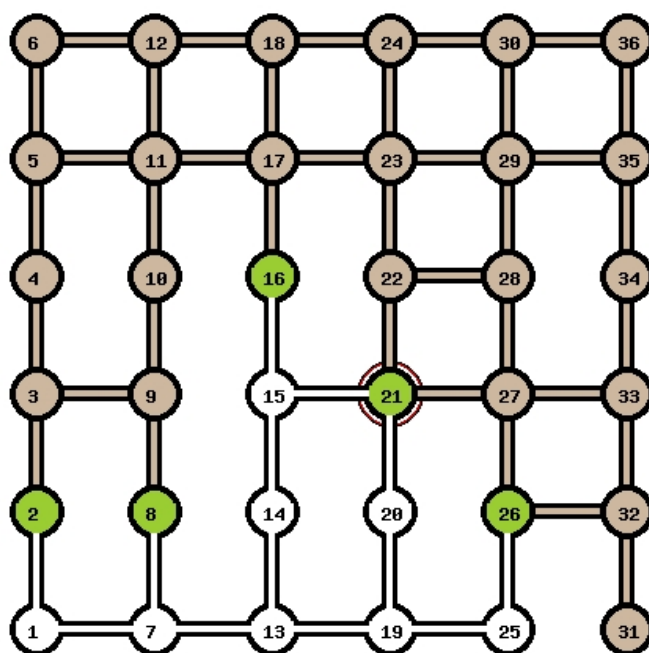




ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΡΟΒΛΗΜΑΤΑ ΚΑΤΑΔΙΩΞΗΣ - ΔΙΑΦΥΓΗΣ
ΜΕ ΜΕΘΟΔΟΥΣ ΤΗΣ ΘΕΩΡΙΑΣ ΓΡΑΦΩΝ:
ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΠΕΙΡΑΜΑΤΙΚΗ ΜΕΛΕΤΗ



Κωνσταντίνος Τομπακίδης
ΑΕΜ:5507

Επιβλέπων καθηγητής: Κεχαγιάς Αθανάσιος

Θεσσαλονίκη, Οκτώβριος 2010

ΠΡΟΛΟΓΟΣ

Η παρούσα διπλωματική εργασία ασχολείται με το πρόβλημα της έρευνας γράφων. Το πρόβλημα έχει δύο κύριες εκδοχές οι οποίες αναλύονται στην παράγραφο 1.1. Από αυτές ασχολούμαστε κυρίως με τη δεύτερη εκδοχή, την έρευνα ακμών. Οι βασικοί στόχοι της εργασίας είναι οι εξής τέσσερις, κατά σειρά προτεραιότητας:

1. Η ανάπτυξη ενός λογισμικού που υλοποιεί τους αλγόριθμους έρευνας ακμών που περιγράφονται στο κεφάλαιο 4. Το πρόγραμμα θέλουμε να είναι εύχρηστο και γρήγορο και να μπορεί να συνεργαστεί με το πρόγραμμα υλοποίησης του αλγορίθμου GSST (κεφάλαιο 3), το gsearch.exe (παράγραφος 5.1.1), χρησιμοποιώντας ίδια δομή στα αρχεία εισόδου και εξόδου του.
2. Η επέκταση του γραφικού περιβάλλοντος του gsearch.exe έτσι ώστε να μπορεί να χρησιμοποιηθεί όχι μόνο για έρευνα κόμβων αλλά και για έρευνα ακμών. Επίσης η βελτίωση της οπτικοποίησης των ερευνών έτσι ώστε να γίνεται καλύτερη εποπτεία των καταστάσεων των στοιχείων του γράφου (παράγραφος 5.1.4) για τους δυο διαφορετικούς τύπους ερευνών.
3. Η μαθηματική μελέτη και απόδειξη των δυο αλγορίθμων έρευνας ακμών που επινοήσαμε.
4. Η συγκριτική αξιολόγηση των αλγορίθμων έρευνας ακμών μέσω πειραμάτων σε γράφους, είτε κατασκευασμένους για αυτό το σκοπό, είτε μοντέλων πραγματικών χώρων (κεφάλαιο 6).

Την εργασία αφιερώνω στην οικογένεια και τους φίλους μου.

Κ. Τομπακίδης,
Οκτώβρης 2010
Θεσσαλονίκη

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΡΟΛΟΓΟΣ.....	3
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	5
1.Εισαγωγή.....	7
1.1 Κτίρια και σπηλιές	7
1.2 Προηγούμενες εργασίες.....	8
1.3 Πιθανές εφαρμογές	9
2.Περιγραφή του προβλήματος.....	11
2.1 Βασικές έννοιες και συμβολισμοί.....	11
2.2 Έρευνα κόμβων και έρευνα ακμών.....	13
2.2.1 Κανόνες των παιχνιδιών	13
2.2.2 Σύγκριση των δύο παιχνιδιών.....	19
2.3 Έρευνες σε δένδρα.....	20
3.Αλγόριθμος GSST.....	23
3.1 Κεντρική Ιδέα.....	23
3.2.1 Παρουσίαση αλγορίθμου	26
3.2.2 Πληρότητα	30
3.3 Συμπεράσματα προηγούμενων πειραμάτων	31
4.Αλγόριθμοι έρευνας ακμών.....	33
4.1 Αλγόριθμος Extra Nodes.....	33
4.2 Αλγόριθμος Extra Searcher.....	35
4.2.1 Κεντρική Ιδέα	35
4.2.2 Παρουσίαση αλγορίθμου.....	36
4.2.3 Μαθηματική Ανάλυση-Απόδειξη	38
5.Περιγραφή λογισμικού	47
5.1 Εργαλεία γραμμής εντολών	47
5.1.1 Gsearch.exe.....	47
5.1.2 Eclear.exe.....	49
5.2 Η δυναμική βιβλιοθήκη plot.dll	50
5.2.1 Η συνάρτηση plotgraph	50
5.2.2 Οι συναρτήσεις plotnsearch και ploteseach	51
5.3 Γραφικό περιβάλλον GsearchGui.exe.....	54
5.3.1 Το Front end	55
5.3.2 Σενάρια χρήσης.....	58
6.Πειράματα	73
6.1 Εισαγωγικά.....	73
6.1 Γράφοι.....	75
6.1.1 Απλός γράφος 1	75
6.1.2 Απλός γράφος 2	80
6.1.3 Δένδρο	84
6.1.4. Πλέγμα 5X5	88
6.1.5. Δένδρο-Πλέγμα 4X4.....	93
6.1.6. Γράφος του NSH	98
6.1.7. Εθνική πινακοθήκη (Washington D.C.)	103
6.2 Οικογένειες γράφων.....	108
6.2.1 Πλήρη Πλέγματα	108
6.2.2 Ελαττωμένα Πλέγματα	119
6.3 Συμπεράσματα πειραμάτων	129
7.Επίλογος	131
7.1 Απολογισμός.....	131
7.2 Προτεινόμενη μελλοντική έρευνα	131
8.Βιβλιογραφία.....	133

1.Εισαγωγή

1.1 Κτίρια και σπηλιές

Σε αυτή την εργασία μελετάμε μια παραλλαγή της έρευνας γράφων. Το κίνητρό μας πηγάζει από προβλήματα εφαρμοσμένης ρομποτικής, όπως θα εξηγηθούν στο κεφάλαιο 1.3. Σαν εισαγωγή, εξετάζουμε τα ακόλουθα δύο προβλήματα, τα οποία μπορούν και τα δύο να ανοιχθούν σε έρευνες γράφων:

1. Ένας φυγάδας κρύβεται σε ένα κτίριο και μια ομάδα ενός ή περισσότερων ερευνητών προσπαθεί να τον συλλάβει. Το κτίριο αντιπροσωπεύεται από ένα γράφο G : οι κόμβοι είναι δωμάτια και οι ακμές είναι πόρτες μεταξύ των δωματίων. Και οι ερευνητές και ο φυγάδας καταλαμβάνουν τους κόμβους της γραφικής παράστασης και κινούνται από κόμβο σε κόμβο ολισθαίνοντας κατά μήκος των ακμών. Ο φυγάδας συλλαμβάνεται όταν βρίσκεται στον ίδιο κόμβο με έναν ερευνητή. Υποτίθεται ότι ο φυγάδας έχει τα ακόλουθα χαρακτηριστικά: (α) θέλει να αποφύγει τη σύλληψη, (β) είναι αόρατος στους ερευνητές (εκτός αν βρίσκεται στον ίδιο κόμβο με έναν από αυτούς), (γ) γνωρίζει πάντα τις θέσεις των ερευνητών και (δ) είναι απείρως γρήγορος.

2. Ένας φυγάδας και μια ομάδα των ερευνητών βρίσκονται μέσα σε μια σπηλιά. Η σπηλιά αντιπροσωπεύεται πάλι από ένα γράφο G : οι ακμές είναι σήραγγες και οι κόμβοι είναι οι διασταυρώσεις των σηράγγων. Οι ερευνητές κινούνται από τον κόμβο σε κόμβο, ολισθαίνοντας κατά μήκος των ακμών: σύλληψη πραγματοποιείται εάν ένας ερευνητής διαβεί κατά μήκος μιας ακμής που καταλαμβάνεται από το φυγάδα, ή εάν ο φυγάδας κινηθεί διαμέσου ενός κόμβου που καταλαμβάνεται από έναν ερευνητή. Οι προαναφερθείσες ιδιότητες του φυγάδα υποτίθεται ότι ισχύουν και σε αυτήν την περίπτωση επίσης.

Τα δύο προβλήματα είναι παρόμοια αλλά διαφέρουν από μια σημαντική άποψη. Στο πρώτο πρόβλημα ο φυγάδας υποτίθεται ότι βρίσκεται στους κόμβους του γράφου (τα δωμάτια) , ενώ στο δεύτερο πρόβλημα υποτίθεται ότι βρίσκεται στις ακμές (οι σήραγγες). Συνεπώς, θα ονομάσουμε το πρώτο πρόβλημα *έρευνα κόμβων* και το δεύτερο *έρευνα ακμών*. Η έρευνα ακμών, όπως θα δειχθεί στην παράγραφο 2.2.2, είναι γενικότερη της έρευνας κόμβων.

Σε αυτή την εργασία θα ασχοληθούμε κυρίως με την έρευνα ακμών και τη σχέση αυτής με την έρευνα κόμβων. Η κύρια συμβολή της παρούσας εργασίας είναι η παρουσίαση, τεκμηρίωση και πειραματική μελέτη δύο αλγορίθμων: *Extra Nodes(EN)* και *Extra Searcher(ES)* οι οποίοι ανάγουν μια *εσωτερική μονότονη και συνδεδεμένη (IMC)* έρευνα κόμβων σε *IMC* έρευνα ακμών. Ως βάση για αυτούς θα χρησιμοποιηθεί ο αλγόριθμος έρευνας κόμβων GSST (*Guaranteed Search by Spanning Tree*), όπως παρουσιάστηκε και υλοποιήθηκε από τους Hollinger, Kehagias και Gelastopoulos στο [41]. Επίσης αναπτύσσεται και διατίθεται ελεύθερα (στην τοποθεσία <http://users.auth.gr/~kehagiat/KehagiasSoftware.htm>) το λογισμικό που υλοποιεί τους δύο αλγορίθμους καθώς και το γραφικό περιβάλλον των GSST, EN και ES που απεικονίζει τις λειτουργίες τους.

1.2 Προηγούμενες εργασίες

Η θέση του φυγάδα (ακμή ή κόμβος) είναι μια σημαντική πλευρά του προβλήματος της έρευνας γράφων και μπορεί να χρησιμοποιηθεί για να ταξινομήσουμε τη σχετική βιβλιογραφία.

Ο όρος *έρευνα κόμβων* έχει χρησιμοποιηθεί στο παρελθόν για να περιγράψει μια εκδοχή έρευνας γράφων του *τοποθετημένου σε ακμή φυγάδα* (*e-φυγάδα*) [25]. Ως εκ τούτου, για να αποφεύγεται η σύγχυση, επαναλαμβάνεται ότι χρησιμοποιούμε την *έρευνα κόμβων* για να αναφερθούμε στην αναζήτηση ενός *τοποθετημένου σε κόμβο φυγάδα* (*n-φυγάδα*).

Μια άλλη σημαντική πτυχή του προβλήματος της έρευνας γράφων είναι η ορατότητα των ερευνητών. Δηλαδή, λέμε ότι ο φυγάδας είναι *ορατός* εάν οι ερευνητές ξέρουν πάντα τη θέση του και λέμε ότι ο φυγάδας είναι *αόρατος* εάν οι ερευνητές μαθαίνουν τη θέση του μόνο όταν είναι στον ίδιο κόμβο (ή την ακμή) με αυτόν.

Πρόσθετα χαρακτηριστικά της έρευνας γράφων είναι εάν αυτή είναι *εσωτερική* (οι ερευνητές μπορούν μόνο να κινηθούν κατά μήκος των ακμών του γράφου) ή όχι (ένας ερευνητής μπορεί, σε ένα βήμα, να μεταβεί σε οποιοδήποτε κόμβο της γραφικής παράστασης –αυτό το ονομάζουμε *διακίνηση*), *μονότονη* (ο φυγάδας δεν μπορεί ποτέ να επιστρέψει σε ένα ήδη καθαρισμένο μέρος του γράφου) και *συνεκτική* (τα καθαρισμένα μέρη διαμορφώνουν ένα συνδεδεμένο υπογράφο). Αυτοί οι όροι θα οριστούν αυστηρά στην ενότητα 2.

Παρουσιάζουμε τώρα μια συνοπτική επισκόπηση της βιβλιογραφίας πάνω στην έρευνα σε γράφους. Οι πρώιμες εργασίες περιλαμβάνουν το [6] όπου διατυπώθηκε το πρόβλημα της έρευνας σπηλιών και το [31] όπου παρουσιάστηκε η πρώτη μαθηματική μελέτη του προβλήματος. Το πρόβλημα ανακαλύφθηκε πάλι ανεξάρτητα λίγο αργότερα από την πιο πρόσφατη [32]. Το πρόβλημα της *έρευνας ακμών* εξετάστηκε στα πλαίσια της θεωρίας γράφων στην [29]. Η ήδη αναφερθείσα παραλλαγή έρευνας κόμβων (που εξετάζει στην πραγματικότητα τον *e-φυγάδα*) εμφανίστηκε ελάχιστα νωρίτερα στην [25]. Μια άλλη παραλλαγή, αποκαλούμενη *μικτή αναζήτηση* (και πάλι εξέταση του *e-φυγάδα*) εμφανίστηκε στην [5] και μελετήθηκε περαιτέρω στις [35,38]. Η μελέτη της συνδεδεμένης αναζήτησης ακμών άρχισε σχετικά πρόσφατα δείτε [2,3,11,13,14,39]. Λίγα πράγματα έχουν δημοσιευτεί σχετικά με την αληθινή έρευνα κόμβων, δηλαδή την αναζήτηση ενός *τοποθετημένου σε κόμβο φυγάδα*. Υπάρχει μια ιδιαίτερη βιβλιογραφία στην αναζήτηση των ορατών *τοποθετημένων σε κόμβο φυγάδων*, την οποία δεν θα συζητήσουμε, δεδομένου ότι η αναζήτηση ορατών φυγάδων είναι έξω από το αντικείμενο της παρούσας εργασίας. Απλά αναφέρεται ότι μια σημαντική πρώιμη εργασία είναι η [30]. Η έρευνα γράφων συσχετίζεται επίσης με διάφορες *παραμέτρους* των γράφων, από τις οποίες αρκετά σημαντικές το *πλάτος μονοπατιού* και ο *διαχωρισμός των κορυφών*. Αυτές οι παράμετροι συζητούνται σε διάφορες δημοσιεύσεις, π.χ., στις [4,8,9,24,25,36]. Τέλος υπάρχουν διάφορες επισκοπήσεις της βιβλιογραφίας αναζήτησης σε γράφους: μια παλαιά και βαθιά είναι η [4], μια πιο πρόσφατη είναι η [1] και πολύ πρόσφατη και εκτενής είναι η [12].

Οι προαναφερθείσες εργασίες υιοθετούν μια θεωρητική προσέγγιση. Ενώ υπάρχει πολλή συζήτηση για τους αλγορίθμους έρευνας γράφων, έχουμε βρει λίγους πραγματικά εφαρμοσμένους αλγορίθμους που μπορούν να αντιμετωπίσουν σχετικά μεγάλους γράφους (με εξαίρεση τους αλγορίθμους αναζήτησης σε δέντρα). Μια πιο εφαρμοσμένη μορφή έρευνας εμφανίζεται στη βιβλιογραφία της ρομποτικής, παραδείγματος χάριν στις [15,16,17,27] και το βιβλίο [28].

Αυτές οι εργασίες αποτελούν ουσιαστικά εφαρμογές αλγορίθμων, και παρουσιάζουν αριθμητικά και φυσικά πειράματα, αλλά παρέχουν ελάχιστη θεωρητική αιτιολόγηση των μεθόδων τους. Πιο συγκεκριμένα, η διάκριση μεταξύ της έρευνας ακμών και κόμβων συχνά παρανοείται. Η διευκρίνιση αυτής της διάκρισης και ο συνδυασμός της ρομποτικής και στοιχείων της θεωρίας γράφων γίνεται σε δύο τεχνικές αναφορές των [20,23].

Τέλος, στην [41] αναλύεται το πρόβλημα της αναζήτησης ενός *αόρατου*, *n-φυγάδα*. Εξετάζεται επίσης η σχέση με το πρόβλημα του *αόρατου*, *e-φυγάδα*. Επίσης παρουσιάζεται και τεκμηριώνεται, θεωρητικά και πειραματικά, ο αλγόριθμος έρευνας κόμβων GSST (*Guaranteed Search by Spanning Tree*), σε διάφορες παραλλαγές. Το εν λόγω έγγραφο ασχολείται κυρίως με εσωτερική, μονότονη, συνδεδεμένη (IMC) έρευνα κόμβων καθώς αυτή η εκδοχή του προβλήματος είναι και η πιο σχετική με τις εφαρμογές ρομποτικής.

1.3 Πιθανές εφαρμογές

Το πρόβλημα με το οποίο ασχολούμαστε στην παρούσα εργασία, δηλ. Η έρευνα γράφων, έχει πολυποίκιλες εφαρμογές.

1. Ρομποτική: Έρευνες σε φυσικούς χώρους εσωτερικούς ή εξωτερικούς από ομάδες ρομπότ ή ανθρώπων και ρομπότ. Τα προβλήματα της έρευνας κόμβων και έρευνας ακμών μπορούν να μοντελοποιήσουν την αναζήτηση επιζώντων σε καταστάσεις ανάγκης από ομάδες διασωστικών ρομπότ (Urban Search And Rescue robots). Δεδομένου ότι οι γράφοι είναι εκ των προτέρων γνωστοί, μπορούμε να χρησιμοποιήσουμε την έρευνα γράφων μόνο για προβλήματα με γνωστό περιβάλλον (π.χ. πυρκαγιές, εγκλωβισμούς αλλά όχι σεισμούς με κατάρρευση). Ωστόσο, όπως αναλύεται και στην [42], οι αλγόριθμοι που προαναφέρθηκαν μπορούν πάντα να συνδυαστούν με άλλους και να αποτελέσουν ένα μέρος μόνο των τελικών αποφάσεων για τις κινήσεις των USAR robots, ανάλογα με την περίπτωση.
2. Στατιστική: Παραγοντοποίηση Bayesianών δικτύων. Τα Bayesian δίκτυα χρησιμοποιούνται για την απεικόνιση πολλών εξαρτώμενων γεγονότων (κόμβοι) και τις πιθανότητες της μετάβασης από το ένα στο άλλο (ακμές). Χρησιμοποιώντας τις τιμές των δεσμευμένων αυτών πιθανοτήτων μπορούμε να υπολογίσουμε την συνδυασμένη πιθανότητα κάθε γεγονότος συνολικά. Η διαδικασία εύρεσης της αντίστοιχης συνάρτησης ονομάζεται παραγοντοποίηση του δικτύου και η πολυπλοκότητά της μπορεί να μειωθεί από εκθετική (στο μέγεθος του δικτύου) μέχρι και σε γραμμική σε αρκετές περιπτώσεις με τη χρήση της έρευνας ακμών και της αποσύνδεσης (treewidth) που αναφέρεται παρακάτω.
3. Γενικότερα, πολυπλοκότητα αλγορίθμων: Οι αλγόριθμοι έρευνας ακμών μπορούν να χρησιμοποιηθούν για να μειώσουν την, αλλιώς εκθετική, πολυπλοκότητα των αλγορίθμων αποσύνδεσης γράφου (treewidth, pathwidth decomposition). Αυτοί οι αλγόριθμοι βρίσκουν εφαρμογή σε πολλούς τομείς όπως η σχεδίαση συστημάτων VLSI, η αξιοπιστία δικτύων, η σχεδίαση τηλεπικοινωνιακών δικτύων, η παραγοντοποίηση Cholesky, η επεξεργασία της φυσικής γλώσσας, η βελτιστοποίηση μεταγλώττισης γλωσσών προγραμματισμού κ.ά.

Σε κάποιες από τις παραπάνω εφαρμογές μας ενδιαφέρει η έρευνα κόμβων, αλλά σε πολύ περισσότερες η έρευνα ακμών. Επομένως οι αλγόριθμοι που θα παρουσιάσουμε στο Κεφ. 4 για την τροποποίηση έρευνας κόμβων σε έρευνα ακμών, αν και προσεγγιστικοί, μπορούν να αποδειχθούν ιδιαίτερα χρήσιμοι.

Επιπλέον, πιστεύουμε ότι τα προγράμματα γραμμής εντολών και το γραφικό περιβάλλον έρχονται να καλύψουν ένα μεγάλο κενό, δεδομένου ότι (ίσως αναπάντεχα) δεν υπάρχουν δημόσια διαθέσιμες υλοποιήσεις των διαφόρων αλγορίθμων έρευνας γράφων οι οποίοι έχουν κατά καιρούς δημοσιευθεί στην βιβλιογραφία. Το δε GUI είναι αρκετά εύχρηστο και δίνει εποπτική αντίληψη του προβλήματος .

2.Περιγραφή του προβλήματος

Για την καλύτερη κατανόηση του προβλήματος, για την περιγραφή των καταστάσεων που προκύπτουν στα στοιχεία των γράφων και για την περιγραφή των αλγορίθμων είναι απαραίτητη η παρουσίαση των βασικών εννοιών και συμβολισμών. Αυτή γίνεται στο κεφάλαιο που ακολουθεί (Κεφ. 2.1), ενώ στο Κεφ. 2.2 αναλύεται η σχέση των δύο τύπων έρευνας που μας απασχολούν. Τα θεωρήματα και ορισμοί που ακολουθούν προέρχονται από την τεχνική έκθεση [41].

2.1 Βασικές έννοιες και συμβολισμοί

Συμβολίζουμε τους γράφους με έντονα γράμματα, π.χ. $G = (V, E)$ όπου V είναι το σύνολο των κόμβων και E το σύνολο των ακμών. Πάντα θα ονομάζουμε τους κόμβους του G ως $V = \{1, 2, \dots, N\}$ (συνεπώς ο γράφος περιέχει N κόμβους, δηλαδή $|V| = N$). Οι ακμές αναφέρονται ως $\{u, v\}$ με $u, v \in V$. Συνήθως θα γράφουμε uv , για απλούστευση, αλλά πρέπει να σημειώσουμε ότι το vu και το uv είναι η ίδια ακμή (με άλλα λόγια μελετάμε μη κατευθυνόμενους γράφους). Ασχολούμαστε μόνο με συνδεδεμένους γράφους χωρίς βρόγχους ή πολλαπλές ακμές. Επιπροσθέτως, για να αποφευχθούν μερικές εκφυλισμένες περιπτώσεις, πάντα θεωρούμε γράφους με τουλάχιστον μία ακμή (και τουλάχιστον δύο κόμβους).

Οι κόμβοι u, v ονομάζονται γειτονικοί αν και μόνο αν $uv \in E$.

Δεδομένης μιας ακολουθίας $u_1 u_2 \dots u_L$, με $u_i u_{i+1} \in E$ για $i = 1, 2, \dots, L-1$, λέμε ότι το $u_1 u_2 \dots u_L$ είναι:

1. μονοπάτι αν και μόνο αν $u_i \neq u_j$ για $i, j \in \{1, 2, \dots, L\}$ και $i \neq j$.
2. κύκλος αν και μόνο αν $u_i \neq u_j$ για $i, j \in \{1, 2, \dots, L\}$ (και $i \neq j$) εξόν από $u_1 = u_L$.

Δένδρο ονομάζουμε ένα συνεκτικό γράφο χωρίς κύκλους. Ισοδύναμοι ορισμοί είναι οι εξής:

α) δένδρο είναι ένας γράφος στον οποίο υπάρχει μοναδικό μονοπάτι μεταξύ κάθε ζευγαριού κόμβων του.

β) δένδρο ονομάζουμε ένα συνδεδεμένο γράφο με N κόμβους και $N - 1$ ακμές. Τα φύλλα ενός δένδρου είναι οι κόμβοι του που έχουν ακριβώς έναν γείτονα.

Ριζωμένο ονομάζουμε ένα δέντρο με ένα διακεκριμένο κόμβο u_0 που ονομάζεται ρίζα του δέντρου. Δεδομένου ενός δένδρου $T = (V, E)$, θα συμβολίζουμε το ίδιο δένδρο, αλλά πλέον ριζωμένο με ρίζα u_0 , ως $T_{u_0} = (V, E, u_0)$. Αν πάρουμε οποιονδήποτε κόμβο $u_L \neq u_0$ και θεωρώντας $u_0 u_1 \dots u_{L-1} u_L$ το μοναδικό μονοπάτι από το u_0 στον u_L , τότε ο u_{L-1} είναι ο γονέας του u_L , και ο u_L είναι το παιδί του u_{L-1} . Δεδομένου ενός κόμβου x , τα παιδιά του, τα παιδιά των παιδιών του και ακολούθως ονομάζονται απόγονοι του x . Δεδομένου ενός ριζωμένου δένδρου $T_x = (V, E, x)$ και ενός κόμβου y , θεωρούμε το σύνολο των κόμβων

$$V[y] = \{z : z \text{ είναι ο } y \text{ ή απόγονος του } y\}$$

και το σύνολο των ακμών

$$E[y] = \{zu : zu \in E \text{ και } z, u \in V[y]\}$$

Ας σημειωθεί ότι και το $V[y]$ και το $E[y]$ εξαρτώνται και από το y και από το x . Ο γράφος $(V[y], E[y])$ είναι δένδρο (υποδένδρο του T_x). Το ριζωμένο δένδρο $(V[y], E[y], y)$ θα συμβολίζεται ως $T_x[y]$.

Γεννητορικό δένδρο (spanning tree) ενός γράφου G ονομάζουμε κάθε δένδρο T που περιέχει όλους τους κόμβους του G και κάθε ακμή του οποίου είναι και ακμή του G .

Με άλλα λόγια το T ορίζει κάποιο υποσύνολο των ακμών του G το οποίο

1. εξακολουθεί να κρατά τον G συνεκτικό, και
2. δεν περιέχει κύκλο.

Στρατηγική έρευνας σε ένα γράφο $G = (V, E)$ λέγεται μια ακολουθία διατεταγμένων ζευγών κόμβων:

$$S = ((u(1), v(1)), (u(2), v(2)), \dots, (u(t_{fin}), v(t_{fin}))),$$

για τα οποία ισχύει $\{u(t), v(t)\} \in E$ για $t = 1, 2, \dots, t_{fin}$. Ονομάζουμε $S(t) = (u(t), v(t))$ την t -οστή κίνηση (ή βήμα) της στρατηγικής έρευνας. Η t -οστή κίνηση μπορεί να είναι μία από τις ακόλουθες:

1. τοποθέτηση ενός ερευνητή στον κόμβο v (στην οποία περίπτωση $u(t) = 0$ και $v(t) \in V$) ή
2. ολίσθηση ενός ερευνητή από κόμβο $u(t) \in V$ σε κόμβο $v(t) \in V$ ή
3. αφαίρεση ενός ερευνητή από κόμβο u (στην οποία περίπτωση $u(t) \in V$ και $v(t) = 0$).

Θα χρησιμοποιήσουμε επίσης τον πιο υπαινικτικό συμβολισμό $S(t) = (u \rightarrow v)$, που σημαίνει ότι ένας ερευνητής μετακινείται από τον κόμβο u στον κόμβο v . Οι κινήσεις της μορφής: $0 \rightarrow v$ (ένας νέος ερευνητής τοποθετείται στο γράφο) και $u \rightarrow 0$ (ένας ερευνητής αφαιρείται από το γράφο) ενέχουν τη φανταστική *αφετηρία* 0 (δεν είναι στοιχείο του συνόλου V), στον οποίο οι ερευνητές στέκονται όταν δεν συμμετέχουν ενεργά στην έρευνα του γράφου. Ο φυγάδας δεν έχει πρόσβαση στην αφετηρία.

Δεδομένης μιας έρευνας S , ο αριθμός των ερευνητών μέσα στο γράφο τη στιγμή t θα συμβολίζεται ως $sn(S, t)$. Ο μέγιστος αριθμός ερευνητών που χρησιμοποιεί η S θα συμβολίζεται ως $\overline{sn}(S)$, δηλαδή

$$\overline{sn}(S) = \max_t sn(S, t).$$

Αυτοί οι αριθμοί δεν πρέπει να συγχέονται με τον αριθμό ερευνητών ενός γράφου, ο οποίος θα οριστεί παρακάτω στην παράγραφο 2.2.1.

Τέλος, λέμε ότι ένας κόμβος u *φυλάσσεται* τη στιγμή t αν και μόνο αν ένας ερευνητής βρίσκεται στον u (τη στιγμή t). Ειδικά λέμε ότι ο u είναι *αφύλακτος*.

Ένα μονοπάτι $u_1 u_2 \dots u_L$ αποκαλείται *n-αφύλακτο* (node unguarded) αν και μόνο αν οι κόμβοι u_i ($i=1,2,\dots,L$) είναι αφύλακτοι (τη στιγμή t). Αλλιώς αποκαλείται *n-φυλασσόμενο* (n-guarded). Το μονοπάτι αποκαλείται *e-αφύλακτο* (edge unguarded) αν και μόνο αν οι κόμβοι u_i ($i=1,2,\dots,L$) είναι αφύλακτοι (τη στιγμή t) και *e-φυλασσόμενο* αλλιώς. Είναι προφανές ότι

$$\begin{aligned} u_1 u_2 \dots u_L \text{ είναι } e\text{-αφύλακτο} &\Rightarrow u_2 u_3 \dots u_{L-1} \text{ είναι } n\text{-αφύλακτο} \\ u_1 u_2 \dots u_L \text{ είναι } n\text{-αφύλακτο} &\Rightarrow u_1 u_2 \dots u_L \text{ είναι } e\text{-αφύλακτο} \end{aligned}$$

Ο λόγος ύπαρξης δύο ορισμών είναι ότι ο πρώτος σχετίζεται με *επαναμόλυνση κόμβων*, ενώ ο δεύτερος με *επαναμόλυνση ακμών*, όπως θα δούμε στην παράγραφο 2.2.2.

2.2 Έρευνα κόμβων και έρευνα ακμών

2.2.1 Κανόνες των παιχνιδιών

Επαναλαμβάνουμε τις υποθέσεις που εισήχθηκαν στην ενότητα 1 σχετικά με το φυγάδα. Δηλαδή, ο φυγάδας θέλει να αποφύγει τη σύλληψη, είναι αόρατος στους ερευνητές (εκτός αν είναι τοποθετημένος στον ίδιο κόμβο), γνωρίζει πάντα τις θέσεις των ερευνητών και κινείται απείρως γρήγορα. Τελική απόρροια όλων αυτών των υποθέσεων είναι ότι (και στην έρευνα κόμβων και στην έρευνα ακμών) ο φυγάδας μπορεί (και θα το κάνει) πάντα να αποφύγει τη σύλληψη εάν ένας δρόμος διαφυγής είναι διαθέσιμος. Ως εκ τούτου, από την άποψη των ερευνητών, μπορούμε να σκεφτούμε την έρευνα γράφων ως διαδικασία αποκλεισμού των δρόμων διαφυγής. Αυτό εκφράζεται ως εξής:

1. μια ακμή (ένας κόμβος) θεωρείται *βρώμικη* αν θα μπορούσε πιθανώς να περιέχει τον φυγάδα και *καθαρή* σε αντίθετη περίπτωση (π.χ. ένας κόμβος που περιέχει ερευνητή είναι καθαρός).
2. Μια ήδη καθαρή ακμή ή ένας ήδη καθαρός κόμβος ενδέχεται να ξαναγίνουν βρώμικοι (π.χ. όταν ένα προηγούμενος φυλασσόμενο μονοπάτι μεταξύ ενός καθαρού και ενός βρώμικου κόμβου γίνει ξαφνικά αφύλακτο). Η περίπτωση αυτή ονομάζεται *επαναμόλυνση*.
3. Έρευνα γράφου είναι η διαδικασία κατά την οποία σταδιακά μειώνεται το *μολυσμένο σύνολο* (από κόμβους και ακμές) ωσότου γίνει κενό (δηλαδή ο φυγάδας δεν έχει πλέον καμιά διαδρομή διαφυγής).

Τα παραπάνω αποτελούν μια *συντηρητική προσέγγιση* που στην ουσία απαλείφει τον φυγάδα από την έρευνα γράφων και των αντικαθιστά με το βρώμικο σύνολο. Έτσι, μπορούμε πλέον να αντιληφθούμε την έρευνα κόμβων (ακμών) ως ένα παιχνίδι κόμβων (ακμών) για ένα παίκτη. Στα δύο αυτά παιχνίδια ο παίκτης ελέγχει όλους τους ερευνητές, σαν τα κομμάτια σε ένα παιχνίδι σκάκι. Για λόγους που θα εξηγηθούν παρακάτω, αντικαθιστούμε τους όρους καθαρός και βρώμικος με τους όρους n-καθαρός και n-βρώμικος (στο παιχνίδι κόμβων) και e-καθαρός και e-βρώμικος (στο παιχνίδι ακμών).

Κανόνες του παιχνιδιού κόμβων

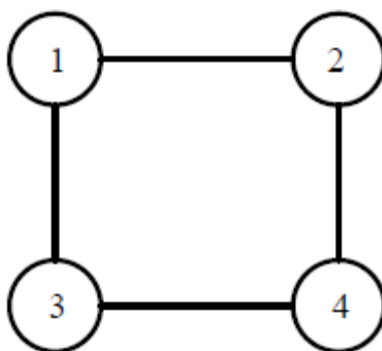
- N1.** Σε χρόνο $t = 0$ όλοι οι κόμβοι είναι *n-βρώμικοι* και κανένας ερευνητής δεν υπάρχει εντός του γράφου.
- N2.** Σε χρόνους $t = 1, 2, \dots$ ο παίκτης εκτελεί *μία* από τις παρακάτω κινήσεις:
- N2a.** τοποθέτηση ερευνητή σε κόμβο,
 - N2b.** αφαίρεση ερευνητή από κόμβο,
 - N2c.** ολίσθηση ενός ερευνητή κατά μήκος μιας ακμής.
- N3.** Ένας *n-βρώμικος* κόμβος γίνεται *n-καθαρός* όταν ένας ερευνητής τοποθετηθεί εντός του.
- N4.** Ένας *n-καθαρός* κόμβος u γίνεται *n-βρώμικος* όταν συνδέεται με κάποιον *n-βρώμικο* κόμβο v διαμέσου ενός *n-αφύλακτου μονοπατιού*.
- N5.** Μια ακμή είναι *n-βρώμικη* αν ακουμπά σε κάποιον *n-βρώμικο* κόμβο. Ειδιάλλως είναι *n-καθαρή*.
- N6.** Το παιχνίδι ολοκληρώνεται όταν όλοι οι κόμβοι (και συνεπώς όλες οι ακμές) γίνουν *n-καθαροί*.
-

Κανόνες του παιχνιδιού ακμών

- N1.** Σε χρόνο $t = 0$ όλες οι ακμές είναι *e-βρώμικες* και κανένας ερευνητής δεν υπάρχει εντός του γράφου.
- N2.** Σε χρόνους $t = 1, 2, \dots$ ο παίκτης εκτελεί *μία* από τις παρακάτω κινήσεις:
- N2a.** τοποθέτηση ερευνητή σε κόμβο,
 - N2b.** αφαίρεση ερευνητή από κόμβο,
 - N2c.** ολίσθηση ενός ερευνητή κατά μήκος μιας ακμής.
- N3.** Μια *e-βρώμικη* ακμή καθίσταται *e-καθαρή* όταν κάποιος ερευνητής την διαβεί.
- N4.** Μια *e-καθαρή* ακμή uv γίνεται *e-βρώμικη* όταν συνδέεται με κάποια *e-βρώμικη* ακμή xy μέσω ενός *e-αφύλακτου μονοπατιού*.
- N5.** Ένας κόμβος είναι *e-βρώμικος* αν είναι *αφύλακτος* και ακουμπά σε κάποια *e-βρώμικη* ακμή. Ειδιάλλως είναι *e-καθαρός*.
- N6.** Το παιχνίδι ολοκληρώνεται όταν όλες οι ακμές (και συνεπώς όλοι οι κόμβοι) γίνουν *e-καθαρές*.
-

Παρατήρηση 2.1. Ο λόγος για τη χρήση των όρων «*n-καθαρός*» και «*e-καθαρός*» (αντί απλά *καθαρός*) είναι ότι μια ακμή μπορεί να είναι *καθαρή* στο παιχνίδι κόμβων και *βρώμικη* στο παιχνίδι ακμών. Ένα παράδειγμα θα διαφωτίσει αυτό το σημείο. Εξετάστε το γράφο στο σχήμα 1 και την στρατηγική έρευνας $0 \rightarrow 1, \quad 0 \rightarrow 1, \quad 1 \rightarrow 2, \quad 2 \rightarrow 4, \quad 4 \rightarrow 3$. Εάν αυτή η στρατηγική ακολουθηθεί σε ένα παιχνίδι

κόμβων, μετά από την τελική κίνηση όλοι οι κόμβοι είναι n-καθαροί και έτσι όλες οι ακμές είναι επίσης n-καθαρές. Αλλά σε ένα παιχνίδι ακμών, μετά από την τελική κίνηση, η ακμή $\{1,3\}$ ακόμα δεν έχει διασχιστεί και ως εκ τούτου είναι ακόμα e-βρώμικη. Γενικότερα, ενώ οι όροι «n-καθαρός» και «e-καθαρός» αναφέρονται σε παρόμοιες φυσικές καταστάσεις, δεν υπάρχει κανένας a priori λόγος για να είναι οι μαθηματικοί ορισμοί τους ισοδύναμοι (στην ενότητα 2.3 θα δείξουμε ότι είναι ισοδύναμοι αν ορισμένες προϋποθέσεις ικανοποιούνται).



Σχήμα 2.1: Ένας γράφος στον οποίο ο καθαρισμός κόμβων και ο καθαρισμός ακμών δεν είναι ισοδύναμοι. Παραδείγματος χάριν, αν τοποθετήσουμε έναν ερευνητή στον κόμβο 0 ($0 \rightarrow 1$), τοποθετώντας έπειτα έναν ερευνητή στο 1 και στέλνοντας τον στους 2, 4, 3 ακολούθως ($0 \rightarrow 1$, $1 \rightarrow 2$, $2 \rightarrow 4$, $4 \rightarrow 3$) ολοκληρώνεται ο καθαρισμός κόμβων αλλά όχι και ο καθαρισμός ακμών.

Παρατήρηση 2.2.. Στο παιχνίδι ακμών οι καθαρές ακμές «καταγράφονται» στο τέλος κάθε κίνησης. Αυτό είναι σημαντικό να ειπωθεί στην περίπτωση που μια ακμή e-καθαρίζει και μετά e-βρωμίζει κατά τη διάρκεια της ίδιας κίνησης (δηλαδή για ένα συγκεκριμένο t). Για να εξηγήσουμε την παρατήρηση, υποθέτουμε ότι στο γράφο του σχήματος 2.1, οι παρακάτω κινήσεις εκτελούνται: $0 \rightarrow 1$ στο $t = 1$, $1 \rightarrow 2$ στο $t = 2$. Σε αυτή την περίπτωση η ακμή $\{1,2\}$ αρχικά e-καθαρίζει (γιατί διασχίζεται) και ακολούθως e-βρωμίζει, πάλι σε χρόνο $t = 2$. Μετά το πέρας της δεύτερης κίνησης, η ακμή $\{1,2\}$ είναι e-βρώμικη.

Θα παρατεθεί τώρα ένας πιο λεπτομερής συμβολισμός σχετικά με καθαρούς (βρώμικους) κόμβους (ακμές). Για το παιχνίδι κόμβων θα ακολουθήσουμε τους παρακάτω συμβολισμούς.

$$\begin{aligned} V_N^C(t) &= \{u \in V : \text{σε χρόνο } t, \text{ ο } u \text{ είναι } n\text{-καθαρός}\}; \\ V_N^D(t) &= \{u \in V : \text{σε χρόνο } t, \text{ ο } u \text{ είναι } n\text{-βρώμικος}\}; \\ E_N^C(t) &= \{uv \in E : \text{σε χρόνο } t, \text{ η } uv \text{ είναι } n\text{-καθαρή}\}; \\ E_N^D(t) &= \{uv \in E : \text{σε χρόνο } t, \text{ η } uv \text{ είναι } n\text{-βρώμικη}\}; \\ \mathbf{G}_N^C(t) &= (V_N^C(t), E_N^C(t)) \text{ είναι ο } n\text{-καθαρός γράφος σε χρόνο } t. \end{aligned}$$

Χρησιμοποιούμε τους αντίστοιχους συμβολισμούς για το παιχνίδι ακμών, αλλά με δείκτη E (αντί για N): $V_E^C(t)$, $V_E^D(t)$, $E_E^C(t)$, $E_E^D(t)$, $\mathbf{G}_E^C(t) = (V_E^C(t), E_E^C(t))$. Προφανώς ισχύουν:

$$\begin{aligned} V_N^C(t) \cup V_N^D(t) &= V, \quad V_N^C(t) \cap V_N^D(t) = \emptyset, \quad E_N^C(t) \cup E_N^D(t) = E, \quad E_N^C(t) \cap E_N^D(t) = \emptyset, \\ V_E^C(t) \cup V_E^D(t) &= V, \quad V_E^C(t) \cap V_E^D(t) = \emptyset, \quad E_E^C(t) \cup E_E^D(t) = E, \quad E_E^C(t) \cap E_E^D(t) = \emptyset. \end{aligned}$$

Παρατήρηση 2.3. Οι ορισμοί των n -καθαρών και e -καθαρών γράφων είναι συνεπείς, δηλαδή τα $(V_N^C(t), E_N^C(t))$ και $(V_E^C(t), E_E^C(t))$ αποτελούν γράφους. Πιο συγκεκριμένα:

1. στο παιχνίδι κόμβων ο $\mathbf{G}_N^C(t)$ αποτελείται από τους n -καθαρούς κόμβους και όλες τις ακμές μεταξύ τέτοιων κόμβων (αυτές είναι ακριβώς το σύνολο $E_N^C(t)$).
2. στο παιχνίδι ακμών ο $\mathbf{G}_E^C(t)$ αποτελείται από e -καθαρές ακμές και καμία από αυτές δεν μπορεί να αφηθεί «ελεύθερη», δηλ. εάν $uv \in E_E^C(t)$ τότε $u, v \in V_E^C(t)$. Επιπλέον το σύνολο $V_E^C(t)$ μπορεί να περιέχει τους φρουρούμενους κόμβους δίπλα στις e -βρώμικες ακμές. Τέτοιοι κόμβοι μπορούν να εμφανιστούν ως απομονωμένα συστατικά του $\mathbf{G}_E^C(t)$.

Στην παρούσα εργασία θα επικεντρωθούμε σε κάποιες *περιορισμένες* εκδοχές των παιχνιδιών ακμών και κόμβων. Πιο συγκεκριμένα, οι περιορισμοί αφορούν τις επιτρεπόμενες στρατηγικές έρευνας και είναι οι ακόλουθοι:

1. Καλούμε μια στρατηγική έρευνας *ρίζωμένη* αν και μόνο αν οι ερευνητές μπορούν να τοποθετηθούν σε μοναδικό, προκαθορισμένο κόμβο u_0 , που ονομάζεται *ρίζα* της έρευνας.
2. Ονομάζουμε μια στρατηγική έρευνας *εσωτερική* αν και μόνο αν οι ερευνητές από τη στιγμή που εισέρχονται στο γράφο (a) μπορούν να κινηθούν μόνο κατά μήκος των ακμών και (b) δεν αφαιρούνται ποτέ τον το γράφο.

Δηλαδή η «διακτίνιση» (η αυθαίρετη μετακίνηση ενός ερευνητή από ένα κόμβο σε έναν άλλο, ασχέτως με την συνεκτικότητα του γράφου) απαγορεύεται.*

3. Στο παιχνίδι κόμβων αποκαλούμε μια στρατηγική έρευνας S *μονότονη* αν και μόνο αν το σύνολο των καθαρών κόμβων ποτέ δεν μειώνεται: $V_N^C(t) \subseteq V_N^C(t+1)$ για κάθε t . Με άλλα λόγια, απ' τη στιγμή που ένας κόμβος καθίσταται n -καθαρός, δεν ξαναγίνεται n -βρώμικος ποτέ. Ο ορισμός είναι όμοιος και για το παιχνίδι ακμών. Δηλαδή, η S είναι *μονότονη* σε ένα παιχνίδι ακμών αν και μόνο αν $E_E^C(t) \subseteq E_E^C(t+1)$ για κάθε t .**

4. Στο παιχνίδι κόμβων (ομοίως και στο παιχνίδι ακμών) καλούμε μια στρατηγική έρευνας *συνδεδεμένη* αν και μόνο αν ο καθαρός γράφος $G_N^C(t)$ (ή ομοίως ο $G_E^C(t)$) είναι συνδεδεμένος για $t = 1, 2, \dots$

Στην παράγραφο 2.3 θα εξετάσουμε γενικές, χωρίς περιορισμούς, έρευνες γράφων. Στην υπόλοιπη εργασία θα επικεντρωθούμε στις εσωτερικές μονότονες (internal monotone $-IM$), στις εσωτερικές συνδεδεμένες (internal connected $-IC$) και ιδίως στις εσωτερικές μονότονες συνδεδεμένες (internal monotone connected $-IMC$) έρευνες κόμβων (ακμών). Η εστίαση αυτή είναι αποτέλεσμα του ενδιαφέροντος σε πρακτικά προβλήματα *καταδίωξης / διαφυγής* που εμφανίζονται στον τομέα της ρομποτικής. Η φύση των προβλημάτων αυτών επιβάλλει τους περιορισμούς στην έρευνα.

1. Ο περιορισμός της εσωτερικότητας προκύπτει από το γεγονός ότι ένα ρομπότ δεν μπορεί να «διακτινιστεί». Μπορεί μόνο να κινείται ανάμεσα σε δωμάτια και τις πόρτες που τα συνδέουν. Στο μοντέλο του γράφου, αυτό το γεγονός αντιστοιχεί στη χρήση μόνο κινήσεων ολίσθησης.
2. Παρομοίως, η ύπαρξη ρίζας υποδηλώνει το γεγονός ότι συνήθως ένα περιβάλλον έχει μία μοναδική είσοδο από την οποία τα ρομπότ μπορούν να εισέλθουν.

*Εφόσον κανείς ερευνητής δεν απομακρύνεται από το γράφο κατά τη διάρκεια μιας εσωτερικής έρευνας S , έχουμε $\overline{sn}(S) = sn(S, t_{fin})$ με t_{fin} τη χρονική διάρκεια της έρευνας.

**Μια σημαντική λεπτομέρεια πρέπει να αναφερθεί σε αυτό το σημείο. Υπάρχει μια ιδιότητα που λέγεται *ισχυρή μονοτονία*, που χαρακτηρίζει στρατηγικές στις οποίες μια ακμή που έχει διασχιστεί δεν ξαναγίνεται ποτέ e -βρώμικη. Η στρατηγική της παρατήρησης 2.2 είναι μονότονη (αφού $E_E^C(1) = E_E^C(2) = \emptyset$) αλλά όχι και ισχυρά μονότονη. Αυτή η επισήμανση γίνεται για λόγους πληρότητας, στα παρακάτω δεν θα χρησιμοποιήσουμε την ισχυρή μονοτονία.

3. Η συνεκτικότητα και η μονοτονία είναι επιθυμητά (αλλά όχι υποχρεωτικά) χαρακτηριστικά σε μία έρευνα από ρομπότ. Παραδείγματος χάριν, σε ένα εχθρικό περιβάλλον είναι προτιμότερο η καθαρισμένη περιοχή να αποτελείται από μια ενιαία συνδεδεμένη περιοχή, η οποία είναι ευκολότερο να φρουρηθεί και να ελεγχθεί από πολλές περιοχές. Ομοίως, εάν ένα υψηλό κόστος συνδέεται με τον καθαρισμό μιας περιοχής, είναι επιθυμητό αυτή η περιοχή να καθαρίζεται μόνο μία φορά, δηλ., η αναζήτηση να είναι μονότονη.

Ένας δεδομένος γράφος G με πλήθος κόμβων N μπορεί σίγουρα να n -καθαριστεί χρησιμοποιώντας N ερευνητές, απλά τοποθετώντας έναν ερευνητή σε κάθε κόμβο. Συνήθως ο G μπορεί να καθαριστεί με πολύ λιγότερους ερευνητές. Ο *αριθμός έρευνας κόμβων* του G , που συμβολίζεται με $s_N(G)$, είναι ο ελάχιστος αριθμός ερευνητών που απαιτούνται για να n -καθαριστεί ο G . Με την εφαρμογή των πρόσθετων περιορισμών στην στρατηγική έρευνας, έχουμε επιπρόσθετους αριθμούς έρευνας:

$s_N^i(G)$:	ελάχιστος αριθμός ερευνητών που απαιτούνται για να καθαριστεί ο G με :	Εσωτερική στρατηγική έρευνας κόμβων.
$s_N^{im}(G)$:	-//-	IM στρατηγική έρευνας κόμβων.
$s_N^{ic}(G)$:	-//-	IC στρατηγική έρευνας κόμβων.
$s_N^{imc}(G)$:	-//-	IMC στρατηγική έρευνας κόμβων.

Για το παιχνίδι ακμών ορίζονται αντίστοιχα οι αριθμοί έρευνας ακμών: $s_E(G)$, $s_E^i(G)$, $s_E^{im}(G)$, $s_E^{ic}(G)$, $s_E^{imc}(G)$.

Στην περίπτωση της ριζωμένης έρευνας με ρίζα τον κόμβο x , ο συμβολισμός γίνεται ως εξής: $s_N(G; x)$ είναι ελάχιστος αριθμός ερευνητών που απαιτούνται για να n -καθαριστεί ο G με στρατηγική με ρίζα το x . Ομοίως για $s_E(G; x)$, $s_E^i(G; x)$ και ούτω καθ' εξής. Αν ο γράφος G είναι ριζωμένο δένδρο T_x , τότε υποθέτουμε ότι πάντα καθαρίζεται από ριζωμένες στρατηγικές. Επομένως $s_E(T_x) = s_E(T; x)$, $s_E^i(T_x) = s_E^i(T; x)$ και ούτω καθ' εξής.

Ελάχιστη ονομάζεται η στρατηγική καθαρισμού κόμβων η οποία καθαρίζει τον G χρησιμοποιώντας $s_N(G)$ ερευνητές. *Ελάχιστη* IM στρατηγική καθαρισμού κόμβων ονομάζεται η IM στρατηγική καθαρισμού που καθαρίζει τον G χρησιμοποιώντας $s_N^{im}(G)$ και παρομοίως για όλες τις άλλες περιπτώσεις, είτε έρευνας κόμβων είτε έρευνας ακμών.

2.2.2 Σύγκριση των δύο παιχνιδιών

Σε αυτή την ενότητα συγκρίνουμε την «χωρίς περιορισμούς» έρευνα ακμών και κόμβων. Με άλλα λόγια, συγκρίνουμε το αποτέλεσμα μιας στρατηγικής έρευνας S όταν χρησιμοποιείται πρώτα σε ένα παιχνίδι κόμβων και έπειτα στο παιχνίδι ακμών, όταν παίζονται και τα δύο στον ίδιο γράφο G και χωρίς προϋποθέσεις όπως την ύπαρξη ρίζας, την εσωτερικότητα, την μονοτονία ή την συνεκτικότητα. Η ανάλυση είναι μετάφραση από την [41] όπου μπορούν να βρεθούν και οι αποδείξεις των θεωρημάτων.

Η συμπύκνωση της ανάλυσης είναι το Θεώρημα 2.6 που, απλουστευμένα, λέει τα εξής: κάθε στρατηγική έρευνας S είναι τουλάχιστον τόσο αποτελεσματική στο παιχνίδι κόμβων όσο στο παιχνίδι ακμών. Ακριβέστερα, σε κάθε βήμα t από την έρευνα, το σύνολο των e -καθαρών κόμβων (και ακμών) είναι ένα υποσύνολο του συνόλου των n -καθαρών κόμβων (και ακμών). Ένα άμεσο πόρισμα είναι ότι μια στρατηγική έρευνας που καθαρίζει ένα γράφο στο παιχνίδι ακμών τον καθαρίζει επίσης και στο παιχνίδι κόμβων. Το αντίστροφο δεν ισχύει, όπως έχουμε ήδη δείξει με το αντιπαράδειγμα της Παρατήρησης 2.1.

Λήμμα 2.4. Δεδομένου ενός γράφου G και μιας στρατηγικής έρευνας S , για $t=0,1,2,\dots$ ισχύουν:

$$u \in V_E^D(t) \Rightarrow (\forall ux \in E : ux \in E_E^D(t)) \quad (1)$$

$$uv \in E_E^C(t) \Rightarrow (u, v \in V_E^C(t)) \quad (2)$$

Με άλλα λόγια: εάν ένας κόμβος u είναι e -βρώμικος, τότε όλες οι ακμές ux , που ακουμπούν στον u , είναι επίσης e -βρώμικες και εάν μια ακμή uv είναι e -καθαρή, τότε οι κόμβοι u, v είναι e -καθαροί.

Λήμμα 2.5. Εάν κάποιος κόμβος u καθίσταται e -καθαρός ακριβώς τη χρονική στιγμή t , τότε κάποιος ερευνητής εισέρχεται στον u τη στιγμή t .

Θεώρημα 2.6. Δεδομένου ενός γράφου G και μιας στρατηγικής έρευνας S , ισχύει:

$$\text{για } t = 0, 1, 2, \dots: \quad V_E^C(t) \subseteq V_N^C(t) \text{ και } E_E^C(t) \subseteq E_N^C(t). \quad (3)$$

Πόρισμα 2.7. Δεδομένου ενός γράφου G και μιας στρατηγικής έρευνας S , αν η S καθαρίζει τον γράφο στο παιχνίδι ακμών, τότε τον καθαρίζει και στο παιχνίδι κόμβων.

