Η Τεχνικη της Χρωματικης Κωδικοποιησης στο Σχεδιασμο Παραμετρικων Αλγοριθμων

$$\mu \prod \lambda \, \forall$$

Σδράκας Κωνσταντίνος

Επιβλέπων: Επίκουρος Καθηγητής Δημήτριος Μ. Θηλυκός

Μάρτιος 2010

Περιεχόμενα

Ευχαριστίες

Σε ένα πολύπλοκο κόσμο γεμάτο προβλήματα, πόσο σπουδαίο ρόλο μπορεί να παίξει μια απλή παράμετρος; Η αναζήτηση της απάντησης σε αυτό το ερώτημα ήταν αρκετή για να με βγάλει από τα μονοπάτια της κρυπτογραφίας, που μονοπωλούσε μέχρι τότε το ενδιαφέρον μου, και να με βάλει στο δρόμο της Παραμετρικής Πολυπλοκότητας.

Αυτή η αλλαγή πορείας όμως δε θα γινόταν αν δεν υπήρχε το ανάλογο σεμιναριακό μάθημα του κου Δ. Θηλυκού. Ο απλός και πάντα ουσιώδης τρόπος διδασκαλίας του, η αμεσότητά του και η σε βάθος γνώση του για το αντικείμενο, μού ξύπνησαν το ενδιαφέρον και εν αγνοία τού μού μετέδωσε το πάθος του για την Παραμετρική Πολυπλοκότητα. Πάνω από όλα όμως, τον ευχαριστώ για την υπέροχη συνεργασία μας, την υπομονή του, την καθοδήγηση, τις σημαντικές του παρατηρήσεις καθώς και για την πολύτιμη βοήθειά του, έτσι ώστε η διπλωματική αυτή εργασία να ολοκληρωθεί.

Πριν φτάσω όμως ως εδώ, στα πρώτα χρόνια σπουδών μου στο ΜΠΛΑ ξεχώρισα έναν καθηγητή. Ο κος Ε. Ράπτης, με το ήθος του, την πραότητά του και το ξεχωριστό ενδιαφέρον που δείχνει για τους φοιτητές του, με κάνει να αισθάνομαι τυχερός που υπήρξα μαθητής του.

Επίσης, θα ήθελα να ευχαριστήσω τον κο Κολλιόπουλο. Ο ιδιαίτερος τρόπος διδασκαλίας του, ο ξεχωριστός του χαρακτήρας και το πάθος του να δίνει απαντήσεις σε κάθε ερώτημα έκαναν τις ώρες παρακολούθησης των μαθημάτων του εξαιρετικά ενδιαφέρουσες.

Στο σημείο αυτό οφείλω να ευχαριστήσω όλους τους συντελεστές του ΜΠΛΑ και ιδιαίτερα τους κκ Ι. Μοσχοβάκη και Κ. Δημητρακόπουλο, που πριν λίγα χρόνια έδειξαν εμπιστοσύνη στο πρόσωπό μου και μου έδωσαν αυτή την πολύτιμη ευκαιρία σπουδών.

Πως θα ήταν όμως τα χρόνια του ΜΠΛΑ χωρίς τον Ηλία Ρόκο; Δε θέλω ούτε να το φανταστώ. Το ξέρω από τώρα ότι, όταν μετά από καιρό τα φοιτητικά χρόνια του ΜΠΛΑ αρχίσουν να ξεθωριάζουν στη μνήμη μου, οι αναμνήσεις που έχω με τον Ηλία, τον οποίο μπορώ να αποκαλώ πραγματικό φίλο, θα μείνουν για πάντα. Δεν τον ευχαριστώ μόνο για τη συνεργασία μας, για τις αμέτρητες ώρες διαβάσματος, για τις εποικοδομητικές μας συζητήσεις, για τη βοήθειά του σε αυτή τη διπλωματική και για το κουράγιο που μου έδινε, αλλά κυρίως για ένα σωρό άλλα πράγματα που δε χωρούν και δεν έχει νόημα να παρουσιάσω σε αυτές τις γραμμές. Αυτός ξέρει.

Τέλος, θέλω να ευχαριστήσω μέσα από την καρδιά μου την αγαπημένη μου αδελφή Ελένη. Η πίστη της σε εμένα και η ανοχή της σε δύσκολους καιρούς, όταν σκεφτόμουν διάφορα, έπαιξαν καθοριστικό ρόλο στις αποφάσεις μου για το μέλλον. Η πολύτιμη παρέα της έκανε λίγο πιο δυνατό το φως στην άκρη του τούνελ.

Όμως, το ότι μπορώ αυτή τη στιγμή να γράφω αυτές εδώ τις γραμμές και να νιώθω λίγο περήφανος για αυτά που έχω καταφέρει ως τώρα το οφείλω, πρώτα και πάνω από όλα, σε δύο συγκεκριμένους ανθρώπους. Δε θα σταθώ στην ανυπολόγιστη οικονομική τους υποστήριξη όλα αυτά τα χρόνια, ούτε καν στην ατελείωτη ηθική υποστήριξη που μου έδιναν απλόχερα. Θα σταθώ στο γεγονός ότι ήταν πάντα δίπλα μου όταν τους χρειαζόμουνα και με αμέριστη υπομονή, υπέρμετρη αγάπη, ενδιαφέρον, φροντίδα και υπευθυνότητα έφεραν εις πέρας με επιτυχία το δυσκολότερο έργο από όλους.

Πατέρα-Μητέρα, ένα απλό ***ευχαριστώ*** *φαντάζει τόσο λίγο μπροστά σε όλα αυτά που μου έχετε προσφέρει.*

Αθήνα, Μάρτιος 2010

Κεφάλαιο 1

Εισαγωγή

Η παραμετρική πολυπλοκότητα είναι μια σχετικά νέα προσέγγιση πάνω στη γενική θεωρία της κλασσικής υπολογιστικής πολυπλοκότητας. Αυτό που κάνει είναι να εξετάζει την πολυπλοκότητα ενός προβλήματος σε σχέση με τις παραμέτρους που υπάρχουν στην είσοδό του. Με τον τρόπο αυτό, πετυχαίνει να δίνει απαντήσεις εκεί που δεν υπήρχαν και να βελτιώνει πολλά μέχρι τώρα γνωστά αποτελέσματα.

Σαν νέος κλάδος στο χώρο της θεωρητικής πληροφορικής είναι αναμενόμενο να είναι ταχύτατα αναπτυσσόμενος. Τα τελευταία χρόνια έχει παρουσιαστεί ένας μεγάλος αριθμός τεχνικών σχεδιασμού παραμετρικών αλγορίθμων. Μία από αυτές είναι και η τεχνική της χρωματικής κωδικοποίησης (color coding) που ανακαλύφθηκε από τους Alon, Yuster και Zwick [2]. Επιγραμματικά, στην τεχνική αυτή οι κόμβοι ενός γραφήματος χρωματίζονται με κάποιον αριθμό χρωμάτων και το εκάστοτε πρόβλημα αντιμετωπίζεται δουλεύοντας με σύνολα χρωμάτων και μαθηματικές συναρτήσεις.

Όλα ξεκίνησαν όταν οι Παπαδημητρίου και Γιαννακάκης είχαν θέσει το ερώτημα για το αν μπορεί να βρεθεί σε ένα γράφημα $G = (V, E)$ μονοπάτι μήκους $\log n$ σε πολυωνυμικό χρόνο [35] . Μέχρι το 1995 το ερώτημα αυτό παρέμενε αναπάντητο.

Στο σημείο αυτό ήρθε η δημοσίευση των Alon, Yuster και Zwick [2] με την πρωτοποριακή ιδέα της τεχνικής της χρωματική κωδικοποίησης. Όχι μόνο έλυσε το παραπάνω πρόβλημα, αλλά μαζί με αυτό δόθηκαν απαντήσεις και σε έναν αριθμό ερωτήσεων-προβλημάτων που ακολουθούσαν. Χρησιμοποιεί δυναμικό προγραμματισμό κι εκμεταλλεύεται κάποιες ειδικού τύπου συναρτήσεις κατακερματισμού.

Αναφορικά με αυτή την εργασία, στο κεφάλαιο 2 κάνουμε μια μικρή εισαγωγή στην παραμετρική πολυπλοκότητα και παρουσιάζουμε την κλάση FPT όπου ανήκουν τα προβλήματα που επιδέχονται παραμετρικούς αλγορίθμους. Στο κεφάλαιο 3 περιγράφουμε αναλυτικά τα αποτελέσματα του Alon . Ξεκινάμε δίνοντας τον αλγόριθμο για την εύρεση απλών μονοπατιών μεγέθους $k - 1$ σε ένα γράφημα. Στην παράγραφο αυτή παρουσιάζονται και οι k-τέλειες οικογένειες συναρτήσεων κατακερματισμού, που αποτελούν κλειδί στην απόδειξη ορθότητας του αλγορίθμου. Έπειτα από το μονοπάτι μήκους $k - 1$ δίνουμε τον αλγόριθμο για την εύρεση δέντρου k κόμβων σε ένα γράφημα. Έχοντας αναλύσει τα παραπάνω στη δημοσίευση των Alon et al υπάρχει ένα θεώρημα που υποστηρίζει ότι, ακολουθώντας την ίδια λογική, αποδεικνύεται πως σε ένα γράφημα μπορεί να βρεθεί ένα άλλο οποιοδήποτε γράφημα με φραγμένο δεντροπλάτος, μη δίνοντας όμως αυτή την απόδειξη. Η σημαντική λοιπόν δουλειά σε αυτή τη διπλωματική γίνεται στην παράγραφο 3.3 όπου παρουσιάζεται, από όσο γνωρίζει ο γράφων, για πρώτη φορά αναλυτικά αυτή η απόδειξη και δίνεται με λεπτομέρια η αναδρομή που χρησιμοποιείται.

Στο κεφάλαιο 4 δίνουμε κάποιες πιθανοτικές προσεγγίσεις των προηγούμενων προβλημάτων και στο κεφάλαιο 5 παρουσιάζουμε μια νέα τεχνική που είναι συνδυασμός της χρωματικής κωδικοποίησης και του διαίρει και βασίλευε και δίνει κάποια σημαντικά αποτελέσματα. Στο κεφάλαιο 6 παρουσιάζονται κάποιοι βελτιωμένοι αλγόριθμοι για προβλήματα μονοπατιού και το σπουδαίο σε αυτό το σημείο είναι ότι παρουσιάζονται για πρώτη φορά κάποια κατώτερα όρια. Τέλος, κλείνουμε με το κεφάλαιο 7 όπου γίνεται μια γενική συζήτηση και παρουσιάζονται κάποια συμπεράσματα και ανοιχτά προβλήματα για κάθε ενδιαφερόμενο.

Στο σημείο αυτό να επισημάνουμε ότι στα παραδείγματα που έχουν επιλεγεί να παρουσιαστούν σε κάθε περίπτωση έχει γίνει μια προσπάθεια να είναι όσο το δυνατόν πιο απλά γίνεται. Αυτό έχει συμβεί γιατί θέλουμε να εστιάσουμε στην ουσία της ιδέας της τεχνικής. Επιπλέον, για κάθε βασική τεχνική δίνουμε τα βασικά βήματα, το σκελετό του αλγορίθμου, σε ψευδοκώδικα με σκοπό να γίνει πιο κατανοητό το πώς δουλεύει ο αλγόριθμος πίσω από την εκάστοτε τεχνική. Όπου είναι εφικτό παραθέτουμε επιπρόσθετα και τις αποδείξεις ορθότητας αυτών των αλγορίθμων.

