

Progetto OC - MaxFlow QUBO

Nicola Barbaro (1070668)

Mario Bifulco (881727)

A.A. 2022/2023

Indice

1	Problema del flusso massimo	2
1.1	Descrizione generale del problema	2
1.2	Minimum Cut, Teorema max-flow min-cut	3
1.3	Riformulazione su programmazione lineare primale-duale	4
1.4	Significato dei vincoli	4
1.5	Riformulazione tramite uguaglianze	4
2	QAC e formulazione QUBO	4
2.1	Quantum Adiabatic Computing	4
2.2	Quantum Unconstraint Binary Optimization	6
2.3	Simulated Annealing	6
3	Min Cut come problema QUBO	8
3.1	Conversione delle variabili di surplus	8
3.2	Rilassamento Lagrangiano	8
4	Implementazione	8
4.1	Test	8

1 Problema del flusso massimo

1.1 Descrizione generale del problema

Introdotta da Harris e Ross nel 1954, il problema del Flusso Massimo (Maximum Flow Problem) richiede di trovare, sulla struttura dati chiamata Rete di Flusso, un flusso plausibile con il rapporto più alto possibile (chiamato, appunto *Flusso Massimo*). Il MF è spesso usato per risolvere problemi legati al trasporto di beni o di instradamento di informazioni su una rete.

Si vuole definire innanzitutto la connotazione della Rete di Flusso. Sia:

- $G = (V, E)$ una rete (rappresentata come un grafo) con $s, t \in V$ rispettivamente la *fonte* e la *foce* di G .
- Dato $(i, j) \in E$ uno qualsiasi spigolo della rete, la sua **capacità** è definita come la massima quantità di flusso che può passare per esso. Il suo valore è definito dalla funzione $c : E \rightarrow \mathbb{R}^+$.

Definizione. Un generico flusso su una rete $G = (V, E)$ è una funzione $f : E \rightarrow \mathbb{R}$ se sono soddisfatti i seguenti:

- *Vincoli di capacità.* Per ogni $(i, j) \in E$, il flusso associato ad esso non supererà mai la sua capacità, ossia $x_{ij} \leq u_{ij}$.
- *Conservazione dei flussi.* Con sola esclusione del nodo fonte s e foce t , la somma dei flussi in entrata in un nodo deve uguagliare la somma dei flussi in uscita dallo stesso nodo. In formula:

$$\forall v \in V \setminus \{s, t\} : \sum_{i:(i,j) \in E} x_{ij} = \sum_{i:(j,i) \in E} x_{ji}$$

Si può osservare come la matrice di adiacenza del flusso G è antisimmetrica: infatti $x_{ij} = -x_{ji}$, $\forall (i, j) \in E$.

Definizione. Il **valore** del flusso è la quantità di flusso passante sulla rete a partire dalla fonte s e terminante nella foce t . Rappresentato in termini formali, un flusso $f : E \rightarrow \mathbb{R}^+$ è una funzione il cui dominio è dato da:

$$|x| = \sum_{j:(s,j) \in E} x_{sj} - \sum_{i:(i,s) \in E} x_{is}$$

A questo punto è possibile fornire una introduzione formale del problema attraverso la seguente:

Definizione.. Il **Problema del Flusso Massimo** è il problema di instradare quanto più flusso possibile dalla fonte alla foce, ossia di trovare il flusso

con massimo valore x_{max} instradabile sulla rete.

Il problema del flusso massimo ammette unica o infinite soluzioni, dato che ci sono infinite combinazioni lineari possibili del valore x_{max} di partenza.