Παρατήρηση 2.8. Από το Θεώρημα 2.6 βλέπουμε πως η έρευνα ακμών είναι πιο «αδύναμη» από την έρευνα κόμβων, δηλαδή για κάθε στρατηγική έρευνας, σε κάθε βήμα, το σύνολο των καθαρών στοιχείων είναι στο παιχνίδι ακμών μικρότερο ή ίσο με αυτό στο παιχνίδι κόμβων. Υπάρχει μια παραλλαγή της έρευνας ακμών, η αποκαλούμενη *μικτή έρευνα ακμών* που, όπως αποδεικνύεται στο Παράρτημα Β της [41], είναι *ισοδύναμη* με την έρευνα κόμβων, δηλαδή οποιαδήποτε στρατηγική έρευνας παράγει τα ίδια καθαρά και βρώμικα σύνολα σε κάθε βήμα και του

παιχνιδιού ακμών και του παιχνιδιού κόμβων. Αυτό το γεγονός έχει ως συνέπεια το ότι η έρευνα κόμβων είναι NP-πλήρης (δεδομένου ότι η μικτή έρευνα ακμών είναι NP-πλήρης [5]).

2.3 Έρευνες σε δένδρα

Θα παραθέσουμε τώρα παρακάτω, χωρίς αποδείξεις, τα θεωρήματα σχετικά με την έρευνα σε δένδρα όπως δίνονται στο [41]. Οι έρευνες με τις οποίες θα ασχοληθούμε από εδώ και πέρα είναι ριζωμένες εσωτερικές, μονότονες και συνδεδεμένες (IMC) και τα θεωρήματα που ακολουθούν ισχύουν και για το παιχνίδι ακμών, και για το παιχνίδι κόμβων.

Θεώρημα 2.9. Για ένα δένδρο T και μια ριζωμένη εσωτερική στρατηγική έρευνας S , αν η S είναι μονότονη και συνδεδεμένη στο παιχνίδι κόμβων, τότε:

$$\text{για } t = 0, 1, 2, \dots: \quad V_E^C(t) = V_N^C(t) \quad \text{και} \quad E_E^C(t) = E_N^C(t)$$

και η S είναι μονότονη και συνδεδεμένη και στο παιχνίδι ακμών.

Θεώρημα 2.10. Για ένα δένδρο T και μια ριζωμένη εσωτερική στρατηγική έρευνας S με ρίζα u_0 , αν η S είναι μονότονη και συνδεδεμένη στο παιχνίδι ακμών και ικανοποιεί μία από τις παρακάτω συνθήκες

$$\Sigma 1. \quad E_E^C(1) = \emptyset, E_E^C(2) = \{u_0 v\},$$

$$\Sigma 2. \quad E_E^C(1) = \emptyset, E_E^C(2) = \emptyset, E_E^C(3) = \{u_0 v\},$$

TOTE

$$\text{για } t = 0, 1, 2, \dots: \quad V_E^C(t) = V_N^C(t) \quad \text{και} \quad E_E^C(t) = E_N^C(t)$$

και η S είναι μονότονη και συνδεδεμένη και στο παιχνίδι κόμβων.

Θεώρημα 2.11. Για κάθε δένδρο T υπάρχει IMC στρατηγική έρευνας S που το καθαρίζει στο παιχνίδι ακμών με $s_E^{ic}(T)$ ερευνητές. Επιπλέον, στην S όλοι οι ερευνητές τοποθετούνται αρχικά στον ίδιο κόμβο u_0 (δηλαδή η στρατηγική έρευνας είναι ριζωμένη) και το πρώτο βήμα (αφού τοποθετηθούν οι ερευνητές) συντελεί σε καθαρισμό ακμής που εφάπτεται στον u_0 .

Πόρισμα 2.12. Για κάθε δένδρο ισχύει $s_E^{ic}(T) = s_E^{imc}(T)$. Υπάρχει μια ριζωμένη στρατηγική έρευνας που επιτυγχάνει αυτό το όριο.

Το Θεώρημα 2.11 μπορεί να αναπτυχθεί στην παρακάτω διαδοχή ανισοτήτων.

Θεώρημα 2.13. Για κάθε δένδρο T ισχύει.

$$s_E(T) = s_E^i(T) = s_E^m(T) \leq s_E^{im}(T) = s_E^c(T) = s_E^{ic}(T) = s_E^{mc}(T) = s_E^{imc}(T) \leq 2s_E(T) - 2.$$

Επιπλέον, υπάρχουν δένδρα στα οποία η ανισότητα $s_E^m(T) \leq s_E^{im}(T)$ είναι αυστηρή.

Η Barriere παρουσιάζει στην [2] τον αλγόριθμο **Search**, ο οποίος υπολογίζει μια ελάχιστη ριζωμένη IMC στρατηγική έρευνας **S** που καθαρίζει, στο παιχνίδι ακμών, για οποιοδήποτε δένδρο **T**. Το επόμενο θεώρημα δείχνει ότι η **S** είναι ελάχιστη IMC στρατηγική που καθαρίζει το δένδρο και για το παιχνίδι κόμβων.

Θεώρημα 2.14. *Για κάθε δένδρο **T** η στρατηγική έρευνας που παράγεται από τον αλγόριθμο **Search** είναι:*

1. *Ελάχιστη IMC ριζωμένη που καθαρίζει το **T** στο παιχνίδι ακμών.*
2. *Ελάχιστη IMC ριζωμένη που καθαρίζει το **T** στο παιχνίδι κόμβων.*

Πόρισμα 2.15. *Για κάθε δένδρο **T** ισχύει $s_E^{imc}(\mathbf{T}) = s_N^{imc}(\mathbf{T})$.*

Ο αλγόριθμος **Search** χρησιμοποιεί τον βοηθητικό αλγόριθμο **Label** που επίσης περιγράφεται στην [2]. Η Barriere χρησιμοποιεί τον αναδρομικό αλγόριθμο **Label** για να αποδώσει σε κάθε ακμή uv ενός δένδρου **T** μια βαθμολογία (η ακριβής μετάφραση του όρου είναι ετικέτα από το «λ-label») που εκφράζει τον αριθμό των ερευνητών που απαιτούνται για να καθαριστεί το υποδένδρο του **T** με ρίζα τον κόμβο v . Με άλλα λόγια, η βαθμολογία κατά Barriere, όπως θα λέμε στο εξής, μας υποδεικνύει πόσοι ερευνητές χρειάζονται για να καθαριστεί το δένδρο, από τον κόμβο v και κάτω.

3.Αλγόριθμος GSST

Σε αυτό το κεφάλαιο θα ασχοληθούμε με τις IMC έρευνες κόμβων σε ένα τυχαίο γράφο $G = (V, E)$ (που υποθέτουμε πάντα ότι έχει N κόμβους, δηλαδή $|V| = N$). Σε κάθε τέτοια έρευνα, η t -οστή κίνηση έχει την μορφή $S(t) = u \rightarrow v$, με $u, v \in V$ ή πιθανώς, $u = 0$ ο «πηγαίος» κόμβος.

Θα παρουσιαστούν αρκετές εκδοχές του βασικού n -καθαριστικού αλγορίθμου. Όλες οι εκδοχές εγγυημένα βρίσκουν μια καθαριστική στρατηγική έρευνας για το παιχνίδι κόμβων. Όπως αποδεικνύεται στην [41], δύο από της εκδοχές βρίσκουν στρατηγική έρευνας με τον ελάχιστο αριθμό ερευνητών με πιθανότητα $1 - a^M$ όπου $a \in (0, 1)$ και M ο αριθμός των επαναλήψεων του αλγορίθμου. Ακόμα και μη ελάχιστες στρατηγικές, απαιτούν σχετικά μικρό αριθμό ερευνητών, όπως φάνηκε από τα πειράματα στην [41]. Η ανάλυση και τα θεωρήματα είναι παρμένα από την τεχνική έκθεση [41]. Οι αποδείξεις των θεωρημάτων παραλείπονται.

3.1 Κεντρική Ιδέα

Η βασική ιδέα του αλγορίθμου είχε ως έναυσμα την σχετικά απλή παρατήρηση ότι *κάθε ριζωμένη IMC n -καθαριστική στρατηγική έρευνας πάνω στον G κατασκευάζει ένα γεννητορικό δένδρο*. Αυτή η παρατήρηση μπορεί να διατυπωθεί ως το θεώρημα παρακάτω.

Θεώρημα 3.1 Δεδομένου ενός γράφου $G = (V, E)$ και μιας ριζωμένης IMC n -καθαριστικής στρατηγικής έρευνας S του G , οι καθαριστικές κινήσεις της S παράγουν μια ακολουθία δένδρων, $(T_0, T_1, T_2, \dots, T_N)$, για τα οποία (για $n = 1, 2, \dots, N$) $T_n = (V_n, E_n)$ και ισχύουν τα παρακάτω:

- D1.** το T_0 είναι ο κενός γράφος ($V_0 = \emptyset, E_0 = \emptyset$).
- D2.** το T_N είναι ένα γεννητορικό δένδρο του G ($V_N = V, E_N \subseteq E$).
- D3.** για $n = 1, 2, \dots, N$: $V_{n-1} \subseteq V_n, E_{n-1} \subseteq E_n$ (με άλλα λόγια το T_{n-1} είναι υποδένδρο του T_n).
- D4.** για $n = 1, 2, \dots, N$: $V_n = V_{n-1} \cup \{u_n\}$, και για $n = 2, 3, \dots$: $E_n = E_{n-1} \cup \{u_i u_n\}$, με $i \in [1, n-1]$.

Παρατήρηση 3.2 Μια συντομότερη διατύπωση του Θεωρήματος 5.1 θα μπορούσε να είναι η εξής: «Κάθε ριζωμένη IMC n -καθαριστική στρατηγική έρευνας του G καθορίζει ένα γεννητορικό δένδρο του G και μια διαδοχή καθαρισμού των κόμβων». Όμως η διαδοχή πρέπει να είναι σύμφωνη με τη δομή των ακμών το γράφου και το γεννητορικό δένδρο (π.χ. δεν μπορούμε να θεωρήσουμε μια διαδοχή καθαρισμού των κόμβων που απαιτεί ανύπαρκτες ακμές). Αυτό το σημείο ακριβώς περιγράφεται από τις συνθήκες **D3** και **D4**.

Πόρισμα 3.3 Δεδομένου ενός γράφου $G = (V, E)$, κάθε ελάχιστη IMC n -καθαριστική έρευνα S του G , παράγει μια ακολουθία δένδρων που ικανοποιούν τις συνθήκες του Θεωρήματος 5.1.

Το Θεώρημα 3.1 και το Πόρισμα 3.3 οδηγούν στον αλγόριθμο GSST (Guaranteed Search with Spanning Trees), που περιγράφεται πρόχειρα παρακάτω σε ψευδογλώσσα.

Αλγόριθμος 1 GSST, αρχική έκδοση

Είσοδος: Γράφος G

Επίλεξε ένα γεννητορικό δένδρο T του G και μια ρίζα u_0 του T

Βρες μια ριζωμένη IMC n -καθαριστική στρατηγική έρευνας S' του T

Εφάρμοσε την S' στον G

ΑΝ σε κάποιο βήμα της S' η κίνηση $u \rightarrow v$ προκαλεί επαναμόλυνση **G TOTE**

 στείλε ένα «φρουρό» στον u

 εκτέλεσε την $u \rightarrow v$

ΤΕΛΟΣ_ΑΝ

Έξοδος: Η στρατηγική S ως συνδυασμός της S' με τις κινήσεις των φρουρών

Σημειώστε ότι, εκ κατασκευής, όλες οι κινήσεις καθαρισμού γίνονται πάνω στις ακμές του γεννητορικού δένδρου T .

Στην παραπάνω περιγραφή του αλγορίθμου χρησιμοποιείται ο όρος «φύλακας». Μπορούμε να πούμε άτυπα ότι οι ερευνητές μπορούν να έχουν δύο ρόλους: είτε είναι «ερευνητές του δένδρου» που εκτελούν τις καθαριστικές κινήσεις, πάντα πάνω στις ακμές του γεννητορικού δένδρου, είτε «φρουροί» που είναι στάσιμοι και φράσσουν πιθανές επαναμολύνσεις. Ωστόσο κάθε ερευνητής μπορεί να εναλλάσσει ρόλους κατά τη διάρκεια της έρευνας.

Το κύριο πλεονέκτημα του GSST αλγορίθμου είναι ότι είναι *γρήγορος*. Ένα τυχαίο γεννητορικό δένδρο T μπορεί να κατασκευαστεί και να ερευνηθεί πολύ γρήγορα. Δεδομένου ότι το T και ο G έχουν το ίδιο σύνολο κόμβων, ο n -καθαρισμός του T με n -μονότονο τρόπο έχει ως αποτέλεσμα τον n -καθαρισμό και του G . Το κύριο ζήτημα είναι: *πόσοι φρουροί απαιτούνται για να αποφευχθεί η επαναμόλυνση μέσω των ακμών του G που δεν ανήκουν στο δένδρο;* Ο αριθμός τους συνήθως προκύπτει σχετικά λογικός, καθώς (α) οι φρουροί μπορούν να ξαναχρησιμοποιηθούν και (β) οι ερευνητές του δένδρου γίνεται να χρησιμοποιηθούν και ως φρουροί όταν δεν εκτελούν καθαριστικές κινήσεις.

Και η φάση βαθμολόγησης κατά Barriere (των στοιχείων του γράφου) και η φάση διάσχισης του GSST μπορούν να εκτελεσθούν με είτε με συγκεντρωμένο με διανεμημένο τρόπο. Ο διανεμημένος τρόπος είναι ιδανικός για ρομποτικές εφαρμογές, που κάθε ρομπότ μπορεί να αναλάβει ένα μέρος του υπολογιστικού φορτίου. Στη διανεμημένη εκδοχή, όλοι οι ερευνητές έχουν κοινό γεννητορικό δένδρο και τις βαθμολογίες των στοιχείων. Όταν ένας ερευνητής φθάνει σε έναν κόμβο, ελέγχει εάν μπορεί να κινηθεί χωρίς επαναμόλυνση. Αν μπορεί, αποφασίζει την επόμενη του κίνηση βάσει της στρατηγικής διάσχισης και ενημερώνει την υπόλοιπη ομάδα για την κίνηση αυτή.

Συμπερασματικά, ο GSST έχει μικρό χρόνο εκτέλεσης και μπορεί να τρέχει συνεχώς (σε λογικά πλαίσια χρόνου), χρησιμοποιώντας κάθε φορά διαφορετικά (τυχαία επιλεγμένα) γεννητορικά δένδρα. Ο αλγόριθμος βασίζεται στην γρήγορη

ανακάλυψη κάποιου γεννητορικού δένδρου που θα οδηγήσει σε μία ελάχιστη, ή πολύ κοντά στην ελάχιστη (από άποψη αριθμού ερευνητών) στρατηγική έρευνας (Θεώρημα 3.1). Τα πειράματα που έγιναν στην [41] έδειξαν ότι ο GSST βρίσκει τέτοιου είδους στρατηγικές σε μικρό χρόνο εκτέλεσης για ένα πλήθος οικογενειών περίπλοκων γράφων.

Ένα σημαντικό χαρακτηριστικό του GSST είναι το ότι μπορεί να δώσει κάποια απάντηση σε οποιοδήποτε χρόνο εκτέλεσης. Η απάντηση βελτιώνεται όσο αυξάνεται ο χρόνος εκτέλεσης.

3.2.1 Παρουσίαση αλγορίθμου

Μια λεπτομερής περιγραφή του αλγορίθμου GSST δίνεται στον παρακάτω ψευδοκώδικα. Ο συμβολισμός $S = S | (u \rightarrow v)$ σημαίνει ότι η κίνηση $(u \rightarrow v)$ προστίθεται στην αποθηκευμένη στρατηγική έρευνας S (δηλαδή γίνεται η επόμενη κίνηση της S). Οι υπορουτίνες που εμφανίζονται θα εξηγηθούν παρακάτω.

Αλγόριθμος 2 GSST

Είσοδοι: G : ο γράφος, M : ο αριθμός των δένδρων που θα χρησιμοποιηθούν.

$$S_{min} = \emptyset$$

$$s_{min} = \infty$$

ΓΙΑ m **ΑΠΟ** 1 **ΕΩΣ** M

$$S = \emptyset$$

T = **GenerateTree**(G)

Επίλεξε τυχαία τη ρίζα u_0

$$S = S | (0 \rightarrow u_0)$$

$$V_N^C = \{u_0\}, E_N^C = \emptyset$$

$$V_N^D = V - \{u_0\}, E_N^D = E$$

$$G_N^C = (V_N^C, E_N^C)$$

$$\lambda = \mathbf{R-Label}(\mathbf{T})$$

ΟΣΟ $V_N^D \neq \emptyset$ **ΕΠΑΝΑΛΑΒΕ**

$$uv = \mathbf{SelectEdge}(G_N^C, \mathbf{T}, E_N^D, \lambda)$$

ΑΝ κάποιος ερευνητής μπορεί να διαβεί την uv χωρίς επαναμόλυνση **TOTE**

Μετακίνησε τον στο u μέσω του καθαρού γράφου

$$S = S | (u \rightarrow v)$$

$$V_C = V_N^C \cup \{v\}, E_N^C = E_C \cup \{uv\}$$

$$V_N^D = V_N^D - \{v\}, E_N^D = E_N^D - \{uv\}$$

$$G_N^C = (V_N^C, E_N^C)$$

ΑΛΛΙΩΣ

$$S = S | (0 \rightarrow u_0)$$

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ $\overline{sn}(S) < s_{min}$ **TOTE**

$$S_{min} = S$$

$$s_{min} = \overline{sn}(S)$$

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Έξοδος: n -καθαριστική στρατηγική έρευνας S_{min}

Τα παρακάτω βήματα εξηγούν τη λειτουργία του αλγορίθμου.

1. Κατασκευή τυχαίου γεννητορικού δένδρου **T** με την υπορουτίνα **GenerateTree**. Χρησιμοποιούνται δύο διαφορετικές μέθοδοι για την κατασκευή τυχαίου γεννητορικού δένδρου.
 - (a) Η μέθοδος *uniform* που είναι μια υλοποίηση του αλγορίθμου *loop erased random walk* του Wilson [37].
 - (b) Η μέθοδος *DFS* (dept first) η οποία επιλέγει μια ρίζα και με τυχαίο τρόπο διασχίζει κατά βάθος το δένδρο. Για κάθε κόμβο του γράφου, επιλέγετε τυχαία μια προσκείμενη ακμή και τοποθετείται στο γεννητορικό δένδρο. Υπάρχει μια λίστα που αποθηκεύονται οι κόμβοι που έχουμε επισκεφτεί και όταν ξαναβρεθούμε σε κάποιον κόμβο, η ακμή που μόλις διαβήκαμε σηματοδοτείται ως ακμή εκτός του δένδρου. Με αυτόν τον τρόπο απαλείφονται οι κύκλοι του γράφου και έτσι κατασκευάζεται ένα δένδρο. Όταν ένα φύλλο ολοκληρωθεί, ο αλγόριθμος επαναλαμβάνεται για να βεβαιωθούμε ότι όλοι οι κόμβοι συμπεριλαμβάνονται στο δένδρο (με άλλα λόγια ότι πρόκειται για γεννητορικό δένδρο του αρχικού γράφου). Το κίνητρο για τη χρήση αυτού του αλγορίθμου είναι η προσπάθεια να δημιουργήσουμε δένδρα που απαιτούν όσο το δυνατό λιγότερους φρουρούς. Περιμένουμε ότι ο DFS διάσχιση θα παράγει λίγους κόμβους προσκείμενους σε ακμές εκτός του δένδρου και άρα λιγότερους απαιτούμενους φρουρούς.
2. Βαθμολόγηση κατά *Barriere* των ακμών του **T** με τον αλγόριθμο **R-Label** (μια ισοδύναμη, μη αναδρομική παραλλαγή του αλγορίθμου **Label** της *Barriere* [2], που μπορεί να βρεθεί στο Appendix A της [41]).
3. Ενώ *n*-βρώμικοι κόμβοι υπάρχουν ακόμα, επιλέγεται μια *n*-βρώμικη ακμή *uv* του **T** (με τον **SelectEdge**). Κάποιος ερευνητής διασχίζει την *uv* αν δεν προκαλείται έτσι επαναμόλυνση κόμβων. Αν δεν είναι δυνατή μια τέτοια κίνηση, χρησιμοποιείται ένας καινούριος ερευνητής (που αρχικά τοποθετείται στην ρίζα) για να διασχίσει την *uv*.
4. Η διαδικασία επαναλαμβάνεται μέχρι όλοι οι κόμβοι να γίνουν *n*-καθαροί. (Επομένως ο αλγόριθμος δεν επιτρέπει επαναμόλυνση κόμβων και θα εισάγει όσους ερευνητές χρειαστεί για να την αποτρέψει).
5. Όταν όλοι οι κόμβοι έχουν *n*-καθαριστεί, έχει κατασκευαστεί μια ριζωμένη IMC στρατηγική καθαρισμού κόμβων η οποία πραγματοποιεί όλες τις καθαριστικές τις κινήσεις κατά μήκος των ακμών του **T**.
6. Επαναλαμβάνεται η διαδικασία από το βήμα 1 με ένα νέο γεννητορικό δένδρο.

7. Όταν έχει ξεπεραστεί ο μέγιστος αριθμός των δένδρων (και άρα των αντίστοιχων στρατηγικών έρευνας), ο αλγόριθμος τερματίζεται επιστρέφοντας την στρατηγική S_{min} που χρησιμοποιεί s_{min} ερευνητές, την ελάχιστη τιμή του $\overline{sn}(S)$.

Η υπορουτίνα **SelectEdge** επιλέγει μια n -βρώμικη ακμή uv που ανήκει στο T και ακουμπά στον μέχρι τώρα καθαρό γράφο G_N^C . Υπάρχουν αρκετές μέθοδοι για αυτή την επιλογή.

1. **Labeled Selection (L).** Η επόμενη ακμή uv που πρόκειται να διασχιστεί επιλέγεται σύμφωνα με την λ βαθμολόγηση της κατά Barriere. Αν ωστόσο η συγκεκριμένη κίνηση προκαλεί επαναμόλυνση επιλέγεται η επόμενη κατά Barriere ακμή. Αν σε κάποιο βήμα της έρευνας η διάσχιση οποιασδήποτε ακμής του T θα προκαλούσε επαναμόλυνση (δηλαδή κανένας ερευνητής δεν μπορεί να κινηθεί), τότε εισάγουμε έναν νέο ερευνητή στην ρίζα.
2. **Labeled Selection with Randomized Tie-breaking (LR).** Η επιλογή γίνεται με τον ίδιο τρόπο εκτός του ότι οι κατά Barriere ισοβαθμίες των ακμών επιλύονται με τυχαίο τρόπο.
3. **Randomized Selection (R).** Η επιλογή της uv γίνεται τυχαία (χωρίς τη χρήση της βαθμολόγησης κατά Barriere) με ομοιόμορφη πιθανότητα για όλες τις n -βρώμικες ακμές που ανήκουν στο T και ακουμπούν στον G_N^C :

$$Pr(uv) = \begin{cases} c & \text{αν } o \text{ } u \text{ είναι } n\text{-καθαρός και η } uv \text{ είναι } n\text{-βρώμικη ακμή του } T; \\ 0 & \text{αλλιώς.} \end{cases}$$

4. **Labeled Weighted Selection (LW).** Ο τρόπος επιλογής είναι το ενδιαμέσο μεταξύ R και L : η επιλογή της ακμής εξακολουθεί να είναι τυχαία αλλά αντί για ομοιόμορφη κατανομή πιθανοτήτων, η πιθανότητα κάθε ακμής είναι ανάλογη με την λ βαθμολόγησή της κατά Barriere.
5. **Label Dominated Selection (LD).** Επιτυγχάνεται βαθμολογώντας τις ακμές που οδηγούν σε κομμάτια του γράφου που αποτελούν δένδρα (υποδένδρα του γράφου). Κατά τη διάρκεια της έρευνας, διατηρείται μια λίστα με τους ερευνητές που μπορούν να κινηθούν χωρίς να κινδυνεύουμε για επαναμόλυνση. Αν κάποια ακμή που εφάπτεται στο V_C οδηγεί σε ένα υποδένδρο του γράφου, και υπάρχουν αρκετοί ερευνητές ελεύθεροι, ο καθαρισμός αυτού του υποδένδρου μπορεί μόνο βελτιώσει τις στρατηγικές έρευνας.

Εφαρμόζοντας καθένα από τους παραπάνω κανόνες στον «βασικό» αλγόριθμο GSST, προκύπτουν δέκα παραλλαγές αυτού, οι: uniform GSST-L, uniform GSST-LR, ... , uniform GSST-LD, DFS GSST-L, ..., DFS GSST-LD. Οι παραλλαγές αυτές (εκτός των δύο GSST-R) χρησιμοποιούν την βαθμολογία κατά *Barriere*, αλλά δεν παράγουν απαραίτητα διάσχιση του γεννητορικού δένδρου κατά *Barriere*. Κατά μια έννοια Από μία άποψη, η τυχαία επιλογή είναι ο απλούστερος ή πιο αφελής κανόνας που μπορεί να χρησιμοποιηθεί για να επιλεγεί η επόμενη κίνηση της στρατηγικής έρευνας: κάθε n -βρώμικη ακμή του **T** (που ακουμπά στον καθαρό γράφο) είναι εξίσου πιθανό να επιλεγεί. Οι υπόλοιποι τρεις κανόνες μπορούν θεωρηθούν μέθοδοι να επηρεαστεί η πιθανότητα με την οποία οι ακμές επιλέγονται, με ουσιώδη τρόπο. Η αποτελεσματικότητα αυτών των κανόνων εξετάστηκε από τα πειράματα της [41]. Από τη θεωρητική άποψη, ο uniform GSST-R και ο uniform GSST-LD βρίσκουν μια ελάχιστη στρατηγική καθαρισμού κόμβων με πιθανότητα $1 - \alpha^M$ όπου M είναι ο αριθμός των επαναλήψεων και $\alpha \in (0, 1)$ (Θεώρημα 3.10). Υποθέτουμε ότι αυτή η ιδιότητα δεν ισχύει για την επιλογή κατά *Barriere*. Πρακτικά, το θέμα είναι πόσο μεγάλο πρέπει να είναι το M για να είναι το $1 - \alpha^M$ αρκετά κοντά στο 1. Εντούτοις, τα πειράματα δείχνουν ότι οι παραπάνω κανόνες βρίσκουν καλές στρατηγικές καθαρισμού σε πολύ λογικό χρόνο.

Παραλλαγές GSST μπορούν επίσης να παραχθούν με την αντικατάσταση της υπορουτίνας **GenerateTree** με μια *εξαντλητική εξέταση* όλων των γεννητορικών δένδρων του **G** (μπορεί να χρησιμοποιηθεί σε συνδυασμό με οποιαδήποτε από τις παραλλαγές του **SelectEdge**). Για να γίνει αυτό χρησιμοποιήθηκε ο αλγόριθμος απαρίθμησης γεννητορικών δένδρων του Char [21]. Η εξαντλητική απαρίθμηση είναι εφικτή μόνο για τους σχετικά μικρούς γράφους. Τέλος, οι στρατηγικές έρευνας που παράγονται από κάθε παραλλαγή του GSST είναι IMC. Αυτό ισχύει για την έρευνα και στα **T** και στον **G**. Για την ακρίβεια, για να λειτουργήσει πάνω στον **G**, η στρατηγική έρευνας στα **T** πρέπει να είναι IMC. Με άλλα λόγια, καμία προφανής τροποποίηση του GSST δεν θα παράγει, π.χ. μια εσωτερική, συνδεδεμένη, μη-μονότονη n -καθαριστική στρατηγική έρευνας του **G**.

Κλείνοντας το κεφάλαιο, θα τονίσουμε ότι η βασική ιδέα του GSST είναι να εκτελεστούν όλες οι καθαριστικές κινήσεις κατά μήκος των ακμών ενός γεννητορικού δένδρου. Αυτή η ιδέα εκμεταλλεύεται τα γεγονότα ότι (α) τα γεννητορικά δέντρα μπορούν και να παραχθούν και να ερευνηθούν γρήγορα και (β) η αποτροπή επαναμόλυνσης δεν απαιτεί υπερβολικά μεγάλο αριθμό φρουρών (επειδή ένας ερευνητής μπορεί να εναλλάσσει ρόλους ως φρουρός και ερευνητής δέντρου).

3.2.2 Πληρότητα

Ορισμός 3.4. Δεδομένου ενός γράφου $\mathbf{G}$ και μιας στρατηγικής έρευνας $\mathbf{S}$ του $\mathbf{G}$, το σύνορο τη στιγμή t (υπό την $\mathbf{S}$) είναι το

$$V_N^F(t) = \{u : u \in V_N^C(t) \text{ και } \exists v : v \in V_N^D(t), uv \in E\},$$

δηλαδή οι n -καθαροί κόμβοι που ενώνονται με n -βρώμικους.

Λήμμα 3.5. Σε μια έρευνα κόμβων, για κάθε t , οι κόμβοι $u \in V_N^F(t)$ φυλάσσονται.

Ορισμός 3.6. Θεωρούμε μία ριζωμένη IMC n -καθαριστική στρατηγική έρευνας $\mathbf{S}$ του $\mathbf{G}$. Με $t_0 = 0$ και θεωρώντας ότι οι καθαριστικές κινήσεις της $\mathbf{S}$ γίνονται τις στιγμές $t_1, \dots, t_N$, με $t_0 = 0$. Ονομάζουμε n -οστή φάση της $\mathbf{S}$ (για $n = 1, 2, \dots, N$) τη χρονική διάρκεια $[t_{n-1} + 1, t_n]$, δηλαδή το χρονικό διάστημα μεταξύ της $(n-1)$ -οστής και n -οστής καθαριστικής κίνησης

Οι παρακάτω παρατηρήσεις είναι σχετικά προφανείς. Για $m = 2, 3, \dots$, μια ακμή-στόχος $u_i u_m$ (με $i \in [1, m-1]$) αντιστοιχεί στην m -οστή φάση (εδώ θεωρούμε ότι η ρίζα είναι ο κόμβος u_1). Καθώς $t \in [t_{m-1} + 1, t_m]$ ο αλγόριθμος μετακινεί έναν ερευνητή προς την $u_i u_m$. Για $t \in [t_{m-1} + 1, t_m - 1]$, η $u_i u_m$ είναι n -βρώμικη, ο u_i είναι n -καθαρός και ο u_m είναι n -βρώμικος. Τη στιγμή $t = t_m$ έχουμε $\mathbf{S}(t_m) = u_i \rightarrow u_m$ και οι $u_i u_m$ και u_m καθίστανται n -καθαρά.

Λήμμα 3.7. Δεδομένου ενός γράφου $\mathbf{G} = (V, E)$ και μιας ριζωμένης IMC στρατηγικής έρευνας $\mathbf{S}$ του $\mathbf{G}$, αποτέλεσμα είτε του uniform GSST-R είτε του uniform GSST-LW, θεωρούμε ότι $t_1, t_2, \dots, t_N$ είναι η στιγμές όπου γίνονται οι καθαριστικές κινήσεις. Επιπλέον $t_0 = 0$. Τότε για $n = 2, \dots, N$:

1. για $t \in [t_{n-1}, t_n - 1] : V_N^F(t) = V_N^F(t_{n-1})$.
2. για $t \in [t_{n-1}, t_n - 1] : \text{όλοι οι } u \in V_N^F(t) \text{ περιέχουν ακριβώς έναν ερευνητή εκτός έναν κόμβο, τον } u(t), \text{ που πιθανώς περιέχει δύο ερευνητές.}$
3. για $t = t_n : \text{κάθε } u \in V_N^F(t_n) \text{ περιέχει ακριβώς έναν ερευνητή.}$

Λήμμα 3.8. Δεδομένου ενός γράφου $\mathbf{G} = (V, E)$ και μιας ακολουθίας δένδρων $(\mathbf{T}_0, \mathbf{T}_1, \dots, \mathbf{T}_N)$ που ικανοποιεί της συνθήκες **D1-D4** του Θεωρήματος 5.1, οι GSST-R και GSST-LW με $M = 1$ (δηλαδή χρησιμοποιώντας ένα μόνο δένδρο) έχουν μη μηδενική πιθανότητα να κατασκευάσουν μια στρατηγική έρευνας $\mathbf{S}$ που παράγει την $(\mathbf{T}_0, \mathbf{T}_1, \dots, \mathbf{T}_N)$.

Λήμμα 3.9. Δεδομένου ενός γράφου $G=(V,E)$ και μιας ριζωμένης IMC n -καθαριστικής στρατηγικής έρευνας S του G , θεωρούμε ότι η $(T_0, T_1, \dots, T_N)$ είναι η ακολουθία δένδρων που παράγεται από την S . Αν η S' είναι μια στρατηγική που παρήχθη είτε από τον GSST-R είτε από τον GSST-LW η οποία παράγει και αυτή την ακολουθία $(T_0, T_1, \dots, T_N)$, τότε $\overline{sn}(S) \geq \overline{sn}(S')$.

Θεώρημα 3.10. Δεδομένου ενός γράφου $G=(V,E)$.

1. Ο GSST-R θα κατασκευάσει μια ελάχιστη ριζωμένη IMC n -καθαριστική στρατηγική έρευνας του G με πιθανότητα μεγαλύτερη ή ίση του $1 - \alpha_1^M$, όπου M ο αριθμός των επαναλήψεων και $\alpha_1 \in (0,1)$.
2. Ο GSST-LR θα κατασκευάσει μια ελάχιστη ριζωμένη IM n -καθαριστική στρατηγική έρευνας του G με πιθανότητα μεγαλύτερη ή ίση του $1 - \alpha_2^M$, όπου M ο αριθμός των επαναλήψεων και $\alpha_2 \in (0,1)$.

Θεώρημα 3.11. Δεδομένου ενός γράφου $G=(V,E)$ και μιας IMC n -καθαριστικής στρατηγικής έρευνας S που χρησιμοποιεί K ερευνητές, υπάρχει μια e -καθαριστική στρατηγική έρευνας S' η οποία χρησιμοποιεί είτε K είτε $K + 1$ ερευνητές.

3.3 Συμπεράσματα προηγούμενων πειραμάτων

Στην [41] πραγματοποιήθηκε μεγάλος αριθμός πειραμάτων με τον GSST στις διάφορες παραλλαγές του πάνω σε μια μεγάλη γκάμα γράφων διαφόρων μορφών. Στους περισσότερους περίπλοκους γράφους, ο GSST βρήκε μια ελάχιστη στρατηγική σε σχετικά καλό χρόνο. Σε όλους τους γράφους τουλάχιστον μια πολύ κοντινή στην ελάχιστη στρατηγική υπολογίστηκε στα αρχικά στάδια εκτέλεσης του αλγορίθμου. Η ικανότητα του GSST να δίνει μια μερική απάντηση σε οποιονδήποτε χρόνο, στην ουσία μεταφράζεται στο ότι μια πολύ καλή λύση δίνεται σε σύντομο χρόνο και στο υπόλοιπο διάστημα της εκτέλεσης, η λύση βελτιώνεται συνεχώς με την πάροδο του χρόνου.

Συγκρίνοντας τις δύο μεθόδους κατασκευής γεννητορικών δένδρων, παρατηρήθηκε ότι η DFS (κατά βάθος) είναι καλύτερη για όλους τους τύπους γράφων εκτός από τα πλέγματα όπου υπερίσχυσε η uniform. Σχετικά με τις μεθόδους διάσχισης των ακμών, δεν υπάρχει κάποια που ήταν συνεχώς καλύτερη (σε όλους τους τύπους γράφων). Ωστόσο ο GSST-L ποτέ δεν υστερεί στον αριθμό ερευνητών και είναι πάντα γρήγορος. Ο GSST-LR έχει τα ίδια πλεονεκτήματα ενώ είναι ταυτόχρονα και αποδείξιμα πλήρης.

Ο GSST είναι ο μόνος υλοποιημένος αλγόριθμος που λύνει γράφους του μεγέθους και της πολυπλοκότητας αυτών των σχετικών πειραμάτων.

4.Αλγόριθμοι έρευνας ακμών

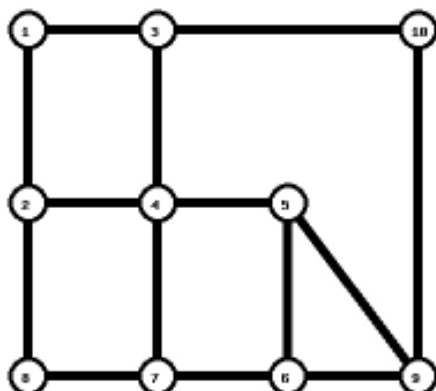
Οι αλγόριθμοι έρευνας που αναπτύχθηκαν για το παιχνίδι ακμών είναι δύο: Ο *Extra Nodes* (EN) και ο *Extra Searcher* (ES). Δεν πρόκειται, ουσιαστικά, για ανεξάρτητους αλγόριθμους καθώς οι λειτουργία και των δύο βασίζεται στον GSST. Ο *Extra Nodes* ουσιαστικά μετατρέπει το παιχνίδι ακμών σε παιχνίδι κόμβων χρησιμοποιώντας έναν ενδιάμεσο *ψευδογράφο*, ενώ ο *Extra Searcher* δρα προσθετικά στην n-καθαριστική στρατηγική έρευνας που προκύπτει από τον GSST εισάγοντας έναν ακόμα ερευνητή στο γράφο.

4.1 Αλγόριθμος *Extra Nodes*

Ο αλγόριθμος είναι ευριστικός και, όσες φορές εφαρμόστηκε, λειτούργησε κανονικά. Ωστόσο, δεν έχουμε αποδείξει ότι παράγει πάντα καθαριστικές έρευνες για το παιχνίδι ακμών.

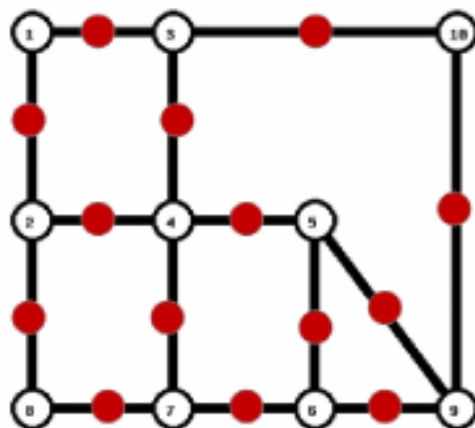
Όπως αναφέρθηκε ο *Extra Nodes* κάνει μια προεργασία στο γράφο, πριν δοθεί στον GSST, δημιουργώντας έναν *ψευδογράφο* με *ενδιάμεσους κόμβους*. Οι κόμβοι αυτοί τοποθετούνται στη μέση κάθε ακμής του γράφου.

π.χ. αν ο αρχικός γράφος είναι ο



Σχήμα 4.1: Αρχικός γράφος

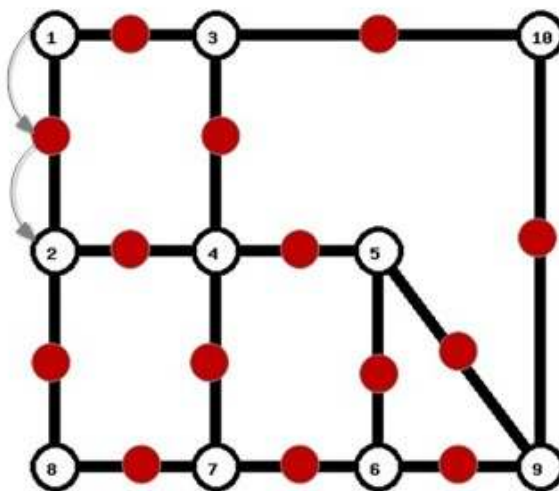
ο *ψευδογράφος* του θα προκύψει.



Σχήμα 4.2: Ο *ψευδογράφος* που προκύπτει από τον γράφο του Σχήματος 4.1 . Με κόκκινο συμβολίζονται οι ενδιάμεσοι κόμβοι

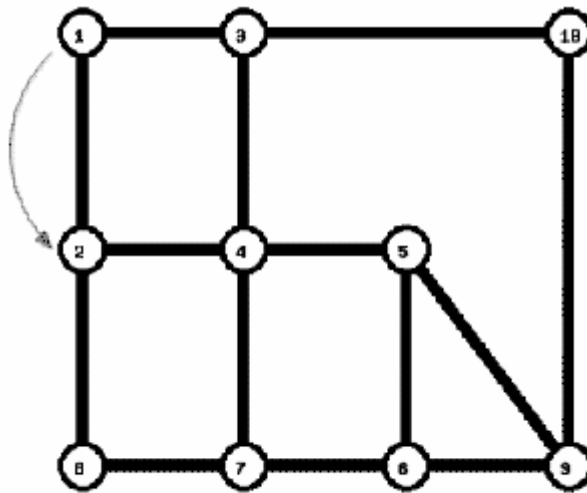
Το τρίτο στάδιο είναι ο καθαρισμός της S από τις *ενδιάμεσες κινήσεις* ώστε να προκύψει η ζητούμενη S' . Αυτό επιτυγχάνεται σε δύο βήματα:

- π.χ. οι κινήσεις.



- 34 -

θα γίνουν



Σχήμα 4.4: Η μεταφορά των κινήσεων του Σχήματος 4.3 στον αρχικό γράφο

Τελικά καταλήγουμε με την S' που είναι καθαριστική IMC στρατηγική για το παιχνίδι ακμών.

4.2 Αλγόριθμος *Extra Searcher*

4.2.1 Κεντρική Ιδέα

Η κεντρική ιδέα του *Extra Searcher* πηγάζει από το Θεώρημα 3.11 . Ο αλγόριθμος, εάν κριθεί απαραίτητο, εισάγει έναν ακόμα ερευνητή στο γράφο, τον *καθαριστή ακμών*, οποίος αναλαμβάνει να διατρέχει τις απαραίτητες e-βρώμικες ακμές ώστε σε μια n-καθαριστική έρευνα S να μην επαναμολύνεται ο γράφος ούτε για το παιχνίδι ακμών. Οι κινήσεις του λαμβάνουν χώρα *ενδιάμεσα* από τα βήματα της S ελέγχοντας κάθε φορά αν η *επόμενη* κίνηση των αρχικών ερευνητών προκαλεί επαναμόλυνση. Στο τέλος της έρευνας ο καθαριστής ακμών διατρέχει όλες τις e-βρώμικες ακμές που πιθανώς υπάρχουν. Έτσι διασφαλίζεται η μονοτονία κατ' αρχάς και η πληρότητα κατά δεύτερον του e-καθαρισμού του γράφου. Οι κινήσεις των αρχικών ερευνητών, καθώς και οι κινήσεις του καθαριστή ακμών, συνδυάζονται (με τη σωστή διαδοχή) και έτσι προκύπτει η S' που είναι καθαριστική IMC στρατηγική για το παιχνίδι ακμών.

4.2.2 Παρουσίαση αλγορίθμου

Θεωρούμε ότι στο n -οστό βήμα, μιας n -καθαριστικής μονότονης στρατηγικής $S = [S_1, S_2, \dots, S_{n_{fin}}]$, η κίνηση είναι $S_n = (u_n \rightarrow v_n)$ και γίνεται από τον ερευνητή σ_n . Ο συμβολισμός $S = S | p$ σημαίνει ότι οι κινήσεις του καθαριστή ακμών, στο μονοπάτι p προστίθενται στην αποθηκευμένη στρατηγική έρευνας S . Η θέση z του καθαριστή ακμών είναι αρχικά εκτός του γράφου. Αν και όταν γίνει απαραίτητο, ο καθαριστής ακμών εισέρχεται στον γράφο μέσω της ρίζας. Η λειτουργία του αλγορίθμου περιγράφεται με τον παρακάτω ψευδοκώδικα.

Αλγόριθμος 3 Extra Searcher

Είσοδος: ο γράφος G , στρατηγική έρευνας (n -καθαριστική) S

$n=1$;

$m=1$;

$S' = \emptyset$;

ΟΣΟ $n < n_{fin}$ **ΕΠΑΝΕΛΑΒΕ**

ΟΣΟ ο σ_n δεν είναι μόνος του στον u_n **ΕΠΑΝΑΛΑΒΕ**

$n = n + 1$;

$m = m + 1$;

$S'_m = S_n$;

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΟΣΟ $\exists \{u_n x_1, u_n x_2, \dots\} : u_n x_i \in E_E^D(n)$ με $(x_i \neq v_n)$ **ΕΠΑΝΑΛΑΒΕ**

ο καθαριστής ακμών διατρέχει την κοντινότερη $u_n x_i$;

Αυτό γίνεται διαμέσου του συντομότερου μονοπατιού p από τον z στο u_n

ή τον v_n μέσα στον e -καθαρό γράφο.*

$S' = S' | p$;

$m = m + \text{sizeof}(p)$;

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

$n = n + 1$;

$m = m + 1$;

$S'_m = S_n$;

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΟΣΟ $\exists \{e_1, e_2, \dots\} : e_i \in E_E^D(n_{fin})$ ** **ΕΠΑΝΑΛΑΒΕ**

Ο καθαριστής ακμών διατρέχει την κοντινότερη e_i στην z διαμέσου του συντομότερου μονοπατιού p μέσα στον G ;

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Έξοδος: Η στρατηγική έρευνας S' που είναι e -καθαριστική και IMC.

* Όπως θα δειχθεί παρακάτω, τέτοια μονοπάτια υπάρχουν πάντα

** Όπως θα δειχθεί, μετά την $S_{n_{fin}}$, αν υπάρχουν ακμές $e_i \in E_N^C$ θα είναι φυλασσόμενες και από τις δύο μεριές.

Όπως θα φανεί από την μαθηματική απόδειξη του αλγορίθμου, η μονοτονία του καθαρισμού στο παιχνίδι ακμών κινδυνεύει όταν αφήνεται μια e-βρώμικη ακμή αφύλακτη (από κάποιον παρακείμενο κόμβο) μέσα στον ως τώρα καθαρό γράφο. Εδώ, λέγοντας καθαρό γράφο εννοούμε τον καθαρό γράφο για το παιχνίδι ακμών (n-καθαρός), ο οποίος, όταν χρησιμοποιείται ο καθαριστής ακμών, είναι ο e-καθαρός γράφος πλην των φυλασσόμενων e-βρώμικων ακμών, κάθε στιγμή. Όπως θα δειχθεί παρακάτω, το σύνολο των e-καθαρών κόμβων και ακμών θα σχηματίσει πάντα έναν συνεκτικό γράφο, επομένως μπορούμε να χρησιμοποιήσουμε τον όρο «e-καθαρός γράφος».

Έτσι ο έλεγχος για την μονοτονία μετατίθεται στον έλεγχο για ύπαρξη τέτοιων n-καθαρών αλλά e-βρώμικων ακμών, που πρόκειται να αφεθούν αφύλακτες. Τέλος η λειτουργία του καθαριστή ακμών είναι ουσιαστικά να διατρέχει της ακμές αυτές *ακριβώς πριν αφεθούν αφύλακτες* εμποδίζοντας έτσι την επαναμόλυνση και για το παιχνίδι ακμών πλέον.

4.2.3 Μαθηματική Ανάλυση-Απόδειξη

Ορισμός 4.1. Μια στρατηγική έρευνας για το παιχνίδι ακμών σε έναν γράφο $G = (V, E)$ ονομάζεται μονότονη αν για κάθε t_1, t_2 με $t_1 \leq t_2$ ισχύει:

$$V_E^C(t_1) \subseteq V_E^C(t_2) \quad \text{και} \quad E_E^C(t_1) \subseteq E_E^C(t_2)$$

Θεώρημα 4.2. Σε κάθε μονότονη στρατηγική έρευνας ακμών S , για κάθε t ισχύουν:

$$\begin{aligned} uv \in E_E^D(t') &\Leftrightarrow (u \rightarrow v, v \rightarrow u \text{ δεν έχουν γίνει στο } t \leq t') \\ uv \in E_E^C(t') &\Leftrightarrow (u \rightarrow v \text{ ή } r \rightarrow u \text{ έχει γίνει στο } t \leq t') \\ u \in V_E^D(t') &\Leftrightarrow (\text{ο } u \text{ δεν έχει καταληφθεί στο } t \leq t') \\ u \in V_E^C(t') &\Leftrightarrow (\text{ο } u \text{ έχει καταληφθεί τουλάχιστον μια φορά στο } t \leq t') \end{aligned}$$

Απόδειξη. Θα αποδείξουμε την πρώτη πρόταση. **Αν** οι κινήσεις $u \rightarrow v, v \rightarrow u$ δεν έχουν γίνει στο $t \leq t'$ είναι προφανές ότι $uv \in E_E^D(t)$ για κάθε $t \leq t'$.

Υποθέτουμε ότι $uv \in E_E^D(t')$ και $u \rightarrow v$ τη στιγμή $t \leq t'$. Θα δείξουμε ότι

$$\forall t \leq s \leq t' : uv \in E_E^C(s)$$

το οποίο οδηγεί σε άτοπο. Εξ' αιτίας της μονοτονίας, αρκεί να δείξουμε ότι $uv \in E_E^C(t)$. Ας υποθέσουμε το αντίθετο. Αυτό σημαίνει ότι, αφού ο uv διασχίστηκε αλλά πάλι σε χρόνο t , επαναμολύνθηκε. Αν ο u εξακολουθεί να φυλάσσεται στο t (περιείχε παραπάνω από έναν ερευνητές προηγουμένως), αυτό είναι αδύνατο καθώς και u και ο v φυλάσσονται ακόμα. Αν ο u είναι αφύλακτος, τότε υπάρχει αφύλακτο μονοπάτι από κάποια βρώμικη ακμή xy προς την uv . Εφόσον, όμως ο v είναι τώρα κατειλημμένος, έχουμε $u \in V_E^C(t-1)$ και $u \in V_E^D(t)$, το οποίο παραβιάζει την μονοτονία.

Οι υπόλοιπες προτάσεις αποδεικνύονται με ανάλογο τρόπο.

Λήμμα 4.3. Σε μια μονότονη στρατηγική έρευνας ακμών, για κάθε t , $e \in E_E^D(t) \cap E_N^C(t) \Rightarrow \eta$ e φυλάσσεται στο t και από τις δυο μεριές.

Απόδειξη: Ας υποθέσουμε ότι μια ακμή e_i γίνεται n -καθαρή, για πρώτη φορά, τη στιγμή t_a . Σύμφωνα με το Θεώρημα 4.2 το γεγονός ότι $e_i \in E_E^D(t_a)$ σημαίνει ότι η e_i δεν έχει διατρεχθεί στο $t \leq t_a$. Αλλά υπάρχουν μόνο δυο τρόποι για μια ακμή να καταστεί n -καθαρή. Είτε να διατρεχθεί, είτε κάποια στιγμή να φυλάσσεται ταυτόχρονα και από τις δύο μεριές. Επομένως, αν η t_a είναι η πρώτη στιγμή που $e_i \in E_N^C$ ξέρουμε ότι απαραίτητα φυλάσσεται και από τις δύο μεριές. Αλλά δεν είναι δυνατό κάποιος εφαπτόμενος κόμβος να αφεθεί αφύλακτος πριν η e_i διατρεχθεί, καθώς πρόκειται για μονότονη στρατηγική έρευνας ακμών και άρα οι εφαπτόμενοι

κόμβοι δεν μπορούν να επαναμολυνθούν από την e_i . Έτσι, για να μετακινηθούν οι φρουροί είναι προφανές ότι πρέπει η e_i να διατρεχθεί, π.χ. σε χρόνο $t' > t_a$, και $e_i \in E_E^C(t) \forall t \geq t'$. Τα παραπάνω μας οδηγούν στο ότι αν η e_i γίνει n-καθαρή χωρίς να έχει διατρεχθεί, θα παραμείνει φυλασσόμενη ωστόσο $e_i \in E_E^C$ (δηλαδή να διατρεχθεί), σε μια μονότονη στρατηγική έρευνας ακμών.

Θεώρημα 4.4. Για τον αλγόριθμο *Extra Searcher* ισχύουν:

$$\begin{aligned} \forall t_1, t_2 \in [0, T_{fin}] \text{ με } t_1 \leq t_2 : V_E^C(t_1) \subseteq V_E^C(t_2), E_E^C(t_1) \subseteq E_E^C(t_2) \\ \forall t \in [0, T_{fin}] : V_N^C(t) = V_E^C(t) \\ \forall t \in [0, T_{fin}] : \forall x, y \in V_E^C(t) (x \neq y) \exists \text{ μονοπάτι } xz_1 \dots z_k y : \{xz_1, \dots, z_k y\} \in E_E^C(t) \end{aligned}$$

Απόδειξη: Θα αποδείξουμε με επαγωγή της παρακάτω προτάσεις για κάθε βήμα $n \in N$ της αρχικής έρευνας **S**, θεωρώντας ότι οι κινήσεις του καθαριστή ακμών λαμβάνουν χώρα σε χρόνους $n+h, n+2h, \dots, n+kh < n+I$.

$$\begin{aligned} P(n) : \forall t_1, t_2 \in [n, n+I] \text{ με } t_1 \leq t_2 : V_E^C(t_1) \subseteq V_E^C(t_2), E_E^C(t_1) \subseteq E_E^C(t_2), \\ Q(n) : \forall t \in [n, n+I] : V_N^C(t) = V_E^C(t) \\ R(n) : \forall t \in [n, n+I] : \forall x, y \in V_E^C(t) \exists \text{ μονοπάτι } xz_1 \dots z_k y : \{xz_1, \dots, z_k y\} \in E_E^C(t) \end{aligned}$$

Για $n = 0$

$$\begin{aligned} \forall t \in [0, I] : V_E^C(t) = V_N^C(t) = \emptyset \text{ και } E_E^C(t) = E_N^C(t) = \emptyset \\ V_E^C(I) = V_N^C(I) = r \text{ (η ρίζα) και } E_E^C(I) = E_N^C(I) = \emptyset \end{aligned}$$

Άρα προφανώς

$$\begin{aligned} V_E^C(t_1) \subseteq V_E^C(t_2), E_E^C(t_1) \subseteq E_E^C(t_2) \forall t_1, t_2 \in [0, I] \\ V_N^C(t) = V_E^C(t) \forall t \in [n, n+I] \end{aligned}$$

Επομένως οι $P(0), Q(0)$ και $R(0)$ ισχύουν.

Υποθέτουμε ότι οι $P(m), Q(m), R(m)$ ισχύουν για όλα τα $m \in [0, n-I]$. Ας εξετάσουμε την περίπτωση όπου $m = n$, με $n < n_{fin}$, θεωρώντας ότι στην έρευνα κόμβων **S** και σε χρόνο $t = n+I$ γίνεται η κίνηση $S(n+I) = (u \rightarrow v)$. Εξετάζουμε τις παρακάτω περιπτώσεις.

