ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής

Εργασία Μαθήματος:

Ευφυής Αλληλεπίδραση με Κοινωνικά Δίκτυα

Αριθμός εργασίας	***Απαλλακτική Εργασία για το Εαρινό Εξάμηνο του Ακαδημαϊκού Έτους 2017-2018***
Όνομα φοιτητή	Σπύρου Άρης – Φλόκας Ζήσης Δελλής Παρασκευάς – Διατσίγκος Γιάννης
Αριθμός Μητρώου	Π15132 – Π15148 – Π15034 – Π15037
Ημερομηνία παράδοσης	2/7/18

Περιεχόμενα

1. Εκφώνηση

Σκοπός της συγκεκριμένης εργασίας είναι η προγραμματιστική διαχείριση του χρονικά μεταβαλλόμενου δικτύου **Stack Overflow Temporal Network**, η περιγραφή του οποίου αλλά και τα δεδομένα του βρίσκονται στην δικτυακή τοποθεσία https://snap.stanford.edu/data/sx-stackoverflow.html . Κάθε ακμή του εν λόγω δικτύου είναι συσχετισμένη με μία χρονοσφραγίδα (timestamp) η οποία αντιστοιχεί στην χρονική στιγμή κατά την οποία δημιουργήθηκε. Το σύνολο των κατευθυνόμενων ακμών του δικτύου με τις αντίστοιχες χρονοσφραγίδες είναι αποθηκευμένο στο αρχείο **sx-stackoverflow.txt** υπό την μορφή διαδοχικών τριάδων **(source_id, target_id,timestamp)**, όπου το **source_id** είναι το αναγνωριστικό του κόμβου προέλευσης της ακμής, το **target_id** είναι το αναγνωριστικό του κόμβου κατάληξης της ακμής ενώ το **timestamp** υποδηλώνει την χρονική στιγμή δημιουργίας της ακμής.

Συνεπώς, τα διαθέσιμα δεδομένα για το υπό εξέταση δίκτυο μπορούν να αποτυπωθούν ως χρονικά συσχετισμένες ακμές της μορφής:

$$e_{ij}(t) = \langle v_i, v_j, t \rangle \ (1) \text{ για } \ t_{min} \leq t \leq t_{max}$$

όπου t_{min} είναι η παλιότερη χρονική παρατήρηση που υπάρχει μέσα στο σύνολο των διαθέσιμων δεδομένων και t_{max} η πιο πρόσφατη χρονική παρατήρηση. Το συγκεκριμένο χρονικό διάστημα $T = [t_{min}, t_{max}]$ διαμερίζεται σε ένα σύνολο (N) μη-επικαλυπτόμενων χρονικών περιόδων $\{T_1, T_2, \cdots, T_j, \dots, T_N\}$ ίσης χρονικής διάρκειας (δt) θεωρώντας ένα σύνολο $(N+1)$ χρονικών στιγμών $\{t_0, t_1, t_2, \cdots, t_{j-1}, t_j \dots, t_{N-1}, t_N\}$ τέτοιων ώστε:

$$t_j = t_{min} + j * \delta t \ (2) \text{ για } 0 \leq j \leq N$$

όπου $\Delta T = t_{max} - t_{min}$ (3) και $\delta t = \frac{\Delta T}{N}$ (4). Σύμφωνα με τις παραπάνω διευκρινήσεις η j-οστή χρονική περίοδος μπορεί να οριστεί σύμφωνα με την παρακάτω σχέση:

$$T_j = \begin{cases} [t_{j-1}, t_j), 1 \leq j \leq N-1; \\ [t_{j-1}, t_j], j = N. \end{cases} (5)$$

Για κάθε μία από τις χρονικές περιόδους T_j για $1 \leq j \leq N$ μπορούμε να θεωρήσουμε το αντίστοιχο υπο-γράφημα του συνολικού δικτύου $G[t_{j-1}, t_j] = (V[t_{j-1}, t_j], E[t_{j-1}, t_j])$ όπου $V[t_{j-1}, t_j]$ είναι το σύνολο των κορυφών που εμφανίζονται στα άκρα των ακμών του δικτύου κατά την χρονική περίοδο T_j. Το σύνολο των ακμών του δικτύου που δημιουργούνται την συγκεκριμένη χρονική περίοδο είναι το σύνολο $E[t_{j-1}, t_j]$.

