**CLOSENESS CENTRALITY**

L’algoritmo per il calcolo della closeness centrality prevede l’esecuzione della BFS (essendo il grafo non pesato e non diretto) a partire da ogni nodo. Il calcolo di tale misura di centralità risulta quindi molto oneroso, da come si è potuto notare dai primi esperimenti.  
La versione Naive di tale algoritmo ha richiesto infatti poco più di un’ora per essere eseguita.  
Di conseguenza è stata implementata una versione parallela attraverso la suddivisione dei nodi in j parti al fine di ottenere tempi di esecuzione migliori. Inoltre, la parallelizzazione ha permesso comunque di ottenere gli stessi risultati della versione Naive, risultando essere un’ottima soluzione.  
Con l’utilizzo di 2 job si è ottenuto però solo un miglioramento di circa 15 minuti, ottenendo un tempo di esecuzione pari a 50 minuti. Con l’utilizzo di 4 job, invece, si è ottenuta una netta velocizzazione dell’algoritmo, il quale ha impiegato 25 minuti per essere eseguito. Non sono stati provati più di 4 job a causa dei limiti fisici del computer su cui è stato testato l’algoritmo, ma, in generale, al fine di ottenere tempi di esecuzione più brevi possibili, si consiglia la scelta di un numero di job pari al numero di core logici presenti sull’elaboratore.  
Tale soluzione è risultata essere soddisfacente in quanto ha permesso di ottenere un tempo di esecuzione di 25 minuti, contro i 65 della versione Naive, senza una perdita di precisione dell’algoritmo.  
Si è però comunque cercato di migliorare tale tempo effettuando un sampling dei nodi su cui calcolare la closeness. Tale sampling è stato fatto andando ad estrarre nodi random dal grafo, escludendoli dal set di nodi su cui effettuare la BFS, ma includendo tutti i suoi vicini. Tale operazione è stata effettuata fin quando ogni nodo o era stato escluso o era il vicino di un escluso. Ciò, quindi ha permesso di ottenere una misura della misura di centralità per tutti i vicini dei nodi esclusivi, potendo così stimare la closeness degli esclusi come media aritmetica della closeness dei suoi vicini. Questa stima, pur causando una diminuzione dell’accuratezza, è risultata adatta a questo problema in quanto per ogni best path calcolato (u,v), andando a considerare il vertice x, vicino escluso di u, il path (x,v) ha lunghezza compresa tra (u,v)+1 e (u,v)-1. Possiamo quindi supporre che utilizzare la media della closeness dei vicini per il calcolo della closeness degli esclusi permette di non discostarci troppo dal valore di centralità reale.  
Tale versione è stata eseguita in maniera parallela con l’utilizzo di 4 job, dato che, come detto in precedenza, la parallelizzazione permette di ottenere tempi di esecuzioni migliori senza perdita di accuratezza, ottenendo così una riduzione del tempo di esecuzione di 10 minuti rispetto alla versione priva di sampling, passando da 25 minuti a 15 minuti. Questa operazione ha permesso di escludere dal calcolo della BFS circa 9000 nodi sui 22470 iniziali, ottenendo però un risultato leggermente distante da quello del Naive. Infatti, all’interno della top 500, solo 270 nodi sono presenti anche nella top 500 della versione Naive, dove, in particolare, solo l’elemento con la closeness maggiore ha visto la propria posizione rimanere inalterata. Inoltre, possiamo notare che, andando a considerare solo i 10 elementi con la closeness maggiore dati dalla versione sampled, solo 6 nodi risultano essere presenti nella top 500 reale. Possiamo quindi concludere che, considerando il trade-off tra accuratezza e tempi di esecuzione, l’utilizzo del sampling per questo tipo di metrica risulta non soddisfacente. Di conseguenza, per grafi di tali dimensioni, risulta preferibile utilizzare una semplice versione parallela.