ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Πτυχιακή Εργασία

Αλγόριθμοι και παράμετροι γύρω από το δενδροπλάτος

Αντώνης Ι. Σκαρλάτος

Επιβλέπων: Σταύρος Κολλιόπουλος, Καθηγητής

ΑΘΗΝΑ

ΟΚΤΩΒΡΙΟΣ 2018

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Αλγόριθμοι και παράμετροι γύρω από το δενδροπλάτος

Αντώνης Ι.Σκαρλάτος

Α.Μ: 1115201400184

Επιβλέπων: Σταύρος Κολλιόπουλος, Καθηγητής

ΠΕΡΙΛΗΨΗ

Η εργασία αυτή επικεντρώνεται στην δενδροαποσύνθεση και το δενδροπλάτος ενός γραφήματος. Το δενδροπλάτος είναι μια παράμετρος στην θεωρία γραφημάτων που είναι σημαντική και για θεωρητικούς και για πρακτικούς σκοπούς. Στο πρώτο κεφάλαιο δίνονται κάποιοι εισαγωγικοί ορισμοί καθώς και μια σύντομη περίληψη του τι είναι το δενδροπλάτος. Στο δεύτερο κεφάλαιο δίνονται οι ορισμοί του δενδροπλάτους και άλλων παραμέτρων και εξηγούνται οι ιδιότητες και οι σχέσεις του δενδροπλάτους με πολλές άλλες παραμέτρους. Τέλος, το τρίτο κεφάλαιο επικεντρώνεται κυρίως στους προσεγγιστικούς αλγορίθμους και δίνονται δύο προσεγγιστικοί αλγόριθμοι για την εύρεση του δενδροπλάτους. Μέσα από το δεύτερο κεφάλαιο εξηγούνται πολλές ιδιότητες που έχει το δενδροπλάτος με άλλες παραμέτρους οι οποίες χρησιμοποιούνται στο τρίτο κεφάλαιο για την δημιουργία των αλγορίθμων.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Θεωρία γραφημάτων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: δενδροπλάτος, δενδροαποσύνθεση, γραφήματα, αλγόριθμοι, παράμετροι

ABSTRACT

This document focuses on tree decomposition and the treewidth of a graph. Treewidth is a parameter in graph theory which is important for theoretical and practical purposes. In the first chapter, we give some definitions, as well as a brief summary on what a treewidth is. In the second chapter, we give the definition of a treewidth, we define some parameters, and we explain some properties and relationships between treewidth and some parameters. Finally we use these properties to build some algorithms in the third chapter, where we focus on approximation algorithms and we present two approximation algorithms for the problem of treewidth.

SUBJECT AREA: Graph theory

KEYWORDS: treewidth, tree decomposition, graphs, algorithms, parameters

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον καθηγητή μου Σταύρο Κολλιόπουλο για αυτά που έμαθα στην θεωρία γραφημάτων μέσω του προπτυχιακού μαθηματός του, καθώς και για την πολύτιμη βοήθειά του και τις συμβουλές του που με βοήθησαν να υλοποιήσω την πτυχιακή μου εργασία.

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΕΙΣΑΓΩΓΗ

1.1 Εισαγωγικές έννοιες και βοηθητικοί ορισμοί

Παρακάτω δίνονται κάποιες βασικές έννοιες και κάποιοι βοηθητικοί ορισμοί που θα χρησιμοποιηθούν στα επόμενα κεφάλαια. Για όλα τα γραφήματα που αναφέρονται παρακάτω και δεν δίνονται οι ιδιοτητές τους, υποθέτουμε ότι είναι μη κετευθυνόμενα, πεπερασμένα, χωρίς βάρη στις ακμές και απλά, δηλαδή δεν περιέχονται πολλαπλές ακμές μεταξύ δύο κορυφών και δεν υπάρχει ακμή από μία κορυφή προς τον εαυτό της.

Ορισμός 1.1. Δέντρο *είναι ένα συνεκτικό γράφημα το οποίο δεν περιέχει κύκλο.*

Για κάθε δέντρο ισχύει ότι ο αριθμός των ακμών του ισούται με το πλήθος των κορυφών του μείον ένα. Επίσης μεταξύ δύο κορυφών ενός δέντρου υπάρχει μοναδικό απλό μονοπάτι, δηλαδή υπάρχει μοναδικό μονοπάτι τέτοιο ώστε κάθε κορυφή που ανήκει στο μονοπάτι αυτό δεν επαναλαμβάνεται.

Ορισμός 1.2. Δάσος *είναι ένα γράφημα για το οποίο ισχύει ότι όλες οι συνεκτικές του συνιστώσες είναι δένδρα.*

Γενικά παρακάτω υποθέτουμε ότι όλα τα γραφήματά μας είναι συνεκτικά, αλλά εύκολα πολλά αποτελέσματα γενικεύονται και για μη συνεκτικά γραφήματα.

Ορισμός 1.3. *Έστω G = (V, E) ένα γράφημα με σύνολο κορυφών το V και σύνολο ακμών το E. Το $H = (V', E')$ είναι* υπογράφημα *του G αν για τις κορυφές του H ισχύει ότι $V' \subseteq V$ και για τις ακμές του H ότι $E' \subseteq E, (\{u, w\} \in E' \Rightarrow u, w \in V')$.*

Ορισμός 1.4. *Έστω G = (V, E) ένα γράφημα με σύνολο κορυφών το V και σύνολο ακμών το E. Για ένα υπογράφημα $H = (V', E')$ του G που ισχύει ότι $V' \subseteq V$ και $E' = \{(u, w) \in E \mid u \in V' \wedge w \in V'\}$ λέμε ότι το H είναι* εναγόμενο υπογράφημα *του G. Το εναγόμενο γράφημα $H = (V', E')$ του G θα το συμβολίζουμε και ως $G[V']$.*

Ένα υπογράφημα δεν είναι απαραίτητο να είναι εναγόμενο. Για ένα γράφημα $G = (V, E)$ και ένα εναγόμενο υπογραφημά του $H = (V', E')$, το $A = (V', E'')$ όπου $E'' \subseteq E'$ είναι ένα υπογράφημα του G. Ουσιαστικά ένα υπογράφημα δεν είναι απαραίτητο να περιέχει όλες τις ακμές που υπάρχουν μεταξύ των κορυφών του.

Ορισμός 1.5. *Μία* σύνθλιψη ακμής (edge contraction) *ορίζεται ως η πράξη στην οποία διαλέγουμε μία ακμή $\{u, w\}$, σβήνουμε την ακμή και τις δύο κορυφές u και w και τις αντικαθιστούμε με μία καινούργια κορυφή k. Η καινούργια κορυφή k ενώνεται με όλες τις κορυφές που ενωνόντουσαν είτε με την u είτε με την w.*

Ορισμός 1.6. Έλασσον γράφημα (minor graph) *ενός γραφήματος G είναι ένα γράφημα το οποίο προκύπτει από το G με αφαίρεση κορυφών, αφαίρεση ακμών και σύνθλιψη ακμών.*

Όταν αφαιρούμε μία κορυφή τότε αφαιρούμε και όλες τις ακμές στις οποίες ανήκει αυτή η κορυφή. Ένα υπογράφημα του G είναι και έλασσον, αλλά ένα έλασσον δεν είναι απαραίτητα υπογράφημα του G καθώς η σύνθλιψη ακμών μπορεί να αυξήσει τον βαθμό μια κορυφής και να αλλάξει την δομή του γραφήματος.

Ορισμός 1.7. *Μια οικογένεια $\{T_i\}_{i\in I}$ υποσυνόλων ενός συνόλου T ικανοποιεί την ιδιότητα Helly αν το $J \subseteq I$ και το $\forall i,j \in J, T_i \cap T_j \neq \emptyset$ υποδηλώνει ότι $\bigcap_{j\in J} T_j \neq \emptyset$.*

Ουσιαστικά η ιδιότητα Helly μας λέει ότι αν μια οικογένεια υποσυνόλων ικανοποιεί την ιδιότητα Helly και η τομή ανά δύο υποσυνόλων που ανήκουν στην οικογένεια είναι μη κενή, τότε και η τομή όλων των υποσυνόλων θα είναι μη κενή. Το υποδέντρο είναι ένα συνεκτικό υπογράφημα ενός δέντρου. Η οικογένεια υποδένδρων ικανοποιεί την ιδιότητα Helly οπως φαίνεται και από το παρακάτω θεώρημα. Αυτό σημαίνει ότι αν σε ένα δέντρο διαλέξουμε υποδένδρα τέτοια ώστε ανά δύο να τέμνονται, τότε και η τομή όλων αυτών θα είναι μη κενή. Διαισθητικά αυτό συμβαίνει επειδή δεν υπάρχει κύκλος σε ένα δέντρο.

Θεώρημα 1.1. *Μια οικογένεια $\{T_i\}_{i\in I}$ υποδένδρων ενός δέντρου T ικανοποιεί την ιδιότητα Helly.*

Απόδειξη. Δες στο [7]. □

Παρακατω δίνονται κάποιοι ορισμοί για κάποιες κλάσεις πολυπλοκότητας που θα τις χρειαστούμε παρακάτω για να δούμε σε ποια ανήκει το πρόβλημα της εύρεσης δενδροπλάτους και πως δύσκολα προβλήματα μπορούν να λυθούν πιο αποδοτικά με την χρήση της δενδροαποσύνθεσης.

Ορισμός 1.8. *Ένα πρόβλημα Π ανήκει στην κλάση* NP *αν και μόνο αν υπάρχουν πολυώνυμα p και q και μία ντετερμινιστική μηχανή Turing M έτσι ώστε να ικανοποιούνται και οι τρεις παρακάτω προϋποθέσεις*

- *$\forall x, w$, η μηχανή M τρέχει σε χρόνο $p(|x|)$ με είσοδο το (x, w).*
- *$\forall x \in \Pi$, υπάρχει ένα w μεγέθους $q(|x|)$ έτσι ώστε $M(x, w) = 1$.*
- *$\forall x \notin \Pi$, και για όλα τα w μεγέθους $q(|x|)$ ισχύει ότι $M(x, w) = 0$.*

Για ένα πολυώνυμο p και μία είσοδο x, αν ισχύει ότι το πρόγραμμα τρέχει σε $p(|x|)$ σημαίνει ότι χρόνος εκτέλεσης αυξάνεται πολυωνυμικά σε σχέση με το μέγεθος της εισόδου. Στον παραπάνω ορισμό το Π είναι το πρόβλημα (για παράδειγμα το Subset sum problem), το x είναι μία οποιαδήποτε περίπτωση του προβλήματος (για παράδειγμα μια τυχαία λίστα με ακέραιους) και το w είναι μία υποψήφια λύση (για παράδειγμα μία υπολίστα της λίστας των ακεραίων). Το $M(x, w)$ σημαίνει ότι για την περιπτωσή μας που είναι το x, δοκιμάζουμε την υποψήφια λύση που είναι το w στην μηχανή M. Αν το w είναι πράγματι μία λύση τότε η μηχανή επιστρέφει 1, διαφορετικά 0. Σε κάθε περίπτωση η μηχανή θα πρέπει να τερματίσει σε πολυωνυμικό χρόνο ως προς το μέγεθος του x. Επιπλέον το μέγεθος του w

θα πρέπει να είναι πολυωνυμικού μεγέθους σε σχέση με το μέγεθος του x. Η κλάση NP περιέχει όλα τα προβλήματα για τα οποία αν δώσουμε μια υποψήφια λύση w, η μηχανή θα μας απαντήσει γρήγορα για το αν το w είναι πραγματική λύση ή όχι του x. Παρόλο που τα προβλήματα που ανήκουν στην κλάση NP λύνονται σε πολυωνυμικό χρόνο αν τους δώσουμε μια υποψήφια λύση ως είσοδο, είναι πιθανό να μην υπάρχει πολυωνυμικός αλγόριθμος ως προς το μέγεθος της εισόδου που να βρίσκει μία λύση του προβλήματος με είσοδο μόνο το x.

Για να τον ορισμό της κλάσης πολυπλοκότητας NP-complete που ανήκει στην κλάση NP θα χρειαστούμε την έννοια της αναγωγής.

Ορισμός 1.9. *Έστω δύο προβλήματα A και B και ένας αλγόριθμος X που λύνει το B. Αν υπάρχει αλγόριθμος που λύνει το πρόβλημα A και εσωτερικά καλεί τον αλγόριθμο X, τότε λέμε ότι ανάγουμε το πρόβλημα A στο B και γράφουμε $A \leq B$.*

Αν έχουμε δηλαδή έναν αλγόριθμο που λύνει το πρόβλημα B, τότε έχουμε και έναν αλγόριθμο που λύνει το πρόβλημα A. Επομένως λέμε ότι το πρόβλημα A είναι πιο εύκολο από το B. Μια ειδική περίπτωση της από πάνω αναγωγής είναι η many-one αναγωγή.

Ορισμός 1.10. *Έστω δύο προβλήματα A και B και $w, f(w)$ να είναι δύο instances του A και του B αντίστοιχα, όπου f είναι μια συνάρτηση που μετατρέπει στοιχεία του A σε στοιχεία του B. Μία* many-one αναγωγή *από το A στο B είναι η συνάρτηση f αν ισχύει ότι το w ανήκει στο A αν και μόνο αν το $f(w)$ ανήκει στο B.*

Με την έννοια της συνάρτησης εννοούμε ότι αν υπάρχει αλγόριθμος που λύνει το πρόβλημα B τότε μπορούμε να μετατρέψουμε την είσοδο του προβλήματος A με την χρήση της συνάρτησης κατάλληλα και να καλέσουμε μία φορά στο τέλος τον αλγόριθμο για το B ο οποίος θα μας δώσει μια απάντηση για το αρχικό πρόβλημα που είναι το A. Στην many-one αναγωγή έχουμε τον περιορισμό ότι καλούμε τον αλγόριθμο για το B μόνο μία φορά στο τέλος, σε αντίθεση με την γενική αναγωγή όπου δεν έχουμε τέτοιους περιορισμούς. Για τις κλάσεις πολυπλοκότητας όμως δεν μας ενδιαφέρει αν υπάρχει μόνο αναγωγή, αλλά και αν ο αλγόριθμος που προκύπτει τρέχει σε πολυωνυμικό χρόνο.

Ορισμός 1.11. *Έστω δύο προβλήματα A και B. Μία* πολυωνυμική many-one αναγωγή *από το A στο B είναι ένας πολυωνυμικός αλγόριθμος ο οποίος μετατρέπει την είσοδο του A κατάλληλα έτσι ώστε αν κληθεί ο αλγόριθμος που λύνει το B με την τροποποιημένη είσοδο, η έξοδος να είναι ίδια με αυτήν του πρόβληματος A.*

Ορισμός 1.12. *Ένα πρόβλημα Π ανήκει στην κλάση* NP*-complete αν το Π ανήκει στην* NP *και κάθε άλλο πρόβλημα που ανήκει στην κλάση* NP *έχει πολυωνυμική many-one αναγωγή στο πρόβλημα Π.*

Για να δείξουμε ότι ένα πρόβλημα Π ανήκει στην κλάση NP-complete χρειάζεται απλώς να δείξουμε ότι υπάρχει πολυωνυμική many-one αναγωγή από ένα γνωστό πρόβλημα της κλάσης NP-complete στο Π. Η κλάση NP-complete περιέχει τα πιο δύσκολα προβλήματα της κλάσης NP. Αυτό γιατί αν ένα πρόβλημα A ανήκει στην κλάση NP και ένα πρόβλημα

B ανήκει στην κλάση NP-complete, τότε λύνοντας το B σε πολυωνυμικό χρόνο μπορούμε να λύσουμε και το A σε πολυωνυμικό χρόνο κάνοντάς το αναγωγή στο B. Αυτό σημαίνει ότι αν καταφέρουμε και λύσουμε ένα πρόβλημα που ανήκει στην κλάση NP-complete σε πολυωνυμικό χρόνο τότε όλα τα προβλήματα της κλάσης NP θα λύνονται σε πολυωνυμικό χρόνο, κάτι που φυσικά δεν έχει συμβεί ακόμα, χωρίς όμως να έχει αποδειχθεί ότι δεν είναι δυνατόν να συμβεί στο μέλλον. Με βάση τα παραπάνω αν για ένα πρόβλημα που ανήκει στην κλάση NP αλλά όχι στην κλάση NP-complete βρεθεί πολυωνυμικός αλγόριθμος αυτό δεν θα σημαίνει ότι όλα τα προβλήματα που ανήκουν στην κλάση NP λύνονται σε πολυωνυμικό χρόνο. Γενικά για τα προβλήματα που έχει βρεθεί πολυωνυμικός αλγόριθμος που να τα λύνει λέμε ότι είναι εύκολα, ενώ τα προβλήματα που ανήκουν στην κλάση NP-complete λέμε ότι είναι δύσκολα.

Τα NP-complete προβλήματα είναι δύσκολα που σημαίνει ότι δεν έχει βρεθεί αλγόριθμος που να τα λύνει σε πολυωνυμικό χρόνο ως προς το μέγεθος της εισόδου τους. Παρόλα αυτά όμως μπορούμε να προσπαθήσουμε να τα λύσουμε με κάποιες διαφορετικές τεχνικές που είναι πιο ελαστικές είτε ως προς την ακρίβεια της λύσης είτε ως προς τον χρόνο. Μια κλάση προβλημάτων η οποία είναι ελαστική ως προς τον χρόνο είναι η FPT (fixed parameter tractable).

Ορισμός 1.13. *Ένα παραμετροποιημένο πρόβλημα Π είναι* FPT *αν υπάρχει αλγόριθμος που το λύνει σε χρόνο $O(f(k) \cdot n^c)$, όπου $f(k)$ είναι μια συνάρτηση που εξαρτάται μόνο ως προς το k, το n είναι το μέγεθος της εισόδου και το c είναι μια σταθερά.*

Είναι σημαντικό το ότι το c είναι σταθερά και όχι μεταβλητή. Δηλαδή κάτι της μορφής n^k για την μεταβλητή k δεν θα ήταν επιτρεπτό. Το $f(k)$ είναι μια αυθαίρετη συνάρτηση που σημαίνει ότι μπορεί να είναι εκθετική ως προς k. Αν όμως θεωρήσουμε το k ως σταθερά τότε και το 2^k για παράδειγμα γίνεται σταθερά. Ουσιαστικά για τα προβλήματα που ανήκουν στην κλάση FPT μπορούμε να βρούμε μια μεταβλητή έτσι ώστε όλη η δυσκολία του προβλήματος να εξαρτάται από αυτήν την μεταβλητή και όχι από το μέγεθος της εισόδου. Επομένως για μικρές τιμές της μεταβλητής μπορούμε να την θεωρήσουμε σταθερά και να πάρουμε πολυωνυμικό αλγόριθμο ακόμα και αν το μέγεθος της εισόδου είναι πολύ μεγάλο. Αυτό βέβαια δεν σημαίνει ότι το πρόβλημα γίνεται εύκολο, καθώς αν δεν θεωρήσουμε το k ως σταθερά θα πρέπει να το δώσουμε ως είσοδο και τότε το $O(f(k))$ για $f(k) = 2^k$ για παράδειγμα δεν θα είναι πολυωνυμικό ως προς το μέγεθος της εισόδου. Ο λόγος που θεωρούμε το k ως σταθερά είναι επειδή πολλά προβλήματα πρακτικά έχουν φραγμένο και μικρό k που είναι ανεξάρτητο από το μέγεθος του γραφήματος. Θεωρητικά όμως το k είναι μεταβλητή και μπορεί να μεγαλώνει αρκετά για συγκεκριμένες κατηγορίες γραφημάτων.

1.2 Εισαγωγή στο δενδροπλάτος

Τα δένδρα είναι μια κατηγορία γραφημάτων στην οποία πολλά δύσκολα προβλήματα λύνονται σε πολυωνυμικό χρόνο. Ένα γενικό γράφημα μπορεί να περιέχει κύκλους και να είναι αρκετά πολύπλοκο. Η έννοια της δενδροαποσύνθεσης είναι μια προσπάθεια να μετατρέψουμε το γραφημά μας σε δέντρο, δημιουργώντας ένα δέντρο που κάθε του κορυφή

αντιστοιχεί σε ένα υπογράφημα του αρχικού γραφήματος. Αυτό το δέντρο είναι η δενδροαποσύνθεση του αρχικού γραφήματος και τις κορυφές του τις ονομάζουμε τσάντες επειδή κάθε κορυφή, της δενδροαποσύνθεσης μπορεί να περιέχει πολλές κορυφές του αρχικού γραφήματος. Το μέγεθος της μεγαλύτερης τσάντας μας μείον ένα ισούται με το δενδροπλάτος της δενδροαποσύνθεσης, το οποίο είναι ένα μέτρο που μας δείχνει πόσο πολύ μοιάζει το αρχικό μας γράφημα με δέντρο. Ανάλογα πόσο πολύ μοιάζει το αρχικό μας γράφημα με δέντρο τόσο πιο μικρές σε μέγεθος μπορούν να γίνουν οι τσάντες της δενδροαποσύνθεσης και τόσο πιο μικρό είναι το δενδροπλάτος.

Η έννοια του δενδροπλάτους και της δενδροαποσύνθεσης εισήχθη από τους Robertson και Seymour και παίζει πολύ σημαντικό ρόλο στην θεωρία των graph minors και στην απόδειξη διάφορων θεωρημάτων που απέδειξαν. Ένα σημαντικό αποτέλεσμα της θεωρίας των graph minors είναι ότι κάθε κλάση που είναι κλειστή ως προς ελάσσονα γραφήματα (δηλαδή κάθε έλασσον γράφημα ανήκει στην ίδια κλάση με το αρχικό γράφημα) έχει ένα πεπερασμένο σύνολο απαγορευμένων ελασσόνων γραφημάτων. Αυτό σημαίνει ότι αν για κάθε G που ανήκει στην κλάση F ισχύει ότι και κάθε H που είναι έλασσον του G ανήκει στην κλάση F, τότε υπάρχει ένα πεπερασμένο σύνολο A από απαγορευμένα γραφήματα. Και από το θεώρημα γνωρίζουμε ότι ένα άλλο γράφημα G' ανήκει στην F αν και μόνο αν δεν περιέχει ως έλασσον κάποιο γράφημα από το σύνολο A. Αρχικά ο Kruskal απέδειξε κάτι παρόμοιο για τα δάση τα οποία αποτελούνται από συνεκτικά δένδρα. Στην συνέχεια οι Robertson και Seymour με την έννοια του δενδροπλάτους και το ότι γραφήματα με φραγμένο δενδροπλάτος έχουν μια δομή που μοιάζει με δέντρο (η δενδροαποσύνθεση) αποδείξαν το θεώρημα για γενικά γραφήματα.

Το δενδροπλάτος (treewidth) πέρα από τους θεωρητικούς λόγους, είναι μια σημαντική έννοια στην θεωρία γραφημάτων και για πρακτικούς λόγους. Υπάρχουν πολλά θεωρητικά αποτελέσματα που δείχνουν ότι πολλά προβλήματα που είναι NP-complete σε γενικά γραφήματα, μπορούν να επιλυθούν σε πολυωνυμικό χρόνο για κλάσεις γραφημάτων με φραγμένο δενδροπλάτος. Ο λόγος είναι ότι όταν το γραφημά μας είναι δέντρο πολλά προβλήματα λύνονται εύκολα με τεχνικές, όπως για παράδειγμα ο δυναμικός προγραμματισμός και όταν γνωρίζουμε ότι το δενδροπλάτος ενός γραφήματος G είναι μικρό, μπορούμε να τροποποιήσουμε ανάλογα τον αλγόριθμο για να λύσουμε το πρόβλημα στο G σε πολυωνυμικό χρόνο ως προς το μέγεθος του G. Δύο από τα πολλά τέτοια προβλήματα είναι το Maximum Independent Set και το Vertex Cover. Η λογική που ακολουθείται σε αυτά τα προβλήματα είναι η ίδια με αυτήν που ακολουθείται στα δένδρα. Στα δένδρα για κάθε κορυφή του γραφήματος κρατάμε μια πληροφορία (δυναμικός προγραμματισμός) και για κάθε καινούργια κορυφή ανανεώνουμε αυτήν την πληροφορία μέσω των παιδιών της τρέχουσας κορυφής. Το γεγονός ότι δεν υπάρχει κύκλος μας διαβεβαιώνει ότι η καινούργια πληροφορία που θα βρούμε για την τρέχουσα κορυφή δεν θα επηρεάσει κάποια πληροφορία που έχουμε ήδη αποθηκεύσει σε κάποιον απόγονο αυτής. Με την ίδια λογική, για ένα γενικό γράφημα G αν έχουμε την δενδροαποσυνθεσή του μπορούμε να ακολουθήσουμε τον προηγούμενο αλγόριθμο τρεχοντάς τον για την δενδροαποσύνθεση του G. Η διαφορά είναι ότι η δενδροαποσύνθεση ως κορυφές έχει τσάντες, επομένως θα πρέπει να ελέγξουμε την σχέση που έχουν οι κορυφές που ανήκουν στην ίδια τσάντα. Αυτός είναι και ο λόγος για τον οποίο όσο πιο μικρό δενδροπλάτος έχουμε τόσο πιο γρήγορος θα είναι

και ο αλγόριθμος.

