Targeting with Partial Incentives (TPI)

Sommario

1. Introduzione

In questa relazione viene analizzato l'algoritmo "TPI" descritto nel paper "*Whom to Befriend to Influence People*"(Con chi fare amicizia per influenzare le persone) scritto nel 2016 da Gennaro Cordasco, Luisa Gargano, Manuel Lafond, Lata Narayanan, Adele A. Rescigno, Ugo Vaccaro e Kangkang Wu. Lo pseudo-codice esposto nel paper è stato convertito in Python (versione 3.7) ed utilizza la libreria **SNAP** (*Stanford Network Analysis Project*). Il Dataset utilizzato è una versione ridotta a 60050 archi e 37873 nodi del dataset "*YouTube social network and ground-truth communities*", un dataset che analizza le relazioni tra utenti (detti *Youtuber* o *canali*) del popolare sito di condivisione video YouTube. Ho scelto tale sito perché non è raro imbattersi in video in cui gli utenti più popolari recensiscono o aprono in diretta (*unboxing*) prodotti appena usciti sul mercato e spesso tali video sono commissionati dalle aziende stesse (che spediscono il prodotto in omaggio allo Youtuber o offrono codici sconto che lo stesso può elargire ai suoi iscritti) al fine di promuovere il lancio sul mercato del prodotto visto nel video.

2. Problema

Data una rete sociale rappresentata con un grafo G = (V, E, t), con V insieme dei nodi della rete, E insieme degli archi che rappresentano i collegamenti tra i nodi della rete e t funzione di soglia di attivazione per ogni nodo della rete, vogliamo trovare un "***link set pervasivo ottimale***", ovvero un insieme di nodi che sia in grado di "attivare" l'intera rete. Questo problema ha applicazione nel marketing virale, in cui bisogna selezionare un particolare insieme di utenti di un social network (***seed set***) tali da poter influenzare il maggior numero possibile di altri utenti al fine di far acquistare loro un nuovo prodotto o servizio. Agli utenti inclusi nel seed set viene fornito un incentivo (parziale, come ad esempio un coupon sconto o totale come ad esempio un campione omaggio del prodotto) al fine di influenzarli e indurli a parlare del prodotto ai loro contatti per sfruttare l'effetto passaparola. Formalmente,

Definizione (*problema dei collegamenti minimi, Min-Links*). Dato un social network G = (V, E, t), dove t è la funzione di soglia su V, e un insieme di nodi esterni U, trova un link set pervasivo ottimale per (G, U). (tradotto da "*Whom to Befriend to Influence People*")

3. Algoritmo

Viene ora illustrato e commentato l'algoritmo TPI realizzato in Python, i numeri di riga (#xx) si riferiscono allo pseudo-codice fornito nel paper. Il codice è liberamente consultabile su GitHub dove è inoltre presente una copia di questa relazione ed una copia del file Excel con i risultati dei test discussi più avanti nel paragrafo 4.
#1: Vengono create le strutture dati a supporto dell'algoritmo: S (che conterrà la soluzione), W (che conterrà una copia temporanea dei nodi), delta (che conterrà i gradi dei nodi), k (le Tresholds) e N (il vicinato di ogni nodo)
#2: Per ogni nodo presente nel grafo,
#3: La soluzione corrispondente viene settata a 0, il nodo viene inserito in W,
#4: si salva il grado in delta
#5: e la soglia in k.
#6: Viene calcolato il vicinato di ogni nodo e salvato in N (eseguito all'esterno del ciclo per ottimizzazione)
#7: L'algoritmo itera finché W non è vuoto
#8: La funzione computeValue(W, k, delta, g) combina le righe 8 e 14 dello pseudo codice. Se presente, restituisce immediatamente un nodo v tale che la soglia t del nodo sia superiore al suo grado d (k > d). Altrimenti, restituisce il nodo che massimizzi la funzione (t*(t+1))/(d*(d+1)). Viene svolto tutto in un'unica funzione per ottimizzazione.
#9: Se il nodo v soddisfa la prima condizione, la soluzione corrispondente a v viene incrementata di k[v]-delta[v] (si fornisce a v un incentivo pari alla soglia meno il suo grado),
#10: La soglia del nodo viene impostata pari al suo grado e
#11: se la soglia diventa pari a 0
#12: il nodo viene eliminato da W e si torna al punto #8
#13, #14: Altrimenti se il nodo soddisfa la seconda condizione
#15: Si itera tra i vicini u di v
#16: e si riduce la soglia di ogni vicino u di 1
#17: Qui andrebbe rimosso v dall'elenco dei vicini di u, ma nella versione in Python, per efficienza, non è necessario in quanto v viene rimosso e viene ridotta la soglia di u.
#18: si rimuove v da W e si torna al punto #8
#19: Quando W è vuoto si restituisce la soluzione S

4. Esecuzione del test

Dopo aver caricato il grafo, viene applicato il principio di "***decisione differita***": per ogni arco del grafo viene generato un numero pseudocasuale compreso tra 0 e 1. Se il numero generato è minore della probabilità presente sull'arco (cioè il nodo infetta con una probabilità inferiore rispetto a quella richiesta), l'arco viene rimosso. Questa può essere applicata in modo uniforme (la probabilità dell'arco è scelta in modo pseudo-casuale, se tale valore è maggiore della probabilità di rimozione, anch'essa scelta in modo pseudo-casuale, l'arco viene rimosso) o proporzionale (la probabilità dell'arco è 1/il grado del nodo, se tale valore è maggiore della probabilità di rimozione, scelta in modo pseudo-casuale, l'arco viene rimosso). Le Tresholds vengono invece calcolate in 3 diversi metodi: costante (pari a 2), casuale (viene scelto un valore pseudo-casuale compreso tra 1 e 5) e proporzionale al grado (se il grado è diverso da 0 la soglia è pari a (1 /(grado del nodo + 5)) * (numero di archi /numero di nodi), se tale valore è inferiore a 1 o il grado è pari a 0 la soglia è 5). Vengono eseguite 6 diverse combinazioni in cui si varia il principio di decisione differita e l'assegnazione della soglia. Ogni combinazione viene ripetuta per 10 volte. Al termine di ogni iterazione viene dato in output la size della soluzione (il numero di nodi che riceve un incentivo maggiore di 0) e il totale degli incentivi da elargire (il budget). Al termine delle 10 iterazioni viene calcolata la media del numero di nodi da influenzare e la media dell'incentivo da spendere su tali nodi.

5. Risultati

I risultati sono allegati nel documento Excel Test_Result.xlsx, vengono qui di seguito inseriti i grafici risultanti da tali dati. I risultati saranno discussi al termine dei grafici.

