



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Σχεδιασμός και ανάπτυξη λογισμικού για ανάλυση σύνθετων δικτύων: Από τις κεντρικότητες και τις κοινότητες στην επιδημιολογία

Design and development of software for complex network analysis: From centralities and communities to epidemics

Διπλωματική εργασία

υπό

Σεγδίτσα Κωνσταντίνου

Επιβλέποντες: Κατσαρός Δημήτριος
Επίκουρος καθηγητής Π.Θ.

Μποζάνης Παναγιώτης
Καθηγητής Π.Θ.

Βόλος, Οκτώβριος 2015

Η σελίδα αυτή είναι σκόπιμα λευκή.

Διπλωματική εργασία για την απόκτηση του διπλώματος του Μηχανικού Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων του Πανεπιστημίου Θεσσαλίας, στα πλαίσια του προγράμματος προπτυχιακών σπουδών του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών της Πολυτεχνικής Σχολής του Πανεπιστημίου Θεσσαλίας.

.....
Σεγδίτσας Κωνσταντίνος

Διπλωματούχος Μηχανικός Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων Πανεπιστημίου Θεσσαλίας

Copyright © Segditsas Konstantinos, 2015

Με επιφύλαξη παντός δικαιώματος. All rights reserved

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Η σελίδα αυτή είναι σκόπιμα λευκή.

Ευχαριστίες

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας, θα ήθελα να ευχαριστήσω μια σειρά ανθρώπων που με βοήθησαν με ποικίλους τρόπους καθ' όλη την διάρκεια της εκπόνησης και της συγγραφής της.

Αρχικά θα ήθελα να ευχαριστήσω θερμά τον κύριο Δημήτριο Κατσαρό, που μου έδωσε την ευκαιρία να εργαστώ δίπλα του πάνω σε ένα τόσο ενδιαφέρον τομέα, καθώς και για την καθοδήγηση που μου παρείχε από την αρχή μέχρι το τέλος. Επίσης τον κύριο Παναγιώτη Μποζάνη, για την πολύτιμη βοήθειά του μέσα από τις εύστοχες παρατηρήσεις και υποδείξεις του ιδιαίτερα σε θέματα αλγορίθμων.

Επίσης, ένα πολύ μεγάλο ευχαριστώ στον συνάδελφο και πολύ καλό μου φίλο κύριο Πεσλή Παντελή, για την πολύτιμη βοήθεια και τις προγραμματιστικές συμβουλές που μου παρείχε κατά την συγγραφή του κώδικα, δίχως τα οποία η περάτωση της διπλωματικής αυτής εργασίας θα ήταν σχεδόν αδύνατη.

Τέλος, θέλω να ευχαριστήσω από καρδιάς την οικογένειά μου για την εμπιστοσύνη και την αμέριστη ψυχολογική τους υποστήριξη καθ' όλη την διάρκεια των σπουδών μου, καθώς και για την υπομονή που επέδειξαν όλα αυτά τα χρόνια.

Η σελίδα αυτή είναι σκόπιμα κενή.

Περίληψη

Τα σύνθετα δίκτυα αποτελούν την ραχοκοκαλιά πολλών πολύπλοκων συστημάτων του σημερινού κόσμου. Χαρακτηριστικά παραδείγματα περιλαμβάνουν τα κοινωνικά δίκτυα, τα δίκτυα συγκοινωνιών και μεταφορών, τα οικονομικά, τα βιολογικά, όπως επίσης και τα δίκτυα τηλεπικοινωνιών. Η οπτικοποίηση και η ανάλυση των σύνθετων δικτύων πραγματοποιείται με την βοήθεια της θεωρίας γραφημάτων.

Αντικείμενο της παρούσας διπλωματικής εργασίας ήταν ο σχεδιασμός και η ανάπτυξη λογισμικού οπτικοποίησης και ανάλυσης σύνθετων δικτύων. Το λογισμικό AniNet χτίστηκε στην γλώσσα προγραμματισμού Java, και φέρει εκτός των άλλων δυνατότητες γραφικής δημιουργίας και αναπαράστασης γραφημάτων, αλγορίθμους εύρεσης κεντρικότητας, εντοπισμού κοινοτήτων και εξάπλωσης επιδημιών, παρέχοντας παράλληλα την δυνατότητα εξαγωγής των αποτελεσμάτων του.

Στο κεφάλαιο 1 γίνεται εισαγωγή στις έννοιες και την χρησιμότητα ανάλυσης των σύνθετων δικτύων. Το κεφάλαιο 2 παρουσιάζει τα δημοφιλέστερα υπάρχοντα λογισμικά, συγκρίνοντας μερικά από αυτά ως προς τα πλεονεκτήματα και μειονεκτήματά τους. Στο κεφάλαιο 3 παρουσιάζονται τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη του AniNet, ενώ στο κεφάλαιο 4 γίνεται μια εισαγωγή στο γραφικό του περιβάλλον, επεξηγώντας παράλληλα τους τρόπους δημιουργίας γραφημάτων και αλληλεπίδρασης με το πρόγραμμα. Εν συνεχεία στα κεφάλαια 5, 6 και 7 παρουσιάζονται οι λειτουργίες εύρεσης κεντρικότητας, εντοπισμού κοινοτήτων και εξάπλωσης επιδημιών αντίστοιχα, που παρέχονται από το AniNet, ακολουθούμενες σε κάθε περίπτωση από ένα χαρακτηριστικό παράδειγμα που επεξηγεί την λειτουργία του προγράμματος, όπως επίσης και τους κανόνες αλληλεπίδρασης με αυτό, όπου κρίνεται απαραίτητο. Το κεφάλαιο 8 επεξηγεί την διαδικασία εξαγωγής των αποτελεσμάτων του AniNet. Τέλος στο κεφάλαιο 9, γίνεται μια συμπερασματική σύνοψη των δυνατοτήτων του λογισμικού που κατασκευάστηκε, κάνοντας παράλληλα μια αναφορά σε πιθανές βελτιστοποιήσεις και μελλοντικές επεκτάσεις.

Abstract

Complex networks constitute the foundation of many complex systems nowadays. Typical examples of these networks are social networks, transportation networks, financial, biological, as well as communication networks. Visualization and analysis of complex networks is achieved by implementing the theory of graphs.

The goal of this specific dissertation was the design and development of a visualization and analysis software for complex networks. “AviNet” software was developed in Java and is able, amongst others, to create and represent graphs, centralities computation algorithms, communities detection and epidemics diffusion, and enables the extraction of the results.

Chapter 1 is an introduction on the meaning and the significance of complex networks analysis. Chapter 2 exhibits the most popular existing software, while comparing some of them according to their advantages and disadvantages. In chapter 3 the tools that were used to develop AviNet are displayed whereas chapter 4 gives an introduction on its graphical environment while explaining the ways of graph creation and ways of interacting with the program. Consequently, the functions of centralities computation, communities detection, and epidemics diffusion are exhibited in chapters 5, 6 and 7. These functions, which are provided by AviNet, are followed, in each case, by a distinctive example, which explains the way the program operates as well as the interaction regulations wherever it is seen fit. Chapter 8 explains how to extract the results in AviNet. Chapter 9 consists of a conclusion and summary of the potentials the created software provides, and reports on possible future enhancement and expansion.

Περιεχόμενα

Ευχαριστίες.....	v
Περίληψη.....	vii
Abstract	viii
Πίνακας Εικόνων	xi
1 Εισαγωγή.....	1
1.1 Γενικά για τα σύνθετα δίκτυα	1
1.2 Ανάλυση και χρησιμότητα.....	5
2 Λογισμικά για σύνθετα δίκτυα.....	8
2.1 Επισκόπηση υπαρχόντων λογισμικών	8
2.2 Συμπεράσματα και ενέργειες.....	10
3 Το πρόγραμμα AviNet	11
3.1 Λογισμικά που χρησιμοποιήθηκαν	11
3.1.1 NetBeans IDE	11
3.1.2 Git και SourceTree	12
3.2 Το περιβάλλον του AviNet	13
4 Ξεκινώντας με το AviNet	15
4.1 Δημιουργία γραφημάτων βήμα προς βήμα.....	15
4.1.1 Παραδείγματα γραφημάτων.....	17
4.2 Δημιουργία γραφημάτων μέσω αλγορίθμων	19
4.2.1 Random Graph.....	19
4.2.2 Small World Graph	21
4.3 Διάβασμα αρχείων για την δημιουργία γραφημάτων.....	24
4.4 Αποθήκευση γραφημάτων για μελλοντική χρήση.....	28
5 Κεντρικότητες στο AviNet.....	31
5.1 Degree Centrality.....	32
5.1.1 Μη κατευθυνόμενα, μη βεβαρημένα γραφήματα	32
5.1.2 Κατευθυνόμενα, μη βεβαρημένα γραφήματα	34
5.1.3 Μη κατευθυνόμενα, βεβαρημένα γραφήματα	35
5.1.4 Κατευθυνόμενα, βεβαρημένα γραφήματα.....	37
5.2 Closeness Centrality	38
5.3 Betweenness Centrality.....	40
5.4 Edge Betweenness Centrality.....	42

5.5 Power Community Index (μ -PCI)	44
5.6 k-Shell/s-Core	47
5.7 Page Rank	50
6 Κοινότητες στο Avinet.....	54
6.1 EBC (Girvan & Newman using Edge Betweenness).....	55
6.2 CiBC (Using node Betweenness).....	57
7 Επιδημιολογίες στο Avinet	61
7.1 Linear Threshold (SIS).....	62
7.2 Independent Cascade (SIR).....	72
8 Εξαγωγή αποτελεσμάτων από το Avinet.....	80
9 Σύνοψη – Μελλοντικές επεκτάσεις.....	82
9.1 Σύνοψη	82
9.2 Μελλοντικές επεκτάσεις	82
Βιβλιογραφία	84

Πίνακας Εικόνων

Εικόνα 1 - Παράδειγμα κοινωνικού δικτύου	2
Εικόνα 2 - Δίκτυο τραπεζών	2
Εικόνα 3 - Ευρωπαϊκό σιδηροδρομικό δίκτυο	3
Εικόνα 4 - Δίκτυο πρωτεΐνης	4
Εικόνα 5 - Δίκτυο υπολογιστών.....	4
Εικόνα 6 - Παράδειγμα Προσομοίωσης Σύνθετου Δικτύου	5
Πίνακας 1 - Συγκριτικός πίνακας λογισμικών	10
Εικόνα 7 - Το AviNet στο NetBeans	11
Εικόνα 8 - Πορεία εργασίας του AviNet στο SourceTree	12
Εικόνα 9 - Περιβάλλον AviNet.....	13
Εικόνα 10 - Preview panel popup menu	15
Εικόνα 11 - Node popup menu.....	16
Εικόνα 12 - Επιλογή βάρους ακμής.....	17
Εικόνα 13 - Παράδειγμα μη κατευθυνόμενου, μη βεβαρημένου γραφήματος.....	18
Εικόνα 14 - Παράδειγμα κατευθυνόμενου, βεβαρημένου γραφήματος	18
Εικόνα 15 - Επιλογή Random Graph	19
Εικόνα 16 - Παράμετροι εισόδου ενός Random Graph	20
Εικόνα 17 - Erdős-Rényi Random Graph	21
Εικόνα 18 - Επιλογή Small world graph.....	22
Εικόνα 19 - Παράμετροι εισόδου Small world graph 1/2	22
Εικόνα 20 - Παράμετροι εισόδου Small world graph 2/2	23
Εικόνα 21 - Small World graph	23
Εικόνα 22 - Ανεπιτυχής ανάγνωση αρχείου εισόδου	25
Εικόνα 23 - Πρότυπο αρχείο εισόδου για την κατασκευή κατευθυνόμενου βεβαρημένου γραφήματος	25
Εικόνα 24 - Πρότυπο αρχείο εισόδου για την κατασκευή μη κατευθυνόμενου μη βεβαρημένου γραφήματος	26
Εικόνα 25 - Μενού για την ανάγνωση γραφημάτων	26
Εικόνα 26 - Αναπαράσταση γραφήματος εικόνας 22	27
Εικόνα 27 - Αναπαράσταση γραφήματος εικόνας 23	28
Εικόνα 28 - Παράδειγμα γραφήματος που θα αποθηκευτεί.....	29
Εικόνα 29 - example.cnt	29
Εικόνα 30 - Προτροπή αποθήκευσης γραφήματος	30
Εικόνα 31 – Centralities.....	31
Εικόνα 32 - Παράθυρο κανονικοποίησης αποτελεσμάτων	32
Εικόνα 33 - Μη κανονικοποιημένο Degree σε undirected unweighted δίκτυο	33
Εικόνα 34 - Κανονικοποιημένο Degree σε undirected unweighted δίκτυο.....	33
Εικόνα 35 - Μη κανονικοποιημένο Degree σε directed unweighted δίκτυο.....	34
Εικόνα 36 - Κανονικοποιημένο Degree σε directed unweighted δίκτυο.....	35
Εικόνα 37 - Μη κανονικοποιημένο Degree σε undirected weighted δίκτυο.....	36
Εικόνα 38 - Κανονικοποιημένο Degree σε undirected weighted δίκτυο.....	36
Εικόνα 39 - Μη κανονικοποιημένο Degree σε directed weighted δίκτυο.....	37

Εικόνα 40 - Κανονικοποιημένο Degree σε directed weighted δίκτυο	38
Εικόνα 41 - Παράδειγμα υπολογισμού της Closeness Centrality	39
Εικόνα 42 - Betweenness Centrality σε undirected unweighted γράφημα	41
Εικόνα 43 - EBC σε directed unweighted γράφημα	43
Εικόνα 44 - EBC σε directed weighted γράφημα	43
Εικόνα 45 - Ενημερωτικό μήνυμα για την εκτέλεση του μ-PCI.....	45
Εικόνα 46 - Επιλογή παραμέτρου μ για το μ-PCI	45
Εικόνα 47 - PCI για $\mu=1$	46
Εικόνα 48 - PCI για $\mu=2$	46
Εικόνα 49 - Ενημερωτικό μήνυμα για την εκτέλεση του k-Shell/s-Core.....	47
Εικόνα 50 - Το k-Shell στο AviNet.....	48
Εικόνα 51 - Το s-Core στο AviNet	49
Εικόνα 52 - Παράδειγμα δικτύου ιστοσελίδων.....	50
Εικόνα 53 - Ενημερωτικό μήνυμα για την εκτέλεση του Page Rank.....	52
Εικόνα 54 - Υπολογισμός Page Rank στο AviNet.....	53
Εικόνα 55 – Communities.....	54
Εικόνα 56 - Εύρεση κοινοτήτων με βάση την Edge Betweenness	56
Εικόνα 57 - Ενημερωτικό μήνυμα για την εκτέλεση του CiBC.....	58
Εικόνα 58 - Καθορισμός παραμέτρου s	58
Εικόνα 59 - CiBC για $s=1$	59
Εικόνα 60 - CiBC για $s=2$	60
Εικόνα 61 - Epidemics στο AviNet	61
Εικόνα 62 - Παράθυρο επιλογής εισόδου για το Linear Threshold	63
Εικόνα 63 - Επιλογή αρχικών κόμβων βάση Κεντρικότητας	64
Εικόνα 64 - Επιλογή αρχικών κόμβων χειροκίνητα.....	64
Εικόνα 65 - Τυχαία επιλογή αρχικά μολυσμένων κόμβων	65
Εικόνα 66 - Κοινό threshold για όλους τους κόμβους	66
Εικόνα 67 - Τυχαίο threshold για κάθε κόμβο του γραφήματος ξεχωριστά	66
Εικόνα 68 - Εισαγωγή συγκεκριμένου κατωφλίου για κάθε κόμβο ξεχωριστά	67
Εικόνα 69 - Εισαγωγή συγκεκριμένης πιθανότητας μόλυνσης για όλες τις ακμές	67
Εικόνα 70 - Τυχαία πιθανότητα μόλυνσης για κάθε διαφορετική ακμή	68
Εικόνα 71 - Χειροκίνητος καθορισμός πιθανότητας μόλυνσης για κάθε ακμή ξεχωριστά....	69
Εικόνα 72 - Αποθήκευση εισόδων για μελλοντική χρήση	69
Εικόνα 73 - Αρχείο δεδομένων για το Linear Threshold	70
Εικόνα 74 - Παράθυρο ενημέρωσης ακατάλληλου αρχείου	70
Εικόνα 75 - Το Linear Threshold στο AviNet	71
Εικόνα 76 - Παράθυρο επιλογής εισόδου για το Independent Cascade	72
Εικόνα 77 - Επιλογή αρχικών κόμβων βάση κεντρικότητας.....	73
Εικόνα 78 - Επιλογή αρχικά μολυσμένων κόμβων χειροκίνητα	74
Εικόνα 79 - Τυχαία επιλογή αρχικών κόμβων.....	74
Εικόνα 80 - Συγκεκριμένη πιθανότητα μόλυνσης, ίδια για όλες τις ακμές	75
Εικόνα 81 - Εισαγωγή διαφορετικής πιθανότητας μόλυνσης κάθε ακμής από τον χρήστη ..	76
Εικόνα 82 - Τυχαία πιθανότητα μόλυνσης για κάθε ακμή ξεχωριστά.....	76
Εικόνα 83 - Αποθήκευση δεδομένων για μελλοντική χρήση.....	77
Εικόνα 84 - Αρχείο δεδομένων για το Independent Cascade	77

Εικόνα 85 - Μήνυμα ακατάλληλου αρχείου στο Independent Cascade	78
Εικόνα 86 - Το Independent Cascade στο AniNet	79
Εικόνα 87 - Εξαγωγή αποτελεσμάτων στο AniNet	80
Εικόνα 88 - Παράδειγμα αρχείου αποτελεσμάτων	81

1 Εισαγωγή

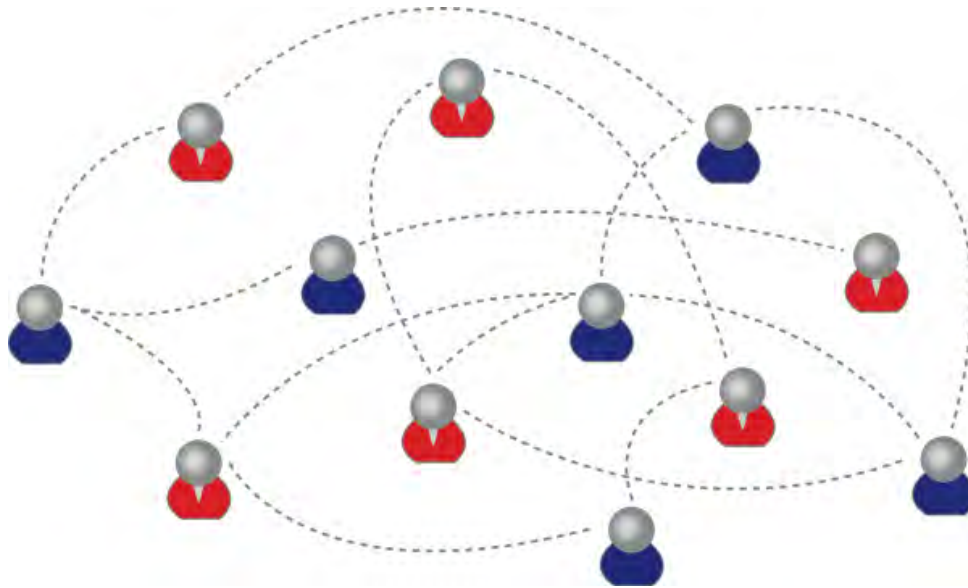
1.1 Γενικά για τα σύνθετα δίκτυα

Στο πλαίσιο της θεωρίας δικτύων, ως σύνθετα δίκτυα ορίζονται τα γραφήματα με μη τετριμμένα τοπολογικά χαρακτηριστικά και ιδιότητες. Τα ιδιαίτερα αυτά χαρακτηριστικά των σύνθετων δικτύων δεν συναντώνται σε απλά δίκτυα, όπως τυχαία γραφήματα και πλέγματα, αλλά σε γραφήματα που μοντελοποιούν πραγματικά συστήματα.

Τα περισσότερα συστήματα που συναντά κανείς στον πραγματικό κόσμο μπορούν να μοντελοποιηθούν ως δίκτυα και να αναπαρασταθούν με γραφήματα, όπου τα μέλη του συστήματος είναι οι κόμβοι, και οι αλληλεπιδράσεις μεταξύ αυτών παρουσιάζονται διαμέσου των ακμών.

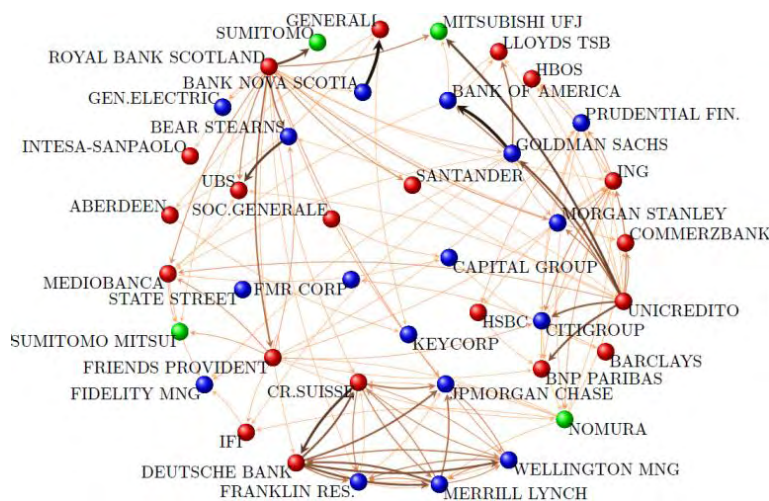
Η συστηματική μελέτη των σύνθετων δικτύων ξεκίνησε από τον τομέα της κοινωνιολογίας. Χαρακτηριστικό παράδειγμα αποτελεί το πείραμα του Milgram. Πρόκειται για μια σειρά πειραμάτων κοινωνικής ψυχολογίας που διεξήχθη από το πανεπιστήμιο του Yale υπό την καθοδήγηση του καθηγητή Stanley Milgram. Στο πείραμα αυτό μετρήθηκε η προθυμία των συμμετεχόντων να υπακούσει σε μια μορφή εξουσίας, που τους έδινε εντολές να προβούν σε ενέργειες οι οποίες συγκρούονταν με την προσωπική τους συνείδηση.

Η ανάλυση των κοινωνικών δικτύων είναι στην ουσία μια στρατηγική για την διερεύνηση των κοινωνικών δομών, με την χρήση της θεωρίας των δικτύων και των γραφημάτων. Οι κόμβοι, που μπορεί να αναπαριστούν από ανθρώπους μέχρι και ολόκληρες κοινωνικές ομάδες, χαρακτηρίζουν τις θεμελιώδεις δομές του κοινωνικού δικτύου, και οι ακμές χαρακτηρίζουν τις εξαρτήσεις μεταξύ αυτών. Παραδείγματα κοινωνικών δομών που συνήθως οπτικοποιούνται μέσω της ανάλυσης των σύνθετων δικτύων περιλαμβάνουν τα κοινωνικά δίκτυα, τα μέσα μαζικής ενημέρωσης, τη γνωριμία και τη φιλία των ανθρώπων ή κοινωνικών ομάδων, και τη συγγένεια.



Εικόνα 1 - Παράδειγμα κοινωνικού δικτύου

Ένας ακόμα τομέας όπου απαντώνται τα σύνθετα δίκτυα, είναι αυτός της οικονομίας. Πιο συγκεκριμένα, μέσα από την θεωρία των σύνθετων δικτύων, παρέχεται η δυνατότητα να κατανοηθεί η δομή και η δυναμική που διέπει τις χρηματοπιστωτικές αγορές, ώστε να προβλεφθούν και να κατανοηθούν οι αλληλεπιδράσεις και τα αποτελέσματα διαφόρων ενεργειών. Η γνώση που παρέχει η ανάλυση των σύνθετων δικτύων, δύναται να οδηγήσει τον άνθρωπο σε αποφάσεις και ενέργειες, που θα μπορούσαν ακόμα και να επηρεάσουν την δυναμική μιας αγοράς. Ακόμα, η κατανόηση των σημαντικότερων παραγόντων σταθερότητας μιας οικονομίας και η προστασία αυτών, θα μπορούσε να περιορίσει σημαντικά τους κινδύνους αποτυχίας της.



Εικόνα 2 - Δίκτυο τραπεζών

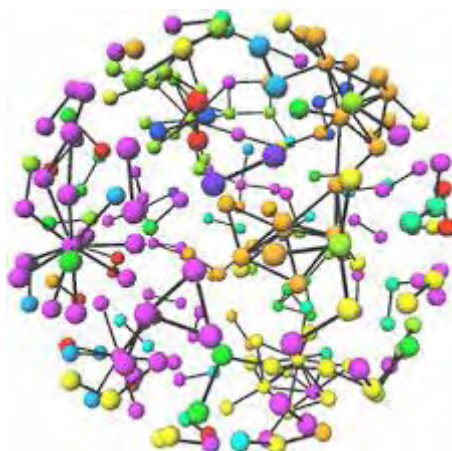
Ιδιαίτερα σημαντική επίσης είναι η ύπαρξη των σύνθετων δικτύων στον τομέα των μεταφορών, είτε στην μεταφορά των ανθρώπων μέσα από το οδικό δίκτυο μιας πόλης ή χώρας, είτε στην μεταφορά προϊόντων η οποία με την σειρά της είναι στενά συνδεδεμένη με τον τομέα της οικονομίας. Ως στρατηγικός παράγοντας για μια χώρα προκειμένου να επιβιώσει στον παγκόσμιο ιστό, τα συστήματα μεταφοράς έχουν προσελκύσει την εκτεταμένη προσοχή από διάφορους επιστημονικούς κλάδους, για αρκετά μεγάλο χρονικό διάστημα. Την τελευταία δεκαετία οι μελέτες πάνω στα δίκτυα μεταφορών, έχουν παρουσιάσει υψηλή πρόοδο. Δρόμοι, ράγες τρένων και σιδηρόδρομοι άλλο μπορεί να χρησιμοποιηθεί στις μεταφορές είναι πλέον οργανωμένα και εξετάζονται ως σύνθετα δίκτυα.



Εικόνα 3 - Ευρωπαϊκό σιδηροδρομικό δίκτυο

Στις επιστήμες της ιατρικής και της βιολογίας, η εισαγωγή και η μελέτη των σύνθετων δικτύων, είναι καθοριστική. Τα οικοσυστήματα μέσα σε έναν οργανισμό μπορούν να μοντελοποιηθούν ως δίκτυα αλληλεπιδρώντων ειδών, ενώ μια πρωτεΐνη μπορεί να μοντελοποιηθεί ως ένα δίκτυο αμινοξέων. Με τον τρόπο αυτό δίνεται η δυνατότητα να μελετηθούν οι αλληλεπιδράσεις των μορίων και των κυττάρων μέσα σε ένα οργανισμό, ενισχύοντας έτσι την εξέλιξη της βιολογίας και κατ' επέκταση της

ιατρικής. Ένα ακόμα χαρακτηριστικό παράδειγμα είναι η μελέτη της εξάπλωσης μιας επιδημίας σε ένα δίκτυο ανθρώπων.



Εικόνα 4 - Δίκτυο πρωτεΐνης

Το διαδίκτυο και οι τηλεπικοινωνίες μπορούν με την σειρά τους να αναπαρασταθούν από γραφήματα και να μελετηθούν ως σύνθετα δίκτυα. Ένα σύνολο ιστοσελίδων δύναται να αποτελεί τους κόμβους ενός γραφήματος, με τις κατευθυνόμενες ακμές να αναπαριστούν τους εισερχόμενους και εξερχόμενους συνδέσμους ανάμεσά τους. Η σημαντικότητα και συνεπώς η αξία μιας ιστοσελίδας, που μπορεί να αποτελεί και μια ολόκληρη επιχείρηση, μπορεί να αξιολογηθεί μελετώντας το σύνθετο δίκτυο και τις ιδιότητές του.



Εικόνα 5 - Δίκτυο υπολογιστών

1.2 Ανάλυση και χρησιμότητα

Τα σύνθετα δίκτυα, επομένως, αποτελούν την ραχοκοκαλιά πολλών πολύπλοκων συστημάτων. Έτσι λοιπόν η θεωρητική και υπολογιστική ανάλυσή τους καθίσταται αναγκαία για την απόκτηση γνώσεων γύρω από τα συστήματα αυτά, με στόχο τόσο την κατανόηση και βελτίωσή τους, όσο και την απόκτηση γνώσεων σε πλήθος άλλων εφαρμογών.

Εδώ και αρκετές δεκαετίες η μελέτη των σύνθετων δικτύων έχει διαδραματίσει αποφασιστικό ρόλο στις επιστήμες, καθώς και στην ανάπτυξη πολλών μαθηματικών και στατιστικών εργαλείων.



Εικόνα 6 - Παράδειγμα Προσομοίωσης Σύνθετου Δικτύου

Τα ακόλουθα ερωτήματα, μπορούν να βοηθήσουν σημαντικά ώστε να γίνει αντιληπτός ο τρόπος με τον οποίο πρέπει να αναλυθεί ένα σύνθετο δίκτυο, η χρησιμότητα της ανάλυσης αυτής, και οι ιδιότητες που μπορούν να προκύψουν.

- Αν χρειάζεται κανείς να προσλάβει νέους ανθρώπους για την επιχείρησή του, ποια πρέπει να είναι τα κριτήριά του;

- Αν υπάρχει η ανάγκη διάδοσης μιας φήμης σε ένα δίκτυο ανθρώπων, ποιοι είναι εκείνοι που πρέπει να την αναλάβουν;
- Αν πρέπει να οριστεί ένας αρχηγός σε μια ομάδα, με ποιο κριτήριο είναι προτιμότερο να γίνει;
- Πως μπορεί κανείς να ανακαλύψει τους κεντρικότερους οδικούς άξονες σε μια πόλη ή χώρα;

Οι πιο διαδεδομένες κατηγορίες ανάλυσης των σύνθετων δικτύων, που δύναται να ικανοποιήσουν ερωτήματα όπως τα παραπάνω, είναι οι εξής:

Εύρεση κεντρικότητων (Centralities)

Στην θεωρία των γραφημάτων και της ανάλυσης δικτύων, οι δείκτες κεντρικότητας εντοπίζουν τις σημαντικότερες κορυφές ενός γραφήματος. Στους δείκτες αυτούς συγκαταλέγονται η κεντρικότητα βαθμού (Degree Centrality), η ενδιάμεση κεντρικότητα (Betweenness Centrality), η κεντρικότητα εγγύτητας (Closeness Centrality), η ενδιάμεση κεντρικότητα ακμών (Edge Betweenness Centrality), ο δείκτης PCI, το Page Rank, καθώς και πολλά άλλα.

Η κεντρικότητα βαθμού δείχνει το πλήθος των κόμβων με τους οποίους συνδέεται ο υπό εξέταση κόμβος, ενώ η ενδιάμεση κεντρικότητα δηλοί το πόσο συχνά συναντάται ένας κόμβος κατά την διαπέραση του γραφήματος. Από την άλλη πλευρά η κεντρικότητα εγγύτητας δείχνει πόσο κεντρικός είναι ένας κόμβος μέσα σε ένα γράφημα, ενώ ο δείκτης PCI δείχνει ανάμεσα σε πόσο δημοφιλείς κόμβους βρισκόμαστε. Οι πλήρεις ορισμοί όλων των προαναφερθέντων κεντρικότητων θα δοθούν σε επόμενα κεφάλαια.

Εύρεση κοινοτήτων (Communities)

Στην μελέτη των σύνθετων δικτύων, οι κόμβοι ενός γραφήματος μπορούν εύκολα να ομαδοποιηθούν σε σύνολα, πολλές φορές επικαλυπτόμενα, τα οποία είναι πυκνά συνδεδεμένα στο εσωτερικό τους. Τα κριτήρια με τα οποία μπορεί να γίνει αυτή η ομαδοποίηση και η εύρεση των κοινοτήτων, ποικίλουν και εξαρτώνται από τις ανάγκες του εκάστοτε προβλήματος. Οι αλγόριθμοι εύρεσης κοινοτήτων είτε μελετούν τις ιδιότητες ενός γραφήματος σε σχέση με τα χαρακτηριστικά των πινάκων που συνδέονται με το γράφημα, είτε βασίζονται σε ιδιότητες όπως είναι οι κεντρικότητες.

Εξάπλωση επιδημιών (Epidemics)

Ιδιαίτερα σημαντική είναι η ανάλυση των σύνθετων δικτύων από την σκοπιά των επιδημιών. Πιο συγκεκριμένα, στα γραφήματα που μοντελοποιούν κυρίως κοινωνικά δίκτυα, η εξάπλωση μιας φήμης και η μεγιστοποίησή της, είναι ένας τομέας που έχει απασχολήσει έντονα τους επιστήμονες. Ο τομέας της διαφήμισης αποτελεί χαρακτηριστικό παράδειγμα, αφού η σωστή επιλογή των κόμβων που θα ξεκινήσουν την διάδοση μιας πληροφορίας, αποτελεί μείζων ζήτημα. Από την άλλη πλευρά στον τομέα της βιολογίας και της ιατρικής η πειραματική εξάπλωση μιας επιδημίας σε ένα σύνθετο δίκτυο που αναπαριστά τους ανθρώπους μιας κοινωνίας, μπορεί να βοηθήσει στον περιορισμό και τελικά την αντιμετώπισή της. Οι αλγόριθμοι της κατηγορίας αυτής, βασίζονται σε δύο βασικά μοντέλα διάχυσης, το Linear Threshold και το Independent Cascade.

2 Λογισμικά για σύνθετα δίκτυα

2.1 Επισκόπηση υπαρχόντων λογισμικών

Οι αλγόριθμοι που οδηγούν στην ανάλυση των γραφημάτων που προσομοιώνουν τα σύνθετα δίκτυα, είναι τις περισσότερες φορές χρονοβόροι και ιδιαίτερα πολύπλοκοι. Είναι προφανές λοιπόν ότι για την εξαγωγή των επιθυμητών αποτελεσμάτων, ειδικότερα σε μεγάλα δίκτυα, είναι αναγκαία η ύπαρξη του κατάλληλου λογισμικού. Στις μέρες μας υπάρχει μια μεγάλη ποικιλία τέτοιων προγραμμάτων, το καθένα από τα οποία φέρει τα πλεονεκτήματα και τα μειονεκτήματά του.

Τα λογισμικά ανάλυσης σύνθετων δικτύων χωρίζονται κατά κύριο λόγο σε δύο βασικές κατηγορίες. Σε εκείνα που βασίζονται σε ένα γραφικό περιβάλλον, και σε εκείνα που είναι χτισμένα για γλώσσες προγραμματισμού.

Σε γενικές γραμμές τα λογισμικά που είναι βασισμένα σε GUI είναι ευκολότερα στην χρήση, καθώς το περιβάλλον εργασίας τους τα καθιστά ιδιαίτερα φιλικά. Μερικά ευρέως διαδεδομένα παραδείγματα, είναι τα NetMiner, UCINet, Rajek, GUESS, ORA, Cytoscape, Gephi και muxViz. Για περισσότερο επαγγελματική χρήση θα μπορούσε κανείς να απευθυνθεί σε ένα από τα Orgnet, το οποίο παρέχει και εκπαίδευση για τη χρήση του λογισμικού της, Polinode, Keyhubs, KeyLines, KXEN και Keynetiq. Τα μειονεκτήματά τους συναντώνται στο γεγονός ότι τα περισσότερα από αυτά περιορίζονται περισσότερο στην γραφική αναπαράσταση των γραφημάτων, χωρίς να δίνουν έμφαση στο εύρος των αλγορίθμων που υποστηρίζουν.

Στα πιο διαδεδομένα και οργανωμένα λογισμικά της δεύτερης κατηγορίας, θα μπορούσε κανείς να συμπεριλάβει τα NetMiner και NetworkX. Παρά την δικαιολογημένη δυσκολία στην εκμάθηση των λογισμικών αυτής της κατηγορίας, μερικά από αυτά αναπτύσσονται πολύ γρήγορα όσον αφορά την λειτουργικότητα και τα χαρακτηριστικά τους. Επίσης παρέχουν εκτεταμένους οδηγούς και βιβλιοθήκες των λειτουργιών τους. Βασικό μειονέκτημα των περισσότερων από αυτά τα λογισμικά, είναι η έλλειψη γραφικής προσομοίωσης των δικτύων. Η οπτικοποίηση διευκολύνει την ποιοτική ερμηνεία και κατανόηση των δεδομένων, καθώς και την ερμηνεία των εξαγόμενων ιδιοτήτων. Πρόκειται λοιπόν για μια σημαντική λειτουργία, που όμως δεν προσφέρει ένα αρκετά μεγάλο ποσοστό των ήδη υπαρχόντων προγραμμάτων, με αποτέλεσμα την δυσκολία ερμηνείας των αποτελεσμάτων που προσφέρουν.

Παρακάτω γίνεται μια πιο εκτεταμένη παρουσίαση των χαρακτηριστικών, των ιδιοτήτων, των πλεονεκτημάτων και των μειονεκτημάτων, τεσσάρων από τα πιο δημοφιλή λογισμικά ανάλυσης σύνθετων δικτύων. Εν συνεχεία παρουσιάζεται ένας συγκριτικός πίνακας.

Gephi: Πρόκειται για ένα λογισμικό ανοιχτού κώδικα, αναπαράστασης και εξερεύνησης γραφημάτων, διαθέσιμο για Windows, Linux και Mac OS. Υποστηρίζει όλους τους τύπους δικτύων, κατευθυνόμενα και μη, και είναι σε θέση να χειρίζεται πολύ μεγάλα γραφήματα (έως και ένα εκατομμύριο κόμβους). Προσφέρει γραφική αναπαράσταση των γραφημάτων, ενώ στα πλεονεκτήματά του συγκαταλέγεται επίσης και η παρουσία των σημαντικότερων Centralities. Βασικά μειονεκτήματά του, είναι η έλλειψη εντοπισμού κοινοτήτων και αλγορίθμων για τα Epidemics.

