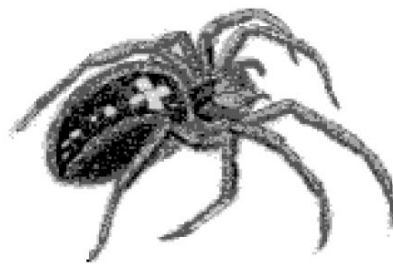


Rajek [πάγιεκ]

Εφαρμογή ανάλυσης μεγάλων δικτύων



Version 4.03
(64bit)

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Πρόγραμμα Μεταπτυχιακών Σπουδών στην Πληροφορική
Επικοινωνιακά Συστήματα και Τεχνολογίες
Προηγμένη Επεξεργασία Σήματος

Δημήτριος Ρούσης (ΑΜ: 532)



Περιεχόμενα

1. Η εφαρμογή Pajek.....	4
2. Τύποι αντικείμενων	6
3. Τα εργαλεία της εφαρμογής Pajek	7
3.1 Το κεντρικό παράθυρο εργασίας	8
3.1.1 Η περιοχή δεδομένων	8
3.1.2 Η περιοχή του κυρίως μενού.....	10
4. Δημιουργία αρχείων δικτύου για το Pajek	18
4.1 Μέσω του Pajek	18
4.2 Μέσω του Matlab	24
4.3 Μέσω της γλώσσας προγραμματισμού Python	27
4.4 Με επεξεργαστή κειμένου.....	28
4.5 Δημιουργώντας σχεσιακή βάση δεδομένων.....	32
4.6 Μέσω άλλων τύπων αρχείων.....	36
5. Καθορισμός διάταξης δικτύου	36
5.1 Αυτόματες ρουτίνες σχεδίασης	36
5.2 Χειροκίνητη σχεδίαση.....	37
6. Αποθήκευση δικτύου	38
7. Δημιουργώντας partitions, vectors.....	38
7.1 Δημιουργία Partition	38
7.2 Δημιουργία Vector	39
8. Το αρχείο .paj	40
9. Μορφές αρχείων εξόδου του Pajek.....	40
10. Τοπικές και ευρείες προβολές δικτύου.....	41
11. Εξαγωγή υποδικτύων.....	42
12. Μείωση μεγέθους δικτύου.....	43
12.1 Μείωση – Συρρίκνωση δικτύου	43
12.2 Αφαίρεση των γραμμών με χαμηλή τιμή	43
13. Εύρεση μονοπατιών	44

13.1	Εύρεση συντομότερου μονοπατιού	44
13.2	Εύρεση διαμέτρου	45
13.3	K-neighbors.....	45
14.	Ισχυρά και Χαλαρά συνδεδεμένα στοιχεία – κορυφές.....	46
15.	Αμφισυνδεδεμένα στοιχεία – κόμβοι.....	47
16.	Άκυκλα δίκτυα.....	48
17.	Κεντρικότητα.....	48
17.1	Βαθμός κεντρικότητας.....	49
17.2	Πυκνότητα κεντρικότητας.....	49
17.3	Ενδιαμεσότητα κεντρικότητας	49
18.	Τριαδική απογραφή	49
19.	Πυρήνες.....	50
20.	Κλίκες.....	51
21.	Αναζήτηση προτύπων	52
22.	Μέθοδοι εντοπισμού δομών κοινοτήτων	53
23.	Δίκτυα 2-mode	53
23.1	Μετασχηματισμός 2-mode σε 1-mode	54
24.	Απεικόνιση δικτύου με βάρη	55
25.	Νησίδες	55
26.	Ιεραρχικό clustering	56
27.	Μοντελοποίηση σε μπλοκ	57
28.	Χρονικώς μεταβαλλόμενα δίκτυα	58
29.	Συνδεσιμότητα μικρών κύκλων.....	60
30.	Εξαγωγή δεδομένων	61
30.1	Εξαγωγή στην R.....	61
30.2	Εξαγωγή σε SPSS	62
30.3	Εξαγωγή σε αρχείο χωρισμένο με tabs.....	62
31.	Σενάριο χρήσης με πραγματικά δεδομένα.....	62
32.	Παραρτήματα	73
32.1	Τα περιεχόμενα του αρχείου <code>mynetwork.net</code>	73

32.2	Οι κώδικες των αρχείων Matlab	74
32.2.1	Ο κώδικας του αρχείου <code>get_random_graph.m</code>	74
32.2.2	Ο κώδικας του αρχείου <code>write_matrix_to_pajek.m</code>	75
32.3	Εύρεση του συντομότερου μονοπατιού με τη βοήθεια βιολογικών οργανισμών που δε διαθέτουν εγκέφαλο	78
33.	Βιβλιογραφία	80

1. Η εφαρμογή Pajek

Το Pajek¹ είναι μια δωρεάν εφαρμογή για Windows η οποία αναπτύχθηκε από τους καθηγητές A. Mrvar και V. Batagelj². Χρησιμοποιείται για την ανάλυση και οπτικοποίηση δικτύων (στο εξής το δίκτυο θα αναφέρεται και ως δικτύωμα) που αποτελούνται από χιλιάδες ή και εκατομμύρια κορυφές (ή κόμβους, ή στοιχεία). Είναι γραμμένη σε Delphi και η ανάπτυξή της ξεκίνησε το 1996. Η λέξη «Pajek» είναι σλοβενική και σημαίνει «αράχνη».

Μερικά από τα δίκτυα που μπορούν να αναλυθούν και να οπτικοποιηθούν με το Pajek είναι:

- Δίκτυα οργανικών μορίων στην οργανική χημεία
- Αλληλεπιδράσεις δικτύων πρωτεϊνών – υποδοχέων
- Κοινωνικά δίκτυα
- Γενεαλογικά δέντρα
- Δίκτυα Internet
- Δίκτυα βιβλιογραφικών παραπομπών
- Δίκτυα διάχυσης (ιών, ειδήσεων κ.α.)
- Δίκτυα με κόμβους δύο ειδών (εξόρυξη δεδομένων από αυτά)
- Και πολλά άλλα³

Ο σχεδιασμός του Pajek βασίστηκε σε προηγούμενη εμπειρία ανάπτυξης λογισμικού των δημιουργών του⁴. Βασίστηκε σε βιβλιοθήκες αλγορίθμων που αφορούν γράφους⁵ (Graph και X-Graph), σε λογισμικό ανάλυσης και οπτικοποίησης (STRAN, RelCalc, Draw, Energ), και στην γλώσσα σήμανσης περιγραφής γράφων NetML η οποία βασίζεται στη γλώσσα σήμανσης SGML (Standard Generalized Markup Language).

¹ <http://vlado.fmf.uni-lj.si/pub/networks/pajek>

² Πανεπιστήμιο Ljubljana, Σλοβενία (www.uni-lj.si)

³ <http://vlado.fmf.uni-lj.si/pub/networks/data>

⁴ <http://vlado.fmf.uni-lj.si/pub/networks>

⁵ http://en.wikipedia.org/wiki/Graph_theory

Μια ειδική έκδοση του Pajek, είναι το Pajek-XXL⁶. Έχει σημαντικά λιγότερες απαιτήσεις σε μνήμη (2 έως 3 φορές) και είναι ταχύτερο σε περιπτώσεις διεργασιών με συχνή χρήση μνήμης (π.χ. παραγωγή τυχαίων δικτύων, εξαγωγή δεδομένων δικτύων, συρρίκνωση δικτύων κ.α.).

Η ανάπτυξη του Pajek έχει γίνει με βάση τους εξής στόχους:

- Να υποστηρίζει απλοποίηση μέσω της (αναδρομικής) αποσύνθεσης ενός μεγάλου δικτύου σε πολλά μικρότερα δίκτυα τα οποία να μπορούν στη συνέχεια να υποστούν περαιτέρω επεξεργασία χρησιμοποιώντας πιο εξελιγμένες μεθόδους.
- Να παρέχει στο χρήστη ισχυρά εργαλεία οπτικοποίησης – παρουσίασης των δεδομένων του.
- Να εφαρμόζει ένα σύνολο αποτελεσματικών αλγορίθμων για την ανάλυση μεγάλων δικτύων των οποίων ο χρόνος εκτέλεσης θα παρουσιάζει υποτετραγωνική πολυπλοκότητα⁷.

Με το Pajek είναι δυνατή:

- Η εύρεση συστάδων (υποσύνολα, γειτονικές κορυφές που παρουσιάζουν κάποιο νόημα, πυρήνες, κλπ.) σε ένα δίκτυο.
- Η εξαγωγή κορυφών που ανήκουν σε ίδιες συστάδες και η παρουσίασή τους ξεχωριστά (λεπτομερής τοπική θέαση).
- Η συμπύκνωση κορυφών σε συστάδες.
- Η παρουσίαση των σχέσεων που υπάρχουν μεταξύ συστάδων (ευρεία σφαιρική θέαση).

Τέλος, εκτός από τα κανονικά δίκτυα (κατευθυνόμενα, μη-κατευθυνόμενα, μικτά), το Pajek υποστηρίζει επίσης:

- Πολυσχεσιακά δίκτυα
- Δίκτυα 2-mode (διμερείς γράφοι – δίκτυα μεταξύ δύο αποκομμένων συνόλων από κορυφές)
- Προσωρινά δίκτυα (δυναμικοί γράφοι – δίκτυα που μεταβάλλονται με τον χρόνο)

⁶ <http://mrvar.fdv.uni-lj.si/pajek>

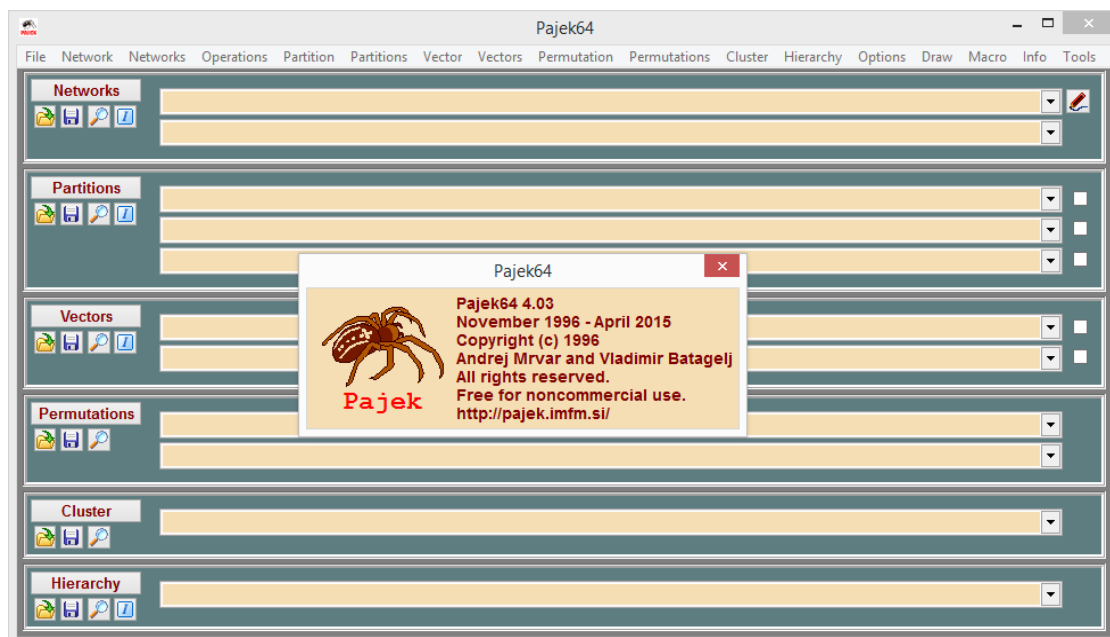
⁷ http://en.wikipedia.org/wiki/Time_complexity#Sub-quadratic_time

2. Τύποι αντικείμενων

Στο Pajek χρησιμοποιούνται έξι τύποι αντικειμένων:

- i. Δίκτυα (Networks). Είναι τα κύρια αντικείμενα (κορυφές και γραμμές). Η προ-επιλεγμένη επέκτασή τους είναι `.net`
- ii. Κατατμήσεις (Partitions) – Δείχνουν σε ποια κλάση κορυφών ανήκει η κάθε κορυφή. Η επέκτασή τους είναι `.clu`
- iii. Διανύσματα (Vectors) – Περιλαμβάνουν τιμές για κάθε κορυφή. Επέκταση `.vec`
- iv. Μεταθέσεις (Permutations) – Αναδιατάξεις των κορυφών. Επέκταση `.per`
- v. Συστάδες (Clusters) – Υποσύνολα κορυφών. Επέκταση `.cls`
- vi. Ιεραρχίες (Hierarchies) – Ιεραρχικά ταξινομημένες κορυφές. Επέκταση `.hie`

Στο κύριο παράθυρο του γραφικού περιβάλλοντος του Pajek περιλαμβάνονται ως πεδία εισόδου, και οι έξι αυτοί τύποι αντικειμένων που αναφέρθηκαν:



Τη στιγμή που γράφονται αυτές οι γραμμές, η τελευταία έκδοση του Pajek είναι η 4.03 (2 Απριλίου 2015) και διατίθεται σε 32 και 64bit εκδόσεις.

Μετά την εγκατάσταση του προγράμματος δημιουργείται το παρακάτω σύστημα αρχείων:

Name	Date modified	Type	Size
 data	4/16/2015 2:57 AM	File folder	
 macro	4/16/2015 2:57 AM	File folder	
 libiomp5md.dll	4/23/2014 8:34 AM	Application extension	1,051 KB
 MtxVec.Lapack4d.dll	4/11/2014 1:27 PM	Application extension	55,060 KB
 MtxVec.Spld4.dll	5/13/2014 4:04 PM	Application extension	8,991 KB
 MtxVec.Vml4d.dll	5/13/2014 4:04 PM	Application extension	8,238 KB
 Pajek.exe	4/1/2015 1:14 PM	Application	8,831 KB
 pajek.ini	4/16/2015 4:06 AM	Configuration settings	5 KB
 Pajek	4/16/2015 2:57 AM	Shortcut	1 KB
 PajekXXL.exe	4/1/2015 1:14 PM	Application	7,119 KB
 PajekXXL	4/16/2015 2:57 AM	Shortcut	1 KB
 unins000.dat	4/16/2015 2:57 AM	Probe Document	6 KB
 unins000.exe	4/16/2015 2:00 AM	Application	704 KB

Διαδρομή εγκατάστασης: C : \Users\<UserName>\AppData\Local\Pajek

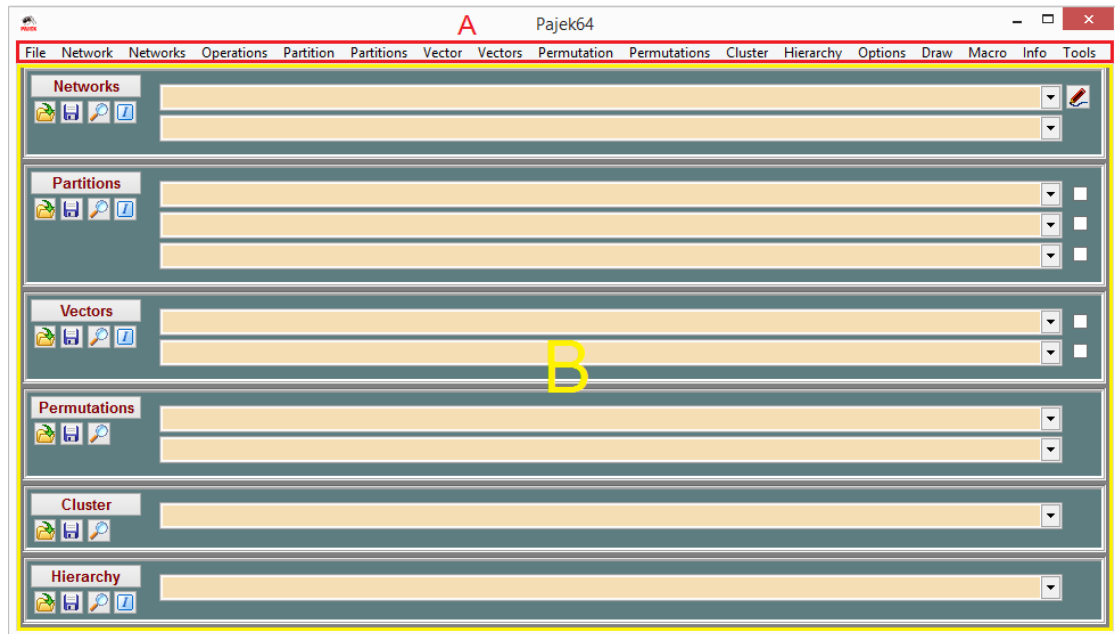
3. Τα εργαλεία της εφαρμογής Pajek

Το γραφικό περιβάλλον του Pajek δεν είναι ιδιαίτερα φιλικό προς τον χρήστη. Διαφέρει πολύ από τα περιβάλλοντα που έχουμε συνηθίσει στις περισσότερες εφαρμογές, όπου ο τρόπος λειτουργίας τους είναι προφανής σχεδόν από την πρώτη ματιά.

Παρ' όλα αυτά, παρατηρώντας το υπό την κατάλληλη οπτική γωνία, και έχοντας το απαραίτητο θεωρητικό υπόβαθρο, γρήγορα καταλαβαίνει κανείς τις αξιόλογες δυνατότητές του. Το Pajek λειτουργεί ουσιαστικά σαν ένα επιστημονικό κομπιουτεράκι που είναι εξειδικευμένο για πράξεις επάνω σε δίκτυα.

3.1 Το κεντρικό παράθυρο εργασίας

Το κεντρικό παράθυρο της εφαρμογής αποτελείται από δύο περιοχές. Την περιοχή του κυρίως μενού (Κόκκινη – Α) και την περιοχή των δεδομένων (Κίτρινη Β):



Πιο κάτω γίνεται μια πολύ σύντομη αναφορά στη λειτουργικότητα του κάθε στοιχείου που απαρτίζει το γραφικό περιβάλλον του Pajek. Μια λεπτομερής ανάλυση γίνεται στο εγχειρίδιο χρήσης του προγράμματος [1] το οποίο έχουν συντάξει οι ίδιοι οι δημιουργοί του.

3.1.1 Η περιοχή δεδομένων

Η περιοχή Β αποτελείται με τη σειρά της από έξι υποπεριοχές (όσες και οι τύποι αντικειμένων που υποστηρίζει το Pajek) οι οποίες χαρακτηρίζονται από την ονομασία των έξι μεγάλων κουμπιών της:

- Networks – Εδώ εισάγεται το αρχείο δικτύου (`filename.net`) που μελετάται.
- Partitions – Το αρχείο (`filename.clu`) με τα δεδομένα ταξινόμησης των κορυφών ενός δικτυώματος. Έχει τέτοια δομή ώστε κάθε κορυφή να αντιστοιχείται σε μία και μόνο κλάση.

- **Vectors** – Το αρχείο (`filename.vec`) με τα δεδομένα στα οποία αποδίδονται αριθμητικές τιμές σε κάθε κορυφή του δικτύου. Πρόκειται για μια λίστα αριθμών όπου κάθε γραμμή (εγγραφή) αντιστοιχεί σε μία τιμή που συνδέεται με ένα κόμβο του δικτυώματος.
- **Permutations** – Αρχείο (`filename.per`) στο οποίο μας επιτρέπεται η αναδιοργάνωση των κόμβων σε διαφορετική σειρά (ακολουθία).
- **Cluster** – Αρχείο (`filename.cls`) ομαδοποίησης κόμβων που έχουν κάποιο ιδιαίτερο ενδιαφέρον για εμάς.
- **Hierarchy** – Αρχείο (`filename.hie`) ταξινόμησης κορυφών για περιπτώσεις που μια κορυφή (ή ένα group κορυφών) ανήκει ταυτόχρονα σε περισσότερες από μία κλάσεις.

Η διαφορά ενός αρχείου με δεδομένα τύπου αντικειμένου “Partition”, με ένα αρχείο τύπου “Hierarchy”, είναι ότι στο πρώτο μια κορυφή (ή ένα group κορυφών) δεν μπορεί να ανήκει ταυτόχρονα σε περισσότερες από μία κλάσεις.

Οι πρώτες τρεις υποπεριοχές χρησιμοποιούνται σε όλα σχεδόν τα δίκτυα που αναλύονται, ενώ οι υπόλοιπες τρεις, μόνο σε περίπλοκα δίκτυα στα οποία υπάρχουν ανάγκες αναδιοργάνωσης, ομαδοποίησης και ιεράρχησης των κόμβων.

Σε καθεμιά από τις έξι υποπεριοχές περιλαμβάνονται γραφικά στοιχεία ελέγχου για την επίτευξη της απαιτούμενης λειτουργικότητας κατά τον χειρισμό των δεδομένων:



Το μεγάλο κουμπί επάνω αριστερά χρησιμεύει για την προσθήκη επιπλέον πεδίων εισαγωγής δικτύων.

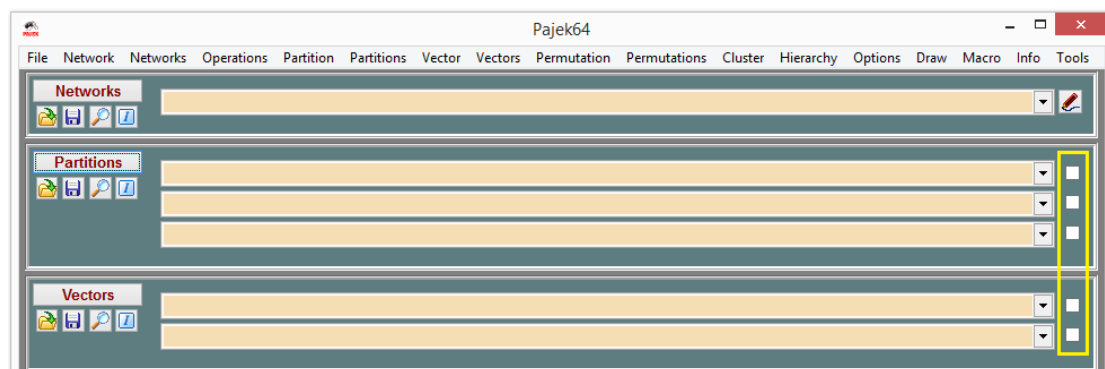
Τα τέσσερα μικρότερα κουμπιά με τα εικονίδια φακέλου, δισκέττας, μεγεθυντικού φακού και του λατινικού “i”, χρησιμεύουν για τις ενέργειες ανοίγματος αρχείου, αποθήκευσης αρχείου, εμφάνισης και επεξεργασίας ιδιοτήτων αντικειμένου – αρχείου, και πληροφοριών (info), αντίστοιχα.

Τα πτυσσόμενα πεδία κειμένου (drop – down lists) περιλαμβάνουν τα ονόματα των αρχείων που μπορεί να υπάρχουν στις έξι υποπεριοχές. Αν για παράδειγμα εναποθέ-

σουμε κάνοντας drag η' drop ένα αρχείο δικτύου .net, τότε θα εμφανιστεί στο αντίστοιχο πεδίο, το όνομά του και η διαδρομή του στο σκληρό δίσκο (και ορισμένες επιπλέον πληροφορίες που θα δούμε στη συνέχεια).

Το κουμπί με το εικονίδιο μολυβιού που υπάρχει μόνο δίπλα από την περιοχή Networks, είναι μια συντόμευση για την εντολή Draw > Network ή οποία απεικονίζει στον χρήστη το τρέχον δίκτυο.

Τα κουτιά επιλογής (check boxes) χρησιμεύουν για την μορφοποίηση της εμφάνισης των αντικειμένων κατά την απεικόνισή τους, όταν υπάρχουν πολλαπλά όμοια αντικείμενα (πχ: περισσότερα του ενός partitions ή/και vectors):



Διατίθενται μόνο για την υποπεριοχή των Partitions και των Vectors μιας και μόνο εκεί υπάρχει νόημα διαφοροποίησης της εμφάνισής τους κατά την απεικόνιση. Για παράδειγμα, κατά την απεικόνιση ενός δικτύου χωρών, θα ήταν πρακτικό να εμφανίζονται διαφορετικά αυτές που ανήκουν σε διαφορετική ήπειρο από τις υπόλοιπες.

3.1.2 Η περιοχή του κυρίως μενού

Το κεντρικό μενού σχετίζεται άμεσα με τις έξι υποπεριοχές (και δη, τύπους αντικειμένων) της περιοχής δεδομένων.

Τα παρακάτω μενού λειτουργούν ως βοηθητικά εργαλεία για τη διαχείριση αρχείων, την αποθήκευση ρυθμίσεων, την γραφική απεικόνιση δεδομένων κ.α. τα οποία θα αναφερθούν στις επόμενες παραγράφους:

- File
- Options

- Draw
- Macro
- Info
- Tools

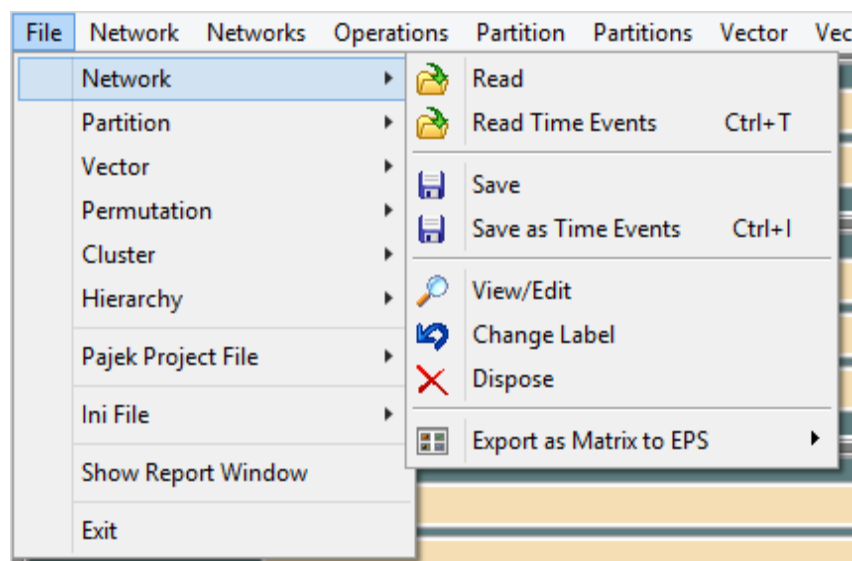
Τα μενού που περιλαμβάνονται στη λίστα που ακολουθεί σχετίζονται με πράξεις και υπολογισμούς που αφορούν τους έξι τύπους αντικειμένων του Pajek:

- Network και Networks
- Operations
- Partition και Partitions
- Vector και Vectors
- Permutation και Permutations
- Cluster
- Hierarchy

Όσα από τα παραπάνω είναι στον πληθυντικό, απαιτούν την είσοδο δύο (η περισσότερων σε ορισμένες περιπτώσεις) αντικειμένων του **ίδιου τύπου** για να πραγματοποιηθούν. Εξάιρεση αποτελεί το μενού “Operations” όπου απαιτούνται δύο είδη αντικειμένων **διαφορετικού τύπου**.