1.2 Minimum Cut, Teorema max-flow min-cut

Nella teoria dei grafi, un **taglio di costo minimo** (o **minimum cut**) per un grafo è una partizione dei vertici del grafo in due insiemi ad disgiunti minimale rispetto ad una qualunque metrica definita dal problema. Nella specifica del problema di Flusso a Costo Massimo, su rete G , è interessante porre l'attenzione sull'insieme di tagli chiamati **tagli s-t** $C = (S, T)$: questo insieme è una suddivisione dei vertici V di G tale che il nodo fonte $s \in S$ e il nodo foce $t \in T$, ossia una ripartizione dei vertici della rete in due parti, con la fonte in una parte e la foce in un'altra. Si definisce il **cut-set** X_C di un taglio C l'insieme degli spigoli che connettono i vertici $i \in S$ ai vertici $j \in T$:

$$X_C := \{(i, j) \in E : i \in S, j \in T\} = (S \times T) \cap E$$

È immediato notare che togliere gli spigoli di X_C da E renderebbe impossibile la computazione di qualsiasi flusso a valore positivo, perché non ci sarebbero più cammini dalla fonte alla foce percorribili.

La **capacità** di un taglio $s - t$ è dato dalla somma delle capacità degli spigoli in X_C :

$$c(S, T) = \sum_{(i, j) \in X_C} u_{ij} = \sum_{(i, j) \in E, i \in S, j \in T} u_{ij}$$

Il problema del **taglio s-t a costo minimo** prevede di minimizzare $u(S, T)$, ossia di determinare S, T tali che la capacità del taglio $s - t$ sia minimale.

Teorema Max-Flow Min-Cut. Il massimo valore di un flusso $s - t$ è uguale al taglio $s - t$ di costo minimo tra tutti i possibili tagli.

1.3 Riformulazione su programmazione lineare primale-duale

È possibile rappresentare il problema del Flusso a Costo massimo come un programma lineare primale il cui duale è il programma lineare associato al problema di Taglio a Costo Minimo.

Formulazione Primale:

$$\text{massimizza} \quad \sum_{(s,i) \in FS_i} x_{si} = x_{ts} \quad (1)$$

$$\text{soggetto a} \quad \sum_{(h,i) \in BS_i} x_{hi} - \sum_{(i,h) \in FS_i} x_{ij} = 0 \quad \forall i \in V \setminus \{s, t\} \quad (2)$$

$$\left(\sum_{(i,t) \in BS_t} x_{it} \right) - x_{ts} = 0 \quad (3)$$

$$x_{ts} - \left(\sum_{(s,i) \in FS_s} x_{si} \right) = 0 \quad (4)$$

$$0 \leq x_{ij} \leq u_{ij} \quad \forall (i, j) \in E \quad (5)$$

1.4 Significato dei vincoli

1.5 Riformulazione tramite uguaglianze

2 QAC e formulazione QUBO

La seguente sezione fa riferimento alle informazioni contenute in [2] e [1].

2.1 Quantum Adiabatic Computing

La risoluzione di problemi intrattabili in modo efficiente utilizzando il calcolo classico richiede la dimostrazione dell'uguaglianza $P = NP$. Tuttavia, al momento attuale, questa dimostrazione rimane un problema aperto.

Un'alternativa per affrontare tali problemi è sfruttare modelli di calcolo non convenzionali che utilizzano principi di calcolo differenti. Tra questi modelli, si trovano le architetture basate sulla meccanica quantistica.

Attualmente, i due principali approcci per la computazione quantistica sono la programmazione tramite gate, concettualmente simile alla programmazione tramite porte logiche, e la programmazione adiabatica quantistica (QAC). Quest'ultimo approccio si basa sulla ricerca di stati a minore energia per trovare la soluzione desiderata.

Nel modello di calcolo descritto da QAC, si utilizza una matrice Hermitiana con elementi complessi (\mathbb{C}). Gli Hamiltoniani, utilizzati per eseguire la computazione, rappresentano il livello di energia istantaneo di un sistema fisico e descrivono l'evoluzione del suo stato.