Η αυστηρότερη περιγραφή της χρονικής εξέλιξης του εξεταζόμενου δικτύου μέσα στο πλαίσιο του προβλήματος της πρόγνωσης ακμών επιβάλλει την διατύπωση μερικών συμπληρωματικών σχέσεων. Συγκεκριμένα, κατά την μετάβαση του δικτύου από την χρονική περίοδο T_j στην χρονική περίοδο T_{j+1} μας ενδιαφέρει το σύνολο των κορυφών που παραμένει κοινό μεταξύ των χρονικών διαστημάτων $[t_{j-1}, t_j]$ και $[t_j, t_{j+1}]$, το οποίο θα υποδηλώνεται ώς το σύνολο $V^*[t_{j-1}, t_{j+1}]$ που θα δίνεται από την σχέση:

$$V^*[t_{j-1}, t_{j+1}] = V\,[t_{j-1}, t_j] \cap V[t_j, t_{j+1}] (6) \text{ για } 1 \leq j \leq N-1.$$

Αντίστοιχα, μας ενδιαφέρει ο περιορισμός των συνόλων $E[t_{j-1}, t_j]$ και $E[t_j, t_{j+1}]$ σε εκείνα τα υποσύνολα των ακμών που οι κορυφές τους ανήκουν αυστηρά στο σύνολο $V^*[t_{j-1}, t_{j+1}]$. Αυτά τα περιορισμένα σύνολα ακμών θα υποδηλώνονται ως το σύνολο $E^*[t_j, t_{j+1}]$ και θα δίνονται από τις σχέσεις:

$$E^*[t_{j-1}, t_j] = \{(u,v) \in E[t_{j-1}, t_j] : u \in V^*[t_{j-1}, t_{j+1}] \; και \; v \in V^*[t_{j-1}, t_{j+1}]\} \; (7)$$

$$E^*[t_j, t_{j+1}] = \{(u,v) \in E[t_j, t_{j+1}] : u \in V^*[t_{j-1}, t_{j+1}] \; και \; v \in V^*[t_{j-1}, t_{j+1}]\} \; (8)$$

Η προγραμματιστική διαχείριση του προαναφερθέντος χρονικά μεταβαλλόμενου δικτύου συνίστανται στην συγγραφή κώδικα είτε στο προγραμματιστικό περιβάλλον του **MatLab** είτε της **Python** προκειμένου να υλοποιηθούν οι ακόλουθες διαδικασίες:

1. Υπολογισμός των χρονικών στιγμών t_{min} και t_{max}.
2. Διαμέριση του συνολικού χρονικού διαστήματος $T = [t_{min}, t_{max}]$ στα υποδιαστήματα $\{T_1, T_2, \cdots, T_j, \dots, T_N\}$ και υπολογισμός των αντίστοιχων χρονικών στιγμών $\{t_0, t_1, t_2, \cdots, t_{j-1}, t_j \dots, t_{N-1}, t_N\}$ συναρτήσει της παραμέτρου (N). Η παράμετρος (N) θα μπορεί να μεταβληθεί από τον χρήστη του προγράμματος πριν από την εκτέλεσή του.
3. Προγραμματιστική αποτύπωση (είτε μέσω της μήτρας γειτνίασης είτε μέσω κάποιου εγγενούς για το εργαλείο που θα χρησιμοποιήσετε τρόπου, π.χ. ενός αντικειμένου Graph του module NetworkX της Python) του συνόλου των υποδικτύων $G[t_{j-1}, t_j]$ για $1 \le j \le N$.
4. Για κάθε ένα από τα υποδίκτυα $G[t_{j-1}, t_j]$ για $1 \le j \le N$ να υπολογίσετε και να παρουσιάσετε γραφικά την κατανομή των τιμών των παρακάτω μέτρων κεντρικότητας:
   i. Degree Centrality
   ii. In-Degree Centrality
   iii. Out-Degree Centrality
   iv. Closeness Centrality
   v. Betweenness Centrality
   vi. Eigenvector Centrality
   vii. Katz Centrality
5. Για κάθε ζεύγος διαδοχικών υποδικτύων $(G[t_{j-1}, t_j], G[t_j, t_{j+1}])$ για $1 \le j \le N-1$να υπολογιστούν τα σύνολα $V^*[t_{j-1}, t_{j+1}]$, $E^*[t_{j-1}, t_j]$ και $E^*[t_j, t_{j+1}]$.
6. Για κάθε ζεύγος κόμβων $(u,v) \in V^*[t_{j-1}, t_{j+1}]$ και κάθε σύνολο $V^*[t_{j-1}, t_{j+1}]$ με $1 \le j \le N-1$να υπολογιστούν οι παρακάτω πίνακες ομοιότητας:
   i. $S_{GD} = [S_{GD}(u,v)] = -Length\ of\ Shortest\ Path\ Between\ u\ and\ v$ [**Graph Distance**]
   ii. $S_{CN} = [S_{CN}(u,v)] = |\Gamma(u) \cap \Gamma(v)|$ [**Common Neighbors**] όπου $\Gamma(u)$ το σύνολο των γειτόνων του κόμβου u.
   iii. $S_{JC} = [S_{JC}(u,v)] = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}$ [**Jaccard's Coefficient**]
   iv. $S_A = [S_A(u,v)] = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log(|\Gamma(z)|)}$ **[Adamic / Adar]**
   v. $S_{PA} = [S_{PA}(u,v)] = |\Gamma(u)| * |\Gamma(v)|$ **[Preferential Attachment]**