Pajek: Είναι ένα δωρεάν πρόγραμμα για Windows, το οποίο χρησιμοποιείται κυρίως για τις ανάγκες της ακαδημαϊκής έρευνας. Είναι γρήγορο, παρέχει εργαλεία οπτικοποίησης και ανάλυσης σύνθετων δικτύων, όπως επίσης υποστηρίζει πίνακες και λίστες γειτνίασης ως εναλλακτικές εισόδους. Βασικό μειονέκτημα του, η έλλειψη παρουσίας αλγορίθμων εξάπλωσης επιδημιών.

NetworkX: Το NetworkX είναι ένα πακέτο λογισμικού της γλώσσας Python για την δημιουργία, τον χειρισμό και την μελέτη της δομής και των λειτουργιών των σύνθετων δικτύων. Έχει την δυνατότητα να παράγει πολλά είδη τυχαίων γραφημάτων, όπως επίσης να αναλύσει σε αρκετά ικανοποιητικό βαθμό την δομή τους. Υπολείπεται σε μεγάλο βαθμό στον εντοπισμό κοινοτήτων και εξάπλωσης επιδημιών. Επίσης στηρίζεται σε άλλα προγράμματα για την οπτικοποίηση των γραφημάτων.

Igraph: Πρόκειται για μια συλλογή εργαλείων ανάλυσης δικτύων, με έμφαση στην αποδοτικότητα και στην ευκολία χρήσης. Είναι δωρεάν, ανοιχτού κώδικα, και μπορεί να προγραμματιστεί σε R, Python, C και C++. Παρέχει την δυνατότητα εντοπισμού κοινοτήτων, όχι όμως αλγορίθμων για τα Epidemics. Βασικό του μειονέκτημα επίσης είναι η δυσκαμψία που παρουσιάζει στον τομέα της παραμετροποίησης από τον χρήστη.

	Pajek	Gephi	NetworkX	igraph
Είσοδος/Εξοδος	✓	✓✓	✓✓	✓
Παραμετροποίηση	✓	✓	✓✓	✗
Οπτικοποίηση	✓✓	✓	✗	✓✓
Κεντρικότητες	✓	✓	✓	✓
Κοινοτήτες	✓	✗	✗	✓
Επιδημίες	✗	✗	✗	✗

Πίνακας 1 - Συγκριτικός πίνακας λογισμικών

2.2 Συμπεράσματα και ενέργειες

Από τα λεγόμενα της προηγούμενης ενότητας και από την εμπειρία χρήσης μερίδας των προαναφερθέντων προγραμμάτων, κρίθηκε αναγκαίο να δημιουργηθεί ένα λογισμικό το οποίο θα φέρει κάποια βασικά χαρακτηριστικά, τα οποία δεν συναντώνται μέχρι στιγμής.

Το λογισμικό αυτό, θα πρέπει να είναι ιδιαίτερα φιλικό προς τον χρήστη, δίνοντάς του την δυνατότητα να εισάγει και εν συνεχεία να προσομοιώσει ένα σύνθετο δίκτυο με ποικίλους τρόπους. Να παρέχει επίσης την λειτουργία εξαγωγής των αποτελεσμάτων του σε αρχεία προσβάσιμα από διαφορετικά προγράμματα, για περαιτέρω ανάλυσή τους. Ακόμα να εμπεριέχει την δυνατότητα εξαγωγής των βασικότερων ιδιοτήτων ενός σύνθετου δικτύου, ταυτόχρονα όμως να περιλαμβάνει αλγόριθμους εύρεσης κοινοτήτων και εξάπλωσης επιδημιών. Τα δύο τελευταία χαρακτηριστικά, είναι αυτά που εκλείπουν περισσότερο από τα προγράμματα που υπάρχουν ήδη, ειδικά αν τα αναζητήσει κανείς σε συνδυασμό με όλα τα παραπάνω.

Έτσι λοιπόν το πρόγραμμα αυτό, δεν θα επικεντρώνεται σε έναν συγκεκριμένο τρόπο εξέτασης γραφημάτων, και συνεπώς δεν θα περιορίζεται στην ανάλυση μιας συγκεκριμένης κατηγορίας ιδιοτήτων, όπως συμβαίνει με το μεγαλύτερο ποσοστό των λογισμικών που είναι ήδη υλοποιημένα.

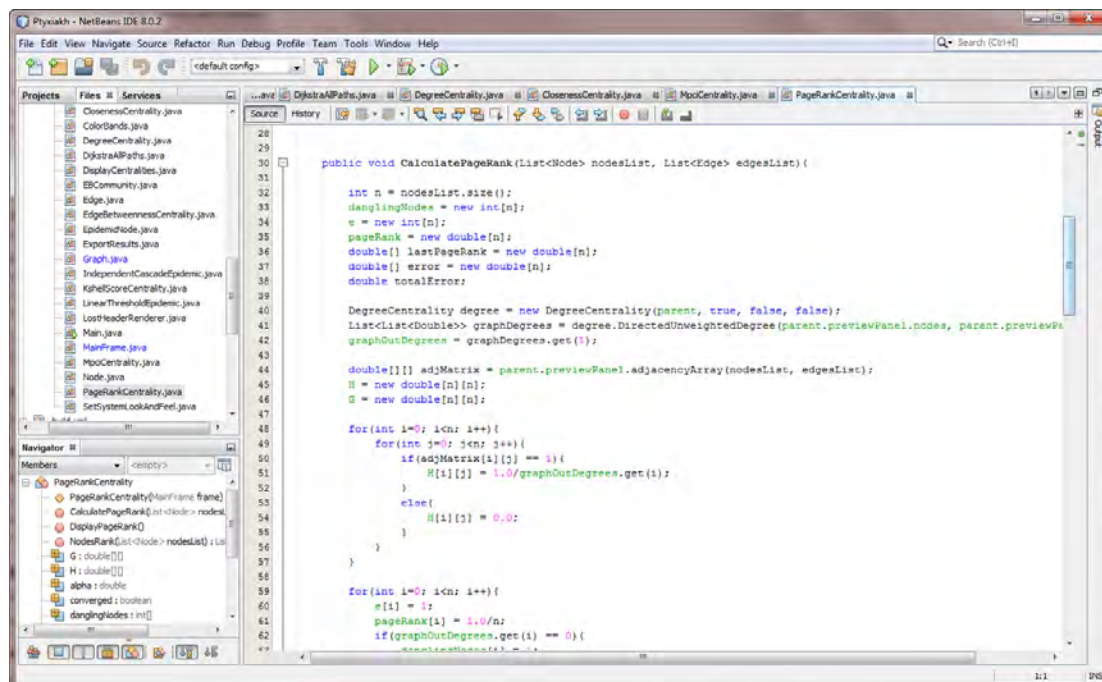
3 Το πρόγραμμα AniNet

3.1 Λογισμικά που χρησιμοποιήθηκαν

Αντικείμενο της διπλωματικής αυτής εργασίας ήταν η δημιουργία ενός λογισμικού προσομοίωσης και ανάλυσης σύνθετων δικτύων, που θα πληροί τα χαρακτηριστικά που περιγράφηκαν στην προηγούμενη παράγραφο. Η υλοποίηση του AniNet έγινε στην γλώσσα προγραμματισμού Java και στηρίχθηκε στα ακόλουθα προγράμματα.

3.1.1 NetBeans IDE

Για την συγγραφή του κώδικα χρησιμοποιήθηκε το πρόγραμμα NetBeans. Πρόκειται για ένα ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού (IDE) γραμμένο σε Java, που υποστηρίζει επίσης γλώσσες όπως PHP, C, C++ και HTML 5. Έχει την δυνατότητα να τρέξει σε Microsoft Windows, Mac OS X, Linux, Solaris, καθώς και άλλες πλατφόρμες που υποστηρίζουν ένα συμβατό Java Virtual Machine.



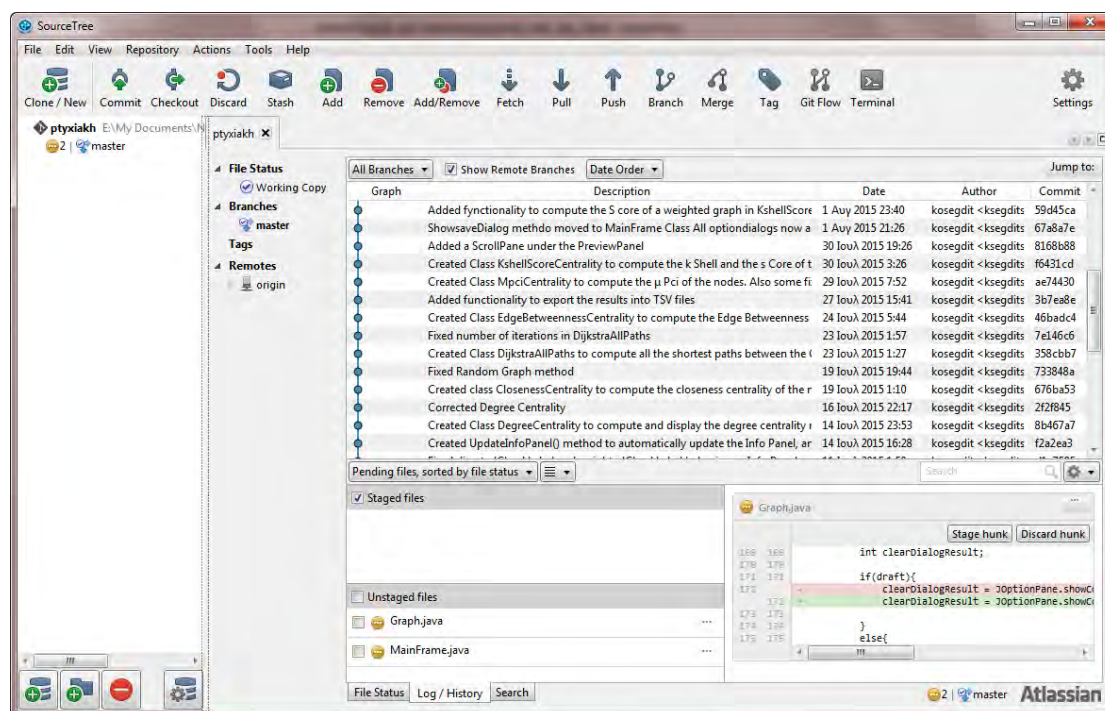
Εικόνα 7 - Το AniNet στο NetBeans

3.1.2 Git και SourceTree

Για την σωστή οργάνωση και παρακολούθηση του κώδικα, χρησιμοποιήθηκε το Git μέσα από το GUI του λογισμικού SourceTree.

Το Git είναι ένα ευρέως διαδεδομένο σύστημα ελέγχου εκδόσεων (version control system), για την ανάπτυξη κώδικα. Πρόκειται για ένα σύστημα ελέγχου με έμφαση στην ταχύτητα, την ακεραιότητα των δεδομένων, καθώς και την υποστήριξη για καταναμημένες και μη, ροές εργασίας.

Για την εκμετάλλευση των δυνατοτήτων του Git, χρησιμοποιήθηκε το λογισμικό SourceTree, το οποίο με το εξελιγμένο και άκρως φιλικό περιβάλλον χρήσης του, παρέχει την δυνατότητα παρακολούθησης των αλλαγών του κώδικα που χτίζεται στο NetBeans, σε πραγματικό χρόνο. Μέσα από την δημιουργία του κατάλληλου repository, ο χρήστης έχει την δυνατότητα να παρατηρεί τις αλλαγές που πραγματοποιούνται κατά την πορεία ανάπτυξης του κώδικά του, όπως επίσης και να τις ενσωματώνει ή να τις απορρίπτει στο περιβάλλον του NetBeans, ανάλογα με την λειτουργικότητα που προσφέρουν. Με αυτό τον τρόπο το SourceTree δημιουργεί το ανάλογο ιστορικό του κώδικα, κάτι που είναι άκρως σημαντικό για την ομαλή ανάπτυξη και την σωστή οργάνωση της εργασίας.



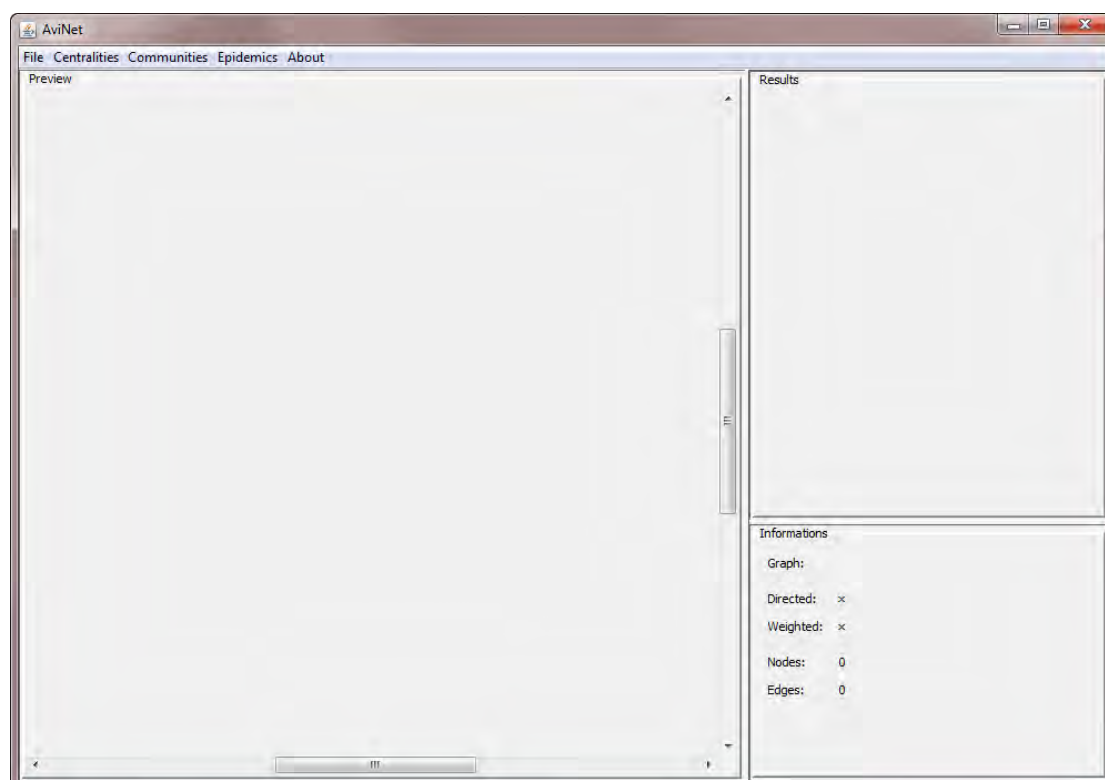
Εικόνα 8 - Πορεία εργασίας του AviNet στο SourceTree

3.2 Το περιβάλλον του AviNet

Με την εκκίνηση του AviNet, ο χρήστης έρχεται αντιμέτωπος με το κυρίως γραφικό περιβάλλον.

Στην κορυφή παρατηρείται η μπάρα με το μενού που περιλαμβάνει τις βασικές λειτουργίες για δημιουργία γραφημάτων, ανάγνωση και αποθήκευση αρχείων και εξαγωγή αποτελεσμάτων. Στο ίδιο σημείο βρίσκονται επίσης και οι απαραίτητες επιλογές για την εξαγωγή των ιδιοτήτων του εκάστοτε σύνθετου δικτύου, μέσα από τους υποστηριζόμενους αλγορίθμους του προγράμματος.

Το υπόλοιπο μέρος του παραθύρου καταλαμβάνεται από τα απαραίτητα πάνελ για την οπτική αναπαράσταση του δικτύου, την παρουσίαση των αποτελεσμάτων του κάθε αλγορίθμου, καθώς και την παρουσίαση των βασικότερων ιδιοτήτων του υπό εξέταση γραφήματος.



Εικόνα 9 - Περιβάλλον AviNet

Πιο συγκεκριμένα, το μεγαλύτερο μέρος του παραθύρου βρίσκεται στην αριστερή πλευρά και αφορά το πάνελ στο οποίο λαμβάνει χώρα η γραφική αναπαράσταση του σύνθετου δικτύου. Ο χώρος αυτός φέρει τον τίτλο “Preview”. Εκεί μπορεί επίσης ο χρήστης να δημιουργήσει το δικό του γράφημα, βήμα προς βήμα, μια λειτουργία που θα αναλυθεί εκτενέστερα στο επόμενο κεφάλαιο. Η κάθετη μπάρα που διαχωρίζει το Preview πάνελ με τα υπόλοιπα, είναι συρόμενη, δίνοντας έτσι την δυνατότητα της προσαρμογής σε οποιοδήποτε επιθυμητό σημείο.

Στο άνω δεξί κομμάτι του παραθύρου παρατηρείται ο χώρος που προορίζεται για την παρουσίαση των επιθυμητών αποτελεσμάτων, και φέρει τον τίτλο “Results”. Το Results πάνελ χωρίζεται με την σειρά του με μια οριζόντια μπάρα, επίσης συρόμενη, από το τρίτο και τελευταίο πάνελ του κυρίως παραθύρου.

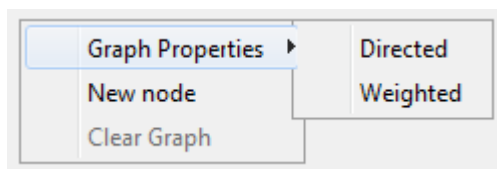
Κάτω και δεξιά βρίσκεται ο χώρος που είναι αφιερωμένος για την προβολή των σημαντικότερων πληροφοριών του γραφήματος που είναι είτε ολοκληρωμένο, είτε υπό κατασκευή, και παρουσιάζεται στο Preview πάνελ. Πιο αναλυτικά, στο πεδίο “Graph: ...” προβάλλεται το όνομα του γραφήματος που εξετάζεται από τον χρήστη, το οποίο πολλές φορές μπορεί να είναι και το πλήρες μονοπάτι στον κατάλογο του υπολογιστή, εάν το γράφημα αυτό βρίσκεται αποθηκευμένο σε ανάλογο αρχείο. Οι δύο επόμενες γραμμές δεικνύουν αν το δίκτυο είναι κατευθυνόμενο και βεβαρημένο. Σειρά παίρνει το πλήθος των κόμβων και των ακμών που απαρτίζουν το δίκτυο, ενώ σε περίπτωση που έχει ζητηθεί από τον χρήστη η εκτέλεση κάποιου αλγορίθμου, το όνομα αυτού εμφανίζεται στην τελευταία γραμμή των πληροφοριών, η οποία αρχικά είναι κενή.

4 Ξεκινώντας με το AviNet

Για να μπορέσει κανείς να εξάγει την οποιαδήποτε ιδιότητα, πρέπει αρχικά να έχει δημιουργήσει το επιθυμητό δίκτυο. Το AviNet προσφέρει πληθώρα επιλογών στον τομέα αυτό. Ο χρήστης μπορεί είτε να δημιουργήσει το δικό του γράφημα βήμα προς βήμα, είτε να φορτώσει ένα γράφημα από αρχείο που συμμορφώνεται με τους κανόνες του AviNet, είτε ακόμα να χρησιμοποιήσει τους αλγορίθμους δημιουργίας τυχαίων γραφημάτων, τα οποία στη συνέχεια μπορεί να παραμετροποιήσει.

4.1 Δημιουργία γραφημάτων βήμα προς βήμα

Η δημιουργία γραφημάτων βήμα προς βήμα, μπορεί να γίνει εύκολα χρησιμοποιώντας το Preview πάνελ. Ξεκινώντας, με δεξί κλικ σε οποιοδήποτε κενό σημείο του εν λόγω πάνελ, παρατηρούμε το μενού που εμφανίζεται με τις εξής επιλογές:



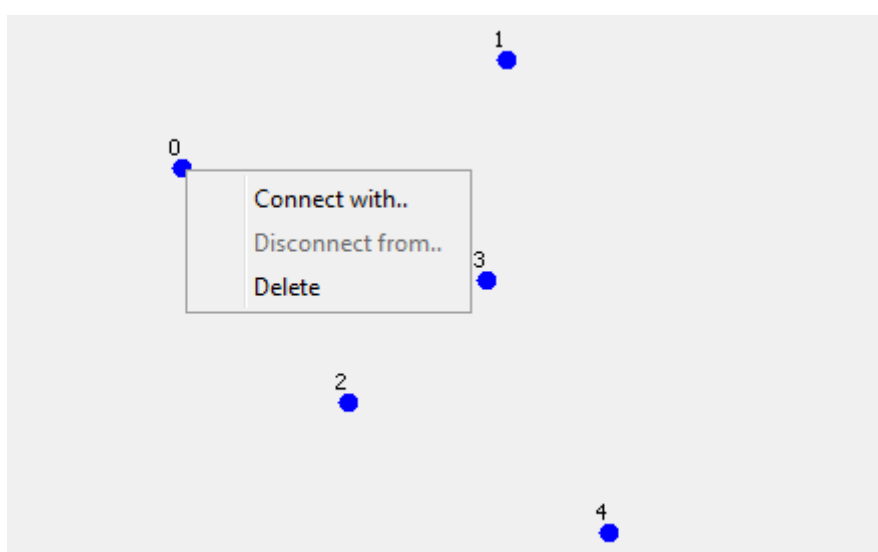
Εικόνα 10 - Preview panel popup menu

Graph Properties: Η πρώτη επιλογή που συναντάμε αφορά τις ιδιότητες του γραφήματος, και συγκεκριμένα το αν αυτό θα είναι βεβαρημένο ή κατευθυνόμενο. Εξ ορισμού οι επιλογές αυτές είναι απενεργοποιημένες και ο χρήστης μπορεί να επιλέξει οποιαδήποτε επιθυμεί. Από την στιγμή που δεν υπάρχουν ακμές στο γράφημα, και για όσο χρονικό διάστημα ισχύει αυτό, η επιλογή αυτή θα παραμένει ενεργή σε κάθε πάτημα του δεξιού κλικ. Αν ο χρήστης επιλέξει να δημιουργήσει έστω και μια ακμή, τότε το μενού “Graph Properties” καθίσταται ανενεργό, και προκειμένου να γίνει ξανά προσβάσιμο, θα πρέπει να εξαλειφθούν τυχόν υπάρχουσες ακμές.

Clear Graph: Η επιλογή Clear Graph μας δίνει την δυνατότητα να σβήσουμε ένα γράφημα και να καθαρίσουμε το Preview πάνελ. Στην προκειμένη αλλά και στην γενικότερη περίπτωση που δεν υπάρχει γράφημα φορτωμένο στο AniNet, η επιλογή αυτή είναι ανενεργή.

New node: Για την δημιουργία κόμβου, αρκεί να κάνουμε κλικ στην επιλογή “New node”. Στο επιλεγμένο σημείο θα εμφανιστεί ένας νέος κόμβος, ο οποίος θα φέρει και το ανάλογο αναγνωριστικό-όνομα, προκειμένου να ξεχωρίζει εύκολα από τους υπόλοιπους. Το αναγνωριστικό αυτό, για λόγους ευκολίας προγραμματιστικούς, αλλά και για την αποφυγή λαθών από μεριάς του χρήστη, είναι πάντα ακέραιος αριθμός και κατά 1 μεγαλύτερος από αυτόν του τελευταίου κόμβου που δημιουργήθηκε στο συγκεκριμένο γράφημα. Επαναλαμβάνοντας την διαδικασία, μπορούμε να δημιουργήσουμε περισσότερους κόμβους στις θέσεις που επιθυμούμε.

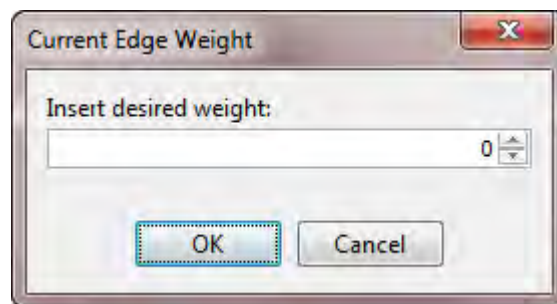
Για την δημιουργία και την διαγραφή ακμής, καθώς και την διαγραφή κόμβου, αρκεί ο χρήστης να επιλέξει τον κόμβο που επιθυμεί, κάνοντας δεξί κλικ σε αυτόν. Έτσι θα εμφανιστεί το εξής μενού:



Εικόνα 11 - Node popup menu

Connect with..: Η επιλογή αυτή δηλώνει την πρόθεση του χρήστη, να ενώσει με μια ακμή τον κόμβο που έχει αρχικά επιλέξει, με έναν ακόμα κόμβο του γραφήματος. Σε περίπτωση που υπάρχει μόνο ένας κόμβος, το “Connect with..” είναι ανενεργό, μιας και στο AniNet δεν επιτρέπονται ακμές που δημιουργούν βρόγχους και συνεπώς δεν υπάρχει η δυνατότητα δημιουργίας ακμής. Επίσης ο χρήστης θα

πρέπει να έχει αποφασίσει για το τι είδους γράφημα θέλει να κατασκευάσει, με την έννοια αν αυτό θα είναι κατευθυνόμενο ή βεβαρημένο, προκειμένου να μπορέσει να κάνει τις απαραίτητες ρυθμίσεις στο πρόγραμμα και να δημιουργήσει τις ανάλογες ακμές. Επιλέγοντας το “Connect with..” το μενού εξαφανίζεται και το πρόγραμμα περιμένει από τον χρήστη να κάνει αριστερό κλικ στον κόμβο που θέλει να ενώσει, προκειμένου τελικά να δημιουργήσει την επιθυμητή ακμή. Σε περίπτωση που ο χρήστης επιθυμεί να κατασκευάσει βεβαρημένο γράφημα, αφού επιλεγεί ο έτερος κόμβος, εμφανίζεται το παρακάτω παράθυρο για την επιλογή του επιθυμητού βάρους της ακμής. Τα βάρη που επιτρέπονται στο AniNet, είναι μη αρνητικοί αριθμοί, με έως και τρία δεκαδικά ψηφία.



Εικόνα 12 - Επιλογή βάρους ακμής

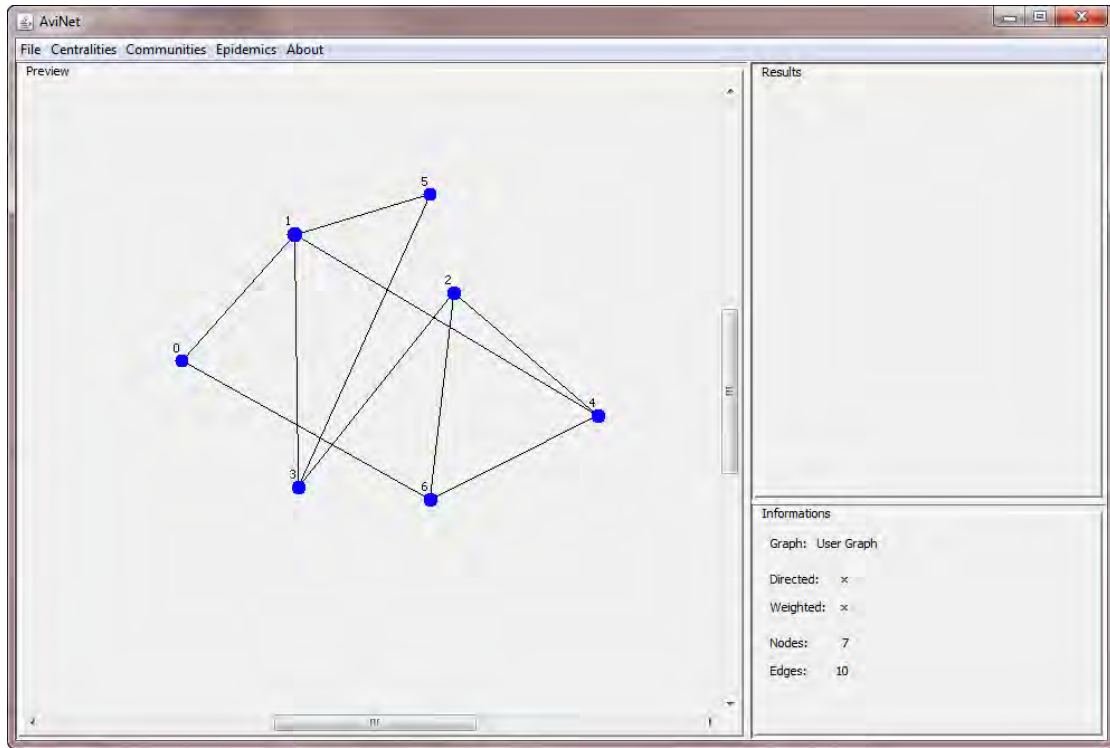
Disconnect from..: Αν υπάρχουν εξερχόμενες ακμές από τον επιλεγμένο κόμβο, τότε η επιλογή “Disconnect from..” είναι ενεργή και δίνει την δυνατότητα στον χρήστη να τις σβήσει. Μετά το πάτημά της, το μενού εξαφανίζεται και το πρόγραμμα περιμένει την επιλογή του επόμενου κόμβου, ώστε να εξαλείψει την ακμή που τους ενώνει.

Delete: Τέλος, η επιλογή “Delete”, χρησιμοποιείται για την διαγραφή του επιλεγμένου κόμβου από το δίκτυο.

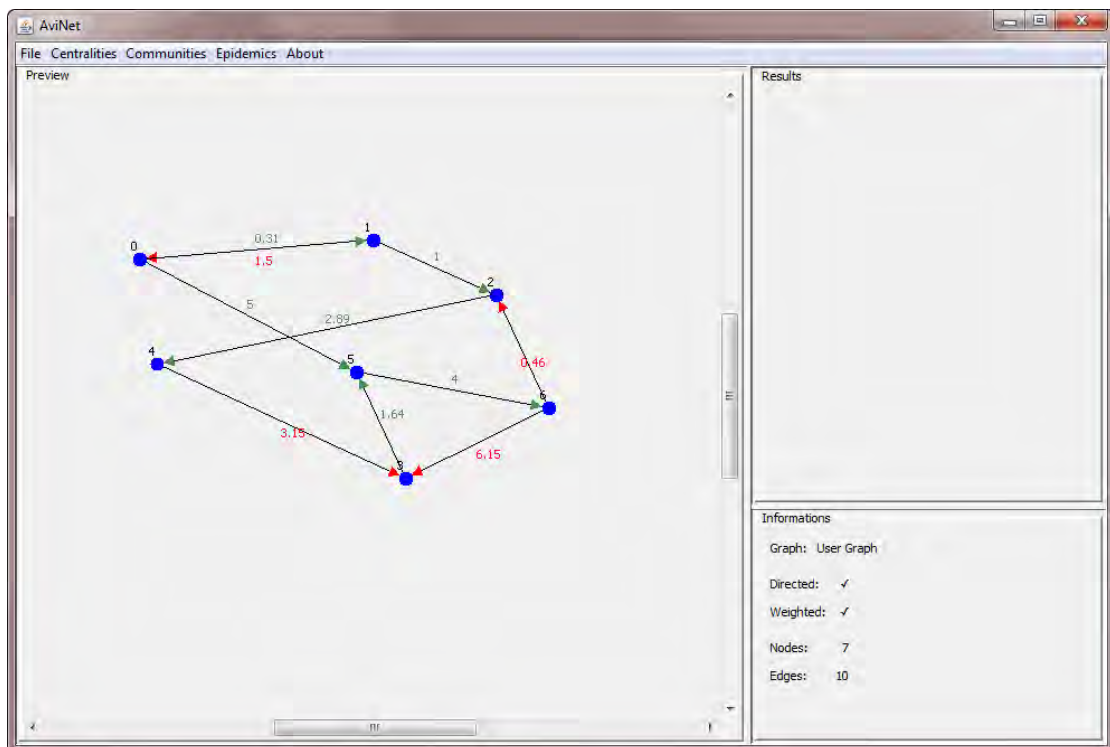
4.1.1 Παραδείγματα γραφημάτων

Οι εικόνες που ακολουθούν παρουσιάζουν δύο παραδείγματα γραφημάτων που έχουν δημιουργηθεί βήμα προς βήμα, με την διαδικασία που περιγράφεται παραπάνω, και έχουν σκοπό την κατανόηση και την επίδειξη της γραφικής αναπαράστασης γραφημάτων από το AniNet. Αξίζει να προσέξει κανείς πως στην δεύτερη περίπτωση όπου το γράφημα είναι κατευθυνόμενο και βεβαρημένο, οι

κατευθύνσεις και τα βάρη των ακμών, από μαύρα χρωματίζονται κόκκινα και πράσινα, προκειμένου να γίνεται κατανοητή η αντιστοιχία τους.



Εικόνα 13 - Παράδειγμα μη κατευθυνόμενου, μη βεβαρημένου γραφήματος



Εικόνα 14 - Παράδειγμα κατευθυνόμενου, βεβαρημένου γραφήματος

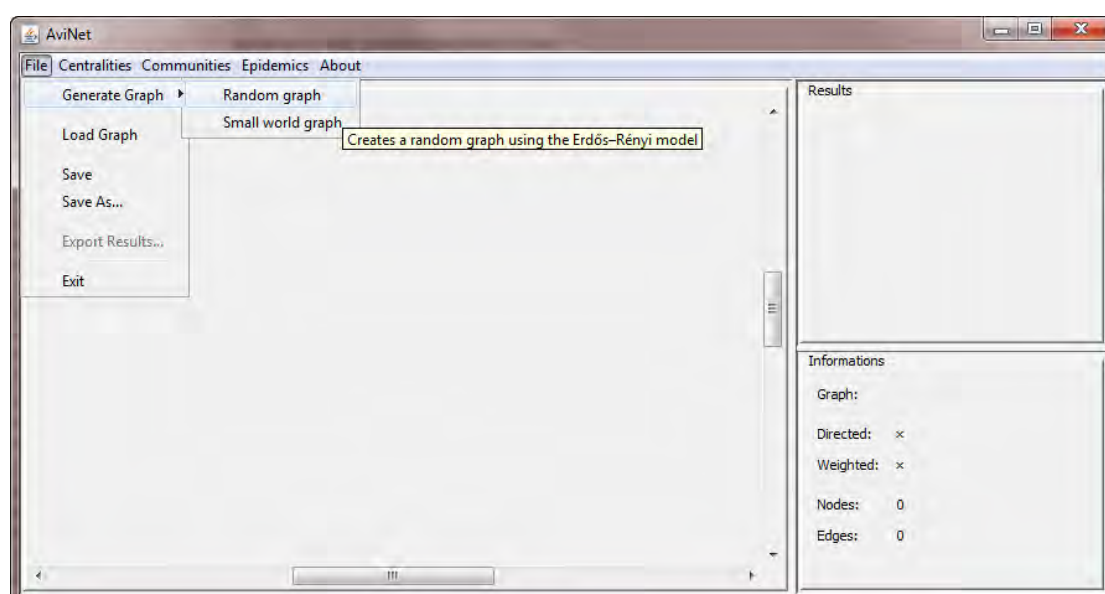
4.2 Δημιουργία γραφημάτων μέσω αλγορίθμων

Το AviNet παρέχει την δυνατότητα της απευθείας δημιουργίας γραφημάτων, μέσα από δύο διαφορετικούς αλγορίθμους. Ο χρήστης μπορεί να επιλέξει να δημιουργήσει ένα Random Graph, ή ένα Small World Graph, τα οποία αν θέλει μπορεί να παραμετροποιήσει επιπλέον με τα εργαλεία που περιγράφηκαν στην προηγούμενη παράγραφο.

4.2.1 Random Graph

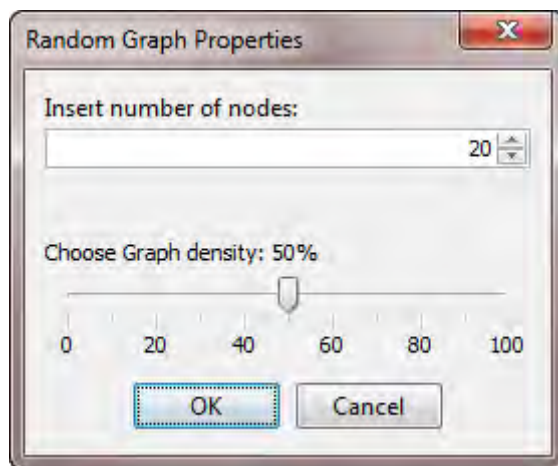
Το AviNet παράγει τυχαία γραφήματα (Random Graph), βασισμένα στο μοντέλο Erdős-Rényi. Στην θεωρία των γραφημάτων, το μοντέλο Erdős-Rényi περιλαμβάνει δύο στενά συσχετιζόμενα μοντέλα για την δημιουργία τυχαίων γραφημάτων. Πήρε το όνομα του από τους Paul Erdős και Alfréd Rényi, οι οποίοι εισήγαγαν για πρώτη φορά ένα από τα δύο μοντέλα το 1959. Το δεύτερο εισήχθη ανεξάρτητα από τον Edgar Gilbert. Σύμφωνα με το πρώτο, το μοντέλο των Erdős-Rényi, μια ακμή δημιουργείται ανάμεσα σε ένα οποιοδήποτε ζευγάρι κόμβων, με ανεξάρτητη πιθανότητα.

Στο AviNet για την δημιουργία ενός Random Graph, αρκεί ο χρήστης να ακολουθήσει το μενού “File → Generate Graph → Random graph”, όπως φαίνεται και στην εικόνα 15.



Εικόνα 15 - Επιλογή Random Graph

Κάνοντας κλικ στην επιλογή Random Graph, εμφανίζεται το παρακάτω μενού, προκειμένου ο χρήστης να δώσει τις επιθυμητές παραμέτρους για την δημιουργία του Random Graph.