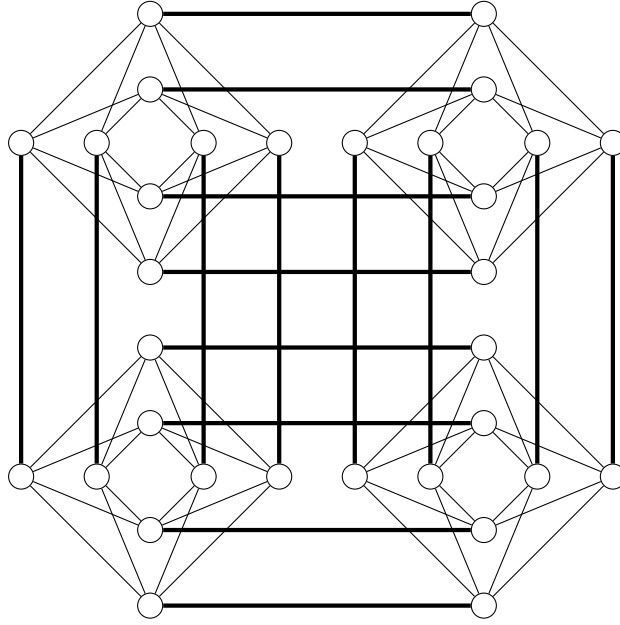


Figura 1: Esempio di QCPU a quattro qubit

L'idea fondamentale è sfruttare le proprietà fisiche per cercare stati a minore energia e quindi più stabili.

La matrice Hamiltoniana è un operatore in uno spazio vettoriale in cui le configurazioni delle variabili sono rappresentate come vettori, che rappresentano gli autovettori della matrice, e i valori assunti dalla funzione da minimizzare corrispondono agli autovalori di \mathcal{H}_P .

Per risolvere problemi espressi tramite la programmazione lineare utilizzando un Quantum CPU, è necessario convertirli in una formulazione QUBO (Quantum Unconstrained Binary Optimization), che presenta una naturale rappresentazione grafica in forma di grafo, successivamente mappato nel grafo dei qubit.

La mappatura viene effettuata mediante l'utilizzo di un algoritmo di *minor-embedding*, che preserva le relazioni aggiungendo eventualmente nuovi nodi se necessario.

Tuttavia, l'algoritmo di *minor-embedding* è NP-completo. Fortunatamente, esistono algoritmi che funzionano bene sperimentalmente, quindi, nonostante la sua complessità teorica, rappresenta comunque una procedura utilizzabile nella pratica.

Ci sono alcuni punti critici da considerare in questo approccio computazionale:

- È difficile fornire un limite inferiore al tempo di annealing (il tempo

minimo necessario per raggiungere lo stato finale).

- In assenza di tale limite inferiore, la formulazione QUBO può essere considerata come un'euristica e non è possibile valutare in generale l'efficienza e la correttezza dell'algoritmo.
- Questo approccio non è valido in modo generale, poiché ci sono casi in cui gli algoritmi classici mostrano un comportamento lineare, mentre l'implementazione quantistica può degradare all'aumentare delle dimensioni del problema.

2.2 Quantum Unconstrained Binary Optimization

Il problema QUBO, Quadratic Unconstrained Binary Optimization, consiste nell'individuare il vettore $x^* = (x_1^*, \dots, x_n^*)$ che minimizza l'equazione $\bar{x}^t Q \bar{x}$, dove \bar{x} è un vettore colonna di variabili binarie e Q è una matrice triangolare superiore che rappresenta i coefficienti della funzione da minimizzare.

In particolare, l'obiettivo del problema QUBO è determinare il valore delle variabili binarie x_i che minimizza l'espressione $\sum_{i=1}^n \sum_{j=1}^n Q_{ij} x_i x_j$, dove Q_{ij} sono gli elementi della matrice Q .

$$\text{minimizzare } \underbrace{\begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}}_{\bar{x}^T} \underbrace{\begin{bmatrix} a_1 & \cdots & a_3 \\ \vdots & \ddots & b_3 \\ 0 & \cdots & c_3 \end{bmatrix}}_Q \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}}_{\bar{x}}$$

È importante notare che nella funzione obiettivo espansa, è possibile riscrivere i termini quadratici, cioè le variabili al quadrato, utilizzando il loro equivalente lineare. Questa riscrittura è possibile grazie al dominio delle variabili x_i , che assumono solo i valori 0 o 1. Pertanto, il termine x_i^2 può essere sostituito con x_i nella formulazione del problema QUBO.

