ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ
ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΟΜΕΑΣ ΜΑΘΗΜΑΤΙΚΩΝ

Υλοποίηση και πειραματική αξιολόγηση αλγορίθμων μονότονων απεικονίσεων γραφημάτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΥΑΓΓΕΛΙΔΑΚΗΣ ΛΕΑΝΔΡΟΣ

Επιβλέπων : Αντώνιος Συμβώνης
Καθηγητής ΕΜΠ

Αθήνα, Σεπτέμβριος 2019

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ
ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΟΜΕΑΣ ΜΑΘΗΜΑΤΙΚΩΝ

Υλοποίηση και πειραματική αξιολόγηση αλγορίθμων μονότονων απεικονίσεων γραφημάτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΥΑΓΓΕΛΙΔΑΚΗΣ ΛΕΑΝΔΡΟΣ

Επιβλέπων : Αντώνιος Συμβώνης
Καθηγητής ΕΜΠ

Τριμελής επιτροπή

……………………..	…………………….	…………………….
Αντώνιος Συμβώνης	Δημήτριος Φωτάκης	Πέτρος Στεφανέας
Καθηγητής ΕΜΠ	Επ. Καθηγητής ΕΜΠ	Επ. Καθηγητής ΕΜΠ

Αθήνα, Σεπτέμβριος 2019

Ευαγγελιδάκης Λέανδρος

Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών Ε.Μ.Π

Περίληψη

Οι μονότονες απεικονίσεις γραφημάτων είναι μια πρόσφατη σχετικά μέθοδος για την οπτικοποίηση γραφημάτων στο επίπεδο. Μια μονότονη απεικόνιση ενός γραφήματος G, είναι μια απεικόνιση του G στο πλέγμα, όπου για κάθε ζεύγος κόμβων, υπάρχει ένα μονοπάτι που είναι μονότονο ως προς κάποια κατεύθυνση. Το αντικείμενο της παρούσας διπλωματικής εργασίας, είναι η υλοποίηση και η πειραματική αξιολόγηση κάποιων επιλεγμένων αλγορίθμων που παράγουν μονότονες απεικονίσεις, για μια ειδική κατηγορία γραφημάτων, τα δένδρα.

Αρχικά, δείχνουμε ότι όλα τα δένδρα με ρίζα μπορούν να αναπαρασταθούν μέσω ισορροπημένων συμβολοσειρών από παρενθέσεις και περιγράφουμε τη διαδικασία μετατροπής τους. Βασιζόμενοι σε αυτό, περιγράφουμε δύο αλγορίθμους παραγωγής δένδρων, εξαντλητικά αλλά και τυχαία. Για την υλοποίηση, αναπτύξαμε ένα διαδραστικό εργαλείο που επιτρέπει την εφαρμογή τους, σε ένα δένδρο που επιλέγεται από το χρήστη, και την οπτικοποίηση των αποτελεσμάτων τους. Για την αξιολόγηση, παράγουμε ένα σύνολο από διακριτά δένδρα με ρίζα και εφαρμόζουμε τους αλγορίθμους σε αυτά, αποθηκεύοντας τις διαστάσεις και το εμβαδό του πλέγματος που καταλαμβάνουν σε μια βάση δεδομένων. Στη συνέχεια, αναλύουμε τα δεδομένα και τα συγκρίνουμε με τα θεωρητικά μεγέθη πλέγματος των αλγορίθμων, ώστε να υπολογίσουμε το μεγαλύτερο ποσοστό αυτών που χρησιμοποίησε ο καθένας.

Λέξεις κλειδιά

απεικόνιση γραφημάτων, μονότονες απεικονίσεις, παραγωγή δένδρων, ανάπτυξη εφαρμογών

Abstract

Monotone drawings of graphs is a relatively new method for visualizing graphs. A monotone drawing of a graph G is a straight-line drawing of G in a grid, where for every pair of nodes, there is a path between them which is monotone with respect to some direction. The subject of this diploma thesis is the implementation and experimental evaluation of certain algorithms for monotone drawing of a special class of graphs, trees.

We first show that all rooted trees can be represented with balanced strings of parenthesis and we describe the procedure through which this is done. Using this fact, we describe two algorithms for generating rooted trees exhaustively and randomly. For the implementations, we developed an interactive tool which applies the algorithms to a user selected tree, and visualizes their drawings. For the evaluation part, we generate a set of distinct rooted trees, we apply the algorithms to them and we save the grid dimensions of their drawings in a database. Next, we analyze the data and we compare them with the theoretical grid size of the algorithms, in order to compute the maximum portion each one of them used.

Keywords

Graph drawing, monotone drawings, tree generation, software development

Ευχαριστίες

Αρχικά θέλω να ευχαριστήσω τον καθηγητή μου κ. Αντώνιο Συμβώνη που μου έδωσε την ευκαιρία να ασχοληθώ με αυτό το ενδιαφέρον θέμα και για την καθοδήγησή του. Επίσης θέλω να εκφράσω τις ευχαριστίες μου στους καθηγητές τις τριμελής επιτροπής κ. Δημήτριο Φωτάκη και κ. Πέτρο Στεφανέα για το χρόνο που διέθεσαν. Τέλος θέλω να ευχαριστήσω θερμά την οικογένειά μου και τους φίλους μου, για την υποστήριξη και την υπομονή τους όλο αυτό το διάστημα.

ΠΕΡΙΕΧΟΜΕΝΑ

Εισαγωγή

1.1 Πρόλογος

Η απεικόνιση γραφημάτων, είναι ένας κλάδος των μαθηματικών και της επιστήμης των υπολογιστών, ο οποίος ασχολείται με την οπτική αναπαράσταση γραφημάτων. Ένα από τα βασικά ζητούμενα είναι η απεικόνιση των γραφημάτων με τέτοιους τρόπους, ώστε να αναδεικνύονται τα δομικά χαρακτηριστικά τους, όπως η ύπαρξη ιεραρχιών, συμμετριών, ομαδοποιήσεων κ.λπ. Αυτό συνήθως συνεπάγεται στην κατάλληλη τοποθέτηση κάθε κόμβου του γραφήματος σε σημεία στο επίπεδο, έτσι ώστε η παραγόμενη απεικόνιση να τηρεί διάφορα υψηλά κριτήρια αισθητικής, όπως η ελαχιστοποίηση τεμνόμενων ακμών, ο περιορισμός του μεγέθους που καταλαμβάνουν οι απεικονίσεις, η μη ύπαρξη κύκλων κ.λπ. Για το σκοπό αυτό γίνεται ιδιαίτερη χρήση της υπολογιστικής γεωμετρίας, ενός κλάδου της επιστήμης των υπολογιστών που ασχολείται με αλγορίθμους που βασίζονται στη γεωμετρία.

1.2 Μονότονες απεικονίσεις γραφημάτων

Οι Angelini, Colasante, Di Battista, Frati και Patrignani εισήγαγαν την ιδέα των μονότονων απεικονίσεων το 2010 ως μια νέα μέθοδο για την οπτικοποίηση γραφημάτων. Ένα μονοπάτι ονομάζεται μονότονο, όταν υπάρχει μια ευθεία τέτοια ώστε οι προβολές των κόμβων του μονοπατιού στην ευθεία αυτή, να εμφανίζονται με την ίδια σειρά όπως και στο μονοπάτι. Μια ευθύγραμμη απεικόνιση ενός γραφήματος σε πλέγμα ονομάζεται μονότονη, όταν για κάθε ζεύγος κόμβων, υπάρχει τουλάχιστον ένα μονότονο μονοπάτι που να τους συνδέει. Οι μονότονες απεικονίσεις σχετίζονται με προβλήματα της υπολογιστικής γεωμετρίας, όπως αυτό της εύρεσης μονότονων μονοπατιών μεταξύ δύο σημείων, όταν θα πρέπει να αποφεύγονται ορισμένα εμπόδια. Επιπλέον, έχει παρατηρηθεί πειραματικά , ότι η γεωδαιτική τάση, δηλαδή μονοπάτια που έχουν μια συγκεκριμένη διεύθυνση, είναι σημαντική για την κατανόηση της δομής των γραφημάτων.

Τα επόμενα χρόνια μετά τη δημοσίευσή των Angelini et al, οι μονότονες απεικονίσεις έγιναν αντικείμενο μελέτης από αρκετούς επιστήμονες και παρουσιάστηκαν αρκετά νέα ευρήματα. Τα βασικά ερευνητικά ζητήματα είναι η ταυτοποίηση των κατηγοριών των γραφημάτων που δέχονται μονότονη απεικόνιση καθώς και η σχεδίαση αλγορίθμων οπτικοποίησής τους, που να απαιτούν όσο το δυνατό μικρότερο μέγεθος πλέγματος γίνεται. Μέχρι στιγμής, έχει αποδειχθεί ότι όλα τα επίπεδα συνεκτικά γραφήματα αλλά και τα εξωεπίπεδα συνεκτικά γραφήματα μπορούν να δεχθούν μονότονη απεικόνιση σε πολυωνυμικό μέγεθος πλέγματος. Επίσης, μονότονη απεικόνιση μπορούν να δεχθούν και τα επίπεδα 2-συνεκτικά γραφήματα αλλά σε εκθετικό μέγεθος πλέγματος.

1.3 Σκοπός της διπλωματικής

Στις μονότονες απεικονίσεις, έχει δοθεί ιδιαίτερη έμφαση σε μια ειδική κατηγορία γραφημάτων, τα δένδρα, καθώς μια μονότονη απεικόνιση ενός γραφήματος μπορεί να παραχθεί από μια μονότονη απεικόνιση ενός σκελετικού δένδρου του. Στην παρούσα διπλωματική, θα εξετάσουμε κάποιους από τους βασικότερους αλγορίθμους που έχουν δημοσιευτεί έως τώρα, για μονότονες απεικονίσεις δένδρων. Το βασικό γνώρισμα των αλγορίθμων είναι το μέγεθος του πλέγματος που καταλαμβάνουν οι απεικονίσεις τους. Το μέγεθος αυτό αποδεικνύεται θεωρητικά και αντιστοιχεί στη χειρότερη περίπτωση, δηλαδή δεν σημαίνει ότι κάθε απεικόνιση θα καταλαμβάνει αυτό το μέγεθος. Ο σκοπός της διπλωματικής είναι να τους υλοποιήσουμε ώστε να δούμε οπτικά τις απεικονίσεις παράγουν και επίσης να ελέγξουμε πειραματικά, τον βαθμό στον οποίο οι αλγόριθμοι αξιοποιούν το θεωρητικό μέγεθος πλέγματος.

1.3 Οργάνωση της διπλωματικής

Στο Κεφάλαιο 2 περιγράφουμε συνοπτικά τους αλγορίθμους μονότονων απεικονίσεων που θα μελετήσουμε, αναφέροντας κάποιες βασικές μαθηματικές έννοιες που χρησιμοποιούν.

Στο Κεφάλαιο 3 εξετάζουμε το πρόβλημα της αυτόματης παραγωγής δένδρων. Περιγράφουμε μία μέθοδο εξαντλητικής αλλά και τυχαίας παραγωγής δένδρων με n κόμβους, τα οποία θα χρησιμοποιήσουμε κατά την πειραματική αξιολόγηση.

Στο Κεφάλαιο 4 γίνεται η υλοποίηση ενός διαδραστικού εργαλείου εφαρμογής των αλγορίθμων, σε δένδρα που εισάγονται από τον χρήστη, και η οπτικοποίηση των αποτελεσμάτων τους.

Στο Κεφάλαιο 5 γίνεται η πειραματική μελέτη των αλγορίθμων. Αρχικά εφαρμόζουμε τους αλγορίθμους, σε δένδρα που παράγουμε με τους τρόπους του τρίτου κεφαλαίου, και αποθηκεύουμε τα δεδομένα τους σε βάση δεδομένων. Έπειτα προβαίνουμε στην ανάλυση αυτών και στην παρουσίασή τους.

Στο Κεφάλαιο 6 διατυπώνονται τα συμπεράσματα της παρούσας διπλωματικής.

1.4 Τεχνολογίες που χρησιμοποιήθηκαν

Στα επόμενα, θα παρουσιάσουμε συνοπτικά τις τεχνολογίες που χρησιμοποιήθηκαν στην παρούσα διπλωματική εργασία, τόσο για την υλοποίηση όσο και για την πειραματική ανάλυση των αλγορίθμων.

1.4.1 Γλώσσες προγραμματισμού και βιβλιοθήκες

Python

Η Python είναι μια ευρέως διαδεδομένη γλώσσα προγραμματισμού, γενικού σκοπού, η οποία χαρακτηρίζεται από την απλότητά της και από το γεγονός ότι, συγκριτικά με άλλες γνωστές γλώσσες προγραμματισμού, μπορεί να εκτελέσει σύνθετες λειτουργίες απαιτώντας ελάχιστες γραμμές κώδικα. Η Python υποστηρίζει όλες τις σύγχρονες απαιτήσεις ενός προγραμματιστή και χρησιμοποιείται για πολλές εφαρμογές, όπως η κατασκευή διαδικτυακών εφαρμογών, η ανάλυση και επεξεργασία μεγάλων βάσεων δεδομένων, η παραγωγή διεπιφανειών χρήστη κ.λπ. και υποστηρίζεται από όλα τα λειτουργικά συστήματα. (Windows, Linux, Mac κ.λπ.).

Σημαντικές βιβλιοθήκες τις Python:

Matplotlib
Είναι μια εξαιρετικά δημοφιλής βιβλιοθήκη της Python η οποία χρησιμοποιείται για την παραγωγή διαφόρων διαγραμμάτων, όπως διαγράμματα διασποράς, ιστοριογράμματα, ραβδογράμματα κ.λπ. Έχει πολλές δυνατότητες παραμετροποίησης και είναι εύκολη στη χρήση.

NetworkX
Χρησιμοποιείται για την μοντελοποίηση γραφημάτων και σύνθετων δικτύων. Παρέχει πολλούς αλγορίθμους ανάλυσης γραφημάτων όπως αυτοί της διάσχισης δένδρων, της εύρεσης ανεξαρτήτων συνόλων και κλικών, της συσταδοποίησης καθώς και αλγορίθμους δημιουργίας δένδρων διαφόρων τύπων. Επίσης, επιτρέπει την οπτικοποίηση των γραφημάτων χρησιμοποιώντας διάφορες τεχνικές και χρησιμοποιεί ως βάση τη βιβλιοθήκη Matplotlib.

HTML5

Η γλώσσα HTML(Hypertext Markup Language) είναι η γλώσσα σήμανσης που χρησιμοποιείται για να παράξει ιστοσελίδες. Κάθε ιστοσελίδα του παγκοσμίου ιστού αναφέρεται σε ένα αρχείο με περιεχόμενο HTML το οποίο βρίσκεται σε έναν εξυπηρετητή, ο οποίος εμφανίζει το περιεχόμενό του σε κάποιον φυλλομετρητή. Έχει ιδιαίτερα απλό συντακτικό και αρκετές βασικές δυνατότητες όπως η μορφοποίηση κειμένου, η εισαγωγή εικόνων και πολυμέσων, η χρήση διαδραστικών φορμών κ.λπ.

CSS3

Παράλληλα με την HTML, πλέον, η συνήθης πρακτική είναι η χρήση της γλώσσας σήμανσης CSS(Cascading style sheets), η οποία χρησιμοποιείται για τη μορφοποίηση των στοιχείων της HTML, δηλαδή ασχολείται με την εξωτερική εμφάνιση ενός ιστοτόπου. Επιτρέπει την επεξεργασία των ιδιοτήτων της γραμματοσειράς , των χρωμάτων , των γραφικών, της στοίχισης και άλλων χαρακτηριστικών μιας ιστοσελίδας.

Bootstrap

Το Bootstrap είναι ένα πακέτο διαδικτυακών εργαλείων που χρησιμοποιείται συμπληρωματικά για την κατασκευή ιστοσελίδων. Βασίζεται στις τεχνολογίες HTML,CSS και JavaScript για περιέχει πολλές έτοιμες επιλογές για την προσαρμογή των στοιχείων του περιβάλλοντος μιας ιστοσελίδας. Το βασικό γνώρισμα του Bootstrap είναι ότι επιτρέπει την παραγωγή ιστοσελίδων που μεταβάλλουν δυναμικά το περιεχόμενό τους έτσι ώστε να προσαρμόζονται στο εκάστοτε μέγεθος οθόνης. Δηλαδή ανάλογα αν χρησιμοποιείται μια μεγάλη οθόνη ενός υπολογιστή ή μια μικρή οθόνη ενός κινητού τηλεφώνου, η ιστοσελίδα μπορεί αυτόματα να τροποποιεί την ανάλυση των στοιχείων HTML που την απαρτίζουν, έτσι ώστε ο χρήστης να μην χρειάζεται να κάνει μεγέθυνση ή σμίκρυνση χειροκίνητα.

JavaScript

Πρόκειται για την πιο διαδεδομένη γλώσσα προγραμματισμού ηλεκτρονικών ιστοσελίδων και χρησιμοποιείται για την εκτέλεση κώδικα σε αυτές. Δεν έχει κάποια σχέση με τη γλώσσα Java, όπως μπορεί να παρερμηνευθεί λόγω του ονόματός της, και μπορεί να χρησιμοποιηθεί για αντικειμενοστραφές ,προστακτικό αλλά και συναρτησιακό προγραμματισμό. Είναι αρκετά γρήγορη στη χρήση και ενσωματώνεται στον κώδικα HTML μιας ιστοσελίδας, δηλαδή εκτελείται εξολοκλήρου από την πλευρά του χρήστη, παρόλο που έχουν αναπτυχθεί και βιβλιοθήκες που επιτρέπουν και την εκτέλεση κώδικα και από την πλευρά του εξυπηρετητή.Η JavaScript χρησιμοποιείται κυρίως σε ιστοσελίδες στο διαδίκτυο, αλλά και σε κινητά τηλέφωνα, tablets κ.λπ.

Chart.js

Η βιβλιοθήκη της JavaScript Chart.js είναι υπεύθυνη για την παραγωγή διαδραστικών διαγραμμάτων διαφόρων τύπων. Είναι ανοικτού κώδικα, εύκολη στη χρήση και παράγει διαγράμματα υψηλής αισθητικής με πολλές δυνατότητες τροποποίησης. Βασίζεται στις τεχνολογίες της HTML5 και της JavaScript και έτσι προσφέρει ποιοτικά διαγράμματα σε όλους τους σύγχρονους περιηγητές ιστού και προσφέρει responsive design, δηλαδή τα διαγράμματα προσαρμόζονται ανάλογα με τη διάσταση και τον προσανατολισμό της οθόνης.

SQL

Η γλώσσα SQL(Structured Query Language) έχει σχεδιαστεί για την διαχείριση βάσεων δεδομένων και βασίζεται στη σχεσιακή άλγεβρα. Επιτρέπει την αποθήκευση, τροποποίηση και άντληση πληροφοριών και γενικά παρέχει όλα τα εργαλεία για ολοκληρωμένη διαχείριση μιας βάσης δεδομένων.

PHP

Η γλώσσα PHP(Hypertext Preprocessor) αποτελεί τη βασική γλώσσα προγραμματισμού στις ιστοσελίδες με δυναμικό περιεχόμενο. Εκτελείται στην πλευρά ενός εξυπηρετητή (π.χ. Apache), είναι γενικού σκοπού γλώσσα και χρησιμοποιείται ενταγμένη σε κώδικα HTML ή και αυτόνομα. Επίσης, παρέχει τη δυνατότητα σύνδεσης με πολλές δημοφιλείς βάσεις δεδομένων όπως MySql, PostgreSQL,SQLite κ.λπ. καθώς και με FTP εξυπηρετητές. Είναι αρκετά γρήγορη, σχετικά εύκολη στη χρήση και πλέον χρησιμοποιείται από την πλειοψηφία των διασημότερων ιστοσελίδων παγκοσμίως.

1.4.2 Λογισμικό

XAMPP

Αποτελεί μια συλλογή προγραμμάτων ,ελεύθερου λογισμικού, το οποίο παρέχει μια ολοκληρωμένη λύση για την ανάπτυξη και τη δοκιμή ιστοσελίδων. Παρέχει έναν εξυπηρετητή HTTP Apache, τη σχεσιακή βάση δεδομένων MariaDB (επέκταση της MySql) καθώς και τη γλώσσα PHP. Λειτουργεί σε όλα τα λειτουργικά συστήματα και περιέχει πολλές χρήσιμες εφαρμογές γραμμένες σε PHP και Perl όπως οι OpenSSL, Joomla, Wordpress, phpmyadmin κ.λπ.

Sublime text

Είναι μια δημοφιλής διεπαφή προγραμματισμού εφαρμογών, η οποία υποστηρίζεται από όλα τα λειτουργικά συστήματα. Υποστηρίζει τις περισσότερες από τις υπάρχουσες γλώσσες προγραμματισμού καθώς και γλώσσες σήμανσης. Έχει απλό περιβάλλον χρήσης και είναι ιδιαίτερα γρήγορο, με αρκετές δυνατότητες επεκτασιμότητας.

Οι αλγόριθμοι που θα χρησιμοποιήσουμε, υλοποιήθηκαν εξολοκλήρου μέσω της γλώσσας Python3.6. Έγινε ιδιαίτερη χρήση των βιβλιοθηκών της , Matplotlib, Numpy, Pandas, NetworkX και PyMySQL.