Κλείνονταν την εισαγωγή είναι σημαντικό να αναφέρουμε ότι η τεχνική της χρωματικής κωδικοποίησης, ούσα μια νέα τεχνική που δίνει ή βελτιώνει αποτελέσματα, είναι φυσικό να έχει χρησιμοποιηθεί σε μια ποικιλία προβλημάτων. Οι εφαρμογές της είναι πολυάριθμες και αφορούν κυρίως στον κόσμο της γραφοθεωρίας. Κάποιες από αυτές παρουσιάζονται στα επόμενα κεφάλαια, όπως μπορεί να παρατηρήσει κανείς από την παραπάνω επισκόπηση της εργασίας, ενώ πολλές από αυτές καθώς και η σημαντικότητά τους αναφέρονται στον επίλογο.

Κεφάλαιο 2

Παραμετρική Πολυπλοκότητα

2.1 Εισαγωγή

Η παραμετρική πολυπλοκότητα είναι ένας κλάδος της θεωρίας της υπολογιστικής πολυπλοκότητας στη θεωρητική πληροφορική. Επικεντώνεται στη λύση κλασσικών υπολογιστικών προβλημάτων σύμφωνα με τη δυσκολία που αποκτούν σε σχέση με τις πολλαπλές παραμέτρους που υπάρχουν στην είσοδο του προβλήματος. Λίγο πιο συγκεκριμένα, στην παραμετρική πολυπλοκότητα, η πολυπλοκότητα ενός προβλήματος μετράται ως μια συνάρτηση του μεγέθους της εισόδου και μιας ή περισσοτέρων παραμέτρων σε αυτή. Με τον τρόπο αυτό η παραμετρική πολυπλοκότητα πετυχαίνει παραμετροποιήσεις πολλών NP-δύσκολων (NP-hard) προβλημάτων και δίνει ένα πιο ακριβές τοπίο σχετικά με την πρακτική δυσκολία πολλών προβλημάτων. Στο σημείο αυτό να σημειώσουμε ότι από τις πρώτες θεμελιώσεις της παραμετρικής πολυπλοκότητας έχουν γίνει από τους Downey και Fellows το 1999 [10] καθώς και με τις δουλειές των Grohe και Flum [21] και του Niedermeier [33].

Γνωρίζουμε σήμερα ότι η ύπαρξη ενός αποδοτικού, ακριβούς και ντετερμινιστικού αλγορίθμου που να λύνει NP-πλήρη (NP-complete) ή αλλιώς NP-δύσκολα προβλήματα είναι μη αναμενόμενη. Όλοι οι γνωστοί αλγόριθμοι που λύνουν τέτοια προβλήματα απαιτούν χρόνο που είναι εκθετικός ως προς το συνολικό μέγεθος της εισόδου. Παρόλα αυτά, υπάρχουν κάποια προβλήματα που μπορούν να λυθούν από αλγόριθμους που είναι υπερπολυωνυμικοί κατά τρόπο ώστε το μη πολυωνυμικό κομμάτι της πολυπλοκότητάς τους να αφορά αποκλειστικά σε κάποια παράμετρο της εισόδου. Οι αλγόριθμοι αυτοί ονομάζονται παραμετρικά αποδοτικοί (fixed-parameter tractable ή (fpt)-αλγόριθμοι), γιατί το πρόβλημα μπορεί να λυθεί ικανοποιητικά για μικρές τιμές της παραμέτρου.

Στη συνέχεια θα δώσουμε ένα παράδειγμα και θα κλείσουμε με τους ορισμούς των παραπάνω εννοιών.

Υπάρχει ένας μεγάλος αριθμός προβλημάτων που έχει την ακόλουθη μορφή. ''Δοσμένου ενός αντικειμένου x κι ενός μη αρνητικού ακεραίου k, έχει το x κάποια ιδιότητα που να εξαρτάται από το k;''. Χαρακτηριστικό παράδειγμα είναι το πρόβλημα της κάλυψης κορυφών (Vertex Cover) όπου η είσοδος είναι γράφημα και η παράμετρος είναι ο αριθμός των κορυφών στο ζητούμενο κάλυμμα. Είναι εύκολα κατανοητό, ότι σε πολλές πρακτικές εφαρμογές κάποιος μπορεί να υποθέσει ότι η παράμετρος είναι μικρή σε σύγκριση με το συνολικό μέγεθος της εισόδου, που στο συγκεκριμένο παράδειγμα είναι το γράφημα.

Ακριβώς σε αυτό το σημείο εισέρχεται η παραμετρική πολυπλοκότητα και οι ορισμοί που παρουσιάζονται στη συνέχεια.

Ορισμός 2.1 *Παραμετρικό πρόβλημα είναι ένα διατεταγμένο ζευγάρι* $(x, k) \in L$ *όπου* L *είναι μια γλώσσα με* $L \subseteq \Sigma^* \times \mathbb{N}$. *Έχουμε ότι* $x \in \Sigma^*$ *όπου* Σ *είναι ένα πεπερασμένο αλφάβητο και* $k \in \mathbb{N}$ *η παράμετρος του προβλήματος.*

Ορισμός 2.2 *Ένα παραμετρικό πρόβλημα είναι παραμετρικά αποδοτικό (fixed-parameter tractable) αν η ερώτηση "*$(x, k) \in L$*;" μπορεί να απαντηθεί σε χρόνο* $f(k) \cdot |x|^{O(1)}$, *όπου* f *είναι μια συνάρτηση που εξαρτάται μόνο από το* k. *Η αντίστοιχη κλάση πολυπλοκότητας είναι η* FPT .

2.2 Η κλάση FPT

Πριν προχωρήσουμε, είναι σκόπιμο να αναφέρουμε λίγια λόγια για την κλάση FPT. Η κλάση αυτή περιέχει όλα τα παραμετρικά αποδοτικά προβλήματα, δηλαδή, σε συνάρτηση με την προηγούμενη παράγραφο, αυτά που μπορούν να λυθούν σε χρόνο $f(k) \cdot |x|^{O(1)}$ με f μια υπολογίσιμη συνάρτηση που εξαρτάται μόνο από το k και $|x|$ το μέγεθος της εισόδου του προβλήματος. Τυπικά, η συνάρτηση αυτή είναι κατά προτίμηση μια απλή εκθετική συνάρτηση, όπως για παράδειγμα η $2^{O(k)}$, αλλά στην ουσία ο ορισμός επιτρέπει και συναρτήσεις αρκετά πιο περίπλοκες (με μεγαλύτερη αύξηση). Αυτό ήταν πολύ ουσιαστικό τουλάχιστον τα πρώτα χρόνια ύπαρξης αυτής της κλάσης. Το σημαντικό σημείο του ορισμού είναι ότι αποκλείει συναρτήσεις της μορφής $f(n, k)$, όπως για παράδειγμα η n^k.

Ένα χαρακτηριστικό παράδειγμα αυτής της κλάσης είναι το ευρέως γνωστό πρόβλημα της ικανοποιησιμότητας μιας φόρμουλας ϕ όταν παραμετροποιούμε με βάση τον αριθμό των μεταβλητών της φόρμουλας. Είναι εύκολο να διαπιστώσει κανείς ότι μια φόρμουλα ϕ μεγέθους m με k μεταβλητές μπορεί να ελεγχθεί αν ικανοποιείται σε χρόνο $O(2^k \cdot m)$. Έτσι λοιπόν το παραμετροποιημένο αυτό πρόβλημα ανήκει στην FPT. Άλλο παράδειγμα, στο οποίο αναφερθήκαμε και στην Εισαγωγή, είναι το κάλυμμα κορυφών. Ένα κάλυμμα κορυφών μεγέθους k σε ένα γράφημα μεγέθους m μπορεί να βρεθεί σε χρόνο $O(2^k \cdot m)$, οπότε κι αυτό το πρόβλημα ανήκει στην κλάση FPT.

Σύμφωνα με τα παραπάνω θα μπορούσε να σκεφτεί κανείς ότι όλα τα γνωστά προβλήματα, όταν παραμετροποιούνται κατάλληλα, θα μπορούσαν να ανήκουν σε αυτή την κλάση. Κάτι τέτοιο όμως δεν ισχύει. Χαρακτηριστικό παράδειγμα αυτής της περίπτωσης είναι το πρόβλημα χρωματισμού γραφημάτων με k χρώματα (k-χρωματισμός) όταν η παραμετροποίηση γίνεται στον αριθμό των χρωμάτων που χρησιμοποιούνται για το χρωματισμό του γραφήματος. Γνωρίζουμε ότι το 3-χρωματισμός είναι NP-δύσκολο. Έτσι ένας αλγόριθμος που θα χρωματίζει ένα γράφημα με k χρώματα και θα τρέχει σε χρόνο $f(k) \cdot |x|^{O(1)}$, για $k = 3$, θα έτρεχε σε χρόνο πολυωνυμικό ως προς το μέγεθος της εισόδου. Οπότε, αν ο χρωματισμός γραφημάτων ήταν στο FPT τότε θα ίσχυε P = NP, κάτι που θεωρείται (ευρέως) απίθανο.

Γενικότερα υπάρχουν κι άλλοι εναλλακτικοί ορισμοί για την κλάση FPT. Για παράδειγμα, ο απαιτούμενος χρόνος τερματισμού $f(k) \cdot |x|^{O(1)}$ μπορεί να αντικατασταθεί με $f(k) + |x|^{O(1)}$. Επίσης, ένα παραμετρικό πρόβλημα ανήκει στην κλάση

FPT αν έχει έναν λεγόμενο πυρήνα. Η τεχνική που χρησιμοποιείται για αυτόν τον έλεγχο ονομάζεται πυρηνοποίηση. Η πυρηνοποίηση ουσιαστικά είναι μια προεπεξεργαστική τεχνική η οποία ανάγει το αρχικό στιγμιότυπο του προβλήματος σε ένα πολύ μικρότερο στιγμιότυπό του, σε έναν ''πυρήνα'' του, που είναι ισοδύναμος με το αρχική είσοδο αλλά με μέγεθος πολυωνυμικό ως προς την παράμετρο [18]. Η τεχνική αυτή παρουσιάζει εξαιρετικό ενδιαφέρον αλλά δε θα μας απασχολήσει περαιτέρω σε αυτή την εργασία.

Προφανώς η κλάση FPT περιλαμβάνει όλα τα προβλήματα που υπολογίζονται σε πολυωνυμικό χρόνο. Επιπλέον περιλαμβάνει όλα τα προβλήματα βελτιστοποίησης στο NP , που επιτρέπουν ένα πλήρως πολυωνυμικού χρόνου προσεγγιστικό σχήμα.

Κλείνοντας αυτή τη σύντομη περιγραφή για την κλάση FPT θα αναφέρουμε δύο, τρόπον τεινά διαισθητικά, επιχειρήματα για να δώσουμε μια γενική ιδέα της δύναμης και της ομορφιάς αυτής της κλάσης.

1. Η FPT είναι μια εύρωστη κλάση, που δεν εξαρτάται από κάποιο συγκεκριμένο μοντέλο μηχανής και έχει τις απαραίτητες ιδιότητες κλειστότητας.

2. Τα περισσότερα προβλήματα στην FPT έχουν συνήθως χαμηλή εξάρτηση ως προς την παράμετρο.

Κεφάλαιο 3

Χρωματική Κωδικοποίηση

3.1 Εύρεση απλού μονοπατιού μήκους $k-1$

Ο αλγόριθμος Χρωματικής Κωδικοποίησης (Color-Coding) που θα περιγραφεί στη συνέχεια, χρησιμοποιείται για την εύρεση απλών μονοπατιών μεγέθους $k-1$ σε γραφήματα. Πριν όμως περάσουμε στην περιγραφή του αλγορίθμου, είναι χρήσιμο, για την καλύτερη κατανόηση, να αναφέρουμε τους παρακάτω ορισμούς.