Περίπτωση I Η $S(n+I) = (u \rightarrow v)$ δεν είναι n-καθαριστική κίνηση και στο $t = n$ ο κόμβος u περιέχει έναν ακριβώς ερευνητή.

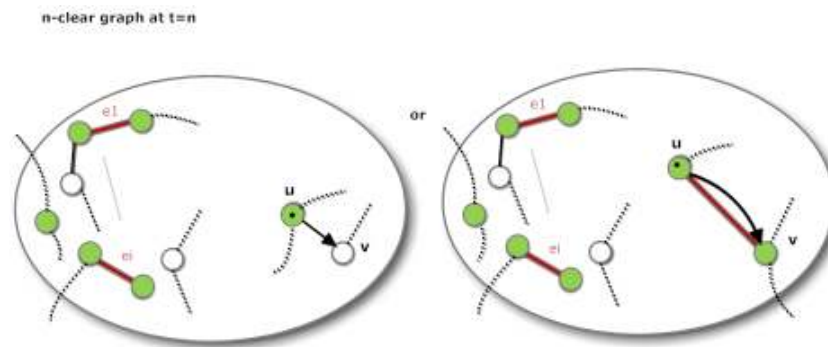
Περίπτωση I.1 Υποθέτουμε ότι δεν υπάρχουν e-βρώμικες ακμές που ακουμπούν στον u (εκτός ίσως της uv). Ο καθαριστής ακμών δεν παρεμβαίνει στην έρευνα. Τότε

$$\begin{aligned} V_E^C(n) &= V_N^C(n) \\ E_E^C(n) &= E_N^C(n) - \{e_1, e_2, \dots\} \quad \text{ή} \\ &= E_N^C(n) - \{uv, e_1, e_2, \dots\} \end{aligned}$$

όπου $e_1, e_2, \dots$ είναι οι φυλασσόμενες e-βρώμικες ακμές, μέσα στον καθαρό γράφο, οι οποίες δεν ακουμπούν στον u . Στο εξής ο συμβολισμός $e_1, e_2, \dots$ θα αναφέρεται σε αυτές τις ακμές. Έπειτα έχουμε

$$\begin{aligned} V_E^C(n+1) &= V_N^C(n+1) = V_N^C(n) = V_E^C(n) \\ E_E^C(n+1) &= E_N^C(n) - \{e_1, e_2, \dots\} = E_E^C(n) \cup \{uv\} \end{aligned}$$

Αρα οι $P(n), Q(n)$ και $R(n)$ ισχύουν.

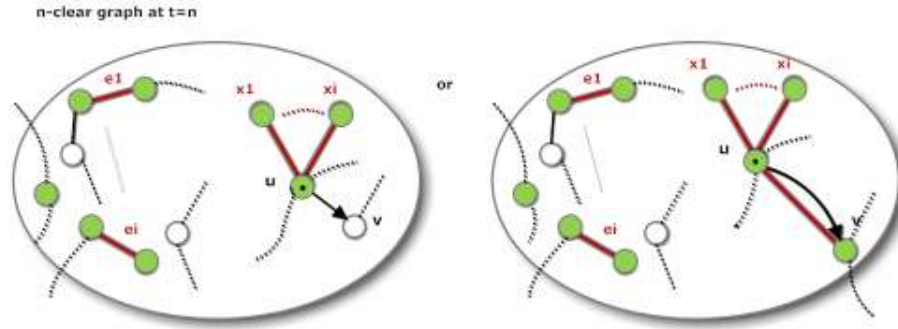


Σχήμα 4.5: Απεικόνιση της Περίπτωσης I.1

Περίπτωση I.2 Υποθέτουμε ότι υπάρχουν e-βρώμικες ακμές $ux_1, ux_2, \dots$ που ακουμπούν στον u , διαφορετικές από την uv (δηλαδή $v \notin \{x_1, x_2, \dots\}$). Πρέπει $x_i \in V_N^C(n) = V_E^C(n)$ διότι διαφορετικά ο ερευνητής στον u δεν θα μπορούσε να κινηθεί στην αρχική IMC έρευνα κόμβων. Συνεπώς από το Λήμμα 4.3, οι x_i πρέπει να φυλάσσονται στο $t=n$ από ερευνητές της αρχικής έρευνας (όχι από τον καθαριστή ακμών). Στο εξής ο συμβολισμός $ux_1, ux_2, \dots$ θα αναφέρεται σε αυτές τις ακμές. Τότε

$$\begin{aligned} V_E^C(n) &= V_N^C(n) \\ E_E^C(n) &= E_N^C(n) - \{e_1, e_2, \dots\} - \{ux_1, ux_2, \dots\} \quad \text{ή} \\ &= E_N^C(n) - \{e_1, e_2, \dots\} - \{uv, ux_1, ux_2, \dots\} \quad \text{με } (x_i \neq v) \end{aligned}$$

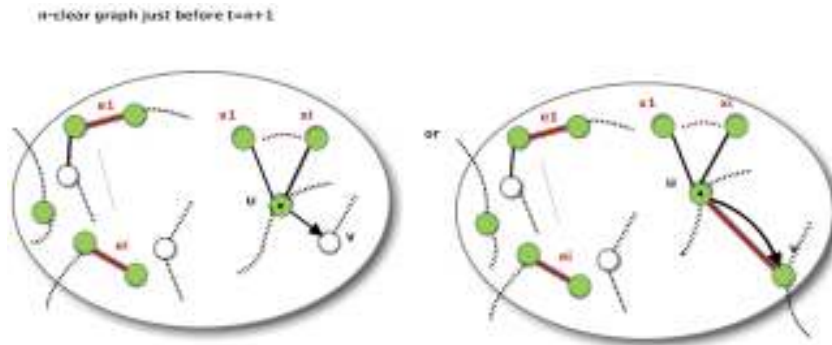
Ο καθαριστής ακμών διατρέχει όλες τις ux_i με κινήσεις του τύπου $u \rightarrow x_i$ ή $x_i \rightarrow u$. Ο καθαριστής ακμών φτάνει στον u ή στον κάθε x_i διαμέσου μόνο e-καθαρών ακμών και κόμβων.



Σχήμα 4.6: Απεικόνιση της Περίπτωσης I.2

Σύμφωνα με την $R(n-1)$ θα υπάρχει πάντα τέτοιο μονοπάτι, άρα το σύνολο $V_E^C(t)$, που συμπίπτει με το $V_N^C(t)$, παραμένει το ίδιο για κάθε $t \in [n, n+1)$. Επίσης, όλες οι ux_i παραμένουν φυλασσόμενες εκατέρωθεν μέχρι την επόμενη κίνηση της αρχικής έρευνας, έτσι γίνουν e-καθαρές (διατρεχθούν), δεν επαναμολυνονται στο $[n, n+1)$. Τότε

$$\begin{aligned} V_E^C(n+1^-) &= V_N^C(n+1^-) = V_N^C(n) \\ E_E^C(n+1^-) &= E_N^C(n) - \{e_1, e_2, \dots\} \text{ ή} \\ &= E_N^C(n) - \{uv, e_1, e_2, \dots\} \end{aligned}$$



Σχήμα 4.7: Απεικόνιση της Περίπτωσης I.2

Και τέλος

$$\begin{aligned} V_E^C(n+1) &= V_N^C(n+1) = V_N^C(n) = V_E^C(n) \\ E_E^C(n+1) &= E_N^C(n) - \{e_1, e_2, \dots\} = E_E^C(n) \cup \{uv, ux_1, ux_2, \dots\} \end{aligned}$$

Άρα οι $P(n)$, $Q(n)$ και $R(n)$ ισχύουν.

Περίπτωση II Η $S(n+1) = (u \rightarrow v)$ δεν είναι n-καθαριστική κίνηση και στο $t = n$ ο κόμβος u περιέχει παραπάνω από έναν ερευνητές. Ο καθαριστής ακμών δεν παρεμβαίνει στην έρευνα. Τότε

$$V_E^C(n) = V_N^C(n)$$

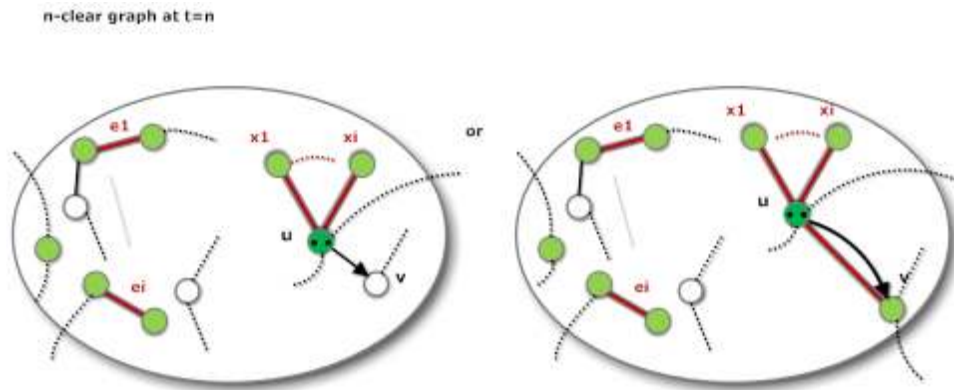
$$E_E^C(n) \left\{ \begin{array}{l} E_N^C(n) - \{e_1, e_2, \dots\} \\ \dot{\eta} \\ E_N^C(n) - \{uv, e_1, e_2, \dots\} \\ \dot{\eta} \\ E_N^C(n) - \{e_1, e_2, \dots\} - \{ux_1, ux_2, \dots\} \\ \dot{\eta} \\ E_N^C(n) - \{e_1, e_2, \dots\} - \{uv, ux_1, ux_2, \dots\} \end{array} \right.$$

Και

$$V_E^C(n+1) = V_N^C(n) = V_E^C(n)$$

$$\begin{aligned} E_E^C(n+1) &= E_N^C(n) - \{e_1, e_2, \dots\} = E_E^C(n) \cup \{uv\} \quad \dot{\eta} \\ &= E_N^C(n) - \{e_1, e_2, \dots\} - \{ux_1, ux_2, \dots\} = E_E^C(n) \cup \{uv\} \end{aligned}$$

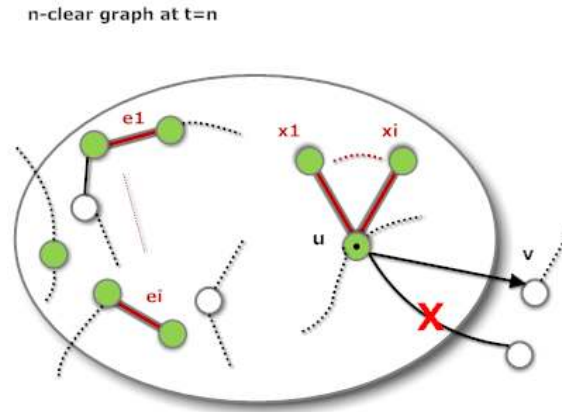
Άρα οι $P(n)$, $Q(n)$ και $R(n)$ ισχύουν.



Σχήμα 4.8: Απεικόνιση της Περίπτωσης II

Περίπτωση III Η $S(n+1) = (u \rightarrow v)$ είναι n-καθαριστική κίνηση και στο $t = n$ ο κόμβος u περιέχει ακριβώς έναν ερευνητή.

Περίπτωση III.1 Υποθέτουμε ότι ο u είναι γείτονας με παραπάνω από έναν ε-βρώμικους κόμβους. Φυσικά ο ένας από αυτούς θα είναι ο v . Αλλά η περίπτωση είναι αδύνατη, γιατί τότε η S , η αρχική έρευνα (για το παιχνίδι ακμών), δεν θα ήταν μονότονη, καθώς ο u θα ξαναγινόταν n-βρώμικος αμέσως μετά την κίνηση.



Σχήμα 4.9: Απεικόνιση της Περίπτωσης III.1

Περίπτωση III.2 Υποθέτουμε ότι ο u είναι γείτονας με ακριβώς έναν e-βρώμικο κόμβο, τον v . Επιπλέον ότι ο v είναι και αυτός γείτονας με ακριβώς έναν e-καθαρό κόμβο, τον u .

Περίπτωση III.2.i Θεωρούμε ότι η μόνη e-βρώμικη ακμή προσκείμενη στον u είναι η uv . Ο καθαριστής ακμών δεν παρεμβαίνει στην έρευνα. Τότε

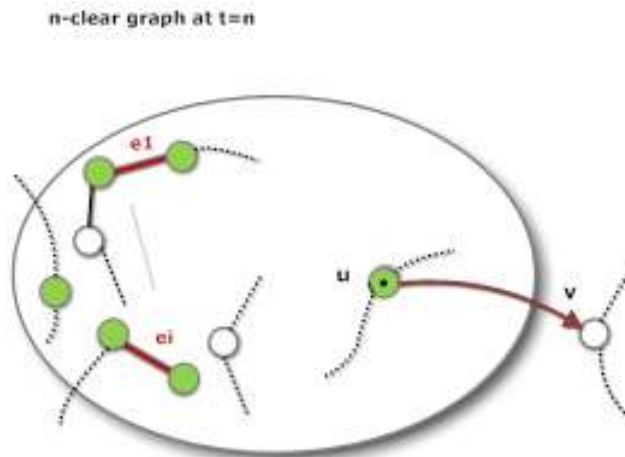
$$V_E^C(n) = V_N^C(n)$$

$$E_E^C(n) = E_N^C(n) - \{e_1, e_2, \dots\}$$

Και

$$V_E^C(n+1) = V_N^C(n+1) = V_E^C(n) \cup \{v\}$$

$$E_E^C(n+1) = E_N^C(n) \cup \{uv\} - \{e_1, e_2, \dots\} = E_E^C(n) \cup \{uv\}$$



Σχήμα 4.10: Απεικόνιση της Περίπτωσης III.2.i

Αρα οι $P(n)$ και $Q(n)$ ισχύουν. Επίσης, αφού η $R(n-1)$ ισχύει και $uv \in E_E^C(n)$, με $V_E^C(n+1) = V_E^C(n) \cup \{v\}$, υπάρχει μονοπάτι προς τον v διαμέσου του uv . Επομένως και η $R(n)$ ισχύει.

Περίπτωση III.2.ii Θεωρούμε ότι η υπάρχουν και άλλες ε-βρώμικες ακμές προσκείμενες στον u , οι $ux_1, ux_2, \dots$ με $x_i \neq v$ και $x_i \in V_E^C(n)$. Τότε

$$V_E^C(n) = V_N^C(n)$$

$$E_E^C(n) = E_N^C(n) - \{e_1, e_2, \dots\} - \{ux_1, ux_2, \dots\}$$

Ο καθαριστής ακμών διατρέχει όλες τις ux_i με κινήσεις του τύπου $u \rightarrow x_i$ ή $x_i \rightarrow u$ ωσότου

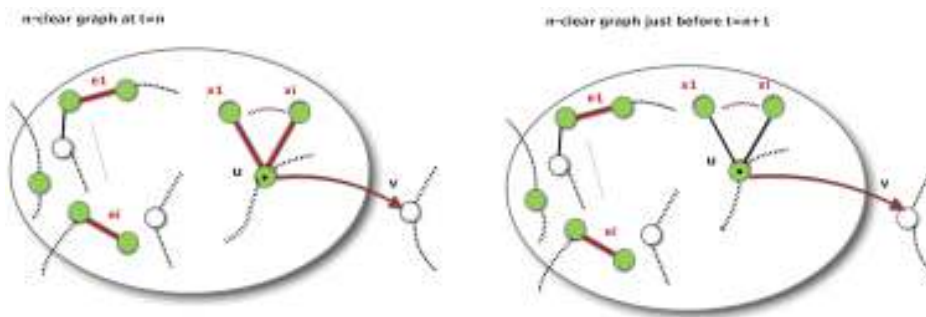
$$V_E^C(n+I^-) = V_N^C(n+I^-) = V_N^C(n)$$

$$E_E^C(n+I^-) = E_N^C(n) - \{e_1, e_2, \dots\}$$

Και τελικά

$$V_E^C(n+I) = V_N^C(n+I) = V_E^C(n) \cup \{v\}$$

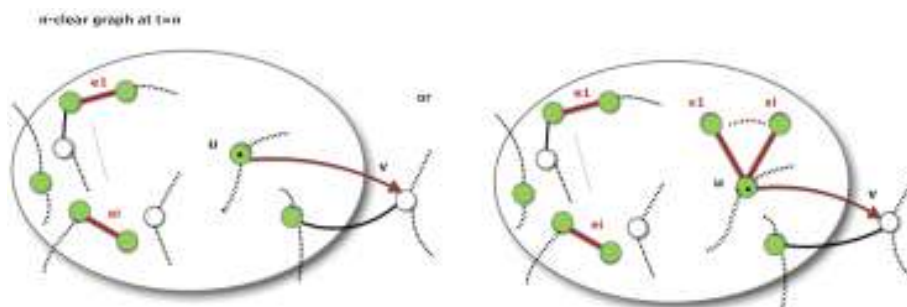
$$E_E^C(n+I) = E_N^C(n) \cup \{uv\} - \{e_1, e_2, \dots\} = E_E^C(n) \cup \{uv, ux_1, ux_2, \dots\}$$



Σχήμα 4.11: Απεικόνιση της Περίπτωσης III.2.ii

Άρα οι $P(n)$, $Q(n)$ και $R(n)$ ισχύουν.

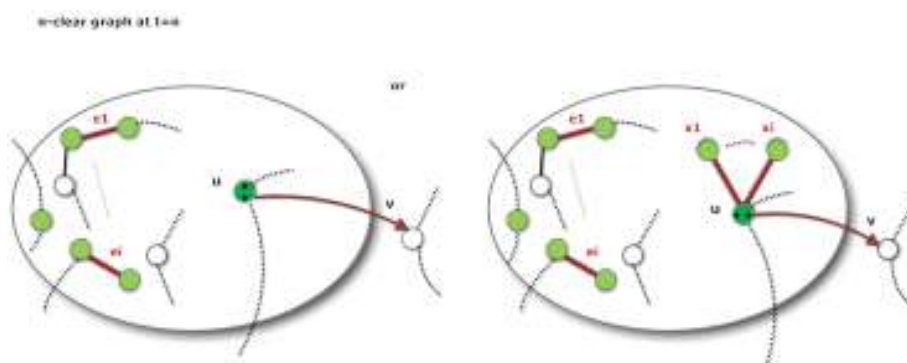
Περίπτωση III.3 Υποθέτουμε πάλι ότι ο u είναι γείτονας με ακριβώς έναν ε-βρώμικο κόμβο, τον v , αλλά ο v είναι γείτονας παραπάνω από έναν ε-καθαρούς κόμβους. Αυτή η περίπτωση όμως είναι ίδια με την III.2. Δηλαδή, οι εξισώσεις παραμένουν ίδιες, δεδομένου ότι αυτοί οι κόμβοι θα πρέπει να είναι αναγκαστικά φυλασσόμενοι.



Σχήμα 4.12: Απεικόνιση της Περίπτωσης III.3

Περίπτωση IV Η $S(n+1) = (u \rightarrow v)$ είναι n-καθαριστική κίνηση και στο $t = n$ ο κόμβος u περιέχει παραπάνω από έναν ερευνητές. Ο καθαριστής ακμών δεν ενεργοποιείται. Τότε

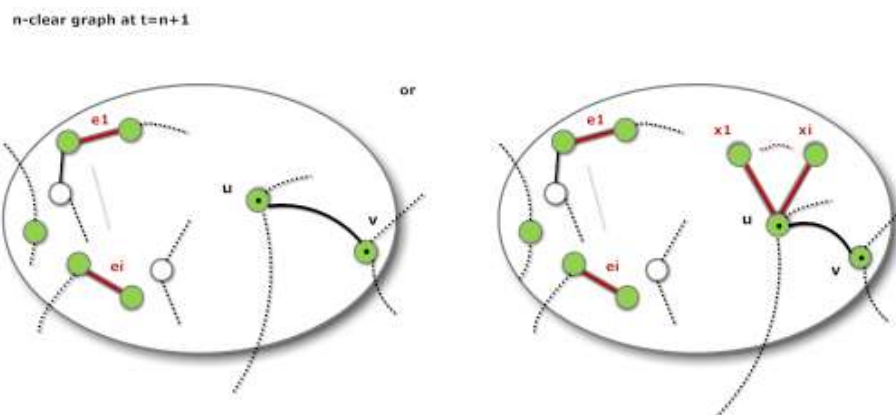
$$\begin{aligned} V_E^C(n) &= V_N^C(n) \\ E_E^C(n) &= E_N^C(n) - \{e_1, e_2, \dots\} \text{ ή} \\ &= E_N^C(n) - \{e_1, e_2, \dots\} - \{ux_1, ux_2, \dots\} \end{aligned}$$



Σχήμα 4.13: Απεικόνιση της Περίπτωσης IV

Και

$$\begin{aligned} V_E^C(n+1) &= V_N^C(n+1) = V_N^C(n) \cup \{v\} = V_E^C(n) \cup \{v\} \\ E_E^C(n+1) &= E_N^C(n) \cup \{uv\} - \{e_1, e_2, \dots\} = E_E^C(n) \cup \{uv\} \text{ ή} \\ &= E_N^C(n) \cup \{uv\} - \{e_1, e_2, \dots\} - \{ux_1, ux_2, \dots\} = E_E^C(n) \cup \{uv\} \end{aligned}$$



Σχήμα 4.14: Απεικόνιση της Περίπτωσης IV

Άρα οι $P(n)$, $Q(n)$ και $R(n)$ ισχύουν.

Επαγωγικά οι $P(n)$, $Q(n)$ και $R(n)$ για όλες τις κινήσεις της αρχικής έρευνας.

Παρατήρηση 4.5. Ως συνέπεια της $R(n)$, τα σύνολα $V_E^C(t)$ και $E_E^C(t)$ πάντα σχηματίζουν ένα συνεκτικό γράφο $G_E^C(t)$.

Θεώρημα 4.6. *Ο αλγόριθμος Extra Searcher, μετά και το τελευταίο πέρασμα του καθαριστή ακμών έχει ως αποτέλεσμα*

$$G_E^C = G$$

Απόδειξη: Σύμφωνα με το προηγούμενο θεώρημα

$$\begin{aligned} V_E^C(n_{fin}) &= V_N^C(n_{fin}) = V(G) \\ E_E^C(n_{fin}) &= E_N^C(n_{fin}) - \{e_1, e_2, \dots\} = V(G) - \{e_1, e_2, \dots\} \end{aligned}$$

Στη συνέχεια ο καθαριστής ακμών ξεκινά να διασχίζει τις $e_1, e_2, \dots$ μία προς μία, μέσω του συντομότερου μονοπατιού κάθε φορά. Προφανώς αυτές οι ακμές είναι συνεχώς φυλασσόμενες εκατέρωθεν, από ερευνητές εκτός του καθαριστή ακμών. Ας θεωρήσουμε ότι αυτές οι κινήσεις γίνονται σε k βήματα, που χρειάζονται $dt = h$ το καθένα. Τότε

$$\begin{aligned} V_E^C(n_{fin} + kh) &= V_N^C(n_{fin} + kh) = V(G) \\ E_E^C(n_{fin} + kh) &= E_E^C(n_{fin}) \cup \{e_1, e_2, \dots\} = E_N^C(n_{fin}) = E(G) \end{aligned}$$

Άρα $G_E^C(n_{fin} + kh) = G$.

5.Περιγραφή λογισμικού

5.1 Εργαλεία γραμμής εντολών

5.1.1 Gsearch.exe

Το gsearch αναπτύχθηκε από τον Geoffrey Hollinger για την πειραματική έρευνα της δημοσίευσης [41] και αποτελεί την υλοποίηση του αλγορίθμου GSST στην γλώσσα C.

Δέχεται ως ορίσματα τα εξής δεδομένα:

1. (-m) Το αρχείο που περιγράφει το γράφο. Είναι ένα απλό αρχείο κειμένου στο οποίο καταγράφονται οι ακμές του γράφου. Κάθε σειρά στο κείμενο αντιστοιχεί σε μια ακμή και περιέχει τους δύο κόμβους που ενώνει αυτή. Για παράδειγμα η ακμή {1 2} καταγράφεται ως 1 2 .
2. (-n) Τον αριθμό των γεννητορικών δέντρων που θέλουμε να αναπτυχθούν.
3. (-s) Τον κόμβο που θα θεωρηθεί ρίζα της έρευνας (με 0 θα έχουμε τυχαία επιλογή)
4. (-g) Τη μέθοδο ανάπτυξης των γεννητορικών δένδρων (όπως αναφέρονται στο κεφάλαιο 4.1.2).
5. (-y) Αν θα αποθηκεύσουμε μία, όλες ή καμία βέλτιστη στρατηγική έρευνας, από άποψη αριθμού ερευνητών. (Δείτε στις εξόδους για τη δομή των αρχείων στρατηγικής)
6. (-w) Αν θα αποθηκεύσουμε ένα, όλα ή κανένα γεννητορικό δέντρο που παράγει βέλτιστη στρατηγική. (Δείτε στις εξόδους για τη δομή των γεννητορικών δένδρων)
7. (-t) Τη μέθοδο που θα σπάνε οι δεσμοί των ακμών.
8. (-i) Αν θα βελτιστοποιούνται τα γεννητορικά δέντρα. (Η επιλογή αυτή συχνά προκαλεί προβλήματα στη μνήμη)
9. (-r) Αν θα ελέγχεται η επικάλυψη των γεννητορικών δέντρων που παράγονται. Δηλαδή αν κάποιο συγκεκριμένο δένδρο παραχθεί παραπάνω από μία φορές, θα εντοπισθεί από το πρόγραμμα και δεν θα ξαναεξεταστεί.
10. (-l) Ο επιθυμητός ελάχιστος αριθμός ερευνητών. Αν βρεθεί μια στρατηγική με αυτόν τον αριθμό η έρευνα σταματά (η προκαθορισμένη τιμή είναι 0 ώστε να διεκπεραιώνεται πλήρως η έρευνα).

Οι εξόδοι του προγράμματος είναι:

- τα αρχεία ./output/strat*.txt, όπου * ο αριθμός της αντίστοιχης βέλτιστης στρατηγικής, από άποψη αριθμού ερευνητών. Σε κάθε γραμμή του αρχείου καταγράφεται είτε ο αριθμός που αντιστοιχεί σε κάποιον κόμβο είτε ο το -1. Οι αριθμοί των κόμβων υποδεικνύουν την παρουσία ερευνητή στον αντίστοιχο κόμβο. Όταν εμφανίζεται το -1 σημαίνει ότι η έρευνα προχωρά κατά ένα βήμα.
- τα αρχεία ./output/tree*.txt, όπου * ο αριθμός του αντίστοιχου βέλτιστου γεννητορικού δένδρου. Τα αρχεία αυτά σε κάθε σειρά αποθηκεύουν μια γονική σχέση μεταξύ κόμβων στο εκάστοτε δένδρο. Δηλαδή η σειρά 2 4 θα σήμαινε ότι ο κόμβος 2 είναι γονέας του 4. Η ρίζα δηλώνεται πάντα στην πρώτη σειρά ως απόγονος της αφετηρίας 0. Έτσι μπορούμε να κατασκευάσουμε το αποθηκευμένο γεννητορικό δένδρο.

- το αρχείο ./output/count.txt που καταγράφει τα δένδρα που εμφανίζουν κάθε φορά μικρότερο αριθμό ερευνητών από το μέχρι τώρα ελάχιστο, καθώς και τον αριθμό που αντιστοιχεί σε αυτά. Κάθε φορά που εντοπίζεται ένας νέος ελάχιστος αριθμός ερευνητών, συμπληρώνεται μια νέα σειρά με αυτά τα στοιχεία.
- το αρχείο ./output/results.txt που αναφέρει τον ελάχιστο αριθμό ερευνητών που εντοπίστηκε καθώς και το χρόνο εκτέλεσης του προγράμματος.
- εκτύπωση στην κονσόλα της προόδου της έρευνας.

Το gsearch έχει μεταγλωττιστεί και για λειτουργικό linux με πρόσθετη δυνατότητα για οπτικοποίηση της έρευνας. Η βοήθεια (στα αγγλικά) για την χρήση του προγράμματος παρατίθεται παρακάτω.

```
USAGE: gsearch -m [graph] -n [no.trees] -s [startnode] -g [gen.tree]
        -y [wr-search] -w [wr-tree] -v [visualize]
        -t [edge traversal], -i [improve tree]
        -l [low number], -r [redundancy check]
```

All arguments are optional EXCEPT -m.
If an arg is omitted, the DEFAULT value is used.

EXAMPLE: gsearch -m edge11.txt -n 100 -s 0 -t bh

LIST OF ARGUMENTS

-m [graph]:	string, name of file with edge list of the graph.
-n [no.trees]:	int, how many sp.trees to generate (DEFAULT is 1).
-s [start node]:	int, which node to start (DEFAULT is 1, random choice is 0)
-g [gen.tree]:	string, method of generating spanning trees (acceptable values: readtree, exhaustive, uniform, dfsrand; DEFAULT is uniform)
-y [wr-search]:	string, how many best searches to write (acceptable values: one, all, none; DEFAULT is one)
-w [wr-tree]:	string, how many best sp.trees to write (acceptable values: one, all, none; DEFAULT is one)
-t [traversal]:	string, how to break edge label ties (acceptable values: bh, random, bhrand, bhweight, bhdom, bhdomweight; DEFAULT is bh)
-i [improve tree]:	boolean, use tree improvement technique or not. (acceptable values are 0 / 1; DEFAULT is 0, do not use it)

```

-r [redundancy check]:  boolean, check for redundant trees
                        (acceptable values are 0 / 1; DEFAULT 1,
                        check for redundancy)

-l [low number]:        break if a tree is found with this number of
                        Searchers (DEFAULT is 0; do not break)

-v [visualize]:         boolean, use visualizer (only supported on
                        linux) (DEFAULT is 0, do not use)

```

5.1.2 Eclear.exe

Αναπτύξαμε το εργαλείο αυτό για την πειραματική έρευνα της εργασίας αυτής σε γλώσσα C. Έχει τέσσερις λειτουργίες που προϋποθέτουν πάντα τη χρήση του σε συνδυασμό με το gsearch.exe. ή κάποιου προγράμματος που υλοποιεί αντίστοιχες έρευνες σε γράφους, με την ίδια δομή εισόδων και εξόδων:

1. (-i) Δέχεται ως όρισμα το αρχείο που περιγράφει τον γράφο (με τις ακμές του) και εκτυπώνει στην κονσόλα το αρχείο ενός ίδιου γράφου με ενδιαμέσους κόμβους. Αποτελεί ουσιαστικά την προεργασία του αλγορίθμου Extra Nodes. Δημιουργεί τον βοηθητικό αυτό γράφο ώστε να δοθεί στον αλγόριθμο GSST (μέσω του gsearch.exe) για έρευνα.
2. (-o) Δέχεται ως ορίσματα αρχείο που περιγράφει τον γράφο (με τις ακμές του) και ένα αρχείο στρατηγικής (strat*.txt) που όμως περιέχει και ενδιαμέσους (ψεύτικους) κόμβους. Εκτυπώνει στην κονσόλα την στρατηγική έρευνας απαλλαγμένη από τους ψεύτικους κόμβους. Αποτελεί ουσιαστικά την επεξεργασία της εξόδου του gsearch.exe που έχει εφαρμοστεί σε βοηθητικό γράφο.

Οι λειτουργίες 1 και 2 αποτελούν μαζί την υλοποίηση του αλγορίθμου Extra Nodes. Για παράδειγμα, η διαδοχή εντολών ενός τέτοιου πειράματος θα ήταν:

```

1.Eclear -i graphs\edge04.txt > FEdge.txt
2.gsearch -m FEdge.txt -n 5000 -s 2 -g uniform -t bh
3.Eclear -o graphs\edge04.txt output\strat1.txt >
realstrat.txt

```

Όπου:

- **edge04.txt** είναι το αρχείο που περιγράφει τον αρχικό γράφο (με τις ακμές του).
- **FEdge.txt** είναι το αρχείο που περιγράφει τον γράφο με τους ενδιαμέσους κόμβους.
- **strat1.txt** είναι το αρχείο στρατηγικής που παρήχθη από το gsearch.exe εφαρμοσμένο πάνω στο **FEdge.txt**. Επομένως, περιέχει και ενδιαμέσους (ψεύτικους) κόμβους.
- **realstrat.txt** είναι η στρατηγική καθαρισμένη από τους ψεύτικους κόμβους που καθαρίζει τον αρχικό γράφο στο παιχνίδι ακμών.

- 3.(-x) Δέχεται ως ορίσματα αρχείο που περιγράφει τον γράφο (με τις ακμές του) και ένα αρχείο στρατηγικής (strat*.txt) που έχει παραχθεί από το gsearch.exe και άρα καθαρίζει τον γράφο στο παιχνίδι κόμβων. Εκτυπώνει στην κονσόλα μια στρατηγική καθαρισμού του γράφου στο παιχνίδι ακμών υλοποιώντας τον αλγόριθμο Extra Searcher. Για παράδειγμα, ένα σενάριο χρήσης της λειτουργίας αυτής:

```
1.gsearch -m graphs\edge04.txt -n 5000 -s 2 -g uniform -t bh
2.Eclear -x graphs\edge04.txt output\strat1.txt >
output\strat1.edge.txt
```

Όπου:

- **edge04.txt** είναι το αρχείο που περιγράφει τον γράφο (με τις ακμές του).
- **strat1.txt** είναι το αρχείο στρατηγικής που παρήχθη από το gsearch.exe εφαρμοσμένο πάνω στο **edge04.txt**.
- **strat1.edge.txt** το αρχείο στρατηγικής που καθαρίζει τον γράφο στο παιχνίδι ακμών υλοποιώντας τον αλγόριθμο Extra Searcher.

- 4.(-t) Δέχεται ως ορίσματα αρχείο που περιγράφει τον γράφο (με τις ακμές του) και ένα αρχείο στρατηγικής (strat*.txt). Εκτυπώνει στην κονσόλα τη στρατηγική καθαρισμού χωρίς όμως *άλματα* στις κινήσεις της. Η λειτουργία αυτή δεν αποτελεί μέρος υλοποίησης κάποιου αλγορίθμου. Κάνει μόνο χρήση της συνάρτησης καθαρισμού των αλμάτων, ώστε να μην γίνεται διακίνηση των ερευνητών, για λόγους ευκολίας στην σύγκριση των διάφορων ερευνών.

Οι στρατηγικές έρευνας που παράγονται από της λειτουργίες 2,3 και 4 είναι πάντα χωρίς άλματα. Τέλος, κάθε φορά που χρησιμοποιούμε μία από αυτές, παράγεται ή συμπληρώνεται το αρχείο *Eclear_oot.txt*, *Eclear_xout.txt* ή *Eclear_tout.txt* με τον αριθμό των ερευνητών της παραγόμενης στρατηγικής και τον χρόνο εκτέλεσης του Eclear.exe. Τα αρχεία αυτά αποθηκεύονται στο φάκελο από τον οποίο εκτελείται το Eclear.

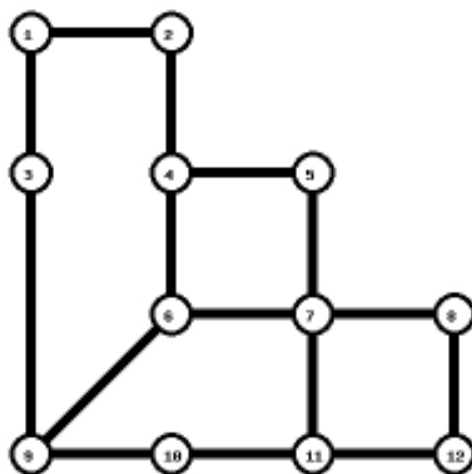
5.2 Η δυναμική βιβλιοθήκη plot.dll

Η δυναμική βιβλιοθήκη plot.dll αναπτύχθηκε σε γλώσσα C και χρησιμοποιεί την βιβλιοθήκη γραφικών ανοιχτού κώδικα GD library (<http://www.libgd.org>). Περιέχει τις συναρτήσεις που οπτικοποιούν τους γράφους και τις στρατηγικές έρευνας δηλαδή τις *plotgraph*, *plotnsearch* και *plotesearch*. Η C επιλέχθηκε για πιο άμεση και ευέλικτη χρήση της βιβλιοθήκης γραφικών και οι συναρτήσεις αναπτύχθηκαν αποκλειστικά για τη χρήση τους από το γραφικό περιβάλλον (GsearchGUI.exe) που παρουσιάζεται στο Κεφάλαιο 5.3.

5.2.1 Η συνάρτηση plotgraph

Η συνάρτηση *plotgraph* έχει ως λειτουργία τη δημιουργία αρχείου εικόνας (graph.jpg) που απεικονίζει τον γράφο που περιγράφεται από τα αρχεία edges.txt και

nodes.txt. Το πρώτο περιγράφει τον γράφο με τις ακμές του (όπως ακριβώς στο gsearch και το Eclear) και το δεύτερο περιέχει τις συντεταγμένες των κόμβων του. Τα αρχεία εισόδου πρέπει να βρίσκονται στον ίδιο φάκελο με το εκτελέσιμο, στον οποίο μάλιστα θα εμφανιστεί και το αρχείο εικόνας.

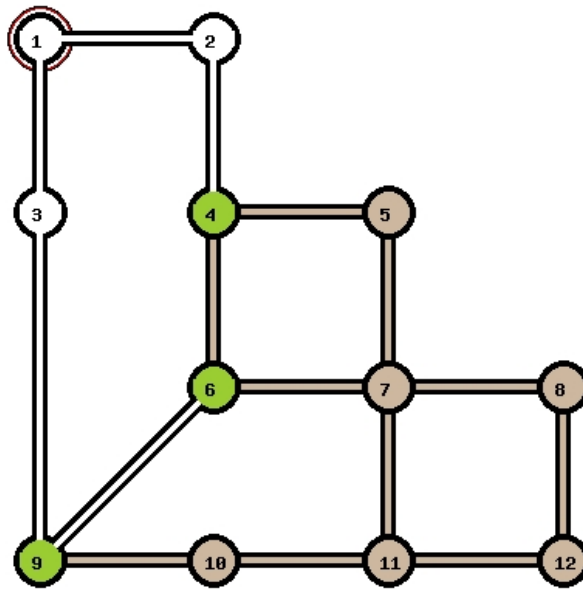


Σχήμα 5.1: Παράδειγμα παραγόμενου αρχείου εικόνας από την plotgraph.

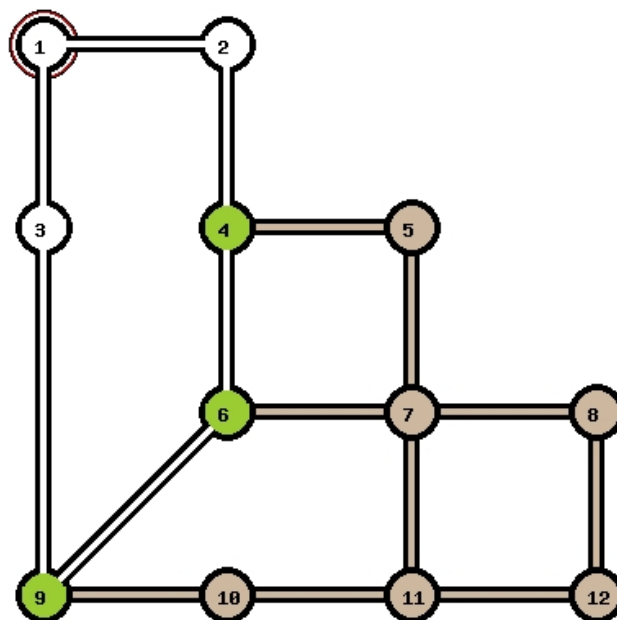
5.2.2 Οι συναρτήσεις plotnsearch και plotesearch

Οι συναρτήσεις αυτές αποτελούν την επέκταση της plotgraph στα βήματα μιας στρατηγικής έρευνας. Η λειτουργία τους είναι όμοια με αυτήν της plotgraph με τη διαφορά ότι διαβάζουν και το αρχείο στρατηγικής ./output/strat1.txt και παράγουν πολλαπλές εικόνες στο φάκελο ./pix/. Η μεταξύ διαφορά είναι ότι η μεν *plotnsearch* οπτικοποιεί το παιχνίδι κόμβων ενώ η *plotesearch* το παιχνίδι ακμών.

Ενώ η plotgraph παράγει λευκούς κόμβους και μαύρες ακμές, οι παραπάνω συναρτήσεις χρησιμοποιούν έναν χρωματικό κώδικα για να υποδηλώσουν την κατάσταση κάθε στοιχείου.



Σχήμα 5.2: Παράδειγμα εικόνας από την plotsearch.



Σχήμα 5.3: Παράδειγμα της ίδιας εικόνας για το παιχνίδι κόμβων (plotnsearch).

Χρωματικός κώδικας:

- Ο **καφέ κύκλος** δείχνει τη ρίζα της εκάστοτε έρευνας
- **λευκά** είναι τα στοιχεία (κόμβοι ή ακμές) που έχουν υπάρξει καθαρά τουλάχιστον μια φορά, μέχρι συγκεκριμένο στιγμιότυπο της έρευνας είτε για το παιχνίδι ακμών (plotesearch) είτε για το παιχνίδι κόμβων (plotnsearch)
- **γκρίζα** είναι τα στοιχεία που δεν έχουν ποτέ καθαριστεί μέχρι το συγκεκριμένο στιγμιότυπο της έρευνας.
- με **ανοικτό πράσινο** χρωματίζονται οι κόμβοι που περιέχουν έναν και μοναδικό ερευνητή στο συγκεκριμένο στιγμιότυπο της έρευνας.
- με **σκούρο πράσινο** χρωματίζονται οι κόμβοι που περιέχουν πάνω από έναν ερευνητή στο συγκεκριμένο στιγμιότυπο της έρευνας.

Παρατήρηση. Τα παραδείγματα εικόνων αναφέρονται στην ίδια στρατηγική έρευνας, για τον ίδιο γράφο, στο ίδιο βήμα αλλά με τη διαφορά στο είδος του παιχνιδιού. Έτσι ενώ στο παιχνίδι ακμών (plotesearch) η ακμή {4,6} είναι βρώμικη, την ίδια στιγμή στο παιχνίδι κόμβων (plotnsearch) θεωρείται καθαρή.

5.3 Γραφικό περιβάλλον GsearchGui.exe

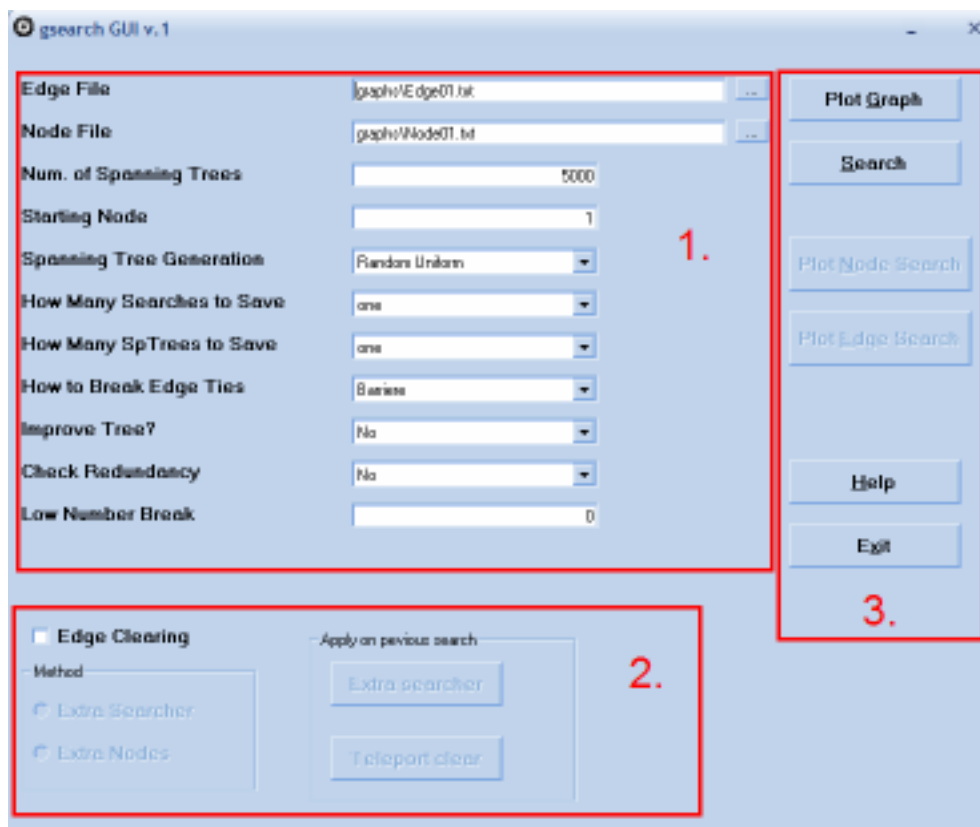
Το γραφικό περιβάλλον GsearchGUI.exe αναπτύχθηκε σε γλώσσα Visual Basic, με την δυνατότητα κλήσης των gsearch.exe, eclear.exe και των συναρτήσεων του plot.dll . Τα διάφορα εκτελέσιμα που χρησιμοποιούνται από το GsearchGUI είναι τα εξής

1. Τα εργαλεία-εντολές που προαναφέρθηκαν:
 - gsearch.exe
 - Eclear.exe
2. Το Autopics.exe, ένα δωρεάν (freeware) standalone πρόγραμμα προβολής εικόνων (<http://www.mydesktophelp.com>). Το autopics καλείται μετά τη χρήση των συναρτήσεων του plot.dll.
3. Το kill.exe ένα πρόγραμμα για τη διακοπή των εντολών του gsearch και του Eclear.

Η χρήση όλων αυτών των διαφορετικών εκτελέσιμων, αντί για την προγραμματιστική ενσωμάτωση τους σε ένα, έγινε για διάφορους σκοπούς. Αρχικά, για τα gsearch και Eclear , δεν υπάρχει λόγος για τον επαναπρογραμματισμό τους αφού έχουν ήδη αναπτυχθεί ως εργαλεία-εντολές για τη διεξαγωγή πειραμάτων. Το kill.exe για τον λόγο ότι η Visual Basic δεν υποστηρίζει multithreading και έτσι εκμεταλλευόμαστε το multitasking του λειτουργικού συστήματος. Τέλος το autopics είναι ένα στοιχείο έτοιμο για χρήση με μικρό όγκο και για λόγους ευκολίας δεν έχει νόημα η ανάπτυξη κάποιου αντίστοιχου από την αρχή.

5.3.1 To Front end

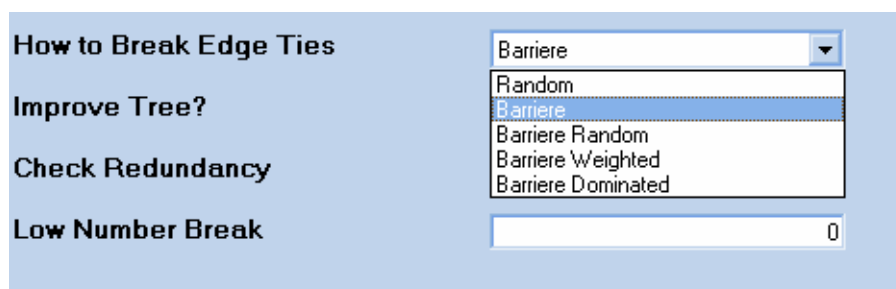
Το Front-end του προγράμματος αποτελείται από 3 περιοχές.



Σχήμα 5.4: Το Interface του GsearchGUI.exe .

1. Η περιοχή των επιλογών.

Αποτελείται από τα πεδία που αντιστοιχούν στα ορίσματα του gsearch.

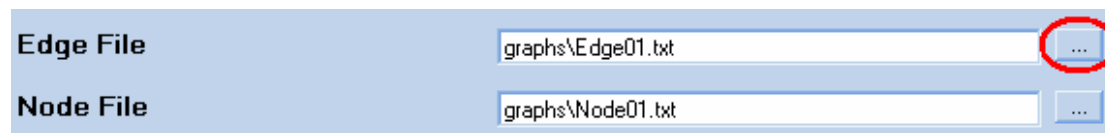


Σχήμα 5.5: Περιοχή επιλογών του gsearch στο GsearchGUI.

Αυτά έχουν είτε τη λειτουργία της επιλογής από μια προκαθορισμένη λίστα, είτε τη μορφή απλού κειμένου που όμως δέχεται μόνο αριθμούς όταν χρειάζεται (number of trees, starting node, low number break). Το πεδίο «Node File» δεν είναι

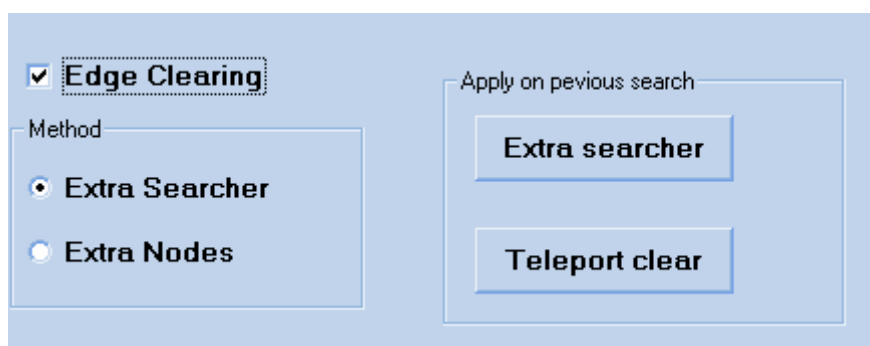
ουσιαστικά όρισμα του gsearch αλλά είναι απαραίτητο για την οπτικοποίηση του γράφου και της έρευνας (Vgraph, Vsearch).

Τέλος τα πεδία επιλογής αρχείων παρέχουν τη δυνατότητα να πλοηγηθούμε στο αρχείο από το λειτουργικό, κλικάρωντας στο μικρό κουμπάκι δεξιά τους.



Σχήμα 5.6: Επιλογές αρχείων περιγραφής γράφου στο GsearchGUI.

2. Οι λειτουργίες του Eclear.



Σχήμα 5.7: Περιοχή λειτουργιών του Eclear στο GsearchGUI.

Επιλέγοντας το checkbox «Edge Clearing» ενεργοποιείται και η επιλογή ενός από τους δύο αλγορίθμους. Ουσιαστικά κάνοντας την παραπάνω επιλογή, δηλώνουμε εξ' αρχής ότι θα εφαρμοστεί κάποιος από τους δυο αλγορίθμους στην επικείμενη έρευνα. Σε αντίθετη περίπτωση η έρευνα μας είναι απλή εφαρμογή του gsearch. Μπορούμε ωστόσο να την καθαρίσουμε από άλματα, με το κουμπί «Teleport clear» ή να εφαρμόσουμε των Extra Searcher με το ομώνυμο κουμπί. Η εκ των υστέρων εφαρμογή του Extra Nodes είναι αδύνατη καθώς ο αλγόριθμος απαιτεί προεργασία στον γράφο-είσοδο του gsearch.

3. Η περιοχή των εντολών

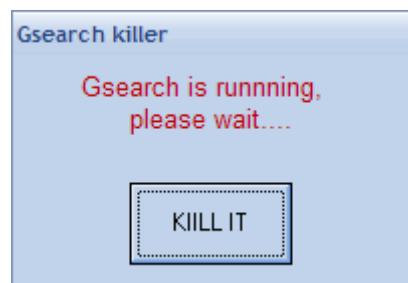
Τα κουμπιά εντολών έχουν τις εξής λειτουργίες.

- Το «Plot Graph» καλεί την συνάρτηση plotgraph από την βιβλιοθήκη plot.dll και έπειτα ανοίγει το πρόγραμμα Autopics στον φάκελο που βρίσκεται η εικόνα του γράφου.
- Το «Search» εκτελεί το gsearch με τις αντίστοιχες επιλογές που έχουν δοθεί από το χρήστη. Εάν έχουμε επιλέξει να γίνει καθαρισμός και στο παιχνίδι ακμών (τσεκάρωντας το «Edge clearing») τότε καλούνται και οι αντίστοιχες λειτουργίες του Eclear.

- Το «Plot Node Search» καλεί την συνάρτηση plotnsearch από την plot.dll και αφού δημιουργηθούν οι εικόνες ανοίγει το πρόγραμμα Autopics στον φάκελο που βρίσκονται αυτές.
- Το «Plot Edge Search» καλεί την συνάρτηση ploteseach από την plot.dll και αφού δημιουργηθούν οι εικόνες ανοίγει το πρόγραμμα Autopics στον φάκελο που βρίσκονται αυτές.
- Το «Help» μας παραπέμπει στο αρχείο βοήθειας του προγράμματος.
- Το «Exit» τερματίζει την εφαρμογή.

Τα κουμπιά αυτής της περιοχής έχουν όλα συντομεύσεις πληκτρολογίου τα γράμματα που είναι υπογραμμισμένα στο κείμενο του καθενός. Η εκτέλεση των λειτουργιών τους από το πληκτρολόγιο γίνεται με το συνδυασμό Alt+<Key> ,όπου <Key> το αντίστοιχο πλήκτρο συντόμευσης.

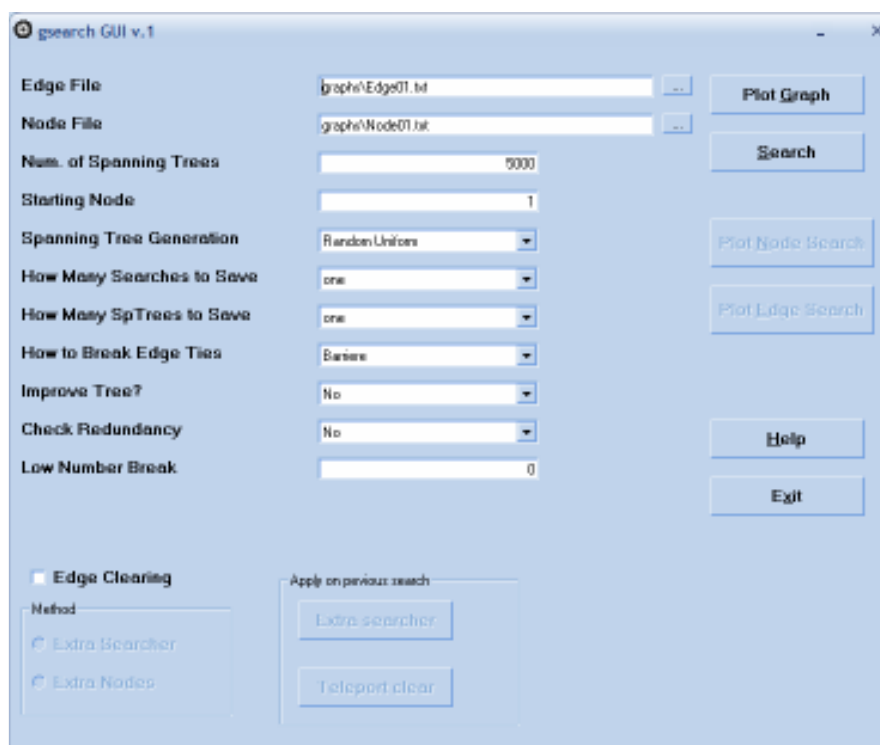
Τέλος, καθώς εκτελείται η εντολή ή η ακολουθία εντολών του «Search» εμφανίζεται στην οθόνη το kill.exe το οποίο μας παρέχει ένα κουμπί για τη διακοπή κάθε εντολής.



