

Social Network Analysis: Group 04

Cirillo Benedetto¹, Dario Civale²

^{1,2}{b.cirillo6, d.civale1}@studenti.unisa.it

1. Task 1

L'obiettivo di questo task è quello di implementare algoritmi di social network mining (**diameter**, **triangles computation**, **clustering**) che siano in grado di operare in modo efficiente sulle reti fornite, garantendo risultati abbastanza precisi. Le reti fornite per i test sono:

- *Facebook Large Page-Page Network*: rete indiretta (nodi: 22470, archi: 171002)
- *High-energy physics theory citation network*: rete diretta (nodi: 27770, archi: 352807)

Per semplicità chiameremo queste due reti rispettivamente **Facebook net** e **Cite net**.

1.1. Triangoli

Per l'individuazione dei triangoli abbiamo utilizzato i seguenti algoritmi:

- **Standard**: algoritmo standard che visita il vicinato di un nodo 2 volte, contando quindi 6 volte un triangolo (3 per la versione diretta)
 - **Parallel standard**: versione parallela dell'algoritmo standard.
 - **Optmized**: versione ottimizzata che distingue tra nodi con un alto grado (*heavy hitters*) e nodi di basso grado. I triangoli che coinvolgono solo gli heavy hitters (che sono il collo di bottiglia dell'algoritmo standard) vengono gestiti in modo diverso rispetto ai triangoli rimanenti.
 - **Node Iterator Plus Plus**: Questo algoritmo permette di conteggiare solo una volta ogni triangolo, sfruttando un id assegnato a ogni nodo. Un triangolo viene contato solo nel caso in cui $id(u) < id(v) < id(w)$.
 - **Node Iterator N**: Algoritmo per il conteggio dei triangoli in un grafo indiretto, analizzato nel paper *Fast Parallel For Counting and Listing Triangles in Big Graphs* [6]. Come *NodeIteratorPlusPlus* nella fase iniziale assegna un id per ogni nodo ma anche un vicinato N_u (un nodo v appartiene al vicinato di u se sono verificate le seguenti condizioni:
 - Esiste un arco tra u e v
 - $id(u) < id(v)$
- Per ogni nodo u , si esplorano i nodi $v \in N_u$ e il conteggio dei triangoli sarà incrementato in base alla cardinalità dell'intersezione tra l'insieme N_u e N_v . I nodi appartenenti a questo insieme hanno un id più alto sia di u che di v quindi vengono contati anche in questo algoritmo solo nel caso in cui $id(u) < id(v) < id(w)$.
- **Parallel Node N**: versione parallela dell'algoritmo Node Iterator N.

Algoritmo	Tempo	Num. triangoli
Standard	3.144	794953
Parallel standard	3.043	794953
Optimized	4.82	794953
Networkx	1.526	794953
nodeIteratorPlusPlus	0.802	794953
nodeIteratorN	0.215	794953
Parallel nodeIteratorN	0.218	794953

Table 1. Numero triangoli su **Facebook net**

Algoritmo	Tempo	Num. triangoli
Standard	1.281	545
Parallel standard	1.522	545
Optimized	8.414	545

Table 2. Numero triangoli su **Cite net**

Per i grafi **diretti** sono stati utilizzati solo i primi 3 algoritmi.

Possiamo vedere come gli algoritmi restituiscano tutti il risultato corretto (quello restituito dalla libreria *Networkx*), come anche che gli algoritmi **Node Iterator Plus Plus** e **Node Iterator N** raggiungano le migliori performance in termini di tempi.

1.2. Diametro

Per il calcolo del diametro della rete abbiamo utilizzato i seguenti algoritmi:

- **Standard**: algoritmo standard che esegue un **BFS** per ogni nodo e restituisce l'altezza dell'albero **BFS** più alto.
- **Parallel standard**: versione parallela dell'algoritmo standard.
- **Optimized**: Questo algoritmo restituisce solo un'approssimazione del diametro. Esplora tutti gli archi più volte per un numero di step che è approssimativamente il diametro del grafo. Pertanto, ha una complessità computazionale $O(\text{diam} \cdot m)$, che di solito è molto più veloce della complessità dell'algoritmo precedente $O(n \cdot m)$.

Algoritmo	Tempo	Diametro
Standard	563.064	15
Parallel standard	204.135	15
Optimized	0.262	6
Networkx	561.102	15

Table 3. Diametro di **Facebook net**

Si nota come l'algoritmo ottimizzato non porta una buona approssimazione.

1.3. Clustering

2. Task 2

Il task 2 si basa sempre sull'implementazione e l'analisi di algoritmi per calcolare altri parametri quantitativi in grado di descrivere una social network, la **centralità**. Gli algoritmi analizzati sono

Algoritmo	Tempo	Diametro
Standard	217.702	37
Parallel standard	95.147	37
Optimized	0.21	5

Table 4. Diametro di Cite net

stati:

- **degree**: Misura l'importanza di un nodo in termini di numero di sue relazioni.
- **closeness**: Misura la velocità con cui un nodo può essere raggiunto (o viene raggiunto) da tutti gli altri nodi della rete.
- **betweenness**: Misura quanto un nodo è cruciale per la trasmissione di informazioni tra ogni coppia di nodi della rete.
- **PageRank**: Misura la frequenza con cui il **random surfer** può essere trovato su una determinata pagina. (Un random surfer è una persona che naviga a caso in una rete Web. Una volta arrivato su una pagina, segue un link casuale dalla pagina corrente e lo segue fino a dove conduce).
- **HITS** (authority, hubiness e entrambi): Misura il valore di un nodo come potenziale autorità e come potenziale hub o entrambi.
- **VoteRank**: Questa misura di centralità è un'estensione della centralità *degree*, che cerca di evitare di scegliere nodi troppo vicini tra loro.

Inoltre anche le misure centralità Shapley che applicano il concetto di valore di Shapley per valutare l'importanza di un nodo nella rete sociale. Queste misure tengono conto di tutte le possibili combinazioni di nodi nella rete e valutano il contributo marginale di un nodo quando viene aggiunto o rimosso da queste combinazioni. Il valore di Shapley di un nodo riflette quindi il suo contributo medio in tutte le possibili configurazioni di coalizione. Si vuole quindi quantificare la sfera di influenza della coalizione di nodi C sugli altri nodi della rete. I metodi **shapley-degree** e **shapley-threshold** calcolano il valore di Shapley per una funzione caratteristica che estende la **degree centrality** alle coalizioni: in particolare, la sfera d'influenza di una coalizione di nodi C è l'insieme dei nodi esterni a C con almeno k vicini in C . questo vale per entrambe le misure per **shapley-degree** $k=1$ per **shapley-threshold** $k>1$. Per il valore di **shapley-closeness** si calcola il valore di Shapley per una funzione caratteristica che estende la **closeness centrality**.

