Alternative: Dijkstra-Algorithmus

- Algorithmus zum Finden des kürzesten Weges
- Übliche in der Praxis angewendete Methode für das Finden der korrekten Lösung
- Komplexität bei Nutzung eines Arrays Worstcase: $O(|E| + |V|^2)$
  - E: Anzahl Routen, V: Anzahl Häfen

```
Q=[Starthafen]

d={(h: kürzeste Distanz(kd)=∞,
        nächster Hafen(nH)=none) für jeden Hafen h}

Solange Q nicht leer:
    h=Q.Erster Eintrag(inkl. Löschen)
    für jeden Nachbarn n von h:
        wenn d[h][kd]+Distanz(n->h) < d[n][kd]:
            d[n]={kd= d[h][kd]+Distanz(n->h), nH=h}
            Q.fügeHinzu(n)
```

Zusammenfassung

- Problemlösung mit Quantencomputer (CSP,CQM) und klassischen Computerressourcen (Dijkstra)
- Problem mit gegebenen Datenumfang konnte in kurzer Zeit mit CQM und Dijkstra gelöst werden
  - Ein Zeitunterschied war nicht erkennbar

- CSP hatte durch das Umwandeln mit der Stitch-Funktion Probleme mit dem Datenumfang
  - Kleinere Datenmenge war möglich
  - Herstellerseitig sind große Probleme nicht abdeckt (nicht dokumentiert!)
    → Nach Fix: Laufzeitschwierigkeiten

Bewertung der Ergebnisse

Bewertung Ergebnisse

- Quantencomputer ist jetzt für Entwickler nutzbar
- D‘Wave hat eine gute API zur Nutzung entworfen, ohne tiefere physikalische Kenntnisse zu besitzen
  - Bsp.: BQM Werte für die Magnetfelder werden automatisch umgerechnet
- Allerdings:
  - Problemdefinition zeigt sich schwierig
    - Hilfsmittel von D‘Wave wie CSP sind nicht immer nutzbar
  - Problemgröße ist beim klassischen Quantencomputing begrenzt
    - Nutzung von D‘Waves hybriden Konzept
  - Einschränkung auf einen Anbieter und Nische
    - Dokumentation und Community sehr eingeschränkt vorhanden
    - Vendor Lockin?

D‘Wave als Anbieter

- Wenige Firmen, die Quantum-Computer mit Quantenannealing umsetzen
- Keine Standards → Wechsel sehr schwierig
  - Insbesondere im Bereich hybriden „Black Box“-Modell
- Datenschutz? → Quantencomputer teilweise außerhalb der EU
- Dokumentation und Hilfe gibt es derzeit nur von D‘Wave
- Kostenmodell unbekannt
  - Business Case für das Tragen ggf. hoher Kosten?

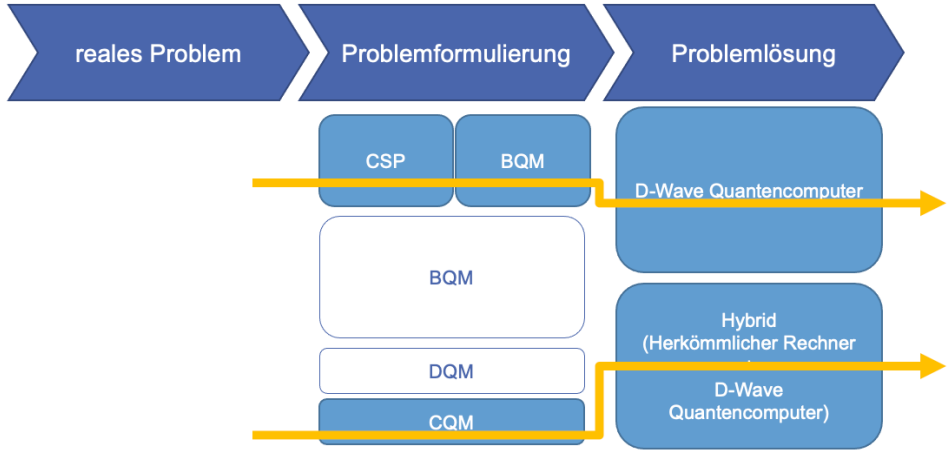
Bewertung Ergebnisse

- Dokumentation Herangehensweise für Laien: insgesamt sehr ausführlich
  - Viele kleine, leicht verständliche Beispiele (Mathematik, Code)
  - keine größeren, realistischen Beispiele, für die ein QC benötigt würde

- technische Dokumentation
  - bereitgestellte Funktionen nicht ausreichend dokumentiert
  - Bsp. Parameter max_graph_size in stitch()
  - Funktion besitzt Default=8, allerdings wird nicht erwähnt, dass größere Modelle nicht möglich sind
  - Problem bekannt: in der Funktion selbst sind Entwickler-Kommentare, die das Problem beschreiben

Fazit

- Was haben wir versucht?
- Was hat funktioniert? Ist das genug?
  - BQM mit geringen Datengröße, limitiert durch CSP
  - CQM mit vollständiger Datenmenge
- Was hätten wir noch versuchen können?
  - Constraints anders definieren
  - BQM ohne CSP erstellen
  - zum Vergleich: BQM auf hybridem Rechner ausführen
  - Größere Datenmenge für CQM und Dijkstra testen



Fazit

- 1. Frage: Funktioniert QC mit Fährhäfen?
  - Ja
  - Aber: sehr begrenzte Größe zum Testen
  - Aber: nur mit CQM, dort unersichtlich, was tatsächlich auf QC gerechnet wird
  - Schwierigkeit liegt in der Problemformulierung

- 2. Frage: Ist QC praktisch anwendbar?
  - Nicht zu beantworten
  - Erst einmal ja, für kleine Probleme
  - Relevanz von Quantencomputing: Problemtypen die exponentiell skallieren und große Datenmengen
  - Da unsere Problemgröße nicht realistisch ist, keine Aussage diesbezüglich möglich
  - Weitere Tests notwendig

Vielen Dank für Ihre Aufmerksamkeit!