

Algoritmi di ottimizzazione su computer quantistici

(seminario didattico!)

F. Palombi



21 Giugno 2018 @ ENEA - Frascati

Problema

Scopo: minimizzare la funzione $f(x): \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$

Problema

Scopo: minimizzare la funzione $f(x): \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$

$x = \{x_1, \dots, x_n\} \equiv$ configurazione o stato del sistema

Problema

Scopo: minimizzare la funzione $f(x): \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$

$x = \{x_1, \dots, x_n\} \equiv$ configurazione o stato del sistema

$f(x) \equiv$ energia del sistema

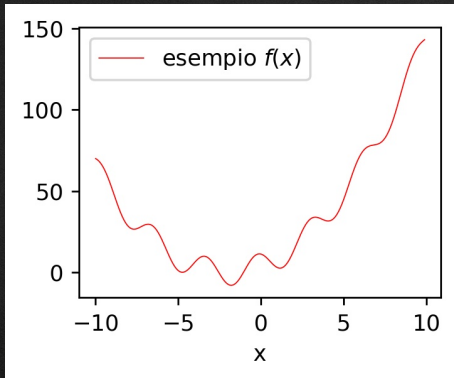
Problema

Scopo: minimizzare la funzione $f(x): \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$

$x = \{x_1, \dots, x_n\} \equiv$ configurazione o stato del sistema

$f(x) \equiv$ energia del sistema

In generale molti minimi locali. Distinguere dal minimo globale!



Simulated Annealing

Scopo: minimizzare la funzione $f(x): \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$

Simulated Annealing

Scopo: minimizzare la funzione $f(x): \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$

Un algoritmo che usa le **fluttuazioni termiche** in meccanica statistica classica

Simulated Annealing

Scopo: minimizzare la funzione $f(x): \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$

Un algoritmo che usa le **fluttuazioni termiche** in meccanica statistica classica

- Costruiamo una successione di configurazioni $x^{(0)} \rightarrow x^{(1)} \rightarrow x^{(2)} \rightarrow \dots$

Simulated Annealing

Scopo: minimizzare la funzione $f(x): \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$

Un algoritmo che usa le **fluttuazioni termiche** in meccanica statistica classica

- Costruiamo una successione di configurazioni $x^{(0)} \rightarrow x^{(1)} \rightarrow x^{(2)} \rightarrow \dots$
- $x^{(0)}$ random. $x^{(k+1)}$ costruita da $x^{(k)}$ modificando a caso un elemento scelto a caso

Simulated Annealing

Scopo: minimizzare la funzione $f(x): \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$

Un algoritmo che usa le **fluttuazioni termiche** in meccanica statistica classica

- Costruiamo una successione di configurazioni $x^{(0)} \rightarrow x^{(1)} \rightarrow x^{(2)} \rightarrow \dots$
- $x^{(0)}$ random. $x^{(k+1)}$ costruita da $x^{(k)}$ modificando a caso un elemento scelto a caso
- $\Delta f = f(x^{(k+1)}) - f(x^{(k)})$ è la variazione di f

Simulated Annealing

Scopo: minimizzare la funzione $f(x): \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$

Un algoritmo che usa le **fluttuazioni termiche** in meccanica statistica classica

- Costruiamo una successione di configurazioni $x^{(0)} \rightarrow x^{(1)} \rightarrow x^{(2)} \rightarrow \dots$
- $x^{(0)}$ random. $x^{(k+1)}$ costruita da $x^{(k)}$ modificando a caso un elemento scelto a caso
- $\Delta f = f(x^{(k+1)}) - f(x^{(k)})$ è la variazione di f

Se $\Delta f < 0$ accettiamo la nuova configurazione (Go down the hill...)

Simulated Annealing

Scopo: minimizzare la funzione $f(x): \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$

Un algoritmo che usa le **fluttuazioni termiche** in meccanica statistica classica

- Costruiamo una successione di configurazioni $x^{(0)} \rightarrow x^{(1)} \rightarrow x^{(2)} \rightarrow \dots$
- $x^{(0)}$ random. $x^{(k+1)}$ costruita da $x^{(k)}$ modificando a caso un elemento scelto a caso
- $\Delta f = f(x^{(k+1)}) - f(x^{(k)})$ è la variazione di f

Se $\Delta f < 0$ accettiamo la nuova configurazione (Go down the hill...)

Se $\Delta f > 0$ accettiamo la nuova configurazione purché Δf ragionevolmente piccolo

Simulated Annealing

Scopo: minimizzare la funzione $f(x): \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$

Un algoritmo che usa le **fluttuazioni termiche** in meccanica statistica classica

- Costruiamo una successione di configurazioni $x^{(0)} \rightarrow x^{(1)} \rightarrow x^{(2)} \rightarrow \dots$
- $x^{(0)}$ random. $x^{(k+1)}$ costruita da $x^{(k)}$ modificando a caso un elemento scelto a caso
- $\Delta f = f(x^{(k+1)}) - f(x^{(k)})$ è la variazione di f

Se $\Delta f < 0$ accettiamo la nuova configurazione (Go down the hill...)

Se $\Delta f > 0$ accettiamo la nuova configurazione purché Δf ragionevolmente piccolo

- introduciamo un parametro di controllo $T > 0$ (temperatura)

Simulated Annealing

Scopo: minimizzare la funzione $f(x): \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$

Un algoritmo che usa le **fluttuazioni termiche** in meccanica statistica classica

- Costruiamo una successione di configurazioni $x^{(0)} \rightarrow x^{(1)} \rightarrow x^{(2)} \rightarrow \dots$
- $x^{(0)}$ random. $x^{(k+1)}$ costruita da $x^{(k)}$ modificando a caso un elemento scelto a caso
- $\Delta f = f(x^{(k+1)}) - f(x^{(k)})$ è la variazione di f

Se $\Delta f < 0$ accettiamo la nuova configurazione (Go down the hill...)

Se $\Delta f > 0$ accettiamo la nuova configurazione purché Δf ragionevolmente piccolo

- introduciamo un parametro di controllo $T > 0$ (temperatura)
estraiamo $r \in U(0,1)$ random uniforme

Simulated Annealing

Scopo: minimizzare la funzione $f(x): \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$

Un algoritmo che usa le **fluttuazioni termiche** in meccanica statistica classica

- Costruiamo una successione di configurazioni $x^{(0)} \rightarrow x^{(1)} \rightarrow x^{(2)} \rightarrow \dots$
- $x^{(0)}$ random. $x^{(k+1)}$ costruita da $x^{(k)}$ modificando a caso un elemento scelto a caso
- $\Delta f = f(x^{(k+1)}) - f(x^{(k)})$ è la variazione di f

Se $\Delta f < 0$ accettiamo la nuova configurazione (Go down the hill...)

Se $\Delta f > 0$ accettiamo la nuova configurazione purché Δf ragionevolmente piccolo

- introduciamo un parametro di controllo $T > 0$ (temperatura)
estriamo $r \in U(0,1)$ random uniforme
accettiamo $x^{(k+1)}$ se $r < e^{-\Delta f/T}$

Simulated Annealing

Scopo: minimizzare la funzione $f(x): \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$

Un algoritmo che usa le **fluttuazioni termiche** in meccanica statistica classica

- Costruiamo una successione di configurazioni $x^{(0)} \rightarrow x^{(1)} \rightarrow x^{(2)} \rightarrow \dots$
- $x^{(0)}$ random. $x^{(k+1)}$ costruita da $x^{(k)}$ modificando a caso un elemento scelto a caso
- $\Delta f = f(x^{(k+1)}) - f(x^{(k)})$ è la variazione di f

Se $\Delta f < 0$ accettiamo la nuova configurazione (Go down the hill...)