Προσοχή!!! Οι παραπάνω πίνακες ομοιότητας θα υπολογίζονται για το σύνολο των κόμβων που είναι κοινοί για δύο διαδοχικά υποδίκτυα, δηλαδή το σύνολο $V^*[t_{j-1}, t_{j+1}]$, αλλά πάνω στην βάση του συνόλου των ακμών $E^*[t_{j-1}, t_j]$. Πρόκειται για τις ακμές της προηγούμενης χρονικής περιόδου που σχηματίζονται όμως μεταξύ κορυφών που ανήκουν στο κοινό σύνολο των κόμβων $V^*[t_{j-1}, t_{j+1}]$.

7. Για κάθε έναν από τους πίνακες ομοιότητας S_{GD}, S_{CN}, S_{JC}, S_Aκαι S_{PA} που υπολογίστηκαν στο προηγούμενο ερώτημα, για κάθε ένα από τα σύνολα κόμβων $V^*[t_{j-1}, t_{j+1}]$, να εξαχθούν οι

κορυφαίες $p_{GD}\%$, $p_{CN}\%$,$p_{JC}\%$,$p_A\%$ και $p_{PA}\%$ (μεγαλύτερες) τιμές ομοιότητας και τα ζεύγη των κορυφών στις οποίες αντιστοιχούν. Το ποσοστό αυτών των ζευγαριών των κορυφών που ανήκουν πράγματι στο σύνολο $E^*[t_j, t_{j+1}]$ υποδηλώνει το ποσοστό επιτυχίας στην πρόγνωση μελλοντικών ακμών της κάθε μετρικής. Να υπολογιστούν τα ποσοστό ορθής πρόγνωσης για κάθε μέτρο ομοιότητας για κάθε σύνολο $V^*[t_{j-1}, t_{j+1}]$ (δηλαδή για κάθε ζεύγος διαδοχικών υποδικτύων). Οι τιμές των παραμέτρων $p_{GD}\%$, $p_{CN}\%$,$p_{JC}\%$,$p_A\%$ και $p_{PA}\%$ θα δίνονται από τον χρήστη του προγράμματος πριν από την εκτέλεση του προγράμματος.

Σημείωση: Μαζί με το κώδικα του προγράμματος θα πρέπει να παραδοθεί αναλυτική τεχνική τεκμηρίωση που να περιγράφει την λογική που ακολουθήσατε ,τις σχεδιαστικές αποφάσεις που λάβατε ή τις συμβάσεις που κάνατε προκειμένου να ολοκληρωθεί η υλοποίηση. Στο παραδοτέο κείμενο θα πρέπει να εμφανίζονται αναλυτικά αποτελέσματα από την εκτέλεση του προγράμματός σας για διάφορες τιμές παραμέτρων που ελέγχει ο χρήστης. Η εργασία μπορεί να εκπονηθεί σε ομάδες των 3 ατόμων το πολύ.

2. Περιγραφή προγράμματος

Για την εκπόνηση του προγράμματος χρησιμοποιήθηκαν οι βιβλιοθήκες NetworkX[1], matplotlib[2] και NumPy[3].

Επίσης κατά το διάβασμα του αρχείου χρησιμοποιείται η τιμή μιας μεταβλητής (N_RL) που καθορίζει το πόσες γραμμές θα διαβαστούν καθώς το διάβασμα ολόκληρου του αρχείου αποδείχθηκε πολύ χρονοβόρο.

2.1. Κύρια ροή

Αρχικά καλείται η συνάρτηση min_max η οποία υπολογίζει το μέγιστο και το ελάχιστο timestamp. Στη συνέχεια το πρόγραμμα χωρίζει το συνολικό χρονικό διάστημα σε υποδιαστήματα συναρτήσει της παραμέτρου *N* που ορίζεται στην αρχή του προγράμματος. Τέλος για κάθε υποδιάστημα σχεδιάζεται το γράφημα, το ιστόγραμμα της κεντρικότητας που επιλέγεται, υπολογίζονται τα σύνολα του 5ου ερωτήματος, υπολογίζονται οι πίνακες ομοιότητας και επίσης τα ποσοστά ορθής πρόγνωσης του κάθε πίνακα ομοιότητας.