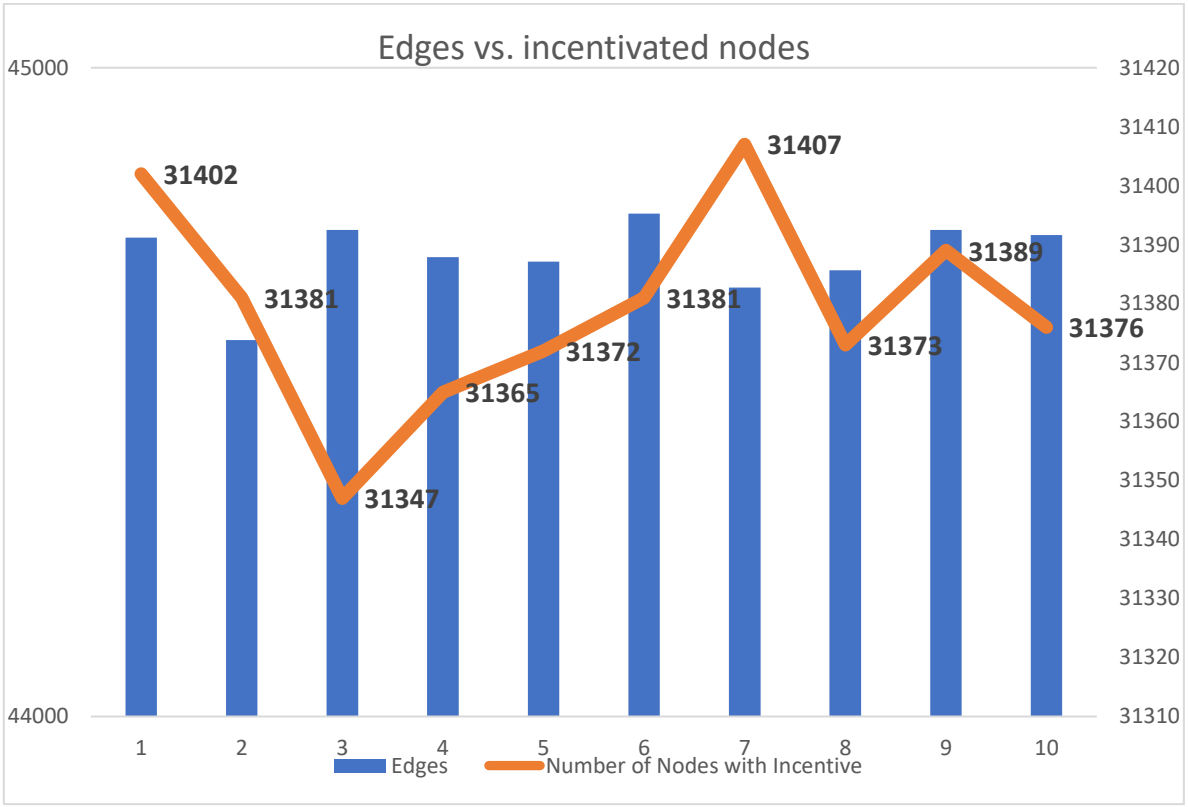

Grafico 1. Decisione Differita Uniforme e Tresholds Costanti – Numero di Archi e numero di nodi incentivati

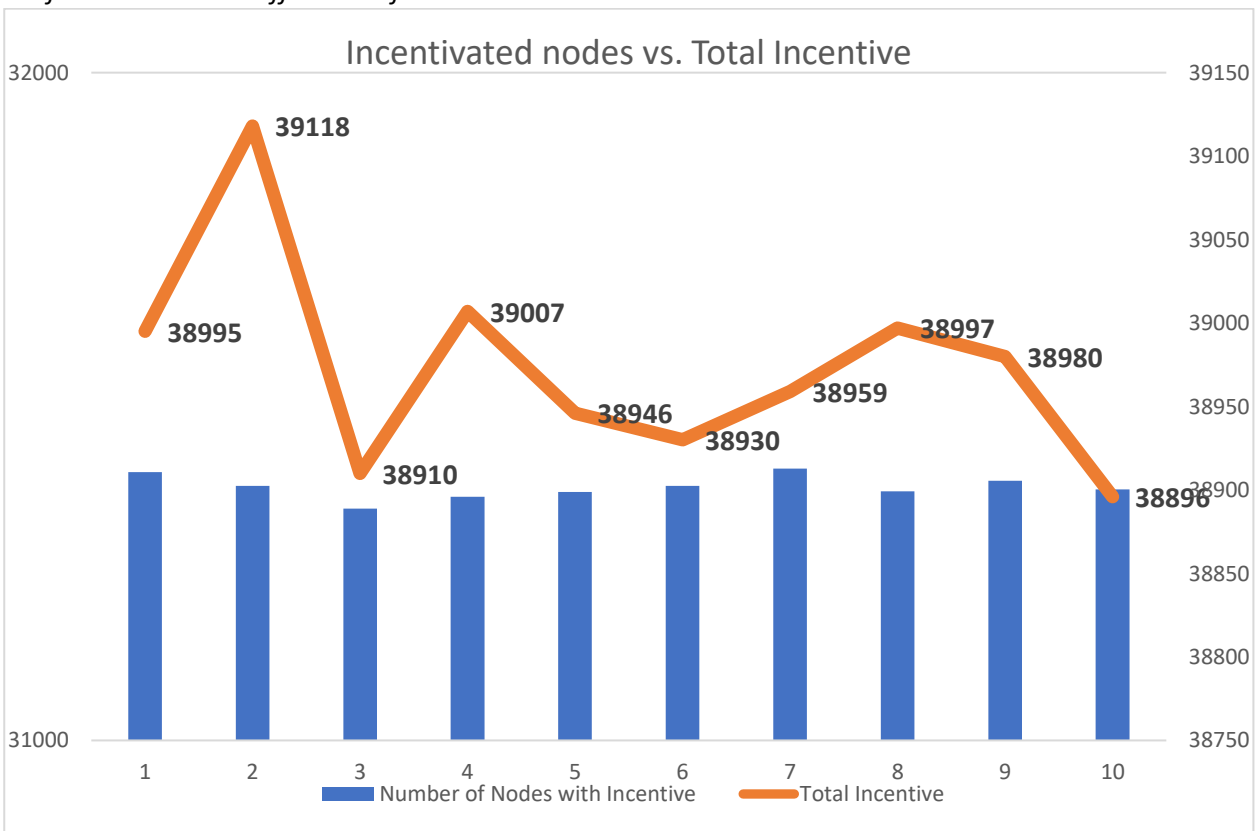

Grafico 2. Decisione Differita Uniforme e Tresholds Costanti – Numero di nodi incentivati e incentivo totale