Se $\Delta f > 0$ accettiamo la nuova configurazione purché Δf ragionevolmente piccolo

- introduciamo un parametro di controllo $T > 0$ (temperatura)

estriamo $r \in U(0,1)$ random uniforme

accettiamo $x^{(k+1)}$ se $r < e^{-\Delta f/T}$

- DOPO N ITERAZIONI RIDUCIAMO T E PROSEGUIAMO (T-schedule)

Simulated Annealing

Scopo: minimizzare la funzione $f(x): \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$

Un algoritmo che usa le **fluttuazioni termiche** in meccanica statistica classica

$T = T_{\text{start}}$

$x = x_{\text{start}}$

while criterio di arresto non soddisfatto **do**

for iter from 1 to max-iter **do**

$y = \text{proposta vicina a } x$

if $f(y) < f(x)$ **then**

$x = y$

else

$r = \text{numero random in } U(0,1)$

$\Delta f = f(y) - f(x) > 0$

if $r < \exp\{-\Delta f / T\}$ **then**

$x = y$

end if

end if

end do

$T = \alpha T$ (con $\alpha < 1$)

end while

Esempio: Sudoku

			9			7	2	8
2	7	8			3		1	
	9					6	4	
	5			6		2		
		6				3		
	1			5				
1			7		6		3	4
			5		4			
7		9	1			8		5

- Un tabellone 9×9 suddiviso in 9 regioni 3×3

Esempio: Sudoku

			9			7	2	8
2	7	8			3		1	
	9					6	4	
	5			6		2		
		6				3		
	1			5				
1			7		6		3	4
			5		4			
7		9	1			8		5

- Un tabellone 9×9 suddiviso in 9 regioni 3×3
- 25÷32 celle sono inizialmente piene (31 celle piene in figura, $81 - 31 = 50$ celle vuote)

Esempio: Sudoku

			9			7	2	8
2	7	8			3		1	
	9					6	4	
	5			6		2		
		6				3		
	1			5				
1			7		6		3	4
			5		4			
7		9	1			8		5

- Un tabellone 9×9 suddiviso in 9 regioni 3×3
- 25÷32 celle sono inizialmente piene (31 celle piene in figura, $81 - 31 = 50$ celle vuote)
- Scopo del gioco: riempire tutte le celle vuote in modo tale che ogni riga, colonna e/o regione contenga tutti i numeri da 1 a 9

Esempio: Sudoku

			9			7	2	8
2	7	8			3		1	
	9					6	4	
	5			6		2		
		6				3		
	1			5				
1			7		6		3	4
			5		4			
7		9	1			8		5

- Calcolo **brute force**: 50 celle da riempire

Esempio: Sudoku

			9			7	2	8
2	7	8			3		1	
	9					6	4	
	5			6		2		
		6				3		
	1			5				
1			7		6		3	4
			5		4			
7		9	1			8		5

- Calcolo **brute force**: 50 celle da riempire
- Ogni cella può contenere un numero 1,...,9

Esempio: Sudoku

			9			7	2	8
2	7	8			3		1	
	9					6	4	
	5			6		2		
		6				3		
	1			5				
1			7		6		3	4
			5		4			
7		9	1			8		5

- Calcolo **brute force**: 50 celle da riempire
- Ogni cella può contenere un numero 1,...,9
- # soluzioni = $9^{50} = 515377520732011331036461129765621272702107522001 \approx 5 \times 10^{47}$

Esempio: Sudoku

			9			7	2	8
2	7	8			3		1	
	9					6	4	
	5			6		2		
		6				3		
	1			5				
1			7		6		3	4
			5		4			
7		9	1			8		5

- Calcolo **brute force**: 50 celle da riempire
- Ogni cella può contenere un numero 1,...,9
- # soluzioni = $9^{50} = 515377520732011331036461129765621272702107522001 \approx 5 \times 10^{47}$
- Su CRESCO6 (700 TFLOP/sec) il tempo richiesto sarebbe $\approx 7.36 \times 10^{32}$ sec

Esempio: Sudoku

			9			7	2	8
2	7	8			3		1	
	9					6	4	
	5			6		2		
		6				3		
	1			5				
1			7		6		3	4
			5		4			
7		9	1			8		5

- Calcolo **brute force**: 50 celle da riempire
- Ogni cella può contenere un numero 1,...,9
- # soluzioni = $9^{50} = 515377520732011331036461129765621272702107522001 \approx 5 \times 10^{47}$
- Su CRESCO6 (700 TFLOP/sec) il tempo richiesto sarebbe $\approx 7.36 \times 10^{32}$ sec
- Età dell'Universo: $O(10^{17})$ sec Non sembra una buona idea!

Esempio: Sudoku

			9			7	2	8
2	7	8			3		1	
	9					6	4	
	5			6		2		
		6				3		
	1			5				
1			7		6		3	4
			5		4			
7		9	1			8		5

- Proviamo a risolvere attraverso simulated annealing

Esempio: Sudoku

			9			7	2	8
2	7	8			3		1	
	9					6	4	
	5			6		2		
		6				3		
	1			5				
1			7		6		3	4
			5		4			
7		9	1			8		5

- Proviamo a risolvere attraverso simulated annealing
- Assegniamo ad ogni riga, colonna, regione un punteggio da 1 a 9

Esempio: Sudoku

			9			7	2	8
2	7	8			3		1	
	9					6	4	
	5			6		2		
		6				3		
	1			5				
1			7		6		3	4
			5		4			
7		9	1			8		5

- Proviamo a risolvere attraverso simulated annealing
- Assegniamo ad ogni riga, colonna, regione un punteggio da 1 a 9
- Il punteggio conta le celle il cui contenuto è unico nella riga, colonna, regione

Esempio: Sudoku

			9			7	2	8
2	7	8			3		1	
	9					6	4	
	5			6		2		
		6				3		
	1			5				
1			7		6		3	4
			5		4			
7		9	1			8		5

- Proviamo a risolvere attraverso simulated annealing
- Assegniamo ad ogni riga, colonna, regione un punteggio da 1 a 9
- Il punteggio conta le celle il cui contenuto è unico nella riga, colonna, regione
- **Punteggio massimo: $9 \times 9 \times 3 = 243$**

Esempio: Sudoku

			9			7	2	8
2	7	8			3		1	
	9					6	4	
	5			6		2		
		6				3		
	1			5				
1			7		6		3	4
			5		4			
7		9	1			8		5

- Proviamo a risolvere attraverso simulated annealing
- Assegniamo ad ogni riga, colonna, regione un punteggio da 1 a 9
- Il punteggio conta le celle il cui contenuto è unico nella riga, colonna, regione
- **Punteggio massimo: $9 \times 9 \times 3 = 243$**
- Definiamo l'energia di una configurazione $E[\text{conf.}] = 243 - \sum_{X=\text{riga,colonna,regione}} \text{punteggio}(X)$

Esempio: Sudoku

6	3	4	9	1	5	7	2	8
2	7	8	6	4	3	5	1	9
5	9	1	2	7	8	6	4	3
4	5	7	3	6	9	2	8	1
9	8	6	4	2	1	3	5	7
3	1	2	8	5	7	4	9	6
1	2	5	7	8	6	9	3	4
8	6	3	5	9	4	1	7	2
7	4	9	1	3	2	8	6	5

- Soluzione in $O(10^4) \div O(10^5)$ passi

Esempio: Sudoku

6	3	4	9	1	5	7	2	8
2	7	8	6	4	3	5	1	9
5	9	1	2	7	8	6	4	3
4	5	7	3	6	9	2	8	1
9	8	6	4	2	1	3	5	7
3	1	2	8	5	7	4	9	6
1	2	5	7	8	6	9	3	4
8	6	3	5	9	4	1	7	2
7	4	9	1	3	2	8	6	5

- Soluzione in $O(10^4) \div O(10^5)$ passi
- Tempo di calcolo $O(10)$ sec su un PC con hardware ragionevole...