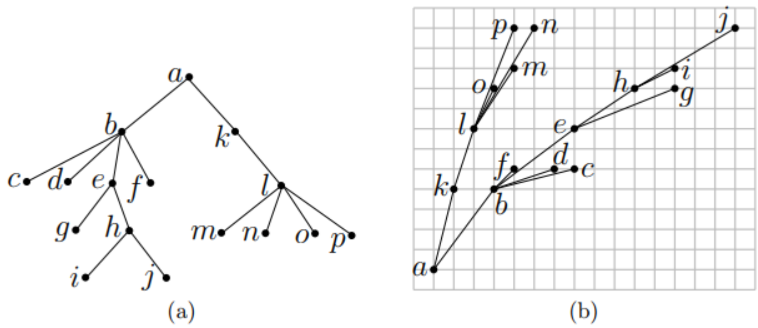
Το εργαλείου οπτικοποίησης αναπτύχθηκε ως διαδικτυακή εφαρμογή, μέσω HTML5,CSS3,Bootstrap,JavaScript και PHP5. Για τον ρόλο του εξυπηρετητή της σελίδας, χρησιμοποιήθηκε το λογισμικό πακέτο XAMPP και για τα πειραματικά δεδομένα καταχωρήθηκαν σε σχεσιακή βάση δεδομένων τύπου MySQL (MariaDB).

Όλο το προγραμματιστικό μέρος έγινε χρησιμοποιώντας το προγραμματιστικό περιβάλλον Sublime Text σε λειτουργικό σύστημα Windows 10.

2. Αλγόριθμοι παραγωγής μονότονων απεικονίσεων δένδρων

2.1 Ιστορική αναδρομή

Οι Angelini et al,το 2010 [1], απέδειξαν ότι όλα τα δένδρα έχουν μονότονη απεικόνιση και περιέγραψαν δύο αλγορίθμους για την κατασκευή μονότονων απεικονίσεων δένδρων χρησιμοποιώντας πλέγμα μεγέθους $O(n^{1.6})\ x\ O(n^{1.6})$ και $O(n)\ x\ O(n^2)$. Οι αλγόριθμοί τους, βασίζονται στη θεωρία αριθμών και συγκεκριμένα στο δένδρο Stern-Brocot, ένα άπειρο δυαδικό δένδρο που βρίσκεται σε αντιστοίχιση με όλους τους θετικούς ρητούς αριθμούς. Οι Kindermann et al. [2] το 2014, χρησιμοποιώντας και αυτοί στοιχεία της θεωρίας αριθμών, τις ακολουθίες Farey, περιέγραψαν αλγόριθμο που καταλαμβάνει πλέγμα μεγέθους $O(n^{1.5})\ x\ O(n^{1.5})$ και παράγει κυρτή απεικόνιση. Έπειτα, οι X.He και D.He , χρησιμοποιώντας και αυτοί τις ακολουθίες Farey, μείωσαν το μέγεθος του πλέγματος σε $O(n^{1.205})\ x\ O(n^{1.205})$ και μετέπειτα σε $O(n \log n)\ x\ O(n \log n)$, αλλά γενικά ακολουθώντας έναν διαφορετικό τρόπο προσέγγισης, συγκριτικά με τους προηγούμενους αλγορίθμους. Στη συνέχεια, οι ίδιοι, πέτυχαν μια ασυμπτωτικά βέλτιστη λύση επιτυγχάνοντας μέγεθος πλέγματος $12n\ x 12n$ [3] το οποίο ήταν και το πρώτο αποτέλεσμα που απαιτεί συνολικό μέγεθος της τάξης του n^2.
Πρόσφατα, οι Οικονόμου-Συμβώνης, περιέγραψαν τρείς νέους αλγορίθμους οι οποίοι σε αντίθεση με όλες τις προηγούμενες προσεγγίσεις, δεν κάνουν χρήση της θεωρίας αριθμών. Αρχικά παρουσίασαν έναν αλγόριθμο που χρησιμοποιεί πλέγμα μεγέθους $n\ x\ n$ [4] για την απεικόνιση δένδρων με ρίζα και βασίζεται στη γεωμετρία. Έπειτα, παρουσίασαν δύο αλγορίθμους στους οποίους η ρίζα μπορεί να μεταβληθεί. Επιλέγοντας προσεκτικά μια νέα ρίζα για το δένδρο και χρησιμοποιώντας δύο τεταρτημόρια για την απεικόνισή του, πέτυχαν μέγεθος πλέγματος $n\ x \frac{n}{2}$ [5]. Τέλος, συνδυάζοντας τους πρώτους δύο αλγορίθμους, και χρησιμοποιώντας τέσσερα τεταρτημόρια, περιέγραψαν αλγόριθμο που χρησιμοποιεί μέγεθος πλέγματος $\left\lfloor\frac{3}{4}(n+1)\right\rfloor \chi \left\lfloor\frac{3}{4}(n+1)\right\rfloor$ [5]. Ο τελευταίος αλγόριθμος απαιτεί το μικρότερο μέγεθος πλέγματος από όλους τους υπάρχοντες αλγορίθμους για μονότονες απεικονίσεις δένδρων μέχρι σήμερα.



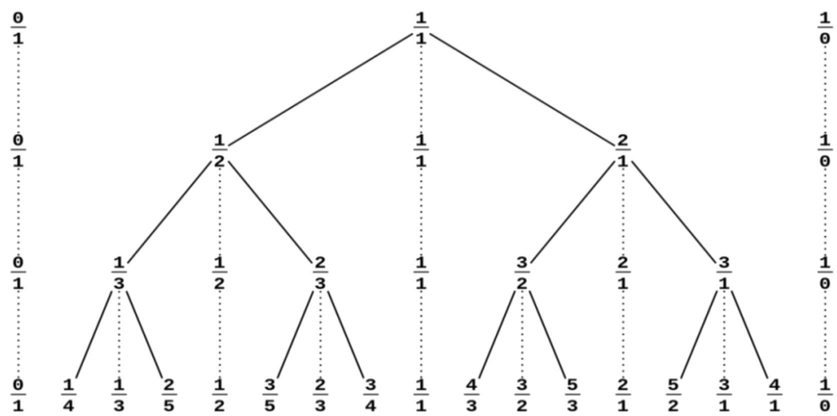
Εικόνα 2.1 Μονότονη απεικόνιση ενός δένδρου. Αλγόριθμος Angelini et al. [1]

2.2 Στοιχεία από θεωρία αριθμών

Ένα βασικό κομμάτι των περισσότερων αλγορίθμων είναι η χρήση της θεωρίας αριθμών για την παραγωγή συνόλων από διακριτά ζεύγη ακεραίων, με τη μορφή κλασμάτων. Σε αυτή την παράγραφο θα εξετάσουμε δύο πολύ βασικές έννοιες που χρησιμοποιούνται για το σκοπό αυτό και τις οποίες χρησιμοποιούν οι περισσότεροι αλγόριθμοι.

2.2.2 Δένδρο Stern-Brocot

Το *δένδρο Stern-Brocot*, είναι ένα άπειρο δυαδικό δένδρο αναζήτησης του οποίου οι κόμβοι έχουν ένα προς ένα αντιστοίχιση με το σύνολο των θετικών ρητών αριθμών. Κάθε κόμβος παράγεται αθροίζοντας κατά μέλη τα κλάσματα των πατέρων του. Όταν δεν υπάρχει στο δένδρο ένας από τους δύο πατέρες, θεωρούμε ως αριστερό ή δεξιό πατέρα, τα κλάσματα $\frac{0}{1}$ και $\frac{1}{0}$ αντίστοιχα, με αρχική ρίζα του δένδρου το κλάσμα $\frac{1}{1}$.Το μονοπάτι που σχηματίζεται από τη ρίζα του δένδρου έως κάποιο κόμβο με τιμή έστω n, αντιστοιχεί σε μια φθίνουσα ακολουθία η οποία προσεγγίζει τον αριθμό αυτό.



Σχήμα 2.1 Τα τέσσερα πρώτα επίπεδα του δένδρου Stern-Brocot [8]

Επίσης, ορίζουμε ως *ακολουθία Stern-Brocot* την ακολουθία :

- $a(0) = 0\,,\ a(1) = 1.$
- $Για\ n > 0,\ a(2n) = a(n)\ \ και\ \ a(2n+1) = a(n) + a(n+1)$

Οι ακολουθίες Stern-Brocot χρησιμοποιούνται για να παράξουμε μια ακολουθία από τους κόμβους ενός δένδρου Stern-Brocot ανά επίπεδο. Αυτό γίνεται παίρνοντας ανά ζεύγη τα στοιχεία μιας τέτοιας ακολουθίας και απορρίπτοντας τον πρώτο όρο, δηλαδή το 0.

2.2.3 Ακολουθίες Farey

Οι ακολουθίες Farey, έχουν στενή σχέση με τα δένδρα Stern-Brocot και είναι επίσης ένας τρόπος προσέγγισης των ρητών αριθμών.

Η ακολουθία Farey τάξης n ορίζεται ως το σύνολο

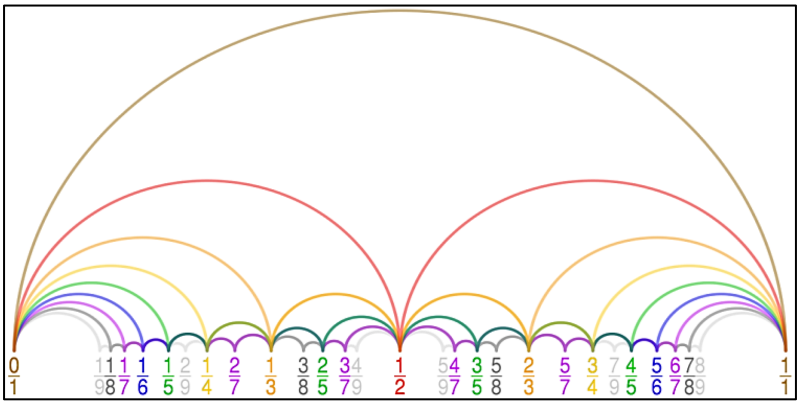
$$F_n = \{ \frac{p}{q} \mid p, q \in \mathbb{N}\, , \text{ΜΚΔ}(p, q) = 1, 0 < x \leq y \leq n \}$$

, διατεταγμένο κατά αύξουσα σειρά .

Συχνά η ακολουθία περικλείεται από τα κλάσματα $\frac{0}{1}$ και $\frac{1}{1}$ έτσι ώστε να είναι φραγμένη στο διάστημα [0,1], αλλά γενικά χρησιμοποιείται και χωρίς αυτόν τον περιορισμό.

Για παράδειγμα, $F_3 = \left\{\frac{0}{1}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{1}{1}\right\}$.

Στο Σχήμα 2.2 απεικονίζονται τα στοιχεία της ακολουθίας F_9, καθώς και κυκλικά τόξα αντίστοιχου μεγέθους για κάθε ένα στοιχείο.



Σχήμα 2.2 Οι ακολουθίες F_9, απεικονιζόμενες με κυκλικά τόξα.[9]

Όσων αφορά την υπολογιστική πολυπλοκότητα παραγωγής των ακολουθιών Farey, έχει αποδειχθεί ότι $|F_n| = \frac{3n^2}{\pi^2} + O(n * logn)$ και πως μπορούμε να υπολογίσουμε την ακολουθία F_n σε $O(|F_n|)$ χρόνο [9].

2.3 Περιγραφή των αλγορίθμων

Στην ενότητα αυτή θα περιγράψουμε συνοπτικά τις βασικές ιδέες πίσω από τους αλγορίθμους που θα υλοποιήσουμε για την παραγωγή μονότονων απεικονίσεων δένδρων.

2.3.1 Αλγόριθμοι Angelini, Colasante, Di Battista, Frati, Patrignani.

Αλγόριθμος BFS-based.

Αρχικά, ο αλγόριθμος παράγει ένα σύνολο από διακριτά κλάσματα της μορφής $\frac{y}{x}$ και τα διατάσσει κατά αύξουσα τιμή του ρητού που προκύπτει από τη διαίρεση. Το σύνολο αυτό ονομάζεται *πρωτογενή(ή πρωτόγονα) διανύσματα* (primitive vectors) και η έννοια αυτή χρησιμοποιείται από την πλειοψηφία των αλγορίθμων που θα δούμε στη συνέχεια. Το σύνολο από τα πρωτογενή διανύσματα παράγεται παίρνοντας τα πρώτα $\lceil \log_2(n) \rceil$ επίπεδα του δένδρου Stern-Brocot. Έπειτα, ο αλγόριθμος αντιστοιχεί ένα σύνολο από αυτά σε κάθε υποδένδρο $T(u)$ που παράγεται από κάθε κόμβο u του δένδρου. Η τοποθέτηση των κόμβων στο πλέγμα γίνεται ως εξής: Τοποθετούμε τη ρίζα του δένδρου r στο σημείο (0,0). Σε κάθε κόμβο του δένδρου u , αντιστοιχούμε τις συντεταγμένες του διανύσματος που αντιστοιχήθηκε στο τελευταίο στοιχείο της ακολουθίας $T(u)$, έστω $(y_u\ ,\ x_u\ )$. Κάθε κόμβος u θα σχεδιαστεί στο σημείο $(p_x(u) + x_u\ , p_y(u) + y_u\ )$, όπου $(p_x(u)\ , p_y(u))$ οι συντεταγμένες του πατέρα του u.

Μέγεθος πλέγματος: $O(n^{1.6})\ x\ O(n^{1.6})$

Αλγόριθμος DFS-based

Πρόκειται για μια παραλλαγή του προηγούμενου αλγορίθμου με τη διαφορά ότι επιλέγονται μόνο τα πρώτα n-1 στοιχεία του δεξιότερου μονοπατιού του δένδρου Stern-Brocot, δηλαδή η ακολουθία $\left[\frac{1}{1}, \frac{2}{1}, \dots, \frac{n-1}{1}\right]$. Στη συνέχεια, ακολουθείται η ίδια διαδικασία όπως ακριβώς με το προηγούμενο αλγόριθμο BFS-based.

Μέγεθος πλέγματος: $O(n)\ x\ O(n^2)$

2.3.2 Αλγόριθμος Kindermann, Schulzm, Spoerchase, Wolf.

Αλγόριθμος inorder-based

Αρχικά χρησιμοποιείται ένας αναδρομικός αλγόριθμος που διασχίζει όλο το δένδρο ξεκινώντας από τη ρίζα, αντιστοιχώντας έναν ακέραιο δείκτη σε κάθε ακμή που διατρέχει. Ο τρόπος διάσχισης του δένδρου είναι παρόμοιος με την ενδοδιαταγμένη διάσχιση ενός δένδρου (inorder traversal). Στη συνέχεια, εξετάζονται τα υποδένδρα της ρίζας με αντιωρολογιακή σειρά και τα διαχωρίζονται σε τρία υποσύνολα. Δεν αναφέρεται κάποιος τρόπος για να γίνει ο διαχωρισμός αυτός, αλλά ο στόχος είναι τα υποσύνολα αυτά να είναι όσο το δυνατόν πιο ισομεγέθη γίνεται. Κάθε υποδένδρο θα χρησιμοποιήσει πρωτογενή διανύσματα που ανήκουν στο 1^{ο} , 2^{ο} και 4^{ο} τεταρτημόριο. Αυτό γίνεται παίρνοντας ένα αρχικό σύνολο από τις ακολουθίες Farey και τις ανακλάσεις του. Οι τελικές συντεταγμένες κάθε κόμβου υπολογίζονται με παρόμοιο τρόπο όπως και στους αλγορίθμους των Angelini et al. Εδώ πρέπει να επισημάνουμε ότι ο αλγόριθμος αυτός απαιτεί η ρίζα να έχει τουλάχιστον τρία παιδιά, έτσι ώστε να μπορεί να χωριστεί σε τρία μέρη.

Μέγεθος πλέγματος: $O(n^{1.5})\ x\ O(n^{1.5})$

2.3.3 Αλγόριθμος D .He -X. He

Αλγόριθμος Optimal Draw

Οι He και He, παρόμοια με τη δουλειά των Kindermann et al., κάνουν χρήση των ακολουθιών Farey για να παράξουν τα απαιτούμενα πρωτογενή διανύσματα σε αύξουσα σειρά. Η παραγωγή των διανυσμάτων δεν καθορίζεται απλά από το συνολικό μέγεθος του δένδρου, όπως στους προηγούμενους αλγορίθμους, αλλά γίνεται με μια διαδικασία όπου εξετάζονται όλα τα μονοπάτια από τη ρίζα μέχρι τα φύλλα του δένδρου από αριστερά προς τα δεξιά. Η βασική ιδέα είναι να χρησιμοποιηθούν μικρά πρωτογενή διανύσματα στα μονοπάτια που είναι μεγάλα σε μήκος. Η απεικόνιση των κόμβων στο πλέγμα γίνεται όπως και στα προηγούμενα.

Μέγεθος πλέγματος: $12n\ x\ 12n$

2.3.4 Αλγόριθμοι Α.Οικονόμου-Α.Συμβώνης

Οι Οικονόμου-Συμβώνης περιέγραψαν τρείς αλγορίθμους στους οποίους δεν γίνεται χρήση της θεωρίας αριθμών, όπως είδαμε σε όλους τους προηγούμενους αλγορίθμους.

Αλγόριθμος 1. Μονότονη Απεικόνιση Δένδρου σε 1 τεταρτημόριο

Ο αλγόριθμος αρχικά αναθέτει ένα εύρος γωνιών σε κάθε κόμβο του δένδρου, αυτή η διαδικασία είναι η αντίστοιχη με τα πρωτογενή διανύσματα που παρήγαγαν οι προηγούμενοι αλγόριθμοι. Η ανάθεση γίνεται αναδρομικά, με βάση το μέγεθος του υποδένδρου που προκύπτει από τον κάθε κόμβο του δένδρου. Στη ρίζα ανατίθεται το εύρος γωνίας $(0, \frac{\pi}{2})$. Η στρατηγική αυτή εξασφαλίζει ότι παράγεται μια μονότονη απεικόνιση, η οποία απεικονίζεται στο 1^{o} τεταρτημόριο.

Μέγεθος πλέγματος: $n\ x\ n$

Αλγόριθμος 2. Μονότονη Απεικόνιση Δένδρου σε 2 τεταρτημόρια

Σε αυτή την περίπτωση επιλέγεται μια νέα ρίζα του δένδρου η οποία ονομάζεται ρίζα βαρύτητας. Η βασική ιδέα είναι να χωριστεί το δένδρο σε δύο μέρη, τα οποία να είναι όσο πιο ισομεγέθη γίνεται. Ο αλγόριθμος, χρησιμοποιεί κάθε κόμβο $u \in T$ ως νέα ρίζα του δένδρου και ελέγχει αν για κάθε παιδί v του u ισχύει ότι $|T_v \leq \frac{n}{2}|$, τότε ο κόμβος u επιλέγεται ως ρίζα βαρύτητας. Έπειτα ακολουθείται παρόμοια διαδικασία με αυτή του πρώτου αλγορίθμου αλλά αυτή τη φορά η ανάθεση συντεταγμένων γίνεται με τέτοιο τρόπο ώστε τα δύο μέρη του δένδρου να απεικονιστούν στα πρώτα δύο τεταρτημόρια.

Μέγεθος πλέγματος: $n\ x\,\frac{n}{2}$

Αλγόριθμος 3. Μονότονη Απεικόνιση Δένδρου σε 4 τεταρτημόρια

Ο αλγόριθμος κάνει χρήση των δύο προηγουμένων αλγορίθμων. Το δένδρο T που δίνεται ως είσοδος, διαχωρίζεται σε δύο μέρη και υπολογίζεται η ρίζα βαρύτητας r. Το μεγαλύτερο σε μέγεθος υποδένδρο από αυτά, έστω το T_1, σχεδιάζεται στα δύο πρώτα τεταρτημόρια χρησιμοποιώντας τον προηγούμενο αλγόριθμο. Στη συνέχεια, το μονοπάτι από τη ρίζα βαρύτητας του T_1 έως και την αρχική ρίζα βαρύτητας r του δένδρου, σχεδιάζεται ως επάνω στον άξονα x΄x. Το δεύτερο υποδένδρο T_2, απεικονίζεται στο 3^{o} και 4^{o} τεταρτημόριο, παίρνοντας την ανάκλαση της απεικόνισης που θα είχε, εάν σχεδιαζόταν μέσω του πρώτου αλγορίθμου.

Μέγεθος πλέγματος: $\left\lfloor \frac{3}{4}(n+1) \right\rfloor x \left\lfloor \frac{3}{4}(n+1) \right\rfloor$

3. Παραγωγή δένδρων

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τη διαδικασία παραγωγής των δένδρων στα οποία θα εφαρμοστούν οι αλγόριθμοι. Τα δένδρα που μας ενδιαφέρουν, είναι τα δένδρα με ρίζα και διατεταγμένα στο επίπεδο (ordered rooted trees).

Περιγράφουμε δύο αλγορίθμους παραγωγής δένδρων, έναν για την εξαντλητική παραγωγή όλων των δυνατών δένδρων με n κόμβους και έναν για τη μεμονωμένη κατασκευή τυχαίων δένδρων.