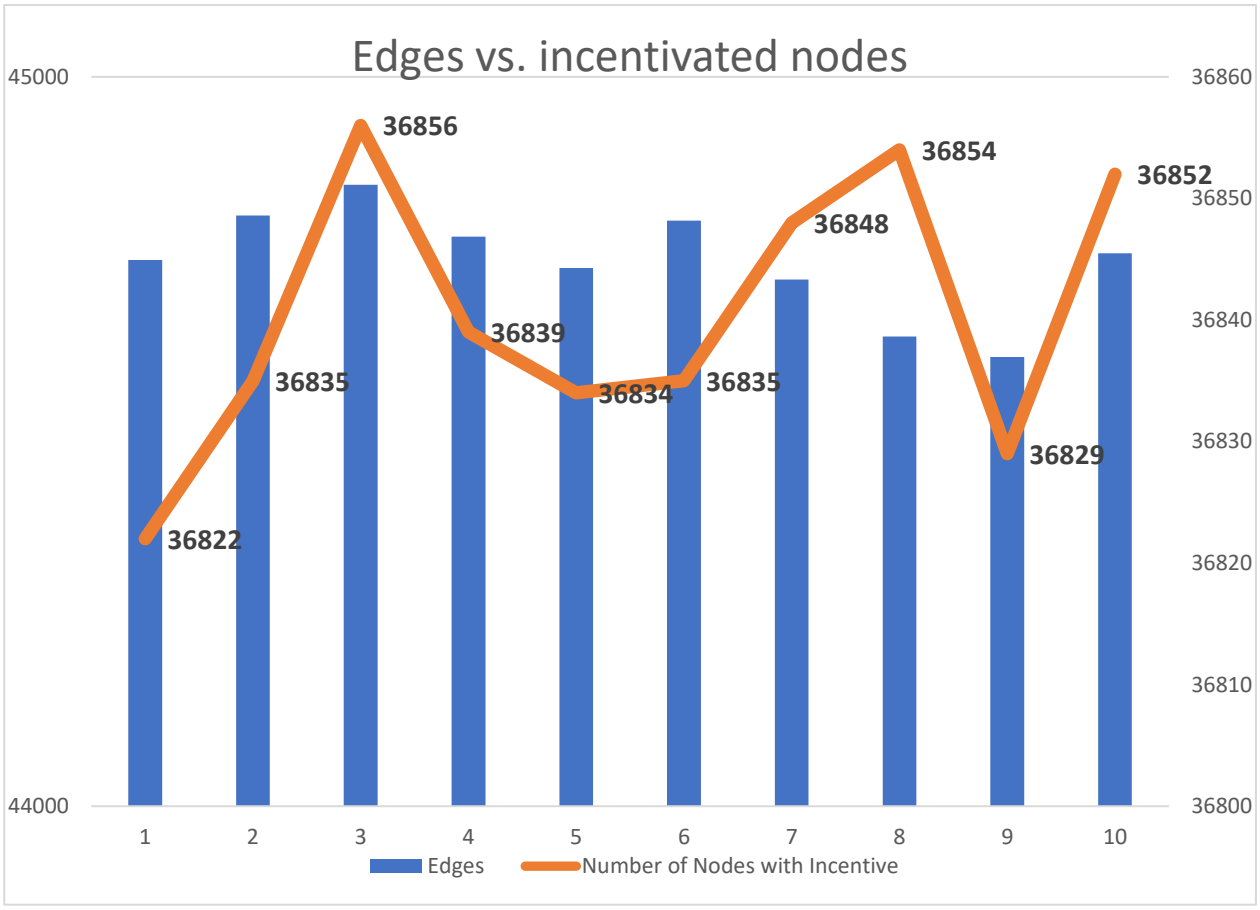

Grafico 3. Decisione Differita Uniforme e Tresholds Proporzionali – Numero di Archi e # di nodi incentivati