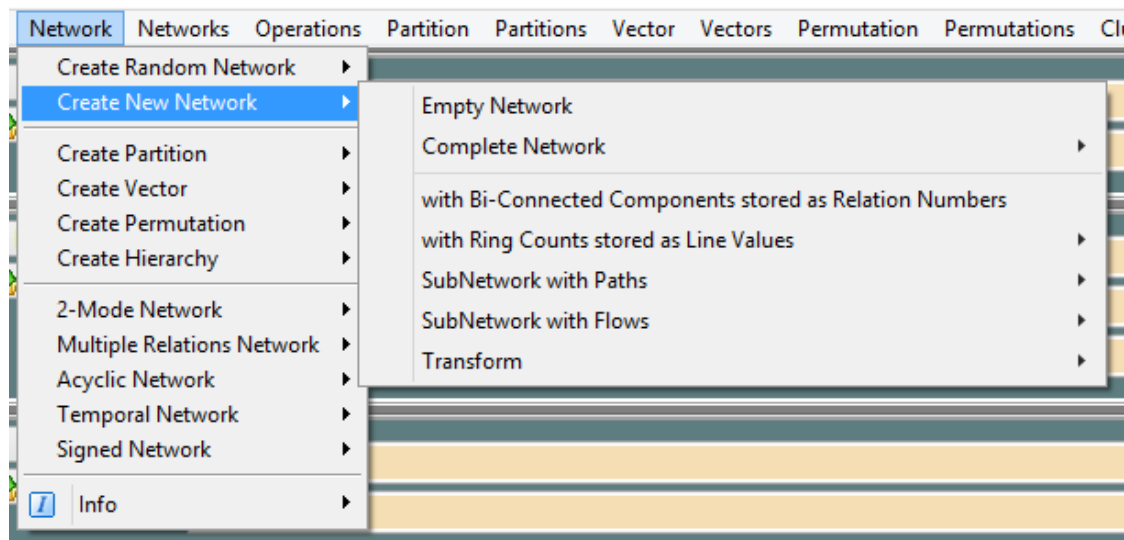
3.1.2.1 Μενού File

Βασικές λειτουργίες εισόδου – εξόδου με τους έξι τύπους αντικειμένων:

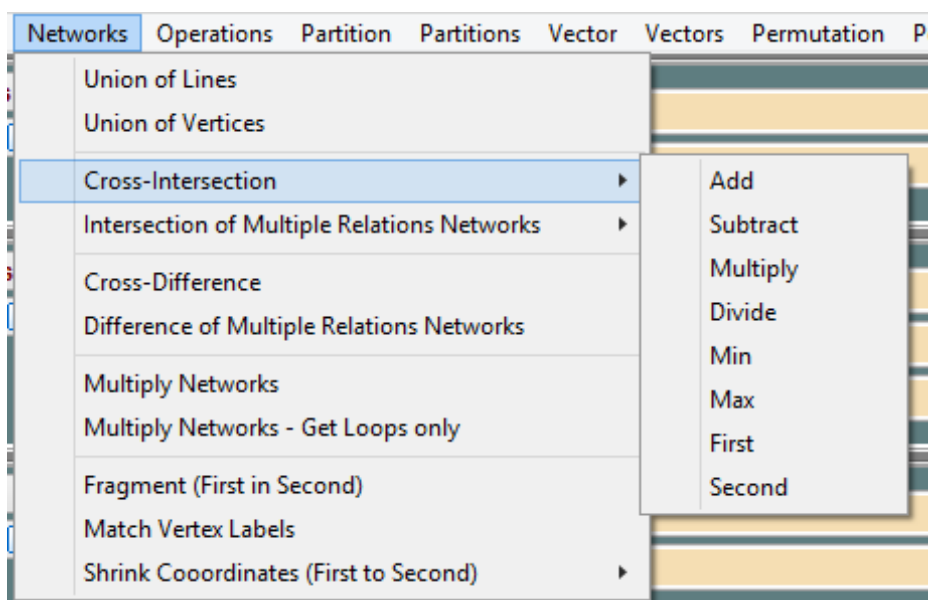


3.1.2.2 Μενού Network και Networks

Το μενού “Network” αφορά λειτουργίες για τις οποίες μόνο ένα δίκτυο (ή τίποτα στην περίπτωση των τυχαίων δικτύων) είναι απαραίτητο ως είσοδο:

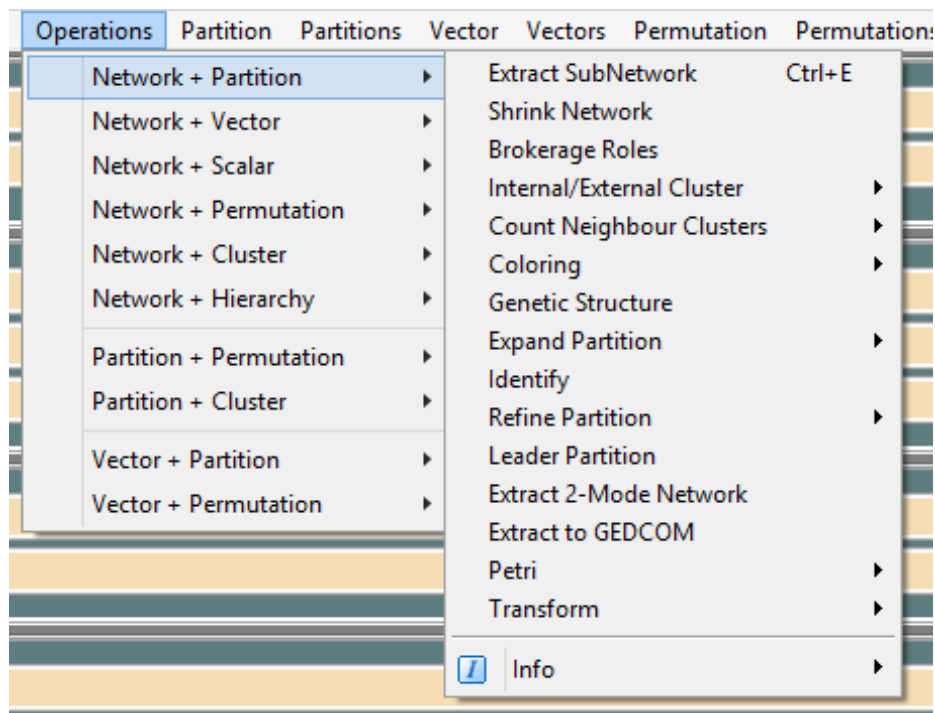


Το μενού “Networks” αφορά λειτουργίες (πράξεις) που εφαρμόζονται σε δύο δίκτυα:



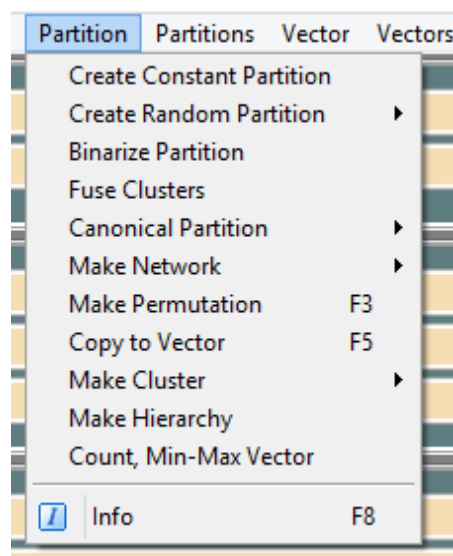
3.1.2.3 Μενού Operations

Ως είσοδος απαιτείται ένα δίκτυο και ένας ακόμη τύπος αντικειμένου:

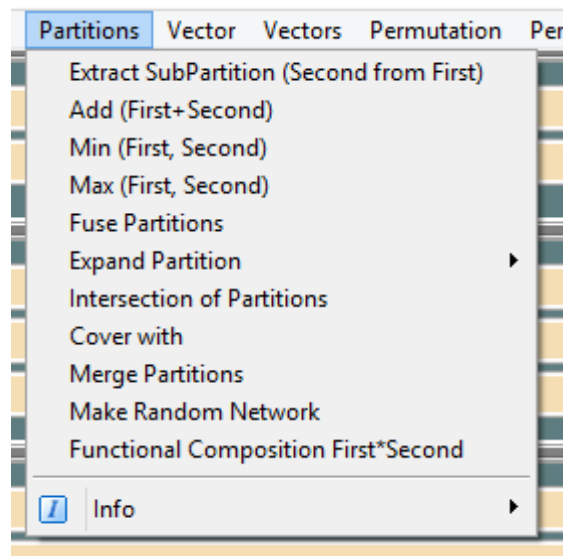


3.1.2.4 Μενού Partition και Partitions

Το μενού “Partition” περιλαμβάνει λειτουργίες για μία μόνο κατάτμηση του δικτύου:

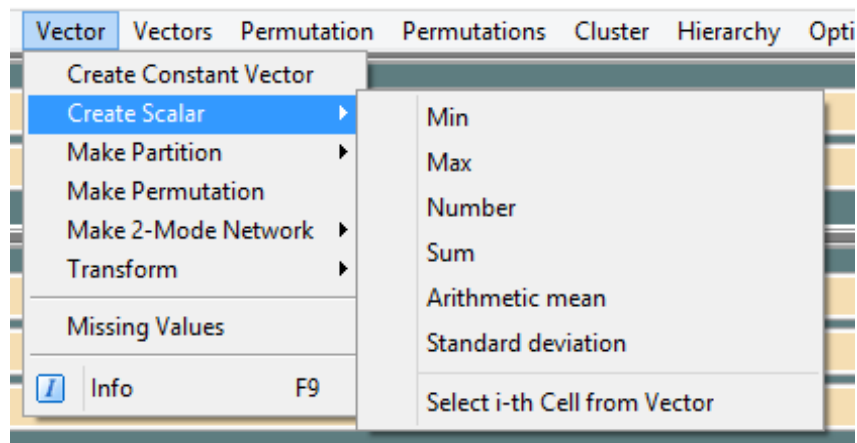


Το μενού “Partitions” αφορά λειτουργίες που απαιτούν δύο κατατμήσεις:

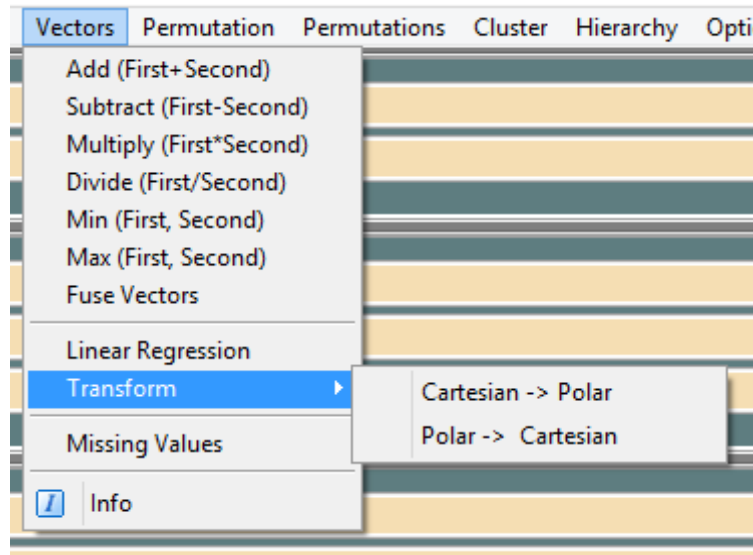


3.1.2.5 Μενού Vector και Vectors

Το μενού “Vector” περιλαμβάνει λειτουργίες για ένα μόνο διάνυσμα του δικτύου:

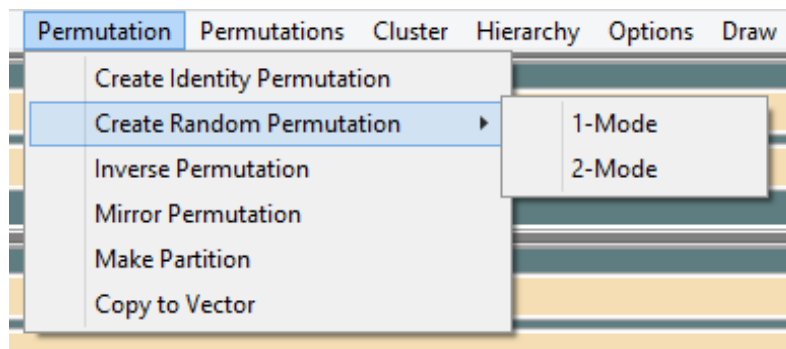


Το μενού “Vectors” αφορά λειτουργίες για δύο διανύσματα. Τα διανύσματα αυτά θα πρέπει, είτε να είναι ίδιας διάστασης, είτε το ένα από τα δύο να βαθμωτό (scalar):

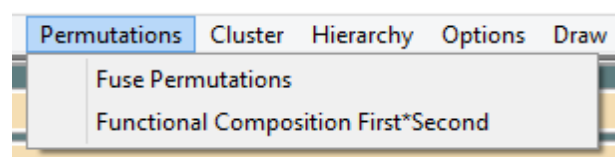


3.1.2.6 Μενού *Permutation* και *Permutations*

Το μενού “Permutation” περιλαμβάνει λειτουργίες για ένα μόνο σύνολο μεταθέσεων του δικτύου:

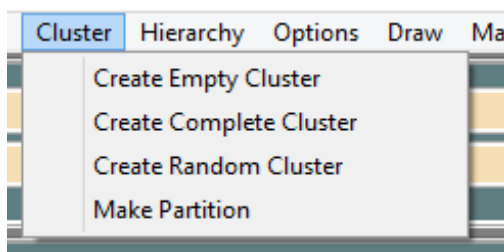


Το μενού “Permutations” αφορά λειτουργίες για δύο σύνολα μεταθέσεων:



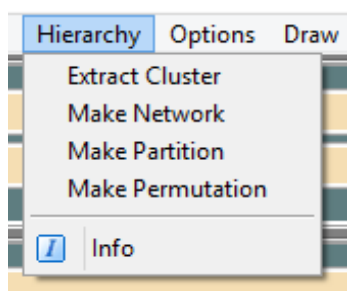
3.1.2.7 Μενού Cluster

Δημιουργεί ή μετατρέπει μία συστάδα, σε κατάτμηση (make partition):



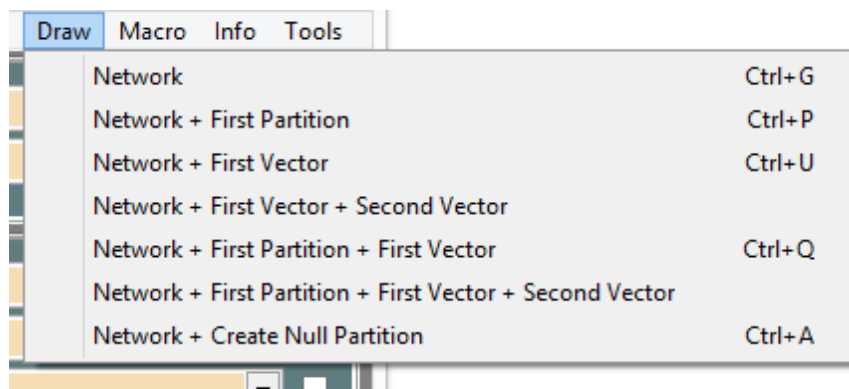
3.1.2.8 Μενού Hierarchy

Λαμβάνει ως είσοδο μια ιεραρχία και, είτε εξάγει από αυτή μια συστάδα, είτε την μετατρέπει σε δίκτυο, ή σε κατάτμηση, ή σε ένα σύνολο μεταθέσεων:



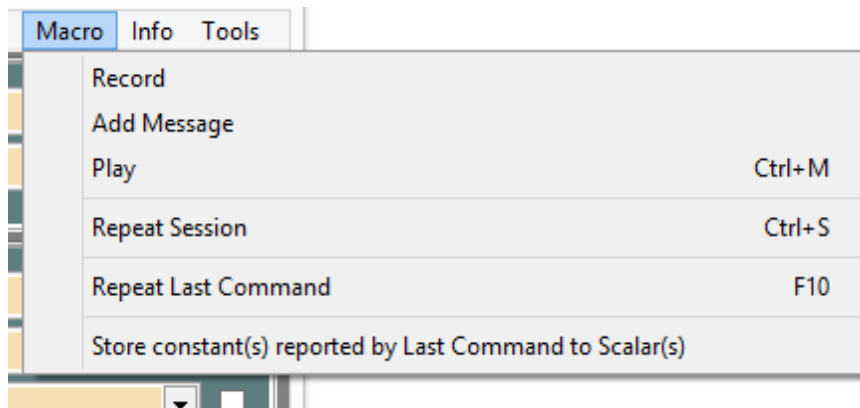
3.1.2.9 Μενού Draw

Σε αυτό το μενού περιλαμβάνονται όλες οι δυνατότητες απεικόνισης δεδομένων που υποστηρίζει το Pajek:



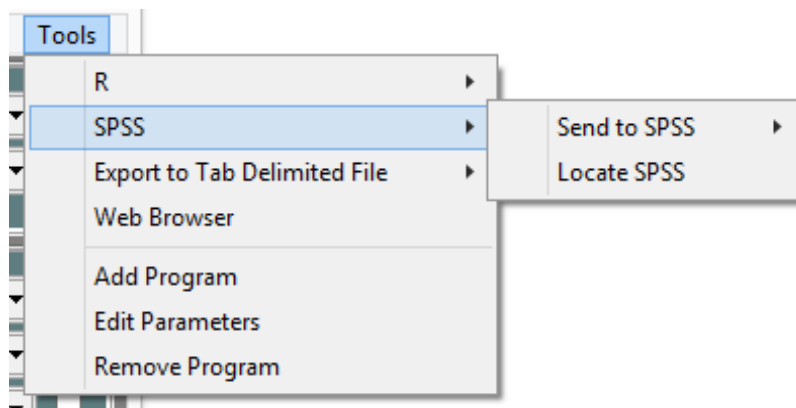
3.1.2.10 Μενού Macro

Μας δίνει τη δυνατότητα να δημιουργήσουμε μακροεντολές⁸:



3.1.2.11 Μενού Tools

Μας δίνει τη δυνατότητα εξαγωγής των δεδομένων μας σε διάφορες μορφές προκειμένου να είναι δυνατή η εισαγωγή τους σε λογισμικό τρίτων και έτσι να μπορεί να επιτευχθεί η περεταίρω επεξεργασία τους (πχ: στατιστική ανάλυση μέσω SPSS⁹):



⁸ [https://en.wikipedia.org/wiki/Macro_\(computer_science\)](https://en.wikipedia.org/wiki/Macro_(computer_science))

⁹ <https://en.wikipedia.org/wiki/SPSS>

4. Δημιουργία αρχείων δικτύου για το Rajek

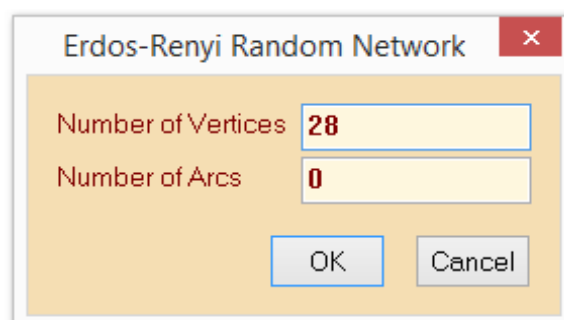
Υπάρχουν αρκετοί τρόποι για την δημιουργία αρχείων δεδομένων που θα μπορούν να διαβαστούν από το Rajek. Οι τρεις κυριότεροι είναι:

- i. Μέσω του Rajek
- ii. Με επεξεργαστή κειμένου (πχ: notepad)
- iii. Μέσω του Matlab
- iv. Μέσω της γλώσσας προγραμματισμού Python
- v. Μέσω της εφαρμογής Excel2Rajek.exe
- vi. Μέσω της εφαρμογής TXT2Rajek.exe
- vii. Δημιουργώντας σχεσιακή βάση δεδομένων (πχ: μέσω Microsoft Access)

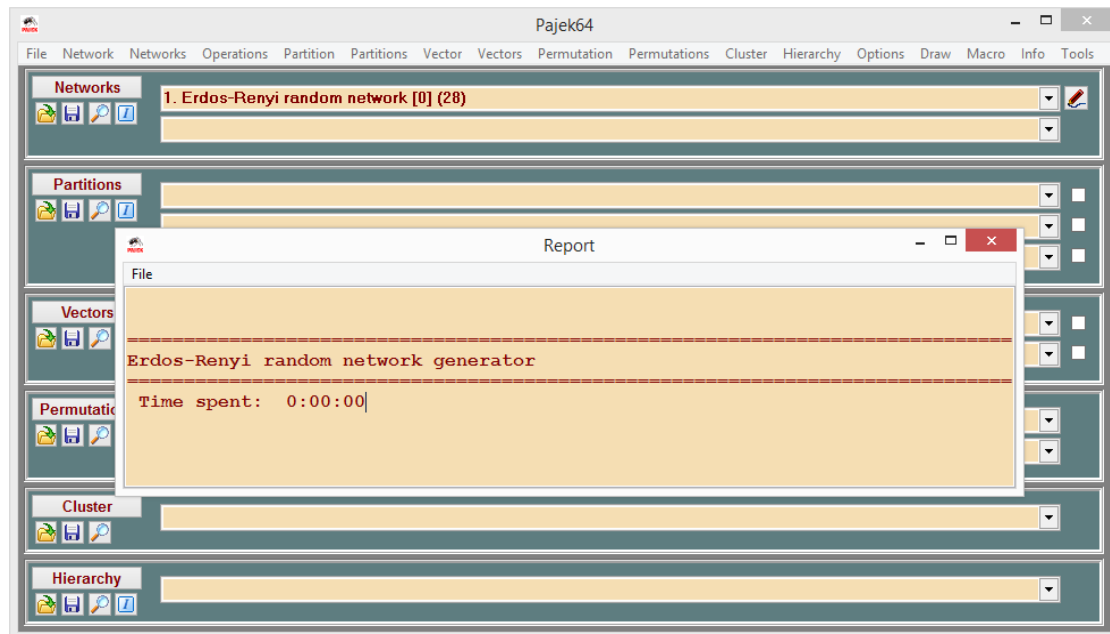
4.1 Μέσω του Rajek

Η δημιουργία ενός δικτύου μέσω του Rajek ή μέσω οποιασδήποτε άλλης μεθόδου από τις τρεις που αναφέρθηκαν στην προηγούμενη παράγραφο, έχει νόημα όταν οι ίδιοι έχουμε συλλέξει τα δεδομένα που επιθυμούμε να αναλύσουμε. Επίσης αποτελεί και ένα πολύ καλό τρόπο εκμάθησης του Rajek.

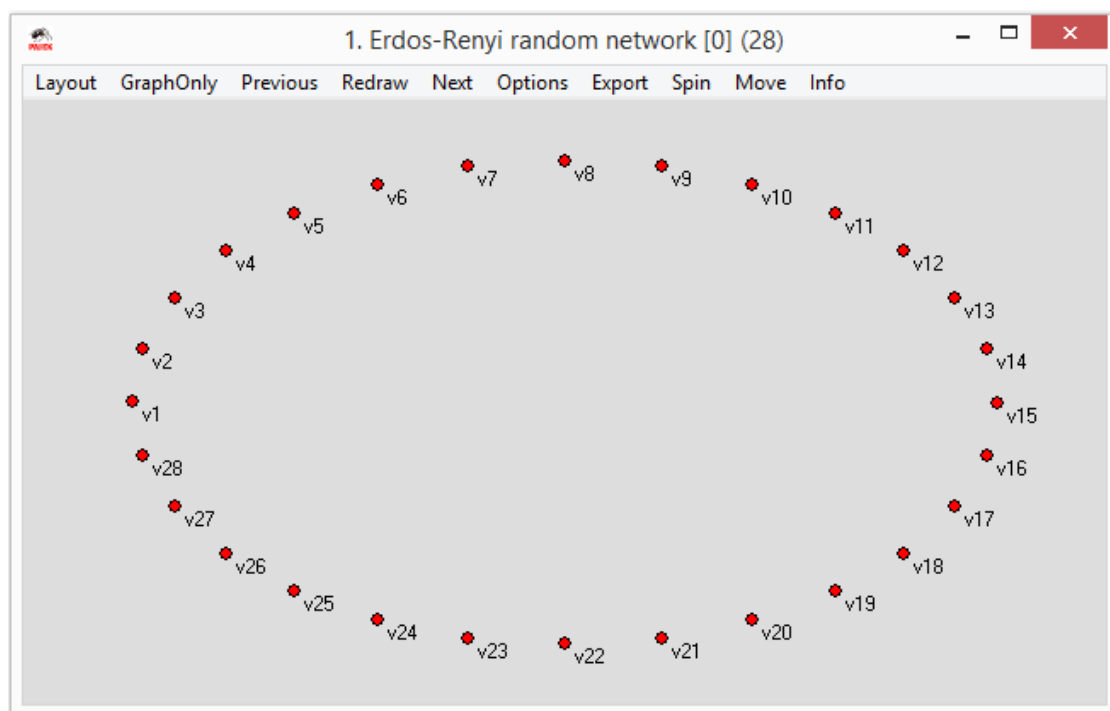
Αρχικά δημιουργούμε ένα τυχαίο δίκτυο επιλέγοντας από το κεντρικό μενού του Rajek: `Network > Create Random Network > Total No. of Arcs`. Στο παράθυρο διαλόγου που εμφανίζεται, θέτουμε έναν αριθμό κορυφών (`Number of Vertices = 28`) και έναν αριθμό γραμμών (`Number of Arcs = 0`) που θα ενώνουν τις κορυφές αυτές:



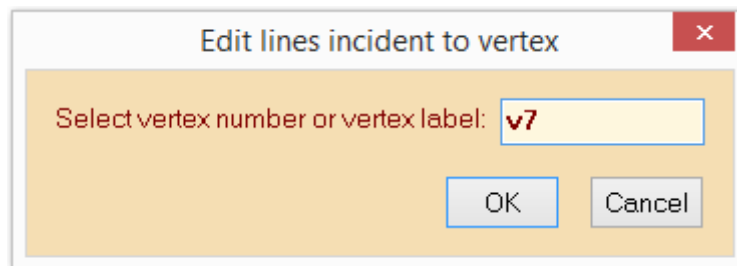
Επιλέγοντας OK, δημιουργείται ένα νέο δίκτυο και προστίθεται στην πτυσσόμενη λίστα διαθέσιμων δικτύων του κεντρικού παραθύρου του Pajek:



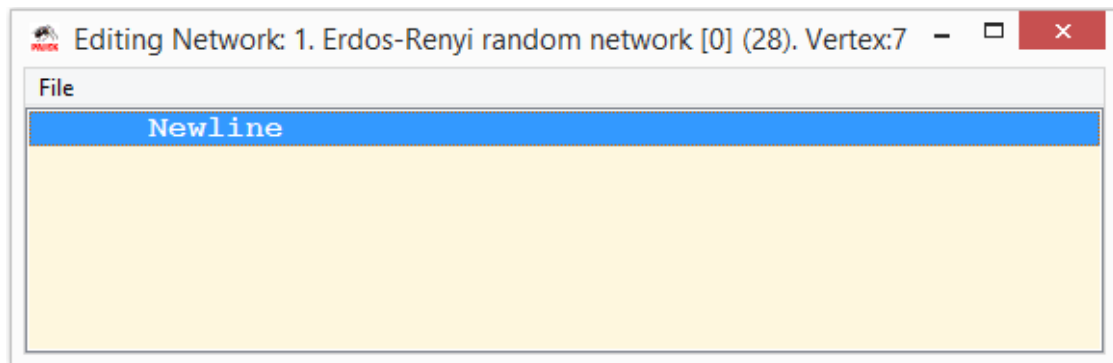
Για να δούμε το δίκτυο που δημιουργήθηκε επιλέγουμε Draw > Network:



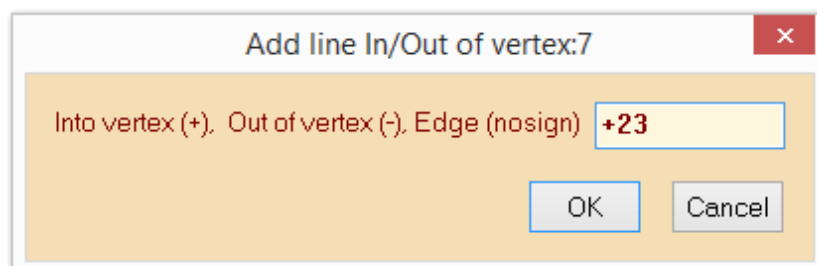
Κατά τη δημιουργία του δικτύου ουσιαστικά παρήχθη ένας ασύνδετος γράφος αφού θέσαμε τον αριθμό των γραμμών να είναι ίσος με το μηδέν. Προκειμένου να προσθέσουμε συνδέσεις στο δίκτυό μας θα πρέπει να επεξεργαστούμε το δίκτυο που δημιουργήθηκε. Αυτό γίνεται, είτε επιλέγοντας: `File > Network > View/Edit`, είτε με κλικ στο κουμπί με το εικονίδιο του μεγεθυντικού φακού, που υπάρχει κάτω από το κουμπί “Networks”. Στο παράθυρο διαλόγου που θα ανοίξει, πληκτρολογούμε, είτε τον αριθμό της κορυφής την οποία θέλουμε να ενώσουμε (vertex number), είτε την ετικέτα της (vertex label, πχ: v7):



Επιλέγοντας OK, εμφανίζεται ένα νέο παράθυρο διαλόγου για τον σχεδιασμό της νέας γραμμής (Newline):



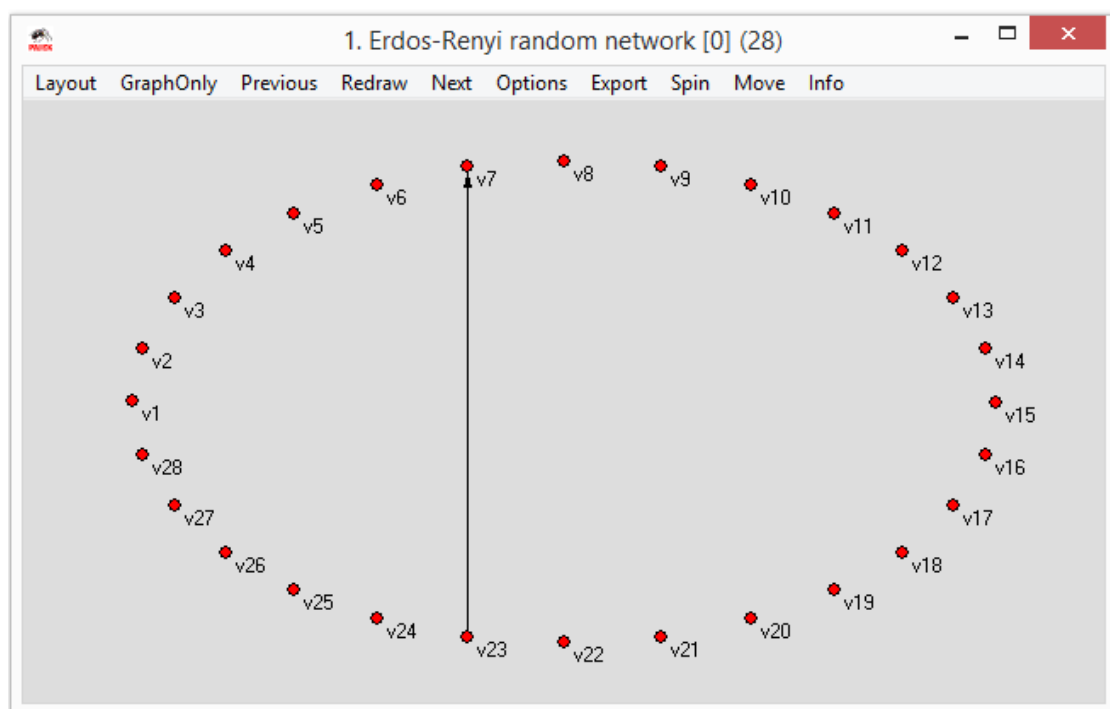
Με διπλό κλικ στο Newline μας δίνεται η δυνατότητα να εισάγουμε τον τρόπο με τον οποίο θα σχεδιαστεί η νέα γραμμή:



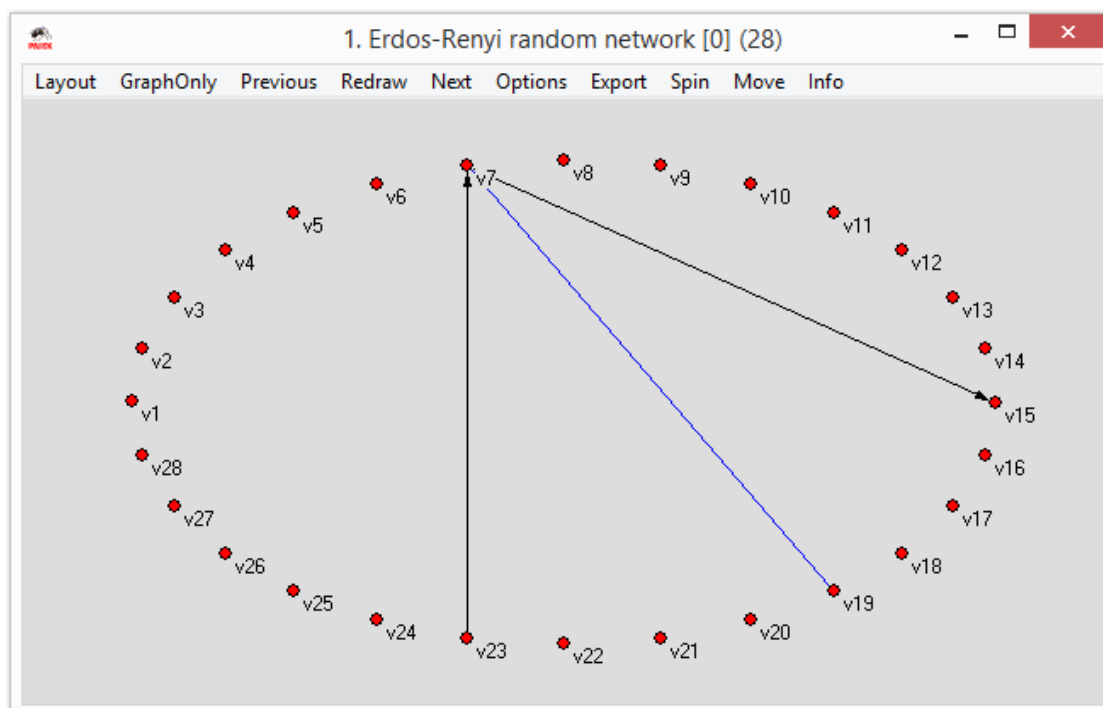
Οι επιλογές σχεδίασης γραμμής που μας δίνονται είναι:

- Από την κορυφή που ενώνουμε, προς τη v7: Into vertex (+)
- Από την v7, προς την κορυφή που ενώνουμε: Out of vertex (-)
- Ένωση της v7 με κάποια άλλη, χωρίς κατεύθυνση: Edge (nosign)

Το αποτέλεσμα της εντολής +23 επομένως, θα είναι μία γραμμή η οποία έχει φορά από την κορυφή 23 προς την κορυφή αναφοράς μας που σε αυτή την περίπτωση είναι η 7:



Προσθέτοντας άλλες δύο συνδέσεις – γραμμές με τους άλλους δύο τρόπους σχεδίασης (Newline, -15 και 19), το δίκτυό μας γίνεται:



Το παράθυρο διαλόγου επεξεργασίας δικτύου, σε αυτό το σημείο έχει ενημερωθεί με τις νέες συνδέσεις:

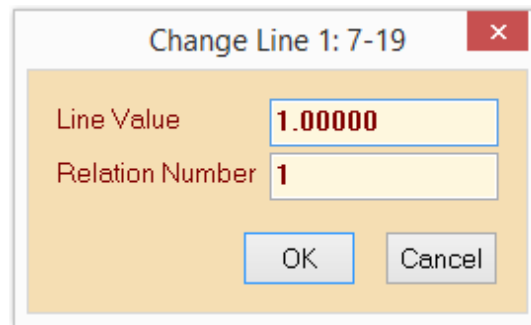
Editing Network: 1. Erdos-Renyi random network [0] (28). Vertex:7

1:	23.7	val=1.00000
1:	7.15	val=1.00000
1:	7-19	val=1.00000

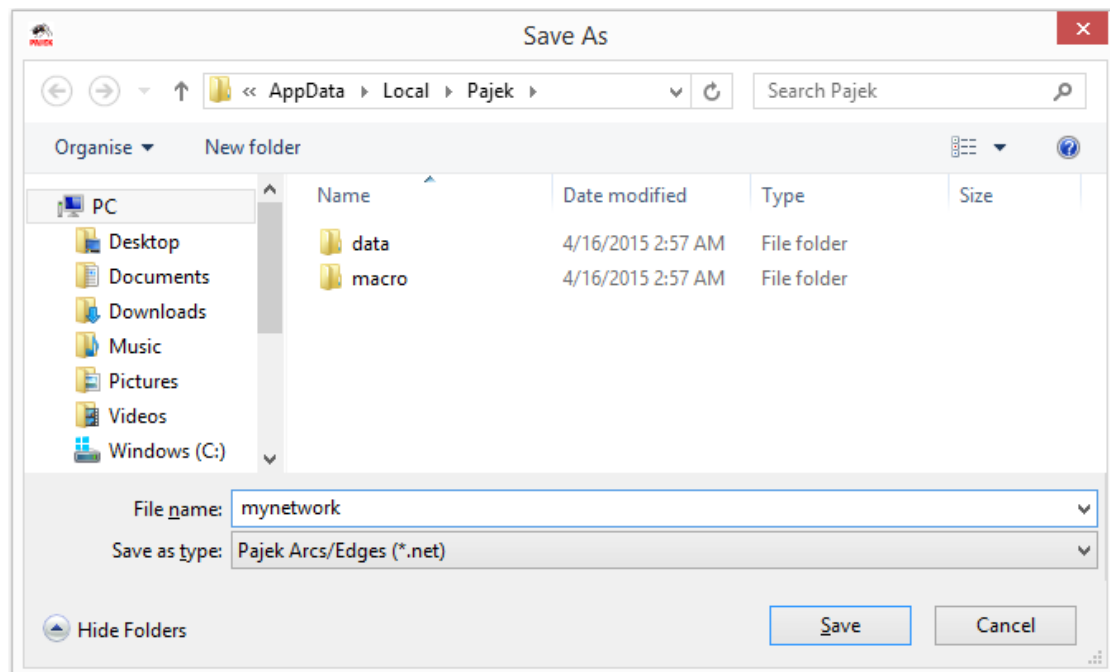
Newline

Με τον ίδιο τρόπο μπορούμε να συνεχίσουμε να δημιουργούμε συνδέσεις προκειμένου να ολοκληρώσουμε το δίκτυο που ενδιαφέρει κάθε φορά.

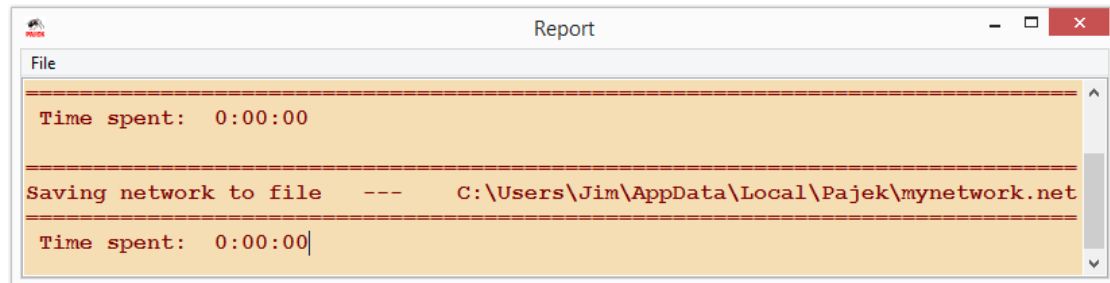
Εξ' ορισμού, όλες οι γραμμές έχουν βαθμό συσχέτισης 1 και τιμή (val) 1.00000, όπως φαίνεται και στην προηγούμενη εικόνα. Μπορούμε να αλλάξουμε αυτές τις τιμές με δεξί κλικ σε όποια μας ενδιαφέρει:



Σε αυτό το σημείο μπορούμε να αποθηκεύσουμε το δίκτυό μας σε αρχείο με επέκταση .net καθώς το Pajek δεν κάνει αυτόματη αποθήκευση:



Μετά την επιτυχή αποθήκευση, μας επιστρέφεται από το Pajek μια σύντομη αναφορά για τις λεπτομέρειες αποθήκευσης:



Τα περιεχόμενα του αρχείου `mynetwork.net` παρατίθενται ως παράρτημα.

4.2 Μέσω του Matlab

Το Matlab αποτελεί πανίσχυρο εργαλείο για την επιστημονική έρευνα. Έχει λοιπόν νόημα να μπορεί να επικοινωνεί με άλλα λογισμικά εισάγοντας ή εξάγοντας δεδομένα σε κατάλληλη μορφή κάθε φορά.

Σε αυτή την παράγραφο παρατίθεται ένα σενάριο δημιουργίας αρχείου `.net`, μέσω του Matlab, για το Pajek.

Για το παράδειγμά μας θα γίνει χρήση των δύο παρακάτω αρχείων Matlab:

- `get_random_graph.m` – Επιστρέφει ένα μη-κατευθυνόμενο, τυχαίο δίκτυο 20 κόμβων, με μέσο όρο 3 συνδέσεις ανά κόμβο (πιθανότητα 3/20 ή 15%), σε μορφή `.mat`.
- `write_matrix_to_pajek.m` – Μετατρέπει δεδομένα Matlab που βρίσκονται σε μορφή `.mat`, σε αρχείο δικτύου Pajek μορφής `.net`.

Ο πηγαίος κώδικας των δύο αυτών αρχείων, παρατίθεται ως παράρτημα στο τέλος του εγγράφου.

Ας υποθέσουμε ότι το παράδειγμά μας αφορά την νευροεπιστήμη. Έστω δηλαδή, ότι έχουμε ένα σύνολο εγκεφαλικών περιοχών R , μεγέθους N . Για κάθε ζεύγος περιοχών i και j (όπου i διάφορο του j), μπορεί να υπάρχει μια νευρωνική σύνδεση ισχύος $S(i,j)$. Επίσης, ας υποθέσουμε ότι δεν γνωρίζουμε την κατευθυντικότητα αυτής της σύνδεσης, πράγμα που σημαίνει ότι $S(i,j) = S(j,i)$. Άρα ο πίνακας γειτνίασης είναι συμμετρικός. Αυτό το εγκεφαλικό δίκτυο, μπορεί να περιγραφεί από ένα πίνακα γειτνίασης

AM ο οποίος θα είναι ένας τετραγωνικός (και συμμετρικός) $M \times N$ πίνακας. Κάθε στοιχείο του πίνακα αυτού περιλαμβάνει μία τιμή η οποία αναπαριστά την ισχύ συνδέσεως μεταξύ των εγκεφαλικών περιοχών i και j . Σε ορολογία δικτύων, ο πίνακας AM αναπαριστά ένα μη-κατευθυνόμενο δίκτυο με βάρη, μεγέθους N .

Αρχικά εκτελούμε στη γραμμή εντολών του Matlab τη συνάρτηση δημιουργίας τυχαίου δικτύου `get_random_graph()` περνώντας τις επιθυμητές παραμέτρους ως εξής:

```
>> AM = get_random_graph(20,3)
```

Το αποτέλεσμα είναι όντως ένα τυχαίο δίκτυο:

```
AM =
    0    1    0    1    0    0    0    0    0    0    1    0    0    0    0    1    0    1    0    0
    1    0    1    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0    0
    0    1    0    0    0    0    1    0    0    0    0    1    0    0    0    1    0    0    0    0
    1    0    0    0    0    0    0    0    0    0    0    1    1    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    0
    0    0    1    0    0    0    0    1    0    0    0    0    0    0    0    0    1    0    0    0
    0    0    0    0    0    1    0    0    0    0    0    0    1    0    0    0    0    1    0    0
    0    0    1    0    0    0    0    0    0    0    0    1    0    0    0    1    0    0    1    0
    0    0    0    0    0    0    0    0    0    0    0    0    1    0    1    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    1    0    0    0    0
    1    1    0    0    0    0    0    0    0    0    0    0    1    1    0    0    0    0    1    1
    0    0    1    1    0    0    0    1    0    1    0    0    0    0    0    0    0    0    1    0
    0    0    0    1    0    0    1    0    0    0    1    0    0    0    0    0    0    0    1    0
    0    0    0    0    0    0    0    0    1    0    1    0    0    0    0    0    0    0    1    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    0
    1    0    1    0    0    0    0    1    0    1    0    0    0    0    1    0    0    0    1    1
    0    0    0    0    0    1    0    0    0    0    0    0    0    0    0    0    1    0    0    1
    1    0    0    0    0    0    1    0    0    0    0    0    0    0    0    1    0    0    0    0
    0    0    0    0    1    0    0    1    0    0    1    1    1    1    0    1    0    0    0    1
    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    1    1    0    1    0
```

Τα περιεχόμενά του είναι όμως μηδενικά και άσσοι. Επειδή θέλουμε το δίκτυό μας να είναι κατευθυνόμενο, εκτελούμε ένα επιπλέον βήμα:

```
>> AM = AM.*rand(20);
```

Μετά όμως από αυτό το βήμα, ο πίνακας AM δεν είναι συμμετρικός. Για να είμαστε ακριβείς θα έπρεπε να καθρεπτίσουμε το άνω τρίγωνό (δηλαδή, τα στοιχεία πάνω από την κύρια διαγώνιο) του στο κάτω. Καθώς όμως η συνάρτηση δημιουργίας του αρχείου .net θα χρησιμοποιήσει μόνο τα στοιχεία επάνω από την κύρια διαγώνιο (λόγω του ότι η παράμετρος `directed` έχει την τιμή `false`), δεν χρειάζεται να προχωρήσουμε σε αυτή την ενέργεια.

Για να αναπαραστήσουμε το δίκτυό μας σε τρισδιάστατη μορφή (όπως είναι διατεταγμένοι και οι πραγματικοί νευρώνες του εγκεφάλου), θα πρέπει να καθορίσουμε

συντεταγμένες για την κάθε κορυφή του δικτύου. Μόνο που και σε αυτή την περίπτωση, οι συντεταγμένες θα είναι τυχαίες αφού θα δημιουργηθούν μέσω της εκτέλεσης της παρακάτω εντολής:

```
>> coordinates = rand(20,3)
```

Και το αποτέλεσμα είναι:

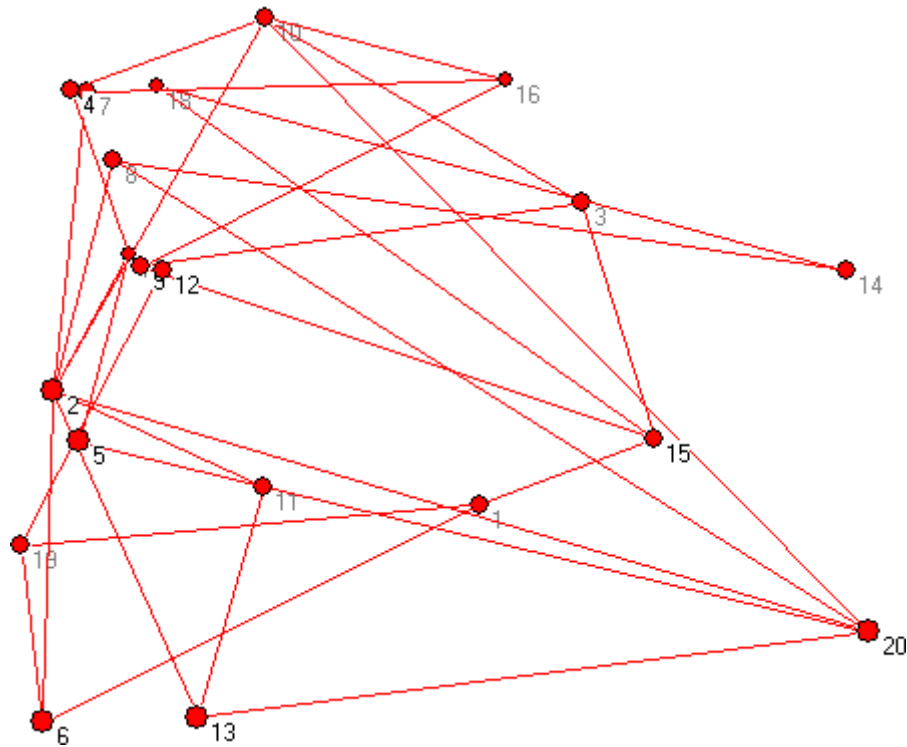
```
coordinates =  
  
    0.5312    0.7067    0.3508  
    0.1088    0.5578    0.6855  
    0.6318    0.3134    0.2941  
    0.1265    0.1662    0.5306  
    0.1343    0.6225    0.8324  
    0.0986    0.9879    0.5975  
    0.1420    0.1704    0.3353  
    0.1683    0.2578    0.2992  
    0.1962    0.3968    0.4526  
    0.3175    0.0740    0.4226  
    0.3164    0.6841    0.3596  
    0.2176    0.4024    0.5583  
    0.2510    0.9828    0.7425  
    0.8929    0.4022    0.4243  
    0.7032    0.6207    0.4294  
    0.5557    0.1544    0.1249  
    0.1844    0.3813    0.0244  
    0.2120    0.1611    0.2902  
    0.0773    0.7581    0.3175  
    0.9138    0.8711    0.6537
```

Σε αυτό το σημείο θα μπορούσαμε να δώσουμε ονόματα στις κορυφές του δικτύου δημιουργώντας ένα vector με strings (τα ονόματα των περιοχών του εγκεφάλου). Όμως, απουσία αυτού του vector, η συνάρτηση θα χρησιμοποιήσει τους δείκτες των περιοχών (1 έως N) σαν ετικέτες (labels), κάτι που για τις ανάγκες αυτού του παραδείγματος είναι αρκετό.

Αφού πλέον έχουμε καθορίσει το δίκτυό μας, μπορούμε να το εξαγάγουμε σε μορφή .net για το Pajek, με την εξής εντολή:

```
write_matrix_to_pajek(AM,'random.net','weighted',true,'directed',false,  
                      'coords',coordinates);
```

Το αρχείο random.net έχει δημιουργηθεί στον τρέχοντα κατάλογο εργασίας του Matlab (Τα περιεχόμενά του παρατίθενται σε αντίστοιχο παράρτημα στο τέλος του εγγράφου), και μπορεί να εισαχθεί στο Pajek. Επιλέγοντας στη συνέχεια, Draw > Network, εμφανίζεται το δίκτυό μας:



Εφόσον είναι τρισδιάστατο, μπορούμε να το περιστρέψουμε χρησιμοποιώντας τα πλήκτρα x, y, z, του πληκτρολογίου.

4.3 Μέσω της γλώσσας προγραμματισμού Python

Η γλώσσα προγραμματισμού Python¹⁰ είναι πλέον η δεύτερη δημοφιλέστερη μετά την Java. Ειδικά στην ακαδημαϊκή κοινότητα, πολλές φορές έρχεται πρώτη, ενώ σε αρκετές περιπτώσεις αντικαθιστά ακόμα και το λογισμικό Matlab. Για το λόγο αυτό υπάρχει τεράστια εγκατεστημένη βάση προγραμματιστών που αναπτύσσουν και επεκτείνουν τη λειτουργικότητά της. Η ανάπτυξη βιβλιοθηκών είναι ίσως ο κυριότερος τρόπος επέκτασής της Python.

¹⁰ [http://en.wikipedia.org/wiki/Python_\(programming_language\)](http://en.wikipedia.org/wiki/Python_(programming_language))

Όσον αφορά την υποστήριξη της Python για επεξεργασία γράφων και δικτύων, υπάρχουν δύο βιβλιοθήκες οι οποίες αναφέρονται συχνότερα στο Διαδίκτυο: Η βιβλιοθήκη NetworkX¹¹ και η βιβλιοθήκη iGraph¹². Και οι δύο βιβλιοθήκες διαθέτουν υποστήριξη για το Pajek.

Συγκεκριμένα, η υλοποίηση¹³ της βιβλιοθήκης NetworkX για το Pajek, υποστηρίζει κατευθυνόμενους και μη-κατευθυνόμενους γράφους, συμπεριλαμβανομένων εκείνων με κόμβους που με βρόχους προς τον εαυτό τους, αλλά και εκείνων με παράλληλες ακμές. Στον παρακάτω πίνακα παρουσιάζονται οι τρεις συναρτήσεις της βιβλιοθήκης για ανάγνωση, εγγραφή και σάρωση γράφων Pajek:

<code>read_pajek</code> (path[, encoding])	Read graph in Pajek format from path.
<code>write_pajek</code> (G, path[, encoding])	Write graph in Pajek format to path.
<code>parse_pajek</code> (lines)	Parse Pajek format graph from string or iterable.

Η βιβλιοθήκη iGraph έχει και αυτή παρόμοιες δυνατότητες με την NetworkX, αλλά παρουσιάζει περισσότερες δυσκολίες κατά την εγκατάστασή της. Σε αντίθεση με την δεύτερη, δεν μπορεί να εγκατασταθεί από τη γραμμή εντολών της Python μέσω της εντολής `pip install`.

4.4 Με επεξεργαστή κειμένου

Η χρήση ενός επεξεργαστή κειμένου για τη δημιουργία ενός γράφου (ή δικτύου) από το πληκτρολόγιο, μπορεί να έχει, σχεδόν αποκλειστικά, εκπαιδευτικό σκοπό. Ακόμα και αν πρόκειται για την ανάγκη εισαγωγής πρωτογενών δεδομένων, αυτά συνήθως θα λαμβάνονται και στη συνέχεια θα μορφοποιούνται με εργαλεία που έχουν σχεδιαστεί ειδικά για το σκοπό αυτό.

Παρ' όλα αυτά, είναι αρκετά χρήσιμο να γνωρίζει κανείς τη δομή ενός αρχείου `.net` αν πρόκειται να χρησιμοποιήσει ως βασικό του εργαλείο το Pajek. Για παράδειγμα, η ανάπτυξη των βιβλιοθηκών python που αναφέρθηκαν στην προηγούμενη παράγραφο, έγινε εφικτή με προαπαιτούμενη αυτή τη γνώση.

