

Quantum annealing und Logistik

Yanick Kind

12. Juli 2023

Fakultät Physik

Übersicht

Einleitung

Simulated und quantum annealing

Quadratic unconstrained binary optimization

Hardware embedding

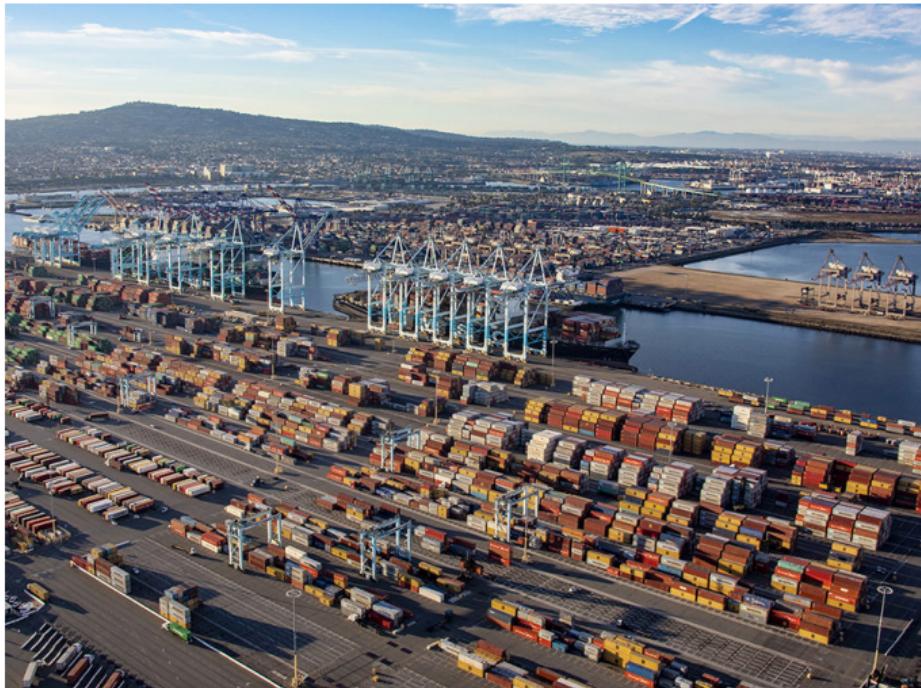
Fallbeispiel Aisin Corporation

Zusammenfassung und Ausblick



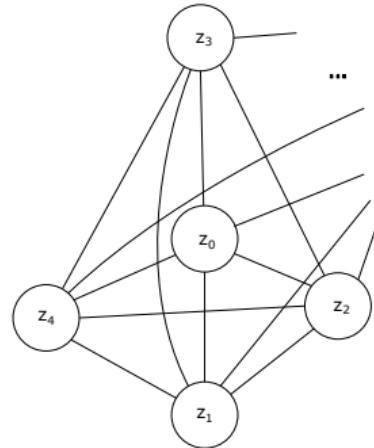
azom.com

Motivation



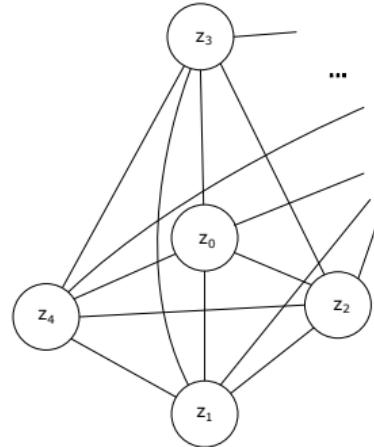
supplychainbrain.com

Traveling Salesman Problem



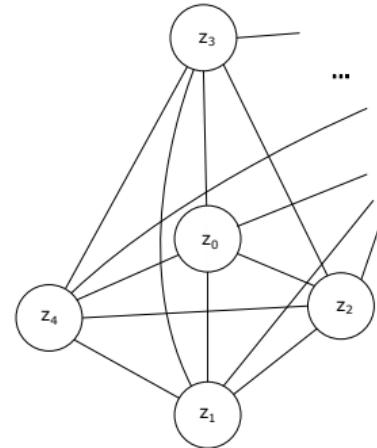
Traveling Salesman Problem

- klassisches kombinatorisches Optimierungsproblem



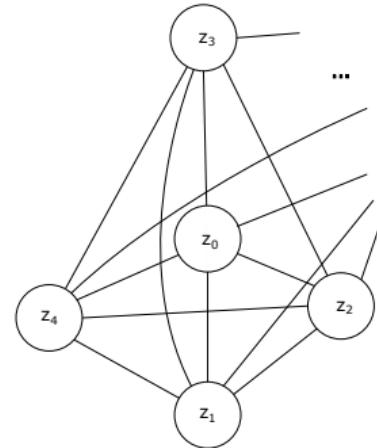
Traveling Salesman Problem

- klassisches kombinatorisches Optimierungsproblem
- Beispiel: Ein Lieferwagen hat n Lieferorte



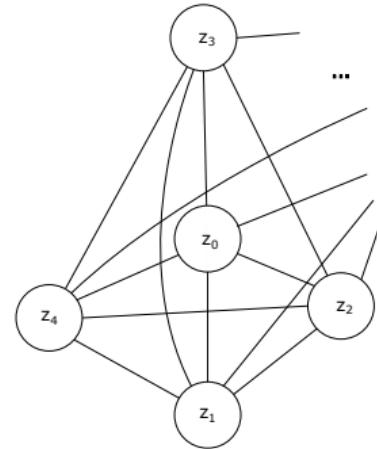
Traveling Salesman Problem

- klassisches kombinatorisches Optimierungsproblem
- Beispiel: Ein Lieferwagen hat n Lieferorte
 - $(n - 1)!$ verschiedene Routen



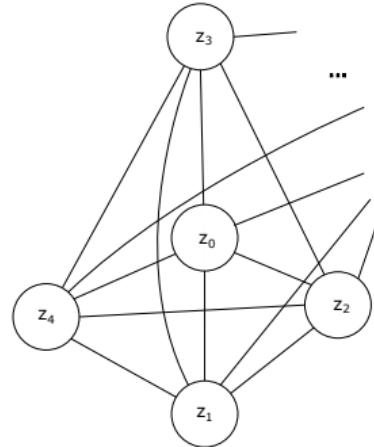
Traveling Salesman Problem

- klassisches kombinatorisches Optimierungsproblem
- Beispiel: Ein Lieferwagen hat n Lieferorte
 - $(n - 1)!$ verschiedene Routen
- NP-hard
 - skaliert nicht polynomiell sondern **exponentiell** mit Systemgröße



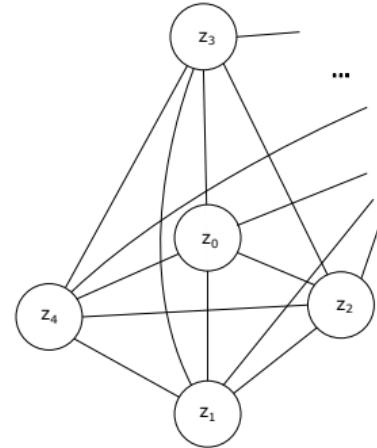
Traveling Salesman Problem

- klassisches kombinatorisches Optimierungsproblem
- Beispiel: Ein Lieferwagen hat n Lieferorte
 - $(n - 1)!$ verschiedene Routen
- NP-hard
 - skaliert nicht polynomiell sondern **exponentiell** mit Systemgröße
- Globales Optimum mittels numerischen Lösungsverfahren finden, ist extrem aufwändig

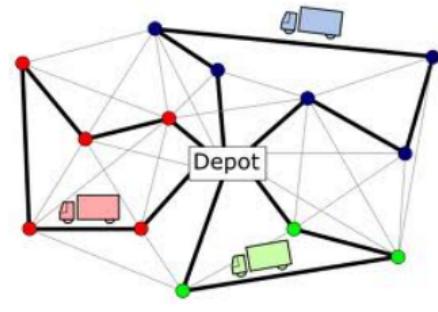
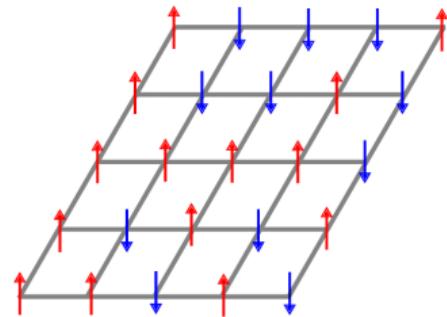


Traveling Salesman Problem

- klassisches kombinatorisches Optimierungsproblem
- Beispiel: Ein Lieferwagen hat n Lieferorte
 - $(n - 1)!$ verschiedene Routen
- NP-hard
 - skaliert nicht polynomiell sondern **exponentiell** mit Systemgröße
- Globales Optimum mittels numerischen Lösungsverfahren finden, ist extrem aufwändig
- Local search Algorithmen als heuristische Näherungsverfahren nutzen



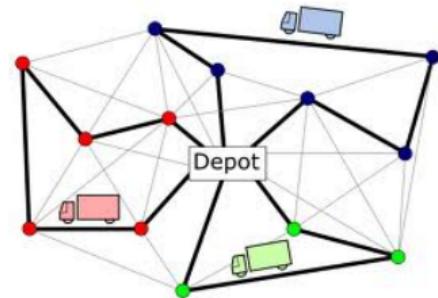
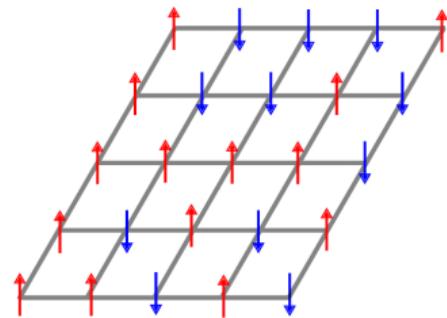
Annealing als Lösung?



r-bloggers.com

Sascha Wald, Thermalisation and Relaxation of Quantum Systems (2017)

Annealing als Lösung?

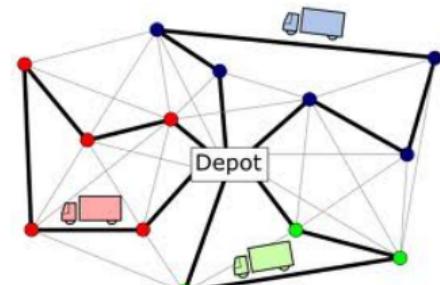
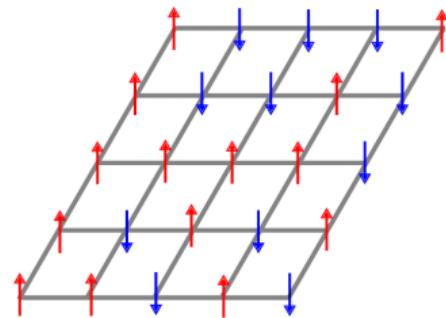


r-bloggers.com

Sascha Wald, Thermalisation and Relaxation of Quantum Systems (2017)

E

Annealing als Lösung?



r-bloggers.com

E

f

Simulated annealing (SA)

Simulated annealing (SA)

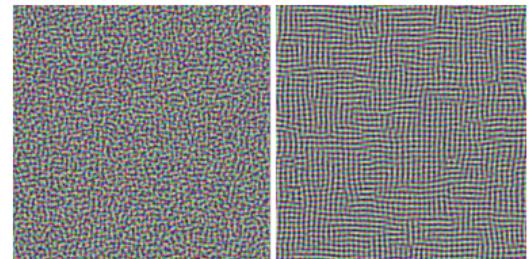
- in Analogie zu auskühlenden Metallen

Simulated annealing (SA)

- in Analogie zu auskühlenden Metallen
 - schneller auskühlen → suboptimaler Zustand (amorph)
 - langsamer auskühlen → optimaler Zustand (kristallin)

Simulated annealing (SA)

- in Analogie zu auskühlenden Metallen
 - schneller auskühlen → suboptimaler Zustand (amorph)
 - langsamer auskühlen → optimaler Zustand (kristallin)

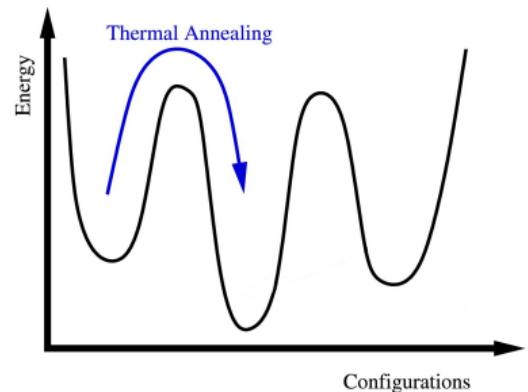


Cyp, wikipedia.org

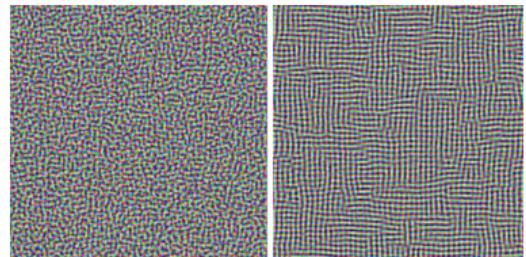
Simulated und quantum annealing

Simulated annealing (SA)

- in Analogie zu auskühlenden Metallen
 - schneller auskühlen → suboptimaler Zustand (amorph)
 - langsamer auskühlen → optimalerer Zustand (kristallin)
- nutzt thermische Fluktuationen, um lokalen Minima zu entkommen



V. Bapst et al., Physics Reports 523 (2013), editiert

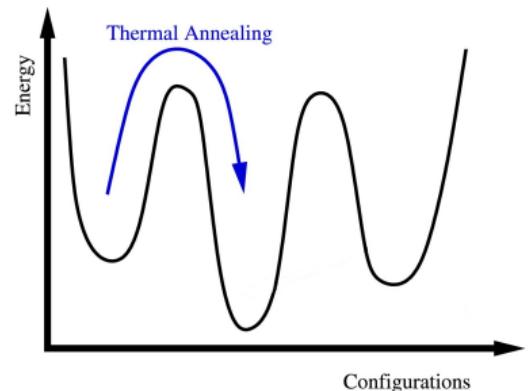


Cyp, wikipedia.org

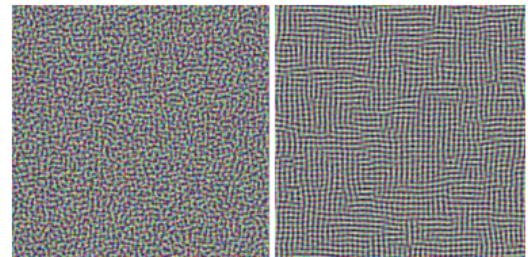
Simulated und quantum annealing

Simulated annealing (SA)

- in Analogie zu auskühlenden Metallen
 - schneller auskühlen → suboptimaler Zustand (amorph)
 - langsamer auskühlen → optimalerer Zustand (kristallin)
- nutzt thermische Fluktuationen, um lokalen Minima zu entkommen
- Wahrscheinlichkeit P_i folgen Mastergleichung $\frac{dP_i}{dt} = \sum_j L_{ij} P_j$



V. Bapst et al., Physics Reports 523 (2013), editiert



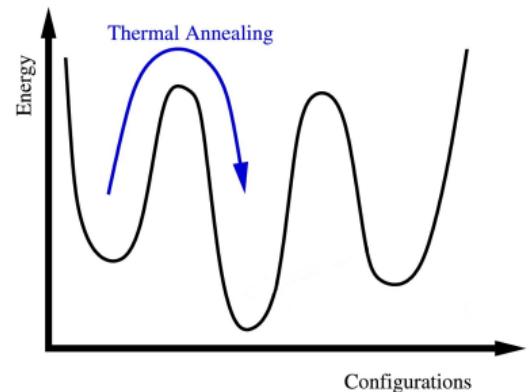
Cyp, wikipedia.org

Simulated und quantum annealing

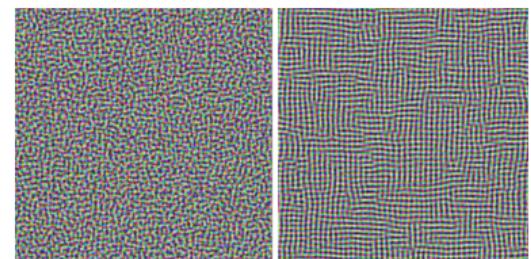
Simulated annealing (SA)

- in Analogie zu auskühlenden Metallen
 - schneller auskühlen → suboptimaler Zustand (amorph)
 - langsamer auskühlen → optimalerer Zustand (kristallin)
- nutzt thermische Fluktuationen, um lokalen Minima zu entkommen
- Wahrscheinlichkeit P_i folgen Mastergleichung $\frac{dP_i}{dt} = \sum_j L_{ij} P_j$

$$L_{ij} = \begin{cases} \left[1 + \exp((E_i - E_j)/T_t)\right]^{-1} & \text{einzelne Spin-Differenz} \\ -\sum_{k \neq i} L_{ki} & i = j \\ 0 & \text{sonst} \end{cases}$$