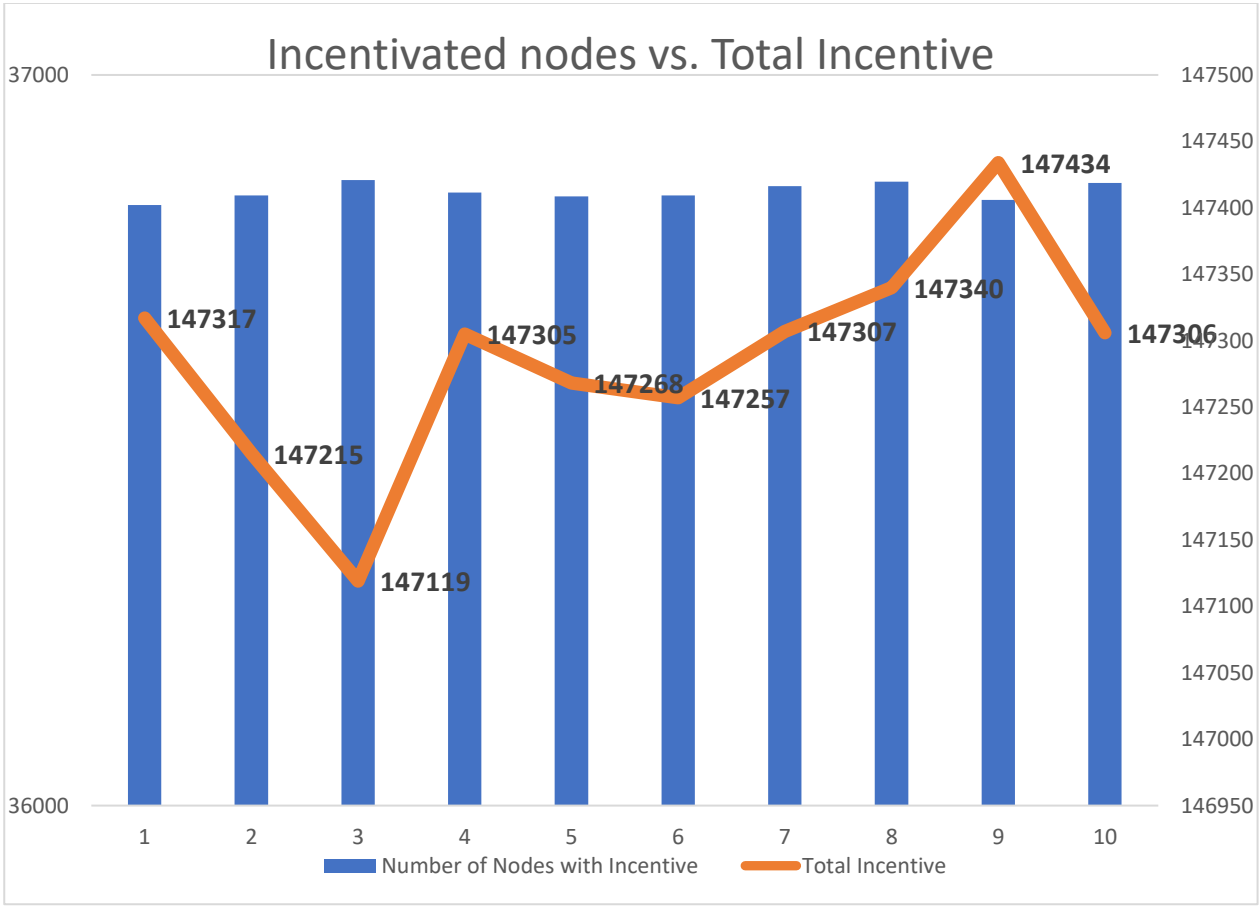

Grafico 4. Decisione Differita Uniforme e Tresholds Proporzionali – # di nodi incentivati e incentivo totale

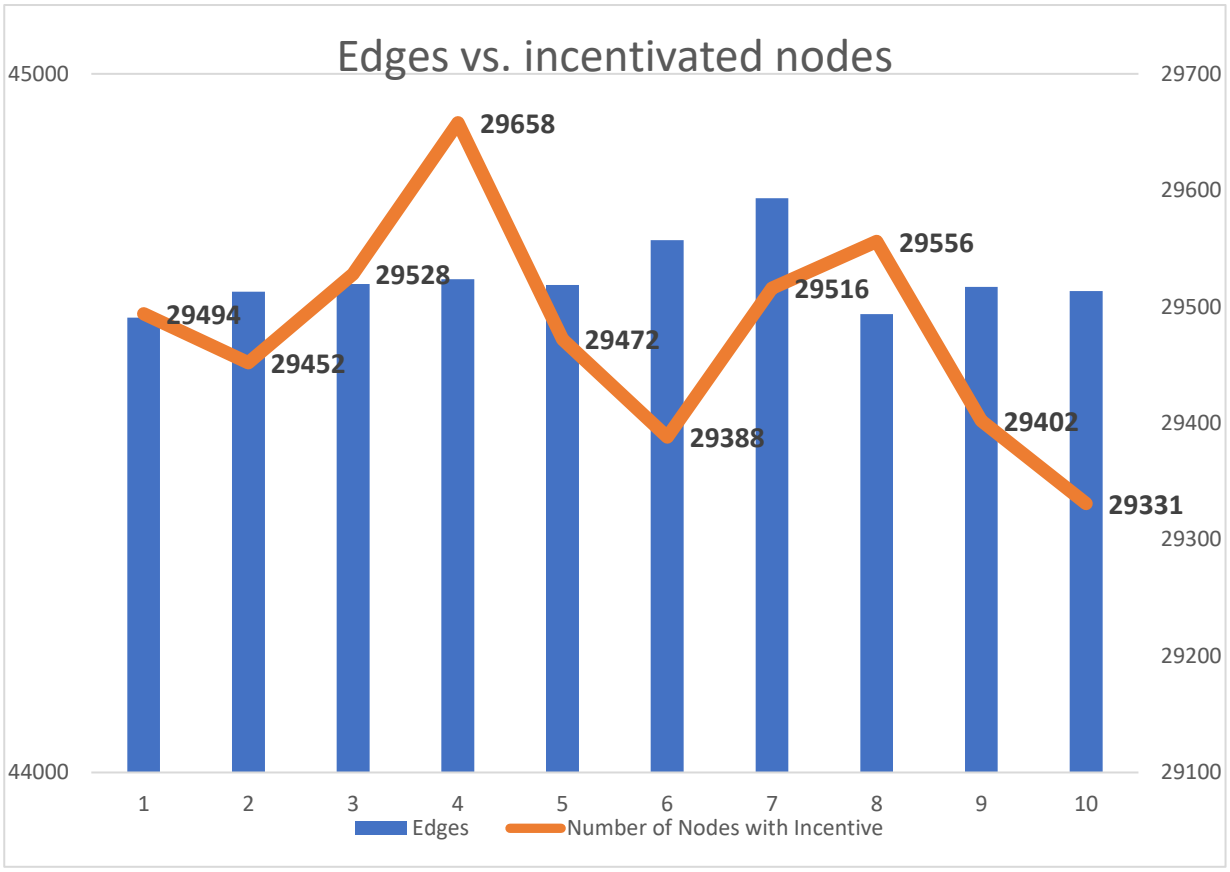

Grafico 5. Decisione Differita Uniforme e Tresholds Casuali – Numero di Archi e numero di nodi incentivati

