

Διπλωματική Εργασία

‘Σύγκριση αλγορίθμων εύρεσης Κυρίαρχου Συνόλου
Γραφημάτων με κεντρικοποιημένο τρόπο
(Connected Dominating Sets)’

Επώνυμο: Τριαντόπουλος

Όνομα: Ιωάννης

Έτος Εγγραφής: 2002

AM: 1702032

AEM:194



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΒΙΒΛΙΟΘΗΚΗ & ΚΕΝΤΡΟ ΠΛΗΡΟΦΟΡΗΣΗΣ
ΕΙΔΙΚΗ ΣΥΛΛΟΓΗ «ΓΚΡΙΖΑ ΒΙΒΛΙΟΓΡΑΦΙΑ»**

Αριθ. Εισ.: 6019/1
Ημερ. Εισ.: 07-11-2007
Δωρεά: Συγγραφέα
Ταξιθετικός Κωδικός: ΠΤ – ΜΗΥΤΔ
2007
ΤΡΙ

ΕΠΙΤΡΟΠΗ ΕΞΕΤΑΣΗΣ

1^{ος} Εξεταστής-Επιβλέπων Καθηγητής

Δρ. Κατσαρός Δημήτριος

Διδάσκων ΠΔ 407/80

Τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών & Δικτύων

Πανεπιστήμιο Θεσσαλίας

2^{ος} Εξεταστής

Δρ. Μποζάνης Παναγιώτης

Επίκουρος Καθηγητής

Τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών & Δικτύων

Πανεπιστήμιο Θεσσαλίας

ΠΕΡΙΕΧΟΜΕΝΑ

Συνοπτική Περιγραφή της Εργασίας.....	6
<u>ΚΕΦΑΛΑΙΟ 1^ο:Εισαγωγή.....</u>	8
1.1.Λίγα λόγια.....	8
1.2.Κατηγορίες των ασυρμάτων δικτύων	9
1.3.Mobile Ad Hoc Networks.....	9
1.4. Αντικείμενο της Εργασίας.....	11
<u>ΚΕΦΑΛΑΙΟ 2^ο:Μερικοί Ορισμοί.....</u>	13
2.1. Εισαγωγή	13
2.2. Ορισμοί.....	13
<u>ΚΕΦΑΛΑΙΟ 3^ο:Περιγραφή των αλγορίθμων.....</u>	18
3.1.Εισαγωγή.....	18
3.2.Node Imporance algorithm for Connected Dominating sets.....	18
3.3.Approximation Algorithms for CDS-Algorithm 1.....	22
3.4.Approximation Algorithms for CDS-Algorithm 2.....	25
3.5.Node Imporance Algorithm with color.....	29
<u>ΚΕΦΑΛΑΙΟ 4^ο:Περιγραφή των πειραμάτων.....</u>	33
4.1 Εισαγωγή.....	34
4.2 Πείραμα 1	34
4.3 Πείραμα 2.....	39
4.4 Πείραμα 3:.....	44
4.5 Πείραμα 4.....	52
4.6 Γενικά συμπεράσματα.....	57

<u>ΚΕΦΑΛΑΙΟ 5°: Αποτίμηση και μελλοντική εργασία.....</u>	58
5.1.Εισαγωγή	58
5.2.Οι αλγόριθμοι που μελετήσαμε.....	58
5.3.Τα πειράματα που έγιναν.....	59
5.4.Τα αποτελέσματα.....	60
5.5.Μελλοντική Εργασία.....	60
<u>Παραρτημα Α'.....</u>	62
A.1.Υλοποίηση του αλγορίθμου NI algorithm	62
A.2. Υλοποίηση του αλγορίθμου AACDS1.....	64
A.3. Υλοποίηση του αλγορίθμου AACDS2.....	65
A.4.Υλοποίηση του αλγορίθμου NI algorithm with color.....	66
A.5.Προγράμματα που δημιουργούν ad hoc δίκτυα-γράφους.....	67
A.6.Πρόγραμμα που αυξάνει ομοιόμορφα την εμβέλεια του κάθε κόμβο.....	68
<u>Αναφορές-Βιβλιογραφία.....</u>	70

Συνοπτική Περιγραφή της Εργασίας

Η εργασία αυτή γίνεται με σκοπό να μελετήσω και να συγκρίνω αλγορίθμους που σχετίζονται με την επιλογή κόμβων σε ένα *Mobile Ad Hoc* δίκτυο έτσι ώστε να καλύπτουν το όλους τους κόμβους του δικτύου. Το πρώτο κεφάλαιο είναι εισαγωγικό , λέω λίγα λόγια για τα ασύρματα δίκτυα και περιγράψω τα *Mobile Ad Hoc* δίκτυα και το τελειώνω αναφέροντας επιγραμματικά του αλγορίθμους που θα περιγράψω αργότερα. Στο δεύτερο κεφάλαιο αναφέρω μετρικούς ορισμούς από την θεωρία γραφημάτων για να γίνεται αντιληπτή η ορολογία που χρησιμοποιώ αργότερα. Στο τρίτο κεφάλαιο περιγράψω τους αλγορίθμους που συγκρίνω και δίνω ένα παράδειγμα του καθενός. Στο τέταρτο κεφάλαιο περιγράψω τα πειράματα που έκανα και τα συμπεράσματα που έβγαλα. Στο πέμπτο κεφάλαιο συνοψίζω όλα όσα ανέφερα στα προηγούμενα κεφάλαια και παραθέτω μελλοντική εργασία.

ΚΕΦΑΛΑΙΟ 1^ο

Εισαγωγή

1.1. Λίγα λόγια

Όπως έχει γίνει αντιληπτό την περασμένη δεκαετία είχαμε τρομερή ανάπτυξη της ασύρματης τεχνολογίας. Πιο συγκεκριμένα αναπτύχθηκαν οι υποδομές των δικτύων, αυξήθηκε η ανάπτυξη και η χρήση των ασύρματων εφαρμογών καθώς και εν τέλει η χρήση ασυρμάτων συσκευών όπως τα κινητά τηλέφωνα , τα PDAs και οι φορητοί υπολογιστές έγινε πιο ευρεία και οι υπολογιστικές τους δυνατότητες πιο ισχυρές σε σύγκριση με ενσύρματες υποδομές. Είναι ευρέως γνωστό πως όλα τα παραπάνω διαδραματίζουν σημαντικό ρόλο στην καθημερινή ζωή μας. Είναι αξιοσημείωτο να πούμε πως με ένα κινητό τηλέφωνο μπορεί κανείς να ελέγξει το ηλεκτρονικό ταχυδρομείο και να περιηγηθεί στο διαδίκτυο , με ένα PDA ή ένα φορητό υπολογιστή μπορεί κανείς να κατεβάσει χάρτες και χρησιμοποιήσει GPS τεχνολογίες, να ανταλλάξει αρχεία μέσω δικτύου και οποιαδήποτε λειτουργία μπορούσε να κάνει κάποιος με ένα σταθερό υπολογιστή μπορεί να τα υλοποιήσει ασύρματα με χρήση φορητών υπολογιστικών συσκευών. Πρόσφατα έχουν αναπτυχθεί συστήματα για αυτοκίνητα με τα οποία ο οδηγός μπορεί να γνωρίζει για την κατάσταση των γειτονικών αυτοκινήτων , ώστε να αποφεύγονται ατυχήματα , να ενημερώνονται οι οδηγοί για την κίνηση στον δρόμο και άλλες πληροφορίες . Ανάλογα ασύρματα δίκτυα συναντάμε στο στρατό, σε πλοία, σε δίκτυα αισθητήρων και άλλα.

1.2. Κατηγορίες των ασυρμάτων δικτύων

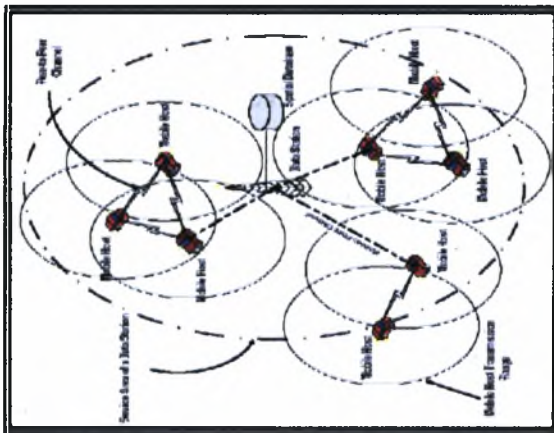
Τα ασύρματα δίκτυα με βάση την αρχιτεκτονική κατηγοριοποιούνται σε δύο βασικές κατηγορίες:

- **Infrastructure-based wireless network.** Στην κατηγορία αυτή ανήκουν τα δίκτυα με προκατασκευασμένη δομή από ενσύρματους κόμβους. Για παράδειγμα τα κυτταροειδή δίκτυα (cellular networks) ανήκουν σε αυτή την κατηγορία, τα οποία έχουν δημιουργηθεί από PSTN backbone switches, MSCs, base station και άλλα. Οι κόμβοι των δικτύων αυτών σταθεροί και η δομή του δικτύου είναι προκαθορισμένη.
- **Infrastructureless (ad hoc) network.** Στην περίπτωση αυτή η μορφή του δικτύου αλλάζει δυναμικά και αποτελείται από αυθαίρετους και ανεξάρτητους κόμβους. Ο ρόλος του κάθε κόμβου δεν είναι προκαθορισμένος, αντιθέτως κάθε κόμβος ενεργεί ανάλογα την κατάσταση του δικτύου χωρίς να υπάρχει προκαθορισμένη δομή του δικτύου. Στην περίπτωση των mobile ad hoc networks (MANETs) οι κόμβοι έχουν ρόλο στην ανακάλυψη και στην διατήρηση μονοπατιού σε σχέση με άλλους κόμβους.

1.3. Mobile Ad Hoc Networks

Στην εργασία αυτή θα ασχοληθώ με την μελέτη των MANET (mobile ad hoc network). Όπως περιγράψαμε και πιο πάνω ένα Manet αποτελείται από ένα σύνολο από κινητούς hosts κάθε ένας εκ των οποίων έχει πομποδέκτη. Ένα τέτοιο δίκτυο δεν υπάρχουν σταθμοί βάσης, ούτε υποδομή σταθερού δικτύου και έχουμε multi-hop επικοινωνία. Σε ένα MANET απαιτείται ένα πρωτόκολλο δρομολόγησης το οποίο να μπορεί να χειρίζεται τη μεταβαλλόμενη τοπολογία του δικτύου.

Στην εικόνα δίπλα παρατηρούμε ένα παράδειγμα ad hoc δικτύου, στο οποίο



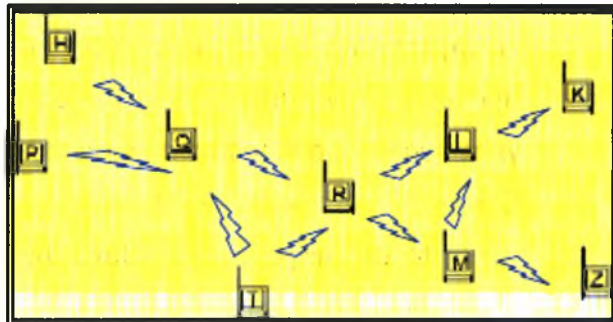
Εικόνα1

τα αυτοκίνητα είναι οι κόμβοι και ο κύκλος γύρω από το καθένα είναι το range του πομποδέκτη που έχει. Παρατηρούμε πως οι επικοινωνία μεταξύ δύο κόμβων είναι εφικτή όταν ο ένας είναι στο κύκλο του άλλου. Μπορούμε να φανταστούμε πόσο γρήγορα αλλάζει η τοπολογία σε ένα τέτοιο

δίκτυο

Σε ένα ad hoc δίκτυο η επικοινωνία μεταξύ δύο κόμβων ,που δεν υπάρχει μεταξύ τους ζεύξη (δεν βρίσκεται ο ένας στη περιοχή εμβέλειας του άλλου), επιτυγχάνεται μέσω ενδιάμεσων κόμβων , οι οποίοι προωθούν τα μηνύματα μέχρι να δημιουργηθεί ένα

κανάλι επικοινωνίας μεταξύ των κόμβων που επιθυμούν να επικοινωνήσουν μεταξύ τους. Για παράδειγμα στη διπλανή εικόνα για στήλη ο κόμβος 'H' στον 'R' πρέπει να στήλη πρώτα στον Q. Σε ένα τέτοιο δίκτυο μας



Εικόνα2

απασχολεί πολύ η πράξη της παν-

εκπομπής (broadcast task) , η οποία χρησιμοποιείται σε πολλές λειτουργίες στο δίκτυο. Με λίγα λόγια όταν γίνεται πανεκπομπή ένας κόμβος επιθυμεί να στείλει ένα μήνυμα το οποίο πρέπει να το λάβουν όλοι οι κόμβοι του δικτύου . Η πανέκπομπη σε ένα δίκτυο όπου υπάρχει υποδομή σταθερού δικτύου είναι σχετικά εύκολη υπόθεση. Σε ένα ad Hoc δίκτυο στο οποίο η τοπολογία και οι γειτονιά κάθε κόμβου δεν είναι σταθερή τα πράγματα δυσκολεύουν.

Η πιο απλή υλοποίηση της πανεκπομπής είναι με πλημμύρα μηνυμάτων στο δίκτυο. Σε αυτή την περίπτωση ένας κόμβος-host αναμεταδίδει ένα μήνυμα

όταν το λάβει για πρώτη φορά σε όλους τους κόμβους με τους οποίους επικοινωνεί. Με αυτή την υλοποίηση όμως δημιουργούνται προβλήματα όπως η άδικη επανεκπομπή μηνυμάτων, η δημιουργία συγκρούσεων, η υπερφόρτωση του δικτύου από μηνύματα. Με λίγα λόγια η υλοποίηση πανεκπομπής με πλημμύρα αποτελεί ακριβή λύση. Έτσι η ανεύρεση μεθόδων για την επιλογή κόμβων που να καλύπτουν το δίκτυο είναι επιτακτική. Στην προσπάθεια αυτή έχουν προταθεί πολλοί αλγόριθμοι οι οποίοι διακρίνονται σε probability-based, area-based και neighbor-knowledge-based methods. Οι δυο πρώτες οικογένειες αλγορίθμων αν και πετυχαίνουν μείωση στο σύνολο των κόμβων όπου προωθούνται τα μηνύματα δεν εγγυώνται 100% κάλυψη του δικτύου σε αντίθεση με τους neighbor-knowledge-based αλγορίθμους. Οι αλγόριθμοι αυτοί βασίζονται στο να αναγνωρίζουν κόμβους οι οποίοι σχηματίζουν κυρίαρχο σύνολο του γραφήματος (connected dominating set) το οποίο σχηματίζεται από το ad-hoc δίκτυο.

1.4. Αντικείμενο της Εργασίας

Η εργασία αυτή γίνεται με σκοπό να υλοποιήσω και να συγκρίνω αλγορίθμους σχετικά με την επιλογή των κόμβων για την δημιουργία Κυρίαρχου Συνόλου γραφημάτων (connected dominating set) από ένα ad-hoc δίκτυο.

Οι αλγόριθμοι που θα μελετήσω στις παρακάτω ενότητες είναι οι εξής:

- **Node Importance algorithm for Connected Dominating sets:** Δημιουργία connected dominating set με βάση μια ποσότητα που θα υπολογίζω για κάθε κόμβο που θα την ονομάζω node importance .
- **Approximation Algorithms for Connected Dominating sets:** Στην κατηγορία αυτή μελετώ δύο αλγορίθμους υπολογίζουν το Connected Dominating Set με βάση κυρίως τον αριθμό των γειτόνων κάθε κόμβου στο δίκτυο.
- **Node Importance algorithm with color for Connected Dominating Sets:** Δημιουργία ενός Connected Dominating Set με βάση της ποσότητας Node Importance που χρησιμοποιούμε στον πρώτο αλγόριθμο και τον χρωματισμό των κόμβων από τους άλλους δύο αλγορίθμους.

Γενικά ένα Connected Dominating Set για Wireless Ad Hoc δίκτυο είναι το σύνολο των κόμβων το οποίο μπορούμε να το χρησιμοποιήσουμε για να κάνουμε Broadcast στο δίκτυο και να αντιμετωπίσουμε προβλήματα όπως η πλημμύρα στο δίκτυο. Το σύνολο των κόμβων αυτό ουσιαστικά καλύπτει όλο το δίκτυο-γράφημα .

ΚΕΦΑΛΑΙΟ 2^ο

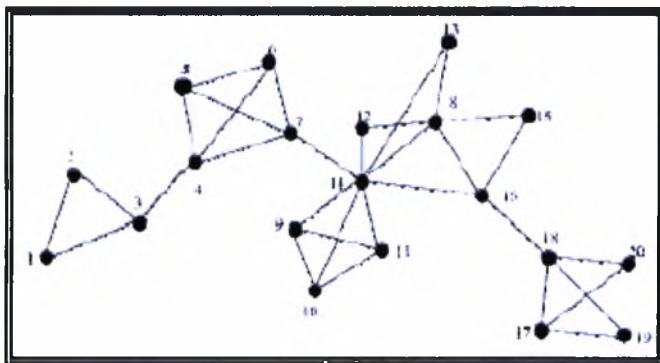
Μερικοί Ορισμοί

2.1. Εισαγωγή

Πριν ξεκινήσουμε την περιγραφή και ανάλυση των αλγορίθμων πρέπει να αναλύσουμε τις βασικές έννοιες και τους βασικούς ορισμούς που θα χρησιμοποιήσω παρακάτω για να περιγράψω τους αλγορίθμους που μελέτησα (τους ανέφερα επιγραμματικά στην ενότητα 1.4), έτσι ώστε να μην υπάρχει σύγχυση από τον αναγνώστη και να είναι όσο το δυνατό πιο ευανάγνωστα τα επόμενα κεφάλαια.

2.2. Ορισμοί

Mobile Ad hoc δίκτυο (MANET) όπως περιγράψαμε και πιο πάνω είναι ένα δίκτυο στο οποίο οι κόμβοι-hosts είναι εξοπλισμένοι με πομποδέκτη, δεν υπάρχει υποδομή σταθερού δικτύου, έχουμε multi-hop επικοινωνία. Σε ένα ad hoc δίκτυο κάθε κόμβος είναι αυτόνομος και οι γειτονιά του μπορεί να μεταβάλλεται. Στην εργασία αυτή θα προσομοιώνω ένα ad hoc δίκτυο με ένα γράφο $G(V,E)$ όπου V οι κορυφές και E οι ακμές του γράφου. Μία ακμή με κορυφές α, β που ανήκει στο γράφο αντιπροσωπεύει το γεγονός ότι το τερματικό- κόμβος α επικοινωνεί με το β και το αντίθετο.

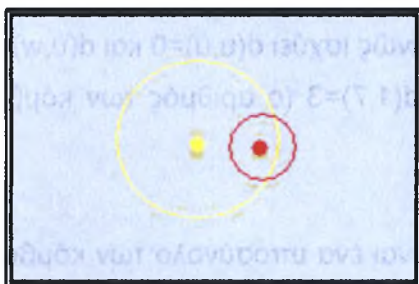


Εικόνα 3

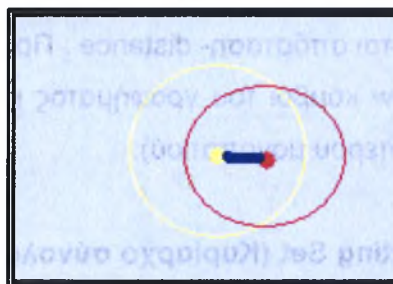
Η εικόνα 3 αναπαριστά ένα παράδειγμα γράφου που περιγράψαμε παραπάνω. Οι αριθμημένοι κόμβοι αναπαριστούν τα μέλη του MANET και οι μεταξύ τους ακμές αναπαριστούν ποιος επικοινωνεί με ποιον.

Για παράδειγμα μπορούμε να δούμε πως ο κόμβος 1 επικοινωνεί με τον 2^ο και τον 3^ο κόμβο.

Για να υπάρχει μια **σύνδεση-link** μεταξύ σε δύο κόμβους πρέπει στον κύκλο, που δημιουργείται με κέντρο την θέση του κόμβου και ακτίνα το range, να περιέχει την θέση του δεύτερου κόμβου και αντίστοιχα ο δεύτερος κόμβος να στον κύκλο που δημιουργεί να περιέχει τον πρώτο κόμβο. Συνεπώς στα δίκτυα που μελετάμε έχουν bydirectional links (αν υπάρχει link από ένα κόμβο α σε ένα β υπάρχει και από τον β στον α).



Εικόνα 4.α



Εικόνα 4.β

Για παράδειγμα στην Εικόνα 4.α παρατηρούμε πως δεν υπάρχει link μεταξύ του κίτρινου και του κόκκινου κόμβου, αφού ο κίτρινος κόμβος δεν περιέχεται στον κύκλο που ορίζει ο κόκκινος κόμβος. Αντίθετα στην Εικόνα 4.β έχουμε σύνδεση μεταξύ των δύο κόμβων.