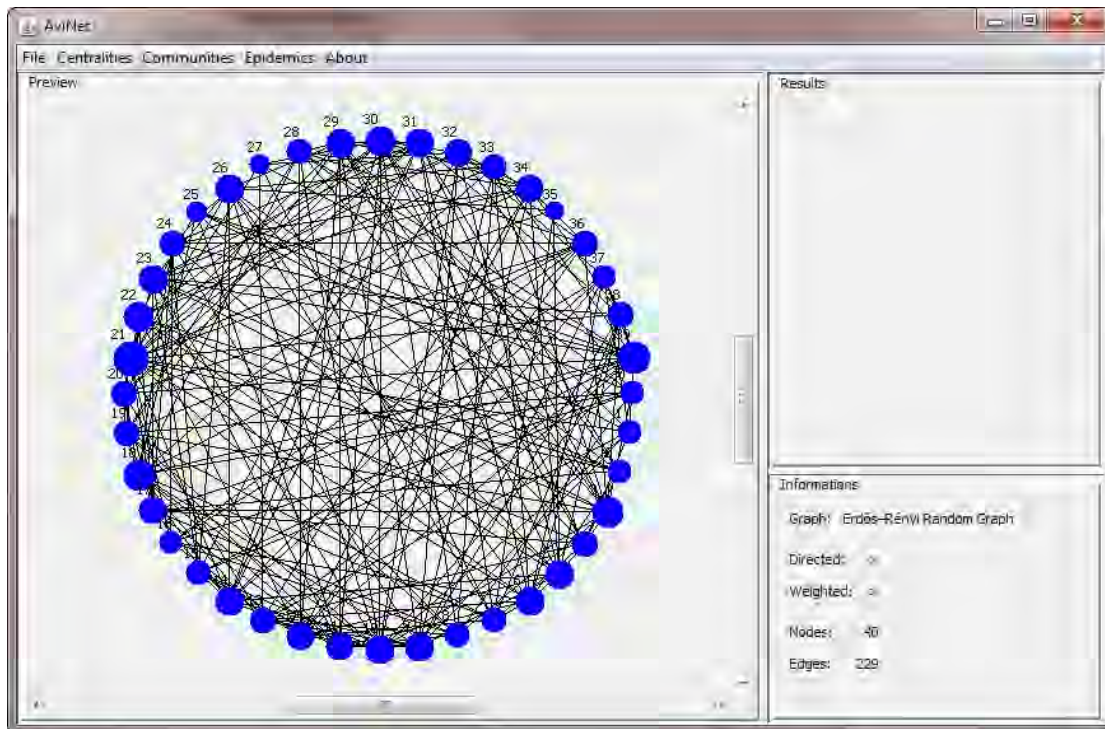


Εικόνα 16 - Παράμετροι εισόδου ενός Random Graph

Η πρώτη επιλογή αφορά το πλήθος των κόμβων που θα περιλαμβάνει το γράφημα, με ελάχιστη τιμή 1, αφού η δημιουργία γραφήματος χωρίς κόμβους δεν θα είχε κανένα απολύτως νόημα. Ο χρήστης μπορεί είτε να πληκτρολογήσει έναν ακέραιο αριθμό, είτε να χρησιμοποιήσει τα βέλη του μετρητή. Η δεύτερη επιλογή αφορά την επιθυμητή πυκνότητα του γραφήματος, και πραγματοποιείται σύροντας το βέλος αριστερά και δεξιά. Με άλλα λόγια η τιμή αυτή αφορά την πιθανότητα για ένα ζευγάρι κόμβων να συνδέεται με ακμή, συνεπώς όσο μεγαλύτερη είναι η πυκνότητα, τόσο περισσότερες ακμές θα δημιουργηθούν στο γράφημα. Οι αρχικές τιμές των δύο αυτών παραμέτρων, είναι 20 για το πλήθος των κόμβων, και 50% για την πυκνότητα του γραφήματος.

4.2.1.1 Παράδειγμα δημιουργίας Random Graph στο AviNet

Στην εικόνα 17 παρουσιάζεται ένα παράδειγμα δημιουργίας Random Graph στο λογισμικό AviNet. Στην περίπτωση αυτή, επιλέχθηκαν 40 αρχικοί κόμβοι, με πυκνότητα γραφήματος 30%. Έτσι δημιουργήθηκε ένα μη κατευθυνόμενο, μη βεβαρημένο γράφημα, 40 κόμβων και 229 ακμών.



Εικόνα 17 - Erdős-Rényi Random Graph

4.2.2 Small World Graph

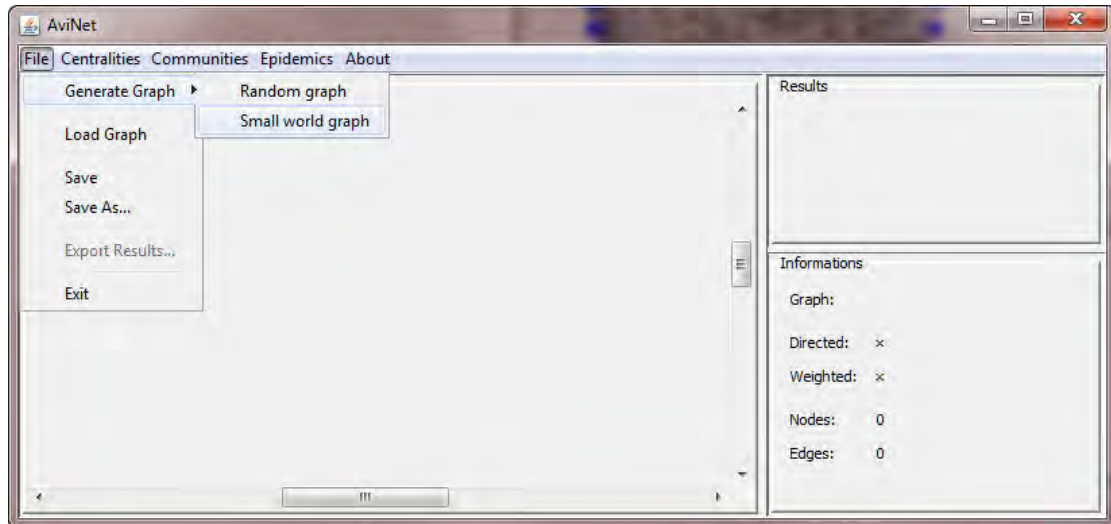
Ένα Small World Graph είναι ένα γράφημα στο οποίο οι περισσότεροι κόμβοι δεν είναι γείτονες μεταξύ τους, μπορούν όμως να προσεγγιστούν από τους υπόλοιπους με μικρό πλήθος βημάτων (hops).

Τα απαιτούμενα δεδομένα για την δημιουργία ενός Small World γραφήματος είναι το πλήθος των κόμβων (L), η πιθανότητα διατήρησης των αρχικών ακμών (p), και τέλος το πλήθος των γειτόνων του κάθε κόμβου (Z), όπου το Z είναι ζυγός, ακέραιος αριθμός, μικρότερος ή ίσος με το 15% του L ($Z \leq 0.15 * L$).

Η διαδικασία κατασκευής ενός τέτοιου γραφήματος, είναι η εξής: Αρχικά, κάθε ένας από τους L κόμβους συνδέεται με τους $Z/2$ κόμβους στα αριστερά του, και με τους $Z/2$ κόμβους στα δεξιά του. Έπειτα, επαναληπτικά διατρέχεται κάθε ακμή για κάθε κόμβο του γραφήματος, και της αντιστοιχίζεται μια τυχαία πιθανότητα. Αν η πιθανότητα αυτή, είναι μεγαλύτερη ή ίση από την πιθανότητα διατήρησης των αρχικών ακμών p , τότε το πρόγραμμα επιλέγει έναν τυχαίο κόμβο του γραφήματος, και αναπροσαρμόζει την επιλεγμένη ακμή στον κόμβο αυτό. Ο αλγόριθμος δεν

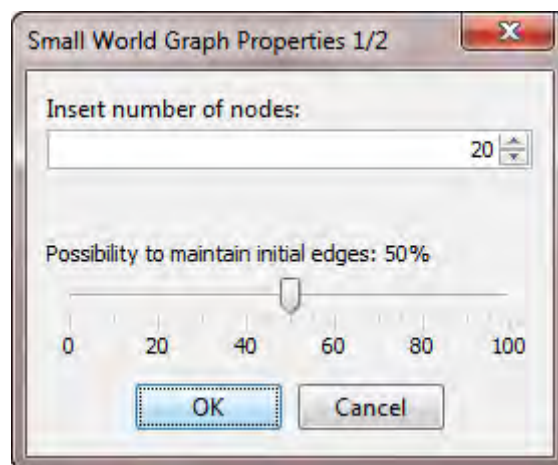
επιτρέπει την δημιουργία βρόγχων ή πολλαπλών ακμών, μιας και τα δύο αυτά χαρακτηριστικά δεν υποστηρίζονται εξ αρχής από το AviNet.

Για την δημιουργία ενός Small World Graph, αρκεί να ακολουθήσει κανείς το μενού “File → Generate Graph → Small world graph”, όπως δείχνει και η εικόνα 18.



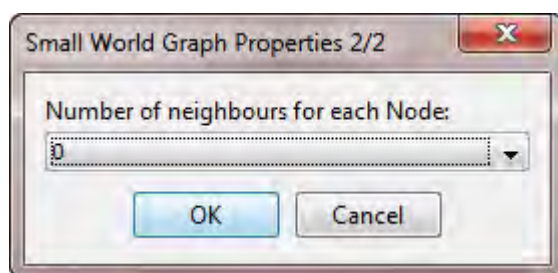
Εικόνα 18 - Επιλογή Small world graph

Μετά την επιλογή δημιουργίας ενός Small world γραφήματος, ακολουθεί η εμφάνιση δύο παραθύρων που ζητούν από τον χρήστη να δώσει τις απαραίτητες εισόδους για την κατασκευή του γραφήματος.



Εικόνα 19 - Παράμετροι εισόδου Small world graph 1/2

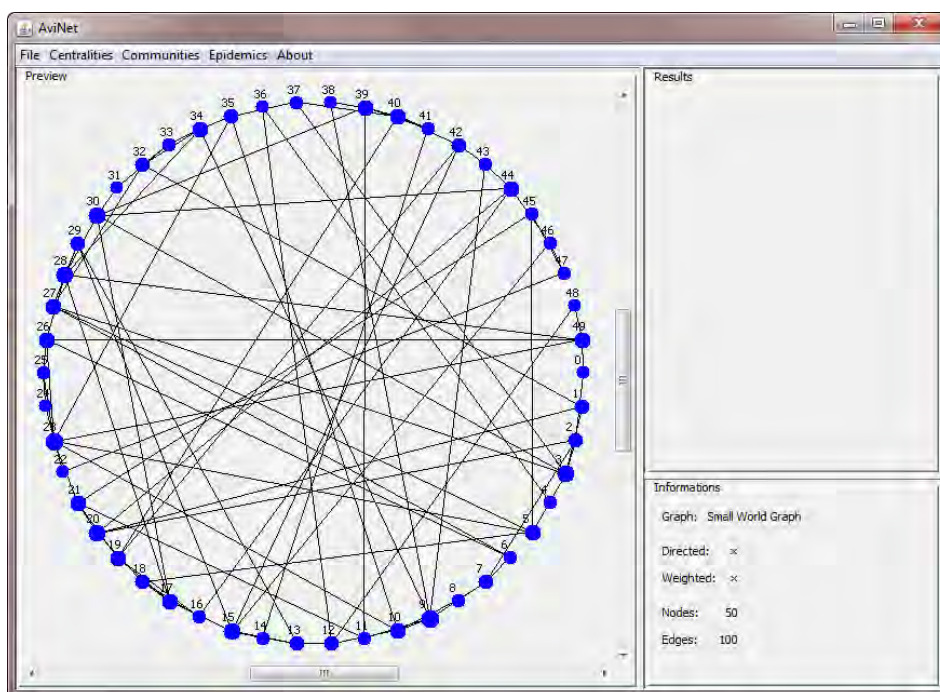
Αρχικά εμφανίζεται το παράθυρο της εικόνας 19, παραπλήσιο με εκείνο που χρησιμοποιείται για το Random graph. Με τον ίδιο τρόπο, όπως και πριν, ο χρήστης καλείται να επιλέξει τον επιθυμητό αριθμό των κόμβων του γραφήματος (L), και στην συνέχεια την πιθανότητα διατήρησης των αρχικών ακμών (p). Αφού γίνουν οι επιλογές αυτές, εμφανίζεται το δεύτερο και τελευταίο παράθυρο, στο οποίο ο χρήστης καλείται να επιλέξει το πλήθος των γειτόνων (Z) κάθε κόμβου του γραφήματος.



Εικόνα 20 - Παράμετροι εισόδου Small world graph 2/2

4.2.2.1 Παράδειγμα δημιουργίας Small world graph στο AviNet

Για αρχικό πλήθος κόμβων 50, πιθανότητα διατήρησης των αρχικών ακμών 60% και πλήθος γειτόνων για κάθε κόμβο ίσο με 4, δημιουργήθηκε το ακόλουθο Small world γράφημα στο AviNet, όπως φαίνεται στην εικόνα 21.



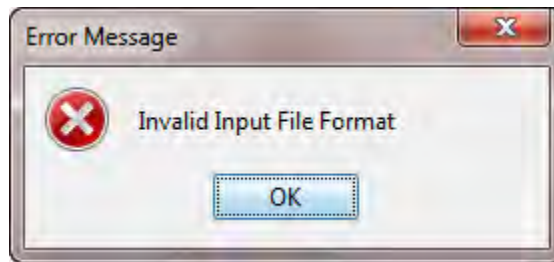
Εικόνα 21 - Small World graph

4.3 Διάβασμα αρχείων για την δημιουργία γραφημάτων

Εκτός από την δυνατότητα δημιουργίας γραφημάτων βήμα προς βήμα ή μέσω των αλγορίθμων της προηγούμενης παραγράφου, το AniNet παρέχει επίσης την λειτουργία της ανάγνωσης αρχείων. Ο χρήστης, κατά την διαδικασία κατασκευής γραφήματος σε αρχείο, οφείλει να συμμορφώνεται με τους παρακάτω κανόνες, προκειμένου το AniNet να είναι σε θέση να ολοκληρώσει σωστά την ανάγνωση του αρχείου και να κατασκευάσει επιτυχώς το επιθυμητό γράφημα.

- Τα αρχεία κειμένου οφείλουν να φέρουν την κατάληξη .cnt
- Κάθε γραμμή που ξεκινά με δίσηση (#) αναγνωρίζεται ως σχόλιο, και συνεπώς δεν λαμβάνεται υπόψιν.
- Για την κατασκευή κατευθυνόμενου γραφήματος, είναι απαραίτητη η ύπαρξη γραμμής που ξεκινά με το αναγνωριστικό “directed”, και χωρίζεται από την λέξη “true” μέσω ενός tab. Σε περίπτωση που η γραμμή αυτή δεν υπάρχει στο αρχείο ή τελειώνει με “false”, το πρόγραμμα θα παράγει ένα μη κατευθυνόμενο γράφημα.
- Ομοίως για βεβαρημένα γραφήματα, είναι απαραίτητη η ύπαρξη γραμμής που ξεκινά με το αναγνωριστικό “weighted”, ακολουθούμενο από ένα tab και την λέξη “true”. Αν λείπει η συγκεκριμένη γραμμή ή τελειώνει με “false”, θα παραχθεί μη βεβαρημένο γράφημα. Επίσης αν η επιλογή για βεβαρημένο γράφημα δεν συνάδει με την ύπαρξη ή όχι βαρών, τότε το πρόγραμμα θα αποτύχει στην δημιουργία του γραφήματος.
- Κάθε άλλη γραμμή του αρχείου, οφείλει να αναγράφει αρχικά το όνομα του κάθε κόμβου, να ακολουθεί ένα tab και στην συνέχεια να αναγράφεται η επιθυμητή γεωγραφική του τοποθεσία στο Preview πάνελ, ξεκινώντας από το σύμβολο “@”. Μιας και κάτι τέτοιο είναι δύσκολο, θα ήταν θεμιτό το σύμβολο “@” να μην ακολουθείται από τίποτα, ώστε το πρόγραμμα να δώσει δικές του συντεταγμένες στους κόμβους. Έπειτα, χωρισμένοι με tabs αναγράφονται οι γείτονες του τρέχοντος κόμβου ώστε να δημιουργηθούν οι ακμές, και αν το γράφημα είναι βεβαρημένο, μετά το όνομα του γείτονα πρέπει να βρίσκεται ο χαρακτήρας “,” συνοδευόμενος από το βάρος της αντίστοιχης ακμής.

Σε περίπτωση που το αρχείο δεν συμμορφώνεται με τους παραπάνω κανόνες, τότε η διαδικασία της ανάγνωσης αποτυγχάνει, και το πρόγραμμα εμφανίζει το παρακάτω μήνυμα λάθους.



Εικόνα 22 - Ανεπιτυχής ανάγνωση αρχείου εισόδου

Στις εικόνες που ακολουθούν, φαίνονται πρότυπα αρχεία εισόδου, που βοηθούν στην κατανόηση των προαναφερθέντων κανόνων.

A screenshot of a Notepad++ window titled "E:\Desktop\input_example.cnt - Notepad++ [Admi...". The window shows a text file with the following content:

```
1 # Ιδιότητες γραφήματος
2
3 directed true
4 weighted true
5
6 # Λίστα κόμβων και ακμών
7
8 0 @ 1,3 5,6 4,2
9 1 @ 2,0 4,1 3,8
10 2 @ 0,6
11 3 @ 4,5 5,1
12 4 @
13 5 @ 2,6 4,9
```

The status bar at the bottom indicates "Ln:5 Col:1 Sel:0 | Dos\Windows UTF-8 w/o BOM INS".

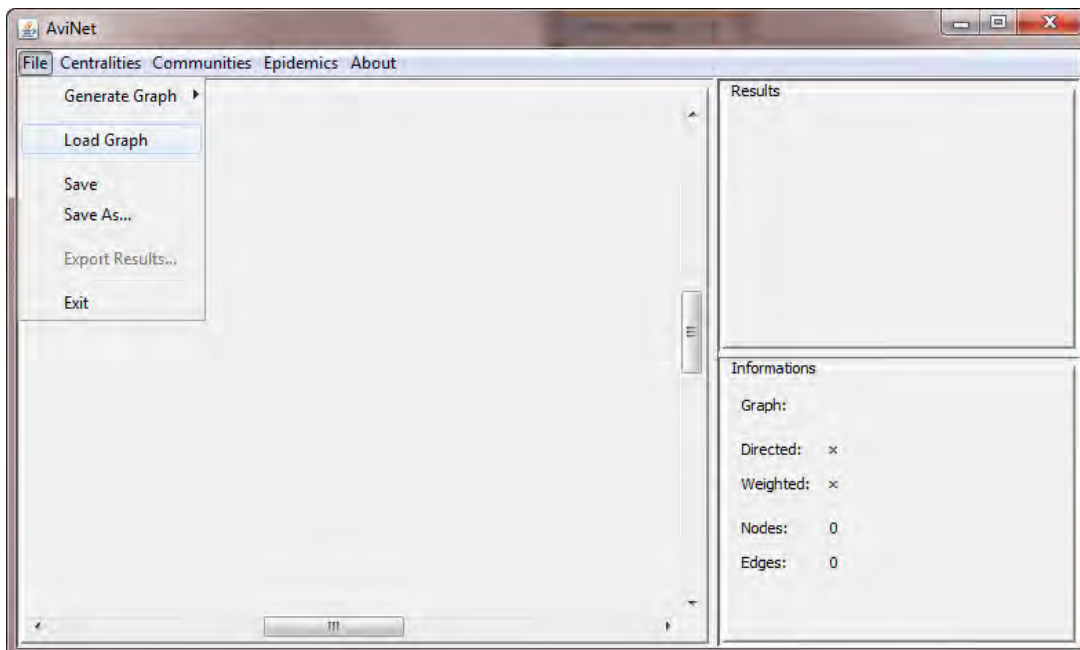
Εικόνα 23 - Πρότυπο αρχείο εισόδου για την κατασκευή κατευθυνόμενου βεβαρημένου γραφήματος

```
1 # Ιδιότητες γραφήματος
2
3 directed      false
4 weighted     false
5
6 # Λίστα κόμβων και ακμών
7
8 0 @ 1 5 4
9 1 @ 2 4 3
10 2 @ 0
11 3 @ 4 5
12 4 @
13 5 @ 2 4
```

Ln:13 Col:14 Sel:| Dos\Windows UTF-8 w/o BOM INS

Εικόνα 24 - Πρότυπο αρχείο εισόδου για την κατασκευή μη κατευθυνόμενου μη βεβαρημένου γραφήματος

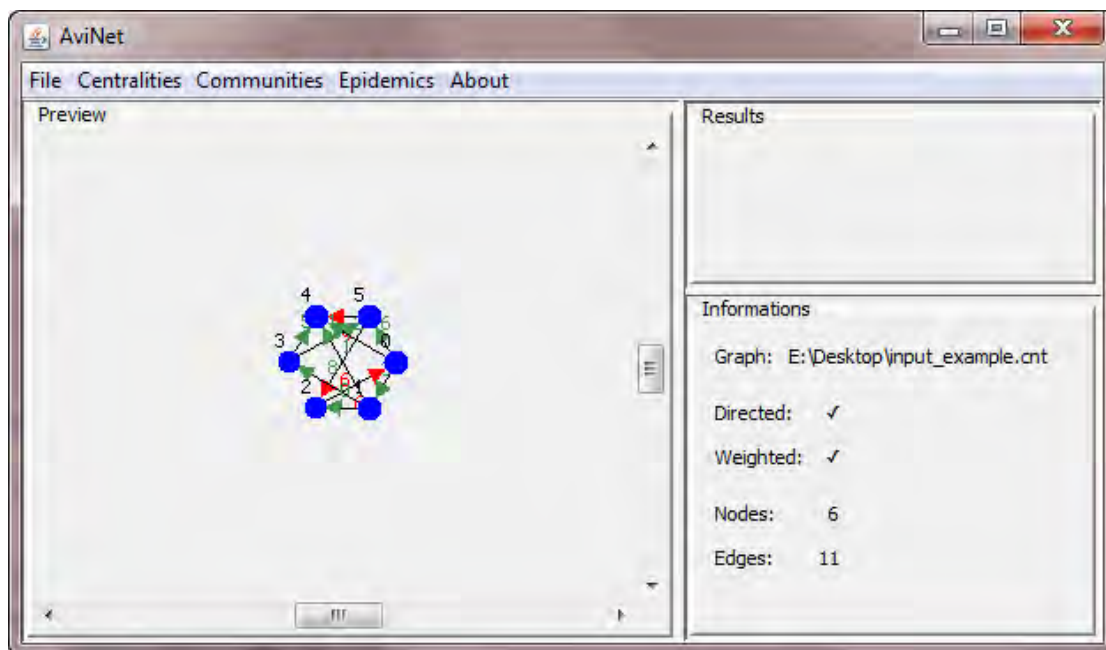
Για την ανάγνωση αρχείων με σκοπό την δημιουργία γραφημάτων, ο χρήστης μπορεί να ακολουθήσει το μενού “File → Load Graph”, όπως φαίνεται στην εικόνα 25.



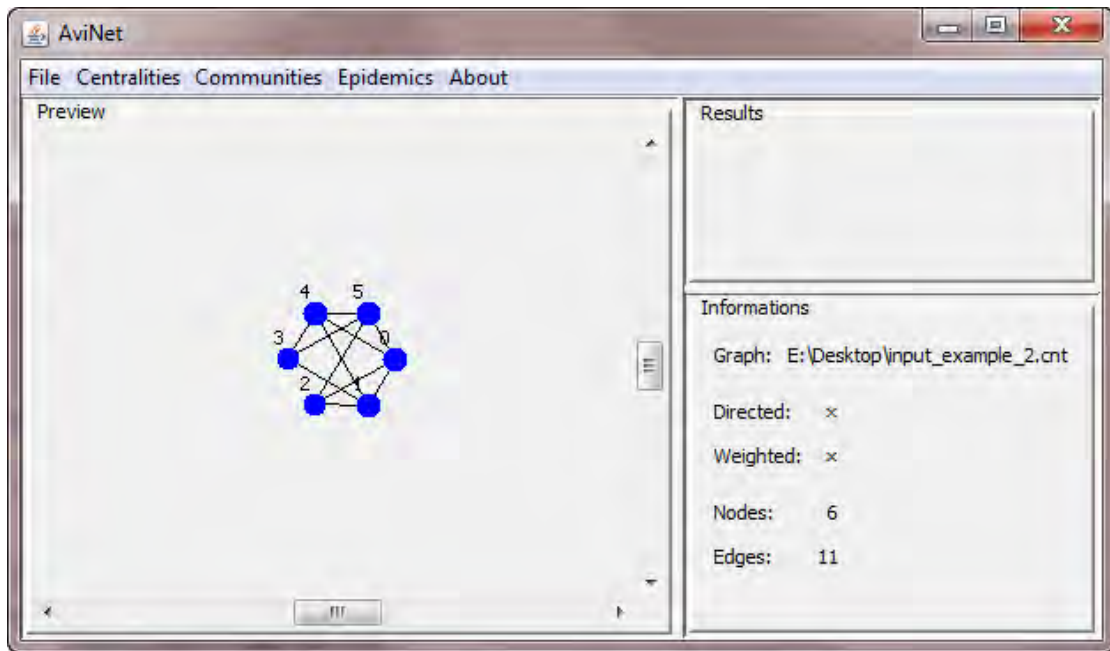
Εικόνα 25 - Μενού για την ανάγνωση γραφημάτων

Με την επιλογή του μενού Load Graph, εμφανίζεται το κατάλληλο παράθυρο διαλόγου, μέσα από το οποίο ο χρήστης μπορεί να αναζητήσει το αρχείο που επιθυμεί, και στην συνέχεια να το φορτώσει στο AviNet για την αναπαράσταση του γραφήματος. Το AviNet έχει προγραμματιστεί έτσι ώστε εκτός από τους φακέλους, να εμφανίζει αποκλειστικά τα αρχεία με κατάληξη .cnt, ώστε να διευκολύνει την περιήγηση και να αποκλείει την επιλογή αρχείων που δεν είναι συμβατά.

Οι εικόνες 26 και 27 αναπαριστούν την εικόνα του προγράμματος μετά την ανάγνωση των αρχείων των εικόνων 23 και 24 αντίστοιχα. Αξίζει να παρατηρήσει κανείς πως από την στιγμή που ο χρήστης δεν έδωσε δικές του συντεταγμένες στους κόμβους, το λογισμικό ανέλαβε να τους παρατάξει από μόνο του σε κυκλική διάταξη. Μετά την φόρτωση του αρχείου, μπορούμε εύκολα να μετακινήσουμε τους κόμβους και τις ακμές στο σημείο που επιθυμούμε.



Εικόνα 26 - Αναπαράσταση γραφήματος εικόνας 22



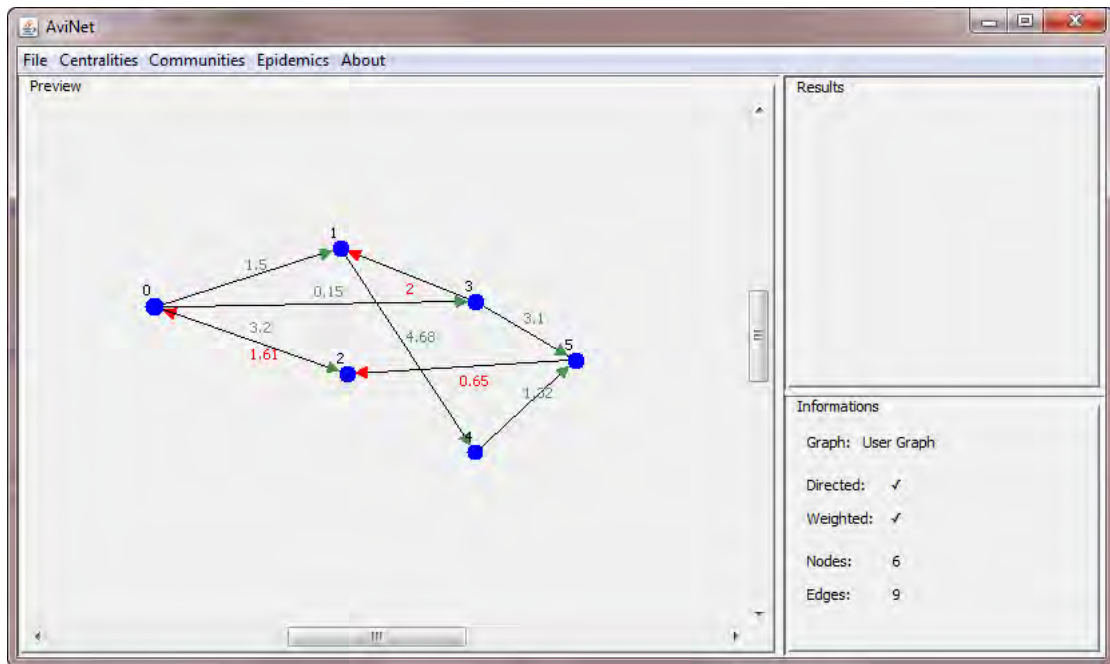
Εικόνα 27 - Αναπαράσταση γραφήματος εικόνας 23

4.4 Αποθήκευση γραφημάτων για μελλοντική χρήση

Από την στιγμή που ο χρήστης θα δημιουργήσει το επιθυμητό γράφημα, το AviNet του παρέχει την δυνατότητα να το αποθηκεύσει σε ένα αρχείο .cnt, προκειμένου να είναι σε θέση να το χρησιμοποιήσει ξανά οποιαδήποτε στιγμή εκείνος επιθυμεί, χωρίς την ανάγκη να το δημιουργήσει από την αρχή.

Για την αποθήκευση του γραφήματος, αρκεί κανείς να ακολουθήσει το μενού “File → Save”, αν το αρχείο υφίσταται ήδη και απλά έχει υποστεί επεξεργασία, ή “File → Save As..”, αν θέλει να δημιουργήσει ένα καινούριο. Το απαραίτητο παράθυρο διαλόγου θα εμφανιστεί, και το μόνο που απομένει είναι ο χρήστης να επιλέξει την τοποθεσία που επιθυμεί να αποθηκεύσει το αρχείο, όπως επίσης και το όνομα του. Το όνομα του αρχείου δεν είναι απαραίτητο να φέρει την κατάληξη .cnt, μιας και το AviNet θα την προσθέσει από μόνο του.

Δημιουργώντας ένα γράφημα και στην συνέχεια αποθηκευόντάς το, μπορεί κανείς να παρατηρήσει την μορφή του εξαγόμενου αρχείου. Για το παράδειγμα του γραφήματος της εικόνας 28, μετά την αποθήκευσή του, προέκυψε το αρχείο της εικόνας 29.



Εικόνα 28 - Παράδειγμα γραφήματος που θα αποθηκευτεί

Μετά την αποθήκευση του παραπάνω γραφήματος, με όνομα “example”, το αρχείο “example.cnt” έχει την ακόλουθη μορφή.

```

1 #Graph Properties
2
3 directed    true
4 weighted    true
5
6
7 #List of Nodes, with their Locations and their Edges
8
9 0  @778,839  1,1.5  2,3.2  3,0.15
10 1  @918,796  4,4.68
11 2  @923,890  0,1.61
12 3  @1019,836 1,2.0  5,3.1
13 4  @1019,949 5,1.32
14 5  @1094,880 2,0.65
15

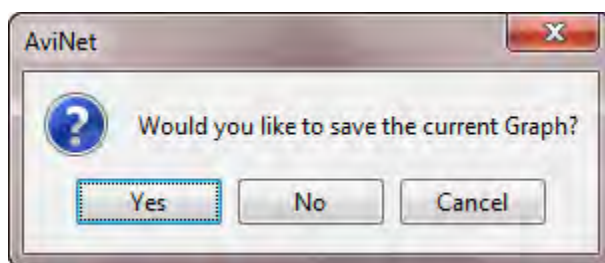
```

The screenshot shows a Notepad++ window titled "E:\Desktop\example.cnt - Notepad++ [Administrator]". The window contains the following text:

Εικόνα 29 - example.cnt

Διαβάζοντας το αρχείο, αξίζει να παρατηρήσουμε τους αριθμούς δίπλα στο σύμβολο “@”, οι οποίοι υποδηλώνουν την γεωγραφική τοποθεσία κάθε κόμβου του γραφήματος. Σε περίπτωση που θελήσουμε να φορτώσουμε ξανά το συγκεκριμένο γράφημα στο AviNet, το λογισμικό θα διαβάσει τις συγκεκριμένες τοποθεσίες και έτσι θα αναπαραστήσει το γράφημα όπως ακριβώς φαίνεται στην εικόνα 28.

Τέλος, στην περίπτωση κατά την οποία στο πρόγραμμα υπάρχει γράφημα το οποίο έχει παραμετροποιηθεί, δίχως την αποθήκευση των τελευταίων αλλαγών του, και ο χρήστης προσπαθήσει είτε να φορτώσει άλλο γράφημα, είτε να σβήσει το παρών από το Preview πάνελ, είτε ακόμα να τερματίσει το πρόγραμμα δίχως την αποθήκευση του, το AviNet θα εμφανίσει το παρακάτω μήνυμα, ρωτώντας τον χρήστη αν επιθυμεί να προχωρήσει πρώτα στην αποθήκευση των αλλαγών.



Εικόνα 30 - Προτροπή αποθήκευσης γραφήματος

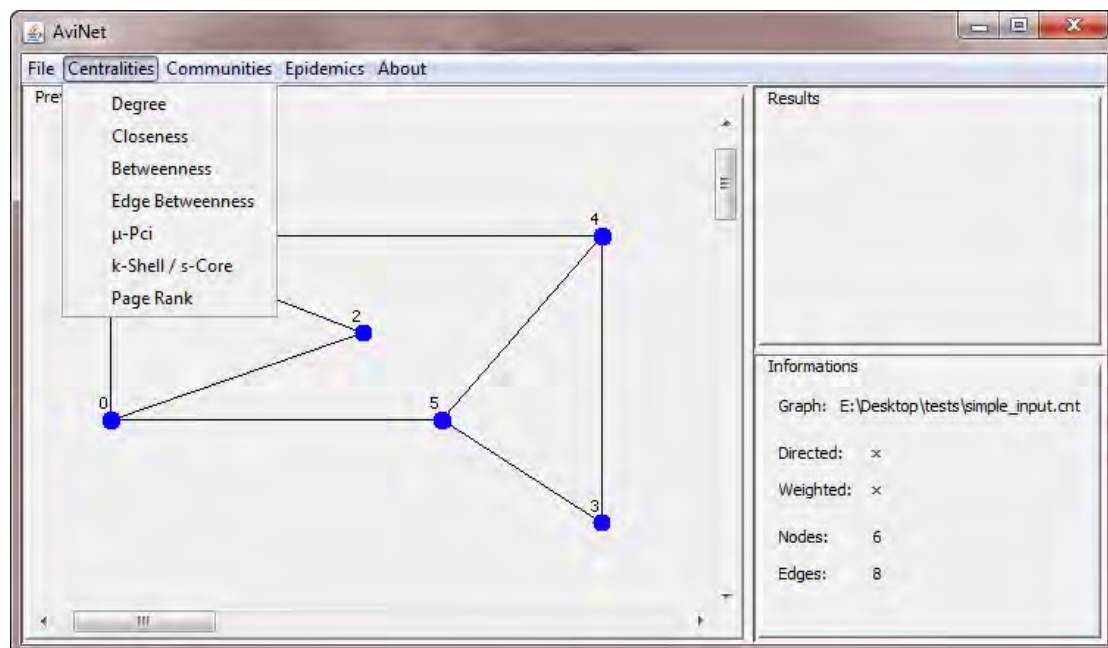
Η λειτουργία αυτή, λειτουργεί ως δίχτυ ασφαλείας, εξαλείφοντας την πιθανότητα απώλειας των αλλαγών που έγιναν σε ένα γράφημα, λόγω παράλειψης του χρήστη.

5 Κεντρικότητες στο AviNet

Η πρώτη κατηγορία αλγορίθμων ανάλυσης σύνθετων δικτύων αφορά τις κεντρικότητες (Centralities). Πιο συγκεκριμένα, στο AviNet υποστηρίζονται οι εξής αλγόριθμοι κεντρικότητας, που δύναται να εφαρμοστούν σε οποιοδήποτε τύπο γραφήματος αυτοί ορίζονται.

1. Degree Centrality
2. Closeness Centrality
3. Betweenness Centrality
4. Edge Betweenness Centrality
5. μ -PCI
6. k-Shell / s-Core
7. Page Rank

Για να μπορέσει κανείς να εφαρμόσει τους αλγορίθμους κεντρικότητας στο AviNet, και αφού πρώτα έχει δημιουργήσει ένα γράφημα με τους τρόπους που παρουσιάστηκαν στο κεφάλαιο 4, αρκεί να επιλέξει από το μενού την επιλογή “Centralities”. Έτσι θα εμφανιστεί το υπομενού της κατηγορίας αυτής, παρουσιάζοντας τις διαθέσιμες επιλογές.

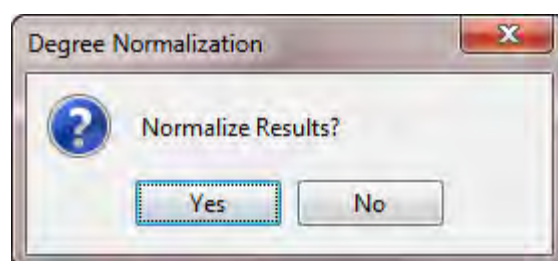


Εικόνα 31 – Centralities

5.1 Degree Centrality

Ιστορικά πρώτος και εννοιολογικά απλούστερος αλγόριθμος στην κατηγορία των κεντρικότητων, είναι αυτός της κεντρικότητας βαθμού (Degree Centrality). Μπορεί να εφαρμοστεί σε οποιοδήποτε τύπο γραφήματος και ορίζεται ως ο αριθμός των συνδέσεων που προσπίπτουν πάνω σε έναν κόμβο.