Επιπλέον το δενδροπλάτος συνδέεται άμεσα με πολλές άλλες σημαντικές έννοιες στην θεωρία γραφημάτων. Μία από αυτές είναι η έννοια του διαχωριστή. Με τον τρόπο που θα οριστεί παρακάτω η δενδροαποσύνθεση, φαίνεται ότι οι τσάντες είναι πιθανοί διαχωριστές του αρχικού γραφήματος και μια δενδροαποσύνθεση με μικρό δενδροπλάτος μας δηλώνει ότι το αρχικό γράφημα περιέχει πολλούς μικρούς κορυφοδιαχωριστές. Το να βρούμε το δενδροπλάτος ενός γραφήματος είναι δύσκολο πρόβλημα και συγκεκριμένα NP-complete και δείχθηκε στο [1] για μια ισοδύναμη κλάση γραφημάτων που είναι τα partial k-trees όπως θα δούμε και παρακάτω. Παρόλα αυτά όμως υπάρχουν κλάσεις γραφημάτων όπως για παράδειγμα τα chordal γραφήματα για τις οποίες η εύρεση του δενδροπλάτους γίνεται εύκολο πρόβλημα. Το δενδροπλάτος σχετίζεται με πολλές έννοιες σε θεωρητικό επίπεδο και υπάρχουν πολλά θεωρητικά αποτελέσματα που δείχνουν το πόσο στενά συνδεδεμένες είναι όλες οι έννοιες μαζί.

Πιο τυπικά το πρόβλημα που ορίζεται ως "Δίνεται ένα γράφημα $G = (V, E)$ και ένας ακέραιος k. Βρες αν το δενδροπλάτος του G είναι το πολύ k" είναι NP-complete. Αυτό μας λέει ότι το πρόβλημα της εύρεσης του δενδροπλάτους ενός γραφήματος είναι δύσκολο πρόβλημα και μάλιστα είναι από τα πιο δύσκολα που ανήκουν στην κλάση NP. Παρόλα αυτά τέτοιου είδους προβλήματα μπορούμε να τα λύσουμε αν γίνουμε πιο ελαστικοί είτε στην ακρίβεια της λύσης είτε στον χρόνο. Συγκεκριμένα υπάρχει πολυωνυμικός αλγόριθμος που δίνεται και παρακάτω, ο οποίος λύνει το πρόβλημα προσεγγιστικά και τα αποτελέσματα δεν είναι ποτέ χειρότερα από $O(\log n)$ της βέλτιστης λύσης. Επίσης αν δούμε το k ως σταθερά και όχι ως μεταβλητή τότε οι παράγοντες που είναι εκθετικοί ως προς k είναι και αυτοί σταθερές. Θέτοντας μια μεταβλητή ως σταθερά και λύνοντας το πρόβλημα πολυωνυμικά ως προς της άλλες μετατρέπει το πρόβλημα σε FPT (fixed parameter tractable). Με βάση αυτό αν θεωρήσουμε το k σταθερά, υπάρχει πολυωνυμικός και συγκεκριμένα γραμμικός αλγόριθμος που λύνει το πρόβλημα. Επειδή όμως η σταθερά που βρίσκεται στο big-O μπορεί να είναι τεράστια έχει αναπτυχθεί αλγόριθμος που δίνεται και παρακάτω, ο οποίος τρέχει σε πολυωνυμικό χρόνο για σταθερό k και μας λέει ότι είτε το γράφημα έχει δενδροπλάτος μεγαλύτερο του k είτε δίνει μια δενδροαποσύνθεση με δενδροπλάτος το πολύ $4 \cdot k + 4$. Αυτός ο αλγόριθμος είναι ελαστικός και ως προς τον χρόνο και ως προς το αποτέλεσμα και για αυτό είναι FPT προσεγγιστικός. Κάτι ενδιαφέρον είναι ότι το θεώρημα στο οποίο χρησιμοποιήθηκε το δενδροπλάτος για την αποδειξή του, μπορεί να χρησιμοποιηθεί για να αποδείξουμε ότι το πρόβλημα που ορίστηκε από πάνω είναι FPT. Το θεώρημα στο οποίο χρησιμοποιήθηκε η έννοια του δενδροπλάτους από τους Robertson και Seymour μας λέει ότι μια κλάση που είναι κλειστή ως προς ελάσσονα γραφήματα έχει ένα πεπερασμένο σύνολο απαγορευμένων ελασσόνων γραφημάτων. Όπως θα φανεί και παρακάτω το δενδροπλάτος είναι κλειστό ως προς ελάσσον γραφήματα. Δηλαδή αν το H είναι έλασσον του G τότε ισχύει ότι το δενδροπλάτος του H είναι μικρότερο ή ίσο του G. Επομένως αν ορίσουμε μια κλάση γραφημάτων G_k η οποία περιέχει όλα τα γραφήματα με δενδροπλάτος μικρότερο ή ίσο του k, αυτή η κλάση είναι κλειστή ως προς ελάσσονα γραφήματα και έτσι υπάρχει ένα πεπερασμένο σύνολο A από απαγορευμένα ελάσσον γραφήματα. Επιπλέον υπάρχει αλγόριθμος που ελέγχει αν ένα γράφημα είναι έλασσον ενός άλλου σε $O(|V|^3)$. Οπότε μπορούμε να λύσουμε το αρχικό μας πρόβλημα

σε χρόνο $O(f(k) \cdot |V|^3)$, όπου $f(k)$ είναι μια συνάρτηση που εξαρτάται μόνο από το k, καθώς το μέγεθος του A εξαρτάται μόνο από το δενδροπλάτος και όχι από το μέγεθος του γραφήματος. Οπότε για σταθερό k έχουμε έναν πολυωνυμικό αλγόριθμο που λύνει το πρόβλημα που ορίσαμε. Βέβαια το σύνολο A είναι γνωστό για πολύ μικρές τιμές του k και ούτε γνωρίζουμε ποιο είναι για όλα τα k ούτε πόσο γρήγορα μεγαλώνει. Για αυτό αυτός ο αλγόριθμος θεωρητικά είναι χρήσιμος αλλά πρακτικά δεν μπορεί να υλοποιηθεί.

Το δενδροπλάτος και η δενδροαποσύνθεση μπορούν να χρησιμοποιηθούν για να λυθούν και πραγματικά προβλήματα στην βιολογία, στην γραμμική άλγεβρα, στην επεξεργασία της φυσικής γλώσσας και σε πολλούς άλλους τομείς. Ο λόγος είναι ότι για πολλά τέτοια προβλήματα έχει παρατηρηθεί ότι τα γραφήματα που προκύπτουν έχουν μικρό και φραγμένο δενδροπλάτος επομένως μπορούμε να θεωρήσουμε το k ως σταθερά και να λύσουμε το πρόβλημα σε πολυωνυμικό χρόνο. Επίσης σε πολλά πρακτικά προβλήματα όπως για παράδειγμα στο Cholesky factorization αυτό που θέλουμε να βρούμε έχει άμεση σχέση με το δενδροπλάτος, οπότε μια γρήγορη λύση για το δενδροπλάτος δίνει και μία γρήγορη λύση για το προβλημά μας. Πιο αναλυτικά για την σχέση του δενδροπλάτους και των πραγματικών προβλημάτων μπορεί κάποιος να δει στο [4].

2. ΔΕΝΔΡΟΠΛΑΤΟΣ ΚΑΙ Η ΣΧΕΣΗ ΤΟΥ ΜΕ ΑΛΛΕΣ ΠΑΡΑΜΕΤΡΟΥΣ

2.1 Δενδροπλάτος

Για αρχή θα πρέπει να δώσουμε έναν ορισμό για το τι είναι το δενδροπλάτος και τι είναι η δενδροαποσύνθεση ενός γραφήματος και στην συνέχεια αναφέρουμε διάφορα λήμματα σχετικά με τις ιδιότητες του δενδροπλάτους. Διαισθητικά το δενδροπλάτος είναι μια ποσότητα που μας λέει πόσο μοιάζει ένα γραφημά G με δέντρο και το συμβολίζουμε με $tw(G)$. Όσο μικρότερο είναι το δενδροπλάτος του G τόσο πιο πολύ το G μοιάζει με δέντρο. Επομένως για κάθε δέντρο T θέλουμε να ισχύει ότι $tw(T) = 1$.

Ορισμός 2.1. *Μία* δενδροαποσύνθεση *του* $G = (V, E)$ *είναι ένα ζευγάρι* $(T = (I, F), \{X_i \mid i \in I\})$ *όπου το* T *είναι δέντρο με κορυφές το σύνολο* I *και ακμές το σύνολο* F *και το* $\{X_i \mid i \in I\}$ *είναι οικογένεια υποσυνόλων του* V*, τέτοια ώστε*

1. $\bigcup_{i \in I} X_i = V$.
2. $\forall \{u, w\} \in E, \exists i \in I$ *τέτοιο ώστε* $u \in X_i, w \in X_i$.
3. $\forall u \in V$ *το σύνολο* $\{i \in I \mid u \in X_i\}$ *ενάγει ένα συνεκτικό υπογράφημα του* T.

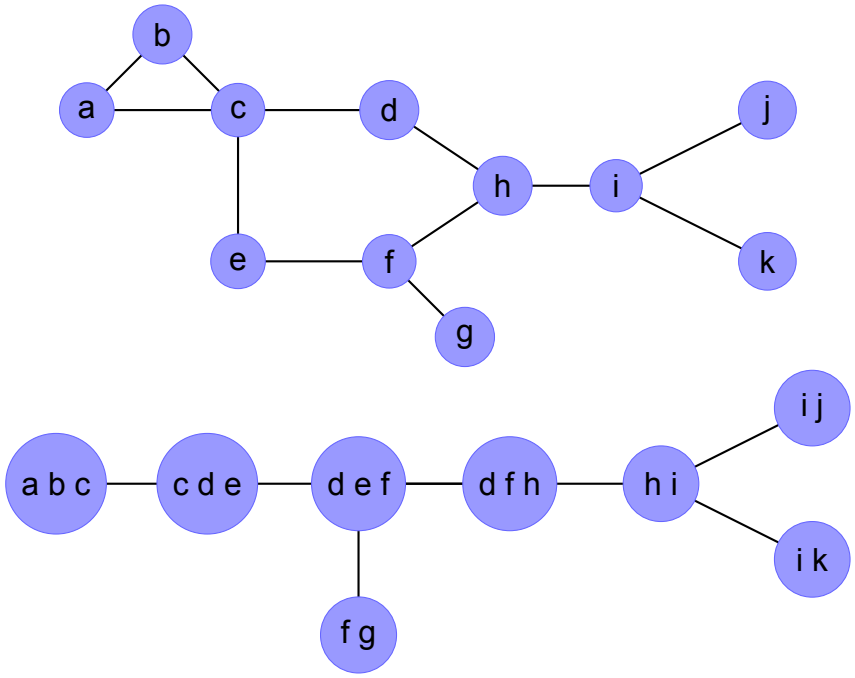
Η τρίτη συνθήκη μπορεί να αντικατασταθεί με την ισοδύναμη

- $\forall i, j, k \in I$, αν το j βρίσκεται στο μοναδικό μονοπάτι $i - k$ στο T, τότε $X_i \cap X_k \subseteq X_j$

Τα X_i θα τα ονομάζουμε τσάντες επειδή μπορούν να περιέχουν πολλές κορυφές από το αρχικό γράφημα και κάθε X_i αντιστοιχεί στην κορυφή i της δενδροαποσύνθεσης T. Η πρώτη συνθήκη μας επιβάλλει να εισάγουμε όλες τις κορυφές του V σε τουλάχιστον μία τσάντα X_i. Η δεύτερη συνθήκη μας επιβάλλει για κάθε ακμή $\{u, w\} \in E$ να υπάρχει τουλάχιστον μία τσάντα X_i που να περιέχει και την κορυφή u και την κορυφή w. Με αυτό τον τρόπο διατηρούνται οι σχέσεις των κορυφών του G και στην δενδροαποσύνθεση.
Η τρίτη συνθήκη μας λέει ότι αν μια κορυφή βρίσκεται σε μια τσάντα X_i και σε μια τσάντα X_k, τότε αναγκαστικά θα πρέπει να βρίσκεται και σε όλες τις τσάντες X_j, όπου j ανήκει στο μοναδικό μονοπάτι $i - k$ στο T. Θυμίζουμε ότι σε ένα δέντρο υπάρχει μόνο ένα απλό μονοπάτι μεταξύ δύο κορυφών και η δενδροαποσυνθεσή μας είναι δέντρο.

Το δενδροπλάτος της δενδροαποσύνθεσης $(T, \{X_i \mid i \in I\})$ ισούται με $max|X_i| - 1$. Δηλαδή ισούται με το πλήθος της μεγαλύτερης τσάντας X_i μείον 1. Το δενδροπλάτος του G ισούται με το μικρότερο δενδροπλάτος από όλες τις πιθανές δενδροαποσυνθέσεις του G. Σύμφωνα με τον παραπάνω ορισμό καταλαβαίνουμε γιατί η τρίτη συνθήκη είναι σημαντική. Χωρίς αυτή θα μπορούσαμε να φτιάξουμε για κάθε γράφημα μια δενδροαποσύνθεση με $tw(G) = 1$, χάνοντας έτσι το μέτρο κατά πόσο ένα γράφημα μοιάζει με δέντρο. Ο τρόπος που θα φτιάχναμε μια τέτοια δενδροαποσύνθεση θα ήταν εισάγοντας για κάθε ακμή $\{u, w\}$ του G, μία τσάντα X_i που να περιέχει μόνο την u και w. Αυτό όμως δεν μπορεί να

συμβεί όταν για μια κορυφή u το υπογράφημα που ενάγεται από τα X_i που περιέχουν την κορυφή u θα πρέπει να είναι συνεκτικό, δεδομένου ότι και το T θα πρέπει να είναι δέντρο και δεν μπορεί να περιέχει κύκλο. Για το δενδροπλάτος δεν έχει σημασία πόσες κορυφές έχει το δέντρο T, αλλά το πόσο μεγάλη είναι η μεγαλύτερη τσάντα X_i. Επιπλέον ο λόγος που υπάρχει το -1 στον ορισμό του δενδροπλάτους είναι επειδή θέλουμε το δέντρο να έχει δενδροπλάτος ίσο με 1 και για μια ακμή και μόνο αναγκαζόμαστε να φτιάξουμε μια τσάντα με δύο κορυφές λόγω της δεύτερης συνθήκης.



Σχήμα 1: Από πάνω φαίνεται ένα γράφημα G και από κάτω μια δενδροαποσύνθεσή του με tw(G) = 2.

Παρακάτω αναφέρονται κάποια λήμματα σχετικά με το δενδροπλάτος τα οποία θα βοηθήσουν στην καλύτερη κατανόηση του. Επιπλέον με την χρήση κάποιων λημμάτων μπορούμε να είμαστε σίγουροι ότι για ένα γράφημα G το δενδροπλάτος του δεν μπορεί να είναι μικρότερο από κάποια τιμή όταν περιέχει κάποια συγκεκριμένα υπογραφήματα όπως για παράδειγμα κύκλο ή κλικα.

Λήμμα 2.1. *Αν $G = (V, E)$ είναι δέντρο με τουλάχιστον μία ακμή τότε $tw(G) = 1$.*

Απόδειξη. Για κάθε ακμή $\{u, w\} \in E$ φτιάξε μια τσάντα X_i που περιέχει μόνο τις κορυφές u και w. Ένωσε δύο τσάντες μεταξύ τους μόνο αν έχουν μια κοινή κορυφή. Η δενδροαποσύνθεση που προκύπτει ικανοποιεί και τις τρεις συνθήκες και έχει $tw(G) = 1$. □

Λήμμα 2.2. *Έστω $(T = (I, F), \{X_i \mid i \in I\})$ δενδροαποσύνθεση του $G = (V, E)$ και $W \subseteq V$ να είναι κλίκα στο G. Τότε $\exists i \in I$ τέτοιο ώστε $W \subseteq X_i$. Δηλαδή το W ανήκει ολόκληρο σε τουλάχιστον μία τσάντα με συνέπεια $tw(G) \geq |W| - 1$.*

Απόδειξη. Για κάθε $u \in V$, ορίζουμε το υπογράφημα $T_u = \{i \in I \mid u \in X_i\}$. Το υπογράφημα αυτό είναι συνεκτικό λόγω της τρίτης συνθήκης. Αν $u, w \in W$ τότε υπάρχει μία τσάντα X_i που περιέχει και την u και την w, οπότε $V(T_u) \cap V(T_w) \neq \emptyset$. Επειδή όμως οι κορυφές που ανήκουν στο W ανά δύο ενώνονται με μία ακμή, τότε η τομή ανά δύο τέτοιων υπογραφημάτων δεν θα είναι κενή. Επίσης στα δένδρα ικανοποιείται η ιδιότητα Helly, οπότε θα πρέπει να ισχύει ότι $\bigcap_{u \in W} V(T_u) \neq \emptyset$. Επομένως $\exists i \in I$, τέτοιο ώστε $W \subseteq X_i$. □

Λήμμα 2.3. *Αν H είναι έλασσον του G, τότε ισχύει $tw(H) \leq tw(G)$.*

Λήμμα 2.4. *Αν το $G = (V, E)$ περιέχει κύκλο τότε $tw(G) \geq 2$.*

Απόδειξη. Από το Λήμμα 2.2 συμπεραίνουμε ότι η κλίκα K_3 έχει δενδροπλάτος ίσο με 2. Η κλίκα K_3 είναι έλασσον ενός γραφήματος που περιέχει κύκλο και από το Λήμμα 2.3 γνωρίζουμε ότι $tw(K_3) \leq tw(G) \Rightarrow tw(G) \geq 2$. □

Λήμμα 2.5. *[3] Έστω $(T = (I, F), \{X_i \mid i \in I\})$ δενδροαποσύνθεση του $G = (V, E)$. Αν το εναγόμενο γράφημα $G[W_1 \cup W_2]$ είναι ένα πλήρες διμερές γράφημα, τότε $\exists i \in I$ τέτοιο ώστε $W_1 \subseteq X_i$ ή $W_2 \subseteq X_i$. Δηλαδή τουλάχιστον ένα από τα δύο ανήκει ολόκληρο σε τουλάχιστον μία τσάντα με συνέπεια $tw(G) \geq min(|W_1|, |W_2|) - 1$.*

Απόδειξη. Θα χρησιμοποιηθούν κάποιες ιδιότητες των chordal γραφημάτων που εξηγούνται παρακάτω. Ας υποθέσουμε ότι δεν υπάρχει τσάντα που να περιέχει όλες τις κορυφές του W_1 ή του W_2. Αν προσθέσουμε όλες εκείνες τις ακμές μεταξύ των κορυφών που βρίσκονται στην ίδια τσάντα προκύπτει το γράφημα G' το οποίο θα πρέπει να είναι chordal. Αν όμως δεν ανήκει όλο το W_1 σε μία τσάντα τότε θα υπάρχουν δύο διαφορετικές κορυφές $a_1, b_1 \in W_1$ που δεν θα ενώνονται με ακμή γιατί και δεν θα βρίσκονται στην ίδια τσάντα της δενδροαποσύνθεσης και εξαρχής το εναγόμενο υπογράφημα ήταν διμερές. Αντίστοιχα θα ισχύει το ίδιο και για δύο διαφορετικές κορυφές $a_2, b_2 \in W_2$. Τότε όμως επειδή το διμερές γράφημα ήταν πλήρες θα έχουμε έναν κύκλο a_1, a_2, b_1, b_2 μεγέθους 4. Αυτό όμως σημαίνει ότι το G' δεν είναι chordal, που είναι άτοπο. Επομένως θα πρέπει να υπάρχει τσάντα που περιέχει ολόκληρο είτε το W_1 είτε το W_2. □

Λήμμα 2.6. *Έστω $(T = (I, F), \{X_i \mid i \in I\})$ δενδροαποσύνθεση του $G = (V, E)$. Αν αφαιρέσουμε μια ακμή $e = \{u, w\}$ του T, τότε το $T \setminus e$ είναι δάσος που αποτελείται από δυο συνιστώσες, την T_u που περιέχει το u και την T_w που περιέχει το w. Ορίζουμε το $A = \bigcup_{i \in T_u} X_i$ και το $B = \bigcup_{i \in T_w} X_i$. Τότε το (A, B) είναι διαχωρισμός του G, με διαχωριστή το $X_u \cap X_w$.*

Αυτό το λήμμα είναι πολύ σημαντικό για την ανάπτυξη αλγορίθμων που βρίσκουν μια δενδροαποσύνθεση. Ουσιαστικά το λήμμα μας πληροφορεί ότι για δύο γειτονικές τσάντες η τομή τους ορίζει έναν κορυφοδιαχωριστή του G. Αυτό σημαίνει ότι και μία ολόκληρη εσωτερική τσάντα αν αφαιρέσουμε, τότε αυτές οι κορυφές της τσάντας ορίζουν έναν κορυφοδιαχωριστή του G. Επομένως ένα γράφημα με μικρό δενδροπλάτος είναι ένα γράφημα με πολλούς μικρούς κορυφοδιαχωριστές. Οπότε μέσω αυτής της ιδιότητας οι αλγόριθμοι χρησιμοποιούν την λογική διαίρει και βασίλευε ώστε να σπάνε το αρχικό γράφημα σε μικρότερα γραφήματα, να βρίσκουν για αυτά μια δενδροαποσύνθεση και μετά να ενώνουν

τα αναδρομικά αποτελέσματα για να βρουν την δενδροαποσύνθεση του αρχικού γραφήματος.

Λήμμα 2.7. *Έστω $(T = (I, F), \{X_i \mid i \in I\})$ δενδροαποσύνθεση του $G = (V, E)$. Αν το G είναι k-συνεκτικό, ισχύει ότι $tw(G) \geq k$.*

Απόδειξη. Αν υπάρχουν δύο συνεχόμενες τσάντες X_i, X_j, ώστε $X_i = X_j$, τότε μπορούμε να αφαιρέσουμε μία από αυτές χωρίς να χαλάσει η δενδροαποσύνθεση. Ισχύει ότι $|X_i| \leq tw(G) + 1, |X_j| \leq tw(G) + 1, |X_i \cap X_j| \leq tw(G)$. Αλλά το $X_i \cap X_j$ είναι διαχωριστής του G, επομένως αν το G είναι k-συνεκτικό, ισχύει ότι $tw(G) \geq k$. □

Λήμμα 2.8. *[2] Για $G = (V, E)$ αν $tw(G) \leq k$, τότε $|E| \leq k \cdot |V| - \frac{1}{2} \cdot k \cdot (k + 1)$.*

Λήμμα 2.9. *Αν $(T = (I, F), \{X_i \mid i \in I\})$ δενδροαποσύνθεση του $G = (V, E)$ και $\exists i \in I : u, w \in X_i, u \neq w$, τότε η ίδια δενδροαποσύνθεση ικανοποιεί και τις τρεις συνθήκες για το γράφημα $G + (u, w) = (V, E \cup \{(u, w)\})$. Αν δηλαδή δύο διαφορετικές κορυφές ανήκουν σε μία τσάντα χωρίς να ενώνονται στο G, τότε αν στο καινούργιο γράφημα $G + (u, w)$ οι κορυφές ενώνονται με ακμή, η δενδροαποσύνθεση του G παραμένει έγκυρη και για το καινούργιο γράφημα $G + (u, w)$.*

Λήμμα 2.10. *[2] Έστω δύο κορυφές u και w στο $G = (V, E)$ οι οποίες έχουν τουλάχιστον $k + 1$ κοινούς γείτονες. Αν το δενδροπλάτος του G είναι το πολύ k, τότε το δενδροπλάτος του $G + (u, w)$ είναι επίσης το πολύ k.*

Απόδειξη. Έστω W το σύνολο των κοινών γειτόνων των u και w. Το εναγόμενο γράφημα $G[u \cup w \cup W]$ είναι ένα πλήρες διμερές γράφημα. Από Λήμμα 2.5 είτε $\exists i \in I : u, w \in X_i$ και από Λήμμα 2.9 το δενδροπλάτος του $G + (u, w)$ είναι το πολύ k, είτε $\exists i \in I : W \in X_i$. Αν όμως ισχύει η δεύτερη υπόθεση, τότε από Λήμμα 2.9 αν ενώσουμε όλες τις κορυφές του W θα πρέπει το καινούργιο γράφημα να έχει δενδροπλάτος το πολύ k. Τότε όμως το εναγόμενο γράφημα $G[u \cup W]$ θα είναι μια κλίκα μεγέθους $k + 2$ και από Λήμμα 2.2 το καινούργιο γράφημα θα έχει δενδροπλάτος τουλάχιστον $k + 1$. Άτοπο. Επομένως δεν μπορεί να ισχύει η δεύτερη υπόθεση και το γράφημα $G+(u, w)$ έχει δενδροπλάτος το πολύ k. □

Για το γράφημα στο Σχήμα 1 αν χρησιμοποιήσουμε το Λήμμα 2.4 επειδή το γράφημα έχει κύκλο θα πρέπει $tw(G) \geq 2$. Επειδή όμως βρήκαμε μία έγκυρη δενδροαποσύνθεση(δηλαδή ικανοποιεί και τις τρεις συνθήκες του ορισμού 2.1) με $tw(G) = 2$, τότε μπορούμε να είμαστε σίγουροι ότι δεν γίνεται να βρούμε κάποια δενδροαποσύνθεση με μικρότερο δενδροπλάτος.