V. Bapst et al., Physics Reports 523 (2013), editiert



Cyp, wikipedia.org

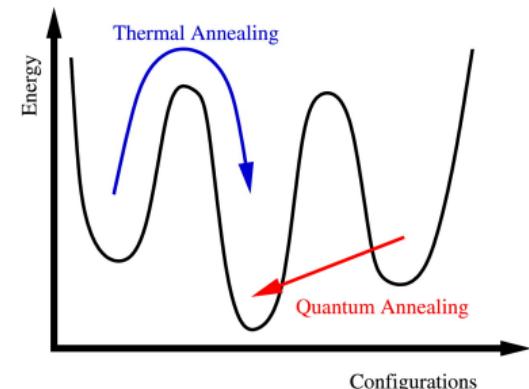
Quantum Annealing (QA)

Quantum Annealing (QA)

- transversales Ising-Modell $H(t) = \underbrace{-\sum_{ij} J_{ij} \sigma_i^z \sigma_j^z}_{H_f} - h \sum_i \sigma_i^z - \Gamma(t) \sum_i \sigma_j^x$

Quantum Annealing (QA)

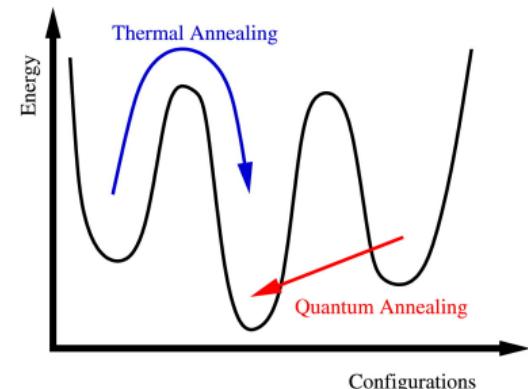
- transversales Ising-Modell $H(t) = \underbrace{-\sum_{ij} J_{ij} \sigma_i^z \sigma_j^z}_{H_f} - h \sum_i \sigma_i^z - \Gamma(t) \sum_i \sigma_i^x$
- $H_i(t)$ verursacht Tunneln zwischen Eigenzuständen von H_f



V. Bapst et al., Physics Reports 523 (2013)

Quantum Annealing (QA)

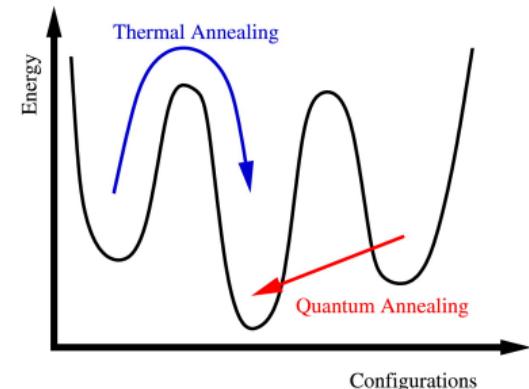
- transversales Ising-Modell $H(t) = \underbrace{-\sum_{ij} J_{ij} \sigma_i^z \sigma_j^z}_{H_f} - h \sum_i \sigma_i^z - \Gamma(t) \sum_i \sigma_i^x$
- $H_i(t)$ verursacht Tunneln zwischen Eigenzuständen von H_f
 - schnellere Konvergenz



V. Bapst et al., Physics Reports 523 (2013)

Quantum Annealing (QA)

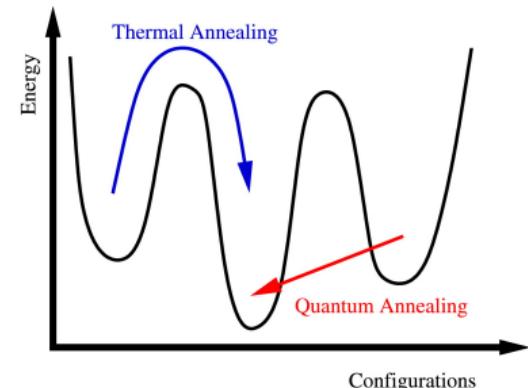
- transversales Ising-Modell $H(t) = \underbrace{-\sum_{ij} J_{ij} \sigma_i^z \sigma_j^z}_{H_f} - h \sum_i \sigma_i^z - \Gamma(t) \sum_i \sigma_i^x$
- $H(t)$ verursacht Tunneln zwischen Eigenzuständen von H_f
→ schnellere Konvergenz
- $|\Psi(t)\rangle$ durch Schrödingergleichung $i \frac{\partial |\Psi(t)\rangle}{\partial t} = H(t)|\Psi(t)\rangle$ festgelegt



V. Bapst et al., Physics Reports 523 (2013)

Quantum Annealing (QA)

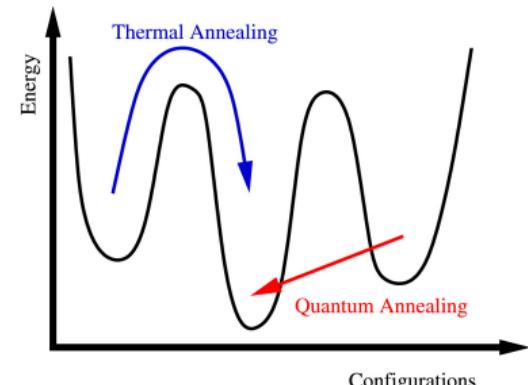
- transversales Ising-Modell $H(t) = \underbrace{-\sum_{ij} J_{ij} \sigma_i^z \sigma_j^z}_{H_f} - h \sum_i \sigma_i^z - \Gamma(t) \sum_i \sigma_i^x$
- $H_i(t)$ verursacht Tunneln zwischen Eigenzuständen von H_f
 - schnellere Konvergenz
- $|\Psi(t)\rangle$ durch Schrödingergleichung $i \frac{\partial |\Psi(t)\rangle}{\partial t} = H(t)|\Psi(t)\rangle$ festgelegt
- $\lim_{t \rightarrow 0} \Gamma(t) = \infty \rightarrow \lim_{t \rightarrow 0} H(t) = H_i(0)$



V. Bapst et al., Physics Reports 523 (2013)

Quantum Annealing (QA)

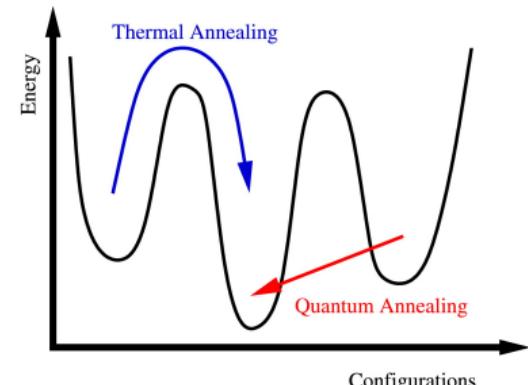
- transversales Ising-Modell $H(t) = \underbrace{-\sum_{ij} J_{ij} \sigma_i^z \sigma_j^z}_{H_f} - h \sum_i \sigma_i^z - \Gamma(t) \sum_i \sigma_i^x$
- $H_i(t)$ verursacht Tunneln zwischen Eigenzuständen von H_f
 - schnellere Konvergenz
- $|\Psi(t)\rangle$ durch Schrödingergleichung $i \frac{\partial |\Psi(t)\rangle}{\partial t} = H(t)|\Psi(t)\rangle$ festgelegt
- $\lim_{t \rightarrow 0} \Gamma(t) = \infty \rightarrow \lim_{t \rightarrow 0} H(t) = H_i(0)$
 - Superposition aller Zustände gleicher Amplitude



V. Bapst et al., Physics Reports 523 (2013)

Quantum Annealing (QA)

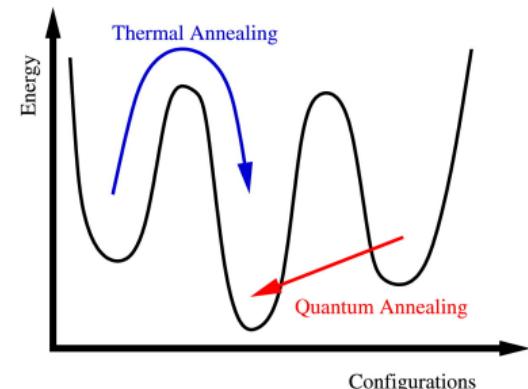
- transversales Ising-Modell $H(t) = \underbrace{-\sum_{ij} J_{ij} \sigma_i^z \sigma_j^z}_{H_f} - h \sum_i \sigma_i^z - \Gamma(t) \sum_i \sigma_i^x$
- $H_i(t)$ verursacht Tunneln zwischen Eigenzuständen von H_f
 - schnellere Konvergenz
- $|\Psi(t)\rangle$ durch Schrödingergleichung $i \frac{\partial |\Psi(t)\rangle}{\partial t} = H(t)|\Psi(t)\rangle$ festgelegt
- $\lim_{t \rightarrow 0} \Gamma(t) = \infty \rightarrow \lim_{t \rightarrow 0} H(t) = H_i(0)$
 - Superposition aller Zustände gleicher Amplitude
- wähle monoton fallende Funktion für $\Gamma(t)$ („ausglühen“)



V. Bapst et al., Physics Reports 523 (2013)

Quantum Annealing (QA)

- transversales Ising-Modell $H(t) = \underbrace{-\sum_{ij} J_{ij} \sigma_i^z \sigma_j^z}_{H_f} - h \sum_i \sigma_i^z - \Gamma(t) \sum_i \sigma_i^x$
- $H_i(t)$ verursacht Tunneln zwischen Eigenzuständen von H_f
 - schnellere Konvergenz
- $|\Psi(t)\rangle$ durch Schrödingergleichung $i \frac{\partial |\Psi(t)\rangle}{\partial t} = H(t)|\Psi(t)\rangle$ festgelegt
- $\lim_{t \rightarrow 0} \Gamma(t) = \infty \rightarrow \lim_{t \rightarrow 0} H(t) = H_i(0)$
 - Superposition aller Zustände gleicher Amplitude
- wähle monoton fallende Funktion für $\Gamma(t)$ („ausglühen“)
 - $\lim_{t \rightarrow \infty} \Gamma(t) = 0 \rightarrow \lim_{t \rightarrow \infty} H(t) = H_f$



V. Bapst et al., Physics Reports 523 (2013)

Adiabatische Zeit

Adiabatische Zeit

Adiabatisches Theorem der Quantenmechanik

System bleibt im selben Eigenzustand von H , wenn sich H hinreichend langsam ändert

Adiabatische Zeit

Adiabatisches Theorem der Quantenmechanik

System bleibt im selben Eigenzustand von H , wenn sich H hinreichend langsam ändert

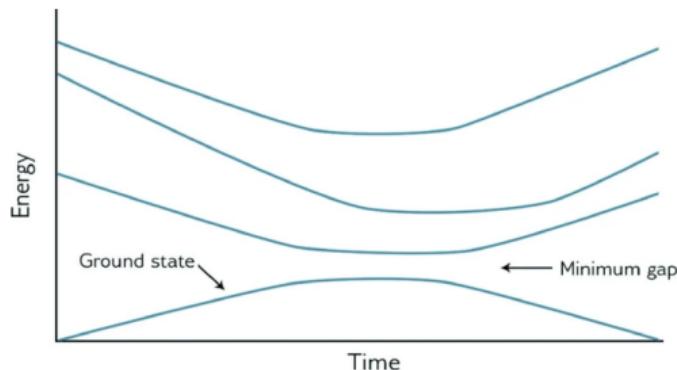
- minimale Zeit, damit sich System adiabatisch ändert τ

Adiabatische Zeit

Adiabatisches Theorem der Quantenmechanik

System bleibt im selben Eigenzustand von H , wenn sich H hinreichend langsam ändert

- minimale Zeit, damit sich System adiabatisch ändert τ
- $\Delta(t) = E_1(t) - E_0(t)$



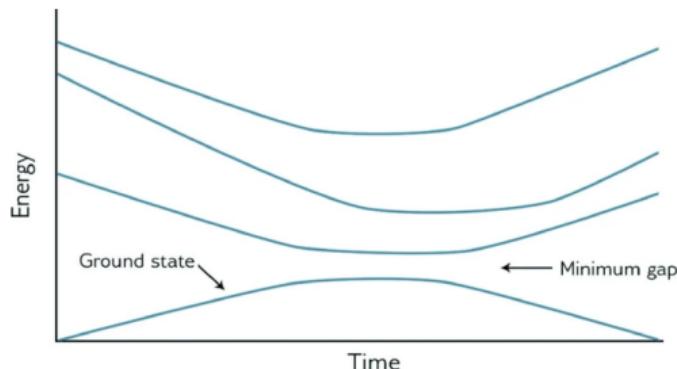
E. R. Miranda (Ed.), Quantum Computer Music (2022)

Adiabatische Zeit

Adiabatisches Theorem der Quantenmechanik

System bleibt im selben Eigenzustand von H , wenn sich H hinreichend langsam ändert

- minimale Zeit, damit sich System adiabatisch ändert τ
- $\Delta(t) = E_1(t) - E_0(t)$
- mindestens $\tau \propto \frac{1}{\Delta(t)^2}$, schlimmstenfalls $\tau \propto \frac{1}{\Delta(t)^3}$



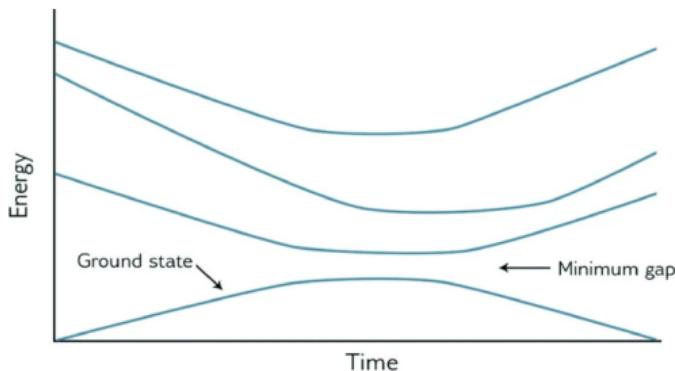
E. R. Miranda (Ed.), Quantum Computer Music (2022)

Adiabatische Zeit

Adiabatisches Theorem der Quantenmechanik

System bleibt im selben Eigenzustand von H , wenn sich H hinreichend langsam ändert

- minimale Zeit, damit sich System adiabatisch ändert τ
- $\Delta(t) = E_1(t) - E_0(t)$
- mindestens $\tau \propto \frac{1}{\Delta(t)^2}$, schlimmstenfalls $\tau \propto \frac{1}{\Delta(t)^3}$
- Problem: $\Delta(t)$ sehr klein



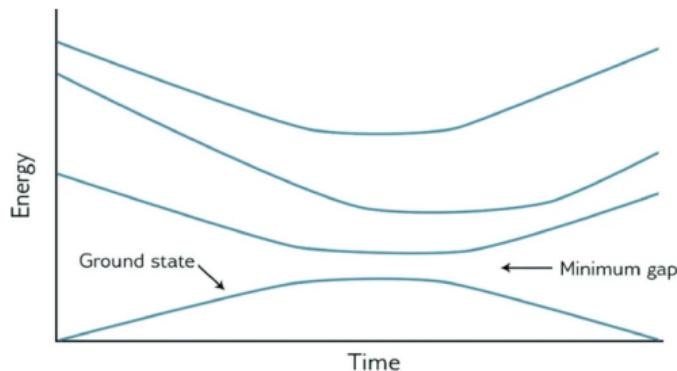
E. R. Miranda (Ed.), Quantum Computer Music (2022)

Adiabatische Zeit

Adiabatisches Theorem der Quantenmechanik

System bleibt im selben Eigenzustand von H , wenn sich H hinreichend langsam ändert

- minimale Zeit, damit sich System adiabatisch ändert τ
- $\Delta(t) = E_1(t) - E_0(t)$
- mindestens $\tau \propto \frac{1}{\Delta(t)^2}$, schlimmstenfalls $\tau \propto \frac{1}{\Delta(t)^3}$
- Problem: $\Delta(t)$ sehr klein
 - $\tau \rightarrow \infty$



E. R. Miranda (Ed.), Quantum Computer Music (2022)

Simulation zu simulated und quantum annealing

$$H(t) = \underbrace{- \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z}_{H_f} - h \sum_i \sigma_i^z - \Gamma(t) \sum_i \sigma_i^x$$

Simulation zu simulated und quantum annealing

$$H(t) = \underbrace{- \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z}_{H_f} - h \sum_i \sigma_i^z - \Gamma(t) \sum_i \sigma_i^x$$

- $P_{SA}(t)$: Wahrscheinlichkeit System in GZ zu finden
- $P_{QA}(t) = |\langle g | \Psi(t) \rangle|^2$: Wahrscheinlichkeit System in GZ $|g\rangle$ von H_f zu finden
- $P_{SA}^{st}(T)$: Boltzmann-Faktor des GZ von H_f
- $P_{QA}^{st}(\Gamma) = |\langle g | \Psi_\Gamma \rangle|^2$ mit $|\Psi_\Gamma\rangle$ als GZ von H

Simulation zu simulated und quantum annealing

$$H(t) = \underbrace{- \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z}_{H_f} - h \sum_i \sigma_i^z - \Gamma(t) \sum_i \sigma_i^x$$