3.1 Οι αριθμοί Catalan

Οι αριθμοί Catalan αποτελούν μια ακολουθία από πραγματικούς αριθμούς η οποία φαίνεται να δίνει λύση σε αρκετά πολλά προβλήματα των διακριτών μαθηματικών που σχετίζονται με την καταμέτρηση συνόλων.

Ορίζουμε την ακολουθία Catalan ως:

$$C_n = \frac{1}{n+1}\binom{2n}{n} = \frac{(2n)!}{(n+1)!\,n!} \quad ,n \geq 0$$

Οι ακολουθίες αυτές αποτελούνται από ακέραιους αριθμούς παρόλο που αυτό δεν είναι προφανές καθώς ο τύπος δίνεται με τη μορφή κλάσματος.

Ο n-οστός όρος της ακολουθίας Catalan , είναι μεταξύ άλλων, το πλήθος όλων των δένδρων με n+1 κόμβους, καθορισμένη ρίζα και συγκεκριμένα διατεταγμέναα στο επίπεδο. Επιπλέον, είναι ο αριθμός όλων των ισορροπημένων συμβολοσειρών από παρενθέσεις που περιέχει ακριβώς n-ζεύγη παρενθέσεων, δηλαδή 2n χαρακτήρες.

Ισορροπημένες (balanced) συμβολοσειρές παρενθέσεων ονομάζονται αυτές όπου κάθε ζεύγος ανοικτής και κλειστής παρένθεσης είναι έγκυρα τοποθετημένο. Για παράδειγμα, για n=4, τα αποδεκτά ζεύγη παρενθέσεων είναι : "((()))", "(()())", "(())()", "()(())", "()()()"

Συνεπώς, τα διατεταγμένα δένδρα με ρίζα και οι ισορροπημένες οι συμβολοσειρές από παρενθέσεις είναι δύο σύνολα ισομορφικά μεταξύ τους και άρα μπορούμε να αντιστοιχήσουμε σε κάθε τέτοιο δένδρο μία συγκεκριμένη συμβολοσειρά και το αντίστροφο.

Στη συνέχεια, θα δούμε πως ακριβώς θα γίνεται αυτή η διαδικασία.

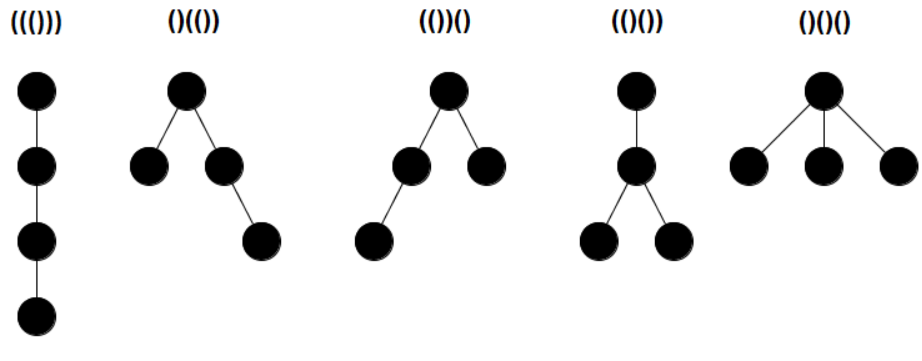
3.2 Αναπαράσταση δένδρων με χρήση παρενθέσεων

Το γεγονός ότι τα δένδρα ταυτίζονται με τις έγκυρες συμβολοσειρές παρενθέσεων είναι εξαιρετικά χρήσιμο καθώς πρώτον, είναι ένας απλός τρόπος να κωδικοποιηθεί ένα δένδρο που έχει συγκεκριμένη διάταξη στο επίπεδο και επιπλέον, αποτελεί έναν πολύ πιο απλό τρόπο παραγωγής τους.

Ο τρόπος με τον οποίο από ένα δένδρο μπορεί να παραχθεί μια συμβολοσειρά παρενθέσεων και το αντίστροφο, είναι ο παρακάτω:

Έστω, ότι έχουμε ένα δένδρο με ρίζα που βρίσκεται και οι κόμβοι έχουν τοποθετηθεί με αντι-ωρολογιακή σειρά. Εκτελούμε έναν περίπατο στο δένδρο, ξεκινώντας από τη ρίζα και ακολουθώντας τη σειρά που είναι διατεταγμένοι οι κόμβοι. Κάθε φορά που μετακινούμαστε προς ένα νέο κόμβο χρησιμοποιούμε μια ανοικτή παρένθεση ΄(΄ ενώ κάθε φορά που μετακινούμαστε προς τα πίσω μια κλειστή ΄)΄.

Η διαδικασία αυτή γίνεται κατανοητή πολύ πιο εύκολα οπτικά. Ενδεικτικά, παρουσιάζουμε όλα τα δένδρα με n=4 και με τις αντίστοιχες παρενθετικοποιήσεις τους στο Σχήμα 3.1:



Σχήμα 3.1. Αναπαράσταση δένδρων μέσω ισορροπημένων συμβολοσειρών από παρενθέσεις

3.3 Εξαντλητική παραγωγή δένδρων

Για να παράξουμε όλα τα δυνατά διατεταγμένα δένδρα με ρίζα και n κόμβους, παράγουμε όλες τις ισορροπημένες συμβολοσειρές από παρενθέσεις μήκους 2(n-1).

Στη συνέχεια, παρουσιάζουμε τον αλγόριθμο που χρησιμοποιήθηκε για την εξαντλητική παραγωγή όλων των δένδρων με n κόμβους, με χρήση ψευδοκώδικα.

Αλγόριθμος 1.1: Εξαντλητική παραγωγή (n)

Είσοδος : Πλήθος κόμβων n
Έξοδος : Συμβολοσειρά παρενθέσεων μήκους 2(n-1)

```
Διαδικασία Επεξεργασία (Συμβολοσειρά = [] , #ανοικτό = 0, #κλειστό = 0):

    Αποτέλεσμα = Λίστα[]
    Αν  μήκος(Συμβολοσειρά) = 2*(n-1) :
        Αποτέλεσμα =+ Συμβολοσειρά
    Αν #ανοικτό < n-1 :
        Επεξεργασία(Συμβολοσειρά + ‘(‘ , #ανοικτό + 1 , κλειστό)
    Αν #κλειστό < #ανοικτό :
        Επεξεργασία(Συμβολοσειρά + ‘)’ , #ανοικτό  , κλειστό+1)

    Επεξεργασία()
    Επέστεψε Αποτέλεσμα
```

Περιγραφή αλγορίθμου:

Ο αλγόριθμος δέχεται για είσοδο τον αριθμό των κόμβων n και επιστρέφει όλες τις δυνατές συμβολοσειρές, δηλαδή όλα τα δυνατά δένδρα με n κόμβους. Η διαδικασία γίνεται αναδρομικά και η ιδέα είναι ότι προσθέτουμε μια κλειστή ή ανοικτή παρένθεση μόνο εάν η παραχθείσα συμβολοσειρά είναι αποδεκτή. Δηλαδή, προσθέτουμε μια ανοικτή παρένθεση με την προϋπόθεση ότι το πλήθος των ανοικτών παρενθέσεων είναι μικρότερο του αριθμού των κόμβων και μια κλειστή παρένθεση όταν το πλήθος τον ανοικτών είναι μεγαλύτερο του πλήθους των κλειστών.

Ανάλυση πολυπλοκότητας:

Για τους αριθμούς Catalan, ισχύει ότι, $C_n \sim \frac{\mathbf{4^n}}{\mathbf{n}^{1.5}\sqrt{\pi}}$, συνεπώς ο αλγόριθμος έχει χρονική και χωρική πολυπλοκότητα $O(\frac{\mathbf{4^n}}{\mathbf{n}^{1.5}})$.

3.4 Παραγωγή τυχαίων δένδρων

Εκτός από την παραγωγή όλων των δυνατών δένδρων, απαιτείται και η παραγωγή τυχαίων δένδρων αφού όπως είδαμε, το να παράξουμε όλα τα δένδρα είναι υπολογιστικά μια ιδιαίτερα κοστοβόρα διαδικασία .

Για να παράξουμε ένα τυχαία δένδρο, κατασκευάζουμε μια τυχαία συμβολοσειρά μήκους 2(n-1) και ελέγχουμε αν αυτή τηρεί τις προϋποθέσεις για να είναι έγκυρη. Εάν δεν είναι, επαναλαμβάνουμε τη διαδικασία έως ότου εντοπίσουμε μια έγκυρη συμβολοσειρά.

Αλγόριθμος 1.2: Γεννήτρια Τυχαίων (n)

Είσοδος : Πλήθος κόμβων n
Έξοδος : Συμβολοσειρά μήκους 2(n-1)

```
Διαδικασία Τυχαία Συμβολοσειρά( n ):
    Χαρακτήρες = [ ‘(’ ,  ’)’  ] * (n-1)
    Τυχαίο_Ανακάτεμα( Χαρακτήρες )
    Επέστρεψε Χαρακτήρες

  Διαδικασία Έγκυρη(Είσοδος):
        Αν Είσοδος[0] = ‘)‘ :
              Επέστρεψε Ψευδές

        Για κάθε χαρακτήρα της εισόδου :
            Αν είναι ‘(‘  :  πλήθος ανοικτών += 1
            Αν είναι ‘)‘  :  πλήθος κλειστών += 1
           Αν (πλήθος κλειστών) > (πλήθος ανοικτών):
                 Επέστρεψε Ψευδές

      Αν (πλήθος ανοικτών) = (πλήθος κλειστών):
          Επέστρεψε Αληθές
      Αλλιώς:
          Επέστρεψε Ψευδές

    Αποτέλεσμα = Ψευδής
    Όσο ( Αποτέλεσμα = Ψευδής):
      Συμβολοσειρά = Τυχαία Συμβολοσειρά(n)
          Αν ( Έγκυρη(Συμβολοσειρά) = Αληθής ):
            Επέστρεψε Συμβολοσειρά
         Αλλιώς:
            Επανέλαβε
```

Περιγραφή αλγορίθμου:

Πρώτα παράγεται μια τυχαία συμβολοσειρά μήκους 2(n-1), παράγοντας n-1 χαρακτήρες ανοικτών και κλειστών παρενθέσεων. Στη συνέχεια γίνεται ένα τυχαίο ανακάτεμα των χαρακτήρων της συμβολοσειράς και στη συνέχεια ελέγχεται αν είναι έγκυρη ως εξής:

Ο αλγόριθμος αρχικά ελέγχει τον πρώτο χαρακτήρα της συμβολοσειράς, και εάν είναι κλειστή παρένθεση επιστρέφει Ψευδής.

Έπειτα ελέγχει σειριακά τη συμβολοσειρά, και εξετάζει αν μέχρι τώρα, ο αριθμός των ανοικτών παρενθέσεων είναι μικρότερος του αριθμού των κλειστών. Η ιδέα είναι ότι, για να είναι η συμβολοσειρά έγκυρη, δεν θα πρέπει σε κανένα σημείο της να έχουμε περισσότερες κλειστές απ’ότι ανοικτές παρενθέσεις έως το σημείο αυτό. Όταν συμβαίνει αυτό, ο αλγόριθμος τερματίζει.

Τέλος, ο αλγόριθμος ελέγχει αν το πλήθος των ανοικτών παρενθέσεων ισούται με το πλήθος των κλειστών και επιστρέφει την τιμή Αληθής ή Ψευδής ανάλογα με το αποτέλεσμα.

Ανάλυση πολυπλοκότητας:

Για να αναλύσουμε τη χρονική πολυπλοκότητα του αλγορίθμου, θα πρέπει να δούμε πόσα είναι όλα τα δυνατά δένδρα με ρίζα, δηλαδή οι αριθμοί Catalan, σε σχέση με όλες τις δυνατές συμβολοσειρές μήκους 2n που είναι 2^{2n}.

Είδαμε προηγουμένως ότι $C_n \sim \frac{\mathbf{4^n}}{\mathbf{n}^{1.5}\sqrt{\boldsymbol{\pi}}}$, άρα η πιθανότητα να παράξουμε τυχαία μια έγκυρη συμβολοσειρά φράζεται από την ποσότητα $\frac{C_n}{2^{2n}} \sim \frac{\frac{\mathbf{4^n}}{\mathbf{n}^{1.5}\sqrt{\boldsymbol{\pi}}}}{2^{2n}} = \frac{1}{\mathbf{n}^{1.5}\sqrt{\boldsymbol{\pi}}}$.

Συνεπώς η χρονική πολυπλοκότητα του αλγορίθμου είναι $O(\mathbf{n}^{1.5})$.

4. Υλοποίηση διαδραστικού εργαλείου οπτικοποίησης των αλγορίθμων

Στα πλαίσια της ανάλυσης των αλγορίθμων, πρέπει να μπορούμε να έχουμε και οπτική εποπτεία των αποτελεσμάτων τους. Έτσι, κατασκευάσαμε ένα διαδικτυακό εργαλείο, στο οποίο μπορούμε να παράξουμε μονότονες απεικονίσεις οποιουδήποτε δένδρου θέλουμε, εφαρμόζοντας τους αλγορίθμους που περιγράψαμε στα προηγούμενα κεφάλαια.

4.1 Λεπτομέρειες υλοποίησης

Για την υλοποίηση των αλγορίθμων, χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python 3.6. Κάθε αλγόριθμος βρίσκεται σε ένα ξεχωριστό αρχείο και χρησιμοποιείται αυτόνομα. Η επεξεργασία και ανάλυση των γραφημάτων έγινε μέσω της βιβλιοθήκης Networkx της Python η οποία χρησιμοποιήθηκε εκτενώς. Για την παραγωγή των οπτικοποιήσεων χρησιμοποιήθηκε η βιβλιοθήκη Matplotlib καθώς και κάποιες βασικές λειτουργίες που παρέχει η βιβλιοθήκη NetworkX και βασίζονται σε αυτή.

Η διεπιφάνεια χρήστη κατασκευάστηκε μέσω της γλώσσας σήμανσης HTML5 και τα αισθητικά χαρακτηριστικά της μέσω των CSS3 και Bootstrap3. Για τα διαδραστικά μέρη, χρησιμοποιήθηκε η JavaScript και η βιβλιοθήκη της jQuery.

Ως εξυπηρετητή (server) χρησιμοποιήσαμε το εργαλείο XAMPP, το οποίο προσφέρει έναν εξυπηρετητή Apache και σε αυτόν βρίσκονται όλα τα αρχεία της υλοποίησης. Για τη διαχείριση του εξυπηρετητή χρησιμοποιήθηκε η γλώσσα προγραμματισμού PHP . Για την παρουσίαση των αλγορίθμων στη διεπιφάνεια χρήστη, μέσω της PHP, στέλνονται εντολές στο λειτουργικό σύστημα του εξυπηρετητή, οι οποίες εκτελούν τα αρχεία της Python των αλγορίθμων, με τις παραμέτρους που επέλεξε ο χρήστης. Στη συνέχεια, παράγονται οι εικόνες που περιέχουν τις μονότονες απεικονίσεις των αλγορίθμων και μετέπειτα παρουσιάζονται στην ιστοσελίδα του εργαλείου.

4.2 Βασικές λειτουργίες

Παρακάτω θα αναφέρουμε συνοπτικά τις βασικές λειτουργίες του εργαλείου.

Εφαρμογή των αλγορίθμων και οπτικοποίηση των απεικονίσεων τους

Εφόσον έχει καθοριστεί το δένδρο προς χρήση, εφαρμόζονται όλοι οι αλγόριθμοι που υλοποιήθηκαν και επιστρέφονται οι οπτικοιποιήσεις τους στο πλέγμα. Για κάθε μία, παράγεται ένα αρχείο εικόνας στο οποίο παρουσιάζεται η μονότονη απεικόνιση του δένδρου και το μέγεθος του πλέγματος που καταλαμβάνει.

Εισαγωγή δένδρου με χρήση παρενθέσεων

Αρχικά, ο χρήστης καλείται να εισάγει ένα δένδρο σε μορφή συμβολοσειράς από παρενθέσεις, όπως είδαμε στο προηγούμενο κεφάλαιο. Η συμβολοσειρά που δίνεται, επεξεργάζεται από το εργαλείο και ελέγχεται κατά πόσο είναι έγκυρη, σε περίπτωση λάθους επιστρέφεται το αντίστοιχο μήνυμα.

Αυτόματη παραγωγή δένδρου

Εκτός από την επιλογή της χειροκίνητης εισαγωγής ενός δένδρου μέσω παρενθέσεων, ο χρήστης μπορεί να επιλέξει την αυτόματη παραγωγή ενός τυχαίου δένδρου με 3 έως και 15 κόμβους. Επιπλέον μπορεί να παράξει το γράφημα-μονοπάτι καθώς και το γράφημα-αστέρι, καθορίζοντας το πλήθος των κόμβων που επιθυμεί.

Εισαγωγή/Εξαγωγή αρχείων GraphML

Εκτός από την εισαγωγή δένδρων με τη μορφή παρενθέσεων, υποστηρίζεται και η επεξεργασία γραφημάτων σε μορφή αρχείου GraphML. Ο χρήστης μπορεί να επιλέξει να χρησιμοποιήσει ένα αρχείο GraphML από τον τοπικό του δίσκο και να εφαρμόσει τους αλγόριθμους σε αυτό. Επιπλέον, κάθε φορά που επιλέγεται από τον χρήστη ένα δένδρο σε μορφή παρενθέσεων, μετατρέπεται και σε αρχείο GraphML και υπάρχει διαθέσιμη επιλογή για την εξαγωγή του στον τοπικό δίσκο.

Σύγκριση δύο αλγορίθμων στο ίδιο διάγραμμα

Εκτός από την χρήση όλων των αλγορίθμων ξεχωριστά, υπάρχει η δυνατότητα να γίνει σύγκριση δύο αλγορίθμων, τους οποίους επιλέγει ο χρήστης, μέσω της εφαρμογής και απεικόνισής τους στο ίδιο πλέγμα , έτσι ώστε να φαίνεται καθαρά η διαφορά σε μέγεθος πλέγματος που καταλαμβάνει ο καθένας.

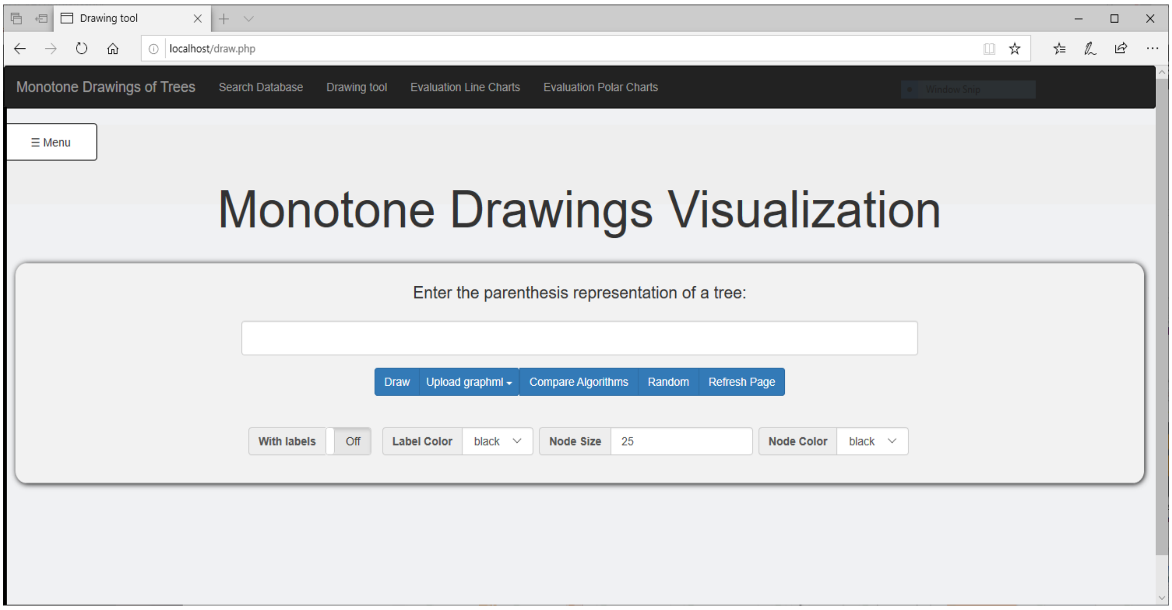
Αισθητικές παραμετροποιήσεις

Ο χρήστης έχει την δυνατότητα να τροποποιήσει κάποιες από τις αισθητικές παραμέτρους για την εμφάνιση των παραγόμενων απεικονίσεων. Συγκεκριμένα, μπορεί να επιλέξει την χρήση ετικετών στους κόμβους του δένδρου οι οποίες είναι κεφαλαία γράμματα του αγγλικού αλφαβήτου, το χρώμα των ετικετών αλλά και των κόμβων καθώς και το μέγεθος των κόμβων(1 έως και 300). Οι προκαθορισμένες παράμετροι είναι: μαύρο χρώμα για τους κόμβους, μέγεθος κόμβων 25 και η μη χρήση ετικετών.

4.3 Τρόπος λειτουργίας - Σενάρια χρήσης

Στη συνέχεια, θα παρουσιάσουμε τον τρόπο λειτουργίας του εργαλείου που υλοποιήθηκε για να επιτελέσει τις βασικές λειτουργίες που αναφέρθηκαν στην προηγούμενη παράγραφο.

