Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών
Τομέας Μαθηματικών

Απεικόνιση Επίπεδων Γραφημάτων σε Πλέγμα

Διπλωματική Εργασία

Διαμαντίδου Ιωάννα

Επιβλέπων: Αντώνιος Συμβώνης, Καθηγητής Ε.Μ.Π.

Σεπτέμβριος 2023

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών
Τομέας Μαθηματικών

Απεικόνιση Επίπεδων Γραφημάτων σε Πλέγμα

Διπλωματική Εργασία

Διαμαντίδου Ιωάννα

Επιβλέπων: Αντώνιος Συμβώνης, Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή:

.....	.....	.....
Αντώνιος Συμβώνης	Πέτρος Στεφανέας	Ιωάννης Κολέτσος
Καθηγητής Ε.Μ.Π.	Αν. Καθηγητής Ε.Μ.Π.	Αν. Καθηγητής Ε.Μ.Π.

Σεπτέμβριος 2023

Περίληψη

Η απεικόνιση γραφημάτων είναι ένας κλάδος των Μαθηματικών και της Επιστήμης Υπλογιστών, με τη βοήθεια του οποίου, τα αφηρημένα γραφήματα μεταλλάσσονται σε ευανάγνωστες και διαισθητικές αναπαραστάσεις.

Στην παρούσα διπλωματική εργασία, ασχολούμαστε με την απεικόνιση ευθύγραμμων επίπεδων γραφημάτων σε πλέγμα και ειδικότερα με τη χρήση της μεθόδου Shift.

Ο σκοπός της εργασίας ήταν η ανάπτυξη ενός λογισμικού που θα υλοποιεί τον αλγόριθμο Shift.
Το λογισμικό επιτρέπει στον χρήστη να δημιουργήσει ή να εισάγει οποιοδήποτε εσωτερικά τριγωνοποιημένο γράφημα, να παρακολουθήσει την διαδικασία της κανονικής διάταξης και του αλγορίθμου Shift βήμα-βήμα και τελικά να λάβει μια ΄καθαρή΄ επίπεδη απεικόνιση, εμφυτευμένη σε πλέγμα μεγέθους $(2n-4)\times(n-2)$.

Λέξεις Κλειδιά

Θεωρία Γραφημάτων, Απεικόνιση Γραφημάτων, Ευθύγραμμες Απεικονίσεις, Επίπεδα Γραφήματα, Απεικονίσεις σε πλέγμα

Abstract

Graph Drawing is an area of Mathematics and Computer Science, helping abstract graphs transform into easily interpretable and intuitive representations.

In this Diploma Thesis, we address straight-line drawings of planar graphs on the grid, created using the Shift algorithm.

The purpose of this thesis was the development of a software tool which implements the Shift algorithm.
The software allows the user to create or import any internally triangulated graph, to watch the procedure of the canonical ordering and the shift method implementation step-by-step, and eventually get a clear planar drawing, embedded on a $(2n-4) \times (n-2)$ grid.

Keywords

Graph Theory, Graph Drawing, Straight-line drawings, Planar Graphs, Grid Drawings

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα της διπλωματικής μου, καθηγητή κ. Αντώνιο Συμβώνη, για την καθοδήγηση του.

Την οικογένειά μου για την σταθερή και ανιδιοτελή στήριξη.

Τους φίλους, συμφοιτητές και όλους τους ανθρώπους με τους οποίους μοιραστήκαμε στιγμές κατά τη διάρκεια των φοιτητικών χρόνων.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

Η **Θεωρία Γραφημάτων** με τις ρίζες της στον 18ο αιώνα, αποτελεί ένα ζωτικής σημασίας πεδίο των μαθηματικών, με σημαντική συνεισφορά τόσο στα θεωρητικά μαθηματικά, όσο και σε πρακτικές εφαρμογές. Λέγεται ότι ξεκίνησε από τον Leonhard Euler και προέκυψε από την ενασχόληση του με το πρόβλημα των Επτά Γεφυρών του Königsberg [1]. Η σπουδαιότητά του πεδίου αυτού, φαίνεται στην ικανότητα του να αναλύει και να μοντελοποιεί πολύπλοκες σχέσεις και συνδέσεις σε πολλούς τομείς της πραγματικής ζωής, όπως η βελτιστοποίηση των δικτύων μεταφοράς, η μελέτη κοινωνικών δικτύων και η αποδοτική ροή δεδομένων στα δίκτυα ηλεκτρονικών υπολογιστών.
Στο κεφάλαιο αυτό, θα αναφέρουμε τις βασικές έννοιες της θεωρίας γραφημάτων καθώς και κάποιους ορισμούς που είναι απαραίτητοι για την κατανόηση των αλγορίθμων που θα αναλυθούν στα πλαίσια αυτής της διπλωματικής εργασίας.

1.1 Σκοπός της Διπλωματικής Εργασίας

Το κύριο αντικείμενο αυτής της διπλωματικής εργασίας, είναι η δημιουργία μιας εφαρμογής που υλοποιεί τη μέθοδο $Shift$ για την ευθύγραμμη απεικόνιση επίπεδων γραφημάτων σε πλέγμα.

Σύντομη περιγραφή της λειτουργικότητας της εφαρμογής

Στη διάρκεια της περιήγησης στην εφαρμογή, ο χρήστης έχει τη δυνατότητα είτε να ανακτήσει κάποιο αποθηκευμένο γράφημα σε μορφή $.graphml$, είτε να δημιουργήσει ένα γράφημα στο πλέγμα που υπάρχει στην αρχική σελίδα. Στη συνέχεια, με το πάτημα του Canonical Ordering, παρουσιάζεται η διαδικασία της διάταξης των κόμβων, η οποία μπορεί έπειτα να παρουσιαστεί βήμα-βήμα ώστε να μπορεί ο χρήστης να παρατηρήσει ακριβώς τι συμβαίνει. Έπειτα, με

το πάτημα του Shift Method, παρουσιάζεται και η διαδικασία της επαναδημιουργίας του γραφήματος πάνω στο πλέγμα, η οποία επίσης δύναται να παρουσιαστεί αναλυτικά. Έπειτα, στο νέο γράφημα, ο χρήστης έχει τη δυνατότητα να μελετήσει τις θέσεις των κόμβων κάνοντας κλικ σε αυτούς και το covering set του εκάστοτε, διαπερνώντας το ποντίκι από πάνω του.
Ο κύριος σκοπός της εφαρμογής είναι να βοηθήσει στην πλήρη κατανόηση του τρόπου λειτουργίας των δύο αυτών αλγορίθμων και της διαδικασίας δημιουργίας ενός τελικού 'όμορφου' γραφήματος, αλλά και η γρήγορη δημιουργία μιας τέτοιας απεικόνισης.

1.2 Χρήσιμες Έννοιες

Γράφημα

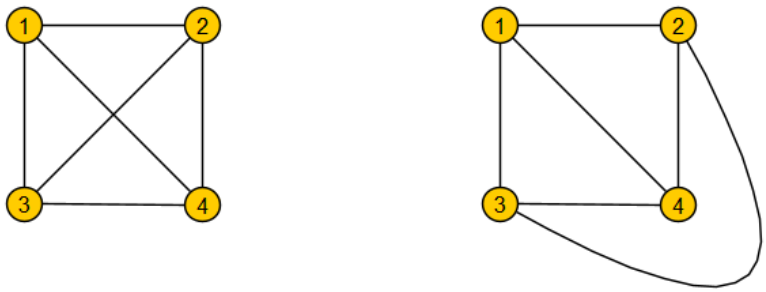
Ένα γράφημα είναι ένα ζευγάρι $G = (V, E)$ συνόλων τέτοιο ώστε $E \subseteq V \times V$. Τα στοιχεία του V είναι οι **κόμβοι** του G και τα στοιχεία του E είναι οι **ακμές** του G. Για δύο κόμβους v_1, v_2 του συνόλου V, που ενώνονται με ακμή e του συνόλου E, συμβολίζουμε $e = (v_1, v_2)$

Επίπεδο Γράφημα

Ένα γράφημα G(V,E) είναι επίπεδο, αν έχει μία τουλάχιστον **επίπεδη απεικόνιση**.

Επίπεδη απεικόνιση γραφήματος

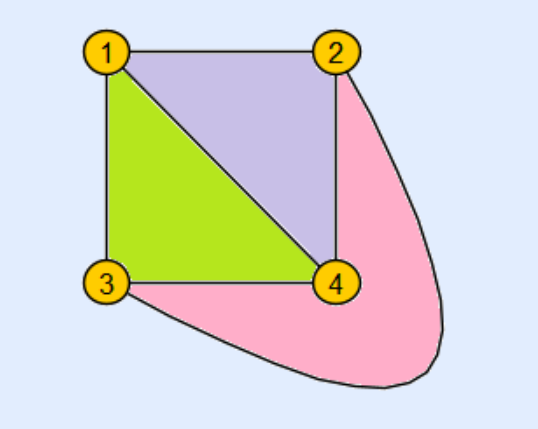
Μια απεικόνιση ενός γραφήματος είναι **επίπεδη** αν οι ακμές του δεν τέμνονται. Στην εικόνα 1.1 φαίνονται μία επίπεδη και μία μη-επίπεδη απεικόνιση για το ίδιο γράφημα.



Σχήμα 1.1: Μη-επίπεδη και επίπεδη απεικόνιση του ίδιου γραφήματος

Όψεις επίπεδου γραφήματος

Αν το γράφημα G(V,E) είναι επίπεδο, τότε οι ακμές του γραφήματος χωρίζουν το επίπεδο σε περιοχές, οι οποίες ονομάζονται **όψεις** του γραφήματος.