- $P_{SA}(t)$: Wahrscheinlichkeit System in GZ zu finden
- $P_{QA}(t) = |\langle g | \Psi(t) \rangle|^2$: Wahrscheinlichkeit System in GZ $|g\rangle$ von H_f zu finden
- $P_{SA}^{st}(T)$: Boltzmann-Faktor des GZ von H_f
- $P_{QA}^{st}(\Gamma) = |\langle g | \Psi_\Gamma \rangle|^2$ mit $|\Psi_\Gamma\rangle$ als GZ von H
- $\lim_{t \rightarrow 0} \Gamma(t) = \lim_{t \rightarrow 0} T(t) = \infty$

Simulation zu simulated und quantum annealing

$$H(t) = \underbrace{- \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z}_{H_f} - h \underbrace{\sum_i \sigma_i^z}_{\Gamma(t)} \underbrace{- \Gamma(t) \sum_i \sigma_i^x}_{H_i(t)}$$

- $P_{SA}(t)$: Wahrscheinlichkeit System in GZ zu finden
- $P_{QA}(t) = |\langle g | \Psi(t) \rangle|^2$: Wahrscheinlichkeit System in GZ $|g\rangle$ von H_f zu finden
- $P_{SA}^{st}(T)$: Boltzmann-Faktor des GZ von H_f
- $P_{QA}^{st}(\Gamma) = |\langle g | \Psi_\Gamma \rangle|^2$ mit $|\Psi_\Gamma\rangle$ als GZ von H
- $\lim_{t \rightarrow 0} \Gamma(t) = \lim_{t \rightarrow 0} T(t) = \infty$
 - Superposition aller Zustände gleicher Amplitude (QA)
 - Alle Zustände gleich wahrscheinlich (SA)

Simulation zu simulated und quantum annealing

$$H(t) = \underbrace{- \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z}_{H_f} - h \underbrace{\sum_i \sigma_i^z}_{\Gamma(t)} \underbrace{- \Gamma(t) \sum_i \sigma_i^x}_{H_i(t)}$$

- $P_{SA}(t)$: Wahrscheinlichkeit System in GZ zu finden
- $P_{QA}(t) = |\langle g | \Psi(t) \rangle|^2$: Wahrscheinlichkeit System in GZ $|g\rangle$ von H_f zu finden
- $P_{SA}^{st}(T)$: Boltzmann-Faktor des GZ von H_f
- $P_{QA}^{st}(\Gamma) = |\langle g | \Psi_\Gamma \rangle|^2$ mit $|\Psi_\Gamma\rangle$ als GZ von H
- $\lim_{t \rightarrow 0} \Gamma(t) = \lim_{t \rightarrow 0} T(t) = \infty$
 - Superposition aller Zustände gleicher Amplitude (QA)
 - Alle Zustände gleich wahrscheinlich (SA)
- Idealfall (adiabatisch): $\lim_{t \rightarrow \infty} P_{QA}(t) = \lim_{t \rightarrow \infty} P_{SA}(t) = 1$

Simulation zu simulated und quantum annealing

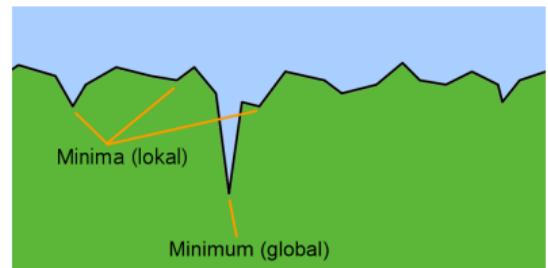
$$H(t) = \underbrace{- \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z}_{H_f} - h \underbrace{\sum_i \sigma_i^z}_{\Gamma(t)} \underbrace{- \Gamma(t) \sum_i \sigma_i^x}_{H_i(t)}$$

- $P_{SA}(t)$: Wahrscheinlichkeit System in GZ zu finden
- $P_{QA}(t) = |\langle g | \Psi(t) \rangle|^2$: Wahrscheinlichkeit System in GZ $|g\rangle$ von H_f zu finden
- $P_{SA}^{st}(T)$: Boltzmann-Faktor des GZ von H_f
- $P_{QA}^{st}(\Gamma) = |\langle g | \Psi_\Gamma \rangle|^2$ mit $|\Psi_\Gamma\rangle$ als GZ von H
- $\lim_{t \rightarrow 0} \Gamma(t) = \lim_{t \rightarrow 0} T(t) = \infty$
 - Superposition aller Zustände gleicher Amplitude (QA)
 - Alle Zustände gleich wahrscheinlich (SA)
- Idealfall (adiabatisch): $\lim_{t \rightarrow \infty} P_{QA}(t) = \lim_{t \rightarrow \infty} P_{SA}(t) = 1$
 - Grundzustand gefunden

Simulation zu simulated und quantum annealing

- $P_{SA}(t)$: Wahrscheinlichkeit System in GZ zu finden
- $P_{QA}(t) = |\langle g | \Psi(t) \rangle|^2$: Wahrscheinlichkeit System in GZ $|g\rangle$ von H_f zu finden
- $P_{SA}^{st}(T)$: Boltzmann-Faktor des GZ von H_f
- $P_{QA}^{st}(\Gamma) = |\langle g | \Psi_\Gamma \rangle|^2$ mit $|\Psi_\Gamma\rangle$ als GZ von H
- $\lim_{t \rightarrow 0} \Gamma(t) = \lim_{t \rightarrow 0} T(t) = \infty$
 - Superposition aller Zustände gleicher Amplitude (QA)
 - Alle Zustände gleich wahrscheinlich (SA)
- Idealfall (adiabatisch): $\lim_{t \rightarrow \infty} P_{QA}(t) = \lim_{t \rightarrow \infty} P_{SA}(t) = 1$
 - Grundzustand gefunden
 - sonst: In einem lokalen Minimum „hängen“ geblieben

$$H(t) = - \underbrace{\sum_{ij} J_{ij} \sigma_i^z \sigma_j^z}_{H_f} - h \underbrace{\sum_i \sigma_i^z}_{\Gamma(t)} \underbrace{- \Gamma(t) \sum_i \sigma_i^x}_{H_i(t)}$$

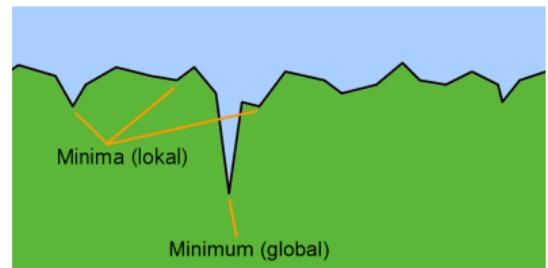


Fleischer, K., wikipedia.org

Simulation zu simulated und quantum annealing

- $P_{SA}(t)$: Wahrscheinlichkeit System in GZ zu finden
- $P_{QA}(t) = |\langle g | \Psi(t) \rangle|^2$: Wahrscheinlichkeit System in GZ $|g\rangle$ von H_f zu finden
- $P_{SA}^{st}(T)$: Boltzmann-Faktor des GZ von H_f
- $P_{QA}^{st}(\Gamma) = |\langle g | \Psi_\Gamma \rangle|^2$ mit $|\Psi_\Gamma\rangle$ als GZ von H
- $\lim_{t \rightarrow 0} \Gamma(t) = \lim_{t \rightarrow 0} T(t) = \infty$
 - Superposition aller Zustände gleicher Amplitude (QA)
 - Alle Zustände gleich wahrscheinlich (SA)
- Idealfall (adiabatisch): $\lim_{t \rightarrow \infty} P_{QA}(t) = \lim_{t \rightarrow \infty} P_{SA}(t) = 1$
 - Grundzustand gefunden
 - sonst: In einem lokalen Minimum „hängen“ geblieben
- 8 Spins

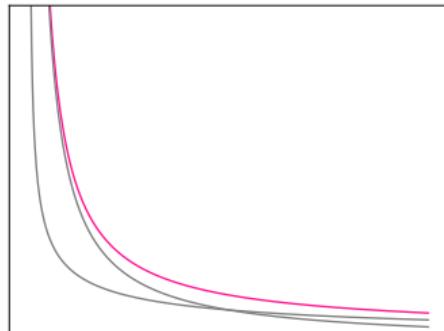
$$H(t) = - \underbrace{\sum_{ij} J_{ij} \sigma_i^z \sigma_j^z}_{H_f} - h \underbrace{\sum_i \sigma_i^z}_{\Gamma(t)} \underbrace{- \Gamma(t) \sum_i \sigma_i^x}_{H_i(t)}$$



Fleischer, K., wikipedia.org

Ergebnisse zu simulated und quantum annealing

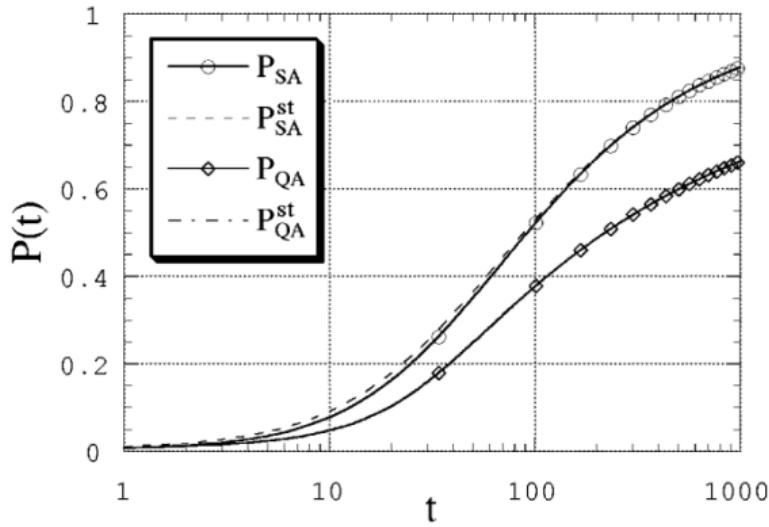
- $P_{SA}(t)$: Wahrscheinlichkeit System in GZ zu finden
- $P_{QA}(t) = |\langle g | \Psi(t) \rangle|^2$: Wahrscheinlichkeit System in GZ $|g\rangle$ von H_f zu finden
- $P_{SA}^{st}(T)$: Boltzmann-Faktor des GZ von H_f
- $P_{QA}^{st}(\Gamma) = |\langle g | \Psi_\Gamma \rangle|^2$ mit $|\Psi_\Gamma\rangle$ als GZ von H



Kadowaki, T., Nishimori, H., PHYSICAL REVIEW E 1998

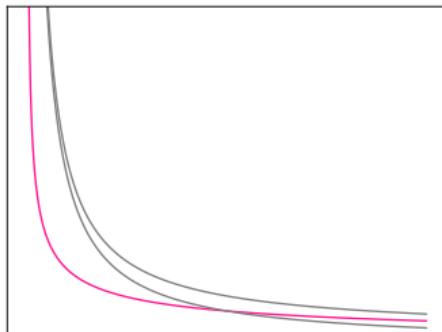
- $J_{ij} = const > 0$
- $h = 0.1$

$$\Gamma(t) = T(t) = 3 / \ln(t + 1)$$



Ergebnisse zu simulated und quantum annealing

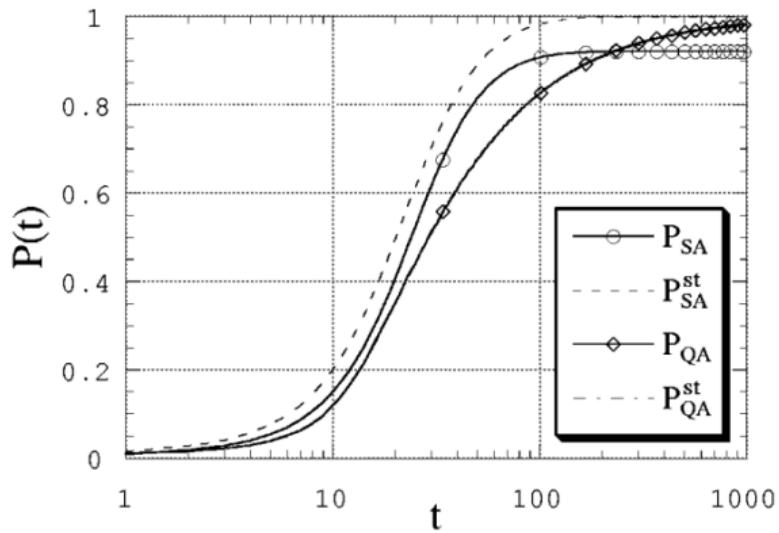
- $P_{SA}(t)$: Wahrscheinlichkeit System in GZ zu finden
- $P_{QA}(t) = |\langle g | \Psi(t) \rangle|^2$: Wahrscheinlichkeit System in GZ $|g\rangle$ von H_f zu finden
- $P_{SA}^{st}(T)$: Boltzmann-Faktor des GZ von H_f
- $P_{QA}^{st}(\Gamma) = |\langle g | \Psi_\Gamma \rangle|^2$ mit $|\Psi_\Gamma\rangle$ als GZ von H



Kadowaki, T., Nishimori, H., PHYSICAL REVIEW E 1998

- $J_{ij} = const > 0$
- $h = 0.1$

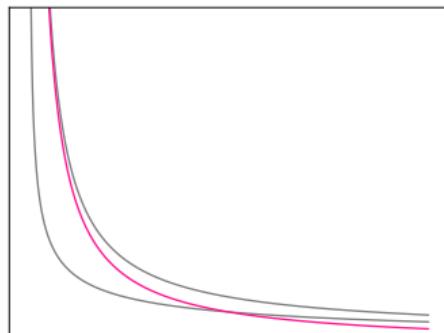
$$\Gamma(t) = T(t) = 3/\sqrt{t}$$



Simulated und quantum annealing

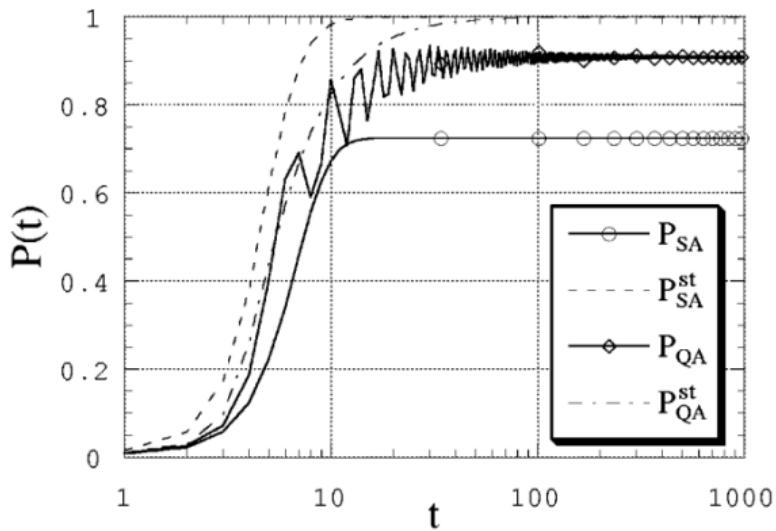
Ergebnisse zu simulated und quantum annealing

- $P_{SA}(t)$: Wahrscheinlichkeit System in GZ zu finden
- $P_{QA}(t) = |\langle g | \Psi(t) \rangle|^2$: Wahrscheinlichkeit System in GZ $|g\rangle$ von H_f zu finden
- $P_{SA}^{st}(T)$: Boltzmann-Faktor des GZ von H_f
- $P_{QA}^{st}(\Gamma) = |\langle g | \Psi_\Gamma \rangle|^2$ mit $|\Psi_\Gamma\rangle$ als GZ von H



■ $J_{ij} = const > 0$
■ $h = 0.1$

$$\Gamma(t) = T(t) = 3/t$$



Kadowaki, T., Nishimori, H., PHYSICAL REVIEW E 1998

Binäre Variable

Binäre Variable

- $x \in \{0, 1\}$

Binäre Variable

- $x \in \{0, 1\}$

Beispiel: Aufenthaltsort von einem Truck

- Formulierung als Graph mit N Knoten i

Binäre Variable

- $x \in \{0, 1\}$

Beispiel: Aufenthaltsort von einem Truck

- Formulierung als Graph mit N Knoten i
- ordne jedem Knoten i eine binäre Variable zu

Binäre Variable

- $x \in \{0, 1\}$

Beispiel: Aufenthaltsort von einem Truck

- Formulierung als Graph mit N Knoten i
- ordne jedem Knoten i eine binäre Variable zu
- jedem Truck kann ein Vektor $x^j = (x_1^j, \dots, x_N^j)$ von binären Variablen zugeordnet werden