Για να εισέλθουμε στη διεπαφή χρήστη του εργαλείου (Εικόνα 4.1) ανοίγουμε έναν περιηγητή ιστού(web browser) και μεταβαίνουμε στην ιστοσελίδα http://localhost/draw.php.

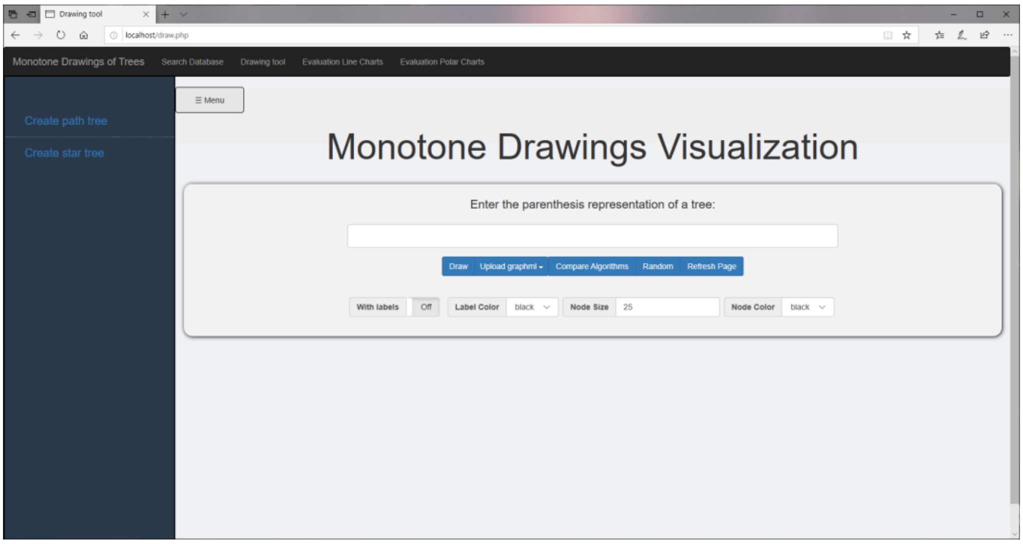


Εικόνα 4.1. Η διεπαφή χρήστη του εργαλείου οπτικοποίησης

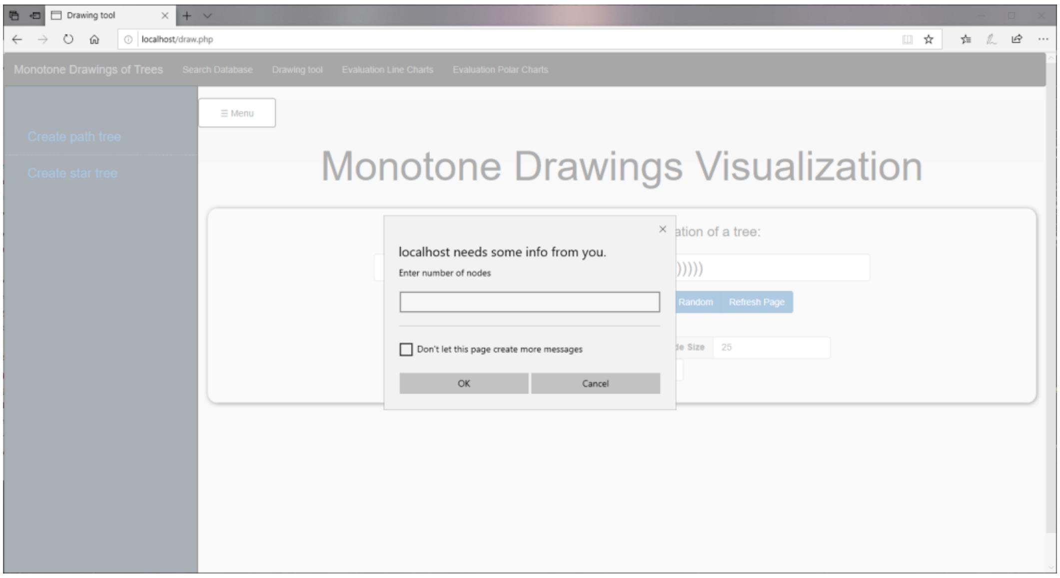
Όπως φαίνεται στην Εικόνα 4.1, υπάρχει ένα μεγάλο πεδίο εισόδου στο οποίο ο χρήστης καταχωρεί ένα δένδρο που επιθυμεί, σε μορφή συμβολοσειράς από παρενθέσεις. Κάτω από αυτό βρίσκονται τα κουμπιά που εκτελούν τις διαθέσιμες λειτουργίες του εργαλείου καθώς και τα πεδία που τροποποιούν τα αισθητικά χαρακτηριστικά. Στη μπάρα με μαύρο φόντο στο άνω τμήμα της σελίδας, βρίσκονται και οι σύνδεσμοι (links) για άλλα εργαλεία που θα δούμε μετέπειτα.

4.3.1 Εφαρμογή των αλγορίθμων και οπτικοποίηση

Αρχικά θα επιλέξουμε ένα δένδρο και θα το χρησιμοποιήσουμε για να δούμε τον τρόπο λειτουργίας του εργαλείου. Για το εν λόγω σενάριο χρήσης, θα επιλέξουμε το γράφημα-αστέρι (δένδρο με n-1 φύλλα) με 16 κόμβους. Για να γίνει αυτό κάνουμε κλικ στο κουμπί '**Menu**' που βρίσκεται πάνω αριστερά από το κεντρικό παράθυρο του εργαλείου, μόλις γίνει αυτό, εμφανίζονται δύο νέες επιλογές, αυτές τις παραγωγής του γραφήματος-μονοπάτι και του γραφήματος-αστέρι, όπως φαίνονται στην παρακάτω Εικόνα 4.2. Εδώ επιλέγουμε την επιλογή '**Create star tree**'. Στη συνέχεια, εμφανίζεται ένα νέο παράθυρο στο οποίο ζητείται ο καθορισμός του πλήθους των κόμβων που επιθυμούμε να χρησιμοποιήσουμε, όπως φαίνεται στην Εικόνα 4.3.



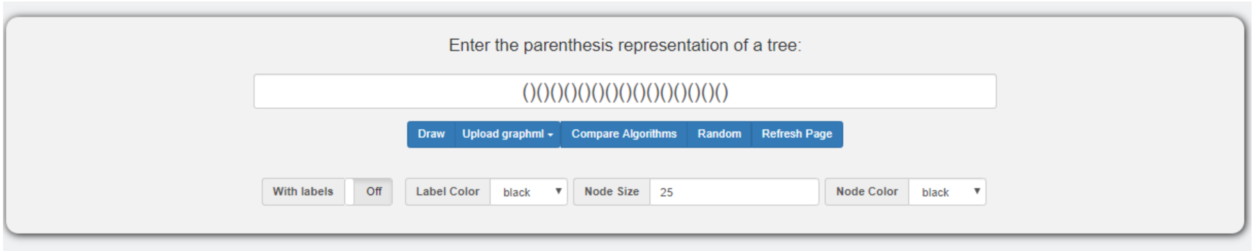
Εικόνα 4.2. Οι επιλογές που εμφανίζονται πατώντας το 'Menu'



Εικόνα 4.3 Παραγωγή γραφήματος-αστεριού, επιλογή πλήθους κόμβων

Καταχωρούμε τον αριθμό 16 στο νέο πεδίο που εμφανίζεται και πατάμε OK. Στη συνέχεια καταγράφεται στο πεδίο εισαγωγή δένδρων, η συμβολοσειρά από παρενθέσεις του γραφήματος αστεριού που παράξαμε, η οποία είναι η ‘()()()()()()()()()()()()()()()()’.

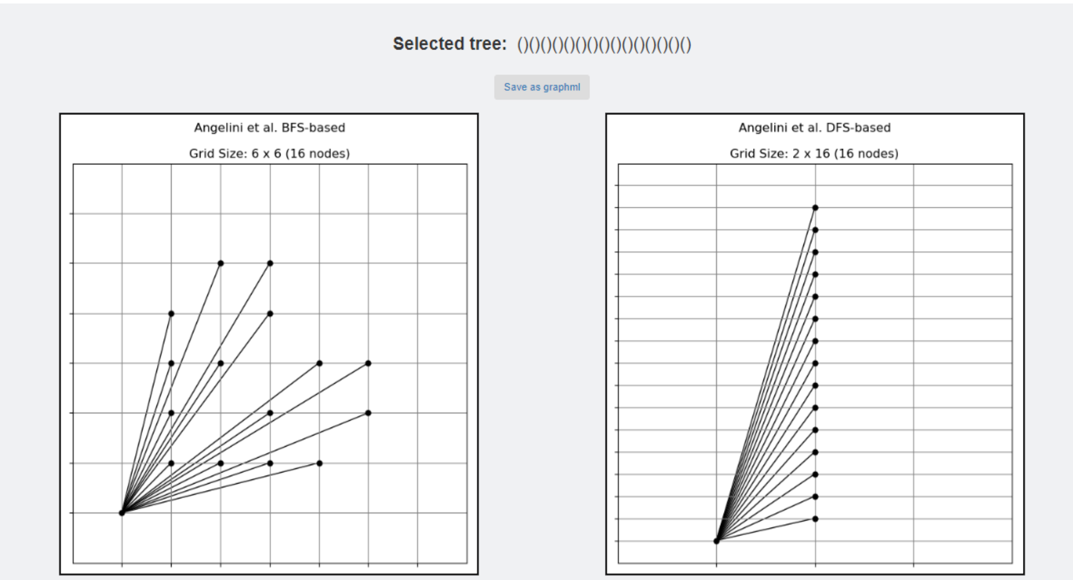
(Εικόνα 4.4).



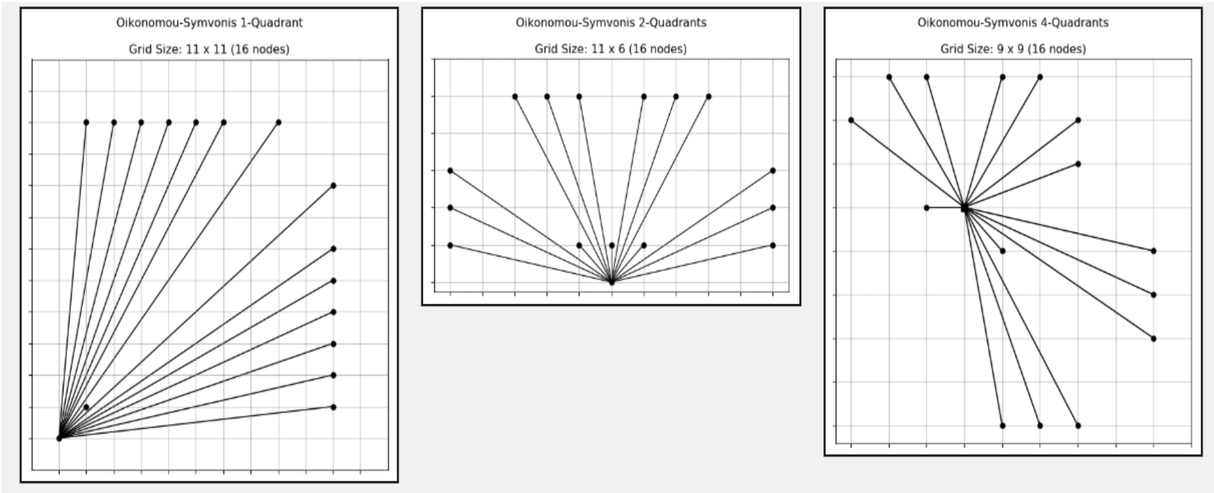
Εικόνα 4.4 Εισαγωγή δένδρου ως ισορροπημένης συμβολοσειράς από παρενθέσεις

Στη συνέχεια, θα επιλέξουμε την επιλογή ‘**Draw**’, που αποτελεί τη βασική λειτουργία του εργαλείου και η οποία θα εφαρμόσει τους αλγορίθμους στο εν λόγω δένδρο και θα επιστρέψει τις απεικονίσεις τους στον παρακάτω κενό χώρο. Για το παράδειγμα αυτό, δεν θα αλλάξουμε τα αισθητικά γνωρίσματα και θα αφήσουμε τα προεπιλεγμένα. Για κάθε απεικόνιση χρησιμοποιούμε μεταβλητό σύστημα αξόνων στο οποίο ο λόγος των x και y ρυθμίζεται αυτόματα έτσι ώστε να γίνεται πιο ευκρινές το τελικό γράφημα.

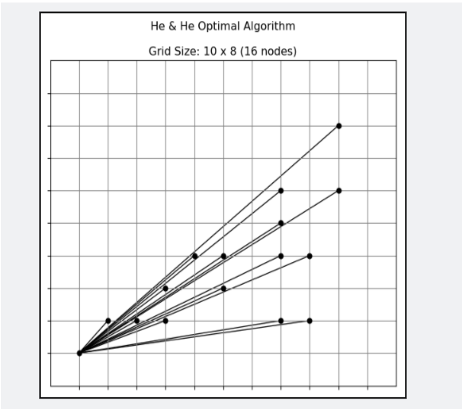
Στις Εικόνες 4.5 έως 4.8 παρουσιάζονται οι απεικονίσεις των αλγορίθμων.



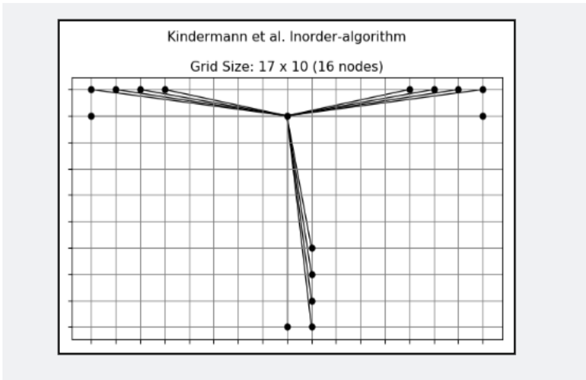
Εικόνα 4.5 Μονότονη απεικόνιση. Αλγόριθμοι Angelini et al.



Εικόνα 4.6 Μονότονη απεικόνιση. Αλγόριθμοι Οικονόμου-Συμβώνη



Εικόνα 4.7 Μονότονη απεικόνιση. Αλγόριθμος He&He



Εικόνα 4.8 Μονότονη απεικόνιση. Αλγόριθμος Kindermann et al.

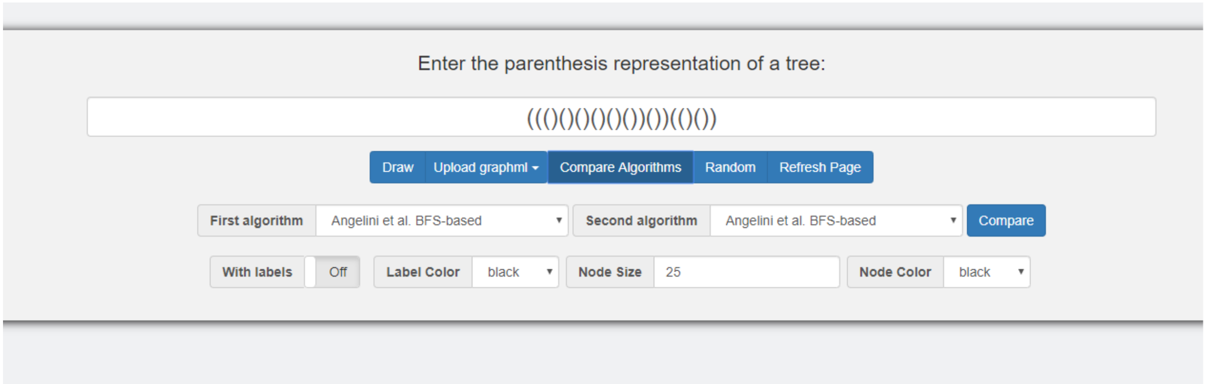
Ο χρήστης μπορεί να επιλέξει να εξάγει το εν λόγο δένδρο ως αρχείο τύπου GraphML μέσω της επιλογής ‘**Save as GraphML**’ που εμφανίζεται πάνω από τα αποτελέσματα (Εικόνα 4.5). Επιπλέον , για να αποθηκεύσει την εικόνα της οπτικοποίησης κάποιου αλγορίθμου, μπορεί να πατήσει δεξί κλικ πάνω σε μία από τις εικόνες και να επιλέξει ‘Αποθήκευση εικόνας ως…’ .

4.3.2 Σύγκριση δύο αλγορίθμων μεταξύ τους

Στη συνέχεια θα εξετάσουμε τη διαδικασία σύγκρισης δύο αλγορίθμων μεταξύ τους.

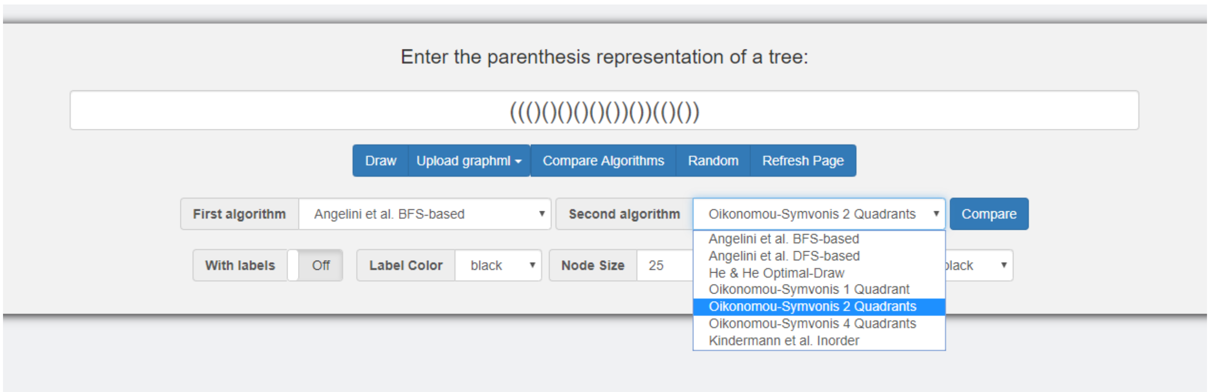
Το δένδρο που θα χρησιμοποιήσουμε θα είναι τυχαίο και γι'αυτό χρησιμοποιούμε τη λειτουργία **'Random'** , μέσω της οποίας παράγεται ένα τυχαίο δένδρο και εμφανίζεται στο πεδίο εισόδου με τη μορφή παρενθέσεων. Το τυχαίο δένδρο που παράχθηκε είναι το δένδρο με 13 κόμβους '((()()()()()())())(()())'.