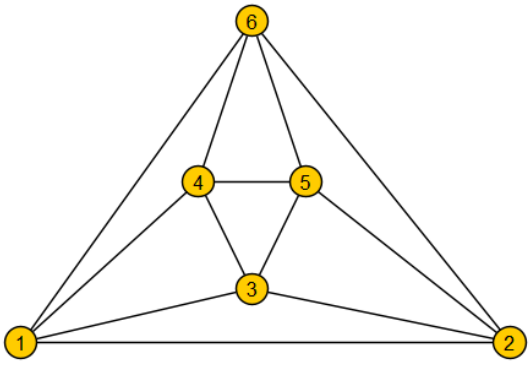
Σχήμα 1.2: Χρωματισμένες όψεις ενός γραφήματος

Εξωτερική όψη γραφήματος/Outerface

Η εξωτερική όψη ενός γραφήματος είναι η περιοχή του επιπέδου η οποία δεν περιορίζεται από ακμές. (βλ. Σχήμα 1.2 η όψη χρωματισμένη με απαλό γαλάζιο)

Τριγωνοποίηση/Τριγωνοποιημένο γράφημα

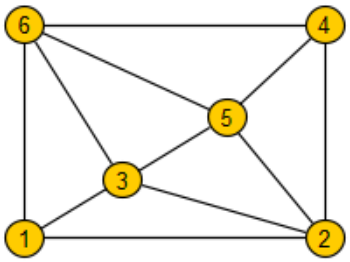
Ένα επίπεδο γράφημα λέγεται **τριγωνοποίηση**, αν κάθε όψη του ορίζεται από ακριβώς **τρεις** ακμές, και δεν είναι δυνατόν να προστεθεί ακμή μεταξύ δύο κορυφών του γραφήματος χωρίς να πάψει να είναι επίπεδο.



Σχήμα 1.3: Τριγωνοποίηση

Εσωτερικά τριγωνοποιημένο γράφημα

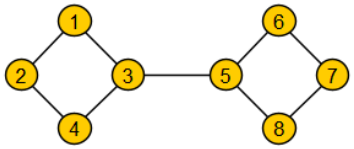
Ένα επίπεδο γράφημα λέγεται **εσωτερικά τριγωνοποιημένο**, αν κάθε όψη του, εκτός από την εξωτερική, ορίζεται από ακριβώς τρεις ακμές.



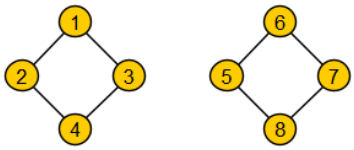
Σχήμα 1.4: Εσωτερικά τριγωνοποιημένο γράφημα

Συνεκτικό γράφημα

Ένα γράφημα G(V,E) λέγεται **συνεκτικό**, όταν για οποιουσδήποτε δύο κόμβους του, υπάρχει μονοπάτι που τους συνδέει.



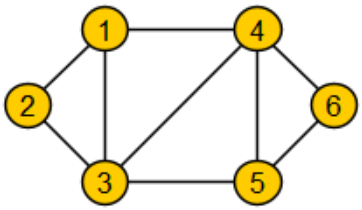
Σχήμα 1.5: Συνεκτικό γράφημα



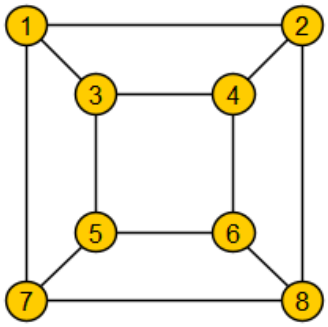
Σχήμα 1.6: Μη-συνεκτικό γράφημα

K-συνεκτικό γράφημα

Ένα γράφημα G(V,E) λέγεται **κ-συνεκτικό**, όταν αφαιρώντας οποιουσδήποτε κ-1 κόμβους, παραμένει συνεκτικό.



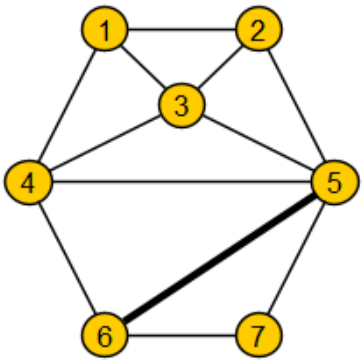
Σχήμα 1.7: 2-συνεκτικό γράφημα



Σχήμα 1.8: 3-συνεκτικό γράφημα

Χορδή

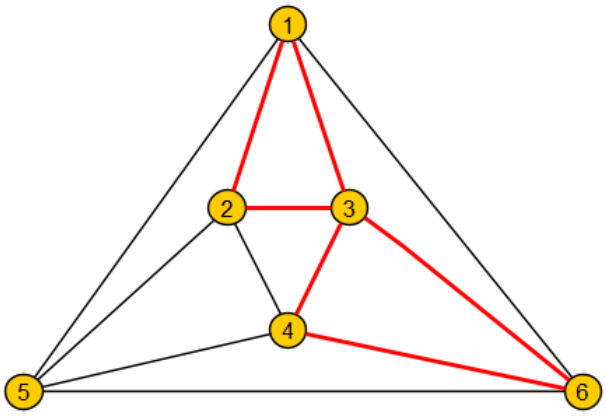
Μια ακμή ενός γραφήματος ονομάζεται **χορδή**, αν δε βρίσκεται στην εξωτερική όψη του γραφήματος, αλλά ενώνει δύο κόμβους οι οποίοι βρίσκονται σε αυτήν.



Σχήμα 1.9: Χορδή εξωτερικού κύκλου

Περίπατος

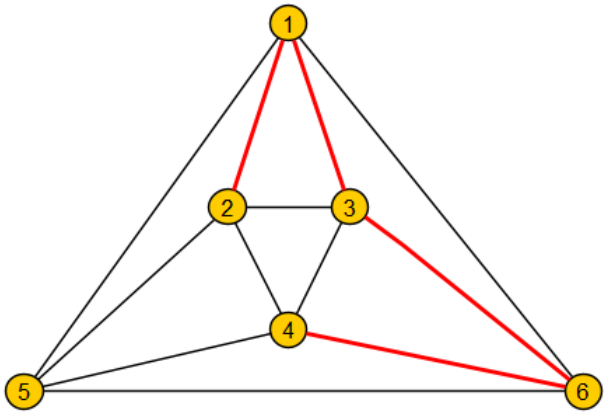
Ένας **περίπατος** σε ένα γράφημα Γ είναι μια ακολουθία $u_0, e_1, u_1, ..., u_{l-1}, e_l, u_l$, που αποτελείται από κόμβους και ακμές (εναλλάξ) του Γ, η οποία αρχίζει από κόμβο και καταλήγει σε κόμβο.



Σχήμα 1.10: Η ακολουθία $(1, 3, 6, 4, 3, 1, 2, 3, 4, 6)$ είναι ένας περίπατος

Μονοπάτι

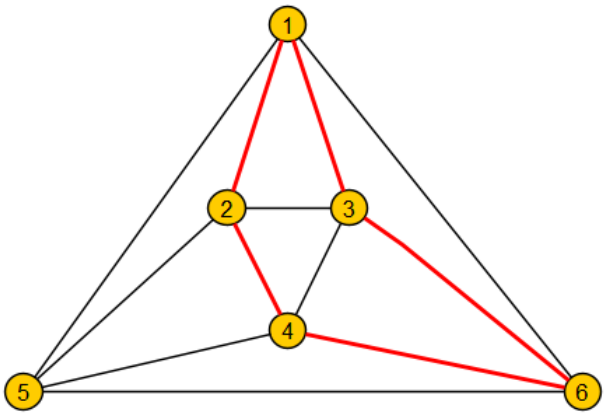
Μονοπάτι ενός γραφήματος ονομάζεται ένας περίπατος, ο οποίος δεν έχει επαναλαμβανόμενες κορυφές.



Σχήμα 1.11: Η ακολουθία $(2, 1, 3, 6, 4)$ είναι ένα μονοπάτι

Κύκλος

Ένα κλειστό μονοπάτι $(u_0 = u_l)$ ονομάζεται **κύκλος**.

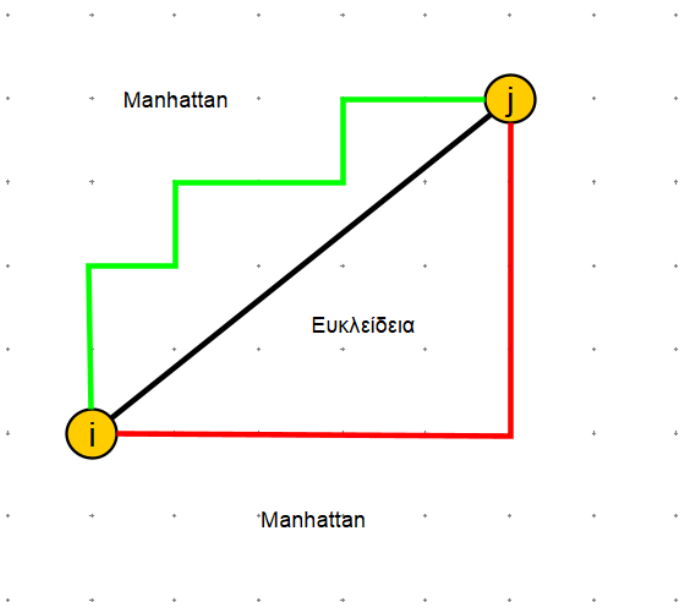


Σχήμα 1.12: Η ακολουθία $(2, 1, 3, 6, 4, 2)$ είναι ένας κύκλος

Απόσταση Manhattan

Η απόσταση Manhattan μεταξύ δύο κόμβων u_i, u_j στο επίπεδο ορίζεται ως η νόρμα l^1 με τύπο:

$$d(u_i, u_j) = |x_i - x_j| + |y_i - y_j|$$



Σχήμα 1.13: Ευκλείδεια και Manhattan απόσταση

Κεφάλαιο 2

Απεικόνιση Γραφημάτων

Ο κύριος σκοπός της απεικόνισης γραφημάτων είναι να προσφέρει ευανάγνωστες αναπαραστάσεις γραφημάτων, οι οποίες βοηθούν στην κατανόηση μοτίβων και σχέσεων που ίσως ήταν δύσκολο να κατανοηθούν από ακατέργαστα δεδομένα ή δυσανάγνωστα γραφήματα.
Το αντικείμενο του συγκεκριμένου κλάδου είναι ο σχεδιασμός των γραφημάτων με τέτοιο τρόπο ώστε να ελαχιστοποιείται κάποια συνάρτηση κόστους, όπως το μέγεθος, η ελάχιστη κλίση των ακμών ή ο συνολικός αριθμός των κυρτώσεων του γραφήματος.
Ενώ η Θεωρία Γραφημάτων προσφέρει μια θεμελιώδη κατανόηση των σχέσεων και των συνδέσεων μεταξύ οντοτήτων, η σπουδαιότητα των αλγορίθμων απεικόνισης γραφημάτων έγγειται στην ικανότητά τους να μεταμορφώσουν αφηρημένα γραφήματα σε αναπαραστάσεις που είναι εύκολα ερμηνεύσιμες οπτικά. Οι αλγόριθμοι αυτοί, γεφυρώνουν το κενό μεταξύ θεωρητικών ιδεών και πρακτικών εφαρμογών.
Σε πεδία όπως η οπτικοποίηση δικτύων, ο πολεοδομικός σχεδιασμός και ο σχεδιασμός κυκλωμάτων, ο τρόπος απεικόνισης των στοιχείων του γραφήματος είναι ζωτικής σημασίας. Γραφήματα τα οποία είναι σχεδιασμένα με αποτελεσματικό και αισθητικά όμορφο τρόπο, ενισχύουν την ικανότητα κατανόησης, βοηθούν στη λύση προβλημάτων και στη λήψη εμπεριστατωμένων στρατηγικών αποφάσεων.
Επομένως, η ανάπτυξη και η υλοποίηση τέτοιων αλγορίθμων είναι κρίσιμη για την πλήρη αξιοποίηση της Θεωρίας Γραφημάτων σε ένα ευρύ φάσμα πρακτικών πλαισίων.