2.1. Risultati

Le misure di centralità HITS sono possibili solo per grafi diretti. La misura di centralità betweenness è quella che ci mette più tempo, come anche il voterank.

3. Task 3

L'obiettivo di questo task è quello di implementare i meccanismi **VCG**, **MUDAN** e **MUDAR**, per la vendita di più oggetti omogenei su una rete sociale, con ogni buyer che richiede solo un singolo oggetto. Oltre a questi meccanismi abbiamo implementato anche l'asta **GIDM**.

Un meccanismo M è:

- è *incentive compatible* (**IC**) se per qualsiasi buyer **reporting truthfully** e **bidding truthfully** è una strategia dominante.

Armset	T (Facebook net)	T (Cite net)
degree	0.011	0.011
closeness	1035.620	287.483
betweenness	11365.809	out of time
PageRank	11.757	12.451
Linear PageRank	24.808	24.375
VoteRank	1347.285	out of time
HITS-both	n.p.	18.102
HITS-authority	n.p.	18.102
HITS-hubiness	n.p.	18.102
linear HITS-both	n.p.	541.732
linear HITS-authority	n.p.	541.732
linear HITS-hubiness	n.p.	541.732
shapley-degree	0.138	0.197
shapley-threshold	0.149	0.145
shapley-closeness	608.344	out of time

Table 5. Tempo in secondi delle varie misure di centralità

Auction	<i>Truthful bidding</i>	<i>Truthful reporting</i>	Non-deficit
VCG [1]	<i>True</i>	<i>True</i>	<i>False</i>
MUDAN [3]	<i>True</i>	<i>True</i>	<i>True</i>
MUDAR [3]	<i>False</i>	<i>True</i>	<i>True</i>
GIDM [4]	<i>True</i>	<i>True</i>	<i>True</i>

Table 6

- **individually rational (IR)** se per qualsiasi buyer **reporting truthfully** non ti fa ricevere utilità negative.
- **non-deficit (ND)** o **Budget Balancedness** se il seller non rievve mai guadagni negativi.
- **efficiente** se raggiunge un social welfare ottimale.

A differenza delle aste tradizionali, per le *aste a diffusione* nessun meccanismo soddisfa contemporaneamente **IC**, **IR**, **ND** e **S.W.** ottimale.

3.1. GIDM

L'asta **IDM** soddisfa il requisito **IC** ma è pensato per il caso della vendita di un solo oggetto. Si potrebbe pensare che le aste multi-unità essendo una naturale generalizzazione della controparte a unità singola, possano essere affrontati con meccanismi simili all'**IDM**: **GIDM**.

L'asta **GIDM** (Generalized Information Diffusion Mechanism) generalizza **IDM** con l'introduzione di nuovi concetti: **genitore critico**, **figlio critico** e **albero di allocazione ottimale**.

Diciamo che i è il **genitore critico** di j se, senza la partecipazione di i , non esiste una catena di inviti dal seller a j . Se i è il **genitore critico** di j , allora j è il **figlio critico** di i . Ogni nodo può avere un set di padri e figli critici. L'**albero di allocazione ottimale** è un albero che ha come radice il *seller* e come nodi i **vincitori** degli oggetti (quelli con la valutazione più alta) e i loro **padri critici**. Ogni nodo nell'albero ha un peso w_i che equivale al numero totale di oggetti assegnati nella sequenza di **figli critici** di i in base all'allocazione efficiente $C_i(\Theta')$.

L'allocazione dell'algoritmo, un pò come quella di **IDM**, si basa sull'idea che se un buyer non riceve un oggetto nell'allocazione efficiente, ma i suoi figli critici ricevono oggetti, allora il buyer

può prendere un oggetto da uno dei suoi figli critici (se nell'allocazione dove i suoi *top-k* figli critici non partecipano all'asta riceve un elemento), ma non da un altro buyer che non sia suo figlio critico.

L'algoritmo inizia dai vicini del seller e controlla se questi nodi possono vincere come descritto prima. Se rispetta questa condizione allora il nodo $i \in W$ pagherà:

$$p_i = SW_{-D_i} - (SW_{-C_i^k} - v_i) \quad (1)$$

e aggiornerà i pesi dei suoi figli critici nell'albero diminuendo di 1 il loro peso. Se non rispetta quella condizione allora i passa i suoi oggetti ai suoi figli critici, e viene ricompensato di:

$$p_i = SW_{-D_i} - SW_{-C_i^k} \quad (2)$$

D_i è l'insieme dei figli critici di i compreso lui. Mentre C_i^k è l'insieme formato da i top-k figli critici di i uniti ai loro padri critici (escluso i e i suoi padri critici) uniti ai figli critici di questi ultimi.

Successivamente vengono analizzati i suoi figli.

Abbiamo sviluppato due versioni dell'algoritmo chiamate *generalize_diffusion_mechanism* e *GIDM* che portano gli stessi risultati.

GIDM è: **IC**, **IR**.

(in tutte le prove che abbiamo fatto il seller non riceve mai delle revenue negative)

3.2. VCG

Si generalizza il caso per la vendita di un elemento a più elementi applicando questa regola:

$$p_i^{vcg}(a') = W(a'_{-d_i}) - (W(a') - \pi_i^*(a')v'_i) \quad (3)$$

dove il primo termine $W(a'_{-d_i})$ è il social welfare quando non partecipa il nodo i e tutti nodi che partecipano solo grazie al suo invito. Il secondo termine $W(a') - \pi_i^*(a')v'_i$ è il social welfare dell'allocazione efficiente meno la valutazione di i se riceve l'oggetto nell'allocazione efficiente ($\pi_i(a') = 1$ se i riceve l'oggetto 0 altrimenti).

VCG è: **IC**, **IR**, **efficiente** ma non è **ND**.

3.3. MUDAN e MUDAR

L'idea è quella di esplorare il grafo a partire dal venditore s , costruendo iterativamente un insieme di *explored buyers*. A ogni iterazione, viene costruito un set di **potential winner** P e la concorrenza viene localizzata all'interno degli explored buyers: viene scelto un nodo **vincitore** w a cui viene assegnato un **tentativo di pagamento** \hat{p}_w . **MUDAN** e **MUDAR** si differenziano su come viene aggiornato il set P e come avviene il tentativo di pagamento \hat{p}_w . L'algoritmo termina quando l'insieme $P - W = \emptyset$.