Επιλέγοντας τη λειτουργία '**Compare Algorithms**', εμφανίζονται δύο νέα πεδία στα οποία ο χρήστης επιλέγει τους δύο αλγόριθμους που επιθυμεί να χρησιμοποιήσει. (Εικόνα 4.9)



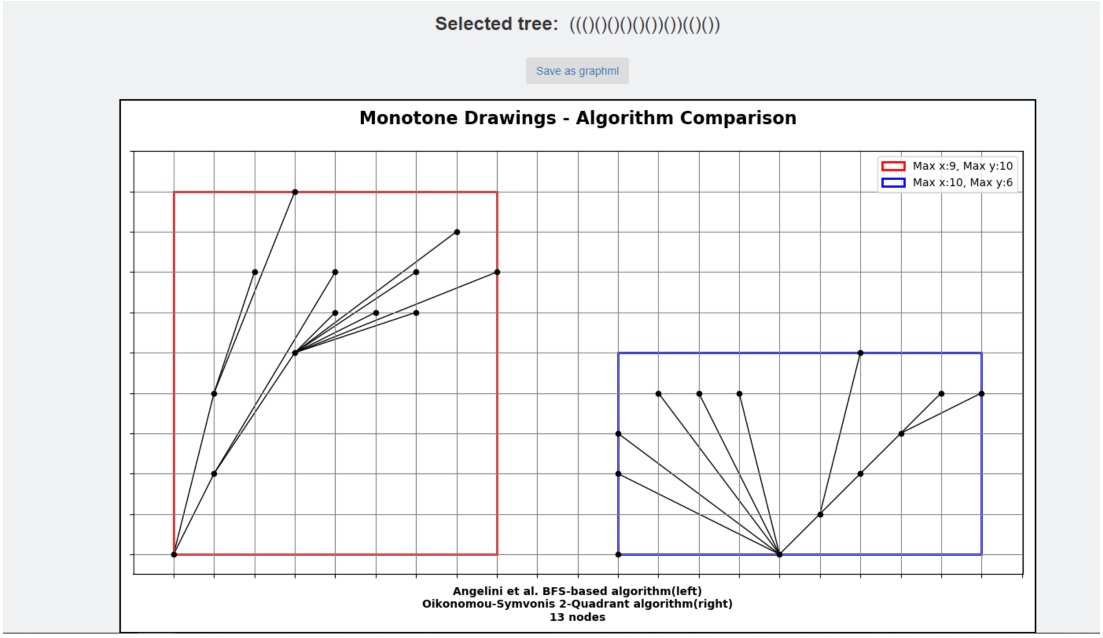
Εικόνα 4.9 Σύγκριση δύο αλγορίθμων. Εισαγωγή δένδρου

Στο συγκεκριμένο παράδειγμα θα επιλέξουμε ως πρώτο αλγόριθμο τον BFS-based αλγόριθμο των Angelini et al. και ως δεύτερο, αυτόν των Οικονόμου-Συμβώνη που χρησιμοποιεί δύο τεταρτημόρια για την απεικόνιση του δένδρου (2-Quadrants). (Εικόνα 4.10)



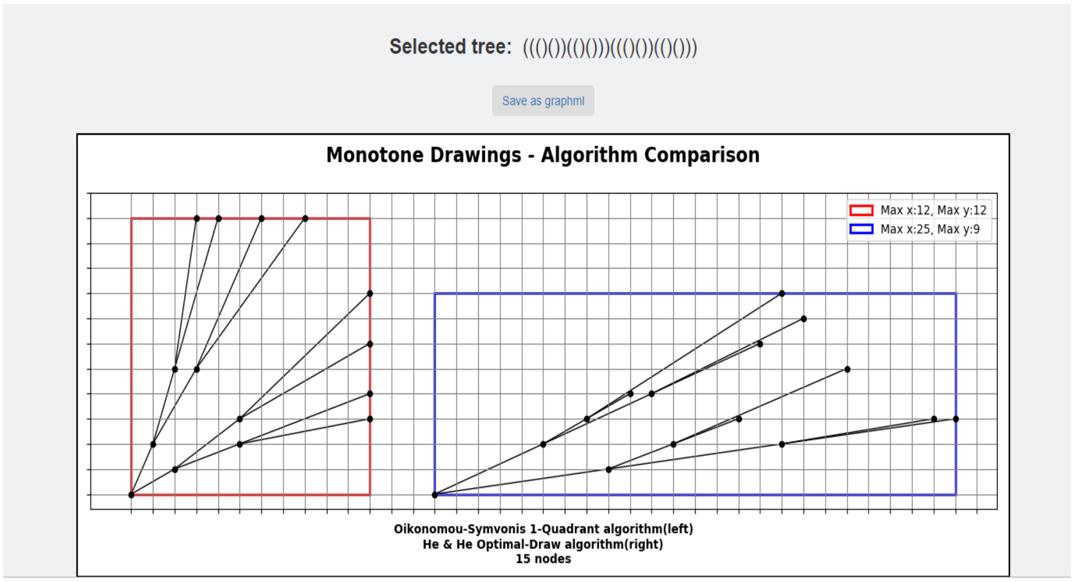
Εικόνα 4.10 Σύγκριση δύο αλγορίθμων. Επιλογή αλγορίθμων

Στη συνέχεια επιλέγουμε την επιλογή ‘**Compare**’ που βρίσκεται δεξιά από τα δύο πεδία της επιλογής των αλγορίθμων και η οποία θα εκτελέσει τη λειτουργία και επιστρέψει το αποτέλεσμα. Στην τελική εικόνα που επιστρέφεται (Εικόνα 4.11.1) , υπάρχουν στο ίδιο πλέγμα και οι δύο απεικονίσεις, οι οποίες περικλείονται σε ένα παραλληλόγραμμο η κάθε μία, έτσι ώστε να φαίνεται καθαρά ο απαιτούμενος χώρος που καλύπτουν του οποίου το μέγεθος αναφέρεται στο πάνω δεξιά τμήμα της εικόνας. Όπως και πριν υπάρχει η δυνατότητα λήψης της εικόνας καθώς και εξαγωγής του δένδρου ως GraphML αρχείο.



Εικόνα 4.11.1 Σύγκριση δύο αλγορίθμων - Τελικό αποτέλεσμα (1)

Ενδεικτικά παρουσιάζουμε (Εικόνα 4.11.2) και τα αποτελέσματα των αλγορίθμων Οικονόμου-Συμβώνης 1 Quadrant, και He&He Optimal-draw, για το πλήρες δυαδικό δένδρο τριών επιπέδων:

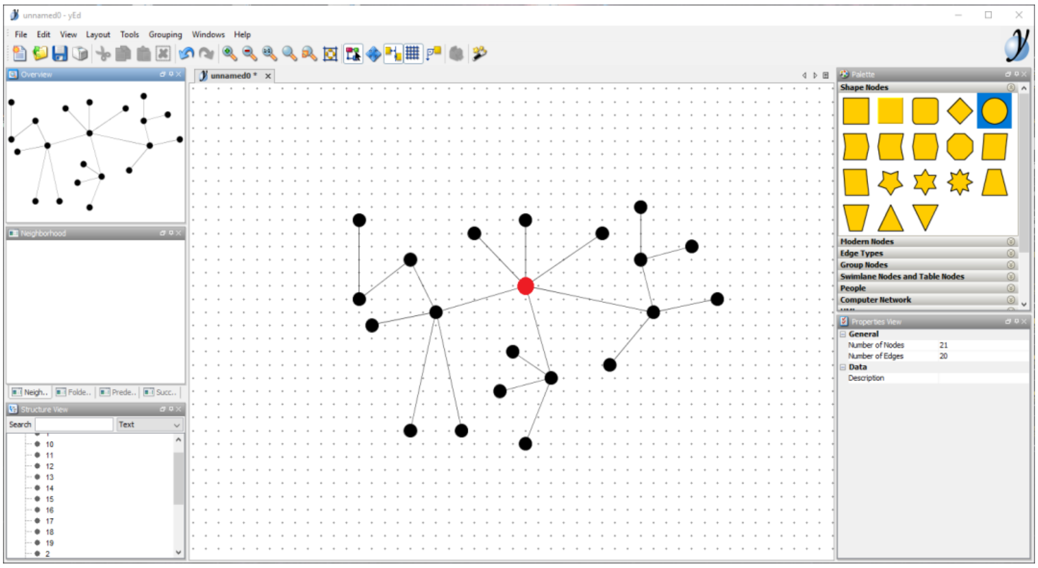


Εικόνα 4.11.2 Σύγκριση δύο αλγορίθμων - Τελικό αποτέλεσμα (2)

4.3.3 Εισαγωγή αρχείων τύπου GraphML.

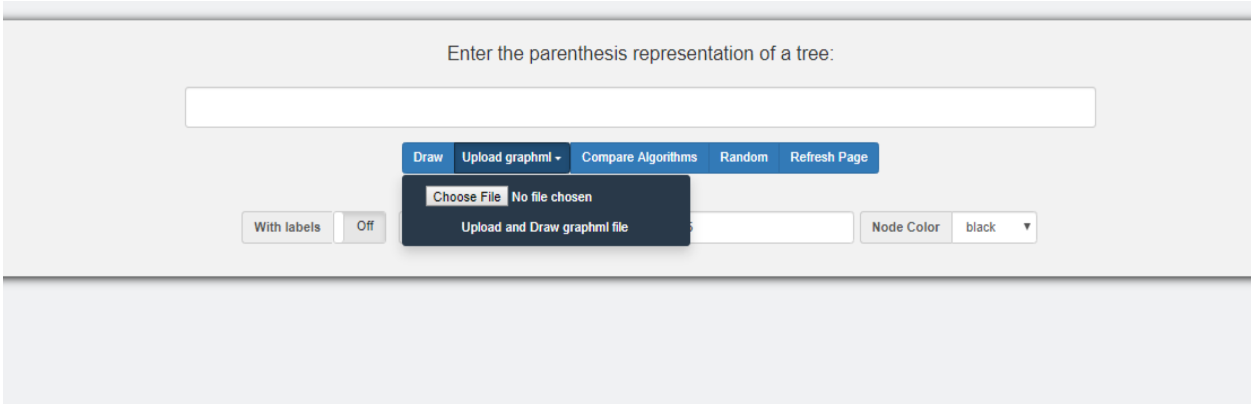
Στη συνέχεια, θα δούμε πως μπορεί κάποιος να χρησιμοποιήσει τη δυνατότητα ανεβάσματος ενός αρχείου GraphML και εφαρμογής των αλγορίθμων σε αυτό. Για τη δημιουργία ενός δοκιμαστικού αρχείου χρησιμοποιήσαμε το λογισμικό yEd Graph Editor της yWorks το οποίο είναι ένα ιδιαίτερα δημοφιλές και εύχρηστο πρόγραμμα μοντελοποίησης και ανάλυσης γραφημάτων. Ο χρήστης έχει τη δυνατότητα χρησιμοποιώντας ένα διαδραστικό περιβάλλον με πολλά διαθέσιμα εργαλεία και επιλογές, να σχεδιάσει χειροκίνητα κάποιο γράφημα αρκετά εύκολα και μπορεί να το αποθηκεύσει απευθείας ως αρχείο τύπου GraphML.

Το γράφημα που σχεδιάστηκε φαίνεται στην Εικόνα 4.14, η ρίζα του δένδρου είναι ο κόμβος με κόκκινο χρώμα.



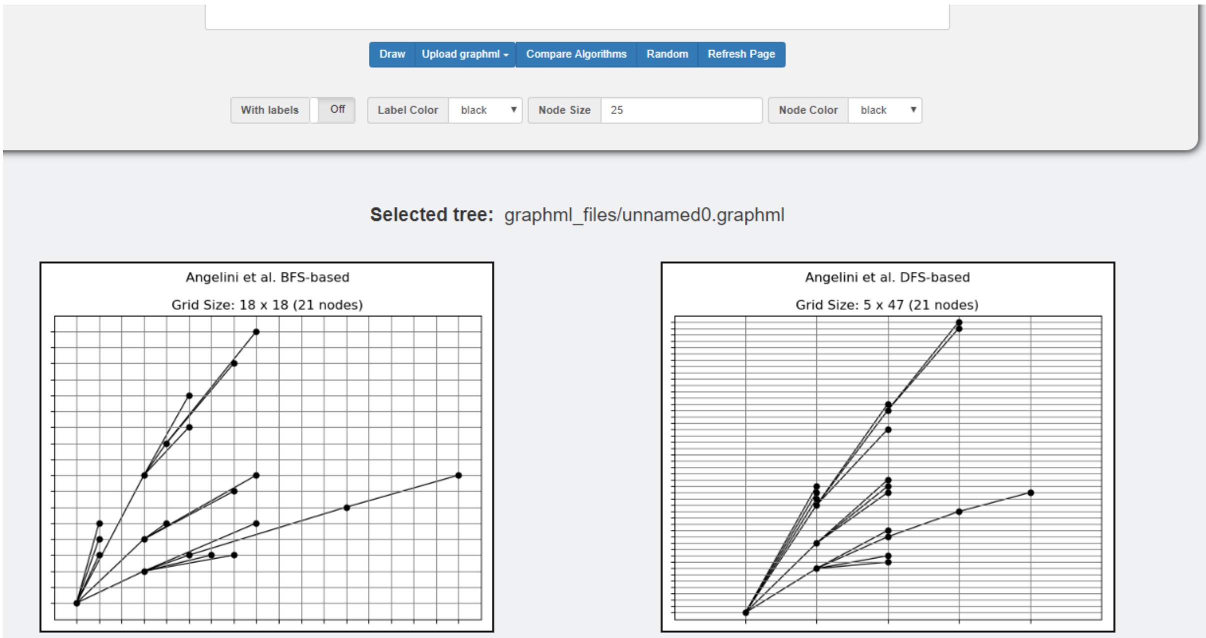
Εικόνα 4.12 Σχεδιασμός γραφήματος μέσω του yEd Graph Editor

Για να οπτικοποιήσουμε το αρχείο χρησιμοποιώντας την εφαρμογή μας, αρχικά επιλέγουμε τη λειτουργία '**Upload GraphML**' η οποία εμφανίζει ένα νέο μενού με δύο δυνατότητες , την επιλογή ενός αρχείου από τον τοπικό δίσκο του χρήστη πατώντας το εικονίδιο '**Choose File'**, και στη συνέχεια την οπτικοποίηση του δοσμένου δένδρου χρησιμοποιώντας τους αλγορίθμους, με την επιλογή '**Upload and Draw GraphML file'**.(Εικόνα 4.13)



Εικόνα 4.13 Επιλογή αρχείου GraphML.

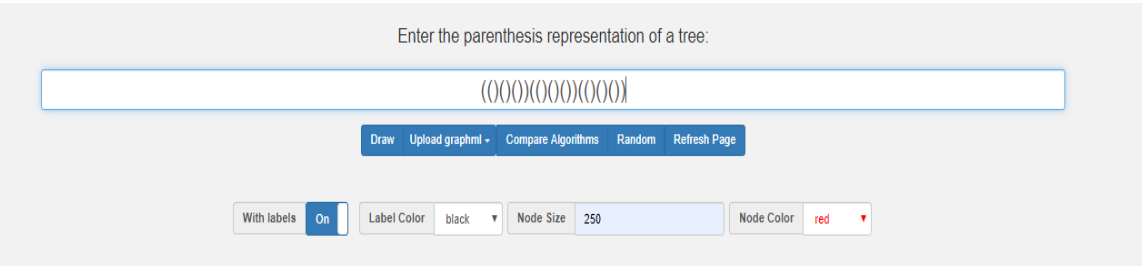
Στη συνέχεια επιστρέφονται τις μονότονες απεικονίσεις του δένδρου, αναφέροντας και το συγκεκριμένο αρχείο που επιλέχθηκε, ενδεικτικά παρουσιάζουμε τις πρώτες δύο απεικονίσεις που εμφανίζονται. (Εικόνα 4.14)



Εικόνα 4.14 Ανέβασμα αρχείου GraphML και οι μονότονες απεικονίσεις του.

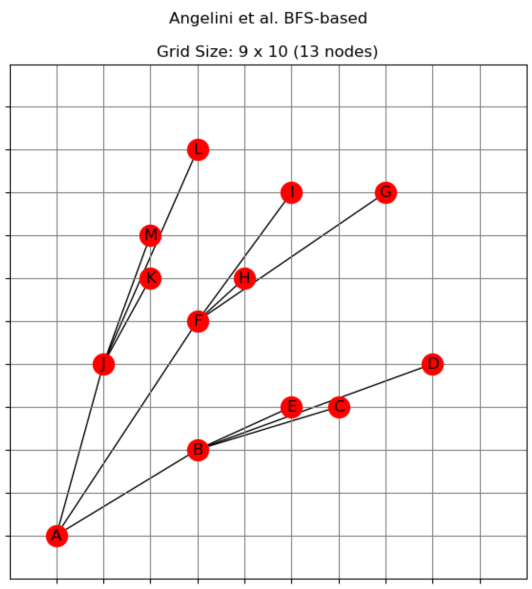
4.3.4 Τροποποίηση αισθητικών λεπτομερειών

Για να τροποποιήσουμε τα αισθητικά γνωρίσματα των τελικών απεικονίσεων, χρησιμοποιούμε τις επιλογές που βρίσκονται κάτω από τα κουμπιά με τις λειτουργίες του εργαλείου. Οι προκαθορισμένες τιμές, όπως αναφέρθηκε, είναι μαύρο χρώμα για του κόμβους, μέγεθος κόμβων 25 και η μη χρήση ετικετών. Για το συγκεκριμένο παράδειγμα θα χρησιμοποιήσουμε κόκκινο χρώμα για τους κόμβους, μέγεθος 250 και θα προσθέσουμε ετικέτες μαύρου χρώματος. Το δένδρο που επιλέχτηκε για το παράδειγμα αυτό είναι το πλήρες τριαδικό δένδρο δύο επιπέδων ‘(()()())(()()())(()()())’.(Εικόνα 4.14)

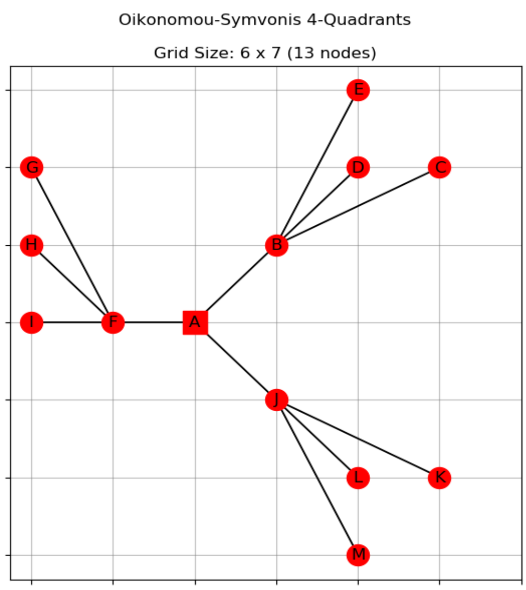


Εικόνα 4.14. Τροποποίηση αισθητικών λεπτομερειών

Παρακάτω παραθέτουμε ενδεικτικά κάποια από τα αποτελέσματα.



Εικόνα 4.15. Τροποποίηση αισθητικών παραμέτρων, αλγόριθμος Angelini et al. BFS-based



Εικόνα 4.16. Τροποποίηση αισθητικών παραμέτρων, αλγόριθμος Οικονόμου-Συμβώνης 4 Quadrants

5. Πειραματική αξιολόγηση των αλγορίθμων

Σε αυτό το κεφάλαιο, θα εξετάσουμε τις μεθόδους που ακολουθήσαμε για την πειραματική μελέτη των αλγορίθμων. Αρχικά θα περιγράψουμε τη διαδικασία για την παραγωγή και την αποθήκευση των πειραματικών δεδομένων σε βάση δεδομένων καθώς και ένα εργαλείο που αναπτύξαμε για την άντληση δεδομένων από αυτή. Στη συνέχεια θα κάνουμε μια συγκεντρωτική ανάλυση των αποτελεσμάτων και έπειτα θα τους εξετάσουμε ξεχωριστά για να προβούμε σε συμπεράσματα για καθέναν από αυτούς.

Το βασικό γνώρισμα των αλγορίθμων είναι το μέγεθος πλέγματος που καταλαμβάνουν οι απεικονίσεις τους, καθώς εξαρχής αναφέρθηκε ότι ένα από τα βασικά ερευνητικά ζητήματα των μονότονων απεικονίσεων, είναι ο περιορισμός του. Για την αξιολόγηση των αλγορίθμων, θα εξετάσουμε το μέγιστο συνολικό ποσοστό των θεωρητικών διαστάσεων που αξιοποίησε ο κάθε αλγόριθμος, για όλο το σύνολο των πειραματικών δεδομένων, και θα το χρησιμοποιήσουμε ως ένα μέτρο καθορισμού της επίδοσής του.

5.1 Παραγωγή και αποθήκευση πειραματικών δεδομένων

Για την παραγωγή των δένδρων στα οποία θα εφαρμοστούν οι αλγόριθμοι, χρησιμοποιούνται οι μέθοδοι που παρουσιάσαμε στο κεφάλαιο 3, για εξαντλητική και για τυχαία παραγωγή δένδρων με n κόμβους. Η εξαντλητική παραγωγή δένδρων, όπως δείξαμε, είναι υπολογιστικά μια ιδιαίτερα κοστοβόρα διαδικασία, καθώς το πλήθος των δένδρων αυξάνεται εκθετικά ως προς το μέγεθός τους. Έτσι, επιλέξαμε να παράξουμε εξαντλητικά, όλα τα ριζωμένα δένδρα με 2 έως και 15 κόμβους. Παρακάτω αναφέρουμε το πλήθος όλων των δυνατών δένδρων, για αυτές τις τιμές .

Πλήθος κόμβων (n)	Δυνατά δένδρα
2	1
3	2
4	5
5	14
6	42
7	132
8	429
9	1430
10	4862
11	16796
12	58786
13	208012
14	742900
15	2674440

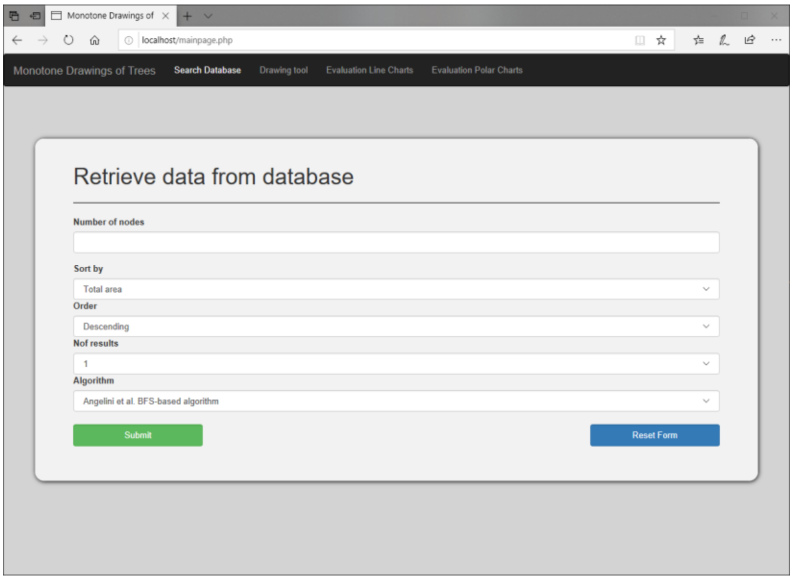
Πίνακας 5.1 Πλήθος όλων των διατεταγμένων δένδρων με ρίζα και n κόμβους

Εκτός από την εξαντλητική παραγωγή δένδρων, παράγουμε επίσης ένα σύνολο από 50.000 τυχαία δένδρα για μεγαλύτερες τιμές του n. Συγκεκριμένα, θα παράξουμε τυχαία δένδρα για n=20,25,30,35,40,45,50. Επίσης, προσθέτουμε χειροκίνητα το γράφημα μονοπάτι καθώς και το γράφημα-αστέρι (δένδρο με n-1 φύλλα), διότι όπως παρατηρήθηκε, αυτά τα δύο παράγουν, για αρκετούς αλγορίθμους, τις απεικονίσεις που χρησιμοποιούν το μεγαλύτερο και ελάχιστο μέγεθος πλέγματος αντίστοιχα.