2.2. Συνάρτηση min_max

Σε κάθε περίπτωση η συνάρτηση διαβάζει ένα αρχείο με όνομα “min_maxN.txt” όπου N η παράμετρος του προγράμματος που ορίζει πόσες γραμμές διαβάζονται (π.χ. min_max10000.txt). Αν βρεθεί το αρχείο τότε οι τιμές διαβάζονται από αυτό εναλλακτικά βρίσκει τις τιμές εκ νέου. Για να βρει τις τιμές αρχικά ανοίγει το αρχείο και διαβάζει ανά γραμμή. Κάθε timestamp συγκρίνεται με την μέγιστη και την ελάχιστη τιμή και αυτές ανανεώνονται. Όταν τελειώσει η αναζήτηση τυπώνει το αποτέλεσμα σε αρχείο και επιστρέφει το μέγιστο και το ελάχιστο.

[1] networkx.github.io
[2] matplotlib.org
[3] www.numpy.org

2.3. Συνάρτηση graph

Αυτή η συνάρτηση καλείται από την κύρια ροή του προγράμματος και είναι υπεύθυνη για την δημιουργία και τον σχεδιασμό ενός γραφήματος. Παίρνει ως όρισμα δύο μεταβλητές: την *t* που ορίζει το υποσύνολο του χρόνου για το οποίο δημιουργείται το γράφημα και την *centrality* που ορίζει ποιο μέτρο κεντρικότητας θα χρησιμοποιηθεί για την δημιουργία του ιστογράμματος βαθμών στη συνέχεια.

Δημιουργεί αρχικά ένα κενό κατευθυνόμενο γράφημα και διαβάζει το αρχείο ανά γραμμή. Αν το timestamp ανήκει στο υποσύνολο του χρόνου για το οποίο έχει κληθεί η συνάρτηση και αν το source_id και το target_id είναι διαφορετικά (δηλαδή η γραμμή δεν περιγράφει ένα self-loop) τότε προστίθεται μία ακμή στο γράφημα από τον κόμβο source_id στο κόμβο target_id. Όταν τελειώσει το διάβασμα τότε γίνεται γραφική αναπαράσταση του γραφήματος και η συνάρτηση καλεί την συνάρτηση centrality_histogram.

2.4. Συνάρτηση centrality_histogram

Η συνάρτηση καλείται από την συνάρτηση graph και είναι υπεύθυνη για τον υπολογισμό και την γραφική απεικόνιση σε ιστόγραμμα των μέτρων κεντρικότητας των γραφημάτων. Παίρνει ως όρισμα δύο μεταβλητές: την g που είναι ένα αντικείμενο γραφήματος της βιβλιοθήκης NetworkX και την *c* που ορίζει το μέτρο κεντρικότητας που θα χρησιμοποιηθεί. Η τελευταία μπορεί να πάρει τις τιμές ‘degree’, ‘in_degree’, ‘out_degree’, closeness’, ‘betweenness’, ‘eigenvector’, ‘katz’.

Βρίσκει τους βαθμούς και των αριθμό εμφανίσεων κάθε βαθμού και στη συνέχεια τους απεικονίζει γραφικά σε ένα ιστόγραμμα.

2.5. Συνάρτηση graph_star

Η συνάρτηση καλείται από την κύρια ροή του προγράμματος και υπολογίζει τα σύνολα $V^*[t_{j-1}, t_{j+1}]$, $E^*[t_{j-1}, t_j]$ και $E^*[t_j, t_{j+1}]$. Παίρνει ως όρισμα την μεταβλητή *t* που ορίζει το υποσύνολο του χρόνου για το οποίο καλείται η συνάρτηση.

Η συνάρτηση αρχικά δημιουργεί δύο κατευθυνόμενα γραφήματα που αντιστοιχούν στα υποδίκτυα ($G[t_{j-1}, t_j], G[t_j, t_{j+1}]$). Στη συνέχεια συγκρίνεται κάθε κόμβος του πρώτου γραφήματος με του δεύτερου και στην περίπτωση που είναι ίδιοι τότε προστίθεται στο σύνολο V*. Επίσης για κάθε ακμή σε καθένα από τα υποδίκτυα ($G[t_{j-1}, t_j], G[t_j, t_{j+1}]$) ελέγχεται αν και οι δύο της κόμβοι βρίσκονται στο σύνολο V*. Σε αυτή την περίπτωση η ακμή προστίθεται στο σύνολο $E^*[t_{j-1}, t_j]$ ή $E^*[t_j, t_{j+1}]$ ανάλογα με το από πού προήλθε η ακμή και τέλος τυπώνονται νέα τρία σύνολα και καλείται η συνάρτηση similarity_matrices.

