**Presentazione DPAP**

L’algoritmo esamina le casistiche di anonimizzazione dei grafi tramite due tecniche due differenti approcci relativi sia alla costruzione del grafo anonimizzato che al costruzione dei gradi del grafo stesso.

In particolare l’implementazione si è concentrata sulle versioni greedy e dynamic dell’algoritmo di anonimizzazione, per poi rigenerare il grafo a partire dall’output delle due componenti sopra citate.

**TECNICHE DI ANONIMIZZAZIONE**

**DP**

Questa variante utilizza la tecnica della programmazione dinamica per generare il grafo con costo di anonimizzazione ottimo, cioè minore. Per ottenere ciò l’algoritmo valuta ad ogni passo se sia conveniente creare un nuovo cluster o unire il nodo considerato a questo passo al cluster precedente. Quest’operazione viene eseguita per ogni nodo e per ogni combinazione di sotto cluster che si è generata finora, tenendo in memoria man mano che si procede nella computazione dei migliori cluster ottenuti in base al costo totale di anonimizzazione che ciascuno di loro (seppur parziale) comporta.

**GREEDY**

Questa variante utilizza un approccio più semplice andando a considerare per ogni nodo da inserire nel grafo anonimizzato il singolo costo di unione all’ultimo cluster confrontandolo con quello di creazione di un nuovo cluster. Questa variante ovviamente non esplora tutte le combinazioni possibili e non tenendo memoria degli ottimi locali per i singoli cluster trovati, e genera quindi soluzioni sub ottime rispetto al DP per quanto riguarda i costi di anonimizzazione.

**TECNICHE DI COSTRUZIONE**

Entrambe le tecniche proposte utilizzano il vettore dei gradi generato da una delle due tecniche di anonimizzazione viste precedentemente per ricreare una versione del grafo che rispetti il vincolo di anonimizzazione K.

**CONSTRUCT**

Questa tecnica di costruzione del grafo riassembla i vari nodi in maniera randomica: ciò comporta che ad ogni persona venga attribuito un numero di archi pari al numero indicato dal grado di anonimizzazione presente nel vettore sopra citato, tuttavia gli archi così generati non avranno alcun riferimento con quelli presenti nel grafo originale non anonimizzato.

**PRIORITY**

Questa tecnica di costruzione del grafo**,** viceversa, parte dal grafo originale e aggiunge solamente ad ogni nodo un numero di archi necessario a portarlo al nuovo grado di anonimizzazione precedentemente calcolato, mantenendo una coerenza col grafo originale. Dato che questa versione dell’algoritmo di costruzione del grafo deve valutare per ogni nodo l’introduzione di ogni possibile arco fino al raggiungimento del grado richiesto per l’anonimizzazione, è atteso che la complessità sia più elevata.

**CONCLUSIONE**

Analizzando i risultati prodotti dagli script si evince che a livello computazionale il carico comportato dall’esecuzione dell’algoritmo di anonimizzazione DP risulta nettamente superiore rispetto alla computazione prodotta tramite l’algoritmo greedy, indipendentemente dalla tecnica di costruzione del grafo scelto. Per quanto riguarda i costi ottenuti, i risultati avvalorano la tesi esposta nel paper, evidenziando il fatto che l’algoritmo di anonimizzazione DP produce in ogni caso soluzioni migliori. Tuttavia va tenuto conto del trade-off tra tempi di esecuzione e reale risparmio in termini di costi: su istanze di dimensione superiori al migliaio di nodi la differenza di costo per lo stesso data set anonimizzato con lo stesso parametro k varia nell’ordine di un migliaio mentre invece le differenze nei tempi di produzione del grafo anonimizzato incrementa fino a raddoppiare nella media i tempi.

Inoltre possiamo concludere che l’algoritmo di costruzione del grafo anonimizzato *construct* produce un a soluzione in tempi molto ristretti, tuttavia non mantiene coerenza col grafo originale introducendo molto rumore e rendendo la soluzione poco utile ad analisi successive.

Per contro, l’algoritmo *priority* mantiene coerenza col grafo originale al prezzo di un onere computazionale decisamente più alto, crescente con la dimensione dell’istanza.

In conclusione l’algoritmo migliore dipende dalle necessità di chi deve manipolare successivamente il grafo anonimizzato:

* Nel caso di frequenti analisi su data set estesi è consigliabile l’utilizzo di una combinazione **DP – Construct** per non incombere in tempi computazionali troppo elevati mantenendo un costo di anonimizzazione ottimo
* Nel caso di dati molto affidabili e coerenti con il grafo di provenienza, la scelta consigliata dai dati ottenuti risulta essere l’utilizzo della combinazione **Greedy – Priority** che conferisce tempi medi di esecuzione leggermente più alti di quelli ottenuti con il caso precedente, introducendo però la coerenza dei dati.
* Per anonimizzazioni rapide su data set molto estesi (utile ad esempio per un web service) la soluzione migliore risulta essere la combinazione **Greedy – Construct**.
* Per la massima precisione e coerenza dei dati è necessario utilizzare **DP – Priority** al costo di tempi di esecuzione molto elevati su istanze anche non estremamente estese.