¹¹ <http://networkx.github.io/documentation/development/reference/readwrite.pajek.html>

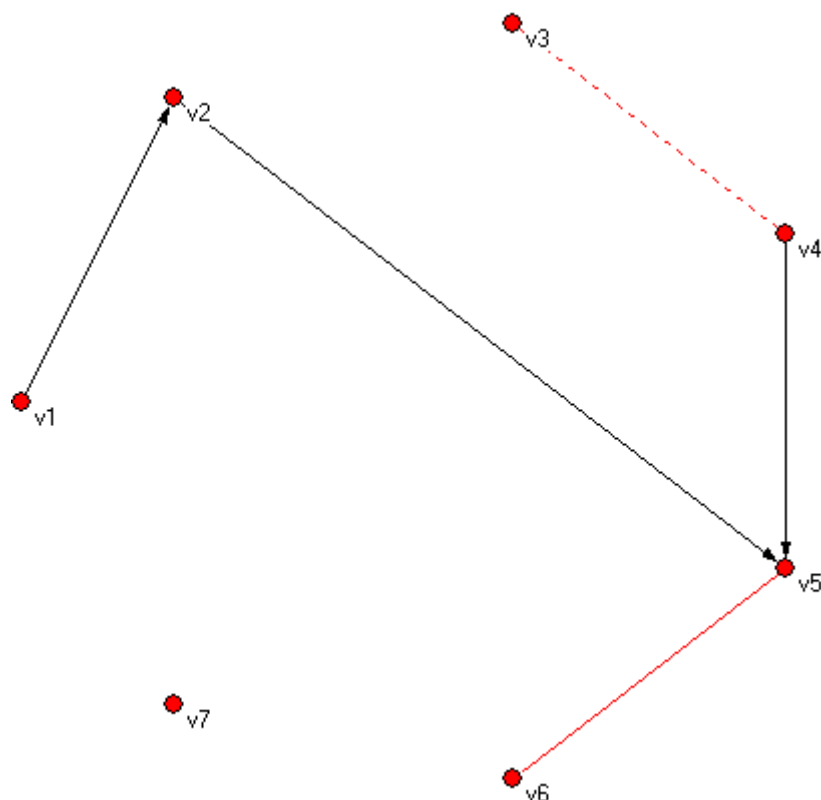
¹² <http://igraph.org/python/doc/tutorial/tutorial.html#igraph-and-the-outside-world>

¹³ www.networkx.github.io/documentation/development/_modules/networkx/readwrite/pajek.html

Ας δούμε ένα παράδειγμα μίας πολύ απλής μορφής αρχείου .net:

```
*Vertices 7
1 "v1"          0.1000    0.5000    0.5000
2 "v2"          0.2506    0.1873    0.5000
3 "v3"          0.5890    0.1100    0.5000
4 "v4"          0.8604    0.3264    0.5000
5 "v5"          0.8604    0.6736    0.5000
6 "v6"          0.5890    0.8900    0.5000
7 "v7"          0.2506    0.8127    0.5000
*Arcs
1 2 1
2 5 2.500000000
4 5 1
*Edges
3 4 -1
5 6 1
```

Η γραφική του απεικόνιση (Draw>Network) φαίνεται στην επόμενη εικόνα:



Στην εικόνα αυτή διακρίνουμε 7 κορυφές (Vertices), 3 τόξα (Arcs) και 2 ακμές (Edges). Στο αντίστοιχο αρχείο .net, κάτω από τις αγγλικές μεταφράσεις των τριών αυτών στοιχείων, περιλαμβάνονται γραμμές κειμένου στις οποίες καθορίζονται ορισμένες από τις λεπτομέρειες σύνδεσης των κορυφών.

Πιο συγκεκριμένα, η πρώτη γραμμή (`*Vertices 7`) ακολουθείται από επτά γραμμές οι οποίες περιλαμβάνουν τις επτά κορυφές του δικτύου. Καθεμιά από αυτές τις γραμμές περιλαμβάνει αντίστοιχα, τον αριθμό της κορυφής (1 έως 7), το σύμβολό της (`v1` έως `v7`), και τρεις τιμές που αντιπροσωπεύουν τις συντεταγμένες της στο χώρο (`x`, `y`, `z`). Δεν επιτρέπονται κενές γραμμές και tabs.

Κάτω από την γραμμή με την επικεφαλίδα `*Arcs`, υπάρχουν τρεις γραμμές που αφορούν τα τόξα κατευθυνόμενων συνδέσεων. Ο πρώτος αριθμός αφορά την κορυφή από την οποία ξεκινά το βέλος και ο δεύτερος, την κορυφή στην οποία δείχνει. Ο τρίτος επηρεάζει το στυλ της γραμμής. Για παράδειγμα αν έχει αρνητικό πρόσημο σχεδιάζεται διακεκομμένη.

Τέλος, οι μη-κατευθυνόμενες συνδέσεις περιγράφονται κάτω από τον τίτλο `*Edges`.

Ένας δεύτερος τρόπος σύνταξης του αρχείου `.net` είναι για τις συνδέσεις των κορυφών να χρησιμοποιηθεί πίνακας. Έτσι, η δομή του προηγούμενου δικτύου θα μπορούσε να συνταχθεί ως εξής:

```
*Vertices 7
1 "v1"          0.1000    0.5000    0.5000
2 "v2"          0.2506    0.1873    0.5000
3 "v3"          0.5890    0.1100    0.5000
4 "v4"          0.8604    0.3264    0.5000
5 "v5"          0.8604    0.6736    0.5000
6 "v6"          0.5890    0.8900    0.5000
7 "v7"          0.2506    0.8127    0.5000

*Matrix
0 1 0 0 0 0 0
0 0 0 0 2.5 0 0
0 0 0 -1 0 0 0
0 0 -1 0 1 0 0
0 0 0 0 0 1 0
0 0 0 0 1 0 0
0 0 0 0 0 0 0
```

Ο πίνακας θα πρέπει να είναι τετραγωνικός και το μέγεθός του να συμφωνεί με τον αριθμό των κορυφών, διαφορετικά το RajeK πέφτει σε σφάλμα.

Ο συγκεκριμένος τρόπος σύνταξης όμως έχει ένα προφανές μειονέκτημα. Το RajeK δεν είναι σε θέση να ξεχωρίσει της κατευθυνόμενες από τις μη-κατευθυνόμενες συν-

δέσεις. Επομένως καλό θα είναι να αποφεύγεται για την περιγραφή σύνθετων δικτύων με αμφίδρομες συνδέσεις μεταξύ των κορυφών. Μπορεί να γίνει μετατροπή των κατευθυνόμενων σε μη, με την εντολή `Network > Create New Network > Transform > Arcs to Edges > Bidirectional only`.

Όλα τα παραπάνω φυσικά, αφορούν δίκτυα one-mode όπου όλοι οι κόμβοι (κορυφές) είναι ίδιου τύπου (πχ: πόλεις). Στα δίκτυα 2-mode που υπάρχουν δύο διαφορετικοί σύνολοι κόμβων, το αρχείο `.net` συντάσσεται ελαφρώς διαφορετικά. Οπότε αν υποθέσουμε ότι έχουμε 5 υπαλλήλους οι οποίοι εργάζονται σε 2 διαφορετικές εταιρείες, το αρχείο `.net` θα μπορούσε να είναι το ακόλουθο:

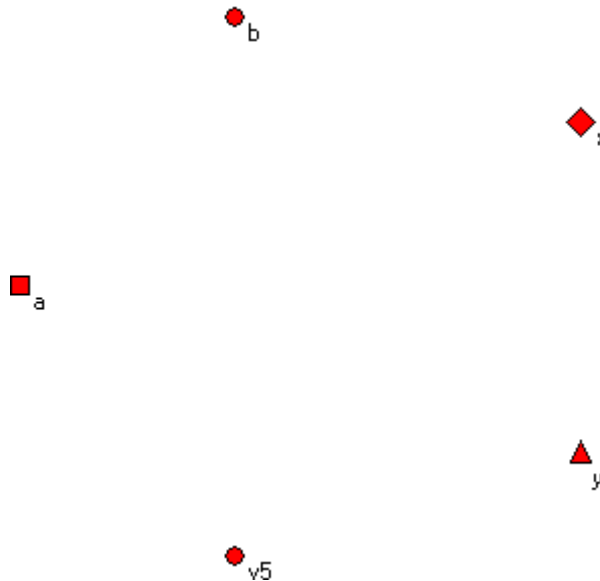
```
*Vertices 6 2
1 "company1"          0.1000    0.5000    0.5000
2 "company2"          0.3000    0.1536    0.5000
3 "employee1"         0.7000    0.1536    0.5000
4 "employee2"         0.9000    0.5000    0.5000
5 "employee3"         0.7000    0.8464    0.5000
6 "employee4"         0.3000    0.8464    0.5000
*Arcs
1 3 1
2 5 2.5000000000
1 4 1
*Edges
2 4 -1
1 6 1
```

Σε αυτή την περίπτωση ο δεύτερος αριθμός στην επικεφαλίδα `*Vertices` μπαίνει για να συμβολίσει το two-mode δίκτυο.

Μέχρι στιγμής αναφέρθηκαν μόνο τα πιο βασικά δομικά στοιχεία τα οποία είναι απαραίτητα για να καθορίσουν τη δομή ενός δικτύου. Το Rajek, επιτρέπει επιπλέον μορφοποίηση και καθορισμό των χαρακτηριστικών των στοιχείων που ανήκουν στο δίκτυο (συντεταγμένες των κορυφών στο χώρο, σχήμα και χρώμα των κορυφών και των γραμμών). Για παράδειγμα, το παρακάτω αρχείο δικτύου:

```
*Vertices 5
1 "a" box
2 "b" ellipse
3 "x" diamond
4 "y" triangle
```


Έχει ως αποτέλεσμα:



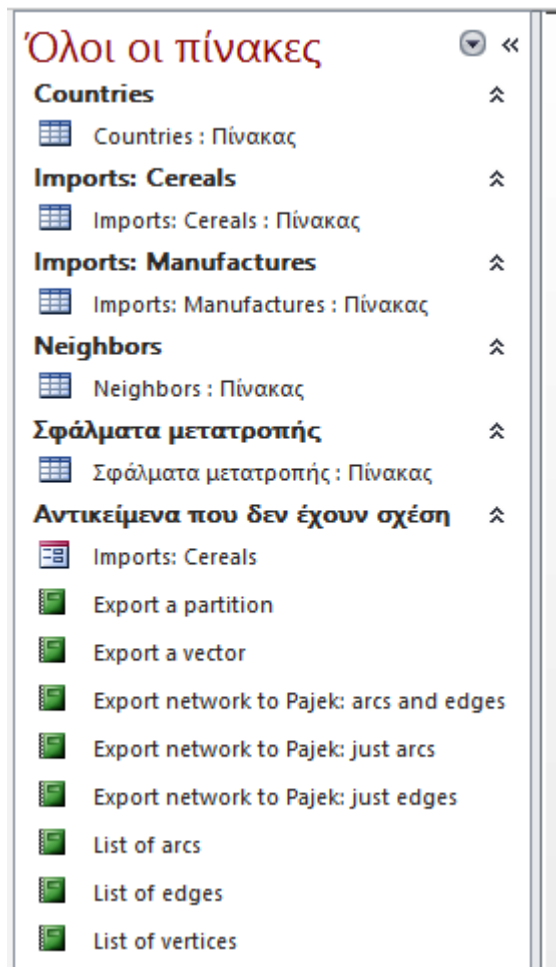
Παρατηρούμε ότι ενώ στο αρχείο περιγράφουμε 4 κορυφές, στο δίκτυο που σχεδιάστηκε υπάρχουν 5. Αυτό θα μπορούσε να συμβεί ακόμα και αν δεν περιγράψαμε καμία κορυφή αλλά αφήναμε μόνο την γραμμή `*Vertices 5` την οποία το Pajek αυτόματα τη μεταφράζει και τη σχεδιάζει ως ένα δίκτυο 5 κορυφών με τις προεπιλεγμένες από το πρόγραμμα ρυθμίσεις τους. Γι' αυτό και στην Πέμπτη κορυφή επελέγησαν αυτόματα οι συντεταγμένες και το σχήμα της.

4.5 Δημιουργώντας σχεσιακή βάση δεδομένων

Τα δεδομένα των πολύ μεγάλων δικτύων είναι δυνατό να αποθηκεύονται σε σχεσιακές βάσεις δεδομένων. Σε μια τέτοια βάση, δύο πίνακες (tables) μπορούν να αντιπροσωπεύσουν ένα δίκτυο one-mode. Ο ένας μπορεί να περιλαμβάνει τις κορυφές του δικτύου και ο άλλος τις γραμμές (συνδέσεις) μεταξύ των κορυφών αυτών.

Ως παράδειγμα μπορούν να χρησιμοποιηθούν δεδομένα¹⁴ που αφορούν τις εισαγωγές μετάλλου σε όλο τον κόσμο (Βλέπε παράγραφο: «Σενάριο χρήσης με πραγματικά δεδομένα»):

¹⁴ <http://vlado.fmf.uni-lj.si/pub/networks/data/esna/metalWT.htm>



Ο πίνακας Countries περιλαμβάνει εγγραφές για όλες τις χώρες που συμμετέχουν και αποτελεί τον πυρήνα όλης της βάσης:

Countries													
vertexnr	vertexlabel	x	y	continent	trade bloc	position	totcereal	totmanuf	GDPgrowth	GDP95	Popugrowth	Popul95	
1	Algeria	0.4380682	0.4957265	1		0	999998	1303138	277449	1.6	1209	2.36	28058
2	Argentina	0.2159091	0.7749288	6	3	2	53155	572691	6.1	8042	1.33	34768	
3	Australia	0.8636364	0.7008547	5	0	2	107963	1237693	3.3	20333	1.22	17946	
4	Austria	0.46875	0.411396	3	0	2	233427	1969990	2	28942	0.75	8001	
5	Barbados	0.2670454	0.5766382	6	4	999998	17550	20868	1	7053	0.55	264	
6	Bangladesh	0.7301137	0.5071225	2	8	999998	66647	18530	4.9	272	1.61	118616	
7	Belgium, Lux.	0.45	0.3931624	3	1	1	1660877	2774659	1.5	27129	0.27	10495	
8	Belize	0.1352273	0.5259259	4	4	999998	6461	13210	3.9	2753	2.59	213	
9	Bolivia	0.2215909	0.6951567	6	3	999998	62671	48506	3.9	885	2.41	7414	
10	Brazil	0.2897727	0.6780627	6	3	2	1597383	439099	2.9	4501	1.49	159346	
11	Canada	0.1363636	0.3304843	4	2	1	667341	4182836	1.8	19362	1.27	29617	
12	Chile	0.1931818	0.7635328	6	3	3	204441	319689	8.3	4589	1.63	14210	
13	China	0.75	0.4643874	2	0	999998	1321437	1587749	11.2	584	1.1	1220516	
14	Colombia	0.2215909	0.5925926	6	3	3	410684	210572	4	2089	1.95	38542	
15	Croatia	0.4886364	0.4273504	3	0	2	18530	110318	-2.8	4187	-0.11	4493	
16	Cyprus	0.5142046	0.4700855	3	0	999998	75393	89035	3.7	11818	1.78	744	
17	Czech Rep.	0.4772727	0.4017094	3	0	999998	76889	423479	-1.3	4922	0.04	10325	
18	Denmark	0.4602273	0.3760684	3	1	2	285238	1006509	2.2	33121	0.33	5225	
19	Ecuador	0.1761364	0.6210826	6	3	3	110140	91762	3.2	1565	2.2	11460	
20	Egypt	0.5170455	0.5071225	1	0	3	1110127	231082	4	970	2.01	62282	
21	El Salvador	0.1306818	0.5413105	4	0	4	93627	62776	5.3	1675	2.07	5669	
22	Fiji	0.9886363	0.6951567	5	0	999998	29275	26847	1.2	2604	1.11	768	
23	Finland	0.5113636	0.3304843	3	0	2	95136	540905	0.9	24653	1.11	5108	
24	France	0.4375	0.4159544	3	0	1	866599	550688	1.3	26458	0.45	58020	
25	French Guiana	0.2784091	0.6096866	6	0	999998	10568	39364	3.9	11415	4.53	147	
26	Germany	0.4659091	0.3988604	3	1	1	1974426	9018540	2.2	29419	0.57	81661	
27	Greece	0.5028409	0.4501424	3	1	2	244815	434988	1.7	10900	0.52	10489	
28	Guadeloupe	0.3613636	0.5754096	6	0	999998	49696	62420	0.2	8704	1.65	474	

Ο πίνακας `Imports` περιλαμβάνει τις πληροφορίες σύνδεσης μεταξύ των χωρών και τα βάρη των συνδέσεων αυτών:

tail ▼	head ▼	value ▼	valperc ▼
Canada	Japan	549807	8.823322
United States	Japan	3092924	49.63536
Argentina	Japan	100280	1.609297
China	Japan	883471	14.17797
Thailand	Japan	428337	6.873968
Germany	Japan	80041	1.284501
Australia	Japan	573428	9.202394
Canada	Italy	68432	4.339227
United States	Italy	85905	5.447178
Belgium, Lux.	Italy	21625	1.371227
Denmark	Italy	19973	1.266474
France	Italy	703680	44.61988
Germany	Italy	227383	14.4182
Greece	Italy	122386	7.760414
Netherlands	Italy	24742	1.568874
Spain	Italy	34926	2.214634
United Kingdom	Italy	149554	9.483119
Hungary	Italy	25070	1.589672
Canada	Australia	2361	2.18686
United States	Australia	18545	17.36081

Ο πίνακας `Manufactures` περιλαμβάνει εγγραφές για τις μη-κατευθυνόμενες συνδέσεις (edges) του δικτύου:

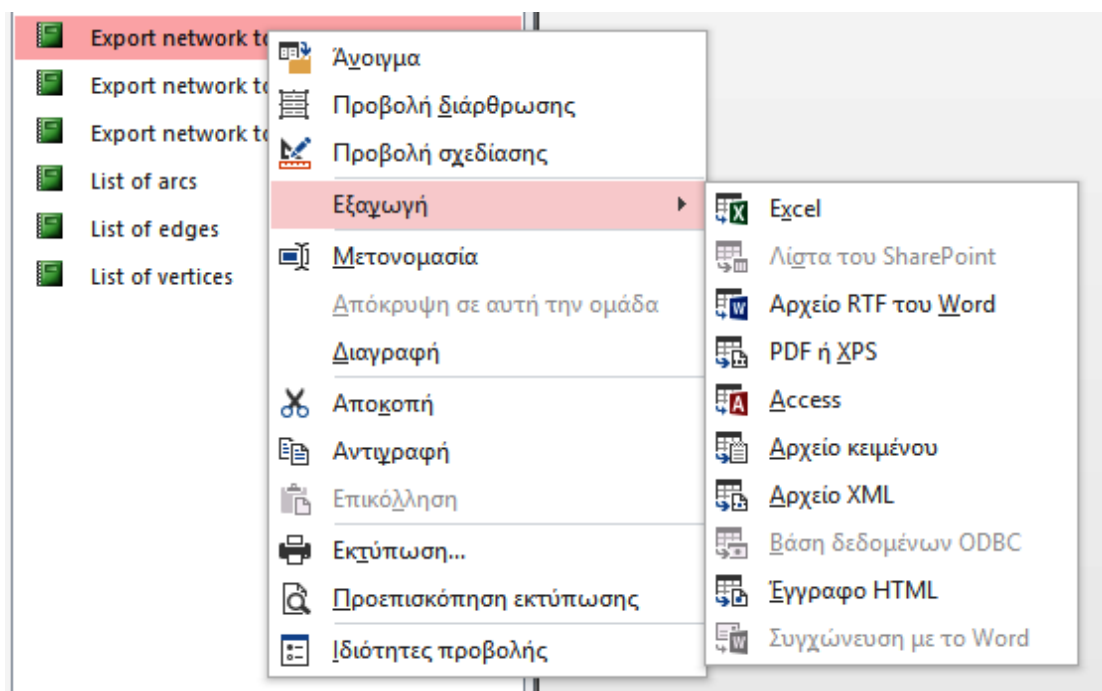
tail ▼	head ▼	value ▼	valperc ▼
78	25	556	1.412458
24	25	23023	58.48745
26	25	12714	32.29855
38	25	1243	3.157707
78	28	2141	3.375374
7	28	1586	2.500394
24	28	47886	75.49425
26	28	1242	1.958064
38	28	6045	9.53019
78	53	55418	4.39644
39	53	14864	1.179196
13	53	13802	1.094945
7	53	21765	1.726668
10	53	91722	6.49407

Ο πίνακας Neighbor, παίζει το ρόλο του πίνακα γειτνίασης:

vertex1	vertex2	value
United States	Canada	1
Egypt	Israel	1
Morocco	Algeria	1
Indonesia	Malaysia	1
Malaysia	Thailand	1
Bangladesh	India	1
India	Pakistan	1
India	China	1
Pakistan	China	1
Jordan	Israel	1
Turkey	Greece	1
Croatia	Slovenia	1
Croatia	Hungary	1

Μέσω της φόρμας Imports: Cereals, γίνεται η εισαγωγή των πρωτογενών δεδομένων.

Τα υπόλοιπα, είναι ερωτήματα προς τη βάση δεδομένων. Ένα από αυτά είναι και το Export network to Pajek: arcs and edges. Μέσω αυτού του ερωτήματος (με δεξί κλικ), καταφέρνουμε τελικά να εξάγουμε τα δεδομένα μας σε μορφή αρχείου που θα μπορεί στη συνέχεια να εισαχθεί στο Pajek:



4.6 Μέσω άλλων τύπων αρχείων

Ένας άλλος τρόπος δημιουργίας δικτύου με το Pajek, είναι πρώτα αυτό να εξαχθεί από κάποια άλλη εφαρμογή, σε κάποιο αποδεκτό format, και στη συνέχεια να γίνει η εισαγωγή του μέσω του Pajek. Οι τύποι που υποστηρίζονται είναι:

- UCINET DL¹⁵ (gephi)
- Vega¹⁶ (Tomaz Pisanski project)
- GEDCOM¹⁷ (Genealogical Data Communication)
- BS (Ball and Stick)
- MAC (Mac Molecule)
- MOL¹⁸ (MDL MOLfile)

5. Καθορισμός διάταξης δικτύου

Η κεντρική ιδέα ενός δικτύου συνήθως λαμβάνεται από τον άνθρωπο μέσω της απεικόνισής της διάταξής του. Στο Pajek περιλαμβάνονται αρκετές αυτόματες και χειροκίνητες ρουτίνες σχεδίασης.

5.1 Αυτόματες ρουτίνες σχεδίασης

Οι αυτόματες ρουτίνες διάταξης του Pajek είναι:

- Ενέργειας
- Pivot MDS¹⁹
- VOS Mapping²⁰

Για την εφαρμογή ενός εκ των τριών, αφού έχουμε σχεδιάσει το δίκτυο μέσω του Pajek (Draw/Network), επιλέγουμε Layout:

¹⁵ <http://gephi.github.io/users/supported-graph-formats/ucinet-dl-format>

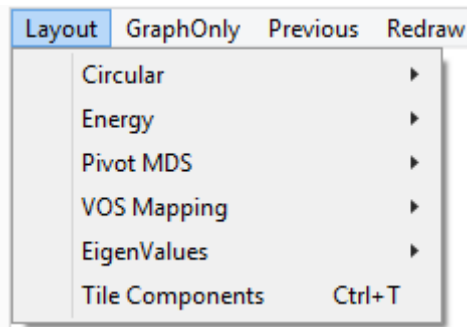
¹⁶ <https://www.google.gr/search?q=Tomaz+Pisanski+Vega+project>

¹⁷ <http://en.wikipedia.org/wiki/GEDCOM>

¹⁸ http://en.wikipedia.org/wiki/Chemical_table_file#Molfile

¹⁹ <http://mrvar.fdv.uni-lj.si/pajek/community/DrawPivotMDS.htm>

²⁰ <http://mrvar.fdv.uni-lj.si/pajek/community/DrawEnergy.htm>



Μέσω των ρουτινών ενέργειας, το δίκτυο παρουσιάζεται ως ένα φυσικό σύστημα (πχ: μια πρωτεΐνη) του οποίου ψάχνουμε την βέλτιστη (ελάχιστη) ενεργειακή κατάσταση. Υπάρχουν δύο αλγόριθμοι:

- Kamada – Kawai [2]. Σχεδίαση σε ένα επίπεδο.
- Fruchterman – Reingold [3]. Σχεδίαση 2D ή 3D.

Οι ρουτίνες Pivot MDS και VOS Mapping εφαρμόζονται σε πολύ μεγάλα δίκτυα, των εκατό χιλιάδων κορυφών (Pivot MDS) και άνω (VOS Mapping). Η ύπαρξη συμμετριών στο δίκτυο, βελτιώνει το αποτέλεσμα της απεικόνισής του, όταν γίνεται χρήση των ρουτινών αυτών.

5.2 Χειροκίνητη σχεδίαση

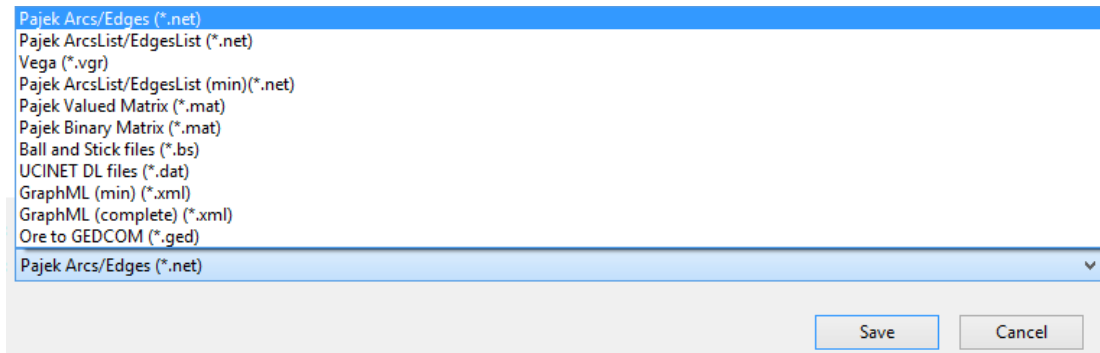
Η χειροκίνητη σχεδίαση, μπορεί να λειτουργεί συμπληρωματικά της αυτόματης όταν κρίνονται χρήσιμες ορισμένες διορθώσεις οι οποίες βελτιώνουν την τελική απεικόνιση του δικτύου. Μπορεί όμως να χρησιμοποιηθεί και μόνο αυτή, ιδιαίτερα όταν πρόκειται για πολύ απλά δίκτυα.

Μπορούμε με το ποντίκι να μετακινούμε την κάθε κορυφή κατά βούληση. Επίσης είναι δυνατό να περιστρέψουμε κάποια εικόνα με τη βοήθεια των πλήκτρων: x , X , y , Y , z , Z (κάθε γράμμα αφορά τον άξονα περιστροφής, ενώ τα κεφαλαία ή τα μικρά αφορούν θετική ή αρνητική διεύθυνση, αντίστοιχα). Μπορεί να επιλεγεί οποιοσδήποτε άξονας περιστροφής επιλέγοντας Spin > Normal. Στη συνέχεια, η περιστροφή γύρω από τον επιλεγμένο άξονα γίνεται με τα πλήκτρα S και s.

Η εστίαση (zoom) σε συγκεκριμένο τμήμα του δικτύου, επιτυγχάνεται κρατώντας το πατημένο το δεξί κλικ και σχεδιάζοντας ένα παράθυρο που περιλαμβάνει το τμήμα αυτό. Η επαναφορά γίνεται επιλέγοντας Redraw.

6. Αποθήκευση δικτύου

Η αποθήκευση του δικτύου γίνεται μέσω της εντολής `File>Network>Save`, σε κάποια από τις επιθυμητές μορφές τις οποίες υποστηρίζει το Pajek:



7. Δημιουργώντας partitions, vectors

7.1 Δημιουργία Partition

Τα partitions χρησιμοποιούνται για να περιγράψουν ονομαστικές ιδιότητες των κορυφών του δικτύου (πχ: 1-άνδρας, 2-γυναίκα). Όπως έχει ήδη αναφερθεί, η επέκταση ενός τέτοιου αρχείου έχει την επέκταση `.clu` (clustering).