Esempio: Sudoku

6	3	4	9	1	5	7	2	8
2	7	8	6	4	3	5	1	9
5	9	1	2	7	8	6	4	3
4	5	7	3	6	9	2	8	1
9	8	6	4	2	1	3	5	7
3	1	2	8	5	7	4	9	6
1	2	5	7	8	6	9	3	4
8	6	3	5	9	4	1	7	2
7	4	9	1	3	2	8	6	5

- Soluzione in $O(10^4) \div O(10^5)$ passi
- Tempo di calcolo $O(10)$ sec su un PC con hardware ragionevole...
- Soluzione facilmente verificabile

Fluttuazioni Quantistiche

E' possibile riformulare quantisticamente il Simulated Annealing in modo da usare l'effetto tunnel al posto delle fluttuazioni termiche?

Fluttuazioni Quantistiche

E' possibile riformulare quantisticamente il Simulated Annealing in modo da usare l'effetto tunnel al posto delle fluttuazioni termiche?

- Consideriamo un'Hamiltoniana $H(t)$ dipendente dal tempo

Fluttuazioni Quantistiche

E' possibile riformulare quantisticamente il Simulated Annealing in modo da usare l'effetto tunnel al posto delle fluttuazioni termiche?

- Consideriamo un'Hamiltoniana $H(t)$ dipendente dal tempo
- Introduciamo una scala temporale esterna τ , poniamo $s = t/\tau$ e definiamo

$$H(t) = \tilde{H}\left(\frac{t}{\tau}\right) = \tilde{H}(s)$$

Fluttuazioni Quantistiche

E' possibile riformulare quantisticamente il Simulated Annealing in modo da usare l'effetto tunnel al posto delle fluttuazioni termiche?

- Consideriamo un'Hamiltoniana $H(t)$ dipendente dal tempo
- Introduciamo una scala temporale esterna τ , poniamo $s = t/\tau$ e definiamo

$$H(t) = \tilde{H}\left(\frac{t}{\tau}\right) = \tilde{H}(s)$$

- Indichiamo con $\epsilon_k(s)$ e $|k(s)\rangle$ autovalori e autoket di $\tilde{H}(s)$

$$\tilde{H}(s)|k(s)\rangle = \epsilon_k(s)|k(s)\rangle$$

Fluttuazioni Quantistiche

E' possibile riformulare quantisticamente il Simulated Annealing in modo da usare l'effetto tunnel al posto delle fluttuazioni termiche?

- Consideriamo un'Hamiltoniana $H(t)$ dipendente dal tempo
- Introduciamo una scala temporale esterna τ , poniamo $s = t/\tau$ e definiamo
$$H(t) = \tilde{H}\left(\frac{t}{\tau}\right) = \tilde{H}(s)$$
- Indichiamo con $\epsilon_k(s)$ e $|k(s)\rangle$ autovalori e autoket di $\tilde{H}(s)$
$$\tilde{H}(s)|k(s)\rangle = \epsilon_k(s)|k(s)\rangle$$
- supponiamo di saper diagonalizzare $\tilde{H}(0)$ e poniamo $|\tilde{\psi}(0)\rangle = |0(0)\rangle$

Fluttuazioni Quantistiche

E' possibile riformulare quantisticamente il Simulated Annealing in modo da usare l'effetto tunnel al posto delle fluttuazioni termiche?

- Consideriamo un'Hamiltoniana $H(t)$ dipendente dal tempo
- Introduciamo una scala temporale esterna τ , poniamo $s = t/\tau$ e definiamo

$$H(t) = \tilde{H}\left(\frac{t}{\tau}\right) = \tilde{H}(s)$$

- Indichiamo con $\epsilon_k(s)$ e $|k(s)\rangle$ autovalori e autoket di $\tilde{H}(s)$

$$\tilde{H}(s)|k(s)\rangle = \epsilon_k(s)|k(s)\rangle$$

- supponiamo di saper diagonalizzare $\tilde{H}(0)$ e poniamo $|\tilde{\psi}(0)\rangle = |0(0)\rangle$

- Il ket di stato $|\tilde{\psi}(s)\rangle$ segue l'equazione di Schrödinger

$$i\hbar \frac{d}{ds} |\tilde{\psi}(s)\rangle = \tau \tilde{H}(s) |\tilde{\psi}(s)\rangle$$

Fluttuazioni Quantistiche

E' possibile riformulare quantisticamente il Simulated Annealing in modo da usare l'effetto tunnel al posto delle fluttuazioni termiche?

- Consideriamo un'Hamiltoniana $H(t)$ dipendente dal tempo
- Introduciamo una scala temporale esterna τ , poniamo $s = t/\tau$ e definiamo

$$H(t) = \tilde{H}\left(\frac{t}{\tau}\right) = \tilde{H}(s)$$

- Indichiamo con $\epsilon_k(s)$ e $|k(s)\rangle$ autovalori e autoket di $\tilde{H}(s)$

$$\tilde{H}(s)|k(s)\rangle = \epsilon_k(s)|k(s)\rangle$$

- supponiamo di saper diagonalizzare $\tilde{H}(0)$ e poniamo $|\tilde{\psi}(0)\rangle = |0(0)\rangle$

- Il ket di stato $|\tilde{\psi}(s)\rangle$ segue l'equazione di Schrödinger

$$i\hbar \frac{d}{ds} |\tilde{\psi}(s)\rangle = \tau \tilde{H}(s) |\tilde{\psi}(s)\rangle$$

- Domanda: quanto $|\tilde{\psi}(s)\rangle$ differisce da $|0(s)\rangle$?

Teorema Adiabatico

Teorema

Se lo stato fondamentale dell'Hamiltoniana $\tilde{H}(s)$ è non degenere per $s \geq 0$ e il sistema si trova nello stato fondamentale per $s = 0$, cioè $|\tilde{\psi}(0)\rangle = |0(0)\rangle$, allora nel limite $\tau \rightarrow \infty$ il vettore di stato $|\psi'(s)\rangle$ ha forma asintotica

$$|\tilde{\psi}(s)\rangle = \sum_k c_k(s) e^{-i\tau\phi_k(s)} |k(s)\rangle$$

dove

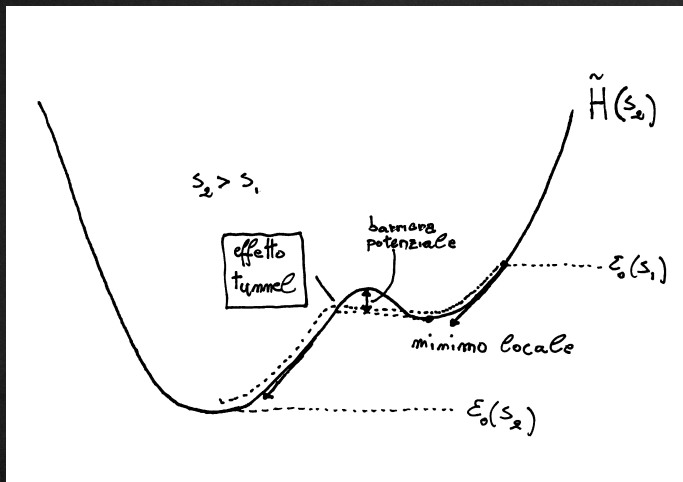
$$c_0(s) = 1 + \mathcal{O}(\tau^{-2})$$

$$c_{k \neq 0}(s) = \frac{i}{\tau} \left[A_k(0) - e^{i\tau[\phi_k(s) - \phi_k(0)]} A_k(s) \right] + \mathcal{O}(\tau^{-2})$$

e

$$\phi_k(s) = \int_0^s ds' \epsilon_k(s'), \quad A_k(s) = \frac{1}{[\epsilon_k(s) - \epsilon_k(0)]^2} \langle k(s) | \frac{d\tilde{H}(s)}{ds} | 0(s) \rangle$$