2.6. Συνάρτηση similarity_matrices

Η συνάρτηση καλείται από την συνάρτηση graph_star, υπολογίζει τους πίνακες ομοιότητας, τα ποσοστά επιτυχίας πρόβλεψης και παίρνει ως όρισμα μία μεταβλητή *edges* η οποία αντιστοιχεί στο σύνολο των ακμών $E^*[t_{j-1}, t_j]$ που υπολογίζεται από την συνάρτηση graph_star και μία μεταβλητή *nodes* η οποία αντιστοιχεί στο σύνολο κόμβων $V^*[t_{j-1}, t_{j+1}]$.

Αρχικά ανακατασκευάζει ένα κατευθυνόμενο γράφημα με ακμές αυτές του ορίσματος και επίσης δημιουργεί το αντίστοιχο μη-κατευθυνόμενο γράφημα που χρειάζεται για τον υπολογισμό των περισσότερων πινάκων ομοιότητας. Για κάθε ζεύγος κόμβων $(u, v) \in$

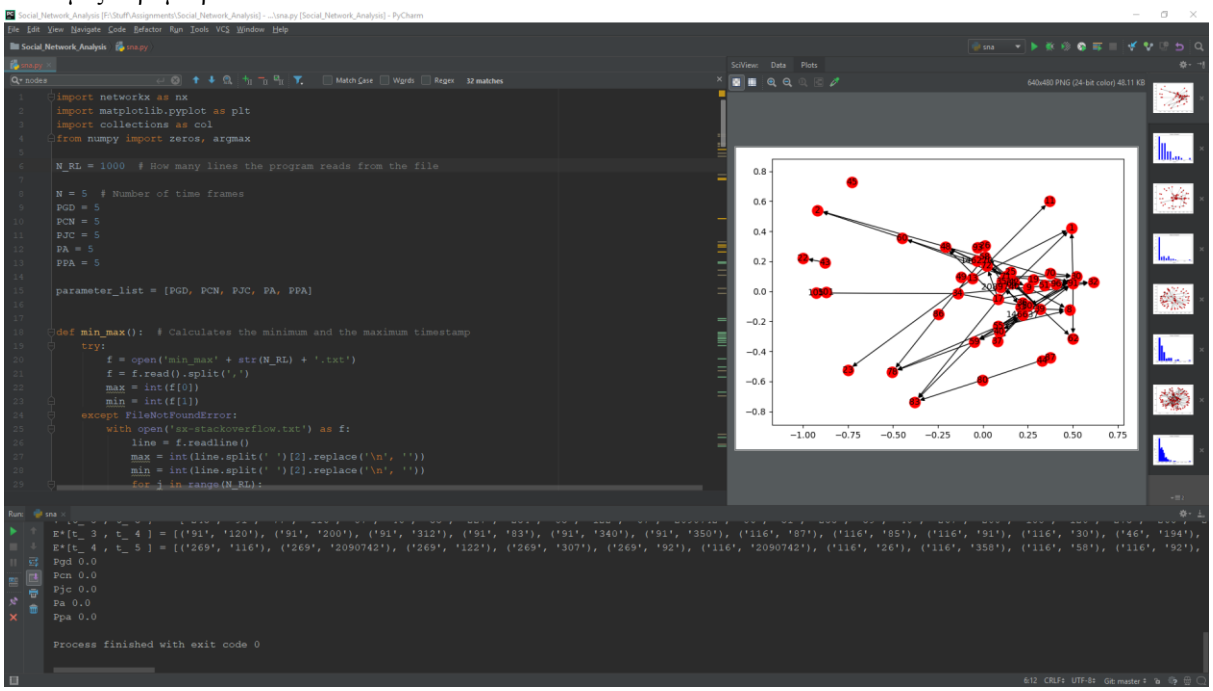
$V^*[t_{j-1}, t_{j+1}]$ υπολογίζει την τιμή της εκάστοτε μετρικής ομοιότητας και την εκχωρεί σε κατάλληλο πίνακα. Τέλος βρίσκει τις n μεγαλύτερες τιμές των πινάκων, (όπου n μία παράμετρος που ορίζεται στην αρχή του προγράμματος και είναι διαφορετική για κάθε μετρική ομοιότητας) σε ποιους κόμβους αντιστοιχούν και υπολογίζει τον λόγο: αριθμός ακμών που αντιστοιχούν στις μεγαλύτερες τιμές και ανήκουν στο σύνολο $E^*[t_{j-1}, t_j]$ προς συνολικός αριθμός ακμών που αντιστοιχούν στις μεγαλύτερες τιμές.