3.3.1. MUDAN

MUDAN non garantisce un meccanismo di ricompensa ma nonostante ciò garantisce **IC**.

- **Potential winner set** P : Ordina i buyers in $A - W$ in base alle loro valutazioni. Se $|A - W| \leq m' \rightarrow P = A - W$ altrimenti $P = \text{top-}m'(A - W)$.
- **Tentative payment** \hat{p}_w : Quando un winner w è selezionato \hat{p}_w come la $(m' + 1)$ -esima alta valutazione in $A - W$. Decrementa m' .

MUDAN, garantisce **IC**, **IR**, **ND**, e una *forma debole* di **efficienza**.

3.3.2. MUDAR

MUDAR garantisce un meccanismo di ricompensa.

- **Potential winner set** P : $P = \text{top-}m(A)$ (Il vincitore viene scelto nell'insieme $P - W$).
- **Tentative payment** \hat{p}_w : Quando un winner w è selezionato \hat{p}_w come la $(m + 1)$ -esima alta valutazione in A .

Il meccanismo divide poi l'insieme dei vincitori W in due sottoinsiemi W_A e W_R :

- W_A contiene gli acquirenti che hanno le **top-m** valutazioni. Ogni vincitore $w \in W_A$ paga $p_w = \hat{p}_w$.
- W_R contiene i vincitori $W - W_A$. Ricevono solo una ricompensa ma nessun oggetto. La ricompensa equivale all'utilità che avrebbe ottenuto se avesse ottenuto l'oggetto, cioè $p_w = \hat{p}_w - v'_w$.

MUDAR, raggiunge l'efficienza, **IR**, **ND**, ma una *forma debole* di **IC** (è **truthful reporting** ma è **truthful bidding** solo in alcuni casi).

3.4. Risultati

Auction	Time (s)	Sw	Rw
MUDAR	0.032	128810.0	116861.0
MUDAN	0.109	126079.0	114554.0
VCG	6.181	128810.0	114463.0
GIDM	18.243	128810.0	127134.0

Table 7. k=13

Auction	Time (s)	Sw	Rw
MUDAR	0.032	208021.0	195968.0
MUDAN	0.433	204393.0	192815.0
VCG	6.205	208021.0	201576.0
GIDM	17.997	208021.0	206884.0

Table 8. k=21

Auction	Time (s)	Sw	Rw
MUDAR	0.039	343390.0	327403.0
MUDAN	0.866	332067.0	318806.0
VCG	6.169	343390.0	290581.0
GIDM	16.384	343390.0	335856.0

Table 9. k=35

Queste tabelle riportano i risultati ottenuti per una grafo non diretto di 1000 nodi (da tabella 7 a 9) per differenti valori di k in termini di tempo, social welfare e reward del seller.

Si può notare come quanto descritto precedente per quanto riguarda l'efficienza è stato rispettato. Infatti **MUDAN** non raggiunge l'ottimo social welfare, mentre **VCG** dà la ricompensa minore al seller.

Inoltre abbiamo testato per tutti i nodi se il requisito di bidding truthfulness fosse verificato e ha dimostrato che MUDAR è l'unica asta che non garantisce questo requisito.

4. Task 4

Gli algoritmi del bandito considerati sono stati i seguenti:

- **Epsilon Greedy** [5]
- **UCB1** [5]
- **Bayesian UCB** [5]
- **Thompson Sampling** [5]

I due algoritmi *Bayesian UCB* e *Thompson Sampling* utilizzano una prospettiva probabilistica ipotizzando che le ricompense dei bracci seguano una distribuzione di probabilità. All'inizio, si assegnano delle priori, che rappresentano le conoscenze iniziali sulle distribuzioni delle ricompense. Man mano che l'agente interagisce con l'environment, osservando le ricompense ottenute, le priori vengono aggiornate. L'aggiornamento delle priori avviene attraverso la combinazione delle priori con i dati osservati e in questo modo, le priori si trasformano in distribuzioni di probabilità posteriori che rappresentano le conoscenze aggiornate dopo aver osservato le ricompense effettive.

Nel nostro caso abbiamo pensato che la distribuzione di probabilità che meglio potesse modellare il nostro problema fosse una distribuzione Gaussiana (sebbene i valori non siano continui) invece che una distribuzione di Bernoulli.

Per la stima della varianza e della media è stato scelto un aggiornamento incrementale:

$$\hat{\mu}_t = \frac{\hat{\mu}_{t-1} * (n-1) + r_t}{n}, \hat{\sigma}_t^2 = \frac{(\hat{\sigma}_{t-1}^2 * (n-2) + (\hat{\mu}_t - r_t)^2)}{n-1} \quad (4)$$

come stime approssimative sull'osservazione più recente in modo da non dover rallentare troppo l'algoritmo.

4.1. Bayesian UCB

Per una variabile aleatoria B gaussiana, fissato l'intervallo di confidenza CI_α con α solitamente scelto uguale a 0.05 (cioè si vuole che il valore stimato sia all'interno di CI con una probabilità $1 - \alpha$) sappiamo che l'intervallo di confidenza è $[\hat{B} - 1.96 \cdot SE(\hat{B}), \hat{B} + 1.96 \cdot SE(\hat{B})]$ con una precisione del 95% e un rischio del 5%.

Nella versione bayesiana dell'algoritmo UCB il raggio di confidenza $rad_t(a) = \sqrt{\frac{2 \log T}{n_t(a)}}$ viene sostituito dalla stima della deviazione standard della variabile aleatoria.

4.2. Thompson Sampling

Ad ogni passo si campiona per ogni braccio a un valore di ricompensa dalla sua distribuzione P_a e si sceglie quindi il braccio con il più alto valore.

4.3. Risultati

Da questa analisi fatta su una rete generata con il modello di Watts-Strogatz sembra che l'algoritmo Epsilon greedy dia risultati migliori

Bandit	T	AVG reward	Last reward
UCB	10000	5271.527132481943	10058.730733718723
EPS	10000	5680.30108	10830.8
Bayes UCB	10000	5390.5712147567	10276.117284345255
Thompson	10000	1062.0500451782207	1417.504599185288

Table 10

5. Task 2: Mechanisms on Weighted Social Networks with Unknown Weights

5.1. Aste

Le aste che abbiamo considerato per questo task sono riportate nella [Table 1](#). In questa tabella viene riportato anche i requisiti soddisfatti da ogni asta. Un asta è:

- incentive compatible (**IC**) se per ogni *buyer* il **truthful reporting** e il **truthful bidding** sono una strategia dominante
- non-deficit (**ND**) se il ricavo del *seller* è sempre **non-negativo**

Come si può notare tutte le aste garantiscono come strategia dominante sia il **truthful reporting** che il **truthful bidding** ad eccezione di **MUDAR** (*Multi- Unit Diffusion Auction with Reward*) che garantisce solo il truthful reporting.