Γειτονιά πρώτης τάξης ενός κόμβου (1 hop neighborhood) είναι οι γειτονικοί του κόμβοι. **Γειτονιά δεύτερης τάξης** ενός κόμβου (2 hop neighborhood) είναι οι κόμβοι που είναι γείτονες της πρώτης τάξης γειτονιάς του κόμβου και δεν ανήκουν στην γειτονιά πρώτης τάξης. Ο συνδυασμός των 1 hop και 2 hop

neighborhood αποτελεί το **12hop-neighborhood**. Για παράδειγμα στην Εικόνα 3 ο κόμβος 1 έχει 1hopneighbors του κόμβους 2,3 - 2hopneighbors τον 4 – 12hopneighbors τους 2,3,4. Πιο αυστηρός ορισμός για το 12hopneighborhood είναι η ένωση του στοιχειοσυνόλου που δημιουργείται από το σύνολο των κόμβων του 1hop neighborhood με το 2hop-neighborhood.

Μονοπάτι μεταξύ δύο κόμβων ορίζουμε ως την ακολουθία ακμών που πρέπει να ακολουθήσουμε για να μεταβούμε από τον ένα στον άλλο. Για παράδειγμα οι ακολουθίες ακμών $(1,3)(3,4)(4,5)(5,6)(6,7)$, $(1,2)(2,3)(3,4)(4,7)$ και $(1,3)(3,4)(4,7)$ αποτελούν μονοπάτια από τον κόμβο 1 στον 7, ενώ το μονοπάτι $(1,3)(3,4)(4,9)$ δεν αποτελεί μονοπάτι στο γράφημα αφού η ακμή που ορίζεται από τους κόμβους 4 και 9 δεν υφίσταται. Το μονοπάτι με της λιγότερες ακμές από ένα κόμβο σε ένα άλλο ονομάζεται **συντομότερο μονοπάτι**. Στο προηγούμενο παράδειγμα και τα τρία μονοπάτια αποτελούν μονοπάτια από τον κόμβο 1 στον 7 αλλά το τρίτο μονοπάτι αποτελεί συντομότερο. Ο αριθμός των ακμών που παρεμβάλλονται σε ένα μονοπάτι ονομάζεται απόσταση- distance . Προφανώς ισχύει $d(u,u)=0$ και $d(u,w)=d(w,u)$ όπου u,w κόμβοι του γραφήματος και $d(1,7)=3$ (ο αριθμός των κόμβων του συντομότερου μονοπατιού).

Dominating Set (Κυρίαρχο σύνολο) είναι ένα υποσύνολο των κόμβων ενός γράφου(στην περίπτωση μας οι κόμβοι του γράφου θα θεωρούμε τους σταθμούς του δικτύου και οι ακμές μεταξύ κόμβων είναι ποιοι κόμβοι επικοινωνούν με ποιους) τέτοιο ώστε κάθε κόμβος είτε ανήκει στο υποσύνολο αυτό είτε είναι προσκείμενος (adjacent) σε τουλάχιστον έναν από τους κόμβους αυτού του υποσυνόλου.

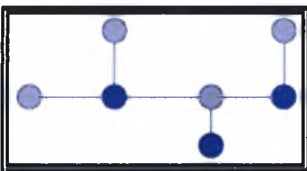


Εικόνα 5

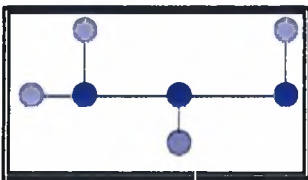
Οι μπλε κόμβοι στο διπλανή εικόνα αποτελούν dominating set του γραφήματος αφού κάθε κόμβος του γραφήματος ή είναι γειτονικός ή ανήκει στο Dominating Set.

Στο σχήμα αυτό οι μπλε κόμβοι δεν αποτελούν Dominating set αφού ο κόμβος που βρίσκεται στον κόκκινο κύκλο που δεν ανήκει στο dominating set δεν γειτονεύει με κανένα μπλε κόμβο.

Connected Dominating Set (Συνδεδεμένο Κυρίαρχο Σύνολο) είναι ένα Dominating Set του γραφήματος, τέτοιο ώστε το υπογράφημα που αντιστοιχεί στο Dominating Set να είναι συνδεδεμένο.



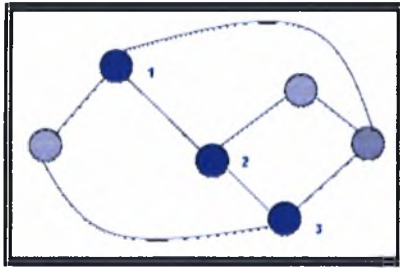
Οι μπλε κόμβοι του διπλανού γραφήματος δεν αποτελούν Connected Dominating set δεν είναι συνδεδεμένοι μεταξύ τους.



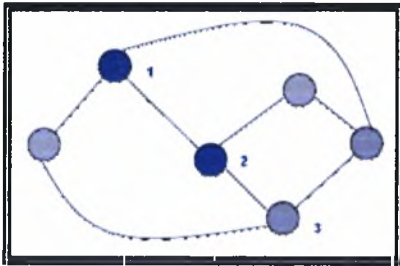
Εικόνα 6

Σε αντίθεση με το πιο πάνω γράφημα οι μπλε κόμβοι του γραφήματος στο διπλανό σχήμα αποτελούν connected dominating set αφού όπως είπαμε και στον ορισμό το υπογράφημα που αντιστοιχεί στο Dominating Set είναι ενωμένο.

Minimum Connected Dominating Set (Ελάχιστο Συνδεδεμένο Κυρίαρχο Σύνολο) είναι ένα Connected Dominating Set, τέτοιο ώστε να έχει το ελάχιστο μέγεθος (δηλ., το μικρότερο αριθμό κόμβων).



Οι μπλε κόμβοι του διπλανού γραφήματος σχηματίζουν Connected Dominating Set , όχι όμως Minimum, αφού παρατηρούμε πως ο κόμβος 3 είναι περιττός επειδή την γειτονιά του την καλύπτει ο κόμβος 1 και τον ίδιο ο κόμβος 2.



Συνεπώς οι μπλε κόμβοι της διπλανής εικόνας αποτελούν Min Connected Dominating Set για τον λόγο που εξηγήσαμε παραπάνω.

Εικόνα 7

ΚΕΦΑΛΑΙΟ 3^ο

Περιγραφή των αλγορίθμων

3.1 Εισαγωγή

Στο κεφάλαιο αυτό θα πούμε δύο λόγια για τα βήματα των αλγορίθμων που μελέτησα και θα δώσω παραδείγματα για το πως τρέχουν έτσι ώστε ο αναγνώστης να καταλάβει επακριβώς πως λειτουργεί και πως κλιμακώνει κάθε αλγόριθμος. Δεν θα αναφερθώ σε ζητήματα υλοποίησης. Το πως υλοποίησα του αλγορίθμους, το τι έκανα στα πειράματα και το πως μπορεί κάποιος να δοκιμάσει τα προγράμματα που έφτιαξα περιγράφεται αναλυτικά στο Παράρτημα Α'.

3.2. Node Importance algorithm for Connected Dominating sets

Η μέθοδος αυτή με λίγα λόγια υπολογίζει ένα μέτρο για κάθε κόμβο που προσεγγίζει την 'σπουδαιότητα' του κόμβου στο δίκτυο (Node Importance). Αφού υπολογίσουμε αυτό το μέγεθος για κάθε κόμβο, το Connected Dominating Set θα υπολογιστεί προσθέτοντας κόμβους με βάση το μεγαλύτερο Node Importance των κόμβων που δεν ανήκουν στο Connected Dominating Set μέχρι να καλυφθεί όλος ο γράφος-δίκτυο. Δηλαδή έχουμε ένα κεντρικοποιημένο πρωτόκολλο.

Αξίζει σε αυτό το σημείο να αναφέρουμε περισσότερα για το Node Importance ενός κόμβου:

Ορισμός: Node Importance ενός κόμβου u είναι ίσο με:

$$NI(u) = \sum_{w \neq u} \sigma_{uw}(u) / \sigma_{uw} \text{ όπου } u \neq w \text{ ανήκουν } V$$

Το μέγεθος σ_{uw} αντιπροσωπεύει το αριθμό των **κοντινότερων** μονοπατιών από τον κόμβο u στον κόμβο w . Παρόμοια το μέγεθος $\sigma_{uw}(u)$ είναι ο αριθμός των κοντινότερων μονοπατιών τα οποία περιέχουν τον κόμβο u .

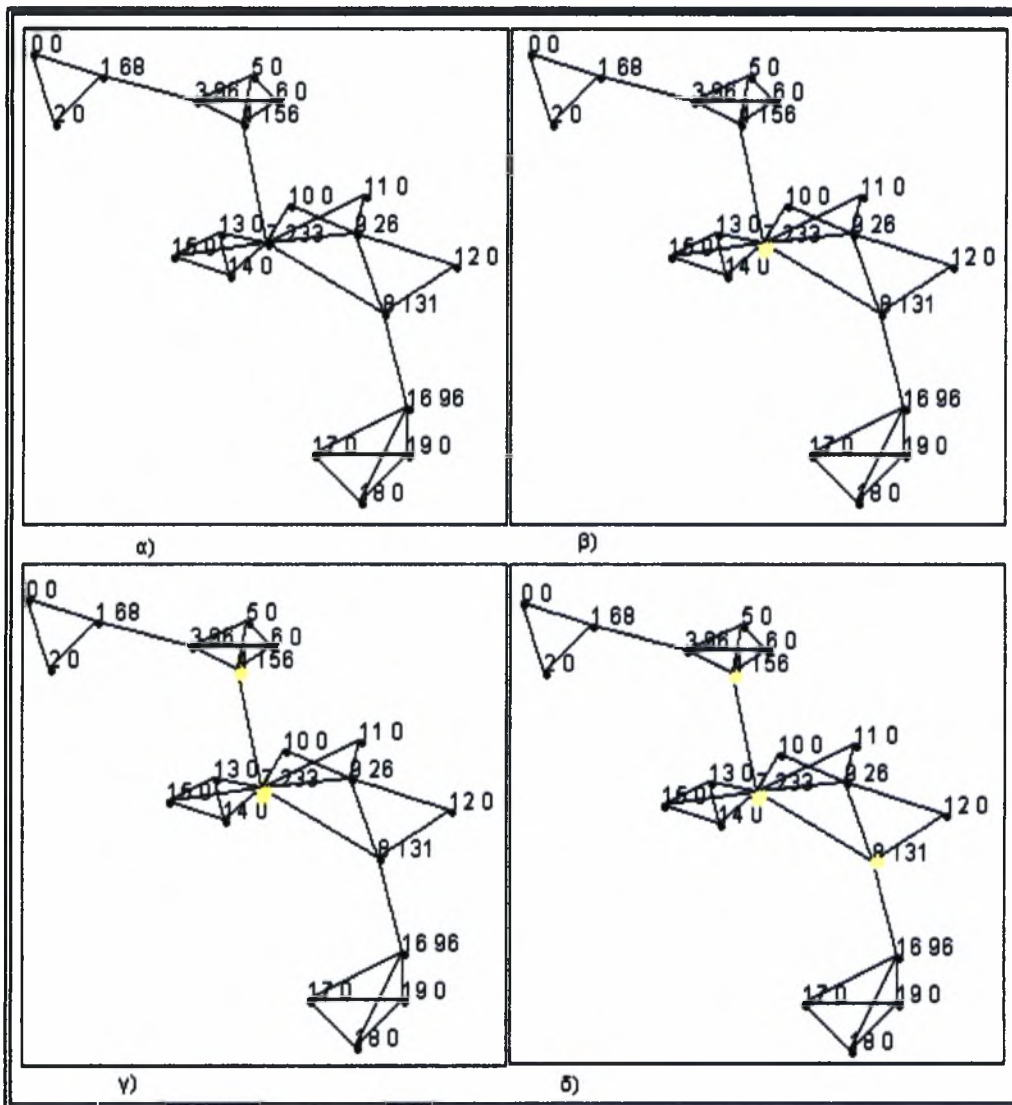
Από τον ορισμό και μόνο παρατηρούμε πως οι κόμβοι με μεγάλο Node Importance αποτελούν κομβικά σημεία στο γράφο. Γι' αυτό και ο αλγόριθμος αυτός προσθέτει στο Connected Dominating Set κόμβους με κριτήριο πόσο μεγάλο Node Importance έχει. Η πολυπλοκότητα για να υπολογιστεί το μέγεθος αυτό σε ένα γράφο που αποτελείται από n ακμές και m κορυφές και είναι της τάξης $O(n \cdot m)$.

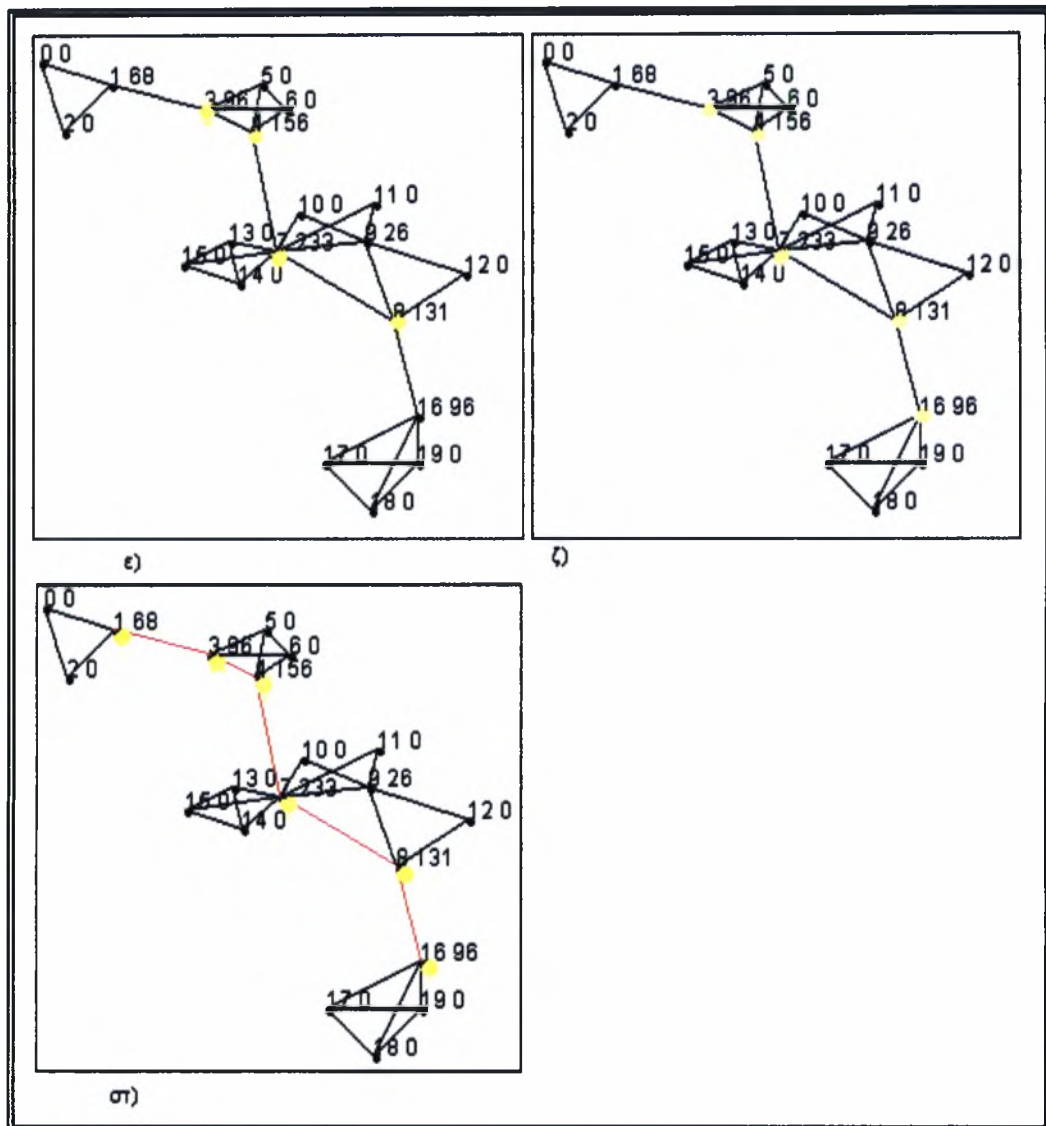
Εν τέλει τα βήματα για τον αλγόριθμο είναι:

1. Υπολογίζω για τον γράφο το Node Importance για κάθε κόμβο.
2. Επιλέγω το κόμβο με το μεγαλύτερο Node Importance και τον προσθέτω στο Connected Dominating Set
3. Επαναλαμβάνω το βήμα 2 μέχρις ότου ο γράφος που θα δημιουργηθεί να καλύπτει όλο το γράφο.

(περισσότερα για το πώς υλοποιώ τον αλγόριθμο στο πρόγραμμα θα βρείτε στο παράρτημα Α').

Ας δώσουμε τώρα ένα αναλυτικό παράδειγμα για το πώς κλιμακώνει ο αλγόριθμος σε ένα γράφημα.





Εικόνα 8

Στην Εικόνα 8 έχουμε ένα παράδειγμα του Node Importance algorithm . Στην 8.α έχουμε το γράφημα του MANET και βλέπουμε δίπλα στο id του κάθε κόμβου την ποσότητα node importance . Όπως φαίνεται στη εικόνα 8.β επιλέγουμε πρώτα τον κόμβο 7 αφού έχει node importance 233 ,που είναι και το μεγαλύτερο σε όλο το γράφημα. Έπειτα επιλέγω τον κόμβο 4 ο οποίος έχει Node Importance 156 όπως φαίνεται στην επόμενη εικόνα. Εν τέλει επιλέγω διαδοχικά του κόμβους 8, 3,16, 1 όπως φαίνεται στα επόμενα σχήματα και καταλήγω στην εικόνα 8.στ όπου έχει δημιουργηθεί ένα Connected Dominating Set .

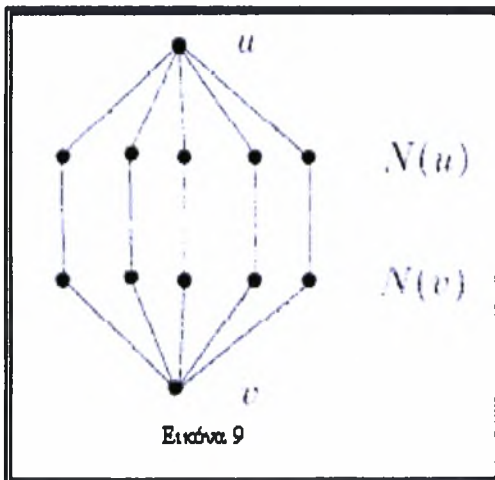
3.3. Approximation Algorithms for Connected Dominating sets- Algorithm 1

Ο αλγόριθμος αυτός βασίζεται στο να επεκτείνεις ένα δέντρο ξεκινώντας από τον κόμβο με τους περισσότερους γείτονες και αργότερα εξετάζοντας τους γείτονες προσθέτει στο δέντρο τον πιο κατάλληλο. Ο αλγόριθμος χρησιμοποιεί 'χρώματα' για κάθε κόμβο για να του δώσει κάποια ιδιότητα. Πιο συγκεκριμένα τους κόμβους που ανήκουν στο Connected Dominated Set χρωματίζονται μαύροι, γκριζοί χρωματίζονται οι κόμβοι που είναι γειτονικοί στους μαύρους και οι υπόλοιποι με το χρώμα λευκό. Μπορούμε να πούμε πως το χρώμα παίζει και κάποιο άλλο ρόλο: με το μαύρο είναι οι κόμβοι που ανήκουν στο Connected Dominating Set, οι γκρι είναι αυτοί που καλύπτονται από το Connected Dominating Set και οι λευκοί είναι αυτοί που δεν καλύπτονται και πρέπει να καλυφθούν.

Πιο συγκεκριμένα τα βήματα του αλγορίθμου είναι τα εξής:

1. Στην αρχή όλοι οι κόμβοι είναι άσπροι.
2. Με βάση τον αριθμό των γειτόνων κάθε κόμβου επιλέγουμε αυτόν με τους περισσότερους γείτονες και το χρωματίζουμε μαύρο, ενώ τους γείτονες του γκρι.
3. Από τους γκρι επιλέγουμε αυτόν με τους περισσότερους λευκούς και τον χρωματίζουμε μαύρο ενώ αντίστοιχα τους γείτονες του γκριζούς.
4. επαναλαμβάνουμε τα βήματα 2,3 μέχρι να μην μείνει κανένας λευκός γείτονας.

