**MC MIN-CUT**

Vella Mirco - S4803218

Min-Cut è un algoritmo che mira a trovare il minimo numero di archi da tagliare per poter ottenere due grafi distinti. Ovviamente di questo algoritmo esistono due versioni distinte, la versione “deterministica” e la versione “randomizzata”. Noi abbiamo trattato in particolare la versione randomizzata di tipo Monte Carlo.

Questa versione prevede la scelta randomica dell’arco da tagliare ad ogni turno. Il taglio dell’arco è seguito dalla “fusione” dei due nodi dell’arco scelto, con la rimozione successiva degli eventuali self-loops, ovvero gli archi che vanno da un nodo a sé stesso. Il risultato finale sarà il Min-Cut candidato, ma essendo un algoritmo randomizzato la risposta finale non è sempre uguale e soprattutto non è sempre giusta. Questo vuol dire che per ottenere una risposta affidabile dovrò eseguire l’algoritmo un congruo numero di volte.

Il mio codice implementa il Min-Cut organizzando vertici e archi in due liste distinte. Fondamentalmente ogni lista funziona da “recipiente”, dentro il quale ho gli archi considerabili e i vertici considerabili ad ogni iterazione. La funzione MCMinCut effettua le operazioni necessarie, ovvero sceglie due vertici casuali, elimina gli archi tra i due nodi e li unisce in un nuovo nodo, appoggiandosi a funzioni ausiliarie di gestione delle liste.

Ho scelto di non creare però un nuovo nodo ma di far “sopravvivere” uno dei due nodi scelti casualmente, al quale vado ad attaccare gli archi del nodo eliminato. Alla fine, MCMinCut restituisce il numero di archi sopravvissuti, e contando quante volte esce 4 io so quante volte esce il Min-Cut effettivo, perché sto analizzando un grafo di Fritsch.

Nel mio caso con 10^5 run dell’algoritmo ho ottenuto il Min-Cut circa 15700 volte (15914 l’ultima volta), quindi ho una frequenza empirica di circa 0.157, quindi il 15.7%, molto più alta della p indicata nel testo, ovvero (2/n^2), che avendo n = 9 mi da circa 0.024, quindi in percentuale il 2.4%. La probabilità con un algoritmo randomizzato come questo di tipo Monte Carlo cresce al crescere degli esperimenti.

**Immagine che contiene rosso, luce, nero, scuro

Descrizione generata automaticamenteFRITSCH GRAPH:**