Teorema Adiabatico



Quantum Annealing

- Consideriamo un operatore Hamiltoniano della forma

$$\tilde{H}(s) = A(s) H_{\text{friend}} + B(s) H_{\text{target}}$$

Quantum Annealing

- Consideriamo un operatore Hamiltoniano della forma

$$\tilde{H}(s) = A(s) H_{\text{friend}} + B(s) H_{\text{target}}$$

- H_{friend} e H_{target} sono indipendenti dal tempo

Quantum Annealing

- Consideriamo un operatore Hamiltoniano della forma

$$\tilde{H}(s) = A(s) H_{\text{friend}} + B(s) H_{\text{target}}$$

- H_{friend} e H_{target} sono indipendenti dal tempo
- H_{friend} è diagonalizzabile esattamente

Quantum Annealing

- Consideriamo un operatore Hamiltoniano della forma

$$\tilde{H}(s) = A(s) H_{\text{friend}} + B(s) H_{\text{target}}$$

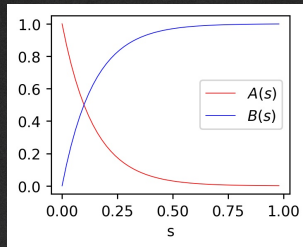
- H_{friend} e H_{target} sono indipendenti dal tempo
- H_{friend} è diagonalizzabile esattamente
- H_{target} coincide con la funzione che vogliamo minimizzare

Quantum Annealing

- Consideriamo un operatore Hamiltoniano della forma

$$\tilde{H}(s) = A(s) H_{\text{friend}} + B(s) H_{\text{target}}$$

- H_{friend} e H_{target} sono indipendenti dal tempo
- H_{friend} è diagonalizzabile esattamente
- H_{target} coincide con la funzione che vogliamo minimizzare
- Le funzioni $A(s)$ e $B(s)$ sono riportate in figura

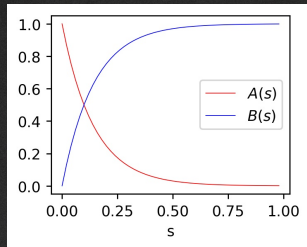


Quantum Annealing

- Consideriamo un operatore Hamiltoniano della forma

$$\tilde{H}(s) = A(s) H_{\text{friend}} + B(s) H_{\text{target}}$$

- H_{friend} e H_{target} sono indipendenti dal tempo
- H_{friend} è diagonalizzabile esattamente
- H_{target} coincide con la funzione che vogliamo minimizzare
- Le funzioni $A(s)$ e $B(s)$ sono riportate in figura



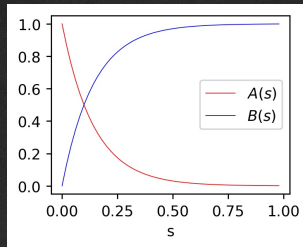
- Prepariamo il sistema nello stato fondamentale di H_{friend} per $s = 0$

Quantum Annealing

- Consideriamo un operatore Hamiltoniano della forma

$$\tilde{H}(s) = A(s) H_{\text{friend}} + B(s) H_{\text{target}}$$

- H_{friend} e H_{target} sono indipendenti dal tempo
- H_{friend} è diagonalizzabile esattamente
- H_{target} coincide con la funzione che vogliamo minimizzare
- Le funzioni $A(s)$ e $B(s)$ sono riportate in figura



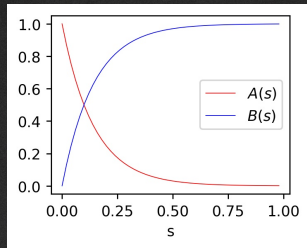
- Prepariamo il sistema nello stato fondamentale di H_{friend} per $s = 0$
- Lasciamo che il sistema evolva liberamente per $s > 0$

Quantum Annealing

- Consideriamo un operatore Hamiltoniano della forma

$$\tilde{H}(s) = A(s) H_{\text{friend}} + B(s) H_{\text{target}}$$

- H_{friend} e H_{target} sono indipendenti dal tempo
- H_{friend} è diagonalizzabile esattamente
- H_{target} coincide con la funzione che vogliamo minimizzare
- Le funzioni $A(s)$ e $B(s)$ sono riportate in figura



- Prepariamo il sistema nello stato fondamentale di H_{friend} per $s = 0$
- Lasciamo che il sistema evolva liberamente per $s > 0$
- Se la variazione di $A(s)$ e $B(s)$ è sufficientemente lenta, per il **teorema adiabatico** il sistema rilassa allo stato fondamentale di H_{target} per $s \rightarrow 1$

Modello di Ising

- Un modello per descrivere la transizione di fase **ferromagnetica**



Modello di Ising

- Un modello per descrivere la transizione di fase **ferromagnetica**
- Lo **spin** (momento angolare intrinseco) è una variabile classica $\sigma = \pm 1$



Modello di Ising

- Un modello per descrivere la transizione di fase **ferromagnetica**
- Lo **spin** (momento angolare intrinseco) è una variabile classica $\sigma = \pm 1$
- Consideriamo un **reticolo 2-dimensionale** di N atomi dotati di spin

$$H = -\sum_k h_k \sigma_k - \sum_{i,k} J_{ik} \sigma_i \sigma_k$$



Modello di Ising



- Un modello per descrivere la transizione di fase **ferromagnetica**
- Lo **spin** (momento angolare intrinseco) è una variabile classica $\sigma = \pm 1$
- Consideriamo un **reticolo 2-dimensionale** di N atomi dotati di spin

$$H = -\sum_k h_k \sigma_k - \sum_{i,k} J_{ik} \sigma_i \sigma_k$$

- h_k = campo magnetico esterno su atomo k
- J_{ik} = costante d'accoppiamento atomi i e k
- $\sigma = \{\sigma_1, \dots, \sigma_N\}$ = configurazione di spin

Modello di Ising



- Un modello per descrivere la transizione di fase **ferromagnetica**
- Lo **spin** (momento angolare intrinseco) è una variabile classica $\sigma = \pm 1$
- Consideriamo un **reticolo 2-dimensionale** di N atomi dotati di spin

$$H = -\sum_k h_k \sigma_k - \sum_{i,k} J_{ik} \sigma_i \sigma_k$$

- h_k = campo magnetico esterno su atomo k
- J_{ik} = costante d'accoppiamento atomi i e k
- $\sigma = \{\sigma_1, \dots, \sigma_N\}$ = configurazione di spin

-
- 1) Se $\{h_k = 0\}$ e $\{J_{ik} > 0\}$, due configurazioni di minimo: $\begin{cases} \sigma_+ = \{+1, +1, \dots, +1\} \\ \sigma_- = \{-1, -1, \dots, -1\} \end{cases}$
- 2) Se $\{h_k > 0\}$ e $\{J_{ik} > 0\}$, una configurazione di minimo: $\sigma_+ = \{+1, +1, \dots, +1\}$
- 3) Se $\{h_k < 0\}$ e $\{J_{ik} > 0\}$, una configurazione di minimo: $\sigma_- = \{-1, -1, \dots, -1\}$