Ορισμός 3.1 *Απλό μονοπάτι σε ένα γράφημα είναι μια διαδρομή στην οποία δεν επαναλαμβάνεται κανένας κόμβος.*

Ορισμός 3.2 *Χρωματισμός c ενός γραφήματος είναι μια ανάθεση χρωμάτων στους κόμβους του γραφήματος σύμφωνα με τη συνάρτηση $c\colon V \to \{1,\ldots,k\}$. Όπου V είναι το σύνολο των κόμβων του γραφήματος και $\{1,\ldots,k\}$ είναι το σύνολο των k διαφορετικών χρωμάτων που χρησιμοποιούμε.*

Ορισμός 3.3 *Πανχρωματικό μονοπάτι σε ένα γράφημα, είναι το μονοπάτι του οποίου οι κόμβοι είναι χρωματισμένοι με διαφορετικά χρώματα ο καθένας.*

Έχοντας λοιπόν τους παραπάνω ορισμούς μπορούμε να περιγράψουμε τι ακριβώς θέλουμε να κάνει ο αλγόριθμος. Έστω ένα κατευθυνόμενο ή μη-κατευθυνόμενο γράφημα $G = (V, E)$. Σε κάθε γράφημα G που θα αναφερόμαστε από εδώ και στο εξής, θα συμβολίζουμε με n τον αριθμό των κόμβων του και με m τον αριθμό των ακμών του (δηλαδή $|V| = n$ και $|E| = m$). Αυτό που θέλουμε είναι να βρούμε ένα απλό κατευθυνόμενο ή μη-κατευθυνόμενο μονοπάτι μεγέθους $k-1$ στο G. Διαλέγουμε ένα τυχαίο χρωματισμό των κόμβων του G με k χρώματα. Προφανώς ένα πανχρωματικό μονοπάτι στο G είναι και απλό. Επίσης κάθε απλό μονοπάτι μεγέθους $k-1$ στο G έχει πιθανότητα $k!/k^k > e^{-k}$ να είναι πανχρωματικό. (Εφόσον το μονοπάτι έχει μήκος $k-1$, θα αποτελείται από k κόμβους. Έτσι λοιπόν για κάθε τέτοιο μονοπάτι έχουμε $k!$ πιθανούς πανχρωματικούς συνδυασμούς και k^k πιθανούς χρωματισμούς. Για αυτό και η παραπάνω πιθανότητα). Να σημειώσουμε ότι η πιθανότητα αυτή είναι εκθετικά μικρή σε σχέση με το k.

Σύμφωνα με τα παραπάνω προκύπτουν τα εξής ερωτήματα. Πόσος χρόνος χρειάζεται για να βρεθεί ένα πανχρωματικό μονοπάτι μήκους $k-1$ στο G, αν αυτό υπάρχει, ή πόσος χρόνος χρειάζεται για να βρεθούν όλα τα ζευγάρια των κόμβων που συνδέονται με πανχρωματικά μονοπάτια μήκους $k-1$ στο G; Τα λήμματα που θα παρουσιαστούν στη συνέχεια δίνουν απαντήσεις σε αυτά τα ερωτήματα.

Λήμμα 3.1 *Έστω ότι $G = (V, E)$ είναι ένα κατευθυνόμενο ή μη-κατευθυνόμενο γράφημα κι έστω ότι $c\colon V \to \{1, \ldots, k\}$ είναι ένας χρωματισμός των κόμβων του με k χρώματα. Ένα πανχρωματικό μονοπάτι μεγέθους $k-1$ στο G, αν αυτό υπάρχει, μπορεί να βρεθεί στη χειρότερη περίπτωση σε χρόνο $2^{O(k)} \cdot n \cdot m$.*

Απόδειξη. Αρχικά περιγράφουμε έναν αλγόριθμο που τρέχει σε χρόνο $2^{O(k)} \cdot m$ και ο οποίος έχει σαν είσοδο ένα γράφημα G, ένα χρωματισμό $c\colon V \to \{1, \ldots, k\}$ και ένα κόμβο $s \in V$ και βρίσκει ένα πανχρωματικό μονοπάτι μήκους $k-1$ που ξεκινάει από τον s, αν αυτό υπάρχει. Ο αλγόριθμος λοιπόν έχει σαν είσοδο ένα χρωματισμένο με k χρώματα γράφημα και σαν έξοδο μια απάντηση ΝΑΙ ή ΟΧΙ για το αν περιέχει ένα πανχρωματικό μονοπάτι μήκους $k-1$ που να αρχίζει από τον s.

Για να το πετύχουμε αυτό χρησιμοποιούμε την αναδρομική σχέση που θα περιγράψουμε στη συνέχεια. Έστω λοιπόν ο χρωματισμός $c\colon V \to \{1, \ldots, k\}$ των κόμβων του γραφήματος κι έστω $s \in V$ ένας κόμβος έναρξης. Για κάθε $v \in V$ και κάθε $1 \leq i \leq k$ ορίζουμε το $\mathcal{A}_{s,i}(v)$ ως τη συλλογή χρωμάτων όλων των (δυνατών) πανχρωματικών μονοπατιών μήκους $i-1$ που ξεκινούν από τον κόμβο έναρξης s και τελειώνουν στον v. Η βάση της αναδρομής που θα περιγράψουμε είναι :

$$\mathcal{A}_{s,1}(v) = \begin{cases} \{c(s)\} & \text{αν } v = s \\ \emptyset & \text{αλλιώς} \end{cases} \tag{3.1}$$

ενώ το βήμα της αναδρομής είναι :

$$\begin{aligned} \mathcal{A}_{s,i+1}(v) = \{C \subseteq \{1, \ldots, k\} \mid |C| = i+1 \\ \& \ \exists u \in N_G(v) \\ \& \ C - c(u) \in \mathcal{A}_{s,i}(u)\} \end{aligned} \tag{3.2}$$

και τελικά αυτό που ρωτάμε είναι αν για κάποια $s, v \in V$ ισχύει:

$$\mathcal{A}_{s,k}(v) \neq \emptyset \tag{3.3}$$

Βλέπουμε ότι ο τρόπος προσέγγισης για την εύρεση ενός πανχρωματικού μονοπατιού μήκους $k-1$, που ξεκινάει από κάποιο συγκεκριμένο κόμβο s, είναι ο δυναμικός προγραμματισμός. Υποθέτουμε ότι για κάθε κόμβο $v \in V$ έχουμε ήδη βρει όλα τα πιθανά σύνολα χρωμάτων που περιέχονται σε πανχρωματικά μονοπάτια μήκους i που ενώνουν το s με τα v. Η σημαντική παρατήρηση, στο σημείο αυτό, είναι ότι δεν καταγράφουμε όλα τα πανχρωματικά μονοπάτια που ενώνουν το s με τα v, αλλά καταγράφουμε μόνο τα σύνολα χρωμάτων που εμφανίζονται σε αυτά τα μονοπάτια. Έτσι λοιπόν, για κάθε κόμβο v έχουμε μια συλλογή με το πολύ $\binom{k}{i}$ σύνολα χρωμάτων. Στη συνέχεια ελέγχουμε κάθε υποσύνολο C που ανήκει στη συλλογή του v και κάθε ακμή $(v, u) \in E$. Αν το $c(u) \notin C$, προσθέτουμε το σύνολο $C \cup \{c(u)\}$ στη συλλογή του u που αντιστοιχεί στα πανχρωματικά μονοπάτια μήκους $i+1$.

Από όλα τα παραπάνω καταλαβαίνουμε ότι το γράφημα G περιέχει ένα πανχρωματικό μονοπάτι μήκους $k-1$ σε σχέση με το χρωματισμό c αν και μόνο αν η τελική συλλογή, που αντιστοιχεί σε μονοπάτια μήκους $k-1$, ενός τουλάχιστον κόμβου είναι μη κενή. Από την αναδρομική σχέση που περιγράφεται από τις εξισώσεις (3.1) και (3.2) είναι εύκολο να διαπιστώσουμε ότι από τη στιγμή που το γράφημα έχει ένα πανχρωματικό μονοπάτι μήκους $k-1$ τότε για αυτό το μονοπάτι (που έστω ότι αρχίζει από τον κόμβο s και τελειώνει στον v) η συλλογή του $\mathcal{A}_{s,k}(v)$ θα είναι μη κενή. Από την άλλη μεριά, αν η συλλογή που αντιστοιχεί σε μονοπάτια μήκους $k-1$ ενός τουλάχιστον κόμβου είναι μη κενή, τότε ξέρουμε σίγουρα ότι υπάρχει τουλάχιστον ένα C σε αυτή τη συλλογή που περιέχει k διαφορετικά χρώματα. Οι κόμβοι αυτοί (που γνωρίζουμε ότι σχηματίζουν μονοπάτι από τον τρόπο που εισέρχονται στο C) είναι που σχηματίζουν το ζητούμενο μονοπάτι.

Ο αριθμός των διεργασιών που εκτελούνται από τον αλγόριθμο που έχει περιγραφεί παραπάνω είναι το πολύ $O(\sum_{i=0}^{k} i\binom{k}{i} \cdot |E|)$ το οποίο είναι της τάξης του $O(k2^k \cdot m)$. Οι παράγοντες του αθροίσματος στην πολυπλοκότητα προκύπτουν ως εξής. Όπως έχουμε ήδη αναφέρει $\binom{k}{i}$ είναι ό αριθμός των συνόλων χρωμάτων που έχει η συλλογή κάθε κόμβου v που εξετάζουμε. Το i προκύπτει από τη σύγκριση των συνόλων που κάνουμε στον τελευταίο έλεγχο του βήματος της αναδρομής στην εξίσωση (3.2) και τέλος το m προκύπτει από το γεγονός ότι κάθε φορά ελέγχουμε τη γειτονιά του υπό εξέταση κόμβου v για να δούμε αν υπάρχει κόμβος του οποίου το χρώμα δεν ανήκει στη συλλογή C.

Κλείνουμε την απόδειξη σημειώνοντας ότι την πολυπλοκόητα $2^{O(k)} \cdot n \cdot m$ την παίρνουμε πολύ απλά τρέχοντας τον παραπάνω αλγόριθμο n φορές, μία για κάθε εναρκτήριο κόμβο. □

Η μορφή του αλγορίθμου που περιγράφεται στο Λήμμα 3.1 είναι αυτή που ακολουθεί.

Algorithm 1 Εύρεση πανχρωματικού μονοπατιού μήκους $k-1$

Είσοδος: Ένα γράφημα $G=(V,E)$, ένας ακέραιος k και χρωματισμός c.
Έξοδος: Απάντηση ΝΑΙ ή ΟΧΙ.

```
for s ∈ V do
    for i = 1, ..., k do
        for v ∈ V do
            Υπολόγισε την A_{s,i}(v)
            if A_{s,k}(v) ≠ ∅ then
                Απάντησε ΝΑΙ
            end if
        end for
    end for
end for
Απάντησε ΟΧΙ
```

Ο αλγόριθμος που έχουμε περιγράψει βρίσκει ένα πανχρωματικό μονοπάτι μήκους $k-1$ σύμφωνα με το χρωματισμό $c\colon V \to \{1,\ldots,k\}$ που του έχει δοθεί σαν είσοδος. Προκύπτει όμως ένα φυσικό ερώτημα. Αν ένα γράφημα έχει ένα μονοπάτι μήκους $k-1$, με ποιο τρόπο εξασφαλίζουμε ότι οι κόμβοι από τους οποίους αποτελείται θα χρωματιστούν όλοι με διαφορετικό χρώμα μεταξύ τους από το χρωματισμό c, έτσι ώστε να ανιχνευθεί τελικά από τον αλγόριθμο; Το γεγονός αυτό μας το εξασφαλίζει μια k-τέλεια οικογένοια συναρτήσεων κατακερματισμού (k-perfect family of hash functions).