2.1 Στυλ απεικόνισης

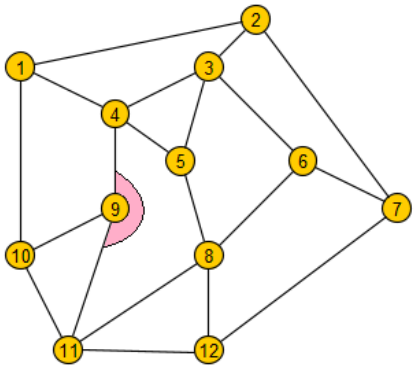
Υπάρχουν πολλοί διαφορετικοί τρόποι απεικόνισης των γραφημάτων, από τους οποίους ο καθένας δίνει έμφαση σε συγκεκριμένα χαρακτηριστικά του γραφήματος. Παρακάτω αναφέρονται συνοπτικά κάποιοι από αυτούς.

Επίπεδη Απεικόνιση

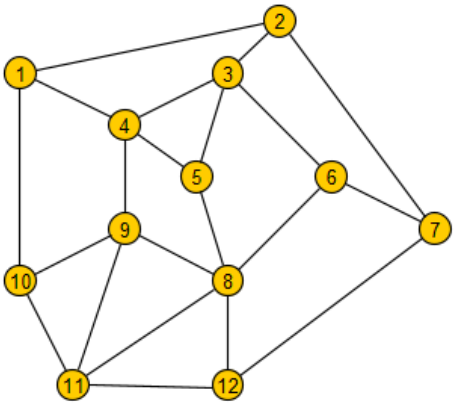
Όπως προαναφέρθηκε, η απεικόνιση ενός γραφήματος είναι **επίπεδη**, αν οι ακμές του δεν τέμνονται.

Κυρτή Απεικόνιση

Κυρτή απεικόνιση ενός γραφήματος, είναι μια επίπεδη απεικόνιση, τέτοια ώστε κάθε όψη του να ορίζεται από ένα κυρτό πολύγωνο, δηλαδή κάθε εσωτερική γωνία να είναι το πολύ 180^o.



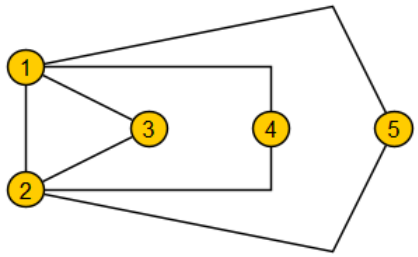
Σχήμα 2.1: Μη κυρτή απεικόνιση



Σχήμα 2.2: Κυρτή απεικόνιση

Πολύγραμμη Απεικόνιση

Πολύγραμμη απεικόνιση ενός γραφήματος, είναι μια απεικόνιση στην οποία οι ακμές του γραφήματος αναπαριστώνται από μια ακολουθία συνεχόμενων ευθυγράμμων τμημάτων.



Σχήμα 2.3: Πολύγραμμη Απεικόνιση

Ευθύγραμμη Απεικόνιση

Ευθύγραμμη απεικόνιση ενός γραφήματος, είναι μια απεικόνιση στην οποία οι ακμές του γραφήματος αναπαριστώνται σαν ευθείες γραμμές. Το σημείο στα οποία τέμνονται δύο τμήματα της ίδιας ακμής, ονομάζεται **κύρτωση**.

Απεικόνιση στο πλέγμα

Απεικόνιση στο πλέγμα ονομάζεται μια απεικόνιση ενός γραφήματος, στην οποία οι κόμβοι και οι κυρτώσεις έχουν ακέραιες συντεταγμένες.

2.2 Γνωστοί Αλγόριθμοι Απεικόνισης

Υπάρχουν δύο βασικοί αλγόριθμοι για την απεικόνιση ευθύγραμμων επίπεδων γραφημάτων, οι οποίοι αναπτύχθηκαν την ίδια περίοδο, εντελώς ανεξάρτητα και βασισμένοι σε άκρως διαφορετικές ιδέες.
Οι De Fraysseix, Pach, Pollack ανέπτυξαν έναν αλγόριθμο που εμφυτεύει ένα επίπεδο γράφημα με n κόμβους σε πλέγμα μεγέθους $(2n-4) \times (n-2)$ και σε χρόνο $O(n \log n)$ [2].
Ο αλγόριθμος αυτός, αργότερα βελτιώθηκε σε πολυπλοκότητα $O(n)$ από τους Chrobak, Payne [3].
Προσεγγιστικά την ίδια περίοδο, ο Schnyder παρουσίασε έναν πολύ διαφορετικό αλγόριθμο, ο οποίος βασίζεται στις -αποκαλούμενες- βαρυκεντρικές αναπαραστάσεις γραφημάτων και εμφυτεύει σε πλέγμα μεγέθους $(n-2) \times (n-2)$ και σε χρόνο $O(n)$ [4].
Οι περισσότερες μέθοδοι που αναπτύχθηκαν έκτοτε για την εμφύτευση ευθύγραμμων επίπεδων γραφημάτων σε πλέγμα, βασίζονται σε αυτούς τους δύο αλγορίθμους.
Οι προαναφερθέντες αλγόριθμοι, προϋποθέτουν ότι το δοθέν γράφημα είναι τριγωνοποιημένο. Μπορούν επίσης να χρησιμοποιηθούν για αυθαίρετα επίπεδα γραφήματα, αν προστεθούν σε αυτά *dummy* ακμές ώστε το αρχικό γράφημα να είναι τριγωνοποιημένο, οι οποίες θα αφαιρεθούν στο τέλος, αφού έχει παραχθεί η εμφύτευση στο πλέγμα. Η μέθοδος αυτή, αν και λειτουργική, παράγει γραφήματα που έχουν όψεις σε απροσδιόριστα και περίπλοκα σχήματα. Επομένως, πολλές φορές για τέτοιου είδους γραφήματα, προτιμώνται οι κυρτές απεικονίσεις.
Ο Kant αρχικά ανέπτυξε έναν αλγόριθμο που κατασκεύαζε κυρτές απεικονίσεις 3-συνεκτικών γραφημάτων σε πλέγμα μεγέθους $(2n-4) \times (n-2)$ [5] σε γραμμικό χρόνο, τον οποίο αργότερα βελτίωσε μαζί με τον Chrobak, περιορίζοντας το πλέγμα σε μέγεθος $(n-2) \times (n-2)$ [6].
Λίγο νωρίτερα, οι Schnyder, Trotter παρουσίασαν έναν αντίστοιχο, αλλά διαφορετικό αλγόριθμο σε πλέγμα μεγέθους $(n-1) \times (n-1)$.
Στη συγκεκριμένη εργασία, θα ασχοληθούμε μόνο με ευθύγραμμες απεικονίσεις γραφημάτων και ειδικότερα με την μέθοδο των De Fraysseix, Pach, Pollack.

Κεφάλαιο 3

Κανονική Διάταξη

Για ένα 2-συνεκτικό, επίπεδο γράφημα G συμβολίζουμε $\boldsymbol{C_o(G)}$ τον εξωτερικό κύκλο του G, δηλαδή το όριο της εξωτερικής όψης του.
Τους κόμβους και τις ακμές που βρίσκονται στο $C_o(G)$, τους αποκαλούμε **εξωτερικούς κόμβους** και **εξωτερικές ακμές** αντίστοιχα.

3.1 Ορισμός

Έστω ένα γράφημα $\boldsymbol{G(V, E)}$ το οποίο είναι τριγωνοποίηση με $n \geq 3$ κόμβους. Αφού το G είναι τριγωνοποίηση, υπάρχουν ακριβώς τρεις κόμβοι στο $C_o(G)$. Υποθέτουμε ότι αυτοί οι τρεις κόμβοι $\boldsymbol{v_1, v_2, v_n}$ εμφανίζονται σε αυτή τη σειρά στον εξωτερικό κύκλο, με φορά αντίστροφη του ρολογιού.
Για κάθε ακέραιο $k, 3 \leq k \leq n$, συμβολίζουμε με G_k το επίπεδο υπογράφημα του G που προκύπτει από τους k κόμβους $v_1, v_2, ..., v_k$. Προφανώς $\boldsymbol{G_n = G}$.
Αποκαλούμε μια διάταξη $\boldsymbol{\pi}$ των κόμβων του G **κανονική** αν οι παρακάτω συνθήκες ισχύουν για κάθε ακέραιο $k, 3 \leq k \leq n$:

(i) το G_k είναι 2-συνεκτικό και τριγωνοποιημένο

(ii) η ακμή (v_1, v_2) είναι εξωτερική ακμή

(iii) αν $k + 1 \leq n$, τότε ο κόμβος v_{k+1} βρίσκεται στην εξωτερική όψη του G_k και όλοι οι γείτονες του v_{k+1}, στο G_k εμφανίζονται διαδοχικά στον εξωτερικό του κύκλο $C_o(G_k)$.