2.2 Partial k-trees

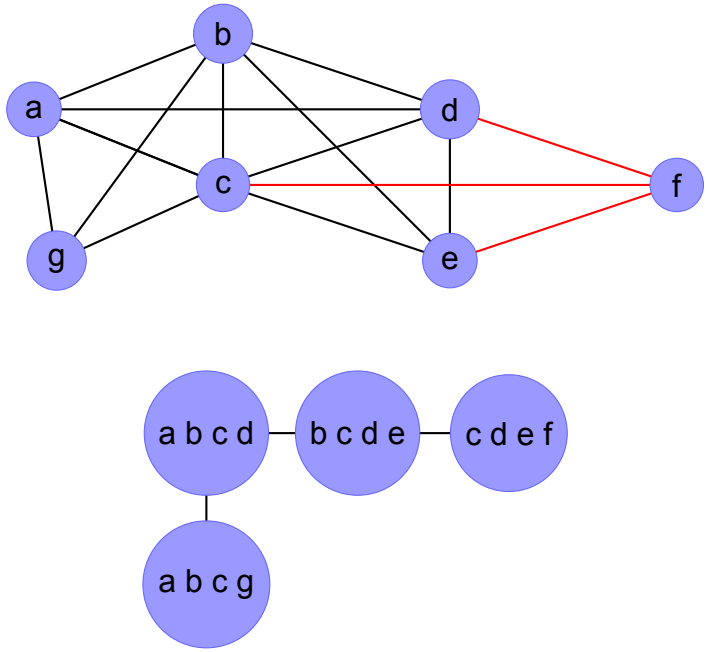
Τα partial k-trees είναι ισοδύναμα με τα γραφήματα που έχουνε δενδροπλάτος το πολύ k. Επομένως, κάθε αλγόριθμος που τρέχει σε πολυωνυμικό χρόνο για γραφήματα φραγμένου δεδροπλάτους, τρέχει σε πολυωνυμικό χρόνο και για τα partial k-trees. Τα partial

k-trees είναι σημαντικά καθώς μας δίνουν μια καλύτερη εικόνα για το πως σχετίζεται το δενδροπλάτος με την κάθε οικογένεια γραφημάτων.

Ορισμός 2.2. *Η ομάδα των* k-trees *ορίζεται αναδρομικά ως εξής:*

- *Το* K_k*(η κλίκα με* k *κορυφές) είναι* k*-tree.*
- *Ένα* k*-tree* G *με* $n+1$ *κορυφές* $(n \geq k)$ *μπορεί να κατασκευαστεί από ένα* k*-tree* H *με* n *κορυφές, ενώνοντας την καινούργια κορυφή του* G *με* k *κορυφές που μεταξύ τους σχηματίζουν μία κλίκα στο* H*.*

Τα k-trees έχουν δενδροπλάτος ίσο με k και είναι μεγιστικά ως προς αυτό. Δηλαδή δεν μπορούμε να προσθέσουμε κάποια ακμή χωρίς να αυξηθεί το δενδροπλάτος. Επίσης οι μεγιστικές κλίκες ενός k-tree είναι μεγέθους $k+1$ και οι ελαχιστικοί διαχωριστές του είναι μεγέθους k. Από την στιγμή που σε ένα γράφημα G που είναι k-tree υπάρχει κλίκα μεγέθους $k+1$ από Λήμμα 2.2 γνωρίζουμε ότι $tw(G) \geq k$. Όμως, όπως φαίνεται από το Σχήμα 2 και από το Θεώρημα 2.1, υπάρχει τρόπος να φτιάξουμε μια δενδροαποσύνθεση ενός k-tree με δενδροπλάτος ίσο με k. Ο τρόπος που γίνεται αυτό είναι να φτιάξουμε μία τσάντα για κάθε κλίκα και να ενώσουσουμε δύο τσάντες αν και μόνο αν η τομή των δύο κλικών που αντιστοιχούν σε αυτές τις τσάντες έχει μέγεθος k.



Σχήμα 2: Από πάνω φαίνεται ένα γράφημα 3-tree και από κάτω μια δενδροαποσύνθεσή του με tw(G) = 3. Όταν εισάγουμε την καινούργια κορυφή f θα πρέπει να την ενώσουμε με τρεις άλλες κορυφές που μεταξύ τους δημιουργούν κλίκα. Για παράδειγμα θα ήταν λάθος αν ενώναμε την f με τις κορυφές a, b, e αφού δεν σχηματίζουν μία κλίκα μεταξύ τους.

Ορισμός 2.3. *Ένα* partial k-tree *γράφημα περιέχει όλες τις κορυφές και ένα υποσύνολο ακμών ενός* k*-tree. Δηλαδή είναι ένα υπογράφημα ενός* k*-tree.*

Για ένα γράφημα G, ο αριθμός k στα partial k-trees ώστε να ισχύει $tw(G) = k$ είναι ο μικρότερος δυνατός. Για παράδειγμα ένα δάσος G μπορεί να είναι υπογράφημα ενός k-tree για $k \geq 1$ και επειδή η μικρότερη τιμή που μπορεί να πάρει το k είναι 1, τότε το δάσος είναι partial 1-tree και $tw(G) = 1$. Αντίστοιχα ένας κύκλος δεν μπορεί να είναι υπογράφημα ενός 1-tree αλλά μπορεί να είναι υπογράφημα ενός 2-tree, επομένως ο κύκλος είναι partial 2-tree με δενδροπλάτος ίσο με 2. Σύφμωνα με τον ορισμό ο κύκλος για παράδειγμα είναι και partial 3-tree αφού και είναι υπογράφημα ενός 3-tree και το δενδροπλάτος του είναι το πολύ 3.

Το επόμενο θεώρημα δείχνει ότι η κλάση γραφημάτων που περιέχει τα γραφήματα με δενδροπλάτος το πολύ k είναι ίση με την κλάση γραφημάτων που περιέχει τα partial k-trees.

Θεώρημα 2.1. *[12], [13] Ένα γράφημα $G = (V, E)$ είναι partial k-tree, αν και μόνο αν $tw(G) \leq k$.*

Απόδειξη. $\Rightarrow$: Αρκεί να δείξουμε ότι κάθε k-tree έχει δενδροπλάτος το πολύ k καθώς το partial k-tree είναι υπογράφημα του k-tree άρα και έλασσον και από Λήμμα 2.3 γνωρίζουμε ότι και το δενδροπλάτος του partial k-tree δεν ξεπερνά το k. Αν $|V| \leq k+1$ τότε τα βάζουμε όλα σε μία τσάντα και τελειώσαμε. Υποθέτουμε ότι $|V| > k + 1$. Από τον ορισμό του k-tree υπάρχει κορυφή u που ενώνεται με k άλλες κορυφές που μεταξύ τους δημιουργούν μια κλίκα. Έστω W το σύνολο των γειτόνων του u που δημιουργούν μια K_k(κλίκα με k κορυφές). Από τον αναδρομικό ορισμό του k-tree το γράφημα $G[V \backslash u]$, δηλαδή το γράφημα G αν αφαιρέσουμε την u θα είναι k-tree. Επαγωγικά υπάρχει μια δενδροαποσύνθεση $(T = (I, F), \{X_i \mid i \in I\})$ του $G[V \setminus u]$ με δενδροπλάτος το πολύ k χρησιμοποιώντας και την βάση της επαγωγής που είναι για $|V| \leq k+1$. Από Λήμμα 2.2, επειδή το W είναι κλίκα, $\exists i \in I$ τέτοιο ώστε X_i περιέχει όλο το W. Μπορούμε τώρα να φτιάξουμε μια καινούργια δενδροαποσύνθεση για το G η οποία είναι η $(T = (I \cup \{j\}, F \cup \{i, j\}), \{X_i \mid i \in I \cup \{j\}\})$ όπου $X_j = \{u\} \cup W$. Σε αυτήν την δενδροαποσύνθεση ικανοποιείται και η συνθήκη 2, αφού στην καινούργια τσάντα X_j υπάρχουν όλες οι ακμές της u και η συνθήκη 3 αφού ενώνουμε την καινούργια τσάντα X_j με την τσάντα X_i που περιέχει ήδη τους γείτονες του u και φυσικά και η συνθήκη 1 καθώς περιέχονται όλες οι κορυφές. Η καινούργια τσάντα έχει μέγεθος $k + 1$ αφού $|W| = k$, επομένως ισχύει ότι $tw(G) \leq k$.

$\Leftarrow$: Έστω μία δενδροαποσύνθεση $(T = (I, F), \{X_i \mid i \in I\})$ του $G = (V, E)$ με δενδροπλάτος το πολύ k. Θα δείξουμε ότι το γράφημα G είναι partial k-tree, δηλαδή είναι υπογράφημα ενός k-tree. Θα το δείξουμε επαγωγικά στο $|V|$ και ότι το $G[X_i]$ ενάγει ένα υπογράφημα που είναι υπογράφημα μίας $k+1$ κλίκας. Η βάση της επαγωγής είναι όταν $|V| \leq k+1$, γιατί μπορούμε να έχουμε μια δενδροαποσύνθεση με μία τσάντα που περιέχει όλες τις κορυφές και να δημιουργήσουμε ένα k-tree που περιέχει όλες τις ακμές μεταξύ των κορυφών. Τότε όμως το G θα είναι υπογράφημα του k-tree και θα είναι partial k-tree. Αν $|V| \geq k + 1$, τότε παίρνουμε μία τσάντα X_i που είναι φύλλο στο T και έστω η τσάντα X_j ο μοναδικός γείτονας της X_i. Αν $X_i \subseteq X_j$, τότε μπορούμε να αφαιρέσουμε την X_i και να πάρουμε ένα άλλο φύλλο στην καινούργια δενδροαποσύνθεση. Έστω u η μοναδική κορυφή που ανήκει στο $X_i \setminus X_j$(Αν ανήκουν περισσότερες κορυφές τότε μπορούμε να σπάσουμε το X_i σε περισσότερες τσάντες μέχρι να ικανοποιηθεί η συνθήκη μας). Από την στιγμή που

η u δεν υπάρχει σε καμία άλλη τσάντα, όλοι οι γειτονές της πρέπει να εμφανίζονται στην τσάντα X_i. Επειδή το δενδροπλάτος είναι το πολύ k, τότε $|X_i| \leq k+1$ και η u έχει το πολύ k γείτονες στο G. Επιπλέον, επειδή το $X_i \setminus X_j$ έχει μόνο μία κορυφή θα πρέπει όλοι οι γείτονες της u να ανήκουν και στο X_j. Από την επαγωγική υπόθεση το γράφημα $G[V \setminus \{u\}]$ με την δενδροαποσύνθεση του(το T χωρίς την τσάντα X_i) είναι partial k-tree, δηλαδή είναι υπογράφημα ενός k-tree. Από την στιγμή όμως που η κορυφή u ενώνεται με k το πολύ κορυφές και οι γειτονές της ενάγουν ένα υπογράφημα μίας k κλίκας(ή ολόκληρη k κλίκα), τότε το u ενώνεται με k γείτονες που μπορούν να φτιάξουν μία k κλίκα, επομένως το G είναι partial k-tree. □

2.3 Chordal γραφήματα

Μια άλλη πολύ σημαντική κατηγορία γραφημάτων είναι τα chordal γραφήματα για τον λόγο ότι πολλά NP-complete προβλήματα σε γενικά γραφήματα λύνονται σε πολυωνυμικό χρόνο στα chordal. Κάποια από αυτά τα προβλήματα είναι η εύρεση όλων των μεγιστικών κλικών ενός γραφήματος και η εύρεση του δενδροπλάτους. Επιπλέον μέσω των chordal γραφημάτων και των ιδιοτήτων τους αποδεικνύονται διάφορες ενδιαφέρουσες σχέσεις μεταξύ του δενδροπλάτους και άλλων παραμέτρων όπως για παράδειγμα το separation number που οδηγούν στην υλοποίηση αλγορίθμων για το δενδροπλάτος και άλλων προβλημάτων. Παρακάτω παραθέτονται οι ορισμοί για τα chordal γραφήματα αλλά και για κάποιες άλλες κατηγορίες γραφημάτων που είναι ισοδύναμες με τα chordal γραφήματα. Επιπλέον αναφέρονται διάφορες ιδιότητες και θεωρήματα σχετικά με chordal γραφημάτα.

Ορισμός 2.4. *Ένα γράφημα G λέγεται* chordal *αν κάθε κύκλος του G με μήκος μεγαλύτερο του 3 περιέχει μια χορδή.*

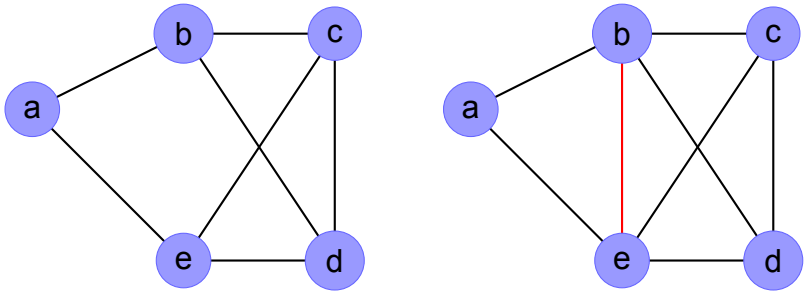
Ορισμός 2.5. Χορδή *είναι μια ακμή που ενώνει δύο μη συνεχόμενες κορυφές ενός κύκλου.*

Ορισμός 2.6. *Ισοδύναμος ορισμός του chordal γραφήματος: Ένα γράφημα G λέγεται* chordal *αν δεν περιέχει εναγόμενο κύκλο μεγαλύτερο του 3.*

Όλα τα εναγόμενα υπογραφήματα ενός chordal γραφήματος είναι επίσης chordal γραφήματα. Αυτό προκύπτει άμεσα από τον ορισμό, καθώς αν ένα υπογράφημα δεν είναι chordal τότε υπάρχει κύκλος στο υπογράφημα μεγαλύτερος του 3. Τότε όμως θα υπάρχει κύκλος μεγαλύτερος του 3 και στο αρχικό γράφημα, επομένως το αρχικό γράφημα δεν μπορεί να είναι chordal.

Θεώρημα 2.2. *[12], [15] Ένα γράφημα G είναι chordal αν και μόνο αν κάθε ελαχιστικός διαχωριστής του G είναι κλίκα.*

Απόδειξη. $\Rightarrow$: Έστω το γράφημα $G = (V, E)$ που είναι chordal και S ένας ελαχιστικός (minimal) διαχωριστής του G. Έστω δύο κορυφές a, b, τέτοιες ώστε ο S να είναι ab-διαχωριστής. Ακόμα, έστω x, y κορυφές που ανήκουν στο S. Αν αποδείξουμε ότι για κάθε ζευγάρι κορυφών στο S υπάρχει ακμή που τις ενώνει, τότε δείχνουμε ότι το S είναι



Σχήμα 3: Στα αριστερά είναι ένα γράφημα που δεν είναι chordal γιατί περιέχει εναγόμενο κύκλο (a, b, c, e) μεγαλύτερο του 3 και στα δεξιά φαίνεται ένα γράφημα που είναι chordal προσθέτοντας στο αριστερό γράφημα την κόκκινη χορδή.

κλίκα. Έστω A και B οι συνεκτικές συνιστώστες του $G[V - S]$ που περιέχουν το a και το b αντίστοιχα. Επειδή ο S είναι ελαχιστικός θα πρέπει κάθε κορυφή του να ενώνεται και με το A και με το B επομένως θα πρέπει να υπάρχει ένα μονοπάτι από το x στο y μέσω των κορυφών του A και ένα μονοπάτι από το x στο y μέσω τον κορυφών του B. Το ελάχιστο δυνατό μήκος αυτών των μονοπατιών είναι 2, επομένως σχηματίζεται σε κάθε περίπτωση ένας κύκλος μεγαλύτερος του 3 αν ενώσουμε αυτά τα μονοπάτια. Επειδή όμως το S διαχωρίζει το a και το b και το G είναι chordal θα πρέπει να υπάρχει μια χορδή μεταξύ του x και του y. Επομένως για κάθε ζευγάρι κορυφών στο S υπάρχει μια ακμή που τις ενώνει και έτσι το S είναι κλίκα.

$\Leftarrow$: Έστω G ένα γράφημα για το οποίο κάθε ελαχιστικός (minimal) διαχωριστής είναι κλίκα. Ας υποθέσουμε ότι το G δεν είναι chordal. Επομένως θα πρέπει να υπάρχει κύκλος χωρίς χορδή $a, x, b, y_1, ..., y_i, a$ με μήκος μεγαλύτερο του 3 $(i \geq 1)$. Κάθε ab-διαχωριστής θα πρέπει να περιέχει το x και τουλάχιστον ένα y_i. Επειδή όμως όλοι οι ελαχιστικοί διαχωριστές είναι κλίκες, θα πρέπει να υπάρχει η ακμή (x, y_i). Επομένως ο κύκλος έχει χορδές. Φτάσαμε σε άτοπο, άρα το G είναι chordal. □

Ορισμός 2.7. *Μια κορυφή λέγεται* simplicial *αν το σύνολο των γειτόνων της ενάγει μία κλίκα.*

Λήμμα 2.11. *[12] Ένα chordal γράφημα είτε είναι κλίκα, είτε περιέχει τουλάχιστον δύο μη γειτονικές simplicial κορυφές.*

Απόδειξη. Αν το γράφημα είναι κλίκα τελειώσαμε. Έστω ότι το $G = (V, E)$ δεν είναι κλίκα. Θα αποδείξουμε με επαγωγή στον αριθμό των κορυφών ότι υπάρχουν τουλάχιστον δύο μη γειτονικές simplicial κορυφές.

- Βάση επαγωγής: Το G περιέχει 2 απομονωμένες κορυφές(γιατί δεν θέλουμε να είναι κλίκα). Θεωρούμε το κενό γράφημα ως κλίκα, επομένως το σύνολο των γειτόνων κάθε κορυφής ενάγει μια κλίκα και έτσι έχουμε δύο μη γειτονικές simplicial κορυφές.

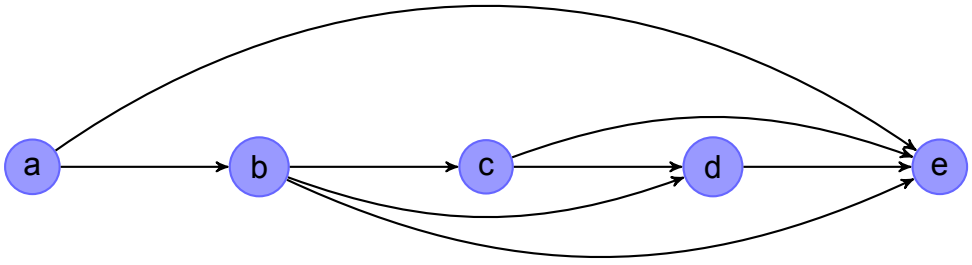
- Επαγωγική υπόθεση: Έστω ότι για κάθε γράφημα με $n \geq 2$ κορυφές υπάρχουν τουλάχιστον δύο μη γειτονικές simplicial κορυφές.

- Επαγωγικό βήμα: Θα πρέπει να δείξουμε ότι για κάθε γράφημα με $n > 2$ κορυφές υπάρχουν τουλάχιστον δύο μη γειτονικές simplicial κορυφές. Έστω δύο μη γειτονικές κορυφές u και w και S ένας ελαχιστικός uw-διαχωριστής. Το γράφημα $G[V - S]$ σπάει σε τουλάχιστον δύο συνεκτικές συνιστώσες. Έστω $G[U]$ η συνιστώσα που περιέχει την κορυφή u και $G[W]$ η συνιστώσα που περιέχει την κορυφή w. Το εναγόμενο γράφημα $G[U \cup S]$ είναι υπογράφημα του G και από τον ορισμό του chordal, θα πρέπει και αυτό να είναι chordal γράφημα. Επιπλέον, ο αριθμός των κορυφών του $G[U \cup S]$ είναι λιγότερος από n αφού υπάρχει τουλάχιστον μία κορυφή (η w) που δεν συμπεριλαμβάνεται σε αυτό το υπογράφημα. Αυτό σημαίνει ότι είτε το $G[U \cup S]$ είναι κλίκα και όλες οι κορυφές του U είναι simplicial, είτε από επαγωγική υπόθεση το εναγόμενο υπογράφημα έχει δύο τουλάχιστον μη γειτονικές simplicial κορυφές. Από το Θεώρημα 2.2 γνωρίζουμε ότι το S είναι κλίκα, άρα όλες οι κορυφές του S είναι γειτονικές και επομένως μία από τις μη γειτονικές simplicial κορυφές θα πρέπει να ανήκει στο U. Σε κάθε περίπτωση το U περιέχει μία τουλάχιστον simplicial κορυφή και επειδή το U δεν έχει γείτονες έξω από το $G[U \cup S]$, τότε αυτή η simplicial κορυφή που υπάρχει στο U θα είναι και simplicial στο G. Με αυτήν την λογική βρήκαμε μία simplicial κορυφή του G που ανήκει στο U. Με την ίδια λογική μπορούμε να βρούμε και μία simplicial κορυφή του G που ανήκει στο W. Επειδή όμως το $G[U]$ δεν ενώνεται με ακμές με το $G[W]$, τότε αυτές οι δύο simplicial κορυφές θα είναι και μη γειτονικές.

□

Οι simplicial κορυφές σχετίζονται άμεσα με την επόμενη έννοια που είναι το perfect elimination ordering το οποίο είναι ένας τρόπος για να αναγνωρίσουμε σε πολυωνυμικό χρόνο αν ένα γράφημα είναι chordal. Επιπλέον μέσω του perfect elimination ordering μπορούν να υλοποιηθούν αλγόριθμοι που λύνουν σε πολυωνυμικό χρόνο NP-complete προβλήματα σε γενικά γραφήματα όπως είναι η εύρεση όλων των μεγιστικών κλικών.

Ορισμός 2.8. *Το* perfect elimination ordering *είναι μία διάταξη κορυφών, έτσι ώστε για κάθε κορυφή u, το σύνολο $\{u \cup N(u)\}$ ενάγει μια κλίκα, όπου $N(u)$ είναι οι γείτονες του u που βρίσκονται στην διάταξη μετά το u.*



Σχήμα 4: Η παραπάνω διάταξη είναι ένα perfect elimination ordering του chordal γραφήματος (στα δεξιά) από το Σχήμα 3.

Το perfect elimination ordering είναι ένας τρόπος για να ανγνωρίσουμε αν ένα γράφημα είναι chordal όπως φαίνεται και από το επόμενο θεώρημα. Το perfect elimination ordering σχετίζεται άμεσα με τις simplicial κορυφές αφού για κάθε simplicial κορυφή οι γειτονές της ενάγουν μια κλίκα, επομένως αν διαλέξουμε αυτήν την κορυφή στην διάταξη πριν τους γειτονές της τότε ικανοποιούμε την συνθήκη του perfect elimination ordering. Για να βρούμε μία simplicial κορυφή και να την τοποθετήσουμε στην διάταξη χρειαζόμαστε πολυωνυμικό χρόνο. Επομένως μπορούμε να κατασκευάσουμε ένα perfect elimination ordering και να ανγνωρίσουμε αν ένα γράφημα είναι chordal σε πολυωνυμικό χρόνο.