Ορισμός 3.4 *Μια k-τέλεια οικογένοια συναρτήσεων κατακερματισμού είναι μια οικογένεια $\mathcal{F}$ συναρτήσεων από $\{1,\ldots,n\}$ επί $\{1,\ldots,k\}$ τέτοια ώστε για κάθε $S \subset \{1,\ldots,n\}$ με $|S|=k$ υπάρχει μια $f \in \mathcal{F}$ η οποία είναι και 1-1 όταν περιορίζεται στο S.*

Χρησιμοποιώντας συναρτήσεις από μια τέτοια k-τέλεια οικογένοια συναρτήσεων κατακερματισμού για το χρωματισμό του γραφήματος εξασφαλίζουμε ότι αν υπάρχει μονοπάτι μήκους $k-1$ τότε αυτό σίγουρα θα γίνει πανχρωματικό από κάποια συνάρτηση (χρωματισμό) της οικογένειας. Στην περίπτωσή μας η επιθυμητή οικογένεια είναι αυτή που παίρνουμε από την κατασκευή των Schmidt και Siegal [40] οι οποίοι στηρίχτηκαν στα βήματα των Fredman, Komlos και Szemeredi [22] .

Αν ονομάσουμε όλη την παραπάνω διαδικασία εύρεσης πανχρωματικού μονοπατιού μήκους $k-1$ σύμφωνα με το χρωματισμό $c\colon V \to \{1,\ldots,k\}$ ως $FindColouredPath(G,k,c)$ τότε ένας αλγόριθμος εύρεσης απλού μονοπατιού μήκους $k-1$ που χρησιμοποιεί μια k-τέλεια οικογένοια συναρτήσεων κατακερματισμού $\mathcal{F}$ για το χρωματισμό του γραφήματος, θα είχε την παρακάτω μορφή.

Algorithm 2 Εύρεση απλού μονοπατιού μήκους $k-1$

Είσοδος: Ένα γράφημα $G=(V,E)$ και ένας ακέραιος k.
Έξοδος: Απάντηση ΝΑΙ ή ΟΧΙ.

```
for f ∈ F do
    Χρησιμοποίησε την f για να χρωματίσεις το γράφημα.
    if FindColouredPath(G, k, f) = NAI then
        Απάντησε ΝΑΙ
    end if
end for
Απάντησε ΟΧΙ
```

Θεώρημα 3.1 *Έστω $G=(V,E)$ ένα κατευθυνόμενο ή μη κατευθυνόμενο γράφημα. Ένα μονοπάτι μήκους $k-1$ στο G, αν αυτό υπάρχει, μπορεί να βρεθεί στη χειρότερη περίπτωση σε χρόνο $2^{O(k)} \cdot n \cdot m$.*

Μένει τώρα να δείξουμε με ποιο τρόπο μπορούμε να βρούμε όλα τα ζευγάρια κόμβων που συνδέονται με πανχρωματικά μονοπάτια μήκους $k-1$ στο G.

Λήμμα 3.2 *Έστω $G=(V,E)$ ένα κατευθυνόμενο ή μη κατευθυνόμενο γράφημα κι έστω $c\colon V \to \{1,\ldots,k\}$ ένας χρωματισμός των κόμβων του με k χρώματα. Όλα τα ζευγάρια των κόμβων που συνδέονται με πανχρωματικά μονοπάτια μήκους $k-1$ στο G μπορούν να βρεθούν σε χρόνο $2^{O(k)} \cdot n^{\omega}$ στη χειρότερη περίπτωση.*

Απόδειξη. Για να πάρουμε τον αλγόριθμο που τρέχει σε χρόνο $2^{O(k)} \cdot n^{\omega}$ χρησιμοποιούμε την ακόλουθη αναδρομική προσέγγιση. Αρχικά βρίσκουμε όλες τις διαχωρίσεις του συνόλου των χρωμάτων $\{1,\ldots,k\}$ σε δύο υποσύνολα C_1, C_2. Έστω λοιπόν ότι για το σύνολο $K=\{1,\ldots,k\}$ (για το οποίο προφανώς ισχύει $|K|=k$) έχουμε το παρακάτω σύνολο όλων των πιθανών διαχωρίσεων.

$$\mathcal{P}(K) = \Big\{\{C_1, C_2\} \Big|\ |C_1| = \left\lceil \tfrac{k}{2} \right\rceil \ \&\ |C_2| = \left\lfloor \tfrac{k}{2} \right\rfloor \Big\} \tag{3.4}$$

Υπάρχουν μόνο $\binom{k}{k/2} = |\mathcal{P}(K)| < 2^k$ τέτοιες διαχωρίσεις. Στο σημείο αυτό έχουμε να κάνουμε δύο παρατηρήσεις. Πρώτον οι διαχωρίσεις είναι τόσες γιατί κάθε φορά μας ενδιαφέρει ποιο είναι το C_1 και ποιο είναι το C_2. Δεύτερον για την απλοποίηση του συμβολισμού στη συνέχεια υποθέτουμε ότι το k είναι ζυγός. Αν ήταν μονός τότε τα σύνολα C_1 και C_2 θα είχαν μέγεθος $\lceil k/2 \rceil$ και $\lfloor k/2 \rfloor$ αντίστοιχα όπως φαίνεται και στην εξίσωση (4.1).

Για κάθε τέτοια διαχώριση, έστω ότι V_1 είναι το σύνολο των κόμβων του G που έχουν χρωματιστεί από τα χρώματα που ανήκουν στο C_1 και V_2 το σύνολο των κόμβων του G που έχουν χρωματιστεί από το C_2. Έστω τώρα ότι G_1 και G_2 είναι τα υπογραφήματα του G αναγόμενα από τα V_1 και V_2 αντίστοιχα. Αναδρομικά βρίσκουμε όλα τα ζευγάρια κόμβων στο G_1 και στο G_2 που συνδέονται με πανχρωματικά μονοπάτια μήκους $k/2-1$. Συλλέγουμε αυτή την πληροφορία σε δύο Boolean πίνακες A_1 και A_2. Δηλαδή τα στοιχεία αυτών των πινάκων είναι άσσοι και μηδενικά. Αυτό που μας ενδιαφέρει είναι το πόσοι άσσοι υπάρχουν και σε ποια ακριβώς θέση είναι. Γιατί, αν ένα στοιχείο του πίνακα είναι άσσος, δηλαδή έστω ότι $a_{i,j}=1$, τότε αυτόματα έχουμε την πληροφορία ότι μεταξύ των κόμβων i και j υπάρχει πανχρωματικό μονοπάτι μήκους $k/2-1$.

Έστω τώρα ότι B είναι ένας Boolean πίνακας που περιγράφει τις σχέσεις γειτνίασης ανάμεσα στους κόμβους του V_1 και αυτούς του V_2. Το Boolean γινόμενο $A_1BA_2 = D$ δίνει όλα τα ζευγάρια των κόμβων στο V που ενώνονται με πανχρωματικά μονοπάτια μήκους ακριβώς $k-1$, όπου οι πρώτοι $k/2$ κόμβοι στα μονοπάτια είναι χρωματισμένοι με χρώματα από το C_1 και οι τελευταίοι $k/2$ κόμβοι είναι χρωματισμένοι με χρώματα από το C_2. Τα ζευγάρια αυτά ορίζουν ουσιαστικά κάποιες συντεταγμένες στον πίνακα D. Το στοιχείο με αυτές τις συντεταγμένες έχει την τιμή 1. Κάνοντας OR στους πίνακες D που παίρνουμε για όλες τις πιθανές διαχωρίσεις παίρνουμε το επιθυμητό αποτέλεσμα. Κι αυτό γιατί ανάλογα με τις διαχωρίσεις C_1, C_2 που έχουμε επιλέξει μπορεί σε κάποιον πίνακα D όλα τα στοιχεία του να είναι μηδενικά. Μας αρκεί τουλάχιστον ένα στοιχείο ενός πίνακα D να είναι άσσος.

Είναι εύκολο να διαπιστώσει κανείς ότι η πολυπλοκότητα αυτής της προσέγγισης είναι όντως $2^{O(k)} \cdot n^{\omega}$, αφού ο αριθμός των πολλαπλασιασμών πινάκων που χρησιμοποιήθηκαν, $t(k)$, ικανοποιεί την αναδρομική σχέση $t(k) \leq 2^k \cdot (2 \cdot t(k/2) + n^{\omega})$. Κι αυτό γιατί έχουμε 2^k διαχωρίσεις και για κάθε διαχώριση καλούμε αναδρομικά την ίδια διαδικασία δύο φορές στο μισό μέγεθος και στη συνέχεια έχουμε και το κόστος του πολλαπλασιασμού για να βρούμε τον πίνακα D. $\square$

Αν λοιπόν ονομάσουμε τον παραπάνω αλγόριθμο $\mathrm{FPP}(G,k,c)$ [$\mathrm{FindPairPath}(G,k,c)$], ο οποίος απαντά στο αν υπάρχουν ζεύγη κόμβων στο γράφημα $G=(V,E)$ που να ενώνονται με πανχρωματικό μονοπάτι μήκους $k-1$, όταν το γράφημα έχει χρωματιστεί από ένα χρωματισμό $c\colon V \to \{1,\ldots,k\}$, τότε αυτός θα έχει την παρακάτω μορφή.

Όπως και πριν, χρησιμοποιώντας μια k-τέλεια οικογένοια συναρτήσεων κατακερματισμού $\mathcal{F}$, για το χρωματισμό του γραφήματος, έχουμε τον παρακάτω αλγόριθμο για την εύρεση ζευγαριών από κόμβους που συνδέονται με μονοπάτια μήκους $k-1$ και το Θεώρημα που ακολουθεί.

Θεώρημα 3.2 *Έστω $G=(V,E)$ ένα κατευθυνόμενο ή μη κατευθυνόμενο γράφημα. Όλα τα ζευγάρια των κόμβων που συνδέονται με μονοπάτι μήκους $k-1$ στο G, αν αυτά υπάρχουν, μπορούν να βρεθούν σε χρόνο $2^{O(k)} \cdot n^{\omega}$ στη χειρότερη περίπτωση.*

Εδώ να σημειώσουμε ότι αυτός ο αλγόριθμος δε βρίσκει ποιά είναι τα μονοπάτια μήκους $k-1$. Το μόνο που μας εξασφαλίζει είναι η ύπαρξή τους ή όχι καθώς και το ποιοί κόμβοι είναι τα δύο άκρα του. Για να βρούμε το ποιά είνα ακριβώς αυτά

Algorithm 3 FindPairPath(G, c) (FPP(G, c)

Είσοδος: Ένα γράφημα $G = (V, E)$, ένας ακέραιος k και χρωματισμός c.
Έξοδος: Απάντηση NAI ή OXI.

for $\{C_1, C_2\} \in \mathcal{P}(\{1, \ldots, k\})$ **do**
    $A_1 := FPP(G[c^{-1}\{C_1\}], k, c)$ ▷ $G_1 = G[c^{-1}\{C_1\}]$
    $A_2 := FPP(G[c^{-1}\{C_2\}], k, c)$ ▷ $G_2 = G[c^{-1}\{C_2\}]$
    $B := (V(G), \{(u, v) \in E(G) |\ u \in V(G_1)\ \&\ v \in V(G_2)\})$
    Υπολόγισε $D = A_1 B A_2$
    if Υπάρχει κάποιο $d_{i,j} = 1$ **then**
        Απάντησε NAI και δώσε $\{(i, j) |\ d_{i,j} = 1\}$
    end if
end for
Απάντησε OXI

Algorithm 4 Εύρεση κόμβων που συνδέονται με μονοπάτι μήκους $k - 1$

Είσοδος: Ένα γράφημα $G = (V, E)$ και ένας ακέραιος k.
Έξοδος: Απάντηση NAI ή OXI.

for $f \in \mathcal{F}$ **do**
    Χρησιμοποίησε την f για να χρωματίσεις το γράφημα.
    if $FPP(G, k, f) = NAI$ **then**
        Απάντησε NAI
    end if
end for
Απάντησε OXI

τα πανχρωματικά μονοπάτια χρησιμοποιούμε έναν άλλο αλγόριθμο των Alon και Naor [1] ο οποίος βρίσκει *πιστοποιητικά* σε Boolean πολλαπλασιασμό πινάκων.