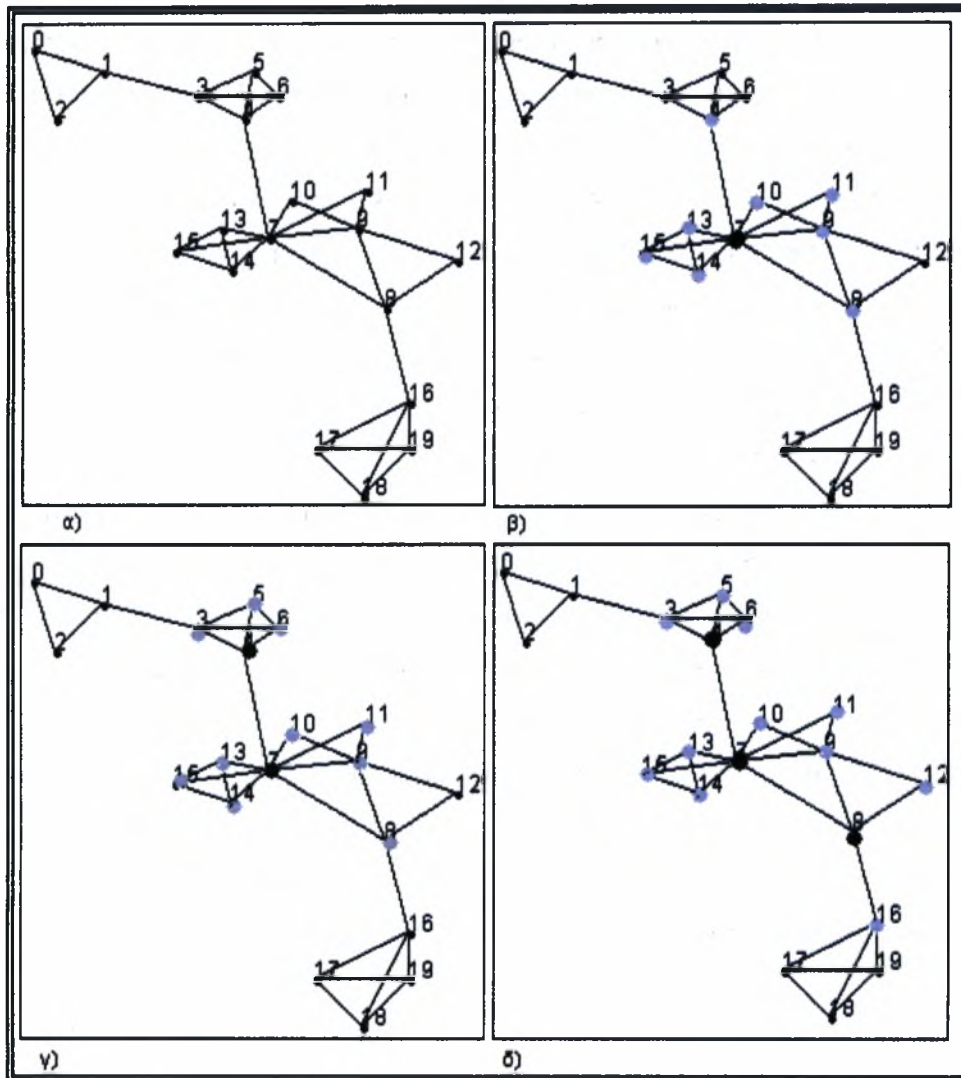
Ο παραπάνω αλγόριθμος μπορεί να βελτιστοποιηθεί ως εξής. Αντί κάθε

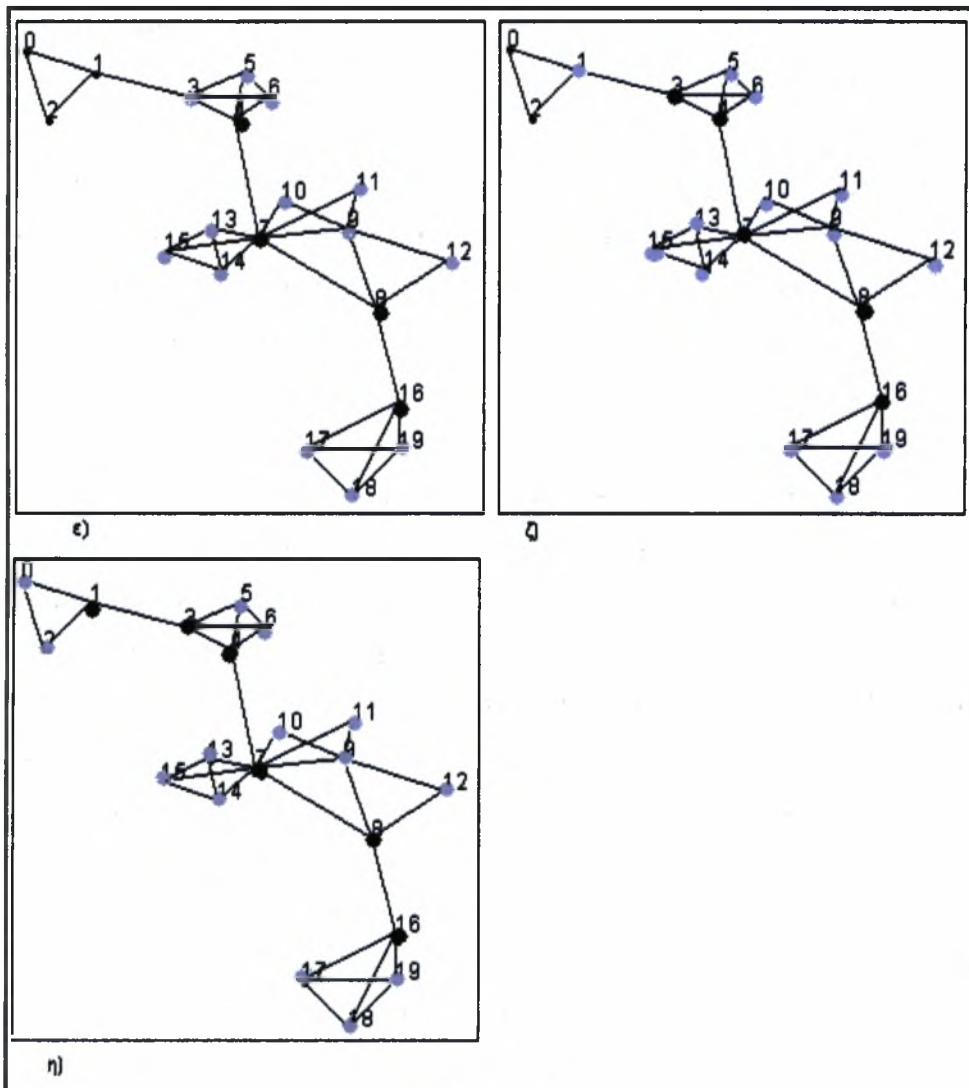


επανάληψη να ελέγχουμε μόνο ένα γκρι, να ελέγχουμε και ζεύγη γειτόνων γκρι-άσπρου, εκτός από μεμονωμένους κόμβους, και όποιος έχει μεγαλύτερο αριθμό λευκών να χρωματίζεται μαύρος. Για παράδειγμα στη Εικόνα 9 αν επιλέγαμε στην αρχή τον u και τον

χρωματίζαμε μαύρο και μετά επιλέγαμε έναν ένα τους γκριζούς κόμβους υπήρχε περίπτωση όλοι οι γείτονες του u να γίνουν μαύροι. Αν ελέγχαμε ζεύγη γκρι και λευκών κόμβων δεν θα είχαμε το παραπάνω πρόβλημα.

Παρακάτω δίνω ένα παράδειγμα του αλγορίθμου που περιγράφει αναλυτικά τα βήματά του :





Εικόνα 10

Στην εικόνα 10 έχουμε ένα δίκτυο στο οποίο παρατηρούμε τα βήματα του αλγορίθμου που περιγράψαμε παραπάνω. Στην αρχή όλοι οι κόμβοι είναι λευκοί. Στην εικόνα 10α επιλέγουμε πρώτα το κόμβο με id 7 αφού έχει τον μέγιστο αριθμό γειτόνων (8) και χρωματίζουμε κάθε γείτονα γκρι. Από τους γκρι επιλέγω τον κόμβο με του περισσότερους λευκούς γείτονες και τον χρωματίζω μαύρο. Στο σχήμα 10β ο κόμβος αυτός είναι ο 4 τον οποίο στο σχήμα 10γ τον χρωματίζω μαύρο και την λευκή γειτονιά του γκρι. Αυτή η διαδικασία γίνεται μέχρι στο σχήμα 10 η) στο οποίο δεν υπάρχουν λευκοί κόμβοι και το σύνολο των μαύρων κόμβων είναι connected και καλύπτουν όλο το γράφημα.

3.4. Approximation Algorithms for Connected Dominating sets

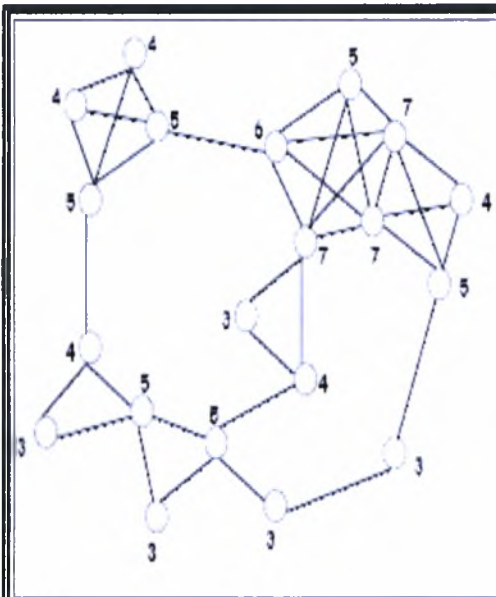
Algorithm 2

Ο αλγόριθμος αυτός αποτελεί μία εναλλακτική προσέγγιση του αλγορίθμου που περιγράψαμε πιο πάνω στην ενότητα 3.2. Η κεντρική ιδέα του αλγορίθμου είναι αντί να επεκτείνει ένα δέντρο πάνω στον γράφο μέχρι να καλύψει τους κόμβους του να επεκτείνει ξεχωριστά κομμάτια και στο τέλος να τα ενώσει.

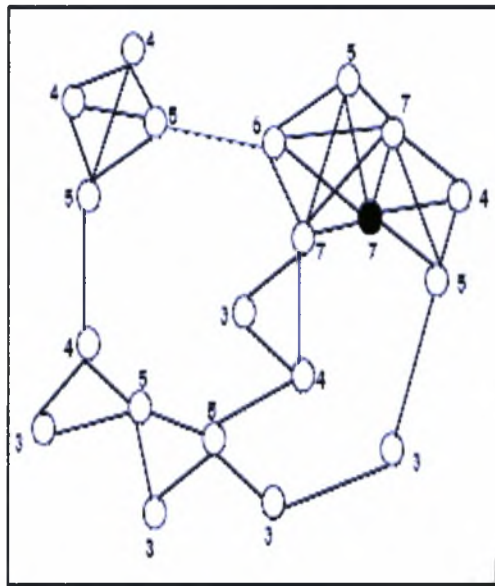
Από' τι συμπεραίνουμε από τα πάνω ο αλγόριθμος έχει δύο φάσεις. Στην πρώτη φάση αρχικά όλοι οι κόμβοι είναι λευκοί και επαναληπτικά επιλέγουμε ένα κόμβο και τον χρωματίζουμε μαύρο. Τους κόμβους που είναι προσκείμενοι κάθε σε ένα μαύρο τους χρωματίζουμε γκρι. Το κριτήριο για να χρωματίσουμε ένα κόμβο μαύρο είναι να έχει τον μεγαλύτερο αριθμό σε pieces. Με τον όρο Piece αναφερόμαστε σε μια επιμέρους υποδομή του γραφήματος που μπορεί να είναι απλά ένας λευκός κόμβος ή να περιέχει ένα maximal σύνολο από black κόμβους των οποίων το weakly induced υπογράφημα είναι συνδεδεμένο συν όποιοι προσκείμενοι gray κόμβοι.

Στην δεύτερη φάση προσθέτουμε μαύρους κόμβους μέχρι να το σύνολο των είδη υπάρχων μαύρων κόμβων που υπάρχουν να γίνει connected. Το κριτήριο με το οποίο επιλέγω τους κόμβους το εξηγώ αναλυτικά στο Παράρτημα Α.

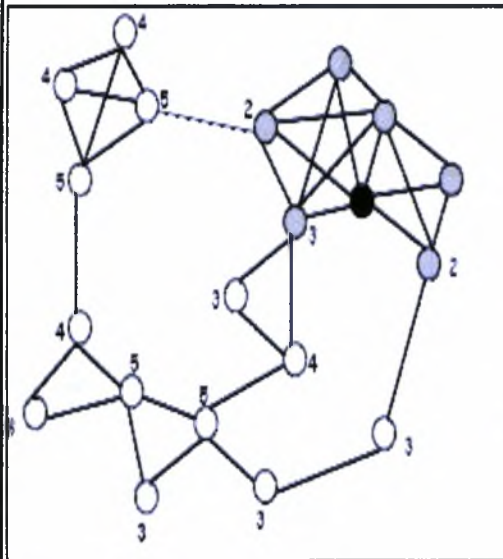
Παρακάτω δίνω ένα παράδειγμα που δείχνει ξεκάθαρα και κάνει πιο κατανοητές της φάσεις του αλγορίθμου.



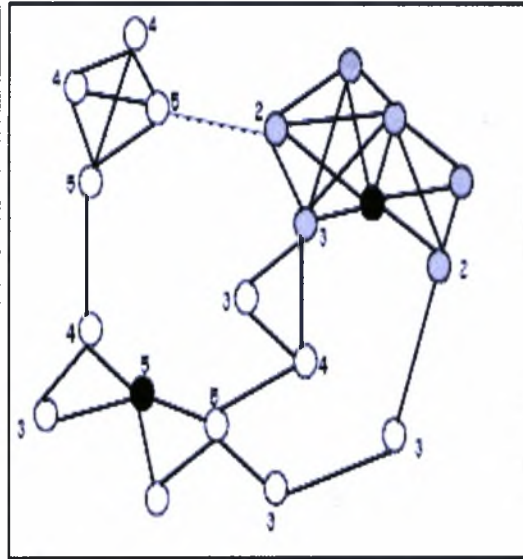
a)



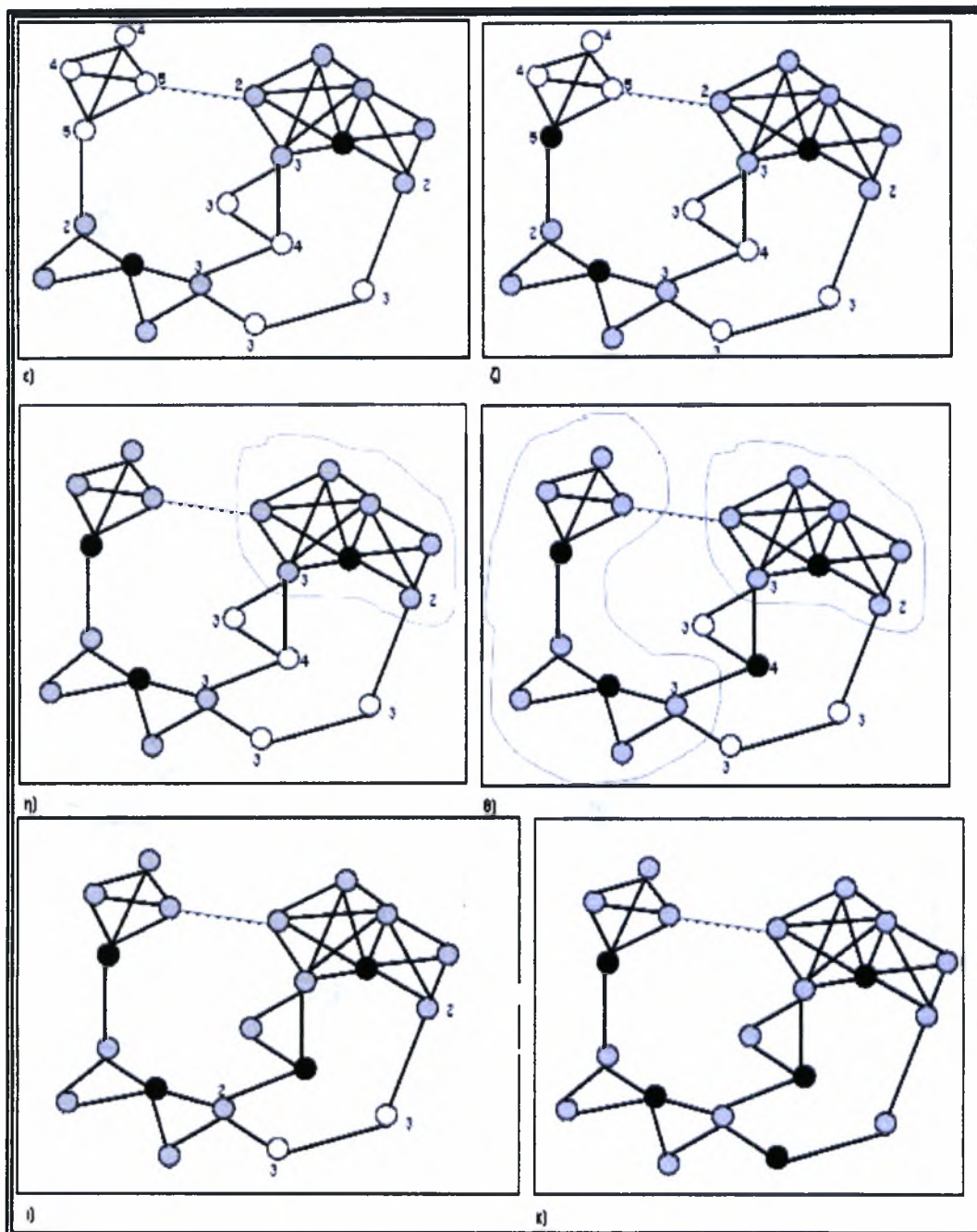
b)



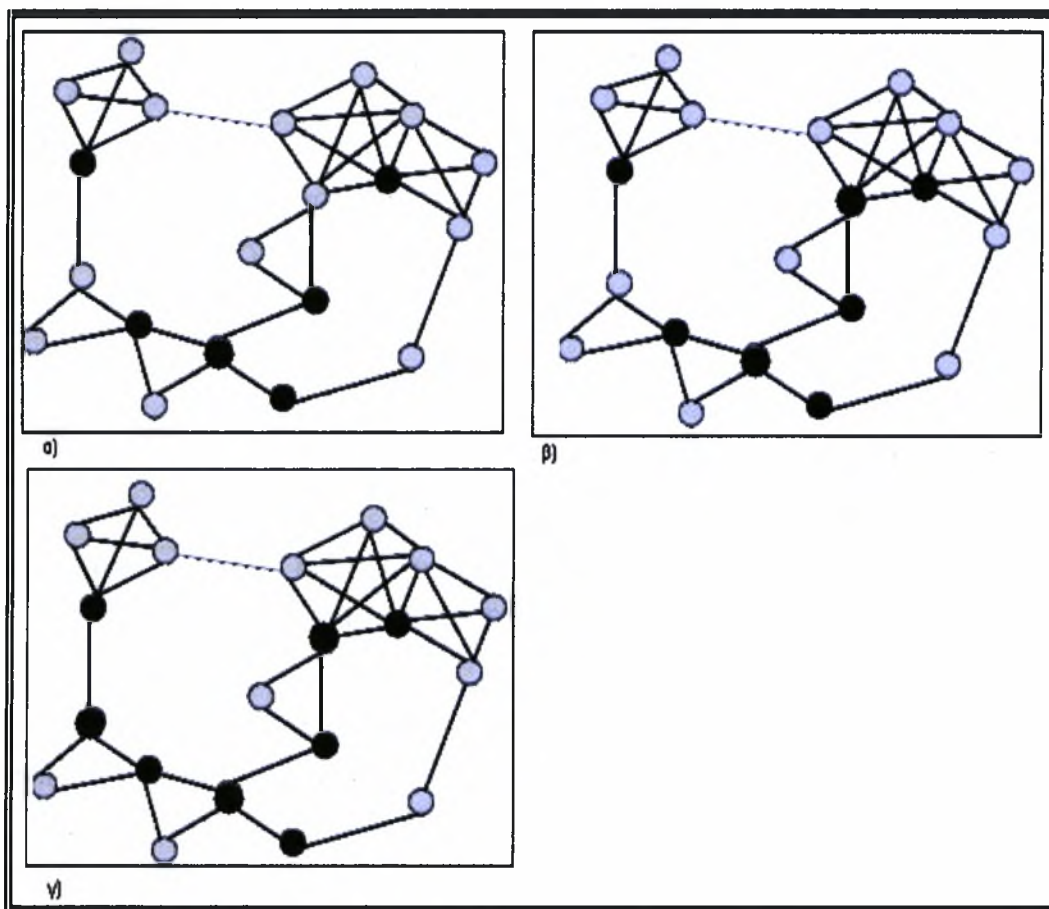
v)



g)



Εικόνα 11



Εικόνα 12

Στην Εικόνα 11 βλέπουμε την πρώτη φάση του αλγορίθμου. Αρχικά όλοι οι κόμβοι του δικτύου είναι λευκοί. Για ευκολία ο αριθμός που γράφεται σε κάθε κόμβο αναπαριστά τον αριθμό από pieces που έχει. Επιλέγουμε τον κόμβο με τον μεγαλύτερο αριθμό από pieces και τον χρωματίζουμε μαύρο (11.β) ενώ τους γείτονες του τους μαύρους(11.γ). Έπειτα ξαναυπολογίζουμε τον αριθμό pieces του κάθε κόμβου. Παρατηρούμε πως ο μαύρος με του γκρι λαμβάνεται σαν ένας κόμβος. Έπειτα επιλέγουμε τον κόμβο με τον μεγαλύτερο αριθμό από pieces τον χρωματίζουμε μαύρο, και η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου να μην υπάρχουν λευκοί κόμβοι. Από την διαδικασία αυτή παρατηρούμε πως το σύνολο των μαύρων κόμβων είναι dominating set όχι όμως connected-συνδεδεμένο.