Στο AniNet τα αποτελέσματα για το Degree Centrality, προσφέρονται τόσο σε κανονικοποιημένη, όσο και σε μη κανονικοποιημένη μορφή. Μετά την επιλογή του μενού “Centralities → Degree”, εμφανίζεται το ακόλουθο παράθυρο, μέσα από το οποίο ο χρήστης μπορεί να επιλέξει αν επιθυμεί την εμφάνιση των αποτελεσμάτων σε κανονικοποιημένη μορφή.



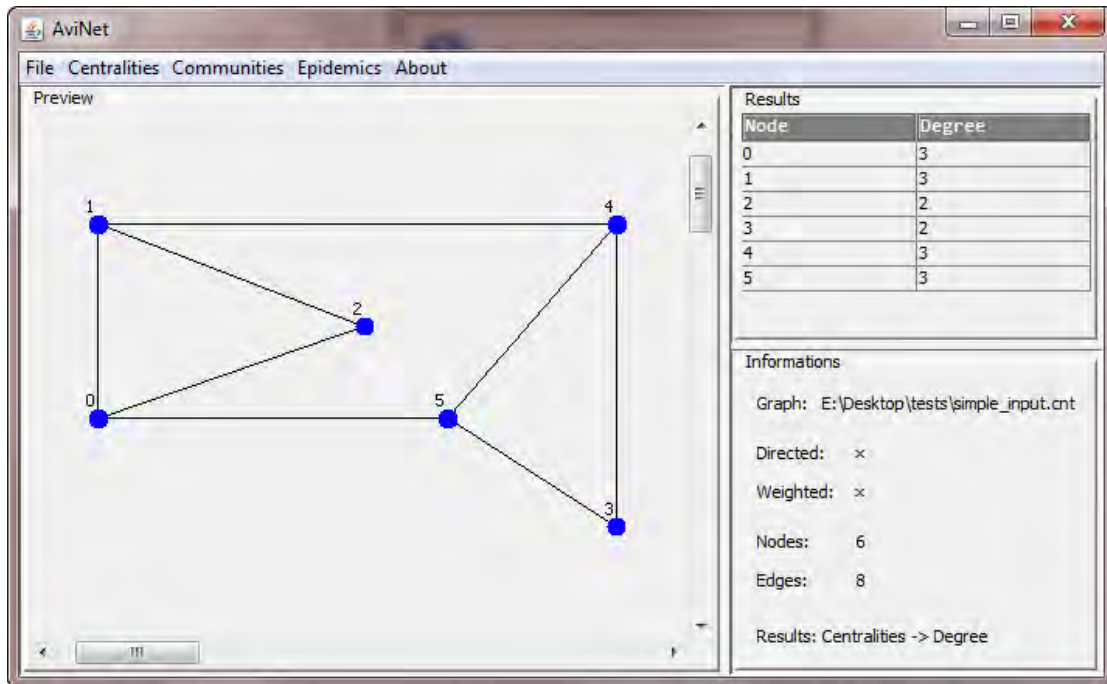
Εικόνα 32 - Παράθυρο κανονικοποίησης αποτελεσμάτων

5.1.1 Μη κατευθυνόμενα, μη βεβαρημένα γραφήματα

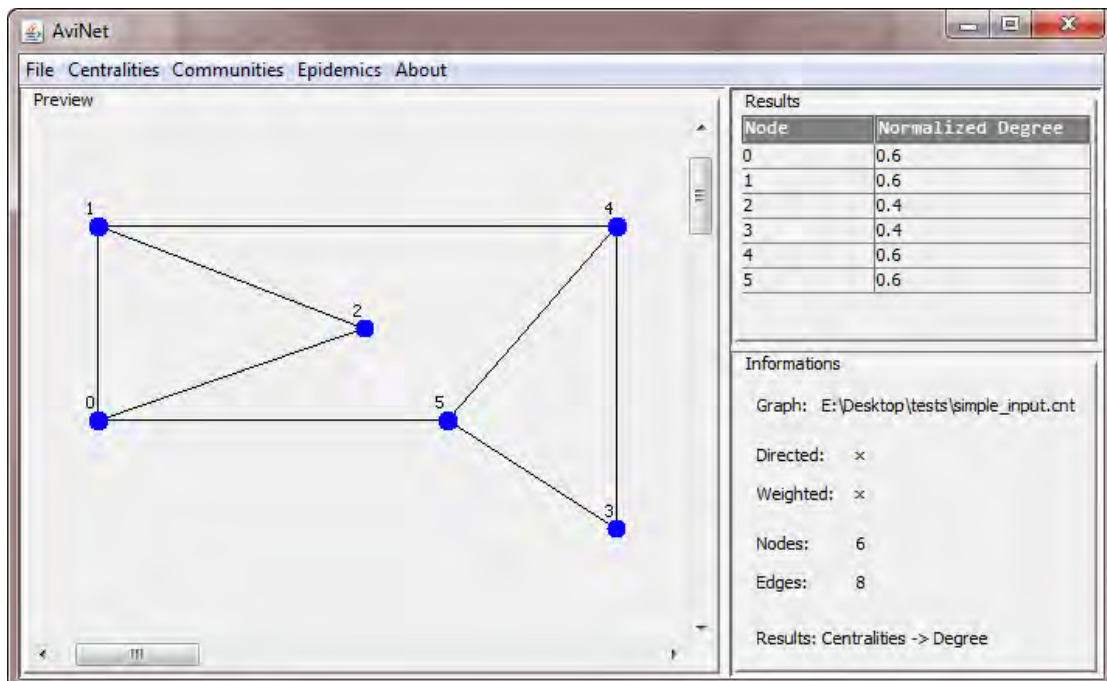
Στα μη κατευθυνόμενα, μη βεβαρημένα γραφήματα, το Degree Centrality ενός κόμβου χαρακτηρίζεται από έναν και μόνο δείκτη, η τιμή του οποίου αναπαριστά τον αριθμό των γειτόνων του.

Για τον υπολογισμό του Degree Centrality σε ένα undirected, unweighted δίκτυο, το AniNet υπολογίζει για κάθε κόμβο του γραφήματος, το πλήθος των ακμών που προσπίπτουν σε αυτόν. Τα μη κανονικοποιημένα αποτελέσματα αντικατοπτρίζουν τον βαθμό του κάθε κόμβου, ενώ τα κανονικοποιημένα προκύπτουν από τον βαθμό του κόμβου διαιρεμένο με το $(n-1)$, όπου το n είναι το συνολικό πλήθος των κόμβων του γραφήματος.

Τα παραδείγματα των εικόνων 33 και 34 δείχνουν τα αποτελέσματα για το Degree Centrality και τον τρόπο εμφάνισής τους στο Results πάνελ. Για μη κανονικοποιημένα αποτελέσματα η δεξιά στήλη φέρει τον τίτλο “Degree”, ενώ για κανονικοποιημένα φέρει τον τίτλο “Normalized Degree”.



Εικόνα 33 - Μη κανονικοποιημένο Degree σε undirected unweighted δίκτυο



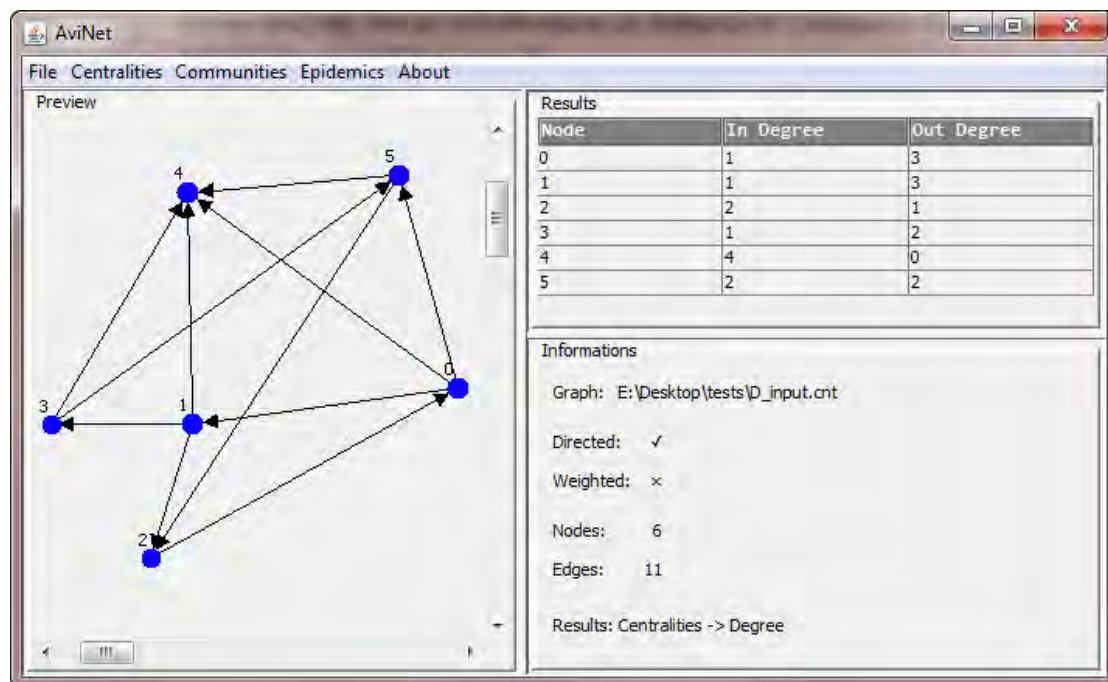
Εικόνα 34 - Κανονικοποιημένο Degree σε undirected unweighted δίκτυο

5.1.2 Κατευθυνόμενα, μη βεβαρημένα γραφήματα

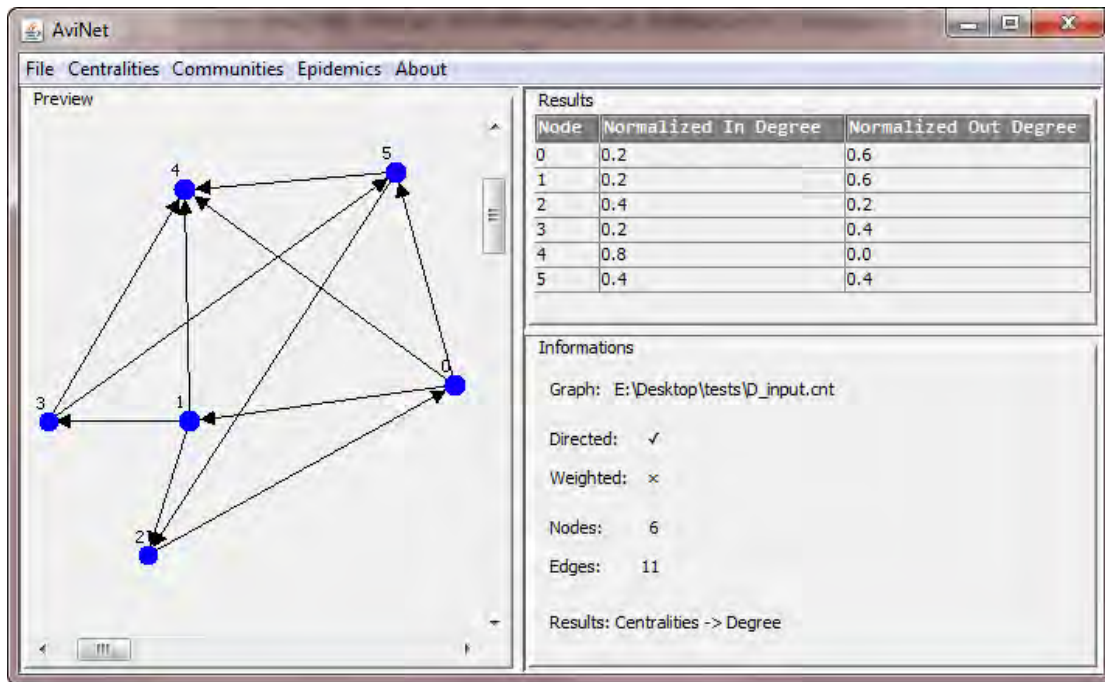
Για την περίπτωση των κατευθυνόμενων μη βεβαρημένων γραφημάτων, το Degree Centrality αντικατοπτρίζει τους βαθμούς εισόδου και εξόδου της κάθε κορυφής.

Ο υπολογισμός του βαθμού εισόδου πραγματοποιείται αθροίζοντας τις εισερχόμενες ακμές για κάθε κορυφή, ενώ αντίστοιχα ο βαθμός εξόδου υπολογίζεται από το πλήθος των εξερχόμενων ακμών. Η κανονικοποίηση των αποτελεσμάτων γίνεται όπως και στα μη κατευθυνόμενα μη βεβαρημένα γραφήματα, διαιρώντας δηλαδή όλα τα αποτελέσματα με $(n-1)$.

Στις εικόνες 35 και 36, παρουσιάζεται η εικόνα του AviNet μετά τον υπολογισμό του Degree Centrality για το ίδιο κατευθυνόμενο, μη βεβαρημένο γράφημα, σε μη κανονικοποιημένη και κανονικοποιημένη μορφή αντίστοιχα.



Εικόνα 35 - Μη κανονικοποιημένο Degree σε directed unweighted δίκτυο



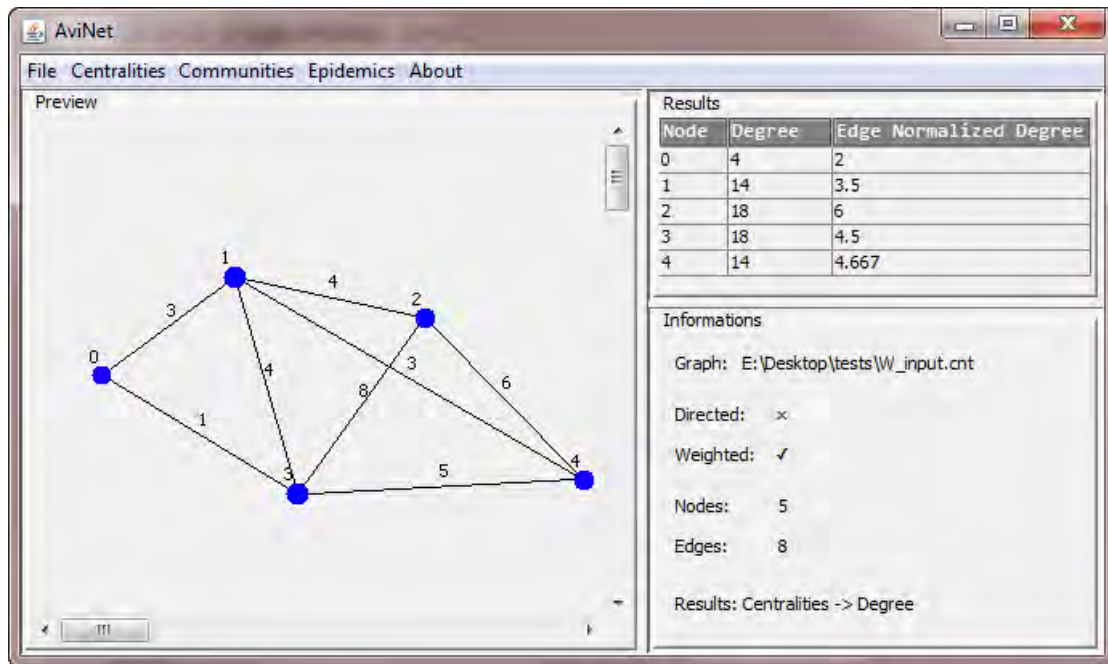
Εικόνα 36 - Κανονικοποιημένο Degree σε directed unweighted δίκτυο

5.1.3 Μη κατευθυνόμενα, βεβαρημένα γραφήματα

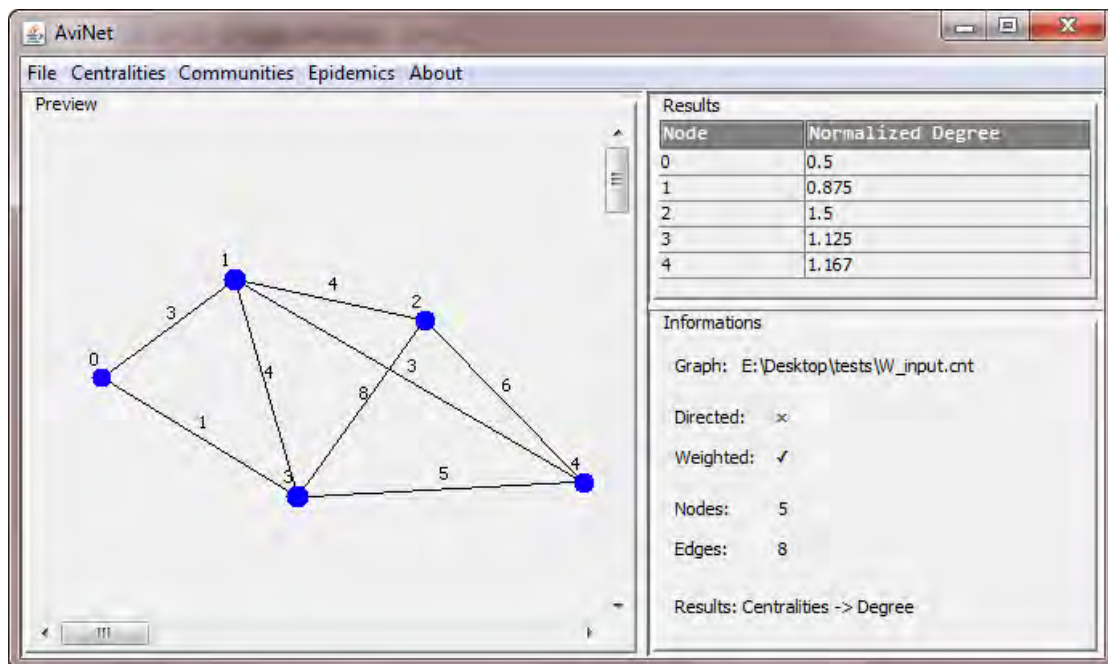
Στα μη κατευθυνόμενα βεβαρημένα γραφήματα, το Degree Centrality αντικατοπτρίζει το συνολικό άθροισμα των βαρών των ακμών που προσπίπτουν σε κάθε κορυφή.

Για τον υπολογισμό του, όπως και στις προηγούμενες κατηγορίες, το AviNet διατρέχει τις ακμές κάθε κόμβου του γραφήματος αθροίζοντας τα βάρη των ακμών. Στην περίπτωση των μη κανονικοποιημένων αποτελεσμάτων, παρέχεται ένας ακόμα δείκτης ο οποίος συμβολίζει τον βαθμό της κάθε κορυφής διαιρεμένο με το πλήθος των ακμών που προσπίπτουν σε αυτήν. Ο δείκτης αυτός φέρει τον τίτλο "Edge Normalized Degree". Αν ο χρήστης ζητήσει την κανονικοποίηση των αποτελεσμάτων, τότε αυτή θα γίνει διαιρώντας τον δείκτη "Edge Normalized Degree" με την ποσότητα $(n-1)$.

Τα παραδείγματα των εικόνων 37 και 38 περιλαμβάνουν τα αποτελέσματα υπολογισμού του Degree Centrality για ένα μη κατευθυνόμενο, βεβαρημένο γράφημα, σε μη κανονικοποιημένη και κανονικοποιημένη μορφή αντίστοιχα.



Εικόνα 37 - Μη κανονικοποιημένο Degree σε undirected weighted δίκτυο



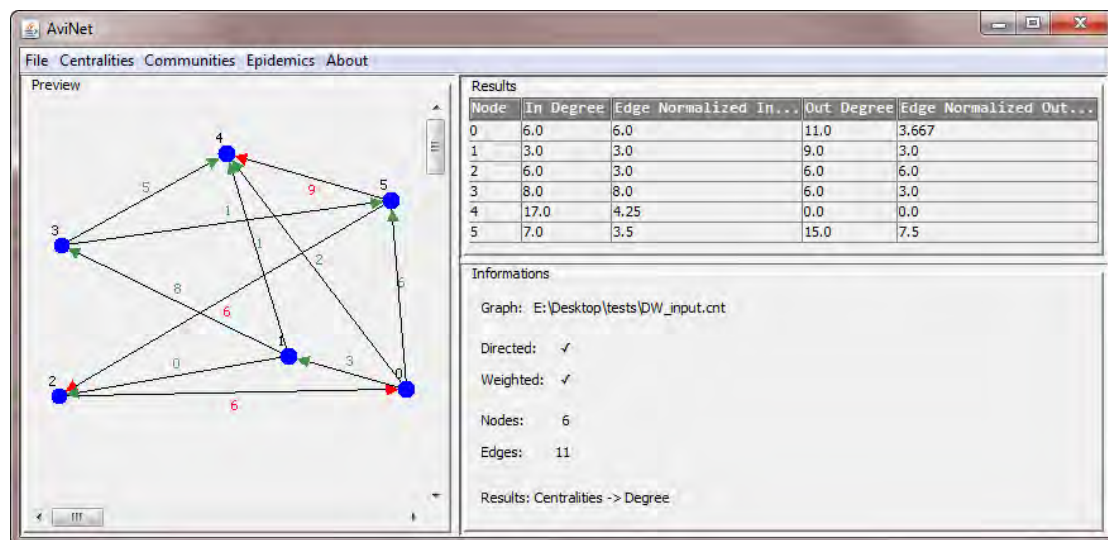
Εικόνα 38 - Κανονικοποιημένο Degree σε undirected weighted δίκτυο

5.1.4 Κατευθυνόμενα, βεβαρημένα γραφήματα

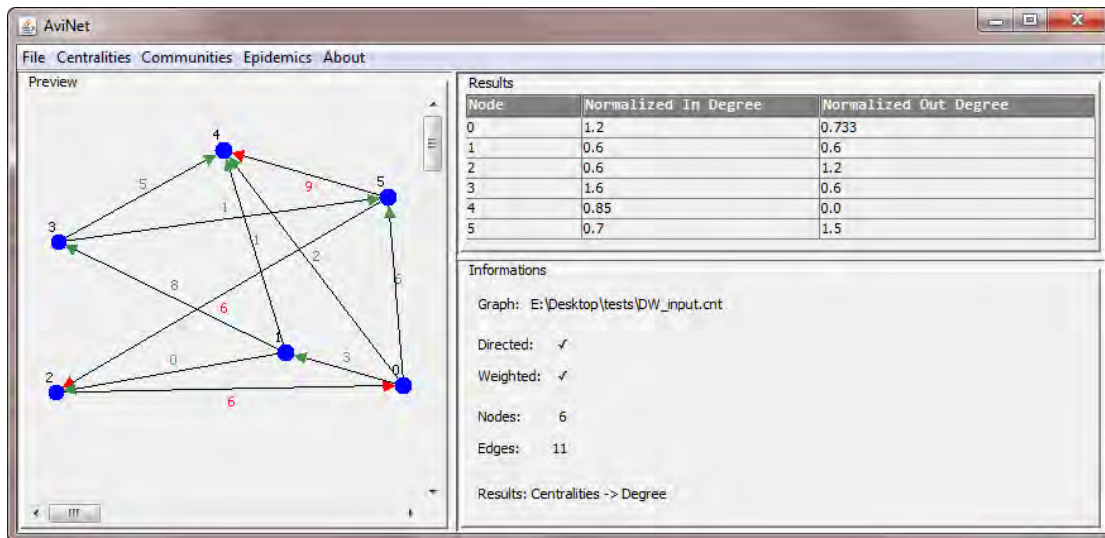
Στον τελευταίο τύπο γραφημάτων, τα κατευθυνόμενα βεβαρημένα γραφήματα, το Degree Centrality δείχνει το συνολικό βάρος των ακμών εισόδου και εξόδου για κάθε κορυφή του γραφήματος.

Το AviNet υπολογίζει το Degree Centrality αυτής της κατηγορίας γραφημάτων, διατρέχοντας για κάθε κόμβο όλες τις εισερχόμενες και εξερχόμενες ακμές, προσθέτοντας τα βάρη τους και βρίσκοντας έτσι τον βαθμό εισόδου και εξόδου αντίστοιχα. Όπως και στην προηγούμενη κατηγορία, έτσι και στα κατευθυνόμενα βεβαρημένα γραφήματα, παρέχονται επιπλέον δείκτες. Για τον υπολογισμό του “Edge Normalized In Degree”, ο βαθμός εισόδου διαιρείται με το πλήθος των εισερχομένων ακμών, ενώ αντίστοιχα το “Edge Normalized Out Degree” προκύπτει διαιρώντας τον βαθμό εξόδου με το πλήθος των εξερχομένων ακμών.

Στις εικόνες 39 και 40 παρουσιάζονται παραδείγματα υπολογισμού του Degree Centrality για ένα γράφημα της κατηγορίας αυτής, σε μη κανονικοποιημένη και κανονικοποιημένη μορφή αντίστοιχα.



Εικόνα 39 - Μη κανονικοποιημένο Degree σε directed weighted δίκτυο



Εικόνα 40 - Κανονικοποιημένο Degree σε directed weighted δίκτυο

5.2 Closeness Centrality

Στα συνδεδεμένα γραφήματα η απόσταση μεταξύ των κόμβων ορίζεται από το μήκος των ελαχίστων μονοπατιών που τους συνδέει. Η κεντρικότητα εγγύτητας (Closeness Centrality) ενός κόμβου, βασίζεται στο μέσο μήκος των ελαχίστων μονοπατιών μεταξύ του ιδίου και όλων των υπολοίπων κόμβων του γραφήματος.

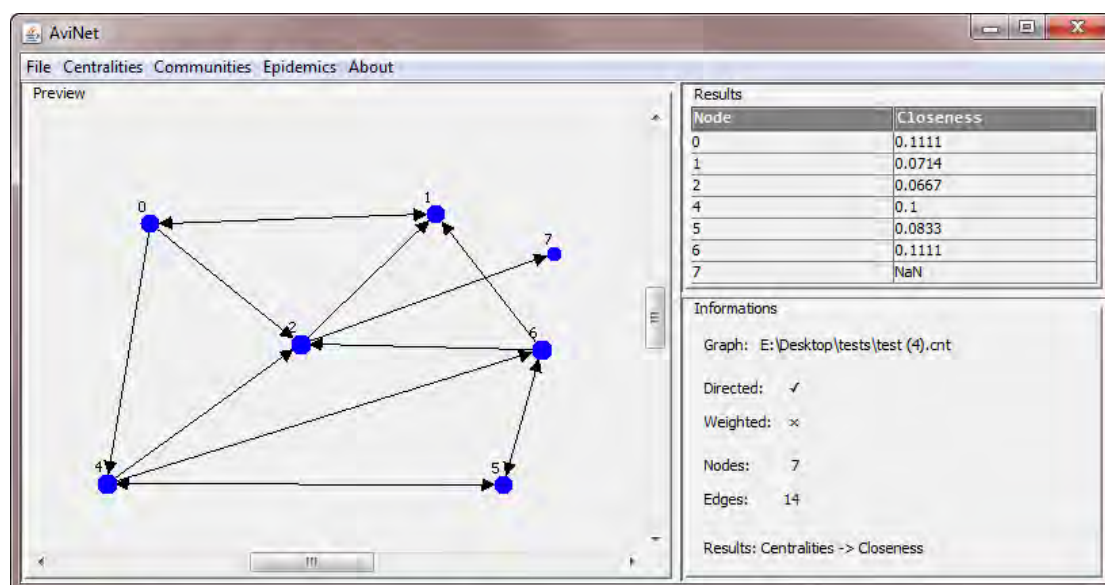
$$C_c(i) = \left[\sum_{j=1}^n d(i,j) \right]^{-1} \quad (5.2.1)$$

Προκειμένου να υπολογίσει κανείς την κανονικοποιημένη Closeness Centrality, αρκεί να την διαιρέσει με το πλήθος των κόμβων του γραφήματος μειωμένο κατά 1 (n-1).

$$C'_c(i) = \frac{C_c(i)}{(n-1)} \quad (5.2.2)$$

Για τον υπολογισμό της Closeness Centrality, το AviNet χρησιμοποιεί τον πίνακα γειτνίασης του γραφήματος, στον οποίο εφαρμόζει τον αλγόριθμο του Floyd. Η εύρεση του πίνακα γειτνίασης και συνεπώς της Closeness Centrality μπορεί να πραγματοποιηθεί με πανομοιότυπο τρόπο για οποιοδήποτε τύπο γραφήματος, ανεξάρτητα δηλαδή από το αν αυτό είναι κατευθυνόμενο ή βεβαρημένο, υποθέτοντας ότι ένα μη βεβαρημένο γράφημα ισοδυναμεί με ένα βεβαρημένο του οποίου όλες οι ακμές έχουν βάρος 1. Αφού κατασκευαστεί ο πίνακας γειτνίασης του γραφήματος, εφαρμόζεται ο αλγόριθμος του Floyd ο οποίος δύναται να υπολογίσει τα μήκη των συντομότερων μονοπατιών από κάθε πηγή του γραφήματος προς όλες τις υπόλοιπες. Τέλος τα μήκη των συντομότερων μονοπατιών που ξεκινούν από κάθε πηγή, προστίθενται μεταξύ τους, και το αποτέλεσμα αντιστρέφεται προκειμένου να υπολογιστεί η τιμή της Closeness Centrality κάθε κόμβου. Στην περίπτωση που από έναν κόμβο του γραφήματος δεν ξεκινάει μονοπάτι προς όλους τους υπόλοιπους, τότε αυτός ο κόμβος θεωρείται ότι έχει μη ορισμένη Closeness Centrality.

Στην εικόνα που ακολουθεί, παρουσιάζεται ένα παράδειγμα υπολογισμού της Closeness Centrality. Αξίζει να παρατηρήσει κανείς το αποτέλεσμα του κόμβου 7, ο οποίος από την στιγμή που δεν έχει μονοπάτια προς όλους τους υπόλοιπους κόμβους του γραφήματος, η τιμή της Closeness Centrality του, είναι "NaN" κάτι που δηλώνει ότι δεν ορίζεται.



Εικόνα 41 - Παράδειγμα υπολογισμού της Closeness Centrality

5.3 Betweenness Centrality

Η Betweenness Centrality αποτελεί ακόμα ένα μέτρο κεντρικότητας των κόμβων ενός γραφήματος. Συγκεκριμένα ποσοτικοποιεί το πόσες φορές ένας κόμβος λειτουργεί ως ενδιάμεσος κατά μήκος της συντομότερης διαδρομής μεταξύ δύο οποιονδήποτε άλλων κόμβων του γραφήματος. Εισήχθη από τον Linton Freeman ως μέτρο για την ποσοτικοποίηση του ελέγχου ενός ανθρώπου στην επικοινωνία των μελών ενός κοινωνικού δικτύου. Σύμφωνα με την αντίληψή του, οι κορυφές με την μεγαλύτερη συχνότητα εμφάνισης σε ένα τυχαίο ελάχιστο μονοπάτι ανάμεσα σε δύο τυχαία επιλεγμένους κόμβους, παρουσιάζουν την μεγαλύτερη Betweenness Centrality.

Η εξίσωση 5.3.1 αποτελεί τον ορισμό της Betweenness Centrality ενός κόμβου

$$C_B(i) = \sum_{j \neq i \neq k \in V} \frac{g_{jk}(i)}{g_{jk}} \quad (5.3.1)$$

,όπου ο παρονομαστής του αθροίσματος αντιστοιχεί στο πλήθος των ελαχίστων μονοπατιών ανάμεσα στους κόμβους j και k , και ο αριθμητής στο πλήθος εκείνων των μονοπατιών που περιλαμβάνουν τον κόμβο i .

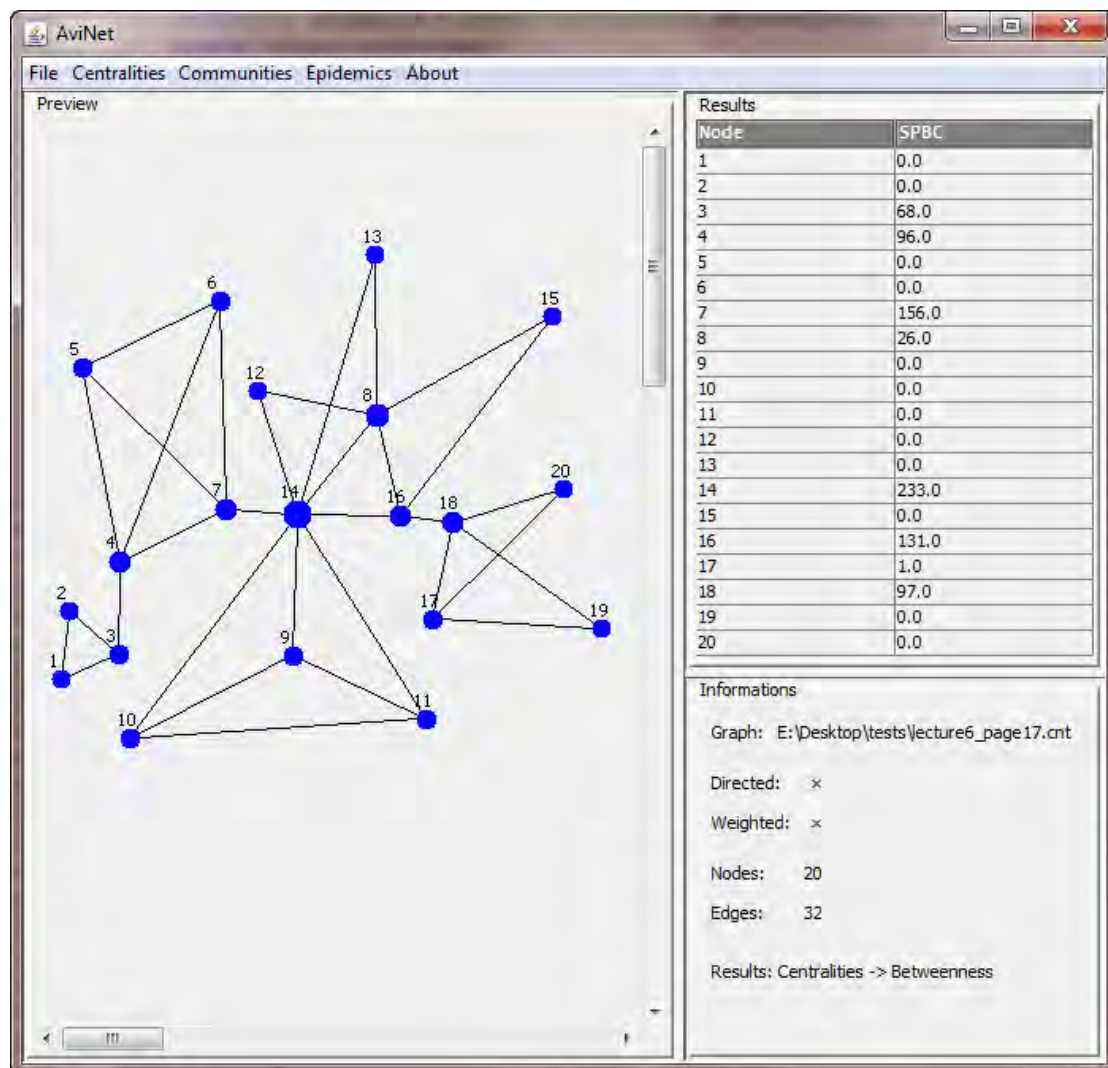
Για την κανονικοποίηση της Betweenness Centrality, αρκεί να διαιρέσει κανείς με $((n-1) \times (n-2))$, όπου το n δηλώνει και εδώ το πλήθος των κόμβων του γραφήματος.

$$C'_B(i) = \frac{C_B(i)}{(n-1) * (n-2)} \quad (5.3.2)$$

Το πρώτο βήμα για τον υπολογισμό της Betweenness Centrality στο Avinet, είναι η εύρεση του πίνακα γειτνίασης του γραφήματος. Μέχρι το σημείο αυτό η διαδικασία είναι όμοια με αυτήν του υπολογισμού της Closeness Centrality, μόνο που στην προκειμένη περίπτωση μας ενδιαφέρει το πλήθος των ελαχίστων μονοπατιών ανάμεσα σε όλους τους κόμβους του γραφήματος, και όχι τα μήκη τους. Για τον λόγο αυτό χρησιμοποιήθηκε μια παραλλαγή του αλγορίθμου του Dijkstra, η οποία δέχεται ως είσοδο τον πίνακα γειτνίασης του γραφήματος και επιστρέφει μια λίστα από λίστες. Πιο συγκεκριμένα, για κάθε κόμβο του γραφήματος, υπολογίζονται όλα τα ελάχιστα μονοπάτια που τον συνδέουν με τους υπόλοιπους. Στη συνέχεια, για τον υπολογισμό της Betweenness Centrality κάθε κόμβου, διατρέχονται οι λίστες των

ελαχίστων μονοπατιών όλων των υπόλοιπων κόμβων, με σκοπό να εντοπιστεί σε ποια από αυτά ανήκει ο υπό εξέταση κόμβος ως ενδιάμεσος. Για κάθε ελάχιστο μονοπάτι από τον κόμβο j στον κόμβο k , στο οποίο συμμετέχει ο κόμβος i , ο τελευταίος πιστώνεται $1/\rho$ μονάδες, όπου το ρ είναι το πλήθος των ελαχίστων αυτών μονοπατιών.

Στην εικόνα 42 παρουσιάζεται ένα παράδειγμα υπολογισμού της Betweenness Centrality σε ένα μη κατευθυνόμενο, μη βεβαρημένο γράφημα. Τα αποτελέσματα είναι σε μη κανονικοποιημένη μορφή.



Εικόνα 42 - Betweenness Centrality σε undirected unweighted γράφημα

Παρατηρούμε ότι οι πιο κεντρικοί κόμβοι του γραφήματος, 7, 14 και 16, είναι και αυτοί με τις μεγαλύτερες τιμές. Αυτό συμβαίνει γιατί από τους κόμβους αυτούς εξαρτάται η επικοινωνία των περισσότερων από τους υπολοίπους του γραφήματος, γεγονός που επαληθεύει και την θεωρία.