3.2 Εύρεση δέντρου k κόμβων

Εφόσον έχουμε δει τι γίνεται στην περίπτωση που ψάχνουμε πανχρωματικό μονοπάτι μήκους $k-1$ θα ήταν ενδιαφέρον να δούμε πως μπορούμε να βρούμε σε ένα γράφημα ένα δέντρο που αποτελείται από k κόμβους. Στόχος μας είναι να φτιάξουμε μια αναδρομική σχέση που θα βασίζεται στη σταδιακή ανακατασκευή του δέντρου που ψάχνουμε κάθε φορά και η οποία θα παράγει τον ζητούμενο αλγόριθμο. Πριν όμως περάσουμε στην ανάλυση του αλγορίθμου είναι σημαντικό να αναφέρουμε τον παρακάτω ορισμό.

Ορισμός 3.5 *Τοπολογική διάταξη ενός κατευθυνόμενου άκυκλου γραφήματος είναι η αρίθμηση των κόμβων του έτσι ώστε για κάθε κατευθυνόμενη ακμή (i, j), δηλαδή από ένα κόμβο με αριθμό i σε ένα κόμβο με αριθμό j, να ισχύει $i < j$.*

Προφανώς ένα δέντρο είναι ένα άκυκλο γράφημα. Κι έστω ότι ένα κατευθυνόμενο δέντρο είναι αυτό στο οποίο όλες οι ακμές από ένα κόμβο δείχνουν προς το γονέα του. Έτσι, δεχόμενοι αυτή τη συνθήκη χωρίς καμία βλάβη της γενικότητας, τοπολογική διάταξη σε ένα τυχαίο δέντρο T είναι μια αρίθμηση των κόμβων του έτσι ώστε κάθε παιδί να έχει μικρότερο αριθμό από το γονέα του. Επιπλέον για κάθε δέντρο T ορίζουμε το σύνολο $\mathcal{T}$ ως το σύνολο που περιέχει τη διατεταγμένη ακολουθία των μερικά ριζωμένων υποδέντρων που σχηματίζουν το αρχικό T καθώς και το ίδιο το δέντρο. Ριζωμένο δέντρο είναι όταν έχουμε το γράφημα σε μορφή δέντρου, δηλαδή ένας κόμβος παίζει το ρόλο της ρίζας (r_T) και όλοι οι άλλοι κόμβοι είναι από κάτω. Μερικά ριζωμένο δέντρο είναι όταν παίρνουμε τη ρίζα (r_T), τη διασπάμε σε όσα παιδιά έχει και παίρνουμε όλα τα υποδέντρα που προκύπτουν. Ακριβώς τα ανάλογα ισχύουν και όταν μιλάμε για ριζωμένα υποδέντρα και μερικώς ριζωμένα υποδέντρα. Όλα τα παραπάνω θα χρειαστούν και θα γίνουν πιο κατανοητά στο θεώρημα που ακολουθεί.

Λήμμα 3.3 *Έστω T ένα κατευθυνόμενο ή μη κατευθυνόμενο δέντρο k κόμβων και $G = (V, E)$ ένα κατευθυνόμενο ή μη κατευθυνόμενο γράφημα. Ένα υπογράφημα του G ισομορφικό με το T, αν αυτό υπάρχει, μπορεί να βρεθεί σε αναμενόμενο χρόνο $2^{O(k)} \cdot m$ στην κατευθυνόμενη περίπτωση και σε χρόνο $2^{O(k)} \cdot n$ στη μη κατευθυνόμενη περίπτωση.*

Απόδειξη. Αρχικά διαλέγουμε έναν τυχαίο χρωματισμό $c\colon V \to \{1, \ldots, k\}$ του γραφήματος G και στη συνέχεια κάνουμε μια τοπολογική διάταξη του δέντρου που ψάχνουμε. Σχηματίζουμε το σύνολο $\mathcal{T}$. Αρχικά σε χρόνο $2^{O(k)} \cdot m$ βρίσκουμε για κάθε κόμβο $v \in V$ όλα τα σύνολα χρωμάτων που εμφανίζονται σε πανχρωματικές κόπιες του ριζωμένου T στο G και στις οποίες ο κόμβος v παίζει το ρόλο της ρίζας του δέντρου, $v = r_T$. Σχηματίζουμε λοιπόν τα παρακάτω σύνολα όπου ο δείκτης $T \in \mathcal{T}$:

$$\begin{aligned}
\mathcal{X}_T(v) = \big\{S \subseteq \{1, \ldots, k\} \mid\ & \exists V' \subseteq V(G) \\
& \&\ \exists \sigma\colon V' \to V(T) \\
& \&\ (\forall \{x, y\} \in E(T))[\sigma^{-1}(x), \sigma^{-1}(y) \in E(G)] \\
& \&\ v \in V' \ \&\ \sigma(v) = r_T \\
& \&\ |c(\sigma^{-1}(V(T)))| = |V(T)|\big\}
\end{aligned} \tag{3.5}$$

Η τιμή των $\mathcal{X}_T(v)$ είναι όλα τα σύνολα χρωμάτων για τα οποία υπάρχει συνάρτηση $\sigma\colon V' \to V(T)$ που αντιστοιχεί τα χρώματα (που έχει δώσει ο αρχικός χρωματισμός c) στους κόμβους του δέντρου, κάθε ακμή που υπάρχει στο δέντρο υπάρχει και στα αρχικό γράφημα G, επιπλέον έχουμε φιξαρισμένο χρώμα για τη ρίζα του δέντρου και τέλος η τελευταία σχέση μας εξασφαλίζει ότι ο αρχικός χρωματισμός c είναι πανχρωματικός.

Πριν προχωρήσουμε στην ανάλυση της αναδρομικής σχέσης που θα χρησιμοποιήσουμε πρέπει να γίνει σαφές το εξής. Το τελικό δέντρο που ψάχνουμε σχηματίζεται σταδιακά από μια διατεταγμένη ακολουθία ριζωμένων υποδέντρων. Για να καταλάβουμε πλήρως ποια είναι αυτή η διατεταγμένη ακολουθία, έστω ότι v είναι ένας τυχαίος κόμβος του δέντρου T που ψάχνουμε και ο οποίος δεν είναι η ρίζα. Ο κόμβος αυτός ορίζει ένα υποδέντρο του αρχικού T, έστω το T_v. Εφόσον όμως δεν είναι η ρίζα θα έχει και κάποιον γονέα, δηλαδή θα υπάρχει ακμή από τον v προς το γονέα του, έστω τον v'. Το υποδέντρο T_v μαζί με την ακμή (v, v') ορίζει ένα νέο υποδέντρο, το T_v^+. Έστω λοιπόν τώρα ότι ο γονέας v' έχει για παράδειγμα τρία παιδιά, τα v, u, w. Με την ίδια λογική όπως και πριν θα έχουμε τα T_u, T_u^+, T_w, T_w^+. Η ακολουθία των μερικά ριζωμένων υποδέντρων που θα σχηματίσει τελικά το ριζωμένο υποδέντρο $T_{v'}$ είναι η $T_v^+, T_u^+, T_w^+, T_{vu}^+, T_{vuw}^+ = T_{v'}$. Το ποιό θα είναι κάθε φορά το επόμενο στοιχείο αυτής της ακολουθίας εξαρτάται από τη μορφή του ριζωμένου δέντρου που ψάχνουμε, σύμφωνα με την τοπολογική διάταξη που του έχουμε δώσει. Κάθε φορά λοιπόν έχουμε δύο επιλογές για την εισαγωγή ενός νέου κόμβου, είτε με επέκταση είτε με σύνδεση.

Έστω λοιπόν τώρα ότι έχουμε ένα πανχρωματικό υποδέντρο T' του τελικού δέντρου T. Στη συνέχεια θα δείξουμε πώς δουλεύει η αναδρομή στις δύο περιπτώσεις που προκύπτουν ανάλογα με το δέντρο που ψάχνουμε.

Επέκταση: Από το ριζωμένο υποδέντρο T' με ρίζα v θέλουμε να συνεχίσουμε με μια ακμή $(v, u) \in E$ για να σχηματίσουμε το T'', ένα νέο μεγαλύτερο ριζωμένο ή μερικά ριζωμένο υποδέντρο του τελικού T. Στην περίπτωση αυτή δουλεύουμε όπως ακριβώς και στην προηγούμενη παράγραφο με το μονοπάτι. Δηλαδή ψάχνουμε έναν νέο κόμβο στη γειτονιά του v του οποίου το χρώμα δεν ανήκει στο $\mathcal{X}_{T'}(v)$. Δηλαδή θα έχουμε :

$$\begin{aligned}\mathcal{X}_{T''}(u) = \{S' \cup \{c(u)\} \mid S' \in \mathcal{X}_{T'}(v) \;\&\; \exists v \in N_G(u) \\ \&\; S' \cap \{c(u)\} = \emptyset\}\end{aligned} \tag{3.6}$$

Σύνδεση: Στο μερικά ριζωμένο υποδέντρο T' με ρίζα v θέλουμε να συνδέσουμε το μερικά ριζωμένο υποδέντρο T'' με ρίζα u, στις ρίζες τους. Δηλαδή να πάρουμε ένα νέο ριζωμένο ή μερικά ριζωμένο υποδέντρο T''' με ρίζα την w_{vu}, δηλαδή την ταύτιση των v και u που, όπως είναι φυσικό, πρέπει να έχουν το ίδιο χρώμα. (Για αυτό το λόγο, όπως είπαμε και στην αρχή, απαιτούμε σε κάθε ριζωμένο υποδέντρο που σχηματίζουμε να έχουμε μια ρίζα με φιξαρισμένο χρώμα). Δηλαδή έχουμε :

$$\begin{aligned}\mathcal{X}_{T'''}(w_{vu}) = \{S' \cup S'' \mid S' \in \mathcal{X}_{T'}(v) \;\&\; S'' \in \mathcal{X}_{T''}(u) \\ \&\; S' \cap S'' = c(w_{vu})\}\end{aligned} \tag{3.7}$$