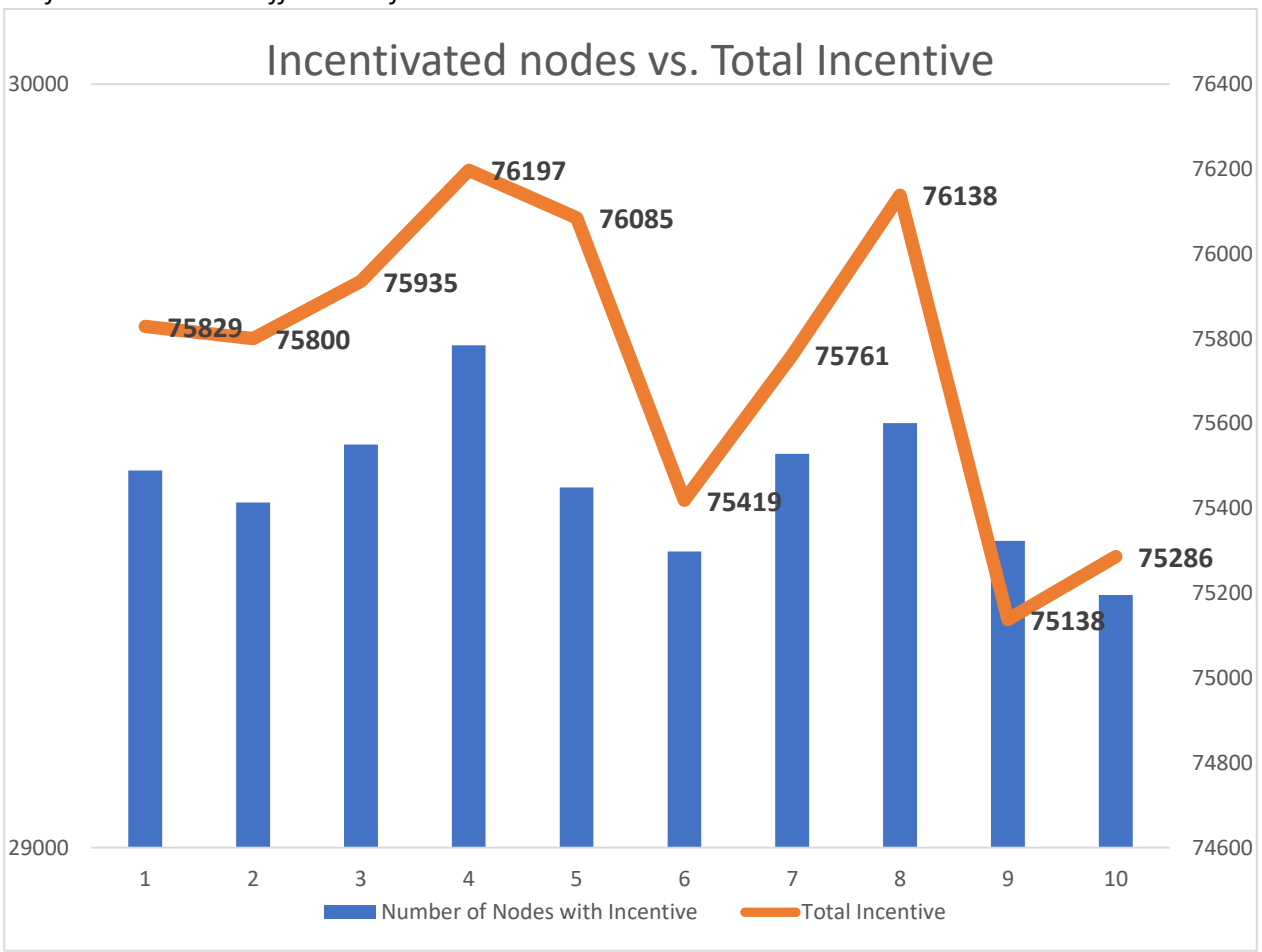

Grafico 6. Decisione Differita Uniforme e Tresholds Casuali – numero di nodi incentivati e incentivo totale

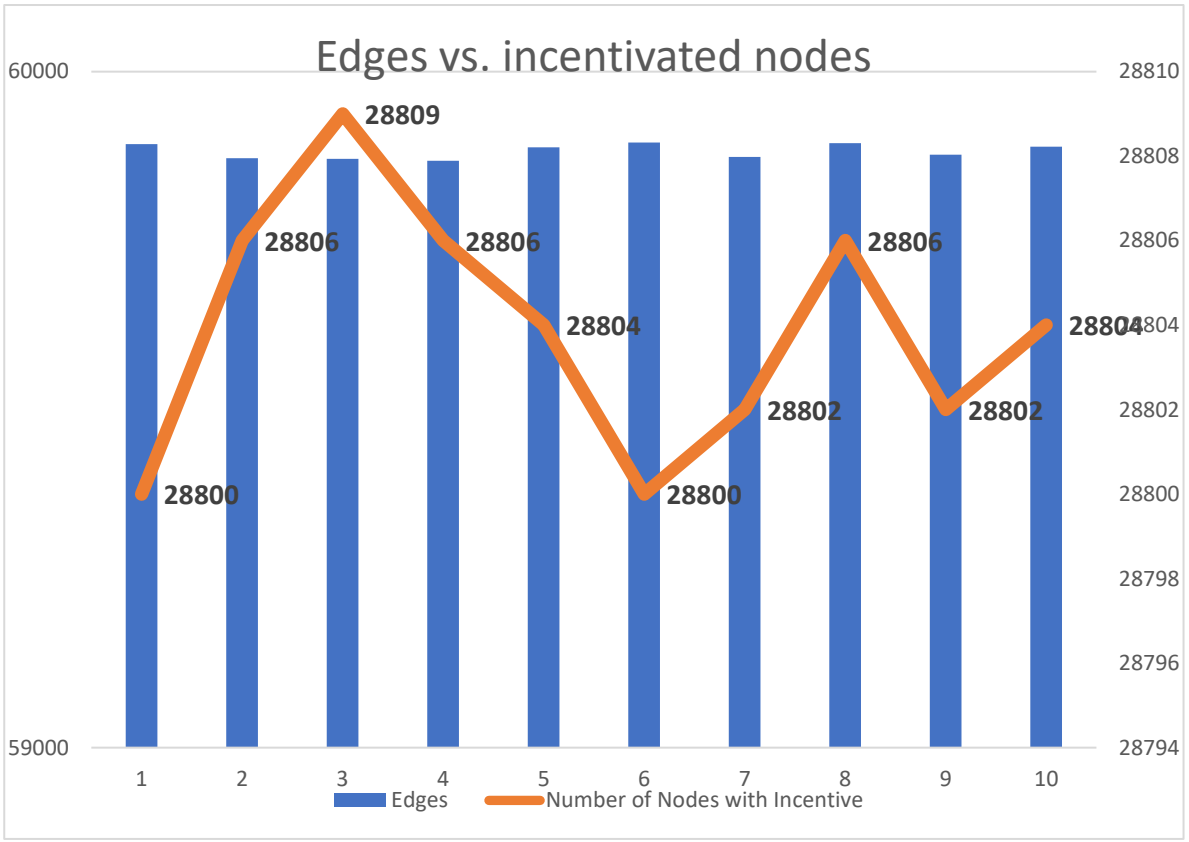

Grafico 7. Decisione Differita Proporzionale e Tresholds Costanti – Numero di Archi e # di nodi incentivati