3. Παραδείγματα εκτέλεσης

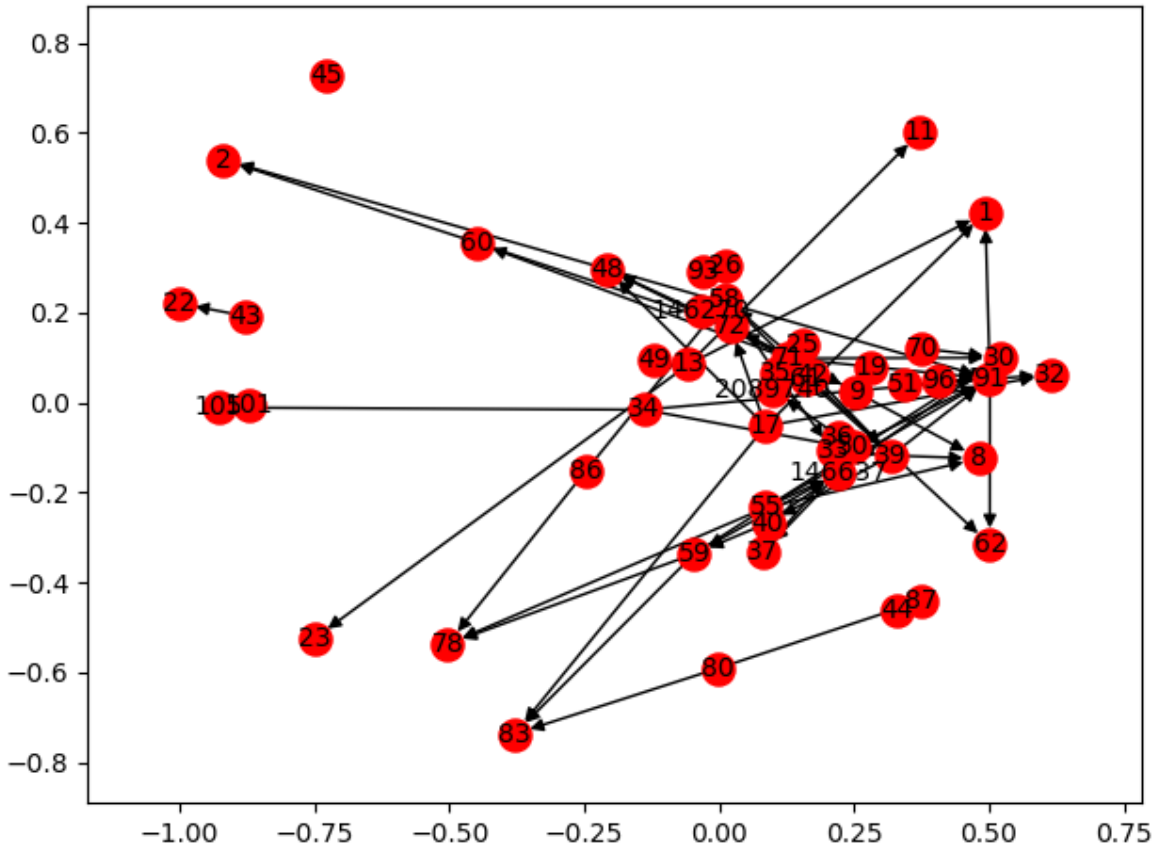
```
N_RL = 1000  # How many lines the program reads from the file

N = 5  # Number of time frames
PGD = 5
PCN = 5
PJC = 5
PA = 5
PPA = 5
```