Αν για παράδειγμα έχουμε ένα δίκτυο πέντε κορυφών και θέλουμε να τοποθετήσουμε τις κορυφές 1 και 5 στο cluster 1, και όλες τις άλλες στο cluster 2, το αντίστοιχο partition θα περιγράφεται από το παρακάτω αρχείο κειμένου:

```
*Vertices
1
2
2
2
1
```

Φυσικά, η δημιουργία ενός partition είναι εφικτή και μέσω του Pajek επιλέγοντας `Partition>Create Constant Partition(constant = 0)`.

Η διάσταση του partition στο Rajek έχει προεπιλεγμένο μέγεθος ίσο με τον αριθμό των κορυφών του δικτύου που επιλέγεται. Με αυτό τον τρόπο τοποθετούμε όλες τις κορυφές στο cluster 0. Στη συνέχεια επιλέγουμε `File>Partition>Edit` και για κάθε μονάδα εισάγουμε τον αριθμό cluster στο οποίο ανήκει.

Όταν προβάλουμε το δίκτυο μαζί με το πρώτο partition (`Draw > Network + First Partition`), τα διαφορετικά χρώματα των κορυφών θα χρησιμοποιηθούν για να δείξουν σε ποιο cluster ανήκει η κάθε κορυφή.

Ένας δεύτερος τρόπος για τον καθορισμό ενός partition επιτυγχάνεται επιλέγοντας `Draw>Network+Create Null Partition`. Με αυτό τον τρόπο εκτελούνται τρεις ενέργειες:

- Δημιουργείται νέο partition μεγέθους ίσου με τον αριθμό των κορυφών
- Όλες οι κορυφές εντάσσονται στο cluster 0
- Το δίκτυο απεικονίζεται χρησιμοποιώντας το null (κενό) partition (όλες οι κορυφές έχουν χρώμα κυανό)

7.2 Δημιουργία Vector

Τα Vectors (διανύσματα) χρησιμοποιούνται για να περιγράφουν τις αριθμητικές ιδιότητες των κορυφών (πχ: centralities²¹). Η επέκτασή τους είναι `.vec` και ένα παράδειγμα vector θα μπορούσε να είναι και το ακόλουθο:

```
*Vertices 5
0.67
0.19
0.567
1.6789
0.05
```

Η απεικόνιση ενός vector γίνεται επιλέγοντας `Draw > Network + First Vector`. Τα μεγέθη των κορυφών θα είναι ανάλογα με τις τιμές των διανυσμάτων.

²¹ <http://en.wikipedia.org/wiki/Centrality>

8. Το αρχείο .paj

Το αρχείο .paj (αρχείο project για το Pajek), είναι ένας βολικός τρόπος αποθήκευσης όλων των δεδομένων ενός δικτύου μαζί με τα δεδομένα που αφορούν partitions, vectors, permutations, clusters και hierarchies. Με τον τρόπο αυτό αποφεύγεται το χειροκίνητο άνοιγμα του κάθε διαφορετικού αρχείου κάθε φορά που ανοίγουμε το Pajek.

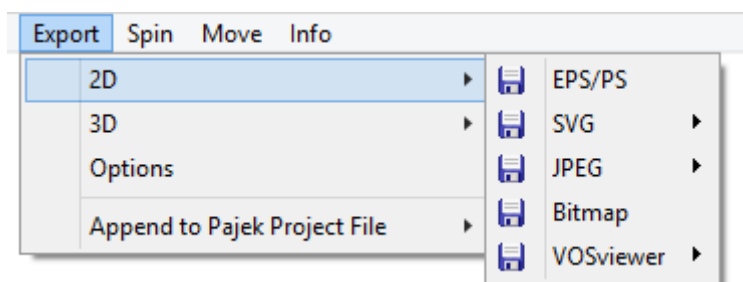
Μπορεί να δημιουργηθεί επιλέγοντας `File>Pajek Project File>Save`. Μετά από αυτή την ενέργεια, όλα τα τρέχοντα αντικείμενα που έχουν φορτωθεί στο Pajek θα αποθηκευτούν στο αρχείο αυτό.

Για το άνοιγμα αρχείου .paj επιλέγουμε `File>Pajek Project File>Read`.

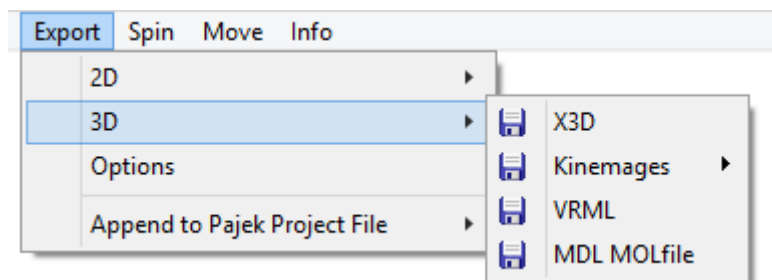
9. Μορφές αρχείων εξόδου του Pajek

Οι απεικονίσεις των δικτύων που δημιουργούνται μέσω του Pajek μπορούν να εξαχθούν σε διάφορες μορφές αρχείων τα οποία στη συνέχεια θα μπορούν να εισαχθούν είτε ως εικόνες σε αρχεία κειμένου, είτε ως αρχεία δεδομένων σε ειδικούς viewers.

Η αποθήκευση σε κάποιο συγκεκριμένο αρχείο εξόδου γίνεται, είτε σε δισδιάστατη μορφή επιλέγοντας `Export>2D`:



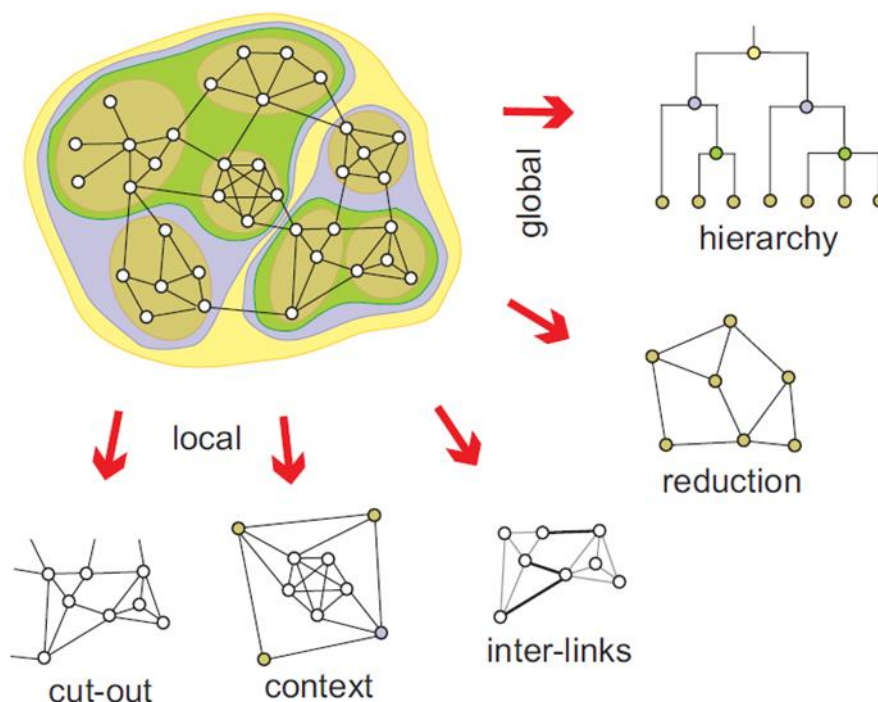
Είτε σε τρισδιάστατη επιλέγοντας `Export>3D`:



10. Τοπικές και ευρείες προβολές δικτύου

Μια τοπική (local) προβολή δημιουργείται επιλέγοντας ένα υποδίκτυο το οποίο εξάγεται από κάποιο σύμπλεγμα κορυφών (πχ: σε μια τάξη μαθητών, οι σχέσεις μεταξύ των αγοριών μόνο).

Μια ευρεία ή οικουμενική (global) προβολή, επιτυγχάνεται μέσω της συρρίκνωσης κορυφών που βρίσκονται στο ίδιο σύμπλεγμα (cluster) σε μια νέα κορυφή. Με αυτό τον τρόπο αναδεικνύονται οι σχέσεις που υπάρχουν μεταξύ των clusters (πχ: σε μια τάξη μαθητών, οι σχέσεις μεταξύ αγοριών και κοριτσιών, δηλαδή ο συνολικός αριθμός των τόξων που ενώνουν τις δύο αυτές ομάδες).

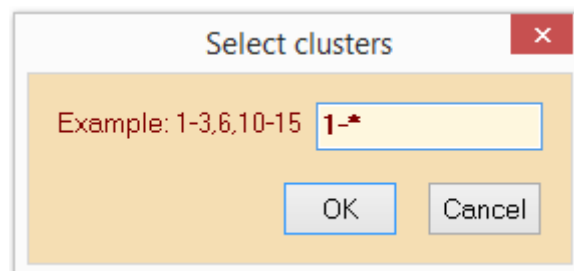


Με λίγα λόγια οι τοπικές προβολές αφορούν συνδέσεις κορυφών μέσα στο ίδιο cluster, ενώ οι ευρείες προβολές, μεταξύ διαφορετικών.

Ο συνδυασμός των δύο παραπάνω διαφορετικών προβολών, ονομάζεται προβολή συνάφειας (contextual view). Μια προβολή συνάφειας προβάλλει τις σχέσεις που υπάρχουν μεταξύ των clusters (πχ: “για κάθε κορίτσι της τάξης, ποιος είναι ο συνολικώς αριθμός των τόξων που συνδέεται με αγόρια).

11. Εξαγωγή υποδικτύων

Εφόσον έχουμε επιλέξει το δίκτυο και το partition που μας ενδιαφέρει, η εξαγωγή του υποδικτύου μπορεί να πραγματοποιηθεί επιλέγοντας `Operations > Network + Partition > Extract > SubNetwork`. Στη συνέχεια μας ζητούνται οι αριθμοί και τα διαστήματα των clusters που θέλουμε να συμπεριληφθούν κατά την εξαγωγή:



Απαραίτητη προϋπόθεση για να μπορεί να γίνει η εξαγωγή ενός υποδικτύου μέσω του Rajek, είναι φυσικά η ύπαρξη ενός αρχείου δικτύου (.net) κι ενός αρχείου partition (.clu) το οποίο έχει προκύψει από το δίκτυο αυτό.

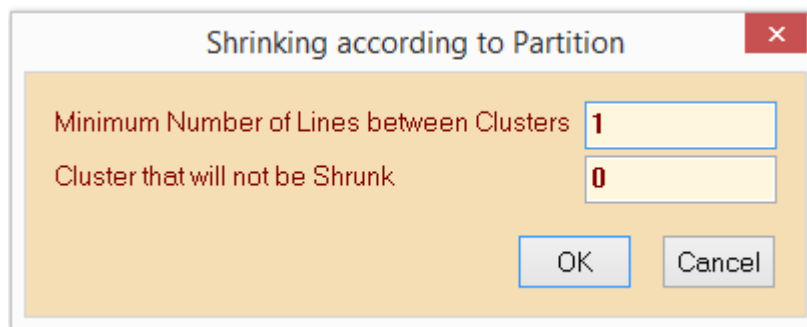
Παράδειγμα θα μπορούσε να αποτελεί και το υποδίκτυο που προκύπτει διαχωρίζοντας τις εισαγωγές και τις εξαγωγές μετάλλου, εντός μιας ηπείρου, από το παγκόσμιο εμπόριο μετάλλου.

12. Μείωση μεγέθους δικτύου

Η μείωση του μεγέθους ενός δικτύου έχει πολλά πλεονεκτήματα κατά τη μελέτη του. Το πιο προφανές, είναι η διευκόλυνση της οπτικής παρατήρησής και της ανάλυσής του. Το Pajek διαθέτει αρκετά εργαλεία για το σκοπό αυτό.

12.1 Μείωση – Συρρίκνωση δικτύου

Η συρρίκνωση (shrinking) πραγματοποιείται επιλέγοντας `Operations > Network + Partition > Shrink Network`. Το Pajek στη συνέχεια ζητά εισαχθούν τα εξής:



Shrinking according to Partition

Minimum Number of Lines between Clusters

Cluster that will not be Shrunk

OK Cancel

Παράδειγμα χρήσης θα μπορούσε να αποτελεί και η συρρίκνωση του δικτύου του παγκόσμιου εμπορίου μετάλλου, από το επίπεδο χωρών στο επίπεδο ηπείρων.

12.2 Αφαίρεση των γραμμών με χαμηλή τιμή

Ένας άλλος τρόπος μείωσης του μεγέθους ενός δικτύου, είναι η αφαίρεση γραμμών σύνδεσης κορυφών οι οποίες έχουν χαμηλή τιμή (ή αλλιώς αξία, ή βάρος, ή κόστος).

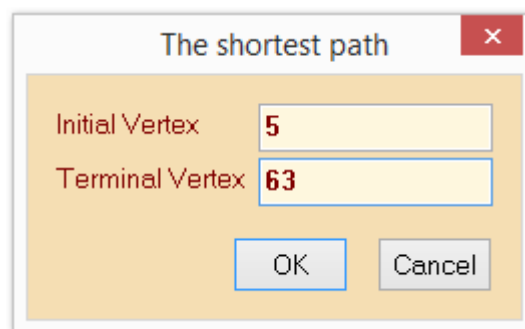
Μπορούμε να αφαιρέσουμε τις γραμμές αυτές επιλέγοντας `Network > Create New Network > Transform > Remove > Lines with Value > lower than`, και δίνοντας εν συνεχεία στο Pajek, το αντίστοιχο αριθμητικό κατώφλι που θα λειτουργεί ως κριτήριο.

13. Εύρεση μονοπατιών

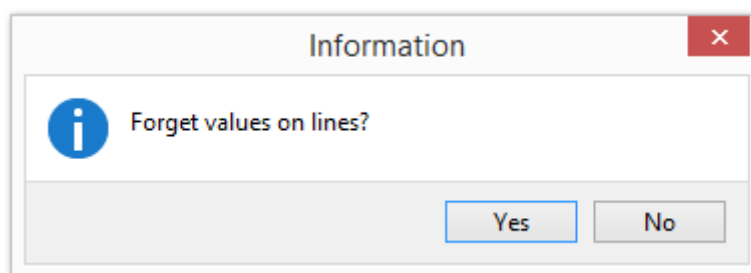
13.1 Εύρεση συντομότερου μονοπατιού

Μπορούμε να βρούμε ένα συντομότερο μονοπάτι μεταξύ δύο κορυφών επιλέγοντας `Network > Create New Network > SubNetwork with Paths > One Shortest Path between Two Vertices`.

Αρχικά μας ζητούνται οι δύο κορυφές εκκίνησης και τερματισμού:



Η επόμενη ερώτηση είναι η παρακάτω:



Απαντούμε καταφατικά αν αναζητούμε το συντομότερο μονοπάτι με βάση τα μήκη, και αρνητικά αν μας ενδιαφέρουν οι τιμές των γραμμών. Τέλος, στην επόμενη ερώτηση “Identify vertices in source network” απαντούμε αρνητικά.

Το αποτέλεσμα είναι ένα νέο (υπό)δίκτυο που περιλαμβάνει τις δύο επιλεγμένες κορυφές με τις γραμμές που τις συνδέουν και μπορεί να απεικονιστεί επιλέγοντας `Layout > Energy > Kamada Kawai > Fix first and last`.

Στο δίκτυο που προκύπτει, οι κορυφές του συντομότερου μονοπατιού παίρνουν αύξουσα αρίθμηση από την εκκίνηση έως τον τερματισμό. Για να φανεί η αντιστοιχία

των κορυφών αυτών με το αρχικό δίκτυο επιλέγουμε `Options > Mark Vertices Using > Labels`. Και για την εύρεση αλυσίδων επιλέγουμε `Network > Create New Network > Transform > Arcs to Edges > All`.

Η εύρεση όλων των συντομότερων μονοπατιών μεταξύ δύο κορυφών είναι εφικτή επιλέγοντας `Network > Create New Network > SubNetwork with Paths > All Shortest Paths between Two Vertices`.

13.2 Εύρεση διαμέτρου

Επιλέγοντας `Network > Create New Network > SubNetwork with Paths > Info on Diameter`, βρίσκουμε τη διάμετρο²² του δικτύου.

Μετά την εκτέλεση της εντολής αυτής, το Rajek επιστρέφει μόνο τις δύο πιο απομακρυσμένες κορυφές. Αν επιθυμούμε να έχουμε και το μονοπάτι ανάμεσά τους, θα πρέπει στο αποτέλεσμα, να εκτελέσουμε επιπλέον και την εντολή εύρεσης του συντομότερου μονοπατιού ανάμεσα σε αυτές τις δύο κορυφές.

13.3 K-neighbors

Η κορυφή j είναι k -neighbor με την κορυφή i εάν το συντομότερο μονοπάτι από την i στην j έχει μήκος k .

Στο Rajek οι αποστάσεις όλων των κορυφών από μία συγκεκριμένη κορυφή υπολογίζονται επιλέγοντας `Network > Create Partition > k-Neighbors > Output`. Στη συνέχεια εισάγουμε την επιλεγμένη κορυφή και την απόσταση που θέλουμε να διερευνήσουμε (εισάγοντας το μηδέν υπολογίζονται όλες οι αποστάσεις).

Το αποτέλεσμα της ενέργειας αυτής, είναι ένα partition – πίνακας $(2 \times n)$, όπου για κάθε κορυφή παίρνουμε και την αντίστοιχη απόστασή της από την επιλεγμένη. Διαφορετικά, θα μπορούσαμε να πούμε ότι πρόκειται για τον υπολογισμό των βημάτων που απαιτούνται για να φτάσουμε από την επιλεγμένη κορυφή σε όλες τις άλλες.

Επιλέγοντας `Network>Create Partition>k-Neighbors>Input`, υπολογίζουμε τα βήματα που απαιτούνται προκειμένου από όλες τις άλλες κορυφές, να φτάσουμε στην επιλεγμένη.

²² http://en.wikipedia.org/wiki/Network_science#Diameter_of_a_network

Τέλος, επιλέγοντας `Network>Create Partition>k-Neighbors>All`, υπολογίζουμε τις αποστάσεις χωρίς να λαμβάνουμε υπόψη τις κατευθύνσεις των γραμμών.

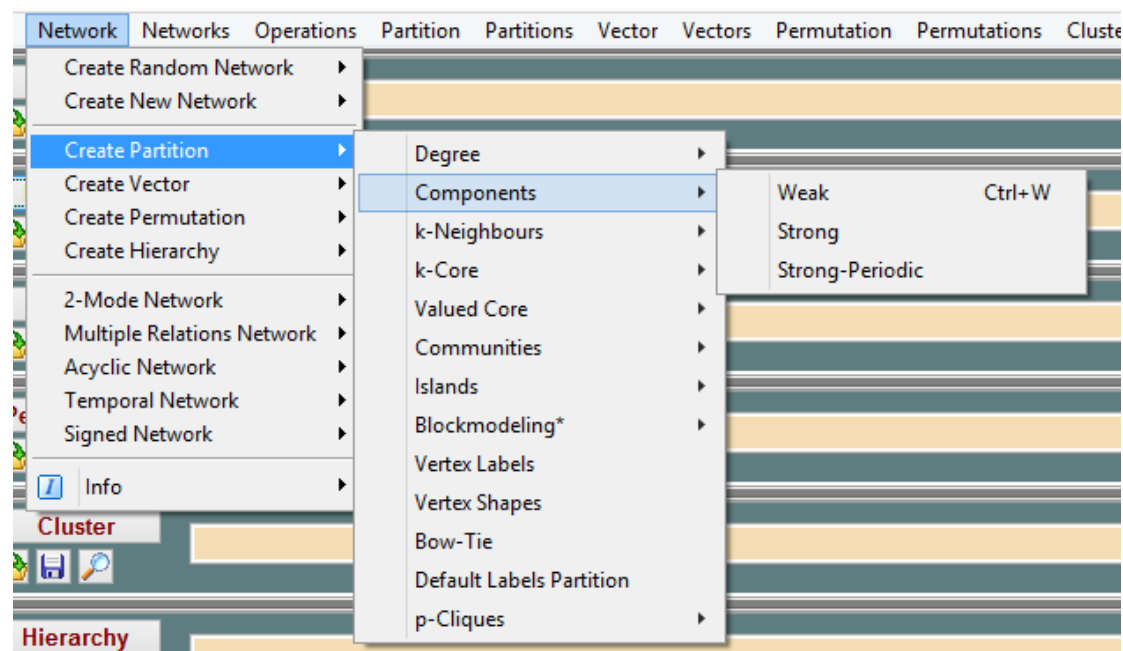
Αφού υπολογιστούν οι αποστάσεις, επιλέγοντας `Partition>Info`, μπορούμε να δούμε τον αριθμό (που θα επιλέξουμε) των κοντινότερων (+) και των πιο απομακρυσμένων (-) κορυφών από την επιλεγμένη κορυφή.

Επίσης, μπορούμε να εξάγουμε από το δίκτυο, ένα υποδίκτυο με κορυφές που βρίσκονται σε συγκεκριμένη απόσταση (την οποία εισάγουμε), από την επιλεγμένη κορυφή επιλέγοντας `Operations > Network + Partition > Extract SubNetwork`.

Το αποτέλεσμα μπορεί να γίνει ορατό με την απεικόνιση του δικτύου.

14. Ισχυρά και Χαλαρά συνδεδεμένα στοιχεία – κορυφές

Τα ισχυρά και τα χαλαρά συνδεδεμένα²³ στοιχεία ενός δικτύου στο Pajek, μπορούν να υπολογιστούν επιλέγοντας:



²³ [http://en.wikipedia.org/wiki/Connectivity_\(graph_theory\)](http://en.wikipedia.org/wiki/Connectivity_(graph_theory))

Το αποτέλεσμα είναι ένα partition στο οποίο οι κορυφές που ανήκουν στο ίδιο στοιχείο (συνιστώσα) έχουν τον ίδιο αριθμό.

15. Αμφισυνδεδεμένα στοιχεία – κόμβοι

Για τον υπολογισμό αμφιστοιχείων (bicomponents) επιλέγουμε `Network > Create New Network > with Bi-Connected Components stored as Relation Numbers`.

Τα αμφισυνδεδεμένα (biconnected²⁴ ή 2-connected) στοιχεία, αποθηκεύονται σε ιεραρχίες. Ένας κόμβος σε μια ιεραρχία αντιπροσωπεύει ένα αμφισυνδεδεμένο στοιχείο.

Σε μια ιεραρχία μπορούμε να πλοηγηθούμε όπως για παράδειγμα και στις δομές καταλόγων ενός παραθυρικού λειτουργικού συστήματος.

Για την εξαγωγή ενός συγκεκριμένου αμφιστοιχείου (ή αμφικορυφής, ή αμφικόμβου), επιλέγουμε `Hierarchy>Extract Cluster`, δίνοντας στη συνέχεια τον αριθμό του κόμβου της ιεραρχίας που μας ενδιαφέρει.

Για την εξαγωγή του υποδικτύου που προκύπτει από την ενέργεια αυτή, επιλέγουμε `Operations>Network+Cluster>Extract SubNetwork`.

Επιπλέον, κατά τον υπολογισμό αμφιστοιχείων, το Rajek επιστρέφει ένα partition με τα σημεία άρθρωσης (articulation points). Οι κορυφές στο πρώτο cluster δεν αποτελούν σημεία άρθρωσης (τα υπόλοιπα αποτελούν). Ο αριθμός του cluster περιλαμβάνει την πληροφορία σχετικά με το σε πόσα κομμάτια θα χωριστεί το (ιεραρχικό - δενδρικό) δίκτυο, όταν αφαιρεθεί ένα συγκεκριμένο σημείο άρθρωσης (ένωσης) από αυτό.

²⁴ http://en.wikipedia.org/wiki/Biconnected_component

16. Άκυκλα δίκτυα

Μερικά παραδείγματα άκυκλων²⁵ δικτύων είναι τα παρακάτω:

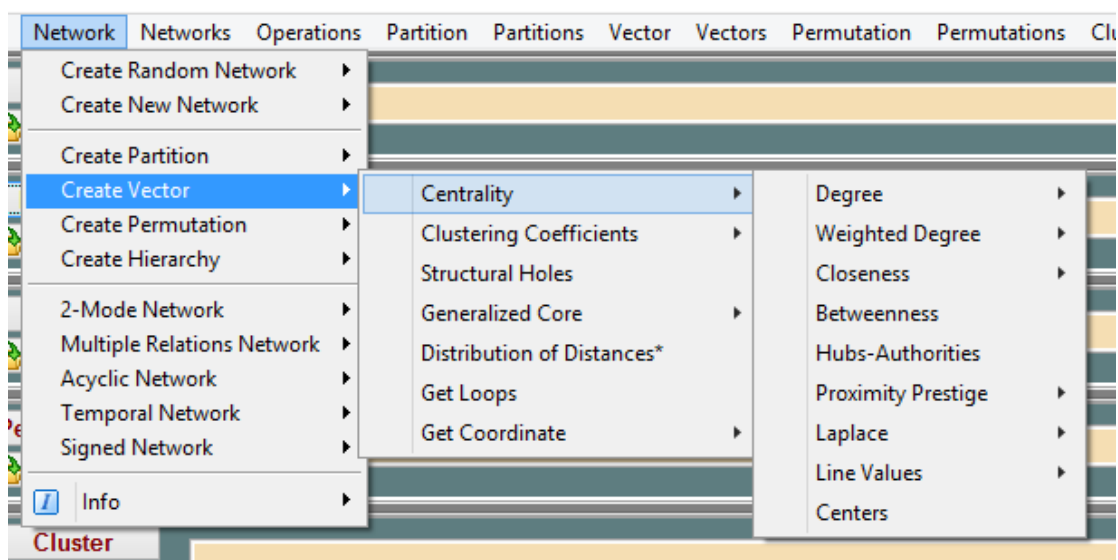
- Δίκτυα παραπομπών (citation networks)
- Γενεαλογίες (genealogies)
- Αλγόριθμοι εύρεσης μονοπατιών (pathfinding algorithms²⁶)

Σε ένα άκυκλο δίκτυο υπάρχουν οι πρώτες κορυφές (επίπεδο 1) στις οποίες δεν φθάνουν γραμμές, οι ενδιάμεσες, και οι τελικές (επίπεδο n) από τις οποίες δε ξεκινά καμία γραμμή σύνδεσης. Επομένως κάθε άκυκλο δίκτυο έχει ένα βάθος, το μέγεθος του οποίου εξαρτάται από το σύνολο των επιπέδων των κορυφών του.

Στο Pajek τα βάθη υπολογίζονται επιλέγοντας Network > Acyclic Network > Depth Partition > Acyclic.

17. Κεντρικότητα

Οι επιλογές υπολογισμού της κεντρικότητας στο Pajek απεικονίζονται στην εικόνα:



²⁵ http://en.wikipedia.org/wiki/Acyclic_graph

²⁶ <http://en.wikipedia.org/wiki/Pathfinding>

17.1 Βαθμός κεντρικότητας

Ο βαθμός κεντρικότητας²⁷ (degree centrality) στο Pajek υπολογίζεται επιλέγοντας `Network > Create Partition > Degree` ή `Network > Create Vector > Centrality > Degree`, και `Input, Output` ή `All`. Με την πρώτη από τις τρεις, παίρνουμε ένα partition ως αποτέλεσμα, ενώ με τη δεύτερη, ένα vector (οι κορυφές με το μεγαλύτερο βαθμό εμφανίζονται επιλέγοντας `Vector>Info`).