Σχήμα 5.8: Το kill.exe, σχεδιασμένο για να διακόπτει της λειτουργίες των gsearch.exe και Eclear.exe όταν είναι αναγκαίο.

5.3.2 Σενάρια χρήσης.

Η αρχική εικόνα του προγράμματος είναι η παρακάτω.



Σχήμα 5.9: Αρχική μορφή του GsearchGUI.

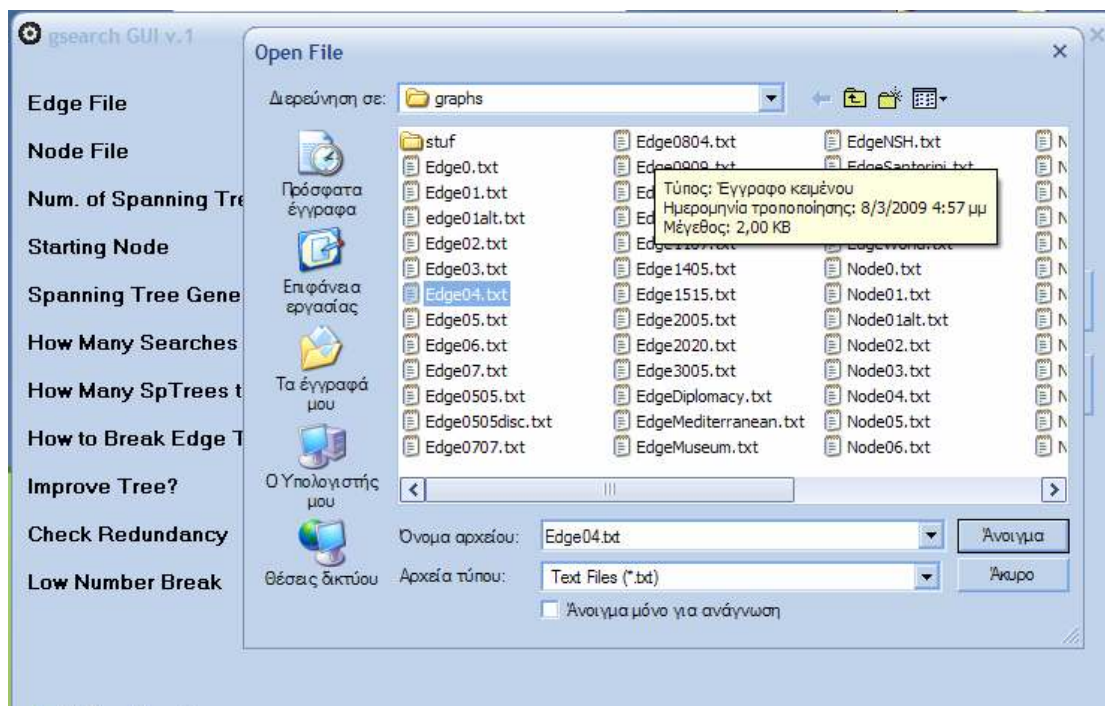
Παρατηρούμε πως οι εντολές «Plot Node Search» και «Plot Edge Search» είναι αρχικά απενεργοποιημένες καθώς δεν υπάρχει ακόμα καμία στρατηγική έρευνας. Επίσης η εντολές για επεξεργασία της έρευνας είναι ανενεργές για τον ίδιο λόγο. Τέλος η επιλογή κάποιου αλγορίθμου για καθαρισμό ακμών θα ενεργοποιηθεί μόνο εφόσον τσεκαριστεί το «Edge Clearing». Ας παρακολουθήσουμε δύο πιθανά σενάρια χρήσης του GsearchGUI.

Αν για παράδειγμα θέλουμε να εκτελέσουμε έρευνα στο παιχνίδι κόμβων πιθανή διαδοχή βημάτων είναι η εξής:

1. Επιλέγουμε τα αρχεία του γράφου.

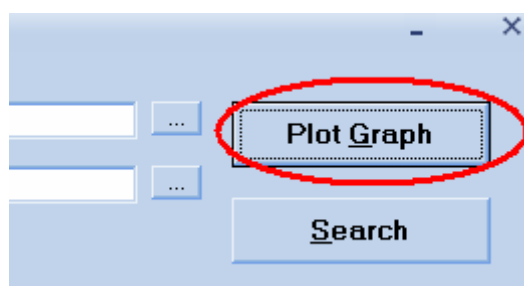


Σχήμα 5.10: Κουμπιά επιλογής αρχείων.



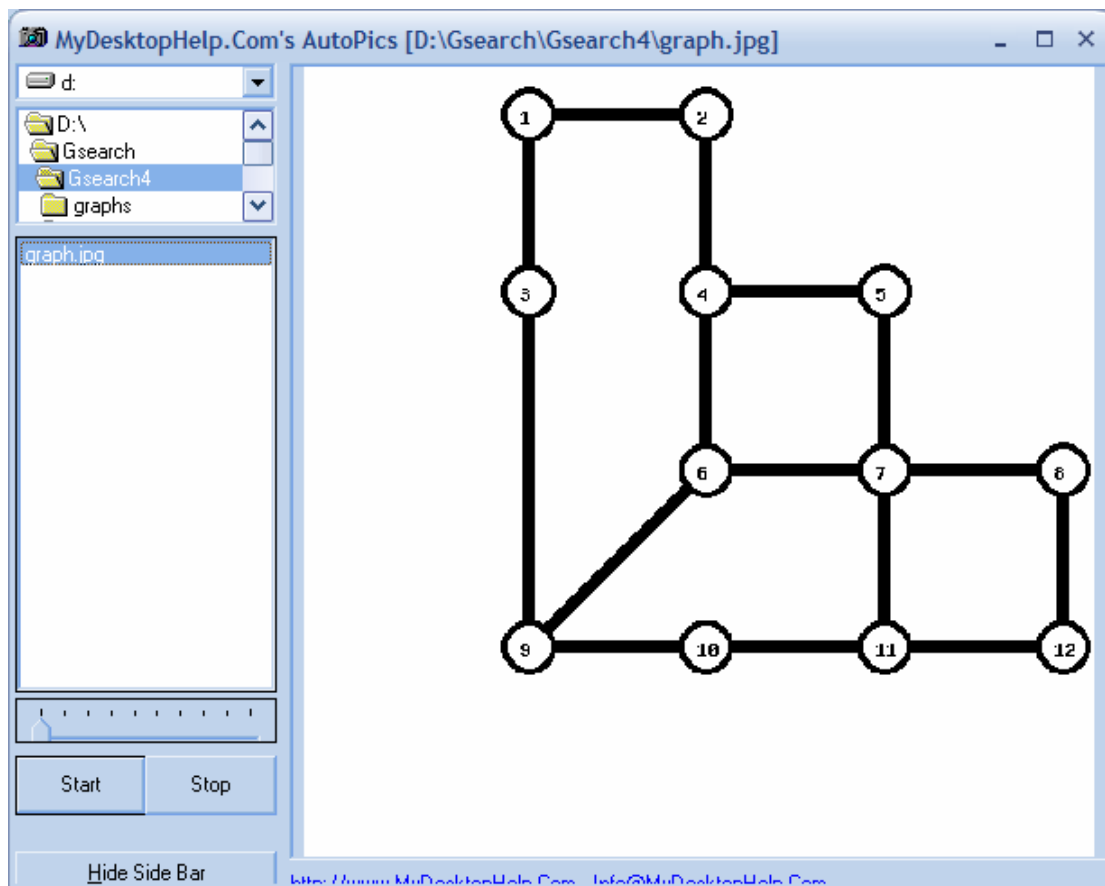
Σχήμα 5.11: Διάλογος περιήγησης στο σύστημα αρχείων.

2. Εκτελούμε την εντολή «Plot Graph» είτε από το κουμπί είτε με την συντόμευση πληκτρολογίου Alt+G.



Σχήμα 5.12: Κουμπί εκτέλεσης της εντολής vgraph.exe στο GsearchGUI.

Αφού δημιουργηθεί η εικόνα (graph.jpg) του γράφου, ανοίγεται από το πρόγραμμα προβολής Autopics.



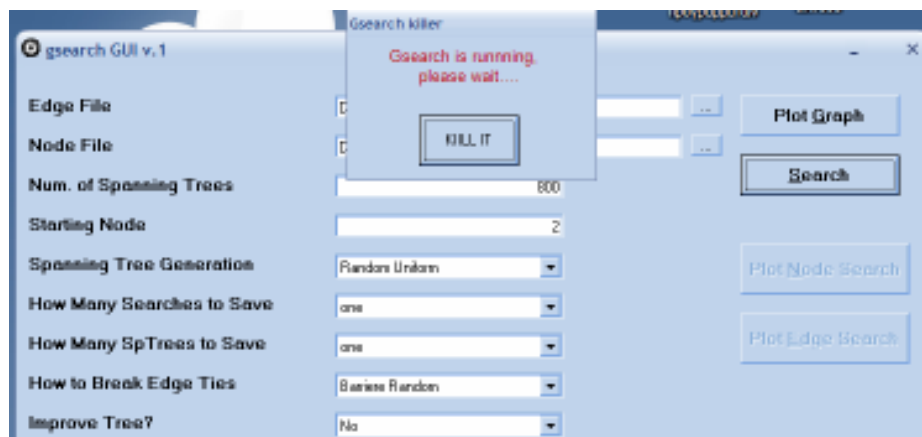
Σχήμα 5.13: Το Autopics.exe παρουσιάζει την εικόνα του γράφου, όπως παρήχθη από την plotgraph.

3.Επιλέγουμε τα επιθυμητά ορίσματα του gsearch.

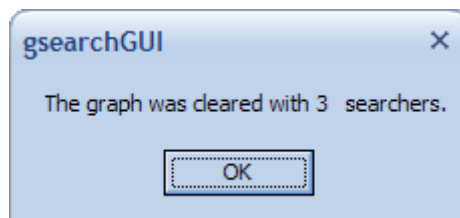
Num. of Spanning Trees	800
Starting Node	2
Spanning Tree Generation	Random Uniform
How Many Searches to Save	one
How Many SpTrees to Save	one
How to Break Edge Ties	Barriere Random
Improve Tree?	No
Check Redundancy	No
Low Number Break	0

Σχήμα 5.14: Επιλογές των ορισμάτων του gsearch.exe. Όπου αυτές είναι συγκεκριμένες, η επιλογή γίνεται μέσω Combobox λιστών που περιέχουν μόνο τις προκαθορισμένες επιλογές.

4. Εκτελούμε την εντολή «Search». Εμφανίζεται για λίγο το kill.exe και το πρόγραμμα μας ενημερώνει για τον ελάχιστο αριθμό ερευνητών που εντοπίστηκε, αφού τελειώσει η έρευνα.

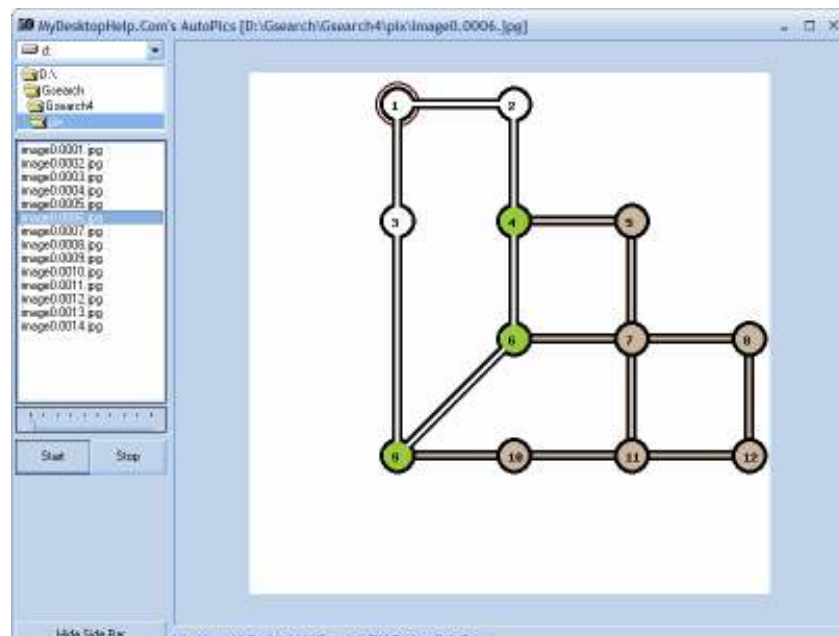


Σχήμα 5.15: Το GsearchGUI κατά τη διάρκεια εκτέλεσης του gsearch.exe.



Σχήμα 5.16: Μήνυμα τέλους εκτέλεσης του gsearch.exe.

5. Επιλέγουμε το «OK» στο παραπάνω μήνυμα και εκτελούμε την «Plot Node Search». Παρακολουθούμε την έρευνα στο Autopics.



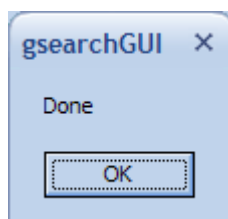
Σχήμα 5.17: Στιγμιότυπο από την παρακολούθηση της στρατηγικής έρευνας με το Autopics.exe. Οι εικόνες παρήχθησαν από την συνάρτηση plotnsearch.

Αν τώρα θέλουμε να εκτελέσουμε έρευνα στο παιχνίδι κόμβων αρχικά, και έπειτα να εφαρμόσουμε πάνω της τον αλγόριθμο Extra Searcher, μια πιθανή διαδοχή βημάτων είναι η εξής:

- 1.Επιλέγουμε τα αρχεία του γράφου.
- 2.Εκτελούμε την εντολή «Plot Graph» είτε από το κουμπί είτε με την συντόμευση πληκτρολογίου Alt+G.
- 3.Επιλέγουμε τα επιθυμητά ορίσματα του gsearch.
- 4.Εκτελούμε την εντολή «Search». Το πρόγραμμα μας ενημερώνει για τον ελάχιστο αριθμό ερευνητών που εντοπίσθηκε.

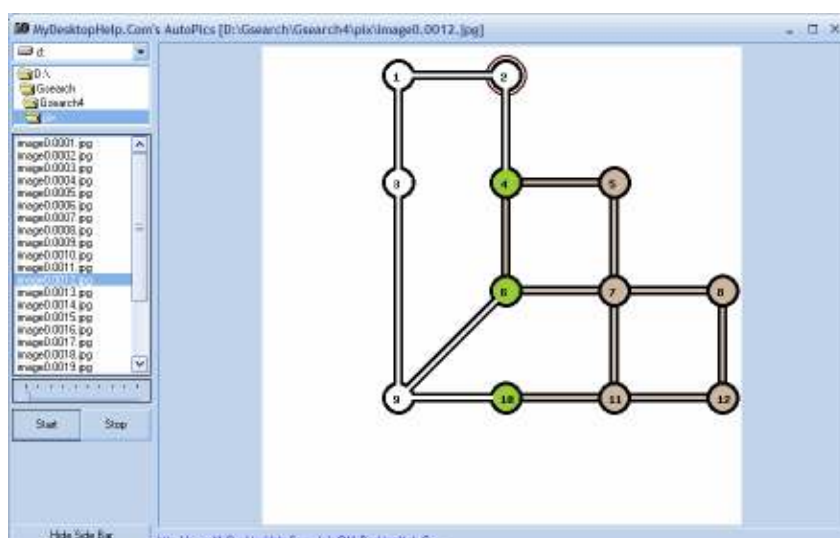
Τα βήματα 1-4 είναι πανομοιότυπα με το πρώτο σενάριο (απλή έρευνα κόμβων). Από εδώ και πέρα όμως εφαρμόζουμε πάνω στην αποθηκευμένη στρατηγική τον καθαρισμό αλμάτων (για να φανεί καλύτερα η διαφορά των δυο ερευνών) και τον Extra Searcher. Ακολουθεί η συνέχεια του σεναρίου.

5. Επιλέγουμε το «OK» στο παραπάνω μήνυμα και εκτελούμε την «Plot Node Search». Παρακολουθούμε την έρευνα με άλματα στις κινήσεις των ερευνητών. Έπειτα εκτελούμε την «Teleport clear» και το πρόγραμμα μας ενημερώνει για την αποπεράτωση της εντολής.

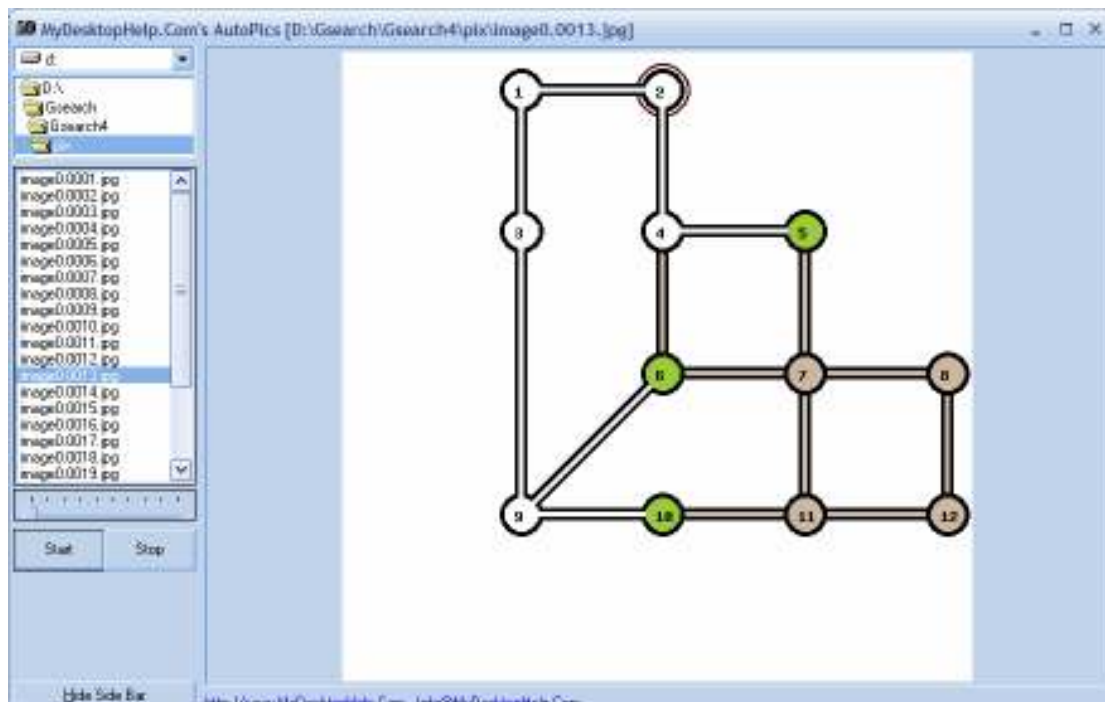


Σχήμα 5.18: Αναγγελία τέλους του καθαρισμού αλμάτων μέσω του Eclear.exe.

6. Επιλέγουμε το «OK» στο παραπάνω μήνυμα και εκτελούμε την «Plot Edge Search» για να παρακολουθήσουμε την αποτελεσματικότητα της στρατηγικής μας για το παιχνίδι ακμών.



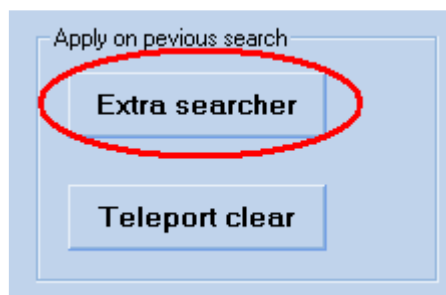
Σχήμα 5.19: Στιγμιότυπο από την παρακολούθηση της στρατηγικής έρευνας με το Autopics.exe. Οι εικόνες αναπαριστούν τις καταστάσεις των στοιχείων του γράφου για το παιχνίδι ακμών πλέον



Σχήμα 5.20: Στιγμιότυπο από την παρακολούθηση της στρατηγικής έρευνας με το Autopics.exe. Η n-καθαριστική στρατηγική έρευνας προκαλεί επαναμόλυνση του γράφου στο παιχνίδι ακμών.

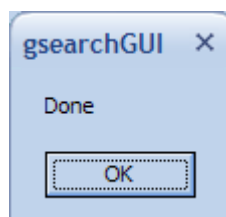
Βλέπουμε πως η στρατηγική μας είναι ανεπαρκής για το παιχνίδι ακμών καθώς στο 13^ο βήμα π.χ. έχουμε επαναμόλυνση του γράφου από την ακμή 4-6.

7. Εκτελούμε την εντολή «Extra Searcher» για να εφαρμόσουμε τον ομώνυμο αλγόριθμο στην ήδη υπάρχουσα στρατηγική έρευνας.



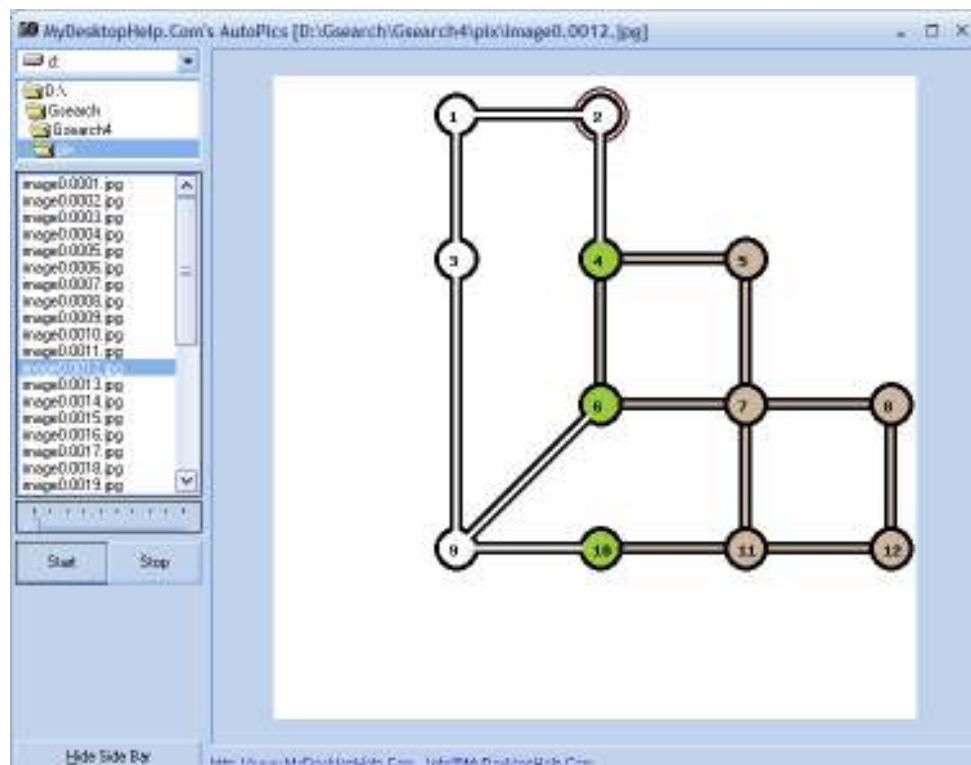
Σχήμα 5.21: Εφαρμογή του Extra Searcher πάνω στην ήδη αποθηκευμένη στρατηγική έρευνας.

Το πρόγραμμα μας ενημερώνει για την αποπεράτωση της εντολής.



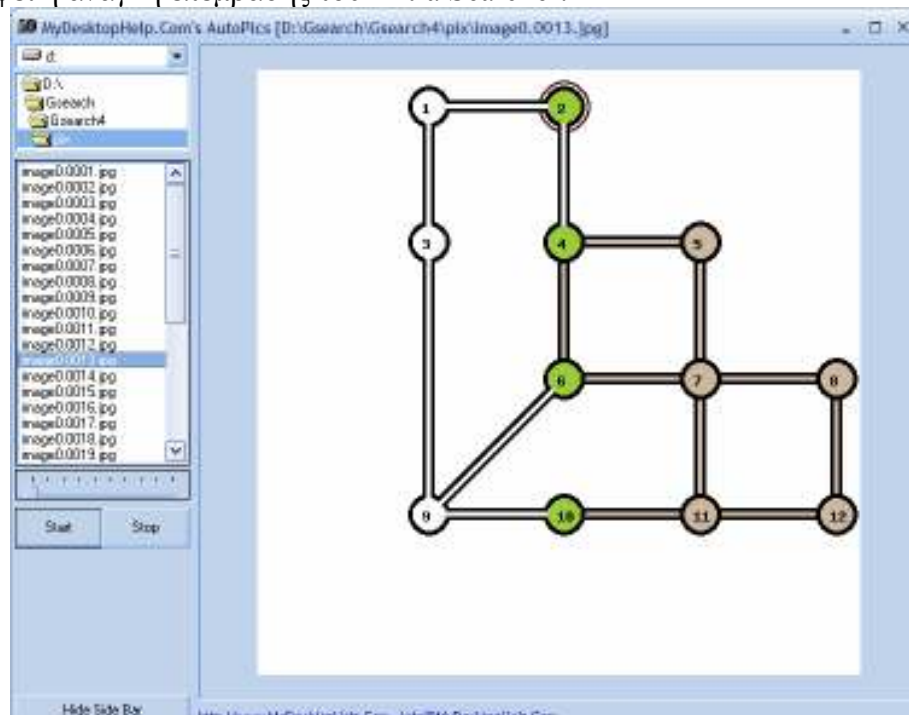
Σχήμα 5.22: Μήνυμα ολοκλήρωσης του αλγορίθμου Extra Searcher.

8. Επιλέγουμε το «OK» στο παραπάνω μήνυμα και εκτελούμε την «Plot Edge Search» για να παρακολουθήσουμε την νέα στρατηγική που προέκυψε.



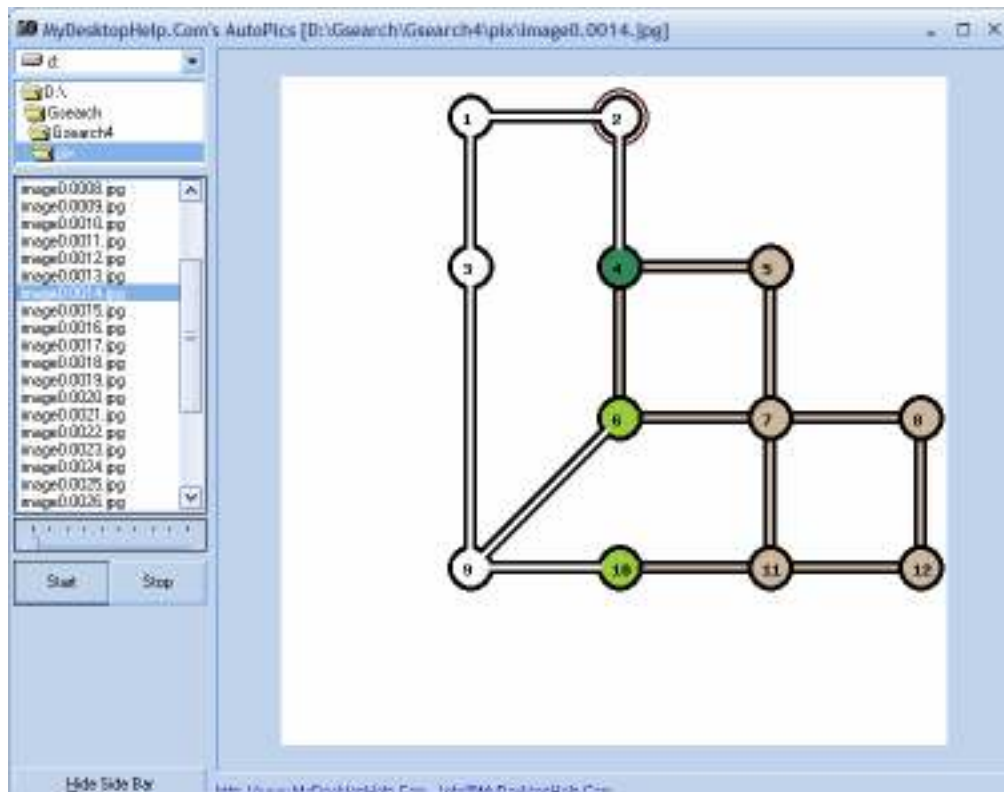
Σχήμα 5.23: Στιγμιότυπο από την παρακολούθηση της στρατηγικής έρευνας με το Autopics.exe.

Ως και το 12 βήμα, η έρευνα είναι πανομοιότυπη με πριν, καθώς δεν έχει ακόμα προκύψει η ανάγκη επέμβασης του Extra Searcher.

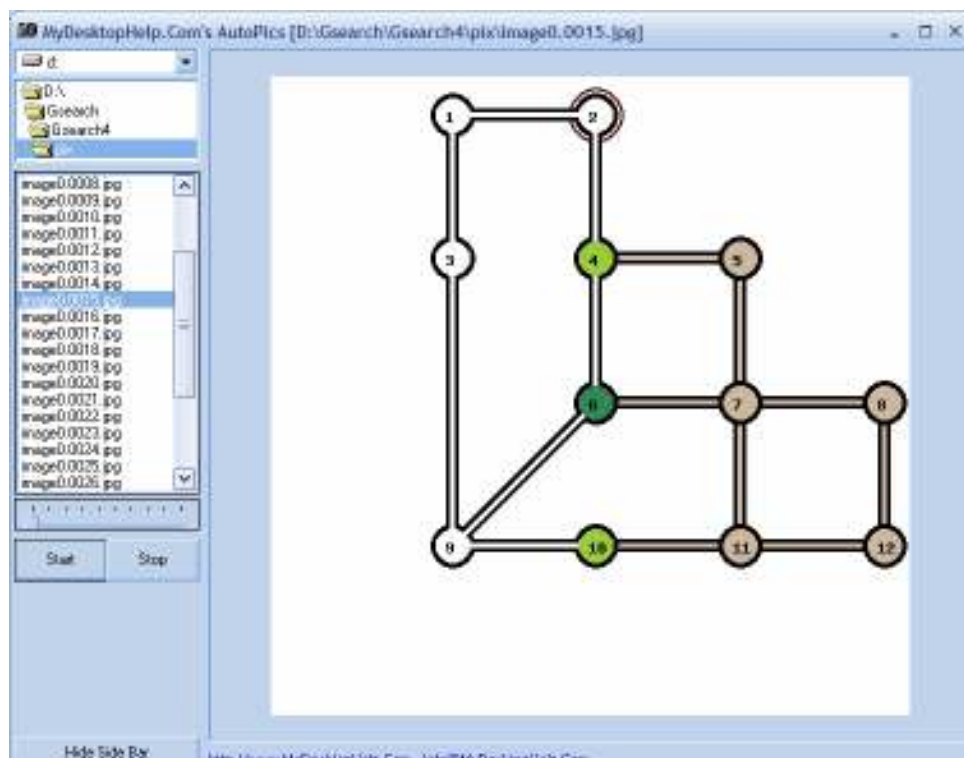


Σχήμα 5.24: Στιγμιότυπο από την παρακολούθηση της στρατηγικής έρευνας με το Autopics.exe. Ο ερευνητής στον κόμβο 2 είναι ο καθαριστής ακμών.

Όμως στο 13^ο βήμα, πριν την μετακίνηση του ερευνητή από τον κόμβο 4, εμφανίζεται ο Edge Cleaner.

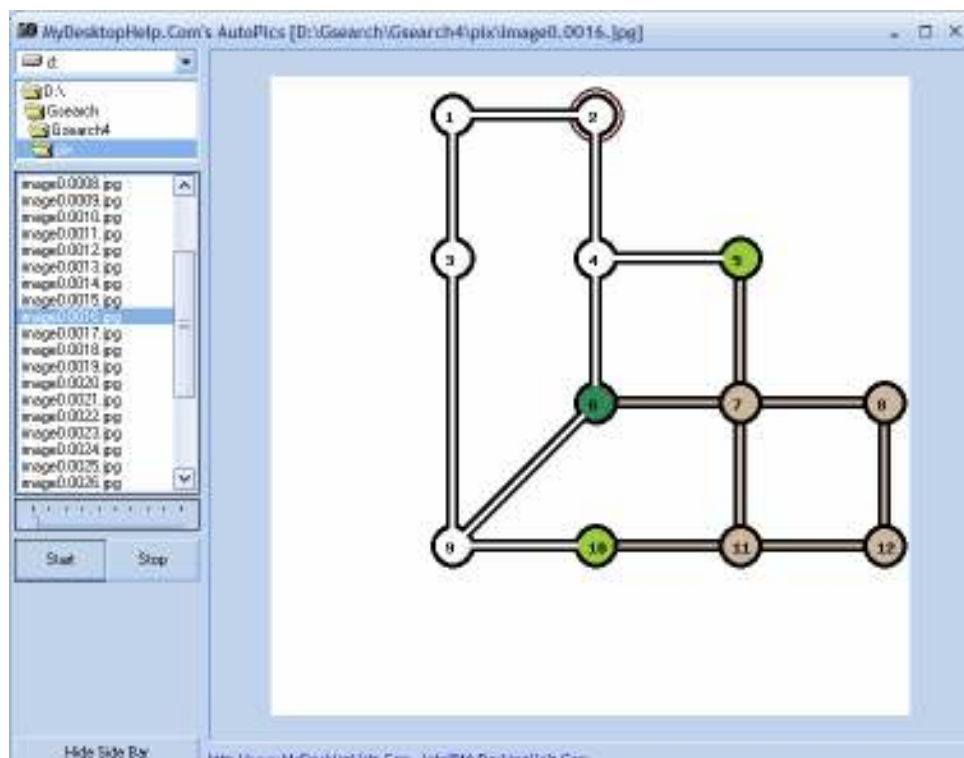


Σχήμα 5.25: Στιγμιότυπο από την παρακολούθηση της στρατηγικής έρευνας με το Autopics.exe. Ο Edge cleaner σε δράση



Σχήμα 5.26: Στιγμιότυπο από την παρακολούθηση της στρατηγικής έρευνας με το Autopics.exe. Ο Edge cleaner σε δράση

Ο Edge Cleaner καθαρίζει την ακμή 4-6 αποτρέποντας την επικείμενη επαναμόλυνση του γράφου.

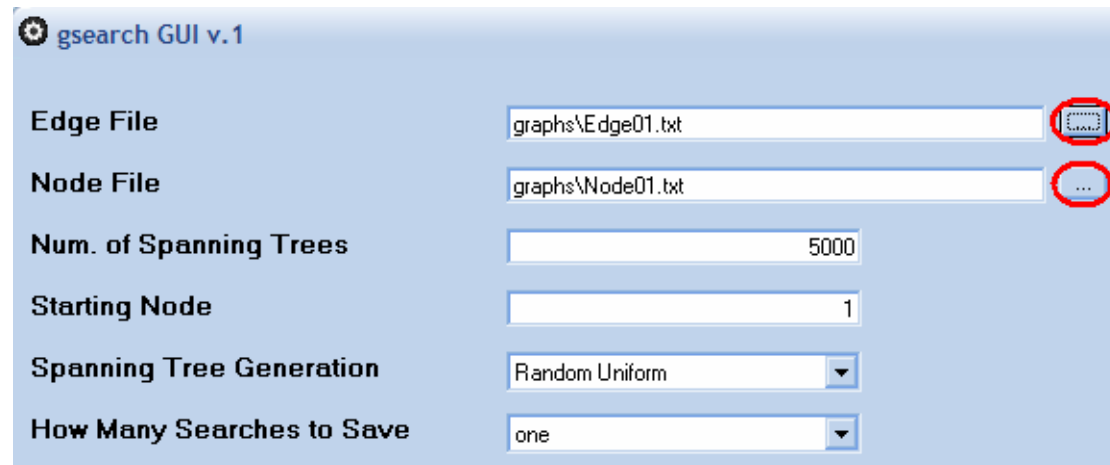


Σχήμα 5.27: Συνέχεια της έρευνας μετά την επέμβαση του Edge cleaner.

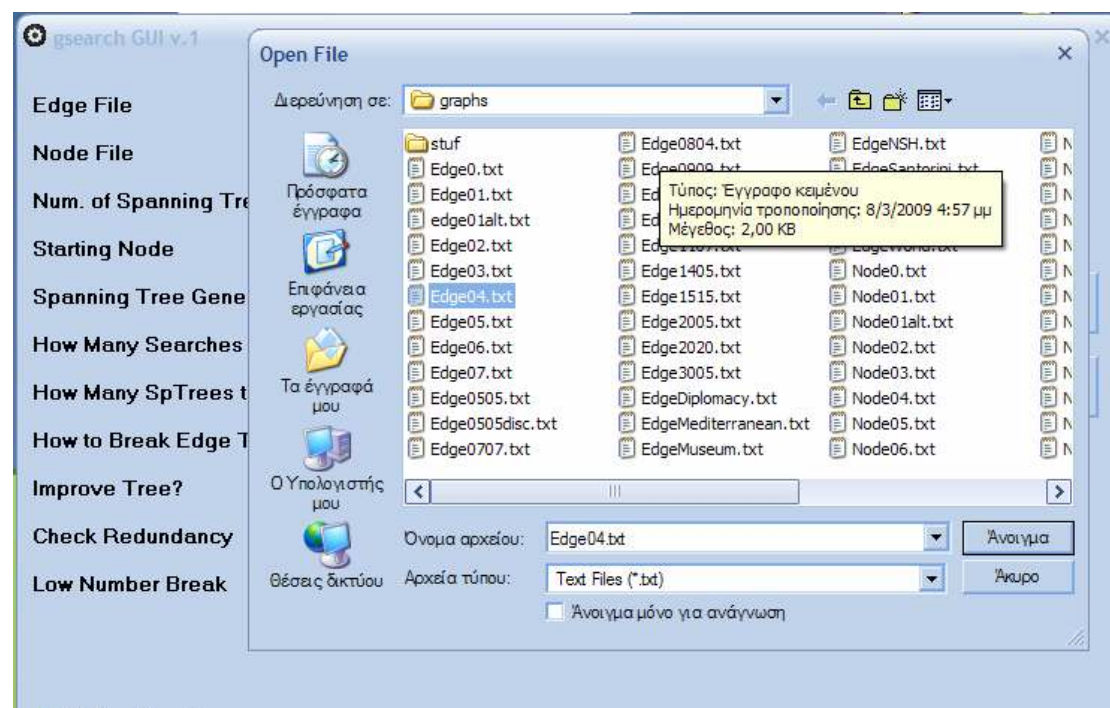
Η έρευνα συνεχίζεται κανονικά, χωρίς πλέον τον κίνδυνο επαναμόλυνσης.

Αν θέλουμε να εκτελέσουμε έρευνα στο παιχνίδι ακμών με τον αλγόριθμο Extra Nodes, μια πιθανή διαδοχή βημάτων είναι η εξής:

1.Επιλέγουμε τα αρχεία του γράφου.

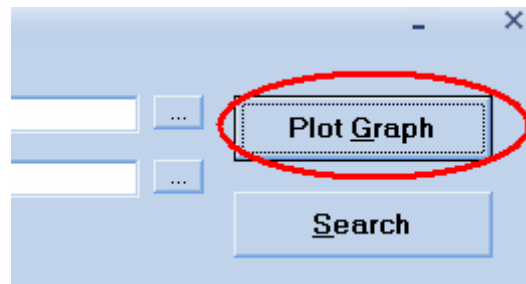


Σχήμα 5.28: Κουμπιά επιλογής αρχείων.



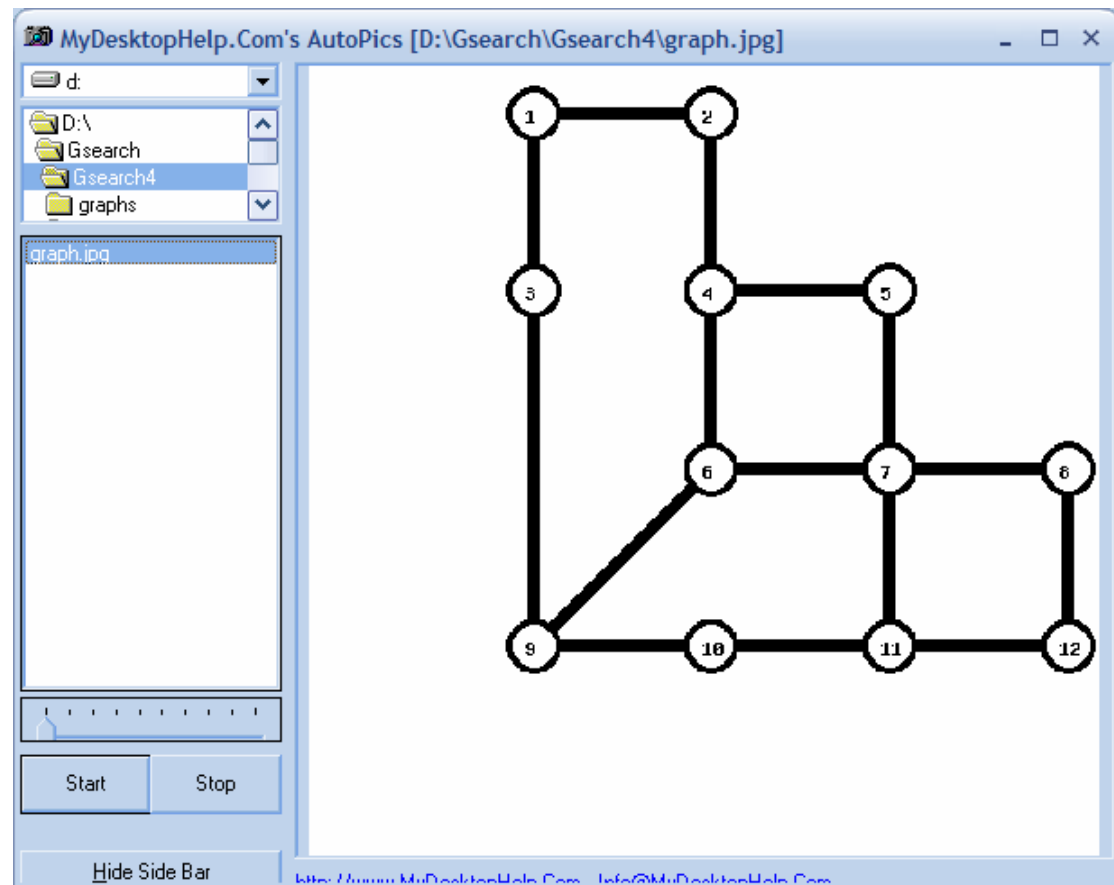
Σχήμα 5.29: Διάλογος περιήγησης στο σύστημα αρχείων.

2. Εκτελούμε την εντολή «Plot Graph» είτε από το κουμπί είτε με την συντόμευση πληκτρολογίου Alt+G.



Σχήμα 5.30: Εκτέλεση της εντολής vgraph.exe στο GsearchGUI.

Αφού δημιουργηθεί η εικόνα (graph.jpg) του γράφου, ανοίγεται από το πρόγραμμα προβολής Autopics.



Σχήμα 5.31: Το Autopics.exe παρουσιάζει την εικόνα του γράφου (graph.jpg).

3.Επιλέγουμε τα επιθυμητά ορίσματα του gsearch.

Num. of Spanning Trees	800
Starting Node	2
Spanning Tree Generation	Random Uniform
How Many Searches to Save	one
How Many SpTrees to Save	one
How to Break Edge Ties	Barriere Random
Improve Tree?	No
Check Redundancy	No
Low Number Break	0

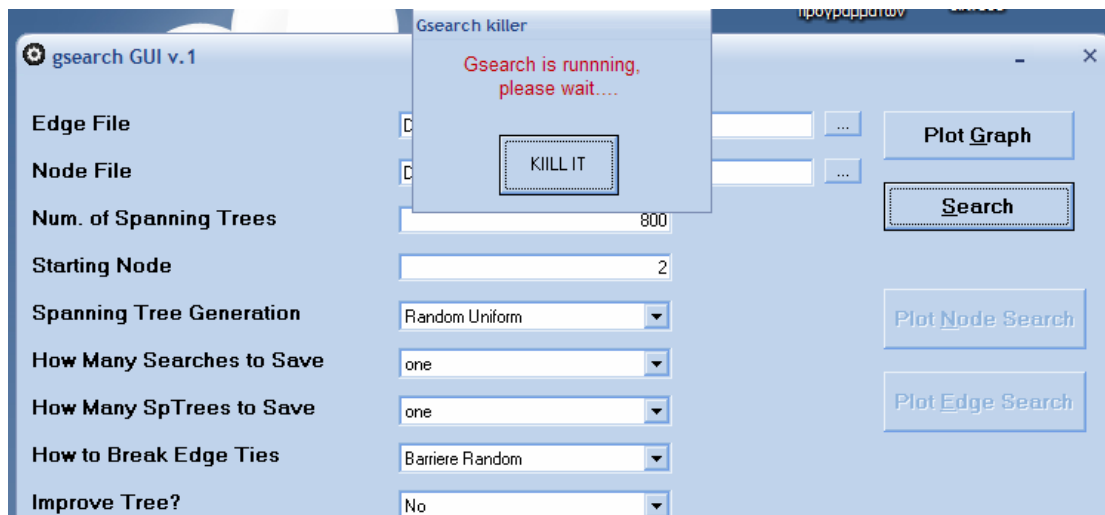
Σχήμα 5.32: Επιλογές των ορισμάτων του gsearch.exe.

4.Επιλέγουμε το παιχνίδι ακμών και τον επιθυμητό αλγόριθμο.

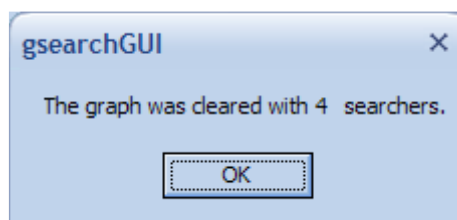
☒ Edge Clearing
Method
☐ Extra Searcher
☒ Extra Nodes

Σχήμα 5.33: Προεπιλογή της έρευνας ακμών και επιλογή εφαρμογής του Extra Nodes.

5.Εκτελούμε την εντολή «Search». Εμφανίζεται για λίγο το kill.exe και το πρόγραμμα μας ενημερώνει για τον ελάχιστο αριθμό ερευνητών που εντοπίστηκε, αφού τελειώσει η έρευνα.



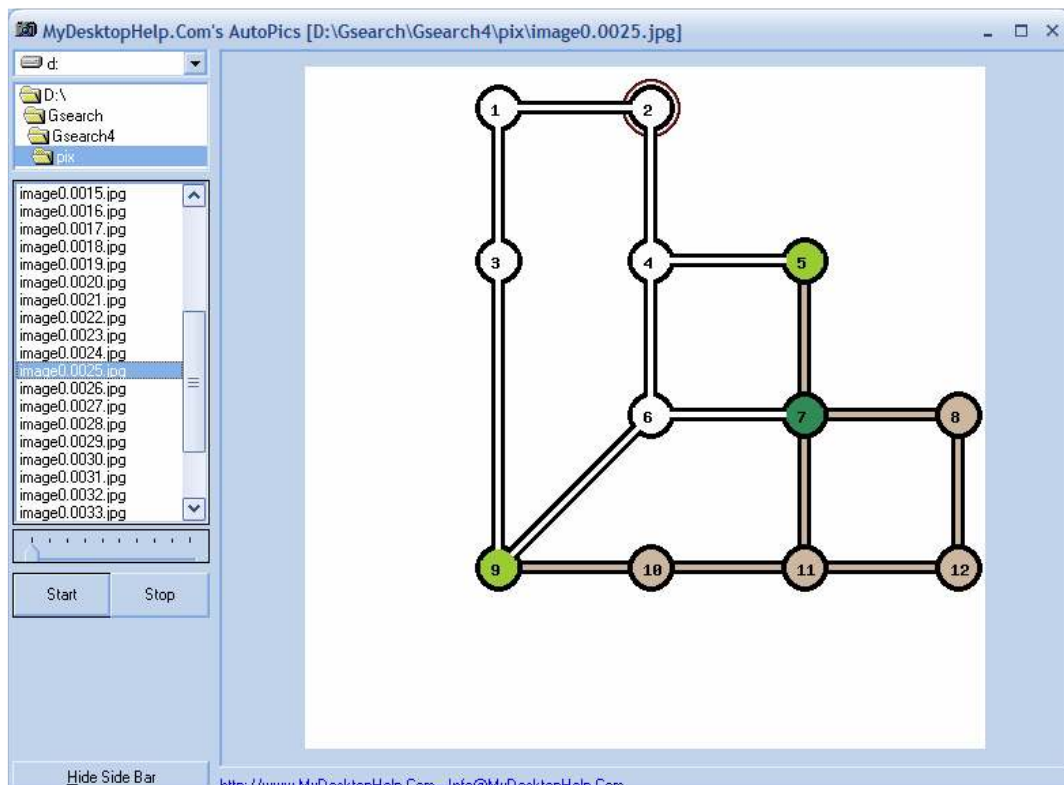
Σχήμα 5.34: Εκτέλεση του gsearch.exe, αυτή τη φορά πάνω στον ψευδογράφο με τους ενδιάμεσους κόμβους



Σχήμα 5.35: Μήνυμα τέλους της εκτέλεσης του gsearch.exe. Ενημέρωση για τον μικρότερο αριθμό ερευνητών που βρέθηκε.

Παρατηρούμε ότι, με τα ίδια δεδομένα με το παράδειγμα στο παιχνίδι κόμβων, τώρα απαιτείται ένας παραπάνω ερευνητής.

6.Επιλέγουμε το «OK» στο παραπάνω μήνυμα και εκτελούμε την «Plot Edge Search» που έχει πλέον ενεργοποιηθεί. Δημιουργούνται οι εικόνες και εκτελείται το Autopics στον αντίστοιχο φάκελο. Παρακολουθούμε τα βήματα της πρώτης αποθηκευμένης στρατηγικής έρευνας για το παιχνίδι ακμών. Η παρακολούθηση μπορεί να γίνει και αυτόματα με το κουμπί «Start» στο Autopics.



Σχήμα 5.36 Στιγμιότυπο από την παρακολούθηση της e-καθαριστικής στρατηγικής έρευνας με το Autopics.exe.

7. Μπορούμε αν θέλουμε να παρακολουθήσουμε την ίδια στρατηγική έρευνας για το παιχνίδι κόμβων εκτελώντας την «Plot Node Search». Όπως έχει αποδειχθεί και θεωρητικά, είναι φανερό ότι αυτή η στρατηγική έρευνας είναι καθαριστική και για το παιχνίδι κόμβων.

6. Πειράματα

6.1 Εισαγωγικά

Για τα πειράματα της εργασίας, αναπτύχθηκαν λογιστικά φύλλα Excel τα οποία, μέσω μακροεντολών VBA, δέχονται τις κατάλληλες εισόδους (ορίσματα gsearch), εκτελούν τις απαραίτητες εντολές (gsearch, Eclear) και τέλος αποθηκεύουν τα αποτελέσματα σε πίνακες και σχηματίζουν τα αντίστοιχα διαγράμματα στα φύλλα εργασίας.

Για τους διάφορους γράφους εκτελέστηκαν τρεις τύποι πειραμάτων για να υπάρξει μια σχετικά ολοκληρωμένη εικόνα της συμπεριφοράς των αλγορίθμων:

1. Στο πρώτο είδος πειραμάτων εκτελούνται όλες οι παραλλαγές του GSST για το παιχνίδι κόμβων, για το παιχνίδι ακμών με τον αλγόριθμο Extra Searcher και τέλος για το παιχνίδι ακμών με τον Extra Nodes. Στην η-καθαριστική στρατηγική έρευνας που προκύπτει στην πρώτη περίπτωση εφαρμόζουμε έπειτα τον καθαρισμό διακτινισμού, έτσι ώστε να μπορούμε να συγκρίνουμε πιο αντικειμενικά τον αριθμό των βημάτων της με αυτόν των άλλων περιπτώσεων. Οι εισοδοί του πειράματος είναι το αρχείο του γράφου και ο αριθμός των δένδρων που θα κατασκευαστούν, ενώ η ρίζα των ερευνών επιλέγεται κάθε φορά τυχαία. Αποθηκεύονται κάθε φορά όλες οι ελάχιστες στρατηγικές έρευνας έτσι ώστε να μπορούμε να μετρήσουμε όλα τα επιθυμητά μεγέθη. Τα μεγέθη που καταγράφονται στους πίνακες είναι τα εξής:
 - a. **ο ελάχιστος αριθμός ερευνητών** (Min Searchers) που χρειάστηκε για να καθαριστεί ο γράφος,
 - b. **ο ελάχιστος αριθμός βημάτων** (Min Steps) των ελάχιστων στρατηγικών που εντοπίστηκαν ,
 - c. **ο μέσος αριθμός βημάτων** (average steps) των ελάχιστων στρατηγικών,
 - d. **το ποσοστό των ελάχιστων ερευνών** (min Searchers %) σε σχέση με όλες τις έρευνες που δοκιμάστηκαν,
 - e. **το ποσοστό των ελάχιστων στρατηγικών με ελάχιστο αριθμό βημάτων** (min Steps %) σε σχέση με όλες τις έρευνες που δοκιμάστηκαν,
 - f. **ο χρόνος εκτέλεσης του gsearch** (Time (gsearch)),
 - g. **ο μέσος χρόνος εκτέλεσης του Eclear** (Av. time (Eclear)) για τις αποθηκευμένες ελάχιστες στρατηγικές έρευνας (στην περίπτωση του παιχνιδιού κόμβων είναι ο χρόνος του καθαρισμού από διακτινισμό). Οι χρόνοι μετρούνται σε sec.

Τέλος τα διαγράμματα είναι δύο ιστογράμματα που δείχνουν τη συμπεριφορά των διάφορων ερευνών σε σχέση με τον ελάχιστο αριθμό ερευνητών στο πρώτο και τον ελάχιστο αριθμό βημάτων στο δεύτερο.

2. Το δεύτερο είδος πειραμάτων είναι μια συγκριτική εξέταση των δύο αλγορίθμων για το παιχνίδι ακμών καθώς ο αριθμός των δένδρων αυξάνεται. Δέχεται ως εισόδους το αρχείο του γράφου, την μέθοδο κατασκευής των δέντρων, την μέθοδο επιλογής ακμής, μια αρχική τιμή για τον αριθμό των δένδρων, το βήμα αύξησης των δένδρων και τέλος τον αριθμό επαναλήψεων για κάθε βήμα αύξησης. Οι επαναλήψεις για συγκεκριμένο αριθμό δένδρων έχουν ως σκοπό την ελαχιστοποίηση του τυχαίου στοιχείου. Ο πίνακας των αποτελεσμάτων παραθέτει τους μέσους όρους
- a. του **ελάχιστου αριθμού ερευνητών** και
 - b. του **ελάχιστου αριθμού βημάτων της έρευνας** που πέτυχε ο κάθε αλγόριθμος σε κάθε βήμα αύξησης των δέντρων.