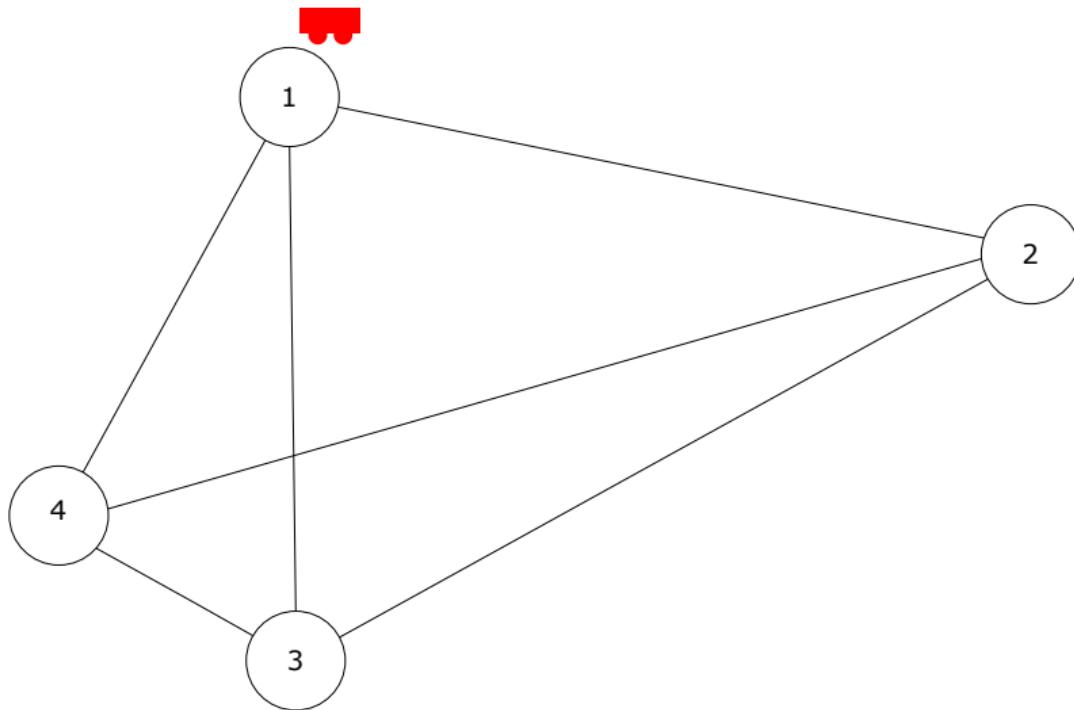
Binäre Variable

- $x \in \{0, 1\}$

Beispiel: Aufenthaltsort von einem Truck

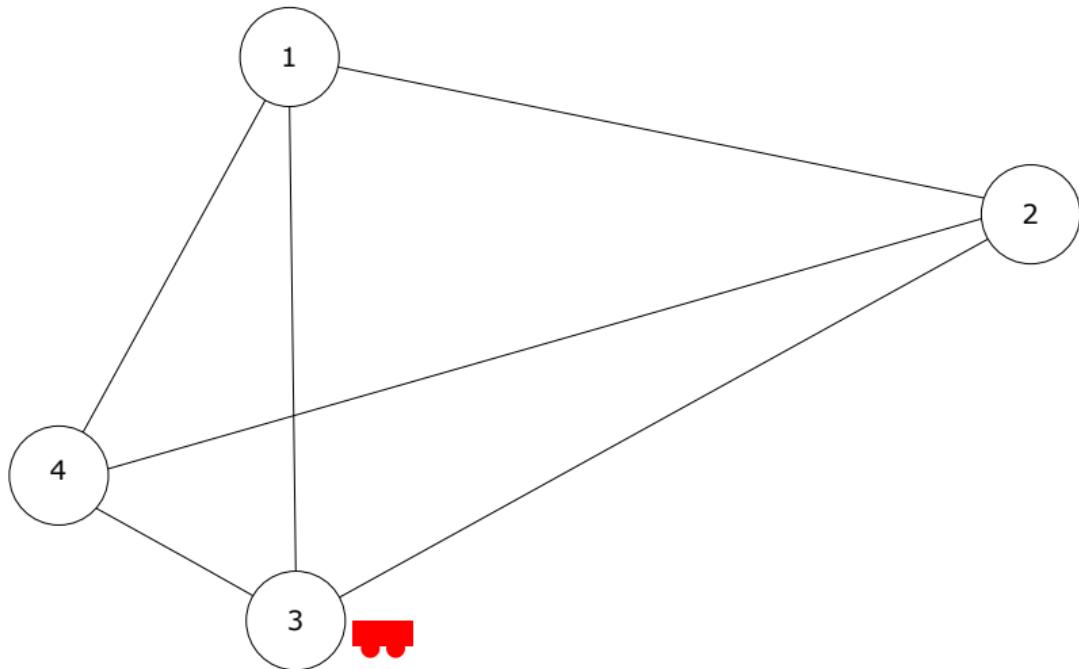
- Formulierung als Graph mit N Knoten i
- ordne jedem Knoten i eine binäre Variable zu
- jedem Truck kann ein Vektor $x^j = (x_1^j, \dots, x_N^j)$ von binären Variablen zugeordnet werden
- Truck j an Knoten $i \rightarrow x_i^j = 1$ sonst $x_k^j = 0$

Binäre Variable



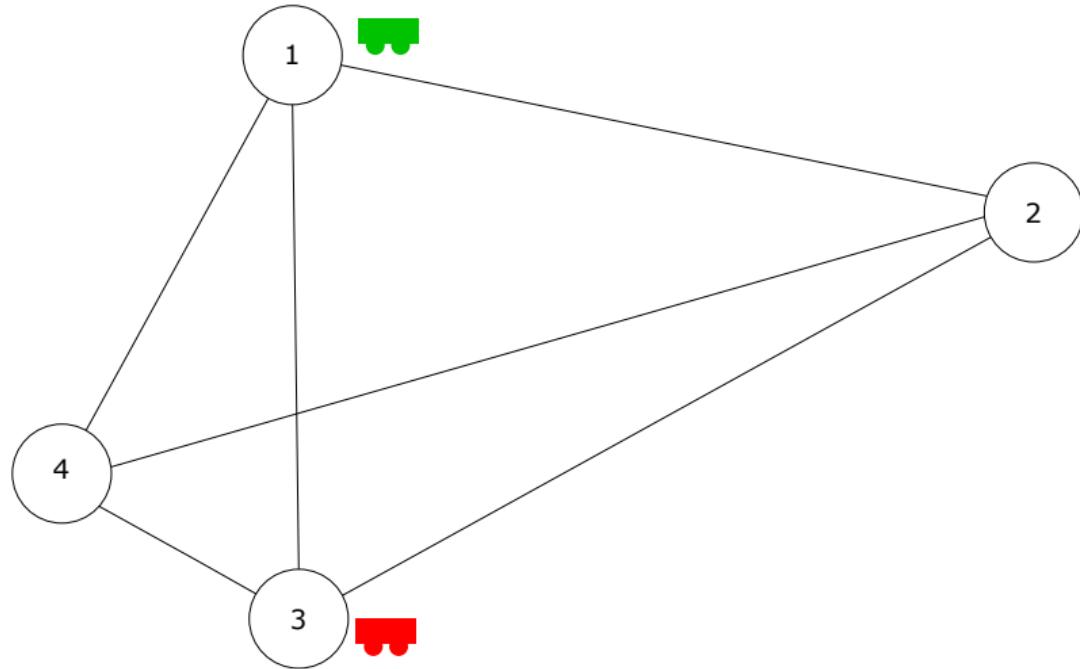
$$\underline{x}^{\text{rot}} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Binäre Variable



$$\underline{x}^{\text{rot}} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

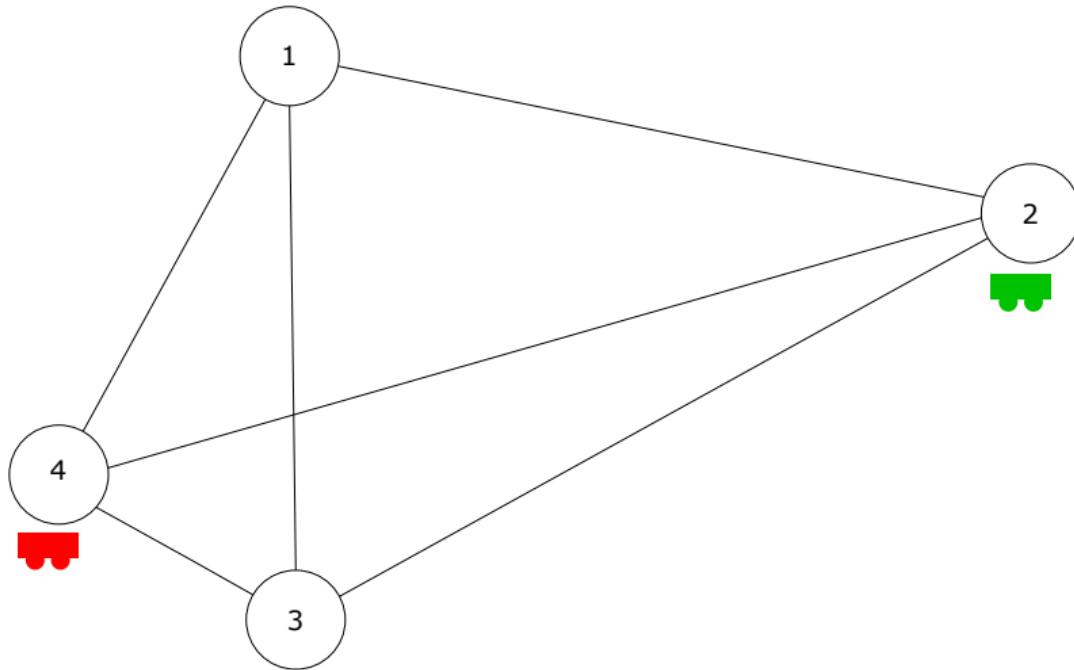
Binäre Variable



$$\underline{x}^{\text{rot}} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\underline{x}^{\text{grün}} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Binäre Variable



$$\underline{x}^{\text{rot}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\underline{x}^{\text{grün}} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Quadratic unconstrained binary optimization (QUBO)

Quadratic unconstrained binary optimization (QUBO)

Aufgabe

$$\text{Minimiere } f(\underline{x}, \underline{\underline{Q}}) = \sum_{ij} Q_{ij} x_i x_j$$

Quadratic unconstrained binary optimization (QUBO)

Aufgabe

$$\text{Minimiere } f(\underline{x}, \underline{\underline{Q}}) = \sum_{ij} Q_{ij} x_i x_j$$

- Q_{ij} definiert Wechselwirkung zwischen x_i

Quadratic unconstrained binary optimization (QUBO)

Aufgabe

$$\text{Minimiere } f(\underline{x}, \underline{\underline{Q}}) = \sum_{ij} Q_{ij} x_i x_j$$

- Q_{ij} definiert Wechselwirkung zwischen x_i
 - allg. hermitesch (symmetrisch im Ising-Modell)
- $x_i \in \{0, 1\}$ als binäre Variable

Quadratic unconstrained binary optimization (QUBO)

Aufgabe

$$\text{Minimiere } f(\underline{x}, \underline{\underline{Q}}) = \sum_{ij} Q_{ij} x_i x_j$$

- Q_{ij} definiert Wechselwirkung zwischen x_i
 - allg. hermitesch (symmetrisch im Ising-Modell)
- $x_i \in \{0, 1\}$ als binäre Variable
 - lässt sich mittels $s_i = 2x_i - 1 \in \{-1, 1\}$ auf Ising-Variablen („Spin-1/2“) transformieren

Quadratic unconstrained binary optimization (QUBO)

Aufgabe

$$\text{Minimiere } f(\underline{x}, \underline{\underline{Q}}) = \sum_{ij} Q_{ij} x_i x_j$$

- Q_{ij} definiert Wechselwirkung zwischen x_i
 - allg. hermitesch (symmetrisch im Ising-Modell)
- $x_i \in \{0, 1\}$ als binäre Variable
 - lässt sich mittels $s_i = 2x_i - 1 \in \{-1, 1\}$ auf Ising-Variablen („Spin-1/2“) transformieren

Quadratic unconstrained binary optimization constraints

Quadratic unconstrained binary optimization constraints

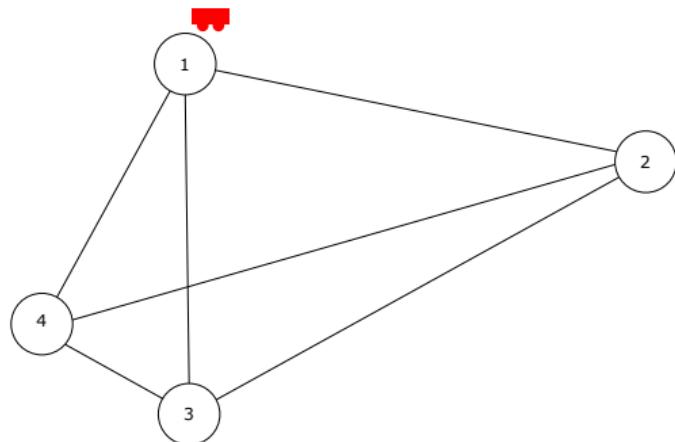
- constraints lassen sich nicht direkt in der Hardware implementieren

Quadratic unconstrained binary optimization constraints

- constraints lassen sich nicht direkt in der Hardware implementieren
 - constraints als „Strafe“ in der Kostenfunktion implementieren

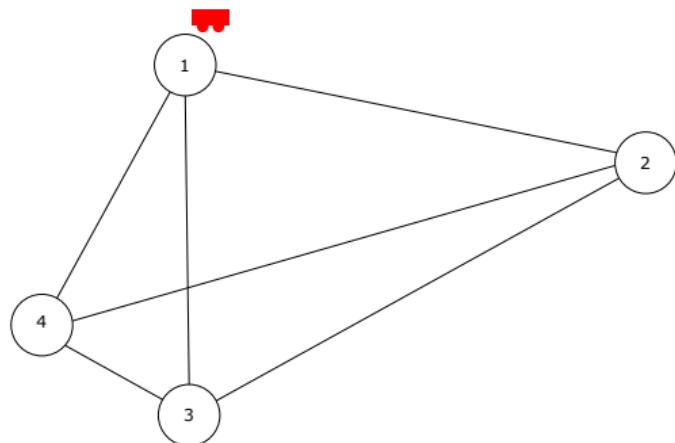
Quadratic unconstrained binary optimization constraints

- constraints lassen sich nicht direkt in der Hardware implementieren
 - constraints als „Strafe“ in der Kostenfunktion implementieren
- Beispiel: Ein Truck darf sich nur an einem Punkt aufhalten



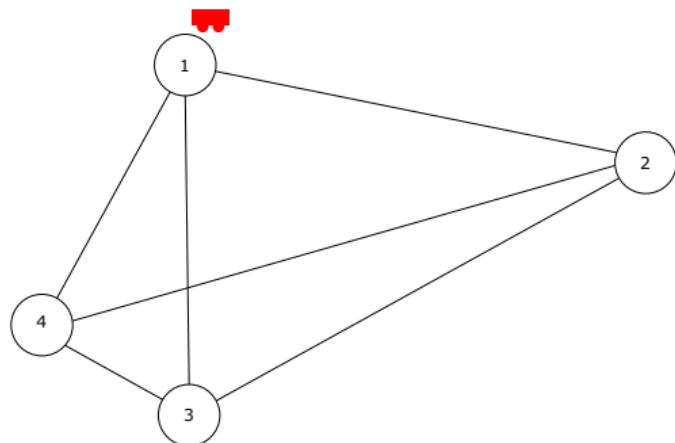
Quadratic unconstrained binary optimization constraints

- constraints lassen sich nicht direkt in der Hardware implementieren
 - constraints als „Strafe“ in der Kostenfunktion implementieren
- Beispiel: Ein Truck darf sich nur an einem Punkt aufhalten
 - $\sum_i x_i = 1 \iff \sum_i x_i - 1 = 0$



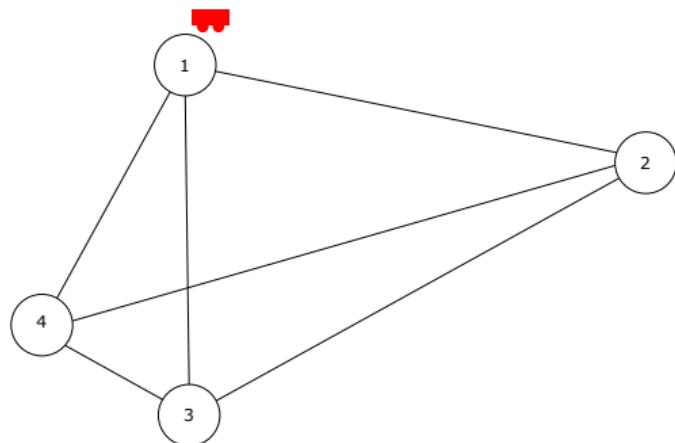
Quadratic unconstrained binary optimization constraints

- constraints lassen sich nicht direkt in der Hardware implementieren
 - constraints als „Strafe“ in der Kostenfunktion implementieren
- Beispiel: Ein Truck darf sich nur an einem Punkt aufhalten
 - $\sum_i x_i = 1 \iff \sum_i x_i - 1 = 0$
 - in QUBO-Problem umwandeln $\rightarrow g = (\sum_i x_i - 1)^2$