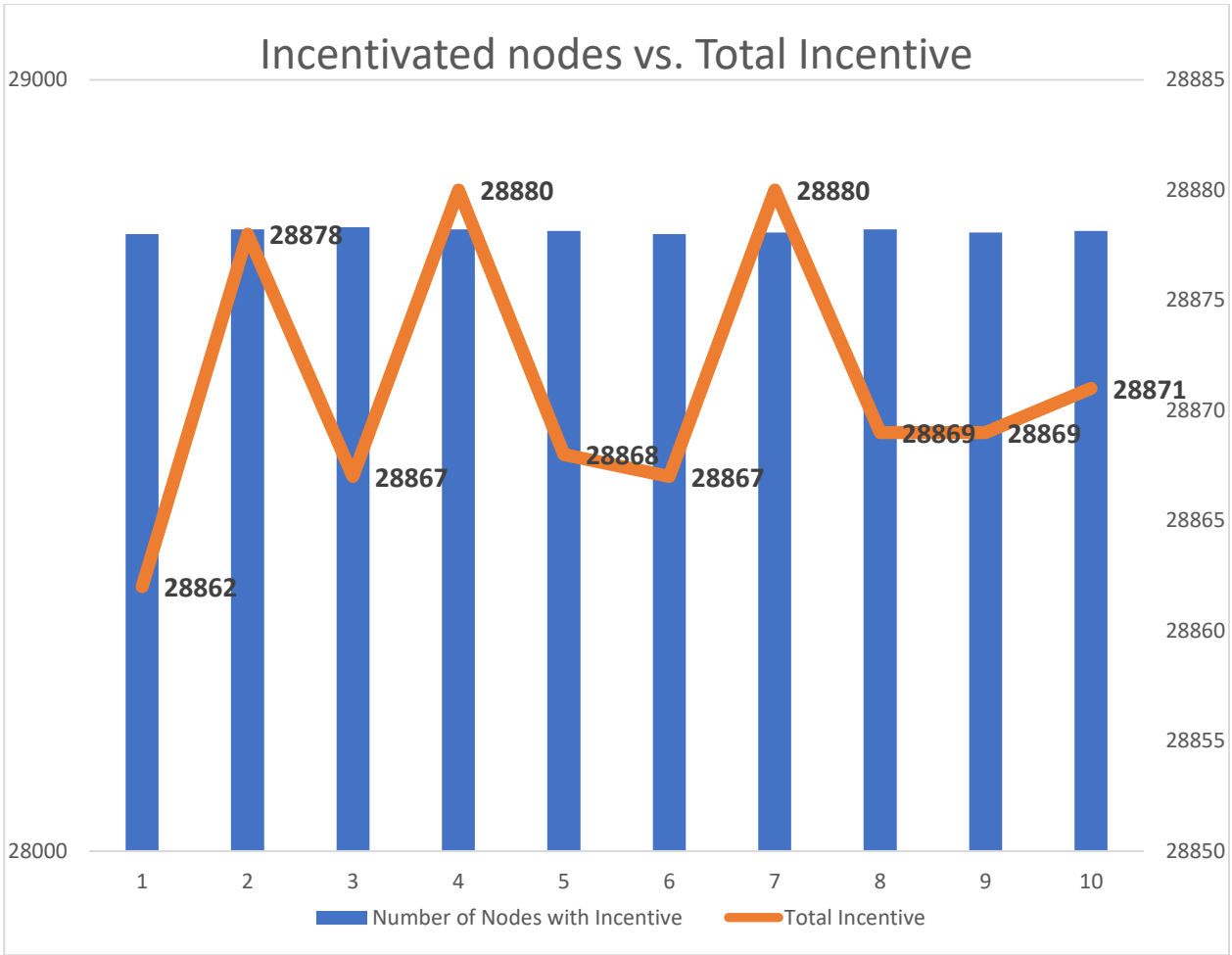

Grafico 8. Decisione Differita Proporzionale e Tresholds Costanti – # di nodi incentivati e incentivo totale

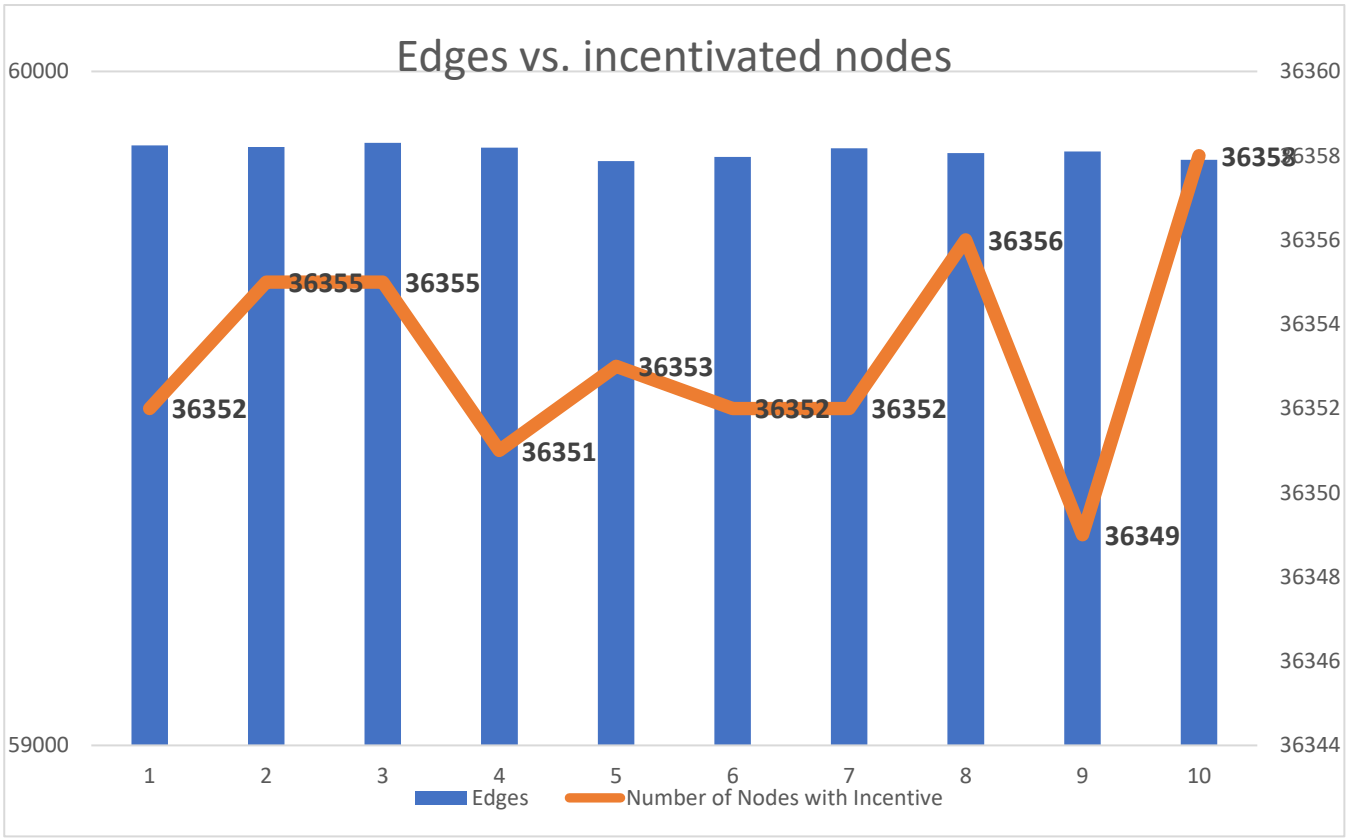

Grafico 9. Decisione Differita Proporzionale e Tresholds Proporzionali – Numero di Archi e # di nodi incentivati