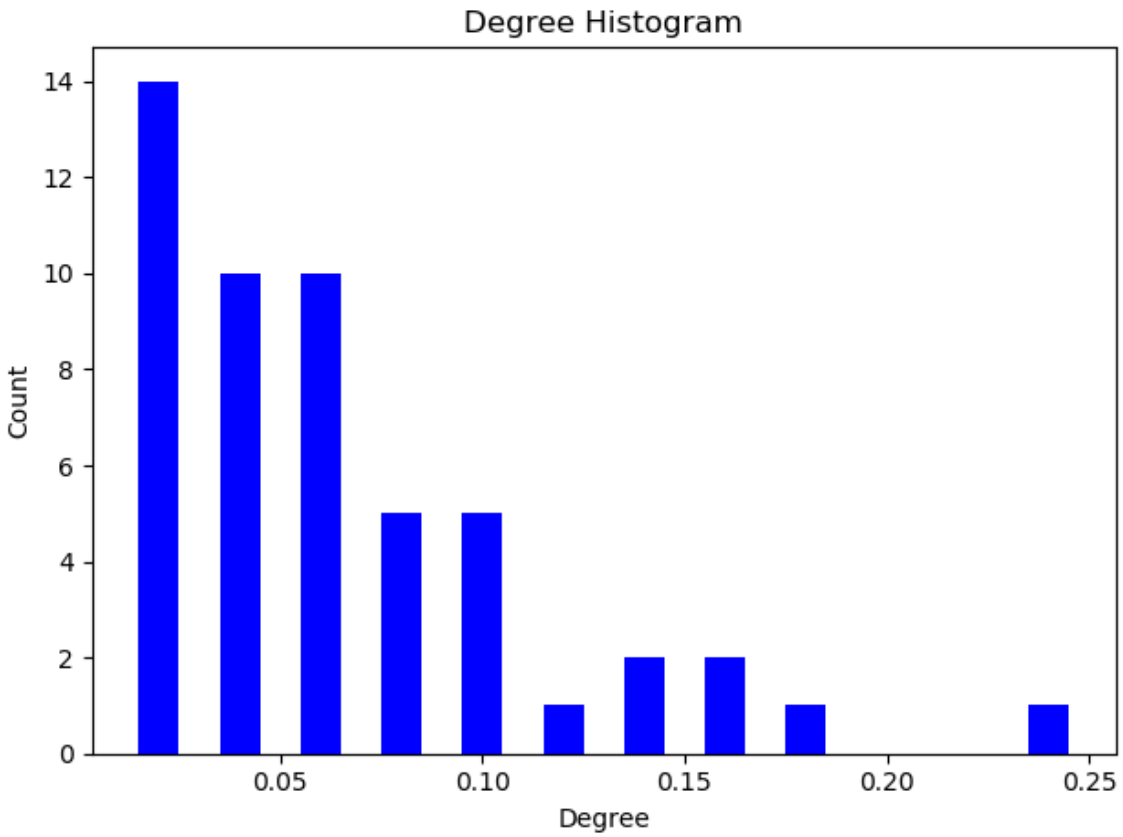
3.1 Τιμές παραμέτρων



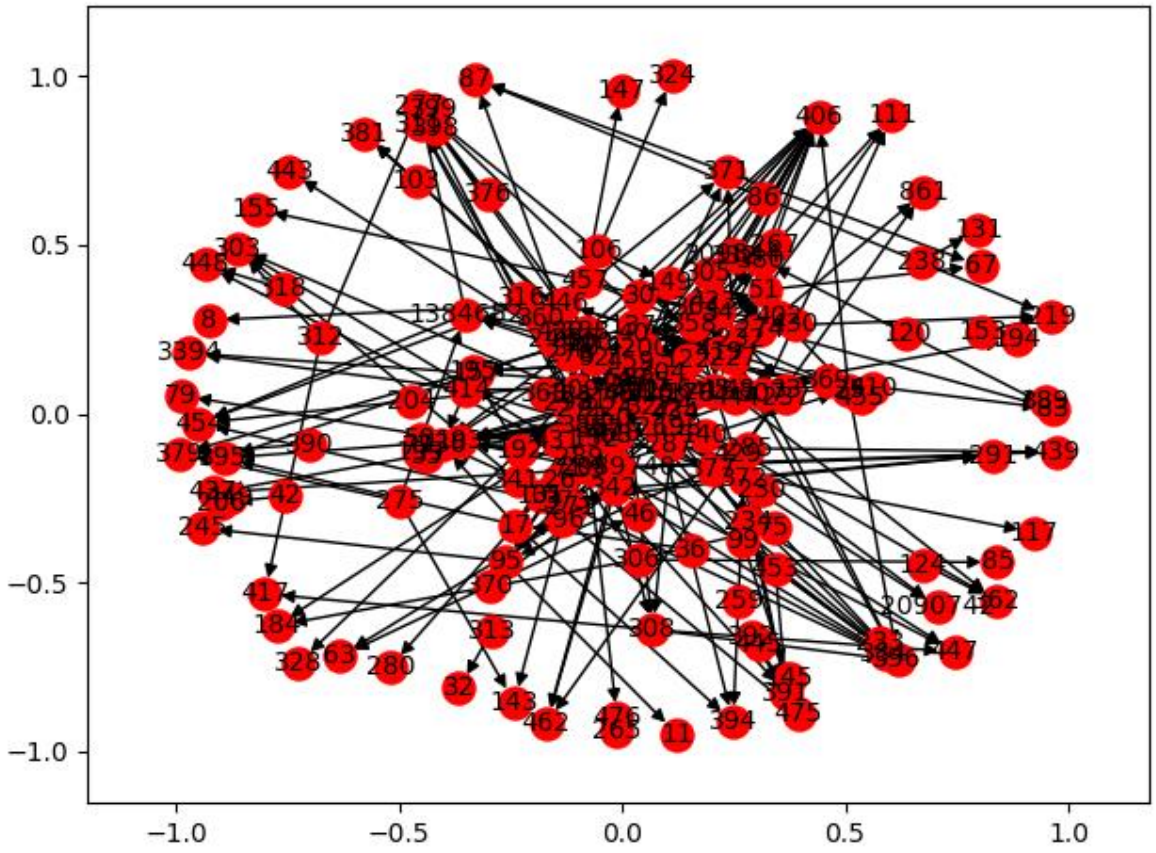
3.2 Συνολική εικόνα παραθύρου



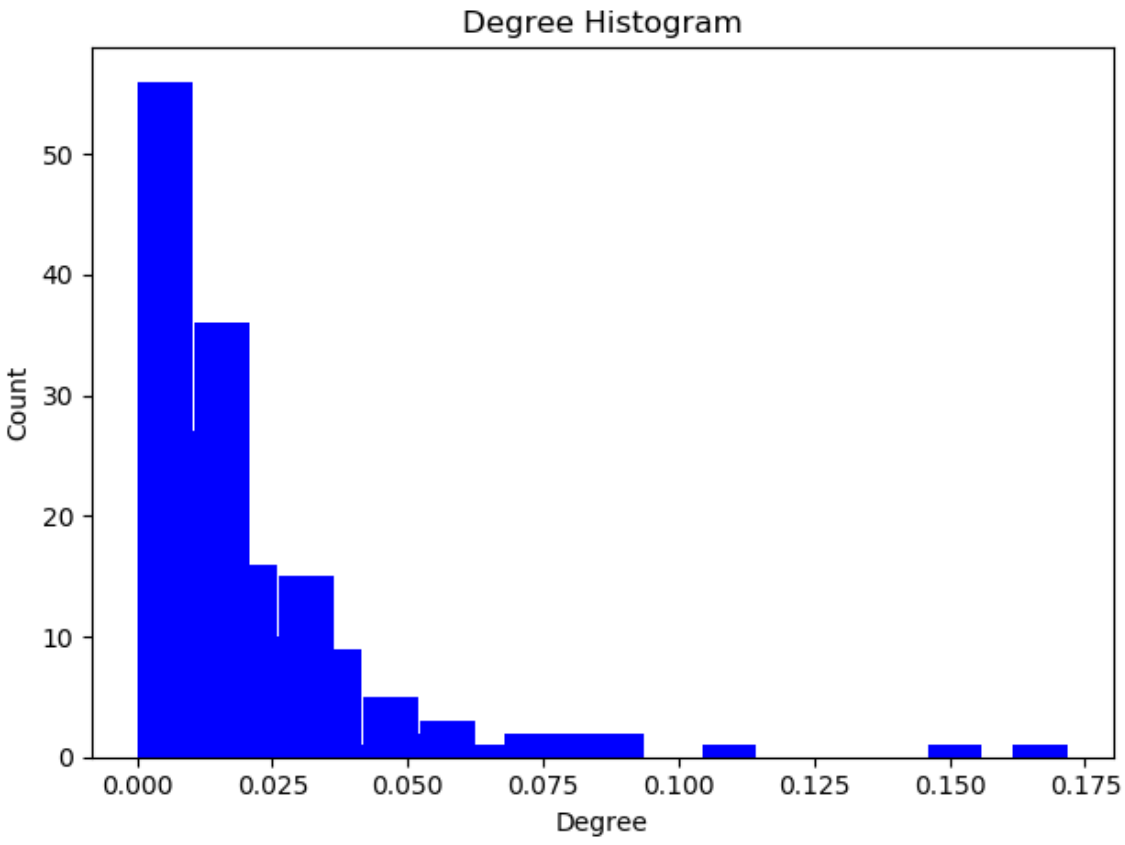
3.3 Γράφημα για t = 0



3.4 Αντίστοιχο ιστόγραμμα για degree centrality



3.5 Γράφημα για t = 4



3.6 Αντίστοιχο ιστόγραμμα για degree centrality