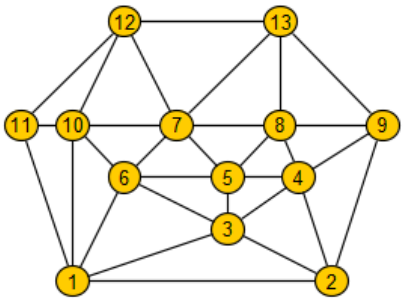
3.2 Ύπαρξη κανονικής διάταξης για κάθε τριγωνοποίηση

Λήμμα 3.2.1 Κάθε τριγωνοποιημένο επίπεδο γράφημα G έχει κανονική διάταξη.

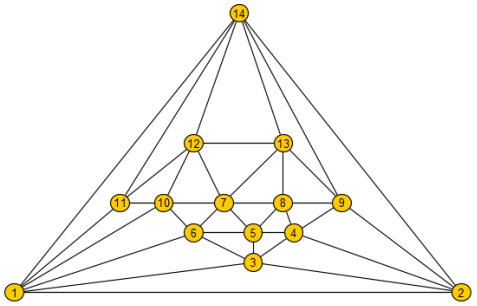
Απόδειξη. Προφανώς το G έχει κανονική διάταξη αν $n = 3$. Έστω ότι $n \geq 4$. Με $G = G_n$, οι συνθήκες $(i), (ii), (iii)$ ισχύουν για $k = n$. Επομένως, θα επιλέξουμε αντίστροφα τους κόμβους $v_{n-1}, v_{n-2}, ..., v_3$ και θα δείξουμε ότι ισχύουν οι συνθήκες για κάθε $k \in \{n-1, n-2, ..., 3\}$.
Υποθέτουμε επαγωγικά, ότι οι κόμβοι $v_n, v_{n-1}, ..., v_{k+1}$, με $k+1 \geq 4$ έχουν επιλεχθεί κατάλληλα και οι συνθήκες $(i), (ii), (iii)$ ισχύουν για το k. Αν μπορούμε στη συνέχεια να επιλέξουμε ως v_k έναν κόμβο $w \neq v_1, v_2$ ο οποίος βρίσκεται πάνω στο $C_o(G_k)$ και δεν είναι μέρος κάποιας χορδής, τότε προφανώς οι $(i), (ii), (iii)$ ισχύουν για $k-1$, αφού $G_{k-1} = G_k - v_k$. Άρα, αρκεί να αποδείξουμε ότι υπάρχει τέτοιος κόμβος w.
Έστω $C_o(G_k) = w_1, w_2, ..., w_t$, όπου $w_1 = v_1$ και $w_t = v_2$. Αν ο $C_o(G_k)$ δεν έχει καμία χορδή, τότε οποιοσδήποτε από τους κόμβους $w_2, w_3, ..., w_{t-1}$ είναι κατάλληλος για επιλογή. Έστω ότι ο $C_o(G_k)$ έχει χορδές. Τότε, το G_k θα έχει μια ελάχιστη χορδή $(w_p, w_q), p+2 \leq q$, δηλαδή τέτοια ώστε σε κανέναν από τους ενδιάμεσους κόμβους $w_{p+1}, w_{p+2}, ..., w_{q-1}$ να μην καταλήγει κάποια χορδή. Σε αυτή την περίπτωση, οποιοσδήποτε από τους κόμβους $w_{p+1}, w_{p+2}, ..., w_{q-1}$ είναι κατάλληλος για επιλογή. □

Πρόταση 1 Κάθε εσωτερικά τριγωνοποιημένο επίπεδο γράφημα G με n κόμβους,έχει κανονική διάταξη.

Απόδειξη. Έστω $w_1, w_2, ..., w_t$ οι κόμβοι της εξωτερικής όψης του G διαδοχικά. Αν προσθέσουμε έναν νέο κόμβο v_{n+1} και τον ενώσουμε με ακμές με τους κόμβους $w_1, w_2, w_3, w_4, .., w_t$ μετακινώντας τους κατάλληλα, έτσι ώστε οι ακμές να μην τέμνονται (Σχήμα 3.2), τότε το γράφημα G' που προκύπτει είναι ένα τριγωνοποιημένο επίπεδο γράφημα, επομένως έχει κανονική διάταξη. Έστω $\pi = \{v_1, v_2, ..., v_{n+1}\}$ η κανονική διάταξη του G'. Προφανώς, η διάταξη $\pi' = \{v_1, v_2, ..., v_n\}$ είναι μια κανονική διάταξη για το G. □



Σχήμα 3.1: G



Σχήμα 3.2: G'

3.3 Αλγόριθμος υπολογισμού της κανονικής διάταξης ενός γραφήματος

Για να είμαστε σίγουροι πως ισχύουν οι προϋποθέσεις (i),(ii),(iii) σε όλα τα βήματα του αλγορίθμου, πρέπει να επιλέγουμε προσεκτικά τους κόμβους που τοποθετούμε στην διάταξη.
Για κάθε κόμβο υ διατηρούμε τρεις μεταβλητές:

1. mark(υ) = 1 αν ο κόμβος έχει προστεθεί ήδη στην διάταξη και 0 αν όχι
2. out(υ) = 1 αν ο κόμβος βρίσκεται στην εξωτερική όψη του γραφήματος και 0 αν όχι
3. chords(υ) = ο αριθμός των χορδών που καταλήγουν στον υ.

Κανονική Διάταξη

1 Έστω ότι οι κόμβοι $\upsilon_1, \upsilon_2, \upsilon_n$ εμφανίζονται στην εξωτερική όψη σε αυτή τη σειρά, αντίστροφα με τη φορά του ρολογιού
2 Θέτουμε mark(υ) = 0, out(υ) = 0, chords(υ) = 0 για κάθε κόμβο $\upsilon \in V$
3 Θέτουμε out(υ_1)=1, out(υ_2)=1, out(υ_n)=1.
4 **Για** $k = n$ ως 3
5 **Εκτέλεσε**
6 Επίλεξε x τέτοιο ώστε mark(x)=0, out(x)=1, chords(x)=0 και $x \neq \upsilon_1, \upsilon_2$
7 Θέσε $\upsilon_k = x$ και mark(x) = 1
8 Έστω $C_o(G_{k-1}) = w_1, w_2, ..., w_t$ όπου $w_1 = \upsilon_1$, $w_t = \upsilon_2$
9 Έστω $w_p, w_{p+1}, ..., w_q$ οι γείτονες του υ_k με $mark(w_i) = 0$
10 $\forall w_i, p < i < q$, $out(w_i) \leftarrow 1$
11 **Για κάθε** $w_i, p \leq i \leq q$,
12 Ενημέρωσε το $chords(w_i)$
13 **Για κάθε** γείτονα n του w_i
14 Ενημέρωσε το $chords(n)$

Κεφάλαιο 4

Μέθοδος **Shift**

Σε αυτή την ενότητα, θα περιγράψουμε τον αλγόριθμο ***Shift***.
Ο αλγόριθμος απεικονίζει το γράφημα G στο επίπεδο, τοποθετώντας έναν κόμβο τη φορά, βασισμένος σε κάποια κανονική διάταξη π=$(\upsilon_1, \upsilon_2, ..., \upsilon_n)$ και προσαρμόζοντας την τρέχουσα μερική απεικόνιση σε κάθε στάδιο.
Με την μετακίνηση κάθε κόμβου υ_i, πρέπει να μετακινηθεί μαζί του και ένα σύνολο κόμβων. Το σύνολο αυτών των κόμβων ονομάζεται *Covering Set* και το συμβολίζουμε ως $L(\upsilon_i)$.
Κάθε κόμβος ανήκει στο covering set του και ως covering set του ορίζουμε την ένωση του εαυτού του και των covering sets των γειτονικών του κόμβων οι οποίοι έπαψαν να βρίσκονται στην εξωτερική όψη του γραφήματος μετά την προσθήκη του υ_i (ή πιο απλά των γειτονικών του κόμβων, εκτός από τον αριστερότερο και τον δεξιότερο).
Έστω $w_p, w_{p+1}, ..., w_q$ με $x(w_p) < x(w_{p+1}) < ... < w_q$ οι γείτονες του υ_i και συμβολίζουμε:

$$L(\upsilon_i) = \{\upsilon_i\} \cup (\bigcup_{i=p+1}^{q-1} L(w_i)) \tag{4.1}$$

Συμβολίζουμε τη θέση ενός κόμβου υ με $P(\upsilon) = (x(\upsilon), y(\upsilon))$.
Αν $P_1 = (x_1, y_1)$ και $P_2 = (x_2, y_2)$ είναι δύο σημεία του πλέγματος με άρτια απόσταση $Manhattan$, τότε το σημείο τομής της ευθείας με κλίση +1 που διέρχεται από το P_1 και της ευθείας με κλίση -1 που διέρχεται από το P_2, είναι επίσης σημείο του πλέγματος και το συμβολίζουμε ως μ$(P_1, P2)$.

$$\mu(P_1, P_2) = (\frac{1}{2}(x_1 - y_1 + x_2 + y_2), \frac{1}{2}(-x_1 + y_1 + x_2 + y_2)). \tag{4.2}$$

4.1 Περιγραφή Αλγορίθμου

Απεικονίζουμε αρχικά το G_3 ως ένα τρίγωνο με $P(\upsilon_1) = (0, 0)$, $P(\upsilon_2) = (2, 0)$, $P(\upsilon_3) = (1, 1)$ και $L(\upsilon_i) = \{\upsilon_i\}$ για $i = 1, 2, 3$.
Υποθέτουμε ότι $k - 1 \leq 3$ και έχουμε ήδη απεικονίσει το G_{k-1} με τέτοιο τρόπο ώστε να ισχύουν οι ακόλουθες συνθήκες:

1. $P(\upsilon_1) = (0, 0)$ και $P(\upsilon_2) = (2k - 4, 0)$
2. $x(w_1) < x(w_2) < ... < x(w_t)$, όπου $C_0(G_{k-1}) = w_1, w_2, ..., w_t$ με $w_1 = \upsilon_1$ και $w_t = \upsilon_2$
3. κάθε ακμή (w_i, w_{i+1}) στο $C_0(G_{k-1})$ είναι μία ευθεία γραμμή με κλίση -1 ή +1