Modello di Ising



- Un modello per descrivere la transizione di fase **ferromagnetica**
- Lo **spin** (momento angolare intrinseco) è una variabile classica $\sigma = \pm 1$
- Consideriamo un **reticolo 2-dimensionale** di N atomi dotati di spin

$$H = -\sum_k h_k \sigma_k - \sum_{i,k} J_{ik} \sigma_i \sigma_k$$

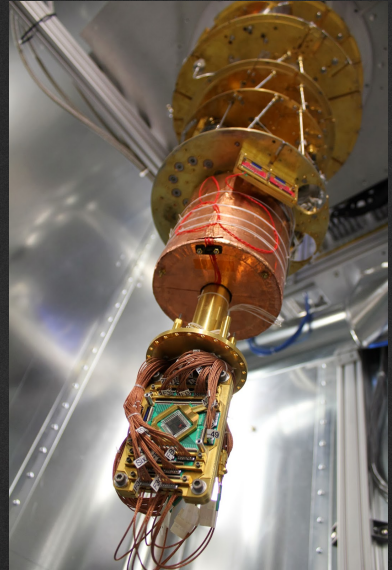
- h_k = campo magnetico esterno su atomo k
- J_{ik} = costante d'accoppiamento atomi i e k
- $\sigma = \{\sigma_1, \dots, \sigma_N\}$ = configurazione di spin

-
- 1) Se $\{h_k = 0\}$ e $\{J_{ik} > 0\}$, due configurazioni di minimo: $\begin{cases} \sigma_+ = \{+1, +1, \dots, +1\} \\ \sigma_- = \{-1, -1, \dots, -1\} \end{cases}$
- 2) Se $\{h_k > 0\}$ e $\{J_{ik} > 0\}$, una configurazione di minimo: $\sigma_+ = \{+1, +1, \dots, +1\}$
- 3) Se $\{h_k < 0\}$ e $\{J_{ik} > 0\}$, una configurazione di minimo: $\sigma_- = \{-1, -1, \dots, -1\}$
- X) Se $\{J_{ik}\}$ ha entrambi i segni (**vetri di spin**), il minimo globale è **complicato!!!**

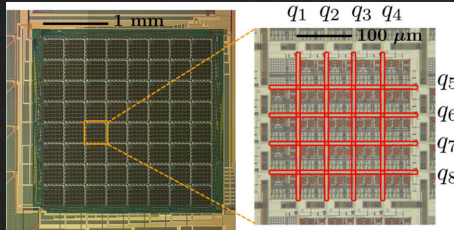
Sistemi D-Wave

Un Sistema D-Wave comprende 3 elementi:

- Un chip QPU (Quantum Processing Unit) che implementa l'algoritmo di Quantum Annealing
- Un computer convenzionale contenente un server di front end detto SAPI (Solver Application Programming Interface)
- Un sistema di Back end che comunica con la QPU

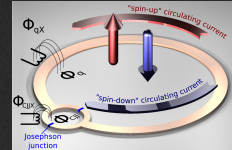
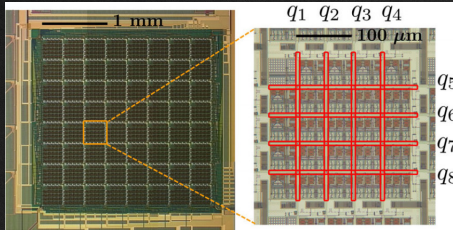


Sistemi D-Wave



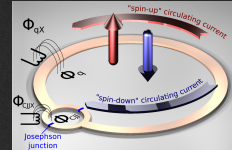
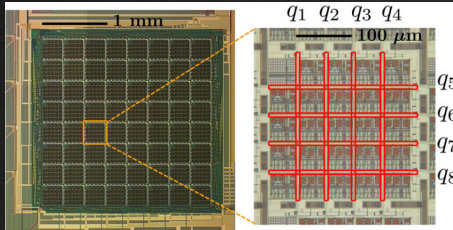
- Fotografia di un chip Chimera 8×8 celle (sx) e di una singola cella contenente 8 qubit (dx)

Sistemi D-Wave



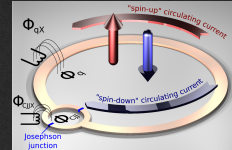
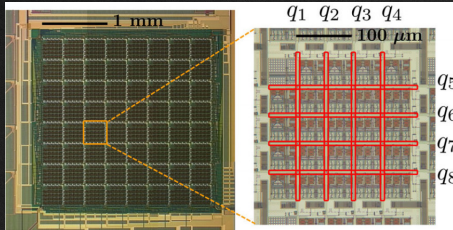
- Fotografia di un chip Chimera 8×8 celle (sx) e di una singola cella contenente 8 qubit (dx)
- Ciascun qubit consiste di un doppio anello di niobio (Nb) interrotto e controllato da due giunzioni Josephson

Sistemi D-Wave



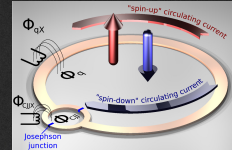
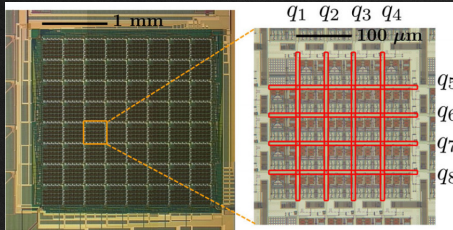
- Fotografia di un chip Chimera 8 × 8 celle (sx) e di una singola cella contenente 8 qubit (dx)
- Ciascun qubit consiste di un doppio anello di niobio (Nb) interrotto e controllato da due giunzioni Josephson
- La corrente può fluire in senso orario o antiorario o in entrambi i sensi

Sistemi D-Wave



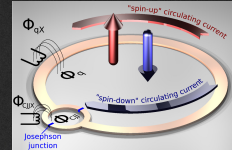
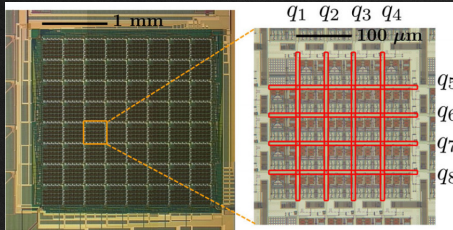
- Fotografia di un chip Chimera 8×8 celle (sx) e di una singola cella contenente 8 qubit (dx)
- Ciascun qubit consiste di un doppio anello di niobio (Nb) interrotto e controllato da due giunzioni Josephson
- La corrente può fluire in senso orario o antiorario o in entrambi i sensi
- Il flusso di corrente crea forze magnetiche rappresentate come spin-up o spin-down o una loro combinazione (sovrapposizione quantistica!)

Sistemi D-Wave



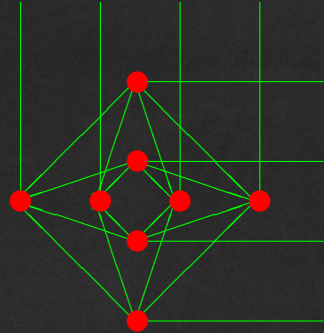
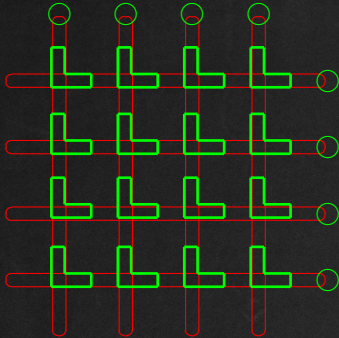
- Fotografia di un chip Chimera 8×8 celle (sx) e di una singola cella contenente 8 qubit (dx)
- Ciascun qubit consiste di un doppio anello di niobio (Nb) interrotto e controllato da due giunzioni Josephson
- La corrente può fluire in senso orario o antiorario o in entrambi i sensi
- Il flusso di corrente crea forze magnetiche rappresentate come spin-up o spin-down o una loro combinazione (sovrapposizione quantistica!)
- Anche i couplers, che connettono coppie di qubit, sono costituiti da anelli di niobio, tuttavia essi hanno una circuiteria più semplice dei qubit