Il meccanismo generico di **VCG** può essere convenientemente esteso a un'asta a diffusione, tuttavia questo meccanismo non aumenta i ricavi del venditore e può portare a un deficit per il seller (come verrà analizzato più avanti).

L'asta IDM soddisfa il requisito IC ma è pensato per il caso della singola unità. Si potrebbe pensare che le aste multi-unità essendo una naturale generalizzazione della controparte a unità singola, possano essere affrontati con meccanismi simili all'IDM: **GIDM**.

Infine, a differenza delle aste tradizionali, per le *aste a diffusione* nessun meccanismo soddisfa contemporaneamente **IC**, **IR**, **ND** e **S.W.** ottimale. Infatti **MUDAN**, garantisce *IC*, *IR*, *ND*, e una *forma debole* di *efficienza*. Il secondo meccanismo, **MUDAR**, raggiunge l'*efficienza*, *IR*, *ND*, ma una *forma debole* di *IC*

5.2. Bandit Algorithm

Gli algoritmi del bandito considerati sono stati i seguenti:

- Epsilon Greedy [\[5\]](#)
- UCB1 [\[5\]](#)
- Bayesian UCB [\[5\]](#)
- Thompson Sampling [\[5\]](#)

Descritti nella sezione [4](#)

5.3. Analisi preliminari

Dati i tempi lunghi di simulazione per scegliere il tipo di asta e il tipo di algoritmo del bandito che dessero la più alta reward cumulativa, abbiamo analizzato come si comportassero le specifiche della traccia alle caratteristiche del tipo di rete assegnata.

Dalla traccia sappiamo che se il formato dell'asta soddisfa **IC** ogni nodo che accetta l'invito lo estenderà a tutti i suoi vicini che non partecipano all'asta, altrimenti inviterà ogni vicino con una probabilità di 0,15. Quindi abbiamo osservato quanti nodi partecipassero in media alla fine dell'orizzonte T .

Inoltre essendo il numero di nodi presente nella rete molto alto (20000) abbiamo pensato di poter sfruttare le misure di centralità per poter selezionare i *nodi seller*, scegliendo i 200 nodi con la misura più alta. Le misure di centralità usate in questa prima fase sono: *degree*, *PageRank*, *VoteRank*, *shapley-degree*, *shapley-threshold*, *shapley-closeness*.

Essendo che le aste utilizzate sono tutte *truthful reporting* il meccanismo di invito è lo stesso per tutti casi, negli esempi che riporteremo nelle prossime tabelle l'asta utilizzata è stata *MUDAR*, mentre l'algoritmo del bandito usato è stato *UCB1*. Nelle tabelle 2 e 3 possiamo analizzare come sia per la rete assegnata per il progetto *net_4* che una rete generata con il modello *Watts-Strogatz* (i parametri sono quelli selezionati per il task 1) abbiano un comportamento simile e mantengano questo comportamento anche per orizzonti maggiori

Armset	$T=20000$	$T=80000$
All	4.1056	4.8542
degree	5.52035	4.9855
PageRank	5.86985	5.1240
VoteRank	5.86985	5.5854
shapley-degree	5.22975	5.5535
shapley-threshold	5.6796	4.9643
shapley-closeness	4.30915	5.4464

Table 11. Media di nodi raggiunti ad ogni asta da *net_4* dopo l'orizzonte T

Armset	$T=20000$	$T=80000$
All	4.1757	4.8542
degree	5.4922	4.9855
PageRank	4.8858	5.1240
VoteRank	5.6821	5.5854
shapley-degree	5.7797	5.5535
shapley-threshold	5.6189	4.9643
shapley-closeness	4.8172	5.4464

Table 12. Media di nodi raggiunti ad ogni asta da una rete generata con il modello di Watts-Strogatz dopo l'orizzonte T

Per questo motivo il numero di elementi da vendere che analizzeremo sono $k \in [2, 5]$

5.4. Prima analisi

La prima analisi è stata fatta utilizzando tutti gli algoritmi del bandito su tutte le aste presentate precedentemente, utilizzando come armset (set di nodi) a disposizione dell'algoritmo del bandito tutti i nodi della rete. Le priori dei bracci per gli algoritmi bayesiani sono state assunte come delle gaussiane standard (media 0 e varianza 1).

Possiamo notare dai plot 1, 2, 3 e 4 che le revenue generate su entrambi le reti (**Net_4** e quella generata con il modello di **Watts-Strogatz**) seguono lo stesso andamento anche per orizzonti temporali diversi. Invece in questa prima analisi notiamo che per $T = 20000$ le aste che ottengono in diverse occasioni la revenue più alta sono **MUDAN** e **GIDM**, mentre per $T = 80000$ l'asta che ottiene la revenue più alta è quasi sempre **GIDM**. Si nota inoltre che l'asta **VCG** per valori di $k > 1$, restituisce al seller sempre una revenue negativa (per il caso $T = 80000$ non

k	bandit algorithm	<i>asta</i>	<i>cumulative revenue</i>
2	TH	GIDM	420489.0
3	TH	MUDAN	498433.0
4	TH	MUDAN	529409.0
5	TH	MUDAN	610472.0

Table 13. Best revenue per orizzonte **T=20000** su **Net_4** al variare di k

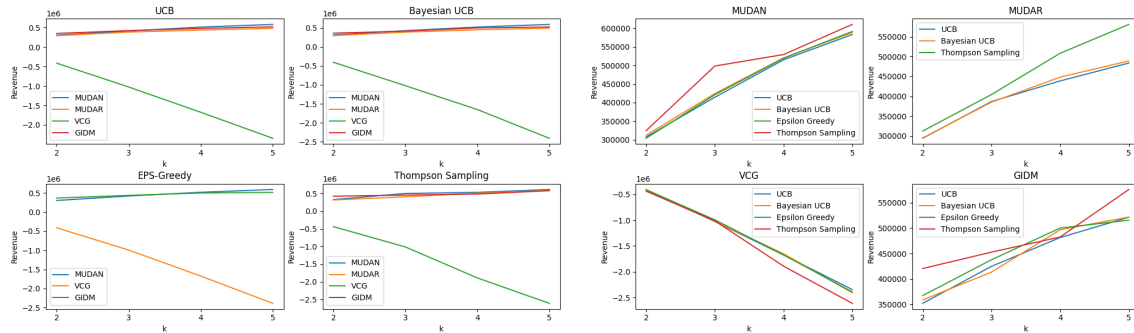
k	bandit algorithm	<i>asta</i>	<i>cumulative revenue</i>
2	TH	GIDM	382784.0
3	TH	GIDM	514874.0
4	TH	GIDM	573051.0
5	TH	MUDAN	609632.0