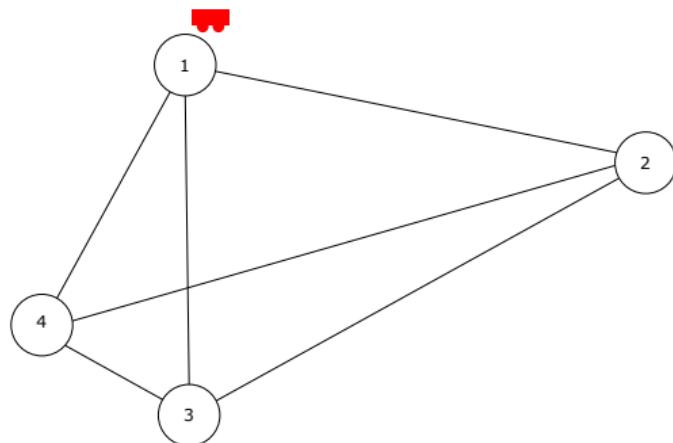
Quadratic unconstrained binary optimization constraints

- constraints lassen sich nicht direkt in der Hardware implementieren
 - constraints als „Strafe“ in der Kostenfunktion implementieren
- Beispiel: Ein Truck darf sich nur an einem Punkt aufhalten
 - $\sum_i x_i = 1 \iff \sum_i x_i - 1 = 0$
 - in QUBO-Problem umwandeln $\rightarrow g = (\sum_i x_i - 1)^2$
 - $f \rightarrow f + \lambda g$



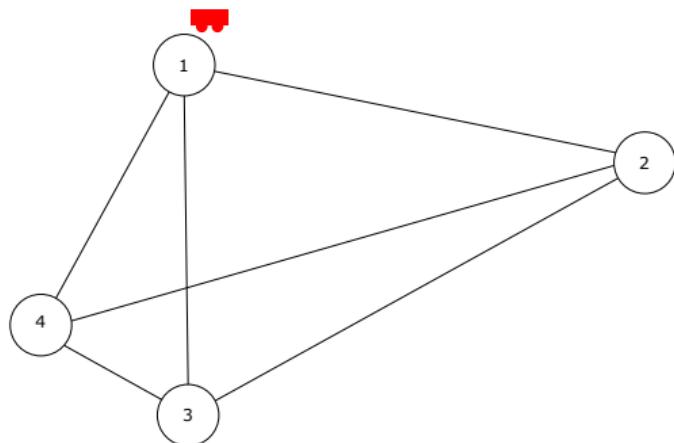
Quadratic unconstrained binary optimization constraints

- constraints lassen sich nicht direkt in der Hardware implementieren
 - constraints als „Strafe“ in der Kostenfunktion implementieren
- Beispiel: Ein Truck darf sich nur an einem Punkt aufhalten
 - $\sum_i x_i = 1 \iff \sum_i x_i - 1 = 0$
 - in QUBO-Problem umwandeln $\rightarrow g = (\sum_i x_i - 1)^2$
 - $f \rightarrow f + \lambda g$
 - constraint verletzt → Kostenfunktion erhöht



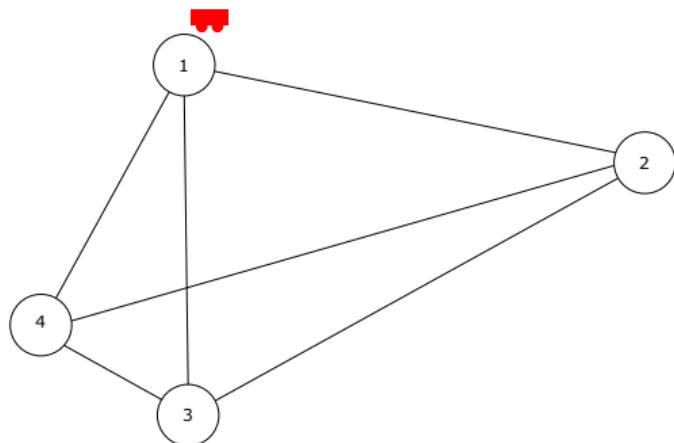
Quadratic unconstrained binary optimization constraints

- constraints lassen sich nicht direkt in der Hardware implementieren
 - constraints als „Strafe“ in der Kostenfunktion implementieren
- Beispiel: Ein Truck darf sich nur an einem Punkt aufhalten
 - $\sum_i x_i = 1 \iff \sum_i x_i - 1 = 0$
 - in QUBO-Problem umwandeln $\rightarrow g = (\sum_i x_i - 1)^2$
 - $f \rightarrow f + \lambda g$
 - constraint verletzt → Kostenfunktion erhöht
- Lagrangeparameter λ sehr groß wählen



Quadratic unconstrained binary optimization constraints

- constraints lassen sich nicht direkt in der Hardware implementieren
 - constraints als „Strafe“ in der Kostenfunktion implementieren
- Beispiel: Ein Truck darf sich nur an einem Punkt aufhalten
 - $\sum_i x_i = 1 \iff \sum_i x_i - 1 = 0$
 - in QUBO-Problem umwandeln $\rightarrow g = (\sum_i x_i - 1)^2$
 - $f \rightarrow f + \lambda g$
 - constraint verletzt → Kostenfunktion erhöht
- Lagrangeparameter λ sehr groß wählen
 - constraint eingehalten

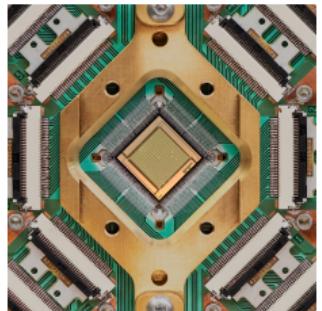


Quantum annealer von D-Wave Systems

- D-Wave Systems als Hersteller von quantum annealer



techcrunch.com



dwavesys.com

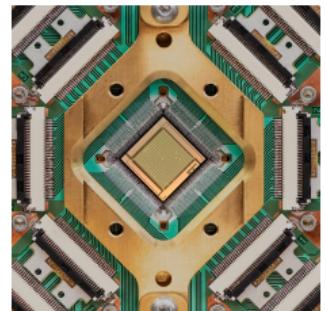
Hardware embedding

Quantum annealer von D-Wave Systems

- D-Wave Systems als Hersteller von quantum annealer
- Quantum Processing Unit (QPU) mit qubits und coupler



techcrunch.com



dwavesys.com

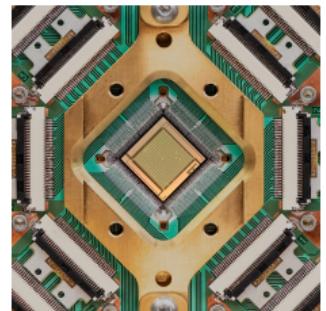
Hardware embedding

Quantum annealer von D-Wave Systems

- D-Wave Systems als Hersteller von quantum annealer
- Quantum Processing Unit (QPU) mit qubits und coupler
 - qubits und coupler sind gewichtet



techcrunch.com



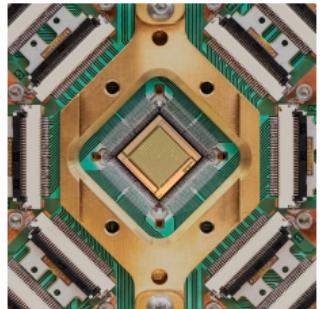
dwavesys.com

Quantum annealer von D-Wave Systems

- D-Wave Systems als Hersteller von quantum annealer
- Qantum Processing Unit (QPU) mit qubits und coupler
 - qubits und coupler sind gewichtet
- neuste Version: 5000 qubits mit 15 coupler pro qubit



techcrunch.com



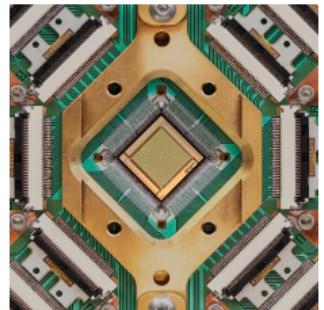
dwavesys.com

Quantum annealer von D-Wave Systems

- D-Wave Systems als Hersteller von quantum annealer
- Qantum Processing Unit (QPU) mit qubits und coupler
 - qubits und coupler sind gewichtet
- neuste Version: 5000 qubits mit 15 coupler pro qubit
- Nach annealing-Prozess die qubits auslesen



techcrunch.com



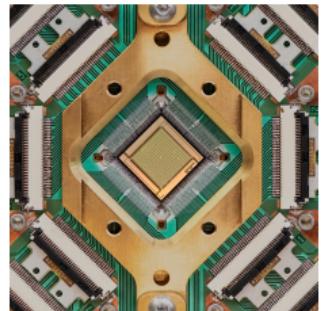
dwavesys.com

Quantum annealer von D-Wave Systems

- D-Wave Systems als Hersteller von quantum annealer
- Qantum Processing Unit (QPU) mit qubits und coupler
 - qubits und coupler sind gewichtet
- neuste Version: 5000 qubits mit 15 coupler pro qubit
- Nach annealing-Prozess die qubits auslesen
 - mögliches Energieminimum gefunden



techcrunch.com



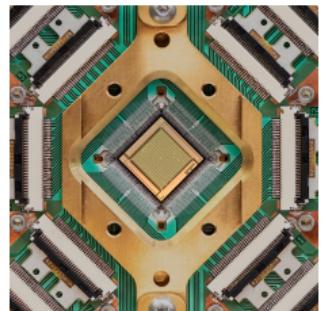
dwavesys.com

Quantum annealer von D-Wave Systems

- D-Wave Systems als Hersteller von quantum annealer
- Qantum Processing Unit (QPU) mit qubits und coupler
 - qubits und coupler sind gewichtet
- neuste Version: 5000 qubits mit 15 coupler pro qubit
- Nach annealing-Prozess die qubits auslesen
 - mögliches Energieminimum gefunden



techcrunch.com



dwavesys.com

Unterschied zu Quantencomputer

Quantum annealer von D-Wave Systems

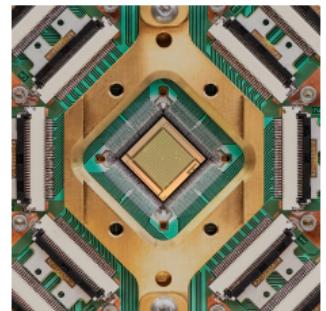
- D-Wave Systems als Hersteller von quantum annealer
- Qantum Processing Unit (QPU) mit qubits und coupler
 - qubits und coupler sind gewichtet
- neuste Version: 5000 qubits mit 15 coupler pro qubit
- Nach annealing-Prozess die qubits auslesen
 - mögliches Energieminimum gefunden



techcrunch.com

Unterschied zu Quantencomputer

- Quantum annealer nur für kombinatorische Optimierungsprobleme



dwavesys.com

Quantum annealer von D-Wave Systems

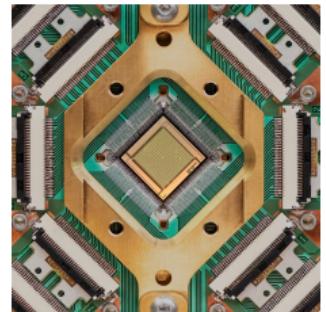
- D-Wave Systems als Hersteller von quantum annealer
- Qantum Processing Unit (QPU) mit qubits und coupler
 - qubits und coupler sind gewichtet
- neuste Version: 5000 qubits mit 15 coupler pro qubit
- Nach annealing-Prozess die qubits auslesen
 - mögliches Energieminimum gefunden



techcrunch.com

Unterschied zu Quantencomputer

- Quantum annealer nur für kombinatorische Optimierungsprobleme
- Quantencomputer nutzen Quantengatter (unitäre Operationen)



dwavesys.com

Quantum annealer von D-Wave Systems

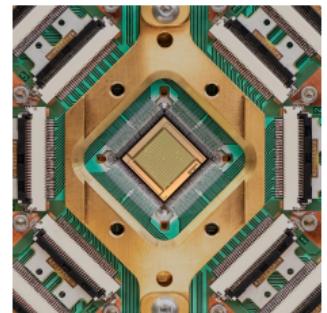
- D-Wave Systems als Hersteller von quantum annealer
- Quantum Processing Unit (QPU) mit qubits und coupler
 - qubits und coupler sind gewichtet
- neuste Version: 5000 qubits mit 15 coupler pro qubit
- Nach annealing-Prozess die qubits auslesen
 - mögliches Energieminimum gefunden



techcrunch.com

Unterschied zu Quantencomputer

- Quantum annealer nur für kombinatorische Optimierungsprobleme
- Quantencomputer nutzen Quantengatter (unitäre Operationen)
 - kann mehr Probleme lösen



dwavesys.com

Embedding

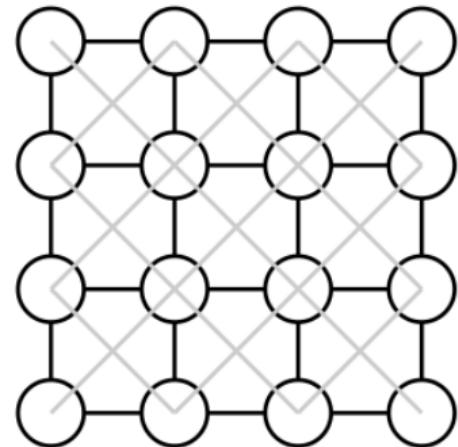
V. Choi, D-Wave Systems Inc. (2008)

Y. Kind | 12. Juli 2023

Hardware embedding

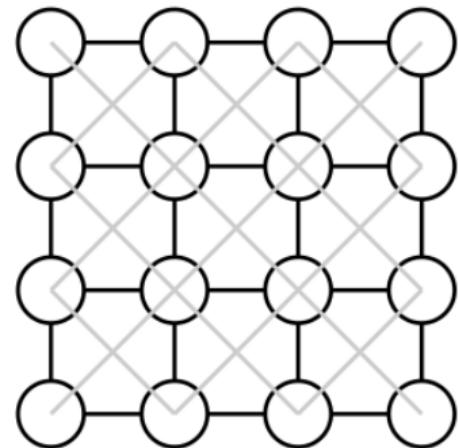
Embedding

- D-Wave's QPU kann als Graph U gesehen werden



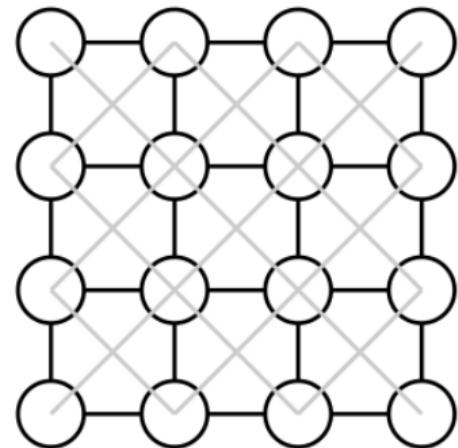
Embedding

- D-Wave's QPU kann als Graph U gesehen werden
 - Gewichtete Vertices (qubits) $i \in V(U)$



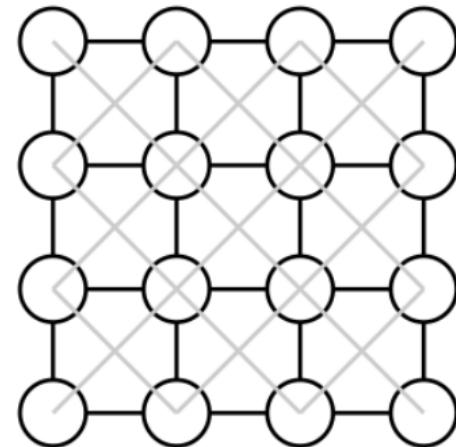
Embedding

- D-Wave's QPU kann als Graph U gesehen werden
 - Gewichtete Vertices (qubits) $i \in V(U)$
 - Gewichtete Kanten (coupler) $ij \in E(U)$



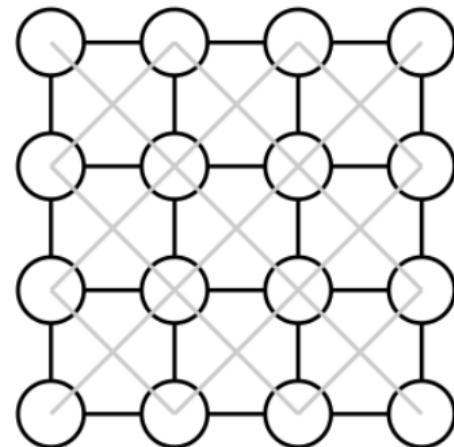
Embedding

- D-Wave's QPU kann als Graph U gesehen werden
 - Gewichtete Vertices (qubits) $i \in V(U)$
 - Gewichtete Kanten (coupler) $ij \in E(U)$
- Gewichte:
 - $h_i(t)$ „qubit bias“
 - $\Delta_i(t)$ „tunneling amplitude“
 - $J_{ij}(t)$ „coupler strength“