Στην Εικόνα 12 εκτελούμε την δεύτερη φάση του αλγορίθμου κατά την οποία επιλέγουμε όσο το δυνατόν τον ελάχιστο αριθμό κόμβων έτσι ώστε τα dominating set της πρώτης φάσης να γίνει connected. Το κριτήριο επιλογής

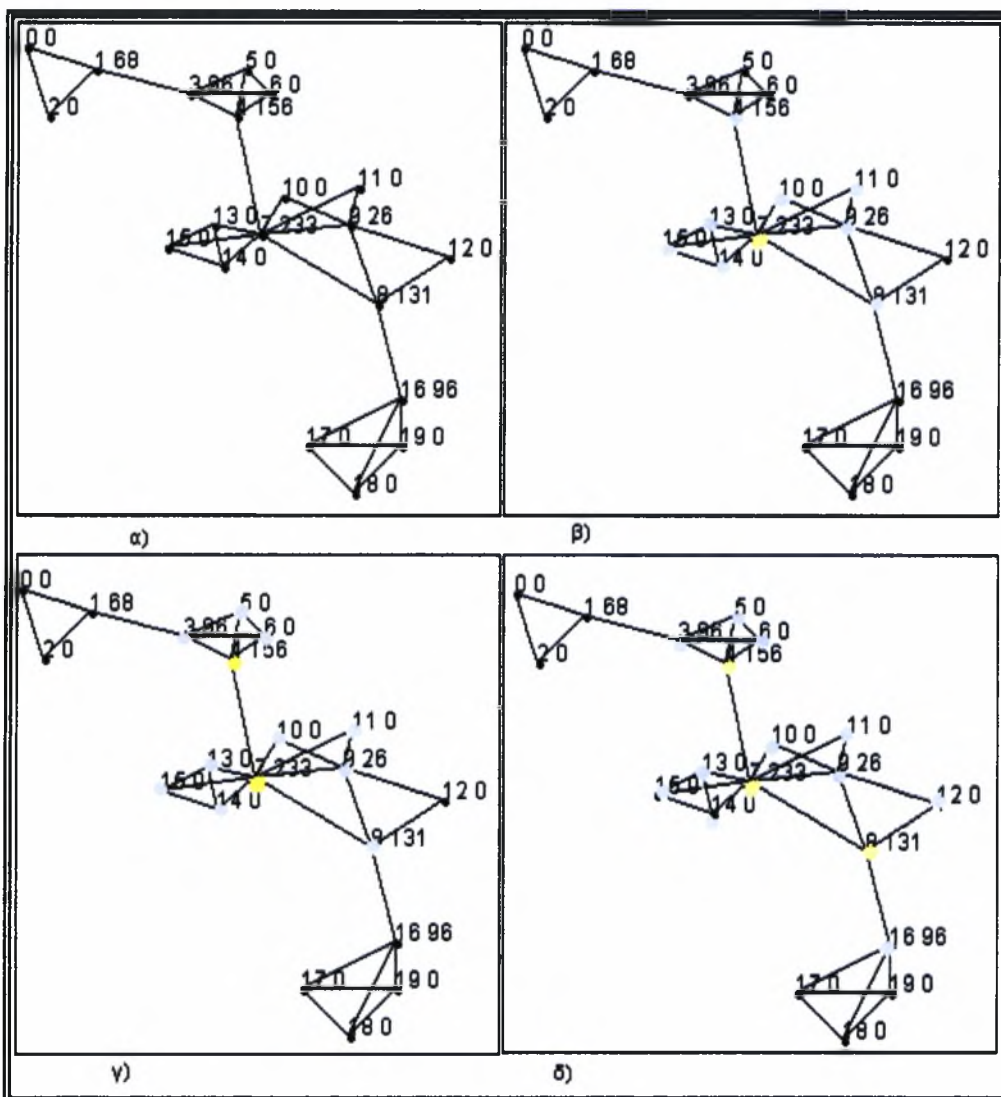
ενός κόμβου είναι ο αριθμός των μη συνδεδεμένων μαύρων κόμβων που βρίσκονται στην γειτονιά του και σε δεύτερο ρόλο είναι και ο αριθμός των μαύρων κόμβων που βρίσκονται στην '1-2' γειτονιά . Έτσι παρατηρούμε στο σχήμα 12.α επιλέξαμε τον κόμβο με του περισσότερους μαύρους μη-συνδεδεμένους γείτονες. Ομοίως και στο 12.β και στο 12.γ.

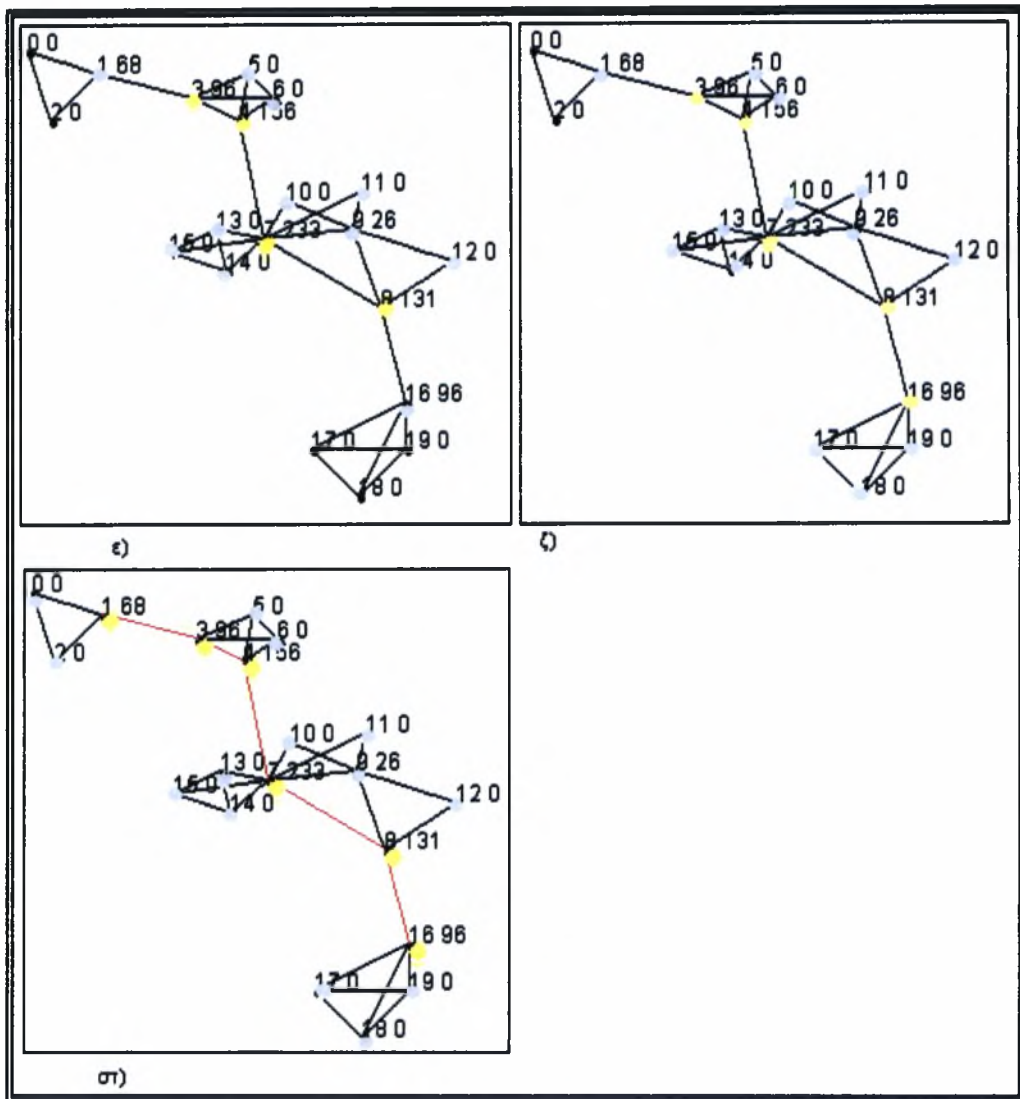
3.5. Node Importance Algorithm with color.

Ο αλγόριθμος αυτός αποτελεί έναν συνδυασμό των αλγορίθμων που περιγράψαμε στις ενότητες 3.1 και 3.2. Πιο συγκεκριμένα αποτελείται από τα εξής βήματα:

1. Αρχικά όλοι οι κόμβοι είναι λευκοί. Από τους λευκούς επιλέγουμε αυτόν με τον μεγαλύτερο Node Importance και τον χρωματίζουμε μαύρο(τον εισάγουμε στο connected dominating set) , ενώ τους γείτονες του χρωματίζουμε γκρι.
2. Επαναληπτικά μέχρι να μην υπάρχουν λευκοί κόμβοι επιλέγω από τους γκρι τον κόμβο με το μεγαλύτερο node importance και τον χρωματίζω μαύρο και τους γείτονες του γκρι.

Από ότι παρατηρούμε έχουμε ένα υβριδικό κεντρικοποιημένο αλγόριθμο, ο οποίος παίρνει το μέγεθος του Node Importance από τον αλγόριθμο 3.1 και το χρώμα από τον αλγόριθμο 3.2. Το υπογράφημα που δημιουργείται από τον αλγόριθμο καλύπτει όλο το γράφο-δίκτυο αφού κάθε γκρι κόμβος έχει γείτονα κόμβο του υπογράφου (μαύρο κόμβο) και οι μαύροι κόμβοι είναι μεταξύ τους ενωμένοι αφού το κριτήριο για να γίνει κάποιος κόμβος μαύρος είναι να είναι γκρι, οι οποίοι γκρι κόμβοι γειτονεύουν με τους είδη μαύρους. Συνεπώς το υπογράφημα από τους μαύρους κόμβους που παράγεται από τον αλγόριθμο είναι Connected Dominating Set.





Εικόνα 13

Στην παραπάνω εικόνα (εικόνα 13) βλέπουμε ένα παράδειγμα εφαρμογής του αλγορίθμου που περιγράψαμε παραπάνω. Στην αρχή όλοι οι κόμβοι είναι λευκοί. Επιλέγουμε τον κόμβο με το μεγαλύτερο Node Importance και τον χρωματίζουμε μαύρο ενώ τους γείτονες του τους χρωματίζουμε γκρι, όπως φαίνεται στην εικόνα 13.β. Έπειτα από το σύνολο των γκριζων κόμβων επιλέγω τον κόμβο με το μεγαλύτερο node importance και τον χρωματίζω μαύρο και τους λευκούς γείτονες του γκρι. Επαναλαμβάνω στις επόμενες εικόνες την διαδικασία αυτή μέχρι να μην υπάρχουν λευκοί κόμβοι. Από 'τι παρατηρούμε το αποτέλεσμα του αλγορίθμου είναι connected dominating set.

Τα αποτελέσματα του αλγορίθμου και η σειρά που επιλέχθηκαν οι κόμβοι στο παραπάνω γράφημα είναι ίδια με την σειρά του πρώτου αλγορίθμου –Node Importance algorithm (Ενότητα 3.2). Όντως από ότι θα δούμε και παρακάτω και οι δύο αλγόριθμοι επιλέγουν σχεδόν με την ίδια σειρά τους κόμβους για το Connected Dominating Set. Αυτό γίνεται αν παρατηρήσουμε πως στα περισσότερα δίκτυα η ποσότητα node importance στους κεντρικούς κόμβους είναι μεγάλη και όσο πηγαίνουμε στους ακριανούς κόμβους το Node Importance μειώνεται, το οποίο είναι λογικό αφού από τους κεντρικούς κόμβους διέρχονται περισσότερα μονοπάτια.

ΚΕΦΑΛΑΙΟ 4^ο

Περιγραφή των πειραμάτων

4.1. Εισαγωγή

Στο κεφάλαιο αυτό θα περιγράψω τα πειράματα που έκανα πάνω στους αλγόριθμους που αναλύσαμε στο Κεφάλαιο 3. Γενικά μας ενδιαφέρει ο αριθμός των κόμβων που υπολογίζει ο κάθε αλγόριθμος για το Connected Dominating Set και να δούμε ποια αλγόριθμου το σύνολο κόμβων τείνει να ανήκει στο υποσύνολο dominating set που είναι γνωστά ως minimum connected dominating set (οι κόμβοι που ανήκουν στο dominating set είναι connected και το γράφημα καλύπτεται από τον μικρότερο δυνατό αριθμό κόμβων). Με τον χρόνο κλιμάκωσης των αλγορίθμων, δηλαδή ποιος υπολογίζει το dominating set πιο γρήγορα δεν θα ασχοληθούμε. Τα πειράματα που κάνω με λίγα λόγια είναι τα εξής:

- Εφαρμογή των αλγορίθμων σε δίκτυα με κόμβους σε τυχαίες θέσεις και με τυχαία ακτίνα για τον καθένα. Θα αυξάνω κάθε φορά τους κόμβους και θα βλέπω πως κλιμακώνει κάθε αλγόριθμος.
- Εφαρμογή των αλγορίθμων σε δίκτυα με κόμβους στους οποίους θα είναι σε clusters. Σε αυτά τα δίκτυα θα μεταβάλλω εκτός από των αριθμών των κόμβων, των αριθμό των clusterheads και των κόμβων που πρόσκεινται σε κάθε clusterhead καθώς και τις αντίστοιχες ακτίνες.
- Θα έχω σταθερό αριθμό κόμβων και σταδιακά θα μεταβάλλω την ακτίνα-range ομοιόμορφα σε κάθε κόμβο.

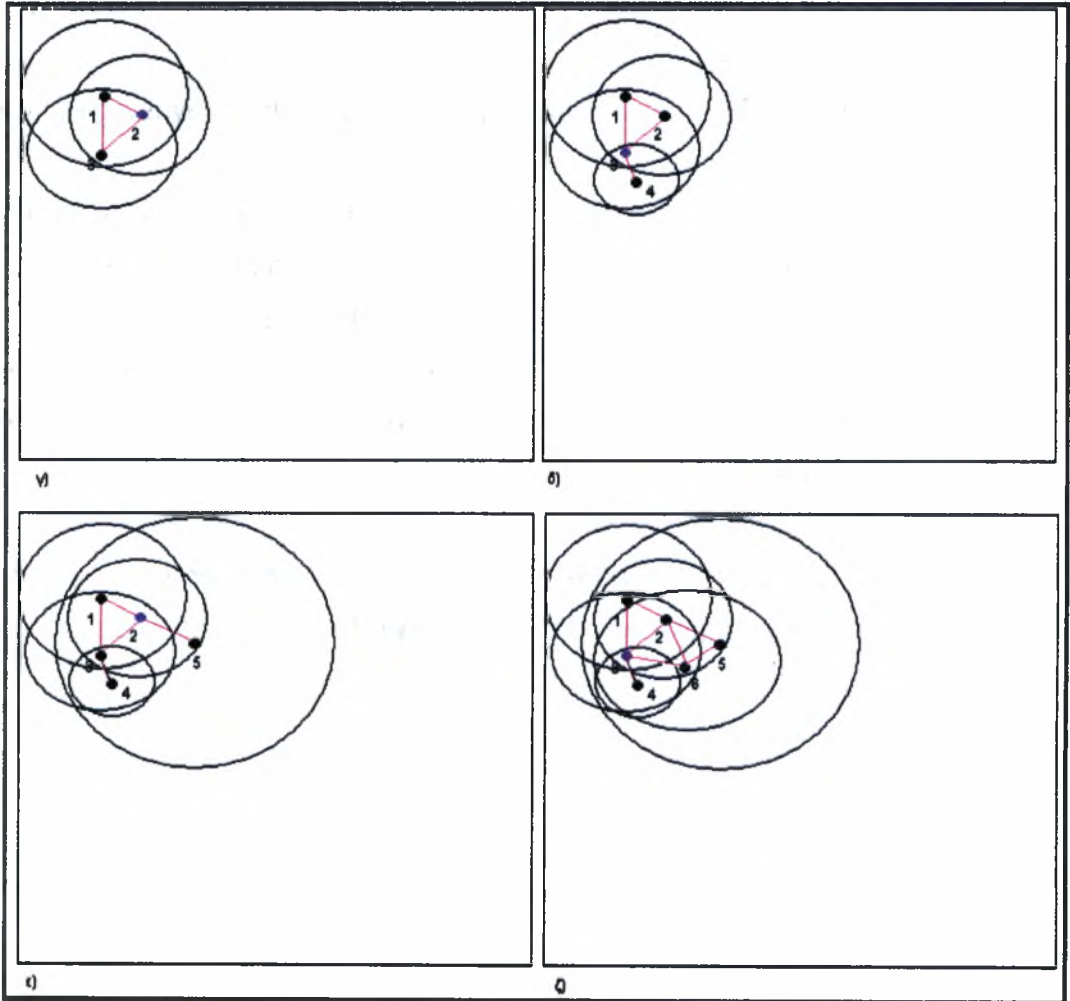
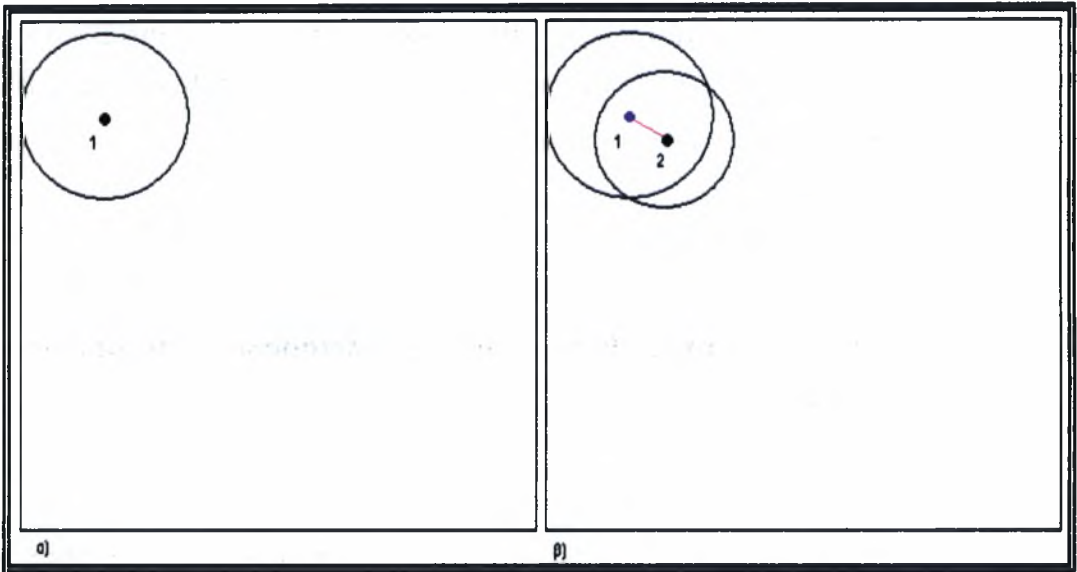
- Εφαρμογή των αλγορίθμων σε δίκτυα με κόμβους οι οποίοι θα έχουν ψευδοτυχαία θέση άλλα όμοια ακτίνα και θα μεταβάλλω σταδιακά των αριθμό των κόμβων.

4.2. Πείραμα 1: Τυχαία θέση – ακτίνα . Μεταβάλλω τον αριθμό των κόμβων

Όπως είπαμε παραπάνω σε πρώτη φάση θα συγκρίνω την επίδοση των αλγορίθμων ως προς τον αριθμό των κόμβων των *connected dominating set* που παράγουν σε δίκτυα των οποίων οι κόμβοι έχουν ψευδοτυχαίες συντεταγμένες και εμβέλεια-range. Πιο συγκεκριμένα τα δίκτυα αυτά με λίγα λόγια παράγονται ως εξής:

- Στην αρχή δημιουργώ ένα κόμβο με τυχαίες συντεταγμένες και τυχαία ακτίνα
- Έπειτα για να προσθέσω ένα καινούριο κόμβο στο γράφο επιλέγω τυχαία ένα κόμβο k από τους ήδη υπάρχοντες και θέτω ως συντρεγμένες του καινούριου κόμβου αριθμούς από τον κύκλο που ορίζει ο κόμβος k , που επέλεξα, και ως ακτίνα έναν αριθμό μεγαλύτερο ή ίσο με την απόσταση του σημείου που επέλεξα με την θέση του κόμβου k .

Παρακάτω δίνουμε ένα παράδειγμα που κάνει πιο κατανοητό την φιλοσοφία που περιγράψαμε παραπάνω στην δημιουργία δικτύου για το πείραμα.