5.4 Edge Betweenness Centrality

Η Edge Betweenness Centrality είναι ένα μέτρο κεντρικότητας των ακμών ενός γραφήματος. Στη ίδια φιλοσοφία με την Betweenness Centrality των κόμβων, η Edge Betweenness Centrality μιας ακμής, ορίζεται ως το πλήθος των συντομότερων μονοπατιών του γραφήματος που την διασχίζουν. Θα μπορούσε να πει κανείς, ότι μια ακμή με υψηλή Edge Betweenness Centrality αναπαριστά έναν σύνδεσμο που ομοιάζει με γέφυρα μεταξύ δύο τμημάτων του δικτύου, η απομάκρυνση της οποίας μπορεί να επηρεάσει σημαντικά την επικοινωνία των κόμβων, αν υποθέσουμε ότι αυτή πραγματοποιείται μέσω συντομότερων μονοπατιών.

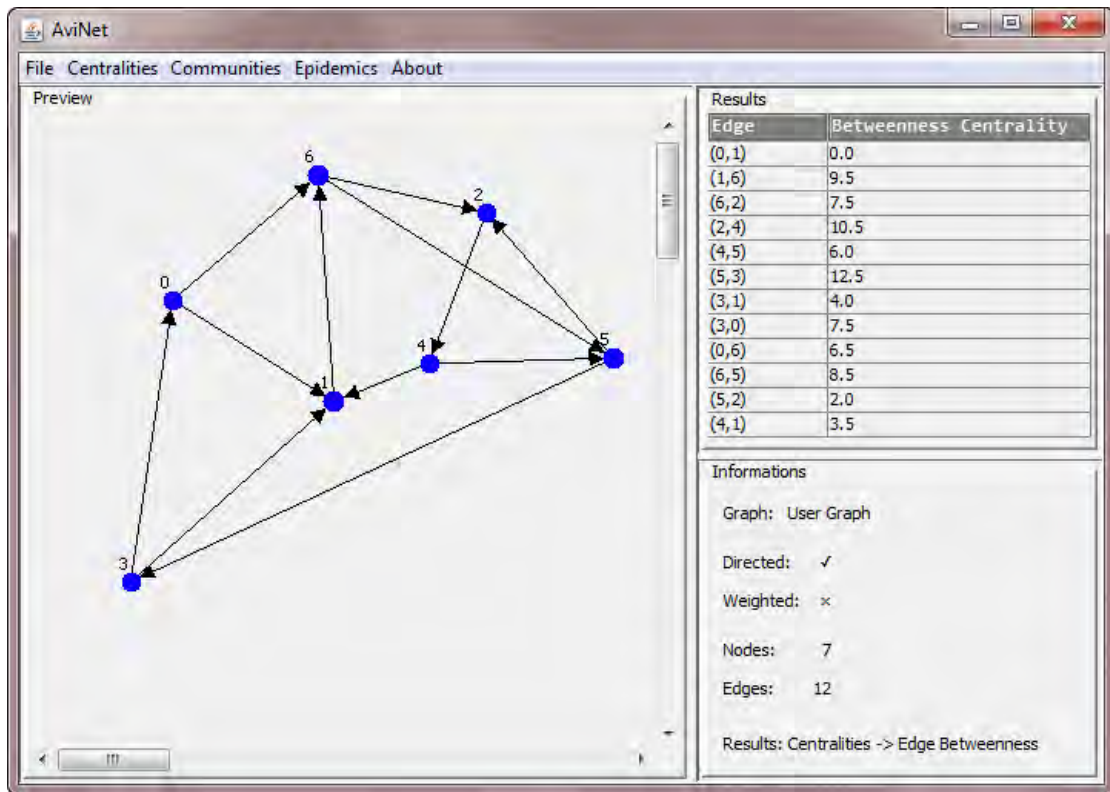
Ο ορισμός της Edge Betweenness Centrality μιας ακμής e , μπορεί να παρασταθεί από την επόμενη εξίσωση

$$C_B(e) = \sum_{s \neq t \in V} \frac{g_{st}(e)}{g_{st}} \quad (5.4.1)$$

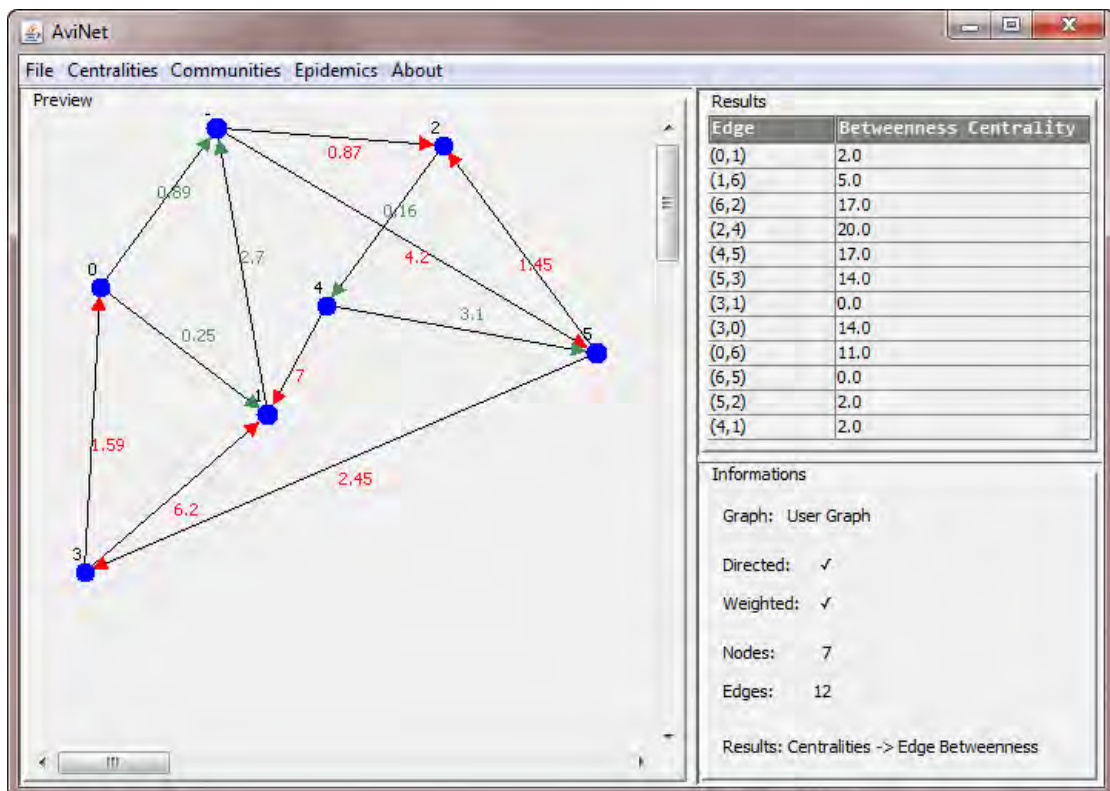
,με τον παρονομαστή να δηλώνει το πλήθος των ελαχίστων μονοπατιών ανάμεσα στους κόμβους s και t , και τον αριθμητή το πλήθος εκείνων των μονοπατιών, τα οποία διασχίζουν την ακμή e . Οι τιμές της Edge Betweenness Centrality δεν κανονικοποιούνται.

Ο υπολογισμός της Edge Betweenness Centrality από το Anynet, παρουσιάζει αρκετές ομοιότητες με αυτόν της Betweenness Centrality για τους κόμβους. Ο πίνακας γεινίασης του γραφήματος υπολογίζεται, και στην συνέχεια δίνεται στον ήδη υλοποιημένο αλγόριθμο Dijkstra, ο οποίος επιστρέφει και πάλι τα ελάχιστα μονοπάτια μεταξύ όλων των κόμβων του γραφήματος. Στη συνέχεια οι λίστες με τα ελάχιστα μονοπάτια διατρέχονται για κάθε ακμή ξεχωριστά. Για κάθε ελάχιστο μονοπάτι από τον κόμβο s στον κόμβο t , στο οποίο εντοπίζεται η ακμή e , αυτή χρεώνεται $1/\rho$ μονάδες, όπου το ρ είναι το πλήθος των ελαχίστων αυτών μονοπατιών.

Οι εικόνες 43 και 44 παρουσιάζουν δύο χαρακτηριστικά παραδείγματα υπολογισμού της Edge Betweenness Centrality. Στο πάνελ των αποτελεσμάτων, η πρώτη στήλη φέρει τα ονόματα των ακμών, έτσι όπως δημιουργήθηκαν κατά την κατασκευή των γραφημάτων. Η ακμή (i, j) δηλώνει την ύπαρξη ακμής από τον κόμβο i , στον κόμβο j , ανεξαρτήτως αν το γράφημα είναι κατευθυνόμενο ή όχι.



Εικόνα 43 - EBC σε directed unweighted γράφημα



Εικόνα 44 - EBC σε directed weighted γράφημα

Το γράφημα της εικόνας 44, έχει προέλθει από εκείνο της εικόνας 43, με την προσθήκη βαρών στις ακμές. Από τα αποτελέσματα φαίνεται ξεκάθαρα πως τα βάρη των ακμών επηρεάζουν την αξία των ακμών, και κατά συνέπεια τις τιμές της Edge Betweenness Centrality.

5.5 Power Community Index (μ -PCI)

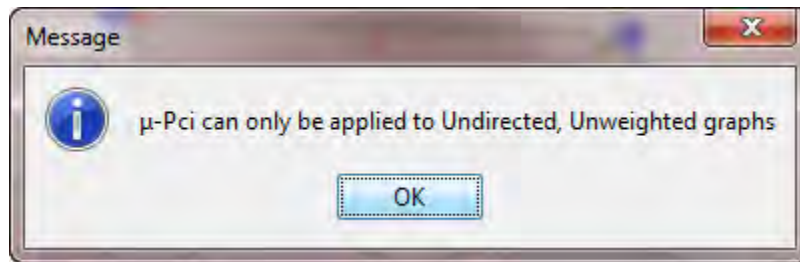
Το μ -PCI είναι ένας ακόμα σημαντικός δείκτης κεντρικότητας των κόμβων ενός γραφήματος, ο οποίος δύναται να εφαρμοστεί μόνο σε μη κατευθυνόμενα, μη βεβαρημένα δίκτυα. Διαισθητικά θα μπορούσε να πει κανείς ότι το Power Community Index, δείχνει το πλήθος των σημαντικότερων κόμβων του γραφήματος, από άποψη κεντρικότητας βαθμού, με τους οποίους γειτονεύει ο εκάστοτε κόμβος. Στην αποσαφήνιση και την κατανόηση του μ -PCI, μπορεί να βοηθήσει ο επόμενος ορισμός.

Ο μ -PCI δείκτης ενός κόμβου ισούται με k , αν και μόνο αν δεν υπάρχουν περισσότεροι από μk κόμβοι στην “ μ -hop” γειτονιά του, με Degree Centrality μεγαλύτερο από k .

Με τον όρο “ μ -hop” γειτονιά ενός κόμβου, ορίζεται το σύνολο των κόμβων που ο καθένας απέχει από αυτόν μονοπάτι μήκους ίσο με “ μ ”. Για παράδειγμα, η 1-hop γειτονιά ενός κόμβου k , αποτελείται από όλους τους κόμβους με τους οποίους ο k συνδέεται με απευθείας ακμή, ενώ η 2-hop γειτονιά του k , περιλαμβάνει το σύνολο των κόμβων οι οποίοι απέχουν από τον k μονοπάτια μήκους 2.

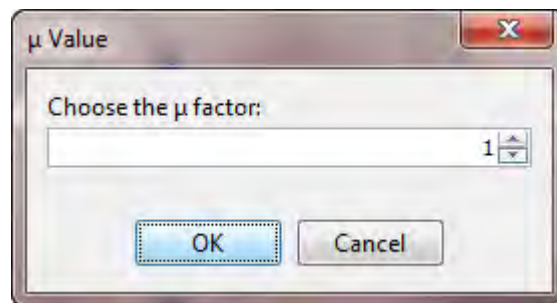
Η εκτέλεση του συγκεκριμένου αλγορίθμου στο AnviNet, πραγματοποιείται με την επιλογή του μενού “Centralities \rightarrow μ -PCI”.

Αν ο χρήστης χρησιμοποιεί γράφημα με κατευθυνόμενες ή βεβαρημένες ακμές, ο αλγόριθμος του μ -PCI αδυνατεί να εξ ορισμού να εφαρμοστεί. Έτσι εμφανίζεται το μήνυμα της επόμενης εικόνας.



Εικόνα 45 - Ενημερωτικό μήνυμα για την εκτέλεση του μ-PCI

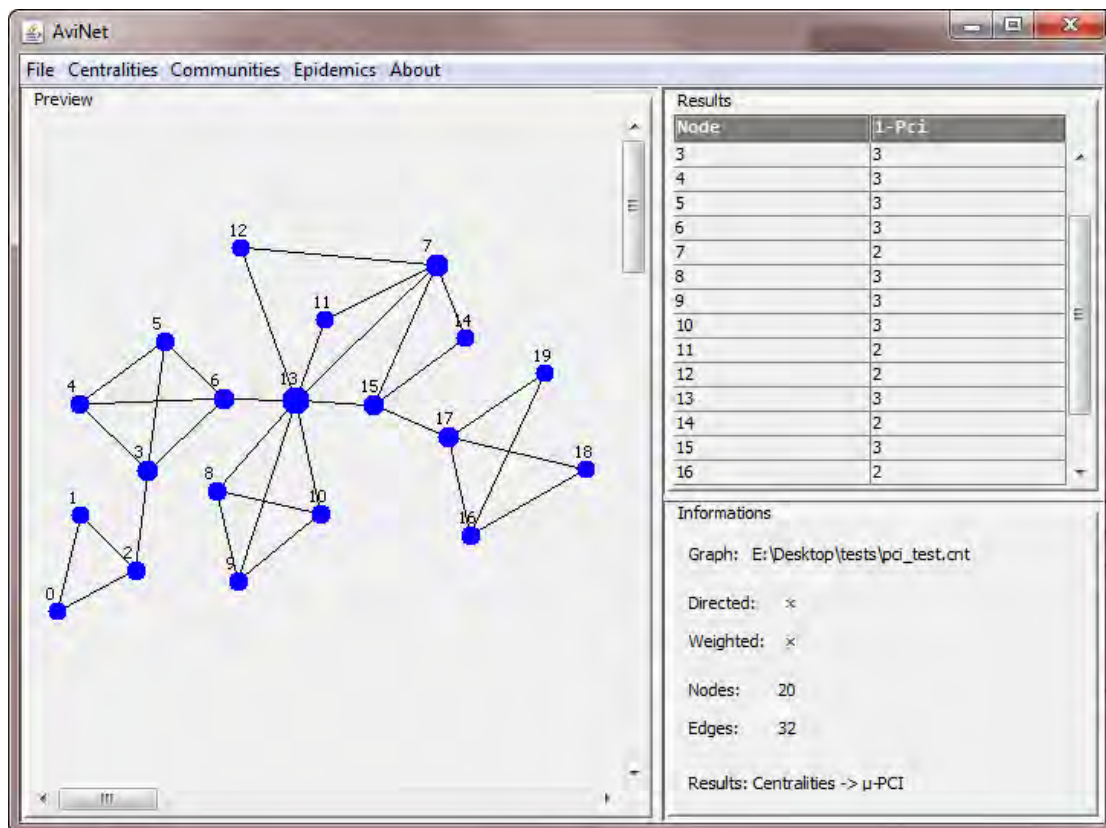
Με την ύπαρξη μη κατευθυνόμενου, μη βεβαρημένου γραφήματος στο Preview πάνελ, εμφανίζεται το μενού της εικόνας 46, το οποίο ζητά από τον χρήστη να εισάγει την επιθυμητή τιμή της παραμέτρου “μ”, ώστε να εκτελεστεί ο αλγόριθμος εμφανίζοντας τα ανάλογα αποτελέσματα.



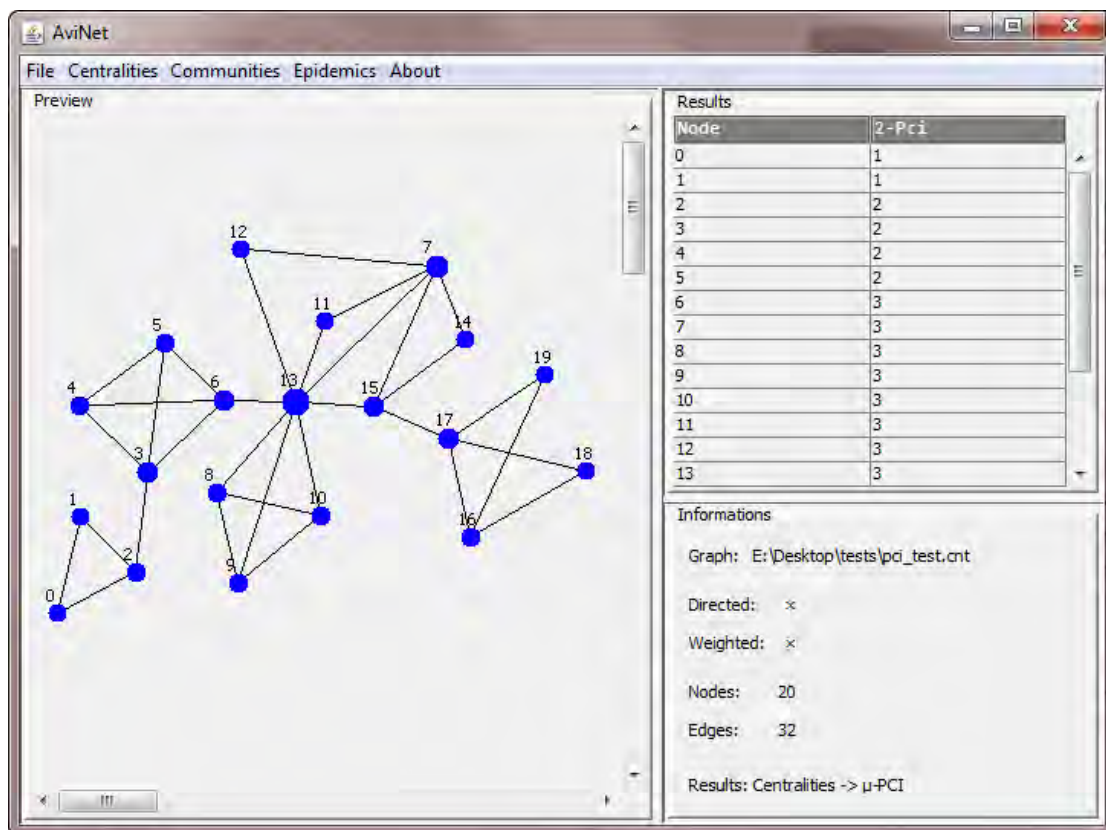
Εικόνα 46 - Επιλογή παραμέτρου μ για το μ-PCI

Πρώτο βήμα για τον υπολογισμό του μ-PCI στο AniNet, είναι η εύρεση των βαθμών, όλων των κόμβων του γραφήματος. Η λίστα με τα Degree Centrality, καθώς επίσης και η τιμή της παραμέτρου “μ” περνούν ως είσοδοι στην αλγόριθμο, ο οποίος με την σειρά του ζητά από το πρόγραμμα τον υπολογισμό του πίνακα γειτνίασης. Έπειτα για τον κάθε κόμβο επαναληπτικά, εντοπίζονται οι γείτονες της “μ-hop” γειτονιάς του, για τους οποίους καταγράφονται οι τιμές των Degree Centrality. Τέλος μέσω επαναληπτικής διαδικασίας, υπολογίζεται για κάθε κόμβο η τιμή του k, η οποία ικανοποιεί τον ορισμό του μ-PCI.

Τα αποτελέσματα του AniNet για το ίδιο παράδειγμα γραφήματος, και για τιμές του “μ” 1 και 2, φαίνονται στις εικόνες 47 και 48 αντίστοιχα.



Εικόνα 47 - PCI για $\mu=1$



Εικόνα 48 - PCI για $\mu=2$

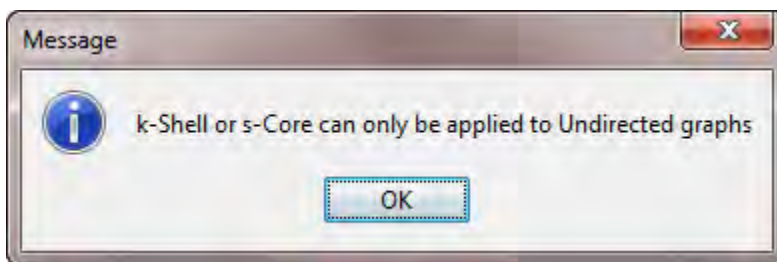
5.6 k-Shell/s-Core

Οι αλγόριθμοι k-Shell και s-Core εισήχθησαν ως ένα εργαλείο μελέτης της δομής των μεγάλων δικτύων. Χρησιμοποιούνται για τον εντοπισμό των κόμβων με τις μεγαλύτερες ικανότητες επιρροής σε ένα γράφημα, παρέχοντας μια μέθοδο αναγνώρισης των ιεραρχιών. Μέθοδοι σαν και αυτές εφαρμόζονται περισσότερο σε στατικά δίκτυα, δίκτυα δηλαδή των οποίων η τοπολογία παραμένει σταθερή κατά το πέρασμα του χρόνου.

Η ανάλυση κατά k-Shell και s-Core δημιουργεί υπογραφήματα σύμφωνα με τα Degree Centralities των κόμβων. Έτσι προκύπτει ένα είδος ομαδοποίησης, κατά την οποία οι κόμβοι του γραφήματος χωρίζονται με βάση την συνδεσιμότητά τους. Με τον τρόπο αυτό αναγνωρίζονται οι κόμβοι του γραφήματος με τις μεγαλύτερες ικανότητες στην διάδοση πληροφοριών και την εξάπλωση επιδημιών.

Η εφαρμογή αυτού του είδους της ανάλυσης προϋποθέτει μη κατευθυνόμενα γραφήματα. Στην περίπτωση των μη βεβαρημένων γραφημάτων, παρουσιάζεται ως k-Shell, ενώ για τα βεβαρημένα γραφήματα αναφέρεται ως s-Core. Η διαφορά έγκειται ότι στα μη βεβαρημένα γραφήματα, ο βαθμός ενός κόμβου ισούται με το άθροισμα των 1-hop γειτόνων του, ενώ στα βεβαρημένα ορίζεται ως το άθροισμα των βαρών των ακμών του, και συνεπώς το πλήθος των ακμών αδυνατεί από μόνο του να ορίσει την σημαντικότητα του κόμβου.

Το Avinet, σε περίπτωση που εντοπίσει κατευθυνόμενο γράφημα, στο οποίο είναι αδύνατο να εφαρμοστεί η ανάλυση κατά k-Shell/s-Core, εμφανίζει στον χρήστη το ακόλουθο μήνυμα.



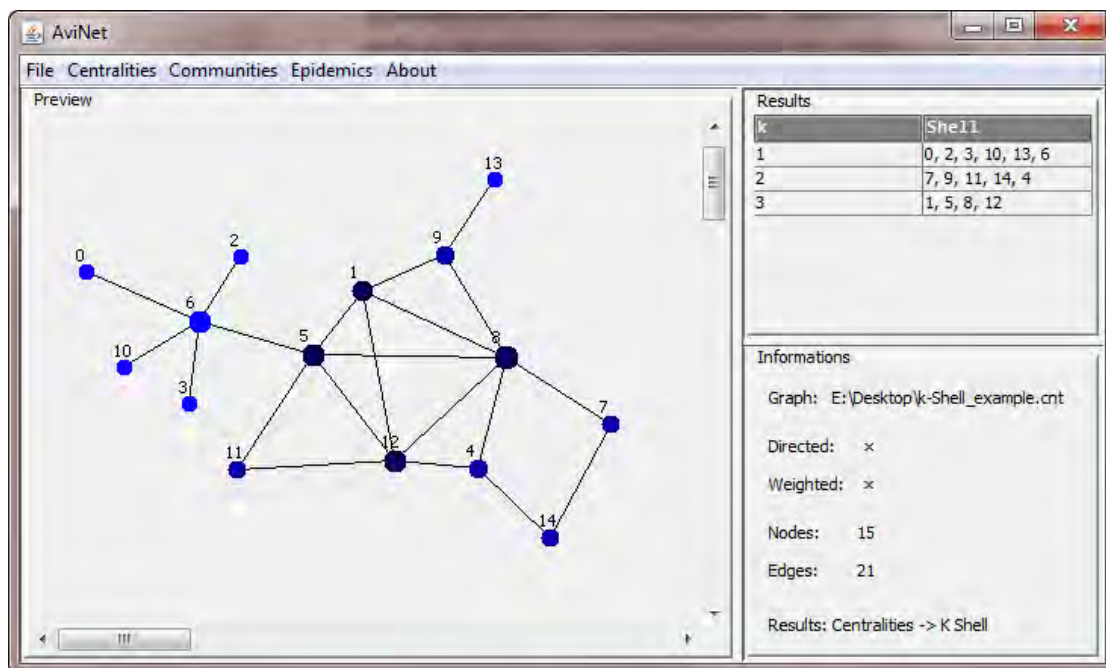
Εικόνα 49 - Ενημερωτικό μήνυμα για την εκτέλεση του k-Shell/s-Core

Ο υπολογισμός των k-Shell και s-Core στο Avinet, γίνεται με παρόμοιο τρόπο. Αρχικά το πρόγραμμα δημιουργεί ένα αντίγραφο των κόμβων και των ακμών του γραφήματος, υπολογίζοντας τις τιμές της Degree Centrality. Κατά το πρώτο βήμα, για

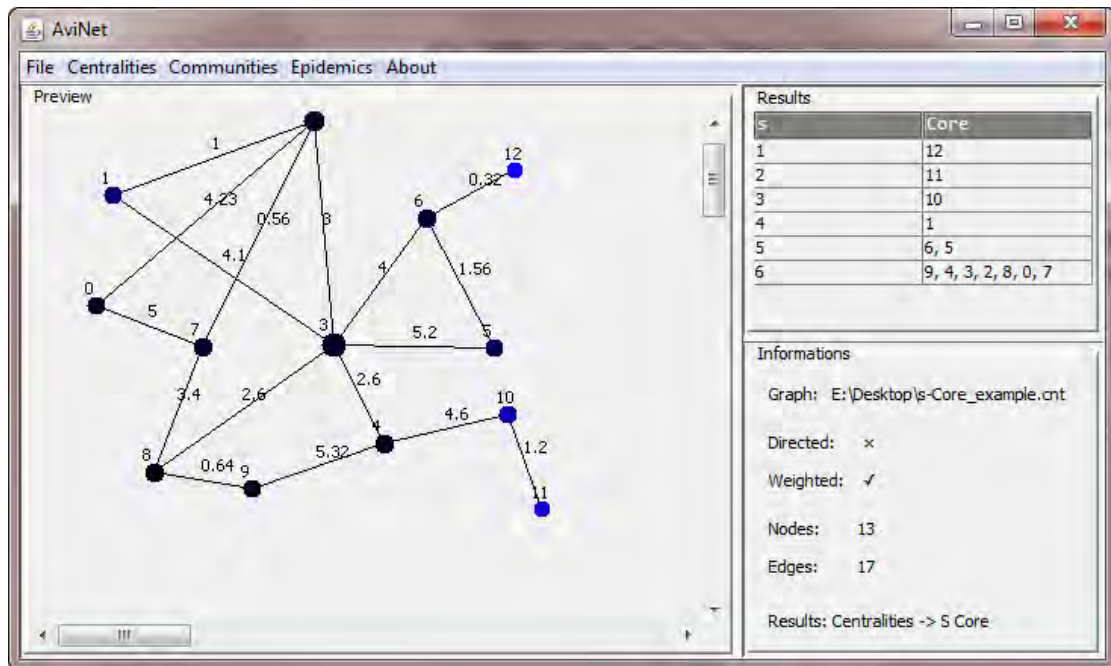
των υπολογισμό των κόμβων της πρώτης ομάδας, αφαιρούνται από το αντίγραφο του γραφήματος οι κόμβοι με Degree ίσο με 1, και κρατούνται σε μια βοηθητική λίστα. Έπειτα υπολογίζονται ξανά τα Degree Centralities των κόμβων που απέμειναν, με σκοπό να αφαιρεθούν οι καινούριοι των οποίων οι βαθμοί είναι ίσοι με 1. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να δημιουργηθεί διαμέριση του αρχικού γραφήματος στην οποία κανένας κόμβος δεν παρουσιάζει Degree ίσο με 1. Με τον τρόπο αυτό έχουν βρεθεί οι κόμβοι που ανήκουν στο $k=1$ για μη βεβαρημένα γραφήματα και στο $s=1$ για βεβαρημένα. Η παραπάνω διαδικασία επαναλαμβάνεται αυξάνοντας τα k και s , και ομαδοποιώντας τους κόμβους μέχρι να εξαλειφθούν όλοι.

Στο AviNet, εκτός από την λίστα με τους κόμβους της κάθε ομάδας που εμφανίζεται στο Results πάνελ, οι κόμβοι του γραφήματος που αναπαρίσταται στο Preview πάνελ χρωματίζονται σε διαφορετικές αποχρώσεις του ίδιου μπλε χρώματος, βοηθώντας έτσι στην κατανόηση των αποτελεσμάτων μέσω της χρωματικής εναλλαγής τους.

Η εικόνα 50 παρουσιάζει ένα παράδειγμα τρεξίματος του k -Shell σε μη βεβαρημένο γράφημα, ενώ η εικόνα 51 περιλαμβάνει τα αποτελέσματα του s -Core για ένα βεβαρημένο γράφημα.



Εικόνα 50 - Το k -Shell στο AviNet



Εικόνα 51 - Το s-Core στο AviNet

Αν μετά την εκτέλεση των συγκεκριμένων αλγορίθμων, ο χρήστης επιλέξει να αλλάξει την τοπολογία του γραφήματος, εισάγοντας ή διαγράφοντας κόμβο ή ακμή, τα χρώματα θα επανέλθουν στις αρχικές τους τιμές, μιας και τα αποτελέσματα δεν θα ανταποκρίνονται πλέον στο αναπαριστώμενο γράφημα.

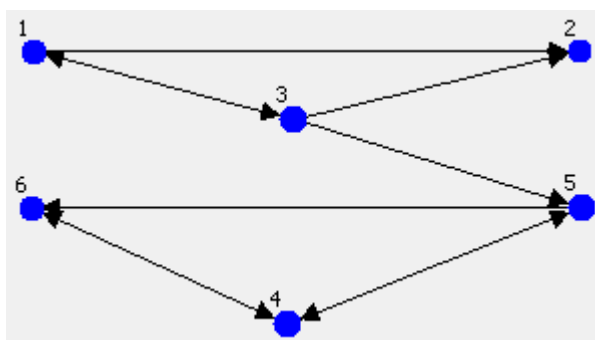
5.7 Page Rank

Ο κόσμος του διαδικτύου δύναται να μελετηθεί από τα σύνθετα δίκτυα, και συνεπώς να αναπαρασταθεί μέσα από γραφήματα. Έτσι ένα σύνολο ιστοσελίδων μπορεί να αποτελεί τους κόμβους ενός γραφήματος, με τις κατευθυνόμενες ακμές να αναπαριστούν τους εισερχόμενους και εξερχόμενους συνδέσμους μεταξύ τους.

Μέχρι το 1998 το διαδίκτυο αποτελούσε σε μεγάλο βαθμό μια ανεκμετάλλευτη πηγή πληροφοριών. Καθώς οι ερευνητές αναγνώριζαν τις προοπτικές του διαδικτύου, ο περισσότερος κόσμος αναρωτιόταν για το πώς σχετίζεται αυτό με τις μηχανές αναζήτησης. Ο συσχετισμός μπορεί να γίνει εύκολα κατανοητός, θεωρώντας τους συνδέσμους μεταξύ των ιστοσελίδων, ως συστάσεις. Ένας σύνδεσμος από την ιστοσελίδα k στην ιστοσελίδα m , αποτελεί ένα τρόπο αναγνώρισης και επισήμανσης της ιστοσελίδας m από την k . Συνεπώς μια ιστοσελίδα με περισσότερους εισερχόμενους συνδέσμους (inlinks), πρέπει να είναι πιο σημαντική από μια ιστοσελίδα με λιγότερους. Ωστόσο δεν θα πρέπει να ξεχνά κανείς και την διαφορετική αξία που προσφέρουν οι εισερχόμενοι σύνδεσμοι από ιστοσελίδες διαφορετικής σημαντικότητας.

Κρίθηκε αναγκαίο λοιπόν να βρεθεί μία φόρμουλα κατάταξης των ιστοσελίδων με βάση την αναγνωρισιμότητά τους, και για τον λόγο αυτό οι ιδρυτές της Google, Brin και Page, εφηύραν τον αλγόριθμο του Page Rank. Ο αλγόριθμος αυτός χρησιμοποιείται από την Google για την κατάταξη των ιστοσελίδων στις μηχανές αναζήτησης.

Προκειμένου να γίνει κατανοητός ο αλγόριθμος υπολογισμού του Page Rank, θα τον παρουσιάσουμε μέσα από ένα απλό παράδειγμα, βήμα προς βήμα. Έστω το γράφημα της εικόνας 52, το οποίο αναπαριστά ένα σύνολο 6 ιστοσελίδων με τις ακμές να δεικνύουν τα inlinks (εισερχόμενους συνδέσμους) και τα outlinks (εξερχόμενους συνδέσμους) μεταξύ τους.



Εικόνα 52 - Παράδειγμα δικτύου ιστοσελίδων

Ο υπολογισμός του Page Rank έγκειται στην επαναληπτική επίλυση της εξίσωσης ιδιοτιμών,

$$\pi^{(k+1)T} = \pi^{(k)T} * G \quad (5.7.1)$$

,με το π^T να αποτελεί το διάνυσμα-γραμμή που συγκρατεί τα Page Ranks όλων των κόμβων του γραφήματος, και το G να αποτελεί τον πίνακα Google. Η αρχική τιμή κάθε θέσης του διανύσματος π , είναι $1/n$, με n το συνολικό πλήθος των κόμβων-ιστοσελίδων του δικτύου. Στο συγκεκριμένο παράδειγμα,

$$\pi^{(0)T} = [1/6 \quad 1/6 \quad 1/6 \quad 1/6 \quad 1/6 \quad 1/6].$$

Το διάνυσμα π^T των Page Ranks υπολογίζεται επαναληπτικά, μέχρι να βρεθούν τα σημεία εκείνα, στα οποία οι τιμές του συγκλίνουν. Η ανοχή σύγκλισης που χρησιμοποιείται στο AniNet είναι 10^{-8} . Για τον υπολογισμό της μεταβολής ανάμεσα στις διαδοχικές τιμές του διανύσματος π^T , χρησιμοποιείται ένα διάνυσμα σφάλματος ιδίου μεγέθους, το οποίο σε κάθε επανάληψη κατά την οποία υπολογίζεται εκ νέου το διάνυσμα π^T , συγκρατεί την διαφορά της προηγούμενης από την τρέχουσα τιμή του. Όταν το άθροισμα των τιμών του διανύσματος σφάλματος είναι μικρότερο ή ίσο από την ανοχή σύγκλισης, θεωρούμε πως το διάνυσμα π^T έχει συγκλίνει σε ικανοποιητικό βαθμό, και η διαδικασία σταματά.

Ο πίνακας Google υπολογίζεται μόνο μια φορά κατά την εκκίνηση του αλγορίθμου, με τον ακόλουθο τύπο,

$$G = \alpha * H + ((\alpha * a) + (1 - \alpha) * e) * 1/n * e^T \quad (5.7.2)$$

,όπου η μεταβλητή “α” αποτελεί τον παράγοντα κλιμάκωσης. Ουσιαστικά αντικατοπτρίζει τον χρόνο κατά τον οποίο ένας περιηγητής του δικτύου ακολουθεί τους υπάρχοντες συνδέσμους. Για παράδειγμα, για $\alpha = 0,6$ ο περιηγητής καταναλώνει το 60% του χρόνου του ακολουθώντας τα links, και το υπόλοιπο 40% κάνοντας τυχαίες μεταφορές. Η παράμετρος “α” συμβάλει στην αντιμετώπιση του αδιεξόδου στο δίκτυο, την περίπτωση δηλαδή κατά την οποία ο περιηγητής μπορεί να βρεθεί σε ιστοσελίδα που δεν φέρει εξερχόμενους συνδέσμους με αποτέλεσμα να εγκλωβιστεί. Η τιμή του παράγοντα κλιμάκωσης που χρησιμοποιείται από την Google αλλά και το AniNet είναι 0,85. Συνεχίζοντας, το “a” είναι το διάνυσμα που υποδηλώνει ποιοι κόμβοι λειτουργούν ως καταβόθρες στο δίκτυο, όπου ως καταβόθρα ορίζεται ένας κόμβος ο οποίος δεν έχει κανέναν εξερχόμενο σύνδεσμο προς άλλους κόμβους.

