

# **Modellazione del problema MAX3-SAT in forma QUBO e sua risoluzione tramite Simulated Annealing**

Tesi di Laurea in  
Ingegneria Informatica

**Candidato**

Francesco Vesigna

**Relatori**

Prof. Marco Cococcioni



UNIVERSITÀ DI PISA

# Introduzione e Problema

- Il problema SAT riguarda la soddisfacibilità di una formula logica ed è centrale nella teoria della complessità.
- Il 3-SAT è la versione del SAT con formule in CNF dove ogni clausola ha esattamente 3 letterali (variabile booleana negata o non). Il MAX-3-SAT è la versione di ottimizzazione: massimizzare il numero di clausole 3-CNF soddisfatte.

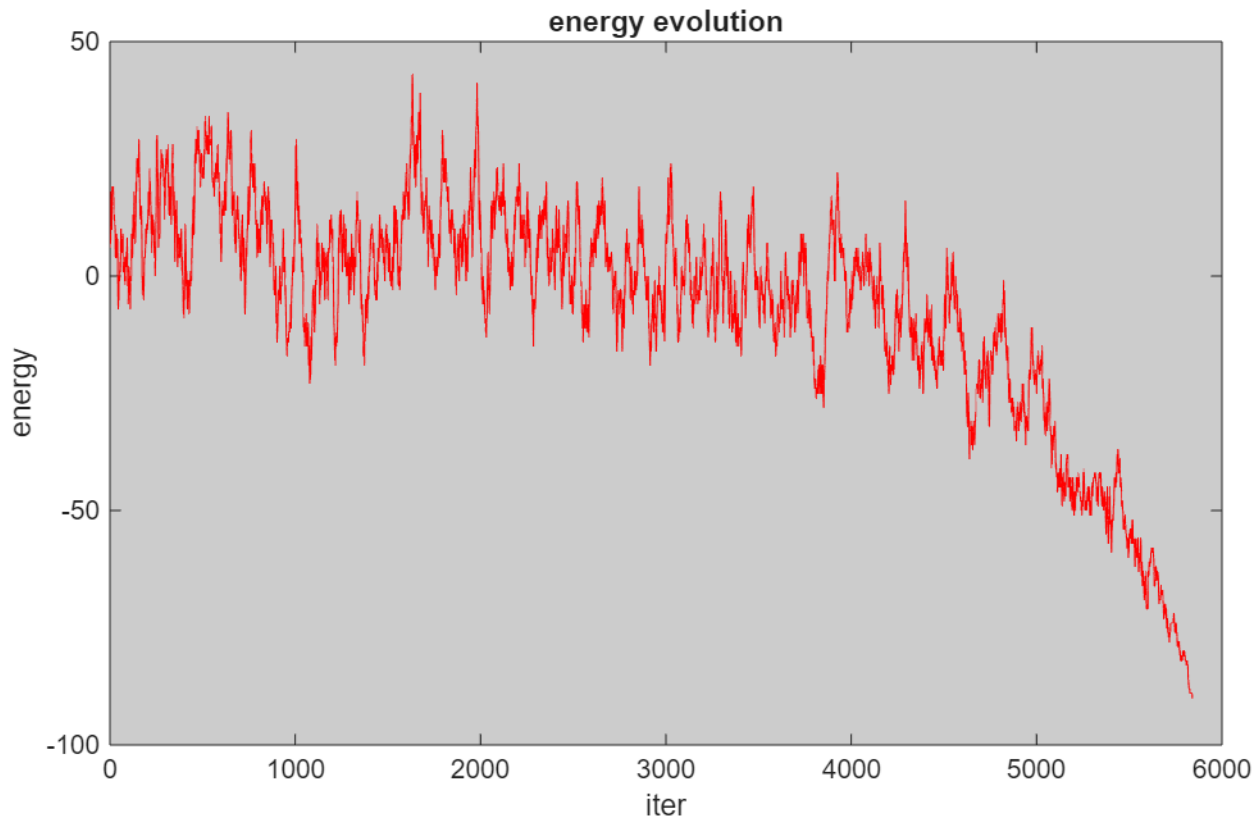
$$\bigwedge_i \left( \bigvee_j^N y_{ij} \right) \text{ con } N = 3$$

- MAX3-SAT è NP-completo, perciò sfruttiamo l'hardware quantistico che opera su modelli QUBO per ridurre il tempo di calcolo.
- QUBO codifica la funzione obiettivo di un problema avente al massimo termini quadratici tramite una matrice  $Q$ .
- Come trasformiamo un'istanza MAX3-SAT in una istanza QUBO?