Στη συνέχεια εφαρμόζουμε τους αλγορίθμους στα δένδρα που παράξαμε και αποθηκεύουμε τα αποτελέσματά τους σε σχεσιακή βάση δεδομένων MariaDB. Το μεγαλύτερο μέρος της διαδικασίας υλοποιήθηκε μέσω της γλώσσας Python 3.6, μέσω της οποίας γίνεται η παραγωγή των δένδρων, η εφαρμογή των αλγορίθμων σε αυτά καθώς και η σύνδεση με τη βάση δεδομένων για την καταχώρηση και ανάλυση των αποτελεσμάτων. Εφαρμόζοντας κάθε αλγόριθμο, καταχωρούμε στη βάση δεδομένων τη συμβολοσειρά από παρενθέσεις του εκάστοτε δένδρου, το συνολικό εμβαδό που καταλαμβάνει η απεικόνισή του, και τις δύο μέγιστες τιμές του πλάτους και του ύψους της, δηλαδή των αξόνων x και y αντίστοιχα.

5.1.1 Υλοποίηση εργαλείου άντλησης πληροφοριών

Για να έχουμε καλύτερη εποπτεία των δεδομένων που παράξαμε, υλοποιήσαμε ένα εργαλείο το οποίο συνδέεται με τη βάση δεδομένων και αντλεί και επιστρέφει τις αποθηκευμένες καταχωρήσεις. Επιπλέον, δίνεται η δυνατότητα οπτικοποίησης του κάθε δένδρου που εμπεριέχεται στα αποτελέσματα ενός επερωτήματος. Χρησιμοποιώντας το, μπορούμε εύκολα να δούμε οπτικά ποιά είναι τα δένδρα που χρησιμοποίησαν τα μεγαλύτερα και τα μικρότερα μεγέθη πλέγματος των απεικονίσεων των αλγορίθμων.



Εικόνα 5.1 Η διεπιφάνεια χρήστη του εργαλείου άντλησης πληροφοριών

Αρχικά φαίνονται τα πεδία στα οποία καθορίζονται οι διαθέσιμες επιλογές για τον καθορισμό των αποτελεσμάτων που θέλουμε να επιστραφούν (Εικόνα 5.1). Αυτά είναι τα εξής :

Number of nodes: Συνολικός αριθμός των κόμβων του δένδρου

Sort by: Χαρακτηριστικό της απεικόνισης, διαθέσιμες επιλογές: συνολικό εμβαδό πλέγματος (Total area), μέγιστο πλάτος (max x), μέγιστο ύψος (max y)

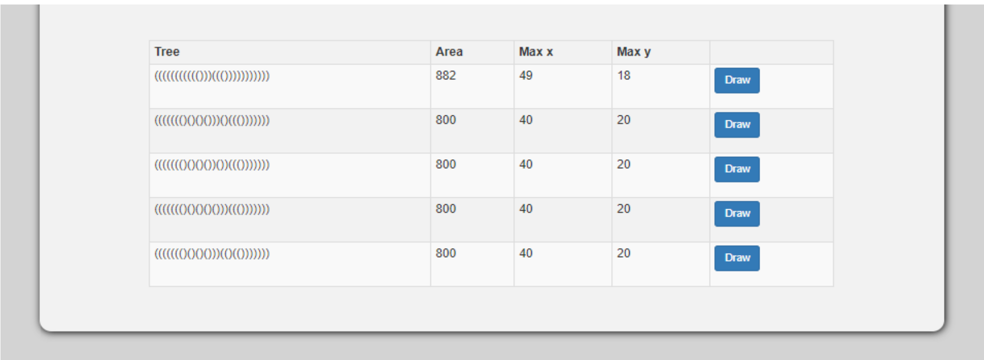
Order: Σειρά κατάταξης των αποτελεσμάτων, επιλογή μεταξύ αύξουσας (ascending) ή φθίνουσας (descending).

Nof results: Πλήθος αποτελεσμάτων που θα επιστραφούν.

Algorithm: Ο αλγόριθμος που χρησιμοποιήθηκε

Για να επιδείξουμε τη λειτουργία του εργαλείου, θα το χρησιμοποιήσουμε για να δούμε τα 5 πρώτα δένδρα με 15 κόμβους των οποίων οι απεικονίσεις καταλαμβάνουν το μεγαλύτερο και το μικρότερο εμβαδό, χρησιμοποιώντας τον αλγόριθμο των He&He Optimal-Draw.

Αφού καταχωρήσουμε τις αντίστοιχες επιλογές στα διαθέσιμα πεδία, επιλέγουμε το κουμπί '**Submit**'**.** Τα αποτελέσματα εμφανίζονται από κάτω και όπως φαίνεται αναγράφονται, η συμβολοσειρά από παρενθέσεις του αντίστοιχου δένδρου, το συνολικό εμβαδό που καταλαμβάνει η απεικόνιση καθώς και οι διαστάσεις του πλάτους και του ύψους (x,y). (Εικόνα 5.2)

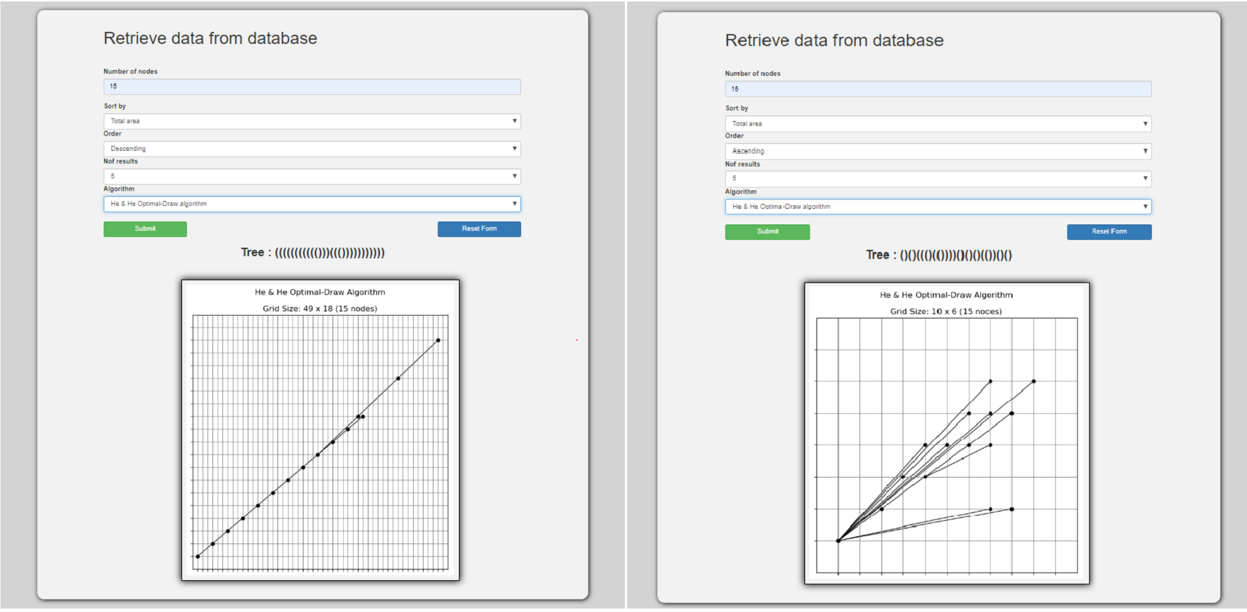


Tree	Area	Max x	Max y	
((((((((((()))((())))))))))))	882	49	18	Draw
(((((((()()()()))()((())))))))	800	40	20	Draw
(((((((()()()())())((())))))))	800	40	20	Draw
(((((((()()()()()))((())))))))	800	40	20	Draw
((((((()()()()))(()((())))))))	800	40	20	Draw

Εικόνα 5.2 Εργαλείο άντλησης πληροφοριών, αποτελέσματα επερωτήματος

Επιπλέον, στο τέλος κάθε αποτελέσματος της Εικόνας 5.3, εμφανίζεται το κουμπί '**Draw**', μέσω του οποίου μπορούμε να δούμε την οπτικοποίηση της μονότονης απεικόνισής του.

Στην Εικόνα 5.3 απεικονίζονται τα δύο δένδρα που παρήγαγαν το μεγαλύτερο και το μικρότερο συνολικό εμβαδό του αλγορίθμου.



Εικόνα 5.3 Εργαλείο άντλησης πληροφοριών, οπτικοποίηση αποτελεσμάτων (1)

5.2 Περιγραφική ανάλυση των δεδομένων

Όπως είδαμε, για κάθε αλγόριθμο καταχωρήθηκαν στη βάση δεδομένων τα γεωμετρικά χαρακτηριστικά των απεικονίσεων που παρήγαγε, δηλαδή το συνολικό εμβαδό και οι δύο επιμέρους διαστάσεις του πλέγματος. Για κάθε ένα από αυτά, υπολογίσαμε τις μέγιστες και τις ελάχιστες τιμές που προέκυψαν από τα δεδομένα. Στη συνέχεια, θα παρουσιάσουμε τα αποτελέσματα αυτά συγκεντρωτικά για όλους τους αλγορίθμους.

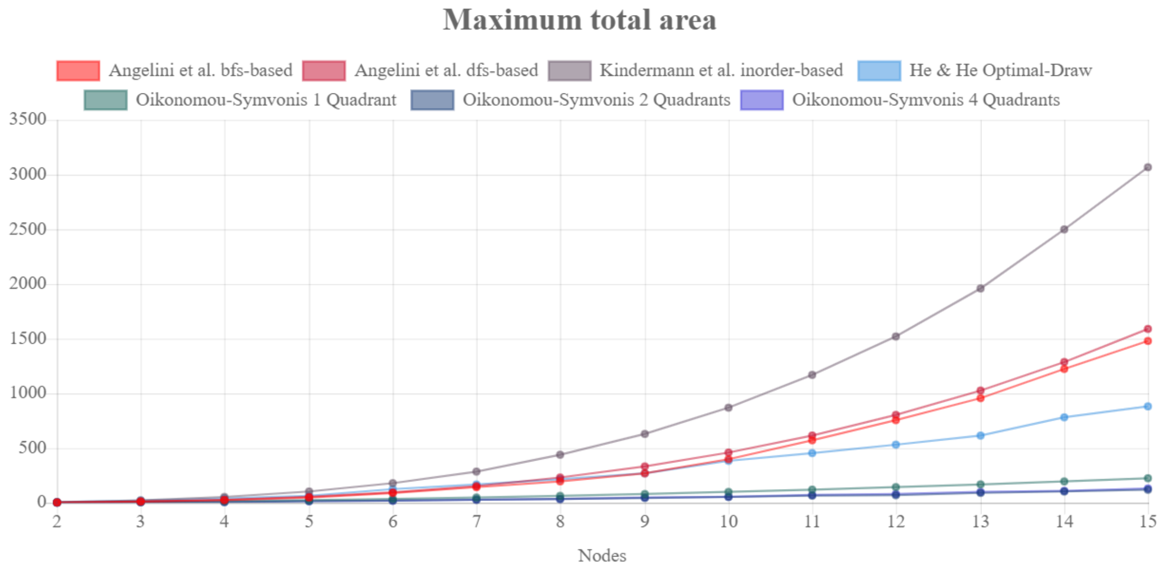
Τα δεδομένα που θα παρουσιάσουμε αφορούν τα δένδρα που παράξαμε εξαντλητικά, δηλαδή με 2 έως και 15 κόμβους, για τα οποία έχουμε πλήρη εικόνα.

Τα επόμενα διαγράμματα παράχθηκαν χρησιμοποιώντας τη βιβλιοθήκη της JavaScript, Chart.js.

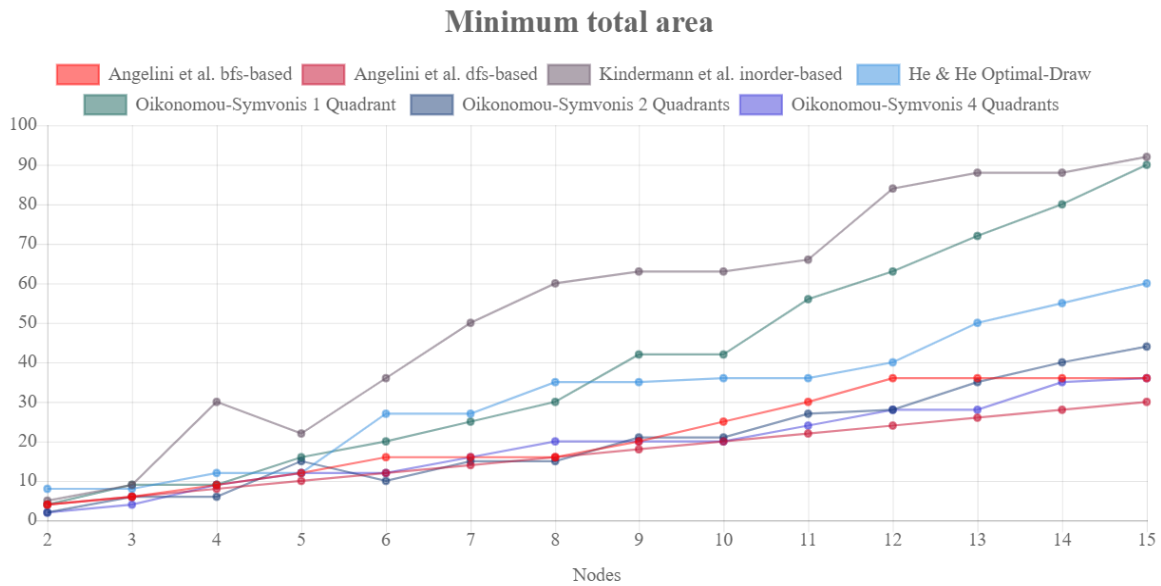
<u>Συνολικό εμβαδό απεικονίσεων</u>

Στο Διάγραμμα 5.1 φαίνονται τα μέγιστα εμβαδά που χρησιμοποίησαν οι αλγόριθμοι για τις απεικονίσεις τους. Τα αποτελέσματα, είναι γενικά με τη σειρά κατάταξης που αναμέναμε από τα θεωρητικά μεγέθη πλέγματος των αλγορίθμων. Ο τρίτος αλγόριθμος των Οικονόμου-Συμβώνη χρησιμοποιεί το μικρότερο εμβαδό πλέγματος .

Στο Διάγραμμα 5.2 που περιέχει τις ελάχιστες τιμές του εμβαδού, φαίνεται ότι το ελάχιστο αυτών χρησιμοποιήθηκε από τον αλγόριθμο DFS-based των Angelini et al. και τον τρίτο αλγόριθμο των Οικονόμου-Συμβώνη. Εδώ η σειρά κατάταξης των αλγορίθμων δεν είναι η ίδια όπως στις μέγιστες.



Διάγραμμα 5.1 Μέγιστο εμβαδό, 2-15 κόμβοι

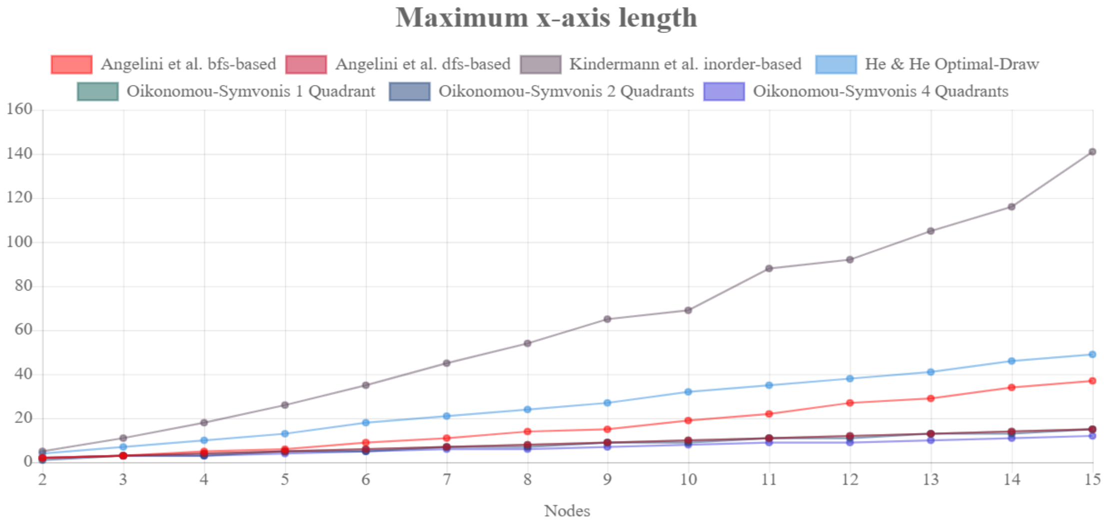


Διάγραμμα 5.2 Ελάχιστο εμβαδό, 2-15 κόμβοι

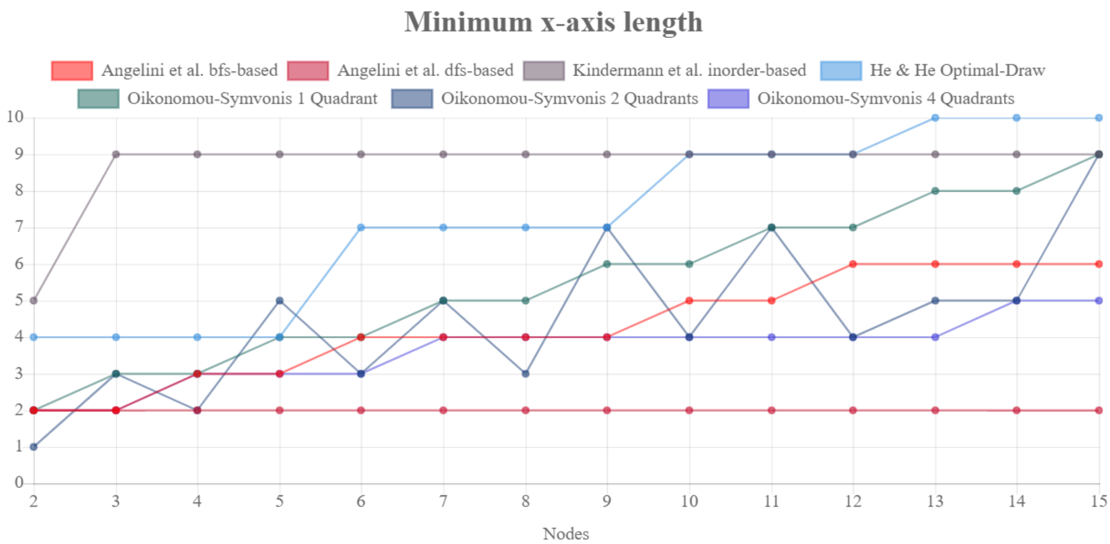
Πλάτος απεικονίσεων (άξονας x)

Όπως φαίνεται στο Διάγραμμα 5.3 που αφορά τα μέγιστα πλάτη, οι τρεις αλγόριθμοι των Οικονόμου-Συμβώνη και ο αλγόριθμος DFS-based των Angelini et al. χρησιμοποιούν το μικρότερο πλάτος συγκριτικά με τους υπολοίπους, έχοντας παρόμοια αποτελέσματα μεταξύ τους. Το μικρότερο από αυτά, καταλαμβάνει ο τρίτος αλγόριθμος των Οικονόμου-Συμβώνη 4-Quadrants.

Όσων αφορά τις ελάχιστες τιμές του πλάτους, ο αλγόριθμος DFS-based χρησιμοποιεί το μικρότερο, το οποίο μάλιστα παραμένει σταθερό για όλους τους κόμβους, όπως φαίνεται στο Διάγραμμα 5.4 .