Εικόνα 14

Στην παραπάνω εικόνα δίνουμε ένα παράδειγμα δημιουργίας ενός ψευδοτυχαίου δικτύου από έξι κόμβους. Στην εικόνα 14.α δημιουργούμε τον πρώτο κόμβο , του οποίου οι συντεταγμένες και η ακτίνα-range επιλέγονται τυχαία. Έπειτα στην εικόνα 14.β από τους είδη υπάρχοντες κόμβους (τον πρώτο δηλαδή) δημιουργούμε τον δεύτερο κόμβο του οποίου οι συντεταγμένες ανήκουν στον κύκλου (ο οποίος ορίζεται από τις συντεταγμένες και την ακτίνα) του κόμβου 1 και επιλέγουμε ως ακτίνα ένα τυχαίο αριθμό μεγαλύτερο ή ίσο από την απόσταση του από τον κόμβου 1. Στην εικόνα 14.γ επιλέγω (τυχαία) από το ήδη υπάρχον δίκτυο ένα κόμβο και προσθέτω τον κόμβο με id 3. Παρατηρούμε πως ο κόμβος εκτός από τον κόμβο με id 2 ενώνεται και με τον 1 αφού η ακτίνα που επιλέχθηκε περιλαμβάνει και τον πρώτο κόμβο. Ομοίως και στο σχήμα 14.δ. Στο επόμενο σχήμα παρατηρούμε πως αν και η ακτίνα του προς εισαγωγή κόμβου περιλαμβάνει όλους τους κόμβους ο νέος κόμβος συνδέεται με τους κόμβους που βρίσκεται στην ακτίνα.

Για να έχω πιο αντικειμενικά αποτελέσματα , και επειδή ο αριθμός των κόμβων ενός connected dominating set εξαρτάται από το πόσο 'πυκνό' ή 'αραιό' είναι το γράφημα για κάθε αριθμό κόμβων θα δημιουργώ ένα πλήθος γραφημάτων και σαν αποτέλεσμα του αριθμού των κόμβων του connected dominating set κάθε αλγορίθμου θα βάζω τον μέσο όρο από τα επιμέρους αποτελέσματα .

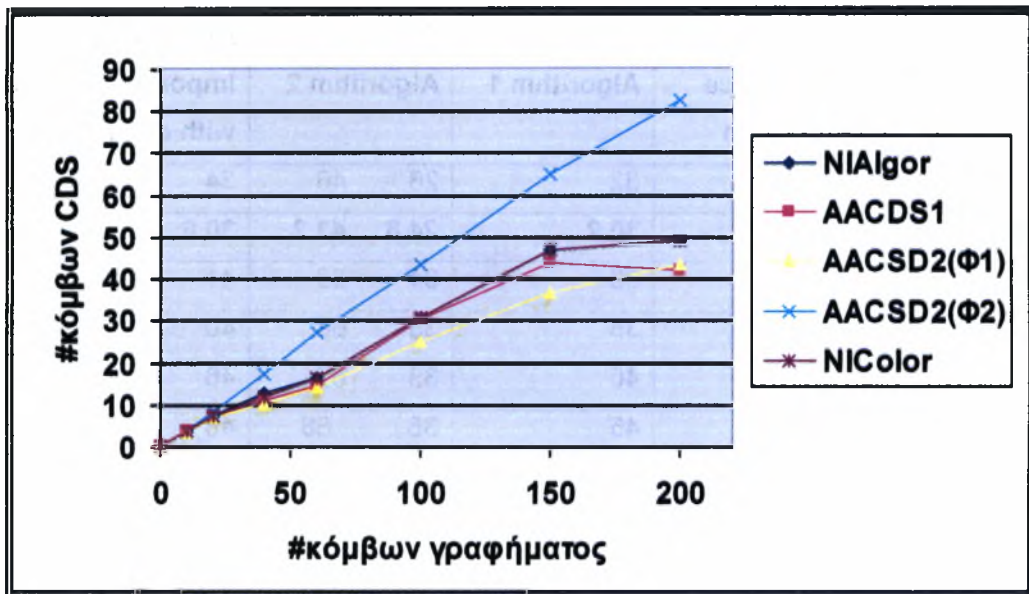
Στον πίνακα που ακολουθεί δίνω τα αποτελέσματα των πειραμάτων που έκανα για καθένα από τους αλγορίθμους που περιγράψαμε στο προηγούμενο κεφάλαιο. Η πρώτη στήλη περιγράφει τον αριθμό των κόμβων του γράφου, η δεύτερη το αποτέλεσμα του αλγορίθμου Node Importance algorithm for Connected Dominating sets που παρουσιάστηκε στην ενότητα 3.2, η τρίτη για τον αλγόριθμο Approximation Algorithms for Connected Dominating sets-Algorithm 1 που παρουσιάστηκε στην ενότητα 3.3, η τέταρτη για τον αλγόριθμο Approximation Algorithms for Connected Dominating sets-Algorithm 2 που παρουσιάστηκε στην ενότητα 3.4 .Για τον αλγόριθμο αυτό μελετώ των αριθμό των κόμβων που παράγει στην πρώτη φάση του αλγορίθμου και τον αριθμό των κόμβων που παράγει στην δεύτερη φάση. Η

πέμπτη για τον αλγόριθμο Node Importance with color algorithm που παρουσιάστηκε στην ενότητα 3.5 . Στην τελευταία στήλη υπάρχει το όνομα του αρχείου το οποίο περιέχει της συντεταγμένες και την εμβέλεια κάθε κόμβου σε μορφή που χρησιμοποιείται στο πρόγραμμα που εξομοιώνει το περιβάλλον ad-hoc δικτύων , στα οποίο εφαρμόζονται οι αλγόριθμοι(περισσότερα για το πειραματικό μέρος , δηλαδή για το πώς έκανα τα πειράματα περιγράφονται στο ΠΑΡΑΡΤΗΜΑ Α').

#Κόμβων Δικτύου	Node Importance Algorithm	AACDS Algorithm 1	AACDS Algorithm 2	Node Importance with color	Όνομα αρχείου για το δίκτυο
10	3	3	3 3	3	rand_net10_1.txt
10	4	4	4 4	4	rand_net10_2.txt
10	4	4	3 4	4	rand_net10_3.txt
M.O	3.66	3.66	3.33 3.66	3.66	
20	8	8	7 10	8	rand_net20_1.txt
20	8	8	7 8	8	rand_net20_2.txt
20	6	5	6 6	6	rand_net20_3.txt
M.O	7.33	7	6.66 8	7.33	
40	16	16	12 25	12	rand_net40_1.txt
40	12	10	10 18	12	rand_net40_2.txt
40	9	8	8 14	9	rand_net40_3.txt
40	13	9	9 12	13	rand_net40_4.txt
M.O	12.5	10.75	9.75 17.25	11.5	
60	15	14	14 26	15	rand_net60_1.txt
60	13	12	11 15	13	rand_net60_2.txt
60	20	15	17 25	20	rand_net60_3.txt
60	20	18	13 24	20	rand_net60_4.txt
60	14	13	14 18	14	rand_net60_5.txt
M.O	16.4	14.4	13.8 26.8	16.4	
100	30	30	23 43	30	rand_net100_1.txt
100	30	26	22 43	31	rand_net100_2.txt
100	33	31	29 45	33	rand_net100_3.txt
100	25	22	24 39	25	rand_net100_4.txt

#Κόμβων Δικτύου	Node Importance Algorithm	AACDS Algorithm 1	AACDS Algorithm 2	Node Importance with color	Όνομα αρχείου για το δίκτυο
100	34	32	26 46	34	rand_net100_5 .txt
M.O	30.4	30.2	24.8 43.2	30.6	
150	41	39	34 63	41	rand_net150_1 .txt
150	40	35	35 66	40	rand_net150_2 .txt
150	46	40	33 67	46	rand_net150_3 .txt
150	48	45	35 68	48	rand_net150_4 .txt
150	58	59	46 61	58	rand_net150_5 .txt
M.O	46.6	43.6	36.6 65	46.6	
200	51	43	45 82	51	rand_net200_1 .txt
200	43	38	38 92	43	rand_net200_2 .txt
200	44	37	40 93	44	rand_net200_2 .txt
200	43	33	33 66	43	rand_net200_3 .txt
200	64	59	59 78	64	rand_net200_5 .txt
M.O	49	42	43 82.2	49	

Μπορούμε να συνοψίσουμε τα αποτελέσματα του παραπάνω πίνακα στο γράφημα της εικόνας 15. Σε αυτό παρατηρούμε πως ο αλγόριθμος Approximation Algorithms for Connected Dominating sets-Algorithm 1 (AACDS1) δημιουργεί connected dominating set με τους λιγότερους κατά μέσω όρο κόμβους σε σύγκριση με τους υπόλοιπους αλγορίθμους. Έπειτα οι αλγόριθμοι Node Importance Algorithm και Node Importance algorithm with color απαιτούν λίγο περισσότερους κόμβους από ότι ο αλγόριθμος AACDS1. Τέλος την χειρότερη απόδοση την έχει ο αλγόριθμος Approximation Algorithms for Connected Dominating sets-Algorithm 2 (AACDS(Φ2)) , αν και κατά την πρώτη φάση του (AACDS2(Φ1)) δημιουργεί ένα dominating set με τους λιγότερους κόμβους.



Εικόνα 15

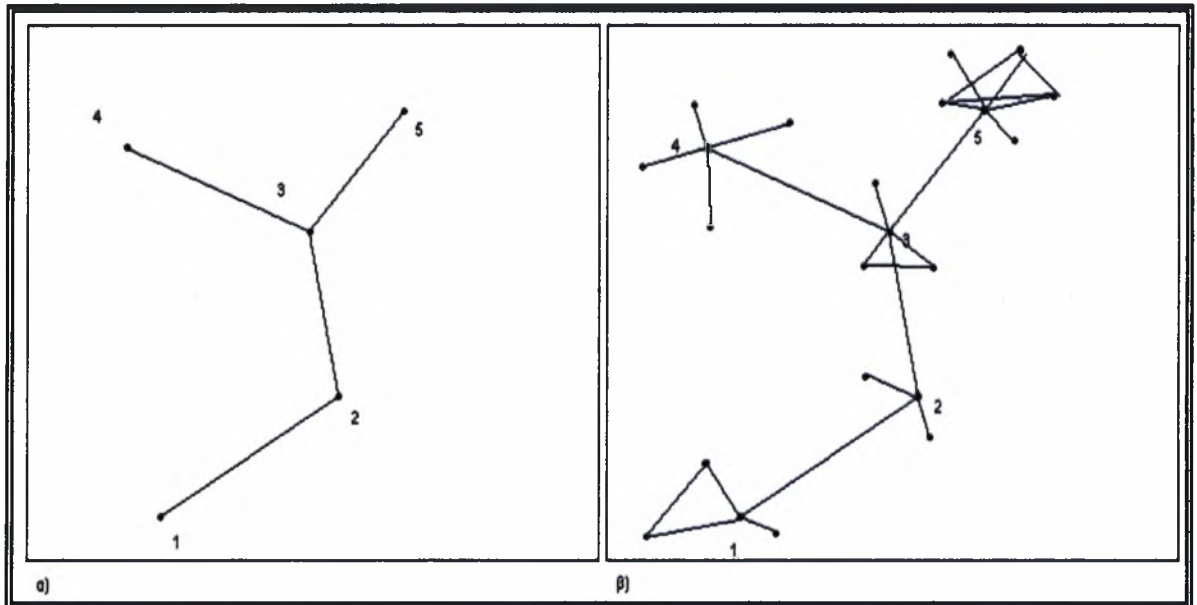
4.3. Πείραμα 2: Γραφήματα στα οποία οι κόμβοι είναι διατεταγμένοι σε clusters

Σε αυτό το πείραμα , όπως περιγράψαμε στην εισαγωγή αυτής της ενότητας , θα εξετάσω την επίδοση των αλγορίθμων που περιγράψαμε στην ενότητα 3 όταν εφαρμόζονται σε γραφήματα τα οποία είναι οργανωμένα σε clusters. Με τον όρο clusters εννοούμε ότι οι κόμβοι στο γράφημα είναι σε ομάδες , από τις οποίες ένα κόμβος ονομάζεται clusterhead. Κάθε clusterhead ενώνεται τουλάχιστον με ένα άλλο.

Πιο συγκεκριμένα η μεθοδολογία που χρησιμοποιώ για να δημιουργήσω γραφήματα με clusters είναι η εξής:

- Στην αρχή δημιουργώ τα clusterheads , τα οποία έχουν μια συγκεκριμένη ελάχιστη απόσταση d το ένα με το άλλο και range μεγαλύτερο από d .
- Για κάθε κόμβο-clusterhead που δημιούργησα δημιουργώ τους 'τοπικούς' κόμβους , οι οποίοι έχουν συντεταγμένες στον κύκλο που ορίζεται από το εκάστοτε clusterhead. Για κάθε τοπικό κόμβο πρέπει να προσέξουμε την επιλογή της εμβέλειας ώστε να μην έχει δύο clusterheads.

Παρακάτω δίνω ένα παράδειγμα δημιουργίας ενός τέτοιου γραφήματος στο οποίο φαίνονται τα δύο βήματα που περιγράψαμε παραπάνω.



Εικόνα 16

Στην εικόνα 15 βλέπουμε τα δύο βήματα που περιγράψαμε παραπάνω. Στην 15 α δημιουργούμε τα clusterheads και στην εικόνα 15 β προσθέτουμε σε κάθε clusterhead τους 'τοπικούς' κόμβους. Παρατηρούμε πως οι τοπικοί κόμβοι μπορούν να επικοινωνούν μεταξύ τους όπως βλέπουμε στους τοπικούς κόμβους του clusterhead 1, 3 και 5.

Τα πειράματα που θα κάνω δεν θα μεταβάλλω μόνο τον αριθμό των κόμβων αλλά και την αναλογία clusterhead- 'τοπικών κόμβων'. Για παράδειγμα για συνολικό αριθμό κόμβων 20 θα πειραματιστώ με γραφήματα με 2 clusterheads και 9 τοπικών κόμβων στο κάθε ένα, 4 clusterheads και 4 τοπικών κόμβων στο κάθε ένα κ.α. Ο παρακάτω πίνακας περιέχει τα αποτελέσματα από τα πειράματα και οι στήλες είναι στο ίδιο μοτίβο όπως στο πρώτο πείραμα με μια επιπλέον πληροφορία ότι ο αριθμός σε παρένθεση

στην πρώτη στήλη είναι ο αριθμός των clusterhead του γραφήματος και ο αριθμός στην παρένθεση την τελευταία στήλη (σε μερικές εγγραφές) είναι η απόσταση στον κόμβων-γειτόνων σε ένα clusterhead :

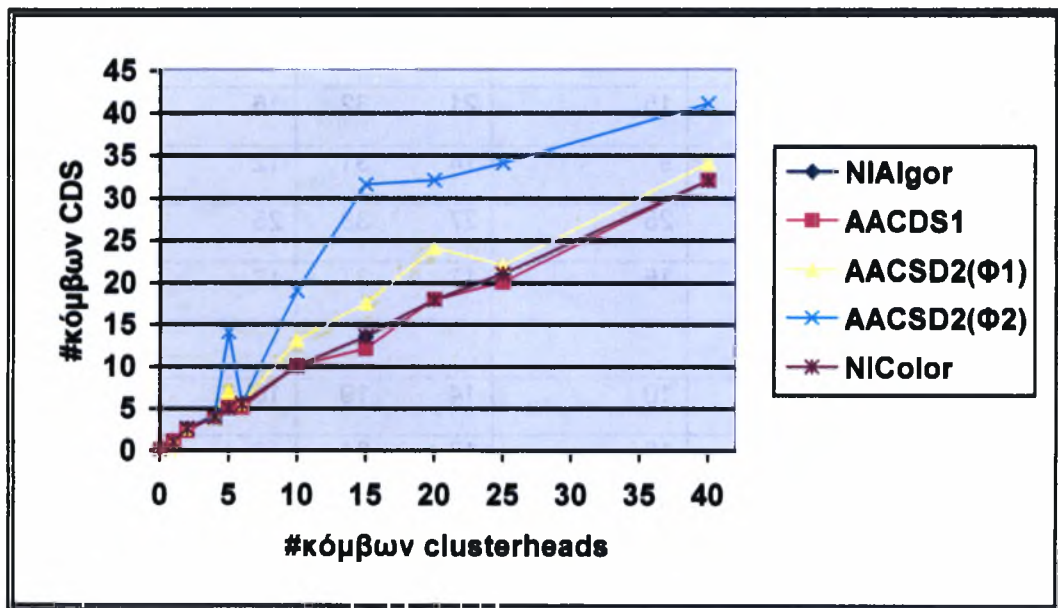
#Κόμβων Δικτύου	Node Importance Algorithm	AACDS Algorithm 1	AACDS Algorithm 2	Node Importance with color	Όνομα αρχείου για το δίκτυο
10 (1)	1	1	1 1	1	clust1_10.txt
10 (2)	3	3	3 3	3	clust2_5.txt
20 (1)	1	1	1 1	1	clust20_1.txt
20 (2)	2	2	2 2	2	clust20_2.txt
20 (4)	4	4	4 4	4	clust4_5.txt
20 (5)	4	5	4 4	4	clust4_5_1.txt
40 (1)	1	1	1 1	1	clust40_1.txt
40 (4)	4	4	4 4	4	clust4_10_1.txt
40 (4)	3	2	4 4	3	clust4_10_2.txt
40 (4)	4	4	4 4	4	clust4_10_3.txt
40 (8)	8	8	8 8	8	clust8_5_1.txt
40 (8)	5	5	6 6	5	clust8_5_2.txt
60 (1)	1	1	1 1	1	clust60_1.txt
60 (4)	4	4	4 4	4	clust15_4.txt
60 (4)	4	4	6 6	4	clust15_4_2.txt
60 (4)	3	3	3 3	3	clust15_4_3.txt
60 (6)	6	6	6 6	6	clust10_6_1.txt
60 (6)	5	4	5 5	5	clust10_6_2.txt

#Κόμβων Δικτύου	Node Importance Algorithm	AACDS Algorithm 1	AACDS Algorithm 2	Node Importance with color	Όνομα αρχείου για το δίκτυο
100 (2)	2	2	2 2	2	clust50_2.txt
100 (2)	2	2	2 2	2	clust50_2_1.txt
100 (2)	2	2	2 2	2	clust50_2_2.txt
100 (2)	4	2	4 2	4	clust50_2_3.txt
100 (2)	3	2	3 3	3	clust50_2_4.txt
100 (2)	2	2	2 2	2	clust50_2_5.txt
100 (5)	5	5	5 5	5	clust20_5_1.txt
100 (5)	5	5	5 11	5	clust20_5_2.txt
150 (5)	5	5	5 5	5	clust30_5_1.txt(50)
150 (5)	5	5	9 11	5	clust30_5_2.txt(100)
150 (5)	5	5	10 51	5	clust30_5_3.txt(150)
150 (10)	10	10	14 17	10	clust15_10_1.txt(50)
150 (10)	10	10	11 21	10	clust15_10_2.txt(100)
150 (10)	10	10	13 21	10	clust15_10_3.txt(100)
150 (15)	15	15	21 32	15	clust10_15_1.txt(50)
150 (15)	12	9	14 31	12	clust10_15_2.txt(120)
150 (25)	25	25	27 33	25	clust6_25_1.txt (50)
150 (25)	17	15	17 37	17	clust6_25_2.txt (100)
200 (10)	10	10	14 19	10	clust20_10_1.txt(50)
200 (10)	10	10	17 21	10	clust20_10_2.txt(100)
200 (20)	19	20	27 32	19	clust10_20_1.txt(50)
200 (20)	17	16	22 34	17	clust10_20_1.txt(100)
200 (40)	39	39	42 51	39	clust5_40_1.txt (50)
200 (40)	25	21	26 32	25	clust5_40_2.txt(100)

Με μια πιο προσεκτική ματιά στο παραπάνω πίνακα παρατηρούμε το εξής πόρισμα το οποίο θα εξηγήσουμε αργότερα στη εικόνα16:

Πόρισμα 4.1: Σε ένα γράφο με clusters ο αριθμός των κόμβων που ανήκουν στο connected dominating set , που δημιουργείται από τους αλγορίθμους που μελετήσαμε παραπάνω εξαρτάται κυρίως από τον αριθμό των κόμβων των clusterheads του δικτύου

Όντως αν κάνουμε την γραφική παράσταση με άξονα χ των αριθμό των clusterheads του γραφήματος και γ των αριθμό των κόμβων του connected dominating set παρατηρούμε που οι αλγόριθμοι Node Importance Algorithm, Node Importance with color και ο AACDS1 έχουν τον ίδιο αριθμό από κόμβους στο connected dominating set που παράγουν και είναι μικρότερος ή ίσος με τον αριθμό των clusterheads του γράφου. Αυτό από ότι βλέπουμε δεν ισχύει για τον AACDS2 αλγόριθμο ο οποίος έχει την χειρότερη επίδοση από τους υπόλοιπους.