Sistemi D-Wave



- Fotografia di un chip Chimera 8×8 celle (sx) e di una singola cella contenente 8 qubit (dx)
- Ciascun qubit consiste di un doppio anello di niobio (Nb) interrotto e controllato da due giunzioni Josephson
- La corrente può fluire in senso orario o antiorario o in entrambi i sensi
- Il flusso di corrente crea forze magnetiche rappresentate come spin-up o spin-down o una loro combinazione (sovrapposizione quantistica!)
- Anche i couplers, che connettono coppie di qubit, sono costituiti da anelli di niobio, tuttavia essi hanno una circuiteria più semplice dei qubit
- Pros $\neq$ Cons

Topologia Chimera



(sx) Layout fisico di qubit e coupler; (dx) Rappresentazione di qubit (rossi) e coupler (verdi) come nodi ed archi in un grafo Chimera;

Minimum Multicut

Problema Dato un grafo non direzionato $G = (V, E)$ ed un insieme di k coppie di nodi $H = \{(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)\} \subset V \times V$, trovare un sottoinsieme di archi di cardinalità minima $E' \subset E$, la cui rimozione disconnette ciascuna coppia in H

Minimum Multicut

Problema Dato un grafo non direzionato $G = (V, E)$ ed un insieme di k coppie di nodi $H = \{(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)\} \subset V \times V$, trovare un sottoinsieme di archi di cardinalità minima $E' \subset E$, la cui rimozione disconnette ciascuna coppia in H

- Per $k=2$, il problema è P
- Per $k>2$, il problema è $NP-hard$

Minimum Multicut

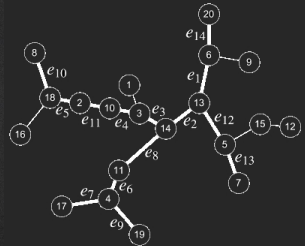
Problema Dato un grafo non direzionato $G = (V, E)$ ed un insieme di k coppie di nodi $H = \{(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)\} \subset V \times V$, trovare un sottoinsieme di archi di cardinalità minima $E' \subset E$, la cui rimozione disconnette ciascuna coppia in H

- Per $k=2$, il problema è P
- Per $k>2$, il problema è $NP-hard$
- Se G è un grafo ad albero, il problema rimane $NP-hard$

Minimum Multicut

Problema Dato un grafo non direzionato $G=(V,E)$ ed un insieme di k coppie di nodi $H=\{(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)\} \subset V \times V$, trovare un sottoinsieme di archi di cardinalità minima $E' \subset E$, la cui rimozione disconnette ciascuna coppia in H

- Per $k=2$, il problema è P
- Per $k>2$, il problema è $NP-hard$
- Se G è un grafo ad albero, il problema rimane $NP-hard$



Minimum Multicut

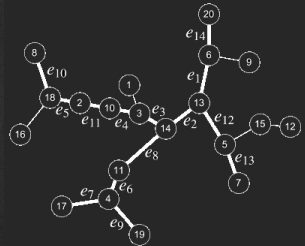
Problema Dato un grafo non direzionato $G=(V,E)$ ed un insieme di k coppie di nodi $H=\{(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)\} \subset V \times V$, trovare un sottoinsieme di archi di cardinalità minima $E' \subset E$, la cui rimozione disconnette ciascuna coppia in H

- Per $k=2$, il problema è P
- Per $k>2$, il problema è $NP-hard$
- Se G è un grafo ad albero, il problema rimane $NP-hard$
- Nell'esempio a dx abbiamo

G grafo ad albero

$|V|=20, k=10$

$H = \{(6,10), (2,18), (11,17), (14,19), (8,13),$
 $(10,11), (3,5), (13,17), (7,14), (6,20)\}$



Applicazioni

- Telecomunicazioni
- Routing
- Programmazione VLSI
- Partizionamento Circuitale

Minimum Multicut

- Consideriamo l'arco e . Definiamo $x_e = \begin{cases} 0 & \text{se } e \text{ è rimosso} \\ 1 & \text{se } e \text{ non è rimosso} \end{cases}$

Minimum Multicut

- Consideriamo l'arco e . Definiamo $x_e = \begin{cases} 0 & \text{se } e \text{ è rimosso} \\ 1 & \text{se } e \text{ non è rimosso} \end{cases}$
- Sia p_i il cammino che connette s_i a t_i (NB.: p_i è unico!)
- Sia $p = \bigcup_{i=1}^k p_i$

Minimum Multicut

- Consideriamo l'arco e . Definiamo $x_e = \begin{cases} 0 & \text{se } e \text{ è rimosso} \\ 1 & \text{se } e \text{ non è rimosso} \end{cases}$
- Sia p_i il cammino che connette s_i a t_i (NB.: p_i è unico!)
- Sia $p = \bigcup_{i=1}^k p_i$
- Introduciamo la funzione da minimizzare

$$h_G = h_{\text{weight}} + h_{\text{penalty}}$$

$$h_{\text{weight}} = \sum_{e \in p} (1 - x_e)$$

$$h_{\text{penalty}} = \lambda_p \sum_{i=1}^k \prod_{e \in p_i} x_e$$

Minimum Multicut

- Consideriamo l'arco e . Definiamo $x_e = \begin{cases} 0 & \text{se } e \text{ è rimosso} \\ 1 & \text{se } e \text{ non è rimosso} \end{cases}$
- Sia p_i il cammino che connette s_i a t_i (NB.: p_i è unico!)
- Sia $p = \bigcup_{i=1}^k p_i$
- Introduciamo la funzione da minimizzare

$$h_G = h_{\text{weight}} + h_{\text{penalty}}$$

$$h_{\text{weight}} = \sum_{e \in p} (1 - x_e)$$

$$h_{\text{penalty}} = \lambda_p \sum_{i=1}^k \prod_{e \in p_i} x_e$$

- h_{weight} è minima se nessun arco è rimosso
- h_{penalty} è massima se nessun arco è rimosso

Minimum Multicut

- Consideriamo l'arco e . Definiamo $x_e = \begin{cases} 0 & \text{se } e \text{ è rimosso} \\ 1 & \text{se } e \text{ non è rimosso} \end{cases}$

- Sia p_i il cammino che connette s_i a t_i (NB.: p_i è unico!)

- Sia $p = \bigcup_{i=1}^k p_i$

- Introduciamo la funzione da minimizzare

$$h_G = h_{\text{weight}} + h_{\text{penalty}}$$

$$h_{\text{weight}} = \sum_{e \in p} (1 - x_e)$$

$$h_{\text{penalty}} = \lambda_p \sum_{i=1}^k \prod_{e \in p_i} x_e$$

- h_{weight} è minima se nessun arco è rimosso
- h_{penalty} è massima se nessun arco è rimosso
- Nell'esempio precedente abbiamo (con $\lambda_p = 10$, numero di coppie di nodi da disconnettere):

$$\begin{aligned} h_G = & 14 - x_1 - x_2 - x_3 - x_4 + 9x_5 - x_6 - x_7 - x_8 - x_9 - x_{10} - x_{11} - x_{12} - x_{13} + 9x_{14} + 10x_1x_2x_3x_4 + 10x_6x_7 + 10x_6x_8x_9 \\ & + 10x_2x_3x_4x_5x_{10}x_{11} + 10x_3x_4x_8 + 10x_2x_3x_{12} + 10x_2x_6x_7x_8 + 10x_2x_{12}x_{13} \end{aligned}$$