Για τον υπολογισμό του βαθμού κεντρικότητας σε δίκτυο με βάρη (weighted degree), επιλέγουμε `Network > Create Vector Centrality > Weighted Degree`.

17.2 Πυκνότητα κεντρικότητας

Η πυκνότητα κεντρικότητας (closeness centrality²⁸) υπολογίζεται επιλέγοντας `Create Vector > Centrality > Closeness`.

Όταν υπολογίζουμε την ενδιαμεσότητα σε δίκτυο με βάρη, πρέπει επιπρόσθετα να επιλέξουμε `Input, Output` ή `All`. Αν το δίκτυο δεν περιλαμβάνει τόξα, μπορούμε να επιλέξουμε, είτε `Input`, είτε `Output`. Το αποτέλεσμα σε αυτή την περίπτωση θα είναι το ίδιο.

17.3 Ενδιαμεσότητα κεντρικότητας

Η ενδιαμεσότητα κεντρικότητας (betweenness centrality²⁹) υπολογίζεται επιλέγοντας `Create Vector > Centrality > Betweenness`.

18. Τριαδική απογραφή

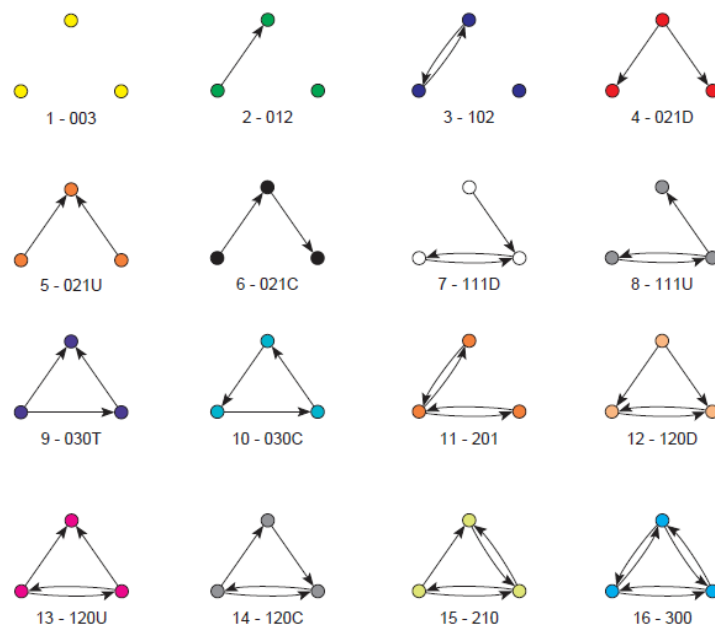
Τριαδική απογραφή (triadic census [4]) ονομάζεται η εύρεση όλων των μοτίβων (patterns ή cliques) τριάδων ενός κατευθυνόμενου γράφου – δικτύου. Στο Pajek η τριαδική απογραφή επιτυγχάνεται επιλέγοντας `Network > Info > Triadic Census`.

²⁷ http://en.wikipedia.org/wiki/Centrality#Degree_centrality

²⁸ http://en.wikipedia.org/wiki/Centrality#Closeness_centrality

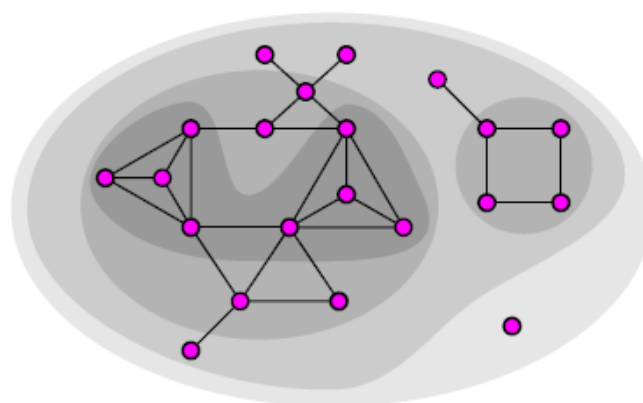
²⁹ http://en.wikipedia.org/wiki/Centrality#Betweenness_centrality

Τα συνήθη τριαδικά μοτίβα είναι 16 και φαίνονται στην εικόνα:



19. Πυρήνες

Ένα υποσύνολο κορυφών ονομάζεται k -core³⁰ (k -πυρήνας) αν κάθε κορυφή του υποσυνόλου είναι συνδεδεμένη με τουλάχιστον k κορυφές του υποσυνόλου αυτού. Στην εικόνα διακρίνονται πυρήνες διαφόρων μεγεθών (0, 1, 2 και 3):



Οι πυρήνες³¹ μπορούν να υπολογιστούν επιλέγοντας Network > Create Partition > K-Core, και επιλέγοντας Input, Output ή All. Το αποτέλεσμα είναι

³⁰ [http://en.wikipedia.org/wiki/Degeneracy_\(graph_theory\)](http://en.wikipedia.org/wiki/Degeneracy_(graph_theory))

³¹ [http://en.wikipedia.org/wiki/Core_\(graph_theory\)](http://en.wikipedia.org/wiki/Core_(graph_theory))

ένα partition για κάθε κορυφή του οποίου δίνεται και ο αριθμός του πυρήνα που ανήκει.

Στις περισσότερες περιπτώσεις ενδιαφερόμαστε μόνο για τον μεγαλύτερο πυρήνα (ή πυρήνες). Το αντίστοιχο υποδίκτυο μπορεί να εξαχθεί επιλέγοντας `Operations > Network + Partition > Extract SubNetwork`. Και εισάγοντας το χαμηλότερο και το υψηλότερο όριο του αριθμού του πυρήνα που μας ενδιαφέρει. Αν θέλουμε να εξάγουμε μόνο τον μέγιστο πυρήνα, εισάγουμε τον ίδιο αριθμό δύο φορές.

Ως παραδείγματα πυρήνων μπορούν να θεωρηθούν και οι πλήρεις γράφοι, οι κύκλοι περιττού μήκους, και κάθε δυο κύκλοι άρτιου μήκους, τις ιδιότητες των οποίων θα εξετάσουμε πολύ συνοπτικά σε επόμενες παραγράφους.

20. Κλίκες

Η κλίκα είναι ένας ειδικός τύπος πυρήνα. Ένα υποσύνολο ενός δικτύου ονομάζεται κλίκα³², εάν καθεμία κορυφή του υποσυνόλου αυτού συνδέεται με την καθεμία από τις άλλες κορυφές του.

Η αναζήτηση κλικών είναι επεξεργαστικά πιο απαιτητική σε σχέση με την αναζήτηση πυρήνων. Γι' αυτό προτιμάται συνήθως η αναζήτηση κλικών μεγέθους 3 ή 4 σε μικρά δίκτυα.

Οι κλίκες μεγέθους 3 (3-cliques) μπορούν να δημιουργήσουν 16 διαφορετικούς τύπους τριάδων (triads) όπως είδαμε και σε προηγούμενη παράγραφο. Κάνοντας αναζήτηση για κλίκες μεγέθους 3, μπορούμε να βρούμε τον αριθμό των τριάδων που υπάρχουν σε ένα δίκτυο.

Αλλά αν θέλουμε να βρούμε όλες τις εμφανίσεις των κλικών μεγέθους 3 (δηλαδή και τύπους τριάδων που εμφανίζονται περισσότερες από μία φορές), θα πρέπει να κάνουμε χρήση μιας γενικότερης διαδικασίας, κατά την οποία γίνεται αναζήτηση τμημάτων (fragments) στο δίκτυο.

³² [http://en.wikipedia.org/wiki/Clique_\(graph_theory\)](http://en.wikipedia.org/wiki/Clique_(graph_theory))

21. Αναζήτηση προτύπων

Ουσιαστικά, η αναζήτηση προτύπων (ή τμημάτων) σε ένα δίκτυο, είναι η διαδικασία κατά την οποία γίνεται μια προσπάθεια εντοπισμού μικρότερων δικτύων σε αυτό. Μία τέτοια αναζήτηση θα μπορούσε να αφορά την εύρεση κλικών που αποτελούνται από τρεις κορυφές (3-cliques).

Στο Pajek, ο καλύτερος τρόπος για να αναζητήσουμε ένα pattern σε ένα δίκτυο, είναι να το σχεδιάσουμε οι ίδιοι ως ένα νέο, ανεξάρτητο και πλήρες κατευθυνόμενο δίκτυο, επιλέγοντας `Network > Create New Network > Complete Network > Directed`.

Στη συνέχεια, για να καθορίσουμε αυτό το νέο μικρό δίκτυο ως fragment, το επιλέγουμε (ή το ανοίγουμε) στο Pajek ώστε να είναι το πρώτο δίκτυο στην αντίστοιχη dropdown λίστα. Επίσης φροντίζουμε το αρχικό δίκτυο να είναι το δεύτερο.

Επειδή ο αλγόριθμος αναζήτησης για fragments κάνει διάκριση μεταξύ γραμμών (edges) και αμφίδρομων τόξων (bidirected arcs), αν στο δίκτυο υπάρχουν γραμμές θα πρέπει πρώτα να μετατραπούν σε αμφίδρομα τόξα επιλέγοντας `Network > Create New Network > Transform > Edges -> Arcs`.

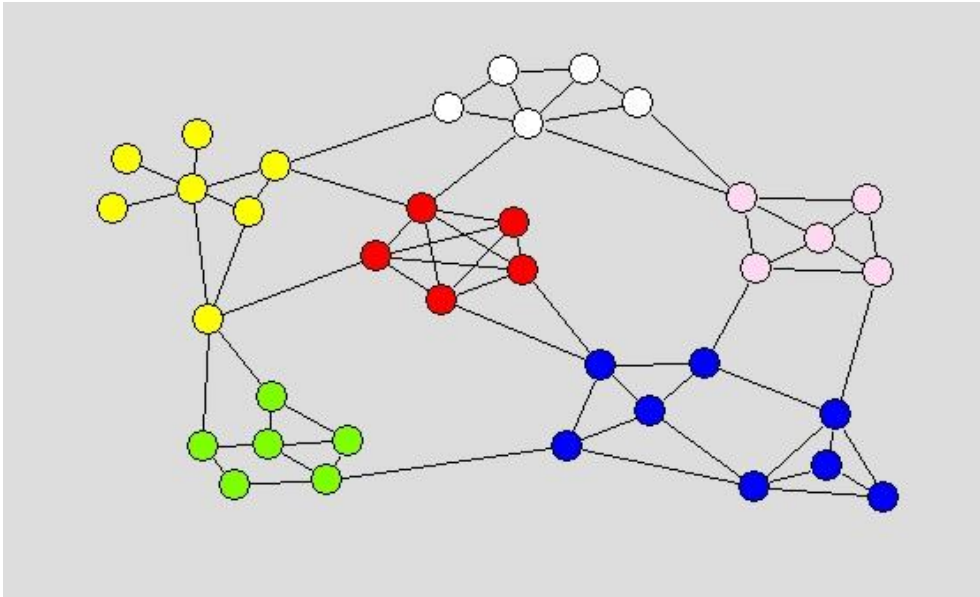
Αφού επιλεγούν το fragment (πρώτο δίκτυο) και το αρχικό (δεύτερο) δίκτυο στο οποίο θέλουμε να εντοπίσουμε τέτοια fragments, επιλέγουμε `Networks > Fragment (First in Second) > Find`.

Ως αποτέλεσμα θα λάβουμε τα παρακάτω τρία αντικείμενα:

- Ένα υποδίκτυο που περιέχει μόνο όλα τα fragments που βρέθηκαν.
- Ένα partition με τιμές από μηδέν και άνω. Η τιμή μηδέν σημαίνει ότι η αντίστοιχη κορυφή δεν ανήκει σε κάποιο fragment. Μία τιμή a , σημαίνει ότι η αντίστοιχη κορυφή ανήκει σε a fragments.
- Μια ιεραρχία με όλα τα fragments

22. Μέθοδοι εντοπισμού δομών κοινοτήτων

Οι δομές κοινοτήτων (community structures³³) είναι πυκνά clusters όπου στο εσωτερικό τους υπάρχουν περισσότερες συνδέσεις απ' ότι στο εξωτερικό τους. Ένα παράδειγμα φαίνεται και στην εικόνα:



Το Pajek διαθέτει δύο μεθόδους εντοπισμού δομών κοινοτήτων. Τη μέθοδο Louvain³⁴ και τη μέθοδο VOS clustering.

23. Δίκτυα 2-mode

Η μόνη διαφορά μεταξύ ενός συνηθισμένου δικτύου κι ενός δικτύου 2-mode³⁵, είναι ότι το δεύτερο αποτελείται από δύο είδη μονάδων - υποσυνόλων (πχ: υπάλληλοι και εταιρείες), τα οποία συσχετίζονται με συνδέσεις.

Η δημιουργία των δύο υποσυνόλων επιτυγχάνεται επιλέγοντας `Network > 2-Mode Network > Partition into 2 Modes`, όπου στις κορυφές του πρώτου υποσυνόλου (υπάλληλοι) δίνεται ο αριθμός 1, και σε εκείνες του δεύτερου (εταιρείες), δίνεται ο αριθμός 2.

³³ http://en.wikipedia.org/wiki/Community_structure

³⁴ <http://mrvar.fdv.uni-lj.si/pajek/community/LouvainVOS.htm>

³⁵ <http://mrvar.fdv.uni-lj.si/sola/info4/uvod/twomode.pdf>

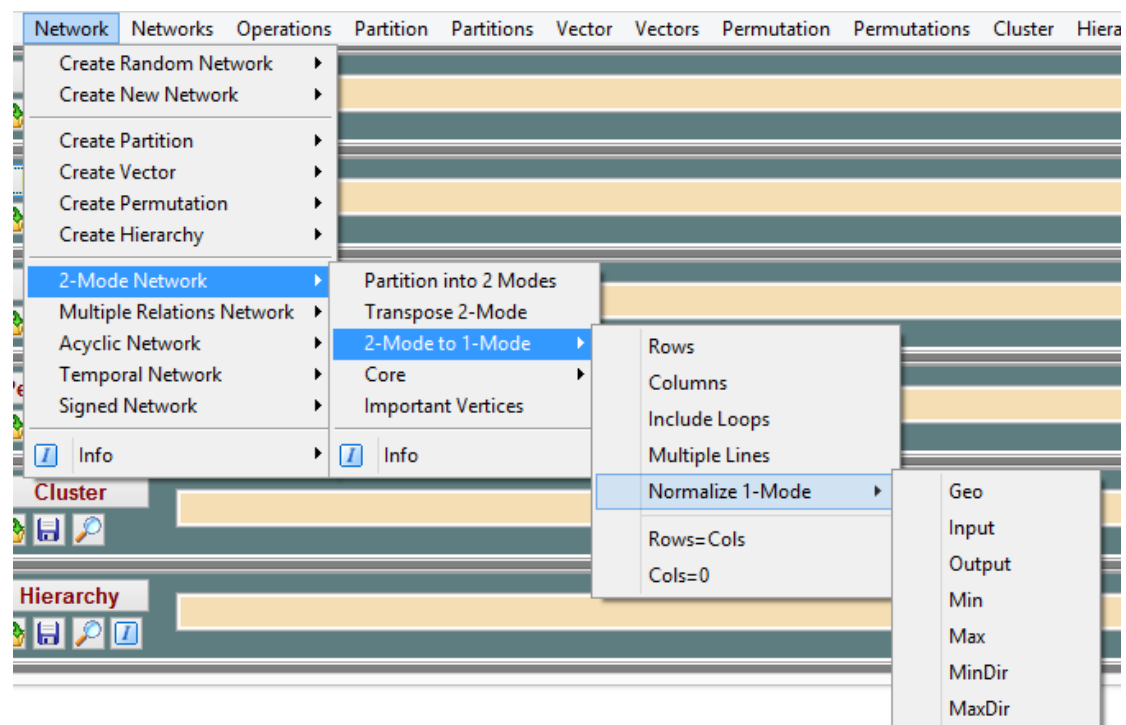
23.1 Μετασχηματισμός 2-mode σε 1-mode

Μπορούμε να μετατρέψουμε ένα 2-mode δίκτυο σε απλό (1-mode) σε δύο βήματα. Στο πρώτο εξάγουμε το πρώτο υποσύνολο των κορυφών επιλέγοντας `Network > 2-Mode Network > 2-Mode to 1-Mode > Rows`, και έπειτα εξάγουμε το δεύτερο υποσύνολο επιλέγοντας `Network > 2-Mode Network > 2-Mode to 1-Mode > Columns`.

Επίσης μπορούμε να δημιουργήσουμε δίκτυο με βρόχους ή χωρίς βρόχους επιλέγοντας `Network > 2-Mode Network > 2-Mode to 1-Mode > Include Loops`.

Για τη δημιουργία δικτύων με γραμμές οι οποίες έχουν κάποια αξία επάνω τους (για παράδειγμα αριθμός υπαλλήλων που δουλεύουν στη ίδια εταιρεία), ή δικτύων με πολλαπλές γραμμές, επιλέγουμε `Network > 2-Mode Network > 2-Mode to 1-Mode > Multiple Lines`.

Άλλες επιλογές μετατροπής φαίνονται και στην εικόνα:



24. Απεικόνιση δικτύου με βάρη

Για την απεικόνιση ενός δικτύου που περιλαμβάνει βρόχους με αξίες (πχ: συνολικός αριθμός υπαλλήλων για κάθε εταιρεία), αποθηκεύουμε πρώτα τις αξίες αυτές σε ένα αρχείο vector επιλέγοντας `Network > Create Vector > Get Loops`, και στη συνέχεια χρησιμοποιούμε το vector αυτό για να καθορίσουμε τα μεγέθη των κορυφών επιλέγοντας `Draw > Network + Vector`.

Αφού δημιουργηθεί δίκτυο με αξιολογημένες γραμμές, μπορούμε να το απεικονίσουμε με πολλούς τρόπους:

- Απεικόνιση πλήρους δικτύου. Αρχικά απεικονίζουμε το δίκτυο χρησιμοποιώντας την μέθοδο Energy έχοντας επιλέξει `Options > Values of Lines > Similarities` (για να εμφανιστούν πιο κοντά οι γραμμές με τις υψηλότερες τιμές). Οι αξίες των γραμμών μπορούν να αναπαρασταθούν επιλέγοντας διαφορετικά πάχη γραμμής (`Options > Lines > Different Widths`) ή/και με χρήση διαβαθμίσεων του γκρι (`Options > Lines > GreyScale`). Αφού εξάγουμε την εικόνα σε μορφή SVG, μπορούμε να προσθέσουμε γραμμές ανάλογα με την αξία τους επιλέγοντας `Export > SVG > LineValues > Nested Classes`.
- Απεικόνιση του πιο σημαντικού μέρους του δικτύου. Εμφανίζονται μόνο οι γραμμές με αξία πάνω από ένα κατώφλι. Για να αποκτήσουμε μια εικόνα της κατανομής των αξιών αυτών, επιλέγουμε `Network > Info > Line Values`. Στη συνέχεια αφαιρούμε τις γραμμές μικρής αξίας θέτοντας το κατώφλι μέσω της εντολής `Network > Create New Network > Transform > Remove > lines with value > lower than`.

25. Νησίδες

Σε ένα δίκτυο όπου δίνονται ορισμένες ιδιότητες (τιμές) των κορυφών ή των γραμμών, μπορούμε να βρούμε νησίδες (islands³⁶). Οι νησίδες αυτές ονομάζονται νησίδες κορυφών αν δίνονται οι αξίες των κορυφών, και νησίδες γραμμών, αν δίνονται οι αξίες των γραμμών.

³⁶ <http://vlado.fmf.uni-lj.si/pub/networks/doc/sunbelt/islands.pdf>

Οι νησίδες μοιάζουν με τις κλίκες, δηλαδή, μπορούμε να τις εντοπίσουμε σε clusters κορυφών με αξίες σύνδεσης μεγαλύτερες εντός του cluster και μικρότερες εκτός.

Για τον εντοπισμό τους θα πρέπει επίσης να επιλέξουμε το μικρότερο και το μεγαλύτερο επιτρεπόμενο μέγεθος της νησίδας. Στο Rajeκ οι γραμμές των νησίδων υπολογίζονται επιλέγοντας `Network > Create Partition > Islands > Lines Weights`.

26. Ιεραρχικό clustering

Η διαδικασία του ιεραρχικού clustering³⁷ αποτελείται από δύο βήματα:

- Υπολογισμός του dissimilarity matrix (πίνακας ανομοιοτήτων)
- Ιεραρχικό clustering με βάση τον dissimilarity matrix που υπολογίστηκε

Πριν εκτελέσουμε την παραπάνω διαδικασία, δημιουργούμε ένα πλήρες (complete³⁸) cluster, επιλέγοντας `Cluster > Create Complete Cluster`. Με αυτό τον τρόπο οι ανομοιοότητες (dissimilarities) θα υπολογιστούν για όλες τις μονάδες.

Στη συνέχεια εκτελούμε την εντολή `Operations > Network + Cluster > Dissimilarity > Network Based` και επιλέγουμε:

- `d1/All` αν θέλουμε να θεωρήσουμε το δίκτυο ως ένα δυαδικό πίνακα.
- `Corrected Euclidian` ή `Corrected Manhattan`, απόσταση για δίκτυα με βάρη - αξίες. Σε αυτή την περίπτωση θα πρέπει να εισαχθεί και η παράμετρος p (0, 1 ή 2). Η παράμετρος αυτή είναι μια οδηγία για τον τρόπο υπολογισμού της διαγώνιου και της απευθείας σύνδεσης μεταξύ δύο μονάδων του δικτύου.

Επιπλέον, μας ζητείται να ονομάσουμε το αρχείο EPS που θα δημιουργηθεί στο οποίο θα αποθηκευτεί το δενδρόγραμμα που θα προκύψει (το αρχείο αυτό ανοίγει με το λογισμικό GSVIEW). Αν σε αυτό το σημείο θέλουμε να υπολογιστεί απλά ο dissimilarity matrix και να μην προχωρήσουμε στο ιεραρχικό clustering, επιλέγουμε `Cancel`.

³⁷ http://en.wikipedia.org/wiki/Hierarchical_clustering

³⁸ http://en.wikipedia.org/wiki/Complete-linkage_clustering

Αν επιτρέψουμε να ολοκληρωθεί το ιεραρχικό clustering, τα αποτελέσματα θα είναι:

- Ο dissimilarity matrix
- Η εικόνα EPS το δένδρογράμματος
- Ένα αρχείο ανακατέματος .per (permutation) σύμφωνα με το δένδρογράμμο το οποίο μπορεί να χρησιμοποιηθεί για την απεικόνιση του πίνακα αναδιάταξης (reordered matrix) στο EPS (File > Network > Export Matrix to EPS > Using Permutation).
- Ιεραρχία που αναπαριστά το ιεραρχικό clustering

Στο Pajek χρησιμοποιείται από προεπιλογή η μέθοδος – κριτήριο Ward³⁹. Για να την αλλάξουμε επιλέγουμε Network > Create Hierarchy > Clustering* > Options.

27. Μοντελοποίηση σε μπλοκ

Η μοντελοποίηση σε μπλοκ ή αλλιώς blockmodeling στο Pajek [5], μπορεί να πραγματοποιηθεί με δύο τρόπους:

- Ξεκινώντας με τυχαίο partition σε δοσμένο αριθμό clusters επιλέγοντας Network > Create Partition > Blockmodeling* > Random Start.
- Βελτιστοποιώντας το δοσμένο partition (πχ: partition που προέκυψε από ιεραρχικό clustering), επιλέγοντας Network > Create Partition > Blockmodeling* > Optimize Partition.

Αφού επιλέξουμε τον τρόπο που θα πραγματοποιηθεί το blockmodeling, επιλέγουμε:

- Τον τύπο ισοδυναμίας (type of equivalence – Structural ή Regular). Μπορούμε επίσης να καθορίσουμε τους δικούς μας τύπους blocks επιλέγοντας 3-Define.
- Τον αριθμό των επαναλήψεων
- Τον αριθμό των clusters

Αφού εισάγουμε όλες τις απαιτούμενες ρυθμίσεις επιλέγουμε Run.

³⁹ http://en.wikipedia.org/wiki/Ward's_method

Στην περίπτωση που έχουμε επιλέξει την βελτιστοποίηση (δεύτερος τρόπος), δεν μας ζητείται ο αριθμός των clusters και ο αριθμός των επαναλήψεων.

Το αποτέλεσμα της διαδικασίας blockmodeling στο Pajek, είναι όλα τα διαφορετικά partitions με την χαμηλότερη τιμή της συνάρτησης του κριτηρίου (ένα ή περισσότερα partitions).

Τέλος, αν θέλουμε να εξάγουμε τον αναδιαταγμένο πίνακα (με γραμμές μεταξύ των clusters), θα πρέπει πρώτα να δημιουργήσουμε ένα permutation επιλέγοντας: `Partition > Make Permutation.`

28. Χρονικώς μεταβαλλόμενα δίκτυα

Τα δίκτυα μεταβάλλονται με το πέρασμα του χρόνου. Στα χρονικώς μεταβαλλόμενα (temporal networks⁴⁰) ή δυναμικά δίκτυα, οι κορυφές και οι γραμμές δεν είναι απαραίτητως παρούσες σε κάθε στιγμή.

Στο αρχείο εισόδου που δημιουργούμε για το Pajek μπορούμε για ένα δυναμικό δίκτυο να χρησιμοποιούμε τις παρακάτω μορφές χρονικής υπόστασης:

- t_i – μια μόνο χρονική στιγμή
- t_i-t_j – ένα χρονικό διάστημα
- t_i-* – από μία χρονική στιγμή και μετά

Για παράδειγμα, αν είχαμε ένα δυναμικό δίκτυο με κορυφές a, b, c οι οποίες εμφανίζονται:

- **a** – Σε δύο χρονικά διαστήματα. Από το 5 δευτερόλεπτο έως το 10 και από το 12 έως το 14.
- **b** – Από το διάστημα 1 έως 3 και έπειτα μόνο στο 7
- **c** – Από το 4 και μετά

⁴⁰ <http://vlado.fmf.uni-lj.si/pub/networks/doc/seminar/SFI-SAS04tn.pdf>

Και γραμμές μεταξύ των a-b, και των a-c οι οποίες εμφανίζονται:

- **a-b** – Μόνο τη χρονική στιγμή 7
- **a-c** – Στο χρονικό διάστημα 6 έως 8

Το αρχείο δικτύου `.net` θα έπρεπε να συνταχθεί ως εξής:

```
*Vertices 3
1 "a" [5-10,12-14]
2 "b" [1-3,7]
3 "c" [4-*]
*Edges
1 2 1 [7]
1 3 1 [6-8]
```

Αφού ανοίξουμε το αρχείο `.net` μέσω του Rajek, επιλέγουμε `Network > Temporal Network > Generate in Time`.