Θεώρημα 2.3. *[12], [15] Ένα γράφημα G είναι chordal αν και μόνο αν έχει perfect elimination ordering.*

Απόδειξη. $\Rightarrow$: Από Λήμμα 2.11 ξέρουμε ότι αν το G είναι chordal τότε έχει κάποια simplicial κορυφή. Δηλαδή μία κορυφή u για την οποία οι γειτονές της $N(u)$ δημιουργούν μια κλίκα. Επιπλέον το γράφημα $G[V - \{u\}]$ είναι και αυτό chordal γιατί η αφαίρεση κορυφής δεν δημιουργεί κάποιον καινούργιο κύκλο. Επομένως επαγωγικά υποθέτουμε ότι το $G[V-\{u\}]$ έχει ένα perfect elimination ordering π. Έστω ένα perfect elimination ordering για το G που βάζει πρώτα το u και στην συνέχεια τις κορυφές με βάση το π. Αυτό είναι ένα έγκυρο perfect elimination ordering για το G.

$\Leftarrow$: Έστω $\pi = u_1, u_2, ..., u_n$ ένα perfect elimination ordering για το G. Υποθέτουμε ότι το G δεν είναι chordal οπότε υπάρχει κύκλος χωρίς χορδή $u_i, u_{i+1}, ..., u_{i+j}, u_i$ με μήκος μεγαλύτερο του 3 $(j \geq 3)$ (Τα u_{i+j} δεν χρειάζεται να είναι συνεχόμενα στην διάταξη, παρά μόνο να διατηρείται η σειρά). Από την στιγμή όμως που το π είναι perfect elimination ordering για το u_i ξέρουμε ότι οι γείτονές του που έρχονται μετά από αυτό στην σειρά θα φτιάχνουν μία κλίκα. Επομένως ο κύκλος έχει χορδές και δεν έχει μέγεθος μεγαλύτερο του 3. Φτάσαμε σε άτοπο, οπότε το G είναι chordal. □

Παρακάτω εξηγούμε δύο άλλες κατηγορίες γραφημάτων που είναι τα intersection graphs και τα clique trees οι οποίες σχετίζονται άμεσα με τα chordal γραφήματα. Ο λόγος είναι ότι μέσω αυτών μπορούμε να αποδείξουμε διάφορες σχέσεις μεταξύ του δενδροπλάτους και άλλων παραμέτρων καθώς και να αντλήσουμε πληροφορίες για τα chordal γραφήματα, όπως για παράδειγμα το πόσο είναι το δενδροπλάτος τους, ποια είναι η δενδροαποσύνθεσή τους, ποιες είναι οι μεγιστικές κλίκες και ποιοι οι ελαχιστικοί διαχωριστές του γραφήματος.

2.3.1 Intersection γραφήματα και Clique trees

Στην συνέχεια αναφέρουμε κάποιες κατηγορίες γραφημάτων που συνδέονται άμεσα με τα chordal γραφήματα. Αυτές είναι τα intersection γραφήματα και τα clique trees. Θα δείξουμε κάποιες ισότητες μεταξύ αυτών των κατηγοριών και μέσω αυτών θα φανεί η στενή σχέση μεταξύ των chordal γραφημάτων και του δενδροπλάτους.

Ορισμός 2.9. *Έστω A μια οικογένεια μη κενών συνόλων. Το* intersection γράφημα *του A είναι το $G = (V, E)$, όπου στο V αντιστοιχούμε για κάθε σύνολο (στοιχείο του A) μία*

κορυφή και στο E υπάρχει η ακμή $\{u, w\}$ αν και μόνο αν τα σύνολα που αντιστοιχούν στη u και w τέμνονται.

Μια οικογένεια A υποδένδρων ενός δέντρου T περιέχει ως στοιχεία κάποια υπογραφήματα του δέντρου T που με τη σειρά τους είναι και αυτά δένδρα. Κάποια υπογραφήματα μπορεί να τέμνονται, δηλαδή να μοιράζονται τουλάχιστον μία κορυφή. Αυτό σημαίνει ότι και οι κορυφές που θα αντιστοιχούν σε αυτά τα υπογραφήματα θα τέμνονται στο $G = (V, E)$ που είναι το intersection γραφημά τους. Προσοχή ότι τα στοιχεία του A είναι και αυτά δένδρα και όχι δάση. Αυτό σημαίνει ότι κάθε στοιχείο του A πρέπει να είναι μία συνεκτική συνιστώσα.

Θεώρημα 2.4. *[10] Ένα γράφημα είναι chordal αν και μόνο αν είναι το intersection γράφημα μιας οικογένειας μη κενών υποδένδρων ενός δέντρου.*

Το θεώρημα αυτό μας λέει ότι αν ένα γράφημα G είναι chordal, τότε υπάρχει ένα δέντρο T και κάποιο σύνολο A που περιέχει υποδένδρα του , τέτοια ώστε το G να είναι το intersection γράφημα της οικογένειας A. Ακόμα, για κάθε οικογένεια A που περιέχει ως στοιχεία υποδένδρα ενός δέντρου T, το intersection γράφημα που προκύπτει είναι chordal, δηλαδή δεν περιέχει εναγόμενο κύκλο μεγαλύτερου του 3.

Ορισμός 2.10. *Ένα* clique tree *ενός γραφήματος G είναι ένα δέντρο $T = (V, E)$, όπου στο V αντιστοιχούμε για κάθε μεγιστική κλίκα του G μία κορυφή και στο E υπάρχει η ακμή $\{u, w\}$ αν και μόνο αν οι κλίκες που αντιστοιχούν στη u και w τέμνονται στο G. Επιπλέον $\forall u \in G$ θα πρέπει το υπογράφημα του T που περιέχει όλες τις κορυφές που αντιστοιχούν στις κλίκες που περιέχουν το u να είναι συνεκτικό.*

Αν ορίσουμε ως K το σύνολο που περιέχει όλες τις μεγιστικές κλίκες ενός γραφήματος και K_u όλες τις μεγιστικές κλίκες που περιέχουν το u, τότε παίρνουμε τον παρακάτω ισοδύναμο ορισμό.

Ορισμός 2.11. *Ισοδύναμος ορισμός του clique tree: Ένα* clique tree *ενός γραφήματος G είναι ένα δέντρο $T = (V, E)$, όπου το σύνολο κορυφών του είναι το K (όπως ορίστηκε από πάνω) και $\forall u \in V$ το $T[K_u]$ (δηλαδή το υπογράφημα που περιέχει όλες τις κορυφές του T που αντιστιχούν στις μεγιστικές κλίκες που περιέχουν το u) είναι συνεκτικό.*

Θεώρημα 2.5. *[10] Ένα γράφημα G είναι το intersection γράφημα μιας οικογένειας υποδένδρων ενός δέντρου αν και μόνο αν υπάρχει T τέτοιο ώστε να είναι το clique tree του G.*

Θεώρημα 2.6. *Ένα γράφημα είναι chordal αν και μόνο αν υπάρχει T ώστε να είναι το clique tree του G.*

Απόδειξη. Το Θεώρημα 2.4 δείχνει ότι το σύνολο των chordal γραφημάτων είναι το ίδιο με αυτό των intersection γραφημάτων των υποδένδρων ενός δέντρου. Επίσης το Θεώρημα 2.5 δείχνει ότι το σύνολο των intersection γραφημάτων των υποδένδρων ενός δέντρου είναι ίδιο με αυτό των clique trees. Επομένως και το σύνολο των chordal γραφημάτων είναι το ίδιο με αυτό των clique trees. □

2.3.2 Σχέσεις μεταξύ των chordal γραφημάτων και του δενδροπλάτους

Τα chordal γραφήματα έχουν στενή σχέση με το δενδροπλάτος λόγω των διάφορων ιδιοτήτων τους όπως θα φανεί παρακάτω αλλά και γιατί μπορούμε να βρούμε το δενδροπλάτος τους και μια δενδροαποσυνθεσή τους σε πολυωνυμικό χρόνο.

Λήμμα 2.12. *Τα k-trees γραφήματα είναι chordal. Δηλαδή δεν περιέχουν εναγόμενο κύκλο μεγαλύτερο του 3.*

Απόδειξη. Από τον Ορισμό 2.2 των k-trees βλέπουμε ότι κάθε φορά που εισάγουμε μια καινούργια κορυφή u, όλοι οι γειτονές της δημιουργούν μια κλίκα. Αυτό σημαίνει ότι η κορυφή u είναι simplicial. Επομένως στα k-trees μπορούμε πάντα να κάνουμε την αντίστοφη ενέργεια της δημιουργίας τους, δηλαδή να βγάζουμε κάθε φορά μια simplicial κορυφή και να καταλήξουμε σε κενό γράφημα. Αυτή όμως η ενέργεια θα μας δώσει ένα perfect elimination ordering και από το Θεώρημα 2.3 ξέρουμε ότι αυτό το γράφημα θα είναι chordal. □

Λήμμα 2.13. *Αν το G είναι chordal και το T είναι ένα clique tree του, τότε το T είναι μια δενδροαποσύνθεση του G.*

Απόδειξη. Στο T κάθε κορυφή αντιστοιχεί σε μία μεγιστική κλίκα του G, οπότε μπορούμε να φανταστούμε κάθε κορυφή του T ως μια τσάντα που περιέχει την κλίκα. Από τον ορισμό της δενδροαποσύνθεσης θα πρέπει το clique tree T να ικανοποιεί τρεις συνθήκες. Η πρώτη ικανοποιείται αφού κάθε κορυφή ανήκει σε τουλάχιστον μία κλίκα, επομένως θα υπάρχει και σε τουλάχιστον μία τσάντα του T. Η δεύτερη ικανοποιείται επειδή αν υπάρχει η ακμή $\{u, w\}$ στο G, τότε υπάρχει κλίκα K στην οποία ανήκει και το u και το w, επομένως υπάρχει και η αντίστοιχη τσάντα στο T. Τέλος, ικανοποιείται και η τρίτη συνθήκη αφού το υπογράφημα που περιέχει όλες τις μεγιστικές κλίκες που περιέχουν την κορυφή u θα πρέπει να είναι συνεκτικό με βάση τον ορισμό των clique trees. □

Το αντίθετο βέβαια δεν ισχύει καθώς στην δενδροαποσύνθεση δεν είναι απαραίτητο κάθε τσάντα να περιέχει μία μεγιστική κλίκα. Με βάση το βάση το Λήμμα 2.13 προκύπτει και το επόμενο σημαντικό λήμμα.

Λήμμα 2.14. *Το δενδροπλάτος (treewidth) ενός chordal γραφήματος G ισούται με την μέγιστη κλίκα του μείον 1. Δηλαδή ισχύει ότι αν το G είναι chordal τότε $tw(G) = \omega(G) - 1$.*

Απόδειξη. $\geq$: Από το Λήμμα 2.2 ξέρουμε ότι αν το G έχει μία κλίκα τότε αυτή θα πρέπει να ανήκει σε μία τσάντα. Επομένως παίρνοντας την μέγιστη κλίκα ισχύει ότι $tw(G) \geq \omega(G)-1$.

$\leq$: Από το Θεώρημα 2.6 ξέρουμε ότι αν το G είναι chordal τότε υπάρχει T που είναι το clique tree του G και από Λήμμα 2.13 το T θα είναι μια δενδροαποσύνθεση του G. Η κάθε τσάντα του T όμως αποτελείται μόνο από μεγιστικές κλίκες που δεν περιέχουν κορυφές εκτός της κλίκας. Το δενδροπλάτος αυτής της δενδροαποσύνθεσης ισούται με την μέγιστη τσάντα που δεν πρόκειται ποτέ να ξεπερνάει το μέγεθος της μέγιστης κλίκας. Επομένως ισχύει ότι $tw(G) \leq \omega(G) - 1$.

Τελικά ισχύει ότι αν το G είναι chordal τότε $tw(G) = \omega(G) - 1$. □

Λήμμα 2.15. *Αν $G = (V, E)$ είναι ένα γράφημα και $(T = (I, F), \{X_i \mid i \in I\})$ η δενδροαποσυνθεσή του, τότε αν ενώσουμε με ακμή όλες τις κορυφές που ανήκουν στην ίδια τσάντα X_i το καινούργιο γράφημα που παίρνουμε είναι chordal.*

Απόδειξη. Το T είναι μια δενδροαποσύνθεση και επομένως είναι δέντρο. Αν ενώσουμε με ακμή στο G όλες τις κορυφές που ανήκουν στην ίδια τσάντα X_i τότε οι κουρυφές που βρίσκονται μέσα στο X_i θα δημιουργήσουν μία κλίκα μεταξύ τους. Επομένως το καινούργιο γράφημα G' που προκύπτει είναι το intersection γράφημα των υποδένδρων $T_u = T[\{i \in I \mid u \in X_i\}]$ και από Θεώρημα 2.4 το καινούργιο γράφημα είναι chordal. Ο λόγος που το G' είναι το intersection γράφημα των υποδένδρων T_i είναι επειδή αν δύο T_u και T_v τέμνονται τότε αυτό σημαίνει ότι τα u και v ανήκουν σε μια κοινή τσάντα άρα θα ενώνονται στο G'. Αντίστοιχα αν δύο κορυφές u, v ενώνονται στο G ή στο G', τότε θα πρέπει να υπάρχει τσάντα στην δενδροαποσύνθεση που να περιέχει και τις δύο κορυφές. Τότε όμως τα T_u και T_v θα τέμνονται. □

Στο προηγούμενο λήμμα δεν χρησιμοποιήθηκε η έννοια των clique trees γιατί οι κλίκες που προκύπτουν δεν είναι απαραίτητα μεγιστικές. Βέβαια από την στιγμή που το γράφημα που προκύπτει είναι chordal, γνωρίζουμε ότι υπάρχει δέντρο T που είναι και το clique tree και μια δενδροαποσύνθεση του καινούργιου γραφήματος.

Όταν έχουμε ένα γράφημα G και με την πρόσθεση μηδέν ή παραπάνω ακμών παίρνουμε το γράφημα H που είναι chordal, τότε λέμε ότι το H είναι το chordal completion του G. Αυτό θα μας χρειαστεί στο επόμενο θεώρημα που συνοψίζει την στενή σχέση μεταξύ των chordal γραφημάτων και του δενδροπλάτους.

Θεώρημα 2.7. *Έστω το σύνολο A που περιέχει όλα τα chordal completion του G. Το G' είναι το γράφημα που ανήκει στο A και για όλα τα άλλα γραφήματα που ανήκουν στο A ισχύει ότι η μέγιστη κλίκα τους είναι μεγαλύτερη ή ίση της μέγιστης κλίκας του G'. Δηλαδή το G' είναι το chordal completion του G με την μικρότερη δυνατή μέγιστη κλίκα. Ένα γράφημα $G = (V, E)$ έχει $tw(G) = k$ αν και μόνο αν υπάρχει γράφημα G' (με όσα υποθέσαμε πριν) ώστε η μέγιστη κλίκα του να είναι μεγέθους $k + 1$.*

Απόδειξη. $\Rightarrow$: Αν το γράφημα G έχει δενδροπλάτος ίσο με k, τότε υπάρχει δενδροαποσύνθεση του ώστε κάθε τσάντα να περιέχει το πολύ $k + 1$ κορυφές. Αν τώρα ενώσουμε όλες τις κορυφές που βρίσκονται στην ίδια τσάντα με ακμή θα πάρουμε το γράφημα G' το οποίο από Λήμμα 2.15 είναι chordal. Επειδή όμως η μεγαλύτερη τσάντα περιέχει $k + 1$ κορυφές (αν περιέχει λιγότερες, τότε προσθέτουμε αυθαίρετα κάποιες κορυφές σε όλες τις τσάντες μέχρι η μεγαλύτερη τσάντα να περιέχει $k+1$ κορυφές), τότε η μέγιστη κλίκα του G' θα είναι μεγέθους $k+1$. Δεν γίνεται να υπάρχει μεγαλύτερη κλίκα μετά την πρόσθεση των ακμών επειδή για δύο διαφορετικές τσάντες θα υπάρχουν κορυφές που δεν ανήκουν στην ίδια τσάντα οπότε ακόμα και αν έχουν τους ίδιους γείτονες δεν θα ενωθούν ποτέ μεταξύ τους και δεν θα έχουμε μεγαλύτερη κλίκα.

$\Leftarrow$: Έστω G' γράφημα με μέγιστη κλίκα μεγέθους $k + 1$ ώστε να είναι chordal completion του G. Από την στιγμή που είναι chordal το Λήμμα 2.14 μας λέει ότι το δενδροπλάτος του ισούται με k. Το γράφημα G είναι υπογράφημα του G' επομένως το δενδροπλάτος του

δεν μπορεί να ξεπεράνει το k. Αν ήταν λιγότερο από k τότε από την περίπτωση $\Rightarrow$ θα υπήρχε chordal completion με μικρότερη μέγιστη κλίκα κάτι που δεν γίνεται σύμφωνα με την υπόθεση που κάναμε για το G'. □

Με βάση το παραπάνω θεώρημα ένα γράφημα έχει δενδροπλάτος το πολύ k, δηλαδή είναι partial-k tree αν και μόνο αν υπάρχει τρόπος να του προσθέσουμε ακμές ώστε να το μετατρέψουμε σε chordal και η μέγιστη κλίκα του chordal να είναι το πολύ $k+1$. Επιπλέον μέσω του perfect elimination ordering μπορεί να κατασκευαστεί για ένα chordal ένα clique tree του άρα και μια δενδροαποσύνθεση του. Επειδή η κατασκευή του clique tree ενός chordal γραφήματος γίνεται σε πολυωνυμικό χρόνο, αν υπάρχει πολυωνυμικός αλγόριθμος που να λύνει το πρόβλημα της μετατροπής ενός γραφήματος σε chordal με την μικρότερη δυνατή μέγιστη κλίκα, τότε θα λύναμε και σε πολυωνυμικό χρόνο το πρόβλημα του δενδροπλάτους.

2.4 Elimination Game

Το elimination game είναι ένα παιχνίδι που παίζεται πάνω σε ένα γράφημα G στο οποίο επιλέγουμε όλες τις κορυφές ακριβώς μία φορά και προσθέτουμε ακμές στο γράφημα με βάση κάποια κριτήρια. Σε ένα γράφημα G όταν παίξουμε το elimination game το καινούργιο γράφημα που προκύπτει είναι chordal. Μέσω του καινούργιου chordal γραφήματος και του elimination tree που θα οριστεί παρακάτω μπορούν να δειχθούν ενδιαφέρουσες σχέσεις μεταξύ των παραμέτρων που σχετίζονται με το δενδροπλάτος.

Ορισμός 2.12. *Το* elimination game *παίζεται πάνω σε ένα γράφημα G και ορίζεται με τον παρακάτω αλγόριθμο.*

- *Επίλεξε μία κορυφή u*
- *Πρόσθεσε ακμές έτσι ώστε να ενώσεις όλους τους γείτονες του u που δεν τους έχεις επιλέξει (αφαιρέσει) ακόμα.*
- *Θέσε τον u ως επιλεγμένη κορυφή ή απλώς αφαιρεσέ την. Επέστρεψε στο 1ο βήμα.*

Το παιχνίδι τελειώνει όταν έχουν επιλεχθεί όλες οι κορυφές (ή έχουν αφαιρεθεί όλες). Οι καινούργιες ακμές που προστίθενται λέγονται fill edges και το τελικό γράφημα που προκύπτει G^+_π λέγεται filled graph. Το π είναι η σειρά με την οποία επιλέχθηκαν οι κορυφές. Για διαφορετική σειρά μπορούμε να πάρουμε διαφορετικό τελικό γράφημα. Επίσης αν παίξουμε το ίδιο παιχνίδι στο G^+_π με την ίδια σειρά π τότε δεν θα προστεθούν καινούργιες ακμές. Ο λόγος είναι ότι για το u που επιλέγουμε κάθε φορά στο παιχνίδι οι γειτονές του είτε θα ενώνονται εξαρχής είτε θα έχουν ενωθεί από την πρώτη φορά που παίξαμε το παιχνίδι. Αν παίξουμε το παιχνίδι παρόλα αυτά με διαφορετική σειρά τότε είναι πιθανόν να προστεθούν πάλι καινούργιες ακμές. Για μια σειρά π, το $\pi(u)$ μας δίνει την στιγμή που επιλέξαμε μια κορυφή u. Η πρώτη κορυφή που επιλέγουμε έχει την τιμή 1 και οι επόμενες έχουν την τιμή της προηγούμενης τους συν 1. Για παράδειγμα σε ένα γράφημα

$G = (V, E)$ με 3 κορυφές, αν επιλέξαμε πρώτα την κορυφή a, μετά την b και μετά την c, τότε $\pi(a) = 1, \pi(b) = 2, \pi(c) = 3$. Όταν λέμε ότι παίζουμε το παιχνίδι ξανά με την ίδια σειρά π εννοούμε ότι διαλέγουμε τις κορυφές με βάση την συνάρτηση $\pi^{-1}(u)$. Στο παραδειγμά μας δηλαδή θα είχαμε την σειρά $\{\pi^{-1}(i) \mid i \in [1, |V|]\} = \{\pi^{-1}(1), \pi^{-1}(2), \pi^{-1}(3)\} = \{a, b, c\}$.

Ουσιαστικά αυτό που κάνει το παιχνίδι σε κάθε βήμα είναι να δημιουργεί κλίκα μεταξύ των γειτόνων του u στο G_π^+ που δεν έχουν επιλεχθεί ακόμα, δηλαδή να μετατρέπει την κορυφή u σε simplicial σε σχέση με το γράφημα που δεν έχει αφαιρεθεί ακόμα. Επομένως παρατηρούμε ότι η σειρά π των κορυφών στο G_π^+ είναι ένα perfect elimination ordering γιατί κάθε φορά που συναντάμε ένα u στην διάταξη τότε οι γείτονες του u που βρίσκονται μετά το u στην διάταξη θα δημιουργούν μια κλίκα. Άρα και το $\{u \cup N(u)\}$ ενάγει μια κλίκα, όπου $N(u)$ είναι οι γείτονες του u που βρίσκονται στην διάταξη μετά το u. Αυτό σημαίνει ότι από Θεώρημα 2.3 το τελικό γράφημα G_π^+ είναι chordal. Το G_π^+ λέγεται και αλλιώς perfect elimination graph. Ένα chordal γράφημα που προκύπτει από την πρόσθεση ακμών στο G λέγεται chordal completion του G. Άρα και το G_π^+ είναι ένα chordal completion του G. Είναι ενδιαφέρον ότι για κάθε ελαχιστικό chordal completion G' του G υπάρχει μια διάταξη π ώστε το G_π^+ να είναι το G'. Αυτό σημαίνει ότι υπάρχει διάταξη π που δημιουργεί ένα G_π^+, τέτοιο ώστε οι ακμές που προσθέτονται να είναι οι λιγότερες δυνατές. Το πρόβλημα αυτό όμως έχει αποδειχθεί ότι είναι NP-complete στο [16] που σημαίνει ότι δεν γνωρίζουμε κάποιον πολυωνυμικό αλγόριθμο για αυτό το πρόβλημα.

Ορισμός 2.13. *Το $C_\pi(u)$ είναι το σύνολο των γειτόνων του u στο G_π^+ που δεν έχουν επιλεχθεί (αφαιρεθεί) μέχρι την στιγμή που επιλέξαμε το u. Είναι δηλαδή οι γείτονες του u στο G_π^+ που βρίσκονται μετά το u στην διάταξη π. Δηλαδή αν η κορυφή w είναι γείτονας της u στο G_π^+ και $\pi(u) < \pi(w)$, τότε $w \in C_\pi(u)$.*

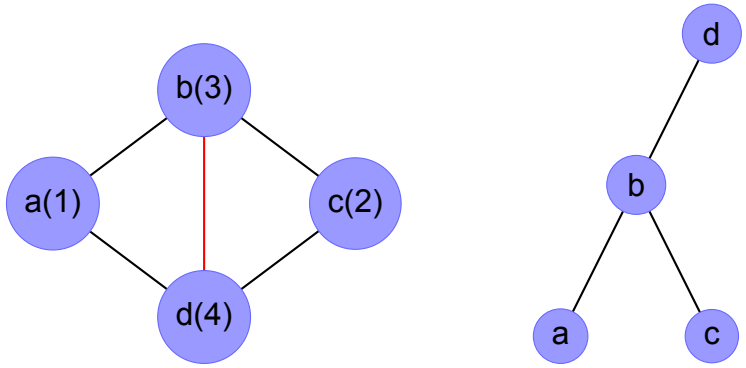
Όταν μιλάμε για τους γείτονες του u αναφερόμαστε στο γράφημα G_π^+. Δηλαδή λαμβάνουμε υπόψιν και τις fill edges που προσθέτουμε.