Έχοντας ήδη απεικονίσει το G_{k-1}, τοποθετούμε τον κόμβο υ_k με την ακόλουθη μέθοδο:
Έστω $w_p, w_{p+1}, ..., w_q$ με $x(w_p) < x(w_{p+1}) < ... < x(w_q)$ οι γείτονες του υ_k στο $C_o(G_{k-1})$. Λέμε ότι ο υ_k **καλύπτει** τους κόμβους $w_{p+1}, w_{p+2}, ..., w_{q-1}$.
Από τη συνθήκη (3) έχουμε ότι η απόσταση Manhattan μεταξύ των w_p και w_q είναι ζυγός αριθμός, επομένως το $\mu(w_p, w_q)$ είναι σημείο του πλέγματος.
Αν τοποθετήσουμε τον κόμβο υ_k στο $\mu(w_p, w_q)$, τότε η ακμή (w_p, υ_k) θα υπερκαλύπτει την (w_p, w_{p+1}), σε περίπτωση που αυτή έχει επίσης κλίση +1.
Για να το αποφύγουμε αυτό, ακολουθούμε τα παρακάτω βήματα:

α′) μετακινούμε τους κόμβους $w_1, w_2, ..., w_p$, μαζί με τα Covering Set τους **κατά ένα προς τα αριστερά**

β′) Παρομοίως, μετακινούμε τους κόμβους $w_q, w_{q+1}, ..., w_t(= \upsilon_2)$ μαζί με τα Covering Set τους **κατά ένα προς τα δεξιά**.

γ′) Έπειτα, τοποθετούμε τον κόμβο υ_k στο σημείο $\mu(w_p, w_q)$ το οποίο, υπολογισμένο με τις νέες θέσεις των $\boldsymbol{w_p, w_q}$, είναι επίσης σημείο του πλέγματος.

. Με τη μέθοδο αυτή, εξασφαλίζουμε πως όλοι οι γείτονες του υ_k θα είναι «ορατοί» από το σημείο $\mu(w_p, w_q)$ και δεν θα υπάρχουν επικαλυπτόμενες ακμές.
Για να βρίσκεται στο τελικό γράφημα ο κόμβος υ_1 στο σημείο (0,0) και να ισχύουν οι συνθήκες (1),(2),(3) , αντικαθιστούμε τα βήματα α′, β′ με τα παρακάτω:

(i) μετακινούμε τους κόμβους $w_{p+1}, w_{p+2}, ..., w_{q-1}$, μαζί με τα Covering Set τους **κατά ένα προς τα δεξιά**

(ii) Παρομοίως, μετακινούμε τους κόμβους $w_q, w_{q+1}, ..., w_t (= \upsilon_2)$ μαζί με τα Covering Set τους **κατά δύο προς τα δεξιά**.

Η υλοποίηση του αλγορίθμου που περιγράφηκε, γίνεται σε χρόνο $\mathbf{O(n^2)}$.

Λήμμα 4.1 Έστω G_k, $3 \leq k \leq n$, ένα ευθύγραμμο γράφημα εμφυτευμένο στο πλέγμα όπως περιγράφηκε παραπάνω. Έστω ότι ο $C_o(G_k)$ περιέχει t κόμβους και έστω $\delta_1 \leq \delta_2 \leq ... \leq \delta_t$ μία αύξουσα ακολουθία μη αρνητικών ακεραίων. Αν, για κάθε i, μετακινήσουμε τους κόμβους του $L(w_i)$ κατά δ_i προς τα δεξιά, τότε προκύπτει και πάλι μια ευθύγραμμη απεικόνιση του G_k.

Απόδειξη. Η απόδειξη γίνεται με επαγωγή στο k. Για το G_3 το λήμμα προφανώς ισχύει. Υποθέτουμε ότι το λήμμα ισχύει για το G_{k-1}, $k \geq 4$. Έστω $C_o(G_{k-1}) = w_1, w_2, ..., w_t$ όπως στον αλγόριθμο. Είμαστε έτοιμοι να προσθέσουμε τον κόμβο υ_k στο G_{k-1} ακολουθώντας τα βήματα που περιγράφηκαν στον αλγόριθμο. Προφανώς έχουμε
$C_o(G_k) = w_1, w_2, ..., w_p, \upsilon_k, w_q, w_{q+1}, ..., w_t$. Έστω
$\delta_1 \leq \delta_2 \leq ... \leq \delta_p \leq \delta \leq ... \leq \delta_t$ μία αύξουσα ακολουθία από t' μη αρνητικούς ακεραίους. Μετακινούμε κάθε $L(w_i)$ κατά δ_i και το $L(\upsilon_k)$ κατά δ. Θα αποδείξουμε ότι η απεικόνιση του G_k παραμένει ευθύγραμμη.
Έστω ακολουθία $\delta'_1 \leq \delta'_2 \leq ... \leq \delta'_t$ για το G_{k-1} τέτοια ώστε:

$$\delta'_i = \begin{cases} \delta_i, & if\ 1 \leq i \leq p; \\ \delta + 1, & if\ p+1 \leq i \leq q-1; \\ \delta_i + 2, & if\ q \leq i \leq t. \end{cases}$$

Η $\delta'_1 \leq \delta'_2 \leq ... \leq \delta'_t$ είναι μια αύξουσα ακολουθία από t μη αρνητικούς ακεραίους. Από την επαγωγική υπόθεση, αν μετακινήσουμε κατά δ'_i τα $L(w_i)$ για την απεικόνιση του G_{k-1}, πριν να προσθέσουμε τον κόμβο υ_k, τότε η απεικόνιση του G_{k-1} παραμένει ευθύγραμμη απεικόνιση στο πλέγμα. Επομένως, και το G_k είναι ευθύγραμμη απεικόνιση στο πλέγμα, αφού ο υ_k θα μετακινηθεί κατά δ και $L(\upsilon_k) = \{\upsilon_k\} \cup (\bigcup_{i=p+1}^{q-1} L(w_i))$, άρα μετακινείται ομοιόμορφα μαζί με τους $w_1, w_2, ..., w_q$.

□

4.2 Μέγεθος πλέγματος

Από την περιγραφή της μεθόδου, έχουμε ότι στο γράφημα G_k ισχύει ότι $P(v_1) = (0, 0)$, $P(v_2) = (2k - 4)$, επομένως στο G_n θα έχουμε:

1. $P(v_1) = (0, 0)$, $P(v_2) = (2n - 4)$
2. $P(v_n) = \mu(v_1, v_2) = \frac{(2n-4)}{2} = n - 2$

Επομένως, το μέγεθος του πλέγματος θα είναι $(2n - 4) \times (n - 2)$

Κεφάλαιο 5

Υλοποίηση του αλγορίθμου Shift σε προγραμματιστικό περιβάλλον

5.1 Τεχνολογικά Εργαλεία

Σε αυτή την ενότητα, θα αναφερθούν επιγραμματικά τα κυριότερα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση του αλγορίθμου.

5.1.1 Πρόγραμμα επεξεργασίας κώδικα

Το πρόγραμμα που χρησιμοποιήθηκε για τη συγγραφή του κώδικα είναι το Visual Studio Code, ένα πρόγραμμα επεξεργασίας πηγαίου κώδικα φτιαγμένο από την Microsoft το οποίο υποστηρίζει διάφορες λειτουργίες όπως υποστήριξη για εντοπισμό σφαλμάτων και ενσωματωμένη «αποθήκη» git για έυκολη πρόσβαση σε παλαιότερες εκδοχές του κώδικα.

5.1.2 Γλώσσα Προγραμματισμού

Η κύρια γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η Javascript, σε συνδυασμό με την HTML και την CSS οι οποίες βοήθησαν στην εξατομίκευση του User Interface.

5.1.3 yFiles for HTML

Το yFiles είναι μια βιβλιοθήκη λογισμικού, η οποία παρέχει βοηθητικά εργαλεία, συναρτήσεις και μεθόδους για την οπτικοποίηση, την επεξεργασία και την ανάλυση γραφημάτων.

5.2 Ευρετικές μέθοδοι για την υλοποίηση του αλγορίθμου

Κατά τη διάρκεια της υλοποίησης του αλγορίθμου σε Javascript, χρειάστηκε να αναπτυχθούν κάποιες ευρετικές μέθοδοι οι οποίες παρουσιάζονται συνοπτικά παρακάτω.

5.2.1 findFaces

Η βιβλιοθήκη των yFiles, περιέχει έναν built-in αλγόριθμο ο οποίος βρίσκει μια επίπεδη απεικόνιση του εισαγόμενου γραφήματος, καθώς και τις όψεις αυτού. Το πρόβλημα αυτού του αλγορίθμου στη συγκεκριμένη περίπτωση, είναι πως δεν λαμβάνει υπόψιν την απεικόνιση που δίνεται από τον χρήστη, επομένως οι όψεις που παρουσιάζει δεν είναι αυτές που φαίνονται στην οθόνη. Λόγω του προαναφερθέντος προβλήματος, ήταν αναγκαία η δημιουργία ενός ευρετικού αλγορίθμου που θα υπολογίζει τις όψεις για την ορατή απεικόνιση του γραφήματος. Έτσι δημιουργήθηκε η μέθοδος $findFaces$.
Η βασική ιδέα στην οποία βασίστηκε, είναι πως μπορούμε να βρούμε τις όψεις ενός γραφήματος, αν για κάθε ακμή $e = (u, v)$, ξεκινώντας από τον κόμβο u, διατρέξουμε διαδοχικά την επόμενη ακμή αντίστροφα με τη φορά του ρολογιού που ξεκινάει από τον «απέναντι» κόμβο και συνεχίσουμε την ίδια διαδικασία μέχρι η ακμή που θα συναντήσουμε να ταυτίζεται με την αρχική.
Η μέθοδος $findFaces$ βρίσκει και επιστρέφει τις όψεις της επίπεδης απεικόνισης σε έναν πίνακα, του οποίου κάθε στοιχείο είναι μια όψη και κάθε όψη αποτελείται από τις ακμές που την ορίζουν στο επίπεδο.

```
1  faces = new List()
2  Για κάθε κόμβο v ∈ G_k
3        Για κάθε ακμή e adj to v
4              Αν trav[e] < 2
5              face = new List()
6              Όσο nextCCWEdge ≠ e
7                    AddItem(face, nextCCWEdge)
8                    nextCCWEdge=getNextCCWEdge
9              Αν face ∉ faces
10             AddItem(faces, face)
11             Για κάθε e' ∈ face : trav[e'] = trav[e'] + 1
12 Επίστρεψε faces
```