Εικόνα 17

Παραπάνω είπαμε πως οι αλγόριθμοι Node Importance Algorithm, Node Importance with color και ο AACDS1 έχουν τον ίδιο αριθμό από κόμβους στο connected dominating set που παράγουν και είναι μικρότερος ή ίσος με τον αριθμό των clusterheads του γράφου. Για την ισότητα είναι προφανές αφού τα κάθε κόμβος του δικτύου ή θα γειτονεύει με ένα clusterhead ή θα είναι ο ίδιος clusterhead. Τα clusterheads ενώνονται μεταξύ τους . Συνεπώς τα clusterheads αποτελούν ένα connected dominating set. Για τον αλγόριθμο AACDS1 τα clusterheads αποτελούν του κόμβους με τους περισσότερους γείτονες και για τους αλγορίθμους με Node Importance από τα clusterheads διέρχονται τα περισσότερα μονοπάτια άρα έχουν και τις μεγαλύτερες node importance τιμές , άρα και στον ένα και στον άλλον θα επιλέγουν οι clusterheads θα επιλέγουν για το connected dominating set. Όταν δύο clusterheads είναι κοντά τότε μπορεί να καλύπτει το ένα clusterhead το άλλο και τους κόμβους του σε αυτή την περίπτωση το connected dominating set που επιλέγουν οι αλγόριθμοι αποτελείται από μικρότερο αριθμό κόμβων από των αριθμό των clusterheads.

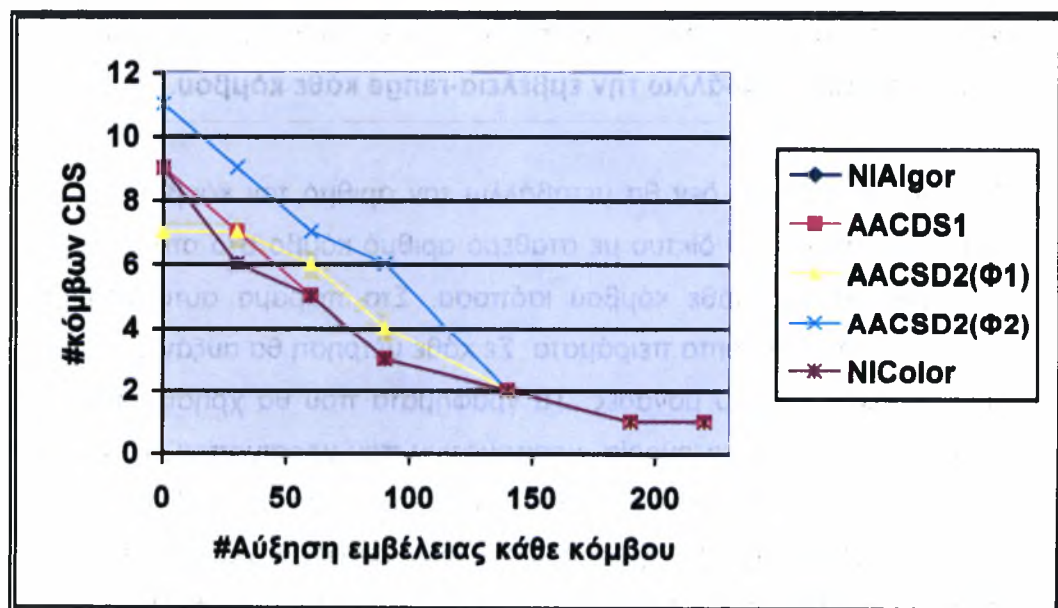
4.4. Πείραμα 3: Γραφήματα στα οποία ο αριθμός των κόμβων είναι σταθερός και μεταβάλλω την εμβέλεια-range κάθε κόμβου.

Σε αυτό το πείραμα δεν θα μεταβάλλω τον αριθμό των κόμβων στο δίκτυο, αλλά θα παίρνω ένα δίκτυο με σταθερό αριθμό κόμβο στο οποίο θα αυξάνω την εμβέλεια του κάθε κόμβου ισόποσα. Στο πείραμα αυτό χρησιμοποιώ δίκτυα από το υπόλοιπα πειράματα. Σε κάθε μέτρηση θα αυξάνω την εμβέλεια των κόμβων κατά 20 μονάδες. Τα γραφήματα που θα χρησιμοποιηθούν θα είναι δυο από κάθε κατηγορία γραφημάτων που χρησιμοποιώ στα υπόλοιπα πειράματα. Δηλαδή δύο γραφήματα με τυχαία θέση και εμβέλεια το ένα με είκοσι κόμβους και με πενήντα κόμβους, δύο γραφήματα με clusters το ένα με πέντε clusterheads και είκοσι κόμβους γείτονες στο κάθε clusterhead και το άλλο με δέκα clusterheads και πέντε κόμβους γείτονες στο κάθε clusterhead και δυο γραφήματα από το τέταρτο πείραμα δηλαδή με τυχαία θέση και αλλά με ίδια εμβέλεια για τον κάθε κόμβο (το ένα γράφημα με είκοσι κόμβους και το άλλο με πενήντα κόμβους).

Τα αποτελέσματα του πρώτου γραφήματος (γράφημα από κόμβους με τυχαία θέση και τυχαία εμβέλεια που αποτελείται από είκοσι κόμβους) είναι τα εξής:

Αύξηση εμβέλειας κάθε κόμβου	Node Importance Algorithm	AACDS Alg 1	AACDS Alg 2	Node Importance with color	Όνομα αρχείου για το δίκτυο
0	9	9	7 11	9	rand_net20_1p3.txt
+30	6	7	7 9	6	rand_net20_1p3_1.txt
+30	5	5	6 7	5	rand_net20_1p3_2.txt
+30	3	3	4 6	3	rand_net20_1p3_3.txt
+50	2	2	2 2	2	rand_net20_1p3_4.txt
+50	1	1	1 1	1	rand_net20_1p3_5.txt

Τα αποτελέσματα του παρακάτω πίνακα μπορούμε να τα συνοψίσουμε στο παρακάτω γράφημα:

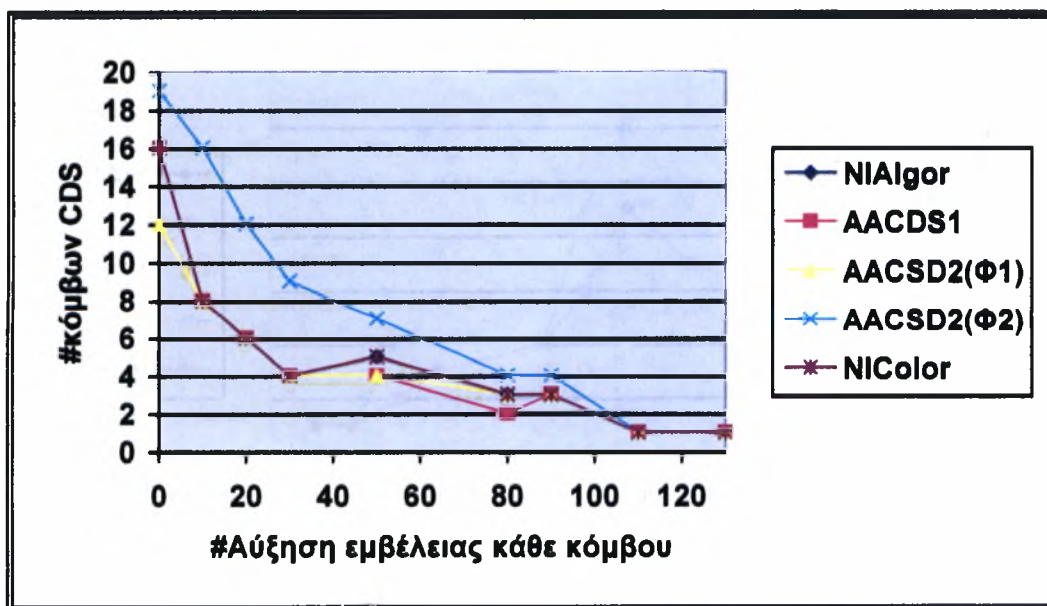


Εικόνα 18

Τα αποτελέσματα του δεύτερου γραφήματος (γράφημα από κόμβους με τυχαία θέση και τυχαία εμβέλεια που αποτελείται από πενήντα κόμβους) είναι τα εξής:

Αύξηση εμβέλειας κάθε κόμβου	Node Importance Algorithm	AACDS Alg 1	AACDS Alg 2	Node Importance with color	Όνομα αρχείου για το δίκτυο
0	16	16	12 19	16	rand_net50_1p3.txt
+10	8	8	8 16	8	rand_net50_1p3_1.txt
+10	6	6	6 12	6	rand_net50_1p3_2.txt
+10	4	4	4 9	4	rand_net50_1p3_3.txt
+20	5	4	4 7	5	rand_net50_1p3_4.txt
+30	3	2	3 4	3	rand_net50_1p3_5.txt
+10	3	3	3 4	3	rand_net50_1p3_6.txt
+20	1	1	1 1	1	rand_net50_1p3_7.txt

Τα αποτελέσματα του παρακάτω πίνακα μπορούμε να τα συνοψίσουμε στο παρακάτω γράφημα:

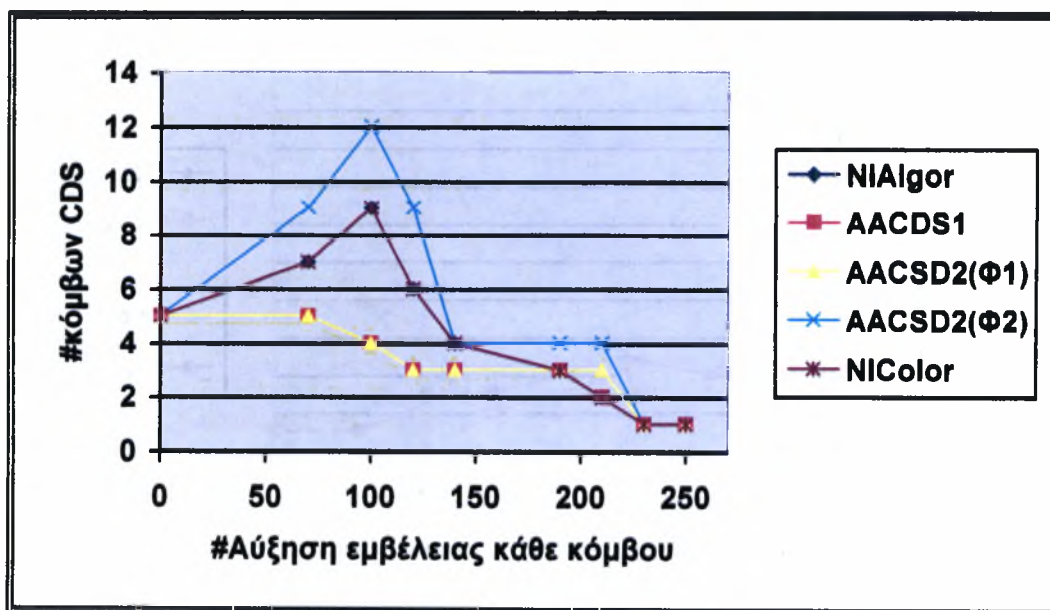


Εικόνα 19

Τα αποτελέσματα του τρίτου γραφήματος (γράφημα από κόμβους σε clusters με πέντε clusterheads και είκοσι κόμβους-γείτονες σε κάθε clusterhead) είναι τα εξής:

Αύξηση εμβέλειας κάθε κόμβου	Node Importance Algorithm	AACDS Alg 1	AACDS Alg 2	Alg	Node Importance with color	Όνομα αρχείου για το δίκτυο
0	5	5	5	5	5	clust20_5p3.txt
+70	7	5	5	9	7	clust20_5p3_1.txt
+30	9	4	4	12	9	clust20_5p3_2.txt
+20	6	3	3	9	6	clust20_5p3_3.txt
+20	4	3	3	4	4	clust20_5p3_4.txt
+50	3	3	3	4	3	clust20_5p3_5.txt
+20	2	2	3	4	2	clust20_5p3_6.txt
+20	1	1	1	1	1	clust20_5p3_7.txt

Τα αποτελέσματα του παρακάτω πίνακα μπορούμε να τα συνοψίσουμε στο παρακάτω γράφημα:

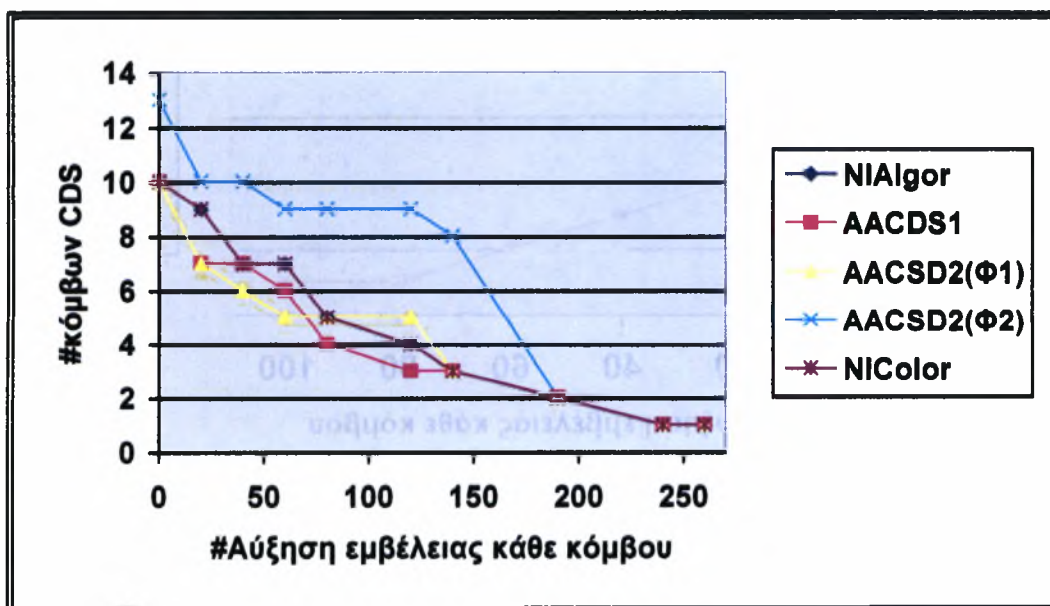


Εικόνα 20

Τα αποτελέσματα του τέταρτου γραφήματος (γράφημα από κόμβους σε clusters με δέκα clusterheads και πέντε κόμβους-γείτονες σε κάθε clusterhead) είναι τα εξής:

Αύξηση εμβέλειας κάθε κόμβου	Node Importance Algorithm	AACDS Alg 1	AACDS Alg 2	Alg	Node Importance with color	Όνομα αρχείου για το δίκτυο
0	10	10	10	13	10	clust5_20p3.txt
+20	9	7	7	10	9	clust5_20p3_1.txt
+20	7	7	6	10	7	clust5_20p3_2.txt
+20	7	6	5	9	7	clust5_20p3_3.txt
+20	5	4	5	9	5	clust5_20p3_4.txt
+40	4	3	5	9	4	clust5_20p3_5.txt
+20	3	3	3	8	3	clust5_20p3_6.txt
+50	2	2	2	2	2	clust5_20p3_7.txt
+50	1	1	1	1	1	clust5_20p3_8.txt

Τα αποτελέσματα του παρακάτω πίνακα μπορούμε να τα συνοψίσουμε στο παρακάτω γράφημα:

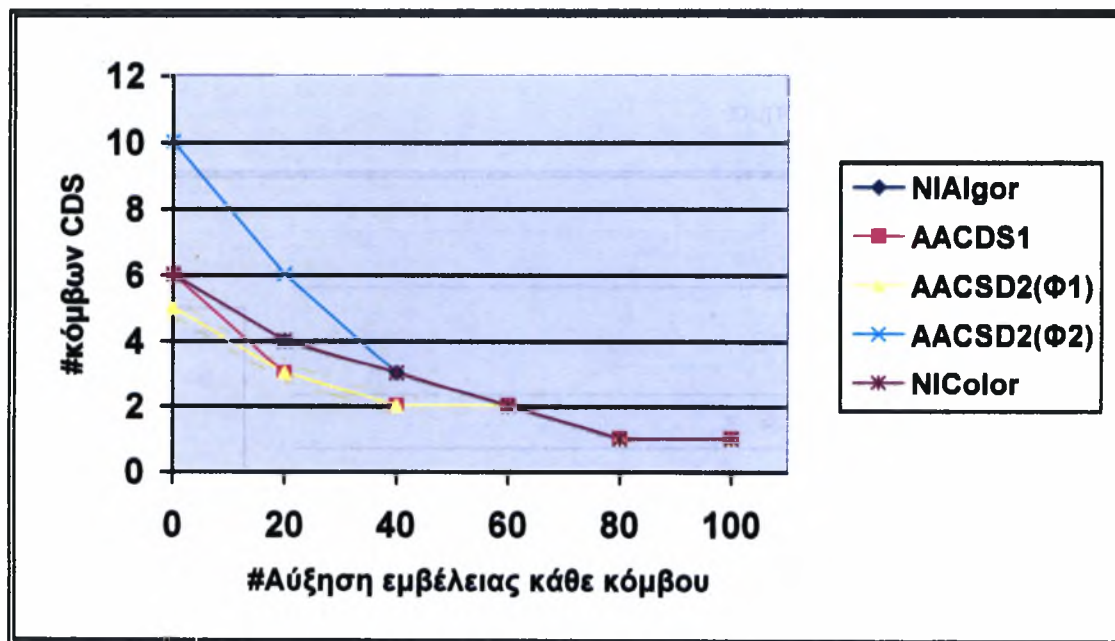


Εικόνα 21

Τα αποτελέσματα του πέμπτου γραφήματος (γράφημα από κόμβους με τυχαία θέση και κοινή εμβέλεια το οποίο αποτελείται από είκοσι κόμβους) είναι τα εξής:

Αύξηση εμβέλειας κάθε κόμβου	Node Importance Algorithm	AACDS Algorithm 1	AACDS Algorithm 2	Node Importance with color	Όνομα αρχείου για το δίκτυο
0	6	6	5 10	6	same_range20_1p3.txt
+20	4	3	3 6	4	same_range20_1p3_1.txt
+20	3	2	2 3	3	same_range20_1p3_2.txt
+20	2	2	2 2	2	same_range20_1p3_3.txt
+20	1	1	1 1	1	same_range20_1p3_4.txt

Τα αποτελέσματα του παρακάτω πίνακα μπορούμε να τα συνοψίσουμε στο παρακάτω γράφημα:

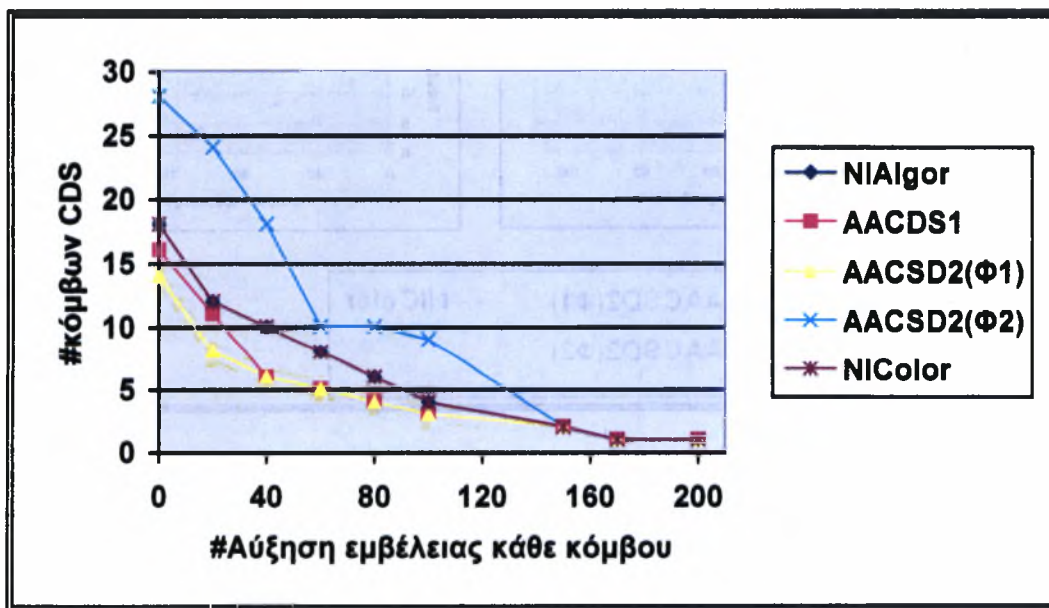


Εικόνα 22

Τα αποτελέσματα του έκτου γραφήματος (γράφημα από κόμβους με τυχαία θέση και κοινή εμβέλεια το οποίο αποτελείται από είκοσι κόμβους) είναι τα εξής:

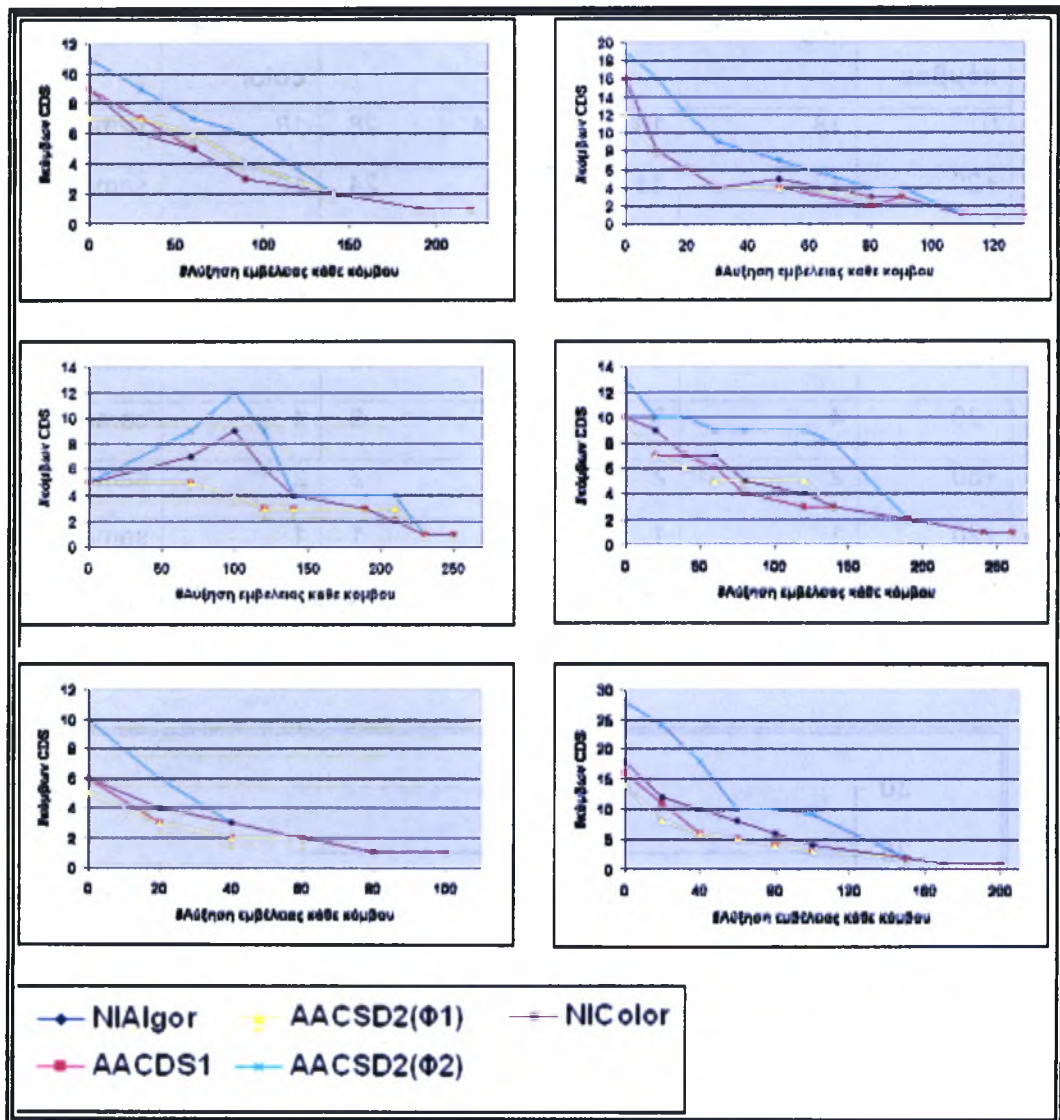
Αύξηση εμβέλειας κάθε κόμβου	Node Importance Algorithm	AACDS Algorithm 1	AACDS Algorithm 2	Node Importance with color	Όνομα αρχείου για το δίκτυο
0	18	16	14 28	18	same_range50_1p3.txt
+20	12	11	8 24	12	same_range50_1p3_1.txt
+20	10	6	6 18	10	same_range50_1p3_2.txt
+20	8	5	5 10	8	same_range50_1p3_3.txt
+20	6	4	4 10	6	same_range50_1p3_4.txt
+20	4	3	3 9	4	same_range50_1p3_5.txt
+50	2	2	2 2	2	same_range50_1p3_6.txt
+20	1	1	1 1	1	same_range50_1p3_7.txt

Τα αποτελέσματα του παρακάτω πίνακα μπορούμε να τα συνοψίσουμε στο παρακάτω γράφημα:



Εικόνα 23

Στην παρακάτω εικόνα έχω προσθέσει όλα τα παραπάνω γραφήματα που παρήγαγα από τα επιμέρους παραπάνω πειράματα από τα οποία θα προσπαθήσουμε να εξάγουμε κάποια συλλογικά συμπεράσματα για το πώς κλιμακώνουν οι αλγόριθμοι σε ένα γράφω στον οποίο σταδιακά αυξάνουμε την ακτίνα των κόμβων του.



Εικόνα 24

Καταρχάς με μια πρώτη ματιά παρατηρούμε πως σε όλα τα γραφήματα αυξάνοντας την εμβέλεια των κόμβων όλοι οι αλγόριθμοι έχουν σαν αποτέλεσμα ένα κόμβο στο connected dominating set που παράγουν.

Από αυτό το συμπέρασμα μπορούμε να βγάλουμε το παρακάτω πόρισμα:

Πόρισμα 4.2: Όλοι οι αλγόριθμοι, που μελετήσαμε, σε ένα δίκτυο στο οποίο αυξάνουμε την εμβέλεια των κόμβων από ένα σημείο και μετά παράγουν *connected dominating set* το οποίο αποτελείται από ένα κόμβο

Το παραπάνω πόρισμα είναι προφανές αφού αυξάνοντας την εμβέλεια των κόμβων μετά από κάποιο σημείο θα υπάρχει ένας κόμβος u ο οποίος θα έχει τόσο ευρύ range που θα καλύπτει όλο το γράφο και κάθε άλλος κόμβος θα έχει αντίστοιχα τόσο ευρύ range που θα 'βλέπουν' τον u . Άρα ο κόμβος u θα καλύπτει όλο το γράφημα και θα αποτελεί *connected dominating set*. Οι αλγόριθμοι που επιλέγουν κόμβους με βάση τους γείτονες κάποιου κόμβου (AACDS1 και AACDS2) θα επιλέξουν τον κόμβο u , ο οποίος προφανώς έχει τους περισσότερους γείτονες και θα χρωματίσει την γειτονιά του γκρι (δηλαδή όλους τους υπόλοιπους κόμβους) και θα τερματίσει. Ομοίως ο κόμβος u στους αλγορίθμους με Node Importance θα έχει το μεγαλύτερο node importance και θα επιλεγεί πρώτος.

Γενικότερα παρατηρούμε πως αυξάνοντας την εμβέλεια ο AACDS1 αλγόριθμος επιλέγει κάθε φορά του λιγότερους κόμβους για να σχηματιστεί το *connected dominating set*. Έπειτα οι αλγόριθμοι που χρησιμοποιούν την ποσότητα Node Importance για να επιλέξουν τους κόμβους έχουν την ίδια επίδοση. Τέλος ο αλγόριθμος AACDS2 έχει την χειρότερη επίδοση.

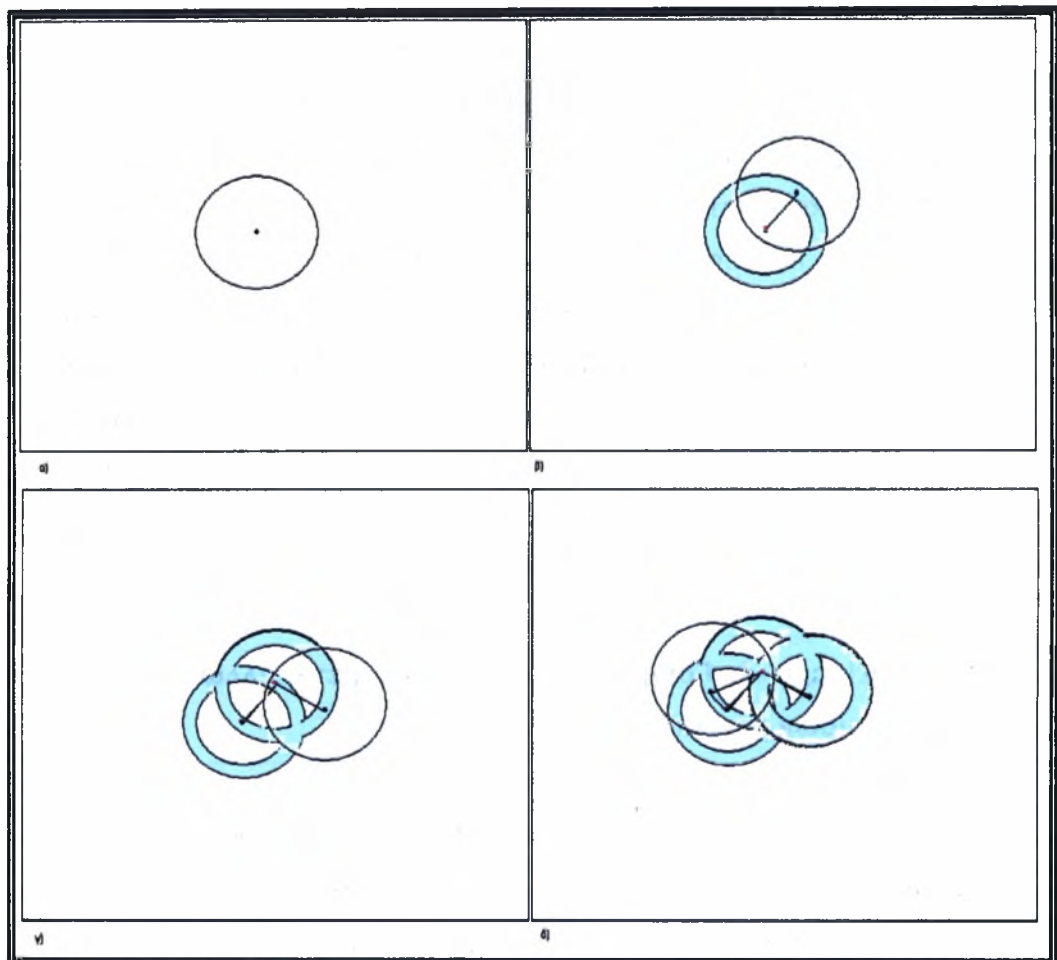
4.5. Πείραμα 4: Γραφήματα στα οποία μεταβάλλω των αριθμό των κόμβων με σταθερό range.

Σε αυτό το πείραμα αυτό μεταβάλλω των αριθμό των κόμβων των γραφημάτων. Οι θέσεις των κόμβων επιλέγονται ψευδοτυχαία, αλλά η εμβέλεια-range για κάθε κόμβο είναι προκαθορισμένη.

Πιο συγκεκριμένα η μεθοδολογία που χρησιμοποιώ για να δημιουργώ τέτοιου είδους γραφήματα είναι η εξής:

- Επιλέγω ένα αριθμό που αντιπροσωπεύει την κοινή εμβέλεια d που έχουν οι κόμβοι
- Επιλέγω τυχαία ένα κόμβο από τους ήδη υπάρχοντες (για τον πρώτο κόμβο επιλέγω την θέση του τυχαία) και με βάση την περιοχή που 'εκπέμπει' επιλέγω την θέση για τον νέο κόμβο. Την καινούρια θέση την επιλέγω ψευδοτυχαία ώστε να ανήκει στον δακτύλιο που δημιουργείται από τους κύκλους με ακτίνες $2/3 d$ και d .
- Το παραπάνω βήμα το επαναλαμβάνω μέχρι να συμπληρωθεί ο επιθυμητός αριθμός από

Παρακάτω δίνω ένα παράδειγμα που δείχνω που δείξω πως δημιουργείται ένα τέτοιο δίκτυο



Εικόνα 24

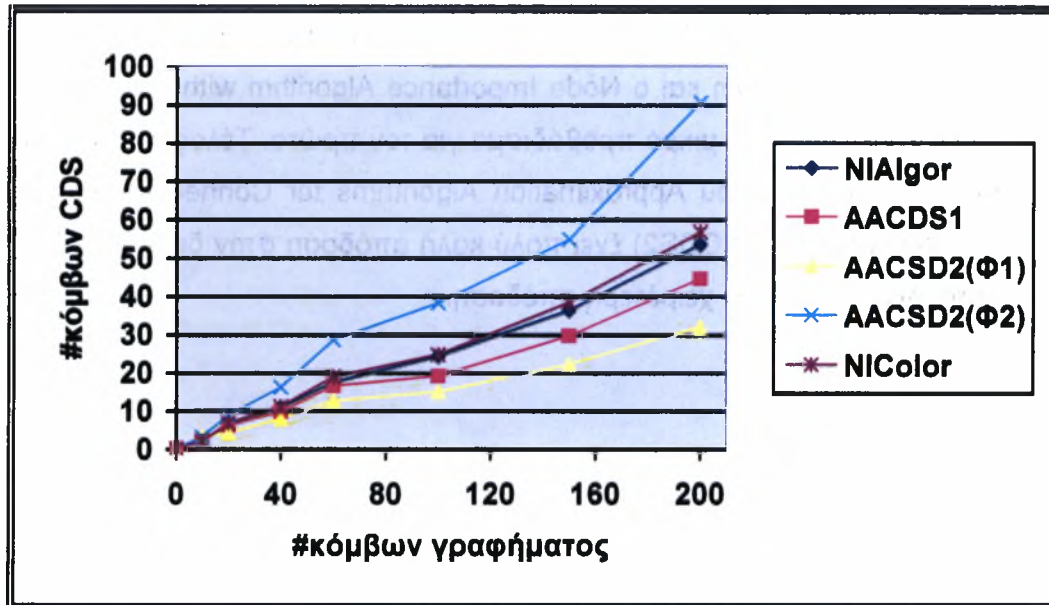
Στην παραπάνω εικόνα αναπαριστάμε το πώς δημιουργείται ένα γράφημα του τύπου που περιγράψαμε παραπάνω. Στην εικόνα 16.α ο πρώτος κόμβος δημιουργείται αυθαίρετα σε ψευδοτυχαία θέση με range d . Στην 16.β επιλέγουμε στον δακτύλιο που σχηματίζεται από τους κύκλους με ακτίνες $2/3 d$ και d ένα σημείο στο οποίο δημιουργούμε τον δεύτερο κόμβο με την ίδια ακτίνα μήκους d με τον πρώτο. Ομοίως και στα σχήματα 16 γ και δ. Στο τελευταίο παρατηρούμε πως ο καινούριος κόμβος ενώ έχει την απαιτούμενη απόσταση ($>2/3d$) από τον 'πατρικό' κόμβο συνδέεται επίσης και με άλλον κόμβο με μικρότερη απόσταση από $2/3 d$.

Παρακάτω δίνω ένα πίνακα με τα αποτελέσματα του πειράματος. Οι στήλες είναι οι ίδιες με αυτές στα πειράματα 1 και 2. Σε κάθε δίκτυο θα μεταβάλλω τον αριθμό των κόμβων ενώ το range του κάθε κόμβου θα είναι ίδιο για κάθε δίκτυο που χρησιμοποιείται για το πείραμα.

#Κόμβων Δικτύου	Node Importance Algorithm	AACDS Algorithm 1	AACDS Algorithm 2	Node Importance with color	Όνομα αρχείου για το δίκτυο
10	2	2	4 4	2	same_range10_1.txt
10	2	2	2 2	2	same_range10_2.txt
M.O	2	2	3 3	2	
20	5	3	3 5	5	same_range20_1.txt
20	10	10	5 13	10	same_range20_2.txt
20	5	5	4 8	5	same_range20_3.txt
M.O	6.66	6	4 8.66	6.66	
40	8	7	5 13	8	same_range40_1.txt
40	8	7	5 15	8	same_range40_2.txt
40	16	16	12 17	17	same_range40_3.txt
40	10	7	8 16	10	same_range40_4.txt
40	12	11	9 19	12	same_range40_5.txt
M.O	10.8	9.6	7.8 16	11	

#Κόμβων Δικτύου	Node Importance Algorithm	AACDS Algorithm 1	AACDS Algorithm 2	Node Importance with color	Όνομα αρχείου για το δίκτυο	
60	20	22	13	35	24	same_range60_1.txt
60	14	14	9	36	16	same_range60_2.txt
60	12	13	11	26	13	same_range60_3.txt
60	22	20	16	41	22	same_range60_4.txt
60	19	13	13	24	19	same_range60_5.txt
M.O	17.4	16.4	12.4	28.4	18.8	
100	21	16	14	16	21	same_range100_1.txt
100	22	15	12	31	23	same_range100_2.txt
100	23	18	15	43	23	same_range100_3.txt
100	24	19	15	44	24	same_range100_4.txt
100	31	27	18	56	32	same_range100_5.txt
M.O	24.2	19	14.8	38	24.6	
150	36	24	21	43	36	same_range150_1.txt
150	31	23	19	54	32	same_range150_2.txt
150	35	24	20	46	36	same_range150_3.txt
150	48	48	30	74	55	same_range150_4.txt
150	31	28	20	56	32	same_range150_5.txt
M.O	36.2	29.4	22	54.6	38.2	
200	52	46	32	87	55	same_range200_1.txt
200	51	37	27	88	53	same_range200_2.txt
200	53	46	33	95	58	same_range200_3.txt
200	52	37	29	86	55	same_range200_4.txt
200	59	56	38	96	62	same_range200_5.txt
M.O	53.4	44.4	31.8	90.4	56.6	

Παρακάτω έχουμε συνοψίσει τα αποτελέσματα του παραπάνω πίνακα στο γράφημα που φαίνεται στην εικόνα 25. Τα συμπεράσματα που βγάζουμε από το γράφημα είναι παρόμοια με αυτά στο πρώτο πείραμα (γραφήματα με τυχαία θέση και τυχαία ακτίνα).



Εικόνα 25

Συγκεκριμένα ο αλγόριθμος AACDS1 κλιμακώνει καλύτερα από τους υπόλοιπους. Οι αλγόριθμοι node importance algorithm και ο υβριδικός node importance with color έχουν σχεδόν την ίδια απόδοση με ένα μικρό προβάδισμα στον πρώτο. Τέλος ο AACDS2 αλγόριθμος αν και το dominating set που σχηματίζει κατά την πρώτη φάση έχει καλή απόδοση κατά την δεύτερη φάση προστίθενται πολλοί κόμβοι.

4.6. Κάποια γενικά συμπεράσματα για τα αποτελέσματα των παραπάνω πειραμάτων.

Από τα παραπάνω πειράματα μπορούμε να εξάγουμε κάποια συμπεράσματα. Γενικά σε ένα γράφημα ο Approximation Algorithms for Connected Dominating sets-Algorithm 1 έχει την καλύτερη απόδοση. Έπειτα ο Node Importance Algorithm και ο Node Importance Algorithm with color έχουν την ίδια απόδοση με ένα μικρό προβάδισμα για τον πρώτο. Τέλος αν και η πρώτη φάση του αλγορίθμου Approximation Algorithms for Connected Dominating sets-Algorithm 2 (AACDS2) έχει πολύ καλή απόδοση στην δεύτερη φάση του καταλήγει να έχει την χειρότερη απόδοση.

ΚΕΦΑΛΑΙΟ 5^ο

Αποτίμηση και μελλοντική εργασία

5.1. Εισαγωγή

Στο κεφάλαιο αυτό θα δούμε τι είπαμε στα προηγούμενα κεφάλαια. Δηλαδή θα συνοψίσουμε τους αλγόριθμους, τα πειράματα και τα αποτελέσματα που μελετήσαμε αναλυτικά στα προηγούμενα κεφάλαια, έτσι ώστε ο αναγνώστης να εμπεδώσει πιο εύκολα τις έννοιες και γενικότερα όσα είπα παραπάνω.

5.2. Οι αλγόριθμοι που μελετήσαμε

Στην εργασία αυτή ασχοληθήκαμε με αλγόριθμους που μπορούμε να τους εφαρμόσουμε σε ένα MANET (Mobile Ad hoc Network) με σκοπό να επιλέξουμε ένα σύνολο από κόμβους οι οποίοι θα καλύπτουν όλο το δίκτυο. Το σύνολο αυτό το ονομάσαμε *connected dominating set* (Κυρίαρχο σύνολο γραφήματος). Με ένα τέτοιο σύνολο κόμβων μπορούμε να υλοποιήσουμε πανεκπομπή (broadcast) στο δίκτυο. Αυτό που μελετήσαμε στην εργασία είναι το ποιος αλγόριθμος επιλέγει τους λιγότερους κόμβους στο CDS (*connected dominating set*) που παράγει.

Οι αλγόριθμοι που μελετήσαμε επιλέγουν τους κόμβους με κεντρικοποιημένο τρόπο. Αυτό σημαίνει ότι οι κόμβοι επιλέγονται ανάλογα με το ποιους ο κάθε αλγόριθμος θεωρεί ποιο 'σημαντικούς'. Γενικά από τους αλγόριθμους που είδαμε οι δύο χρησιμοποιούν μια ποσότητα το `node importance` για να ταξινομήσουν τους κόμβους (η ποσότητα αυτή εξαρτάται από τα πόσα μονοπάτια περνάνε από τον κάθε κόμβο) και οι άλλοι δύο χρησιμοποιούν το πλήθος των γειτόνων του κάθε κόμβου.