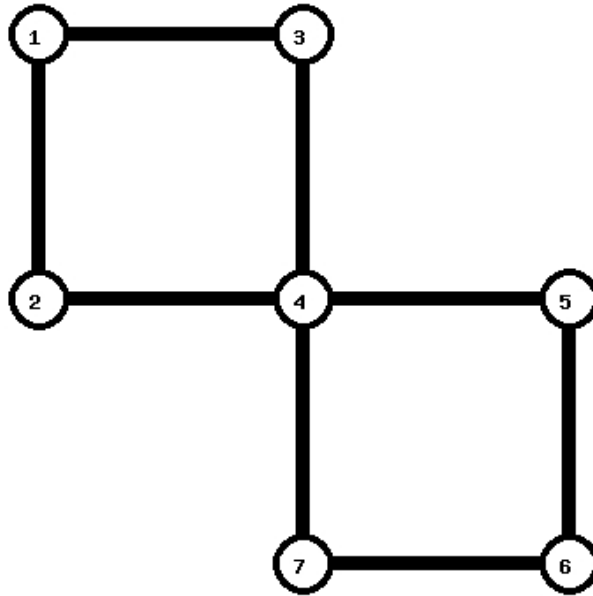
Τα διαγράμματα είναι οι αντίστοιχες γραφικές παραστάσεις των δύο μεγεθών καθώς αυξάνονται τα δένδρα. Τέλος σε κάθε διάγραμμα εμφανίζονται δύο καμπύλες διαφορετικού χρώματος. Η μία αναφέρεται στον Extra Searcher ενώ η άλλη στον Extra Nodes.

3. Το τρίτο πείραμα αποτελεί και αυτό μια συγκριτική εξέταση των δύο αλγορίθμων για το παιχνίδι ακμών καθώς ο αριθμός των δένδρων αυξάνεται. Η σύγκριση γίνεται μόνο ως προς τον **ελάχιστο αριθμό ερευνητών**. Η διαφορά με το παραπάνω πείραμα είναι ότι ο κάθε αλγόριθμος εκτελείται μία μόνο φορά. Εξετάζεται ακριβώς η ιδιότητα του GSST να δίνει λύση σε οποιονδήποτε χρόνο. Η είσοδοι είναι το αρχείο του γράφου, την μέθοδο κατασκευής των δέντρων, την μέθοδο επιλογής ακμής και ο αριθμός των δένδρων. Στον πίνακα και στο διάγραμμα αποτελεσμάτων παρακολουθούμε πως βελτιώνεται ο αριθμός των ερευνητών καθώς εξελίσσεται η έρευνα προς το 100% των δένδρων που πρόκειται να εξεταστούν. Καθώς η καμπύλη αυτή μπορεί να διαφέρει πολύ από έρευνα σε έρευνα λόγω του τυχαίου παράγοντα (μπορεί να βρεθεί πολύ γρήγορα κατά τύχη μια ελάχιστη έρευνα), το πείραμα επαναλήφθηκε πολλές φορές και παρουσιάζεται ένα ενδεικτικό αποτέλεσμα που εμφανίζεται συχνότερα από τα υπόλοιπα.

6.1 Γράφοι

6.1.1 Απλός γράφος 1

Η πρώτη σειρά πειραμάτων έγινε στο πολύ απλό γράφο του Σχήματος 6.1. Εύκολα διακρίνουμε ότι είτε για το παιχνίδι κόμβων, είτε για το παιχνίδι ακμών, ο γράφος έχει $s^{inc} = 2$.



Σχήμα 6.1: Απλός γράφος 1.

Εκτελούμε τις τρεις εκδοχές έρευνας του πρώτου πειράματος για τον παραπάνω γράφο, για όλες τις παραλλαγές του GSST με $M=20$ γεννητορικά δένδρα. Ακολουθούν τα αποτελέσματα του πειράματος.

Node Search	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	2	10	10,86	70%	20%	0,063	0,001
GSST-LR	2	10	10,78	45%	15%	0,047	0,003
GSST-R	2	9	9,75	40%	10%	0,047	0,000
GSST-LW	2	9	9,36	55%	35%	0,047	0,001
GSST-LD	2	9	9,00	45%	45%	0,047	0,002
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)
GSST-L	2	10	10,44	45%	25%	0,047	0,002
GSST-LR	2	10	10,43	35%	20%	0,031	0,002
GSST-R	2	9	9,67	60%	20%	0,047	0,003
GSST-LW	2	9	9,50	50%	25%	0,047	0,002
GSST-LD	2	9	9,00	45%	45%	0,031	0,002

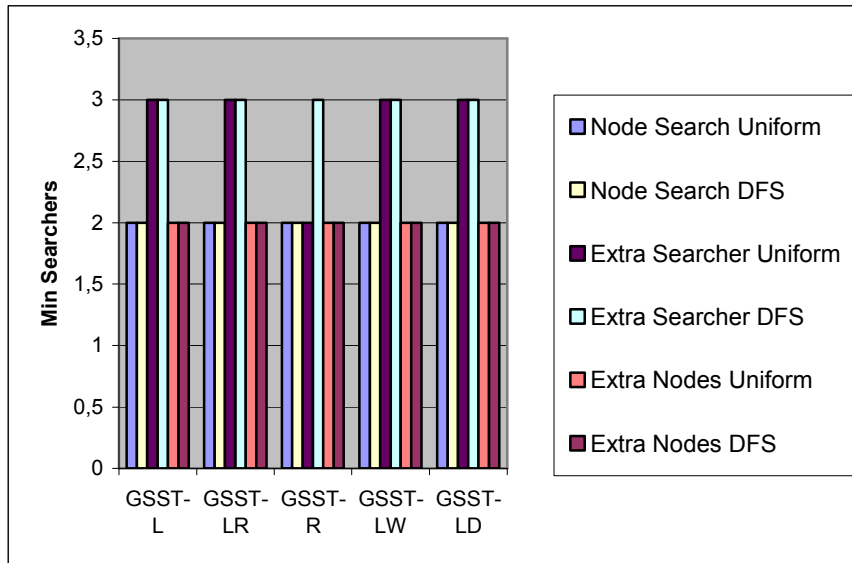
Πίνακας 6.2: Αποτελέσματα της έρευνας κόμβων στον γράφο του Σχήματος 6.1. Πείραμα 1^ο.

Edge Search(Xser.)	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	3	13	15,21	70%	20%	0,063	0,001
GSST-LR	3	13	14,67	45%	15%	0,047	0,003
GSST-R	2	10	10,50	40%	30%	0,047	0,004
GSST-LW	3	10	11,73	55%	20%	0,047	0,003
GSST-LD	3	13	13,44	45%	25%	0,047	0,003
	DFS ST generation						
Edge Traversal							
GSST-L	3	13	13,89	45%	25%	0,047	0,002
GSST-LR	3	13	13,86	35%	20%	0,031	0,000
GSST-R	3	10	10,33	60%	40%	0,047	0,003
GSST-LW	3	10	10,50	50%	25%	0,047	0,002
GSST-LD	3	11	11,44	45%	25%	0,031	0,003
Edge Search(Xnodes)	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	2	16	18,00	30%	5%	0,047	0,003
GSST-LR	2	14	14,83	30%	5%	0,047	0,003
GSST-R	2	11	11,50	20%	10%	0,031	0,004
GSST-LW	2	10	10,67	30%	10%	0,031	0,003
GSST-LD	2	10	10,50	30%	15%	0,047	0,000
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	2	12	13,20	25%	10%	0,031	0,003
GSST-LR	2	12	12,40	25%	20%	0,031	0,003
GSST-R	2	12	12,00	15%	15%	0,031	0,000
GSST-LW	2	10	11,00	30%	15%	0,031	0,003
GSST-LD	2	10	10,50	20%	10%	0,031	0,004

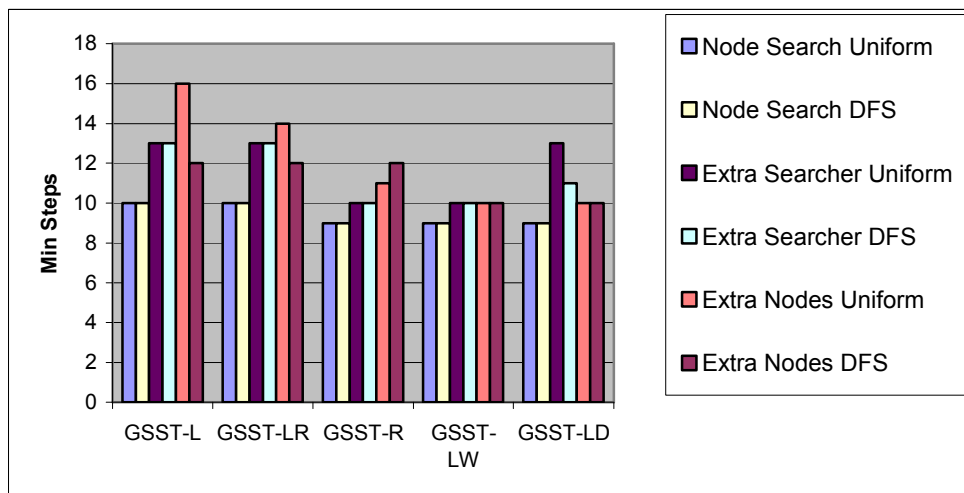
Πίνακας 6.3 Αποτελέσματα της έρευνας ακμών στον γράφο του Σχήματος 6.1. Πείραμα 1^ο.

Η πρώτη και πιο σημαντική παρατήρηση στους παραπάνω πίνακες είναι ότι ο Extra Searcher (εκτός από την περίπτωση του uniform GSST-R) τοποθετεί έναν παραπάνω ερευνητή στο γράφο. Αυτό το γεγονός έχει να κάνει με τη φύση του αλγορίθμου αλλά και του γράφου. Όπως αποδείχθηκε στο Θεώρημα 3.11, αν ένας γράφος μπορεί να καθαριστεί για το παιχνίδι κόμβων με K ερευνητές, τότε για το παιχνίδι ακμών χρειάζονται K ή $K+1$ ερευνητές¹. Η πλειοψηφία των γράφων μάλιστα ανήκει στη δεύτερη κατηγορία. Όμως η τοπολογία του συγκεκριμένου γράφου έχει ως αποτέλεσμα να αρκούν K . Έτσι ο καθαριστής ακμών που εισάγεται από τον Extra Searcher δεν είναι απαραίτητος. Βλέπουμε δηλαδή ότι ο γράφος αυτός αποτελεί μια εξαίρεση στην οποία η φύση του αλγορίθμου Extra Searcher τον καθιστά πιο αδύναμο από τον Extra Nodes. Με τον uniform GSST-R έχουμε όμως πάλι δυο ερευνητές. Λόγω ακριβώς της τυχαίας επιλογής ακμής προέκυψαν στην έρευνα κόμβων κάποιες στρατηγικές που καθαρίζουν τον γράφο και για το παιχνίδι ακμών. Στον αλγόριθμο Extra Searcher δεν απαιτήθηκε εισαγωγή ερευνητή και έτσι η στρατηγική παρέμεινε ως είχε.

¹ Θυμηθείτε ότι κάθε καθαριστική έρευνα ακμών είναι και καθαριστική έρευνα κόμβων, αλλά το αντίθετο δεν ισχύει. Άρα περιμένουμε για μια καθαριστική έρευνα ακμών να απαιτούνται τουλάχιστον τόσοι ερευνητές όσοι και για την αντίστοιχη έρευνα κόμβων.



Σχήμα 6.4: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Πείραμα 1^ο, στον απλό γράφο 1.



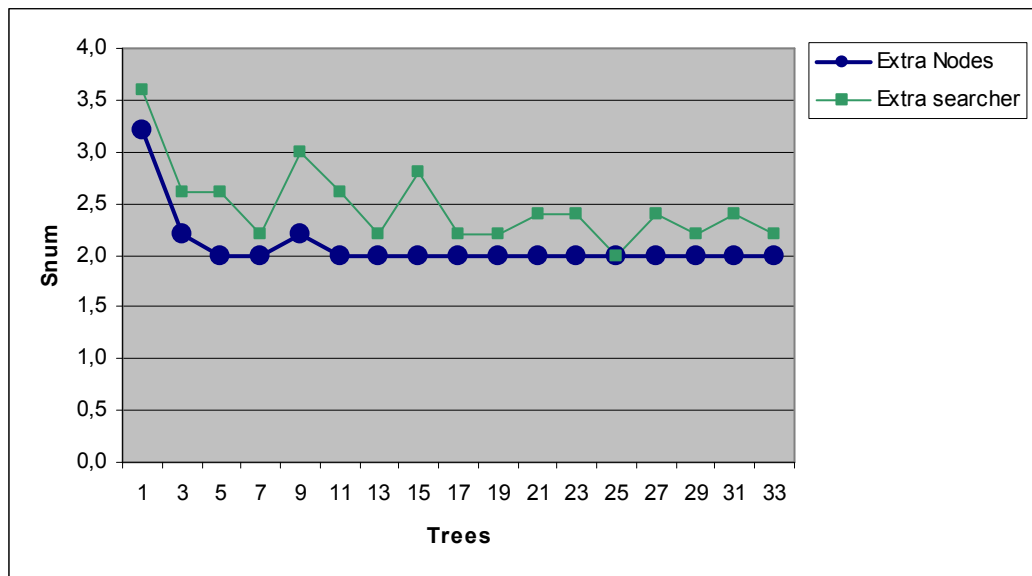
Σχήμα 6.5: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Πείραμα 1^ο, στον απλό γράφο 1.

Σχετικά με τον αριθμό βημάτων δεν μπορούμε να κρίνουμε ότι κάποιος αλγόριθμος είναι γενικά καλύτερος. Φαίνεται όμως καθαρά ότι, όπως και περιμένουμε άλλωστε, η έρευνα ακμών απαιτεί παραπάνω βήματα. Τα ποσοστά είναι λίγο καλύτερα στον Extra Searcher ενώ οι χρόνοι εκτέλεσης γενικά είναι γενικά αμελητέοι.

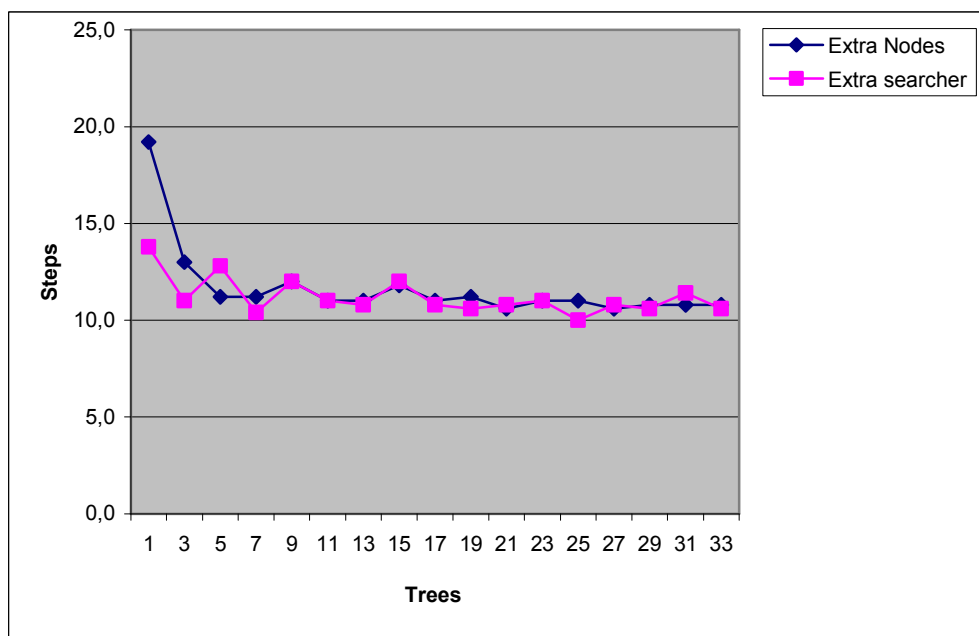
Ας δούμε τώρα την σύγκριση των δύο αλγορίθμων ως προς τον αριθμό βημάτων. Επιλέξαμε την παραλλαγή uniform GSST-R για τη σύγκριση με τις παρακάτω τιμές.

trees minimum:	1
step:	2
traversal	random
gen.tree	uniform
Repeats	5

Πίνακας 6.6: Πίνακας εισόδου δεδομένων για το 2^ο Πείραμα στον απλό γράφο 1.



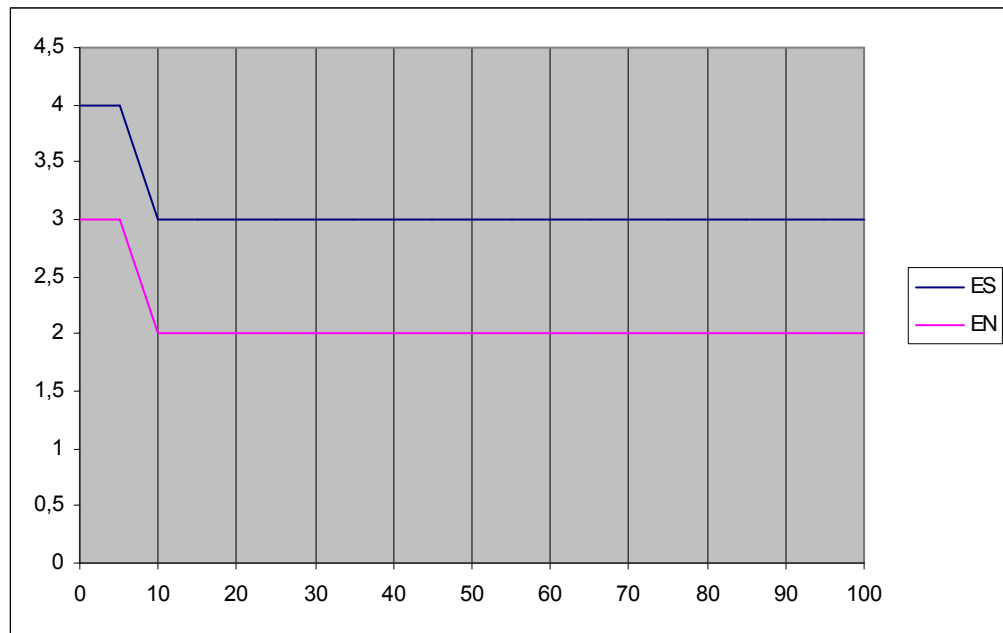
Σχήμα 6.7: Ελάχιστος αριθμός ερευνητών. Πείραμα 2^ο στον απλό γράφο 1.



Σχήμα 6.8: Ελάχιστος αριθμός βημάτων. Πείραμα 2^ο στον απλό γράφο 1.

Παρατηρούμε ότι ενώ ο ελάχιστος αριθμός των βημάτων δίνει την εντύπωση ότι είναι πάντα παρόμοιος, ο αριθμός των ερευνητών δείχνει ξεκάθαρα ότι εδώ ο Extra Nodes υπερτερεί.

Στο τρίτο πείραμα παρακολουθούμε την μείωση του αριθμού ερευνητών που απαιτεί ο κάθε αλγόριθμος καθώς δοκιμάζει περισσότερα δέντρα (στο συγκεκριμένο πείραμα χρησιμοποιείται ο uniform GSST-L και ο μέγιστος αριθμός δέντρων είναι 40).



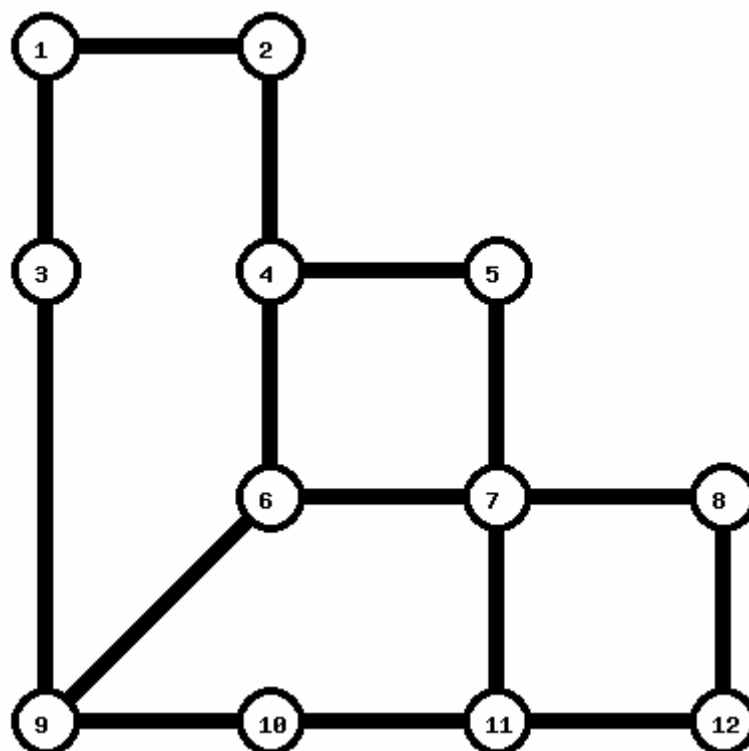
Σχήμα 6.9: Ελάχιστος αριθμός ερευνητών. Πείραμα 3^ο στον απλό γράφο 1.

Λόγω της χρήσης της συγκεκριμένης παραλλαγής του GGST, βλέπουμε ότι ο Extra Nodes δεν καταφέρνει να φτάσει το s_N^{imc} . Το σημείο της αλλαγής, συμπίπτει τυχαία με τη βελτίωση και του Extra Node.

Συμπερασματικά μπορούμε να πούμε ότι στον απλό αυτό γράφο, που απαιτεί ελάχιστο χρόνο εκτέλεσης, ο Extra Nodes είναι καλύτερος. Σημαντικό είναι το γεγονός ότι ο γράφος έχει ιδιαίτερη τοπολογία.

6.1.2 Απλός γράφος 2

Ο επόμενος γράφος που εξετάζεται εξακολουθεί να ανήκει στην κατηγορία των σχετικά απλών γράφων, αλλά αυτή την φορά δεν έχει κάποια ιδιαίτερη τοπολογία. Επίσης έχει παραπάνω στοιχεία (κόμβους και ακμές). Οι δύο παραπάνω ιδιότητες τον καθιστούν πιο γενικό χαρακτηριστικό παράδειγμα των απλών γράφων.



Σχήμα 6.10: Απλός γράφος 2

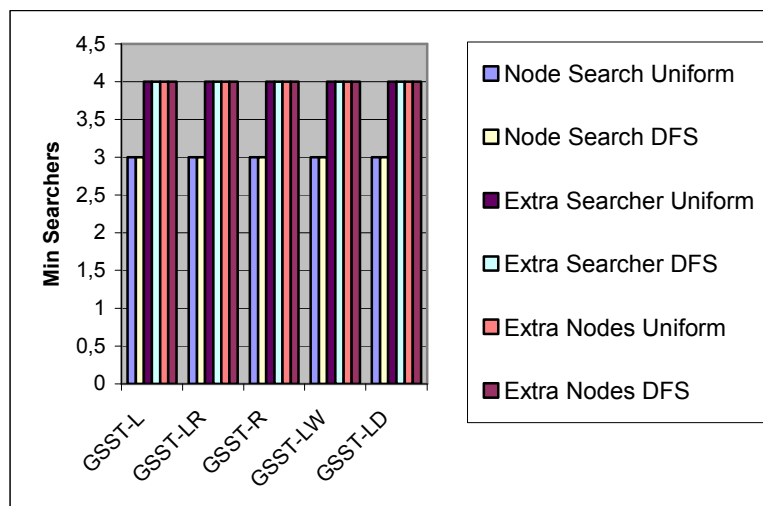
Ακολουθούν τα αποτελέσματα του πρώτου φύλλου πειραμάτων, με $M=300$.

Node Search	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	3	17	21,62	4,33%	0,33%	0,078	0,005
GSST-LR	3	17	24,33	4,00%	0,33%	0,094	0,004
GSST-R	3	15	19,41	13,00%	0,33%	0,125	0,003
GSST-LW	3	16	19,97	12,33%	0,67%	0,172	0,004
GSST-LD	3	14	18,09	7,33%	0,33%	0,094	0,005
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)
GSST-L	3	16	16,33	3,00%	2,00%	0,063	0,002
GSST-LR	3	16	26,08	24,67%	1,33%	0,203	0,006
GSST-R	3	17	21,80	5,00%	0,33%	0,094	0,003
GSST-LW	3	18	23,60	5,00%	1,00%	0,094	0,003
GSST-LD	3	17	19,54	4,33%	2,00%	0,078	0,002

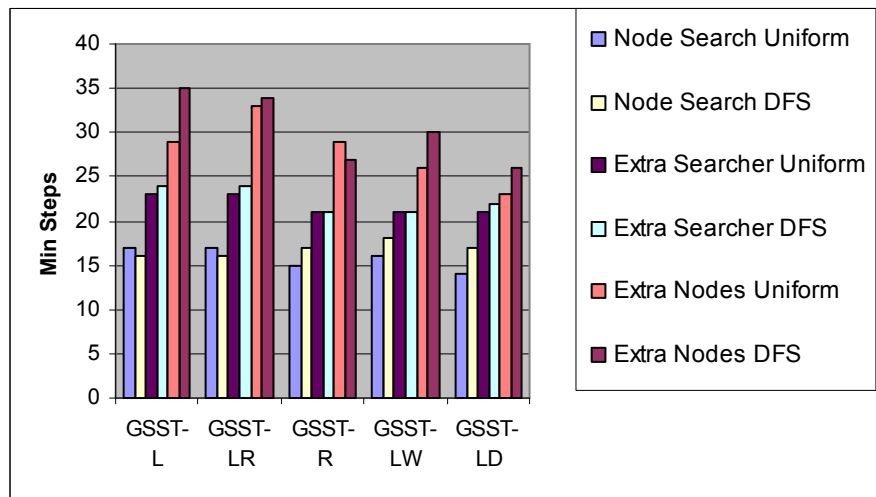
Πίνακας 6.11: Αποτελέσματα της έρευνας κόμβων στον απλό γράφο 4. Πείραμα 1^ο.

Edge Search(Xser.)	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	4	23	31,08	4,33%	0,33%	0,078	0,008
GSST-LR	4	23	35,08	4,00%	0,33%	0,094	0,010
GSST-R	4	21	25,13	13,00%	0,33%	0,125	0,005
GSST-LW	4	21	25,11	12,33%	0,33%	0,172	0,006
GSST-LD	4	21	26,27	7,33%	0,33%	0,094	0,005
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	4	24	24,56	3,00%	1,33%	0,063	0,005
GSST-LR	4	24	35,81	24,67%	1,33%	0,203	0,011
GSST-R	4	21	27,20	5,00%	0,67%	0,094	0,008
GSST-LW	4	21	28,53	5,00%	0,33%	0,094	0,007
GSST-LD	4	22	26,62	4,33%	0,33%	0,078	0,003
Edge Search(Xnodes)	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	4	29	40,64	17,67%	0,33%	0,234	0,012
GSST-LR	4	33	40,59	9,00%	0,33%	0,219	0,013
GSST-R	4	29	36,65	17,33%	1,00%	0,188	0,017
GSST-LW	4	26	36,32	21,00%	0,67%	0,234	0,016
GSST-LD	4	23	32,98	21,00%	0,67%	0,203	0,013
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	4	35	38,09	18,33%	2,33%	0,219	0,011
GSST-LR	4	34	38,25	18,67%	0,33%	0,250	0,011
GSST-R	4	27	38,85	11,33%	0,33%	0,141	0,016
GSST-LW	4	30	38,19	12,00%	0,33%	0,172	0,015
GSST-LD	4	26	33,64	14,67%	0,33%	0,188	0,013

Πίνακας 6.12: Αποτελέσματα της έρευνας ακμών στον απλό γράφο 2. Πείραμα 1^ο.



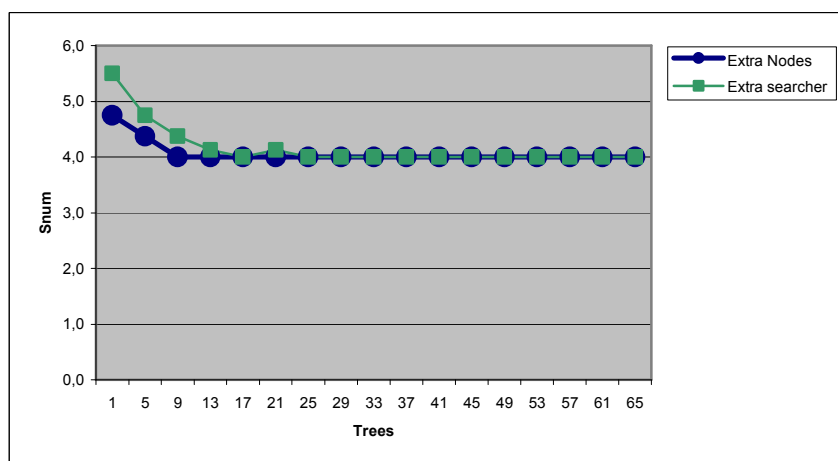
Σχήμα 6.13: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Πείραμα 1^ο, στον απλό γράφο 2.



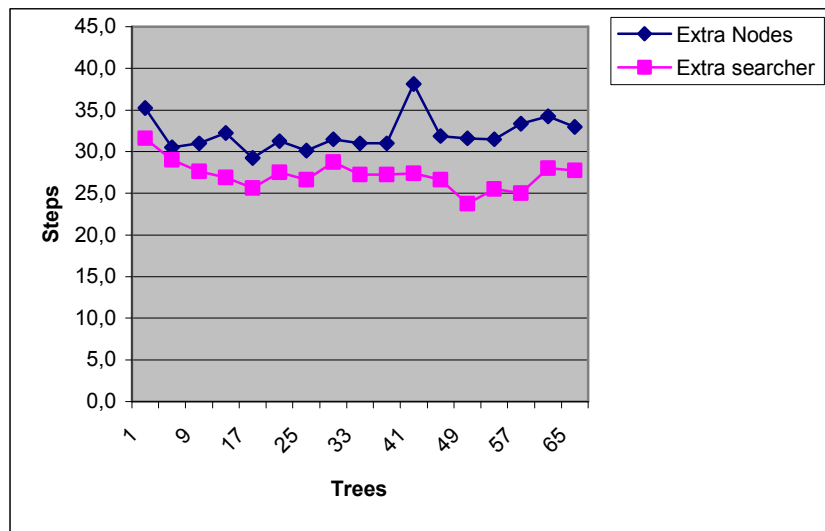
Σχήμα 6.14: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Πείραμα 1^ο, στον απλό γράφο 2.

Βλέπουμε πως, σε αυτήν την περίπτωση, ενώ απαιτούνται 3 ερευνητές για το παιχνίδι κόμβων, και οι δύο αλγόριθμοι για το παιχνίδι ακμών χρειάζονται 4 ερευνητές. Ο ελάχιστος αριθμός απαιτούμενων βημάτων και ο μέσος αριθμός βημάτων στις ελάχιστες στρατηγικές αυξάνονται αισθητά στον Extra Nodes. Όπως είναι αναμενόμενο η έρευνα κόμβων απαιτεί επίσης και λιγότερο αριθμό βημάτων. Είναι επίσης φανερό πως ο Extra Nodes βρίσκει μεγαλύτερο ποσοστό ελάχιστων ερευνών. Τέλος παρατηρούμε ότι ο χρόνος εκτέλεσης του gsearch είναι γενικά μεγαλύτερος στον Extra Nodes. Το γεγονός αυτό μπορεί να εξηγηθεί από τη φύση του αλγορίθμου, καθώς ο ψευδογράφος που δίνεται στον GSST είναι πολυπλοκότερος του αρχικού (με $n+e$ κόμβους και $2 \times e$ ακμές -κεφάλαιο 4.1).

Τα αποτελέσματα του δεύτερου φύλλου πειραμάτων συνοψίζονται στο παρακάτω διάγραμμα. Οι δοκιμές έγιναν με 8 επαναλήψεις σε κάθε βήμα και με τον uniform GSST-LD καθώς είναι οι πιο ευνοϊκές για τον Extra Nodes που είναι γενικά χειρότερος σε αυτό το παράδειγμα.



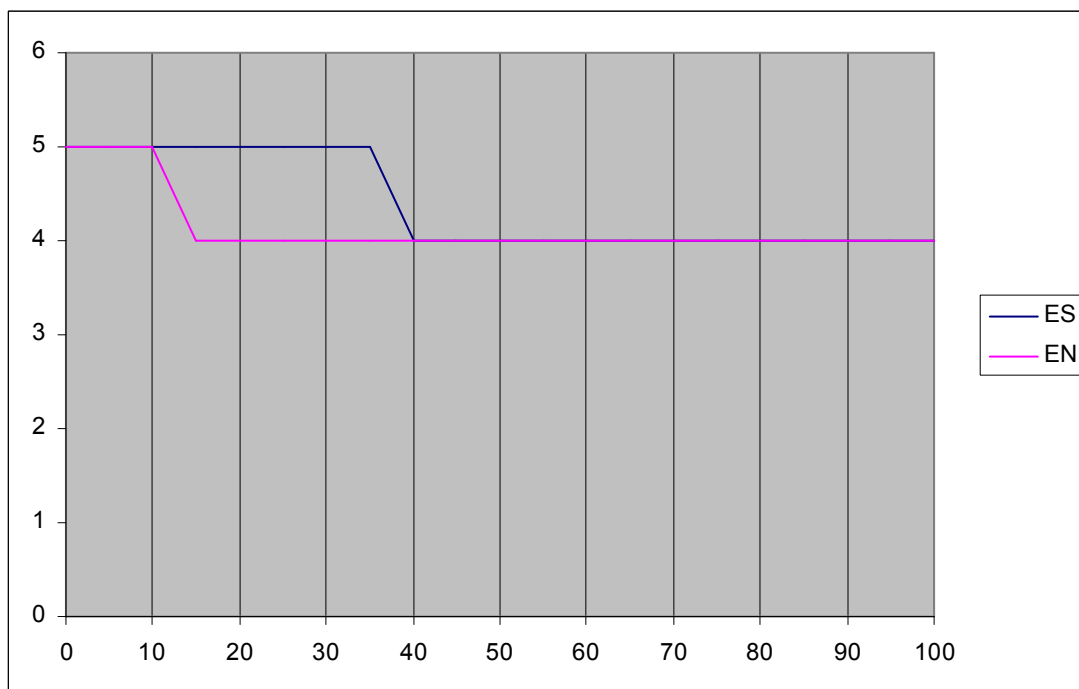
Σχήμα 6.15: Ελάχιστος αριθμός ερευνητών. Πείραμα 2^ο στον απλό γράφο 2



Σχήμα 6.16: Ελάχιστος αριθμός βημάτων. Πείραμα 2^ο στον απλό γράφο 2.

Παρατηρούμε πως ο Extra Nodes βρίσκει πιο γρήγορα τον ελάχιστο αριθμό ερευνητών, όμως πολύ γρήγορα ο Extra Searcher φτάνει στο ίδιο αποτέλεσμα. Σχετικά με τον αριθμό βημάτων, βλέπουμε ότι ο Extra Searcher είναι σταθερά καλύτερος. Αυτό οφείλεται στην μεγαλύτερη πολυπλοκότητα του ψευδογράφου.

Στο τρίτο πείραμα, με τον DFS GSST-LD αυτή τη φορά, παρακολουθούμε πως οι δύο αλγόριθμοι βελτιώνουν τη λύση τους για $M=40$ γεννητορικά δένδρα.

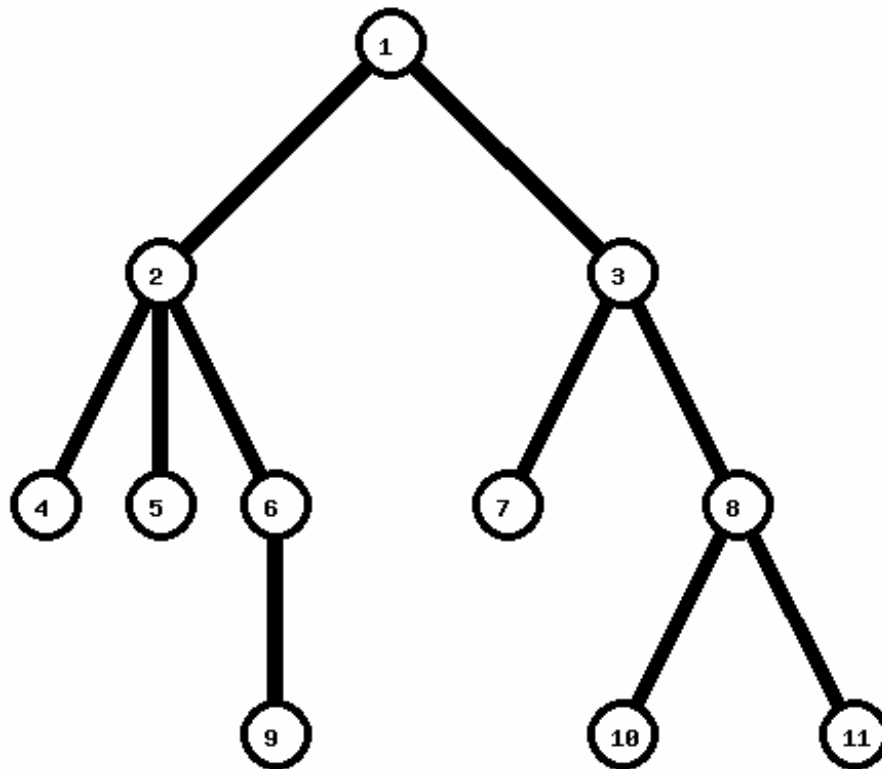


Σχήμα 6.17: Ελάχιστος αριθμός ερευνητών. Πείραμα 3^ο στον απλό γράφο 2.

Σε αυτό τον γράφο, παρατηρούμε ότι καθώς η πολυπλοκότητα αυξάνεται οι χρόνοι των δύο αλγορίθμων αρχίζουν να διαφέρουν. Επίσης φαίνεται μια αδυναμία του Extra Nodes τον αριθμό βημάτων των στρατηγικών που παράγει.

6.1.3 Δένδρο

Θα πραγματοποιήσουμε τα πειράματα ενδεικτικά για το δένδρο του σχήματος 6.15. Όπως έχει αποδειχθεί θεωρητικά τα παιχνίδια κόμβων και ακμών είναι ισοδύναμα στα δένδρα. Μια στρατηγική έρευνας που καθαρίζει ένα δένδρο για το παιχνίδι κόμβων, το καθαρίζει επίσης και για το παιχνίδι ακμών. Επομένως περιμένουμε να δούμε γενικά παρόμοια αποτελέσματα και για τους τρεις τύπους ερευνών.



Σχήμα 6.18: Γράφος - δένδρο

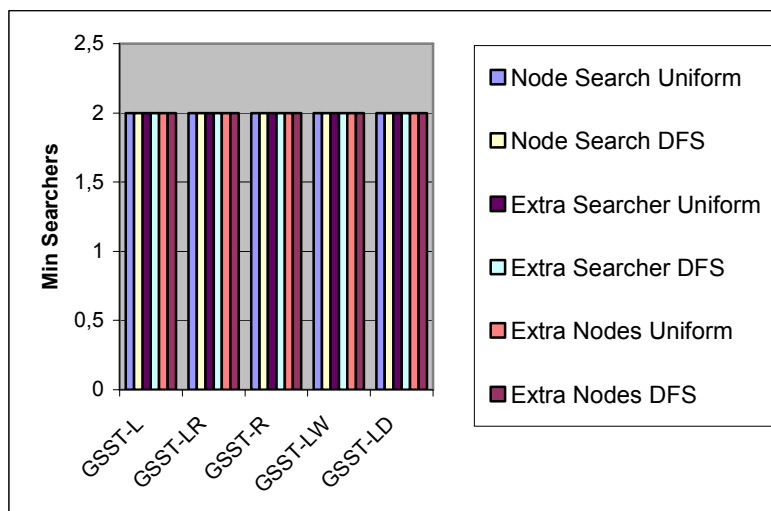
Οι δοκιμές του πρώτου φύλλου πειραμάτων έγιναν για $M=20$.

Node Search	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	2	21	21	5%	5%	0,031	0,015
GSST-LR	2	21	21	5%	5%	0,031	0,000
GSST-R	2	22	22	5%	5%	0,047	0,000
GSST-LW	2	22	22	5%	5%	0,016	0,000
GSST-LD	2	22	22	5%	5%	0,031	0,000
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)
GSST-L	2	21	21	5%	5%	0,031	0,000
GSST-LR	2	21	21	5%	5%	0,031	0,000
GSST-R	2	20	20	5%	5%	0,031	0,000
GSST-LW	2	20	20	5%	5%	0,031	0,000
GSST-LD	2	22	22	5%	5%	0,031	0,000

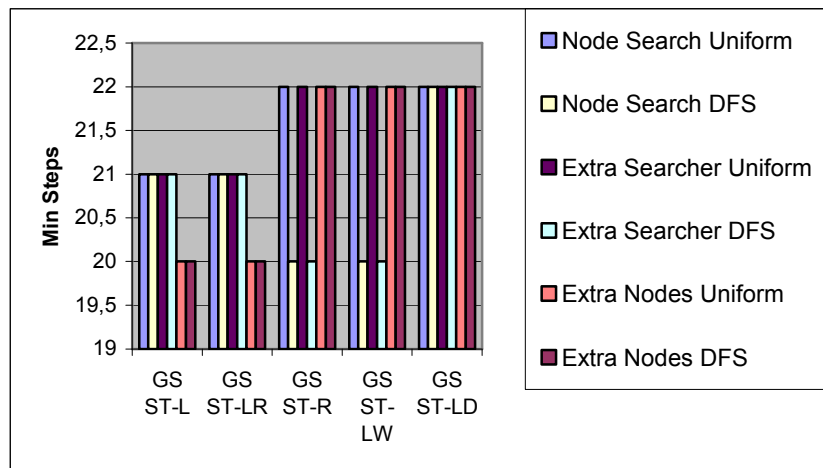
Πίνακας 6.19: Αποτελέσματα της έρευνας κόμβων στο δένδρο. Πείραμα 1^ο.

Edge Search(Xser.)	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	2	21	21	5%	5%	0,031	0,000
GSST-LR	2	21	21	5%	5%	0,031	0,000
GSST-R	2	22	22	5%	5%	0,047	0,015
GSST-LW	2	22	22	5%	5%	0,016	0,000
GSST-LD	2	22	22	5%	5%	0,031	0,015
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	2	21	21	5%	5%	0,031	0,015
GSST-LR	2	21	21	5%	5%	0,031	0,015
GSST-R	2	20	20	5%	5%	0,031	0,015
GSST-LW	2	20	20	5%	5%	0,031	0,000
GSST-LD	2	22	22	5%	5%	0,031	0,000
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	2	20	20	5%	5%	0,031	0,000
GSST-LR	2	20	20	5%	5%	0,031	0,000
GSST-R	2	22	22	5%	5%	0,016	0,015
GSST-LW	2	22	22	5%	5%	0,031	0,000
GSST-LD	2	22	22	5%	5%	0,031	0,015

Πίνακας 6.20: Αποτελέσματα της έρευνας ακμών στο δένδρο. Πείραμα 1^ο.



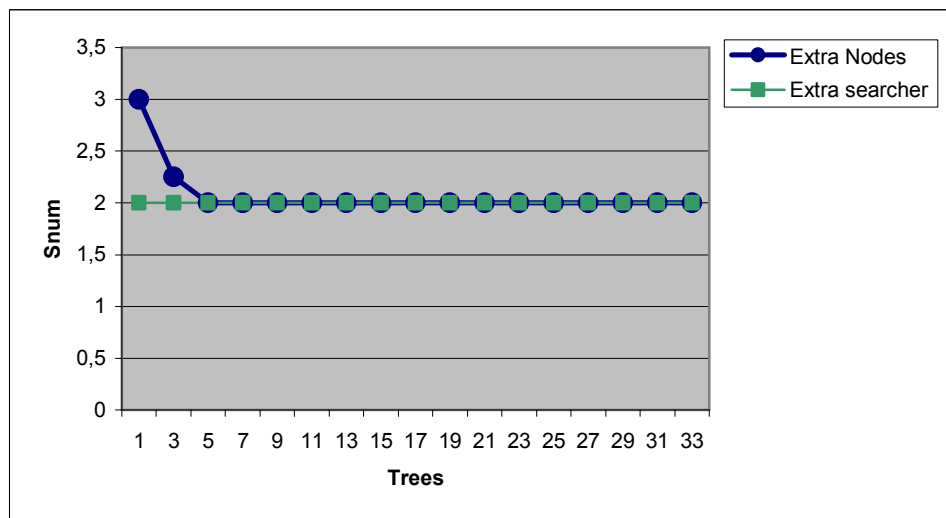
Σχήμα 6.21: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Πείραμα 1^ο, στο δένδρο.



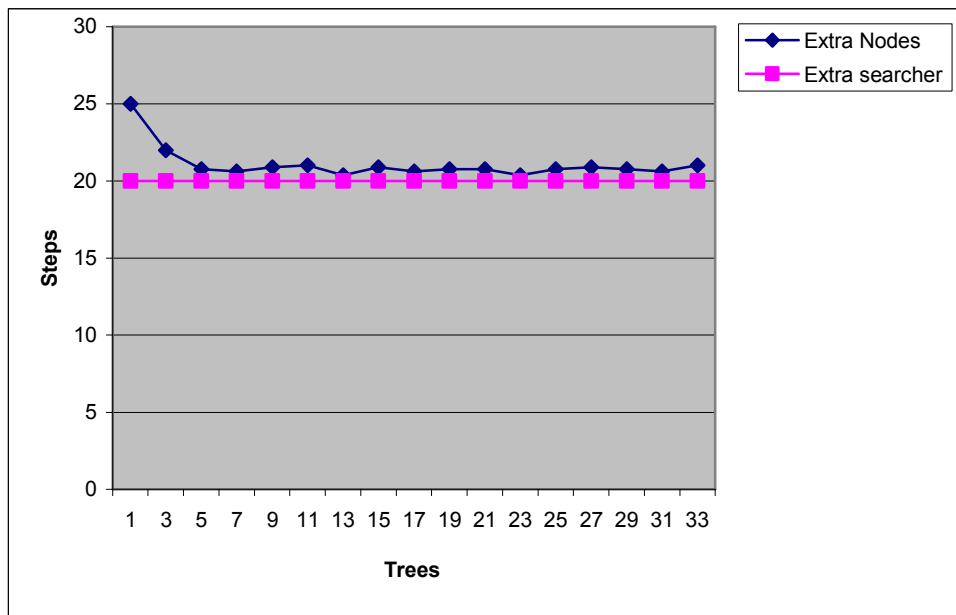
Σχήμα 6.22: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Πείραμα 1^ο, στο δένδρο.

Βλέπουμε, στον πίνακα αποτελεσμάτων, ότι επιβεβαιώνεται η θεωρία. Τα αποτελέσματα είναι ίδια για όλες τις έρευνες. Το μόνο μέγεθος που διαφέρει είναι ο ελάχιστος αριθμός βημάτων, ο οποίος όμως κάθε φορά συμπίπτει με τον μέσο όρο τους. Η διακύμανση αυτή δεν έχει να κάνει με την αποτελεσματικότητα της κάθε έρευνας αλλά με την ρίζα που επιλέχθηκε τυχαία κάθε φορά. Αν για παράδειγμα επιλεγθεί ο κόμβος 11 θα απαιτούνται λιγότερα βήματα απ' ό,τι με τον 8, δεν θα είναι απαραίτητη η κίνηση επιστροφής $11 \rightarrow 8$ μετά τον καθαρισμό του κόμβου 11 (με την $8 \rightarrow 11$).

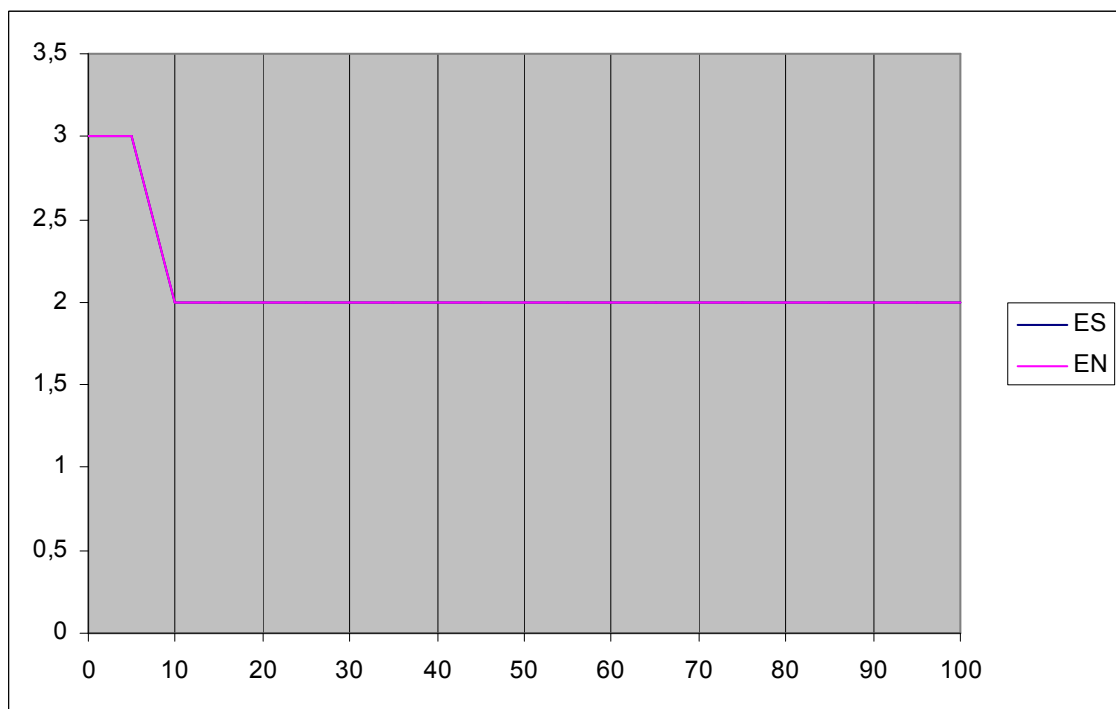
Ας δούμε τα αποτελέσματα από τα άλλα δυο φύλλα πειραμάτων με uniform GSST-L.



Σχήμα 6.23: Ελάχιστος αριθμός ερευνητών. Πείραμα 2^ο στο δένδρο.



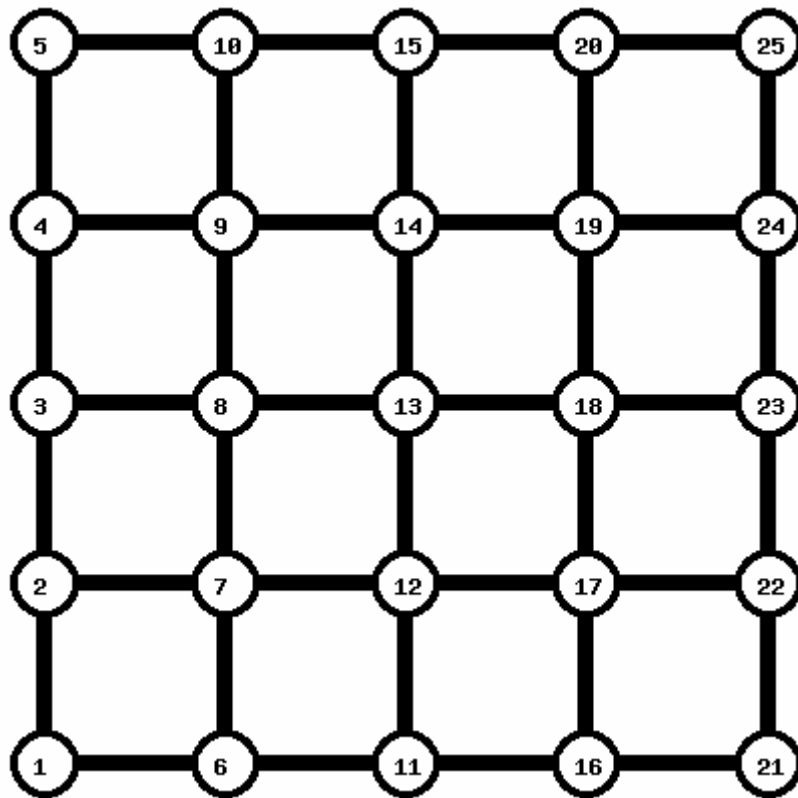
Σχήμα 6.24: Ελάχιστος αριθμός βημάτων. Πείραμα 2^ο στο δένδρο.



Σχήμα 6.25: Ελάχιστος αριθμός ερευνητών. Πείραμα 3^ο στο δένδρο.

6.1.4. Πλέγμα 5X5

Το πλήρες πλέγμα 5X5 έχει $s_N^{inc} = 5$ επομένως σίγουρα μπορεί να καθαριστεί με 6 ερευνητές στο παιχνίδι ακμών. Σε τέτοιας πολυπλοκότητας γράφους περιμένουμε ότι η πολυπλοκότητα του ψευδογράφου θα αυξηθεί έντονα.



Σχήμα 6.26: Γράφος πλέγματος 5x5

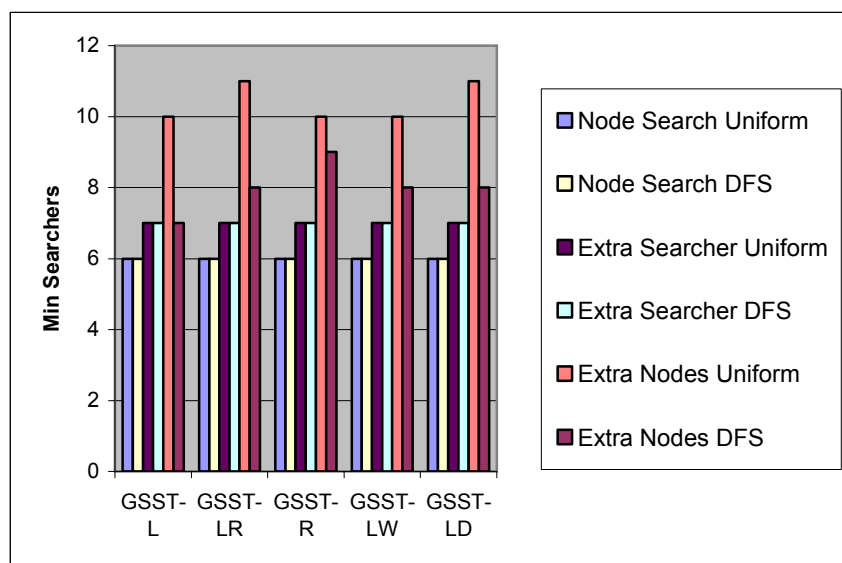
Ακολουθούν τα αποτελέσματα του πρώτου φύλλου πειραμάτων για $M=5000$.