Εδώ, $trav[e]$ = αριθμός διαφορετικών $faces$ που έχουν αναγνωριστεί, στα οποία ανήκει η ακμή e. Προφανώς, μια ακμή μπορεί να «συμμετέχει» μόνο σε

δύο διαφορετικά $faces$.
Η μέθοδος **getNextCCWEdge**, που χρησιμοποιείται στον παραπάνω αλγόριθμο, δέχεται ως είσοδο έναν κόμβο v και μια ακμή του e και επιστρέφει την επόμενη ακμή του απέναντι κόμβου, αντίστροφα από τη φορά του ρολογιού.
Έχει δημιουργηθεί ακόμη μια μέθοδος, η οποία λαμβάνοντας ως είσοδο τα faces που προέκυψαν, δημιουργεί τον πίνακα facesNodes. Κάθε στοιχείο του πίνακα αντιστοιχεί σε μία όψη, όμως αυτή τη φορά περιλαμβάνει τους κόμβους που οριοθετούν την όψη.

5.2.2 findOuterface

Όμοια με τον προηγούμενο αλγόριθμο, και για την εύρεση της εξωτερικής όψης του γραφήματος, η βιβλιοθήκη $yFiles$ δεν λάμβανε υπόψιν την τρέχουσα απεικόνισή του.
Επομένως, αναπτύχθηκε ο παρακάτω ευρετικός αλγόριθμος. Πριν την επεξήγησή του, θα δοθούν δύο χρήσιμοι ορισμοί.

Κυρτό σύνολο σημείων
Ένα σύνολο σημείων M είναι κυρτό, αν για κάθε ζεύγος σημείων x, y του M το ευθύγραμμο τμήμα xy που συνδέει τα δύο σημεία, ανήκει πλήρως στο M.

Κυρτό Περίβλημα συνόλου σημείων
Το κυρτό περίβλημα ενός συνόλου σημείων S στο επίπεδο, είναι το μικρότερο κυρτό πολύγωνο P που περικλείει το S.

Η βιβλιοθήκη yFiles περιέχει μια γεωμετρική μέθοδο (Geom.calcConvexHull), η οποία δοθείσης μιας λίστας, της οποίας κάθε στοιχείο είναι ένα σημείο του επιπέδου, υπολογίζει και επιστρέφει τα σημεία τα οποία βρίσκονται πάνω στο κυρτό περίβλημα του συνόλου.
Δημιουργούμε μια λίστα σημείων τέτοια ώστε κάθε σημείο να αντιστοιχεί στο κέντρο ενός κόμβου του γραφήματος και τρέχουμε τον αλγόριθμο convexHull που μας επιστρέφει μια λίστα σημείων, τα οποία ορίζουν το κυρτό περίβλημα του συνόλου. Προφανώς, οι κόμβοι που βρίσκονται σε αυτά τα σημεία θα ανήκουν στην εξωτερική όψη του γραφήματος.
Αφού έχουμε εντοπίσει όλες τις όψεις του γραφήματος με τη βοήθεια του αλγορίθμου findFaces και την αντιστοίχιση με κόμβους στον πίνακα facesNodes, αναζητούμε σε ποια από αυτές ανήκουν όλοι οι κόμβοι του κυρτού περιβλήματος. Η όψη αυτή είναι μοναδική και αποτελεί την εξωτερική όψη του γραφήματος.
Στο συγκεκριμένο πλαίσιο, επειδή το γράφημα που έχουμε είναι πάντα εσωτερικά τριγωνοποιημένο, αν υπάρχει face με περισσότερες από 3 ακμές, τότε το θέτουμε αυτόματα ως εξωτερική όψη.

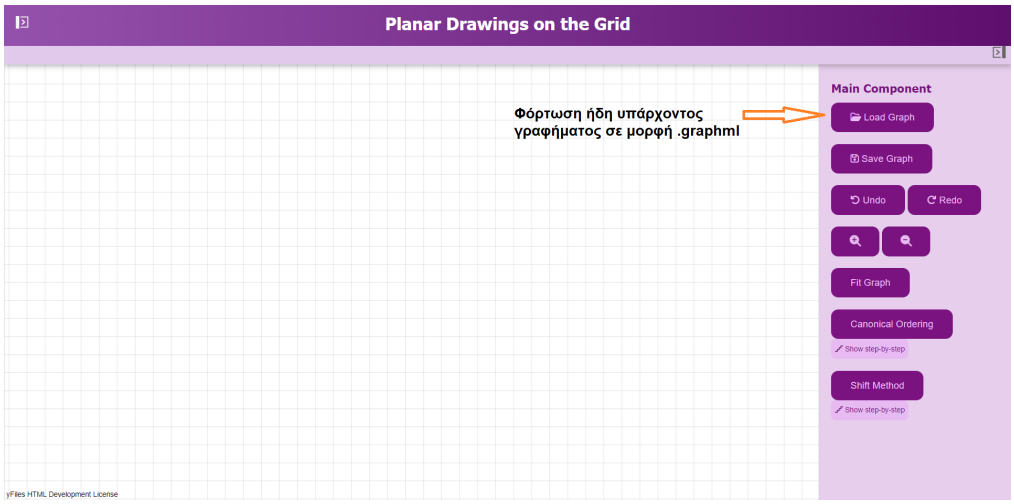
```
1  Αν ∃ face1 ∈ facesNodes με length > 3
2        Επίστρεψε face1
3  Αλλιώς
4        Υπολόγισε ConvexHull(nodeCenters)
5              Βρες face1 ∈ facesNodes τέτοιο ώστε
6              Για κάθε p ∈ convexHull, face1.includes(p) = true,
7        outerface = face1
8        Επίστρεψε outerface
```

5.3 Περιήγηση και Οδηγίες Χρήσης της Εφαρμογής

5.3.1 Δημιουργία/Φόρτωση γραφήματος

Στην εφαρμογή υπάρχουν δύο δυνατές λειτουργίες:

- Φόρτωση έτοιμου γραφήματος σε μορφή .graphml με πάτημα του κουμπιού Canonical Ordering, όπως υποδεικνύεται στο Σχήμα 5.1.
- Χειροκίνητη δημιουργία γραφήματος στο πλέγμα: Η δημιουργία γραφήματος στο πλέγμα της εφαρμογής γίνεται πολύ απλά. Με κλικ στο πλέγμα προστίθεται κόμβος και με κλικ σε κόμβο και τράβηγμα ως έναν άλλο κόμβο, δημιουργείται η αντίστοιχη ακμή.



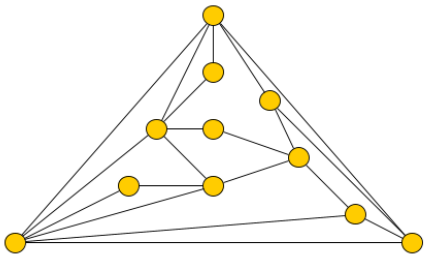
Σχήμα 5.1: Φόρτωση έτοιμου γραφήματος

Αντίστοιχα, με τη χρήση του Save Graph , διατίθεται η δυνατότητα αποθήκευσης ενός γραφήματος, με σκοπό τη γρήγορη ανάκτηση του σε επόμενη χρήση της εφαρμογής.

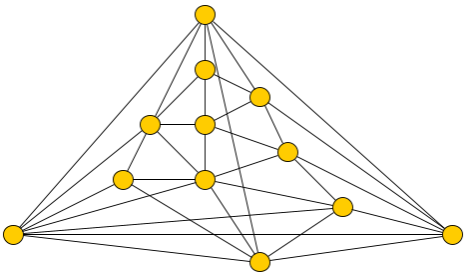
5.3.2 Κατάλληλη μορφή του εισαγόμενου γραφήματος

Ο αλγόριθμος Shift, είναι ένας αλγόριθμος ο οποίος λειτουργεί για συγκεκριμένα γραφήματα. Για τη σωστή λειτουργία της εφαρμογής, το γράφημα που θα δοθεί από τον χρήστη θα πρέπει να είναι:

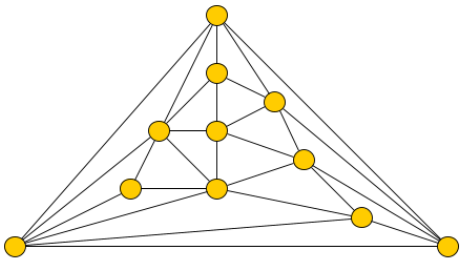
1. **Επίπεδο**: Το εισαγόμενο γράφημα θα πρέπει να είναι επίπεδο και σε επίπεδη απεικόνιση.
2. **Εσωτερικά Τριγωνοποιημένο**: Όλες οι εσωτερικές όψεις του εισαγόμενου γραφήματος θα πρέπει να είναι τριγωνικές.