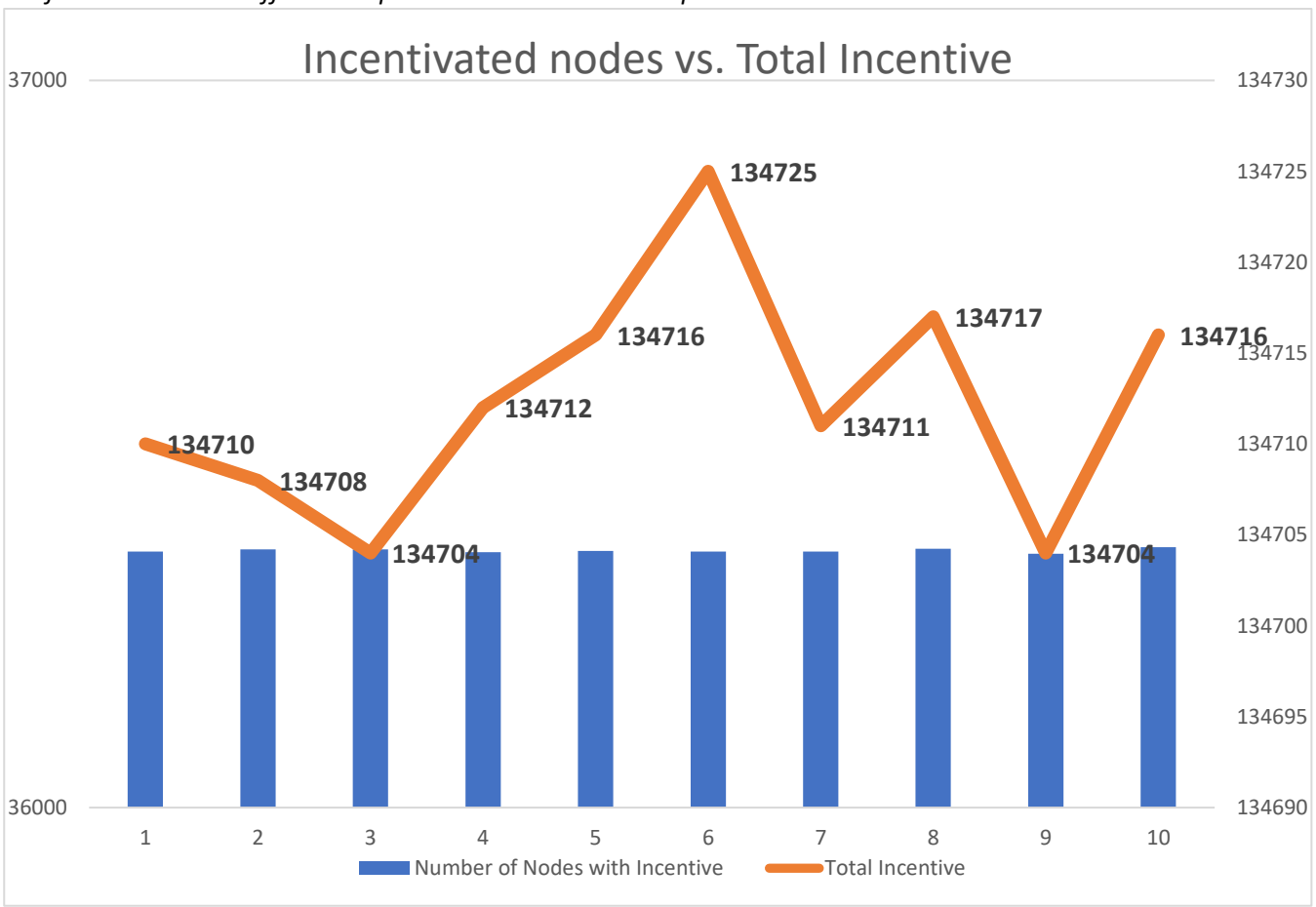

Grafico 10. Decisione Differita Proporzionale e Tresholds Proporzionali – # di nodi incentivati e incentivo totale

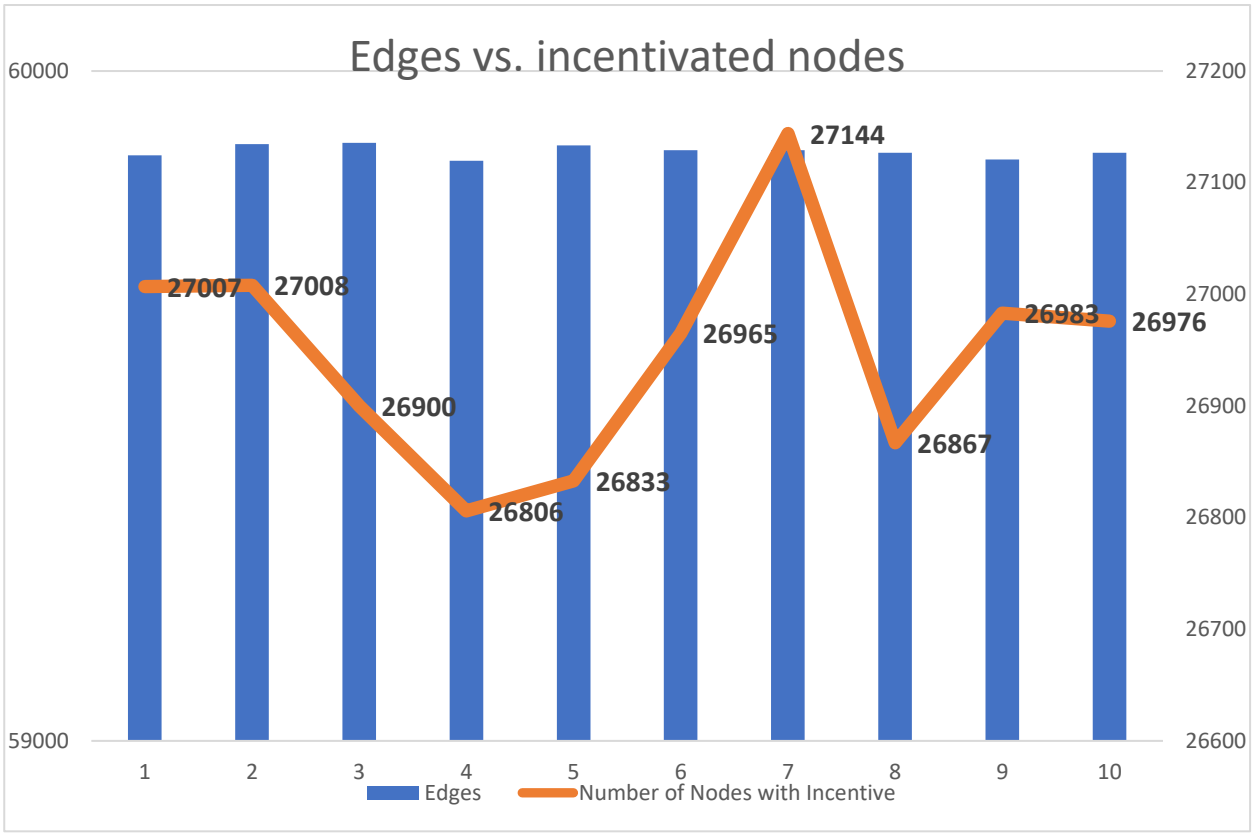

Grafico 11. Decisione Differita Proporzionale e Tresholds Casuali – Numero di Archi e # di nodi incentivati