Ορισμός 2.14. *Το μέγεθος του μεγαλύτερου $C_\pi(u)$ ονομάζεται* front size*. Το μικρότερο front size από όλους τους δυνατούς συνδυασμούς του π ονομάζεται* min front size*.*

Για διαφορετική σειρά επιλογής των κορυφών παίρνουμε διαφορετικό π και διαφορετικό front size. Η έννοια του front size σχετίζεται άμεσα με το δενδροπλάτος και είναι χρήσιμη για τον αλγόριθμο multifrontal που υπολογίζει το Cholesky factorization ενός πίνακα. Στην συνέχεια δίνεται ένας ορισμός για το elimination tree το οποίο κατασκευάζεται αφού παίξουμε το elimination game και κατασκευαστεί το G_π^+.

Ορισμός 2.15. *Το* elimination tree *T_π έχει τις ίδιες κορυφές με το G και η σχέση γονιού πατέρα ορίζεται ως εξης: Ο γονιός της κορυφής u είναι η κορυφή w που έχει την μικρότερη τιμή $\pi(w)$ από όλες τις κορυφές που ανήκουν στο σύνολο $C_\pi(u)$. Με άλλα λόγια είναι η κορυφή w που είναι γείτονας της u στο G_π^+, ακόμα ισχύει ότι $\pi(u) < \pi(w)$ και για κάθε άλλον γείτονα k του u στο G_π^+ τέτοιον ώστε $\pi(u) < \pi(k)$, ισχύει ότι $\pi(w) < \pi(k)$.*

Το elimination tree είναι ένα spanning forest του G_π^+ που δημιουργείται μέσω αναζήτησης κατά βάθος(dfs). Αυτό σημαίνει ότι για κάθε μία συνεκτική συνιστώσα του G_π^+ (ή ισοδύναμα του G) έχουμε και μία αντίστοιχη συνεκτική συνιστώσα του T_π. Ο λόγος που ο αριθμός



Σχήμα 5: Αριστερά: Με μαύρες ακμές είναι το αρχικό γράφημα. Με (i) συμβολίζουμε την σειρά που επιλέξαμε τις κορυφές για να παίξουμε το elimination game. Στην αρχή διαλέξαμε την κορυφή a και για αυτό προστέθηκε η fill edge(η κόκκινη ακμή). Δεξιά: Είναι το elimination tree που προκύπτει αν παίξουμε το elimination game με την σειρά που φαίνεται στα αριστερά.

των συνεκτικών συνιστωσών του G_π^+ και του G είναι ίσος είναι επειδή κάθε φορά που προσθέτουμε στο G_π^+ μία καινούργια ακμή (fill edge) $\{w, k\}$ όταν επιλέγουμε το u, θα πρέπει και το w και το k να είναι γείτονες του u. Οπότε υπάρχει ήδη το μονοπάτι $w - k$ μέσω του u και δεν δημιουργείται κάποια καινούργια συνεκτική συνιστώσα με την πρόσθεση της ακμής. Μια τιμή που μας ενδιαφέρει για διάφορες εφαρμογές όπως το Cholesky factorization είναι το ύψος του T_π. Συγκεκριμένα αυτό που θέλουμε είναι το ύψος του T_π να είναι όσο το δυνατόν μικρότερο και για διαφορετικές επιλογές του π παίρνουμε διαφορετικό T_π με διαφορετικό ύψος.

Ορισμός 2.16. *Το μικρότερο ύψος από όλα τα πιθανά T_π που δημιουργούνται από κάθε πιθανό συνδυασμό του π ονομάζεται* min etree height*.*

Με τον τρόπο που δημιουργείται το elimination tree αν δύο κορυφές ενώνονται με ακμή (έστω $\{u, w\}$) στο G_π^+ τότε μία από αυτές είναι πρόγονος της άλλης, δηλαδή χωρίς βλάβη της γενικότητας αν ξεκινήσουμε από την κορυφή με το μεγαλύτερο βάθος που είναι η u και από μία κορυφή μετακινούμαστε μόνο στον πατέρα της, τότε θα καταλήξουμε στον πρόγονο της u που είναι η w. Αυτό είναι πολύ σημαντικό όπως θα φανεί αργότερα και με τους διαχωριστές καθώς αν δύο κορυφές δεν έχουν την σχέση προγόνου-απογόνου στο elimination tree αυτές οι κορυφές δεν ενώνονται ούτε στο G_π^+ και επομένως ούτε στο G. Το αντίθετο βέβαια δεν ισχύει, δηλαδή αν δύο κορυφές έχουν την σχέση προγόνου-απογόνου στο elimination tree δεν ενώνονται υποχρεωτικά στο G_π^+ και στο G.

Λήμμα 2.16. *Αν δύο κορυφές u και w ενώνονται με ακμή στο G_π^+ τότε η u και w έχουν σχέση προγόνου-απογόνου στο elimination tree. Δηλαδή αν ξεκινήσουμε από την κορυφή με το μεγαλύτερο βάθος (τον απόγονο) και μετακινούμαστε κάθε φορά μόνο στον πατέρα της τρέχουσας κορυφής τότε θα καταλήξουμε στον πρόγονο.*

Απόδειξη. Έστω δύο κορυφές u και w που ενώνονται στο G_π^+. Χωρίς βλάβη της γενικότητας διαλέγουμε στην σειρά π πρώτα την κορυφή u. Αυτό σημαίνει ότι στο $C_\pi(u)$ υπάρχει η κορυφή w και όλοι οι γείτονες του u στο G_π^+ που δεν έχουν επιλεχθεί ακόμα ενώνονται με

ακμή. Αν από τους γείτονες της u στο G_π^+ η επόμενη που διαλέξουμε είναι η w τότε από τον ορισμό του $C_\pi(u)$ και του T_π, το w θα είναι ο πατέρας του u. Αν δεν συμβεί αυτό και η επόμενη κορυφή από τους γείτονες του u που επιλεχθεί είναι μια άλλη κορυφή a τότε η a είναι ο ο πατέρας της u και επειδή η w δεν έχει επιλεχθεί ακόμα και ένώθηκε προηγουμένως με την a, τότε το w ανήκει στο $C_\pi(a)$. Επίσης γενικά ισχύει ότι από την στιγμή που επιλέξουμε μία κορυφή δεν γίνεται να υπάρξουν μελλοντικές fill edges προς αυτήν ξανά. Επομένως όταν αποφεύγουμε να διαλέξουμε την w και διαλέγουμε μια διαφορετική κορυφή k, τότε επειδή προηγουμένως οι κορυφές w και k ενωθήκαν με ακμή λόγω μιας κοινής προηγούμενης κορυφής, το w θα ανήκει στο $C_\pi(k)$ και τελικά όταν επιλεχθεί θα είναι ο πατέρας της k που είναι πρόγονος της αρχικής κορυφής u. Επομένως και η w θα είναι πρόγονος της u. □

Ορισμός 2.17. *Για κάθε συνδυασμό του π παίρνουμε ένα chordal completion του G που είναι το G_π^+. Ως* clique number *ορίζουμε το μέγεθος της μέγιστης κλίκας ενός γραφήματος. Ακόμα το min max clique ενός γραφήματος ορίζεται ως το ελάχιστο clique number του G_π^+ για κάθε συνδυασμό του π.*

Λήμμα 2.17. *Για ένα γράφημα $G = (V, E)$ ισχύει ότι το $tw(G) =$ min max clique -1.*

Απόδειξη. Από το Θεώρημα 2.7 και από το γεγονός ότι παίρνουμε όλα τα πιθανά chordal completion του G παίζοντας το elimination game με διαφορετική σειρά π, προκύπτει ότι το $tw(G) =$ min max clique -1. □

Λήμμα 2.18. *[6] Για ένα γράφημα $G = (V, E)$ ισχύει ότι το $tw(G) =$ min front size. Δηλαδή αν παίξουμε το elimination game για όλους τους συνδυασμούς του π και για κάθε συνδυασμό κρατήσουμε την τιμή του μεγαλύτερου $C_\pi(u)$ και από αυτά τα $C_\pi(u)$ κρατήσουμε το μικρότερο, τότε αυτή η τιμή ισούται με το δενδροπλάτος του G.*

Απόδειξη. Θα δείξουμε πρώτα ότι ισχύει η σχέση: min front size $=$ min max clique - 1

$\geq$: Θυμίζουμε ότι αν μία κορυφή έχει επιλεχθεί τότε δεν γίνεται να υπάρξουν μελλοντικές fill edges προς αυτήν. Αυτό σημαίνει ότι αν στο G_π^+ υπάρχει μία κλίκα μεγέθους a τότε η κορυφή u που επιλέχθηκε πρώτη από όλες τις κορυφές που ανήκουν στην κλίκα (η κορυφή με το μικρότερο $\pi(u)$) θα έχει $C_\pi(u) \geq a - 1$, καθώς το $u \cup N(u)$ ενάγει μία κλίκα μεγέθους a, όπου $N(u)$ οι γείτονες της u που δεν έχουν επιλεχθεί ακόμα. Επομένως αν για κάθε επιλογή του π η μέγιστη κλίκα στο G_π^+ είναι μεγέθους a τότε υπάρχει $\exists u$ ώστε $C_\pi(u) \geq a-1$, δηλαδή front size $\geq$ clique number -1 και min front size $\geq$ min max clique -1

$\leq$: Αν υπάρχει κορυφή u ώστε $C_\pi(u) = a - 1$ τότε όταν διαλέξουμε την κορυφή u στο elimination game θα δημιουργηθεί η κλίκα $u \cup N(u)$ όπου $N(u)$ οι γείτονες της u που δεν έχουν επιλεχθεί ακόμα. Αυτή η κλίκα είναι μεγέθους a, δηλαδή για το G_π^+ ισχύει ότι front size $\leq$ clique number - 1 και min front size $\leq$ min max clique -1.

Επομένως ισχύει ότι min front size $=$ min max clique -1 και από το Λήμμα 2.17 συμπεραίνουμε ότι $tw(G) =$ min front size. □

2.5 Διαχωριστής

Οι διαχωριστές έχουν άμεση σχέση με τον δενδροπλάτος επειδή κάθε εσωτερική τσάντα της δενδροαποσύνθεσης είναι ένας υποψήφιος κορυφοδιαχωριστής. Πάνω σε αυτό βασίζονται οι περισσότεροι αλγόριθμοι που βρίσκουν το δενδροπλάτος ενός γραφήματος. Δηλαδή προσπαθούν να βρουν έναν κορυφοδιαχωριστή και μετά αναδρομικά προσπαθούν να λύσουν το αρχικό πρόβλημα σε μικρότερα υπογραφήματα. Επιπλέον μέσω του διαχωριστή αποδεικνύονται και πολλές άλλες σχέσεις του δενδροπλάτους με άλλες παραμέτρους. Σε αυτήν την ενότητα θα δοθούν κάποιοι βασικοί ορισμοί γύρω από την έννοια του διαχωριστή και κάποιες σημαντικές ιδιοτητές με σκοπό να τις χρησιμοποιήσουμε αυτές στα επόμενα κεφάλαια για την δημιουργία αλγορίθμων.

Ορισμός 2.18. *Έστω a ένας πραγματικός αριθμός μεταξύ 0 και 1. a-κορυφοδιαχωριστής ενός γραφήματος $G = (V, E)$ είναι ένα σύνολο $S \subseteq V$ κορυφών, έτσι ώστε κάθε συνεκτική συνιστώσα του $G[V - S]$ που προκύπτει από το G αφαιρώντας το S, να έχει το πολύ $a \cdot |V|$ κορυφές.*

Ορισμός 2.19. *Για $W \subseteq V$, ένας a-κορυφοδιαχωριστής του W στο $G = (V, E)$ είναι ένα σύνολο $S \subseteq V$ κορυφών, έτσι ώστε κάθε συνεκτική συνιστώσα του $G[V - S]$ που προκύπτει από το G αφαιρώντας το S, να έχει το πολύ $a \cdot |W|$ κορυφές από το W. Αυτόν τον κορυφοδιαχωριστή (το σύνολο S δηλαδή) θα τον συμβολίζουμε και ως (k, W, a)-διαχωριστής, όπου $|S| \leq k$.*

Κάθε (k, W, a)-διαχωριστής είναι και (k, W, a')-διαχωριστής με $a' \geq a$. Ο λόγος είναι ότι το a θέτει το άνω όριο για το μέγεθος των συνεκτικών συνιστωσών. Επομένως ανεβάζοντας το άνω όριο δεν αλλάζει κάτι στον κορυφοδιαχωριστή.

Ορισμός 2.20. *Το* separation number *ενός γραφήματος $G = (V, E)$ το συμβολίζουμε με $sep_a(G)$ και είναι ο μικρότερος αριθμός k ώστε να υπάρχει κάποιος (k, W, a)-διαχωριστής για κάθε $W \subseteq V$.*

Δεν είναι απαρραίτητο το σύνολο κορυφών από το οποίο αποτελείται ο διαχωριστής να είναι το ίδιο για κάθε W. Αυτό που μας ενδιαφέρει στον από πάνω ορισμό είναι το μέγεθος του κάθε διαχωριστή να μην υπερβαίνει το k. Το separation number αναφέρεται σε κάθε δυνατό υποσύνολο κορυφών του G και για αυτό είναι ένα μέτρο κατά πόσο δύσκολο είναι να διαχωριστεί το δυσκολότερο υποσύνολο κορυφών του G. Για αυτόν τον λόγο κάθε υπογράφημα του G έχει separation number το πολύ ίσο με το separation number του G. Αυτό γιατί αν στο υπογράφημα του G υπάρχει W που είναι δύσκολο να διαχωριστεί, τότε αυτό το W θα υπάρχει και στο G. Με την φράση ”δύσκολο να διαχωριστεί” εννούμε ότι ο διαχωριστής του W έχει μεγαλύτερο μέγεθος σε σχέση με κάποιο άλλο W που είναι πιο εύκολο να διαχωριστεί στο G.

Μια παραλλαγή του ορισμού 2.19 είναι στις συνεκτικές συνιστώσες που προκύπτουν να επιτρέπουμε το μεγέθος τους να είναι το πολύ $a \cdot |W \setminus S|$. Δηλαδή τώρα δεν λαμβάνουμε υπόψιν τον διαχωριστή στο μέγεθος των συνεκτικών συνιστωσών. Αν τον Ορισμό 2.19 τον ονομάσουμε $type_1$ και τον ορισμό της παραλλαγής τον ονομάσουμε $type_2$ τότε

μπορούμε αντίστοιχα να ορίσουμε και το $sep_a(G)$ του $type_2$. Αν έχουμε έναν (k, W, a)-διαχωριστή($type_2$) τότε όλες οι συνεκτικές συνιστώσες έχουν μέγεθος το πολύ $a \cdot |W \setminus S|$. Ισχύει ότι $a \cdot |W \setminus S| \leq a \cdot |W|$, επομένως για τον διαχωριστή που έχουμε ικανοποιούνται οι συνθήκες για το μέγεθος των συνεκτικών συνιστωσών ακόμα και αν ήταν $type_1$. Επομένως ένας (k, W, a)-διαχωριστής($type_2$) είναι και (k, W, a)-διαχωριστής($type_1$). Το αντίθετο βέβαια δεν ισχύει καθώς ο διαχωριστής $type_2$ έχει πιο χαμηλό άνω όριο, επομένως υπάρχει το ενδεχόμενο να χρειαζόμαστε έναν μεγαλύτερο διαχωριστή S για να ικανοποιήσουμε την συνθήκη. Για αυτόν τον λόγο ισχύει ότι $sep_a(G)(type_1) \leq sep_a(G)(type_2)$.

Θα δείξουμε τώρα την σχέση που έχει το δενδροπλάτος με το separation number για τα γενικά γραφήματα. Πρώτα όμως θα δείξουμε τι συμβαίνει στα δένδρα γιατί η απόδειξη των γενικών γραφημάτων στηρίζεται πάνω στα δένδρα και συγκεκριμένα στο elimination tree.

Λήμμα 2.19. *Έστω T δέντρο με n κορυφές και W ένα υποσύνολο κορυφών. Υπάρχει αλγόριθμος που τρέχει σε $O(n)$ και βρίσκει μία κορυφή u, τέτοια ώστε κάθε συνεκτική συνιστώσα του $T - u$ να περιέχει το πολύ $\frac{1}{2} \cdot W$ κορυφές του W.*

Απόδειξη. Θέτουμε $a = \frac{1}{2} \cdot W$. Ο αλγόριθμος ξεκινάει από μια τυχαία κορυφή u και ακολουθεί τα παρακάτω βήματα.

- Αν καμία συνεκτική συνιστώσα του $T - u$ δεν περιέχει περισσότερες κορυφές από a που ανήκουν στο W, τότε σταμάτα.
- Διαφορετικά διάλεξε τον γείτονα w της τρέχουσας κορυφής u που περιέχει περισσότερες κορυφές από a που ανήκουν στο W.
- Μετακινήσου στην κορυφή w και πήγαινε στο πρώτο βήμα του αλγορίθμου.

Για κάθε κορυφή u μόνο ένας γείτονας w με κορυφές περισσότερες από a που ανήκουν στο W μπορεί να υπάρχει. Αν υπήρχαν τουλάχιστον δύο τέτοιοι γείτονες τότε το άθροισμα των κορυφών τους θα ξεπερνούσε το $|W|$, κάτι που δεν γίνεται. Επιπλέον όταν μετακινούμαστε σε έναν γείτονα w τότε αυτό σημαίνει ότι η συνιστώσα που ανήκει το w περιέχει περισσότερες από a κορυφές του W. Επομένως όλες οι άλλες συνιστώσες μαζί έχουν λιγότερες από a κορυφές του W και έτσι όταν μετακινηθούμε στον γείτονα w η συνιστώσα που περιέχει το u δεν θα έχει περισσότερες κορυφές από a του W. Αυτό σημαίνει ότι ο αλγόριθμος ποτέ δεν θα γυρίσει σε κορυφή που έχει ήδη επισκεφτεί και για αυτό τρέχει σε $O(n)$. □

Το Λήμμα 2.19 ουσιαστικά μας λέει ότι το separation number για $a = \frac{1}{2}$ ενός δέντρου ισούται με 1. Πιο ειδικά ισχύει ότι $sep_{\frac{1}{2}}(T)(type_1) = 1$. Ο από πάνω αλγόριθμος αναφέρεται στον διαχωριστή του $type_1$ και όχι του $type_2$ γιατί για παράδειγμα αν έχουμε ένα δέντρο με δύο κορυφές που ενώνονται με μία ακμή και για W έχουμε όλο το V, τότε αν ο διαχωριστής S είναι μεγέθους 1 θα πρέπει οι συνεκτικές συνιστώσες να έχουν μέγεθος το πολύ $\frac{1}{2} \cdot |V \setminus S|$ που στην περιπτωσή μας ισούται με $\frac{1}{2}$. Η μία και μοναδική όμως συνιστώσα που απομένει έχει μέγεθος 1, οπότε για τον διαχωριστή $type_2$ το separation number για $a = \frac{1}{2}$ δεν είναι

πάντα 1 αλλά και ποτέ δεν είναι πάνω από 2. Το επόμενο λήμμα δίνει την γενική σχέση μεταξύ του δενδροπλάτους και του separation number, δηλαδή γενικεύει την από πάνω σχέση από τα δένδρα(trees) στα partial k-trees.

Θεώρημα 2.8. *[6] Έστω $G = (V, E)$ με δενδροπλάτος το πολύ k και W ένα υποσύνολο κορυφών του G. Υπάρχει ένας $\frac{1}{2}$-κορυφοδιαχωριστής του W στο G μεγέθους το πολύ $k+1$. Δηλαδή ισχύει ότι $sep_{\frac{1}{2}}(G) \leq k + 1$.*

Απόδειξη. Θέτουμε $a = \frac{1}{2} \cdot W$. Για το γράφημα G παίζουμε το elimination game και παίρνουμε το chordal completion του που είναι το γράφημα G^+_π και το elimination tree που είναι το T_π. Τρέχουμε τον ίδιο αλγόριθμο όπως στο Λήμμα 2.19 στο T_π και βρίσκουμε την κορυφή u για την οποία ισχύει ότι κάθε συνεκτική συνιστώσα του $T_\pi - u$ δεν περιέχει περισσότερες κορυφές από a που ανήκουν στο W. Σκοπός μας είναι να διατηρήσουμε αυτές τις συνεκτικές συνιστώσες όπως είναι και στο G, δηλαδή να μην προσθέσουμε επιπλέον κορυφές σε αυτές και να βρούμε έναν διαχωριστή μεγέθους το πολύ $k+1$ στο G^+_π(άρα και στο G). Με βάση το Λήμμα 2.16 αν δύο κορυφές δεν έχουν την σχέση προγόνου-απογόνου στο T_π δεν θα ενώνονται ούτε στο G^+_π. Ένας άλλος τρόπος να υπάρχει μονοπάτι μεταξύ δύο κορυφών στο G^+_π είναι μέσω άλλων κορυφών που είναι πρόγονοι και των δύο στο T_π. Επομένως αν αφαιρέσουμε την κορυφή u και όλους τους προγόνους της, τότε δεν υπάρχει μονοπάτι προς τις συνεκτικές συνιστώσες του $T_\pi - u$ ούτε στο G^+_π ούτε στο G. Ο διαχωριστής $S = \{u\} \cup A(u)$(όπου $A(u)$ οι πρόγονοι του u στο T_π) όμως μπορεί να υπερβαίνει το μέγεθος $k + 1$ και για αυτό θα δείξουμε ότι μπορούμε να διαχωρίσουμε τις συνιστώσες και με μικρότερο διαχωριστή. Ο διαχωριστής δεν χρειάζεται να περιέχει όλο το $A(u)$, αλλά μόνο τις κορυφές που ανήκουν στο $A(u)$ και ενώνονται με ακμή στο G^+_π με κάποιον απόγονο του u στο T_π. Αυτούς τους ξεχωριστούς προγόνους του u στο T_π θα τους ονομάσουμε $B(u)$. Με την ίδια λογική που ακολουθήθηκε στην απόδειξη του Λήμματος 2.16, αν μία κορυφή w ενώνεται με κάποια κορυφή από το $B(u)$, τότε όταν διαλέξουμε την κορυφή w στο elimination game οι γειτονές του θα πρέπει να δημιουργήσουν κλίκα και η κορυφή από το $B(u)$ θα μπει στο C_π όλων των γειτόνων του w που θα διαλεχθούν πριν την κορυφή του $B(u)$. Πριν από την κορυφή του $B(u)$ όμως στο elimination game θα διαλεχθεί η κορυφή u, επομένως η κορυφή από το $B(u)$ θα ανήκει στο $C_\pi(u)$. Αυτό σημαίνει ότι όλες οι κορυφές του $B(u)$ για κάθε απόγονο του u θα ανήκουν στο $C_\pi(u)$. Επομένως αρκεί να θέσουμε ως διαχωριστή το $S = \{u\} \cup C_\pi(u)$. Από Λήμμα 2.18 όμως ισχύει ότι αν το G έχει δενδροπλάτος ίσο με k τότε υπάρχει π τέτοιο ώστε $\forall w \in V$ να ισχύει ότι $C_\pi(w) \leq k$. Οπότε το μέγεθος του S ισούται το πολύ με $|C_\pi(u) \cup \{u\}| = k + 1$ και ο S διαχωρίζει το W στο G^+_π(άρα και στο G) έτσι ώστε κάθε συνεκτική συνιστώσα να μην περιέχει περισσότερες από a κορυφές του W. □

Αυτό το θεώρημα δείχνει την στενή σχέση όλων των εννοιών που αναφερθήκαν πιο πάνω, καθώς μέσω του elimination game και των chordal γραφημάτων αποδείχθηκε η σχέση που έχει το δενδροπλάτος με το separation number. Επιπλέον το θεώρημα αυτό είναι πολύ σημαντικό για την συνέχεια, καθώς πάνω σε αυτό στηρίζεται η απόδειξη του προσεγγιστικού αλγόριθμου για το δενδροπλάτος που αναφέρεται πιο κάτω. Μια πιο διαισθητική απόδειξη του Θεωρήματος 2.8 υπάρχει στο Λήμμα 7.19 του [8] και χρησιμοποιεί την ίδια

λογική με αυτήν του Λήμματος 2.19 με την διαφορά ότι τρέχει τον αλγόριθμο στην δενδροαποσύνθεση, οπότε βρίσκει μία τσάντα που διαχωρίζει το γράφημα έτσι ώστε κάθε συνεκτική συνιστώσα του να περιέχει το πολύ $\frac{1}{2} \cdot |W|$ κορυφές. Η τσάντα είναι ο διαχωριστής και επειδή το δενδροπλάτος είναι το πολύ k, το μέγεθος της τσάντας και επομένως του διαχωριστή είναι το πολύ $k+1$.