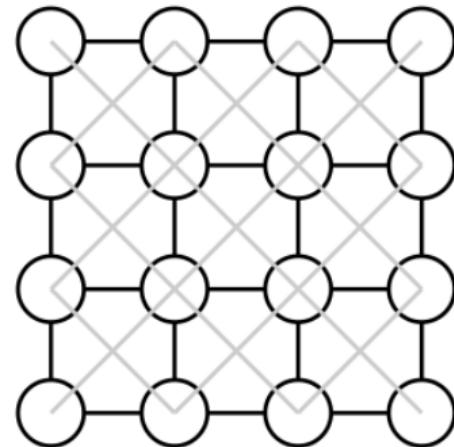
Embedding

- D-Wave's QPU kann als Graph U gesehen werden
 - Gewichtete Vertices (qubits) $i \in V(U)$
 - Gewichtete Kanten (coupler) $ij \in E(U)$
- Gewichte:
 - $h_i(t)$ „qubit bias“
 - $\Delta_i(t)$ „tunneling amplitude“
 - $J_{ij}(t)$ „coupler strength“
- formuliere transversalen Ising-Hamiltonian auf Graph G
 - $H(t) = \sum_{i \in V(G)} h_i(t) \sigma_i^z + \sum_{ij \in E(G)} J_{ij}(t) \sigma_i^z \sigma_j^z + \sum_{i \in V(G)} \Delta_i(t) \sigma_i^x$



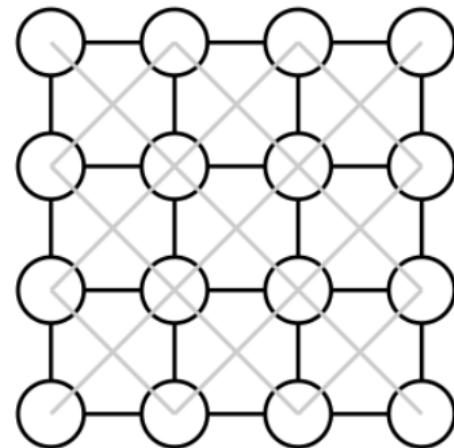
Embedding

- D-Wave's QPU kann als Graph U gesehen werden
 - Gewichtete Vertices (qubits) $i \in V(U)$
 - Gewichtete Kanten (coupler) $ij \in E(U)$
- Gewichte:
 - $h_i(t)$ „qubit bias“
 - $\Delta_i(t)$ „tunneling amplitude“
 - $J_{ij}(t)$ „coupler strength“
- formuliere transversalen Ising-Hamiltonian auf Graph G
 - $H(t) = \sum_{i \in V(G)} h_i(t) \sigma_i^z + \sum_{ij \in E(G)} J_{ij}(t) \sigma_i^z \sigma_j^z + \sum_{i \in V(G)} \Delta_i(t) \sigma_i^x$
- Eigenergie des Ising-Hamiltonians $\epsilon(s_1, \dots, s_n) = \sum_{i \in V(G)} h_i s_i + \sum_{ij \in E(G)} J_{ij} s_i s_j$



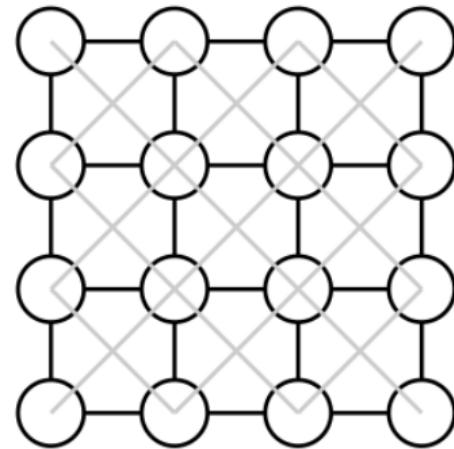
Embedding

- D-Wave's QPU kann als Graph U gesehen werden
 - Gewichtete Vertices (qubits) $i \in V(U)$
 - Gewichtete Kanten (coupler) $ij \in E(U)$
- Gewichte:
 - $h_i(t)$ „qubit bias“
 - $\Delta_i(t)$ „tunneling amplitude“
 - $J_{ij}(t)$ „coupler strength“
- formuliere transversalen Ising-Hamiltonian auf Graph G
 - $H(t) = \sum_{i \in V(G)} h_i(t) \sigma_i^z + \sum_{ij \in E(G)} J_{ij}(t) \sigma_i^z \sigma_j^z + \sum_{i \in V(G)} \Delta_i(t) \sigma_i^x$
- Eigenergie des Ising-Hamiltonians $\epsilon(s_1, \dots, s_n) = \sum_{i \in V(G)} h_i s_i + \sum_{ij \in E(G)} J_{ij} s_i s_j$
 - Spins $s_i \in \{-1, 1\}$



Embedding

- D-Wave's QPU kann als Graph U gesehen werden
 - Gewichtete Vertices (qubits) $i \in V(U)$
 - Gewichtete Kanten (coupler) $ij \in E(U)$
- Gewichte:
 - $h_i(t)$ „qubit bias“
 - $\Delta_i(t)$ „tunneling amplitude“
 - $J_{ij}(t)$ „coupler strength“
- formuliere transversalen Ising-Hamiltonian auf Graph G
 - $H(t) = \sum_{i \in V(G)} h_i(t) \sigma_i^z + \sum_{ij \in E(G)} J_{ij}(t) \sigma_i^z \sigma_j^z + \sum_{i \in V(G)} \Delta_i(t) \sigma_i^x$
- Eigenergie des Ising-Hamiltonians $\epsilon(s_1, \dots, s_n) = \sum_{i \in V(G)} h_i s_i + \sum_{ij \in E(G)} J_{ij} s_i s_j$
 - Spins $s_i \in \{-1, 1\}$
- Suche nach Grundzustandsenergie ist das „Ising-Problem“



Embedding

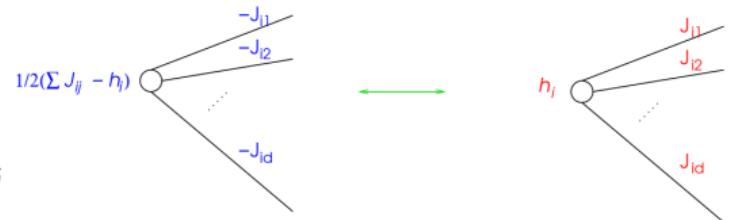
V. Choi, D-Wave Systems Inc. (2008)

Y. Kind | 12. Juli 2023

Hardware embedding

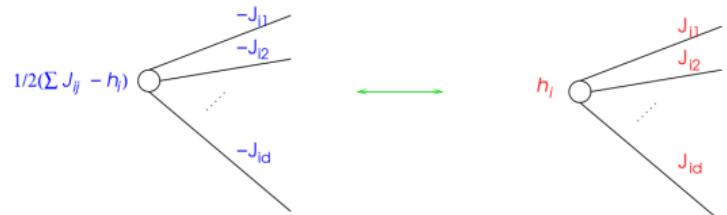
Embedding

- Ising- äquivalent zu QUBO-Problem auf dem selben Graph G



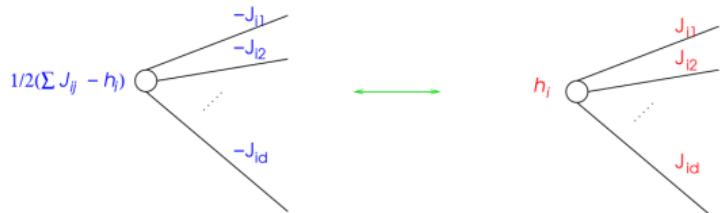
Embedding

- Ising- äquivalent zu QUBO-Problem auf dem selben Graph G
 - $Y(x_1, \dots, x_n) = \sum_{i \in V(G)} c_i x_i - \sum_{ij \in E(G)} J_{ij} x_i x_j$ maximieren
 - $x_i \in \{0, 1\}$



Embedding

- Ising- äquivalent zu QUBO-Problem auf dem selben Graph G
 - $Y(x_1, \dots, x_n) = \sum_{i \in V(G)} c_i x_i - \sum_{ij \in E(G)} J_{ij} x_i x_j$ maximieren
 - $x_i \in \{0, 1\}$

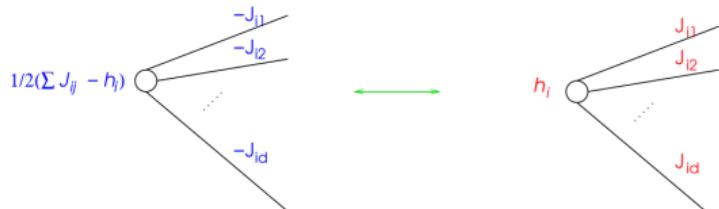


Ziel

G in U einbetten („subgraph-embedding“) und somit das Ising/QUBO-Problem lösen

Embedding

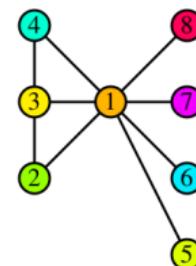
- Ising- äquivalent zu QUBO-Problem auf dem selben Graph G
 - $Y(x_1, \dots, x_n) = \sum_{i \in V(G)} c_i x_i - \sum_{ij \in E(G)} J_{ij} x_i x_j$ maximieren
 - $x_i \in \{0, 1\}$



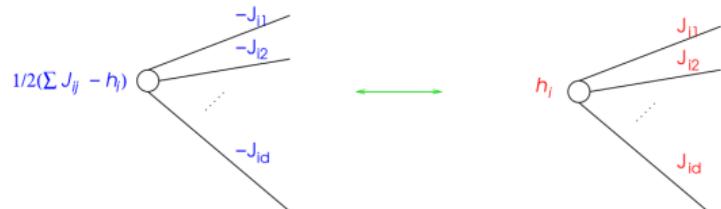
Ziel

G in U einbetten („subgraph-embedding“) und somit das Ising/QUBO-Problem lösen

- Problem: Einschränkungen durch Anzahl der coupler auf U übertragen sich auf G



Embedding

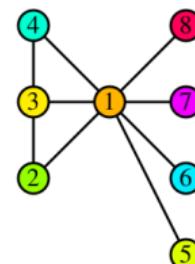


- Ising- äquivalent zu QUBO-Problem auf dem selben Graph G
 - $Y(x_1, \dots, x_n) = \sum_{i \in V(G)} c_i x_i - \sum_{ij \in E(G)} J_{ij} x_i x_j$ maximieren
 - $x_i \in \{0, 1\}$

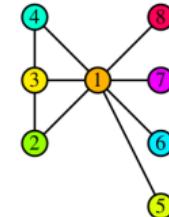
Ziel

G in U einbetten („subgraph-embedding“) und somit das Ising/QUBO-Problem lösen

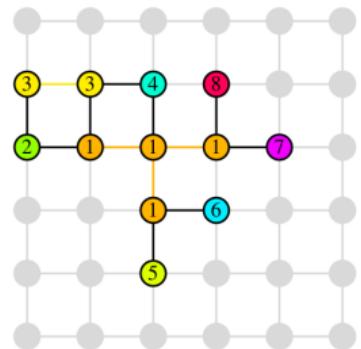
- Problem: Einschränkungen durch Anzahl der coupler auf U übertragen sich auf G
 - „dummy“-Vertices mit ferromagnetischer Kopplung einführen



Minor embedding



G

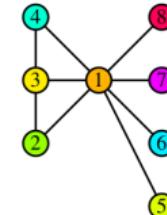


G_{emb}

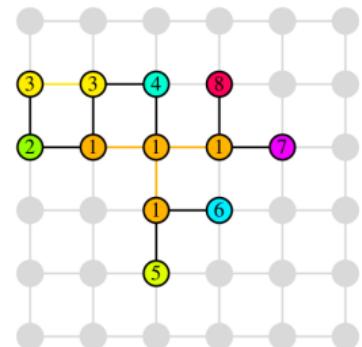
V. Choi, D-Wave Systems Inc. (2008)

Minor embedding

- logischer qubit i aus G wird auf subtree T_i physischer qubits aus $G_{\text{emb}} \subset U$ abgebildet



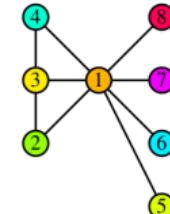
G



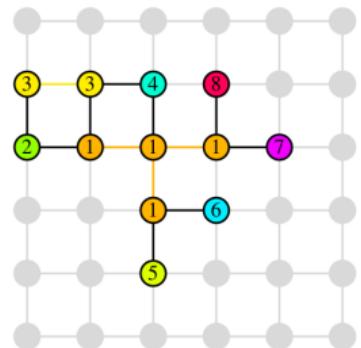
G_{emb}

Minor embedding

- logischer qubit i aus G wird auf subtree T_i physischer qubits aus $G_{\text{emb}} \subset U$ abgebildet
- ferromagnetische Kopplung zwischen „dummy“-Vertices muss hinreichend groß sein



G



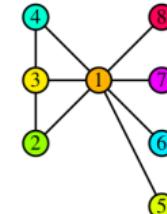
G_{emb}

Minor embedding

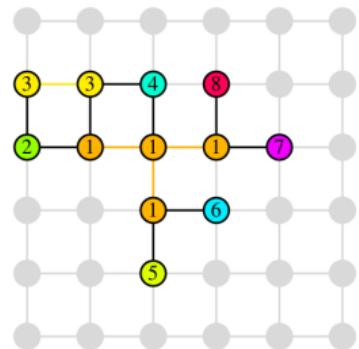
- logischer qubit i aus G wird auf subtree T_i physischer qubits aus $G_{\text{emb}} \subset U$ abgebildet
- ferromagnetische Kopplung zwischen „dummy“-Vertices muss hinreichend groß sein

Ziel

Parameter des minor-embedding auf ursprüngliche Ising/QUBO-Parameter zurückführen



G



G_{emb}

Minor embedding

V. Choi, D-Wave Systems Inc. (2008)

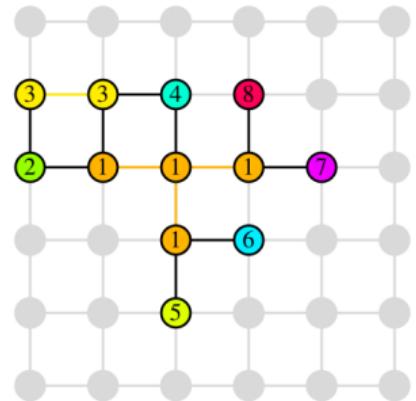
Y. Kind | 12. Juli 2023

Hardware embedding

24

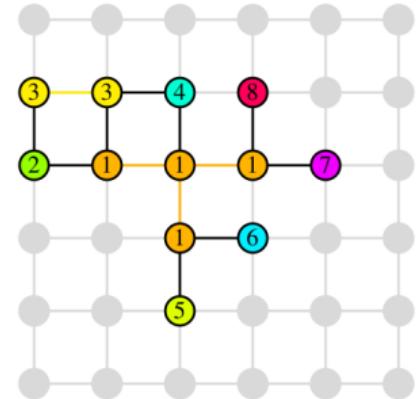
Minor embedding

- $\epsilon^{\text{emb}}(s_1, \dots, s_N) = \sum_{i \in V(G_{\text{emb}})} h'_i s_i + \sum_{ij \in E(G_{\text{emb}})} J'_{ij} s_i s_j$ minimieren



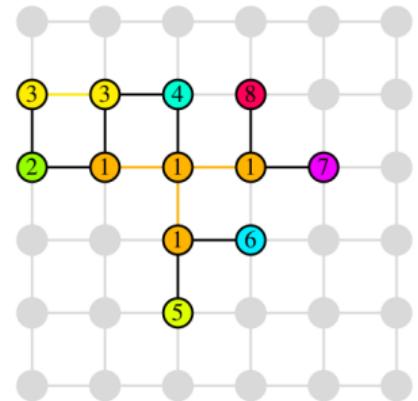
Minor embedding

- $\epsilon^{\text{emb}}(s_1, \dots, s_N) = \sum_{i \in V(G_{\text{emb}})} h'_i s_i + \sum_{ij \in E(G_{\text{emb}})} J'_{ij} s_i s_j$ minimieren
- $\tau : V(G) \times V(G) \rightarrow V(U)$, so dass $i_{\tau(i,j)} \in V(T_i)$, $j_{\tau(j,i)} \in V(T_j)$ mit $i_{\tau(i,j)} j_{\tau(j,i)} \in E(U)$



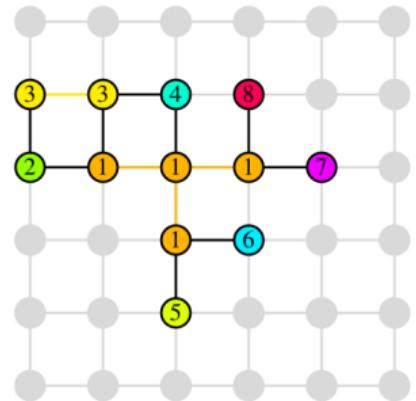
Minor embedding

- $\epsilon^{\text{emb}}(s_1, \dots, s_N) = \sum_{i \in V(G_{\text{emb}})} h'_i s_i + \sum_{ij \in E(G_{\text{emb}})} J'_{ij} s_i s_j$ minimieren
- $\tau : V(G) \times V(G) \rightarrow V(U)$, so dass $i_{\tau(i,j)} \in V(T_i)$, $j_{\tau(j,i)} \in V(T_j)$ mit $i_{\tau(i,j)} j_{\tau(j,i)} \in E(U)$
- $OE(G_{\text{emb}}) = \cup_{ij \in E(G)} i_{\tau(i,j)} j_{\tau(j,i)}$ sind ursprüngliche coupler