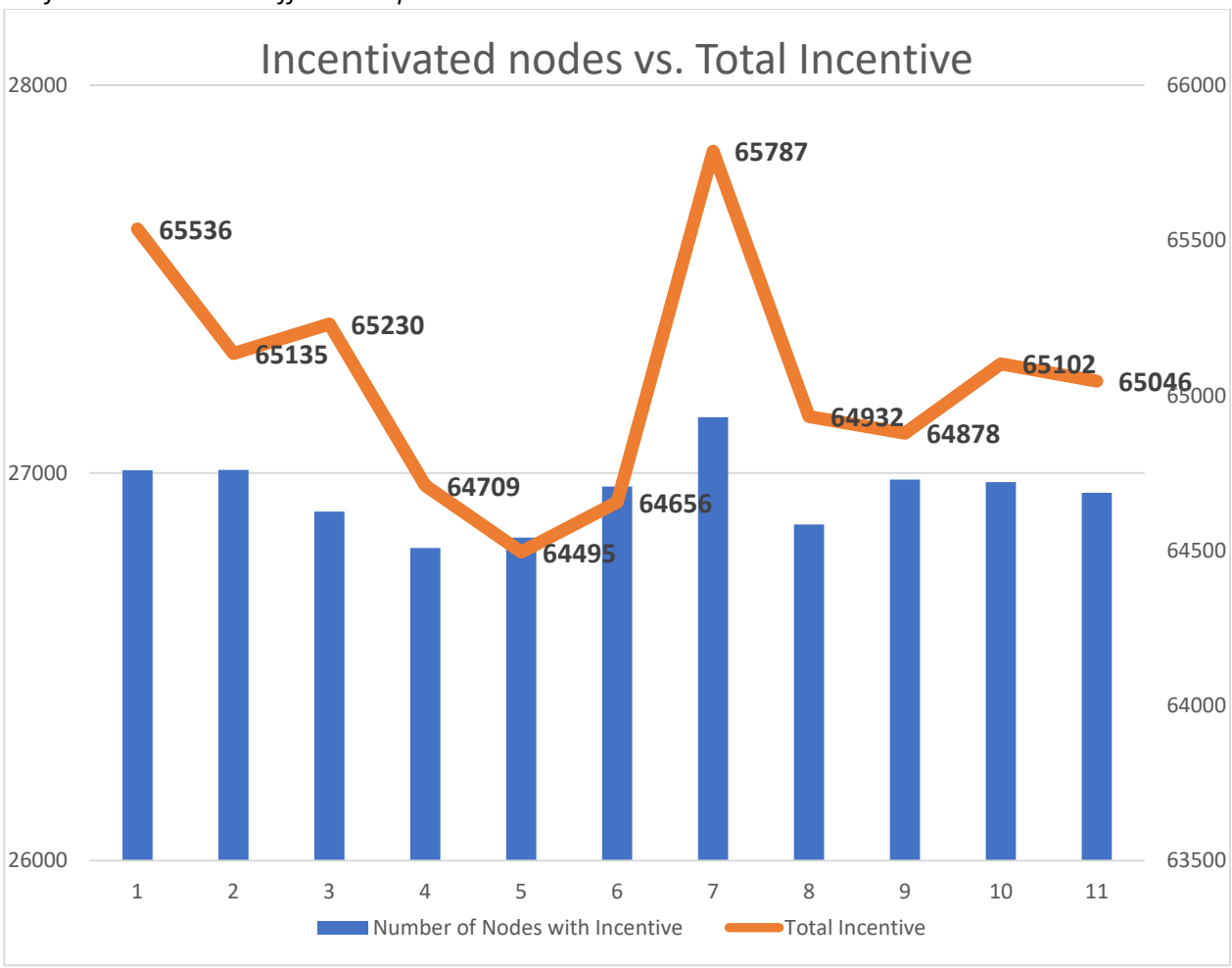

Grafico 12. Decisione Differita Proporzionale e Tresholds Casuali – #di nodi incentivati e incentivo totale

Applicando il principio di decisione differita uniforme e Tresholds costanti si ottiene che il numero medio di nodi che ricevono un incentivo è 31.379,3 mentre la media degli incentivi da elargire è 38.973,8.
Applicando il principio di decisione differita uniforme e Tresholds proporzionali si ottiene che il numero medio di nodi che ricevono un incentivo è **36.840,4** mentre la media degli incentivi da elargire è **147.286,8**.
Applicando il principio di decisione differita uniforme e Tresholds casuali si ottiene che il numero medio di nodi che ricevono un incentivo è 29.479,7 mentre la media degli incentivi da elargire è 75.758,8.
Applicando il principio di decisione differita proporzionale e Tresholds costanti si ottiene che il numero medio di nodi che ricevono un incentivo è 28.803,9 mentre la media degli incentivi da elargire è **28.871,1**.
Applicando il principio di decisione differita proporzionale e Tresholds proporzionali si ottiene che il numero medio di nodi che ricevono un incentivo è 36.353,3 mentre la media degli incentivi da elargire è 134.712,3.
Applicando il principio di decisione differita proporzionale e Tresholds casuali si ottiene che il numero medio di nodi che ricevono un incentivo è **26.948,9** mentre la media degli incentivi da elargire è 65.046.
L'incentivo minore (in verde) viene elargito applicando il principio di decisione differita proporzionale e Tresholds costanti, quello più grande (in rosso) con il principio di decisione differita uniforme e Tresholds proporzionali, che è anche quello che fornisce il seed set più grande. Il seed set più piccolo si ottiene applicando il principio di decisione differita proporzionale e Tresholds casuali.

6. Link Esterni

Repository su GitHub: https://github.com/vinsan/tpi
Whom to Befriend to Influence People: https://link.springer.com/chapter/10.1007/978-3-319-48314-6_22
SNAP: https://snap.stanford.edu/
YouTube social network and ground-truth communities: https://snap.stanford.edu/data/com-Youtube.html