

Al momento il mio piano è quello di trovare un algoritmo basato sul individuare un augmenting path. Vorrei trovare un percorso dalla sorgente al pozzo del grafo (sink). Parto dal considerare grafi aciclici, per poi eventualmente trovare il modo di escludere i cicli dai grafi ciclici.

Quello che sono riuscito a fare finora è:

-Dato un insieme di archi, selezionarne uno e uno solo casualmente tra questi

Mi serve per poter definire l'arco iniziale da scegliere. Se riuscirò a manipolare le probabilità riducendo la probabilità che venga scelto un arco che porta a un percorso morto, ho risolto il mio problema

-Dato un arco, se e solo se questo è attivo, attiva anche il successivo se unico (per poter definire il concetto di flusso). Al momento riesco nell'obiettivo applicando un semplice CNOT. Devo controllare eventuali effetti collaterali a riguardo.

I miei attuali obiettivi sono:

-Provare a riapplicare il concetto "Dato un arco, se e solo se questo è attivo, attiva anche il successivo" se l'arco successore non fosse unico

(ho alcune idee, riuserei il primo punto "selezionarne uno e uno solo casualmente tra questi", ma devo assicurarmi di formularlo in modo tale da annullare la scelta (ottenere come output tutti 0 al fine di

-Trovare un modo per cui un percorso venga ridotto di probabilità se dovesse essere un percorso morto (per ora non ho idee sul come farlo, sto studiando Grover per capire come potrei manipolare le probabilità/pesi tra i vari percorsi)

-Simulare lo stato teorico della funzione d'onda su  $Q^{\#}$  (non ho ancora cercato come si possa fare)

Conclusioni di questa settimana:

- Quando simulati a mano, due qbit che non si toccano in realtà potrebbero avere un comportamento anomalo in cui si influenzano a vicenda  
vedi il circuito, la funzione d'onda prima della misurazione

$Q_0$  H H Measurement

$Q_1$  H Measurement

-Per quanto riguarda la rappresentazione del problema, è necessario utilizzare ALMENO un qbit per arco

La ragione per questa affermazione sta nel fatto che la rappresentazione come prodotto tensoriale (di Kronecker per essere precisi) una volta misurata collassa praticamente in un array di bits. Per poter rappresentare senza ambiguità tutte le possibili combinazioni dell'esito dell'algoritmo (arco 1 preso/non preso. arco 2 preso/non preso....) servono almeno  $\#(\text{archi})$