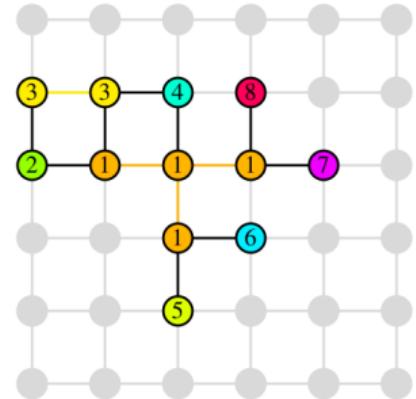
Minor embedding

- $\epsilon^{\text{emb}}(s_1, \dots, s_N) = \sum_{i \in V(G_{\text{emb}})} h'_i s_i + \sum_{ij \in E(G_{\text{emb}})} J'_{ij} s_i s_j$ minimieren
- $\tau : V(G) \times V(G) \rightarrow V(U)$, so dass $i_{\tau(i,j)} \in V(T_i)$, $j_{\tau(j,i)} \in V(T_j)$ mit $i_{\tau(i,j)} j_{\tau(j,i)} \in E(U)$
- $OE(G_{\text{emb}}) = \cup_{ij \in E(G)} i_{\tau(i,j)} j_{\tau(j,i)}$ sind ursprüngliche coupler
 → $J'_{i_{\tau(i,j)} j_{\tau(j,i)}} = J_{ij}$ für $i_{\tau(i,j)} j_{\tau(j,i)} \in OE(G_{\text{emb}})$



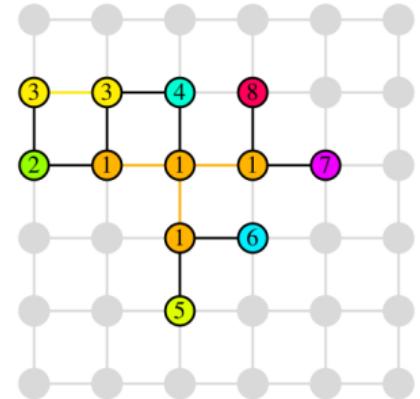
Minor embedding

- $\epsilon^{\text{emb}}(s_1, \dots, s_N) = \sum_{i \in V(G_{\text{emb}})} h'_i s_i + \sum_{ij \in E(G_{\text{emb}})} J'_{ij} s_i s_j$ minimieren
- $\tau : V(G) \times V(G) \rightarrow V(U)$, so dass $i_{\tau(i,j)} \in V(T_i)$, $j_{\tau(j,i)} \in V(T_j)$ mit $i_{\tau(i,j)} j_{\tau(j,i)} \in E(U)$
- $OE(G_{\text{emb}}) = \cup_{ij \in E(G)} i_{\tau(i,j)} j_{\tau(j,i)}$ sind ursprüngliche coupler
 → $J'_{i_{\tau(i,j)} j_{\tau(j,i)}} = J_{ij}$ für $i_{\tau(i,j)} j_{\tau(j,i)} \in OE(G_{\text{emb}})$
- ferromagnetische coupler strength F_k^{ij} für $ij \in E(T_i)$



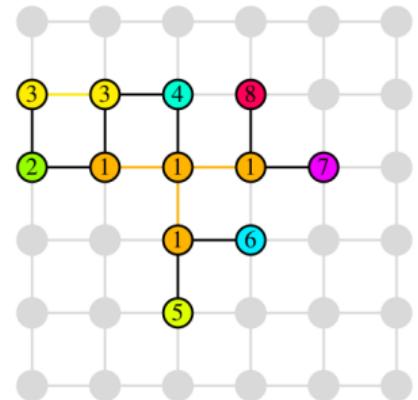
Minor embedding

- $\epsilon^{\text{emb}}(s_1, \dots, s_N) = \sum_{i \in V(G_{\text{emb}})} h'_i s_i + \sum_{ij \in E(G_{\text{emb}})} J'_{ij} s_i s_j$ minimieren
- $\tau : V(G) \times V(G) \rightarrow V(U)$, so dass $i_{\tau(i,j)} \in V(T_i)$, $j_{\tau(j,i)} \in V(T_j)$ mit $i_{\tau(i,j)} j_{\tau(j,i)} \in E(U)$
- $OE(G_{\text{emb}}) = \cup_{ij \in E(G)} i_{\tau(i,j)} j_{\tau(j,i)}$ sind ursprüngliche coupler
 → $J'_{i_{\tau(i,j)} j_{\tau(j,i)}} = J_{ij}$ für $i_{\tau(i,j)} j_{\tau(j,i)} \in OE(G_{\text{emb}})$
- ferromagnetische coupler strength F_k^{ij} für $ij \in E(T_i)$
- bias h_i der logischen qubits auf physische qubits verteilen



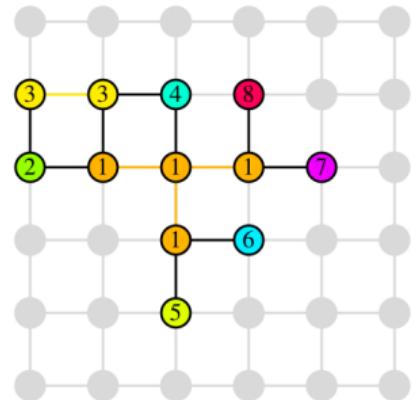
Minor embedding

- $\epsilon^{\text{emb}}(s_1, \dots, s_N) = \sum_{i \in V(G_{\text{emb}})} h'_i s_i + \sum_{ij \in E(G_{\text{emb}})} J'_{ij} s_i s_j$ minimieren
- $\tau : V(G) \times V(G) \rightarrow V(U)$, so dass $i_{\tau(i,j)} \in V(T_i)$, $j_{\tau(j,i)} \in V(T_j)$ mit $i_{\tau(i,j)} j_{\tau(j,i)} \in E(U)$
- $OE(G_{\text{emb}}) = \cup_{ij \in E(G)} i_{\tau(i,j)} j_{\tau(j,i)}$ sind ursprüngliche coupler
 → $J'_{i_{\tau(i,j)} j_{\tau(j,i)}} = J_{ij}$ für $i_{\tau(i,j)} j_{\tau(j,i)} \in OE(G_{\text{emb}})$
- ferromagnetische coupler strength F_k^{ij} für $ij \in E(T_i)$
- bias h_i der logischen qubits auf physische qubits verteilen
 → $\sum_{i_k \in V(T_i)} h'_{i_k} = h_i$



Minor embedding

- $\epsilon^{\text{emb}}(s_1, \dots, s_N) = \sum_{i \in V(G_{\text{emb}})} h'_i s_i + \sum_{ij \in E(G_{\text{emb}})} J'_{ij} s_i s_j$ minimieren
- $\tau : V(G) \times V(G) \rightarrow V(U)$, so dass $i_{\tau(i,j)} \in V(T_i)$, $j_{\tau(j,i)} \in V(T_j)$ mit $i_{\tau(i,j)} j_{\tau(j,i)} \in E(U)$
- $OE(G_{\text{emb}}) = \cup_{ij \in E(G)} i_{\tau(i,j)} j_{\tau(j,i)}$ sind ursprüngliche coupler
 → $J'_{i_{\tau(i,j)} j_{\tau(j,i)}} = J_{ij}$ für $i_{\tau(i,j)} j_{\tau(j,i)} \in OE(G_{\text{emb}})$
- ferromagnetische coupler strength F_k^{ij} für $ij \in E(T_i)$
- bias h_i der logischen qubits auf physische qubits verteilen
 → $\sum_{i_k \in V(T_i)} h'_{i_k} = h_i$

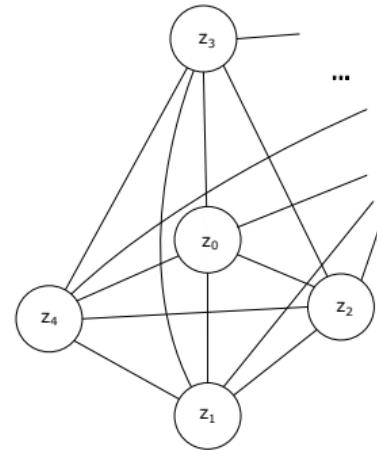


Finale Energie

$$\epsilon^{\text{emb}}(s_1, \dots, s_N) = \sum_{i \in V(G)} \left(\sum_{i_k \in V(T_i)} h'_{i_k} s_{i_k} + \sum_{i_p i_q \in E(T_i)} F_i^{pq} s_{i_p} s_{i_q} \right) + \sum_{ij \in E(G)} J_{ij} s_{i_{\tau(i,j)}} s_{j_{\tau(j,i)}}$$

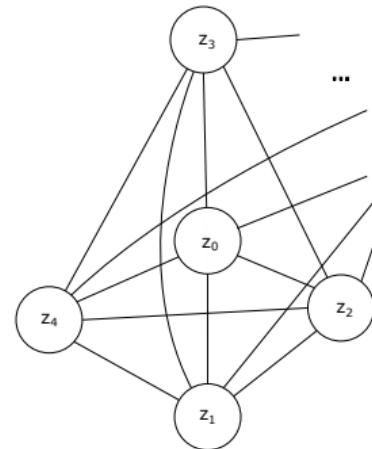
Fallbeispiel Aisin Corporation

Fallbeispiel Aisin Corporation



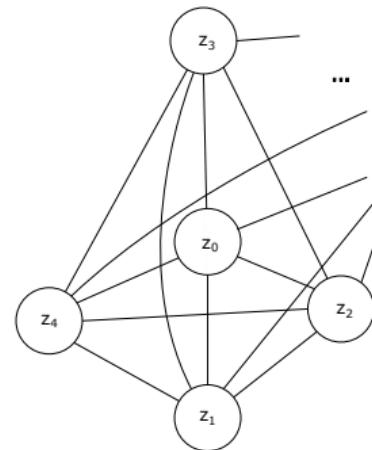
Fallbeispiel Aisin Corporation

- Zeit für Route ξ : $\text{time}(\xi) = \sum_{t=1}^{k-1} T_{\xi_t, \xi_{t+1}}$



Fallbeispiel Aisin Corporation

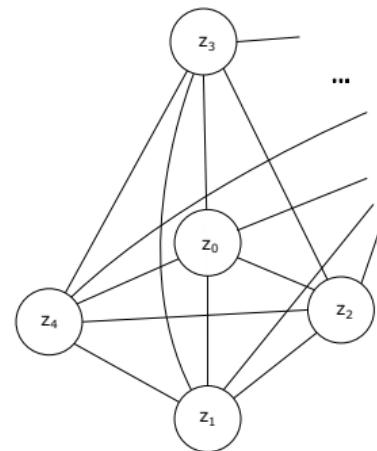
- Zeit für Route ξ : $\text{time}(\xi) = \sum_{t=1}^{k-1} T_{\xi_t, \xi_{t+1}}$
- off-board-demand D_{ij} , der von i nach j gebracht werden muss



Fallbeispiel Aisin Corporation

- Zeit für Route ξ : $\text{time}(\xi) = \sum_{t=1}^{k-1} T_{\xi_t, \xi_{t+1}}$
- off-board-demand D_{ij} , der von i nach j gebracht werden muss

Terme für QUBO-Formulierung

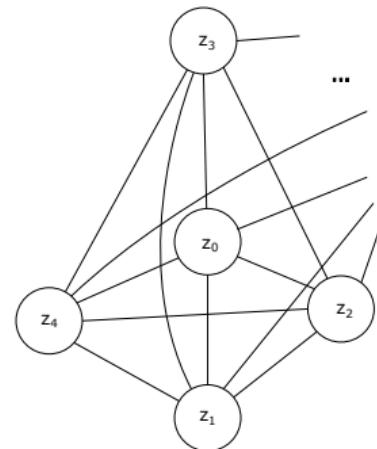


Fallbeispiel Aisin Corporation

- Zeit für Route ξ : $\text{time}(\xi) = \sum_{t=1}^{k-1} T_{\xi_t, \xi_{t+1}}$
- off-board-demand D_{ij} , der von i nach j gebracht werden muss

Terme für QUBO-Formulierung

- $f_{\text{local}}(\{x_i\}) = A_{\text{local}} \sum_t (1 - \sum_i x_{it})^2$

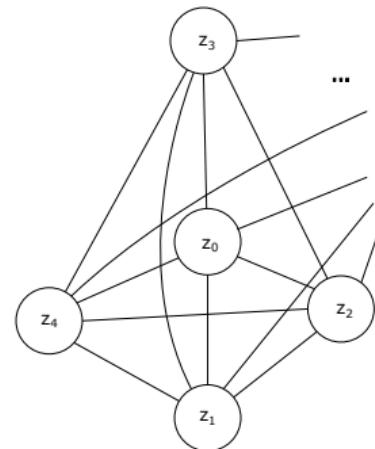


Fallbeispiel Aisin Corporation

- Zeit für Route ξ : $\text{time}(\xi) = \sum_{t=1}^{k-1} T_{\xi_t, \xi_{t+1}}$
- off-board-demand D_{ij} , der von i nach j gebracht werden muss

Terme für QUBO-Formulierung

- $f_{\text{local}}(\{x_i\}) = A_{\text{local}} \sum_t (1 - \sum_i x_{it})^2$
- $f_{\text{time}}(\{x_i\}) = A_{\text{time}} \sum_{ijt} T_{ij} x_{it} x_{jt+1}$

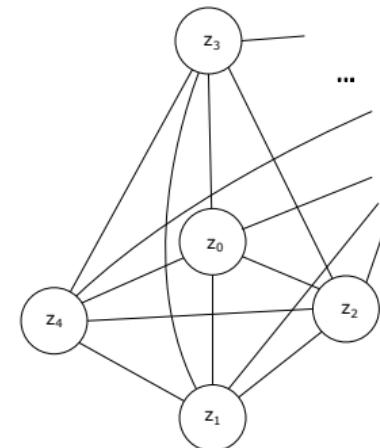


Fallbeispiel Aisin Corporation

- Zeit für Route ξ : $\text{time}(\xi) = \sum_{t=1}^{k-1} T_{\xi_t, \xi_{t+1}}$
- off-board-demand D_{ij} , der von i nach j gebracht werden muss

Terme für QUBO-Formulierung

- $f_{\text{local}}(\{x_i\}) = A_{\text{local}} \sum_t (1 - \sum_i x_{it})^2$
- $f_{\text{time}}(\{x_i\}) = A_{\text{time}} \sum_{ijt} T_{ij} x_{it} x_{jt+1}$
- $f_{\text{demand}}(\{x_i\}) = -A_{\text{demand}} \sum_{ijt} \sum_{\delta=1}^{\delta_*(t)} D_{ij} x_{it} x_{jt+\delta}$ mit $\delta_*(t) = \min(\delta_{\max}, \tau - 1 - t)$

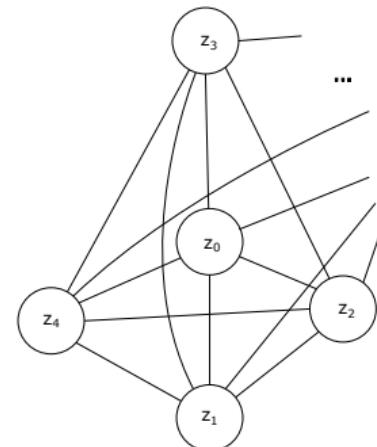


Fallbeispiel Aisin Corporation

- Zeit für Route ξ : $\text{time}(\xi) = \sum_{t=1}^{k-1} T_{\xi_t, \xi_{t+1}}$
- off-board-demand D_{ij} , der von i nach j gebracht werden muss

Terme für QUBO-Formulierung

- $f_{\text{local}}(\{x_i\}) = A_{\text{local}} \sum_t (1 - \sum_i x_{it})^2$
- $f_{\text{time}}(\{x_i\}) = A_{\text{time}} \sum_{ijt} T_{ij} x_{it} x_{jt+1}$
- $f_{\text{demand}}(\{x_i\}) = -A_{\text{demand}} \sum_{ijt} \sum_{\delta=1}^{\delta_*(t)} D_{ij} x_{it} x_{jt+\delta}$ mit $\delta_*(t) = \min(\delta_{\max}, \tau - 1 - t)$
- $f_{\text{nonredundant}}^l(\{x_i\}) = A_{\text{nonredundant}} \sum_{ij} l_{ij} \sum_{\delta=2}^{\tau-2} \sum_{t=0}^{\tau-2-\delta} x_{it} x_{jt+1} x_{it+\delta} x_{jt+1+\delta}$

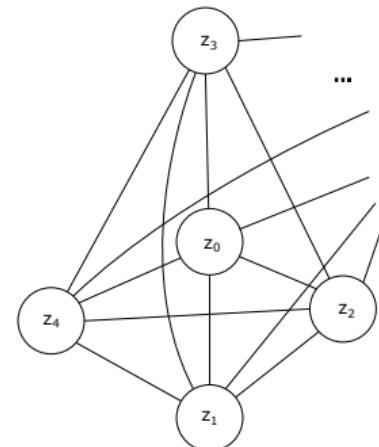


Fallbeispiel Aisin Corporation

- Zeit für Route ξ : $\text{time}(\xi) = \sum_{t=1}^{k-1} T_{\xi_t, \xi_{t+1}}$
- off-board-demand D_{ij} , der von i nach j gebracht werden muss