Στο συγκεκριμένο παράδειγμα, ο κόμβος 2 αποτελεί την μοναδική καταβόθρα, επομένως το διάνυσμα φέρει τις ακόλουθες τιμές

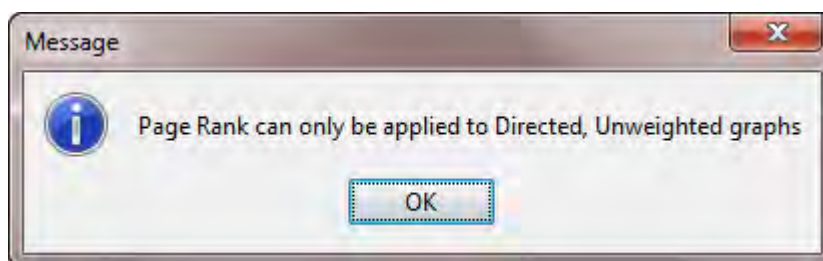
$$a = [0 \ 1 \ 0 \ 0 \ 0 \ 0].$$

Το διάνυσμα “e” είναι το μοναδιαίο διάνυσμα και φέρει την τιμή 1 σε κάθε του θέση. Τέλος ο πίνακας “H” είναι ένας αραιός υποστοχαστικός πίνακας, ο οποίος περιλαμβάνει τους εσωτερικούς και εξωτερικούς συνδέσμους του γραφήματος. Στην θέση (i, j) του πίνακα H υπάρχει η τιμή $1/k$, αν υπάρχει σύνδεσμος από την σελίδα i στην σελίδα j, με το k να δηλώνει το συνολικό πλήθος των εξερχομένων συνδέσμων της σελίδας i. Σε διαφορετική περίπτωση η τιμή της θέσης αυτής είναι 0. Έτσι, για το δίκτυο του παραδείγματος, ο πίνακας H έχει την ακόλουθη μορφή

$$H = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

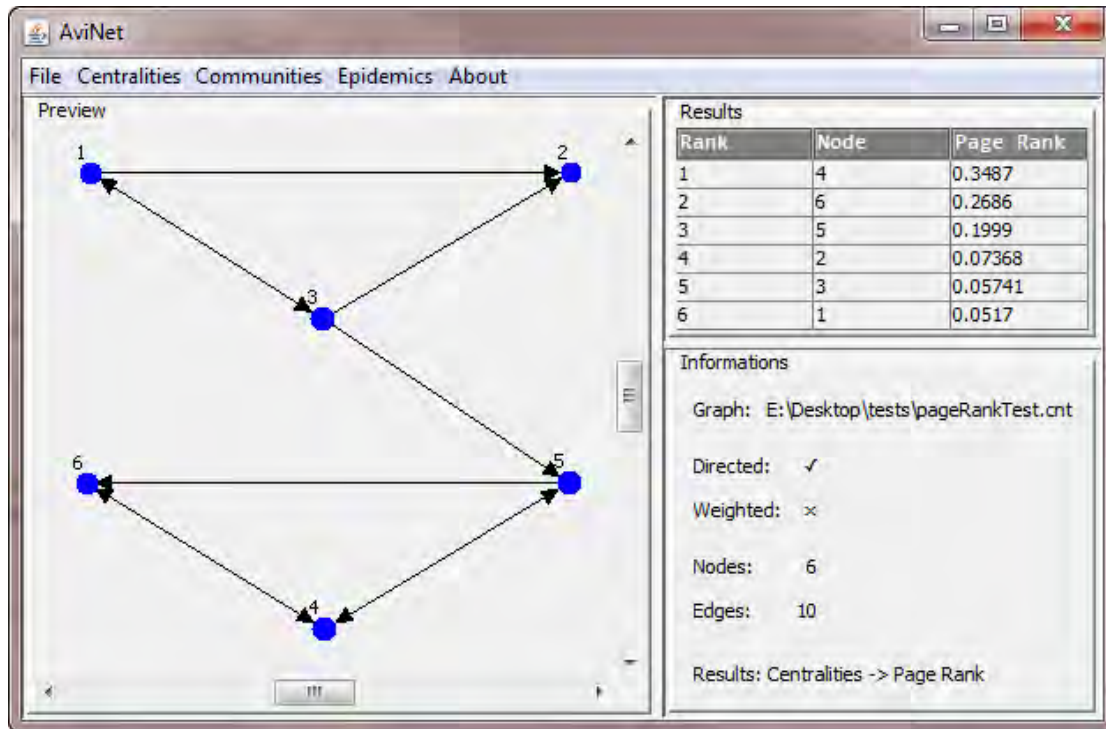
Το AniNet κατά την εκκίνηση του αλγορίθμου για τον υπολογισμό του Page Rank, βρίσκει τον πίνακα G του δικτύου, και στην συνέχεια επαναληπτικά υπολογίζει το διάνυσμα π^T με την διαδικασία που αναφέρθηκε.

Για την εκτέλεση του Page Rank, αρκεί κανείς να ακολουθήσει το μενού “Centralities → Page Rank”, έχοντας πρώτα εισάγει στο πρόγραμμα ένα κατευθυνόμενο, μη βεβαρημένο γράφημα. Σε περίπτωση που εντοπιστεί διαφορετικός τύπος γραφήματος, το AniNet ενημερώνει τον χρήστη με την εμφάνιση του παρακάτω μηνύματος.



Εικόνα 53 - Ενημερωτικό μήνυμα για την εκτέλεση του Page Rank

Η εικόνα 54 δείχνει τα αποτελέσματα του προγράμματος μετά την εκτέλεση του αλγορίθμου Page Rank για το γράφημα της εικόνας 52. Στο Results πάνελ οι κόμβοι-ιστοσελίδες βρίσκονται διατεταγμένοι στην στήλη 2, κατά φθίνουσα τιμή Page Rank, η οποία αναγράφεται στην στήλη 3.



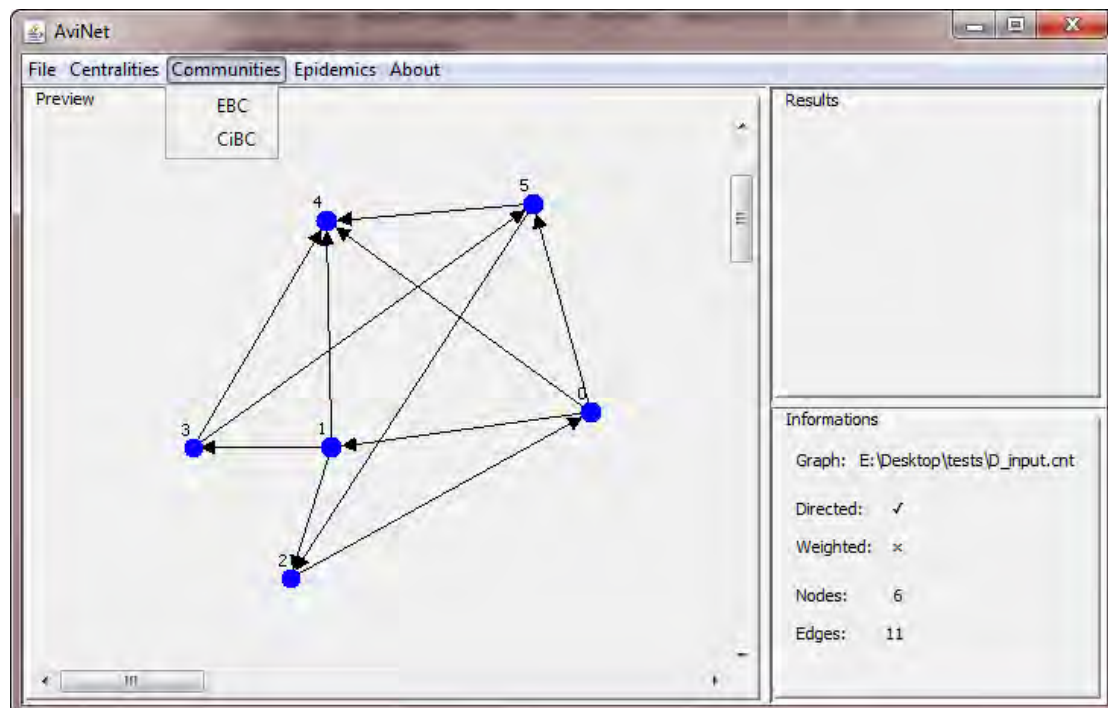
Εικόνα 54 - Υπολογισμός Page Rank στο AviNet

6 Κοινότητες στο AviNet

Οι κοινότητες στα σύνθετα δίκτυα είναι σύνολα κόμβων οι οποίοι μοιράζονται κοινά χαρακτηριστικά και ιδιότητες σε ένα γράφημα. Ουσιαστικά οι κοινότητες αποτελούνται από κόμβους οι οποίοι είναι στενά συνδεδεμένοι μεταξύ τους, την ίδια στιγμή που μεταξύ διαφορετικών κοινοτήτων εμφανίζονται μικρότεροι βαθμοί σύνδεσης. Η ανάλυση των σύνθετων δικτύων, κατά κύριο λόγο με βάση τις κεντρικότητές τους, δύναται να οδηγήσει στην ομαδοποίηση των κόμβων με βάση τα κοινά τους χαρακτηριστικά. Στο AviNet παρέχονται δύο βασικοί αλγόριθμοι εντοπισμού κοινοτήτων.

1. EBC (Using Edge Betweenness)
2. CiBC (Using Node Betweenness)

Για την εκτέλεση των αλγορίθμων εντοπισμού κοινοτήτων, αρκεί να ακολουθήσει κανείς την επιλογή “Communities” όπως αυτή παρέχεται από το βασικό κεντρικό μενού.



Εικόνα 55 – Communities

6.1 EBC (Girvan & Newman using Edge Betweenness)

Όπως παρουσιάστηκε και στο κεφάλαιο 5, ο αλγόριθμος υπολογισμού της Edge Betweenness Centrality εντοπίζει το πλήθος των συντομότερων μονοπατιών που διασχίζουν κάθε ακμή του γραφήματος. Οι ακμές με τις μεγαλύτερες τιμές θεωρούνται σημαντικότερες για την διατήρηση της συνδεσιμότητας του δικτύου, καθώς παίζουν καθοριστικό ρόλο στην επικοινωνία των κόμβων του. Συνεπώς η απομάκρυνσή τους δημιουργεί διάσπαση του αρχικού γραφήματος σε επιμέρους υπογραφήματα.

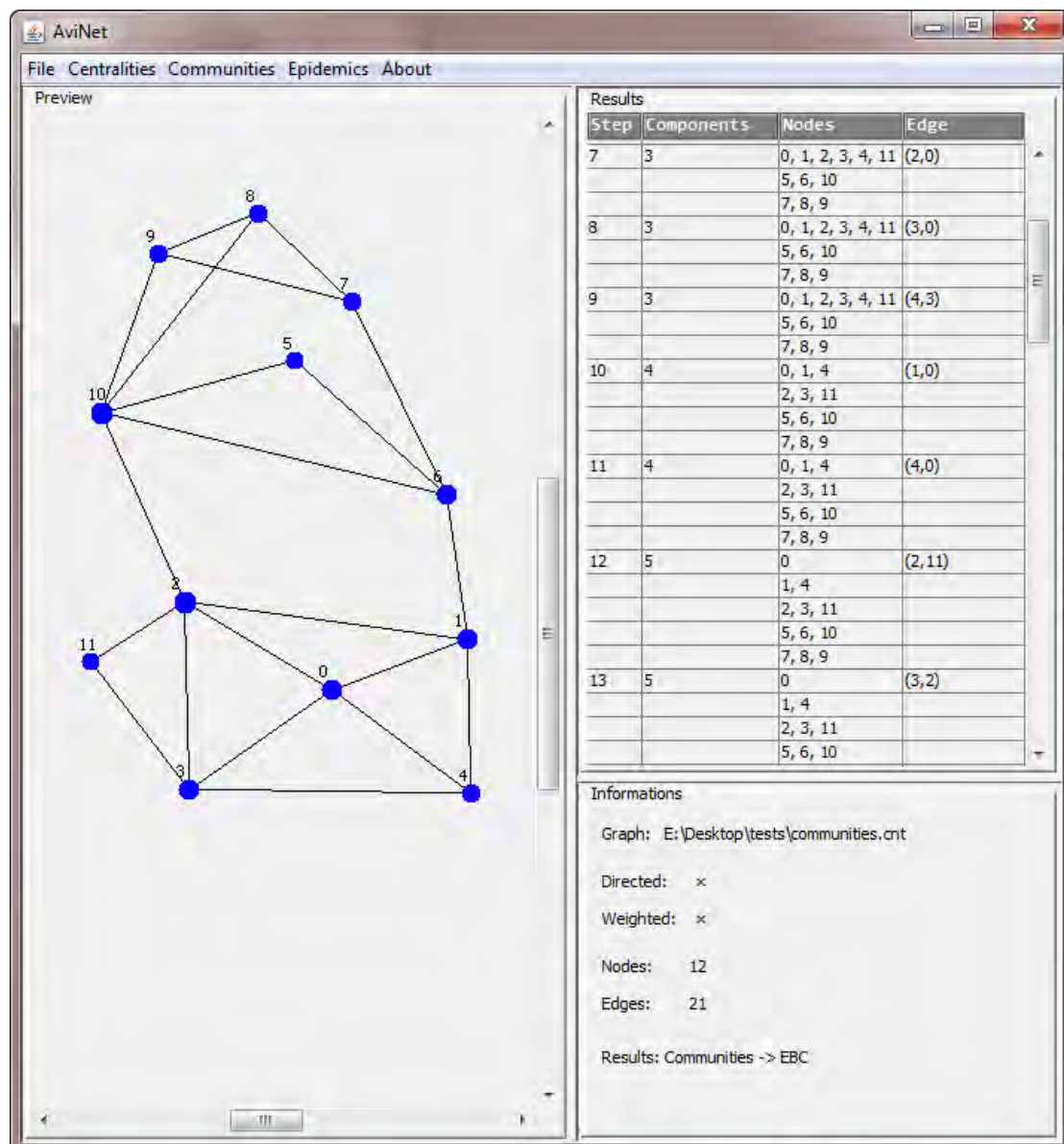
Ο αλγόριθμος των Girvan και Newman εντοπίζει τις κοινότητες με σταδιακή αφαίρεση των ακμών του αρχικού γραφήματος. Η αφαίρεση των ακμών με τις μεγαλύτερες τιμές της Edge Betweenness δημιουργεί σταδιακές διαμερίσεις του γραφήματος, οι οποίες αποτελούν τις ζητούμενες κοινότητες. Η διαδικασία περιγράφεται από τα ακόλουθα βήματα.

1. Υπολογισμός της Edge Betweenness Centrality του γραφήματος.
2. Αφαίρεση της ακμής με το μεγαλύτερο Edge Betweenness.
3. Επανυπολογισμός της Edge Betweenness του γραφήματος που έχει προκύψει μετά το βήμα 2.
4. Επανάληψη από το βήμα 2, μέχρι να εξαλειφθούν όλες οι ακμές.

Για την εκτέλεση του αλγορίθμου σε οποιοδήποτε τύπο γραφήματος, αρκεί ο χρήστης να ακολουθήσει το μενού “Communities → EBC”.

Για τον εντοπισμό των κοινοτήτων με τον αλγόριθμο των Girvan & Newman, το Aninet εκτός από τον πίνακα γειτνίασης, υπολογίζει τις τιμές της Edge Betweenness Centrality όλων των ακμών του τρέχοντος γραφήματος. Εν συνεχεία δημιουργείται ένα αντίγραφο του γραφήματος, για την προγραμματιστική διευκόλυνση της διαδικασίας. Στο αντίγραφο αυτό, μέσα από τον πίνακα γειτνίασης πραγματοποιείται αναζήτηση κατά βάθος, προκειμένου να εντοπιστεί το πλήθος των συνιστωσών του, καθώς επίσης και τα μέλη τους. Έπειτα αφαιρείται η ακμή με το μεγαλύτερο Edge Betweenness και η διαδικασία της παρούσας παραγράφου επαναλαμβάνεται μέχρι να εξαλειφθούν όλες οι ακμές του γραφήματος.

Η εικόνα 56 παρουσιάζει ένα τρέξιμο του συγκεκριμένου αλγορίθμου στο AniNet. Η πρώτη στήλη του πίνακα αποτελεσμάτων δηλώνει τα βήματα του αλγορίθμου. Το πρώτο βήμα αποτελεί την αρχικοποίηση, ενώ κάθε επόμενο αναφέρεται και σε μια ξεχωριστή επανάληψη κατά την οποία αφαιρείται η ακμή με το υψηλότερο τρέχον Edge Betweenness. Η ακμή που αφαιρείται σε κάθε βήμα του αλγορίθμου, παρουσιάζεται στην στήλη τέσσερα. Στην δεύτερη στήλη δηλώνεται το πλήθος των κοινοτήτων του γραφήματος, ενώ στην τρίτη παρουσιάζονται οι κόμβοι που απαρτίζουν την κάθε κοινότητα.



Εικόνα 56 - Εύρεση κοινοτήτων με βάση την Edge Betweenness

6.2 CiBC (Using node Betweenness)

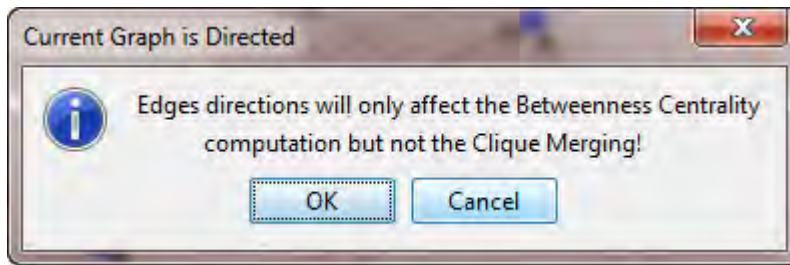
Ο δεύτερος αλγόριθμος στην κατηγορία των κοινοτήτων δημοσιεύθηκε το 2009 από ομάδα Ελλήνων επιστημόνων. Ο CiBC αποτελεί μια υβριδική προσέγγιση στον εντοπισμό κοινοτήτων, από την στιγμή που βασίζεται τόσο στις τοπικές όσο και στις γενικές ιδιότητες του γραφήματος.

Η υπολογιστική διαδικασία του συγκεκριμένου αλγορίθμου χωρίζεται σε δύο στάδια. Κατά το πρώτο στάδιο υπολογίζεται το Betweenness Centrality του γραφήματος, και στη συνέχεια οι κόμβοι του ομαδοποιούνται γύρω από αυτούς με την χαμηλότερη ενδιαμεσότητα. Στο δεύτερο στάδιο οι ομάδες αυτές επεξεργάζονται, και βάση μιας παραμέτρου “ s ” που αποτελεί είσοδο του χρήστη, σχηματίζονται τελικά οι κοινότητες του δικτύου. Για την συγχώνευση των ομάδων σε κοινότητες, πρέπει να ικανοποιείται η επόμενη εξίσωση

$$\frac{d_{out}(C)}{d_{in}(C)} \geq s \quad (6.2.1)$$

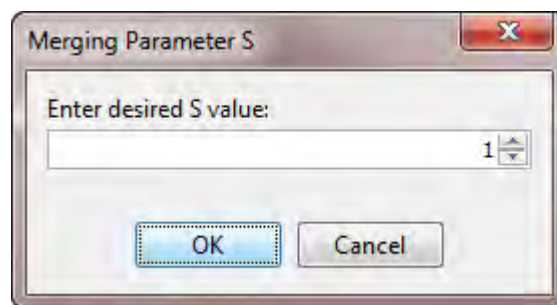
,όπου $d_{out}(C)$ είναι το πλήθος των εξερχομένων ακμών μιας ομάδας προς μια άλλη, και $d_{in}(C)$ είναι το πλήθος των ακμών που ανήκουν εξ ολοκλήρου στο εσωτερικό της υπό εξέταση ομάδας. Η τιμή της παραμέτρου εισόδου s , είναι μείζονος σημασίας για την δημιουργία των κοινοτήτων. Όπως φαίνεται και από την εξίσωση (6.2.1), όσο μεγαλύτερη τιμή έχει το s , τόσο λιγότερες είναι οι πιθανότητες συγχώνευσης των ομάδων σε κοινότητες. Επομένως, μικρή τιμή του s συνεπάγεται μεγαλύτερο βαθμό συγχώνευσης, άρα τελικά την εμφάνιση λιγότερων κοινοτήτων με περισσότερα μέλη.

Για την εκτέλεση του CiBC στο Anynet, ο χρήστης χρειάζεται να ακολουθήσει το μενού “Communities → CiBC”. Μπορεί να γίνει δεκτός οποιοσδήποτε τύπος γραφήματος, όμως η ενδεχόμενη παρουσία κατευθυνόμενων ή βεβαρημένων ακμών θα επηρεάσει μόνο τον υπολογισμό της Betweenness Centrality, και όχι την διαδικασία σχηματισμού κοινοτήτων. Έτσι σε περίπτωση που εντοπιστεί γράφημα με οποιοδήποτε από αυτά τα χαρακτηριστικά, το Anynet πριν προχωρήσει στην εκτέλεση του αλγορίθμου, εμφανίζει το παρακάτω ενημερωτικό μήνυμα.



Εικόνα 57 - Ενημερωτικό μήνυμα για την εκτέλεση του CiBC

Έπειτα ζητείται από τον χρήστη ο προσδιορισμός της επιθυμητής τιμής της παραμέτρου s . Συνήθως το s παίρνει την τιμή 1, στην προκειμένη όμως περίπτωση ο χρήστης μπορεί να δώσει οποιαδήποτε τιμή επιθυμεί, αρκεί αυτή να είναι μεγαλύτερη ή ίση του μηδενός με έως και τρία δεκαδικά ψηφία. Η είσοδος του s πραγματοποιείται από το παράθυρο της εικόνας 58.



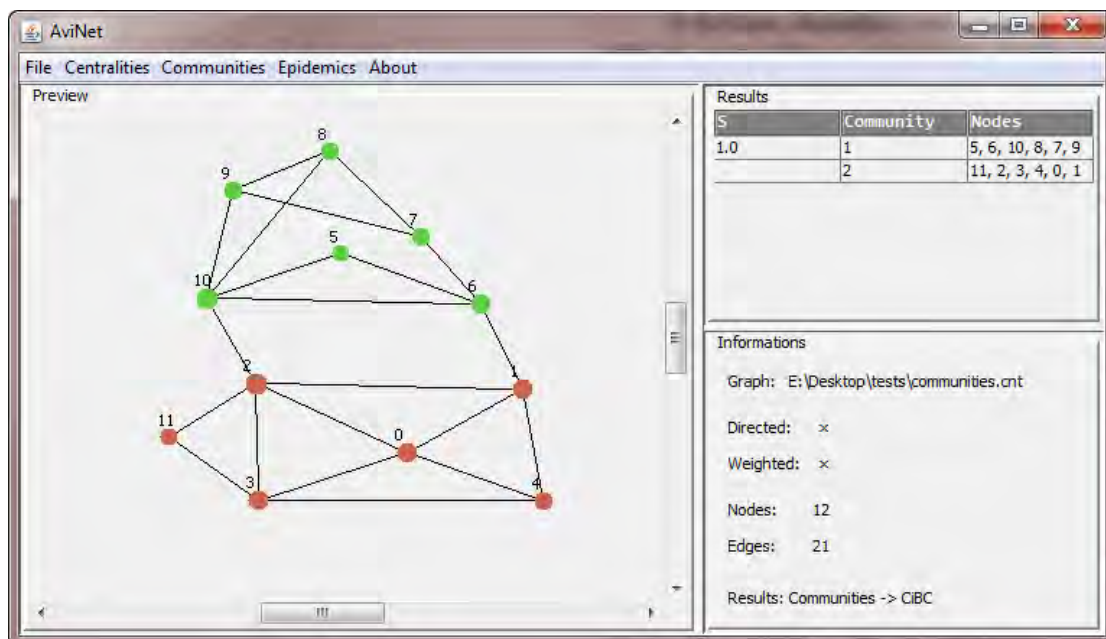
Εικόνα 58 - Καθορισμός παραμέτρου s

Αμέσως μετά ξεκινάει το πρώτο στάδιο εκτέλεσης του αλγορίθμου, ο οποίος καλεί για πρώτη και μοναδική φορά την διαδικασία υπολογισμού της Betweenness Centrality του γραφήματος, η οποία επιστρέφει σε μια λίστα τις ζητούμενες τιμές. Ο κόμβος με την μικρότερη Betweenness Centrality εντοπίζεται και δημιουργεί μια ομάδα προσκαλώντας σε αυτήν τους άμεσους γείτονές του, χαρακτηρίζοντας επίσης τόσο τον εαυτό του, όσο και αυτούς ως χρησιμοποιημένους. Η διαδικασία σύστασης των αρχικών ομάδων επαναλαμβάνεται κάθε φορά με την συμμετοχή των εναπομείναντων κόμβων μέχρι να εξεταστούν όλοι.

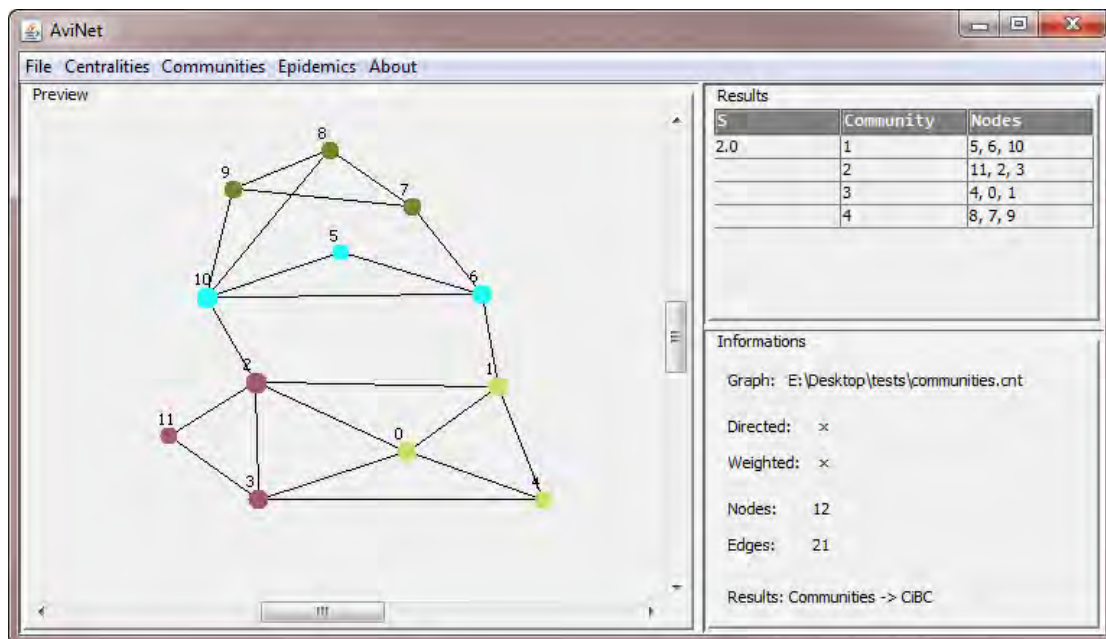
Στο δεύτερο στάδιο, δημιουργείται ένας πίνακας γεινίασης “cliqueMatrix” για τις ομάδες που έχουν προκύψει. Στην θέση (i, i) του εν λόγω πίνακα αποτυπώνεται το πλήθος των ακμών που ανήκουν καθ’ ολοκληρίαν στο εσωτερικό της ομάδας i , ενώ η θέση (i, j) φέρει τον αριθμό των ακμών που ξεκινούν από την ομάδα i και καταλήγουν στην ομάδα j . Μετά την κατασκευή του πίνακα, υπολογίζονται όλα τα

κλάσματα της μορφής $\frac{cliqueMatrix(i,j)}{cliqueMatrix(i,i)}$ και το μεγαλύτερο από αυτά συγκρίνεται με την παράμετρο s . Αν το επιλεγμένο κλάσμα είναι μεγαλύτερο ή ίσο από την τιμή του s , τότε οι ομάδες i και j συγχωνεύονται σε μια. Το δεύτερο στάδιο του αλγορίθμου επαναλαμβάνεται μέχρι να μην προκύπτει καμία συγχώνευση, γεγονός που θα σημαίνει και τον τερματισμό του αλγορίθμου.

Οι εικόνες 59 και 60 παρουσιάζουν τα αποτελέσματα του αλγορίθμου CiBC για το ίδιο γράφημα, με διαφορετικές όμως τιμές της παραμέτρου s . Στην πρώτη περίπτωση, έχουμε $s = 1$, κάτι που οδηγεί σε μεγαλύτερη συγχώνευση των ομάδων με συνέπεια την τελική δημιουργία δύο κοινοτήτων. Στην δεύτερη περίπτωση, παρατηρούμε ότι η αύξηση της τιμής του s , σε $s = 2$, δυσχεραίνει την διαδικασία ομαδοποίησης, με αποτέλεσμα τον τερματισμό του αλγορίθμου με την εμφάνιση τεσσάρων κοινοτήτων. Η πρώτη στήλη του πίνακα αποτελεσμάτων περιλαμβάνει την τιμή που δόθηκε στο s , ενώ η τρίτη περιέχει την λίστα με τους κόμβους κάθε κοινότητας.



Εικόνα 59 - CiBC για $s=1$



Εικόνα 60 - CiBC για $s=2$

Για την καλύτερη οπτικοποίηση των αποτελεσμάτων, το AviNet δίνει ένα τυχαίο χρώμα σε κάθε διαφορετική κοινότητα που εντοπίζεται, προκειμένου με τον τρόπο αυτό να διευκολύνεται και η ερμηνεία από πλευράς χρήστη.

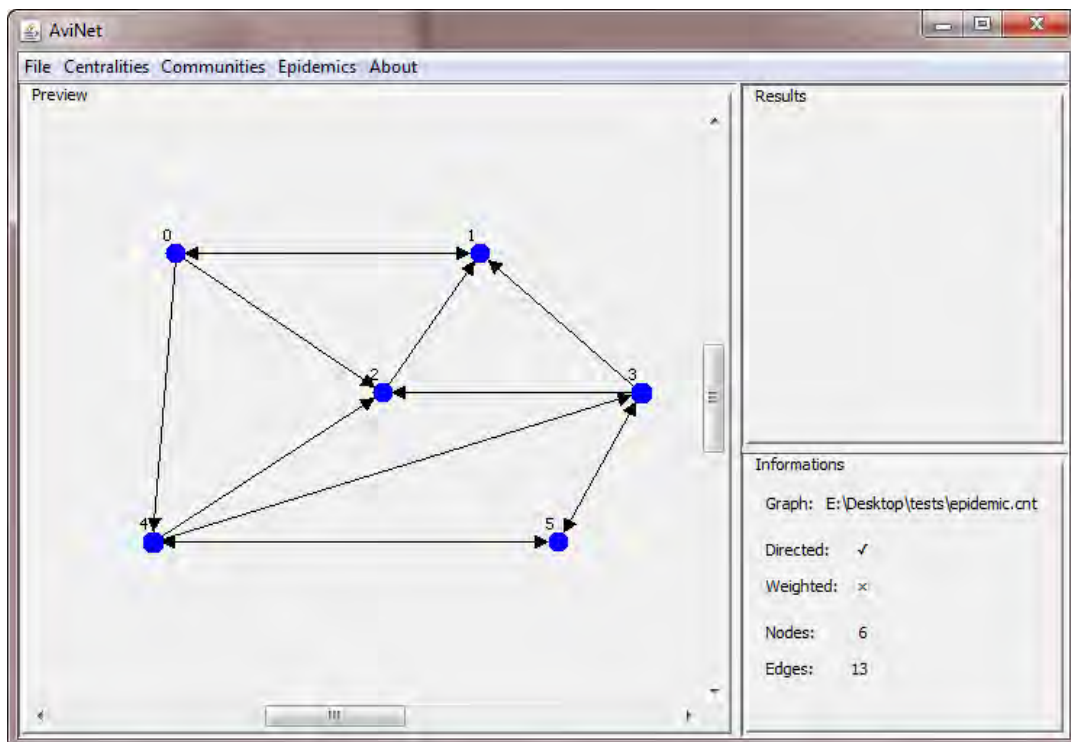
Όπως και στην παράγραφο 5.6, έτσι και εδώ σε περίπτωση που ο χρήστης αλλάξει την τοπολογία του γραφήματος μετά την εκτέλεση του αλγορίθμου προσθαφαιρώντας κόμβο ή ακμή, τα χρώματα θα επανέλθουν στην αρχική τους κατάσταση.

7 Επιδημιολογίες στο AviNet

Η μελέτη της εξάπλωσης επιδημιών στα σύνθετα δίκτυα, αφορά είτε την διάδοση μιας φήμης ή πληροφορίας, είτε την εξάπλωση μιας ασθένειας. Το πρόβλημα της μεγιστοποίησης επιρροής ορίζεται ως η εύρεση των σημαντικότερων κόμβων ενός κοινωνικού δικτύου, οι οποίοι μεγιστοποιούν την επιρροή στο εσωτερικό του. Η εξάπλωση των επιδημιών στα δίκτυα, δύναται να μελετηθεί με την χρήση μοντέλων διάχυσης. Το AviNet προσφέρει δύο από τα βασικότερα και ευρέως μελετημένα.

1. Linear Threshold (SIS)
2. Independent Cascade (SIR)

Για την εκτέλεση των αλγορίθμων που υλοποιούν τα δύο αυτά βασικά μοντέλα, αρκεί κανείς να ακολουθήσει το μενού “Epidemics”, μέσα από το οποίο εμφανίζονται οι διαθέσιμες επιλογές.



Εικόνα 61 - Epidemics στο AviNet

7.1 Linear Threshold (SIS)

Το αναγνωριστικό SIS δίπλα στο μοντέλο του Linear Threshold αφορά τις καταστάσεις που μπορεί να βρεθεί ένας κόμβος του γραφήματος κατά την διαδικασία εξάπλωσης της επιδημίας. Πιο συγκεκριμένα ένας κόμβος στο μοντέλο SIS (Susceptible Infected Susceptible) μπορεί κάθε χρονική στιγμή να είναι είτε ευπαθής στην μόλυνση (Susceptible), είτε ήδη μολυσμένος (Infected). Το Linear Threshold περιγράφεται αρχικά από τους εξής κανόνες.

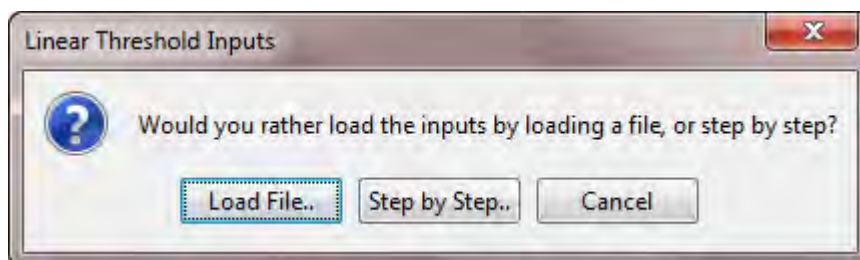
- Ένας κόμβος v έχει ένα τυχαίο κατώφλι $\theta_v \sim U[0,1]$, το οποίο ουσιαστικά αντικατοπτρίζει την πιθανότητα μόλυνσής του.
- Ένας κόμβος v υφίσταται επιρροή από έναν γείτονά του w , σύμφωνα με το βάρος b_{vw} , τέτοιο ώστε $\sum_w \text{neighbor of } v b_{vw} \leq 1$.
- Ένας κόμβος v μολύνεται όταν τουλάχιστον ένα κλάσμα των γειτόνων του είναι μολυσμένο $\sum_w \text{neighbor of } v b_{vw} \geq \theta$.

Σύμφωνα με το μοντέλο του Linear Threshold, σε κάθε διαφορετική επανάληψη οι μολυσμένοι κόμβοι του γραφήματος επιχειρούν να μολύνουν το σύνολο των μη μολυσμένων γειτόνων τους, για τους οποίους φέρουν άμεσες ακμές. Όπως περιγράφει και ο τρίτος κανόνας, ένας ανενεργός κόμβος μολύνεται, αν και μόνο αν η συνολική προσπάθεια επιρροής προς αυτόν ξεπερνάει το κατώφλι του. Ενδεχόμενη αποτυχία στην προσπάθεια επηρεασμού ενός κόμβου, δεν σηματοδοτεί την ανοσία του, αφού στις επόμενες επαναλήψεις, η συνεργασία περισσότερων γειτόνων του, μπορεί να επιφέρει τελικά την αλλαγή της κατάστασής του από ανενεργό σε ενεργό. Έτσι από το επόμενο κιάλας βήμα, οι νέοι ενεργοί-μολυσμένοι κόμβοι, συνεισφέρουν στην εξάπλωση της μόλυνσης του δικτύου σε συνεργασία με τους ήδη μολυσμένους κόμβους των προηγούμενων επαναλήψεων. Η εξάπλωση της μόλυνσης σταματά, τερματίζοντας τον αλγόριθμο, στην περίπτωση κατά την οποία κανένας από τους εναπομείναντες ανενεργούς κόμβους του γραφήματος δεν παρουσιάζει αλλαγή της κατάστασής του.

Για την εκτέλεση του μοντέλου Linear Threshold στο Avinet, ακολουθείται το μενού “Epidemics \rightarrow Linear Threshold”.

Βασική ωστόσο προϋπόθεση για την εκκίνηση του αλγορίθμου είναι ο καθορισμός των αρχικών τιμών του προβλήματος. Η διαδικασία αυτή μπορεί να γίνει είτε με το διάβασμα του ανάλογου αρχείου, είτε βήμα προς βήμα. Ζητώντας την εκτέλεση του συγκεκριμένου μοντέλου, ο χρήστης έρχεται αντιμέτωπος με το

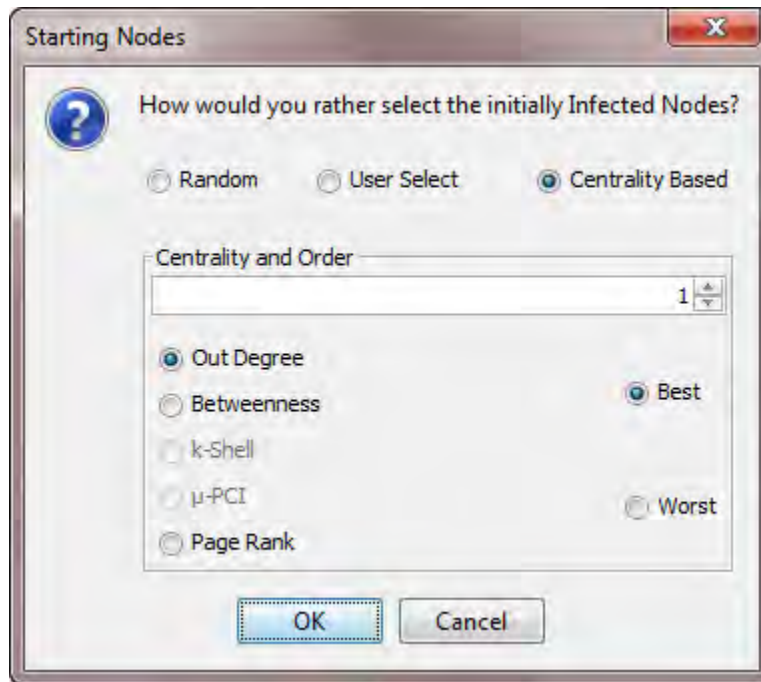
επόμενο παράθυρο που του ζητά να επιλέξει τον τρόπο με τον οποίο επιθυμεί να ορίσει τις εισόδους.