Table 14. Best revenue per orizzonte **T=20000** su rete generata con il modello di **Watts-Strogatz** al variare di k

k	bandit algorithm	<i>asta</i>	<i>cumulative revenue</i>
2	UCB	GIDM	1849378.0
3	UCB	GIDM	2163322.0
4	UCB	GIDM	2494297.0
5	UCB	MUDAN	2801173.0

Table 15. Best revenue per orizzonte **T=(80000)** su **Net_4** al variare di k

k	bandit algorithm	<i>asta</i>	<i>cumulative revenue</i>
2	UCB	GIDM	1947706.0
3	UCB	GIDM	2406461.0
4	UCB	GIDM	2707041.0
5	UCB	GIDM	3022707.0

Table 16. Best revenue per orizzonte **T=80000** su rete generata con il modello di **Watts-Strogatz** al variare di k**Figure 1.** Andamento delle revenue totali ottenute sulla rete **Net_4** dopo T=20000

abbiamo analizzato VCG per questioni di tempo visto che dava revenue negative). Dalle tabelle 4 e 5 possiamo notare come gli algoritmi del bandito che sono risultati migliori per diversi valori di k è **Thompson sampling** per T=20000 mentre dalle tabelle 6 e 7 notiamo che solo l'algoritmo **UCB1** risulta essere il migliore per T=80000.

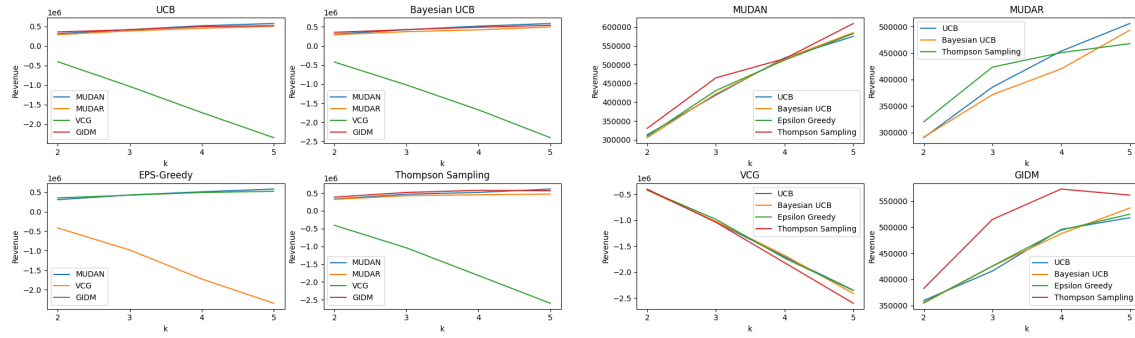


Figure 2. Andamento delle revenue totali ottenute su una **Watts-Strogatz network** dopo $T=20000$

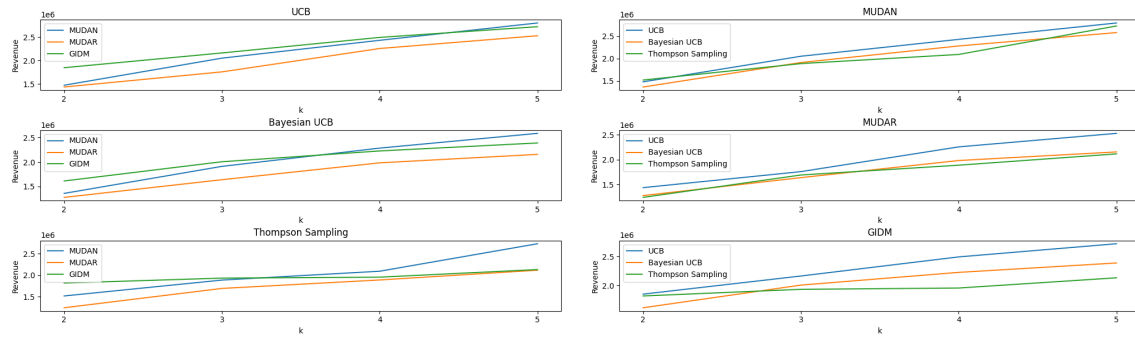


Figure 3. Andamento delle revenue totali ottenute sulla rete **Net_4** dopo $T=80000$

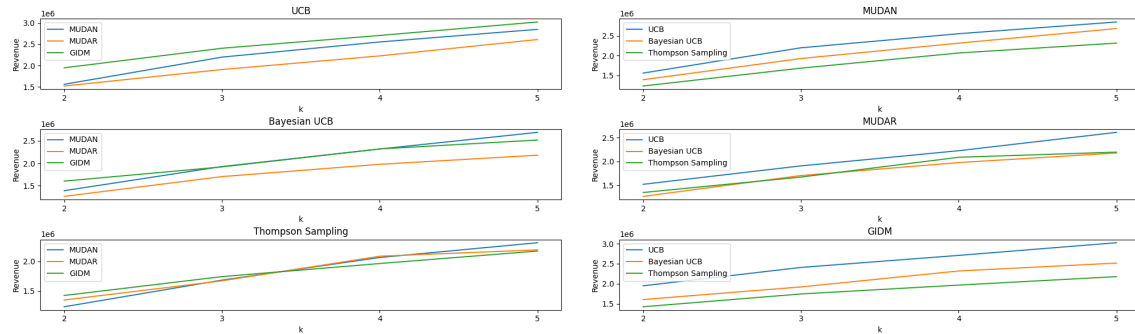


Figure 4. Andamento delle revenue totali ottenute su una **Watts-Strogatz network** dopo $T=80000$

Abbiamo fatto anche dei test dove le braccia dell'algoritmo del bandito erano le coppie (nodo, asta) ma non hanno portato ai risultati sperati.