Διάγραμμα 5.3 Μέγιστο πλάτος, 2-15 κόμβοι

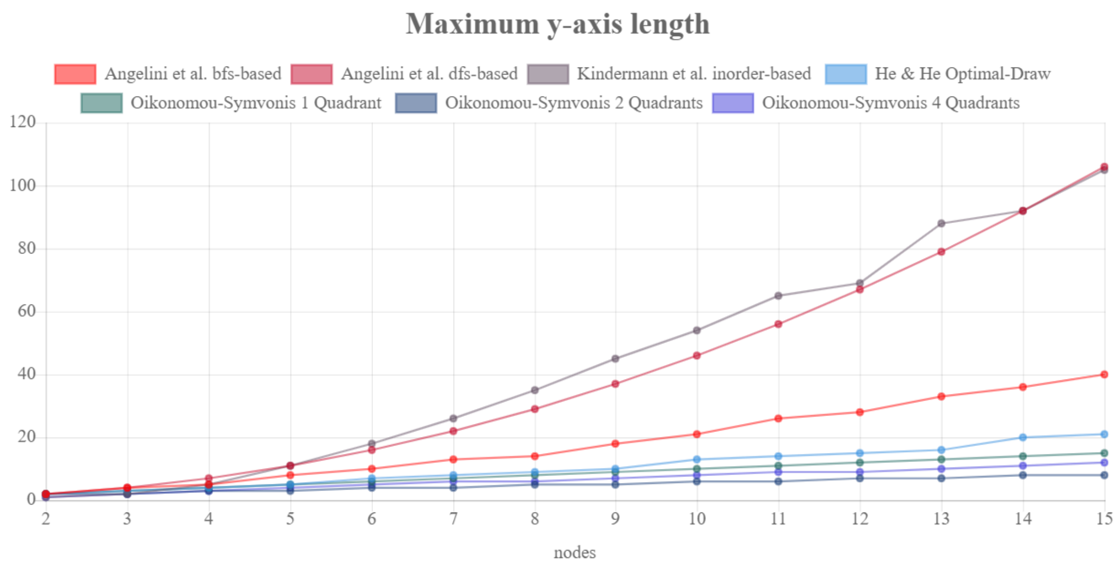


Διάγραμμα 5.4 Ελάχιστο πλάτος, 2-15 κόμβοι

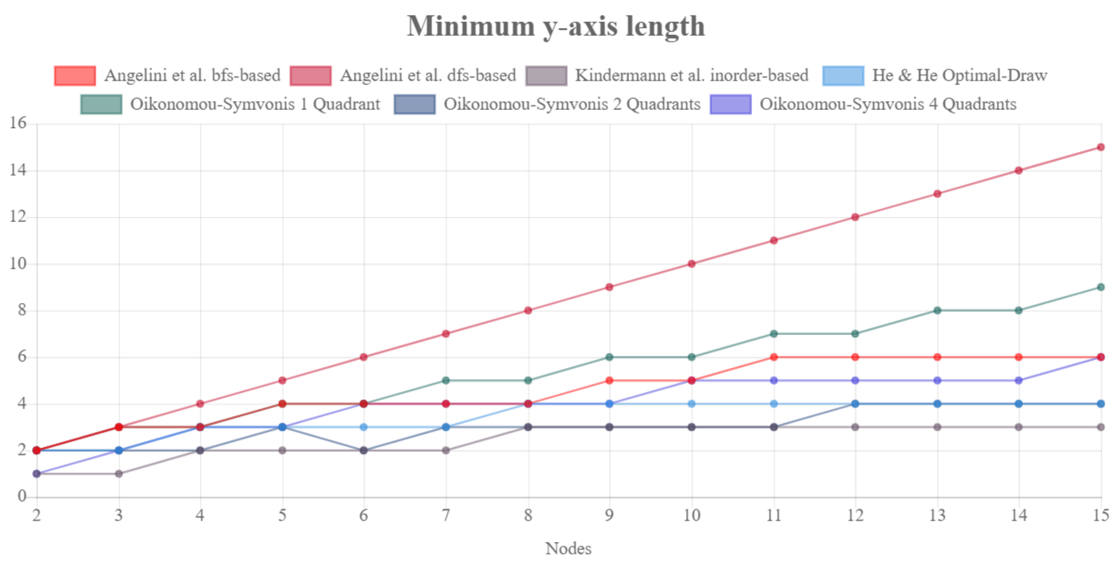
Ύψος απεικονίσεων (άξονας y)

Όπως φαίνεται στο Διάγραμμα 5.5 , ο αλγόριθμος δύο τεταρτημορίων των Οικονόμου-Συμβώνη χρησιμοποιεί το μικρότερο ύψος για τις απεικονίσεις που παράγει. Η σειρά κατάταξης των αλγορίθμων είναι παρόμοια με αυτή του μεγίστου εμβαδού.

Από τις ελάχιστες τιμές του ύψους, ο αλγόριθμος των Kindermnn et al. χρησιμοποιεί το μικρότερο για τις απεικονίσεις των δένδρων .



Διάγραμμα 5.5 Μέγιστο ύψος, 2-15 κόμβοι



Διάγραμμα 5.6 Ελάχιστο ύψος, 2-15 κόμβοι

5.3 Ανάλυση της επίδοσης των αλγορίθμων

Για τον προσδιορισμό της επίδοσης των αλγορίθμων, θα συγκρίνουμε το μέγιστο συνολικό εμβαδό των απεικονίσεων (total area) καθώς και το μέγιστο πλάτος (x) και ύψος (y), με τις αντίστοιχες θεωρητικές τιμές τους. Αυτό που μας ενδιαφέρει είναι να εξετάσουμε το κατά πόσο το μέγεθος των απεικονίσεων των πειραματικών δεδομένων, προσεγγίζει τις θεωρητικές διαστάσεις του κάθε αλγορίθμου.

Συγκεκριμένα, αναλύουμε, για κάθε έναν από αυτούς, όλα τα δεδομένα για όλα τα δένδρα που παράξαμε εξαντλητικά αλλά και τυχαία. Για κάθε πίνακα της βάσης δεδομένων, βρίσκουμε τις απεικονίσεις που απαίτησαν το μεγαλύτερο μέγεθος πλέγματος, συγκρίνουμε τις διαστάσεις τους με τις αντίστοιχες θεωρητικές τιμές και υπολογίζουμε το μέγιστο ποσοστό αυτών που αξιοποίησαν οι αλγόριθμοι. Στο τέλος, παίρνουμε από αυτά τα ποσοστά το μεγαλύτερο και αυτό θα αναφέρουμε ως ποσοστό επίτευξης των θεωρητικών διαστάσεων.

Στα επόμενα, θα παρουσιάσουμε τα αποτελέσματα για κάθε αλγόριθμο, προσθέτοντας και διαγράμματα όπου συγκρίνονται οι θεωρητικές και οι πειραματικές διαστάσεις των απεικονίσεων. Παρόλο που η ανάλυση έγινε εξετάζοντας όλο το σύνολο από δεδομένα που παράξαμε, στα διαγράμματα θα αναφερθούμε για τα δένδρα που παράξαμε εξαντλητικά, και συνεπώς έχουμε πλήρη εικόνα.

Τα διαγράμματα παράχθηκαν χρησιμοποιώντας τη βιβλιοθήκη Matplotlib της Python.

Αλγόριθμοι Angelini et al.

Τα θεωρητικά μεγέθη πλέγματος των δύο αλγορίθμων των Angelini et al. παρουσιάζονται στον Πίνακα 5.1. Ο αλγόριθμος BFS-based, σε σχέση με τα αυτά, αξιοποίησε συνολικά μέχρι και 54.41% του πλάτους, το 68.97% του ύψους και το 35.68% του συνολικού εμβαδού. Ο αλγόριθμος DFS-based, αξιοποίησε όλο το θεωρητικό πλάτος (100%) και σχεδόν το μισό θεωρητικό ύψος, 49.04%. (Πίνακας 5.2)

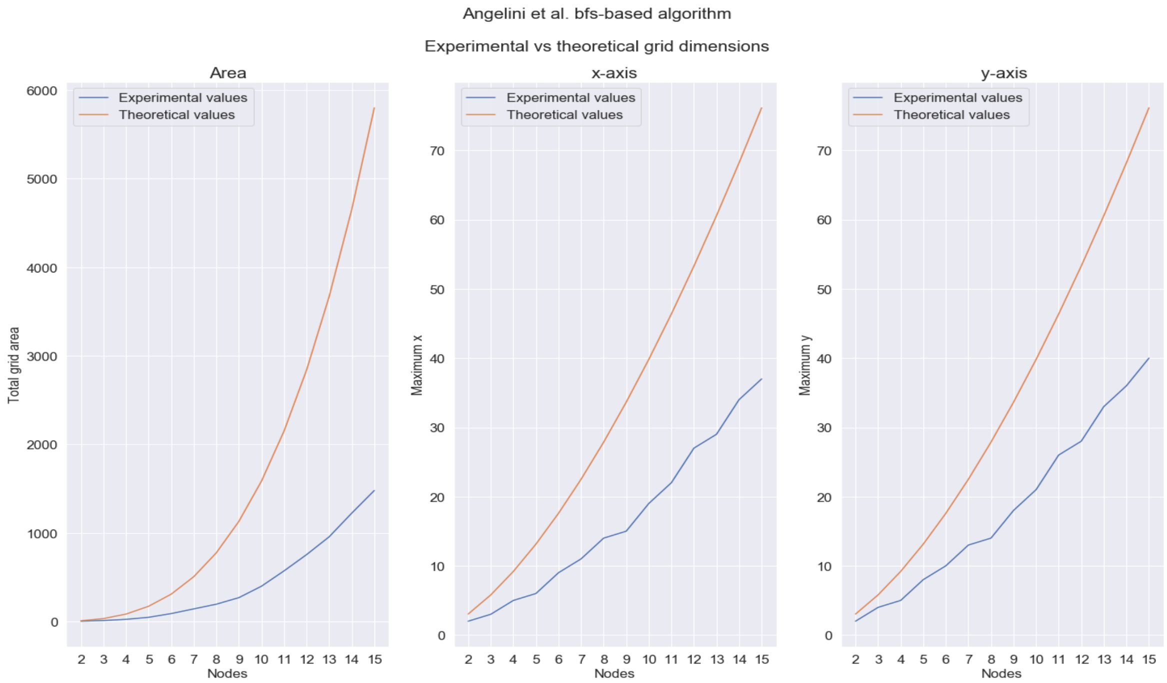
Για τον πρώτο αλγόριθμο παρατηρήσαμε οπτικά ότι οι απεικονίσεις του φαίνονται να έχουν σχετικά ομοιόμορφο πλέγμα, δηλαδή χωρίς η μία διάσταση να υπερβαίνει της άλλης, σε αντίθεση με τον αλγόριθμο DFS-based ο οποίο αξιοποιεί όλο το πλάτος και οι τελικές απεικονίσεις έχουν πολύ μεγάλο ύψος.

	BFS-based	DFS-based
Πλάτος	$O(n^{1.6})$	$O(n)$
Ύψος	$O(n^{1.6})$	$O(n^2)$
Συνολικό εμβαδό	$O(n^{3.2})$	$O(n^3)$

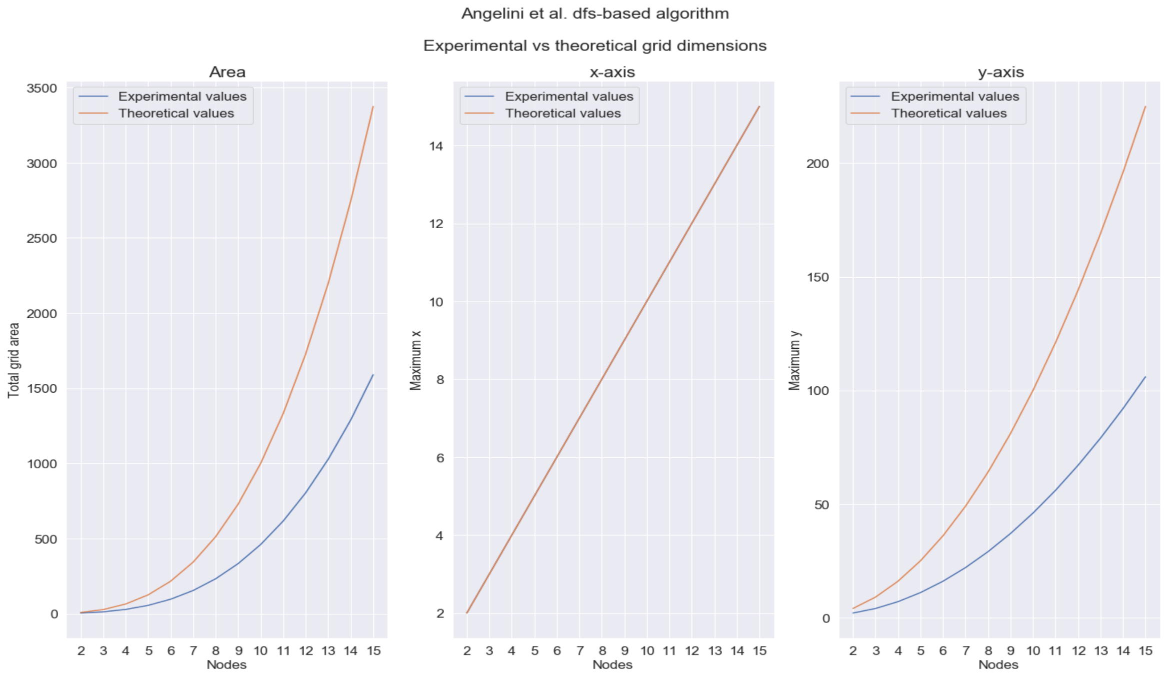
Πίνακας 5.1 Θεωρητικές διαστάσεις πλέγματος αλγορίθμων Angelini et al.

	BFS-based	DFS-based
Πλάτος	54.41%	100%
Ύψος	68.97%	49.04%
Συνολικό εμβαδό	35.68%	49.04%

Πίνακας 5.2 Ποσοστά επίτευξης θεωρητικών διαστάσεων πλέγματος, αλγόριθμοι Angelini et al.



Διάγραμμα 5.7. Angelini et al. αλγόριθμος BFS-based, πειραματικά δεδομένα-θεωρητικές τιμές



Διάγραμμα 5.8. Angelini et al. αλγόριθμος DFS-based, πειραματικά δεδομένα-θεωρητικές τιμές

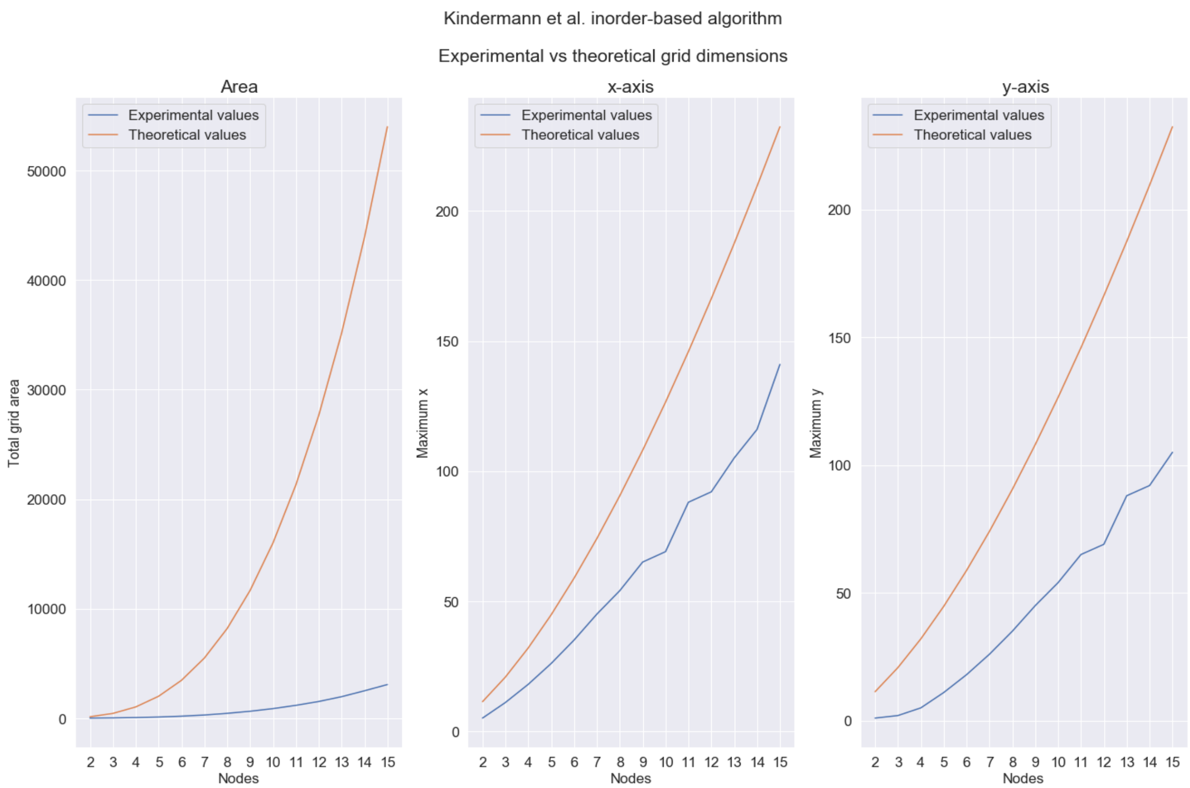
Αλγόριθμος Kindermann et al.

Ο αλγόριθμος των Kindermann et al. παράγει μονότονες απεικονίσεις σε πλέγμα μεγέθους $O(n^{1.5})$x$O(n^{1.5})$. Γνωρίζουμε όμως ότι το μέγεθος αυτό φράζεται από την ποσότητα $4n^{1.5}$ x $4n^{1.5}$ [2]. Συνεπώς, η σύγκριση των αποτελεσμάτων θα γίνει χρησιμοποιώντας αυτό ως μέγεθος πλάτους και ύψους και το γινόμενό τους $16n^3$ ως το συνολικό εμβαδό.

Πειραματικά προέκυψε ότι ο αλγόριθμος αξιοποίησε το 60.06% του πλάτους, το 46.81% του ύψους και το 5.09% του θεωρητικού εμβαδού. (Πίνακας 5.3). Παρατηρούμε ότι παρόλο που χρησιμοποιεί αρκετό μέρος και των δύο διαστάσεων, αυτό δεν συμβαίνει ταυτόχρονα, γι'αυτό και το συνολικό εμβαδό είναι μικρό.

Inorder-based	Θεωρητικές διαστάσεις	Ποσοστό κάλυψης
Πλάτος	$O(n^{1.5})$	60.06%
Ύψος	$O(n^{1.5})$	46.81%
Συνολικό εμβαδό	$O(n^3)$	5.09%

Πίνακας 5.3 Θεωρητικές διαστάσεις και ποσοστά κάλυψης αλγορίθμου Kindermann et al.



Διάγραμμα 5.9. Kindermann et al. inorder, πειραματικά δεδομένα και θεωρητικές τιμές.

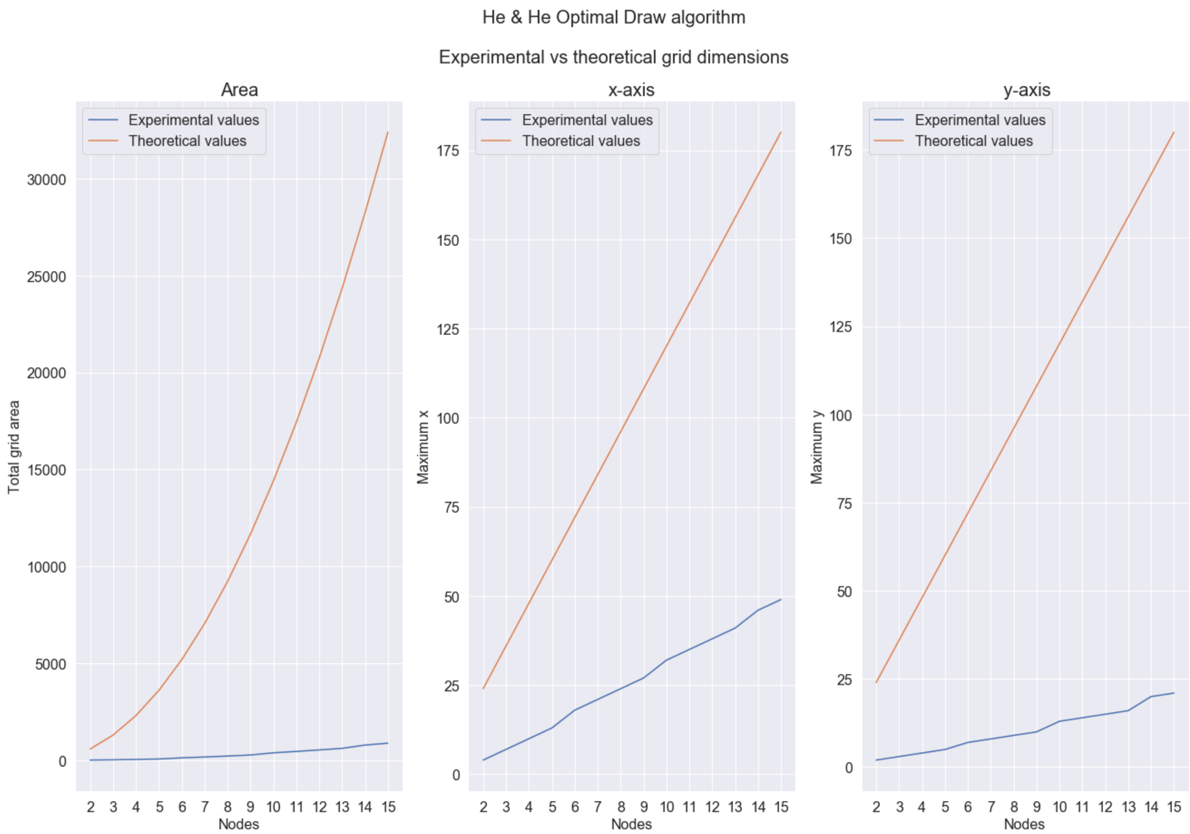
Αλγόριθμος D.He -X.He

Ο αλγόριθμος Optimal-Draw, πειραματικά, χρησιμοποίησε το 28.75% του πλάτους, το 12.92% του ύψους και μόλις το 2.96% του θεωρητικού εμβαδού. (Πίνακας 5.4).

Σε αυτόν τον αλγόριθμο φαίνεται να υπάρχει η μεγαλύτερη απόκλιση από τις θεωρητικές τιμές, τόσο για το εμβαδό όσο και για της επιμέρους διαστάσεις.