Το Rajek υποστηρίζει και ένα σύνολο χρονικών συμβάντων για μεταβαλλόμενα δίκτυα πραγματικού χρόνου. Τα συμβάντα αυτά περιγράφονται στον πίνακα που ακολουθεί:

Event	Explanation
TI <i>t</i>	initial events – following events happen when time point <i>t</i> starts
TE <i>t</i>	end events – following events happen when time point <i>t</i> is finished
AV <i>vn</i> <i>s</i>	add vertex <i>v</i> with label <i>n</i> and properties <i>s</i>
HV <i>v</i>	hide vertex <i>v</i>
SV <i>v</i>	show vertex <i>v</i>
DV <i>v</i>	delete vertex <i>v</i>
AA <i>uvs</i>	add arc (<i>u,v</i>) with properties <i>s</i>
HA <i>uv</i>	hide arc (<i>u,v</i>)
SA <i>uv</i>	show arc (<i>u,v</i>)
DA <i>uv</i>	delete arc (<i>u,v</i>)
AE <i>uvs</i>	add edge (<i>u:v</i>) with properties <i>s</i>
HE <i>uv</i>	hide edge (<i>u:v</i>)
SE <i>uv</i>	show edge (<i>u:v</i>)
DE <i>uv</i>	delete edge (<i>u:v</i>)
CV <i>vs</i>	change vertex property – change property of vertex <i>v</i> to <i>s</i>
CA <i>uvs</i>	change arc property – change property of arc (<i>u,v</i>) to <i>s</i>
CE <i>uvs</i>	change edge property – change property of edge (<i>u:v</i>) to <i>s</i>
CT <i>uv</i>	change type – change (un)directedness of line (<i>u,v</i>)
CD <i>uv</i>	change direction of arc (<i>u,v</i>)
PE <i>uvs</i>	replace pair of arcs (<i>u,v</i>) and (<i>v,u</i>) by single edge (<i>u:v</i>) with properties <i>s</i>
AP <i>uvs</i>	add pair of arcs (<i>u,v</i>) and (<i>v,u</i>) with properties <i>s</i>
DP <i>uv</i>	delete pair of arcs (<i>u,v</i>) and (<i>v,u</i>)
EP <i>uvs</i>	replace edge (<i>u:v</i>) by pair of arcs (<i>u,v</i>) and (<i>v,u</i>) with properties <i>s</i>

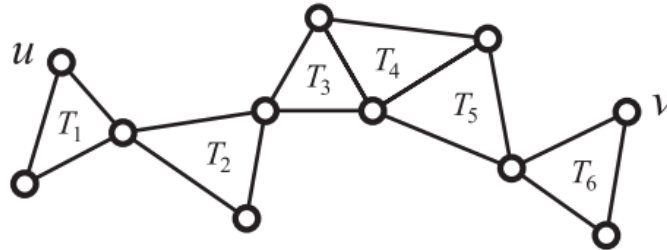
Το παρακάτω κείμενο είναι ένα παράδειγμα αρχείου δικτύου με χρονικά συμβάντα:

```
*Vertices 3
*Events
TI 1
AV 2 "b"
TE 3
HV 2
TI 4
AV 3 "c"
TI 5
AV 1 "a"
TI 6
AE 1 3 1
TI 7
SV 2
AE 1 2 1
TE 7
DE 1 2
DV 2
TE 8
DE 1 3
TE 10
HV 1
TI 12
SV 1
TE 14
DV 1
```

29. Συνδεσιμότητα μικρών κύκλων

Η συνδεσιμότητα μικρών “κύκλων” (short cycle connectivity [6]), είναι μια γενίκευση της συνήθους σύνδεσης. Αντί ενός μονοπατιού (ακολουθία γραμμών σύνδεσης), δύο κορυφές πρέπει να συνδέονται από μια ακολουθία συνδέσεων που σχηματίζουν μικρούς κύκλους (πχ: τριάδες) όπου οι δύο γειτονικοί κύκλοι θα πρέπει να έχουν τουλάχιστον μια κοινή κορυφή μεταξύ τους. Αν αυτό συμβαίνει σε όλους τους παρακείμενους κύκλους, τότε μπορούμε να μιλάμε για συνδεσιμότητα μικρών κύκλων ή αλλιώς, δακτυλίων (rings).

Στην εικόνα βλέπουμε ένα παράδειγμα συνδεσιμότητας μικρών κύκλων (T_1 έως T_6) οι οποίοι σχηματίζουν τριγωνική αλυσίδα (και μονοπάτι $u - v$):



Μέσω του Rajek μπορούμε να δημιουργήσουμε δίκτυα μικρών κύκλων που οι δακτύλιοί τους να είναι τριγωνικοί (3-Rings) επιλέγοντας `Network > Create New Network > with Ring Counts stored as Line Values > 3-Rings`, ή τετραγωνικοί (4-Rings), επιλέγοντας `Network > Create New Network > with Ring Counts stored as Line Values > 4-Rings`.

Οι δακτύλιοι μπορεί να είναι κατευθυνόμενοι ή μη κατευθυνόμενοι.

30. Εξαγωγή δεδομένων

Μέσω του μενού Tools το Rajek διαθέτει τη δυνατότητα εξαγωγής δεδομένων σε διάφορες μορφές (τύπους αρχείων) για την περαιτέρω ανάλυσή τους από άλλα λογισμικά.

30.1 Εξαγωγή στην R

Τα δίκτυα και τα διανύσματα μπορούν να εξαχθούν από το Rajek για το στατιστικό πακέτο R⁴¹ (και το αντίστροφο). Με τη γλώσσα R μπορούμε να πραγματοποιήσουμε στατιστική ανάλυση στα δεδομένα αυτά.

Για παράδειγμα, μπορούμε να εξάγουμε διανύσματα τα οποία περιλαμβάνουν τα δεδομένα για την πυκνότητα (closeness) και την ενδιαμεσότητα (betweenness) της κεντρικότητας (centrality) ενός δικτύου, στην R, και να εκτελέσουμε τις παρακάτω ενέργειες:

⁴¹ <http://cran.r-project.org>

- Σχεδίαση διαγράμματος διασποράς (scatterplot)
- Σχεδίαση ιστογράμματος
- Υπολογισμός συσχέτισης μεταξύ των δύο μετρήσεων

Για να συνεργαστεί το Pajek με την R θα πρέπει να εντοπιστεί το εκτελέσιμο της R που δημιουργεί τα γραφικά επιλέγοντας `Tools > R > Locate R` (εντοπισμός του αρχείου `C:\Program Files\...\bin\Rgui.exe` ή του αρχείου `Rterm.exe`). Αφού γίνει αυτό τα δεδομένα μπορούν να σταλούν επιλέγοντας `Tools > R > Send to R`.

30.2 Εξαγωγή σε SPSS

Προκειμένου να συνδέσουμε το Pajek με τη σουίτα SPSS, επιλέγουμε `Tools > SPSS > Locate SPSS` και εντοπίζουμε το αρχείο `runsyntx.exe` που βρίσκεται συνήθως στον κατάλογο `C:\Program Files\SPSS`.

Στη συνέχεια μπορούμε να στείλουμε δίκτυα, partitions και vectors, επιλέγοντας `Tools > SPSS > Send to SPSS`.

30.3 Εξαγωγή σε αρχείο χωρισμένο με tabs

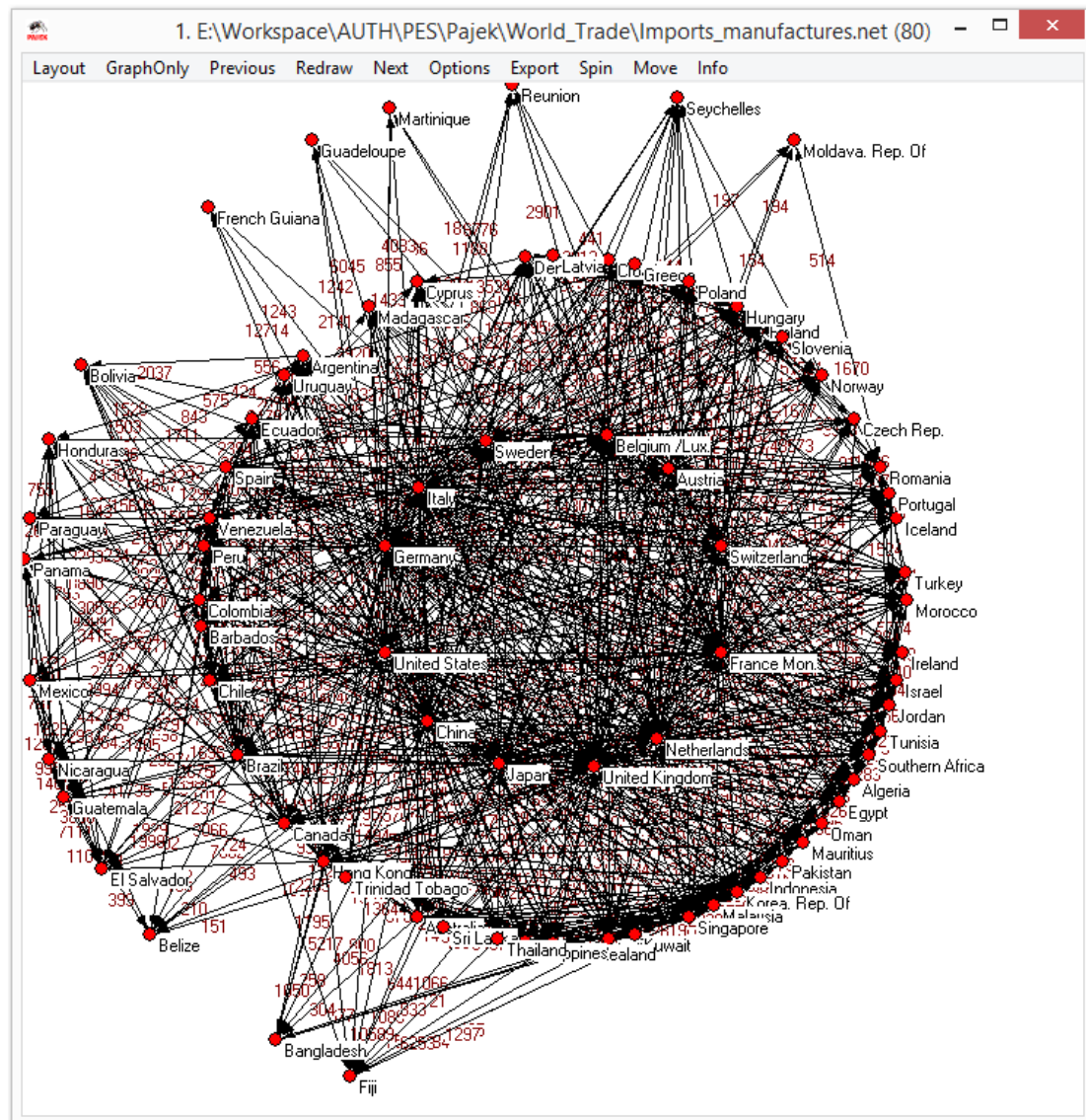
Εξαγωγή partitions και vectors σε αρχείο το οποίο μπορεί στη συνέχεια να διαβαστεί από κάποιο άλλο στατιστικό πακέτο (πχ: Stata), ή από το Excel.

31. Σενάριο χρήσης με πραγματικά δεδομένα

Σε αυτή την παράγραφο θα εξετάσουμε ένα σενάριο χρήσης του Pajek σε πραγματικά δεδομένα⁴², που αφορούν το παγκόσμιο εμπόριο μετάλλου (το έτος 1994), και το οποίο χρησιμοποιήθηκε ως παράδειγμα και σε προηγούμενη παράγραφο.

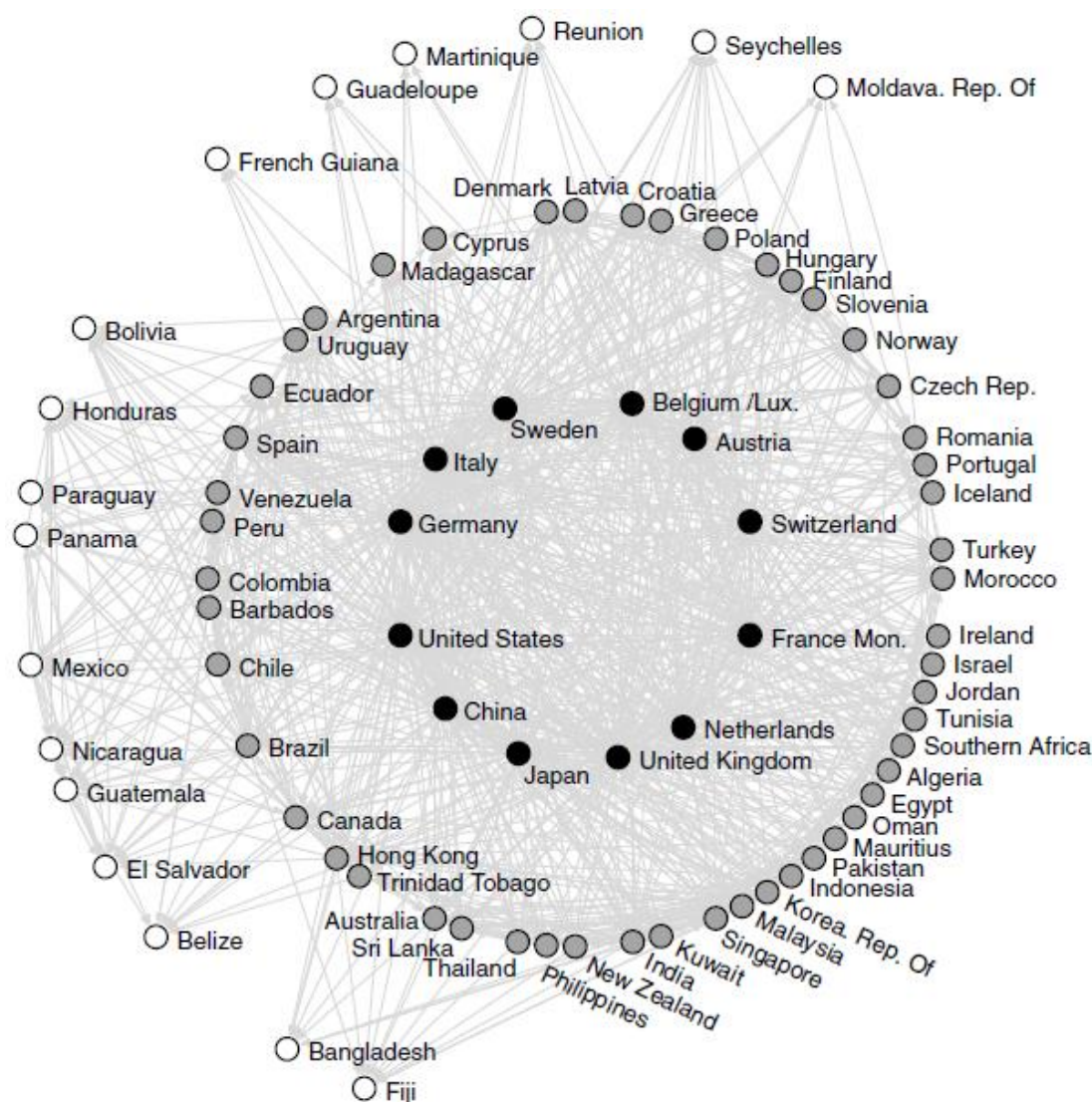
Τα τόξα του δικτύου (`Imports_manufactures.net`) αντιπροσωπεύουν τις εισαγωγές μετάλλου που κάνει μία χώρα από μία άλλη. Αφού εισάγουμε το αρχείο του δικτύου στο Pajek (`File>Network>Read>Imports_manufactures.net`), επιλέγουμε `Draw>Network` και προκύπτει ο παρακάτω γράφος:

⁴² <http://vlado.fmf.uni-lj.si/pub/networks/data/esna/metalWT.htm>



Το μέγεθος των κόκκινων κουκίδων εισάγεται από εμάς μέσω του μενού, επιλέγοντας Options>Size>Of Vertices. Τα βάρη των συνδέσεων ενεργοποιούνται και απενεργοποιούνται μέσω του Options>Lines>Mark Lines.

Μια πιο ευκρινής απεικόνιση του δικτύου:

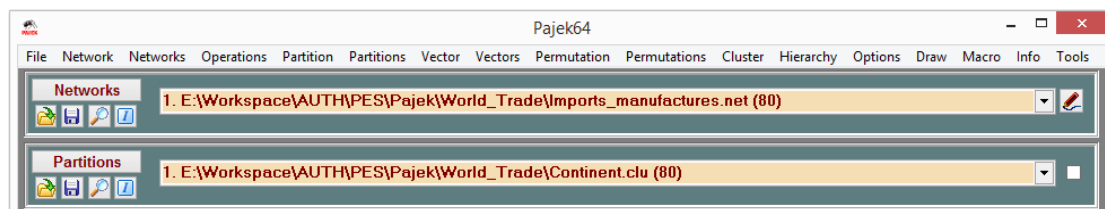


Εύκολα παρατηρούμε από την εικόνα ότι σχηματίζονται τρεις νοητοί «ομόκεντροι» κύκλοι. Τρεις ομάδες χωρών. Η κεντρική ομάδα αποτελεί τον πυρήνα⁴³ του εμπορίου του μετάλλου, και καθώς κινούμαστε προς τα έξω, συναντάμε τις χώρες που παίζουν δευτερεύων και τριτεύων ρόλο, αντιστοίχως. Αυτό είναι το πρώτο και το ευκολότερο συμπέρασμα που μπορούμε να βγάλουμε.

⁴³ Με τον όρο «πυρήνας» εννοούμε τις χώρες που παίρνουν τις πρώτες ύλες από τις (βλ βιβλίο)+++++

Είναι προφανές ότι ο αριθμός των γραμμών είναι αρκετά μεγάλος και δυσχεραίνει την περεταίρω οπτική ανάλυση του δικτύου μας. Για το λόγο αυτό θα πρέπει να επιλεγεί κάποιο κριτήριο - εργαλείο απλοποίησης ή σήμανσης που θα μας επιτρέψει να βγάλουμε επιπλέον συμπεράσματα.

Ένας τρόπος για να επιτευχθεί κάτι τέτοιο, είναι η διαμέριση (partitioning) του δικτύου των χωρών σε ηπείρους. Αυτό επιτυγχάνεται μέσω της εισαγωγής του αντίστοιχου αρχείου (File>Partition>Read>Continent.clu). Μετά την ολοκλήρωση της εισαγωγής, θα πρέπει να επιβεβαιώσουμε οπτικά ότι το αρχείο διαμέρισης .clu έχει τον ίδιο αριθμό κόμβων με το αρχείο του δικτύου .net (που στην περίπτωση μας είναι 80):

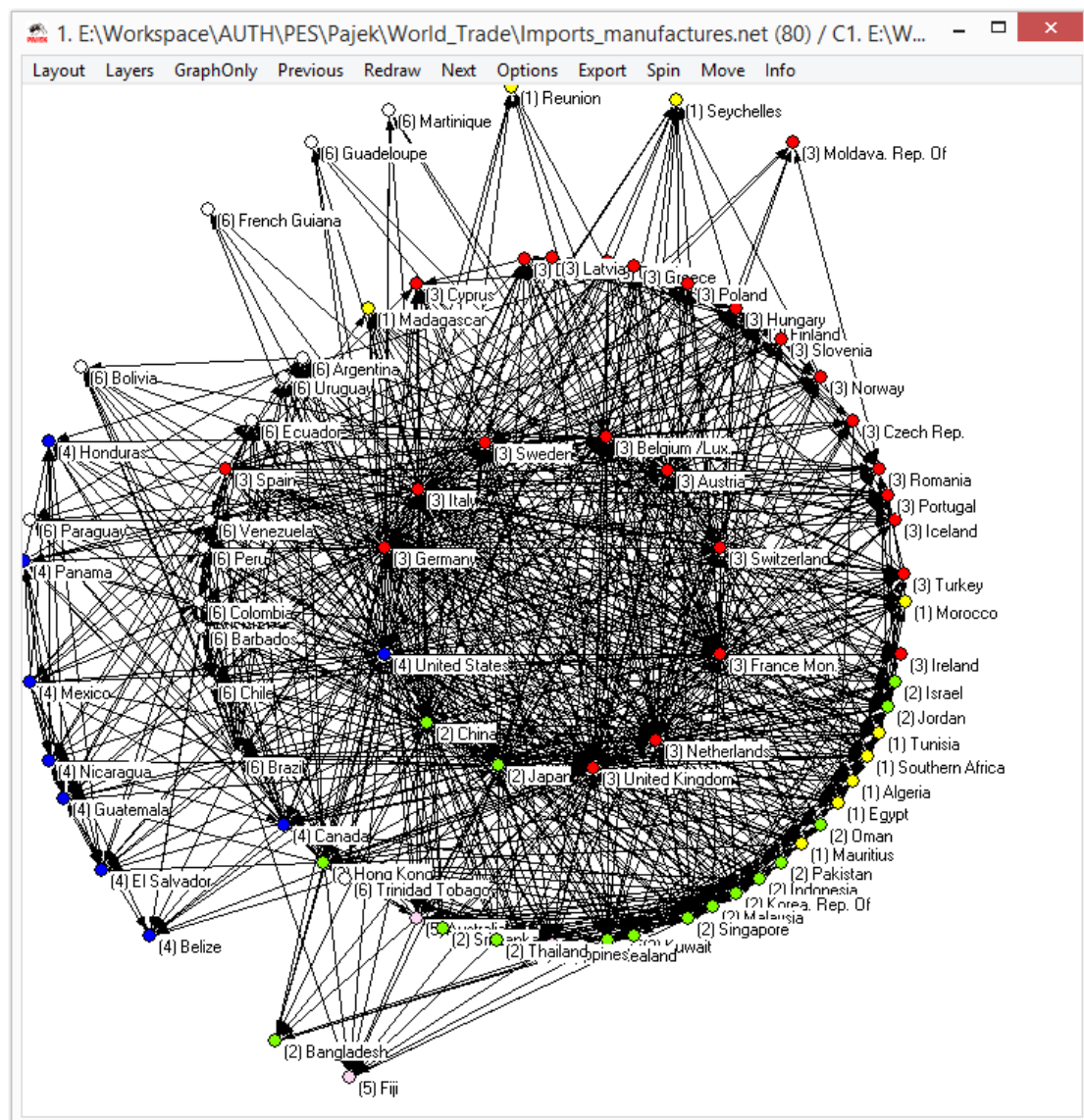


Κάνοντας κλικ στο εικονίδιο του μεγεθυντικού φακού (View/Edit Partition) κάτω από το κουμπί Partition, μπορούμε να δούμε την διαμέριση βάσει των ηπείρων:

Editing Partition: 1. E:\Workspace\AUTH\PES\Pajek\World_Trade\Continent.clu (80)		
Redisplay		
Vertex	Val	Label
1	1	Algeria
2	6	Argentina
3	5	Australia
4	3	Austria
5	6	Barbados
6	2	Bangladesh
7	3	Belgium /Lux.
8	4	Belize
9	6	Bolivia
10	6	Brazil
11	4	Canada
12	6	Chile
13	2	China
14	6	Colombia
15	3	Croatia
16	3	Cyprus
17	3	Czech Rep.
18	3	Denmark
19	6	Ecuador
20	1	Egypt
21	4	El Salvador
22	5	Fiji
23	3	Finland
24	3	France Mon.
25	6	French Guiana
26	3	Germany
27	3	Greece
28	6	Guadeloupe

Η στήλη Vertex συμβολίζει τις κορυφές του δικτύου και παίρνει τιμές 1 – 80 (αριθμός συμμετεχουσών χωρών). Η στήλη με το όνομα Val παίρνει τιμές 1 – 6 και συμβολίζει τις 6 από τις 7 ηπείρους, μιας και δεν υπάρχουν χώρες που να αντιπροσωπεύουν την έβδομη (πιθανότατα επειδή δεν συμμετέχουν στο δίκτυο εισαγωγών μετallου). Η Ελλάδα ανήκει στην Ευρώπη, επομένως εύκολα βγαίνει το συμπέρασμα ότι όποια χώρα (Label) έχει την τιμή (Val) 3, ανήκει στην Ευρώπη.

Επιλέγοντας `Draw>Network + First Partition`, παίρνουμε μία λίγο πιο κατατοπιστική απεικόνιση του δικτύου, μιας και το Rajeκ εφαρμόζει αυτόματα διαφορετικό χρώμα για καθεμιά από τις 6 ηπείρους:

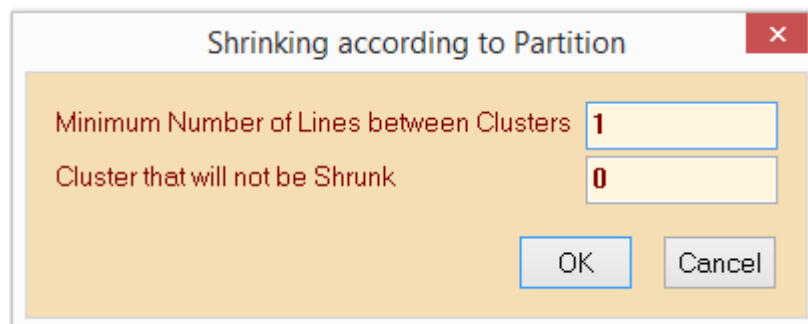


Η εμφάνιση του αριθμού της ηπείρου εφαρμόζεται επιλέγοντας Options>Mark Vertices Using>Partition Clusters.

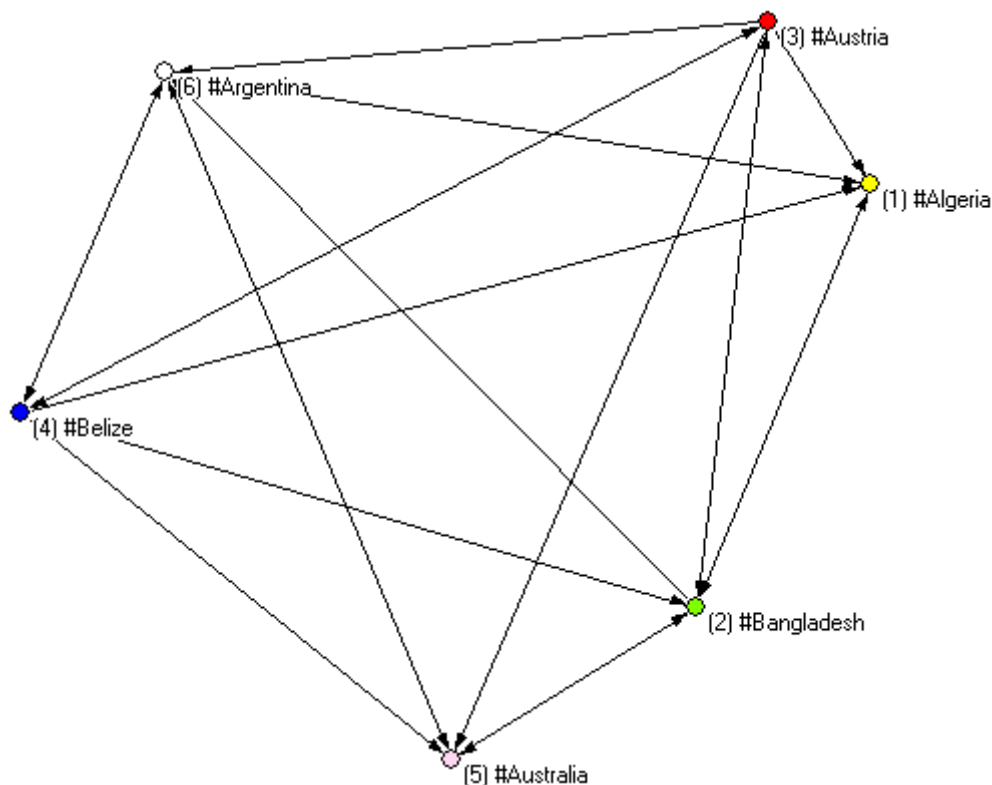
Με μια σύντομη παρατήρηση συμπεραίνουμε ότι κάποιες χώρες της Ευρώπης (κόκκινο χρώμα) ανήκουν στον πυρήνα, ενώ κάποιες άλλες όχι. Οι χώρες εκτός πυρήνα είναι σίγουρα κατώτερες οικονομικά από τις χώρες εντός.