Τελικά αυτό που ρωτάμε είναι αν για κάποιο $r \in V$, όπου r είναι η ρίζα του δέντρου T που ψάχνουμε, ισχύει :

$$\mathcal{X}_T(r) \neq \emptyset \tag{3.8}$$

Με μια παρόμοια ανάλυση όπως και στο Λήμμα 3.3 , για την περίπτωση του μονοπατιού, είναι εύκολο να δούμε ότι η πολυπλοκότητα του παραπάνω αναδρομικού αλγορίθμου είναι $2^{O(k)} \cdot m$. Για να πάρουμε την καλύτερη πολυπλοκότητα στην περίπτωση της μη κατευθυνόμενης περίπτωσης, δηλαδή να τρέχει σε $2^{O(k)} \cdot n$, εκμεταλλευόμαστε το γεγονός ότι ένα γράφημα $G = (V, E)$ με τουλάχιστον $k \cdot n$ ακμές περιέχει ως υπογράφημα οποιοδήποτε δέντρο k κόμβων. $\square$

Μια παρατήρηση που είναι χρήσιμο να κάνουμε σε αυτό το σημείο είναι ότι, στην περίπτωση της σύνδεσης, αν κάποιος γονέας έχει περισσότερα από δύο παιδιά, έστω ρ, τότε θα μπορούσαμε να έχουμε τη σύνδεση ρ μερικά ριζωμένων υποδέντρων $T_1, \ldots, T_\rho$ τα οποία έχουν ρίζες $v_1, \ldots, v_\rho$ αντίστοιχα. Δηλαδή :

$$\begin{aligned}\mathcal{X}_T(w_{1,\ldots,\rho}) = \Big\{ & S_1 \cup \cdots \cup S_\rho \mid \forall i \in \{1, \ldots, \rho\}\ S_i \in \mathcal{X}_T(v_i) \\ & \& \bigcap_{i=1,\ldots,\rho} S_i = c(w_{1,\ldots,\rho}) \Big\}\end{aligned} \tag{3.9}$$

Στην παραπάνω εξίσωση ο κόμβος $w_{1,\ldots,\rho}$ είναι η σύνδεση των $v_1, \ldots, v_\rho$ και είναι ξεκάθαρο ότι η εξίσωση (3.7) είναι ειδική περίπτωση της (3.9) .

3.3 Εύρεση γραφημάτων φραγμένου δεντροπλάτους

Η βασική ιδέα του αλγορίθμου της προηγούμενης ενότητας μπορεί να χρησιμοποιηθεί όχι μόνο για την εύρεση δέντρων αλλά και για την εύρεση γραφημάτων με φραγμένο δεντροπλάτος. Στη συνέχεια ακολουθεί ο ορισμός του δεντροπλάτους που προκύπτει από τη δεντροαποσύνθεση οποιουδήποτε τυχαίου γραφήματος, όπως δόθηκε από τους Robertson και Seymour [39] .

Ορισμός 3.6 *Μια δεντροαποσύνθεση ενός γραφήματος $G = (V, E)$ είναι ένα ζευγάρι (X, T) όπου $T = (I, F)$ είναι ένα δέντρο και $X = \{X_i : i \in I\}$ είναι μια οικογένοια υποσυνόλων του V τέτοια ώστε :*
(i) $\cup_{i \in I} X_i = V$.
(ii) Για κάθε ακμή $(u, v) \in E$ υπάρχει ένα $i \in I$ τέτοιο ώστε $u, v \in X_i$.
(iii) Αν τα $i, j, k \in I$ και και το j είναι στο μονοπάτι από το i στο k στο T, τότε $X_i \cap X_k \subseteq X_j$.
Το δεντροπλάτος μιας δεντροαποσύνθεσης (X, T) είναι το $\max_{i \in I} |X_i| - 1$. Το δεντροπλάτος ενός γραφήματος G είναι το ελάχιστο δεντροπλάτος όλων των δεντροαποσυνθέσεων του G.

Εκτός όμως από την απλή δεντροαποσύνθεση υπάρχει και η λεγόμενη *ωραία* δεντροαποσύνθεση με τα εξής χαρακτηριστικά.

Ορισμός 3.7 *Μια δεντροαποσύνθεση $D = (X, T)$ είναι ωραία αν το δέντρο T είναι ριζωμένο σε κάποιο κόμβο r με $|X_r| = 1$ και κάθε φύλλο αυτού του δέντρου αποτελείται επίσης από μια κορυφή, δηλαδή $|X_l| = 1$. Κάθε $t \in V(T)$ που δεν είναι φύλλο (συμπεριλαμβανομένου και της ρίζας) έχει ένα ή δύο παιδιά.*
(i) Αν το t έχει δύο παιδιά, t_1 και t_2, τότε $X_t = X_{t_1} = X_{t_2}$ και καλούμε τον X_t κόμβο σύνδεσης.
(ii) Αν το t έχει ένα παιδί t' τότε είτε $X_t = X_{t'} \cup \{v\}$ και καλούμε τον X_t κόμβο εισαγωγής και την v κορυφή εισαγωγής είτε $X_{t'} = X_t \cup \{v\}$ και καλούμε τον X_t κόμβο λήθης και την v κορυφή λήθης.

Ο τρόπος προσέγγισης του δυναμικού προγραμματισμού σε γραφήματα με μικρό δεντροπλάτος είναι να βρίσκουμε για κάθε $t \in V(T)$ ένα σύνολο που θα περιέχει πληροφορίες μιας μερικής λύσης για το G_t (δηλαδή το γράφημα από τον κόμβο t και κάτω). Η τιμή του συνόλου για τη ρίζα θα πρέπει να δίνει την τελική απάντηση. Στη συνέχεια ορίζουμε την τιμή του συνόλου για τα φύλλα. Τέλος, υπολογίζουμε αναδρομικά το σύνολο για κάθε κόμβο σύμφωνα με το σύνολο του παιδιού του ή των παιδιών του, ανάλογα με το αν έχουμε κόμβο εισαγωγής ή λήθης για την πρώτη περίπτωση ή κόμβο σύνδεσης για τη δεύτερη. Ακριβώς αυτή η λογική ακολουθείται και στον αλγόριθμο που παρουσιάζεται στη συνέχεια.

Θεώρημα 3.3 *Έστω H ένα μη κατευθυνόμενο γράφημα με k κόμβους και δεντροπλάτος t. Έστω $G = (V, E)$ ένα μη κατευθυνόμενο γράφημα. Ένα υπογράφημα του G ισομορφικό με το H, αν αυτό υπάρχει, μπορεί να βρεθεί σε αναμενόμενο χρόνο $2^{O(k)} \cdot V^{t+1}$ και στη χειρότερη περίπτωση σε χρόνο $2^{O(k)} \cdot V^{t+1} \log V$.*

Απόδειξη. Πριν περάσουμε στην περιγραφή του αλγορίθμου είναι απαραίτητο να ορίσουμε τα ακόλουθα. Σε ένα δέντρο T, ως $ch(i)$ συμβολίζουμε τα παιδιά του κόμβου i και ως $des(i)$ συμβολίζουμε όλους τους απογόνους του i. Δηλαδή οι απόγονοι του κόμβου i περιγράφονται από την αναδρομική σχέση $des(i) = \bigcup_{j \in ch(i)} des(j)$. Τώρα μπορούμε να περάσουμε στην περιγραφή. Αρχικά παίρνουμε μια ωραία δεντροαποσύνθεση του H και αυτό που θέλουμε είναι να ελέγξουμε αν το H υπάρχει στο γράφημα G. Στη συνέχεια χρωματίζουμε το γράφημα G με το χρωματισμό $c\colon V \to \{1, \ldots, k\}$. Τώρα, για κάθε κόμβο i της ωραίας δεντροαποσύνθεσης ορίζουμε το $H_i = G[\bigcup_{j \in des(i) \cup \{i\}} X_j]$ δηλαδή το εναγόμενο γράφημα του G από τις κορυφές που βρίσκονται στον κόμβο i και τους απογόνους του της ωραίας δεντροαποσύνθεσης. Επίσης ορίζουμε $V_i = V(H_i)$ δηλαδή V_i είναι οι κορυφές του αρχικού G που βρίσκονται στους κόμβους του H_i. Έστω $V' \subseteq V(G)$ το σύνολο των κορυφών που βρίσκονται στους κόμβους του H_i και $X \subseteq V'$ το σύνολο των κορυφών του G που βρίσκονται στον κόμβο i του H_i. Για κάθε τέτοιο σύνολο X παίρνουμε όλες τις πιθανές διαμερίσεις του $\{X^-, X^+\}$. Σύμφωνα με όλα τα παραπάνω για κάθε κόμβο i της ωραίας δεντροαποσύνθεσης υπολογίζουμε τα παρακάτω σύνολα :

$$\begin{aligned} \mathcal{Q}_i(X^+, X^-) = \Big\{ S \subseteq \{1, \ldots, k\} \mid\ & \exists V' \subseteq V(G) \;\&\; X \subseteq V' \\ & \&\; X = X^+ \cup X^- \;\&\; X^+ \cap X^- = \emptyset \\ & \&\; \exists \sigma\colon V_i \xrightarrow[\text{επί}]{1-1} V' \\ & \&\; (\forall e \in H_i)[\sigma(e) \in E(G[V'])] \\ & \&\; \sigma(X_i^+) = X^+ \;\&\; \sigma(X_i^-) = X^- \\ & \&\; |c(\sigma(V_i))| = |V_i| \Big\} \end{aligned} \tag{3.10}$$

Τα σύνολα αυτά ορίζονται σε όλες τις πιθανές διαμερίσεις του υποσυνόλου X του V'. Εξασφαλίζουμε ότι υπάρχει συνάρτηση σ από το V_i στο V' η οποία είναι ισομορφισμός και για κάθε ακμή που υπάρχει στο H_i μας δίνει ακμή που υπάρχει στο εναγόμενο υπογράφημα του G από το V' και, όπως είναι φυσικό, οι διαμερίσεις του X_i είναι οι αντίστοιχες με αυτές του X σύμφωνα με τη συνάρτηση σ. Επιπλέον, η τελευταία σχέση μάς λέει ότι ο αρχικός χρωματισμός είναι πανχρωματικός στο σύνολο V_i. Στη συνέχεια θα φτιάξουμε τις αναδρομικές σχέσεις για κάθε μία

από τις τέσσερις διαφορετικές περιπτώσεις που έχουμε.
Κόμβος φύλλο: Από τον τρόπο που έχουμε ορίσει την ωραία δεντροαποσύνθεση κάθε κόμβος φύλλο έχει μόνο μία κορυφή του αρχικού γραφήματος G, έστω τη v_ϕ. Έτσι λοιπόν το X είναι ένα μονοσύνολο $X = \{v_\phi\}$ και η τιμή της $\mathcal{Q}$ είναι το χρώμα που δίνεται από τον αρχικό χρωματισμό. Δηλαδή :

$$\mathcal{Q}_\phi(\{v_\phi\}, \emptyset) = \{c(v_\phi)\} \tag{3.11}$$

Κόμβος εισαγωγής: Όπως γνωρίζουμε στον κόμβο εισαγωγής έχουμε μια νέα κορυφή v_ν σε σχέση με το παιδί του. Δηλαδή ισχύει $X_i = X_j \cup \{v_\nu\}$. Το λεπτό σημείο σε αυτή την περίπτωση είναι ότι πρέπει να εξασφαλίσουμε ότι οι πιθανές ακμές που υπάρχουν ανάμεσα στη v_ν και στις κορυφές του κόμβου παιδιού X_j θα υπάρχουν και ανάμεσα στην κορυφή v (όπου $v = \sigma(v_\nu)$) του X με τις υπόλοιπες κορυφές που ανήκουν στο X. Οπότε θα έχουμε :

$$\mathcal{Q}_i(X^+, X^-) = \bigcup_{\substack{v \in X \\ X^+ - v \subseteq N_{G[X]}(v)}} \Big\{ S \subseteq \{1, \ldots, k\} \mid S - \{c(v_\nu)\} \in \mathcal{Q}_j(X^+ - v, X^- - v) \Big\} \tag{3.12}$$