5.5. Seconda Analisi

Come analizzato nelle tabelle 2 e 3 a seconda dell'utilizzo di diversi *armset* è possibile raggiungere più nodi. Per questo motivo abbiamo svolto una seconda analisi dove abbiamo utilizzato come *armset* i top 200 nodi con il valore di centralità (quelle elencate precedentemente) più alto insieme, ovviamente, a un *armset* contenente tutti i nodi. Nonostante nella prima analisi **MUDAN** e **GIDM** risultano essere le aste migliori, in questa analisi abbiamo voluto utilizzare anche l'asta **MUDAR** per vedere se potessero esserci dei miglioramenti rispetto alla prima analisi. Gli algoritmi del bandito analizzati invece sono tutti tranne *Epsilon Greedy*.

Inoltre visto che gli algoritmi bayesiani prevedono l'inizializzazione della priori per ogni braccio

abbiamo pensato di poter sfruttare l'informazione delle misure di centralità per dare un'importanza maggiore alle braccia con centralità più alta in questo modo:

$$\mu_a = \frac{centrality_a - \mu}{\sigma} \quad (5)$$

$$\sigma_a^2 = 2 * (1 - \frac{centrality_a}{max_centrality}) \quad (6)$$

La media di ogni braccio è stata inizializzata al suo valore della centralità standardizzato (μ e σ sono media e deviazione standard di tutti i valori di centralità). La varianza è stata inizializzata in questo modo per dare un valore minore a quei nodi con valore di centralità più alto.

Quindi per ogni armset abbiamo previsto 2 casi per gli algoritmi bayesiani:

1. $P_{arm} \sim N(0, 1)$
2. $P_{arm} \sim N(\mu_{arm}, \sigma_{arm}^2)$

Gli oggetti vendibili sono sempre $k \in [2, 5]$ e gli orizzonti scelti sono stati nuovamente **T = 20000** e **60000**.

k	bandit algorithm	<i>asta</i>	<i>cumulative revenue</i>	<i>armset</i>
2	UCB	GIDM	595698.0	shapley degree
3	UCB	MUDAN	632581.0	degree
4	UCB	MUDAN	828586.0	pagerank
5	UCB	GIDM	931791.0	pagerank

Table 17. Best revenue per orizzonte **T=20000** su **Net_4** al variare di k

k	bandit algorithm	<i>asta</i>	<i>cumulative revenue</i>	<i>armset</i>
2	UCB	GIDM	649546.0	degree
2	UCB	GIDM	795898.0	degree
3	UCB	GIDM	882022.0	voterank
4	UCB	MUDAN	954375.0	shapley closeness

Table 18. Best revenue per orizzonte **T=20000** su una rete generata con il modello di **Watts-Strogatz** al variare di k

k	bandit algorithm	<i>asta</i>	<i>cumulative revenue</i>	<i>armset</i>
2	UCB	GIDM	1894210.0	shaple threshold
3	UCB	GIDM	2335563.0	shapley degree
4	UCB	GIDM	2720157.0	shapley degree
5	UCB	GIDM	2959281.0	voterank

Table 19. Best revenue per orizzonte **T=60000** su **Net_4** al variare di k

Come possiamo notare dalle tabelle 8, 9, 10 e 11 **GIDM** e **MUDAN** continuano ad essere le aste che fanno ottenere al seller il più alto guadagno cumulativo. Inoltre **UCB** si conferma il migliore tra tutti gli algoritmi. Confrontando i risultati delle tabelle appena riportate con quelle della prima analisi (tabella 4, 5, 6 e 7) possiamo notare come l'utilizzo di un set ridotto scelto in base alle misure di centralità aumentano i ricavi per il venditore. Questo risultato si nota meglio se analizziamo i plot 5 e 6 dove le revenue generate dall'algoritmo **UCB** con l'armset compost da tutti i nodi (chiamato *normal* nei plot) sono le più basse o tra le più basse sempre.

k	bandit algorithm	asta	cumulative revenue	armset
2	UCB	GIDM	1835719.0	voterank
3	UCB	GIDM	2149290.0	voterank
4	UCB	GIDM	2315599.0	shapley degree
5	UCB	MUDAN	2555035.0	shapley degree

Table 20. Best revenue per orizzonte $T=60000$ su una rete generata con il modello di **Watts-Strogatz** al variare di k

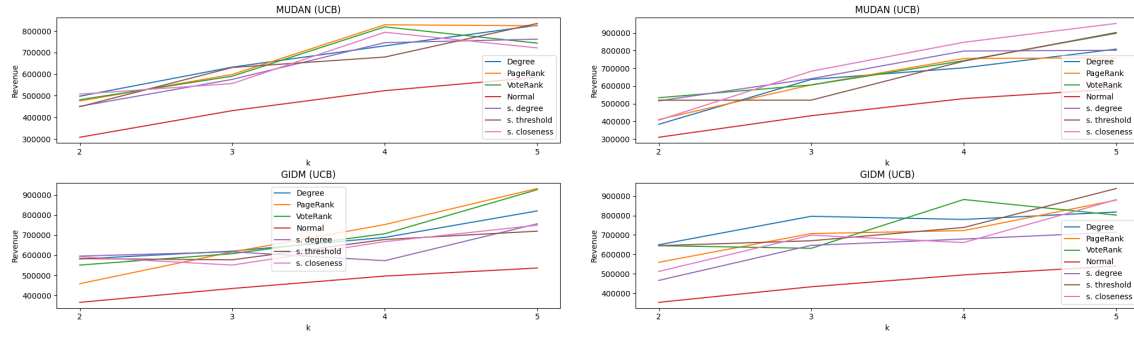


Figure 5. Andamento delle revenue totali ottenute sulla rete **Net_4** (sx) e su una rete generata con il modello di **Watts-Strogatz** (dx) dopo $T=20000$