Θα δούμε τώρα κάποιες ιδιότητες των διαχωριστών τις οποίες θα χρησιμοποιήσουμε κυρίως στον FPT προσεγγιστικό αλγόριθμο για την εύρεση του δενδροπλάτους που δίνεται παρακάτω. Στον αλγόριθμο θέλουμε να διαχωρίζουμε το γράφημα σε δύο συνεκτικές συνιστώστες. Σύμφωνα με το παραπάνω θεώρημα αν το δενδροπλάτος του γραφήματος είναι το πολύ k τότε για κάθε W υπάρχει $\frac{1}{2}$-κορυφοδιαχωριστής μεγέθους το πολύ $k+1$. Παρόλα αυτά δεν ξέρουμε τίποτα για το πλήθος των συνιστωσών. Επειδή θέλουμε το πλήθος να είναι δύο, θα γίνουμε πιο ελαστικοί και θα ψάξουμε για $\frac{2}{3}$-κορυφοδιαχωριστή.

Λήμμα 2.20. *[9] Έστω $G = (V, E)$ με δενδροπλάτος το πολύ k και W ένα υποσύνολο κορυφών του G. Υπάρχει ένας $\frac{2}{3}$-κορυφοδιαχωριστής του W στο G μεγέθους το πολύ $k+1$ όπου διαχωρίζει το G σε δύο συνιστώστες.*

Απόδειξη. Αν το δενδροπλάτος είναι το πολύ k, τότε από το Θεώρημα 2.8 υπάρχει κορυφοδιαχωρστής S μεγέθους το πολύ $k+1$ που διαχωρίζει το γράφημα σε συνεκτικές συνιστώσες για κάθε μία για την οποία ισχύει ότι δεν περιέχει περισσότερες από $\frac{1}{2} \cdot |W|$ κορυφές του W. Αν οι συνεκτικές συνιστώσες είναι δύο τότε ο S είναι σωστός κορυφοδιαχωριστής. Αν οι συνεκτικές συνιστώσες είναι τρεις, τότε χωρίς βλάβη της γενικότητας ας υποθέσουμε ότι αυτές είναι οι W_1, W_2, W_3 και ισχύει $\frac{1}{2} \cdot |W| \geq |W_1| \geq |W_2| \geq |W_3|$. Θα ενώσουμε τις δύο μικρότερες και θα δείξουμε ότι $|W_2 \cup W_3| \leq \frac{2}{3} \cdot |W|$. Ας υποθέσουμε ότι αυτό δεν ισχύει και $|W_2 \cup W_3| > \frac{2}{3} \cdot |W|$, τότε όμως δεν γίνεται και οι δύο συνιστώσες W_2 και W_3 να έχουν μέγεθος μικρότερο ή ίσο του $\frac{1}{3} \cdot |W|$ και επειδή έχουμε υποθέσει ότι $|W_2| \geq |W_3|$ θα πρέπει $|W_2| > \frac{1}{3} \cdot |W|$. Επειδή υποθέσαμε ότι $|W_2 \cup W_3| > \frac{2}{3} \cdot |W|$ για την συνιστώσα W_1 που απομένει θα πρέπει να ισχύει ότι $|W_1| \leq \frac{1}{3} \cdot |W|$ γιατί το άθροισμα και των τριών συνιστωσών θα πρέπει να είναι ίσο με $|W|$. Τότε όμως θα ισχύει ότι $|W_1| < |W_2|$ που είναι άτοπο. Επομένως μπορούμε να ενώσουμε τις δύο μικρότερες συνιστώσες και το μέγεθός τους δεν θα ξεπερνάει το $\frac{2}{3} \cdot |W|$. Τώρα αν το πλήθος των συνιστωσών είναι μεγαλύτερο του τρία μπορούμε να ενώνουμε κάθε φορά τις δύο μικρότερες σε μέγεθος συνιστώσες μέχρι να μείνουνε με δύο. Πάντα θα υπάρχουν δύο συνεκτικές συνιστώσες με μέγεθος μικρότερο ή ίσο του $\frac{1}{3} \cdot |W|$ τις οποίες θα μπορούμε να ενώσουμε σε μία, διαφορετικά το άθροισμα όλων θα υπερβαίνει το $|W|$, κάτι που δεν γίνεται. □

Με βάση το από πάνω λήμμα παίρνουμε το επόμενο λήμμα το οποίο θα χρησιμοποιήσουμε στον FPT προσεγγιστικό αλγόριθμο που θα παρουσιαστεί στην συνέχεια. Η χρησιμότητά του έγκειται στο γεγονός ότι ο αλγόριθμος δουλεύει με την χρήση διαχωριστών και εμείς θέλουμε να φράξουμε και το μέγεθος του διαχωριστή αλλά και το μέγεθος των συνεκτικών συνιστωσών.

Λήμμα 2.21. *[8] Έστω $G = (V, E)$ με δενδροπλάτος το πολύ k και W ένα υποσύνολο κορυφών του G έτσι ώστε $|W| = 3 \cdot k + 4$. Υπάρχει ένας $\frac{2}{3}$-κορυφοδιαχωριστής του W στο G μεγέθους το πολύ $k+1$ όπου διαχωρίζει το G σε δύο συνιστώστες W_1 και W_2 για τις*

οποίες ισχύει ότι $k+2 \leq |W_1|+|S_1|, |W_2|+|S_2| \leq 2 \cdot k+2$*. Όπου* $|S_1|$ *και* $|S_2|$ *είναι το πλήθος των κορυφών του συνόλου* $S \cap W$ *που προσθέτουμε στο σύνολο* W_1 *και* W_2 *αντίστοιχα.*

Απόδειξη. Από το Λήμμα 2.20 γνωρίζουμε ότι υπάρχει $\frac{2}{3}$-κορυφοδιαχωριστής S που διαχωρίζει το W σε δύο συνεκτικές συνιστώσες W_1 και W_2. Οπότε $|W_1|, |W_2| \leq \frac{2}{3} \cdot |W| \leq \frac{2}{3} \cdot (3 \cdot k + 4) \leq 2 \cdot k + \frac{8}{3}$. Επειδή όμως το πλήθος των υπογραφημάτων είναι ακέραιος αριθμός το $\frac{8}{3}$ γίνεται 2 και έχουμε ότι $|W_1|, |W_2| \leq 2 \cdot k + 2$.

Οι κορυφές που ανήκουν και στον κορυφοδιαχωριστή S και στο W πρέπει να τις εισάγουμε είτε στο υποσύνολο W_1 είτε στο W_2 και τις συμβολίζουμε S_1 και S_2 αντίστοιχα. Η στρατηγική είναι για κάθε μία κορυφή του υποσυνόλου $S \cap W$ να την εισάγουμε στο μικρότερο σε μέγεθος υποσύνολο μεταξύ των $W_1 \cup S1'$ και $W_2 \cup S2'$ κάθε φορά, όπου $S1'$ και $S2'$ οι κορυφές που έχουμε επιλέξει μέχρι εκείνη την στιγμή. Επειδή $|W| = 3 \cdot k+4 \leq 2 \cdot (2 \cdot k+2)$, πάντα υπάρχει τρόπος να βάλουμε τις κορυφές σε ένα υποσύνολο ώστε να ισχύει ότι $|W_1| + |S_1|, |W_2| + |S_2| \leq 2 \cdot k + 2$.

Θυμίζουμε ότι οι κορυφές του S_1 είναι κορυφές που ανήκαν στο $S \cap W$ και διαλέξαμε να τις εισάγουμε στο σύνολο W_1. Επειδή όμως $|W_1| + |S_1| \leq 2 \cdot k + 2$ τότε το υπόλοιπο του W που μένει είναι το $W_2 \cup S_2$. Ισχύει ότι $|W_1| + |S_1| + |W_2| + |S_2| = |W| \Rightarrow |W_2| + |S_2| \geq 3 \cdot k + 4 - (2 \cdot k + 2) = k + 2$. Αντίστοιχα αποδεικνύεται ότι και $|W_1| + |S_1| \geq k + 2$. □

Ο λόγος που φράξαμε το μέγεθος από πάνω είναι για να φράξουμε στον αλγόριθμο το δενδροπλάτος. Ο λόγος όμως που φράξαμε το μέγεθος από κάτω είναι για τεχνικούς λόγους ώστε να μας βοηθήσει στην απόδειξη της πολυπλοκότητας του αλγορίθμου. Επιπλέον η επιλογή του $|W|$ να είναι ακριβώς $3 \cdot k + 4$ δεν είναι τυχαία και το δείχνουμε αυτό στην εξήγηση του FPT προσεγγιστικού αλγορίθμου.

Για τον πολυωνυμικό αλγόριθμο πάλι θα εκμεταλλευτούμε τις ιδιότητες του κορυφοδιαχωριστή αλλά θα βασιστούμε σε ένα άλλο θεώρημα που συσχετίζει τον $\frac{2}{3}$-κορυφοδιαχωριστή με τον $\frac{1}{2}$-κορυφοδιαχωριστή και επομένως και με το δενδροπλάτος.

Θεώρημα 2.9. *[14] Για ένα γράφημα* $G = (V, E)$ *και ένα σύνολο κορυφών* $W \subseteq V$*, υπάρχει πολυωνυμικός αλγόριθμος που βρίσκει έναν* $(w \cdot \log n, W, \frac{2}{3})$*-κορυφοδιαχωριστή, όπου* w *είναι το ελάχιστο μέγεθος ενός* $(w, W, \frac{1}{2})$*-κορυφοδιαχωριστή.*

Ο συμβολισμός που χρησιμοποιείται για τους κορυφοδιαχωριστές έχει οριστεί προηγουμένως. Το θεώρημα μας εγγυάται ότι για κάθε υποσύνολο κορυφών W, υπάρχει πολυωνυμικός αλγόριθμος που να βρίσκει έναν $\frac{2}{3}$-κορυφοδιαχωριστή του W και το μέγεθος του να μην είναι ποτέ μεγαλύτερο από $O(\log n)$ φορές το $sep_{\frac{1}{2}}(G)$. Με βάση αυτό διατυπώνεται και το επόμενο λήμμα.

Λήμμα 2.22. *[6] Για ένα γράφημα* $G = (V, E)$ *και ένα σύνολο κορυφών* $W \subseteq V$*, υπάρχει σταθερά* $\beta \geq 1$ *και πολυωνυμικός αλγόριθμος που βρίσκει έναν* $\frac{2}{3}$*-κορυφοδιαχωριστή του* W *στο* G *με μέγεθος* $\beta \cdot k \cdot \log n$*, όπου* $n = |V|$ *και* k *το δενδροπλάτος του* G*.*

Απόδειξη. Το separation number αναφέρεται σε όλα τα δυνατά υποσύνολα $W \subseteq V$. Οπότε για ένα W, αν το S είναι ένας $\frac{1}{2}$-κορυφοδιαχωριστής του W στο G, ισχύει ότι $|S| \leq$

$sep_{\frac{1}{2}}(G)$ και από το Θεώρημα 2.8 παίρνουμε ότι $|S| \leq k+1$. Από το Θεώρημα 2.9 όμως γνωρίζουμε ότι υπάρχει πολυωνυμικός αλγόριθμος που βρίσκει έναν $\frac{2}{3}$-κορυφοδιαχωριστή του W στο G με μέγεθος $O(w \cdot \log n)$ όπου w είναι το ελάχιστο μέγεθος ενός $(w, W, \frac{1}{2})$-κορυφοδιαχωριστή. Επειδή όμως $w \leq k+1$ ο κορυφοδιαχωριστής του πολυωνυμικού αλγορίθμου είναι μεγέθους το πολύ $O((k+1) \cdot \log n)$. Αν όμως πάρουμε μία σταθερά $\beta \geq \frac{k+1}{k}$, τότε ο κορυφοδιαχωριστής του πολυωνυμικού αλγορίθμου είναι μεγέθους $\beta \cdot k \cdot \log n$. □

Αυτό το λήμμα είναι που θα χρησιμοποιήσουμε για την υλοποιήση και την απόδειξη του πολυωνυμικού προσεγγιστικού αλγορίθμου που προσπαθεί να βρει το δενδροπλάτος ενός γραφήματος.

2.6 Πλάτος μονοπατιού

Το πλάτος μονοπατιού (pathwidth) ορίζεται ακριβώς όπως το δενδροπλάτος με την διαφορά ότι τώρα η αποσύνθεση θέλουμε να είναι μονοπάτι και όχι δένδρο. Αυτό σημαίνει ότι όλες οι τσάντες, δηλαδή οι κορυφές της αποσύνθεσης μονοπατιού θα έχουν βαθμό ίσο με δύο εκτός τις δύο τσάντες που βρίσκονται στα όρια, οι οποίες θα έχουν βαθμό ίσο με ένα. Βέβαια ένα μονοπάτι είναι και δένδρο, που σημαίνει ότι αν βρούμε μια αποσύνθεση μονοπατιού τότε έχουμε βρει και μία δενδροαποσύνθεση. Αυτός είναι και ο λόγος που το δενδροπλάτος ενός γραφήματος είναι μικρότερο ή ίσο με από το πλάτος μονοπατιού. Η εύρεση του πλάτους μονοπατιού είναι και αυτό δύσκολο πρόβλημα και συγκεκριμένα NP-complete παρόλα αυτά υπάρχουν FPT αλγόριθμοι που το λύνουν. Το πλάτος μονοπατιού ξεκίνησε και αυτό για θεωρητικούς σκοπούς για την θεωρία των graph minors και ορίστηκε από τους Neil Robertson και Paul Seymour. Αυτό όπως και το δενδροπλάτος έχει αρκετές ενδιαφέρουσες σχέσεις με άλλες παραμέτρους. Πέρα από το θεωρητικό κομμάτι έχει χρησιμοποιηθεί το πλάτος μονοπατιού και για πρακτικές εφαρμογές, κάποιες από τις οποίες είναι ο σχεδιασμός VLSI, ο σχεδιασμός γραφημάτων και η γλωσσολογία. Ένας πιο τυπικός ορισμός του πλάτους μονοπατιού φαίνεται παρακάτω.

Ορισμός 2.21. *Μία* αποσύνθεση μονοπατιού *του* $G = (V, E)$ *είναι ένα ζευγάρι* $(P = (I, F), \{X_i \mid i \in I\})$ *όπου το* P *είναι μονοπάτι με κορυφές το σύνολο* I *και ακμές το σύνολο* F *και το* $\{X_i \mid i \in I\}$ *είναι οικογένεια υποσυνόλων του* V*, τέτοια ώστε*

1. $\bigcup_{i \in I} X_i = V$.
2. $\forall \{u, w\} \in E, \exists i \in I$ *τέτοιο ώστε* $u \in X_i, w \in X_i$.
3. $\forall u \in V$ *το σύνολο* $\{i \in I \mid u \in X_i\}$ *ενάγει ένα συνεκτικό υπογράφημα του* P.

Η τρίτη συνθήκη μπορεί να αντικατασταθεί με την ισοδύναμη

- $\forall i, j, k \in I$, αν το j βρίσκεται στο μοναδικό μονοπάτι $i - k$ στο T, τότε $X_i \cap X_k \subseteq X_j$

Επειδή το πλάτος μονοπατιού είναι μια ειδική περίπτωση του δενδροπλάτους, δεν αναλύουμε ξανά εδώ τον ορισμό. Αντίστοιχα με την δενδροαποσύνθεση, λέμε ότι το πλάτος της αποσύνθεσης μονοπατιού είναι το μέγεθος της μέγιστης τσάντας μείον ένα και το πλάτος μονοπατιού είναι το ελάχιστο δυνατό πλάτος μιας αποσύνθεσης μονοπατιού. Το πλάτος μονοπατιού ενός γραφήματος G, το συμβολίζουμε με $pw(G)$.

Θυμίζουμε ξανά ότι η αποσύνθεση μονοπατιού είναι μια πιθανή δενδροαποσύνθεση, που σημαίνει πως κληρονομεί πολλές από τις ιδιοτητές της. Μία από αυτές είναι το γεγονός ότι κάθε εσωτερική τσάντας της αποσύνθεσης μονοπατιού είναι ένας κορυφοδιαχωριστής του αρχικού γραφήματος. Παρακάτω αναφέρονται κάποιες ακόμα ιδιότητες που έχει το πλάτος μονοπατιού με άλλες παραμέτρους.

Λήμμα 2.23. *Έστω γράφημα $G = (V, E)$. Ισχύει ότι $tw(G) \leq pw(G)$.*

Το παραπάνω λήμμα μας λέει αυτό που αναφέραμε ξανά. Ότι το δενδροπλάτος δεν είναι μεγαλύτερο από το πλάτος μονοπατιού. Το ερώτημα είναι όμως, πόσο μεγάλο μπορεί να είναι το πλάτος μονοπατιού σε σχέση με το δενδροπλάτος. Θα απαντηθεί αυτό, αφού πρώτα δείξουμε την σχέση που έχει το πλάτος μονοπατιού με το elimination tree που ορίσαμε στην ενότητα του elimination game.

Λήμμα 2.24. *Έστω γράφημα $G = (V, E)$ και π μία διάταξη κορυφών του τέτοια ώστε το elimination tree T_π που προκύπτει να έχει ύψος h. Τότε ισχύει ότι το πλάτος μονοπατιού του G είναι μικρότερο ή ίσο του h, δηλαδή $pw(G) \leq h$. Οπότε $pw(G) \leq$ min etree height.*

Απόδειξη. Ονόμασε τα φύλλα του δένδρου T_π $u_1, \ldots, u_r$ από αριστερά προς τα δεξιά όπου r το πλήθος των φύλλων. Για $1 \leq j \leq r$ φτιάχνουμε μια αποσύνθεση μονοπατιού με τσάντες τις X_j. Η κάθε X_j ενώνεται με την X_{j-1} αν $j > 1$ και με την X_{j+1} αν $j < r$. Η τσάντα X_j περιέχει το φύλλο u_j και όλους τους προγόνους του μέχρι την ρίζα του δένδρου. Το ύψος του δένδρου είναι h, επομένως μαζί με το φύλλο u_j, η τσάντα X_j περιέχει συνολικά $h + 1$ κορυφές και έτσι το πλάτος μονοπατιού είναι ίσο με h. Θα πρέπει να δείξουμε όμως ότι οι τσάντες X_j ικανοποιούν και τις τρεις συνθήκες του ορισμού της αποσύνθεση μονοπατιού. Με την υπόθεση ότι το G είναι συνεκτικό (διαφορετικά θα κάναμε την ίδια διαδικασία σε κάθε συνεκτική συνιστώσα), όλες οι κορυφές ανήκουν στο T_π, άρα όλες ανήκουν σε κάποια τσάντα και ικανοποιείται η πρώτη συνθήκη. Θυμίζουμε ότι με βάση το Λήμμα 2.16, αν δυο κορυφές ενώνονται τότε έχουν την σχέση προγόνου- απογόνου επομένως θα ανήκουν και στην ίδια τσάντα με τον τρόπο που φτιάξαμε την αποσύνθεση μονοπατιού και έτσι ικανοποιείται και η δεύτερη συνθήκη. Τέλος ισχύει και η τρίτη συνθήκη επειδή κάθε κορυφή u του T_π ανήκει σε τόσες τσάντες όσοι είναι και οι απόγονοι του u που είναι φύλλα. Αυτές όμως οι τσάντες ενώνονται μεταξύ τους λόγω του τρόπου κατασκευής και έτσι οι τσάντες που περιέχουν την κορυφή u ενάγουν ένα συνεκτικό υπογράφημα της αποσύνθεση μονοπατιού. □

Το επόμενο λήμμα θα μας φράξει το min etree height ως προς το δενδροπλάτος. Αυτό το λήμμα δεν ανήκει στην ενότητα του πλάτους μονοπατιού αλλά επειδή η απόδειξη χρησιμοποιεί διαχωριστές και το χρειαζόμαστε για να δείξουμε την σχέση του πλάτος μονοπατιού με το δενδροπλάτος, το παραθέτουμε εδώ. Στο [6] δίνει μια πιο γενική απόδειξη του επόμενου λήμματος αλλά εμείς θα περιοριστούμε σε μια ειδική περιπτωσή του.

Λήμμα 2.25. *Έστω γράφημα $G = (V, E)$ με δενδροπλάτος ίσο το πολύ με k. Τότε υπάρχει διάταξη π τέτοια ώστε το ύψος του T_π να είναι το πολύ $k \cdot \log |V|$. Δηλαδή min etree height $\leq k \cdot \log |V|$.*

Απόδειξη. Από το Θεώρημα 2.8 γνωρίζουμε ότι αν ένα γράφημα έχει δενδροπλάτος το πολύ ίσο με k, τότε για ένα υποσύνολο του υπάρχει ένας $\frac{1}{2}$-κορυφοδιαχωριστής μεγέθους το πολύ $k + 1$. Θα φτιάξουμε μία διάταξη π ως εξής. Βρίσκουμε έναν κορυφοδιαχωριστή S μεγέθους το πολύ $k + 1$ και τοποθετούμε τις κορυφές του στο τέλος της διάταξης και κάνουμε την ίδια αναδρομική διαδικασία στα υπογραφήματα $G - S$. Παρατηρούμε τώρα ότι τα υπογραφήματα δεν ενώνονται μεταξύ τους, επομένως αν παίξουμε το elimination game στην διάταξη π πάλι αυτά τα υπογραφήματα δεν θα ενώνονται μεταξύ τους. Αν κάθε κορυφοδιαχωριστή τον θεωρήσουμε ένα σύνολο κορυφών, τότε μια κορυφή u έχει σαν πατέρα μια άλλη κορυφή w στο δέντρο T_π αν το w είτε ανήκει στο ίδιο σύνολο κορυφών με το u, είτε το w ανήκει στον κορυφοδιαχωριστή που προηγήθηκε αμέσως πριν τον κορυφοδιαχωριστή που ανήκει το u. Κάθε κορυφοδιαχωριστής περιέχει το πολύ $k+1$ κορυφές και επειδή κάθε συνιστώσα έχει κάθε φορά το πολύ τις μισές κορυφές από το υποσύνολο που διαχωρίζουμε, στην χειρότερη περίπτωση το ύψος του δένδρου θα είναι $O(k \cdot \log |V|)$. □

Τώρα είμαστε έτοιμοι να φράξουμε το πλάτος μονοπατιού γιατί γνωρίζουμε ότι το ύψος του T_π είναι φραγμένο ως προς το δενδροπλάτος και το πλάτος μονοπατιού φραγμένο ως προς το min etree height.

Θεώρημα 2.10. *Έστω $G = (V, E)$. Το πλάτος μονοπατιού είναι μεγαλύτερο του δενδροπλάτους το πολύ κατά έναν παράγοντα $\log |V|$. Δηλαδή $pw(G) \leq tw(G) \cdot \log |V|$.*

Απόδειξη. Από Λήμμα 2.24 ισχύει ότι $pw(G) \leq$ min etree height και από Λήμμα 2.25 ότι min etree height $\leq tw(G) \cdot \log |V|$, οπότε $pw(G) \leq tw(G) \cdot \log |V|$. □

Γίνεται εύκολα αντιληπτό ότι μπορούμε να βρούμε και άλλες σχέσεις μεταξύ των παραμέτρων που ορίστηκαν στις ενότητες συνδυάζοντας τις σχέσεις που έχουμε αναφέρει. Ένα τέτοιο θεώρημα που συνοψίζει όλες τις σχέσεις κάποιων παραμέτρων υπάρχει στο paper [6].

3. ΑΛΓΟΡΙΘΜΟΙ ΓΙΑ ΤΟ ΔΕΝΔΡΟΠΛΑΤΟΣ

3.1 Εισαγωγή στους προσεγγιστικούς αλγορίθμους

Όταν μιλάμε για προσεγγιστικούς αλγόριθμους χρειαζόμαστε ένα μέτρο για το πόσο καλή λύση παίρνουμε. Δηλαδή πόσο κοντά είναι η λύση που δίνει ο προσεγγιστικός αλγόριθμος στην πραγματική. Μερικές φορές είναι δύσκολο να το αποδείξουμε αυτό και έτσι δεν μπορούμε να γνωρίζουμε πόσο κοντά στην πραγματική λύση είναι το αποτέλεσμα που παίρνουμε. Υπάρχουν όμως προβλήματα για τα οποία γνωρίζουμε προσεγγιστικούς αλγορίθμους που τα λύνουν και γνωρίζουμε και πόσο καλή είναι η λύση που μας δίνει ο αλγόριθμος.