Σχήμα 5.2: Μη τριγωνοποιημένο



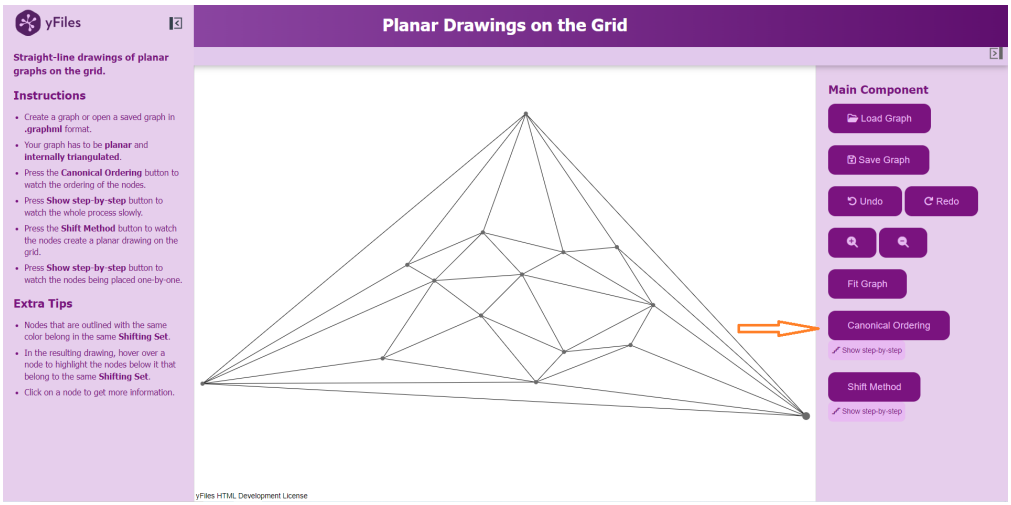
Σχήμα 5.3: Μη επίπεδο



Σχήμα 5.4: Κατάλληλο γράφημα

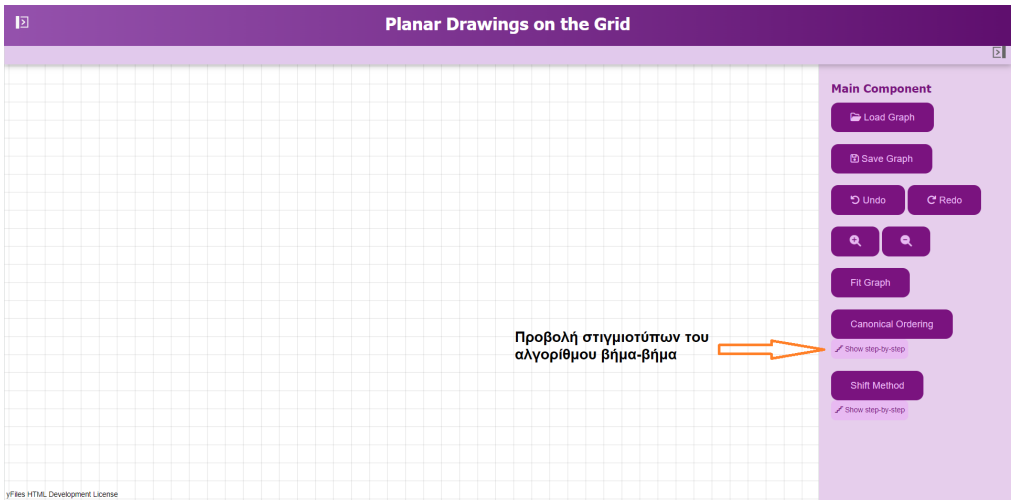
5.3.3 Διάταξη των κόμβων

Η έναρξη του αλγορίθμου κανονικής διάταξης γίνεται με το πάτημα του κουμπιού Canonical Ordering (Σχήμα 5.5).

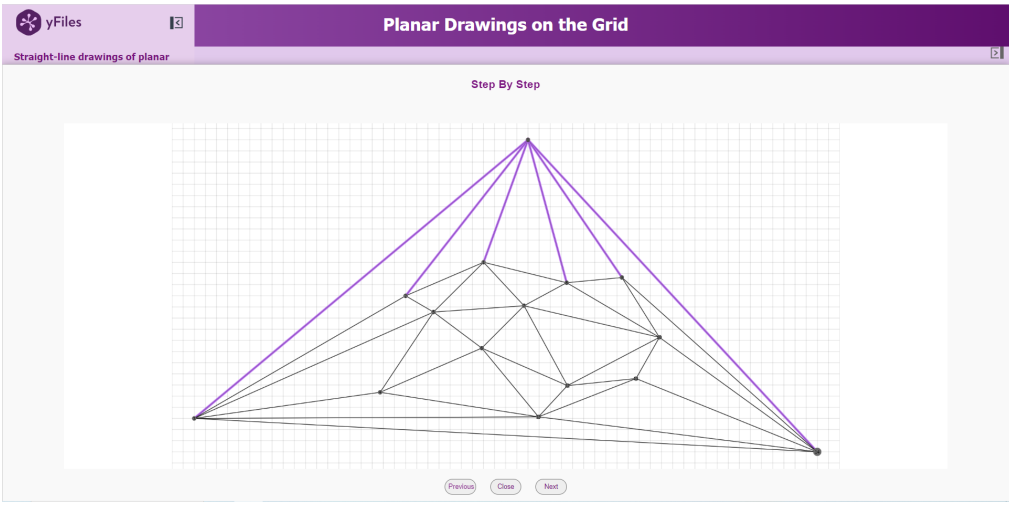


Σχήμα 5.5: Έναρξη Αλγορίθμου Κανονικής Διάταξης

Στη συνέχεια, υπάρχει η δυνατότητα προβολής στιγμιοτύπων των βημάτων του αλγορίθμου βήμα-βήμα, με πάτημα του κουμπιού Show step-by-step (Σχήμα 5.6).



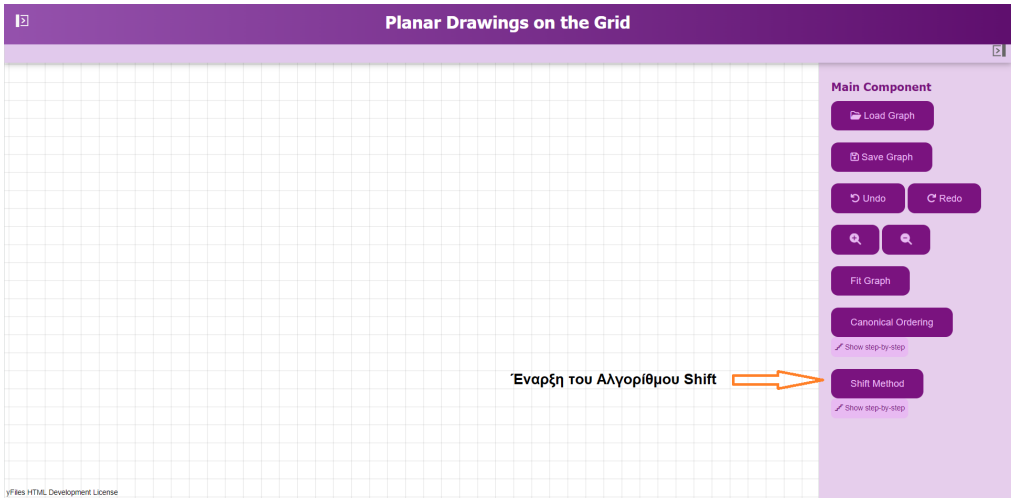
Σχήμα 5.6: Παρουσίαση αλγορίθμου βήμα-βήμα



Σχήμα 5.7: Βήματα Αλγορίθμου

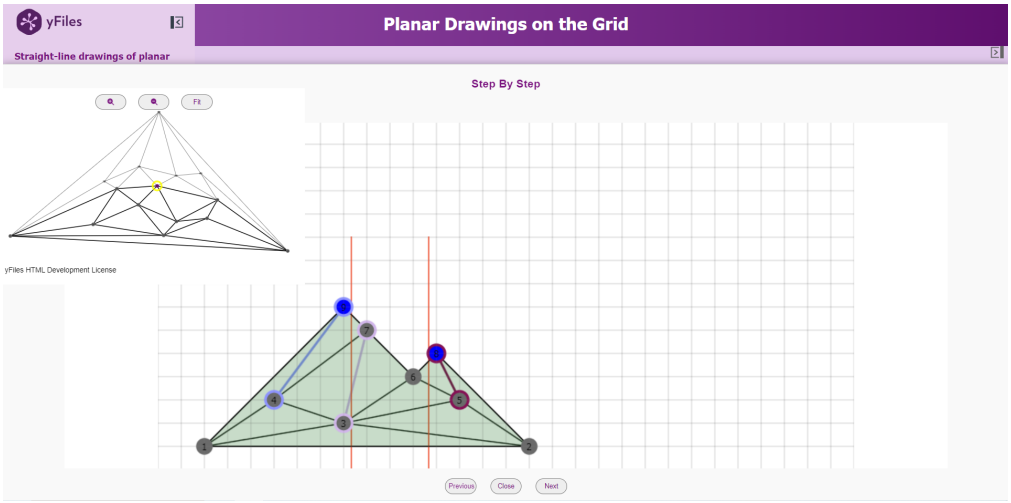
5.3.4 Επαναδημιουργία του γραφήματος

Αφού οι κόμβοι ενταχθούν σε διάταξη, με το πάτημα του κουμπιού Shift Method (Σχήμα 5.8), ξεκινάει η επανασχεδίαση του γραφήματος, με προσαρμογή του κατά την προσθήκη του εκάστοτε κόμβου.