Node Search	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	6	62	76,0	1,10%	0,02%	3,234	0,042
GSST-LR	6	56	75,7	1,20%	0,02%	4,344	0,046
GSST-R	5	47	65,4	2,94%	0,02%	2,547	0,238
GSST-LW	5	43	66,0	3,72%	0,02%	12,609	0,689
GSST-LD	6	40	55,7	4,08%	0,02%	3,891	0,111
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)
GSST-L	6	52	74,3	0,68%	0,06%	2,469	0,042
GSST-LR	6	50	71,4	0,34%	0,02%	3,672	0,036
GSST-R	6	61	77,8	1,14%	0,02%	1,969	0,060
GSST-LW	7	61	78,9	3,80%	0,06%	3,625	0,135
GSST-LD	6	46	55,8	0,24%	0,02%	2,422	0,203

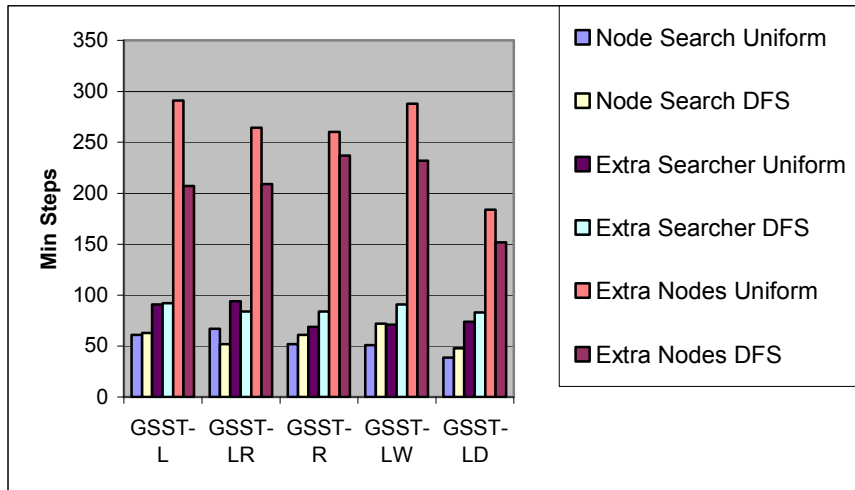
Πίνακας 6.27: Αποτελέσματα της έρευνας κόμβων στο Πλέγμα 5X5. Πείραμα 1^ο.

Edge Search(Xser.)	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	7	87	109,2	1,10%	0,02%	3,234	0,286
GSST-LR	7	87	108,4	1,20%	0,02%	4,344	0,302
GSST-R	6	70	90,4	2,94%	0,02%	2,547	0,665
GSST-LW	6	69	91,0	3,72%	0,02%	12,609	1,093
GSST-LD	7	71	84,5	4,08%	0,02%	3,891	0,322
	DFS ST generation						
Edge Traversal							
GSST-L	7	80	104,9	0,68%	0,02%	2,469	0,245
GSST-LR	7	82	102,3	0,34%	0,04%	3,672	0,266
GSST-R	7	78	98,3	1,14%	0,02%	1,969	0,461
GSST-LW	8	81	100,6	3,80%	0,04%	3,625	0,357
GSST-LD	7	75	80,3	0,24%	0,02%	2,422	0,732
Edge Search(Xnodes)	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	10	275	295,4	0,26%	0,02%	21,422	8,817
GSST-LR	11	269	285,0	0,18%	0,02%	30,109	8,331
GSST-R	11	247	280,3	0,06%	0,02%	13,859	33,708
GSST-LW	11	232	295,0	0,22%	0,02%	20,656	44,275
GSST-LD	10	180	197,4	0,18%	0,04%	20,719	21,819
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	7	215	248,9	0,30%	0,02%	14,328	3,668
GSST-LR	7	215	249,4	0,40%	0,02%	19,625	3,520
GSST-R	8	246	276,3	0,48%	0,02%	7,516	21,138
GSST-LW	8	221	240,6	0,16%	0,02%	11,453	17,769
GSST-LD	7	152	175,2	0,56%	0,02%	12,391	9,399

Πίνακας 6.28: Αποτελέσματα της έρευνας ακμών στο Πλέγμα 5X5. Πείραμα 1^ο.



Σχήμα 6.29: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Πείραμα 1^ο, στο Πλέγμα 5X5.



Σχήμα 6.30: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Πείραμα 1^ο, στο Πλέγμα 5X5.

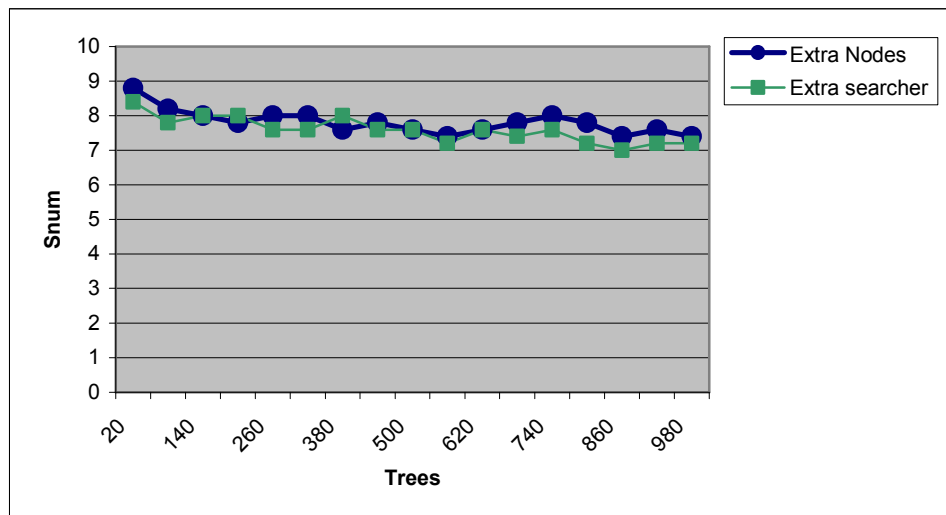
Κατ' αρχάς παρατηρούμε ότι ο πραγματικός $s_N^{imc} = 5$ επιτεύχθηκε από τον uniform GSST μόνο.

Η δεύτερη παρατήρηση είναι ο πολύ μικρότερος αριθμός βημάτων που χρειάζεται στο παιχνίδι κόμβων, σε σχέση με το παιχνίδι ακμών.

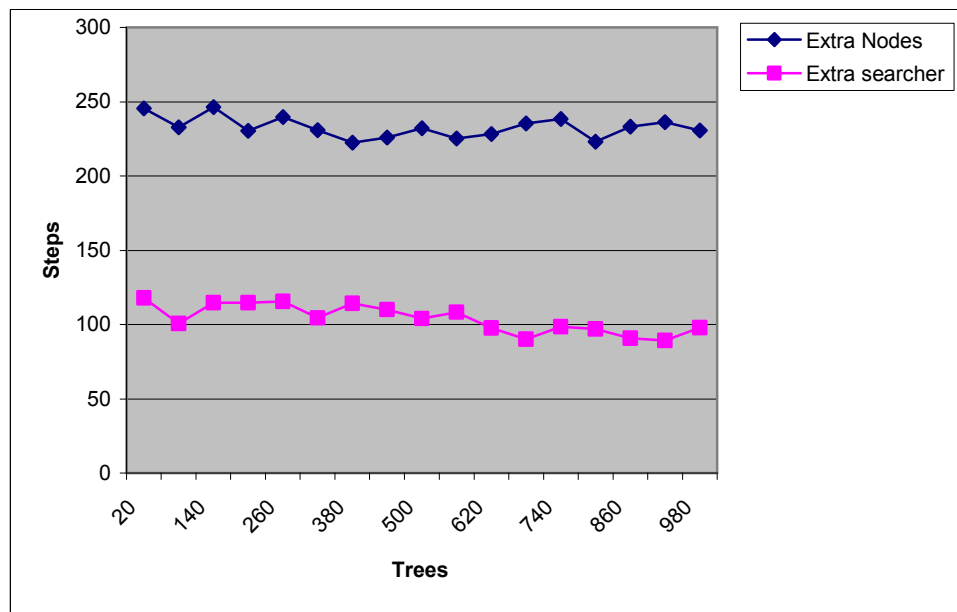
Επίσης βλέπουμε πως ο Extra Nodes λειτουργεί καλύτερα με DFS κατασκευή δένδρων. Πιθανότατα, αυτό οφείλεται στο γεγονός ότι οι ενδιαμέσοι κόμβοι έχουν μόνο δυο γειτονικούς ο καθένας και έτσι ο ψευδογράφος έρχεται πιο κοντά στους γενικούς γράφους από ότι τα πλέγματα.

Τέλος συγκρίνοντας τους δύο αλγορίθμους για το παιχνίδι ακμών, ο Extra Searcher είναι εμφανώς καλύτερος. Ο αριθμός ερευνητών είναι πολύ μικρότερος για τις παραλλαγές uniform και μικρότερος για τις DFS από αυτόν του Extra Nodes. Επιπλέον τα βήματα των ελάχιστων στρατηγικών, και ως ελάχιστος αριθμός και ως μέσος όρος, είναι σε κάθε περίπτωση λιγότερα από τα μισά αυτών του Extra Nodes. Τέλος οι χρόνοι εκτέλεσης του gsearch και του Eclearn γίνονται ασύμφορα μεγάλοι. Αυτό οφείλεται στην πολυπλοκότητα του ψευδογράφου για το gsearch, και στην υπολογιστική πολυπλοκότητα του καθαρισμού των παραγόμενων στρατηγικών από τους ενδιαμέσους κόμβους για το Eclearn.

Για το δεύτερο φύλλο πειραμάτων επιλέγουμε τον DFS GSST-LR ως ευνοϊκή περίπτωση για τον Extra Nodes (όχι όμως η ευνοϊκότερη). Ακολουθούν τα διαγράμματα των αποτελεσμάτων με 5 επαναλήψεις για κάθε βήμα.



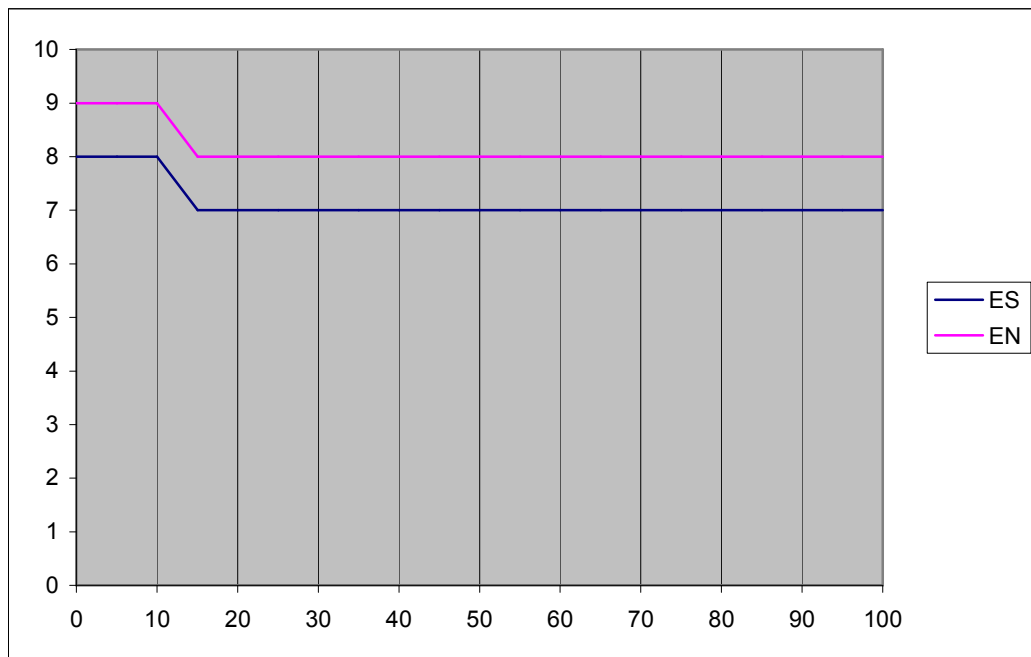
Σχήμα 6.31: Ελάχιστος αριθμός ερευνητών. Πείραμα 2^ο στο Πλέγμα 5X5.



Σχήμα 6.32: Ελάχιστος αριθμός βημάτων. Πείραμα 2^ο στο Πλέγμα 5X5.

Βλέπουμε ότι στα 980 πρώτα ο αριθμός ερευνητών του Extra Searcher είναι ελάχιστα καλύτερος, καθώς πλησιάζει πιο κοντά στο 7. Στον αριθμό των βημάτων φαίνεται καθαρά πώς η πολυπλοκότητα του ψευδογράφου παράγει ανούσιες, για τον καθαρισμό κινήσεις.

Στο τελευταίο φύλλο παρατηρούμε την πορεία βελτίωσης μιας έρευνας με 1600 γεννητορικά δένδρα.

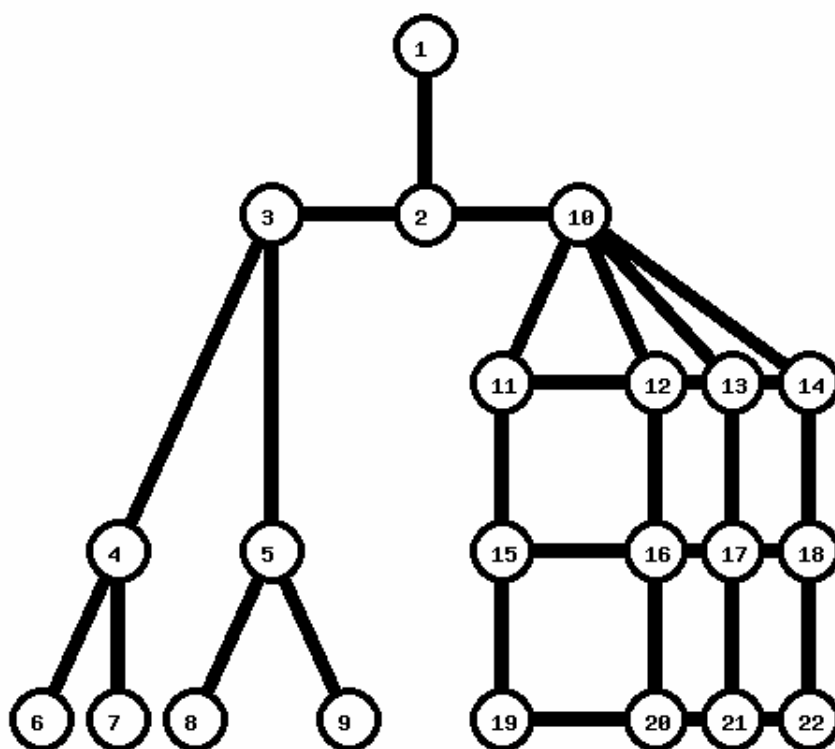


Σχήμα 6.33: Ελάχιστος αριθμός ερευνητών. Πείραμα 3^ο στο Πλέγμα 5X5.

Από τα τρία πειράματα, συμπεραίνουμε ασφαλώς ότι ο Extra Searcher κερδίζει γρήγορα έδαφος, όσο αυξάνεται η πολυπλοκότητα του γράφου.

6.1.5. Δένδρο-Πλέγμα 4X4

Με το γράφο αυτό επιχειρείται ο συνδυασμός ενός πλέγματος με ένα δένδρο. Θα δούμε έτσι ποια συμπεριφορά υπερισχύει. Ο γράφος του σχήματος 6.5 έχει $s_N^{imc} = 4$ καθώς αν ξεκινήσει μια παράλληλη σάρωσή του από κάτω προς τα πάνω, το πλέγμα καθαρίζεται με 4 ερευνητές. Έπειτα είναι αρκετοί για να καθαρίσουν και τα δένδρο. Επομένως σίγουρα αρκούν 5 ερευνητές για τον καθαρισμό του στο παιχνίδι ακμών, σύμφωνα με το θεώρημα 3.11.



Σχήμα 6.34: Δένδρο-Πλέγμα 4X4

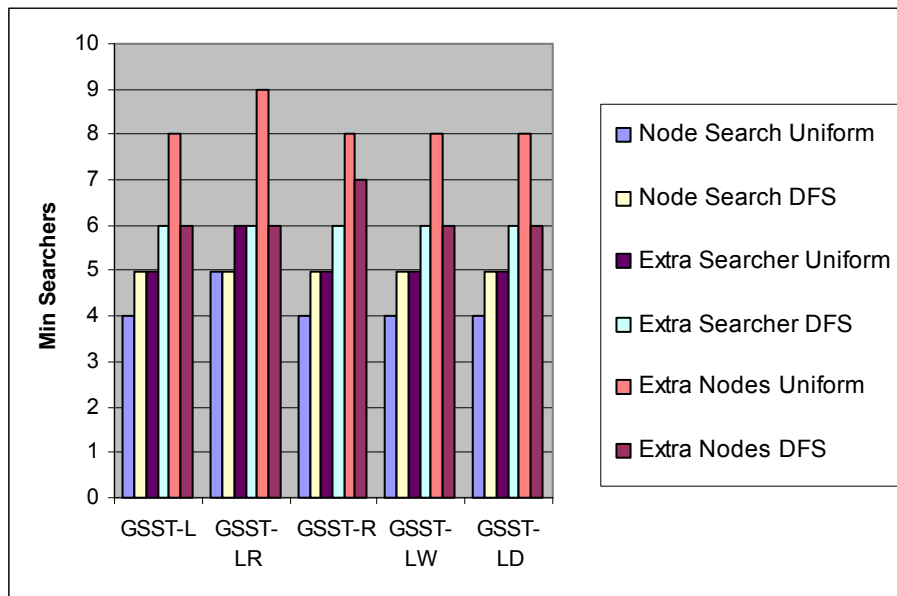
Ακολουθούν τα αποτελέσματα του πρώτου φύλλου πειραμάτων με $M=1000$.

Node Search	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	4	51	65,5	2,20%	0,10%	0,328	0,015
GSST-LR	5	55	64,8	10,20%	0,10%	0,563	0,016
GSST-R	4	47	64,3	6,30%	0,10%	0,297	0,027
GSST-LW	4	51	68,6	9,30%	0,10%	0,438	0,030
GSST-LD	4	48	58,4	4,10%	0,10%	0,328	0,025
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)
GSST-L	5	55	65,5	5,70%	0,10%	0,406	0,018
GSST-LR	5	53	65,5	7,20%	0,10%	0,531	0,019
GSST-R	5	58	68,7	6,90%	0,10%	0,328	0,030
GSST-LW	5	56	69,7	6,80%	0,10%	0,422	0,029
GSST-LD	5	49	60,1	7,80%	0,10%	0,422	0,025

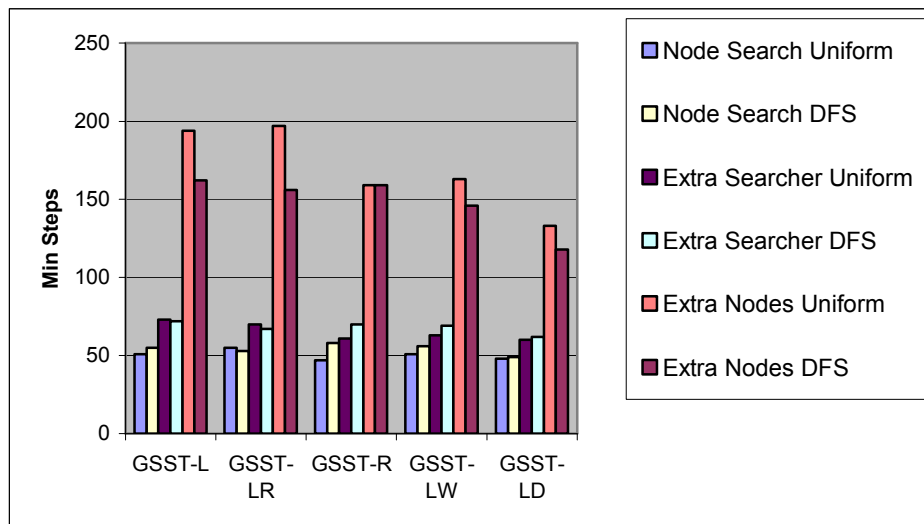
Πίνακας 6.35: Αποτελέσματα της έρευνας κόμβων στο Δένδρο-Πλέγμα. Πείραμα 1^ο.

Edge Search(Xser.)	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	5	73	83,3	2,20%	0,10%	0,328	0,057
GSST-LR	6	70	82,9	10,20%	0,10%	0,563	0,061
GSST-R	5	61	79,4	6,30%	0,10%	0,297	0,063
GSST-LW	5	63	83,6	9,30%	0,10%	0,438	0,062
GSST-LD	5	60	77,1	4,10%	0,10%	0,328	0,070
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	6	72	82,6	5,70%	0,10%	0,406	0,069
GSST-LR	6	67	82,6	7,20%	0,10%	0,531	0,067
GSST-R	6	70	81,2	6,90%	0,10%	0,328	0,059
GSST-LW	6	69	82,5	6,80%	0,10%	0,422	0,059
GSST-LD	6	62	77,6	7,80%	0,30%	0,422	0,074
Edge Search(Xnodes)	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	8	194	203,2	1,30%	0,20%	2,078	1,900
GSST-LR	9	197	216,6	1,20%	0,20%	2,953	2,181
GSST-R	8	159	218,3	1,10%	0,10%	0,984	6,453
GSST-LW	8	163	214,3	1,20%	0,10%	1,297	6,992
GSST-LD	8	133	146,5	0,40%	0,10%	1,500	3,187
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	6	162	171,3	2,50%	0,30%	1,625	0,703
GSST-LR	6	156	171,2	1,90%	0,10%	2,281	0,798
GSST-R	7	159	178,0	0,90%	0,10%	0,969	4,585
GSST-LW	6	146	180,1	3,30%	0,10%	1,547	5,251
GSST-LD	6	118	129,8	1,90%	0,10%	1,594	2,760

Πίνακας 6.36: Αποτελέσματα της έρευνας ακμών στο Δένδρο-Πλέγμα. Πείραμα 1^ο.



Σχήμα 6.37: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Πείραμα 1^ο, στο Δένδρο-Πλέγμα.



Σχήμα 6.38: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Πείραμα 1^ο, στο Δένδρο-Πλέγμα.

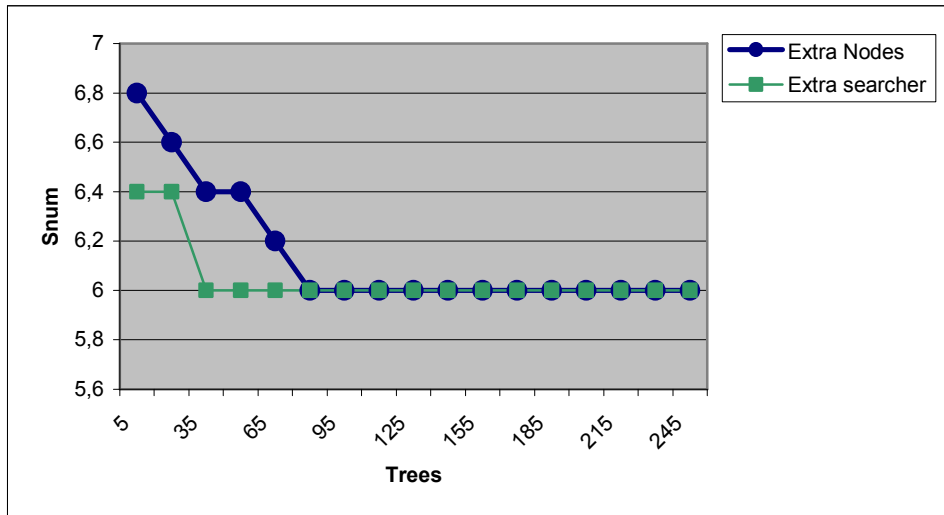
Αρχικά παρατηρούμε ότι ο GSST κατάφερε να βρει τον ελάχιστο αριθμό ερευνητών με τις uniform παραλλαγές του, ενώ οι DFS χρειάζονται έναν παραπάνω. Όπως φάνηκε και από τα πειράματα της [41] η καλύτερη συμπεριφορά των uniform παραλλαγών είναι χαρακτηριστικό των πλεγμάτων. Το γεγονός αυτό, σε συνδυασμό με την απλότητα των δένδρων, μας υποδεικνύει ότι επικρατούν τα χαρακτηριστικά του πλέγματος.

Επίσης παρατηρούμε για άλλη μια φορά ότι ο Extra Nodes προτιμά τις DFS παραλλαγές του GSST. Αντιθέτως ο Extra Searcher φαίνεται, όπως λογικά περιμέναμε, να υιοθετεί την συμπεριφορά του GSST και να προτιμά τις uniform παραλλαγές για τα πλέγματα (άρα και για τον γράφο του σχήματος 6.5 όπου επικρατούν τα χαρακτηριστικά του πλέγματος).

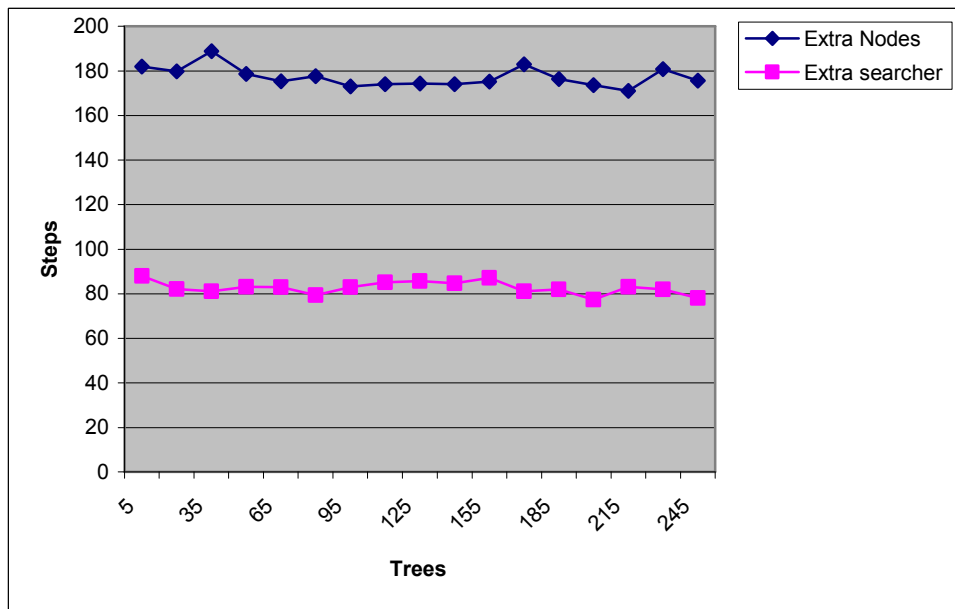
Για άλλη μια φορά βλέπουμε ότι η έρευνα κόμβων απαιτεί σαφώς λιγότερα βήματα.

Συγκρίνοντας τους δύο αλγορίθμους, και πάλι ο Extra Searcher είναι καλύτερος από άποψη αριθμού ερευνητών, αριθμού βημάτων, ποσοστού ελάχιστων ερευνών αλλά και χρόνου εκτέλεσης. Βλέπουμε μάλιστα πόσο μεγαλύτεροι είναι οι χρόνοι του Extra Nodes.

Για το δεύτερο φύλλο πειραμάτων επιλέγουμε τον DFS GSST-L ως ευνοϊκό για τον Extra Nodes. Έγιναν 5 επαναλήψεις των δοκιμών σε κάθε βήμα της έρευνας.



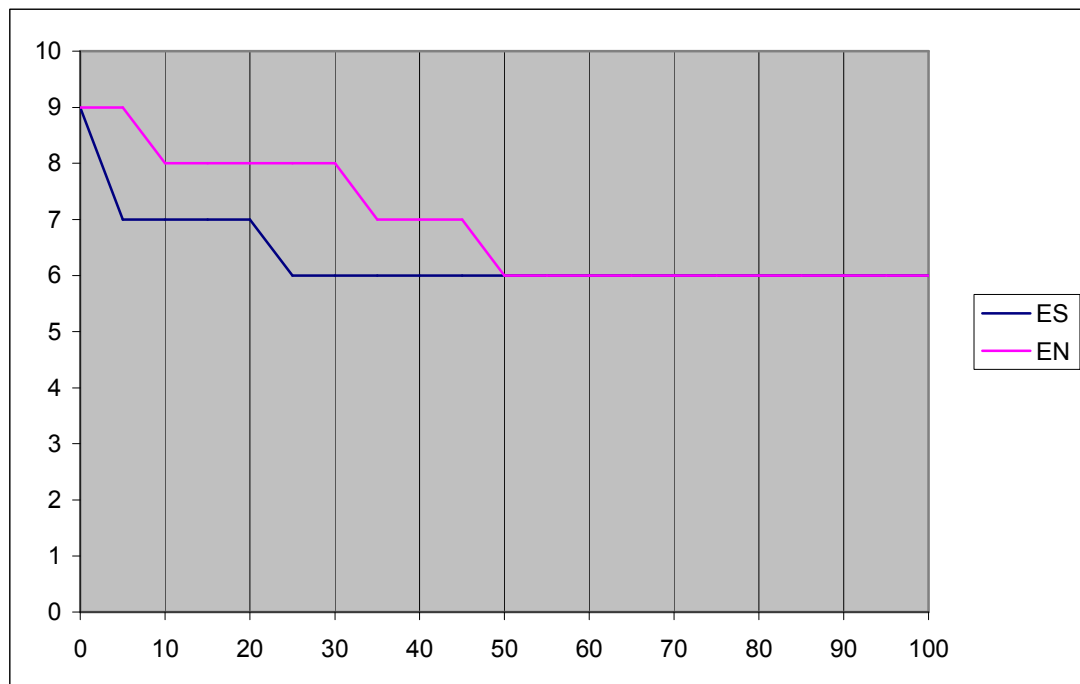
Σχήμα 6.39: Ελάχιστος αριθμός ερευνητών. Πείραμα 2^ο στο Δένδρο-Πλέγμα.



Σχήμα 6.40: Ελάχιστος αριθμός βημάτων. Πείραμα 2^ο στο Δένδρο-Πλέγμα.

Βλέπουμε ότι παρά την ευνοϊκή προς τον Extra Nodes επιλογή της παραλλαγής του GSST ο Extra Searcher φτάνει πιο γρήγορα στους 6 ερευνητές και μάλιστα με λιγότερα από τα μισά βήματα στην στρατηγική έρευνας.

Τέλος το διάγραμμα της βελτίωσης του τρίτου φύλλου με $M=40$.

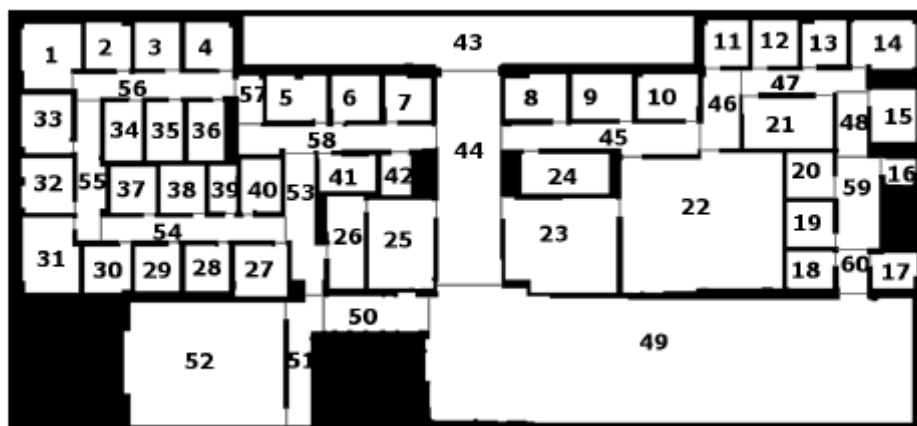


Σχήμα 6.41: Ελάχιστος αριθμός ερευνητών. Πείραμα 3^ο στο Δένδρο-Πλέγμα.

Τα συμπεράσματα και από τα τρία φύλλα είναι ξεκάθαρα ευνοϊκά για τον Extra Searcher.

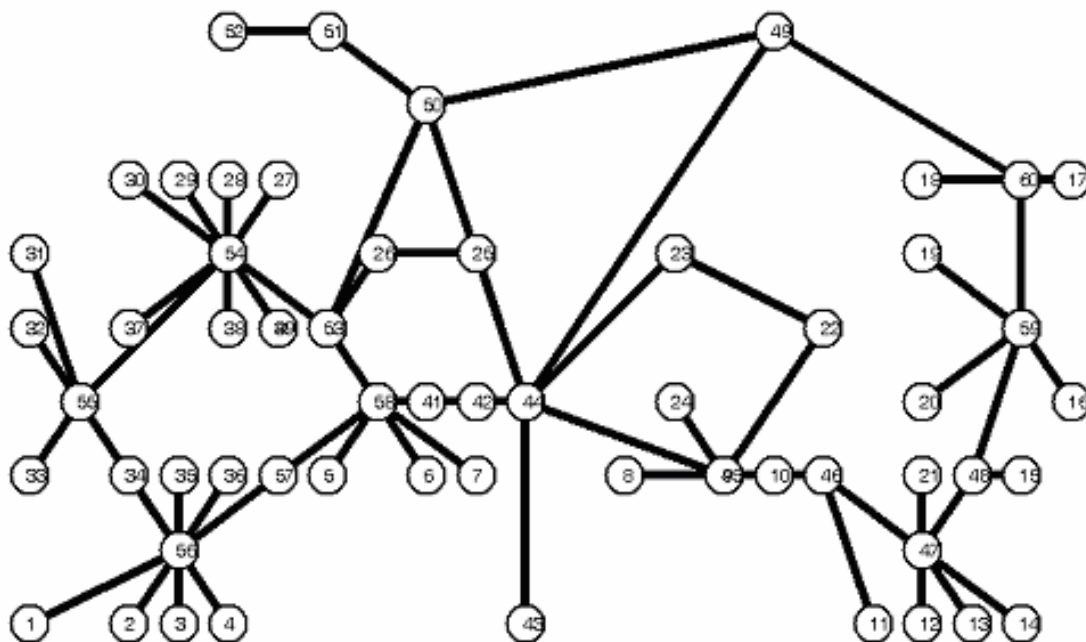
6.1.6. Γράφος του NSH

Στο σχήμα 6.39 βλέπουμε την κάτοψη του ισογείου του κτιρίου Newel-Simon στο Carnegie Mellon University.



Σχήμα 6.42: Κάτοψη του ισογείου του κτιρίου Newel-Simon.

Η κάτοψη του Σχ.6.39 μπορεί να αναχθεί στον γράφο του Σχ. 6.40



Σχήμα 6.43:Γράφος του ισογείου του κτιρίου Newel-Simon.

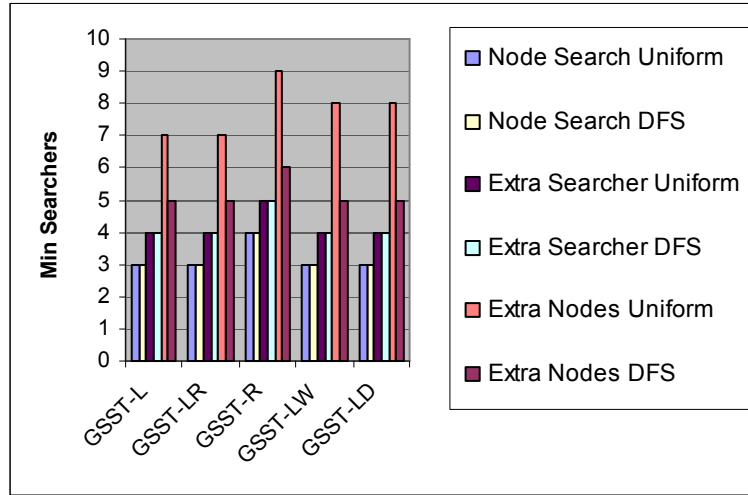
Ο γράφος του σχήματος 6.40 έχει 61 κόμβους (δωμάτια) και 67 ακμές (πόρτες) , είναι πιο απλός ως γράφος από τους δύο προηγούμενους. Η απλή node-clearing έρευνα, έχει δώσει σε προηγούμενα πειράματα $s_N^{imc} = 3$. Επομένως ο extra searcher αλγόριθμος, κατά πάσα πιθανότητα θα καθαρίσει πλήρως το γράφο με 4 ερευνητές.

Επίσης το γεγονός ότι οι ακμές είναι σχετικά αραιές μας προδιαθέτει θετικά προς τον extra nodes, αλλά ο βαθμός πολυπλοκότητας του γράφου εξακολουθεί να είναι μεγάλος.

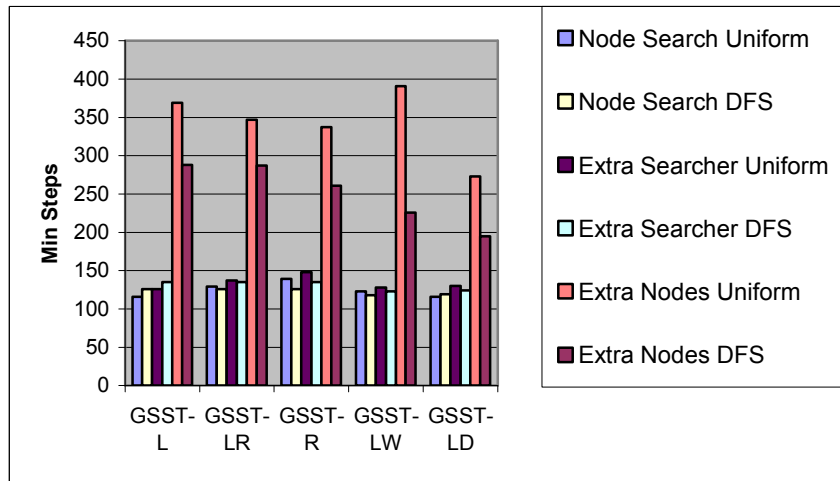
Το πρώτο φύλλο πειραμάτων εκτελέστηκε με $M=4000$.

Node Search	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	3	116	143,3	1,050%	0,025%	3,094	0,071
GSST-LR	3	129	139,7	0,375%	0,025%	3,781	0,026
GSST-R	4	139	166,5	1,425%	0,025%	1,766	0,578
GSST-LW	3	123	152,9	1,375%	0,025%	2,266	0,593
GSST-LD	3	116	137,0	0,975%	0,025%	2,469	0,463
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)
GSST-L	3	126	128,3	1,100%	0,225%	2,922	0,035
GSST-LR	3	126	131,1	1,425%	0,150%	4,047	0,041
GSST-R	4	126	157,8	3,225%	0,025%	2,016	0,595
GSST-LW	3	118	138,0	1,500%	0,025%	2,500	0,577
GSST-LD	3	119	121,6	1,000%	0,250%	2,672	0,462
Edge Search(Xser.)	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	4	126	157,1	1,050%	0,025%	3,094	0,311
GSST-LR	4	137	152,1	0,375%	0,025%	3,781	0,228
GSST-R	5	148	174,5	1,425%	0,025%	1,766	0,725
GSST-LW	4	128	159,6	1,375%	0,025%	2,266	0,722
GSST-LD	4	130	137,9	0,975%	0,025%	2,469	0,479
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	4	135	137,3	1,100%	0,225%	2,922	0,188
GSST-LR	4	135	140,8	1,425%	0,150%	4,047	0,227
GSST-R	5	135	162,9	3,225%	0,025%	2,016	0,758
GSST-LW	4	123	142,6	1,500%	0,025%	2,500	0,665
GSST-LD	4	124	128,6	1,000%	0,225%	2,672	0,532
Edge Search(Xnodes)	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	7	369	397,0	0,850%	0,050%	17,531	44,159
GSST-LR	7	347	378,9	0,300%	0,025%	25,875	37,796
GSST-R	9	337	400,8	0,275%	0,025%	9,172	129,262
GSST-LW	8	391	444,9	0,375%	0,025%	12,047	152,991
GSST-LD	8	273	291,0	0,175%	0,025%	14,094	71,473
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	5	288	316,7	8,125%	0,025%	14,047	10,183
GSST-LR	5	287	312,2	8,275%	0,025%	19,063	10,093
GSST-R	6	261	290,2	0,550%	0,025%	8,266	63,917
GSST-LW	5	226	286,2	2,425%	0,025%	12,453	78,104
GSST-LD	5	195	212,2	1,225%	0,025%	12,750	42,925

Πίνακας 6.44: Αποτελέσματα του 1^{ου} πειράματος στο γράφο του NSH.



Σχήμα 6.45: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Πείραμα 1^ο, στο γράφο του NSH.



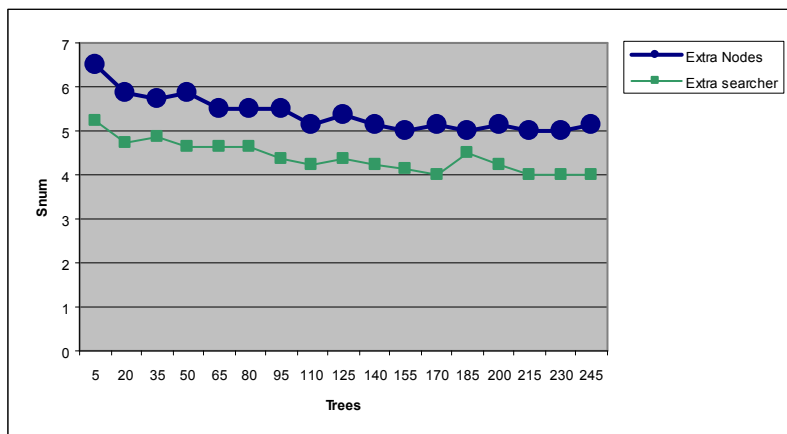
Σχήμα 6.46: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Πείραμα 1^ο, στο γράφο του NSH.

Παρατηρούμε ότι ο ελάχιστος αριθμός ερευνητών επιτεύχθηκε για το παιχνίδι ακμών και ο Extra Searcher, ακολουθώντας τις n-καθαριστικές στρατηγικές καθάρισε τον γράφο για το παιχνίδι ακμών με 4 ερευνητές. Ο Extra Nodes χρειάζεται έναν παραπάνω ερευνητή και εξακολουθεί να βγάζει καλύτερα αποτελέσματα με τις DFS παραλλαγές του GSST.

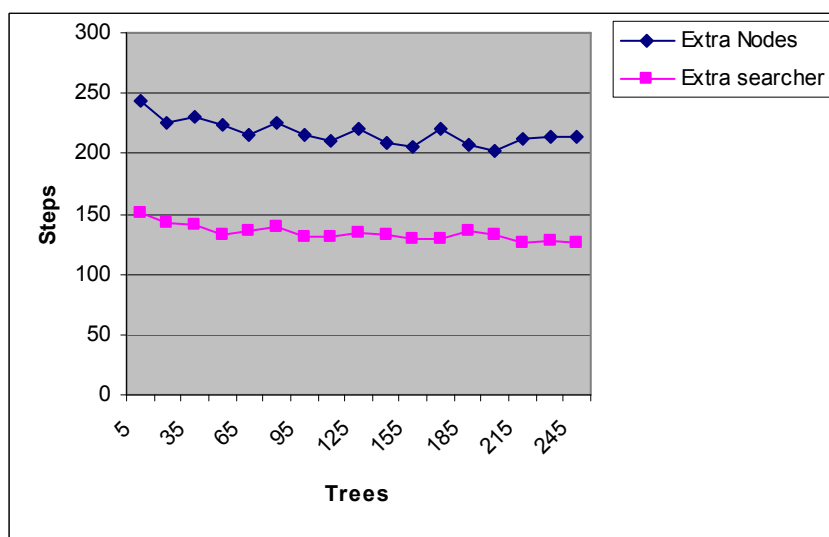
Επίσης βλέπουμε η έρευνα ακμών με τον Extra Searcher δεν απαιτεί πολύ περισσότερα βήματα από την έρευνα κόμβων. Αυτό είναι απόρροια της αραιής δομής του γράφου που καθιστά την χρήση του καθαριστή ακμών ελάχιστη. Αντίθετα ο Extra Nodes παράγει, για άλλη μια φορά, στρατηγικές που χρειάζονται πολύ περισσότερα βήματα. Τα συμπεράσματα για τα διάφορα μεγέθη των ερευνών εξακολουθούν να είναι ευνοϊκά για τον Extra Searcher. Δεν πρέπει να παραπλανηθούμε από τα υψηλά ποσοστά των ελάχιστων ερευνών του Extra Nodes για τις DFS παραλλαγές, καθώς αυτά αναφέρονται στις έρευνες με $s_E^{inc} = 5$, ενώ αυτά του Extra Nodes σ' αυτές με $s_E^{inc} = 4$ (εκτός του GSST-R).

Τέλος ενδιαφέρον είναι ότι οι GSST-R παραλλαγές δεν είναι τόσο αποδοτικές. Αυτό μπορεί να εξηγηθεί από τη δομή του γράφου. Με την τυχαία επιλογή σε ένα αραιό αλλά ταυτόχρονα πολύπλοκο γράφο μπορεί πολύ εύκολα να παραχθούν ανούσιες κινήσεις. Η επιστροφή όμως του ερευνητή στον ενεργητικό καθαρισμό του γράφου θα απαιτήσει πολλά βήματα καθώς η αραιότητα του γράφου έχει ως αποτέλεσμα να μην υπάρχουν πάντα σύντομα μονοπάτια.

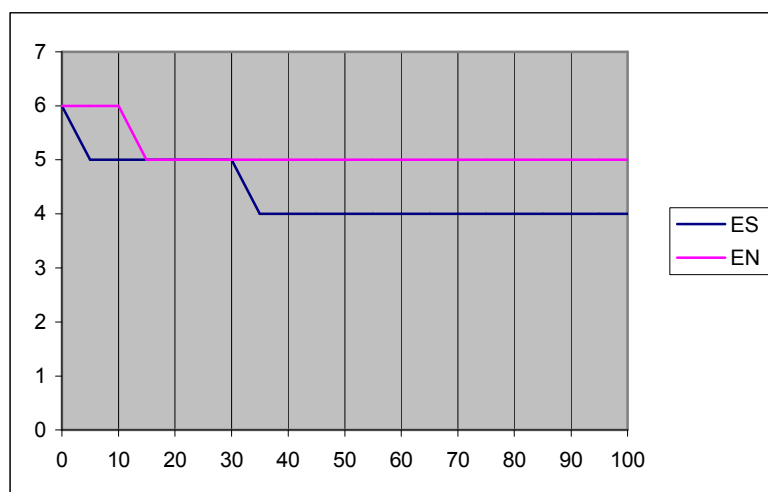
Ακολουθούν τα αποτελέσματα του δεύτερου και του τρίτου φύλλου πειραμάτων για τον DFS GSST-LR . Εκτελέστηκαν 8 επαναλήψεις σε κάθε βήμα της έρευνας, ενώ στο τρίτο φύλλο επιλέχθηκε $M=100$



Σχήμα 6.47: Ελάχιστος αριθμός ερευνητών. Πείραμα 2^ο στο γράφο του NSH.



Σχήμα 6.48: Ελάχιστος αριθμός βημάτων. Πείραμα 2^ο στο γράφο του NSH.

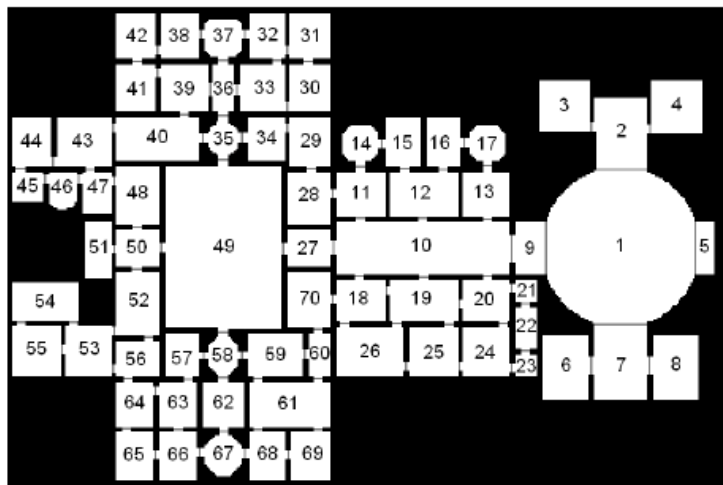


Σχήμα 6.49: Ελάχιστος αριθμός ερευνητών. Πείραμα 2^ο στο γράφο του NSH.

Ο Extra Searcher φαίνεται να είναι καλύτερος για άλλη μια φορά.

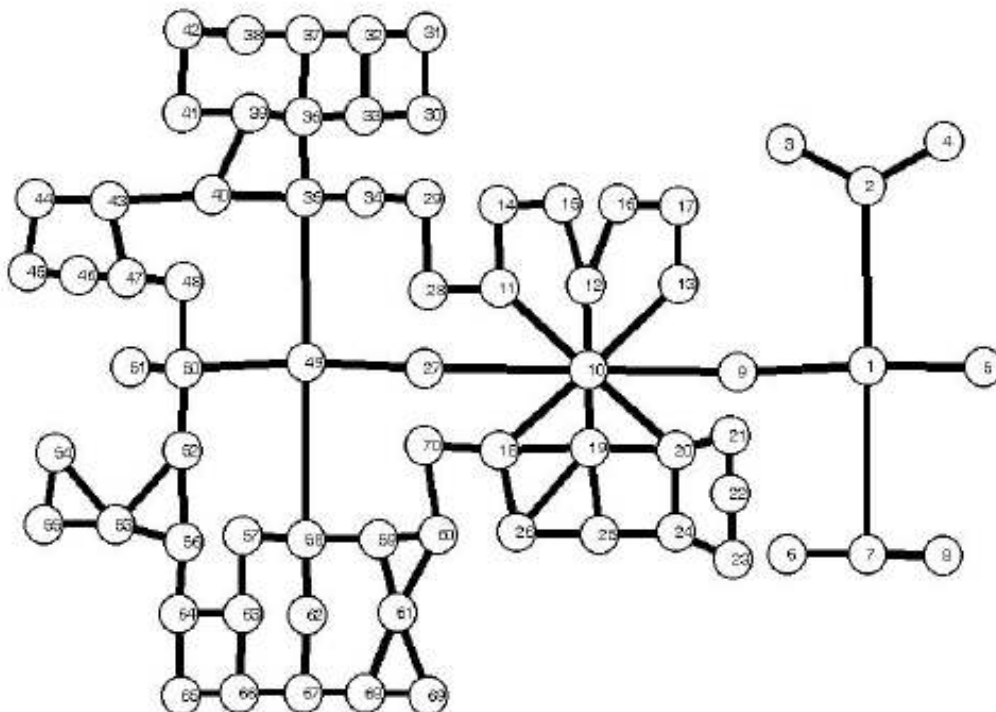
6.1.7. Εθνική πινακοθήκη (Washington D.C.)

Στο Σχ.6.46 βλέπουμε την κάτοψη του ισογείου της εθνικής πινακοθήκης στην Washington D.C.



Σχήμα 6.50: Κάτοψη του ισογείου της εθνικής πινακοθήκης (Washington D.C.)

Η κάτοψη του σχήματος 6.46 μπορεί να αναπτυχθεί στον γράφο του σχήματος 6.47



Σχήμα 6.51: του ισογείου της εθνικής πινακοθήκης (Washington D.C.)

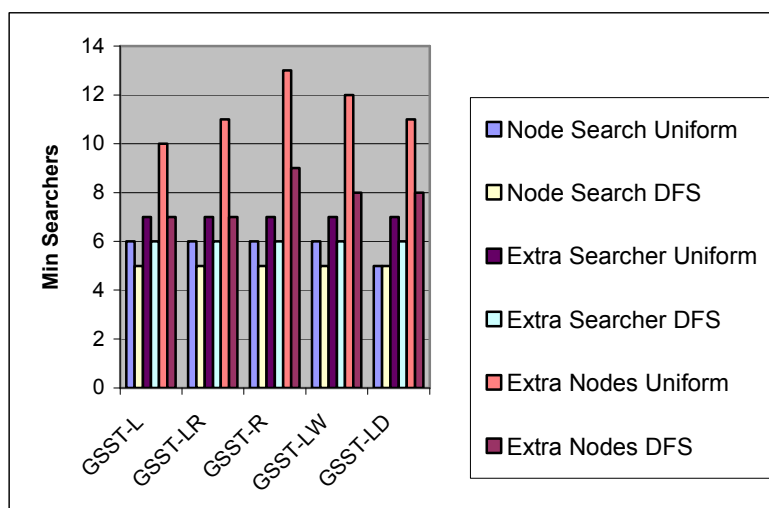
Ο γράφος του σχήματος 6.8 αποτελείται από 70 κόμβους και 93 ακμές και είναι πολυπλοκότερος από όλους τους προηγούμενους γράφους. Έχει μιάμιση φορά της ακμές του NSH αλλά έχει πάρα πολύ μεγάλο αριθμό γεννητορικών δένδρων (περίπου

$5,3 \times 10^{14}$ υπολογισμένα με το θεώρημα Kirchoff) σε σχέση με το NSH (3604). Επιπλέον, δεν μπορούμε να προβλέψουμε τους ελάχιστους αριθμούς ερευνητών αλλά προηγούμενα πειράματα με $M=1,5 \times 10^{15}$, στην [41] έδωσαν ελάχιστο αριθμό ερευνητών ίσο με 5 για το παιχνίδι ακμών.

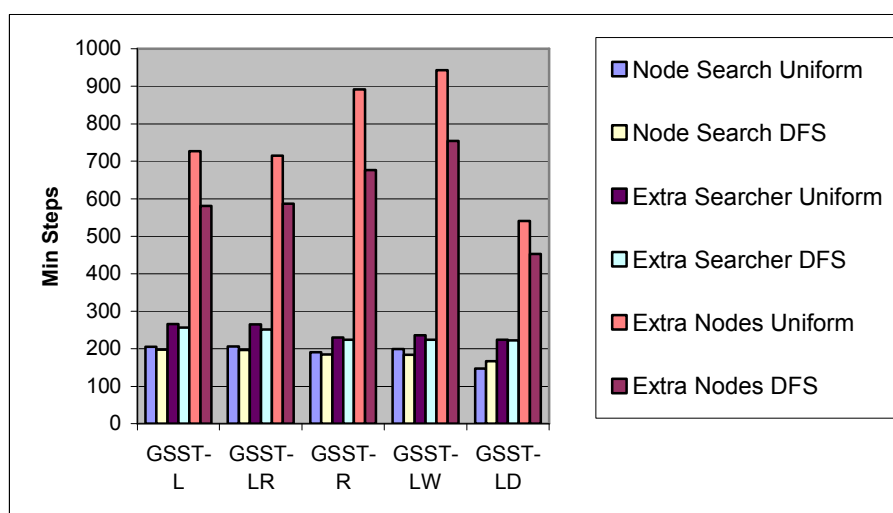
Ακολουθούν τα αποτελέσματα του πρώτου φύλλου πειραμάτων με $M=60000$.