Terme für QUBO-Formulierung

- $f_{\text{local}}(\{x_i\}) = A_{\text{local}} \sum_t (1 - \sum_i x_{it})^2$
- $f_{\text{time}}(\{x_i\}) = A_{\text{time}} \sum_{ijt} T_{ij} x_{it} x_{jt+1}$
- $f_{\text{demand}}(\{x_i\}) = -A_{\text{demand}} \sum_{ijt} \sum_{\delta=1}^{\delta_*(t)} D_{ij} x_{it} x_{jt+\delta}$ mit $\delta_*(t) = \min(\delta_{\max}, \tau - 1 - t)$
- $f_{\text{nonredundant}}^l(\{x_i\}) = A_{\text{nonredundant}} \sum_{ij} I_{ij} \sum_{\delta=2}^{\tau-2} \sum_{t=0}^{\tau-2-\delta} x_{it} x_{jt+1} x_{it+\delta} x_{jt+1+\delta}$
 - Vorsicht: Grad des Polynoms muss noch auf zwei reduziert werden



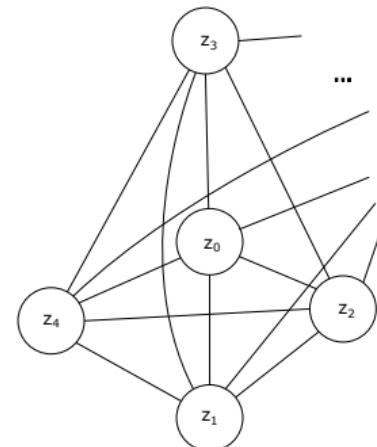
Fallbeispiel Aisin Corporation

- Zeit für Route ξ : $\text{time}(\xi) = \sum_{t=1}^{k-1} T_{\xi_t, \xi_{t+1}}$
- off-board-demand D_{ij} , der von i nach j gebracht werden muss

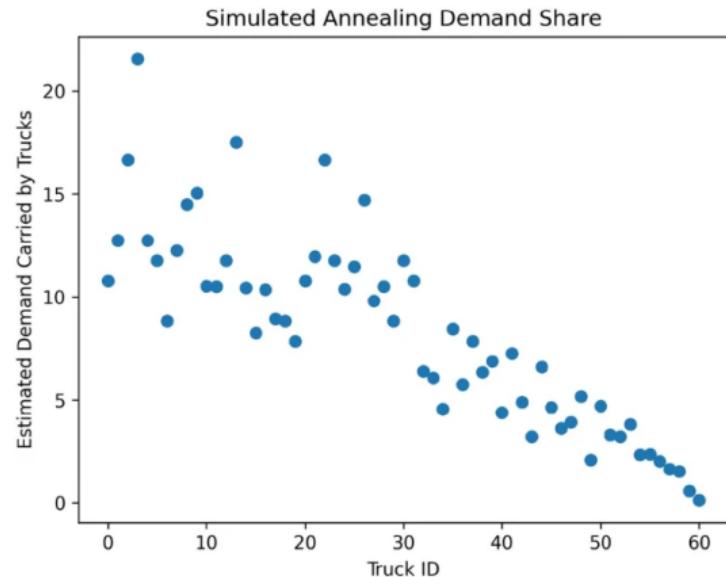
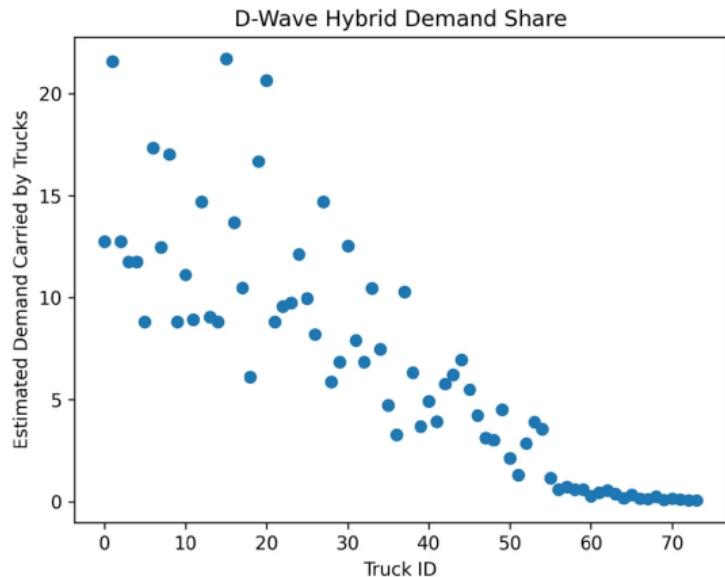
Terme für QUBO-Formulierung

- $f_{\text{local}}(\{x_i\}) = A_{\text{local}} \sum_t (1 - \sum_i x_{it})^2$
- $f_{\text{time}}(\{x_i\}) = A_{\text{time}} \sum_{ijt} T_{ij} x_{it} x_{jt+1}$
- $f_{\text{demand}}(\{x_i\}) = -A_{\text{demand}} \sum_{ijt} \sum_{\delta=1}^{\delta_*(t)} D_{ij} x_{it} x_{jt+\delta}$ mit $\delta_*(t) = \min(\delta_{\max}, \tau - 1 - t)$
- $f_{\text{nonredundant}}^l(\{x_i\}) = A_{\text{nonredundant}} \sum_{ij} I_{ij} \sum_{\delta=2}^{\tau-2} \sum_{t=0}^{\tau-2-\delta} x_{it} x_{jt+1} x_{it+\delta} x_{jt+1+\delta}$
 - Vorsicht: Grad des Polynoms muss noch auf zwei reduziert werden

$$A_{\text{local}} = 5000, \quad A_{\text{demand}} = 320, \quad A_{\text{time}} = 0.01, \quad A_{\text{nonredundant}} = 1$$

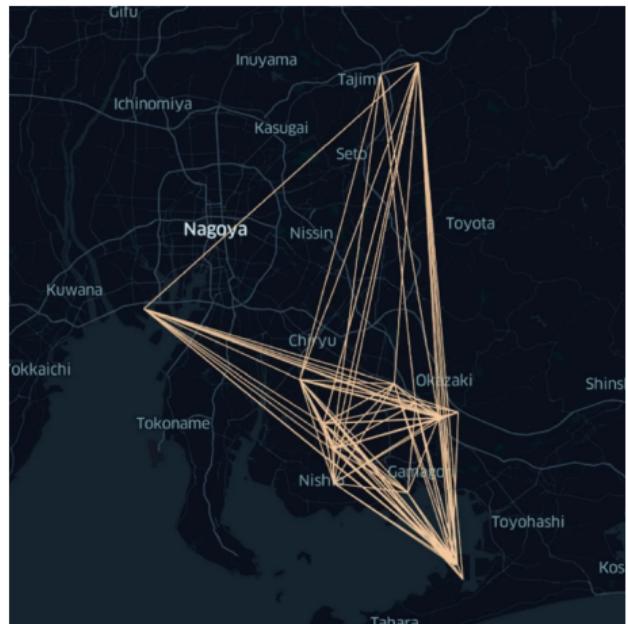
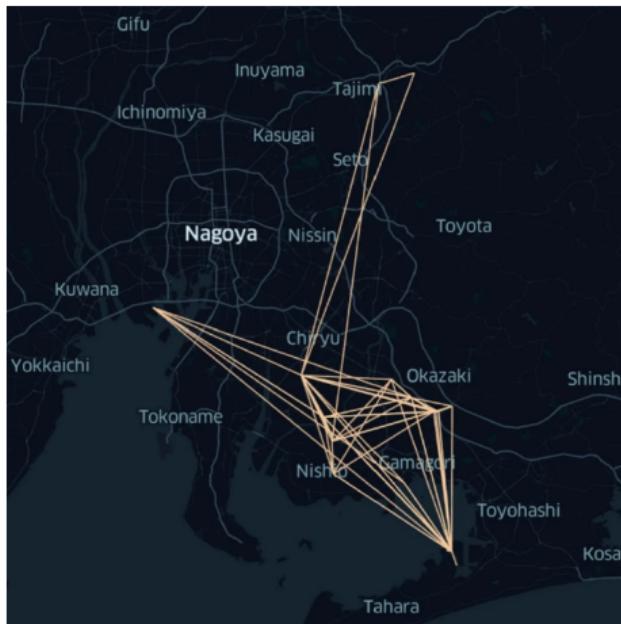


Fallbeispiel Aisin Corporation



Weinberg, S.J., Sanches, F., Ide, T. et al. Sci Rep 13, 4770 (2023)

Fallbeispiel Aisin Corporation



Weinberg, S.J., Sanches, F., Ide, T. et al. Sci Rep 13, 4770 (2023)

Zusammenfassung und Ausblick

Zusammenfassung und Ausblick

- QA als heuristische Methode, um Grundzustand zu finden

Zusammenfassung und Ausblick

- QA als heuristische Methode, um Grundzustand zu finden
- QA bereits angewandt um Logistik (deutlich) zu verbessern

Zusammenfassung und Ausblick

- QA als heuristische Methode, um Grundzustand zu finden
- QA bereits angewandt um Logistik (deutlich) zu verbessern
- QA konvergiert wahrscheinlicher als SA

Zusammenfassung und Ausblick

- QA als heuristische Methode, um Grundzustand zu finden
- QA bereits angewandt um Logistik (deutlich) zu verbessern
- QA konvergiert wahrscheinlicher als SA
- QA und SA nur auf kombinatorische Optimierungsprobleme anwendbar

Zusammenfassung und Ausblick

- QA als heuristische Methode, um Grundzustand zu finden
- QA bereits angewandt um Logistik (deutlich) zu verbessern
- QA konvergiert wahrscheinlicher als SA
- QA und SA nur auf kombinatorische Optimierungsprobleme anwendbar
- Komplexität wächst bei QA und SA exponentiell mit Systemgröße

Zusammenfassung und Ausblick

- QA als heuristische Methode, um Grundzustand zu finden
 - QA bereits angewandt um Logistik (deutlich) zu verbessern
 - QA konvergiert wahrscheinlicher als SA
 - QA und SA nur auf kombinatorische Optimierungsprobleme anwendbar
 - Komplexität wächst bei QA und SA exponentiell mit Systemgröße
-
- mehr qubits
 - größere Probleme optimieren
 - mehr coupler
 - weniger „dummy“-Vertices nötig
 - weniger physische qubits für einen logischen qubit nötig

Quellen

- Bapst, V. et al., The quantum adiabatic algorithm applied to random optimization problems: The quantum Spin glass perspective, ScienceDirect 523 (2013), <https://doi.org/10.1016/j.physrep.2012.10.002>
- Kadowaki, T. et al. Quantum annealing in the transverse Ising model, PHYSICAL REVIEW E 58 (1998), <https://doi.org/10.48550/arXiv.cond-mat/9804280>
- Rieffel, E.G., Venturelli, D., O’Gorman, B. et al. A case study in programming a quantum annealer for hard operational planning problems. Quantum Inf Process 14, 1–36 (2015). <https://doi.org/10.1007/s11128-014-0892-x>
- Choi, V. Minor-embedding in adiabatic quantum computation: I. The parameter setting problem. Quantum Inf Process 7, 193–209 (2008). <https://doi.org/10.1007/s11128-008-0082-9>
- Jain, S, Solving the Traveling Salesman Problem on the D-Wave Quantum Computer, Frontiers in Physics, Volume 9, id.646 (2021) ,doi: 10.3389/fphy.2021.760783
- Sangwan, Shabnam. (2018). Literature Review on Travelling Salesman Problem. International Journal of Research. 5. 1152.
- Weinberg, S.J., Sanches, F., Ide, T. et al. Supply chain logistics with quantum and classical annealing algorithms. Sci Rep 13, 4770 (2023). <https://doi.org/10.1038/s41598-023-31765-8>
- Yarkoni, S et al., Quantum annealing for industry applications: introduction and review, IOP Publishing 85-10 (2022), <https://dx.doi.org/10.1088/1361-6633/ac8c54>

Quellen

- <https://www.dwavesys.com/solutions-and-products/systems/>
- <https://techhq.com/2023/05/industrial-applications-of-quantum-computing-add-up-for-dwave/>
- <https://www.science.org/content/article/quantum-or-not-controversial-computer-runs-no-faster-normal-one>

Backup-Folien für Detailfragen

$$\begin{aligned}
 \mathcal{H}_{\text{final}}|z\rangle &= \left(\sum_{i \in V(G)} h_i \sigma_i^z + \sum_{ij \in E(G)} J_{ij} \sigma_i^z \sigma_j^z \right) |z_1\rangle \otimes \dots \otimes |z_n\rangle \\
 &= \sum_{i \in V(G)} h_i |z_1\rangle \otimes \dots \otimes (-1)^{z_i} |z_i\rangle \dots \otimes |z_n\rangle + \sum_{ij \in E(G)} J_{ij} |z_1\rangle \otimes \dots \otimes (-1)^{z_i} |z_i\rangle \dots \otimes (-1)^{z_j} |z_j\rangle \dots \otimes |z_n\rangle \\
 &= \left(\sum_{i \in V(G)} h_i (-1)^{z_i} + \sum_{ij \in E(G)} J_{ij} (-1)^{z_i} (-1)^{z_j} \right) |z_1\rangle \otimes \dots \otimes |z_n\rangle
 \end{aligned}$$

Therefore, the energy function of $\mathcal{H}_{\text{final}}$ is

$$\mathcal{E}(z_1, \dots, z_n) = \sum_{i \in V(G)} h_i (-1)^{z_i} + \sum_{ij \in E(G)} J_{ij} (-1)^{z_i} (-1)^{z_j}$$

where $z_i \in \{0, 1\}$, for $i = 1, \dots, n$. Replace $(-1)^{z_i}$ by $s_i \in \{+1, -1\}$, we thus have Eq.(2).

Backup-Folien für Detailfragen

Execution and performance

The purpose of the truck loop algorithm of “[Supply chain workflow](#)” is to have an algorithm that can break a very large commercial problem into sufficiently small pieces that simulated and quantum annealing algorithms can viably be applied in the near term. As a proof-of-concept, we applied our algorithm to proprietary supply chain data of Aisin Corporation using both simulated annealing and the D-Wave Hybrid algorithm. Application of direct quantum annealing rather than the D-Wave Hybrid algorithm remains unrealistic for this problem until the hardware matures to some extent.

We referred to the Aisin problem, which was referred to as the ALRP in “[Aisin Corporation vehicle routing problem](#)”. Their data consists of approximately 350,000 boxes to be shipped by trucks among 23 nodes in Japan. The travel times between nodes is shown in Fig. 2. The boxes have rank-2 and rank-3 demand structure. Thus, some boxes have direct delivery requirements and others have a required stop along the way. To apply the truck loop algorithm, we first group boxes by their required routes and then we apply the “box soup” simplification of “[Individual boxes and the box soupsimplification](#)” to convert from a discrete to continuous demand structure. Moreover, we use Eq. (5) to compute the overall demand matrix. With truck volumes normalized to 1, the overall demand matrix is illustrated in Fig. 3.

Truck loop annealing runs. We ran the truck loop algorithm of “[Supply chain workflow](#)” using both a simulated annealing algorithm (“B. 1. Simulated annealing”) and the D-Wave Hybrid solver (“B. 2. Quantum annealing”).

During each step of the loop, we computed an estimated overall demand carried by each truck. That estimated overall demand is shown for the case of simulated annealing and D-Wave Hybrid in Figs. 4 and 5, respectively. The simulated annealing loop terminated after 61 trucks, and the D-Wave Hybrid loop terminated after 74 trucks. This termination condition, defining `stop` from “[Supply chain workflow](#)”, is simply to stop all entries of D under a very small cutoff (.0005 in our case).

To construct the single-truck PUBO, we used a different value of τ (the number of time steps) for simulated annealing and for the D-Wave Hybrid solver: 15 for simulated annealing and 5 for D-Wave Hybrid. With 15 time steps, the typical number of PUBO variables is approximately 350, corresponding to approximately 2500 QUBO variables. With only 5 time steps, there are around 100 PUBO variables which are equivalent to about 200 QUBO variables.

Backup-Folien für Detailfragen

or both the simulated annealing routes and the D-Wave Hybrid routes, we kept only the first 61 trucks. We found that these trucks satisfied 96.45 percent of the overall Aisin Corp. demand when using simulated annealing and 99.39 percent with the D-Wave Hybrid solver.

Kritik

- QA skaliert mit Systemgröße wie normaler Computer
- Rechenzeit sollte bei Quantencomputer mit Systemgröße langsamer als bei normalen Computer skalieren
- In particular, a full-fledged "universal" quantum computer should be able to factor huge numbers ever faster than an ordinary computer as the size of the numbers grow.
- Daniel Lidar (University of Southern California in Los Angeles) hat random WW zw. qubits simuliert und gemerkt, dass das Problem exponentiell mit Systemgröße wächst wie bei normalem Computer
- Texas A und M's Katzgraber sagt aber, dass GZ bei random WW typischerweise nur bei $T = 0$ erreicht werden kann und die Tests somit sinnlos sind
- Hartmut Neven, director of engineering at Google, sagt, dass es Klassen von Problemen gibt, wo ein next-generation quantum annealer jeglichen klassischen Algorithmus schlagen wird
- Quelle <https://www.science.org/content/article/quantum-or-not-controversial-computer-runs-no-faster-normal-one>