Ποιο συγκεκριμένα ο πρώτος αλγόριθμος που είδαμε στο Κεφάλαιο 3 είναι ο `Node importance Algorithm` ο οποίος αφού υπολογίσει την ποσότητα `Node importance` για κάθε κόμβο επιλέγει αυτούς με το μεγαλύτερο `Node Importance` για να υπολογίσει το CDS. Έπειτα είδαμε τους `Approximation Algorithms for Connected Dominating sets-Algorithm 1` και `Algorithm 2 (AACDS 1 & 2)` οι οποίοι παράγουν το CDS ανάλογα με το πλήθος των γειτόνων που έχει ο κάθε κόμβος στο γράφημα. Οι δύο αυτοί αλγόριθμοι είδαμε ότι βασίζονται στον χρωματισμό των κόμβων, οι οποίοι ανάλογα με το αν ο κόμβος ανήκει ή αν καλύπτεται ή όχι από το CDS παίρνει και το ανάλογο χρώμα (μαύρος, γκρι, λευκός). Τέλος είδαμε ένα υβριδικό αλγόριθμο τον `Node Importance with color` ο οποίος δανείζεται από τον πρώτο το μέγεθος `Node Importance` και από τους άλλους τον χρωματισμό των κόμβων.

5.3. Τα πειράματα που έγιναν

Στο Κεφάλαιο 4 πειραματιστήκαμε με τους αλγορίθμους που μελετήσαμε σε πολλά γραφήματα. Δοκιμάσαμε αυτούς τους αλγορίθμους σε τέσσερα πειράματα. Το πρώτο πείραμα είναι να δούμε πως κλιμακώνουν οι αλγόριθμοι σε δίκτυα με τυχαίες συντεταγμένες και τυχαία εμβέλεια. Το δεύτερο ήταν σε δίκτυα το οποία είναι σχηματισμένα σε clusters. Στο τρίτο πείραμα είδαμε πως κλιμακώνουν οι αλγόριθμοι σε γραφήματα με σταθερό αριθμό κόμβων και σταθερή θέση στους οποίους σταδιακά αυξάνουμε την εμβέλεια τους. Τέλος στο τελευταίο πείραμα χρησιμοποιήσαμε δίκτυα με κόμβους με τυχαία θέση αλλά ίδια εμβέλεια. Για κάθε πείραμα δημιουργήσαμε πολλά γραφήματα και έτσι ώστε να έχουμε ποιο αντικειμενικά αποτελέσματα.

5.4. Τα αποτελέσματα

Από τα πειράματα που κάναμε στο κεφάλαιο 4 βγάλαμε κάποια συμπεράσματα και πορίσματα για τους αλγορίθμους που μελετούμε. Τα δύο πορίσματα που παράχθηκαν από τα συμφραζόμενα είναι ότι σε ένα γράφημα με clusters ο αριθμός των κόμβων που παράγουν οι αλγόριθμοι Node Importance , AACDS1 και Node Importance algorithm with color είναι μικρότερος ή ίσος από τον αριθμό των clusterheads του γράφου και όλοι οι αλγόριθμοι σε ένα γράφο όταν αυξάνουμε την εμβέλεια των κόμβων από ένα σημείο και μετά το CDS που παράγουν αποτελείται από ένα κόμβο.

Το γενικότερο συμπέρασμα από όλα τα πειράματα που έγιναν είναι πως σε ένα 'τυχαίο' δίκτυο ο αλγόριθμος AACDS 1 παράγει CDS με τους λιγότερους κόμβους. Έπειτα οι αλγόριθμοι που χρησιμοποιούν την ποσότητα Node Importance έχουν την ίδια σχεδόν επίδοση (βέβαια είδαμε πως ο Node Importance Algorithm είναι λίγο καλύτερος από τον Node Importance with color). Τέλος ο AACDS2 αν και στην πρώτη φάση του αλγορίθμου έχει καλή επίδοση – δημιουργεί ένα dominating set με ικανοποιητικά πλήθος κόμβων-στην δεύτερη φάση του είδαμε πως έχει την χειρότερη επίδοση.

5.5. Μελλοντική Εργασία

Στην εργασία αυτή μελετήσαμε την απόδοση των αλγορίθμων σε σχέση με τον αριθμό των κόμβων που παράγουν. Όμως για μία πανεκπομπή (broadcast task) σε ένα MANET δεν παίζει ρόλο μόνο ο αριθμός των κόμβων που έχει το CDS. Πρέπει να δοκιμάσουμε τους αλγορίθμους για το πόσο γρήγορα ολοκληρώνετε η πανεκπομπή στο γράφημα και γενικότερα πόσο γρήγορα επικοινωνούν δύο κόμβοι μέσω του CDS που παράγονται από τον έκαστο αλγόριθμο. Δηλαδή να μετρήσουμε τον μέσο αριθμό από hops που χρειάζεται κάθε CDS για να σταλεί ένα μήνυμα από ένα κόμβο σε ένα άλλο. Δηλαδή να δούμε ποια από τα κεντροποιημένα πρωτόκολλα που είδαμε είναι βέλτιστο και ως προς το πλήθος των κόμβων που παράγουν αλλά και ως προς τον μέσο αριθμό hops που απαιτείται για επικοινωνία.

Επίσης πρέπει να δούμε ποιος από τους αλγόριθμους έχει καλή επίδοση όταν οι κόμβοι αλλάζουν θέση. Δηλαδή να εφαρμόσουμε τους αλγορίθμους αυτούς σε καταστάσεις όπου οι κόμβοι είναι κινητοί. Θα έχει ενδιαφέρον να δούμε πόσο γρήγορα προσαρμόζονται οι αλγόριθμοι σε MANET όπου οι θέσεις των κόμβων μεταβάλλονται και η μορφολογία του δικτύου συνεχώς αλλάζει.

ΠΑΡΑΡΤΗΜΑ Α'

A.1. Γενικά για την υλοποίηση

Στο σημείο αυτό πρέπει να γράψω την φιλοσοφία του προγράμματος που υλοποίησα και να σχολιάσω τον κώδικα από τον οποίο προσομοίωσα ad hoc δίκτυα και έκανα τα πειράματα για να καταλήξω στα συμπεράσματα στα οποία κατέληξα. Επίσης πρέπει να πω δυο λόγια για την υλοποίηση των αλγορίθμων που υλοποίησα.

Καταρχάς η γλώσσα προγραμματισμού με την οποία υλοποίησα τα προγράμματα είναι η Java.

Ένα κόμβο στο δίκτυο τον υλοποιώ σαν μια κλάση η οποία περιέχει:

- Το αναγνωριστικό (id) του κόμβου το οποίο είναι μοναδικό για κάθε κόμβο.
- Τις συντεταγμένες x,y που δείχνουν την θέση του κόμβου στο δίκτυο
- Την ακτίνα-εκπομπής (range) η οποία είναι η ακτίνα που εκπέμπει ο κόμβος μέσα στην οποία 'βλέπει' άλλους κόμβους
- Το Node Importance : μέγεθος το οποίο χρησιμοποιείται σε ένα από τους αλγορίθμους .
- Μία μεταβλητή που προσδιορίζει το 'χρώμα' του κόμβου (λευκό-μαύρο- γκρι).

- Μια λίστα με τους κόμβους που ανήκουν στην πρώτη τάξη γειτονία
- Μια λίστα με τους κόμβους που ανήκουν στην δεύτερη τάξη γειτονία
- Μια λίστα με τους κόμβους που ανήκουν στην πρώτη και στην δεύτερη τάξη γειτονία

Τρέχοντας το πρόγραμμα διαβάζει σαν είσοδο ένα αρχείο .txt το οποίο περιέχει εγγραφές του τύπου id,x,y,range . Για παράδειγμα τα περιεχόμενα ενός τέτοιου αρχείου θα μπορούσε να είναι:

```
0,5.0,7.0,244.0
1,239.0,20.0,235.0
2,36.0,110.0,108.0
3,282.0,67.0,64.0
4,142.0,35.0,130.0
5,297.0,106.0,42.0
6,419.0,81.0,191.0
7,336.0,11.0,98.0
8,201.0,117.0,105.0
9,248.0,175.0,196.0
```

Τις εγγραφές αυτές τις αποθηκεύω σε μία λίστα και κάθε node της λίστα είναι μία κλάση `ad_node`.

Αργότερα υπολογίζω την 1^η και 2^η τάξης γειτονιά του κάθε κόμβου και το Node Importance.

- Για την εύρεση της γειτονιάς χρησιμοποιώ της συναρτήσεις `find_1hop_neigh(ad_node x)` , `find_2hop_neigh(ad_node x)` , `find_12hop_neigh(ad_node x)` οι οποίες χρησιμοποιούν τις συντεταγμένες κάθε κόμβου και το range του για να βρει την αντίστοιχες γειτονιές.
- Για την εύρεση του node importance ενός κόμβου χρησιμοποιώ την συνάρτηση `node_importance()` η οποία υπολογίζει το node importance κάθε κόμβου σύμφωνα με τον αλγόριθμο

CalculateNodeImportanceIndex από το paper 'The Geodesic Broadcast Scheme for Wireless Ad Hoc Networks'.

A.2. Υλοποίηση του αλγορίθμου Node Importance algorithm for Connected Dominating sets

Ο Αλγόριθμος Node importance algorithm for Connected Dominating sets υλοποιείται από την μέθοδο **cds_NI_centralized()** η οποία καλείται αφού υπολογίσω τις 1^{ης} και 2^{ης} τάξης γειτονιές και το node importance του κάθε κόμβου. Στην συνάρτηση αυτή επαναληπτικά κάνω τα εξής:

1. Βρίσκω από τον κόμβο από την λίστα με τους κόμβους οι οποίοι είναι εκτός Connected Dominating set με το μεγαλύτερο node importance
2. Αφού βρω τον κόμβο ελέγχω κατά πόσο οι μέχρι τώρα κόμβοι που ανήκουν στο Connected Dominating Set καλύπτουν την γειτονιά του προς εισαγωγή κόμβου.
3. Αν καλύπτουν όλη την γειτονιά δεν προστίθεται ο κόμβος
4. Αν δεν την καλύπτουν ή την καλύπτουν μερικώς τότε ο κόμβος προστίθεται στο σύνολο των κόμβων που βρίσκονται στο μέχρι τώρα Connected Dominating Set.
5. Επαναλαμβάνουμε το βήμα 1 μέχρις ότου οι κόμβοι που συλλέγω να είναι ενωμένοι και να καλύπτουν το δίκτυο-γράφο.

(η συνάρτηση **cds_NI_centralized_sort()** κάνει το ίδιο με την διαφορά ότι ταξινομεί τους κόμβους ανάλογα με το Node Importance και επαναληπτικά επιλέγει το πρώτο και τον αφαιρεί από την λίστα – η ταξινόμηση γίνεται με την μέθοδο της φουσαλίδας) .

A.3. Υλοποίηση του αλγορίθμου Approximation Algorithms for Connected Dominating sets-Algorithm 1

Ο αλγόριθμος Approximation Algorithms for Connected Dominating sets-Algorithm 1 υλοποιείται στο πρόγραμμα από την μέθοδο AACDS(). Στην μέθοδο αυτοί αφού αρχικοποιήσω τους κόμβους χρωματίζοντας τους άσπρους (ουσιαστικά θέτω την μεταβλητή color της κλάσης ad_node ίση με 0) βρίσκω τον κόμβο που έχει τους περισσότερους λευκούς γείτονες τον χρωματίζω μαύρο (θέτω το ad_node.color=2) και τους γείτονές του τους χρωματίζω γκρι (θέτω ad_node.color=1) και τους αποθηκεύω σε μία λίστα. Επαναληπτικά μέχρι να μην υπάρχουν λευκοί κόμβοι επιλέγω από την λίστα με τους γκρι κόμβους τον κόμβο με τους περισσότερους λευκούς γείτονες και τον χρωματίζω μαύρο και του λευκούς γείτονες του γκρι και τους προσθέτω στην λίστα με τους υπολοίπους γκρι.

A.3. Υλοποίηση του αλγορίθμου Approximation Algorithms for Connected Dominating sets-Algorithm 2

Ο αλγόριθμος Approximation Algorithms for Connected Dominating sets-Algorithm 2 υλοποιείται στο πρόγραμμα από την μέθοδο AACDS_2(). Τα βήματα του αλγορίθμου που υλοποιώ είναι:

1. Αρχικά χρωματίζω όλους του κόμβους άσπρους (ad_node.color=0).
2. Επιλέγω τον κόμβο με τους περισσότερους λευκούς και τον χρωματίζω μαύρο (ad_node.color=2) ενώ του γείτονες του χρωματίζω γκρι.
3. Επαναληπτικά μέχρι να μην υπάρχουν λευκοί κόμβοι (η συνάρτηση have_whites() επιστρέφει true ή false ανάλογα αν υπάρχουν λευκοί ή όχι κόμβοι) κάνω τα εξής:
 1. υπολογίζω για κάθε κόμβο την λευκή γειτονιά του με την συνάρτηση calc_white_AACDS_2()
 2. υπολογίζω των αριθμό pieces για κάθε κόμβο

3. Επιλέγω τον κόμβο με τον μεγαλύτερο αριθμό από pieces και τον χρωματίζω άσπρο και τους γείτονες του γκρι.

4. Επαναληπτικά μέχρι οι κόμβοι που υπολόγισα πιο πάνω να γίνουν connected (η συνάρτηση `is_cds(cds_AACDS_2)` επιστρέφει true αν τα στοιχεία της λίστας εισόδου αποτελούν connected dominating set , δηλαδή είναι ενωμένοι μεταξύ τους και καλύπτουν όλο τον γράφο)

1. Για κάθε γκριζο κόμβο μετράω πόσους μαύρους μη συνδεδεμένους κόμβους βρίσκονται στην Neigh12hop λίστα του καθενός και τον κόμβο με τον μεγαλύτερο αριθμό τον χρωματίζω μαύρο. (ο αριθμός των μη συνδεδεμένων μαύρων κόμβων δίνεται από την συνάρτηση `int black_clust(LinkedList black_list)`.)

Όπως είπαμε πιο πάνω και όπως βλέπουμε τώρα η πρώτη φάση του αλγορίθμου είναι τα πρώτα τρία βήματα ενώ το τέταρτο είναι η δεύτερη φάση του αλγορίθμου.

A.4. Υλοποίηση του Node Importance Algorithm with color

Ο αλγόριθμος Node Importance Algorithm with color υλοποιείται από την μέθοδο `NIBB_with_color()` στην οποία γίνονται τα εξής:

1. Αρχικά επιλέγω από τον πίνακα `node_array` τον κόμβο με τον μεγαλύτερο NI (node importance) .
2. Τον χρωματίζω μαύρο (`node_array[id].color=2`) .
3. Τον αφαιρώ από την λίστα με τους γκρι (`grey_nodes`) αν υπάρχει και τον προσθέτω στην λίστα με τους κόμβους του connected dominating set
4. Χρωματίζω την λευκή γειτονιά του κόμβου που μόλις πρόσθεσα στο παραπάνω βήμα γκρι (`node_array[temp.id].color=1;`) και τους προσθέτω στην λίστα με τους γκρι.
5. Στην λίστα με τους γκρι βρίσκω τον κόμβο με το μεγαλύτερο id
6. Όσο υπάρχουν λευκοί κόμβοι (`while(have_whites())`) – η συνάρτηση `have_whites()` επιστρέφει true false ανάλογα με το αν υπάρχουν λευκοί ή όχι κόμβοι- επιστρέφω στο βήμα 2.

A.5. Προγράμματα που δημιουργούν ad hoc δίκτυα-γράφους

Για να παράγω εύκολα και ψευδοτυχαία δίκτυα έχω δημιουργήσει τρία προγράμματα τα οποία παίρνουν σαν είσοδο τον αριθμό των κόμβων του δικτύου , δημιουργούν ένα ψευδοτυχαίο γράφο και έχουμε έξοδο ένα αρχείο με κατάληξη .txt το οποίο περιέχει κόμβους στην κωδικοποίηση που περιγράψαμε στην αρχή του παραρτήματος.

Το πρώτο πρόγραμμα είναι το `ad_hoc_creator.java` το οποίο παίρνει απλά τον αριθμό των κόμβων και την τάξη του `range` που θα έχουν οι κόμβοι (με το πρόγραμμα αυτό παράγω γραφήματα τα οποία τα χρησιμοποιώ κυρίως στο πρώτο πείραμα του Κεφαλαίου 4) . Ο τρόπος με τον οποίο παράγεται ένα δίκτυο με αυτό το πρόγραμμα είναι:

Κάνω τόσες επαναλήψεις όσες ο αριθμός των κόμβων του δικτύου που θέλω να παράγω. Σε κάθε επανάληψη κάνω τα εξής:

- 1.κάθε κόμβο του παράγω τον αποθηκεύω σε μία λίστα

- 2.από την λίστα επιλέγω τυχαία ένα κόμβο

```
id_toadd=(int)((Math.random()*(temp_list.size()-1)));
```

```
temp_node=(node)temp_list.get(id_toadd);
```

- 3.επιλέγω ένα τυχαίο x,y τα οποία πρέπει να είναι μεταξύ 0-1000

- 4.επιλέγω το `range` του καινούριου κόμβου την απόσταση του τυχαίου κόμβου με τα τις συντεταγμένες που υπολόγισα παραπάνω.

Η συνάρτηση που επιλέγω ψευδοτυχαίους αριθμούς είναι η `Math.random()`.

Το δεύτερο πρόγραμμα είναι το `cluster_net_creator.java` το οποίο παίρνει σαν είσοδο τον αριθμό των κόμβων `clusterheads` και των αριθμό των κόμβων που θα είναι γείτονες σε κάθε `clusterhead` καθώς επίσης και την σχετική απόσταση των `clusterhead` και την σχετική απόσταση των κόμβων που γειτονεύουν με τα `cluster heads` (με αυτό το πρόγραμμα παράγω γραφήματα κυρίως για το τρίτο πείραμα) . Για να δημιουργήσω το δίκτυο κάνω τα εξής:

1. δημιουργώ τα `clusterheads` ανάλογα με την σχετική απόσταση και τον αριθμό `clusterheads` που έδωσα σαν είσοδο

2. για κάθε κόμβο `clusterhead` δημιουργώ κόμβους γύρω από αυτόν ανάλογα με τον αριθμό των κόμβων που θα είναι γείτονες σε ένα `clusterhead` και την σχετική ακτίνα .

Το τρίτο πρόγραμμα είναι το `same_range.java` το οποίο παράγει γραφήματα των οποίων οι κόμβοι έχουν ψευδοτυχαία θέση αλλά σταθερή εμβέλεια ο κάθε ένας. Σαν είσοδο αυτό το πρόγραμμα παίρνει τον αριθμό των κόμβων και την εμβέλεια που θα έχει ο κάθε κόμβος (με αυτό το πρόγραμμα παράγω γράφους για το τέταρτο πείραμα). Για να δημιουργήσω το γράφημα κάνω ακριβώς τα ίδια βήματα με το πρώτο πρόγραμμα , `ad_hoc_creator.java` , με την διαφορά ότι την θέση ενός καινούριου κόμβου την επιλέγω στον δακτύλιο που αποτελείται από τους κύκλους με ακτίνες $2/3$ `range` και `range` του κόμβου που θα στηριχθώ για να προσθέσω τον καινούριο κόμβο και επίσης όλοι οι κόμβοι έχουν την ίδια εμβέλεια την οποία την προκαθορίζει ο χρήστης.

A.6. Πρόγραμμα που αυξάνει ομοιόμορφα την εμβέλεια του κάθε κόμβου

Στην ενότητα αυτή περιγράφω το πρόγραμμα που χρησιμοποιώ στο τρίτο πείραμα του τετάρτου κεφαλαίου για να αυξάνω ομοιόμορφα την εμβέλεια του κάθε κόμβου του γραφήματος. Το πρόγραμμα αυτό παίρνει σαν είσοδο το αρχείο με το γράφημα στο οποίο θα αυξήσω την εμβέλεια των κόμβων, το όνομα του αρχείου που θα αποθηκεύσω το νέο γράφημα και ένα αριθμό `a` που αναπαριστά τις μονάδες που θα προσθέσω στην εμβέλεια του κάθε κόμβου. Ουσιαστικά το πρόγραμμα αυτό απλά προσθέτει τις μονάδες `a` στην εμβέλεια του κάθε κόμβου και τα αποθηκεύει στο νέο αρχείο.

Αναφορές - Βιβλιογραφία

- [1] The Geodesic Broadcast Scheme for Wireless Ad Hoc Networks*
Δ. Κασσαρός , Γ. Μανολόπουλος
- [2] Approximation Algorithms for Connected Dominating Sets
S. Guha² , S. Khuller³
- [3] Connected Dominating Set in Sensor Networks and MANETs
J. Blum, M. Ding, A. Thaeler, X. Cheng
- [4] Mobile Ad hoc Networking
S.Basagni , M.Conti , S.Giordano , I. Stojmenovic
- [5] A dominating-set-based routing scheme in ad hoc wireless networks.
J. Wu , H. Li
- [6] A generic distributed broadcast scheme for ad hoc wireless networks
J. Wu , F.Dai
- [7] The broadcast storm problem in a mobile ad hoc network.
Y. Tseng , S. Ni , Y.Chen , J. Sheu.
- [8] Dominating Sets and neighbor elimination-based broadcasting algorithms in wireless networks
I.Stojmenovic, M.Seddigh , J Zunic.

- [9] **Wireless Information Highways**
Δ. Κατσαρός , Γ. Μανολόπουλος, Α. Νανόπουλος
- [10] **Solving Connected Dominating Set Faster than 2^n**
F. Fomin, F. Grandoni, D. Kratsch



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ



004000091265