3 Problemi da risolvere...

1. h_G contiene accoppiamenti più che quadratici

3 Problemi da risolvere...

1. h_G contiene accoppiamenti più che quadratici

Soluzione: possiamo ridurre h_G ad un problema QUBO

(Quadratic Unconstrained Pseudo-Boolean Optimisation)

$$h_{QUBO} = \sum_k u_k x_k + \sum_{i < k} e_{ik} x_i x_k$$

3 Problemi da risolvere...

1. h_G contiene accoppiamenti più che quadratici

Soluzione: possiamo ridurre h_G ad un problema QUBO

(Quadratic Unconstrained Pseudo-Boolean Optimisation)

$$h_{QUBO} = \sum_k u_k x_k + \sum_{i < k} e_{ik} x_i x_k$$

2. h_G dipende da variabili Booleane, non da variabili di spin

3 Problemi da risolvere...

1. h_G contiene accoppiamenti più che quadratici

Soluzione: possiamo ridurre h_G ad un problema QUBO

(Quadratic Unconstrained Pseudo-Boolean Optimisation)

$$h_{QUBO} = \sum_k u_k x_k + \sum_{i < k} e_{ik} x_i x_k$$

2. h_G dipende da variabili Booleane, non da variabili di spin

Soluzione: poniamo $x_e = \frac{1+\sigma_e}{2}$ [$x_e = 0, 1 \Leftrightarrow \sigma_e = -1, +1$] (spiniificazione)

3 Problemi da risolvere...

1. h_G contiene accoppiamenti più che quadratici

Soluzione: possiamo ridurre h_G ad un problema QUBO

(Quadratic Unconstrained Pseudo-Boolean Optimisation)

$$h_{QUBO} = \sum_k u_k x_k + \sum_{i < k} e_{ik} x_i x_k$$

2. h_G dipende da variabili Booleane, non da variabili di spin

Soluzione: poniamo $x_e = \frac{1+\sigma_e}{2}$ [$x_e = 0, 1 \Leftrightarrow \sigma_e = -1, +1$] (spinficazione)

3. Gli accoppiamenti non rispettano la topologia Chimera

3 Problemi da risolvere...

1. h_G contiene accoppiamenti più che quadratici

Soluzione: possiamo ridurre h_G ad un problema QUBO

(Quadratic Unconstrained Pseudo-Boolean Optimisation)

$$h_{QUBO} = \sum_k u_k x_k + \sum_{i < k} e_{ik} x_i x_k$$

2. h_G dipende da variabili Booleane, non da variabili di spin

Soluzione: poniamo $x_e = \frac{1+\sigma_e}{2}$ [$x_e = 0, 1 \Leftrightarrow \sigma_e = -1, +1$] (spinficazione)

3. Gli accoppiamenti non rispettano la topologia Chimera

Soluzione: il grafo G può essere "immerso" nella topologia Chimera attraverso un processo detto Minor Embedding

Riduzione ad un problema QUBO

- !!! IDEA !!! introduciamo variabili Booleane *aggiuntive* in modo tale che

$$\min_{x \in \{0,1\}^N} h_G(x) = \min_{x \in \{0,1\}^M} h_{QUBO}(x) \quad (M > N)$$

Riduzione ad un problema QUBO

- !!! IDEA !!! introduciamo variabili Booleane **aggiuntive** in modo tale che

$$\min_{x \in \{0,1\}^N} h_G(x) = \min_{x \in \{0,1\}^M} h_{QUBO}(x) \quad (M > N)$$

- **Metodo di Freedman**: si applica a termini **negativi** e richiede **una sola** nuova variabile

$$-\prod_{j=1}^d x_j = \min_{w \in \{0,1\}} w \left[(d-1) - \sum_{j=1}^d x_j \right]$$

Riduzione ad un problema QUBO

- !!! IDEA !!! introduciamo variabili Booleane **aggiuntive** in modo tale che

$$\min_{x \in \{0,1\}^N} h_G(x) = \min_{x \in \{0,1\}^M} h_{QUBO}(x) \quad (M > N)$$

- **Metodo di Freedman**: si applica a termini **negativi** e richiede **una sola** nuova variabile

$$-\prod_{j=1}^d x_j = \min_{w \in \{0,1\}} w \left[(d-1) - \sum_{j=1}^d x_j \right]$$

- **Metodo di Ishikawa**: si applica a termini **positivi** e richiede **un numero** $\left\lceil \frac{d-1}{2} \right\rceil$ di nuove variabili

$$t(x) = \prod_{k=j}^d x_j \quad \leftarrow \quad \text{espressione da ridurre}$$

Riduzione ad un problema QUBO

- !!! IDEA !!! introduciamo variabili Booleane **aggiuntive** in modo tale che

$$\min_{x \in \{0,1\}^N} h_G(x) = \min_{x \in \{0,1\}^M} h_{QUBO}(x) \quad (M > N)$$

- **Metodo di Freedman**: si applica a termini **negativi** e richiede **una sola** nuova variabile

$$-\prod_{j=1}^d x_j = \min_{w \in \{0,1\}} w \left[(d-1) - \sum_{j=1}^d x_j \right]$$

- **Metodo di Ishikawa**: si applica a termini **positivi** e richiede **un numero** $\left\lfloor \frac{d-1}{2} \right\rfloor$ di nuove variabili

$$t(x) = \prod_{k=1}^d x_k \quad \leftarrow \quad \text{espressione da ridurre}$$

$$S_1 = \sum_{j=1}^d x_j, \quad S_2 = \sum_{1 \leq i < j \leq d} x_i x_j,$$

$$W_1 = \sum_{j=1}^k w_j, \quad W_2 = \sum_{j=1}^k (4j-1) w_j$$

Riduzione ad un problema QUBO

- !!! **IDEA** !!! introduciamo variabili Booleane **aggiuntive** in modo tale che

$$\min_{x \in \{0,1\}^N} h_G(x) = \min_{x \in \{0,1\}^M} h_{QUBO}(x) \quad (M > N)$$

- Metodo di Freedman**: si applica a termini **negativi** e richiede **una sola** nuova variabile

$$-\prod_{j=1}^d x_j = \min_{w \in \{0,1\}} w \left[(d-1) - \sum_{j=1}^d x_j \right]$$

- Metodo di Ishikawa**: si applica a termini **positivi** e richiede **un numero** $\left\lfloor \frac{d-1}{2} \right\rfloor$ di nuove variabili

$$t(x) = \prod_{k=1}^d x_k \quad \leftarrow \quad \text{espressione da ridurre}$$

$$S_1 = \sum_{j=1}^d x_j, \quad S_2 = \sum_{1 \leq i < j \leq d} x_i x_j,$$

$$W_1 = \sum_{j=1}^k w_j, \quad W_2 = \sum_{j=1}^k (4j-1) w_j$$

$$t(x) = \begin{cases} S_2 + \min_{w \in \{0,1\}^k} \{W_2 - 2W_1 S_1\} & \text{se } d \text{ è pari,} \\ S_2 + \min_{w \in \{0,1\}^k} \{W_2 - 2W_1 S_1 + w_k (S_1 - d + 1)\} & \text{se } d \text{ è dispari.} \end{cases}$$

Esempio

$$\begin{aligned}h_{QUBO} = & 14 - x_1 - x_2 - x_3 - x_4 + 9x_5 - x_6 - x_7 - x_8 - x_9 - x_{10} - x_{11} - x_{12} - \\& x_{13} + 9x_{14} + 20x_6x_7 + 10x_1x_2 + 10x_1x_3 + 10x_1x_4 + 30x_2x_3 + 20x_2x_4 + \\& 30x_3x_4 + 30x_{15} - 20x_1x_{15} - 20x_2x_{15} - 20x_3x_{15} - 20x_4x_{15} + 20x_6x_8 + \\& 10x_6x_9 + 10x_8x_9 + 10x_{16} - 10x_6x_{16} - 10x_8x_{16} - 10x_9x_{16} + 10x_2x_5 + \\& 10x_2x_{10} + 10x_2x_{11} + 10x_3x_5 + 10x_3x_{10} + 10x_3x_{11} + 10x_4x_5 + 10x_4x_{10} + \\& 10x_4x_{11} + 10x_5x_{10} + 10x_5x_{11} + 10x_{10}x_{11} + 30x_{17} + 70x_{18} - 20x_2x_{17} - \\& 20x_3x_{17} - 20x_4x_{17} - 20x_5x_{17} - 20x_{10}x_{17} - 20x_{11}x_{17} - 20x_2x_{18} - \\& 20x_3x_{18} - 20x_4x_{18} - 20x_5x_{18} - 20x_{10}x_{18} - 20x_{11}x_{18} + 10x_3x_8 + 10x_4x_8 + \\& 10x_{19} - 10x_3x_{19} - 10x_4x_{19} - 10x_8x_{19} + 20x_2x_{12} + 10x_3x_{12} + 10x_{20} - \\& 10x_2x_{20} - 10x_3x_{20} - 10x_{12}x_{20} + 10x_2x_6 + 10x_2x_7 + 10x_2x_8 + 10x_7x_8 + \\& 30x_{21} - 20x_2x_{21} - 20x_6x_{21} - 20x_7x_{21} - 20x_8x_{21} + 10x_2x_{13} + 10x_{12}x_{13} + \\& 10x_{22} - 10x_2x_{22} - 10x_{12}x_{22} - 10x_{13}x_{22}\end{aligned}$$

9 nuove variabili!