Ορισμός 3.1. *Έστω ένα πρόβλημα Π και ένας αλγόριθμος A. Ο A είναι ρ-προσεγγιστικός αν έχει αποδειχθεί ότι η συνάρτηση κόστους $f(x)$ της προσεγγιστικής λύσης $A(x)$, όπου x είναι μια περίπτωση του προβλήματος Π, δεν είναι μεγαλύτερη ή μικρότερη (ανάλογα αν το πρόβλημα είναι ελαχιστοποίησης ή μεγιστοποίησης) κατά έναν παράγοντα ρ από την συνάρτηση κόστους $OPT(x)$ της βέλτιστης λύσης.*

Ισχύει δηλαδή ότι:

- $OPT(x) \leq f(x) \leq \rho \cdot OPT(x)$, $\rho \geq 1$, *αν Π πρόβλημα ελαχιστοποίησης.*
- $\rho \cdot OPT(x) \leq f(x) \leq OPT(x)$, $\rho \leq 1$, *αν Π πρόβλημα μεγιστοποίησης.*

Όπως ειπώθηκε προηγουμένως το πρόβλημα που ορίζεται ως εξής: ”Δίνεται ως είσοδο ένα γράφημα $G = (V, E)$ και ένας ακέραιος k. Αποφάσισε αν το δενδροπλάτος του G είναι το πολύ k” είναι NP-complete πρόβλημα. Αυτό σημαίνει ότι δεν γνωρίζουμε κάποιον πολυωνυμικό αλγόριθμο που να λύνει το πρόβλημα αποδοτικά και με ακρίβεια. Παρόλα αυτά σε πραγματικές εφαρμογές δεν χρειαζόμαστε πάντα ακρίβεια και ταχύτητα στην λύση μας. Για αυτό έχουν αναπτυχθεί διάφοροι αλγόριθμοι που προσπαθούν να βρουν το δενδροπλάτος ενός γραφήματος και είναι ελαστικοί είτε στον χρόνο είτε στην ακρίβεια της λύσης. Κάποιοι από αυτούς τρέχουν σε πολυωνυμικό χρόνο ακόμα και αν το k δεν είναι φραγμένο αλλά η λύση τους είναι προσεγγιστική του πραγματικού δενδροπλάτους. Κάποιοι άλλοι λύνουν με ακρίβεια το πρόβλημα του δενδροπλάτους αλλά τρέχουν σε εκθετικό χρόνο ως προς την σταθερά k. Ένας τέτοιος αλγόριθμος δίνεται στο [2]. Αυτός ο αλγόριθμος δεν είναι προσεγγιστικός αλλά η σταθερά που βρίσκεται στο big-O είναι αρκετά μεγάλη και πρακτικά δεν είναι και πολύ χρήσιμος ο αλγόριθμος γιατί τρέχει αρκετά αργά. Για αυτόν τον λόγο έχουν αναπτυχθεί γρηγορότεροι στην πράξη αλγόριθμοι που είναι εκθετικοί ως προς την σταθερά k και προσεγγιστικοί ως προς την λύση. Θα μπορούσαμε να αναρωτηθούμε ποιος είναι ο λόγος που ένας εκθετικός ως προς την σταθερά k και προσεγγιστικός αλγόριθμος μπορεί να προτιμηθεί από έναν προσεγγιστικό αλγόριθμο που τρέχει σε πολυωνυμικό χρόνο. Ο λόγος είναι ότι οι προσεγγίσεις είναι διαφορετικές και οι αλγόριθμοι που είναι εκθετικοί ως προς k μπορεί να δίνουν ένα αποτέλεσμα πολύ πιο κοντά στην πραγματική λύση. Πιο ειδικά για ένα προσεγγιστικό πρόβλημα ελαχιστο-

ποίησης μπορούμε να κατατάξουμε τους αλγορίθμους σε τρεις κατηγορίες από τις οποίες οι δύο τελευταίες προκύπτουν από τον ορισμό του ρ-προσεγγιστικού αλγορίθμου.

- Τύπου 1: Η προσεγγιστική λύση του αλγορίθμου διαφέρει από την πραγματική λύση κατά μία θετική σταθερά. Αν δηλαδή $A(I)$ είναι η προσεγγιστική λύση και $OPT(I)$ η βέλτιστη λύση, τότε ισχύει ότι $A(I) - OPT(I) \leq K$ για οποιαδήποτε θετική σταθερά K.
- Τύπου 2: Η προσεγγιστική λύση του αλγορίθμου διαφέρει από την πραγματική λύση κατά έναν παράγοντα a. Αν δηλαδή $A(I)$ είναι η προσεγγιστική λύση και $OPT(I)$ η βέλτιστη λύση, τότε ισχύει ότι $OPT(I) \cdot a \leq A(I)$ για οποιαδήποτε θετική σταθερά $a > 1$.
- Τύπου 3: Η διαφορά μεταξύ της προσεγγιστικής λύσης του αλγορίθμου και της βέλτισης λύσης μεγαλώνει ανάλογα με το μέγεθος του προβλήματος. Αν δηλαδή $A(I)$ είναι η προσεγγιστική λύση και $OPT(I)$ η βέλτιστη λύση, ισχύει ότι $A(I) = O(f(n) \cdot OPT(I))$, όπου $f(n)$ μια συνάρτηση που εξαρτάται από το μέγεθος της εισόδου.

Όσο πιο μικρός είναι ο αριθμός του τύπου που ορίστηκε από πάνω, τόσο πιο δύσκολο είναι να βρεθεί ένας τέτοιος προσεγγιστικός αλγόριθμος για ένα πρόβλημα ελαχιστοποίησης. Πολύ λίγα NP-complete προβλήματα έχουν προσεγγιστικούς αλγορίθμους τύπου 1. Για το δικό μας πρόβλημα που είναι η εύρεση του δενδροπλάτους έχει αποδειχθεί στο [6] ότι αν υποθέσουμε πως P $\neq$ NP-complete, τότε δεν υπάρχει πολυωνυμικός προσεγγιστικός αλγόριθμος τύπου 1. Ο αλγόριθμος FPT που θα παρουσιαστεί στην συνέχεια είναι τύπου 2. Είναι ανοιχτό πρόβλημα όμως το αν υπάρχει πολυωνυμικός αλγόριθμος που να είναι τύπου 2. Για τον τύπο 3 έχει βρεθεί πολυωνυμικός αλγόριθμος ο οποίος παρατίθεται παρακάτω. Με βάση τα από πάνω, καταλαβαίνουμε τώρα ότι ένας FPT αλγόριθμος τύπου 2 μπορεί για μεγάλες τιμές του k να είναι πιο αργός από τον προσεγγιστικός πολυωνυμικό αλγόριθμο τύπου 3, αλλά ανάλογα το μέγεθος του προβλήματος μπορεί να μας δίνει πιο ικανοποιητικές απαντήσεις που σημαίνει ότι αν δεν μας απασχολεί ο χρόνος πάρα πολύ θα επιλέξουμε τον FPT προσεγγιστικό αλγόριθμο τύπου 2.

3.2 Πολυωνυμικός προσεγγιστικός αλγόριθμος

Με βάση το Λήμμα 2.22, στο paper [6] δίνεται ένας πολυωνυμικός προσεγγιστικός αλγόριθμος για το δενδροπλάτος ο οποίος βρίσκει και μία δενδροαποσύνθεση του γραφήματος. Το δενδροπλάτος που βρίσκει αυτός ο αλγόριθμος απέχει το πολύ $O(\log n)$ από την βέλτιστη λύση που είναι το πραγματικό δενδροπλάτος του γραφήματος που δίνουμε ως είσοδο και για αυτό λέμε ότι ο αλγόριθμος είναι τύπου 3. Η λογική που ακολουθεί ο αλγόριθμος είναι να σπάσει το αρχικό πρόβλημα σε μικρότερα προβλήματα και αφού βρει την λύση αναδρομικά σε αυτά τότε να επιστρέψει και να βρει την λύση και για το αρχικό πρόβλημα. Ο τρόπος που σπάει το πρόβλημα σε μικρότερα είναι μέσω των διαχωριστών και αυτός

είναι ο λόγος που χρειαζόμαστε και το Λήμμα 2.22 για να δείξουμε την απόδοση του αλγορίθμου. Πιο ειδικά αν έχουμε ένα γράφημα $G = (V, E)$ και δύο συνεκτικές συνιστώστες A και B που προκύπτουν από το $G[V - S]$ όπου S είναι ένας διαχωριστής, τότε δεν υπάρχουν ακμές που να ενώνουν κορυφές του A και του B. Αυτό σημαίνει ότι δεν χρειάζεται οι κορυφές του A με του B να βρίσκονται σε ίδιες τσάντες, οπότε αν βρούμε μια δενδροαποσύνθεση του $G[A \cup S]$ και του $G[B \cup S]$, τότε μπορουμε να τις ενώσουμε με μια κοινή κορυφή που περιέχει μόνο κορυφές του S και να φτιάξουμε μια δενδροαποσύνθεση του G. Ο λόγος που φτιάχνουμε δενδροαποσύνθεση για το $G[A \cup S]$ και όχι για το $G[A]$ είναι ότι υπάρχουν ακμές μεταξύ του A και του S, τις οποίες πρέπει να τις λάβουμε υπόψιν.

Παρακάτω δίνεται ένας ψευδοκώδικας ο οποίος υλοποιεί τον αλγόριθμο. Η συνάρτηση $solve(Z, W)$ επιστρέφει την δενδροαποσύνθεση του $G[Z \cup W]$ και η ρίζα του δέντρου περιέχει όλες τις κορυφές του W. Οπότε για να βρούμε την δενδροαποσύνθεση ενός γραφήματος $G = (V, E)$, καλούμε την συνάρτηση ως $solve(V, \emptyset)$. Κάτι σημαντικό είναι, ότι πάντα τα Z και W είναι ξένα σύνολα μεταξύ τους. Δηλαδή ισχύει ότι $Z \cap W = \emptyset$.

```
function solve(Z, W) {
    if (3 · |Z| ≤ |W|) {
        return a tree decomposition with one node, containg Z ∪ W;
    } else {
        find a 2/3-vertex separator S of W in G[Z ∪ W];
        find a 2/3-vertex separator S' of Z ∪ W in G[Z ∪ W];
        let G_1, ..., G_t be the connected components of G[Z ∪ W − (S ∪ S')];

        for (i = 1; i ≤ t; i = i + 1) {
            let Z_i = the vertices of G_i in Z;
            let W_i = the vertices of G_i in W;
            T_i = solve(Z_i, W_i ∪ S ∪ S');
        }

        take a new root node r_{Z,W} containing vertices W ∪ S ∪ S';
        add all the tree decompositions returned by the recursive calls
        (each T_i) with an edge from the root of each T_i to r_{Z,W};

        return the new tree decomposition with root the node r_{Z,W};
    }
}
```

Αυτό που κάνει ο αλγόριθμος αρχικά είναι να βρει δύο διαχωριστές S και S'. Ο διαχωριστής S διαχωρίζει το W κάθε φορά και ο λόγος που τον χρειαζόμαστε είναι για να φράξουμε το μέγεθος των W_i, όπως φαίνεται και παρακάτω στην ανάλυση της πολυπλοκότητας. Ο διαχωριστής S' διαχωρίζει όλο το τρέχον γράφημα και ο ρόλος του είναι να δημιουργήσει μικρότερα υποπροβλήματα και αφού βρει την λύση για αυτά να τα συνδυάσει κατάλληλα για να βρει την λύση και για το αρχικό πρόβλημα. Τα G_i είναι οι συνεκτικές συνιστώσες μετά την αφαίρεση των S και S' από το $G[Z \cup W]$. Στην συνέχεια καλούνται τα μικρότερα

υποπροβλήματα και στο τέλος δημιουργείται μια τσάντα που περιέχει όλες τις κορυφές του W και των διαχωριστών και η οποία ενώνεται με τις δενδροαποσυνθέσεις των υποπροβλημάτων. Θα δείξουμε τώρα πιο τυπικά ότι η δενδροαποσύνθεση που προκύπτει από τον αλγόριθμο είναι έγκυρη και το δενδροπλάτος που βρίσκει απέχει το πολύ $O(\log n)$ από το πραγματικό.

Λήμμα 3.1. *[6] Αν Z και W είναι δύο ξένα σύνολα κορυφών του $G = (V, E)$ τότε η συνάρτηση $solve(Z, W)$ επιστρέφει μια δενδροαποσύνθεση του $G[Z \cup W]$ τέτοια ώστε η ρίζα της δενδροαποσύνθεσης να περιέχει όλες τις κορυφές του W. Επίσης, αν $|W| \leq 6 \cdot \beta \cdot k \cdot \log n$, όπου k είναι το πραγματικό δενδροπλάτος του G και $n = |V|$, τότε το δενδροπλάτος της δενδροαποσύνθεσης που βρίσκει ο αλγόριθμος είναι το πολύ $8 \cdot \beta \cdot k \cdot \log n$. Δηλαδή το δενδροπλάτος που βρίσκει ο αλγόριθμος είναι $O(k \cdot \log n)$, που σημαίνει ότι απέχει το πολύ $O(\log n)$ από το πραγματικό δενδροπλάτος.*

Απόδειξη. Αρχικά βλέπουμε ότι τα Z και W είναι δύο ξένα σύνολα κορυφών αφού αρχικά έχουμε ότι $Z = V$ και $W = \emptyset$ και σε κάθε αναδρομική κλήση το πρώτο όρισμα Z_i παίρνει στοιχεία μόνο από το Z που δεν ανήκουν στους διαχωριστές, ενώ το δεύτερο όρισμα παίρνει στοιχεία από το W και από τους διαχωριστές. Επομένως σε κάθε περίπτωση τα δύο σύνολα δεν μοιράζονται κοινά στοιχεία. Ακόμα σε κάθε κλήση της συνάρτησης η ρίζα περιέχει όλες τις κορυφές του W όπως φαίνεται από τον κώδικα.

Θα δείξουμε πρώτα ότι η δενδροαποσύνθεση που δημιουργείται είναι έγκυρη, δηλαδή ικανοποιεί τις τρεις συνθήκες του ορισμού 2.1. Και οι τρεις συνθήκες λόγω της αναδρομής αποδεικνύονται επαγωγικά με την βάση να είναι η κλήση της if στον κώδικα. Η πρώτη συνθήκη ικανοποιείται επειδή κάθε κορυφή αρχικά ανήκει στο $Z = V$ και επομένως είτε θα ανήκει στο Z_i είτε στο $S \cup S'$, όπου επαγωγικά θα ανήκει σε μία τσάντα αφού στην βάση της επαγωγής βάζουμε όλες τις κορυφές σε μία τσάντα. Για την δεύτερη συνθήκη θα πρέπει να δειχθεί ότι για κάθε ακμή $\{u, w\} \in E$ με $u, w \in Z \cup W$, υπάρχει τσάντα X_i τέτοια ώστε $u, w \in X_i$. Αν και το u και το w ανήκουν στο W, τότε από τον κώδικα φαίνεται ότι υπάρχει τσάντα X_i που βάζει όλο το W μέσα σε αυτήν. Αν δεν ισχύει αυτό, τότε αποκλείεται να υπάρχει i, j με $i \neq j$ ώστε $u \in Z_i \cup W_i$ και $w \in Z_j \cup W_j$, επειδή τα $Z_i \cup W_i$ και $Z_j \cup W_j$ είναι δύο διαφορετικές συνιστώσες λόγω των κορυφοδιαχωριστών και επειδή υπάρχει η ακμή $\{u, w\}$ θα πρέπει και το u και το w να ανήκουν στο $Z_i \cup W_i \cup S \cup S'$ για κάποιο i. Ο λόγος που χρειαζόμαστε και τους κορυφοδιαχωριστές είναι επειδή υπάρχει το ενδεχόμενο να ανήκουν οι κορυφές σε κάποιον από αυτούς. Ο κώδικας όμως καλεί αναδρομικά την συνάρτηση $solve(Z_i, W_i \cup S \cup S')$, επομένως επαγωγικά και με την χρήση της βάσης βλέπουμε ότι υπάρχει τσάντα που περιέχει και τις δύο κορυφές. Για την τρίτη συνθήκη θα πρέπει να δείξουμε ότι $\forall u \in Z \cup W$ ισχύει ότι το $T[\{i \in I \mid u \in X_i\}]$ ενάγει ένα συνεκτικό υποδέντρο, όπου I το σύνολο κορυφών της δενδροαποσύνθεσης και T η δενδροαποσύνθεση. Αν το u δεν ανήκει στην τσάντα που δημιουργεί η κλήση της συνάρτησης τότε ανήκει απαραίτητα σε ένα τουλάχιστον Z_i και επαγωγικά με την χρήση της βάσης ισχύει αυτό που θέλουμε να δείξουμε. Διαφορετικά το u ανήκει σε μια συνιστώσα $W_i \cup S \cup S'$ και εισάγεται στην τσάντα $r_{Z,W}$ της τρέχουσας αναδρομικής κλήσης $solve(Z_i, W_i \cup S \cup S')$. Αν $u \in W_i$ τότε το u θα εισαχθεί ξανά μόνο σε ένα παιδί της $r_{Z,W}$ και όλοι άλλοι απογονοί της $r_{Z,W}$ δεν θα έχουν το u αφού θα κληθούν ως $solve(Z_j, W_j \cup S \cup S')$ όπου $j \neq i$ και $u \notin Z_j \cup W_j \cup S \cup S'$. Αν όμως

$u \in S \cup S'$ θα ανήκει σε όλα τα παιδιά της τσάντας $r_{Z,W}$ και επαγωγικά με την χρήση της βάσης θα ισχύει ότι το υποδέντρο είναι συνεκτικό. Σε κάθε περίπτωση, από την στιγμή που δεν ξαναβάλουμε την u σε μια τσάντα τότε αποκλείεται να ξαναεμφανιστεί στα παιδιά αυτής της τσάντας.

Θα δείξουμε τώρα ότι το δενδροπλάτος της δενδροαποσύνθεσης που βρίσκει ο αλγόριθμος είναι το πολύ $8 \cdot \beta \cdot k \cdot \log n$. Αρκεί να δείξουμε ότι $|X_i| \leq 8 \cdot \beta \cdot k \cdot \log n$ για κάθε τσάντα X_i. Για να το δείξουμε αυτό θα πρέπει πρώτα να δείξουμε ότι $|W| \leq 6 \cdot \beta \cdot k \cdot \log n$ για το W κάθε αναδρομικής κλήσης. Η πρώτη κλήση της συνάρτησης θέτει το W ίσο με $\emptyset$, οπότε $|W| = 0$. Σε κάθε αναδρομική κλήση το W γίνεται ίσο με $W_i \cup S \cup S'$. Αυτό που θέλουμε είναι να φράξουμε το $|W|$ κάθε αναδρομικής κλήσης. Έστω X ή μέγιστη τιμή που μπορεί να πάρει το $|W|$, οπότε θέλουμε να ισχύουν αυτές οι δύο ανισότητες $(1)|W| \leq X$ και $(2)|W_i \cup S \cup S'| \leq X$. Επειδή το S είναι ένας $\frac{2}{3}$-κορυφοδιαχωριστής του W θα πρέπει για κάθε W_i να ισχύει ότι $|W_i| \leq \frac{2}{3} \cdot |W|$. Επίσης από το Λήμμα 2.22 το μέγεθος των κορυφοδιαχωριστών S και S' ισούται με $\beta \cdot k \cdot \log n$. Οπότε ανοίγοντας την (2) έχουμε ότι $|W_i \cup S \cup S'| \leq |W_i| + |S| + |S'| \leq \frac{2}{3} \cdot |W| + 2 \cdot \beta \cdot k \cdot \log n \leq X \Rightarrow (3)|W| \leq \frac{3}{2} \cdot X - 3 \cdot \beta \cdot k \cdot \log n$. Για να βρούμε την μέγιστη τιμή X, μετατρέπουμε τις ανισότητες σε ισότητες και από (1) και (3) έχουμε ότι $X = 6 \cdot \beta \cdot k \cdot \log n$. Επομένως η μέγιστη τιμή που μπορεί να πάρει το $|W|$ σε κάθε αναδρομική κλήση είναι η $6 \cdot \beta \cdot k \cdot \log n$. Τώρα μπορούμε να δείξουμε και το ζητούμενό μας. Υπάρχουν δύο περιπτώσεις. Στην πρώτη περίπτωση καλείται το $if(3 \cdot |Z| \leq |W|)$ που φτιάχνει μία τσάντα με μέγεθος $|Z \cup W| \leq |Z| + |W| \leq \frac{|W|}{3} + |W| = \frac{4}{3} \cdot |W| \leq \frac{4}{3} \cdot 6 \cdot \beta \cdot k \cdot \log n = 8 \cdot \beta \cdot k \cdot \log n$. Στην δεύτερη περίπτωση καλείται το $else$ που φτιάχνει μια τσάντα μεγέθους $|W \cup S \cup S'| \leq |W| + |S| + |S'| \leq 6 \cdot \beta \cdot k \cdot \log n + 2 \cdot \beta \cdot k \cdot \log n = 8 \cdot \beta \cdot k \cdot \log n$, όπου για το μέγεθος των $|S|$ και $|S'|$ χρησιμοποιήθηκε το Λήμμα 2.22. Και στις δύο περιπτώσεις το μέγεθος της τσάντας δεν υπερβαίνει το $8 \cdot \beta \cdot k \cdot \log n$, οπότε αποδείχθηκε το ζητούμενο. □

Με βάση τα παραπάνω προκύπτει το εξής θεώρημα που ουσιαστικά μας λέει ότι υπάρχει ένας πολυωνυμικός αλγόριθμος που βρίσκει μια προσεγγιστική δενδροαποσύνθεση ενός γραφήματος. Επειδή όμως το δενδροπλάτος συνδέεται άμεσα με πολλές άλλες παραμέτρους, τότε με την χρήση αυτού του αλγορίθμου μπορούμε να βρούμε προσεγγιστικές τιμές και για άλλες πολλές σημαντικές παραμέτρους.

Θεώρημα 3.1. *[6] Για ένα γράφημα $G = (V, E)$ με $n = |V|$, υπάρχει πολυωνυμικός αλγόριθμος ο οποίος βρίσκει μία δενδροαποσύνθεση του G με δενδροπλάτος το πολύ $O(k \cdot \log n)$, όπου k το πραγματικό δενδροπλάτος του G.*

Απόδειξη. Στο Λήμμα 3.1 αποδείξαμε ότι η συνάρτηση $solve(Z, W)$ μας επιστρέφει μια έγκυρη δενδροαποσύνθεση με δενδροπλάτος ίσο με $O(k \cdot \log n)$. Ακόμα, η εύρεση των κορυφοδιαχωριστών γίνεται σε πολυωνυμικό χρόνο και επιπλέον κάθε φορά το $Z_i \cup W_i \cup S \cup S'$ είναι μικρότερο του $Z \cup W$. Επομένως ο αλγόριθμος τρέχει σε πολυωνυμικό χρόνο ως προς το μέγεθος της εισόδου. □

3.3 FPT προσεγγιστικός αλγόριθμος

Σε αυτήν την ενότητα θα παρουσιαστεί ένας FPT προσεγγιστικός αλγόριθμος για το δενδροπλάτος. Σε πολλά πρακτικά προβλήματα το k που είναι το δενδροπλάτος είναι μία φραγμένη μικρή σταθερά και για αυτό δεν μας ενοχλούν στην πολυπλοκότητα του αλγορίθμου παράγοντες που είναι εκθετικοι ως προς το k. Επίσης ο αλγόριθμος είναι προσεγγιστικός γιατί αν το δενδροπλάτος του γραφήματος που δίνουμε σαν είσοδο είναι k ο αλγόριθμος θα επιστρέψει μία δενδροαποσύνθεση με δενδροπλάτος το πολύ $4 \cdot k + 4$. Αυτός ο αλγόριθμος είναι τύπου 2 επειδή η απάντηση που δίνει διαφέρει κατά έναν παράγοντα a από το πραγματικό δενδροπλάτος. Αν $a = 8$, τότε για $k \geq 1$ ισχύει ότι $4 \cdot k + 4 \leq a \cdot k$ που σημαίνει ότι θα μπορούσαμε να πουμε ότι ο αλγόριθμος δίνει μια προσεγγιστική λύση που διαφέρει κατά έναν παράγοντα $a \geq 8$. Η όλη λογική του αλγορίθμου βασίζεται πάλι πάνω στην σχέση που υπάρχει μεταξύ των κορυφοδιαχωριστών και του δενδροπλάτους όπως και στον πολυωνυμικό προσεγγιστικό αλγόριθμο που παρουσιάστηκε προηγουμένως.