Node Search	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	6	205	255,9	0,32%	1,66667E-05	92,906	0,933
GSST-LR	6	206	253,7	0,24%	1,66667E-05	131,719	0,869
GSST-R	6	191	227,6	0,13%	1,66667E-05	45,281	1,554
GSST-LW	6	199	243,3	0,28%	1,66667E-05	62,578	1,817
GSST-LD	5	147	200,7	0,11%	1,66667E-05	62,656	1,154
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)
GSST-L	5	198	219,9	0,07%	1,66667E-05	99,625	0,691
GSST-LR	5	197	212,5	0,02%	1,66667E-05	140,500	0,634
GSST-R	5	185	198,4	0,05%	1,66667E-05	50,859	1,018
GSST-LW	5	184	202,5	0,09%	1,66667E-05	77,797	1,078
GSST-LD	5	167	190,6	0,09%	1,66667E-05	78,359	0,955
Edge Search(Xser.)	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	7	266	326,4	0,32%	1,66667E-05	92,906	6,001
GSST-LR	7	265	324,2	0,24%	1,66667E-05	131,719	5,849
GSST-R	7	230	279,9	0,13%	1,66667E-05	45,281	4,188
GSST-LW	7	236	291,8	0,28%	3,33333E-05	62,578	4,277
GSST-LD	6	224	263,0	0,11%	1,66667E-05	62,656	3,968
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	6	256	283,0	0,07%	1,66667E-05	99,625	4,491
GSST-LR	6	251	276,3	0,02%	1,66667E-05	140,500	4,438
GSST-R	6	224	248,0	0,05%	1,66667E-05	50,859	3,364
GSST-LW	6	224	249,8	0,09%	1,66667E-05	77,797	3,245
GSST-LD	6	222	244,1	0,09%	1,66667E-05	78,359	4,093
Edge Search(Xnodes)	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	10	727	824,6	0,02%	1,66667E-05	771,984	381,254
GSST-LR	11	715	755,8	0,01%	1,66667E-05	483,674	282,222
GSST-R	13	892	1003,7	0,01%	1,66667E-05	294,875	995,398
GSST-LW	12	943	1057,4	0,01%	1,66667E-05	425,078	1403,428
GSST-LD	11	541	597,6	0,02%	1,66667E-05	491,688	420,237
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	7	581	665,2	0,23%	1,66667E-05	566,719	154,485
GSST-LR	7	587	670,8	0,07%	1,66667E-05	480,630	169,045
GSST-R	9	676	854,9	0,02%	1,66667E-05	263,234	867,365
GSST-LW	8	755	855,7	0,03%	1,66667E-05	463,406	1015,267
GSST-LD	8	453	512,2	0,21%	1,66667E-05	475,859	303,319

Πίνακας 6.52: Αποτελέσματα του 1^{ου} πειράματος στο γράφο της εθνικής πινακοθήκης (Washington D.C.)



Σχήμα 6.53: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Πείραμα 1^ο, στο γράφο της εθνικής πινακοθήκης (Washington D.C.)



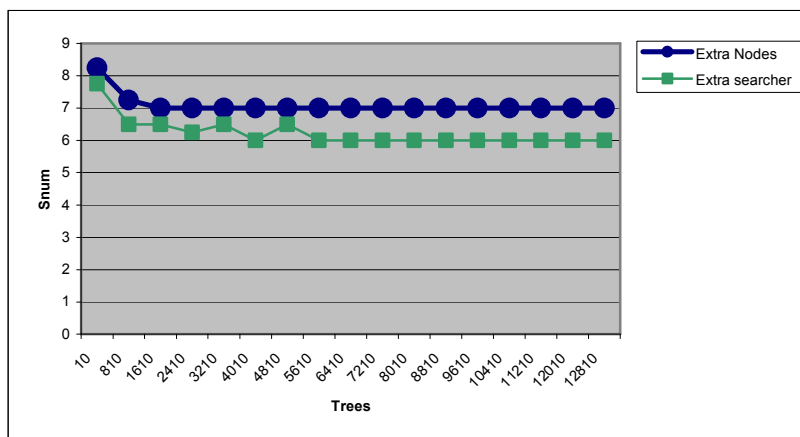
Σχήμα 6.54: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Πείραμα 1^ο, στο γράφο της εθνικής πινακοθήκης (Washington D.C.)

Αρχικά παρατηρούμε ότι ο GSST εντόπισε στρατηγικές με 5 ερευνητές και ακολούθως ο Extra Searcher με 6. Βλέπουμε πως και σε αυτόν το γενικό γράφο οι DFS παραλλαγές λειτουργούν καλύτερα για την απλή έρευνα κόμβων και συνεπώς και για τον Extra Searcher. Σε αυτόν τον πιο πυκνό γράφο βλέπουμε πως απαιτούνται αρκετά περισσότερα βήματα για το παιχνίδι ακμών από το παιχνίδι κόμβων.

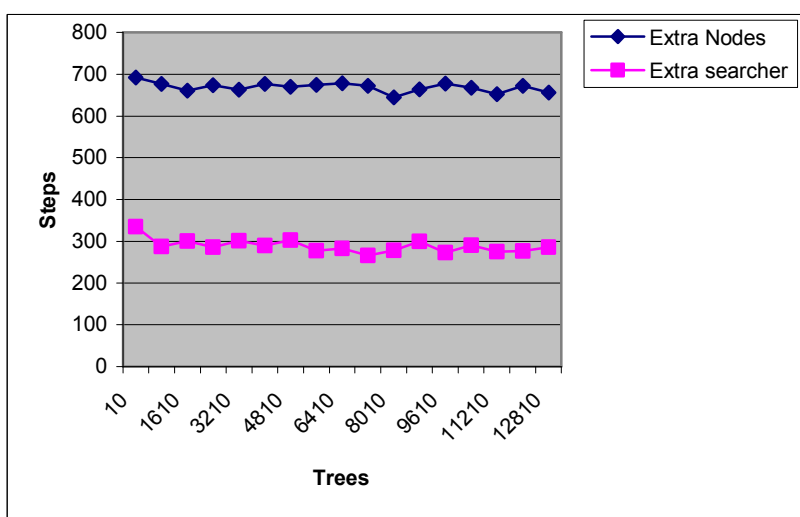
Παρατηρούμε επίσης πώς η μεγάλη πολυπλοκότητα του γράφο (και άρα η πολύ μεγαλύτερη του ψευδογράφου) έχουν ως αποτέλεσμα τους πολύ μεγαλύτερους αριθμούς βημάτων με τον Extra Nodes αλγόριθμο.

Τέλος βλέπουμε ότι και από άποψη αριθμού ερευνητών και από άποψη χρόνου εκτέλεσης ο Extra Searcher υπερισχύει ξανά. Χαρακτηριστικά μεγάλοι είναι και οι χρόνοι εκτέλεσης του gsearch πάνω στον ψευδογράφο.

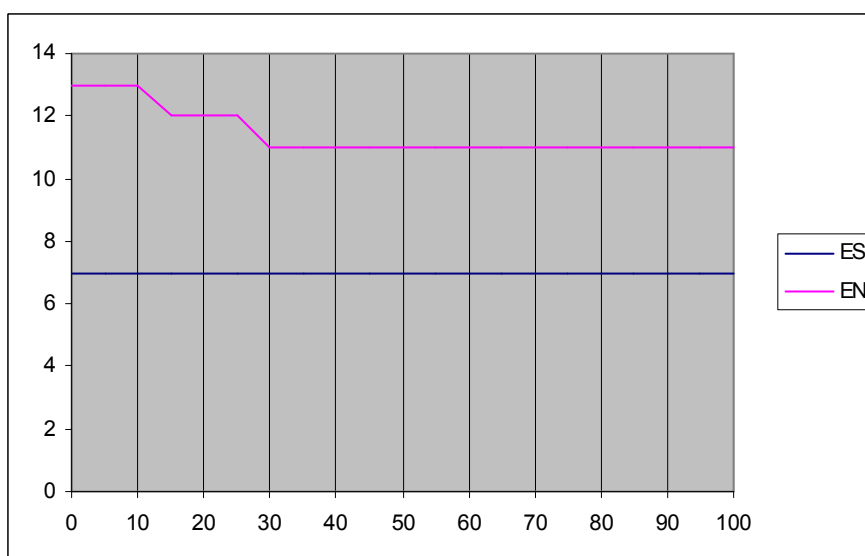
Ακολουθούν τα διαγράμματα των υπολοίπων φύλλων πειραμάτων. Στο δεύτερο φύλλο χρησιμοποιήθηκε ο DFS GSST-LR με 4 επαναλήψεις κάθε φορά και στο τρίτο ο uniform GSST-LR για $M=3000$.



Σχήμα 6.55: Ελάχιστος αριθμός ερευνητών. Πείραμα 2^ο στο γράφο της εθνικής πινακοθήκης (Washington D.C.)



Σχήμα 6.56: Ελάχιστος αριθμός βημάτων. Πείραμα 2^ο στο γράφο της εθνικής πινακοθήκης (Washington D.C.)



Σχήμα 6.57: Ελάχιστος αριθμός ερευνητών. Πείραμα 3^ο στο γράφο της εθνικής πινακοθήκης (Washington D.C.)

Για άλλη μια φορά διαφαίνεται η ανωτερότητα του Extra Searcher στην επίλυση πολύπλοκων γράφων.

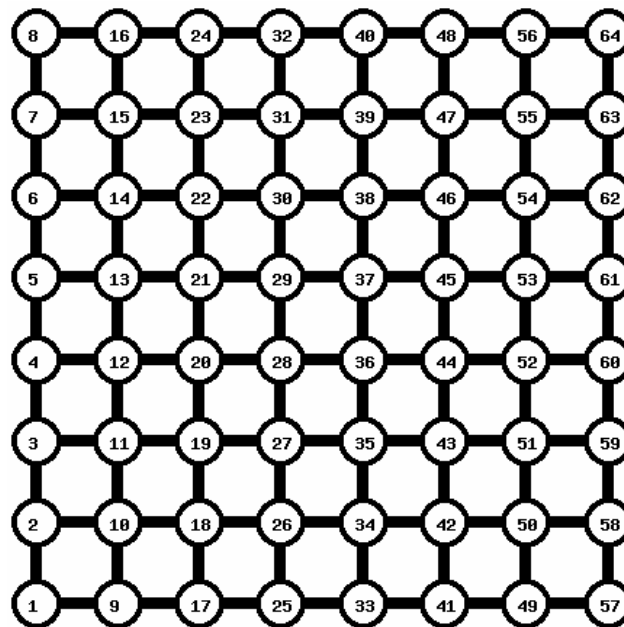
Συμπερασματικά μπορούμε να πούμε ότι ο Extra Nodes λειτουργεί αρκετά καλά στους απλούς γράφους. Ορισμένες φορές μάλιστα, στις εξαιρέσεις που δεν χρειάζεται νέος ερευνητής, λειτουργεί καλύτερα από τον Extra Searcher. Τα πράγματα αλλάζουν πολύ όμως στους πιο περίπλοκους γράφους όπου ο Extra Nodes γίνεται υπερβολικά ασύμφορος από άποψη χρόνου εκτέλεσης. Επίσης αδυνατεί συχνά να βρει το καλύτερο s_E^{inc} , ενώ όσο αυξάνεται η έκταση του γράφου, εκμηδενίζονται οι ελπίδες να αποτελεί εξαίρεση του Θεωρήματος 3.11 και να μην χρειάζεται ο καθαριστής ακμών.

6.2 Οικογένειες γράφων

Τα πειράματα της ενότητας 6.1 έδειξαν ξεκάθαρα ότι το Extra Nodes γίνεται ασύμφορος καθώς αυξάνεται η πολυπλοκότητα των γράφων. Στη δεύτερη ενότητα των πειραμάτων θα ασχοληθούμε με δύο οικογένειες γράφων. Η πολυπλοκότητα τους και η μαζικότητα των πειραμάτων επιβάλλουν να εγκαταλειφθεί ο Extra Nodes και να επικεντρωθούμε πλέον στην αποδοτικότητα του Extra Searcher. Τα συγκριτικά πειράματα (φύλλο 2 και 3) δεν έχουν τώρα κάποιο νόημα, επομένως παραμένει μόνο το πρώτο φύλλο πειραμάτων

6.2.1 Πλήρη Πλέγματα

Αν επεκτείνουμε τις δοκιμές μας από το πλέγμα 5 X 5 (Κεφ.6.1.4) και σε μεγαλύτερα πλέγματα, προκύπτει μια ολόκληρη οικογένεια γράφων με παρόμοιες ιδιότητες. Τα πλήρη πλέγματα είναι γενικά πολύπλοκοι γράφοι με πολλά γεννητορικά δένδρα λόγω της πυκνής τους δομής. Είναι προφανές ότι ένα πλέγμα $J_1 \times J_2$ μπορεί να καθαριστεί για το παιχνίδι ακμών με $J_0 = \min(J_1, J_2)$ ερευνητές, τοποθετώντας τους κατά μήκος της μικρότερης διάστασης και έπειτα ολισθαίνοντάς τους με παράλληλες κινήσεις μέχρι το τέλος της μεγάλης διάστασης. Επομένως ξέρουμε π.χ. ότι ένα πλήρες πλέγμα 5 X 3 μπορεί να η-καθαριστεί με 3 ερευνητές. Επιπλέον σύμφωνα με το Θεώρημα 3.11 χρειάζεται το πολύ ένας ακόμη για να καθαριστεί για το παιχνίδι ακμών.



Σχήμα 6.58: Πλήρες πλέγμα 8 X 8

Στα πειράματά μας ασχοληθήκαμε μόνο με τετραγωνικά πλέγματα από 5 X 5 ως 10 x 10 . Ακολουθεί ένας πίνακας που αναφέρει τους αριθμούς των γεννητορικών δένδρων κάθε εξεταζόμενου πλέγματος.

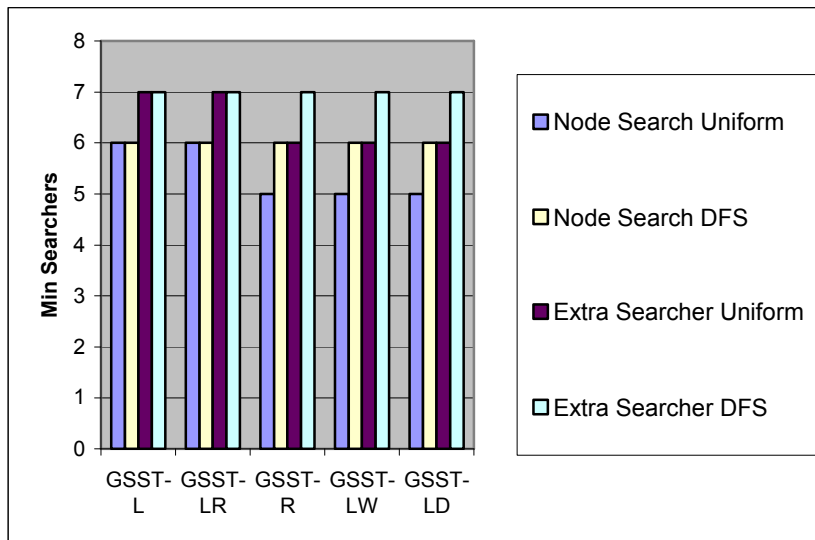
Graph Dim.	No. of spanning trees
5x5	$5.57560 \cdot 10^5$
6x6	$3.2566 \cdot 10^{13}$
7x7	$1.9872 \cdot 10^{19}$
8x8	$1.2623 \cdot 10^{26}$
9x9	$8.3266 \cdot 10^{33}$
10x10	$5.6943 \cdot 10^{42}$

Πίνακας 6.59 Ο αριθμός των γεννητορικών δένδρων για τα πλήρη πλέγματα

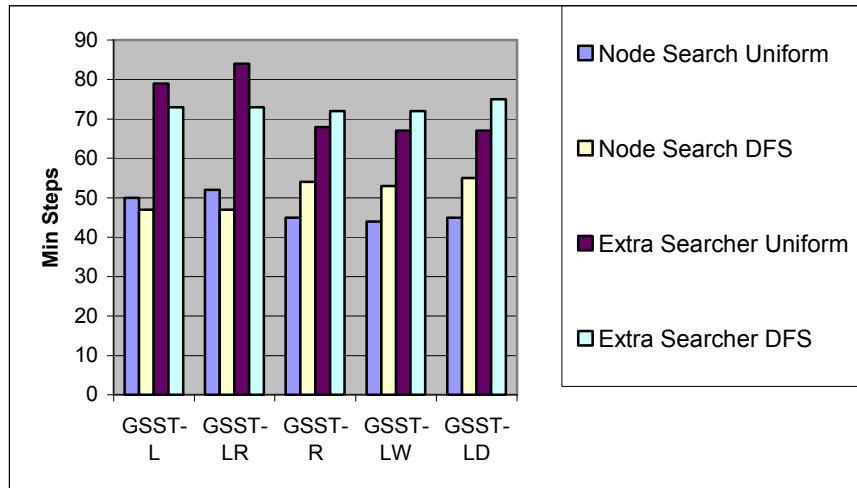
Για το πλέγμα 5 x 5 το πείραμα ξανάγινε με $M = 10^5$ καθώς τώρα, αντίθετα με αυτό της παραγράφου 6.1.4, δεν εξετάζουμε τον Extra Nodes.

Node Search	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	6	50	74,6	1,33%	0,001%	42,359	0,030
GSST-LR	6	52	74,8	1,09%	0,001%	59,141	0,034
GSST-R	5	45	65,9	1,86%	0,001%	28,469	0,047
GSST-LW	5	44	66,0	0,47%	0,001%	32,750	0,054
GSST-LD	5	45	66,7	0,53%	0,003%	34,297	0,047
DFS ST generation							
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)
GSST-L	6	47	63,5	0,12%	0,002%	46,281	0,043
GSST-LR	6	47	65,1	0,12%	0,002%	60,344	0,041
GSST-R	6	54	65,5	0,12%	0,002%	32,375	0,049
GSST-LW	6	53	64,5	0,12%	0,001%	43,844	0,049
GSST-LD	6	55	64,2	0,14%	0,001%	42,484	0,050
Edge Search(Xser.)							
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	7	79	107,1	1,33%	0,001%	42,359	0,193
GSST-LR	7	84	107,3	1,09%	0,001%	59,141	0,209
GSST-R	6	68	91,5	1,86%	0,001%	28,469	0,138
GSST-LW	6	67	91,5	0,47%	0,001%	32,750	0,145
GSST-LD	6	67	82,8	0,53%	0,001%	34,297	0,186
DFS ST generation							
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	7	73	94,9	0,12%	0,001%	46,281	0,211
GSST-LR	7	73	97,2	0,12%	0,001%	60,344	0,223
GSST-R	7	72	86,6	0,12%	0,001%	32,375	0,131
GSST-LW	7	72	86,3	0,12%	0,001%	43,844	0,136
GSST-LD	7	75	86,2	0,14%	0,001%	42,484	0,199

Πίνακας 6.60: Αποτελέσματα του 1^{ου} πειράματος στο Πλέγμα 5 X 5 ($M = 10^5$).



Σχήμα 6.61: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Πείραμα 1^ο στο Πλέγμα 5 X 5 ($M = 10^5$).

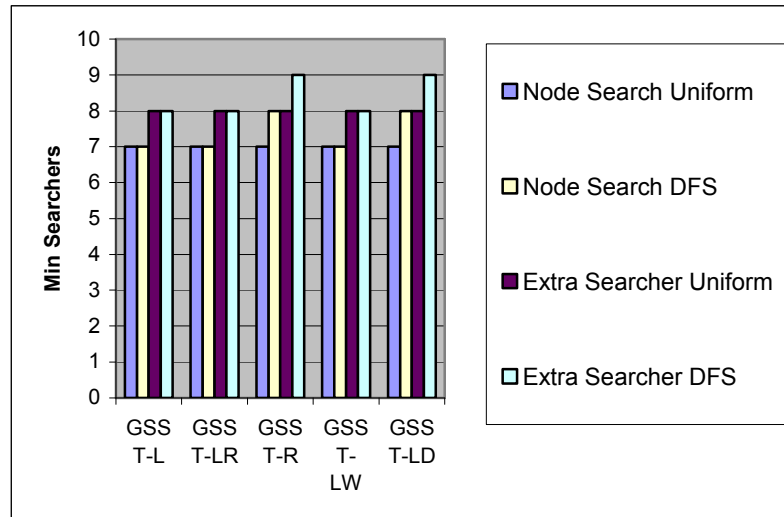


Σχήμα 6.62: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Πείραμα 1^ο στο Πλέγμα 5 X 5 ($M = 10^5$).

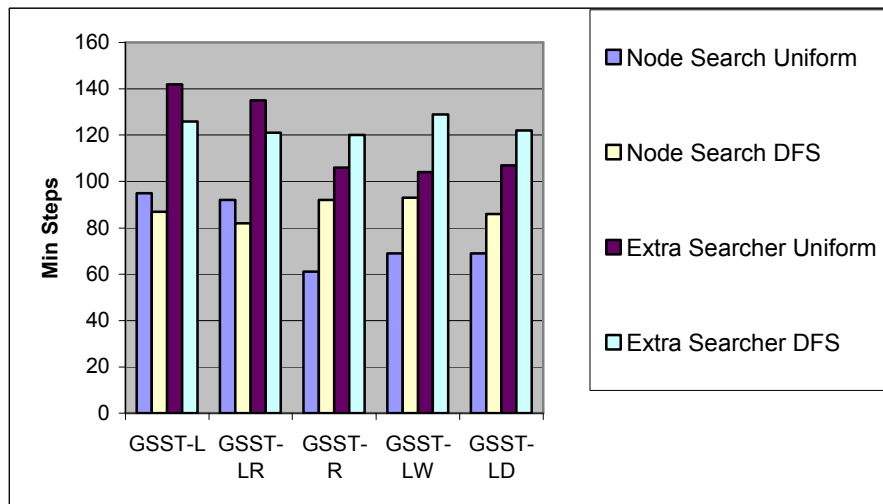
Για το 6 x 6 με $M = 2 \cdot 10^5$

Node Search	Uniform ST generation					
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)
GSST-L	7	95	118,5	0,08%	0,0005%	167,438
GSST-LR	7	92	118,3	0,06%	0,0005%	238,047
GSST-R	7	61	102,6	0,32%	0,0005%	88,953
GSST-LW	7	69	101,8	0,39%	0,0005%	119,375
GSST-LD	7	69	100,8	0,44%	0,0005%	122,344
DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time
GSST-L	7	87	112,9	0,01%	0,0005%	195,984
GSST-LR	7	82	116,4	0,04%	0,0010%	262,141
GSST-R	8	92	118,3	0,13%	0,0005%	126,031
GSST-LW	7	93	117,6	0,04%	0,0005%	174,297
GSST-LD	8	86	114,2	0,12%	0,0005%	166,625
Edge Search(Xser.)						
Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)
GSST-L	8	142	169,6	0,08%	0,0005%	167,438
GSST-LR	8	135	168,9	0,06%	0,0005%	238,047
GSST-R	8	106	146,0	0,32%	0,0005%	88,953
GSST-LW	8	104	144,8	0,39%	0,0005%	119,375
GSST-LD	8	107	148,0	0,44%	0,0010%	122,344
DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)
GSST-L	8	126	157,9	0,01%	0,0005%	195,984
GSST-LR	8	121	163,2	0,04%	0,0005%	262,141
GSST-R	9	120	150,4	0,13%	0,0005%	126,031
GSST-LW	8	129	150,0	0,04%	0,0010%	174,297
GSST-LD	9	122	153,1	0,12%	0,0005%	166,625

Πίνακας 6.63: Αποτελέσματα του 1^{ου} πειράματος στο Πλέγμα 6 X 6 ($M = 2 \cdot 10^5$).



Σχήμα 6.64: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Πείραμα 1^ο στο Πλέγμα 6 X 6 ($M = 2 \cdot 10^5$).

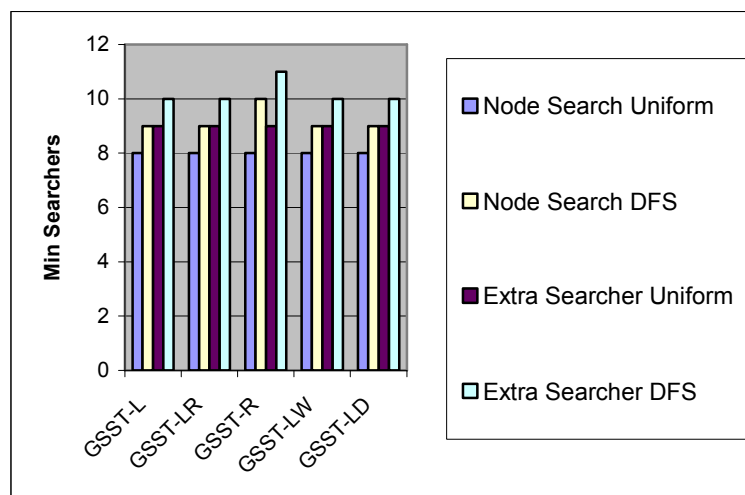


Σχήμα 6.65: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Πείραμα 1^ο στο Πλέγμα 6 X 6 ($M = 2 \cdot 10^5$).

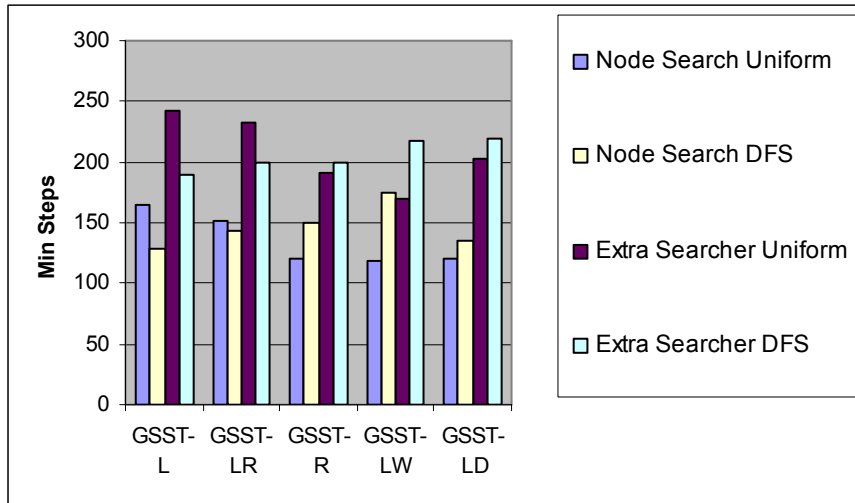
Για το 7×7 με $M = 3 \cdot 10^5$

Node Search	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	8	165	183,8	0,00333%	0,00033%	400,688	0,298
GSST-LR	8	151	173,5	0,00133%	0,00033%	596,531	0,363
GSST-R	8	121	152,3	0,01400%	0,00033%	182,641	0,490
GSST-LW	8	118	149,3	0,02333%	0,00067%	249,703	0,505
GSST-LD	8	120	151,6	0,02733%	0,00033%	262,750	0,389
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)
GSST-L	9	129	154,7	0,00100%	0,00033%	504,203	0,453
GSST-LR	9	143	143,0	0,00033%	0,00033%	689,281	0,343
GSST-R	10	150	195,2	0,16067%	0,00033%	273,563	0,865
GSST-LW	9	174	177	0,00067%	0,00033%	391,750	0,828
GSST-LD	9	135	182,7	0,00100%	0,00033%	383,016	0,505
Edge Search(Xser.)	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	9	242	256,6	0,00333%	0,00033%	400,688	3,442
GSST-LR	9	233	250,0	0,00133%	0,00033%	596,531	3,715
GSST-R	9	192	225,9	0,01400%	0,00067%	182,641	2,460
GSST-LW	9	169	213,3	0,02333%	0,00033%	249,703	2,347
GSST-LD	9	202	223,3	0,02733%	0,00033%	262,750	2,826
	DFS ST generation						
Edge Traversal							
GSST-L	10	189	218,3	0,00100%	0,00033%	504,203	3,796
GSST-LR	10	200	200,0	0,00033%	0,00033%	689,281	3,187
GSST-R	11	200	244,8	0,16067%	0,00033%	273,563	2,861
GSST-LW	10	218	221,5	0,00067%	0,00033%	391,750	2,406
GSST-LD	10	219	222,7	0,00100%	0,00033%	383,016	3,817

Πίνακας 6.66: Αποτελέσματα του 1^{ου} πειράματος στο Πλέγμα 7×7 ($M = 3 \cdot 10^5$).



Σχήμα 6.67: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Πείραμα 1^ο στο Πλέγμα 7×7 ($M = 3 \cdot 10^5$).

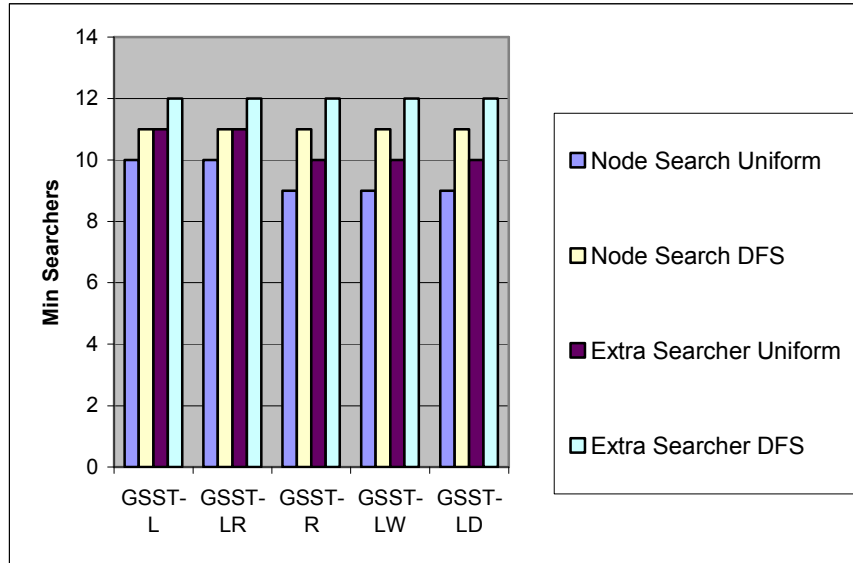


Σχήμα 6.68: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Πείραμα 1^ο στο Πλέγμα 7 X 7 ($M = 3 \cdot 10^5$).

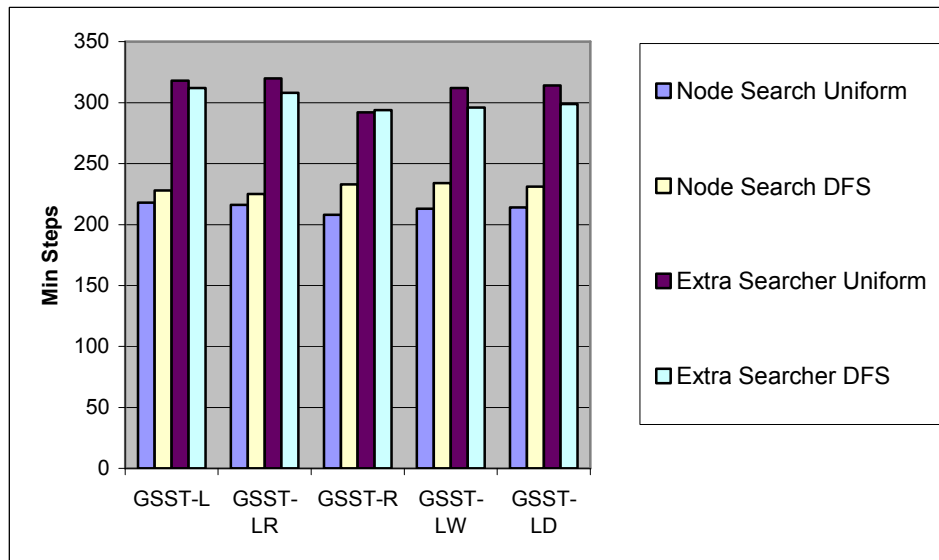
Για το 8 x 8 με $M = 3 \cdot 10^5$

Node Search		Uniform ST generation					
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	10	218	256,3	0,00667%	0,00033%	676,953	1,517
GSST-LR	10	216	271,4	0,00167%	0,00033%	390,225	1,550
GSST-R	9	208	210,7	0,00100%	0,00033%	276,813	1,500
GSST-LW	9	213	216,0	0,00067%	0,00033%	380,125	1,813
GSST-LD	9	214	218,7	0,00100%	0,00033%	407,844	1,218
		DFS ST generation					
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)
GSST-L	11	228	257,7	0,00767%	0,00033%	903,828	1,807
GSST-LR	11	225	258,3	0,00867%	0,00033%	1213,219	1,837
GSST-R	11	233	259,4	0,00600%	0,00033%	431,156	2,794
GSST-LW	11	234	266,4	0,00800%	0,00033%	625,734	2,850
GSST-LD	11	231	268,0	0,00767%	0,00033%	624,609	1,954
Edge Search(Xser.)		Uniform ST generation					
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	11	318	357,8	0,00667%	0,00033%	676,953	14,258
GSST-LR	11	320	365,6	0,00167%	0,00033%	390,225	14,137
GSST-R	10	292	307,7	0,00100%	0,00033%	276,813	7,552
GSST-LW	10	312	316,5	0,00067%	0,00033%	380,125	8,898
GSST-LD	10	314	370,3	0,00100%	0,00033%	407,844	9,458
		DFS ST generation					
Edge Traversal							
GSST-L	12	312	354,1	0,00767%	0,00033%	903,828	17,086
GSST-LR	12	308	353,9	0,00867%	0,00033%	1213,219	16,545
GSST-R	12	294	326,6	0,00600%	0,00033%	431,156	9,147
GSST-LW	12	296	335,1	0,00800%	0,00033%	625,734	9,263
GSST-LD	12	299	330,3	0,00767%	0,00067%	624,609	13,743

Πίνακας 6.69: Αποτελέσματα του 1^{ου} πειράματος στο Πλέγμα 8 X 8 ($M = 3 \cdot 10^5$).



Σχήμα 6.70: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Πείραμα 1^ο στο Πλέγμα 8 X 8 ($M = 3 \cdot 10^5$).

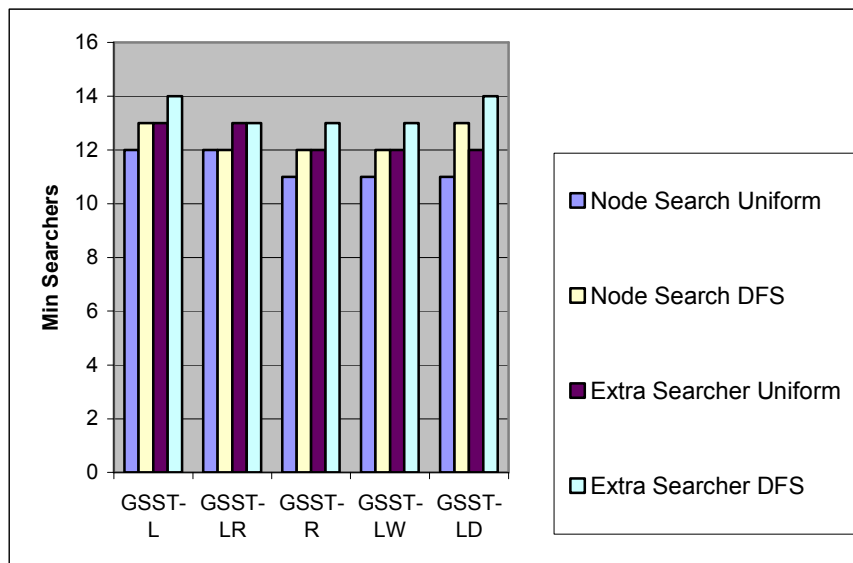


Σχήμα 6.71: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Πείραμα 1^ο στο Πλέγμα 8 X 8 ($M = 3 \cdot 10^5$).

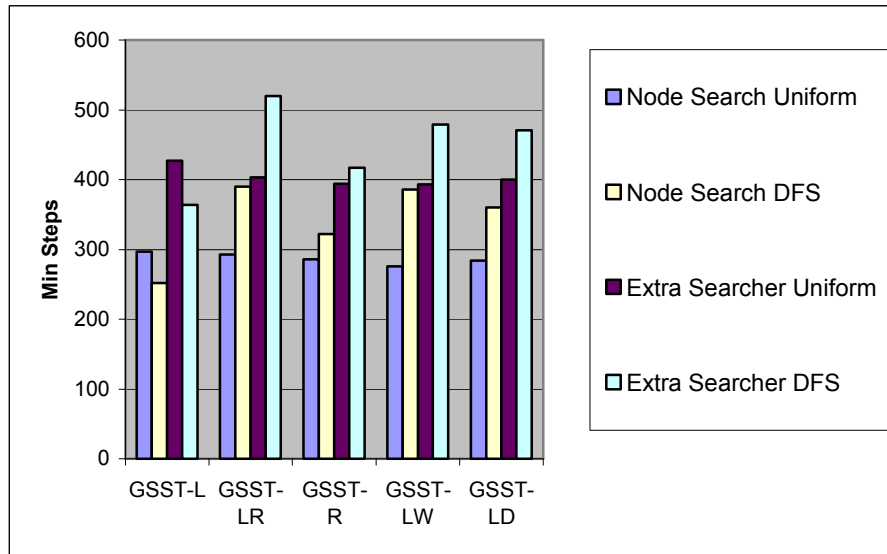
Για το 9×9 με $M = 4 \cdot 10^5$

Node Search	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	12	297	365,7	0,04725%	0,00025%	1445,922	5,324
GSST-LR	12	293	361,9	0,01375%	0,00025%	1578,352	5,287
GSST-R	11	286	320,5	0,00100%	0,00025%	533,734	6,476
GSST-LW	11	276	317,9	0,00250%	0,00025%	752,172	6,375
GSST-LD	11	284	325,0	0,00075%	0,00025%	830,297	4,000
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)
GSST-L	13	252	342,2	0,01425%	0,00025%	2094,688	6,076
GSST-LR	12	390	390,0	0,00025%	0,00025%	1815,665	5,968
GSST-R	12	322	322,0	0,00025%	0,00025%	924,844	7,984
GSST-LW	12	386	386	0,00025%	0,00025%	1332,922	8,015
GSST-LD	13	360	359,5	0,01125%	0,00025%	1353,031	5,910
Edge Search(Xser.)	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	13	427	498,6	0,04725%	0,00025%	1445,922	50,201
GSST-LR	13	403	496,1	0,01375%	0,00025%	1578,352	50,195
GSST-R	12	394	466,5	0,00100%	0,00025%	533,734	30,964
GSST-LW	12	393	449,4	0,00250%	0,00025%	752,172	30,492
GSST-LD	12	400	488,3	0,00075%	0,00025%	830,297	26,390
	DFS ST generation						
Edge Traversal							
GSST-L	14	364	471,0	0,01425%	0,00025%	2094,688	52,991
GSST-LR	13	520	520,0	0,00025%	0,00025%	1815,665	52,593
GSST-R	13	417	417,0	0,00025%	0,00025%	924,844	28,687
GSST-LW	13	479	479,0	0,00025%	0,00025%	1332,922	29,906
GSST-LD	14	471	494,8	0,01125%	0,00025%	1353,031	46,310

Πίνακας 6.72: Αποτελέσματα του 1^{ου} πειράματος στο Πλέγμα 9×9 ($M = 4 \cdot 10^5$).



Σχήμα 6.73: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Πείραμα 1^ο στο Πλέγμα 9×9 ($M = 4 \cdot 10^5$).

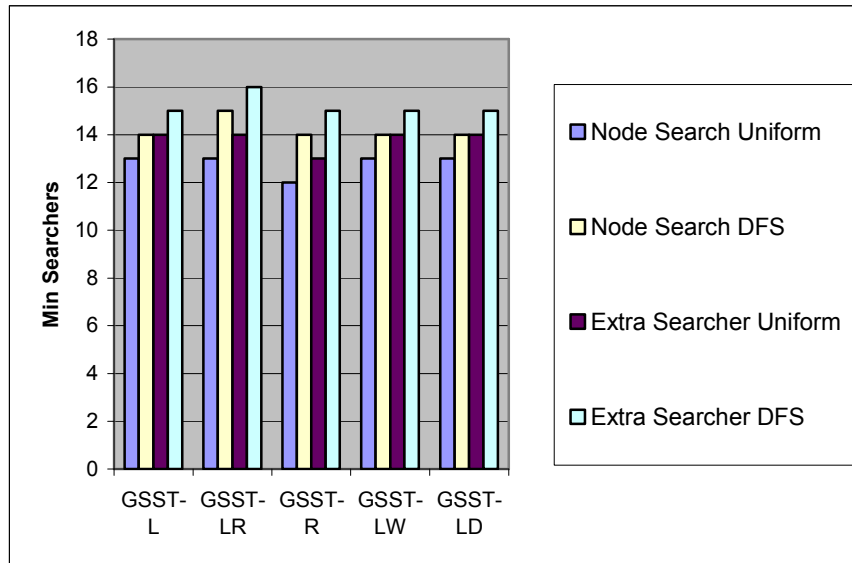


Σχήμα 6.74: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Πείραμα 1^ο στο Πλέγμα 9 X 9 ($M = 4 \cdot 10^5$).

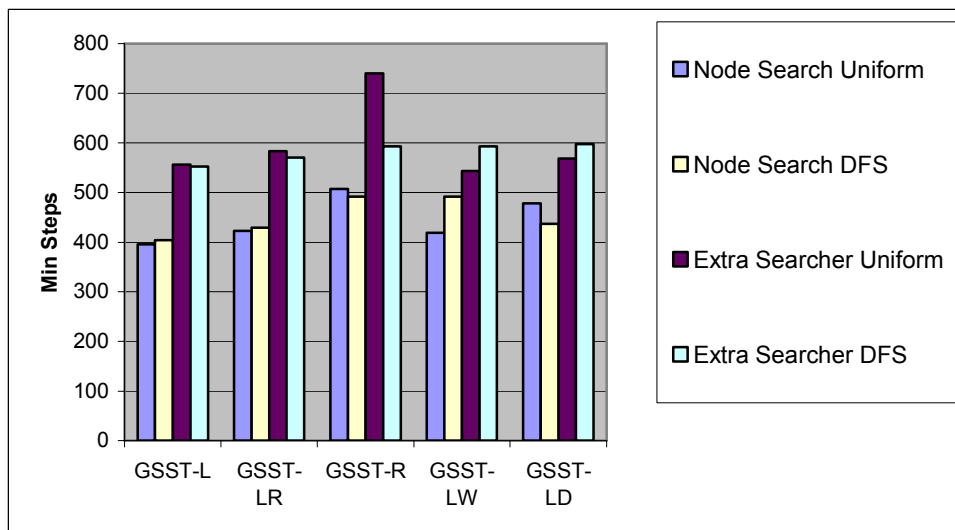
Τέλος, για το 10 x 10 με $M = 5 \cdot 10^5$

Node Search		Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)	
GSST-L	13	396	462,9	0,0066%	0,0002%	3184,719	15,943	
GSST-LR	13	423	486,5	0,0064%	0,0002%	3374,568	14,885	
GSST-R	12	507	507,0	0,0002%	0,0002%	959,422	18,718	
GSST-LW	13	419	488,5	0,0054%	0,0002%	1354,766	23,688	
GSST-LD	13	478	492,5	0,0076%	0,0002%	1529,406	11,179	
		DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)	
GSST-L	14	404	483,2	0,0062%	0,0002%	4231,344	15,559	
GSST-LR	15	429	488,6	0,0032%	0,0002%	4231,344	15,928	
GSST-R	14	492	492,0	0,0002%	0,0002%	2417,422	28,312	
GSST-LW	14	492	492	0,0002%	0,0002%	2425,172	28,125	
GSST-LD	14	437	496,7	0,0006%	0,0002%	2455,484	13,573	
Edge Search(Xser.)		Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)	
GSST-L	14	556	633,5	0,0066%	0,0002%	3184,719	151,183	
GSST-LR	14	583	655,1	0,0064%	0,0002%	3374,568	136,546	
GSST-R	13	740	740,0	0,0002%	0,0002%	959,422	118,656	
GSST-LW	14	543	650,7	0,0054%	0,0002%	1354,766	92,259	
GSST-LD	14	568	679,7	0,0076%	0,0002%	1529,406	103,041	
		DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)	
GSST-L	15	552	648,4	0,0062%	0,0002%	4231,344	151,389	
GSST-LR	16	570	653,3	0,0032%	0,0002%	4231,344	156,776	
GSST-R	15	593	593,0	0,0002%	0,0002%	2417,422	84,031	
GSST-LW	15	593	593,0	0,0002%	0,0002%	2425,172	85,109	
GSST-LD	15	597	599,7	0,0006%	0,0002%	2455,484	119,286	

Πίνακας 6.75: Αποτελέσματα του 1^{ου} πειράματος στο Πλέγμα 10 X 10 ($M = 5 \cdot 10^5$).



Σχήμα 6.76: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Πείραμα 1^ο στο Πλέγμα 10 X 10 ($M = 5 \cdot 10^5$).



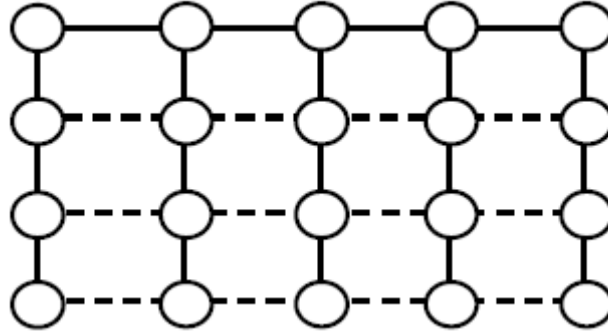
Σχήμα 6.77: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Πλέγμα 10 X 10 ($M = 5 \cdot 10^5$).

Βλέπουμε αρχικά ότι ο GSST δεν εντόπισε τον πραγματικό ελάχιστο αριθμό ερευνητών, παρά μόνο για το πλέγμα 5×5 . Στα υπόλοιπα πλέγματα βρήκε έναν (6×6 , 7×7 , 8×8) ή δύο παραπάνω (9×9 , 10×10). Ωστόσο γνωρίζοντας τον πολύ μεγάλο αριθμό γεννητορικών δένδρων που υπάρχουν για κάθε γράφο (σε σχέση και με τα M που επιλέχθηκαν), μπορούμε να πούμε ότι, για τέτοιας δυσκολίας γράφους, τα αποτελέσματα είναι πολύ καλά. Βλέπουμε επίσης πως η απόδοση του Extra Searcher δεν επηρεάζεται, όπως και περιμέναμε άλλωστε, από το μέγεθος του γράφου, καθώς προσθέτει πάντα έναν επιπλέον ερευνητή. Ο αριθμός των βημάτων στο παιχνίδι ακμών αυξάνεται κατά πολύ σε όλους τους γράφους καθώς, μέσα στην κάθε ήδη πολύπλοκη έρευνα, προστίθενται τα βήματα του edge cleaner.

Γενικά βλέπουμε ότι στα πλέγματα λειτουργούν καλύτερα οι uniform παραλλαγές του GSST και από άποψη αριθμού ερευνητών και βημάτων αλλά και από άποψη ποσοστών. Τέλος βλέπουμε πόσο μικρά γίνονται τα ποσοστά καλύτερων ερευνών όσο αυξάνονται οι διαστάσεις του εξεταζόμενου γράφου.

6.2.2 Ελαττωμένα Πλέγματα

Τα τελευταία μας πειράματα έγιναν πάνω στα λεγόμενα «ελαττωμένα πλέγματα». Αυτά παράγονται με την εξής διαδικασία. Αρχικά επιλέγουμε τις παραμέτρους J_1 και J_2 , δηλαδή το μήκος και το πλάτος του πλέγματος. Όπως και στα πλήρη πλέγματα, τοποθετούμε κόμβους στις θέσεις $(j_1, j_2) \in \{1, 2, \dots, J_1\} \times \{1, 2, \dots, J_2\}$. Στη συνέχεια ενώνουμε (με ακμές) όλους τους διπλανούς κόμβους κατά μήκος των γραμμών $(1, j_2), (J_1, j_2)$ και (j_1, J_2) (με $j_1 \in \{1, 2, \dots, J_1\}$ και $j_2 \in \{1, 2, \dots, J_2\}$) και έτσι δημιουργείται ένα δένδρο της μορφής αυτού του σχήματος 6.78, με τις συμπαγείς γραμμές μόνο. Αυτό γίνεται για να εξασφαλιστεί η συνεκτικότητα του γράφου. Τέλος, εξετάζουμε όλα τα διπλανά ζεύγη κόμβων $(i_1, i_2), (j_1, j_2)$ που δεν είναι ήδη ενωμένα, και τα συνδέουμε με μια ακμή με πιθανότητα p .



Σχήμα 6.78: Κατασκευή ενός ελαττωμένου πλέγματος. Οι συμπαγείς γραμμές αντιστοιχούν στις ακμές που υπάρχουν πάντα στον γράφο. Κάθε διακεκομμένη γραμμή γίνεται ακμή με πιθανότητα p .

Ο γράφος $\mathbf{G}$ που δημιουργείται είναι ένας υπογράφος του πλήρους πλέγματος $J_1 \times J_2$ και έχει $s_N^{imc}(\mathbf{G}) \leq \min(J_1, J_2)$ (και $s_E^{imc}(\mathbf{G}) \leq \min(J_1, J_2) + 1$). Δηλαδή το $\min(J_1, J_2)$ θέτει ένα άνω όριο (χρήσιμο για την εκτίμηση της απόδοσης του GSST) για το $s_N^{imc}(\mathbf{G})$. Φαίνεται λογικό ότι, γενικά, το $s_N^{imc}(\mathbf{G})$ (άρα και το $s_E^{imc}(\mathbf{G})$) θα είναι μια αύξουσα συνάρτηση του p και για $p = 1$ το όριο είναι αυστηρά $s_N^{imc}(\mathbf{G}) = \min(J_1, J_2)$ (και $s_E^{imc}(\mathbf{G}) = \min(J_1, J_2) + 1$). Τέλος, όσο το p πλησιάζει το 0 τόσο ο γράφος μοιάζει στη μορφή με δένδρο (έχει λιγότερους κύκλους).

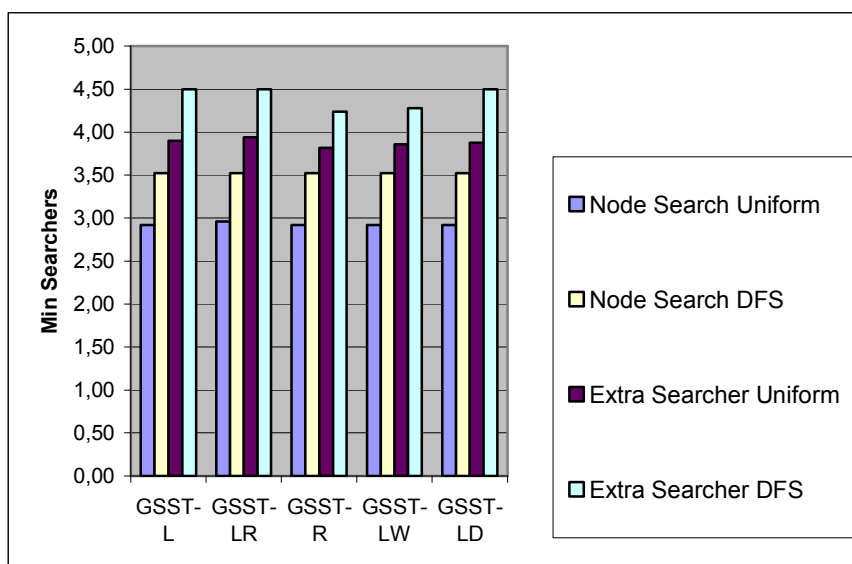
Παράγουμε 6 οικογένειες ελαττωμένων πλεγμάτων, χρησιμοποιώντας τις διαστάσεις 5×5 , 6×6 , 7×7 και τις πιθανότητες $p = 0,4375$ και $p = 0,75$. Παράγουμε 50 γράφους από κάθε οικογένεια και δοκιμάζουμε πάνω τους τις δέκα παραλλαγές του GSST και έπειτα τον Extra Searcher. Χρησιμοποιούμε $5 \cdot 10^4$ γεννητορικά δένδρα σε κάθε εφαρμογή του GSST. Οι τιμές που εμφανίζονται στους πίνακες αποτελεσμάτων είναι πλέον οι μέσοι όροι των αντίστοιχων τιμών, στα προηγούμενα πειράματα. Τέλος, λόγω του μεγάλου αριθμού ελάχιστων στρατηγικών που παράγονται σε κάποιες περιπτώσεις (π.χ. στους 5×5 με $p = 0,4375$ μπορεί να φτάσουν τις δεκάδες χιλιάδες) για να καταστεί δυνατή η εκτέλεση των πειραμάτων, επιλέγουμε τυχαία 100 από αυτές για να εξετασθούν από τον Extra Searcher. Αυτό το τέχνασμα μας δίνει μια αρκετά καλή εικόνα χωρίς να απαιτούνται πολλές ώρες για

κάθε γραμμή του πίνακα (π.χ. για 10.000 στρατηγικές σε 50 γράφους με μέσο χρόνο εκτέλεσης του Extra Searcher 0,06 sec θέλουμε $10.000 \times 50 \times 0,06 = 30000$ sec δηλαδή 8,34 ώρες για μία γραμμή του πίνακα! , χωρίς καν να υπολογίσουμε τους χρόνους εγγραφής και διαγραφής των 50 X 10.000 αρχείων στο δίσκο).

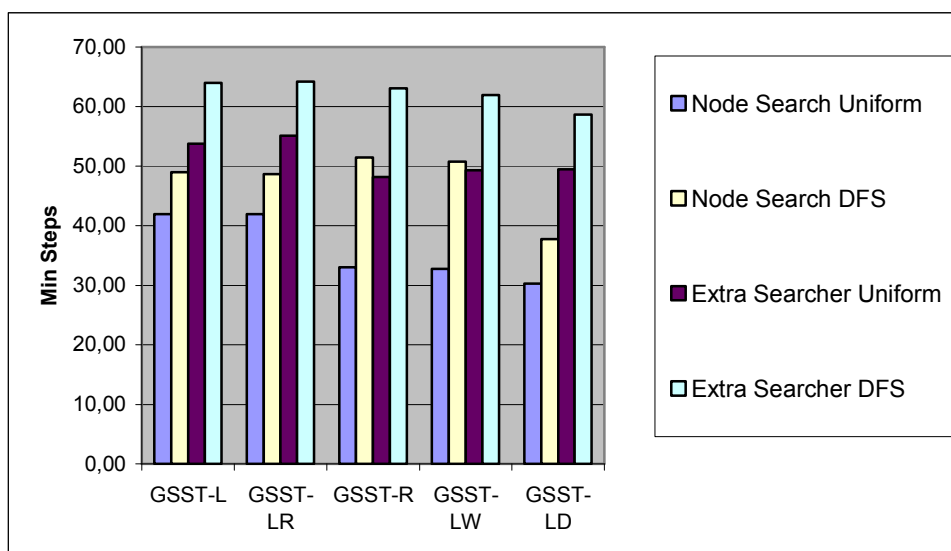
Ακολουθούν οι πίνακες αποτελεσμάτων. Υπενθυμίζουμε πως όλα τα πειράματα έγιναν για $M = 5 \cdot 10^4$ και οι τιμές είναι οι αντίστοιχοι μέσοι όροι από τους 50 γράφους της κάθε οικογένειας ελαττωμένων πλεγμάτων.