$$f(x_1, \dots, x_n) \rightarrow Q$$

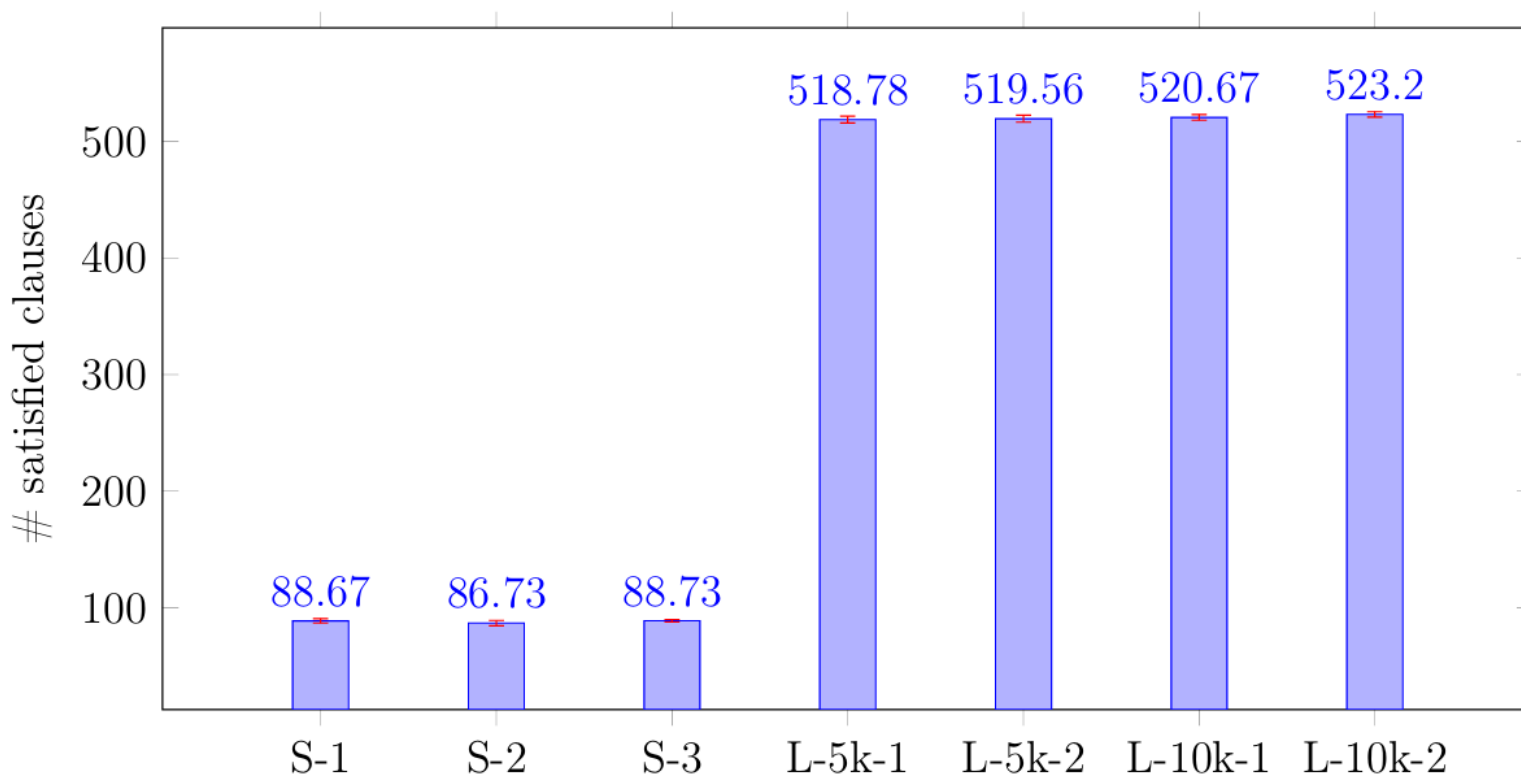
- Per valutare un assegnamento  $x$  si calcola l'energia di  $x$  come  $H(x) = x^T Q x$ . Nel problema QUBO si vuole minimizzare l'hamiltoniana  $H(x)$ .
- Approcci per trovare  $Q$ :
  - Trasformazioni Esatte:  $Q$  codifica perfettamente il problema originale
  - Trasformazioni Approssimate: Esistono assegnamenti ottimi che non hanno energia minima in  $Q$ .
- Nel codice MATLAB si è deciso di utilizzare una trasformazione approssimata per via di matrici  $Q$  più piccole.
- Ogni clausola è associabile a quattro tipi diversi di  $Q$ , a seconda del numero di negazioni presenti.
- Per ogni clausola:
  - si calcola la matrice  $Q$  approssimata corrispondente, se ne mappa il contributo nella matrice  $Q_f$ . Usare una  $Q$  approssimata garantisce una  $Q_f$  più piccola.

- Una volta determinata  $Q_f$ , dobbiamo minimizzare  $H(x)$ .
- Nel Codice MATLAB si è implementato il simulated annealing per via della semplicità e riproducibilità in ambiente classico.
  - Si associa al procedimento una temperatura  $T$  che decresce nel tempo.
  - A ogni passo l'algoritmo genera un nuovo assegnamento, modificando l'assegnamento precedente. Se l'energia del nuovo assegnamento è minore del precedente, viene accettato. Se l'energia è maggiore, l'accettazione è legata a una certa probabilità che dipende da  $\Delta E$  e da  $T$ .
  - Per temperature alte l'algoritmo ha una probabilità alta di accettare assegnamenti con energia maggiore (meno ottimali). Per temperature basse l'algoritmo tende a ignorare assegnamenti meno ottimali.
- L'algoritmo, teoricamente, converge a un ottimo globale per  $t \rightarrow \infty$  e usando una  $T$  che decresce molto lentamente.
  - In pratica, si usano delle temperature che decrescono più velocemente, in quanto il risultato teorico non è utilizzabile. Nel codice si usa una progressione geometrica decrescente.



(max-iter = 10000, ini-temp = 100, fin-temp = 0.1, vars = 125 clauses = 538)

- L'algoritmo si comporta come descritto nella slide precedente.
- Si è deciso di implementare la memoization dei candidati accettati, che funziona bene per istanze QUBO di medie dimensioni.



- Le istanze S sono caratterizzate da 20 vars, 91 clauses, analizzate con 5k iterazioni.
- Le istanze L sono caratterizzate da 125 vars, 538 clauses, analizzate con 5k e 10k iterazioni.