Πιο ειδικά στην συνάρτηση $solve(Z, W)$ το Z είναι το γράφημα για το οποίο θέλουμε να βρούμε την δενδροαποσύνθεση και το W είναι ένα υπογράφημα του Z το οποίο θέλουμε να διασπάσουμε ομοιόμορφα. Το W στην αρχή είναι κενό και ποτέ δεν υπερβαίνει σαν μέγεθος το $3 \cdot k + 4$. Αν είναι μικρότερο από αυτό, τότε αυθαίρετα του προσθέτουμε κορυφές. Ο λόγος είναι ότι με βάση το Λήμμα 2.21 θα βρούμε έναν διαχωριστή που διαχωρίζει το Z σε δύο ακριβώς συνιστώσες έτσι ώστε καμία από αυτές να μην έχει περισσότερες από $\frac{2}{3} \cdot W$ κορυφές από το W. Αφού βρει ο αλγόριθμος τις δύο συνιστώσες, συνεχίζει την δενδροαποσύνθεση καλώντας την συνάρτηση για αυτές. Η τσάντα που δημιουργείται σε κάθε κλήση περιέχει όλον τον διαχωριστή και όλο το W. Είμαστε σίγουροι ότι οι δύο συνιστώσες μεταξύ τους δεν ενώνονται με μία ακμή αφού έχουν διαχωριστεί και για αυτό μπορούμε να βρούμε τις δενδροαποσυνθέσεις τους ξεχωριστά. Παρακάτω φαίνεται ο αλγόριθμος. Στην αρχή καλούμε την συνάρτηση ως $solve(G, \emptyset)$ και γενικά η $solve(Z, W)$ μας επιστρέφει μια δενδροαποσύνθεση του $G[Z]$ έτσι ώστε η ρίζα αυτής να έχει όλο το W.

```
function solve(Z, W) {
    if (|Z| ≤ 4 · k + 5) {
        return a tree decomposition with one node, containing Z;
    }

    if (|W| < 3 · k + 4) {
        augment W arbitarily to the size 3 · k + 4;
    }

    find a 2/3-vertex separator S of W in G[Z];
    let G_1, G_2 be the connected components of G[Z − S];

    for (i = 1; i ≤ 2; i = i + 1) {
        let Z_i = the vertices of induced subgraph G_i ∪ S;
        let W_i = the vertices of (G_i ∩ W) ∪ S;
        T_i = solve(Z_i, W_i);
```

```
    }

    take a new root node r_{Z,W} containing vertices W ∪ S;
    add all the tree decompositions returned by the recursive calls
    (each T_i) with an edge from the root of each T_i to r_{Z,W};

    return the new tree decomposition with root the node r_{Z,W};
}
```

Με βάση τον από πάνω ψευδοκώδικα προκύπτει το επόμενο θεώρημα. Σε όλα τα παρακάτω υποθέτουμε ότι το γράφημα είναι συνεκτικό γιατί σε διαφορετική περίπτωση μπορούμε να καλέσουμε τον αλγόριθμο τόσες φορές όσες και οι συνεκτικές του συνιστώσες και να ενώσουμε τις δενδροαποσυνθέσεις με μια κοινή τσάντα. Ακόμα υποθέτουμε ότι το πλήθος των ακμών είναι το πολύ $k \cdot |V|$, γιατί αλλιώς το δενδροπλάτος του γραφήματος θα υπερβαίνει το k.

Θεώρημα 3.2. *[8], [9] Υπάρχει αλγόριθμος τέτοιος ώστε αν του δώσουμε ως είσοδο ένα γράφημα $G = (V, E)$ και έναν ακέραιο k μας επιστρέφει μια δενδροαποσύνθεση του G με δενδροπλάτος το πολύ $4 \cdot k + 4$ ή συμπεραίνει ότι το δενδροπλάτος του G είναι μεγαλύτερο του k. Ο χρόνος εκτέλεσης του είναι $O(8^k \cdot k^2 \cdot |V|^2)$.*

Απόδειξη. Αρχικά θα δείξουμε ότι η δενδροαποσύνθεση είναι έγκυρη, δηλαδή ικανοποιεί όλες τις συνθήκες του ορισμού 2.1. Οι συνθήκες αποδεικνύονται επαγωγικά και η βάση της επαγωγής είναι η πρώτη if στον κώδικα από πάνω. Η πρώτη συνθήκη ικανοποιείται γιατί κάθε κορυφή ανήκει σε μία συνεκτική συνιστώστα και λόγω της βάσης τελικά κάθε κορυφή εισάγεται σε μία τουλάχιστον τσάντα. Η δεύτερη συνθήκη ικανοποιείται επειδή αν δύο κορυφές ενώνονται με μία ακμή θα ανήκουν και στην ίδια συνιστώστα και επομένως λόγω της βάσης θα υπάρχει τσάντα που τις περιέχει και τις δύο. Για την τρίτη συνθήκη θα πρέπει να δείξουμε ότι οι τσάντες που περιέχουν μια κορυφή ενάγουν ένα υποδέντρο. Αν μία κορυφή εισαχθεί σε μία τσάντα τότε είτε βρισκόμαστε στην βάση όπου και σταματάμε, είτε η κορυφή ανήκει στο σύνολο $W \cup S$. Στην δεύτερη περίπτωση η κορυφή θα ανήκει αποκλειστηκά σε μία συνεκτική συνιστώσα και στην άλλη συνεκτική συνιστώσα δεν πρόκειται να εμφανιστεί ποτέ. Επομένως αν μια κορυφή δεν υπάρχει σε μία τσάντα, τότε δεν θα υπάρχει ούτε στους απογόνους αυτής της τσάντας και για αυτό ισχύει και η τρίτη συνθήκη.

Επιπλέον θα πρέπει να δείχθεί ότι το δενδροπλάτος της δενδροαποσύνθεσης που βρίσκει ο αλγόριθμος είναι το πολύ $4 \cdot k + 4$. Αρκεί να δείξουμε ότι για κάθε τσάντα X_i ισχύει ότι $|X_i| \leq 4 \cdot k + 5$. Πριν από αυτό όμως θα πρέπει να φράξουμε το μέγεθος του W. Η πρώτη κλήση της συνάρτησης θέτει το $W = \emptyset$, οπότε $|W| = 0$ και λόγω της δεύτερης if έχουμε ότι $|W| = 3 \cdot k + 4$. Σε κάθε καινούργια κλήση ισχύει ότι $|W_i| \leq |G_i \cap W| + |S|$. Ισχύει ότι $|G_i \cap W| \leq 2 \cdot k + 2$ από το Λήμμα 2.21 γιατί ο S είναι $\frac{2}{3}$-κορυφοδιαχωριστής. Επίσης το μέγεθος του S είναι το πολύ $k+1$, δηλαδή $|S| \leq k+1$. Επομένως $|W_i| \leq 2 \cdot k + 2 + k + 1 \Rightarrow |W_i| \leq 3 \cdot k + 3$. Αυτό σημαίνει ότι σε κάθε κλήση της συνάρτησης ισχύει ότι $|W| \leq 3 \cdot k + 4$. Τελικά, επειδή σε κάθε κλήση φτιάχνουμε μια τσάντα που περιέχει το $W \cup S$, το μέγεθός

της θα είναι $|X_i| \leq 3 \cdot k + 4 + k + 1 \Rightarrow |X_i| \leq 4 \cdot k + 5$, άρα το δενδροπλάτος θα είναι το πολύ $4 \cdot k + 4$. Αν βρισκόμαστε στην βάση, δηλαδή στην πρώτη if τότε ισχύει τετριμμένα.

Τώρα θα δείξουμε ποια είναι η πολυπλοκότητα του αλγορίθμου. Για αρχή θα πρέπει να βρούμε πόσο γρήγορα μπορούμε να υπολογίσουμε τον κορυφοδιαχωριστή S για το W με μέγεθος $3 \cdot k + 4$. Ένας τρόπος είναι να πάρουμε όλους τους συνδυασμούς που το W σπάει σε δύο σύνολα. Για κάθε κορυφή έχουμε δύο επιλογές(τις δύο συνεκτικές συνιστώσες του W) και επειδή το μέγεθος του W είναι $3 \cdot k + 4$, όλοι οι συνδυασμοί είναι $2^{(3 \cdot k + 4)}$. Για κάθε τέτοιον συνδυασμό έχουμε δύο ξένα υποσύνολα W_1 και W_2 όπου η ενωσή τους είναι το W. Για να βρούμε τώρα τον διαχωριστή μεταξύ του W_1 και W_2 θα πρέπει να υπολογίσουμε από θεώρημα Menger όλα τα διακριτά μονοπάτια και αυτό το κάνουμε με κάποιον αλγόριθμο που υπολογίζει το max flow. Θυμίζουμε ότι οι κορυφές που ανήκουν στο $S \cap W$ τις αναδιανέμουμε ξανά όπως στο Λήμμα 2.21 ώστε το μέγεθος των συνιστωσών να είναι μεταξύ των ορίων που ορίζει το λήμμα. Φυσικά κάποιες κορυφές του μονοπατιού μπορεί να μην ανήκουν στο W γιατί μπορεί να υπάρχει μονοπάτι μεταξύ δύο κορυφών που ανήκουν στο W που να περνάει και από κορυφές εκτός του W. Αν το μέγεθος του κορυφοδιαχωριστή είναι το πολύ $k + 1$ και οι δύο συνιστώσες W_1 και W_2 που προκύπτουν ικανοποιούν τις συνθήκες του Λήμματος 2.21, τότε βρήκαμε έναν έγκυρο κορυφοδιαχωριστή και συνεχίζουμε την διαδικασία. Αν όμως για όλους τους συνδυασμούς δεν βρούμε έναν κορυφοδιαχωριστή με μέγεθος το πολύ $k + 1$, τότε από Λήμμα 2.21 συμπεραίνουμε ότι το δενδροπλάτος του G είναι μεγαλύτερο του k και σταματάμε την διαδικασία. Ένας αλγόριθμος max flow(π.χ ο Edmonds–Karp) κάνει αναζήτηση(π.χ κατά πλάτος) σε χρόνο $O(|V| + |E|)$ και είτε συμπεραίνει ότι το flow είναι μέγιστο είτε το μειώνει στην χειρότερη κατά ένα. Επειδή εμείς θέλουμε το μέγεθος του κορυφοδιαχωριστή να μην υπερβαίνει το $k + 1$, τρέχουμε τον αλγόριθμο max flow, $O(k)$ φορές και συνολικά ο αλγόριθμος τρέχει σε $O(k \cdot (|V| + |E|))$. Η υποθεσή μας πάνω όμως λέει ότι $|E| \leq k \cdot n$ και οπότε έχουμε ότι τρέχει σε $O(k^2 \cdot |V|)$. Επειδή όμως έχουμε $2^{(3 \cdot k + 4)}$ συνδυασμούς, συνολικά μία κλήση της συνάρτησης τρέχει σε $O(2^{(3 \cdot k + 4)} \cdot k^2 \cdot |V|)$. Για τον τελικό χρόνο του αλγορίθμου θα πρέπει να υπολογίσουμε πόσες φορές καλείται η συνάρτηση. Στο [8] ο αλγόριθμος που παρουσιάζεται διαχωρίζει τις δύο συνεκτικές συνιστώσες λίγο διαφορετικά από ότι φαίνεται στον από πάνω κώδικα. Σαν W_i δεν βάζει όλο το $(G_i \cap W) \cup S$ αλλά μόνο ένα κομμάτι αυτού και συγκεκριμένα μόνο τις κορυφές που ενώνονται με το Z_i. Αυτό το κάνει ώστε να διασφαλίσει ότι κάθε κορυφή που ανήκει στο W ενώνεται με κάποια κορυφή εκτός του W, κάτι που θα μας φανεί χρήσιμο στην συνέχεια. Επιπλέον τώρα μπορούμε να δούμε και τον λόγο που θέλουμε τα W_i να είναι μεγαλύτερα ή ίσα του $k + 2$ όπως φαίνεται στο Λήμμα 2.21. Επειδή $|W_i| \geq k + 2$ και $|S| \leq k + 1$, υπάρχει κορυφή στο W_i που δεν ανήκει στο S, αλλά επειδή αυτή η κορυφή θα επικοινωνεί με μία άλλη κορυφή εκτός του W, τότε θα υπάρχει μονοπάτι μεταξύ δύο κορυφών των W_1 και W_2 το οποίο δεν θα ανήκει εξ ολοκλήρου στο W. Επομένως ο διαχωριστής S θα πρέπει να περιέχει μία τουλάχιστον κορυφή εκτός του W για να διασπάσει αυτό το μονοπάτι. Αυτό είναι πολύ σημαντικό για τον υπολογισμό της πολυπλοκότητας επειδή σε κάθε κλήση της συνάρτησης η τσάντα που δημιουργείται περιέχει το σύνολο $W \cup S$ και κάθε S περιέχει μία κορυφή που δεν ανήκει στο W. Έτσι σε κάθε τσάντα υπάρχει μία καινούργια κορυφή κάθε φορά και για αυτό το πλήθος των κορυφών της δενδροαποσύνθεσης δεν ξεπερνά το πλήθος των κορυφών του G. Επομένως η συνάρτηση καλείται $O(|V|)$ φορές και η συνολική πολυπλοκότητα είναι

$O(2^{(3\cdot k+4)} \cdot k^2 \cdot |V|^2) = O(8^k \cdot k^2 \cdot |V|^2)$. □

Τώρα που έχουμε δει πως λειτουργεί ο αλγόριθμος μπορούμε να αναλύσουμε τον λόγο που επιλέχθηκε η τιμή $3 \cdot k + 4$ για το μέγεθος του W. Ο αλγόριθμος εισάγει στην τσάντα της τρέχουσας αναδρομής όλο το W. Επομένως εμείς θέλουμε να το φράξουμε για να μπορούμε να υπολογίσουμε το δενδροπλάτος. Με βάση το Λήμμα 2.20 θέλουμε να διαχωρίσουμε το αρχικό γράφημα σε δύο συνιστώσες, επομένως θέλουμε να βρούμε έναν $\frac{2}{3}$-κορυφοδιαχωριστή. Έστω X η μέγιστη τιμή του W. Σε κάθε αναδρομική κλήση ισχύει ότι $|W_i| \leq \frac{2}{3} \cdot |W| + k + 1$, επειδή $|S| = k+1$. Επειδή το W_i και το W είναι το ίδιο σύνολο που θέλουμε να φράξουμε, υποθέτουμε ότι $|W_i| = |W|$ και μετατρέπουμε την ανισότητα σε ισότητα. Τελικά σε αυτό που καταλήγουμε είναι ότι $|W| = 3 \cdot k + 3$. Αν είχαμε αυτήν την τιμή για το μέγεθος του W, τότε όταν εφαρμόζουμε τον κορυφοδιαχωριστή S στο W, η μεγαλύτερη συνιστώσα του μπορεί να έχει μέγεθος ίσο με $|W_1| = \frac{2 \cdot |W|}{3} = 2 \cdot k + 2$. Τότε όμως η μικρότερη συνιστώσα του θα έχει μέγεθος ίσο με $|W_2| = |W| - |W_1| = 3 \cdot k + 3 - 2 \cdot k - 2 = k + 1$. Σε αυτήν την περίπτωση όμως δεν θα μπορούσαμε να χρησιμοποιήσουμε το επιχείρημα ότι ο κορυφοδιαχωριστής περιέχει τουλάχιστον μία κορυφή εκτός του W γιατί θα μπορούσε να περιέχει όλο το W_2. Αυτό το επιχείρημα όμως μας βοήθησε στην ανάλυση της πολυπλοκότητας, επομένως αυξάνοντας το μέγεθος του W κατά ένα λύνει το προβλημά μας και τελικά αυτός είναι ο λόγος που $|W| = 3 \cdot k + 4$. Από την στιγμή που βρίσκουμε την μέγιστη τιμή του W το δενδροπλάτος το υπολογίζουμε εύκολα όπως δείχθηκε και προηγουμένως.

3.4 Περισσότεροι αλγόριθμοι για το δενδροπλάτος

Μέχρι τώρα παρουσιάστηκαν αναλυτικά δύο αλγόριθμοι για το δενδροπλάτος. Και οι δύο αλγόριθμοι ήταν προσεγγιστικοί και γνωρίζαμε κατά πόσο η λύση που δίνουν διαφέρει στην χειρότερη περίπτωση από την πραγματική λύση. Και οι δύο εκμεταλλευτήκαν τις ιδιότητες των κορυφοδιαχωριστών και το πως σχετίζονται με το δενδροπλάτος. Παρόλα αυτά το δενδροπλάτος σχετίζεται με πολλές άλλες παραμέτρους, με την χρήση των οποίων μπορούμε να δημιουργήσουμε διαφορετικούς αλγορίθμους. Μια παρουσίαση αυτών γίνεται στο [5].

Πολλοί αλγόριθμοι που έχουν δημιουργηθεί και μελετηθεί που λύνουν το πρόβλημα του δενδροπλάτους εκμεταλλεύονται την σχέση που έχει το δενδροπλάτος με το elimination game. Θυμίζουμε ότι όταν παίξουμε το elimination game σε ένα γράφημα τότε το γράφημα που προκύπτει είναι chordal. Από το Θεώρημα 2.7 ξέρουμε ότι αν το δενδροπλάτος του αρχικού γραφήματος είναι k τότε η μέγιστη κλίκα του chordal γραφήματος που προκύπτει μετά το elimination game είναι ίση ή μεγαλύτερη του $k+1$. Επειδή και το elimination game και η εύρεση της μέγιστης κλίκας(μέσω του perfect elimination ordering) στα chordal γραφήματα γίνεται σε πολυωνυμικό χρόνο, ο αλγόριθμος που παίρνουμε είναι προσεγγιστικός και πολυωνυμικός χωρίς ομως να γνωρίζουμε πόσο καλή προσέγγιση έχουμε. Ο τρόπος που επιλέγουμε τις κορυφές στο elimination game είναι πολύ σημαντικός καθώς από αυτήν την επιλογή εξαρτάται κατά πόσο κοντά στην πραγματική λύση θα είναι η απαντησή μας για την τιμή του δενδροπλάτους. Το σίγουρο είναι ότι υπάρχει τρόπος να επιλέξουμε τις κορυφές με τέτοιον τρόπο ώστε να πάρουμε το chordal γράφημα με την μικρότερη μέγιστη

κλίκα. Αυτό είναι επειδή όπως είπαμε στην ενότητα με το elimination game, κάθε ελαχιστικό chordal completion μπορεί να προκύψει για κάποια διάταξη π. Επομένως το chordal completion με την μικρότερη μέγιστη κλίκα θα είναι ένα από αυτά τα ελαχιστικά. Δεν γνωρίζουμε όμως κάποιον πολυωνυμικό αλγόριθμο που να μας δίνει αυτόν τον τρόπο, γιατί τότε θα είχαμε πολυωνυμικό αλγόριθμο για το πρόβλημα του δενδροπλάτους. Για αυτόν τον λόγο χρησιμοποιούμε διάφορες ευρεστικές στην επιλογή των κορυφών στην διάταξη. Κάποιες απλές ευρεστικές μέθοδοι είναι να επιλέγουμε σε κάθε βήμα είτε την κορυφή με τους λιγότερους γείτονες που δεν έχουν αφαιρεθεί ακόμα από το γράφημα, είτε τη κορυφή που θα προσθέσει τις λιγότερες fill edges. Αυτές οι δύο ευρεστικές διαφέρουν, επειδή μπορεί οι γείτονες μιας κορυφής να ενώνονται μεταξύ τους. Μια άλλη εναλλακτική λύση για την επιλογή των κορυφών είναι να χρησιμοποιηθούν αλγόριθμοι που αναγνωρίζουν αν ένα γράφημα είναι chordal. Το να βρούμε ένα chordal completion με την ελάχιστη δυνατή πρόσθεση ακμών, είπαμε ότι είναι NP-complete πρόβλημα, αλλά αν θέλουμε να βρούμε μια ελαχιστική και όχι την ελάχιστη πρόσθεση ακμών, υπάρχουν πολυωνυμικοί αλγόριθμοι που το πετυχαίνουν. Επομένως μπορούμε να χρησιμοποιήσουμε έναν τέτοιον αλγόριθμο για να πάρουμε ένα chordal completion του αρχικού γραφήματος σε πολυωνυμικό χρόνο και μετά να βρούμε σε πολυωνυμικό χρόνο το δενδροπλάτος του. Σε κάθε περίπτωση, οι αλγόριθμοι αυτής της κατηγορίας δίνουν ένα άνω φράγμα για το δενδροπλάτος.

Κάποιοι άλλοι αλγόριθμοι που έχουν υλοποιηθεί για να βρουν το δενδροπλάτος και την δενδροαποσύνθεση ενός γραφήματος χρησιμοποιούν την μέθοδο branch και bound. Συνήθως σε τέτοιους αλγορίθμους χρειαζόμαστε ένα καλό κάτω φράγμα. Κάποια τετριμμένα κάτω φράγματα είναι το μέγεθος της μέγιστης κλίκας ή ο ελάχιστος βαθμός μιας κορυφής του γραφήματος. Έχει γίνει αρκετή μελέτη και έχουν αναπτυχθεί πολλές μέθοδοι που βρίσκουν καλά κάτω φράγματα. Γνωρίζοντας το κάτω φράγμα του δενδροπλάτους ενός γραφήματος, μπορούμε να αποφασίσουμε ποιος αλγόριθμος ταιριάζει καλύτερα στην περιπτωσή μας.

ΑΝΑΦΟΡΕΣ

[1] Stefan Arnborg, Derek G Corneil, and Andrzej Proskurowski. “Complexity of finding embeddings in ak-tree”. In: *SIAM Journal on Algebraic Discrete Methods* 8.2 (1987), pp. 277–284.

[2] Hans L Bodlaender. “A linear-time algorithm for finding tree-decompositions of small treewidth”. In: *SIAM Journal on computing* 25.6 (1996), pp. 1305–1317.

[3] Hans L Bodlaender. “A partial k-arboretum of graphs with bounded treewidth”. In: *Theoretical computer science* 209.1-2 (1998), pp. 1–45.

[4] Hans L. Bodlaender. “A Tourist Guide through Treewidth”. In: *Acta Cybern.* 11.1-2 (1993), pp. 1–21.

[5] Hans L Bodlaender. “Discovering treewidth”. In: *International Conference on Current Trends in Theory and Practice of Computer Science*. Springer. 2005, pp. 1–16.

[6] Hans L Bodlaender, John R Gilbert, Hjálmtyr Hafsteinsson, and Ton Kloks. “Approximating treewidth, pathwidth, frontsize, and shortest elimination tree”. In: *J. Algorithms* 18.2 (1995), pp. 238–255.

[7] Daniela Bronner and Bernard Ries. *An introduction to treewidth*. Tech. rep. EPFL. 2006.

[8] Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*. Vol. 3. Springer, 2015.

[9] Jirı Fiala. *Graph minors, decompositions and algorithms*. 2014.

[10] Fănică Gavril. “The intersection graphs of subtrees in trees are exactly the chordal graphs”. In: *Journal of Combinatorial Theory, Series B* 16.1 (1974), pp. 47–56.

[11] Daniel J Harvey and David R Wood. “Parameters tied to treewidth”. In: *Journal of Graph Theory* 84.4 (2017), pp. 364–385.

[12] Pinar Heggernes. “Treewidth, partial k-trees, and chordal graphs”. In: *Partial curriculum in INF334-Advanced algorithmical techniques, Department of Informatics, University of Bergen, Norway* (2005).

[13] Jan van Leeuwen. “Graph Algorithms”. In: *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity (A)*. 1990, pp. 525–631.

[14] Tom Leighton and Satish Rao. “An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms”. In: *Foundations of Computer Science, 1988., 29th Annual Symposium on*. IEEE. 1988, pp. 422–431.

[15] Donald J Rose. “Triangulated graphs and the elimination process”. In: *Journal of Mathematical Analysis and Applications* 32.3 (1970), pp. 597–609.

[16] Mihalis Yannakakis. “Computing the minimum fill-in is NP-complete”. In: *SIAM Journal on Algebraic Discrete Methods* 2.1 (1981), pp. 77–79.