Εικόνα 62 - Παράθυρο επιλογής εισόδου για το Linear Threshold

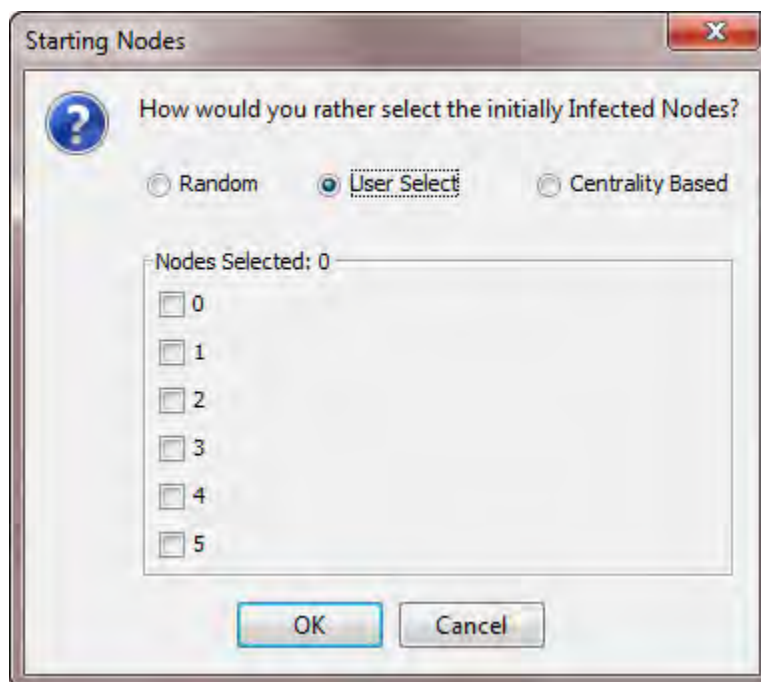
Αρχικά θα αναλύσουμε την διαδικασία εισαγωγής των απαιτούμενων δεδομένων βήμα προς βήμα, εξηγώντας παράλληλα αναλυτικότερα την σημασία της κάθε επιλογής, και στη συνέχεια θα ορίσουμε τους κανόνες που πρέπει να ακολουθεί ένα αρχείο εισόδου προκειμένου να μπορεί να γίνει σωστά η ανάγνωσή του από το Avinet.

Με την επιλογή “Step by Step..” ο χρήστης αντικρίζει το παράθυρο της εικόνας 63. Το παράθυρο με την επιλογή “Centrality Based”, δίνει την δυνατότητα στον χρήστη να επιλέξει τους αρχικούς κόμβους λαμβάνοντας υπόψιν τις τιμές των κεντρικότητων τους. Μέσα από τον μετρητή επιλέγεται το πλήθος των επιθυμητών κόμβων. Στο κάτω αριστερό μέρος ο χρήστης μπορεί να επιλέξει το Centrality βάση του οποίου θα γίνει η επιλογή, ενώ δεξιά αν επιθυμεί τους κόμβους με τις μεγαλύτερες ή τις μικρότερες τιμές. Παραδείγματος χάριν με την επιλογή 2 κόμβων, Page Rank και Best, το Avinet θα επιλέξει τους 2 κόμβους με το υψηλότερο Page Rank και θα τους χαρακτηρίσει ως αρχικά μολυσμένους, ενώ επιλέγοντας 1 κόμβο, Betweenness και Worst, θα επιλεγεί ως αρχικός ο κόμβος με το χειρότερο Betweenness Centrality.



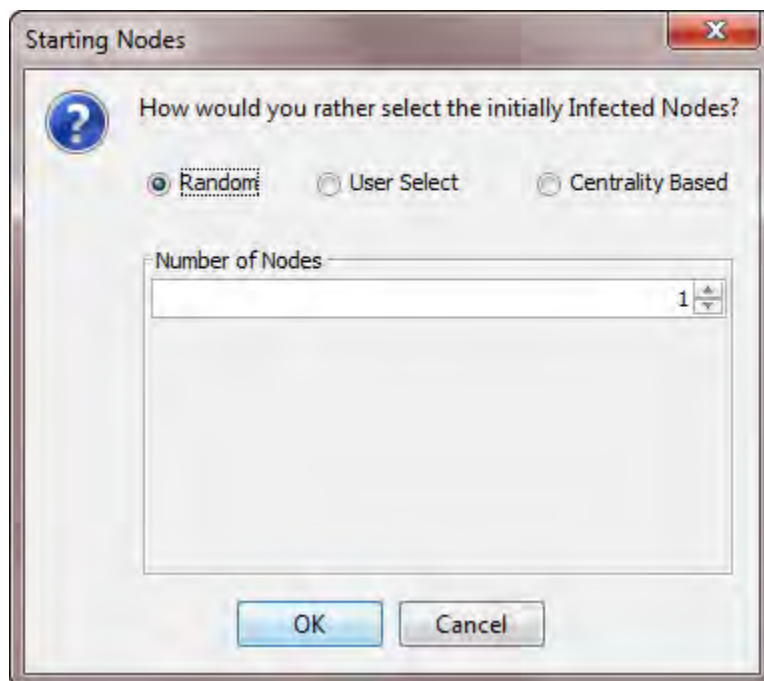
Εικόνα 63 - Επιλογή αρχικών κόμβων βάση Κεντρικότητας

Αν ο χρήστης επιθυμεί να επιλέξει χειροκίνητα τους αρχικά μολυσμένους κόμβους, τότε με την επιλογή "User Select" στα αριστερά του "Centrality Based", το παράθυρο της εικόνας 63 παίρνει την μορφή της εικόνας 64. Το πεδίο "Nodes Selected" λειτουργεί ως μετρητής για το πλήθος των κόμβων που έχουν επιλεγεί.



Εικόνα 64 - Επιλογή αρχικών κόμβων χειροκίνητα

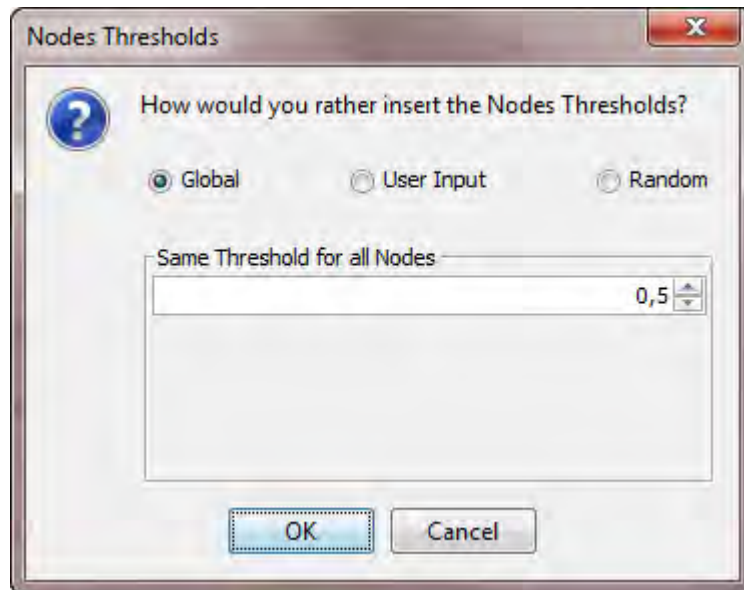
Σε περίπτωση κατά την οποία δεν ενδιαφερόμαστε ιδιαίτερα για την ταυτότητα των αρχικών κόμβων, παρά μόνο για το πλήθος τους, η επιλογή “Random” διαμορφώνει το παράθυρο όπως φαίνεται στην εικόνα 65. Στο σημείο αυτό εισάγεται μόνο το πλήθος των επιθυμητών κόμβων, και το πρόγραμμα τους διαλέγει τυχαία.



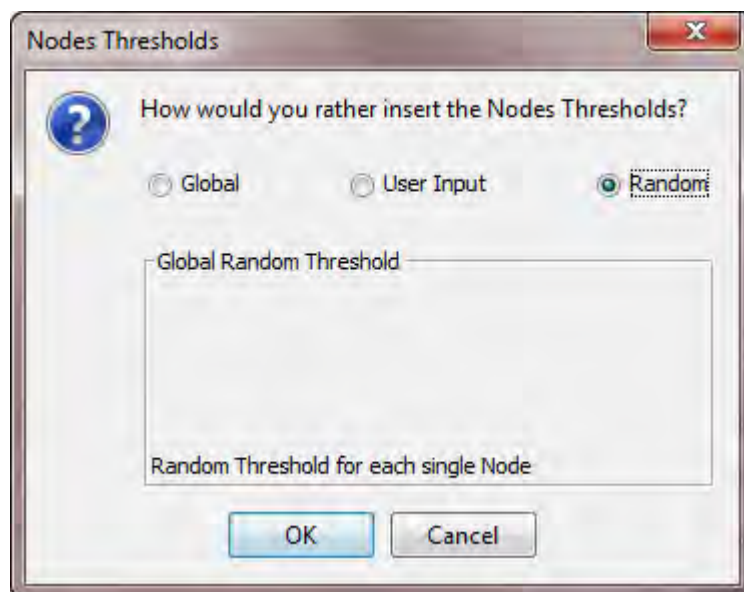
Εικόνα 65 - Τυχαία επιλογή αρχικά μολυσμένων κόμβων

Για τις ανάγκες του παραδείγματος που παρατίθεται στο τέλος της παρούσας παραγράφου, υποθέτουμε ότι ο χρήστης έχει ζητήσει τον κόμβο 5 του γραφήματος της εικόνας 61 ως μοναδικό αρχικά μολυσμένο, με τον τρόπο που παρουσιάζεται στην εικόνα 64.

Στη συνέχεια το πρόγραμμα ζητάει τον καθορισμό των τιμών κατωφλίου για τους κόμβους του γραφήματος που απέμειναν και δεν ορίστηκαν ως αρχικά μολυσμένοι. Όπως και με την είσοδο των αρχικών κόμβων, έτσι και εδώ παρέχεται πληθώρα επιλογών. Το παράθυρο της εικόνας 66, με την επιλογή “Global”, παρέχει την δυνατότητα ορισμού κοινού κατωφλίου για όλους τους κόμβους, μέσα από την πληκτρολόγηση της επιθυμητής τιμής στο πεδίο που παρέχεται. Ως δεύτερη επιλογή, το παράθυρο της εικόνας 67, με την επιλογή “Random” δίνει την εντολή στο πρόγραμμα να παράγει τυχαίες τιμές κατωφλίου για τον κάθε κόμβο ξεχωριστά.

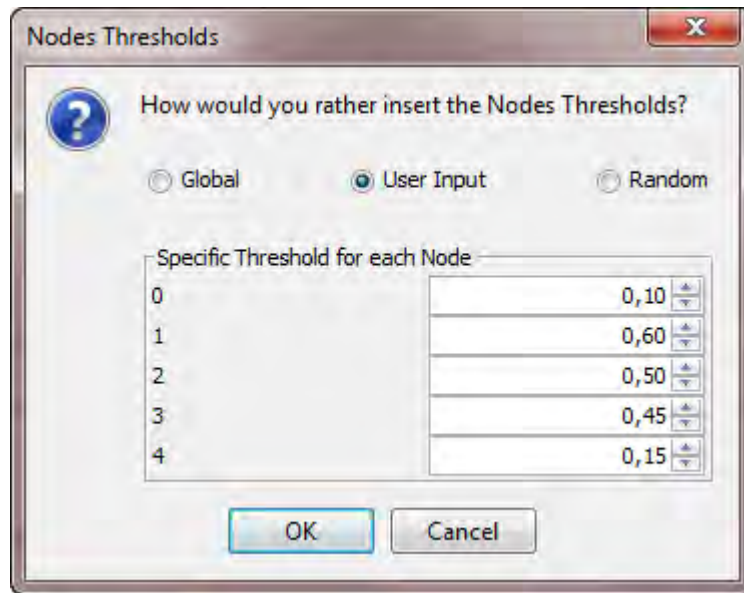


Εικόνα 66 - Κοινό threshold για όλους τους κόμβους



Εικόνα 67 - Τυχαίο threshold για κάθε κόμβο του γραφήματος ξεχωριστά

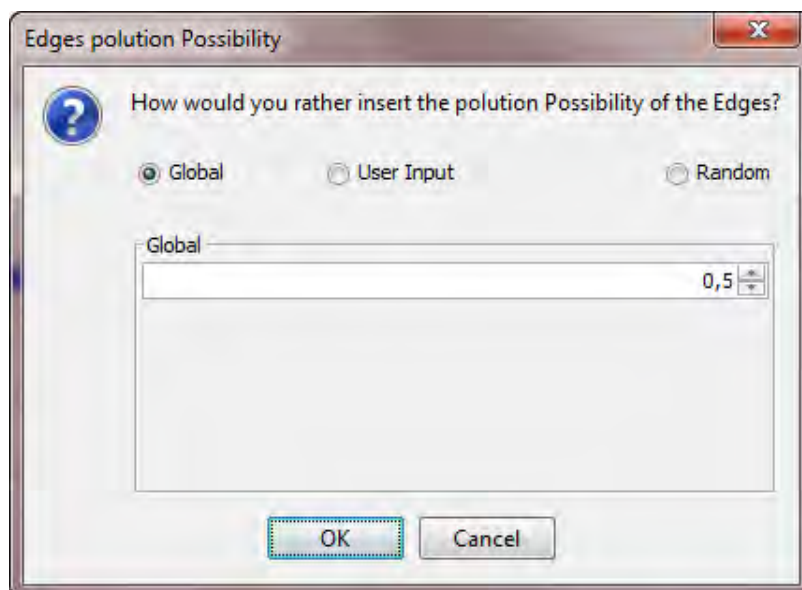
Ως Τρίτη επιλογή για την συγκεκριμένη διαδικασία, κάνοντας κλικ στο “User Input”, εμφανίζεται το παράθυρο της εικόνας 68, στο οποίο ο χρήστης μπορεί να θέσει χειροκίνητα τα thresholds για τον κάθε κόμβο ξεχωριστά. Αξίζει στο σημείο αυτό να παρατηρήσει κανείς την απουσία του κόμβου 5, μιας και ο κόμβος αυτός επιλέχθηκε στο προηγούμενο βήμα ως αρχικά μολυσμένος, συνεπώς θα ήταν ανούσιο να του τεθεί τιμή κατωφλίου. Για τις ανάγκες συνέχισης του παραδείγματος, οι τιμές όπως φαίνονται στην εικόνα τέθηκαν χειροκίνητα.



Εικόνα 68 - Εισαγωγή συγκεκριμένου κατωφλίου για κάθε κόμβο ξεχωριστά

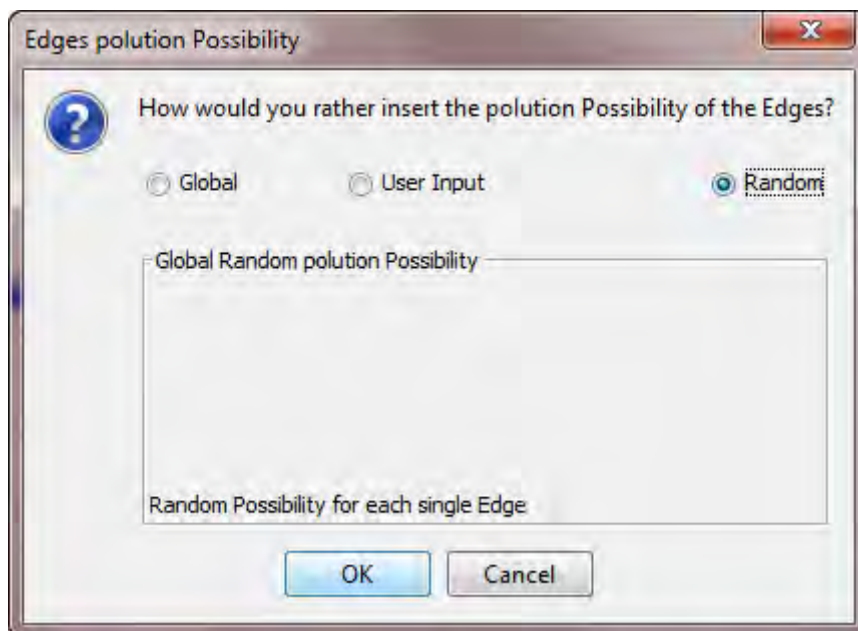
Για την ολοκλήρωση των απαιτούμενων δεδομένων του προβλήματος, μετά τον καθορισμό των τιμών κατωφλίου των κόμβων του γραφήματος, εμφανίζονται τα παράθυρα των εικόνων 69, 70 και 71, όμοια με εκείνα που αφορούν τα thresholds των κόμβων, ζητώντας από τον χρήστη να θέσει τις πιθανότητες κάθε ακμής να μολύνει τον κόμβο στον οποίο οδηγεί. Οι πιθανότητες αυτές ουσιαστικά αντικατοπτρίζουν την δύναμη που έχει ο κόμβος εκκίνησης της ακμής, να μολύνει τον κόμβο κατάληξής της.

Μέσα από το παράθυρο της εικόνας 69, ο χρήστης μπορεί να θέσει μια επιθυμητή πιθανότητα μόλυνσης, κοινή για όλες τις ακμές του γραφήματος.



Εικόνα 69 - Εισαγωγή συγκεκριμένης πιθανότητας μόλυνσης για όλες τις ακμές

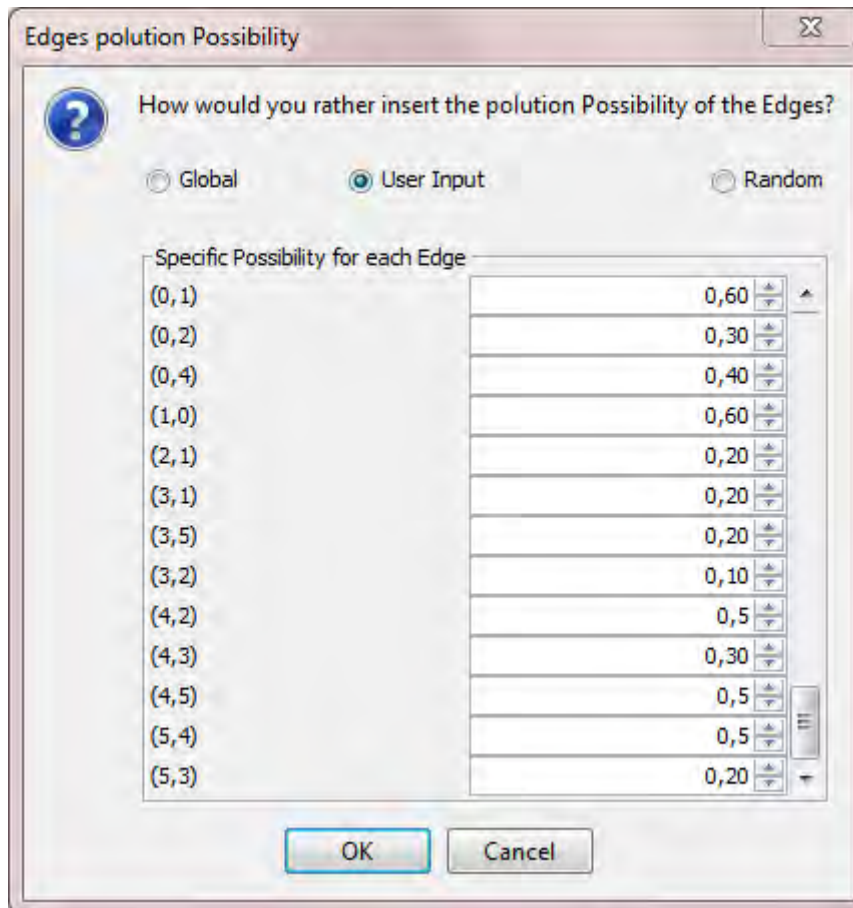
Η εικόνα 70 προκύπτει με την επιλογή “Random”, μέσα από την οποία ο χρήστης αφήνει το πρόγραμμα να επιλέξει τυχαίες πιθανότητες μόλυνσης για κάθε διαφορετική ακμή του γραφήματος.



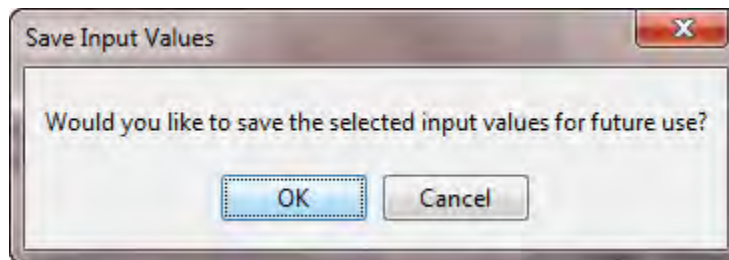
Εικόνα 70 - Τυχαία πιθανότητα μόλυνσης για κάθε διαφορετική ακμή

Τέλος, με την επιλογή “User Input” μέσα από το παράθυρο της εικόνας 71, ο χρήστης μπορεί να θέσει χειροκίνητα τις πιθανότητες μόλυνσης κάθε ακμής ξεχωριστά. Και σε αυτή την περίπτωση, για την ολοκλήρωση των εισόδων του παραδείγματος οι τιμές τέθηκαν μέσω της συγκεκριμένης λειτουργίας.

Με τον καθορισμό των πιθανοτήτων μόλυνσης των ακμών, το AniNet έχει όλα τα δεδομένα που χρειάζεται ώστε να εκτελέσει τον αλγόριθμο του μοντέλου Linear Threshold. Στο σημείο αυτό, και αφού η είσοδος των δεδομένων του προβλήματος πραγματοποιήθηκε χειροκίνητα, το πρόγραμμα ρωτά τον χρήστη αν επιθυμεί να αποθηκεύσει τα δεδομένα αυτά για μελλοντικές χρήσεις. Η λειτουργία αυτή είναι ιδιαίτερα σημαντική, αφού μπορεί να εξοικονομήσει πολύτιμο χρόνο σε επόμενες εκτελέσεις του αλγορίθμου, και παρέχεται μέσα από το παράθυρο της εικόνας 72.



Εικόνα 71 - Χειροκίνητος καθορισμός πιθανότητας μόλυνσης για κάθε ακμή ξεχωριστά



Εικόνα 72 - Αποθήκευση εισόδων για μελλοντική χρήση

Μετά την αποθήκευση των δεδομένων, δημιουργήθηκε το αρχείο της εικόνας 73. Παρατηρώντας το μπορεί κανείς να συμπεράνει την μορφή που πρέπει να έχει σε περίπτωση που θέλει να το δημιουργήσει μόνος του, προκειμένου να το χρησιμοποιήσει για τον καθορισμό των εισόδων του προβλήματος, αντί της διαδικασίας “Step by Step..”. Το διάβασμα των αρχείων εισόδου για τα Epidemics γίνεται μέσα από την επιλογή “Load File..” της εικόνας 62, και το AniNet είναι προγραμματισμένο ώστε να εμφανίζει μόνο τα κατάλληλα αρχεία κατά την περιήγηση, όπως και στην περίπτωση της ανάγνωσης γραφημάτων.

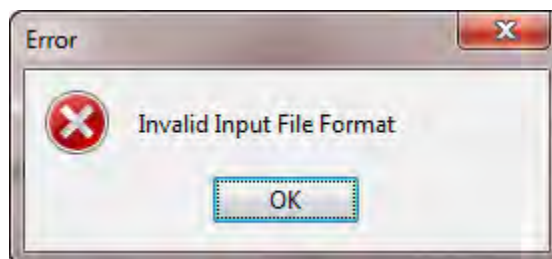
```
1 #Linear Threshold input
2
3 #List of Nodes, with their Thresholds and Edges Thresholds
4
5 0 0.1 1,0.6 2,0.3 4,0.4
6 1 0.6 0,0.6
7 2 0.5 1,0.2
8 3 0.45 1,0.2 5,0.2 2,0.1
9 4 0.15 2,0.5 3,0.3 5,0.5
10 5 -1.0 4,0.5 3,0.2
11
```

Εικόνα 73 - Αρχείο δεδομένων για το Linear Threshold

Πιο συγκεκριμένα, για το μοντέλο Linear Threshold τα αρχεία εισόδου των δεδομένων του προβλήματος, οφείλουν να υπακούν στους ακόλουθους κανόνες.

- Οι γραμμές που ξεκινούν με δίσηση (#) αναγνωρίζονται ως σχόλια, και συνεπώς δεν λαμβάνονται υπόψιν.
- Όλες οι υπόλοιπες γραμμές πρέπει να συμμορφώνονται με τις ακόλουθες προτάσεις. Το πρώτο στοιχείο της γραμμής αφορά το όνομα του εκάστοτε κόμβου, το οποίο χωρίζεται με ένα tab από την τιμή κατωφλίου του. Στη συνέχεια διαχωρισμένοι επίσης με tabs βρίσκονται οι γείτονες του κάθε κόμβου, μετά τα ονόματα των οποίων βρίσκονται οι τιμές των πιθανοτήτων μόλυνσης των ακμών που τους ενώνουν με τον τρέχον κόμβο.

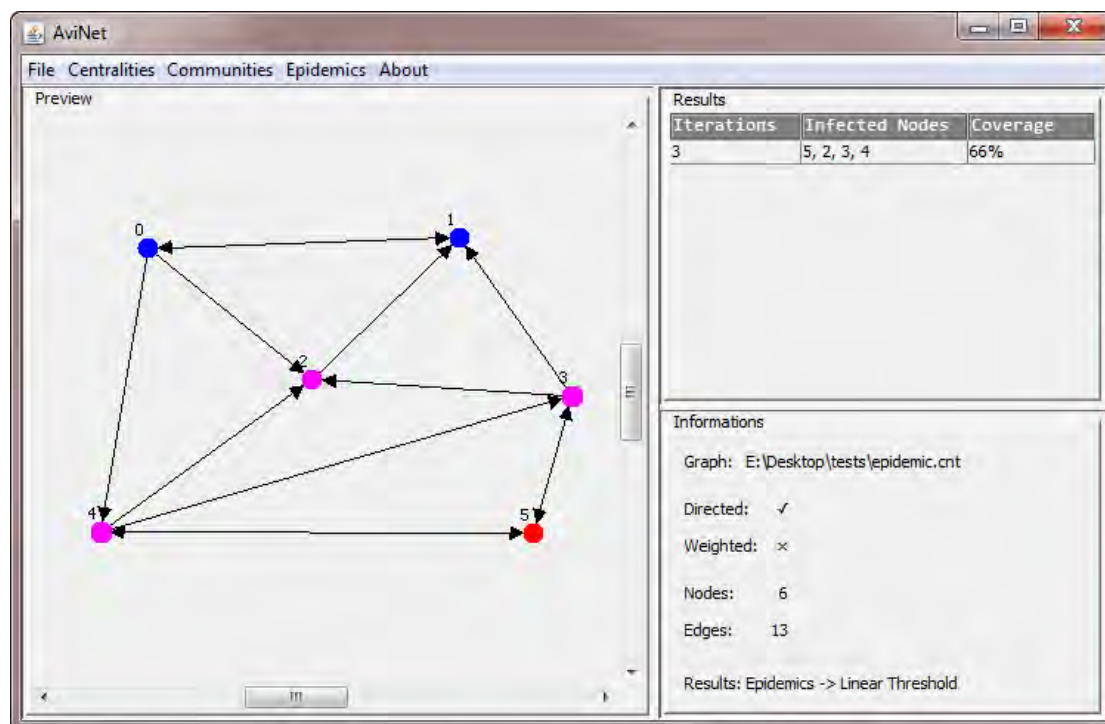
Αν ο χρήστης προσπαθήσει να εισάγει τα δεδομένα του προβλήματος μέσω αρχείου που δεν συμμορφώνεται στους παραπάνω κανόνες, τότε το πρόγραμμα θα εμφανίσει το ακόλουθο μήνυμα λάθους, αδυνατώντας να προχωρήσει στην εκτέλεση του αλγορίθμου.



Εικόνα 74 - Παράθυρο ενημέρωσης ακατάλληλου αρχείου

Το AviNet για την εκτέλεση του μοντέλου Linear Threshold δημιουργεί έναν πίνακα ο οποίος φέρει αρκετές ομοιότητες με τον πίνακα γειτνίασης του γραφήματος. Στην θέση (i, i) βρίσκεται αποθηκευμένο το κατώφλι του κόμβου i , ενώ στην περίπτωση που ο κόμβος αυτός είναι μολυσμένος, λαμβάνει την τιμή -1 . Οι υπόλοιπες θέσεις περιέχουν τις πιθανότητες μόλυνσης των ακμών. Συγκεκριμένα στην θέση (i, j) βρίσκεται η πιθανότητα μόλυνσης της ακμής που ξεκινά από τον κόμβο i και καταλήγει στον κόμβο j . Με τον τρόπο αυτό η στήλη i του πίνακα περιέχει τις πιθανότητες μόλυνσης των ακμών που καταλήγουν στον κόμβο i . Έτσι διευκολύνονται σε μεγάλο βαθμό οι υπολογισμοί, αφού διατρέχοντας κανείς τις στήλες του κατασκευασμένου πίνακα, δύναται να εντοπίσει την συνολική απειλή που συγκεντρώνεται πάνω σε έναν κόμβο, προσέχοντας απλά να αθροίζει μόνο τις τιμές που προέρχονται από μολυσμένους κόμβους. Η εκτέλεση του αλγορίθμου ολοκληρώνεται όταν παρατηρηθεί επανάληψη κατά την οποία δεν μολύνθηκε κανένας κόμβος του γραφήματος.

Η εικόνα 75 παρουσιάζει τα αποτελέσματα του μοντέλου Linear Threshold για το γράφημα της εικόνας 61, με τις εισόδους που παρουσιάζονται στις εικόνες 64, 68 και 71. Η πρώτη στήλη του πίνακα αποτελεσμάτων μαρτυρά το πλήθος των επαναλήψεων που εκτελέστηκαν. Στην δεύτερη στήλη παρουσιάζεται η λίστα που περιλαμβάνει τους μολυσμένους κόμβους, ενώ στην τρίτη παρέχεται το συνολικό ποσοστό κάλυψης του δικτύου. Στο Preview πάνελ με κόκκινο χρώμα παρατηρούνται οι κόμβοι που ορίστηκαν ως αρχικά μολυσμένοι, ενώ με ροζ οι κόμβοι που μολύνθηκαν κατά την εφαρμογή του μοντέλου.



Εικόνα 75 - Το Linear Threshold στο AviNet

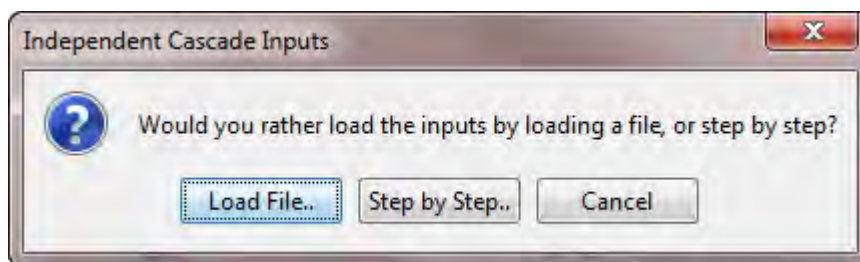
7.2 Independent Cascade (SIR)

Το δεύτερο μοντέλο διάχυσης για την μελέτη εξάπλωσης επιδημιών που προσφέρει το AniNet, ονομάζεται Independent Cascade και στηρίζεται στο SIR, το οποίο επικεντρώνει την προσοχή του στην ατομική και ανεξάρτητη αλληλεπίδραση και επιρροή των κόμβων σε ένα κοινωνικό δίκτυο. Στο μοντέλο SIR, τα αναγνωριστικά του οποίου σημαίνουν Susceptible Infected Recovered, ένας κόμβος μπορεί να είναι είτε ευπαθής στην μόλυνση (Susceptible), είτε μολυσμένος (Infected), είτε να έχει αποκτήσει ανοσία (Recovered).

Στο Independent Cascade, κάθε ακμή φέρει μια πιθανότητα μόλυνσης, όμοια με το μοντέλο Linear Threshold, και η επιρροή του δικτύου πολλαπλασιάζεται μέσω των ενεργών κόμβων, οι οποίοι ανεξάρτητα προσπαθούν να μολύνουν τους γείτονές τους ανάλογα με τις πιθανότητες αυτές. Όταν ένας κόμβος ενεργοποιείται, έχει μια μοναδική ευκαιρία να μολύνει κάθε μη μολυσμένο γείτονά του. Αυτό σημαίνει ότι αν ένας κόμβος v προσπαθήσει να μολύνει έναν άμεσο γείτονά του w , και δεν τα καταφέρει, τότε ο w αποκτά ανοσία προς τον v , όχι όμως προς τους υπόλοιπους κόμβους του δικτύου, οι οποίοι ενδεχομένως να επιχειρήσουν να τον μολύνουν ανεξάρτητα. Η προσπάθεια μόλυνσης ενός κόμβου επιτυγχάνεται με πιθανότητα p_{vw} .

Η εκτέλεση του μοντέλου Independent Cascade στο AniNet πραγματοποιείται ακολουθώντας το μενού “Epidemics → Independent Cascade”.

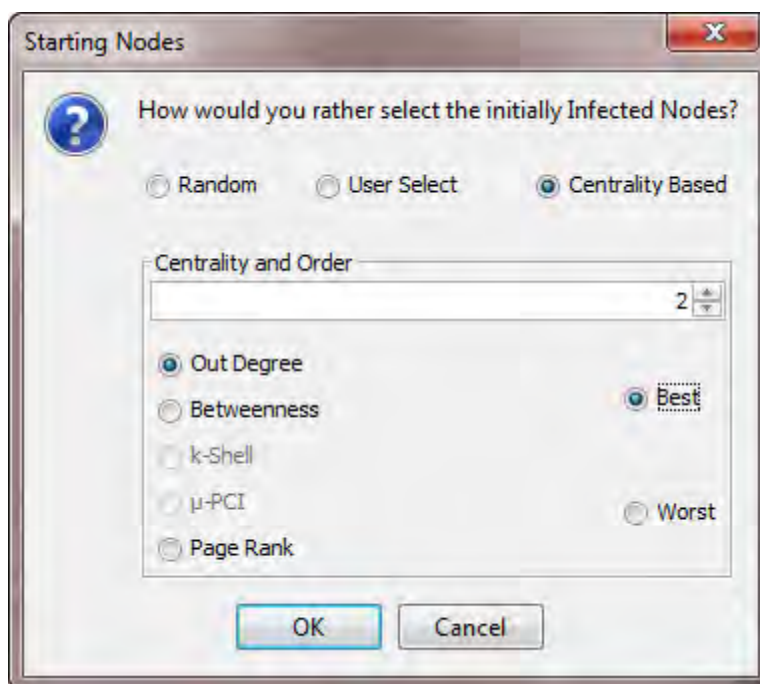
Όπως και στην περίπτωση του Linear Threshold, πριν την εκκίνηση του αλγορίθμου είναι απαραίτητος ο καθορισμός των αρχικών τιμών του προβλήματος. Αυτό μπορεί να γίνει είτε βήμα προς βήμα χρησιμοποιώντας τον οδηγό που προσφέρεται, είτε με το άνοιγμα του κατάλληλου αρχείου. Μετά την επιλογή του μοντέλου Independent Cascade, παρουσιάζεται στον χρήστη το επόμενο παράθυρο επιλογών, προκειμένου να διαλέξει με ποιον από τους δύο τρόπους επιθυμεί να εισάγει τις εισόδους.



Εικόνα 76 - Παράθυρο επιλογής εισόδου για το Independent Cascade

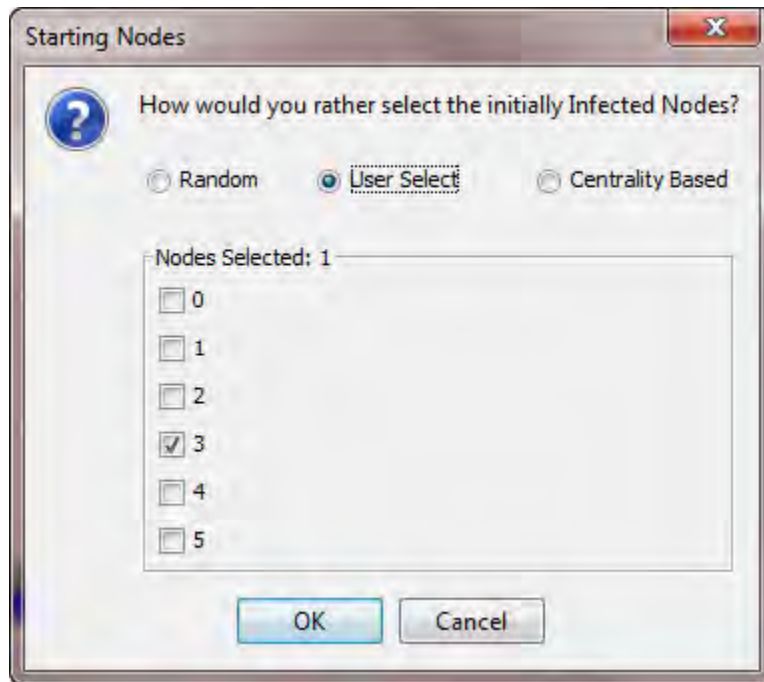
Αναλύοντας σε πρώτη φάση την διαδικασία εισαγωγής των δεδομένων βήμα προς βήμα μέσω της επιλογής “Step by Step..”, θα παρουσιαστούν παράλληλα και οι αρχικές τιμές που εισήχθησαν για τις ανάγκες του γραφήματος της εικόνας 61, που χρησιμοποιήθηκε για το παράδειγμα που παρατίθεται στο τέλος της παραγράφου.