Minor Embedding

- La QPU di un Sistema D-Wave definisce un grafo $\mathcal{G}_{M,N,L}$ con $M \times N$ celle, ciascuna contenente $2L$ qubit

Minor Embedding

- La QPU di un Sistema D-Wave definisce un grafo $\mathcal{G}_{M,N,L}$ con $M \times N$ celle, ciascuna contenente $2L$ qubit
- Gli accoppiamenti dell'Hamiltoniana di Ising ottenuta dalla riduzione QUBO + spinnificazione definiscono un grafo $\mathcal{G}_{Ising}$

Minor Embedding

- La QPU di un Sistema D-Wave definisce un grafo $\mathcal{G}_{M,N,L}$ con $M \times N$ celle, ciascuna contenente $2L$ qubit
- Gli accoppiamenti dell'Hamiltoniana di Ising ottenuta dalla riduzione QUBO + spinnificazione definiscono un grafo $\mathcal{G}_{Ising}$
- In generale non è possibile trovare un sottografo $\tilde{\mathcal{G}} \subset \mathcal{G}_{M,N,L}$ che sia isomorfo a $\mathcal{G}_{Ising}$

Minor Embedding

- La QPU di un Sistema D-Wave definisce un grafo $\mathcal{G}_{M,N,L}$ con $M \times N$ celle, ciascuna contenente $2L$ qubit
- Gli accoppiamenti dell'Hamiltoniana di Ising ottenuta dalla riduzione QUBO + spinnificazione definiscono un grafo $\mathcal{G}_{Ising}$
- In generale non è possibile trovare un sottografo $\tilde{\mathcal{G}} \subset \mathcal{G}_{M,N,L}$ che sia isomorfo a $\mathcal{G}_{Ising}$

!!! IDEA !!! Ciascun vertice in $\mathcal{G}_{Ising}$ viene mappato ad un sottoalbero connesso di qubit che rappresenta una variabile logica

Minor Embedding

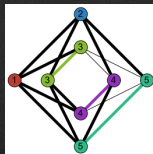
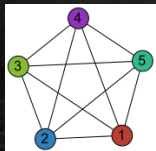
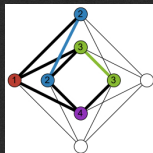
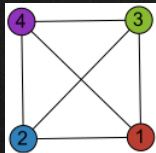
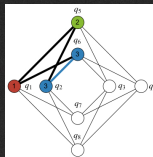
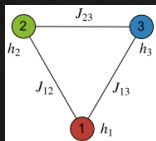
- La QPU di un Sistema D-Wave definisce un grafo $\mathcal{G}_{M,N,L}$ con $M \times N$ celle, ciascuna contenente $2L$ qubit
- Gli accoppiamenti dell'Hamiltoniana di Ising ottenuta dalla riduzione QUBO + spinnificazione definiscono un grafo $\mathcal{G}_{Ising}$
- In generale non è possibile trovare un sottografo $\tilde{\mathcal{G}} \subset \mathcal{G}_{M,N,L}$ che sia isomorfo a $\mathcal{G}_{Ising}$

!!! IDEA !!! Ciascun vertice in $\mathcal{G}_{Ising}$ viene mappato ad un sottoalbero connesso di qubit che rappresenta una variabile logica

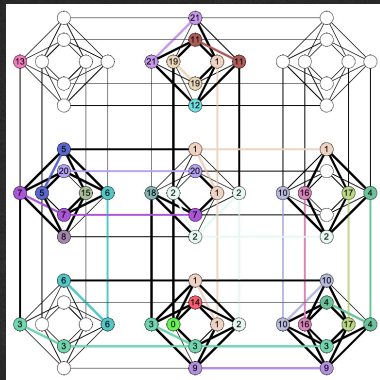
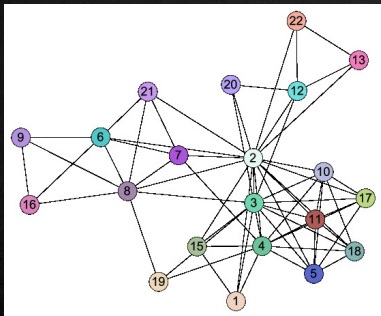
Problema Dato un grafo Chimera $\mathcal{G}_{M,N,L}$ di size (M,N,L) e un grafo logico $\mathcal{G}_{Ising} = (V_{Ising}, E_{Ising})$, trovare un sottografo $\tilde{\mathcal{G}} \subset \mathcal{G}_{M,N,L}$ tale che

- 1) ciascun vertice $j \in V_{Ising}$ venga mappato ad un sottoalbero connesso $T_j \subset \mathcal{G}_{M,N,L}$
 - 2) ciascun arco $(i,j) \in E_{Ising}$ venga mappato ad almeno un coupler in $\mathcal{G}_{M,N,L}$
-

Esempi grafi completi

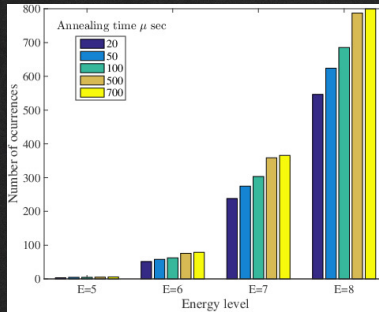


Esempio Minimum Multi-Cut



- 23 nodi (14 iniziali + 9 aggiunti)
- 51 qubit utilizzati!

Risultati



- Quando un problema di Ising viene sottoposto con un certo numero di **readout**, esso ritorna una lista di configurazioni di spin con relativa energia, ordinate da energia minima ad energia massima
- Il simulatore può anche ritornare la frequenza di ciascun livello energetico osservato

Conclusioni

- Un nuovo paradigma computazionale

Conclusioni

- Un nuovo paradigma computazionale
- Non tutti i problemi possono essere trattati

Conclusioni

- Un nuovo paradigma computazionale
- Non tutti i problemi possono essere trattati
- Non è chiaro in quale misura quantum annealing su quantum computer performi meglio rispetto ai migliori algoritmi su supercomputer classici

Conclusioni

- Un nuovo paradigma computazionale
- Non tutti i problemi possono essere trattati
- Non è chiaro in quale misura quantum annealing su quantum computer performi meglio rispetto ai migliori algoritmi su supercomputer classici
- Attualmente solo problemi di size piccola

Conclusioni

- Un nuovo paradigma computazionale
- Non tutti i problemi possono essere trattati
- Non è chiaro in quale misura quantum annealing su quantum computer performi meglio rispetto ai migliori algoritmi su supercomputer classici
- Attualmente solo problemi di size piccola
- Mapping su topologia Chimera quasi mai one-to-one