Il formalismo dei modelli QUBO offre un approccio flessibile per la risoluzione di problemi di ottimizzazione combinatoria mediante l'utilizzo di variabili binarie e l'espressione quadratica della funzione obiettivo. Questo formalismo è di particolare importanza nella progettazione e nell'implementazione di algoritmi e processori quantistici adiabatici per la risoluzione efficiente di problemi complessi.

2.3 Simulated Annealing

L'algoritmo del Simulated Annealing fornisce una prospettiva intuitiva sul funzionamento di un processore quantistico adiabatico, QCPU.

L'obiettivo di questo algoritmo è raggiungere lo stato finale attraverso un cammino casuale, Random Walk, attraverso tutti i possibili stati del sistema. La selezione dei passi da compiere è influenzata dalla probabilità che varia nel

tempo e determina se accettare o meno le variazioni energetiche del sistema fisico su cui si sta operando.

Nell'implementazione di questo algoritmo, vengono utilizzati due attributi fondamentali: l'insieme degli stati adiacenti a un dato nodo e l'energia associata a tale nodo, questi parametri influenzano la sequenza di passi che porta al cammino e dunque allo stato finale.

Durante l'esecuzione dell'algoritmo, un parametro cruciale è rappresentato dalla temperatura, che può decrescere secondo diverse funzioni, più o meno rapidamente, con la diminuzione della temperatura si otterrà una conseguente stabilizzazione su uno degli stati, che quindi rappresenterà la risposta.

Formalizzazione dell'algoritmo del Simulated Annealing

Sia $G = (V, E)$ un grafo rappresentante il sistema fisico di interesse, dove V rappresenta l'insieme dei nodi e E l'insieme degli archi. Ogni nodo $v \in V$ ha associata un'energia $E(v)$, mentre l'insieme degli adiacenti di un nodo v è indicato con $N(v)$.

L'algoritmo del Simulated Annealing può essere descritto attraverso i seguenti passaggi:

1. Inizializzazione: si assegna a ogni nodo del grafo una configurazione casuale iniziale e si fissa una temperatura iniziale T .
2. Ciclo principale:
 - (a) Scelta di un nodo v in modo casuale dal grafo.
 - (b) Selezione di un nodo u in modo casuale dall'insieme $N(v)$.
 - (c) Calcolo della variazione di energia $\Delta E = E(u) - E(v)$.
 - (d) Se $\Delta E \leq 0$, si accetta la mossa e si aggiorna lo stato del sistema considerando u come nuovo stato corrente.
 - (e) Se $\Delta E > 0$, si accetta la mossa con una probabilità $p = \exp(-\frac{\Delta E}{T})$. In caso di accettazione, si aggiorna lo stato del sistema come nel passo precedente.
 - (f) Ripeti i passi da (a) a (e) fino a soddisfare una condizione di terminazione.

Durante l'esecuzione dell'algoritmo, la temperatura T viene ridotta nel tempo seguendo una specifica funzione, che dipende dal problema e dall'obiettivo dell'ottimizzazione.

L'algoritmo del Simulated Annealing rappresenta un importante strumento per l'esplorazione degli spazi di ricerca complessi e ha trovato applicazioni significative nella risoluzione di problemi di ottimizzazione e nell'ambito della computazione quantistica adiabatica.

È importante notare che l'algoritmo del Simulated Annealing può essere implementato anche su processori quantistici adiabatici, sfruttando le peculiarità della computazione quantistica per ottenere potenziali vantaggi prestazionali.

3 Min Cut come problema QUBO

3.1 Conversione delle variabili di surplus

3.2 Rilassamento Lagrangiano

4 Implementazione

4.1 Test

Riferimenti bibliografici

- [1] Fred W. Glover and Gary A. Kochenberger. A tutorial on formulating QUBO models. *CoRR*, abs/1811.11538, 2018.
- [2] Catherine C. McGeoch. Theory versus practice in annealing-based quantum computing. *Theoretical Computer Science*, 816:169–183, 2020.