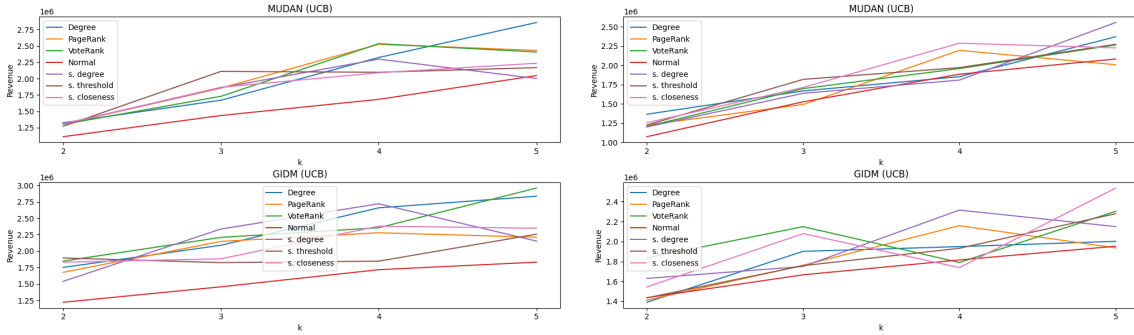


Figure 6. Andamento delle revenue totali ottenute sulla rete **Net_4** (sx) e su una rete generata con il modello di **Watts-Strogatz** (dx) dopo $T=60000$

5.6. Terza Analisi

In questa successiva analisi, abbiamo scelto quelli che si sono rivelati i 4 armset che hanno generato un guadagno maggiore: **pagerank**, **voterank**, **degree** e **shapley degree**. L'algoritmo del bandito utilizzato è **UCB**, mentre le aste utilizzate sono **MUDAN** e **MUDAR**. Gli oggetti vendibili sono sempre $k \in [1, 5]$ e gli orizzonti scelti sono stati nuovamente $T = 20000$, 40000 e 60000 .

k	bandit algorithm	auction	cumulative revenue	armset
2	UCB	GIDM	595729.0	shapley_degree
3	UCB	GIDM	698493.0	shapley_degree
4	UCB	GIDM	794244.0	pagerank
5	UCB	MUDAN	948887.0	degree

Table 21. Best revenue per orizzonte $T=20000$ su **Net_4** al variare di k

k	bandit algorithm	auction	cumulative revenue	armset
2	UCB	GIDM	664262.0	degree
3	UCB	GIDM	689492.0	pagerank
4	UCB	GIDM	916818.0	degree
5	UCB	GIDM	982112.0	voterank

Table 22. Best revenue per orizzonte $T=20000$ su una rete generata con il modello di **Watts-Strogatz** al variare di k

k	bandit algorithm	auction	cumulative revenue	armset
2	UCB	GIDM	1173350.0	shapley_degree
3	UCB	GIDM	1592862.0	pagerank
4	UCB	MUDAN	1455613.0	pagerank
5	UCB	MUDAN	1904066.0	degree

Table 23. Best revenue per orizzonte $T=40000$ su **Net_4** al variare di k

k	bandit algorithm	auction	cumulative revenue	armset
2	UCB	GIDM	1046874.0	pagerank
3	UCB	GIDM	1515597.0	shapley_degree
4	UCB	MUDAN	1653450.0	pagerank
5	UCB	MUDAN	1828475.0	voterank

Table 24. Best revenue per orizzonte $T=40000$ su una rete generata con il modello di **Watts-Strogatz** al variare di k

k	bandit algorithm	auction	cumulative revenue	armset
2	UCB	GIDM	1910936.0	degree
3	UCB	GIDM	2358232.0	degree
4	UCB	GIDM	2714478.0	degree
5	UCB	GIDM	2738174.0	pagerank

Table 25. Best revenue per orizzonte $T=60000$ su **Net_4** al variare di k

k	bandit algorithm	auction	cumulative revenue	armset
2	UCB	GIDM	1694043.0	shapley_degree
3	UCB	GIDM	1979072.0	voterank
4	UCB	GIDM	2316357.0	degree
5	UCB	MUDAN	2501488.0	shapley_degree

Table 26. Best revenue per orizzonte $T=60000$ su una rete generata con il modello di **Watts-Strogatz** al variare di k

Da queste tabelle (da 12 a 18) possiamo notare l'asta **GIDM** sia la migliore quasi sempre. Quindi scelta **GIDM** come asta, nella prossima analisi andremo ad analizzare qual è l'armset che dà migliori risultati tra: **voterank**, **degree**, **pagerank** e **shapley degree**.

5.7. Quarta analisi

.

L'obiettivo di questa ultima analisi è scegliere l'armset che genera la revenue migliore per il

seller tra: **pagerank**, **voterank**, **degree** e **shapley degree**. L'algoritmo del bandito utilizzato è **UCB**, mentre l'asta utilizzata è **GIDM**. Gli oggetti vendibili sono sempre $k \in [2, 5]$ e gli orizzonti scelti sono stati nuovamente **T = 20000, 40000, 60000 e 80000**.

k	bandit algorithm	cumulative revenue	armset
2	UCB	592932.0	voterank
3	UCB	645579.0	pagerank
4	UCB	753338.0	pagerank
5	UCB	809378.0	voterank

Table 27. Best revenue per orizzonte **T=20000** su **Net_4** al variare di k

k	bandit algorithm	cumulative revenue	armset
2	UCB	492087.0	degree
3	UCB	623716.0	degree
4	UCB	692155.0	voterank
5	UCB	804442.0	pagerank

Table 28. Best revenue per orizzonte **T=20000** su una rete generata con il modello di **Watts-Strogatz** al variare di k

k	bandit algorithm	cumulative revenue	armset
2	UCB	622387.0	pagerank
3	UCB	724715.0	shapley degree
4	UCB	790166.0	voterank
5	UCB	884658.0	degree

Table 29. Best revenue per orizzonte **T=20000** su una rete generata con il modello di **Watts-Strogatz** al variare di k

k	bandit algorithm	cumulative revenue	armset
2	UCB	1160987.0	degree
3	UCB	1386619.0	pagerank
4	UCB	1522650.0	pagerank
5	UCB	1913670.0	degree

Table 30. Best revenue per orizzonte **T=40000** su **Net_4** al variare di k

k	bandit algorithm	cumulative revenue	armset
2	UCB	1041324.0	degree
3	UCB	1458971.0	shapley degree
4	UCB	1690165.0	pagerank
5	UCB	1791251.0	degree

Table 31. Best revenue per orizzonte **T=40000** su una rete generata con il modello di **Watts-Strogatz** al variare di k

Dalle tabelle e dai plot possiamo notare come l'armset **shapley degree** dia i risultati peggiori, mentre l'armset **voterank** e **pagerank** sembrano i due che meglio si adattano al problema assegnato.

k	bandit algorithm	cumulative revenue	armset
2	UCB	1139607.0	voterank
3	UCB	1289905.0	voterank
4	UCB	1698424.0	pagerank
5	UCB	1851939.0	pagerank