Optimal draw	Θεωρητικές διαστάσεις	Ποσοστό κάλυψης
Πλάτος	$12n$	28.75%
Ύψος	$12n$	12.92%
Συνολικό εμβαδό	$144n^2$	2.96%

Πίνακας 5.4 Θεωρητικές διαστάσεις και ποσοστά κάλυψης αλγορίθμου He&He



Διάγραμμα 5.10. He & He Optimal draw αλγόριθμος, πειραματικά δεδομένα και θεωρητικές τιμές.

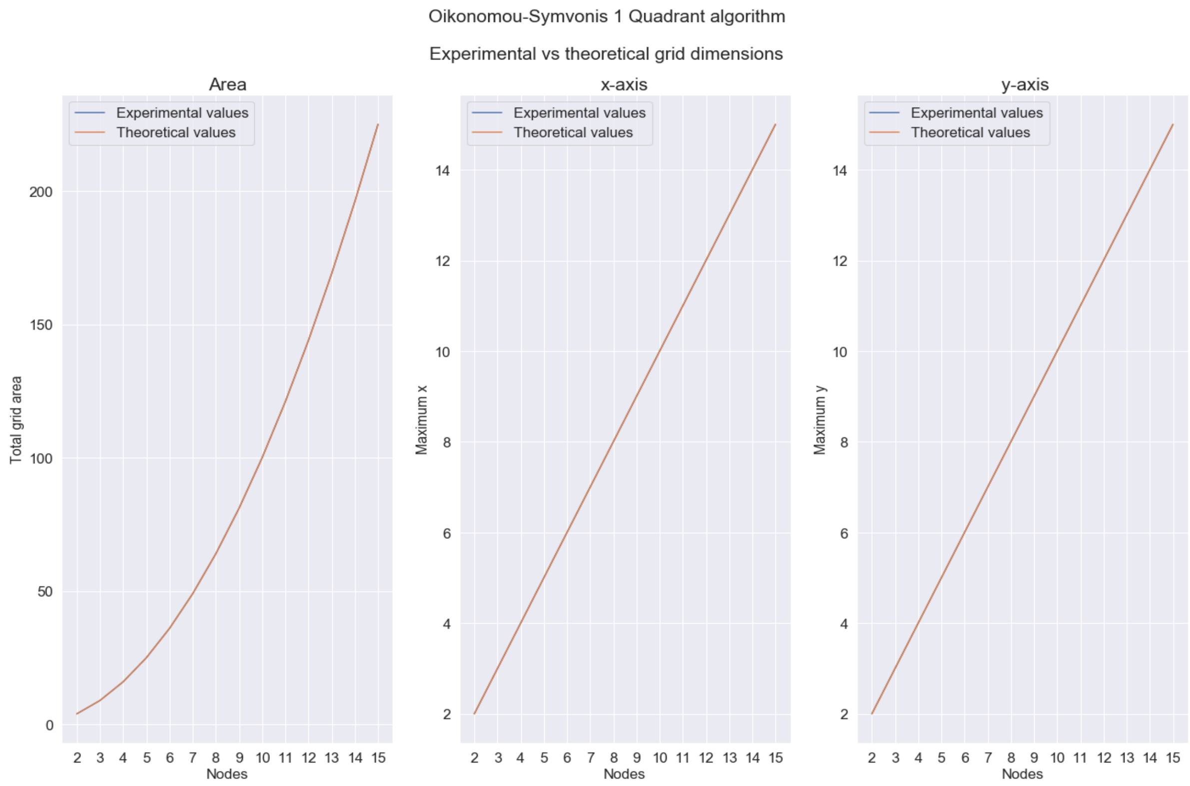
<u>Αλγόριθμοι Οικονόμου-Συμβώνης</u>

Αλγόριθμος απεικόνισης σε ένα τεταρτημόριο (1 Quadrant)

Όπως προκύπτει από τα αποτελέσματα, ο αλγόριθμος χρησιμοποιεί το 100% του θεωρητικού συνολικού εμβαδού και των δύο διαστάσεών του. Στο Διάγραμμα 5.10 βλέπουμε ότι οι θεωρητικές και πειραματικές ευθείες επικαλύπτονται καθώς ταυτίζονται, κάτι που σημαίνει ότι ο αλγόριθμος καταλαμβάνει τις μέγιστες θεωρητικές του διαστάσεις, για όλα τα μεγέθη δένδρων που εξετάσαμε.

1 Quadrant	Θεωρητικές διαστάσεις	Ποσοστό κάλυψης
Πλάτος	n	100%
Ύψος	n	100%
Συνολικό εμβαδό	n^2	100%

Πίνακας 5.5 Θεωρητικές διαστάσεις και ποσοστά κάλυψης αλγορίθμου Οικονόμου-Συμβώνης- 1 Quadrant



Διάγραμμα 5.10. Αλγόριθμος Οικονόμου-Συμβώνης 1 Quadrant, πειραματικά δεδομένα και θεωρητικές τιμές

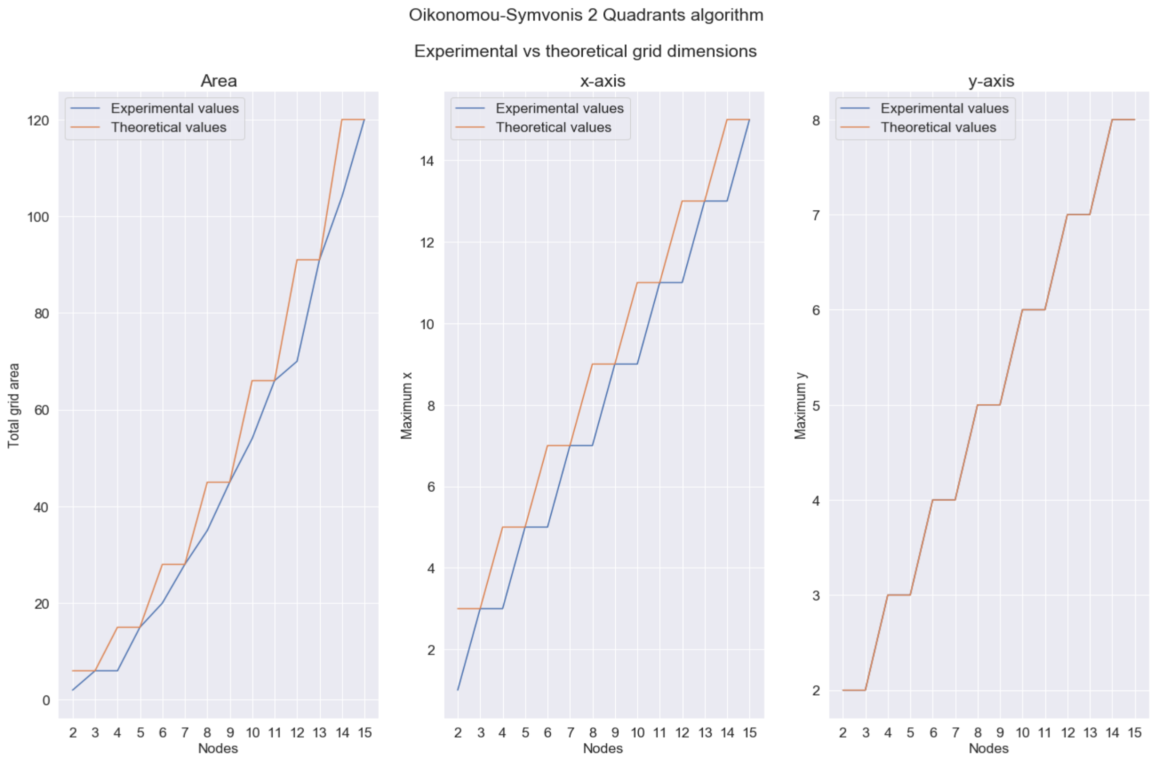
Αλγόριθμος απεικόνισης σε δύο τεταρτημόρια (2 Quadrants)

Οι θεωρητικές τιμές του αλγορίθμου που φαίνονται στον Πίνακα 5.6 είναι οι γενικές που αναφέρει ο αλγόριθμος, αλλά αναλύονται περεταίρω. Όταν το μέγεθος του δένδρου n είναι περιττός αριθμός, το θεωρητικό μέγεθος πλέγματος είναι $n\ x\ \frac{n+1}{2}$, ενώ όταν πρόκειται για άρτιο αριθμό είναι $n + 1\ x\ \frac{n}{2} + 1$.

Όπως και ο προηγούμενος αλγόριθμος , έτσι και αυτός χρησιμοποιεί όλο το θεωρητικό μέγεθος πλέγματος (Πίνακας 5.6). Στο Διάγραμμα 5.11, φαίνεται ότι επιτυγχάνει τις μέγιστες θεωρητικές διαστάσεις πλάτους ανά διαστήματα και όχι συνεχώς, όπως ο πρώτος αλγόριθμος. Μάλιστα, αυτό γίνεται για τα δένδρα με περιττό πλήθος κόμβων. Αντίθετα, για τον άξονα y έχουμε πλήρη ταύτιση θεωρητικών και πειραματικών τιμών για όλα τα μεγέθη.

2 Quadrants	Θεωρητικές διαστάσεις	Ποσοστό κάλυψης
Πλάτος	n	100%
Ύψος	$n/2$	100%
Συνολικό εμβαδό	$n^2/2$	100%

Πίνακας 5.6 Θεωρητικές διαστάσεις και ποσοστά κάλυψης του αλγορίθμου Οικονόμου-Συμβώνης(2 Quadrants)



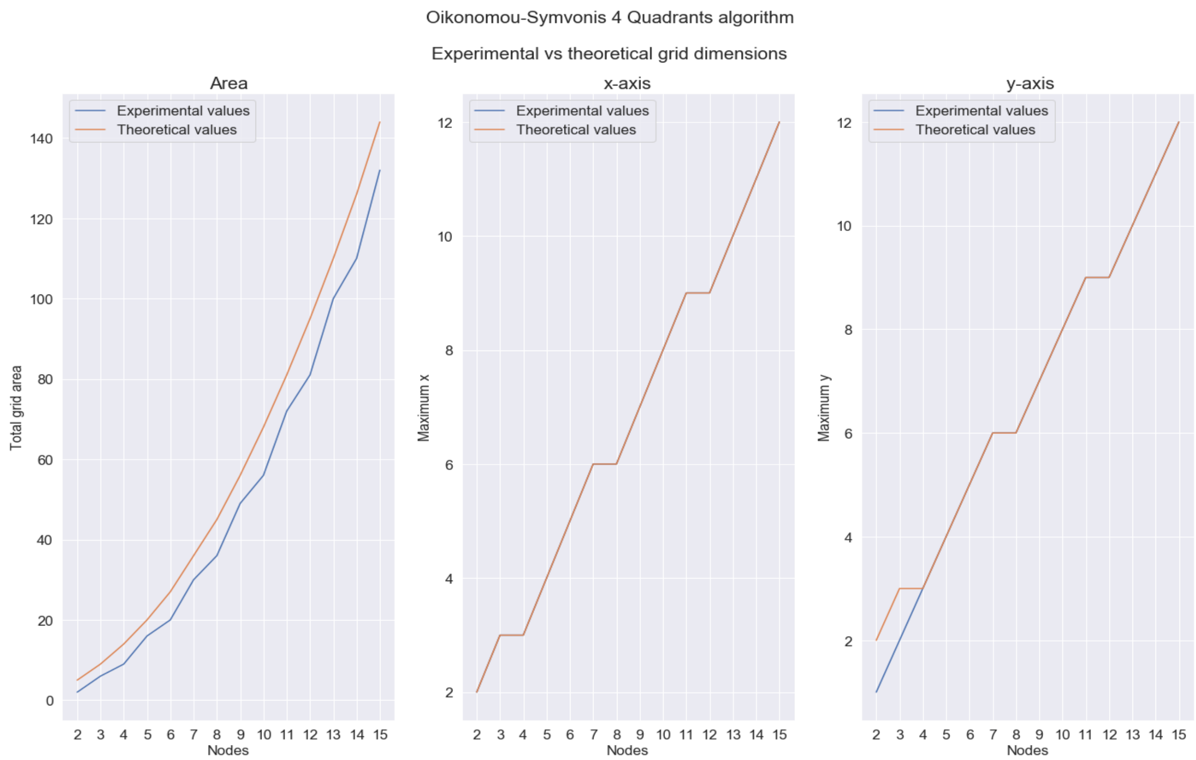
Διάγραμμα 5.11. Αλγόριθμος Οικονόμου-Συμβώνης 2 Quadrant, πειραματικά δεδομένα και θεωρητικές τιμές

Αλγόριθμος απεικόνισης σε τέσσερα τεταρτημόρια (4 Quadrants)

Ο τρίτος αλγόριθμος των Οικονόμου-Συμβώνη αξιοποίησε και αυτός, όπως και οι δύο προηγούμενοι, όλο το θεωρητικό μέγεθος πλέγματος των αξόνων x και y (100%). Όσων αφορά το συνολικό εμβαδό, υπάρχει μια μικρή μόνο απόκλιση από την πλήρη κάλυψη, καθώς χρησιμοποίησε το 91.67% του αντίστοιχου θεωρητικού. Όπως και ο αλγόριθμος των δύο τεταρτημορίων, υπάρχει σχεδόν πλήρης ταύτιση των πειραματικών αποτελεσμάτων με τις θεωρητικές τιμές, καί για τους δύο άξονες (Διάγραμμα 5.13).

4 Quadrants	Θεωρητικές διαστάσεις	Ποσοστό κάλυψης
Πλάτος	$\left\lfloor \frac{3}{4}(n+1) \right\rfloor$	100%
Ύψος	$\left\lfloor \frac{3}{4}(n+1) \right\rfloor$	100%
Συνολικό εμβαδό	$\left\lfloor \frac{9}{16}(n+1)^2 \right\rfloor$	91.67%

Πίνακας 5.7 Θεωρητικές διαστάσεις πλέγματος του αλγορίθμου Οικονόμου-Συμβώνης(4 Quadrants)



Διάγραμμα 5.13. Αλγόριθμος Οικονόμου-Συμβώνης 4 Quadrants , πειραματικά δεδομένα και θεωρητικές τιμές

`

6. Συμπεράσματα

Στην παρούσα διπλωματική εργασία υλοποιήσαμε και αναλύσαμε πειραματικά, αλγορίθμους παραγωγής μονότονων απεικονίσεων γραφημάτων και συγκεκριμένα μιας ειδικής κατηγορίας αυτών, τα δένδρα.

Στο Κεφάλαιο 3, χρησιμοποιώντας τους αριθμούς Catalan, δείξαμε ότι κάθε διατεταγμένο δένδρο με ρίζα μπορεί να αναπαρασταθεί μέσω ισορροπημένων συμβολοσειρών από παρενθέσεις. Το γεγονός αυτό αποδείχθηκε εξαιρετικά χρήσιμο, καθώς αποτελεί έναν εύχρηστο και σύντομο τρόπο αναφοράς σε οποιοδήποτε τέτοιο δένδρο. Επίσης, χρησιμοποιήθηκε ως βάση για τις διαδικασίες παραγωγής δένδρων που περιγράψαμε, δηλαδή για την εξαντλητική παραγωγή όλων των δυνατών δένδρων με ρίζα και n κόμβους αλλά και για την τυχαία παραγωγή ενός δένδρου.

Στο Κεφάλαιο 4, υλοποιήσαμε κάποιους επιλεγμένους αλγορίθμους για την παραγωγή μονότονων απεικονίσεων δένδρων και αναπτύξαμε ένα διαδραστικό εργαλείο, μέσω του οποίου μπορούμε να τους εφαρμόσουμε σε δένδρα που επιλέγονται από τον χρήστη και να δούμε τις απεικονίσεις που παράγουν. Όπως απεδείχθη, η οπτική μελέτη των αλγορίθμων μπορεί να οδηγήσει σε σημαντικά ευρήματα για μελλοντικές μελέτες και βελτιώσεις τους.

Στο Κεφάλαιο 5 έγινε η πειραματική μελέτη και αξιολόγηση των αλγορίθμων. Από την ανάλυση των πειραματικών δεδομένων, παρατηρήσαμε ότι υπάρχουν αλγόριθμοι που χρησιμοποιούν όλο το θεωρητικό μέγεθος πλέγματος ενώ άλλοι ένα ποσοστό αυτού. Όπως προέκυψε, μόνο οι τρείς αλγόριθμοι των Οικονόμου-Συμβώνη χρησιμοποιούν όλο το θεωρητικό μέγεθος πλέγματος για τις απεικονίσεις τους. Οι δύο αλγόριθμοι των Angelini et al., BFS-based και DFS-based, χρησιμοποίησαν περίπου το ένα τρίτο και το μισό, αντίστοιχα, των θεωρητικών τους διαστάσεων, ενώ ο αλγόριθμος των Kindermann et al. και ο αλγόριθμος των D.He και X.He, αξιοποίησαν αρκετά μικρά ποσοστά αυτών. Πρέπει να επισημάνουμε ότι οι δύο τελευταίοι αλγόριθμοι απαιτούν σύνθετες μαθηματικές διαδικασίες οι οποίες δεν έχουν μοναδικές λύσεις, και εξετάζοντάς τες περεταίρω μπορούν να παράξουν καλύτερα αποτελέσματα. Στην προσπάθεια που έγινε για την υλοποίησή τους, ασχοληθήκαμε με απλούς τρόπους προσέγγισης αυτών των ζητημάτων αλλά θα είχε ενδιαφέρον να ασχοληθεί κανείς εις βάθος με αυτά. Επίσης, όπως παρατηρήσαμε, όσο προσθέταμε νέα δεδομένα, οι αλγόριθμοι αξιοποιούσαν όλο και μεγαλύτερο ποσοστό.

Εξετάζοντας οπτικά τις απεικονίσεις των αλγορίθμων, παρατηρήσαμε ότι αυτοί χρησιμοποιούν μικρό ποσοστό των θεωρητικών τους διαστάσεων, παρουσιάζουν μικρή γωνιακή ανάλυση (η μικρότερη γωνία μεταξύ δύο ακμών που συναντούν έναν κοινό κόμβο). Αυτό έχει ως αποτέλεσμα, το συνολικό γράφημα να φαίνεται αρκετά πυκνό. Αντίθετα, οι αλγόριθμοι που αξιοποίησαν μεγάλο ποσοστό των θεωρητικών διαστάσεων, σχηματίζουν μεγαλύτερη γωνιακή ανάλυση και οι απεικονίσεις τους είναι πιο αραιές. και οι ακμές του γραφήματος πιο ομοιόμορφα τοποθετημένες. Σε πρακτικές εφαρμογές, θα είχε ενδιαφέρον η χρήση και των δύο κατηγοριών αυτών, καθώς εάν χρησιμοποιηθούν παράλληλα, ίσως μπορούν να λειτουργήσουν ως ένα είδος μεγέθυνσης ή σμίκρυνσης του γραφήματος.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Patrizio Angelini, Enrico Colasante, Giuseppe Di Battista, Fabrizio Frati and Maurizio Patrignani, “Monotone Drawings of Graphs”, Journal of Graph Algorithms and Applications, vol. 16, no. 1, pp. 5–35, 2012

[2] Philipp Kindermann, André Schulz, Joachim Spoerhase and Alexander Wolff, “On Monotone Drawings of Trees”, 2014

[3] Dayu He and Xin He, “Optimal Monotone Drawings of Trees”, SIAM J. Discrete Math., vol. 31, no. 3, pp. 1867–1877, 2016

[4] Anargyros Oikonomou and Antonios Symvonis, “Simple Compact Monotone Tree Drawings”, CoRR, vol. abs/1708.09653, 2017

[5] Ανάργυρος Οικονόμου, “Απεικόνιση Γραφημάτων με Έμφαση στις Ιδιότητες Μονοπατιών”. Διπλωματική εργασία. , Ε.Μ.Π 2018

[6] W. Huang, P. Eades and S.H. Hong, A Graph Reading Behavior: Geodesic-Path Tendency, in Proceedings of IEEE Pacific Visualization Symposium, pp. 137–144, 2009

[7] E. M. Arkin, R. Connelly, and J. S. Mitchell. On monotone paths among obstacles with applications to planning assemblies. In SoCG ’89, pages 334–343, 1989.

[7] Wikipedia Catalan number https://en.wikipedia.org/wiki/Catalan_number

[8] Wikipedia Stern-Brocot tree https://en.wikipedia.org/wiki/Stern%E2%80%93Brocot_tree

[9] Wikipedia Farey sequence https://en.wikipedia.org/wiki/Farey_sequence

[10] Python 3.6 https://docs.python.org/3.6/

[11] Networkx Python package https://networkx.github.io/

[12] Matplotlib library https://matplotlib.org/

[13] JavaScript https://www.javascript.com/learn

[14] Chart.js library https://www.chartjs.org/docs/latest/

[15] XAMPP environment https://www.apachefriends.org/index.html

[16] Sublime text environment https://www.sublimetext.com/

[17] HTML5 https://developer.mozilla.org/en-US/docs/Web/Tutorials

[18] Bootstrap 3 https://getbootstrap.com/docs/3.3/

[19] CSS3 http://www.css3.info

[20] PHP https://www.php.net/manual/en/index.php