Το πρώτο παράθυρο που εμφανίζεται για την εισαγωγή των δεδομένων, παρουσιάζεται στην εικόνα 77. Μέσω της επιλογής “Centrality Based”, μπορούμε να επιλέξουμε το πλήθος των αρχικά μολυσμένων κόμβων, με βάση τις τιμές που προκύπτουν από τους αλγορίθμους των κεντρικότητων. Συνεπώς με την επιλογή 2 αρχικών κόμβων, “Out Degree” και “Best”, το πρόγραμμα θα ορίσει ως υπεύθυνους για την εξάπλωση της επιδημίας τους 2 κόμβους με τον μεγαλύτερο βαθμό εξόδου, μιας και το γράφημα του παραδείγματος είναι κατευθυνόμενο. Σε περίπτωση που το γράφημα ήταν μη κατευθυνόμενο, θα εμφανιζόταν η επιλογή “Degree” στην θέση της “Out Degree”. Αν κάποιος αλγόριθμος κεντρικότητας δεν εφαρμόζεται λόγω της φύσης του γραφήματος, τότε η ανάλογη επιλογή είναι αχνή και δεν παρέχεται.



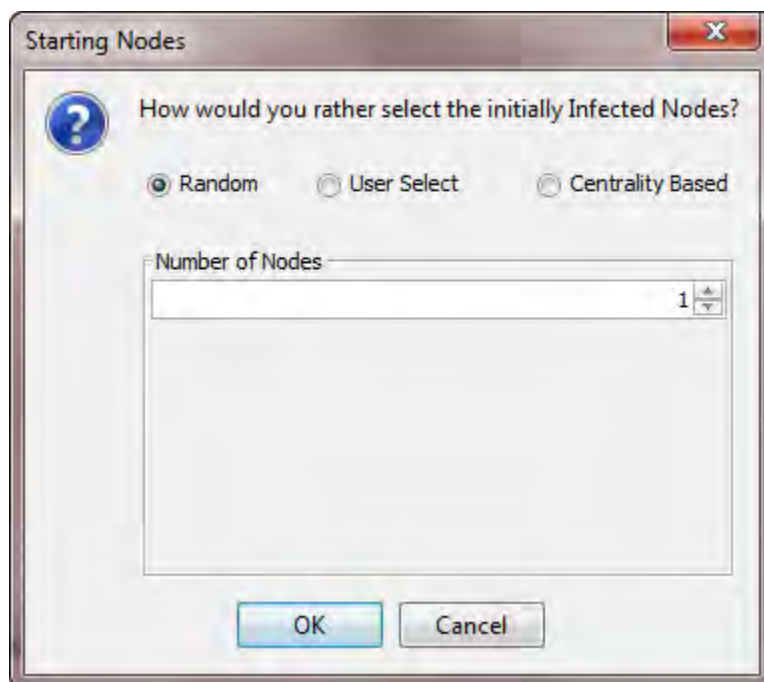
Εικόνα 77 - Επιλογή αρχικών κόμβων βάση κεντρικότητων

Για την χειροκίνητη επιλογή των αρχικών κόμβων του προβλήματος, αρκεί να χρησιμοποιήσει κανείς το “User Select”, δίνοντας στο παράθυρο της εικόνας 77 την μορφή της εικόνας 78. Με κλικ πάνω στους επιθυμητούς κόμβους, αυτοί επιλέγονται ως αρχικοί, με το πεδίο “Nodes Selected” να λειτουργεί και ως μετρητής.



Εικόνα 78 - Επιλογή αρχικά μολυσμένων κόμβων χειροκίνητα

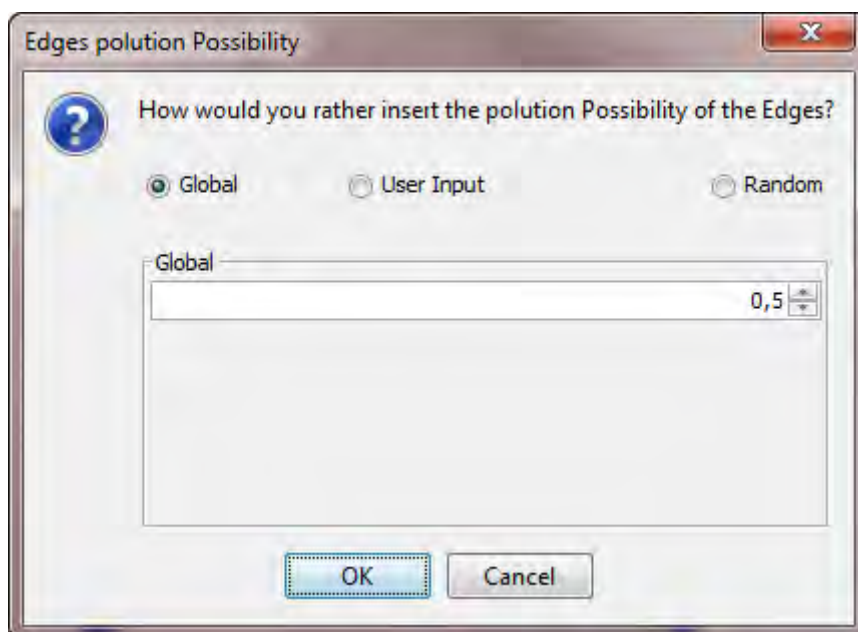
Σε διαφορετική περίπτωση κατά την οποία ο χρήστης ενδιαφέρεται μόνο για το πλήθος των αρχικών κόμβων, και όχι για την ταυτότητά ή ιδιότητά τους, μπορεί να επιλέξει το "Random", όπως δεικνύει και η εικόνα 79, εισάγοντας απλά τον επιθυμητό αριθμό.



Εικόνα 79 - Τυχαία επιλογή αρχικών κόμβων

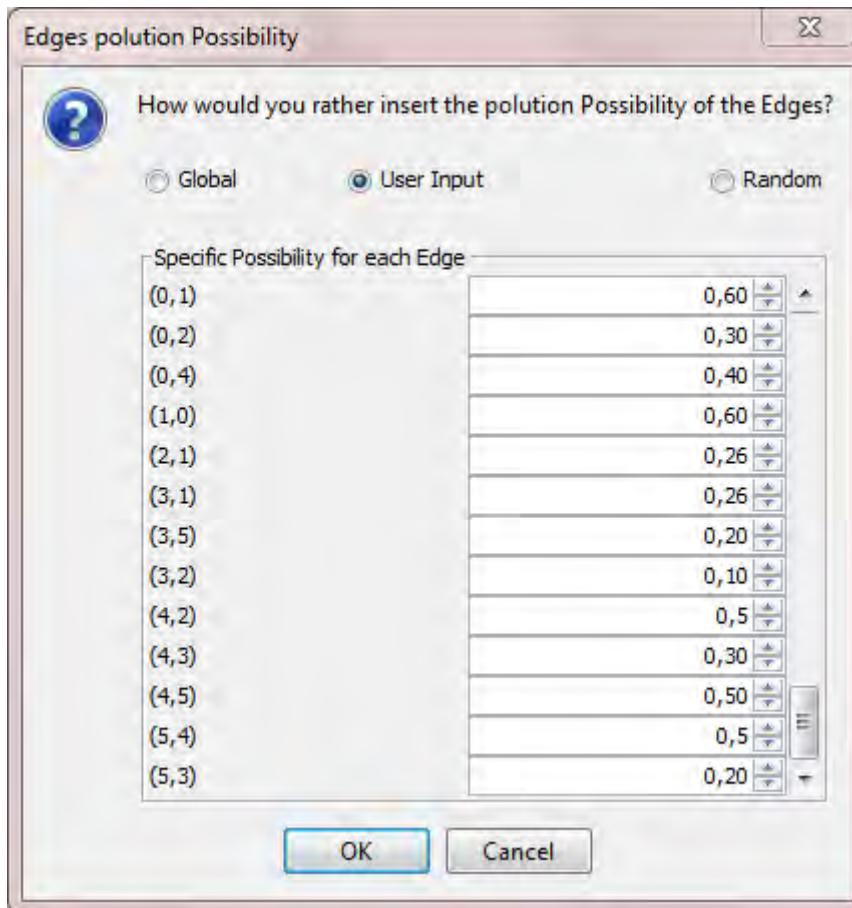
Για τις ανάγκες του παραδείγματος έγινε χρήση του παραθύρου της εικόνας 78, διαλέγοντας με τον τρόπο αυτό τον κόμβο 3 ως αρχικά μολυσμένο. Με κλικ στο κουμπί “OK”, η διαδικασία συνεχίζεται ζητώντας την εισαγωγή των πιθανοτήτων μόλυνσης των ακμών. Όσο μεγαλύτερη είναι η τιμή που αντιπροσωπεύει μια ακμή (i, j), τόσο πιθανότερο είναι αυτή να μολύνει τον κόμβο j , μιας και ουσιαστικά πρόκειται για την ικανότητα επιρροής του κόμβου j από τον i .

Μέσα από το παράθυρο της εικόνας 80, ο χρήστης μπορεί να θέσει μια επιθυμητή πιθανότητα μόλυνσης, κοινή για όλες τις ακμές του γραφήματος.

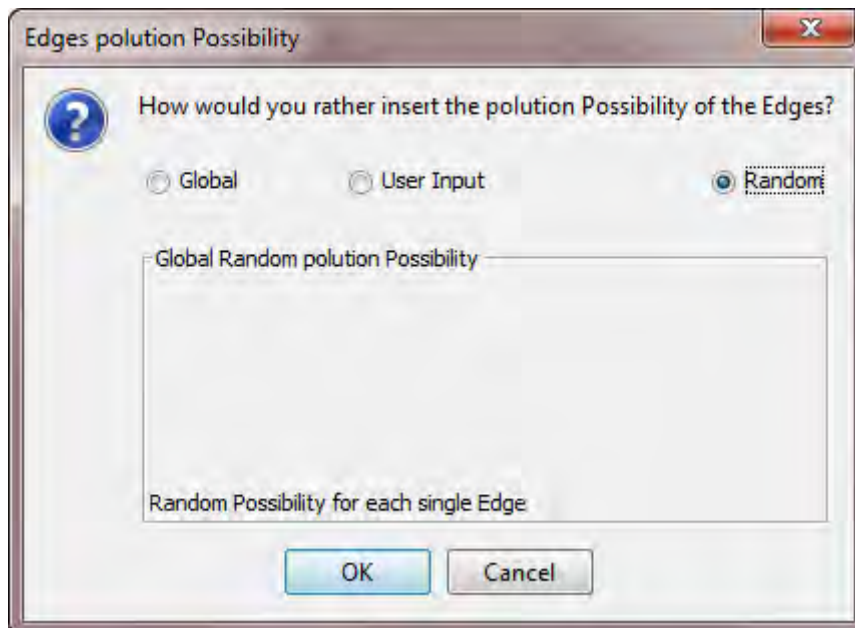


Εικόνα 80 - Συγκεκριμένη πιθανότητα μόλυνσης, ίδια για όλες τις ακμές

Η επιλογή “User Input” μέσω της οποίας ο χρήστης μπορεί να εισάγει την επιθυμητή πιθανότητα μόλυνσης κάθε ακμής ξεχωριστά, απεικονίζεται στο παράθυρο της εικόνας 81. Τέλος, μέσω της επιλογής “Random” ο χρήστης μπορεί να αφήσει στο πρόγραμμα την ελευθερία να επιλέξει εκείνο τυχαία την πιθανότητα μόλυνσης κάθε διαφορετικής ακμής, όπως δείχνει και το παράθυρο της εικόνας 82. Για τις ανάγκες του παραδείγματος, έγινε χρήση του παραθύρου της εικόνας 81, μέσα από την οποία γίνονται εμφανής αναλυτικά οι τιμές που ορίστηκαν.

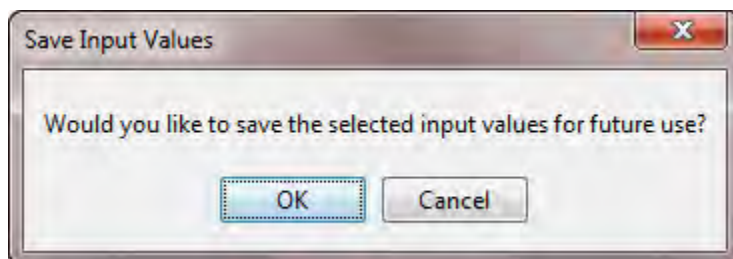


Εικόνα 81 - Εισαγωγή διαφορετικής πιθανότητας μόλυνσης κάθε ακμής από τον χρήστη



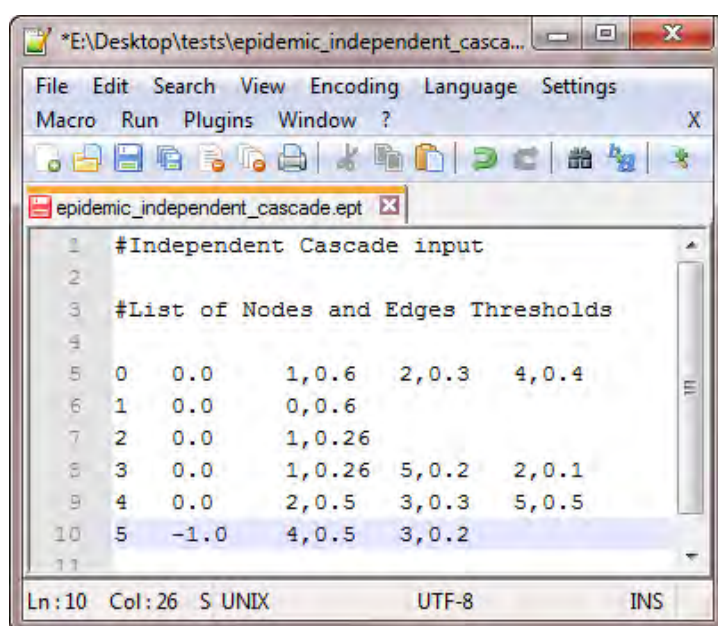
Εικόνα 82 - Τυχαία πιθανότητα μόλυνσης για κάθε ακμή ξεχωριστά

Μετά την ολοκλήρωση εισαγωγής των επιθυμητών πιθανοτήτων μόλυνσης των ακμών, το AniNet είναι σε θέση να προχωρήσει στην εκτέλεση του αλγορίθμου. Ωστόσο παρέχει στον χρήστη την δυνατότητα της αποθήκευσης των τιμών εισόδου, ώστε να είναι σε θέση να τρέξει ξανά το συγκεκριμένο παράδειγμα στο μέλλον, δίχως την ανάγκη εισαγωγής των δεδομένων από την αρχή. Πριν την εκτέλεση του αλγορίθμου, εμφανίζεται το ακόλουθο παράθυρο στον χρήστη, παρέχοντάς του την συγκεκριμένη λειτουργία.



Εικόνα 83 - Αποθήκευση δεδομένων για μελλοντική χρήση

Μετά την αποθήκευση των δεδομένων, δημιουργήθηκε το αρχείο της εικόνας 84. Παρατηρώντας το μπορεί κανείς να αναλύσει την μορφή που πρέπει να έχει σε περίπτωση που θέλει να το δημιουργήσει μόνος του, προκειμένου να το χρησιμοποιήσει για την εισαγωγή των δεδομένων του προβλήματος. Το διάβασμα των αρχείων εισόδου για τα Epidemics μπορεί να γίνει μέσα από την επιλογή "Load File.." της εικόνας 76, και το AniNet για διευκόλυνση του χρήστη είναι προγραμματισμένο ώστε να εμφανίζει μόνο τα κατάλληλα αρχεία κατά την περιήγηση.

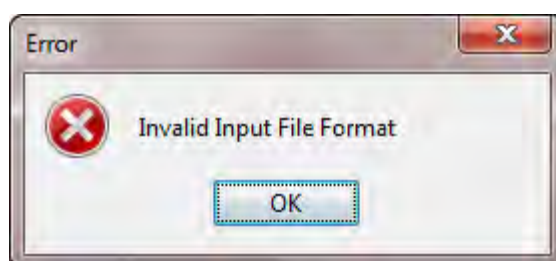


Εικόνα 84 - Αρχείο δεδομένων για το Independent Cascade

Πιο αναλυτικά, για το Independent Cascade μοντέλο, τα αρχεία εισόδου οφείλουν να ακολουθούν τους παρακάτω κανόνες, προκειμένου το Anynet να είναι σε θέση να τα διαβάσει σωστά, και να προχωρήσει στην εκτέλεση του αλγορίθμου.

- Οι γραμμές που ξεκινούν με δίεση (#) αναγνωρίζονται ως σχόλια και δεν λαμβάνονται υπόψιν.
- Το αρχείο πρέπει να περιλαμβάνει ακόμα η γραμμές, όσοι και οι κόμβοι του γραφήματος. Το πρώτο στοιχείο της κάθε γραμμής είναι το όνομα του κόμβου, το οποίο ακολουθείται από ένα tab και εν συνεχεία την δήλωση για το αν ο συγκεκριμένος κόμβος θα θεωρηθεί αρχικά μολυσμένος. Σε περίπτωση που η τιμή που ακολουθεί είναι -1, τότε ο τρέχων κόμβος ξεκινά ως μολυσμένος-ενεργός, διαφορετικά το πεδίο αυτό φέρει την τιμή 0. Έπειτα για κάθε κόμβο ορίζονται οι πιθανότητες μόλυνσης των εξερχόμενων ακμών του, μαζί με τους κόμβους κατάληξής τους. Για παράδειγμα η τιμή (4,0.4) της γραμμής που αναφέρεται στον κόμβο 0, ορίζει πως η ακμή που ξεκινά από τον κόμβο 0 με προορισμό τον 4, θα φέρει πιθανότητα μόλυνσης (0.4).

Αν ο χρήστης εισάγει αρχείο που δεν συμμορφώνεται με την μορφή που περιγράφουν οι παραπάνω κανόνες, τότε το Anynet θα αποτύχει κατά την ανάγνωση του, εμφανίζοντας το μήνυμα της εικόνας 85, με συνέπεια την μη εκτέλεση του μοντέλου.

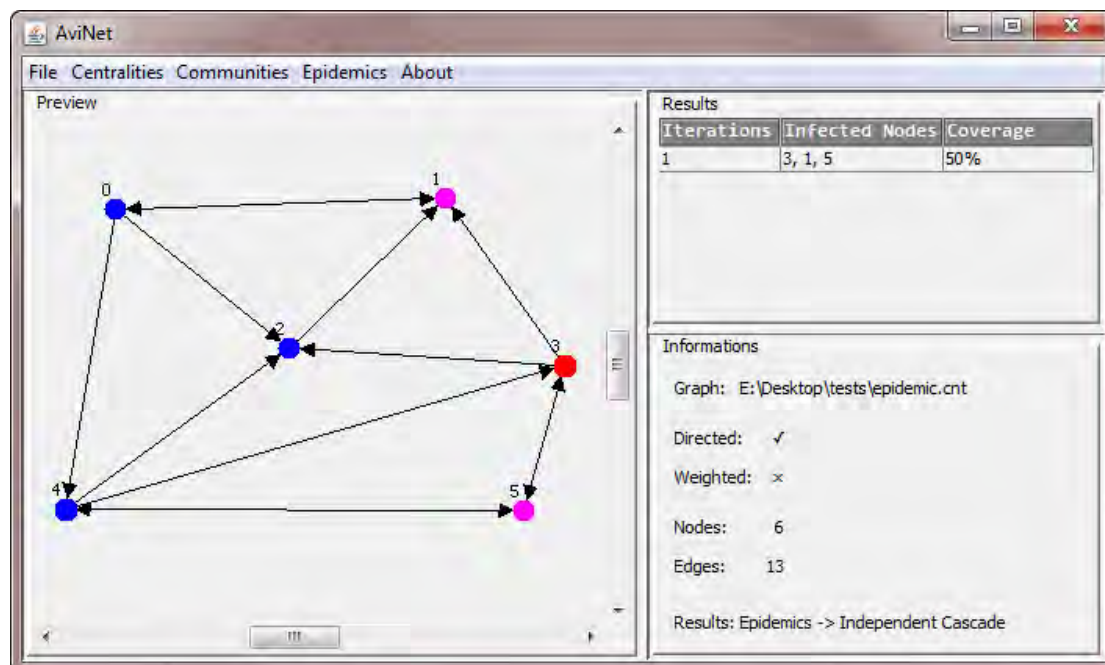


Εικόνα 85 - Μήνυμα ακατάλληλου αρχείου στο Independent Cascade

Ο αλγόριθμος που υλοποιεί το μοντέλο Independent Cascade στο Anynet δημιουργεί αρχικά τον πίνακα “epidemicMatrix”, ο οποίος στην θέση (i, i) έχει την τιμή -1 αν ο κόμβος i είναι αρχικά μολυσμένος, ενώ σε διαφορετική περίπτωση έχει την τιμή 0. Συνεχίζοντας, σε κάθε θέση (i, j), με $i \neq j$, βρίσκεται η πιθανότητα μόλυνσης της ακμής που ξεκινάει από τον κόμβο i και καταλήγει στον j. Έπειτα, και με την βοήθεια του πίνακα “epidemicMatrix”, κάθε ενεργός κόμβος προσπαθεί να μολύνει τους άμεσους ανενεργούς γείτονές του, χαρακτηρίζοντας παράλληλα τον εαυτό του ως χρησιμοποιημένο μετά το πέρας της προσπάθειάς του. Η προσπάθεια μόλυνσης του κόμβου j, από τον κόμβο i, πραγματοποιείται παράγοντας μια τυχαία πιθανότητα

για την ακμή (i, j) . Αν η τιμή της πιθανότητας αυτής είναι μικρότερη από την πιθανότητα μόλυνσης της συγκεκριμένης ακμής, τότε ο κόμβος j μολύνεται, διαφορετικά αποκτά ανοσία ως προς την επιρροή του κόμβου i . Το σύνολο των κόμβων που ενεργοποιήθηκε, θα προσπαθήσει με την σειρά του ακολουθώντας την ίδια διαδικασία να μολύνει τους άμεσους γείτονές του κατά την επόμενη επανάληψη. Η διαδικασία τερματίζει, όταν παρατηρηθεί επανάληψη κατά την οποία κανένας κόμβος δεν μολύνθηκε.

Η εικόνα 86 παρουσιάζει τα αποτελέσματα του μοντέλου Independent Cascade για το γράφημα της εικόνας 61, με τις εισόδους που περιγράφονται στις εικόνες 78 και 81. Λόγω της ιδιαιτερότητας του συγκεκριμένου μοντέλου εξάπλωσης επιδημιών, μέσω της εμπλοκής τυχαίων πιθανοτήτων, κάθε διαφορετικό τρέξιμο του αλγορίθμου δύναται να παραδώσει διαφορετικά αποτελέσματα, ακόμα και για τις ίδιες τιμές της εισόδου. Στο Preview πάνελ, για την διευκόλυνση της οπτικής ερμηνείας των αποτελεσμάτων, με κόκκινο χρώμα παρουσιάζονται οι κόμβοι που επιλέχθηκαν ως αρχικά μολυσμένοι, ενώ με ροζ αυτοί που μολύνθηκαν κατά την εκτέλεση της διαδικασίας. Στον πίνακα αποτελεσμάτων, η πρώτη στήλη δηλώνει τον αριθμό των επαναλήψεων που εκτελέστηκαν μέχρι τον τερματισμό του αλγορίθμου, ενώ η δεύτερη το σύνολο των κόμβων που κατέληξαν μολυσμένοι. Τέλος η τρίτη στήλη δηλώνει το ποσοστό κάλυψης της μόλυνσης στο δίκτυο.

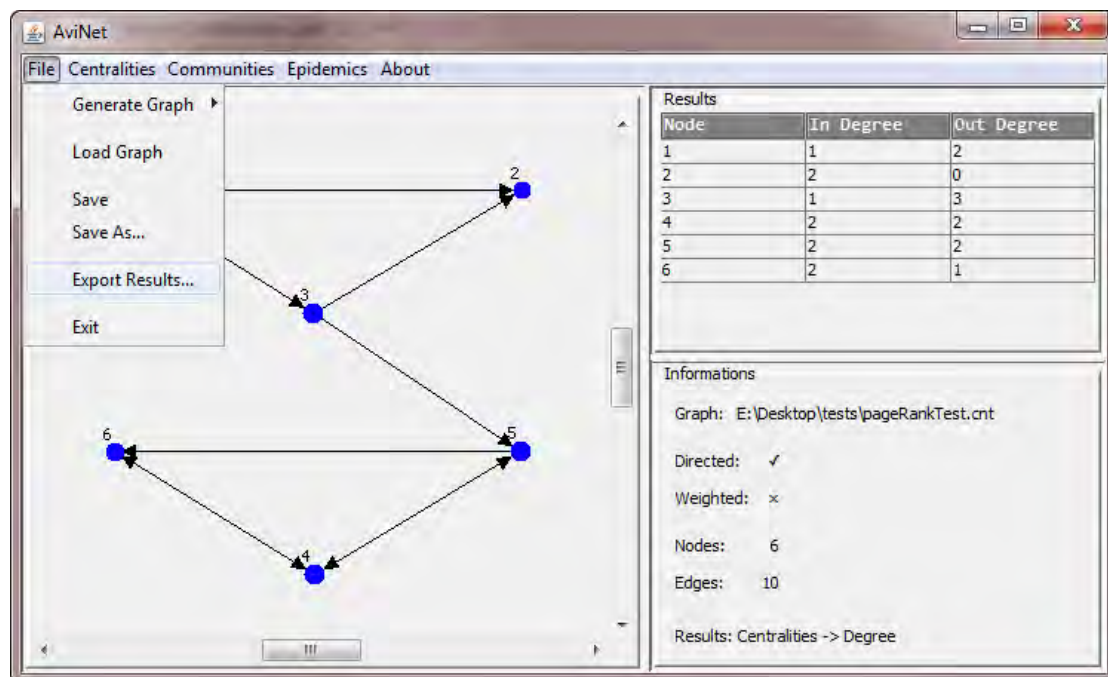


Εικόνα 86 - To Independent Cascade στο AviNet

8 Εξαγωγή αποτελεσμάτων από το AviNet

Η ανάλυση των σύνθετων δικτύων μέσω της θεωρίας γραφημάτων, βοηθά σε μεγάλο βαθμό στην κατανόηση της δομής και των ιδιοτήτων τους. Τα αποτελέσματα των αλγορίθμων που περιλαμβάνονται στις κεντρικότητες, τις κοινότητες και τις επιδημιολογίες, μπορούμε να πούμε ότι προσδίδουν στους κόμβους και τις ακμές ενός γραφήματος χαρακτηριστικά, μέσω των οποίων αποσαφηνίζονται οι ιδιότητες του γραφήματος, όπως επίσης και ο ρόλος που διαδραματίζει στην λειτουργία του δικτύου ο κάθε κόμβος ή ακμή ξεχωριστά. Επιπροσθέτως, η σύγκριση των αποτελεσμάτων διαφορετικών γραφημάτων, εξυπηρετεί τις ανάγκες διαχωρισμού και κατηγοριοποίησής τους, ενώ η παρατήρηση των αλλαγών που επιφέρουν σε ένα γράφημα μικρές διαφοροποιήσεις της τοπολογίας του, δύναται να οδηγήσει στην κατανόηση των αναγκών για την βελτιστοποίησή του.

Συνεπώς η δυνατότητα αποθήκευσης των αποτελεσμάτων που προσφέρει ένα λογισμικό ανάλυσης σύνθετων δικτύων, κρίνεται ιδιαίτερα σημαντική. Στο AviNet η λειτουργία αυτή παρέχεται μέσα από το μενού “File → Export Results...”, με μόνη βασική προϋπόθεση να έχει προηγηθεί η εκτέλεση ενός από τους αλγορίθμους που προσφέρονται, ώστε να είναι διαθέσιμα τα αποτελέσματά του. Διαφορετικά, αν το Results πάνελ είναι κενό, η επιλογή αυτή είναι απενεργοποιημένη.



The screenshot displays the AviNet application window. The main area shows a directed graph with 6 nodes (1-6) and 10 edges. Node 1 is at the top, node 2 is to its right, node 3 is below node 1, node 4 is at the bottom, node 5 is to the right of node 4, and node 6 is to the left of node 5. The edges are: 1→2, 1→3, 3→2, 3→5, 4→5, 4→6, 5→4, 5→6, 6→4, and 6→5.

The 'Results' panel on the right contains the following table:

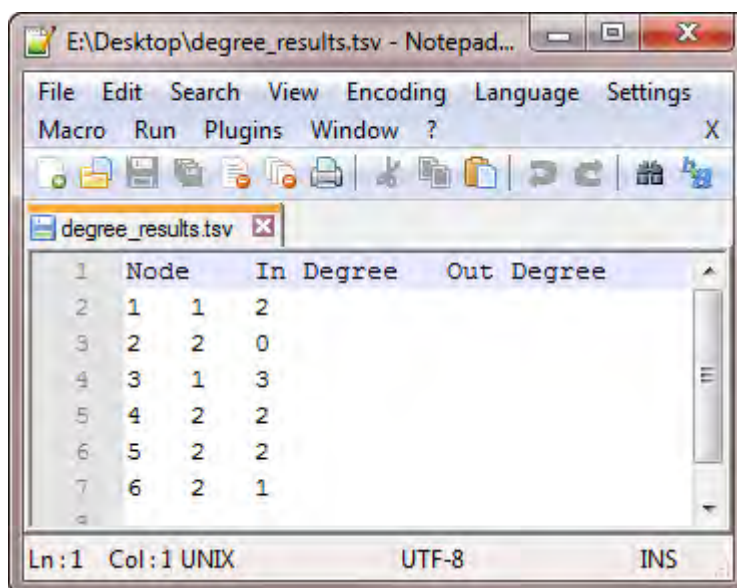
Node	In Degree	Out Degree
1	1	2
2	2	0
3	1	3
4	2	2
5	2	2
6	2	1

The 'Informations' panel below the table shows:

- Graph: E:\Desktop\tests\pageRankTest.cnt
- Directed:
- Weighted:
- Nodes: 6
- Edges: 10
- Results: Centralities -> Degree

Εικόνα 87 - Εξαγωγή αποτελεσμάτων στο AviNet

Με κλικ στο μενού “Export Results”, ανοίγει το κατάλληλο παράθυρο διαλόγου προκειμένου ο χρήστης να αποθηκεύσει τα αποτελέσματα του κάθε αλγορίθμου στην τοποθεσία που επιθυμεί. Το AniNet αποθηκεύει τα αποτελέσματα σε αρχεία tsv (tab separated values), τα οποία είναι συμβατά με το Microsoft Excel. Για το γράφημα της εικόνας 87, και τα αποτελέσματα του Degree Centrality, δημιουργήθηκε το αρχείο της εικόνας 88.



1	Node	In Degree	Out Degree
2	1	1	2
3	2	2	0
4	3	1	3
5	4	2	2
6	5	2	2
7	6	2	1

Εικόνα 88 - Παράδειγμα αρχείου αποτελεσμάτων

Με τον τρόπο αυτό ο χρήστης μπορεί είτε να έχει πρόσβαση στα αποτελέσματα ανοίγοντας τα εξαγόμενα αρχεία με οποιοδήποτε κειμενογράφο, είτε να τα φορτώσει στο Microsoft Excel. Μέσα από την πληθώρα εργαλείων που προσφέρονται από το Excel ο χρήστης μπορεί να παρατηρήσει καλύτερα τις ιδιότητες ενός γραφήματος, υπολογίζοντας συνολικά αθροίσματα ή μετατοπίσεις. Ακόμα έχει την δυνατότητα να παράγει πληθώρα γραφικών παραστάσεων, όπως για παράδειγμα αυτό της κατανομής των τιμών.

9 Σύνοψη – Μελλοντικές επεκτάσεις

9.1 Σύνοψη

Τα σύνθετα δίκτυα μέσα από την θεωρία γραφημάτων δύναται να αναπαραστήσουν και να αναλύσουν την δομή και τις ιδιότητες πολλών πολύπλοκων συστημάτων. Η υπολογιστική και χρονική πολυπλοκότητα των αλγορίθμων που χρησιμοποιούνται στην ανάλυση των σύνθετων δικτύων, επιτάσσουν την χρήση των κατάλληλων λογισμικών. Στις μέρες μας υπάρχει πληθώρα προγραμμάτων για την ανάλυση των σύνθετων δικτύων, το καθένα από τα οποία έχει τα πλεονεκτήματα και τα μειονεκτήματά του. Έτσι κρίθηκε αναγκαία η ανάπτυξη ενός λογισμικού, το οποίο θα φέρει τα ακόλουθα χαρακτηριστικά, βοηθώντας το στην διαφοροποίησή του από τα ήδη υπάρχοντα.

Το AniNet είναι ένα λογισμικό για την δημιουργία, την οπτικοποίηση και την ανάλυση των σύνθετων δικτύων, μέσα από τις βασικές τους ιδιότητες. Το σύνολο των λειτουργιών του AniNet παρέχεται μέσα από ένα εύχρηστο και φιλικό γραφικό περιβάλλον. Προσφέρει πληθώρα επιλογών για την δημιουργία γραφημάτων, είτε μέσα από την κατασκευή τους βήμα προς βήμα μέσω των περιλαμβανομένων εργαλείων, είτε μέσα από την ανάγνωση αρχείων. Επίσης παρέχει την δυνατότητα δημιουργίας μερικών από τους πλέον διαδεδομένους τύπους τυχαίων γραφημάτων, τα οποία ο χρήστης μπορεί είτε να εξετάσει αυτούσια, είτε να προχωρήσει στην παραμετροποίησή τους. Είναι το πρώτο λογισμικό το οποίο περιλαμβάνει ταυτόχρονα αλγορίθμους εύρεσης των σημαντικότερων κεντρικότητων, εντοπισμού κοινοτήτων και εξάπλωσης επιδημιών. Ακόμα παρέχει την δυνατότητα εξαγωγής των γραφημάτων που δημιουργήθηκαν, όπως επίσης και των αποτελεσμάτων τους, με σκοπό την περαιτέρω ανάλυση και την μελλοντική χρήση.

9.2 Μελλοντικές επεκτάσεις

Στις επόμενες εκδόσεις του λογισμικού που αναπτύχθηκε, μπορούν να γίνουν βελτιώσεις σχετικές με την λειτουργικότητά του και το εύρος των αλγορίθμων που υποστηρίζονται. Η δημιουργία τυχαίων γραφημάτων μπορεί να εμπλουτιστεί με περισσότερες επιλογές, όπως για παράδειγμα την δημιουργία ενός Scale Free γραφήματος. Ακόμα η αναπαράσταση των αποτελεσμάτων δύναται να βελτιωθεί με

την προσθήκη λειτουργίας για την παρακολούθηση των μετατοπίσεων των τιμών, μετά από αλλαγές στην τοπολογία του γραφήματος. Το ιστορικό αυτών των αλλαγών θα θέλαμε επίσης να παρέχεται στον χρήστη μέσα από το ανάλογο παράθυρο. Επίσης μπορούν να προστεθούν περισσότεροι αλγόριθμοι εύρεσης κεντρικότητας, όπως ο Bonachich Power Centrality. Στον εντοπισμό κοινοτήτων θα θέλαμε να προστεθεί ο αλγόριθμος του CPM (Clique Percolation Method), ενώ ο τομέας της εξάπλωσης επιδημιών θα μπορούσε να εμπλουτιστεί με την προσθήκη μοντέλων όπως τα SIRS και SEIS.

Βιβλιογραφία

- [1] Dimitrios Katsaros, George Pallis, Konstantinos Stamos, Athena Vakali, Antonis Sidiropoulos and Yannis Manolopoulos, CDNs Content OutSourcing via Generalized Communities.
- [2] W. Chen, Y. Yuan and L. Zhang, Scalable Influence Maximization in Social Networks under the Linear Threshold Model.
- [3] Marius Eidsaa and Eivind Almaas, s-core network decomposition: A generalization of k-core analysis to weighted networks.
- [4] Steven H. Storatz, Exploring complex networks.
- [5] David Combe, Christine Largeron, Előd Egyed-Zsigmond and Mathias Géry, A comparative study of social network analysis tools, 2010.
- [6] Remco van der Hofstad, Random Graphs and Complex Networks. Vol 1. 2015.
- [7] Παναγιώτης Δ. Μποζάνης, Αλγόριθμοι.
- [8] David Kempe, Jon Kleinberg and Eva Tardos, Maximizing the Spread of Influence through a Social Network.
- [9] J. Omic and P. Van Mieghem, Epidemics Spreading in Networks – Variance of the number of infected Nodes.
- [10] U Kang, Spiros Papadimitriou, Jimeng Sun Hanghang Tong, Centralities in Large Networks: Algorithms and Observations.
- [11] Tauhid R. Zaman, Information Extraction with Network Centralities: Finding Rumor Sources, Measuring Influence and Learning Community Structure.
- [12] C. Moore and M. E. J. Newman. Epidemics and percolation in in small-world networks. Phys. Rev. E, 61:5678–5682, 2000.
- [13] M. E. J. Newman, the Structure and Function of Complex Networks.

- [14] Reka Albert and Albert-Laszlo Barabasi, Statistical mechanics of complex networks.
- [15] Maarte van Steen, An introduction to Graph Theory and Complex Networks, January 2010.
- [16] Petter Holme and Beom Jun Kim, Attack vulnerability of Complex Networks