Παίρνουμε δηλαδή την παράπάνω ένωση για κάθε $v \in X$ που μπορεί να παίζει το ρόλο της v_ν και το κλεδί είναι ότι το $X^+ - v$ πρέπει να είναι υποσύνολο της γειτονιάς της v στο εναγόμενο υπογράφημα του G από το X. Όπως είναι φυσικό τα S που μας δίνει αυτή η αναδρομή, χωρίς βέβαια το χρώμα της νέας κορυφής που έχει εισαχθεί, πρέπει να ανήκουν στην $\mathcal{Q}_j$ του παιδιού, με την ίδια διαμέριση του $\mathcal{Q}_i$ στο x, εκτός από τη νέα κορυφή που πρέπει να αφαιρεθεί από το X^+ ή X^- ανάλογα με το πού βρίσκεται.
Κόμβος λήθης: Όπως γνωρίζουμε στον κόμβο λήθης λείπει μια κορυφή, έστω η v_ξ, σε σχέση με το παιδί του. Δηλαδή ισχύει $X_i = X_j - \{v_\xi\}$. Σε αυτή την περίπτωση θα έχουμε :

$$\begin{aligned} \mathcal{Q}_i(X^+, X^-) = \bigcup_{v \in V(G) - X} \Big\{ & S \subseteq \{1, \ldots, k\} \mid S \cup \{c(v_\xi)\} \in \mathcal{Q}_j(X^+ \cup v, X^-) \\ & \vee S \cup \{c(v_\xi)\} \in \mathcal{Q}_j(X^+, X^- \cup v) \Big\} \end{aligned} \tag{3.13}$$

Εδώ τα πράγματα είναι αρκετά πιο απλά αφού αρκεί να πάρουμε την ένωση για όλες τις κορυφές v του αρχικού γραφήματος που μπορούν να παίζουν το ρόλο της v_ξ και δεν ανήκουν στο $X = X^+ \cup X^-$. Προφανώς τα σύνολα S που μας δίνει η αναδρομή μαζί με το χρώμα της v_ξ, δηλαδή της κορυφής που έχει ξεχαστεί, πρέπει να ανήκουν στη $\mathcal{Q}_j$ του παιδιού, είτε η v_ξ ανήκει στο X_j^+ είτε ανήκει στο X_j^-.
Κόμβος σύνδεσης: Ο κόμβος σύνδεσης X_i, που είναι ο μοναδικός που έχει δύο παιδιά, έστω τα X_j και X_p, έχει ακριβώς τις ίδιες κορυφές με τα παιδιά του, δηλαδή ισχύει $X_i = X_j = X_p$. Έχουμε λοιπόν την αναδρομή :

$$\begin{aligned} \mathcal{Q}_i(X^+, X^-) = \big\{ S_1 \cup S_2 \mid & S_1 \in \mathcal{Q}_j(X^+, X^-) \\ & \& \ S_2 \in \mathcal{Q}_p(X^+, X^-) \\ & \& \ S_1 \cap S_2 = c(X^+ \cup X^-) \big\} \end{aligned} \tag{3.14}$$

Εδώ παίρνουμε την ένωση των S_1 και S_2 που το καθένα ανήκει στα $\mathcal{Q}_j$ και $\mathcal{Q}_p$ των παιδιών αντίστοιχα. Επιπλέον θέλουμε η τομή τους να είναι ακριβώς τα χρώματα που δίνει ο αρχικός χρωματισμός του γραφήματος στο υποσύνολο $X = X^+ \cup X^-$ των κορυφών του.

Τελικά αυτό που ρωτάμε είναι αν για κάποιο $r \in V$ ισχύει :

$$\mathcal{Q}_r(\{r\}, \emptyset) \neq \emptyset \tag{3.15}$$

όπου r είναι το μοναδικό μέλος (κορυφή) του κόμβου X_r της ωραίας δεντροαποσύνθεσης και παίζει το ρόλο της ρίζας.

Από τον τρόπο κατασκευής της αναδρομής είναι πλέον εύκολο να δούμε ότι, αν το γράφημα G περιέχει μια πανχρωματική κόπια του γραφήματος H με τη δεδομένη ωραία δεντροαποσύνθεση, τότε η συλλογή $\mathcal{Q}_r(\{r\}, \emptyset)$ δεν θα είναι κενή. Από την άλλη, αν η συλλογή αυτή δεν είναι κενή, τότε θα περιέχει ακριβώς k χρώματα που θα αντιστοιχούν στην πανχρωματική κόπια του H, με τη δεδομένη δεντροαποσύνθεση, στο G.

Οι χρόνοι του Θεωρήματος προκύπτουν άμεσα από την παραπάνω λεπτομερή ανάλυση του αλγορίθμου.$\square$

Πριν κλείσουμε να σημειώσουμε ότι τα τελευταία χρόνια έχει γίνει σημαντική δουλειά πάνω σε γραφοθεωρητικά προβλήματα όπου ο τρόπος προσέγγισηε της λύσης τους έχει να κάνει με χρωματισμό των κόμβων και παραμετροποίηση ως προς το δεντροπλάτος [14] .

Κεφάλαιο 4

Πιθανοτικές προσεγγίσεις

4.1 Εύρεση κύκλων σε κλειστές ως προς ελάσσον οικογένειες γραφημάτων

Στην παράγραφο αυτή θα χρησιμοποιήσουμε τη μέθοδο της χρωματικής κωδικοποίησης που περιγράψαμε για να βρίσκουμε κύκλους σε κλειστές ως προς ελάσσον (minor closed) οικογένειες γραφημάτων. Πριν όμως περάσουμε στην περιγραφή και ανάλυση του αλγορίθμου είναι πάλι απαραίτητο, για την καλύτερη κατανόηση, να δώσουμε κάποιους ορισμούς οι οποίοι μπορούν να βρεθούν αναλυτικά στα [4,31,32].

Ορισμός 4.1 *Ένα μη κατευθυνόμενο γράφημα $G = (V, E)$ είναι d-εκφυλισμένο αν υπάρχει μια άκυκλη κατευθυνόμενη εκδοχή αυτού στην οποία για κάθε $v \in V$ ισχύει $d_{out}(v) \leq d$. Το μικρότερο d για το οποίο το γράφημα G είναι d-εκφυλισμένο καλείται ο εκφυλισμός ή ο max-min βαθμός του G και συμβολίζεται ως $d(G)$.*

Παρατήρηση 4.1 *Εύκολα μπορεί να δει κανείς ότι ο $d(G)$ είναι ο μέγιστος των ελάχιστων βαθμών όλων των υπογραφημάτων του G. Δηλαδή ισχύει :*

$$d(G) = \max\{k \mid G \text{ περιέχει υπογράφημα } H \text{ με } d_{min}(H) \geq k\} \tag{4.1}$$

όπου $d_{min}(H)$ ο ελάχιστος βαθμός του γραφήματος H.

Παρατήρηση 4.2 *Αν ένα G είναι d-εκφυλισμένο τότε ισχύει $|E| \leq d \cdot |V|$.*

Ορισμός 4.2 *Ένα γράφημα H είναι ελάσσον ενός G αν μπορούμε να το πάρουμε από το G με αφαιρέσεις και συνθλίψεις ακμών. Μια οικογένεια $\mathcal{C}$ γραφημάτων λέμε ότι είναι κλειστή ως προς ελάσσον αν το ελάσσον ενός γραφήματος της οικογένειας είναι και αυτό μέλος της οικογένειας. Μια οικογένεια είναι μη τετριμμένη αν δεν περιλαμβάνει όλα τα γραφήματα.*

Είναι γνωστό ότι αν $\mathcal{C}$ είναι μια μη τετριμμένη, κλειστή ως προς ελάσσον οικογένεια γραφημάτων τότε όλα τα γραφήματα στη $\mathcal{C}$ έχουν φραγμένο εκφυλισμό [3]. Δηλαδή υπάρχει μια σταθερά $d = d_{\mathcal{C}}$ τέτοια ώστε για κάθε $G \in \mathcal{C}$ ισχύει $d(G) \leq d$. Ένα προφανές παράδειγμα είναι η οικογένεια των επίπεδων γραφημάτων. Είναι μη τετριμμένη (αφού δεν περιέχει όλα τα γραφήματα), κλειστή ως προς ελάσσον και ο εκφυλισμός κάθε επίπεδου γραφήματος είναι το πολύ 5 (αφού κάθε επίπεδο γράφημα έχει κόμβους των οποίων οι βαθμοί είναι το πολύ 5). Σύμφωνα με όλα τα παραπάνω έχουμε τα θεωρήματα που ακολουθούν.

Θεώρημα 4.1 *Έστω ότι $\mathcal{C}$ είναι μια μη τετριμμένη, κλειστή ως προς ελάσσον οικογένεια γραφημάτων κι έστω $k \geq 3$ ένας σταθερός ακέραιος. Τότε, υπάρχει πιθανοτικός αλγόριθμος ο οποίος δεδομένου ενός γραφήματος $G = (V, E)$, από τη $\mathcal{C}$, βρίσκει ένα C_k (απλός κύκλος μεγέθους k) στο G, αν αυτός υπάρχει, σε $O(V)$ αναμενόμενο χρόνο.*

Απόδειξη. Έστω ένα γράφημα $G = (V, E)$ από τη $\mathcal{C}$ που περιέχει έναν C_k. Διαλέγουμε ένα τυχαίο χρωματισμό $c \colon V \to \{1, \ldots, k\}$ των κόμβων του G. Ένας C_k στο G είναι καλά χρωματισμένος (πανχρωματικός) αν οι κόμβοι του είναι συνεχόμενα χρωματισμένοι από τα $1, 2, \ldots, k$. Με πιθανότητα $2/k^{k-1}$ ο C_k που υπάρχει στο G θα είναι καλά χρωματισμένος. Στη συνέχεια θα περιγράψουμε έναν αλγόριθμο, ο οποίος σε χρόνο $O(k \cdot V)$, δεδομένου ενός γραφήματος $G = (V, E)$ από μια κλειστή ως προς ελάσσον οικογένεια με εκφυλισμό $d = O(1)$ και ενός χρωματισμού $c \colon V \to \{1, \ldots, k\}$, έχει πιθανότητα τουλάχιστον $1/(2d)^k$ να βρει έναν καλά χρωματισμένο C_k στο G, αν αυτός υπάρχει. Συνδυάζοντας αυτόν τον αλγόριθμο με τον αρχικό τυχαίο χρωματισμό, παίρνουμε έναν αλγόριθμο χρόνου $O(k \cdot V)$ που βρίσκει έναν C_k στο G, αν αυτός υπάρχει, με πιθανότητα τουλάχιστον $\frac{2}{(2d)^k k^{k-1}}$. Αυτή μπορεί να φαίνεται ότι είναι μια πολύ μικρή πιθανότητα, αλλά το σημαντικό σε αυτή την περίπτωση είναι ότι εξαρτάται μόνο από τα k και d και όχι από το μέγεθος του γραφήματος. Ξανατρέχοντας τον αλγόριθμο, σε περίπτωση αποτυχίας, από ένα ανεξάρτητο σύνολο επιλογών, παίρνουμε τελικά έναν $O((2dk)^k \cdot V)$ αναμενόμενου χρόνου αλγόριθμο που βρίσκει έναν C_k σε γραφήματα που τον περιέχουν.