Node Search	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	2,92	41,94	50,05	10,92%	0,0107%	29,4794	0,0131
GSST-LR	2,96	41,94	51,25	11,09%	0,0118%	34,7519	0,0145
GSST-R	2,92	33,02	47,49	12,10%	0,0107%	25,1531	0,0226
GSST-LW	2,92	32,80	47,33	12,29%	0,0106%	31,0025	0,0389
GSST-LD	2,92	30,28	44,44	12,50%	0,0104%	27,0213	0,0132
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)
GSST-L	3,52	48,96	56,92	18,49%	0,0222%	46,7863	0,0312
GSST-LR	3,52	48,66	57,27	18,91%	0,0178%	49,1672	0,0332
GSST-R	3,52	51,44	59,28	19,78%	0,0064%	43,5350	0,0596
GSST-LW	3,52	50,72	59,67	20,07%	0,0060%	40,0828	0,0239
GSST-LD	3,52	37,78	54,33	20,35%	0,0102%	42,0881	0,0349
	Edge Search(Xser.)						
	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	3,90	53,80	63,32	10,92%	0,0068%	29,4794	0,0299
GSST-LR	3,94	55,16	64,63	11,09%	0,0049%	34,7519	0,0345
GSST-R	3,82	48,18	52,85	12,10%	0,0042%	25,1531	0,0342
GSST-LW	3,86	49,32	52,22	12,29%	0,0045%	31,0025	0,0499
GSST-LD	3,88	49,44	58,49	12,50%	0,0056%	27,0213	0,0291
	DFS ST generation						
Edge Traversal							
GSST-L	4,50	64,00	69,21	18,49%	0,0196%	46,7863	0,0599
GSST-LR	4,50	64,22	69,64	18,91%	0,0149%	49,1672	0,0633
GSST-R	4,24	63,08	66,45	19,78%	0,0062%	15,3438	0,0736
GSST-LW	4,28	61,94	66,47	20,07%	0,0065%	40,0828	0,0314
GSST-LD	4,50	58,70	67,06	20,35%	0,0157%	42,0881	0,0623

Πίνακας 6.79: Αποτελέσματα του 1^{ου} πειράματος στα 5 X 5 με $p = 0,4375$.



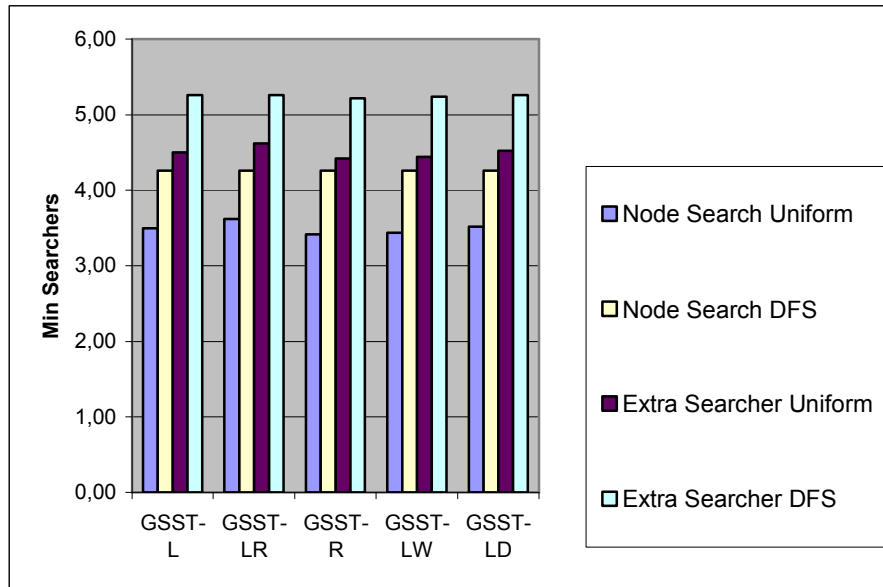
Σχήμα 6.80: Ιστογράμμο ελάχιστου αριθμού ερευνητών. Ελαττωμένα πλέγματα 5 X 5 με $p = 0,4375$.



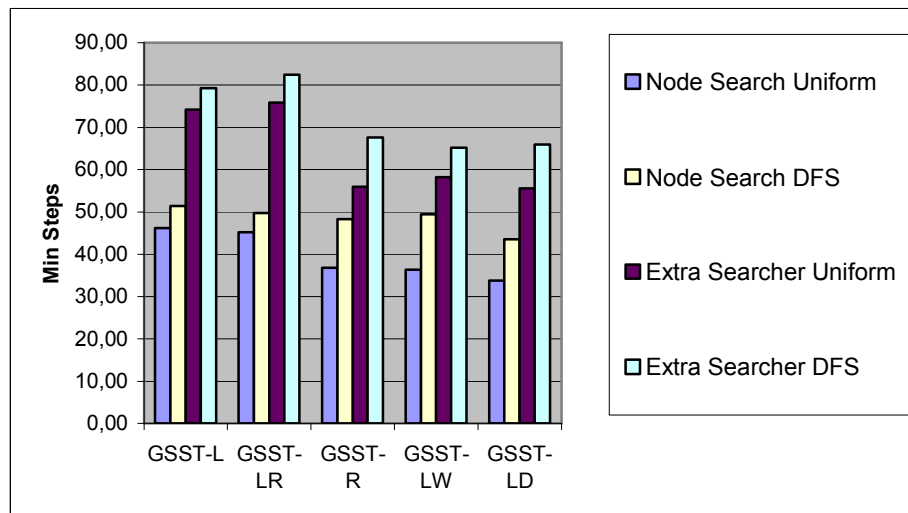
Σχήμα 6.81: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Ελαττωμένα πλέγματα 5 X 5 με $p = 0,4375$.

Node Search		Uniform ST generation					
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	3,50	46,24	54,83	0,46%	0,0033%	15,6731	0,0155
GSST-LR	3,62	45,24	56,79	0,62%	0,0028%	23,1647	0,0458
GSST-R	3,42	36,76	46,41	0,45%	0,0028%	9,6078	0,0190
GSST-LW	3,44	36,30	46,80	0,59%	0,0029%	13,4250	0,0208
GSST-LD	3,52	33,74	44,89	0,89%	0,0027%	14,2631	0,0266
		DFS ST generation					
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)
GSST-L	4,26	51,44	61,66	2,40%	0,0189%	21,5544	0,0616
GSST-LR	4,26	49,76	62,24	2,17%	0,0138%	23,1506	0,0208
GSST-R	4,26	48,34	61,16	2,53%	0,0072%	14,3338	0,0365
GSST-LW	4,26	49,42	61,19	2,56%	0,0082%	17,7781	0,0265
GSST-LD	4,26	43,58	55,32	2,66%	0,0097%	18,1553	0,0352
Edge Search(Xser.)		Uniform ST generation					
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	4,50	74,24	75,79	0,46%	0,0030%	15,6731	0,0564
GSST-LR	4,62	75,86	77,13	0,62%	0,0026%	23,1647	0,0871
GSST-R	4,42	55,96	62,58	0,45%	0,0030%	9,6078	0,0405
GSST-LW	4,44	58,22	62,78	0,59%	0,0027%	13,4250	0,0465
GSST-LD	4,52	55,54	64,76	0,89%	0,0025%	14,2631	0,0687
		DFS ST generation					
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	5,26	79,28	82,30	2,40%	0,0195%	21,5544	0,1303
GSST-LR	5,26	82,46	82,92	2,17%	0,0122%	23,1506	0,0772
GSST-R	5,22	67,60	72,10	2,53%	0,0063%	14,3338	0,0598
GSST-LW	5,24	65,18	71,87	2,56%	0,0059%	17,7781	0,0461
GSST-LD	5,26	65,90	74,04	2,66%	0,0109%	18,1553	0,0838

Πίνακας 6.82: Αποτελέσματα του 1^{ου} πειράματος στα 5 X 5 με $p = 0,75$.



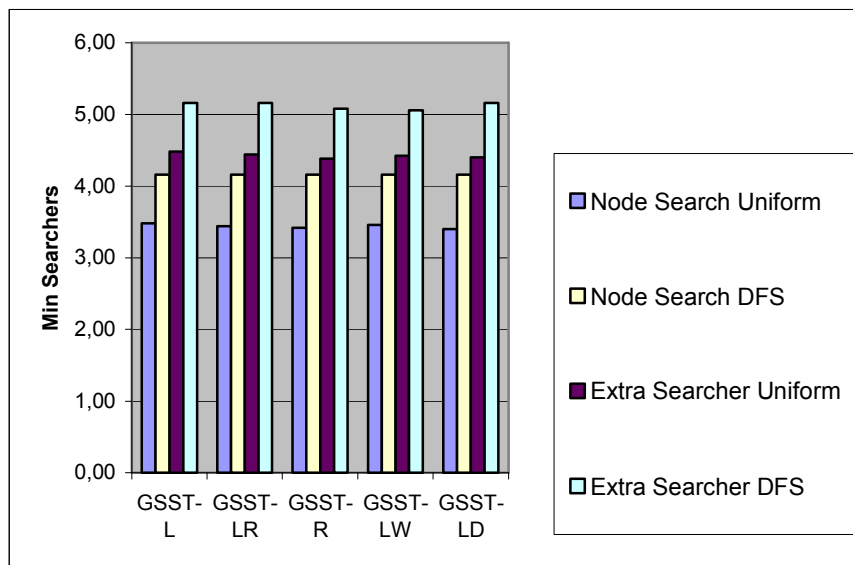
Σχήμα 6.83: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Ελαττωμένα πλέγματα 5 X 5 με $p = 0,75$.



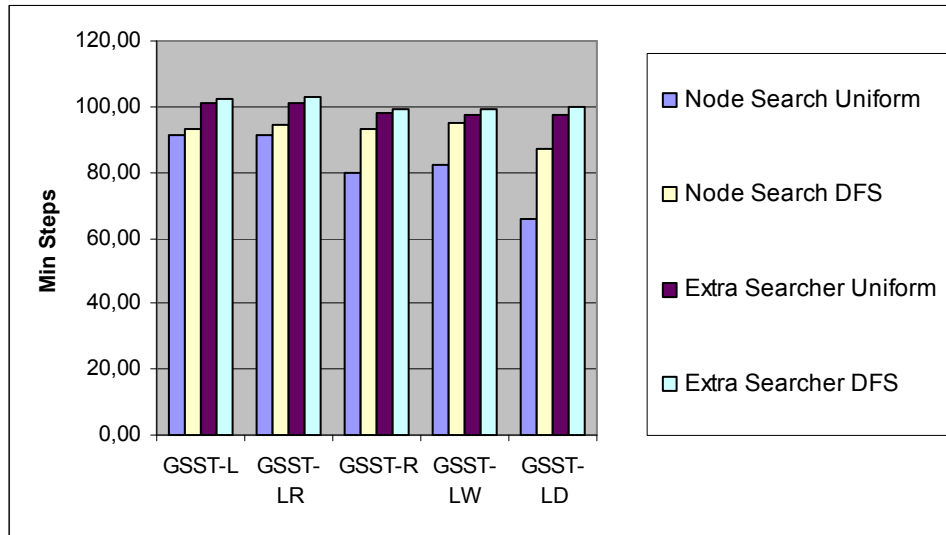
Σχήμα 6.84: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Ελαττωμένα πλέγματα 5 X 5 με $p = 0,75$.

Node Search	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	3,48	91,38	93,95	0,90%	0,0049%	22,3969	0,0271
GSST-LR	3,44	91,46	95,85	0,81%	0,0053%	30,8400	0,0346
GSST-R	3,42	79,84	80,51	1,23%	0,0054%	13,1003	0,0418
GSST-LW	3,46	82,26	83,43	1,40%	0,0055%	18,8006	0,0529
GSST-LD	3,40	66,08	73,64	1,31%	0,0022%	19,8313	0,0541
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)
GSST-L	4,16	93,30	94,88	6,15%	0,0120%	31,4944	0,0421
GSST-LR	4,16	94,18	95,48	6,04%	0,0094%	41,3963	0,0605
GSST-R	4,16	93,36	96,76	6,24%	0,0097%	24,8561	0,0685
GSST-LW	4,16	95,28	97,96	6,31%	0,0096%	27,9384	0,0707
GSST-LD	4,16	87,36	88,63	6,54%	0,0102%	29,5563	0,0747
Edge Search(Xser.)	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	4,48	101,00	108,14	0,90%	0,0042%	22,3969	0,1281
GSST-LR	4,44	101,04	107,53	0,81%	0,0036%	30,8400	0,1302
GSST-R	4,38	98,32	99,84	1,23%	0,0040%	13,1003	0,0892
GSST-LW	4,42	97,48	97,85	1,40%	0,0042%	18,8006	0,1069
GSST-LD	4,40	97,26	98,72	1,31%	0,0049%	19,8313	0,1380
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	5,16	102,54	113,17	6,15%	0,0097%	31,4944	0,1683
GSST-LR	5,16	102,82	113,61	6,04%	0,0093%	41,3963	0,1954
GSST-R	5,08	99,56	104,42	6,24%	0,0079%	24,8561	0,1094
GSST-LW	5,06	99,48	108,62	6,31%	0,0069%	27,9384	0,1064
GSST-LD	5,16	99,78	106,42	6,54%	0,0103%	29,5563	0,1773

Πίνακας 6.85: Αποτελέσματα του 1^{ου} πειράματος στα 6 X 6 με $p = 0,4375$.



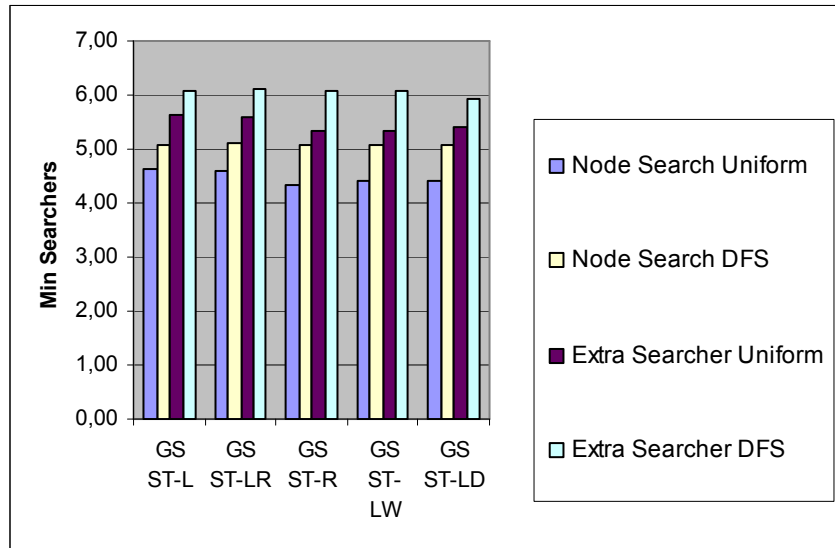
Σχήμα 6.86: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Ελαττωμένα πλέγματα 6 X 6 με $p = 0,4375$.



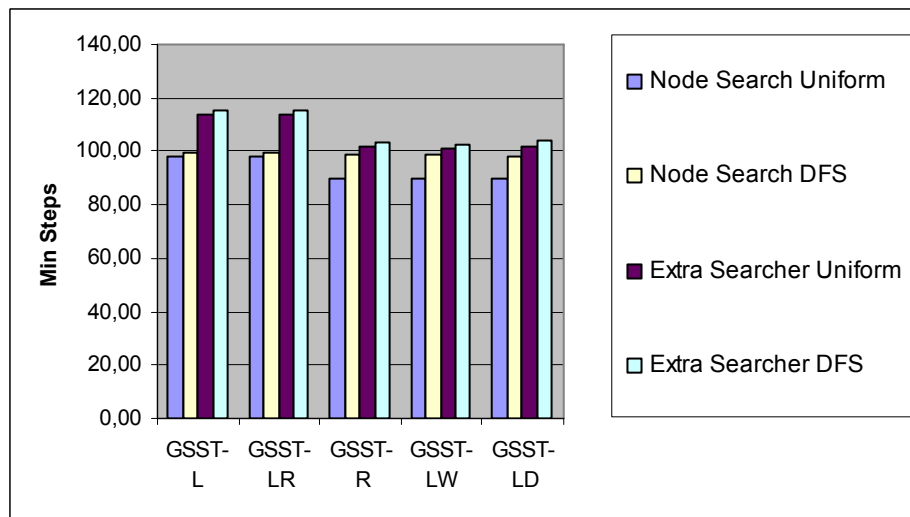
Σχήμα 6.87: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Ελαττωμένα πλέγματα 6 X 6 με $p = 0,4375$.

Node Search	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	4,62	97,98	99,12	0,30%	0,0043%	28,9344	0,0660
GSST-LR	4,58	98,42	98,69	0,20%	0,0038%	41,9819	0,0896
GSST-R	4,32	89,52	91,00	0,24%	0,0033%	14,2981	0,0641
GSST-LW	4,42	89,94	92,49	0,46%	0,0041%	20,2803	0,0703
GSST-LD	4,40	89,78	92,83	0,43%	0,0026%	21,0275	0,0658
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)
GSST-L	5,06	99,60	101,04	0,38%	0,0068%	31,4856	0,0695
GSST-LR	5,10	99,32	101,00	0,44%	0,0062%	41,7791	0,0639
GSST-R	5,06	98,46	100,07	0,38%	0,0050%	18,2806	0,0952
GSST-LW	5,08	98,68	101,40	0,40%	0,0048%	27,3288	0,1111
GSST-LD	5,06	98,40	100,63	0,42%	0,0044%	27,8684	0,1132
	Uniform ST generation						
Edge Search(Xser.)	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	5,62	113,76	132,40	0,30%	0,0023%	28,9344	0,3841
GSST-LR	5,58	113,90	131,55	0,20%	0,0021%	41,9819	0,3902
GSST-R	5,32	101,96	108,63	0,24%	0,0030%	14,2981	0,1768
GSST-LW	5,32	101,24	110,76	0,46%	0,0033%	20,2803	0,1868
GSST-LD	5,40	101,46	108,93	0,43%	0,0036%	21,0275	0,2492
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	6,06	115,64	133,60	0,38%	0,0042%	31,4856	0,3796
GSST-LR	6,10	115,08	133,46	0,44%	0,0028%	41,7791	0,3721
GSST-R	6,06	103,14	117,29	0,38%	0,0037%	18,2806	0,1922
GSST-LW	6,08	102,72	118,18	0,40%	0,0033%	27,3288	0,2110
GSST-LD	5,94	103,96	116,19	0,42%	0,0038%	27,8684	0,3550

Πίνακας 6.88: Αποτελέσματα του 1^{ου} πειράματος στα 6 X 6 με $p = 0,75$.



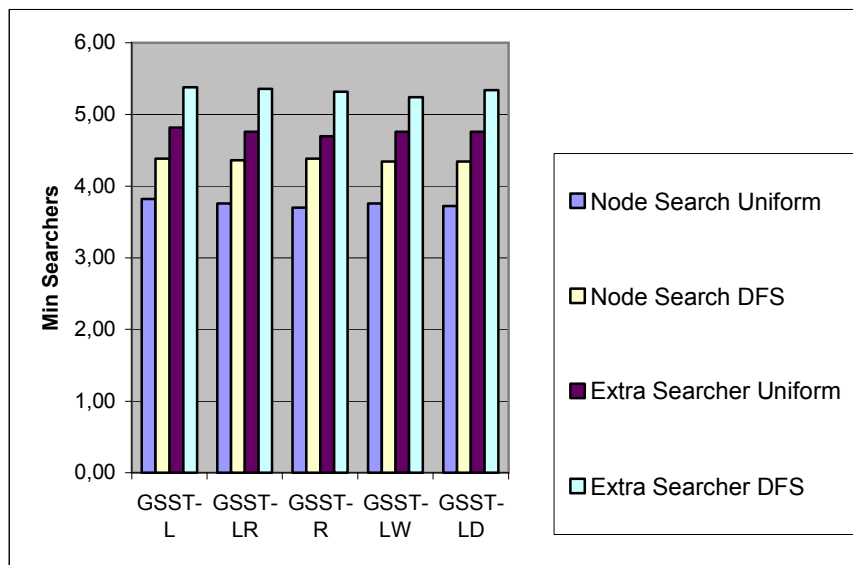
Σχήμα 6.89: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Ελαττωμένα πλέγματα 6 X 6 με $p = 0, 75$.



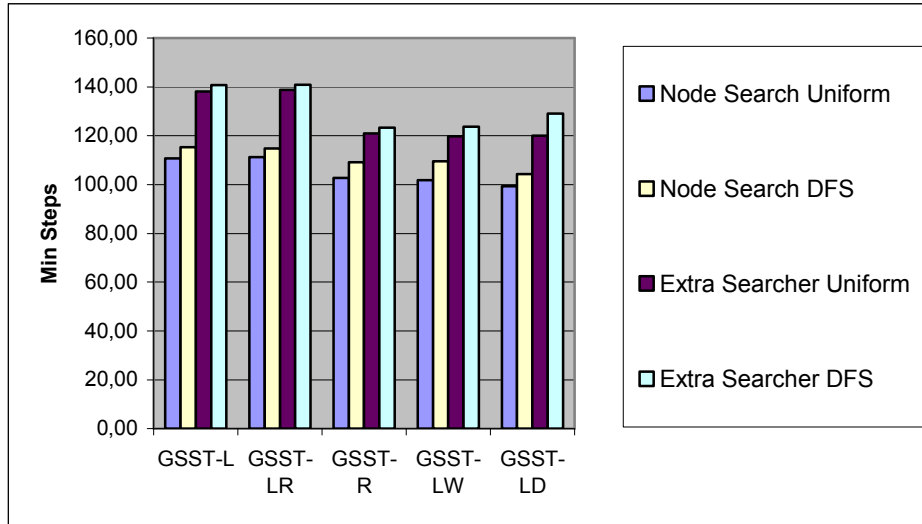
Σχήμα 6.90: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Ελαττωμένα πλέγματα 6 X 6 με $p = 0, 75$.

Node Search	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	3,82	110,64	129,36	1,33%	0,0024%	37,2269	0,0890
GSST-LR	3,76	111,14	127,12	0,98%	0,0024%	47,7703	0,0663
GSST-R	3,70	102,74	117,42	1,14%	0,0029%	18,9994	0,1418
GSST-LW	3,76	101,80	119,67	1,26%	0,0028%	26,3063	0,1423
GSST-LD	3,72	99,30	108,80	1,36%	0,0034%	27,9147	0,1269
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)
GSST-L	4,38	115,24	130,48	2,84%	0,0046%	40,6044	0,1053
GSST-LR	4,36	114,72	131,80	2,85%	0,0046%	56,2069	0,1201
GSST-R	4,38	109,10	138,32	3,25%	0,0027%	24,5091	0,2016
GSST-LW	4,34	109,52	137,48	3,41%	0,0030%	37,0228	0,2307
GSST-LD	4,34	104,28	122,37	3,58%	0,0041%	34,7109	0,1665
Edge Search(Xser.)	Uniform ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	4,82	138,16	160,61	1,33%	0,0023%	37,2269	0,4914
GSST-LR	4,76	138,76	158,41	0,98%	0,0022%	47,7703	0,4231
GSST-R	4,70	120,86	141,54	1,14%	0,0021%	18,9994	0,3139
GSST-LW	4,76	119,66	143,03	1,26%	0,0022%	26,3063	0,3099
GSST-LD	4,76	119,92	139,19	1,36%	0,0023%	27,9147	0,3988
	DFS ST generation						
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	5,38	140,78	160,20	2,84%	0,0040%	40,6044	0,5342
GSST-LR	5,36	140,94	161,35	2,85%	0,0036%	56,2069	0,5687
GSST-R	5,32	123,38	152,64	3,25%	0,0024%	24,5091	0,3199
GSST-LW	5,24	123,78	150,70	3,41%	0,0032%	37,0228	0,3430
GSST-LD	5,34	129,18	149,44	3,58%	0,0040%	34,7109	0,4562

Πίνακας 6.91: Αποτελέσματα του 1^{ου} πειράματος στα 7 X 7 με $p = 0,4375$.



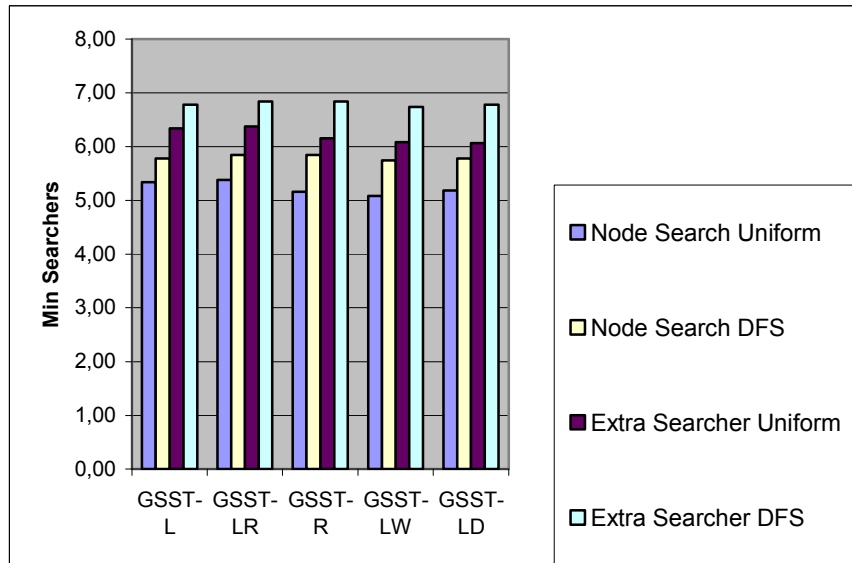
Σχήμα 6.92: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Ελαττωμένα πλέγματα 7X 7 με $p = 0,4375$.



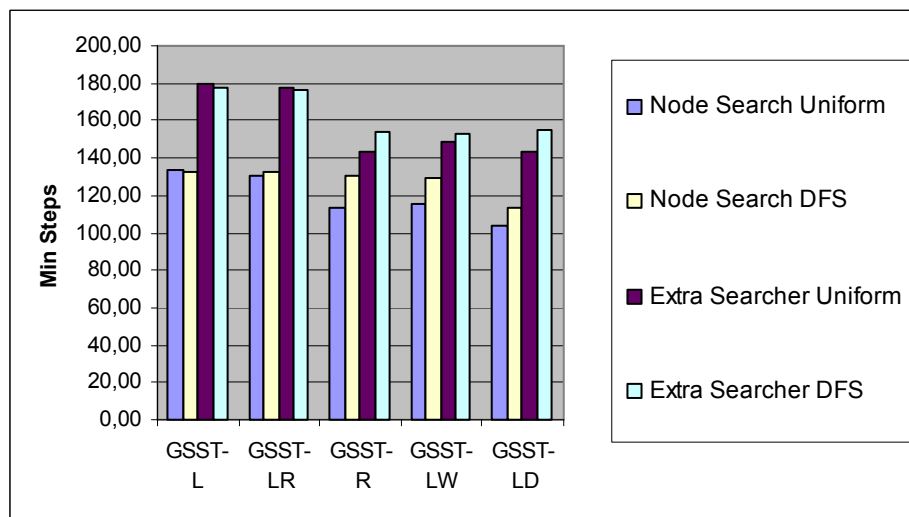
Σχήμα 6.93: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Ελαττωμένα πλέγματα 7 X 7 με $p = 0,4375$.

Node Search		Uniform ST generation					
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	5,34	134,12	154,53	0,13%	0,0022%	45,1763	0,1820
GSST-LR	5,38	130,80	153,13	0,11%	0,0023%	66,5225	0,1892
GSST-R	5,16	113,16	136,73	0,14%	0,0022%	21,5391	0,2860
GSST-LW	5,08	115,54	135,71	0,12%	0,0024%	30,9700	0,2854
GSST-LD	5,18	103,54	117,82	0,19%	0,0022%	33,7113	0,2549
		DFS ST generation					
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time	Av. time (Eclear)
GSST-L	5,78	132,72	153,22	0,12%	0,0026%	52,5806	0,2107
GSST-LR	5,84	133,12	151,37	0,13%	0,0020%	73,3619	0,2192
GSST-R	5,84	130,14	152,34	0,10%	0,0022%	27,8347	0,3444
GSST-LW	5,74	129,42	150,41	0,09%	0,0021%	42,6491	0,3567
GSST-LD	5,78	113,86	128,25	0,12%	0,0022%	43,0491	0,3172
Edge Search(Xser.)		Uniform ST generation					
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	6,34	179,52	203,99	0,13%	0,0021%	45,1763	1,3342
GSST-LR	6,38	177,30	202,47	0,11%	0,0022%	66,5225	1,3824
GSST-R	6,16	143,44	171,38	0,14%	0,0023%	21,5391	0,7810
GSST-LW	6,08	148,14	170,85	0,12%	0,0021%	30,9700	0,7881
GSST-LD	6,06	143,27	164,92	0,19%	0,0022%	33,7113	1,1325
		DFS ST generation					
Edge Traversal	Min Searchers	Min Steps	average steps	min Searchers %	min Steps %	Time (gsearch)	Av. time (Eclear)
GSST-L	6,78	177,54	201,10	0,12%	0,0022%	52,5806	1,4094
GSST-LR	6,84	176,16	199,04	0,13%	0,0021%	73,3619	1,4438
GSST-R	6,84	154,14	178,05	0,10%	0,0020%	27,8347	0,7343
GSST-LW	6,74	153,18	175,92	0,09%	0,0022%	42,6491	0,7388
GSST-LD	6,78	155,08	171,86	0,12%	0,0022%	43,0491	1,2409

Πίνακας 6.94: Αποτελέσματα του 1^{ου} πειράματος στα 7 X 7 με $p = 0,75$.



Σχήμα 6.95: Ιστόγραμμα ελάχιστου αριθμού ερευνητών. Ελαττωμένα πλέγματα 7 X 7 με $p = 0,75$.



Σχήμα 6.96: Ιστόγραμμα ελάχιστου αριθμού βημάτων. Ελαττωμένα πλέγματα 7 X 7 με $p = 0,75$.

Αρχικά παρατηρούμε τις πολύ καλές αποδόσεις της έρευνας κόμβων, δηλαδή του GSST, σε όλες τις περιπτώσεις. Παράδειγμα, πρώτη τιμή με τον GSST-L (2.92) στην οικογένεια 5 X 5 με $p = 0.4375$ υποδηλώνει ότι υπήρχαν σίγουρα γράφοι που καθαρίστηκαν με 2 μόνο ερευνητές, στο παιχνίδι ακμών. Επίσης βλέπουμε ότι, για τις τιμές των πιθανοτήτων που επιλέξαμε, οι uniform παραλλαγές εξακολουθούν να λειτουργούν καλύτερα από τις DFS. Τα καλύτερα ποσοστά ελάχιστων ερευνητών στις DFS παραλλαγές δεν πρέπει να μας παραπλανήσουν καθώς αναφέρονται σε μεγαλύτερους (συνήθως κατά 1) αριθμούς ερευνητών. Έτσι φαίνεται ότι τα ελαττωμένα πλέγματα (τουλάχιστον αυτά με $p \geq 0.4375$) υιοθετούν αυτή την ιδιότητα από τα πλήρη, άρα είναι πιο κοντά σε αυτά από ότι στα δένδρα. Τέλος, όπως λογικά περιμέναμε, τα βήματα, οι χρόνοι και οι αριθμοί των ερευνητών, αυξάνονται καθώς αυξάνονται οι διαστάσεις και οι πιθανότητες, δηλαδή η πυκνότητα των ακμών. Αντίθετα, τα ποσοστά καλύτερων αποτελεσμάτων μειώνονται.

Η μικρότερη πυκνότητα των ελαττωμένων πλεγμάτων σε σχέση με τα πλήρη μπορεί να εξηγήσει και τις μικρότερες διαφορές στον αριθμό βημάτων που απαιτούνται ανάμεσα στην έρευνα κόμβων και ακμών. Επιπλέον, μικρότερες είναι και οι διαφορές του ελάχιστου αριθμού βημάτων σε σχέση με τον μέσο αριθμό βημάτων, απ' ότι στα πλήρη πλέγματα (ως μέσες τιμές στους 50 γράφους πάντα).

Γενικά, οι χρόνοι εκτέλεσης του Eclear, είτε για καθαρισμό των κινήσεων διακτινισμού, είτε για την εφαρμογή του Extra Searcher είναι αρκετά μικροί. Παρόλα αυτά όμως βλέπουμε τον δεκαπλασιασμό τους από τις 5 X 5 οικογένειες στις 7 X 7. Σε μεμονωμένες δοκιμές που έγιναν άτυπα σε 8 X 8 ελαττωμένα πλέγματα, οι χρόνοι αυτοί γίνονται υπερδεκαπλάσιοι από αυτούς των 7 X 7, γεγονός που καθιστά τα ίδια πειράματα σε μεγαλύτερες διαστάσεις χρονικά ασύμφορα (στα πλαίσια αυτής της εργασίας). Απ' ότι φαίνεται, η πυκνότητα των ακμών δεν επηρεάζει τους χρόνους εκτέλεσης του Extra Searcher τόσο όσο η αύξηση των διαστάσεων, δηλαδή ο χρόνος εκτέλεσης του ES εξαρτάται περισσότερο από τον αριθμό κόμβων από ότι των ακμών.

Τέλος, ποιοτικά, φαίνεται ότι γενικά ο Extra Searcher, προσθέτει πάντα έναν και μοναδικό ερευνητή. Στις περισσότερες περιπτώσεις βλέπουμε ότι για τους ελάχιστους αριθμούς ερευνητών που εντοπίσαμε, ισχύει η σχέση $s_E^{imc} = s_N^{imc} + 1$ (ακόμα και κατά μέσο όρο, για 50 γράφους). Η σχέση αυτή είναι αναμενόμενη (και θεωρητικά από το Θεώρημα 3.11). Οι περιπτώσεις που δεν ισχύει εξηγούνται από το τέχνασμα που χρησιμοποιήσαμε (επιλογή 100 τυχαίων στρατηγικών). Από ότι φαίνεται το τέχνασμα εφαρμόστηκε κυρίως στην οικογένεια 5 X 5 με $p = 0.4375$, καθώς σε αυτήν παρήχθησαν συχνά περισσότερες από 100 ελάχιστες στρατηγικές, αλλά δεν προκλήθηκαν μεγάλες διαφοροποιήσεις από τις αναμενόμενες τιμές.

6.3 Συμπεράσματα πειραμάτων

Από το σύνολο των πειραμάτων μπορούμε να καταλήξουμε σε ορισμένα γενικά συμπεράσματα, που περιγράφουν συνοπτικά τη συμπεριφορά των δύο αλγορίθμων έρευνας ακμών.

Στους σχετικά μικρούς γράφους οι δύο αλγόριθμοι έχουν τις ίδιες, περίπου, καλές αποδόσεις, από άποψη χρόνου εκτέλεσης, αριθμού ερευνητών και βημάτων έρευνας. Ωστόσο για λίγες περιπτώσεις δεν ισχύει ο κανόνας $s_E^{imc} = s_N^{imc} + 1$, λόγω ιδιότυπης τοπολογίας του γράφου, και έχουμε $s_E^{imc} = s_N^{imc}$. Τότε ο Extra Searcher θα εισάγει έναν ακόμα ερευνητή, που θα είναι όμως περιττός, εκτός αν ο GSST έχει ήδη καθαρίσει το γράφο και για το παιχνίδι ακμών (πιθανότερο με την παραλλαγή GSST-R).

Καθώς αυξάνεται το μέγεθος των γράφων παρατηρούμε ότι ο Extra Nodes αρχίζει να υστερεί στον αριθμό των βημάτων, διότι αυξάνονται οι ανούσιες κινήσεις που παράγει ο GSST. Το γεγονός αυτό έχει είναι συνέπεια της μεγάλης αύξησης του αριθμού των κόμβων και ακμών στον ψευδογράφο με τους ενδιάμεσους κόμβους. Αν ο αρχικός γράφος έχει n κόμβους και e ακμές, τότε ο ψευδογράφος που δίδεται στον GSST για έρευνα θα έχει $n+e$ κόμβους και $2e$ ακμές. Επίσης, όσο μεγαλύτεροι είναι οι εξεταζόμενοι γράφοι, τόσο μειώνεται η πιθανότητα να αποτελούν εξαίρεση στον κανόνα $s_E^{imc} = s_N^{imc} + 1$, άρα να είναι περιττός ο καθαριστής ακμών.

Στους πολύ μεγάλους γράφους ο Extra Nodes γίνεται πλέον χρονικά ασύμφορος καθώς επίσης μειώνονται οι αποδόσεις και από άποψη αριθμού ερευνητών και

βημάτων έρευνας. Ο Extra Searcher, αντιθέτως, εξακολουθεί να έχει την ίδια συμπεριφορά, δηλαδή να χρησιμοποιεί πάντα έναν μόνο περισσότερο ερευνητή από την απλή έρευνα κόμβων. Επιπροσθέτως, οι χρόνοι εκτέλεσης είναι αρκετά ικανοποιητικοί (ακόμα και στο πλήρες πλέγμα 10 X 10).

Τέλος, παρατηρήσαμε ότι ο Extra Nodes δουλεύει γενικά καλύτερα με τις DFS παραλλαγές του GSST, ακόμα και στα πλέγματα. Αυτό μπορεί να εξηγηθεί από το γεγονός ότι ο ψευδογράφος ακόμα και ενός πλήρους πλέγματος είναι πολύ πιο αραιός, από άποψη πυκνότητας ακμών, από τον αρχικό. Ο Extra Searcher δείχνει να υιοθετεί πάντα την συμπεριφορά του GSST. Δηλαδή λειτουργεί γενικά καλύτερα με την DFS κατασκευή γεννητορικών δένδρων, αλλά στα πλέγματα με την uniform. Από τα πειράματα στα ελαττωμένα πλέγματα φάνηκε επίσης ότι οι χρόνοι εκτέλεσης του Extra Searcher στους μεγάλους γράφους εξαρτώνται κυρίως από τον αριθμό των κόμβων του γράφου και όχι από την πυκνότητα των ακμών του.

Συμπερασματικά, καταλήγουμε ότι και οι δύο αλγόριθμοι είναι κατάλληλοι για την επίλυση μικρών γράφων, με την εξαίρεση πάντα της περίπτωσης $s_E^{imc} = s_N^{imc}$. Για τους υπόλοιπους όμως, ο Extra Searcher υπερτερεί ξεκάθαρα για κάθε μέτρο σύγκρισης.

Συγκρίνοντας τώρα την έρευνα ακμών με την έρευνα κόμβων θα χρησιμοποιήσουμε, για τους λόγους που αναφέρθηκαν παραπάνω, τις αποδόσεις του Extra Searcher μόνο. Αν και ο εν λόγω αλγόριθμος δεν είναι ο βέλτιστος που μπορεί να κατασκευαστεί, δίνει πολύ καλές λύσεις σε σχέση με το χρόνο εκτέλεσής του. Τα αποτελέσματα των πειραμάτων επιβεβαιώνουν την θεωρία στο ότι η έρευνα ακμών είναι γενικότερη της έρευνας κόμβων (Θεώρημα 2.6). Έτσι δικαιολογείται και ο μεγαλύτερος αριθμός ερευνητών, αλλά και ο μεγαλύτερος αριθμός βημάτων στην έρευνα ακμών. Ανάλογα με την πολυπλοκότητα του γράφου και τον αριθμό των γεννητορικών δένδρων που χρησιμοποιεί ο GSST, η αύξηση των βημάτων στην έρευνα ακμών κυμαίνεται από 10% έως και 50% των βημάτων της έρευνας κόμβων.

7.Επίλογος

7.1 Απολογισμός

Οι αρχικοί στόχοι της εργασίας αυτής μπορούμε να πούμε ότι καλύφθηκαν κατά ένα μεγάλο μέρος. Πιο συγκεκριμένα:

1. Το εργαλείο γραμμής εντολών Eclear.exe (Κεφ. 5.1.2) υλοποιεί τους δυο αλγορίθμους με ταχύτητα και χρηστικότητα. Η δομή των αρχείων εισόδου και εξόδου του ακολουθεί αυτή του gsearch.exe (Κεφ. 5.1.1). Έτσι τα δυο προγράμματα μπορούν να συνεργαστούν, άρα και να συνδυαστούν οι αντίστοιχοι αλγόριθμοι.
2. Το γραφικό περιβάλλον GsearchGUI περιλαμβάνει όλες τις επιθυμητές λειτουργίες. Η χρήση του είναι απλή και άμεση και η οπτικοποίηση των δύο διαφορετικών τύπων έρευνας (κόμβων και ακμών) είναι ξεκάθαρη και κατατοπιστική.
3. Η μαθηματική τεκμηρίωση έγινε τελικά μόνο για τον Extra Searcher λόγω χρονικού περιορισμού. Η επιλογή έγινε με βάση την πιο περίπλοκη βασική του ιδέα και τις γενικά καλύτερες αποδόσεις του.
4. Η πειραματική μελέτη έγινε σε μεγάλο πλήθος γράφων και συνεπώς τα συμπεράσματά της (Κεφ. 6.3) μπορούν να θεωρηθούν αξιόπιστα.

7.2 Προτεινόμενη μελλοντική έρευνα

1. Απόδειξη της πληρότητας του extra nodes: Όπως αναφέρθηκε και στο κεφάλαιο 4.1 ο Extra Nodes, όσες φορές εφαρμόστηκε λειτούργησε ορθά. Η πληρότητά του μάλιστα φαίνεται κατά κάποιο τρόπο αναμενόμενη, όπως επίσης και η μονοτονία των στρατηγικών που παράγει. Ωστόσο δεν υπάρχει κάποια μαθηματική απόδειξη για τα παραπάνω. Μια δυσκολία για την απόδειξη αυτών των ιδιοτήτων είναι ότι, στην μετατροπή της έρευνας με τους ενδιάμεσους κόμβους σε έρευνα για τον αρχικό γράφο, μπορεί να αλλάξει σειρά καθαρισμού των αρχικών κόμβων.
2. Πιο αποδοτικοί αλγόριθμοι έρευνας ακμών: Οι αλγόριθμοι που εξετάσαμε, είτε μεταθέτοντας το πρόβλημα (Extra Nodes) είτε δρώντας συμπληρωματικά (Extra Searcher), χρησιμοποιούν και οι δύο την έρευνα κόμβων (GSST). Παρόλ' αυτά, ιδίως ο Extra Searcher, έχουν καλές αποδόσεις στους γράφους που εξετάσαμε. Θα θέλαμε όμως να βρεθούν αποδοτικότεροι, ίσως πιο εξειδικευμένοι, αλγόριθμοι για το παιχνίδι ακμών.
3. Καλύτερη επιλογή γεννητορικών δέντρων: Ο αλγόριθμος GSST εξετάζει το σύνολο των πιθανών γεννητορικών δένδρων με τυχαίο τρόπο. Θα μπορούσε όμως με κάποιο τρόπο να εξετάζει μόνο τα πιο «καλά» δένδρα; Δηλαδή, θα ήταν πολύ χρήσιμο για τον GSST να υπάρχουν κάποια κριτήρια για το ποια γεννητορικά δένδρα θα δώσουν με μεγαλύτερη πιθανότητα ελάχιστες στρατηγικές. Ίσως μια αρχική προσέγγιση θα ήταν η μελέτη και εξήγηση της καλύτερης συμπεριφοράς

των uniform δένδρων στα πλέγματα και των DFS δένδρων στους άλλους γράφους.

4. Αλγόριθμοι καθαρισμού σε ελάχιστο χρόνο και ελάχιστης συνολική απόσταση: Εκτός από τον αριθμό ερευνητών, υπάρχουν και άλλα κριτήρια για την ποιότητα μιας στρατηγικής έρευνας όπως ο αριθμός των βημάτων. Με δεδομένο αριθμό ερευνητών, μεγαλύτερο από τον ελάχιστό, πως θα μπορούσαμε να κατασκευάσουμε την στρατηγική με τα λιγότερα βήματα; Υπάρχει κάποιος αποδοτικός αλγόριθμος που να συνδυάζει τα δύο κριτήρια με οριζόμενα βάρη ακμών; Δηλαδή, σε ένα γράφο που η διάσχιση κάθε ακμής έχει ένα συγκεκριμένο κόστος (που αντιπροσωπεύει π.χ. την απόσταση στον πραγματικό χώρο), πως μπορεί να ελαχιστοποιηθεί το συνολικό κόστος για την στρατηγική καθαρισμού;.
5. Αλγόριθμοι καθαρισμού με λιγότερους περιορισμούς: Για παράδειγμα για εσωτερική μονότονη (αλλά όχι συνδεδεμένη) έρευνα, ή για εσωτερική συνδεδεμένη (αλλά όχι μονότονη έρευνα) ή ακόμα και για έρευνα χωρίς κανένα περιορισμό.
6. Στοχαστική έρευνα: Ο φυγάδας δεν κινείται με άπειρη ταχύτητα και οι κινήσεις του δεν είναι πλέον οι καλύτερες για την διαφυγή του. Κινείται τώρα τυχαία (ή ακολουθεί κάποιο μοντέλο κινήσεων). Έτσι, δεν ακολουθείται πλέον η συντηρητική προσέγγιση της παρούσας εργασίας και το παιχνίδι δεν εκφυλίζεται μονομερές. Αρκετή έρευνα έχει γύρω από αυτό το θέμα όπως στις [19,20], ωστόσο δεν υπάρχει κάποια πειραματική μελέτη των αντίστοιχων αλγορίθμων.
7. Τα παιχνίδια κόμβων και ακμών από την πλευρά του φυγάδα: Για μια στρατηγική που δεν είναι μονότονη και με μερική γνώση της θέσης των ερευνητών (π.χ. «όσφρηση» από απόσταση μικρότερη των δύο ακμών), πώς ποια είναι η καλύτερη διαδοχή κινήσεων για τον φυγάδα; Αν γνωρίζει περίπου τον τρόπο που κινούνται οι ερευνητές μπορεί (και πώς) να τους αποφύγει;

8.Βιβλιογραφία

- [1] B. Alspach, «Searching and sweeping graphs: a brief survey», *Le Matematiche*, τεύχος 59,σελ. 5-7, 2006.
- [2] L. Barriere, P. Flocchini, P. Fraigniaud, και N. Santoro, «Capture of an intruder by mobile agents», πρακτικά του 14^{ου} ετήσιου *ACM symposium on Parallel algorithms and architectures* (SPAA 2002), σελ. 200-209, 2002.
- [3] L. Barriere, P. Fraigniaud, N. Santoro, και D. M. Thilikos, «Searching is not jumping», πρακτικά του 29^{ου} *International Workshop on Graph-Theoretic Concepts in Computer Science* (WG 2003), τεύχος 2880 του Springer LNCS, σελ. 34-45, 2003.
- [4] D. Bienstock, «Graph searching, path-width, tree-width and related problems (a survey)», στο *Reliability of computer and communication networks*, τεύχος 5 του DIMACS Ser. Discrete Math. Theoret. Comput. Sci., σελ. 33-49, 1991.
- [5] D. Bienstock και P. Seymour, «Monotonicity in graph searching», *J. Algorithms*, τεύχος12, σελ. 239-245, 1991.
- [6] R. Breisch, «An intuitive approach to speleotopology», *Southwestern Cavers*, τεύχος 6, σελ. 72-78,1967.
- [7] M. Chavira. *Beyond Treewidth in Probabilistic Inference*. Ph.D. διατριβή, UCLA Dept. of Computer Science, 2007.
- [8] N. D. Dendris, L. M. Kirousis, και D. M. Thilikos, «Fugitive-search games on graphs and related parameters», *Theoret. Comput. Sci.*, τεύχος. 172, σελ. 233-254, 1997.
- [9] J. A. Ellis, I. H. Sudborough, και J. S. Turner, «The vertex separation and search number of a graph», *Inform. and Comput.*, τεύχος 113, σελ. 50-79, 1994.
- [10] F.V. Fomin, P. Heggernes και R. Mihai. «Mixed search number and linear-width of interval and split graphs». *Springer LNCS*, τεύχος 4769, σελ.304-315, 2007
- [11] F. V. Fomin, D. M. Thilikos, και I. Todinca, «Connected graph searching in outerplanar graphs», *Electronic Notes in Discrete Mathematics*, τεύχος 22,σελ. 213-216, 2005.
- [12] F. V. Fomin, D. M. Thilikos, «An annotated bibliography on guaranteed graph searching». *Theor. Comp. Sci.*, τεύχος 399, σελ.236-245, 2008.
- [13] P. Fraigniaud, F. V. Fomin, και D. M. Thilikos, «Connected branch decomposition and graph searching», πρακτικά του *SIAM Conference on Discrete Mathematics*, 2006.

- [14] P. Fraigniaud και N. Nisse, «Connected treewidth and connected graph searching», πρακτικά του 7^{ου} *Latin American Symposium on Theoretical Informatics* (LATIN 2006), τεύχος. 3887 του Springer LNCS., σελ. 479-490, 2006.
- [15] B.P. Gerkey, S. Thrun και G. Gordon. «Visibility-based pursuit-evasion with limited field of view» πρακτικά του *Natl. Conf. on Artificial Intelligence* (AAAI 2004), σελ. 20-27, 2004.
- [16] B.P. Gerkey, S. Thrun και G. Gordon. «Parallel stochastic hill-climbing with small teams». Στο *Multi-Robot Systems: From Swarms to Intelligent Automata III*, σελ. 65-77, 2005.
- [17] L. J. Guibas, J.-C. Latombe, S. M. Lavalley, D. Lin, και R. Motwani, «A visibility-based pursuit-evasion problem», *Int. J. Comput. Geom. Appl.*, τεύχος 9, σελ. 471-493, 1999.
- [18] J.M. Harris, J.L. Hirst και M.J. Mossinghoff. *Combinatorics and Graph Theory*. Springer, 2008.
- [19] G. Hollinger, Ath. Kehagias και S. Singh. «Probabilistic strategies for pursuit in cluttered environments with multiple robots». πρακτικά του ICRA 2007, 2007.
- [20] G. Hollinger, Ath. Kehagias, S. Singh, D. Ferguson, και S. Srinivasa. «Anytime Guaranteed Search using Spanning Trees». Τεχνική Έκθεση CMU-RI-TR-08-36, Robotics Institute, Carnegie Mellon University, 2008.
- [21] R. Jayakumar, K. Thulasiraman και M.N.S. Swamy. «MOD-CHAR: an implementation of Char's spanning tree enumeration algorithm and its complexity analysis». *IEEE Trans. on Circuits and Systems*, τεύχος 36, σελ. 219-228, 1989.
- [22] H Kaplan, R Shamir. «Pathwidth, bandwidth, and completion problems to proper interval graphs with small cliques». *SIAM Journal on Computing*, τεύχος 25, σελ. 540-561, 1996.
- [23] Ath. Kehagias, G. Hollinger, και S. Singh. «A Graph Search Algorithm for Indoor Pursuit / Evasion». Τεχνική Έκθεση CMU-RI-TR-08-38, Robotics Institute, Carnegie Mellon University, 2008.
- [24] N. G. Kinnersley, «The vertex separation number of a graph equals its pathwidth», *Inf. Process. Lett.*, τεύχος. 42, σελ. 345-350, 1992.
- [25] L. M. Kirousis και C. H. Papadimitriou, «Searching and pebbling», *Theoret. Comput. Sci.*, τεύχος 47, σελ. 205-218, 1986.
- [26] A. S. LaPaugh, «Recontamination does not help to search a graph», *J. Assoc. Comput. Mach.*, τεύχος 40, σελ. 224-245, 1993.

- [27] S. M. Lavalle, B. H. Simov, και G. Slutzki, «An algorithm for searching a polygonal region with a flashlight», *Int. J. Comput. Geom. Appl.*, τεύχος 12, σελ. 87-113, 2002.
- [28] S.M. LaValle. *Planning Algorithms*. Cambridge University Press, 2008.
- [29] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, και C. H. Papadimitriou, «The complexity of searching a graph», *J. Assoc. Comput. Mach.*, τεύχος. 35, σελ. 18-44. 16, 1988.
- [30] R. Nowakowski και P. Winkler, «Vertex-to-vertex pursuit in a graph», *Discrete Math.*, τεύχος 43, σελ. 235-239, 1983.
- [31] T. D. Parsons, «Pursuit-evasion in a graph», στο *Theory and applications of graphs*, Springer LNM, τεύχος 642, σελ. 426- 441, 1978.
- [32] N. N. Petrov, A problem of pursuit in the absence of information on the pursued, *Differentsialnye Uravneniya*, τεύχος 18, σελ. 1345-1352, 1982.
- [33] S. Singh και M. Wagh. «Robot path planning using intersecting convex shapes: Analysis and simulation», *IEEE Journal of Robotics and Automation*, 1987.
- [34] J. D. H. Smith, «Minimal trees of given search number», *Discrete Math.*, τεύχος 66, σελ. 191-202, 1987.
- [35] A. Takahashi, S. Ueno, και Y. Kajitani, Mixed searching and proper- path-width, *Theoret. Comput. Sci.*, τεύχος 137, σελ. 253-268, 1995
- [36] D. M. Thilikos, «Algorithms and obstructions for linear-width and related search parameters», *Discrete Appl. Math.*, τεύχος 105, σελ. 239-271, 2000.
- [37] D.B. Wilson, «Generating random spanning trees more quickly than the cover time», πρακτικά του 28^{ου} ετήσιου *ACM symposium on Theory of Computation*, σελ.296-303, 1996.
- [38] B. Yang, «Strong-mixed searching and pathwidth», *J. Comb. Optim.*, τεύχος 13, σελ. 47-59, 2007.
- [39] B. Yang, D. Dyer, and B. Alspach, «Sweeping graphs with large clique number», in Proc. του 15^{ου} *International Symposium on Algorithms and Computation (ISAAC 2004)*, Springer LNCS, τεύχος 3341, σελ. 908-920, 2004.
- [40] S Zilberstein. «Using anytime algorithms in intelligent systems». *AI magazine*, τεύχος 17, σελ. 73-83, 1996.
- [41] Ath. Kehagias, G. Hollinger, και A. Gelastopoulos, «Searching the Nodes of a Graph: Theory and Algorithms» Τεχνική Έκθεση, 2009.
- [42] G. Hollinger, «Search in the physical world», Διδακτορική διατριβή CMU-RI-TR-10-20, Robotics Institute, Carnegie Mellon University, 2010.