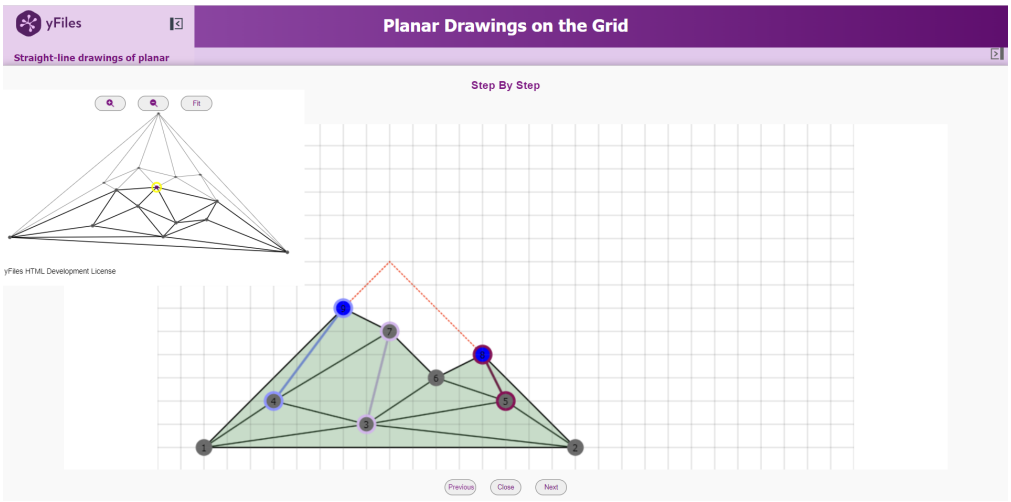


Σχήμα 5.8: Εκκίνηση Αλγορίθμου Shift

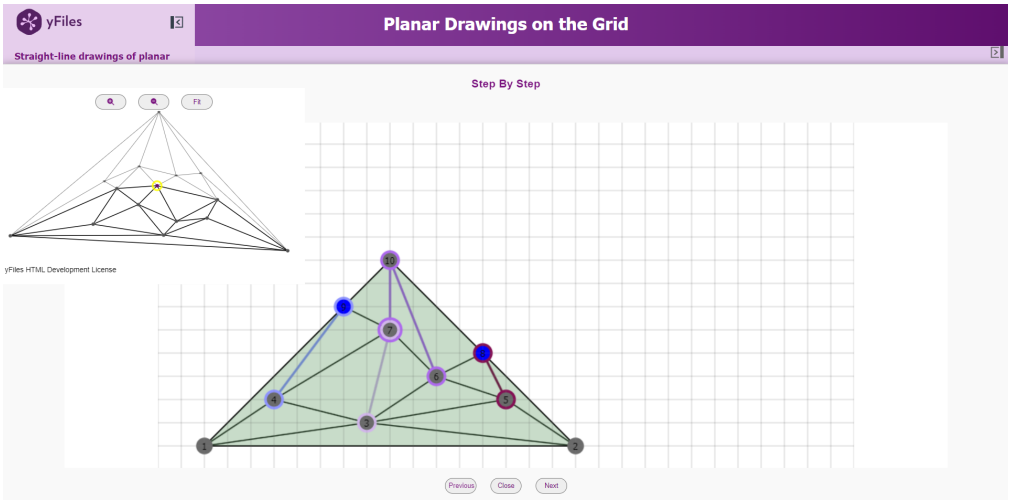
Όμοια με την κανονική διάταξη, αρχικά βλέπουμε όλη την υλοποίηση του αλγορίθμου Shift στο κύριο παράθυρο και στη συνέχεια με το πάτημα του κουμπιού Show step-by-step μπορούμε να δούμε τα βήματα αναλυτικά (Σχήμα 5.9, 5.10, 5.11)
Στην πάνω αριστερά γωνία του pop-up παραθύρου βλέπουμε το αρχικό γράφημα με επισημασμένο τον κόμβο που προστίθεται στο εκάστοτε βήμα. Στο γράφημα που προκύπτει, το covering set ενός κόμβου υποδεικνύεται με ένα χρωματιστό περίγραμμα.



Σχήμα 5.9: Πρώτο βήμα: Μετακίνηση των κόμβων

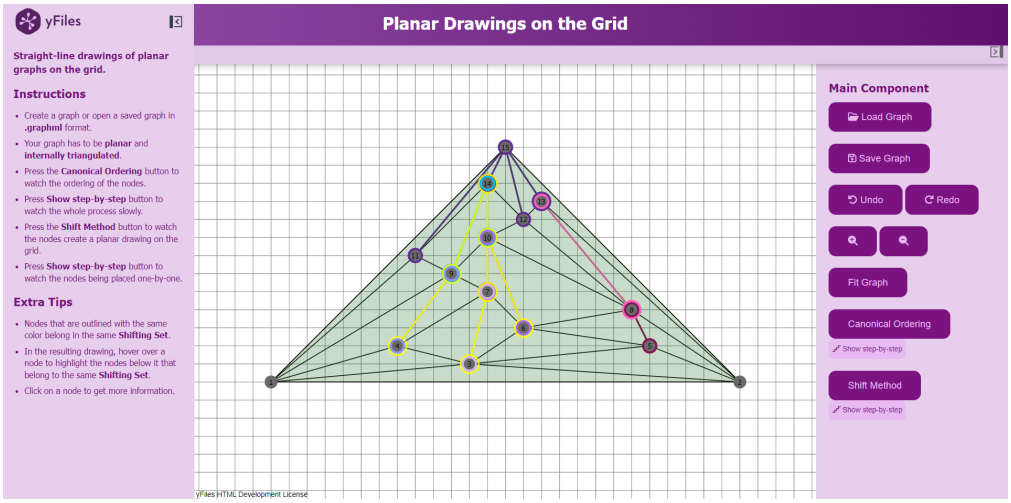


Σχήμα 5.10: Δεύτερο βήμα: Υπολογισμός θέσης του προστιθέμενου κόμβου v_{15}



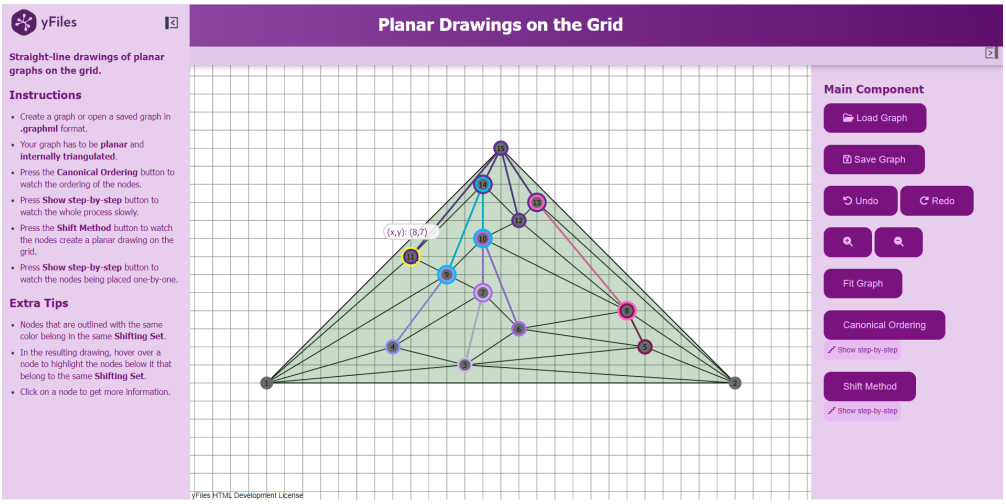
Σχήμα 5.11: Τρίτο βήμα: Τοποθέτηση του κόμβου v_{15}

Μετά την ολοκλήρωση του αλγορίθμου, στο κύριο παράθυρο εμφανίζεται το τελικό γράφημα που προέκυψε.
Στο γράφημα αυτό, με μετακίνηση πάνω σε κάποιον κόμβο, επισημαίνονται οι κατώτεροι κόμβοι του που ανήκουν στο covering set του, όπως φαίνεται στο Σχήμα 5.12.



Σχήμα 5.12: Highlight του covering set του κόμβου v_{18}

Με κλικ πάνω σε κάποιον κόμβο, μπορούμε να δούμε τις συντεταγμένες του στο πλέγμα (Σχήμα 5.13)



Σχήμα 5.13: Κλικ σε κόμβο

Κεφάλαιο 6

Συμπεράσματα και Μελλοντικές Επεκτάσεις

Το λογισμικό που σχεδιάστηκε για αυτή την εργασία, εξυπηρετεί τους αρχικούς του σκοπούς, που ήταν η υλοποίηση του αλγορίθμου Shift και η αναλυτική του παρουσίαση έτσι ώστε να γίνει απολύτως κατανοητός ο αλγόριθμος και να παρατηρηθεί προσεκτικά κάθε του βήμα.
Η τελική εφαρμογή παρέχει ένα περιβάλλον φιλικό προς το χρήστη, που επιτρέπει την εύκολη δημιουργία γραφημάτων και την εύκολη περιήγηση στο επόμενο αλλά και στο προηγούμενο βήμα του αλγορίθμου, με σκοπό την καλύτερη δυνατή κατανόηση του χρήστη.

Μερικές **πιθανές μελλοντικές επεκτάσεις** είναι:

1. Υλοποίηση του αλγορίθμου σε γραμμικό χρόνο $O(n)$
2. Αυτόματη προσθήκη *dummy* ακμών και αυτόματη αφαίρεση στο τελικό στάδιο, ώστε να μπορεί να παρέχει εμφύτευση για όλα τα επίπεδα γραφήματα χωρίς κάποιον επιπλέον περιορισμό
3. Βελτιστοποίηση της πολυπλοκότητας των ευρετικών μεθόδων

Βιβλιογραφία

[1] N. I. Biggs, E. Keith Lloyd, Robin J. Wilson, "Graph Theory, 1736-1936", Clarendon Press, pp. 1-11, 1986

[2] De Fraysseix, H., J. Pach, and R. Pollack. "How to draw a planar graph on a grid." Combinatorica 10 (1990): 41-51.

[3] M. Chrobak, T.H. Payne, A linear-time algorithm for drawing a planar graph on a grid, Information Processing Letters, Volume 54, Issue 4, 1995, pp. 241-246

[4] Schnyder, W. "Embedding planar graphs on the grid." Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms. 1990.

[5] Kant, G. "Drawing planar graphs using the canonical ordering." Algorithmica 16 (1996): pp. 4-32.

[6] Chrobak, M., Kant, G. "Convex grid drawings of 3-connected planar graphs." International Journal of Computational Geometry and Applications 7.03 (1997): pp. 211-223.

[7] Diestel, R., "The Basics". In: Graph Theory. Graduate Texts in Mathematics, vol 173. Springer, Berlin, Heidelberg.

[8] Di Battista, G., and F. Frati. "A survey on small-area planar graph drawing.", arXiv:1410.1006 (2014).

[9] T. Nishizeki and M. S. Rahman. "Planar Graph Drawing", volume 12 of Lecture Notes Series on Computing. World Scientific, 2004

[10] Vismara, L. "Planar Straight-Line Drawing Algorithms." (2013): 193-222.

[11] M. A. Bekos, M. Gronemann, F. Montecchiani, D. Pálvölgyi, A. Symvonis, L. Theocharous, Grid drawings of graphs with constant edge-vertex resolution, Computational Geometry, Volume 98, 2021

[12] M. Chrobak, S. Nakano, Minimum-width grid drawings of plane graphs, Computational Geometry, Volume 11, Issue 1, 1998, Pages 29-54.