Table 32. Best revenue per orizzonte $T=40000$ su una rete generata con il modello di **Watts-Strogatz** al variare di k

k	bandit algorithm	cumulative revenue	armset
2	UCB	1619493.0	shapley degree
3	UCB	2057332.0	voterank
4	UCB	2550961.0	voterank
5	UCB	2399741.0	shapley degree

Table 33. Best revenue per orizzonte $T=60000$ su **Net_4** al variare di k

k	bandit algorithm	cumulative revenue	armset
2	UCB	1589927.0	degree
3	UCB	1858882.0	voterank
4	UCB	2337390.0	voterank
5	UCB	2414294.0	voterank

Table 34. Best revenue per orizzonte $T=60000$ su una rete generata con il modello di **Watts-Strogatz** al variare di k

k	bandit algorithm	cumulative revenue	armset
2	UCB	2525384.0	degree
3	UCB	3096044.0	voterank
4	UCB	3533555.0	degree
5	UCB	3317956.0	voterank

Table 35. Best revenue per orizzonte $T=80000$ su **Net_4** al variare di k

k	bandit algorithm	cumulative revenue	armset
2	UCB	2236621.0	voterank
3	UCB	2859960.0	pagerank
4	UCB	3276390.0	degree
5	UCB	3043272.0	pagerank

Table 36. Best revenue per orizzonte $T=80000$ su una rete generata con il modello di **Watts-Strogatz** al variare di k

Per questo motivo la scelta che abbiamo utilizzato per il problema numero 2 è:

- Algoritmo del bandito: **UCB**
- Asta: **GIDM**
- Armset per l'algoritmo del bandito: top 200 nodi con **pagerank** più alto.

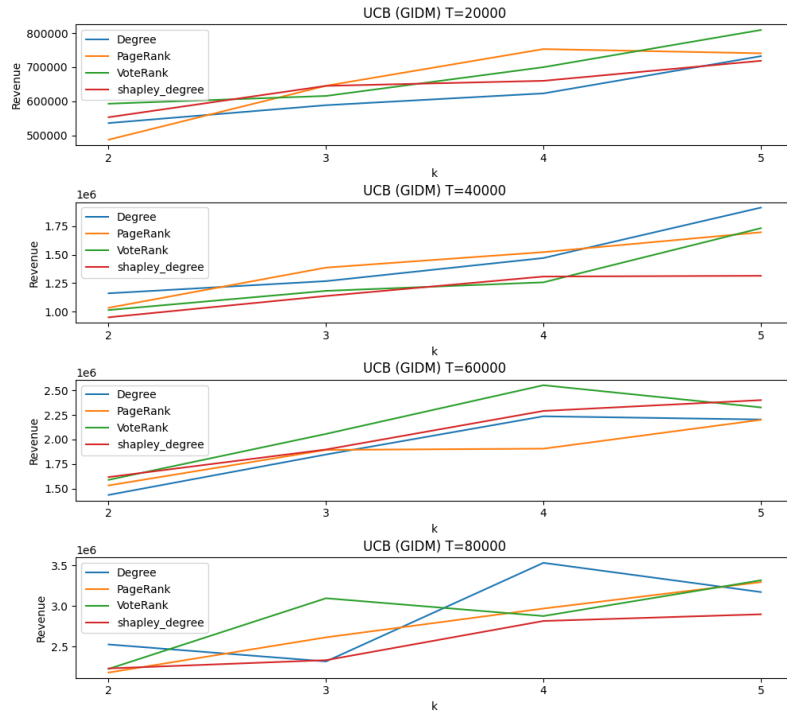


Figure 7. Andamento delle revenue totali ottenute sulla rete **Net₄**

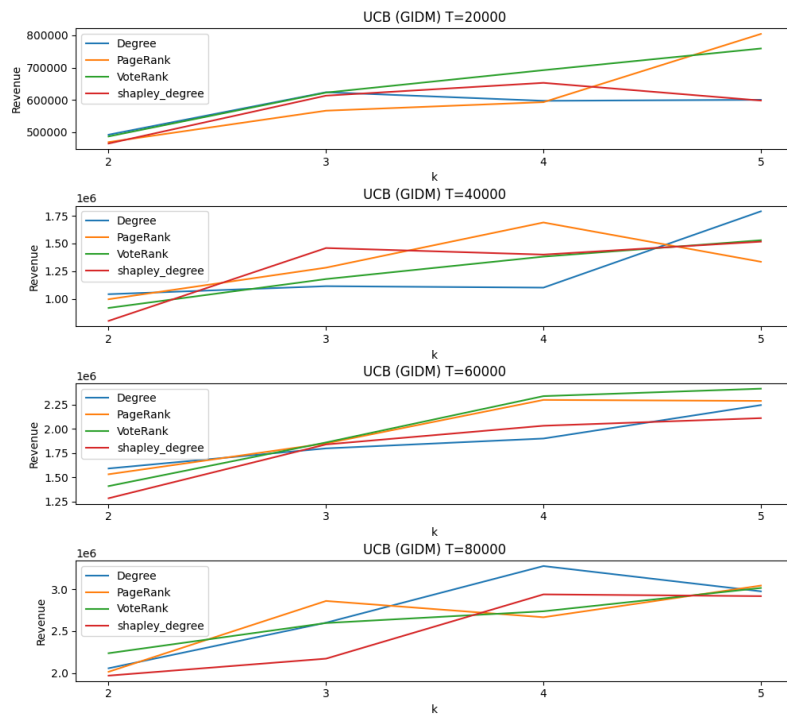


Figure 8. Andamento delle revenue totali ottenute su una rete generata con il modello di **Watts-Strogatz**

References

- [1] Li, Bin, et al. "Mechanism design in social networks." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 31. No. 1. 2017.

- [3] Fang, Yuan, et al. "Multi-unit Auction over a Social Network." arXiv preprint arXiv:2302.08924 (2023) from a single image without facial landmarks. *Int. J. Comput. Vis.* 126(2), 144–157 (2018)
- [4] Zhao, Dengji, et al. "Selling multiple items via social networks." arXiv preprint arXiv:1903.02703 (2019).
- [5] Slivkins, Aleksanders. "Introduction to multi-armed bandits." *Foundations and Trends® in Machine Learning* 12.1-2 (2019): 1-286.
- [6] Shaikh Arifuzzaman, Maleq Khan, and Madhav Marathe. 2019. Fast Parallel Algorithms for Counting and Listing Triangles in Big Graphs. *ACM Trans. Knowl. Discov. Data* 14, 1, Article 5 (February 2020), 34 pages. <https://doi.org/10.1145/3365676>