Παρ' όλα αυτά, το δίκτυό μας συνεχίζει να περιέχει αρκετή ακόμα πληροφορία που δυσχεραίνει την παρατήρησή του. Μία από τις πρώτες επεμβάσεις που μπορούν να γίνουν, είναι η συρρίκνωση του δικτύου. Αυτό μπορεί να γίνει αφού πρώτα θέσουμε ένα λογικό στόχο, όπως είναι για παράδειγμα η αναπαράσταση της συσχέτισης στο εμπόριο μετάλλου μεταξύ των 6 ηπείρων που συμμετέχουν.

Επιλέγοντας από το κεντρικό μενού του Pajek, Operations>Network+Partition>Shrink Network, εμφανίζεται το παρακάτω παράθυρο διαλόγου, στο οποίο ζητείται να εισαχθεί ο ελάχιστος αριθμός συνδέσεων μεταξύ των clusters του partition και ο αριθμός του cluster που δεν επιθυμούμε να συρρικνωθεί:



Ως αποτέλεσμα παίρνουμε κάτι πολύ πιο ευανάγνωστο αλλά με ορισμένα σημεία στα οποία απαιτείται από εμάς χειροκίνητη διόρθωση:



Πρέπει οι κορυφές του νέου δικτύου να αποτελούνται μόνο από ονόματα ηπείρων. Επομένως θα πρέπει να γίνει ανάλογη επεξεργασία στη νέα αυτή διαμέριση που έχει προκύψει μετά τη συρρίκνωση:

Editing Partition: 2. Shrinking C1 [0] (6)

Redisplay

Vertex	Val	Label
1	1	#Algeria
2	6	#Argentina
3	5	#Australia
4	3	#Austria
5	2	#Bangladesh
6	4	#Belize

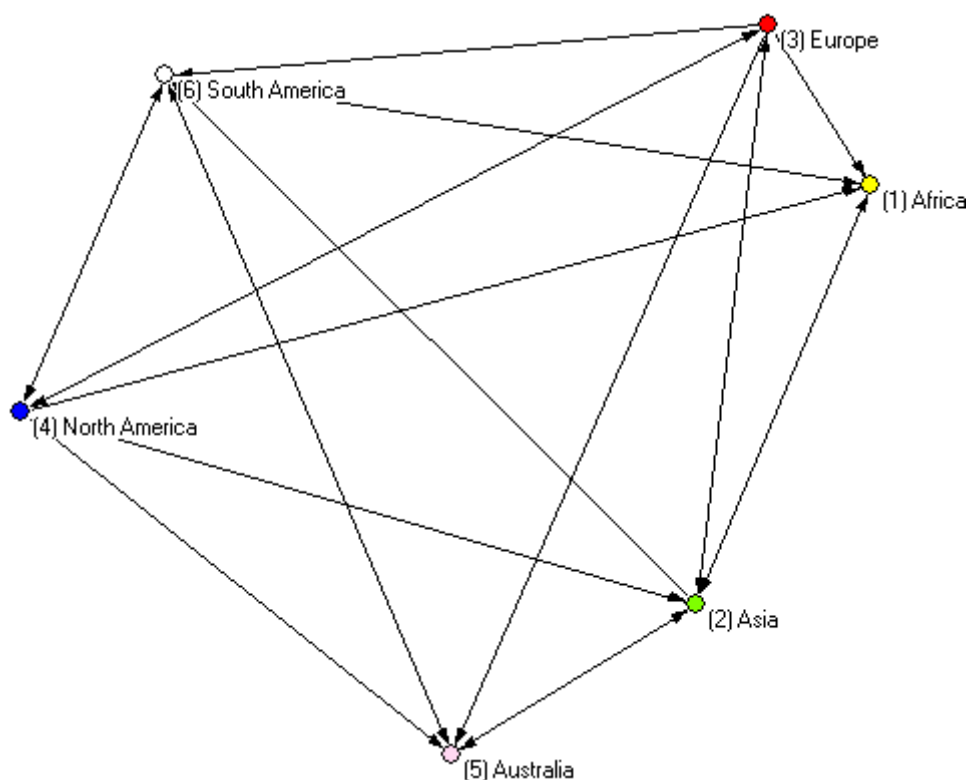
Η αλλαγή των ονομάτων γίνεται στα πεδία της στήλης Label:

Editing Partition: 2. Shrinking C1 [0] (6)

Redisplay

Vertex	Val	Label
1	1	Africa
2	6	South America
3	5	Australia
4	3	Europe
5	2	Asia
6	4	North America

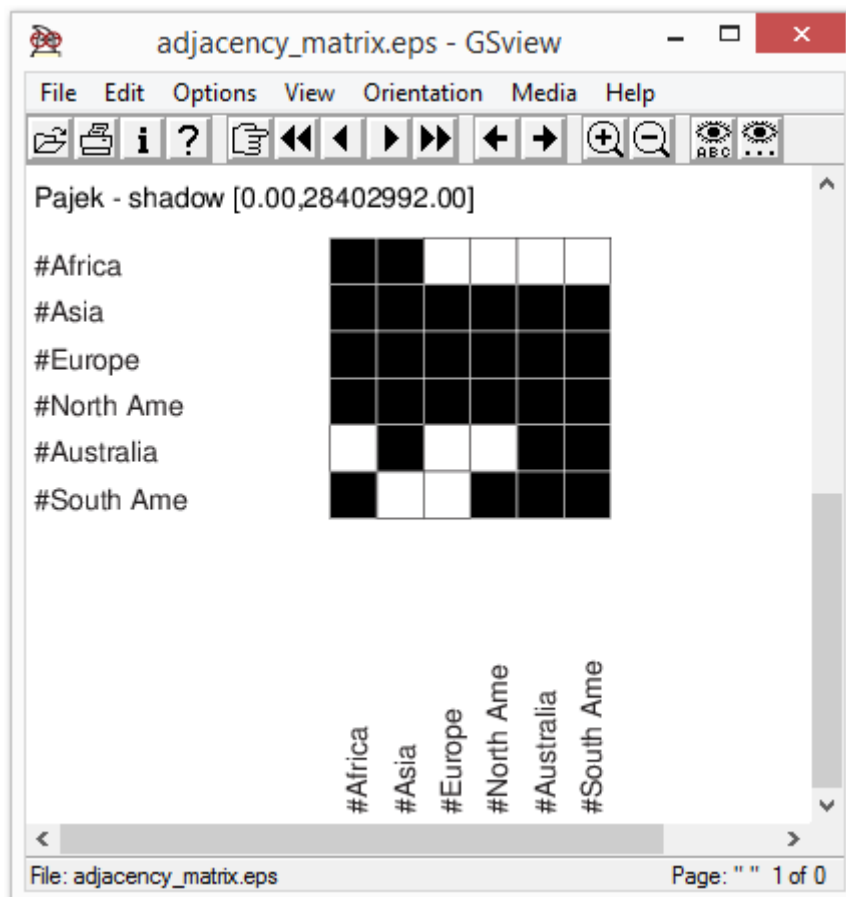
Και το νέο δίκτυο αποτελείται πλέον, μόνο από τα ονόματα των 6 ηπείρων:



Αν και ο παραπάνω γράφος είναι αρκετά απλός, είναι λίγο δύσκολο κάποιος να μπορέσει να καταλάβει αν είναι πλήρης. Η χρήση ενός πίνακα γειτνίασης (adjacency matrix) μπορεί να προσφέρει μια καλύτερη εικόνα. Μέσα από το κεντρικό μενού του Rajek επιλέγουμε File>Network>Export as Matrix to EPS>Using Partition>Structural. Στα παράθυρα διαλόγου που θα προκύψουν αφήνουμε

τις προεπιλεγμένες τιμές και σώζουμε το αρχείο που περιλαμβάνει τον πίνακα γειτνίασης στο αρχείο `adjacency_matrix.eps`.

Στη συνέχεια, αν στο σύστημά μας υπάρχει εγκατεστημένο το απαιτούμενο λογισμικό⁴⁴, μπορούμε να προβάλουμε τον πίνακα γειτνίασης κάνοντας διπλό κλικ στο αντίστοιχο αρχείο:



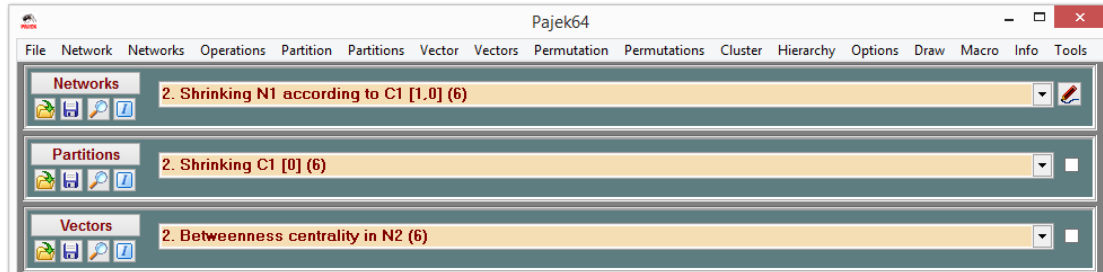
Είναι προφανές με την πρώτη ματιά, ότι ο γράφος μας δεν είναι πλήρως συνδεδεμένος και ότι ο παραπάνω πίνακας δεν είναι συμμετρικός. Δηλαδή, το εμπόριο μετάλλου δεν πραγματοποιείται από, και προς, όλες τις ηπείρους. Για παράδειγμα, η Αυστραλία δε συναλλάσσεται με την Αφρική.

Σε αυτό το σημείο έχει επίσης νόημα, να υπολογίσουμε το μέγεθος της ενδιαμεσότητας (betweenness⁴⁵) δημιουργώντας ένα διάγραμμα μέσω του Pajek. Από το κεντρικό

⁴⁴ Ghostscript και GSview: <http://pages.cs.wisc.edu/~ghost/>

⁴⁵ http://en.wikipedia.org/wiki/Betweenness_centrality

μενού επιλέγουμε `Network>Create Vector>Centrality>Betweenness`. Το νέο διάνυσμα εμφανίζεται στο αντίστοιχο πεδίο:



Κάνοντας κλικ στο εικονίδιο με τον μεγεθυντικό φακό που βρίσκεται κάτω από το κουμπί `Vectors`, εμφανίζεται ο πίνακας με τις τιμές που αφορούν την ενδιαμεσότητα της κάθε Ηπείρου:

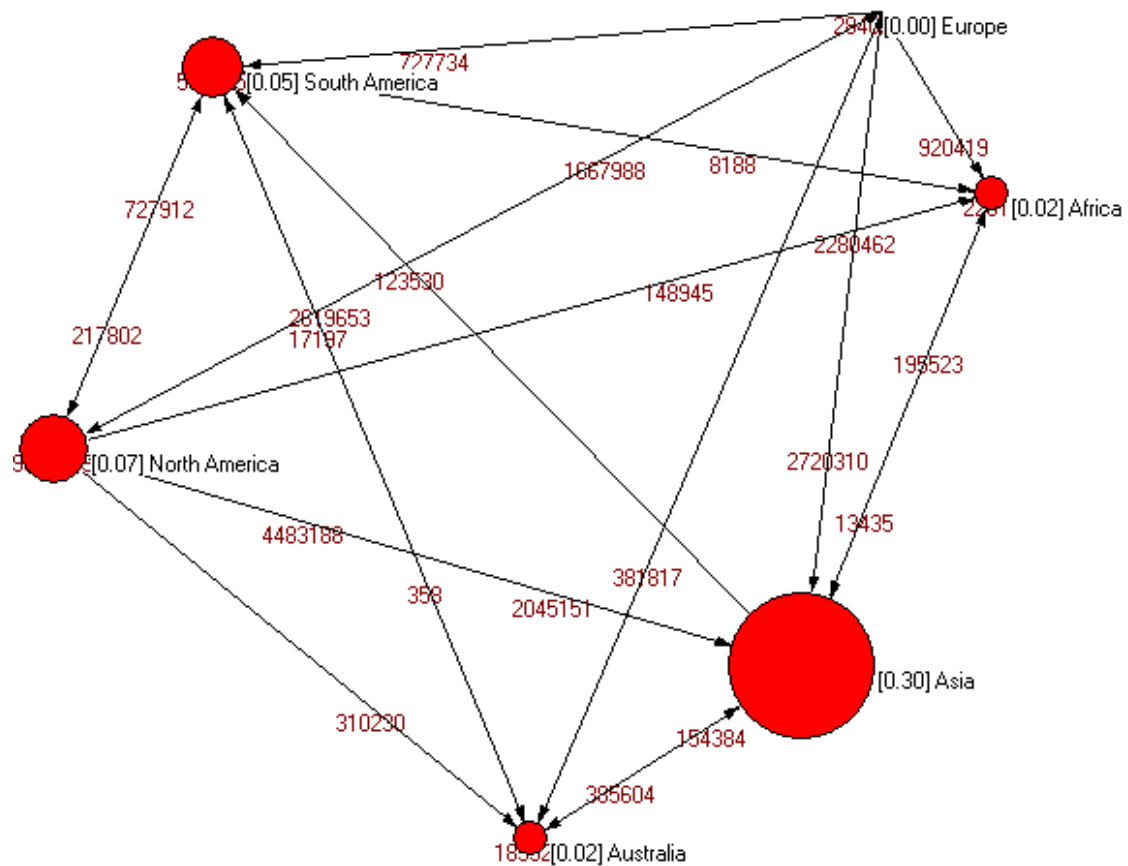
Editing Vector: 2. Betweenness centrality in N2 (6)

Redisplay

Vertex	Val	Label
1	0.016667	Africa
2	0.050000	South America
3	0.016667	Australia
4	0.000000	Europe
5	0.300000	Asia
6	0.066667	North America

Όπως ήταν ίσως αναμενόμενο, η ήπειρος της Ασίας παρουσιάζει τη μεγαλύτερη ενδιαμεσότητα. Αυτό σημαίνει ότι στο παγκόσμιο εμπόριο του μετάλλου (το έτος 1994) οι ασιατικές χώρες παίζουν κομβικό ρόλο και έχουν το μεγαλύτερο μερίδιο των συναλλαγών.

Αυτό είναι ακόμα πιο προφανές αν το διάνυσμα παρουσιαστεί γραφικά, επιλέγοντας
Draw > Network + First Vector:



32. Παραρτήματα

32.1 Τα περιεχόμενα του αρχείου `mynetwork.net`

```
*Vertices 28
 1 "v1"          0.1000    0.5000    0.5000
 2 "v2"          0.1100    0.4110    0.5000
 3 "v3"          0.1396    0.3264    0.5000
 4 "v4"          0.1873    0.2506    0.5000
 5 "v5"          0.2506    0.1873    0.5000
 6 "v6"          0.3264    0.1396    0.5000
 7 "v7"          0.4110    0.1100    0.5000
 8 "v8"          0.5000    0.1000    0.5000
 9 "v9"          0.5890    0.1100    0.5000
10 "v10"         0.6736    0.1396    0.5000
11 "v11"         0.7494    0.1873    0.5000
12 "v12"         0.8127    0.2506    0.5000
13 "v13"         0.8604    0.3264    0.5000
14 "v14"         0.8900    0.4110    0.5000
15 "v15"         0.9000    0.5000    0.5000
16 "v16"         0.8900    0.5890    0.5000
17 "v17"         0.8604    0.6736    0.5000
18 "v18"         0.8127    0.7494    0.5000
19 "v19"         0.7494    0.8127    0.5000
20 "v20"         0.6736    0.8604    0.5000
21 "v21"         0.5890    0.8900    0.5000
22 "v22"         0.5000    0.9000    0.5000
23 "v23"         0.4110    0.8900    0.5000
24 "v24"         0.3264    0.8604    0.5000
25 "v25"         0.2506    0.8127    0.5000
26 "v26"         0.1873    0.7494    0.5000
27 "v27"         0.1396    0.6736    0.5000
28 "v28"         0.1100    0.5890    0.5000

*Arcs
23  7 1
 7 15 1

*Edges
 7 19 1
```

32.2 Οι κώδικες των αρχείων Matlab

32.2.1 Ο κώδικας του αρχείου `get_random_graph.m`

```
function [ G ] = get_random_graph( N, K )

if N < 2
    error('N must be at least 2');
end

if K < 1
    error('K must be at least 1');
end

G = zeros(N);

% For each vertex, connect its K closest neighbours

p = K / N;

for i = 1 : N
    count = 0;
    for j = i + 1 : N
        if rand < p
            G(i,j) = 1;
            G(j,i) = 1;
            count = count + 1;
        end
    end
    if count == 0
        % ensure at least one edge
        j = round(rand * N);
        while (j < 1 || j == i || j > N)
            j = round(rand * N);
        end
        G(i,j) = 1;
        G(j,i) = 1;
    end
end
end
```

32.2.2 Ο κώδικας του αρχείου `write_matrix_to_pajek.m`

```

function write_matrix_to_pajek(matrix, path, varargin)

p.weighted=true;
p.threshold_low=0;
p.threshold_high=realmax;
p.directed=true;
p.labels={};
p.coords={};

validParams      = {      ...
    'weighted',      ...
    'threshold_low',      ...
    'threshold_high',      ...
    'directed',      ...
    'labels',      ...
    'coords',      ...
};

if nargin > 2
    if mod( numel( varargin ), 2 ) ~= 0
        error( 'write_matrix_to_pajek:InvalidInput', ['All
input parameters after the fileName must be in the ' ...
    'form of param-value pairs'] );
    end
    params = lower( varargin(1:2:end) );
    values = varargin(2:2:end);

    if ~all( cellfun( @ischar, params ) )
        error( 'write_matrix_to_pajek:InvalidInput', ['All
input parameters after the fileName must be in the ' ...
    'form of param-value pairs'] );
    end

    lcValidParams = lower( validParams );
    for ii = 1 : numel( params )
        result      = strmatch( params{ii}, lcValidParams
);
        % If unknown param is entered ignore it
        if isempty( result )
            continue
        end
        % If we have multiple matches make sure we don't have
a single unambiguous match before throwing
        % an error
        if numel( result ) > 1

```

```
        exresult      = strmatch( params{ii}, validParams,
'exact' );
        if ~isempty( exresult )
            result      = exresult;
        else
            % We have multiple possible matches, prompt user
            to provide an unambiguous match
            error( 'write_matrix_to_pajek:InvalidInput',
'Cannot find unambiguous match for parameter '%s'', ...
                varargin{ii*2-1} );
        end
    end
    result      = validParams{result};
    p.(result)  = values{ii};
end
end

[fid, msg] = fopen(path, 'w');

if fid < 0
    error('write_matrix_to_pajek:InvalidPath', 'Could not
open %s for writing: %s', path, msg);
end

if size(matrix,1)~=size(matrix,2)
    error('write_matrix_to_pajek:InavlidMatrix', 'Matrix
must be square.');
```

```
end

N=size(matrix,1);

% Write vertices
fprintf(fid, '*vertices %i', N);
for i = 1 : N

    fprintf(fid, '\n %i', i);
    if (~isempty(p.labels))
        fprintf(fid, ' "%s"', char(p.labels(i)));
    else
        fprintf(fid, ' "%i"', i);
    end
    if (~isempty(p.coords))
        fprintf(fid, ' %f %f %f', p.coords(i,1),
p.coords(i,2), p.coords(i,3));
    end
end

end
```

```
% Write edges
if p.directed == true
    fprintf(fid, '\n*arcs');
else
    fprintf(fid, '\n*edges');
end

for i = 1 : N
    start = 1;
    if (p.directed ~= true)
        start = i;
    end
    for j = start : N
        if (i ~= j)
            value = matrix(i,j);
            if value > p.threshold_low && value <
p.threshold_high
                fprintf(fid, '\n %i %i', i, j);
                if p.weighted == true
                    fprintf(fid, ' %f', value);
                end
            end
        end
    end
end

fclose(fid);
```

32.3 Εύρεση του συντομότερου μονοπατιού με τη βοήθεια βιολογικών οργανισμών που δε διαθέτουν εγκέφαλο

Η θεωρία γράφων ορίζεται με μαθηματικά μοντέλα. Τα μαθηματικά είναι ένα ανθρώπινο εργαλείο για την περιγραφή των φαινομένων που ο άνθρωπος μπορεί να αντιληφθεί με τα αισθητήρια όργανά του, αλλά και με τα τεχνολογικά εργαλεία μέτρησης που ο ίδιος έχει αναπτύξει. Όλα αυτά έγιναν εφικτά λόγω της εγκεφαλικής του δομής και δραστηριότητας.

Για τους εξελικτικούς βιολόγους, η ανάπτυξη του εγκεφάλου δεν είναι αποτέλεσμα συμπαντικής σκοπιμότητας. Προέκυψε σε βάθος εκατομμυρίων ετών εξέλιξης, μέσω τυχαίων φυσικών και χημικών γεγονότων. Είναι δηλαδή προϊόν ενός χημικού μονοπατιού το οποίο οι φυσικοί νόμοι επιτρέπουν να υπάρχει και να εκφράζεται.

Πιθανή ένδειξη ορθότητας της ιδέας αυτής, θα μπορούσε να αποτελεί μια πρόσφατη επιστημονική έρευνα κατά την οποία οι ερευνητές απέδειξαν ότι ένα είδος ευκαρυωτικού οργανισμού που ανήκει στα πρώτιστα⁴⁶, μπορεί να «υπολογίσει» το συντομότερο μονοπάτι σε ένα γράφο ή σε ένα λαβύρινθο^{47, 48}. Η επιστημονική του ονομασία είναι *Physarum polycephalum*⁴⁹:



⁴⁶ <http://en.wikipedia.org/wiki/Protist>

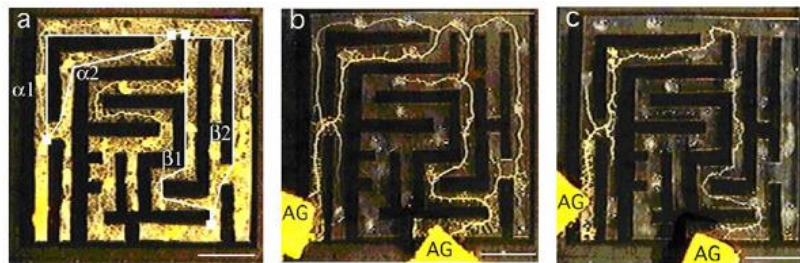
⁴⁷ <https://youtu.be/lIs27hu03yw>

⁴⁸ <https://youtu.be/czk4xgdhdY4>

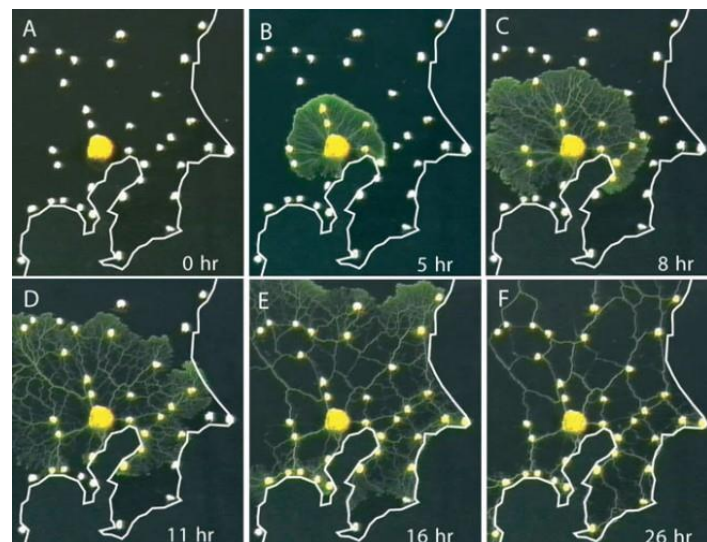
⁴⁹ http://en.wikipedia.org/wiki/Physarum_polycephalum

Δεν διαθέτει εγκέφαλο, ούτε κάποιου είδους νευρικό σύστημα. Όμως, παρ' όλα αυτά, είναι σε θέση να λαμβάνει αρκετά σύνθετες «αποφάσεις». Για παράδειγμα, μπορεί να βρει το συντομότερο μονοπάτι μέσα από έναν λαβύρινθο⁵⁰ προκειμένου να προσεγγίσει την τροφή που βρίσκεται στην έξοδο του λαβυρίνθου αυτού.

Αρχικά επεκτείνει τους δενδρίτες^{51, 52} του σε όλους τους διαδρόμους χαρτογραφώντας έτσι ολόκληρο το λαβύρινθο (α). Στη συνέχεια αποσύρει τον κάθε βλαστό που δεν «εντόπισε» τροφή (b) αφήνοντας πίσω ένα είδος λάσπης που λειτουργεί σαν εξωτερική μνήμη η οποία υπενθυμίζει στο *Physarum* τα αδιέξοδα μονοπάτια. Αποφεύγοντας τα μονοπάτια αυτά, φύεται αποκλειστικά κατά μήκος της συντομότερης διαδρομής (c) από την αρχή του λαβυρίνθου ως την τροφή.



Με τη βοήθεια του *Physarum*, ομάδα επιστημόνων κατάφερε να υπολογίσει με ακρίβεια το σιδηροδρομικό σύστημα του Τόκιο⁵³ (στάδια A έως F σε χρονικό διάστημα 26 ωρών):



⁵⁰ <http://arxiv.org/pdf/1106.0423v3.pdf>

⁵¹ <https://vimeo.com/67686724>

⁵² <https://youtu.be/mvBSkt6LhJE>

⁵³ <https://youtu.be/BZUQQmcR5-g>

Επίσης, υπολόγισαν και τους σημαντικούς δρόμους της Αγγλίας, του Καναδά, της Ισπανίας και της Πορτογαλίας. Ένα απλό κύτταρο, κατάφερε να επιλύσει πραγματικά προβλήματα για τα οποία θα απαιτούνταν άρτια εκπαιδευμένες ομάδες μηχανικών και εκατοντάδες ώρες εργασίας.

Το γεγονός αυτό είναι βέβαιο ότι θα αλλάξει τον τρόπο με τον οποίο η επιστημονική κοινότητα αντιμετωπίζει τον εγκέφαλο. Επίσης θα εμπνεύσει την ανάπτυξη νέων μεθόδων ανάλυσης γράφων, αλλά και νέων βιομαθηματικών μοντέλων.

33. Βιβλιογραφία

- [1] A. Mrvar και V. Batagelj, «Εγχειρίδιο χρήσης του Pajek,» 2 Απρίλιος 2015. [Ηλεκτρονικό]. Available: <http://mrvar.fdv.uni-lj.si/pajek/pajekman.pdf>.
- [2] T. Kamada και S. Kawai, «An algorithm for drawing general undirected graphs,» *Information Processing Letters*, 16 April 2003.
- [3] T. M. J. FRUCHTERMAN και E. M. REINGOLD, «Graph Drawing by Force-directed Placement,» *SOFTWARE-PRACTICE AND EXPERIENCE*, p. 36, November 1991.
- [4] S. Wasserman και K. Faust, «Part V. Dyadic and Triadic Methods,» σε *Social Network Analysis - Methods and Applications*, 1994.
- [5] V. Batagelj, A. Mrvar, A. Ferligoj και P. Doreian, «Generalized Blockmodeling with Pajek,» *Metodoloski zvezki*, pp. 455-467, 2004.
- [6] V. Batagelj και M. Zaversnik, «Short Cycles Connectivity,» 23 July 2013.
- [7] W. d. Nooy, A. Mrvar και V. Batagelj, «Exploratory Social Network Analysis with Pajek,» November 2010. [Ηλεκτρονικό]. Available: <http://vlado.fmf.uni-lj.si/pub/networks/book/esna2.htm>.