Στη συνέχεια ακολουθεί η περιγραφή των βημάτων του αλγορίθμου. Τα ($\triangleright$) είναι κάποια επεξηγηματικά σχόλια για την καλύτερη κατανόησή τους. Οπότε έχουμε :

Βήμα 1. Αν οι ακμές ενώνουν κόμβους που είναι χρωματισμένοι με συνεχόμενα χρώματα $(mod k)$, άφησε το G όπως είναι. Αλλιώς, αφαίρεσε τις ακμές που δεν έχουν αυτή την ιδιότητα.

$\triangleright$ Οι παραπάνω ενέργειες δεν επηρεάζουν έναν καλά χρωματισμένο C_k στο G.

Βήμα 2. Δώσε κατευθύνσεις στις ακμές έτσι ώστε ο βαθμός εξόδου κάθε κόμβου να είναι το πολύ d.

Βήμα 3. Ανάθεσε ετικέτες $1, \ldots, d_{out}(v) \leq d$ με αυθαίρετο τρόπο στις ακμές που φεύγουν από ένα κόμβο.

$\triangleright$ Αυτό παίρνει $O(V)$ χρόνο. Ο υποτιθέμενος καλά χρωματισμένος C_k στο G περιέχει σίγουρα μια ακμή ανάμεσα στον κόμβο v_{k-1} (με χρώμα $k-1$) και στον κόμβο v_k (με χρώμα k).

Βήμα 4. Βρες αυτή την ακμή και μάντεψε (με δίκαιο τρόπο) την κατεύθυνση και την ετικέτα της.

$\triangleright$ Υπάρχουν δύο πιθανές κατευθύνσεις και d πιθανές ετικέτες.

Βήμα 5. Αν μαντέψεις ότι η κατεύθυνση είναι από τον v_{k-1} στον v_k και η ετικέτα είναι i, τότε όλες οι ακμές που φεύγουν από κόμβους με χρώμα $k-1$ των οποίων η ετικέτα δεν είναι i αφαιρούνται από το γράφημα. Αλλιώς, δηλαδή αν μαντέψεις το αντίθετο (την άλλη κατεύθυνση), κάνε το αντίστοιχο με τις ακμές που φεύγουν από κόμβους με χρώμα k.

$\triangleright$ Το νέο γράφημα G' που παίρνουμε περιέχει ακόμη έναν καλά χρωματισμένο C_k με πιθανότητα τουλάχιστον $1/2d$. Το υπογράφημα του G' που ανάγεται από τους

κόμβους με χρώματα $k-1$ και k, είναι δάσος από ριζωμένα αστέρια.
Βήμα 6. Σύνθλιψε κάθε τέτοιο αστέρι σε έναν απλό κόμβο κι ανάθεσε σε κάθε τέτοιο κόμβο το χρώμα $k-1$.
$\triangleright$ Παίρνουμε ένα νέο γράφημα G'' με νέο χρωματισμό c''. Είναι εύκολο να δούμε ότι υπάρχει καλά χρωματισμένος C_{k-1} στο G'' αν και μόνο αν υπάρχει καλά χρωματισμένος C_k στο G'. Αρκεί να σκεφτούμε ότι κάθε ακμή στο G', οπότε και κάθε ακμή στο G'', ενώνουν κόμβους με συνεχόμενα χρώματα. Επίσης παρατηρούμε ότι $G'' \in \mathcal{C}$ γιατί το G'' είναι ελάσσον του G και η $\mathcal{C}$ είναι κλειστή ως προς ελάσσον.
Βήμα 7. Επανεκτέλεσε τα παραπάνω βήματα (αναδρομικά) και τώρα ψάξε για καλά χρωματισμένο C_{k-1} στο G''.
$\triangleright$ Αυτό παίρνει $O((k-1)\cdot V)$ χρόνο και βρίσκει καλά χρωματισμένο C_{k-1} με πιθανότητα τουλάχιστον $1/(2d)^{k-1}$.
Βήμα 8. Αν βρεις έναν καλά χρωματισμένο C_{k-1} στο G'', ανακατασκεύασε τον καλά χρωματισμένο C_k του G (που σίγουρα υπάρχει). Αν όχι, συνέχισε με την αναδρομή.

Ο συνολικός χρόνος είναι $O(k\cdot V)$ και η πιθανότητα εύρεσης ενός καλά χρωματισμένου C_k είναι τουλάχιστον $1/(2d)^k$. Για να ολοκληρώσουμε την εικόνα για το πώς λειτουργεί ο αλγόριθμος πρέπει να ξεκαθαρίσουμε με ποιο τρόπο σταματάει η αναδρομή. Οι συνθλίψεις που εφαρμόζονται σε διάφορα βήματα δημιουργούν self-loops, τα οποία αφαιρούνται αμέσως. Επιπλέον, παράλληλες ακμές μπορούν να προκύψουν μόνο όταν οι συνθλίψεις εφαρμόζονται σε γραφήματα που χρωματίζονται από τρία χρώματα και μετά το έκτο βήμα παίρνουμε ένα γράφημα που είναι χρωματισμένο με δύο χρώματα. Ένας καλά χρωματισμένος C_2 σε ένα τέτοιο γράφημα είναι απλά ένα ζευγάρι παράλληλων ακμών, κι ένα τέτοιο ζευγάρι, αν αυτό υπάρχει, μπορεί να βρεθεί σε $O(V)$ χρόνο. $\square$

Ο παραπάνω αλγόριθμος είναι πιθανοτικός. Το ενδιαφέρον σε αυτή την περίπτωση θα ήταν να είχαμε έναν ντετερμινιστικό αλγόριθμο, χωρίς μεγάλη απώλεια χρόνου. Χρησιμοποιώντας, όπως και στην προηγούμενη παράγραφο, μια k-τέλεια οικογένοια συναρτήσεων κατακερματισμού παίρνουμε το ακόλουθο αποτέλεσμα.

Θεώρημα 4.2 *Έστω ότι $\mathcal{C}$ είναι μια μη τετριμμένη, κλειστή ως προς ελάσσον οικογένεια γραφημάτων κι έστω $k \geq 3$ ένας σταθερός ακέραιος. Υπάρχει ντετερμινιστικός αλγόριθμος που αποφασίζει αν ένα δοσμένο γράφημα $G=(V,E)$ από την $\mathcal{C}$ περιέχει έναν C_k, και τον βρίσκει, αν αυτός υπάρχει, σε χρόνο $O(V\cdot\log V)$ στη χειρότερη περίπτωση.*

Απόδειξη. Ουσιαστικά η σημαντική διαφορά είναι ότι αντί να χρησιμοποιήσουμε τυχαίους χρωματισμούς, εξαντλούμε μια λίστα από $k^{O(k)}\log V$ χρωματισμούς που έχουν την ιδιότητα ότι κάθε ακολουθία $v_1, v_2, \ldots, v_k$, k κόμβων από το V, είναι συνεχόμενα χρωματισμένοι από τα χρώματα $1, 2, \ldots, k$ από τουλάχιστον ένα χρωματισμό της λίστας. Η λίστα αυτή είναι η κατάλληλη k-τέλεια οικογένοια hash συναρτήσεων που χρησιμοποιούμε για το πρόβλημά μας. Από εκεί και πέρα αντί να μαντεύουμε κάθε φορά την κατεύθυνση και την ετικέτα της κάθε ακμής σε έναν καλά χρωματισμένο C_k, εξαντλούμε, για κάθε χρωματισμό, όλες τις $(2d)^k$ πιθανές επιλογές (d είναι ο μέγιστος βαθμός εξόδου στην κατευθυνόμενη εκδοχή επί 2 για τις δύο πιθανές κατευθύνσεις εις την k για όλες τις ακμές του C_k). Αν το G περιέχει έναν C_k τότε τουλάχιστον ένας C_k θα βρεθεί με αυτόν τον τρόπο. $\square$

Κλείνοντας την παράγραφο να σημειώσουμε ότι τα θεωρήματα 4.1 και 4.2 είναι για μη κατευθυνόμενα γραφήματα. Με μόνο κάποιες μικρές τροποποιήσεις μπορούν να χρησιμοποιηθούν για την εύρεση κατευθυνόμενων κύκλων σε κατευθυνόμενα γραφήματα των οποίων η μη κατευθυνόμενη εκδοχή ανήκει σε οποιαδήποτε μη τετριμμένη, κλειστή ως προς ελάσσον οικογένεια $\mathcal{C}$.

4.2 Εύρεση υπογραφημάτων φραγμένου δεντροπλάτους

Στην παράγραφο αυτή θα δείξουμε, πώς χρησιμοποιώντας τη μέθοδο της χρωματικής κωδικοποίσης μπορούμε αρχικά να βρούμε οποιοδήποτε φιξαρισμένο κατευθυνόμενο ή μη κατευθυνόμενο δάσος. Στη συνέχεια χρησιμοποιώντας αυτή τη μέθοδο σε συνδυασμό με τον ορισμό 3.6 θα μπορούμε να βρούμε σε ένα γράφημα οποιοδήποτε υπογράφημα που είναι ισομορφικό με κάποιο κατευθυνόμενο ή μη κατευθυνόμενο γράφημα με φραγμένο δεντροπλάτος. Πρίν περάσουμε στο βασικό θεώρημα της παραγράφου, να σημειώσουμε ότι ένα κατευθυνόμενο δάσος είναι ένα κατευθυνόμενο γράφημα του οποίου η μη κατευθυνόμενη εκδοχή είναι δάσος.

Θεώρημα 4.3 *Έστω ότι F είναι ένα κατευθυνόμενο ή μη κατευθυνόμενο δάσος k κόμβων. Έστω ότι $G = (V, E)$ είναι ένα κατευθυνόμενο ή μη κατευθυνόμενο γράφημα. Ένα υπογράφημα του G που είναι ισομορφικό με το F, αν αυτό υπάρχει, μπορεί να βρεθεί σε αναμενόμενο χρόνο $2^{O(k)} \cdot E$ στην κατευθυνόμενη περίπτωση και σε $2^{O(k)} \cdot V$ στη μη κατευθυνόμενη περίπτωση.*

Απόδειξη. Αρχίζουμε, όπως κάνουμε συνήθως, διαλέγοντας έναν τυχαίο χρωματισμό $c\colon V \to \{1, \ldots, k\}$ του γραφήματος G, στο οποίο υποθέτουμε ότι περιέχει μια κόπια του F. Με πιθανότητα τουλάχιστον e^{-k} (αφού $k!/k^k > e^{-k}$), η κόπια του F στο G θα είναι πανχρωματική, δηλαδή ο κάθε κόμβος θα έχει διαφορετικό χρώμα από τους άλλους. Υποθέτουμε λοιπόν ότι συμβαίνει αυτό το γεγονός. Έστω τώρα ότι το δάσος F αποτελείται από l κατευθυνόμενα δέντρα $T_1, \ldots, T_l$ με $k_1, \ldots, k_l$ κόμβους το καθένα (όπου $k_1 + \cdots + k_l = k$). Έστω ότι F_i, για $1 \leq i \leq l$, είναι ένα κατευθυνόμενο δάσος που αποτελείται από τα δέντρα $T_1, \ldots, T_i$. Βρίσκουμε για κάθε $1 \leq i \leq l$ τα σύνολα χρωμάτων που εμφανίζονται σε πανχρωματικές κόπιες του T_i στο G. Είναι πλέον εύκολο να βρούμε, σε $2^{O(k)}$ χρόνο, τα σύνολα χρωμάτων που εμφανίζονται σε πανχρωματικές κόπιες του F_i για $1 \leq i \leq l$. Να σημειώσουμε ότι οι κόπιες των T_i και T_j, για $i \neq j$, με διαφορετικά σύνολα χρωμάτων είναι αναγκαία διαφορετικές. Αν η συλλογή που αντιστοιχεί στο $F = F_l$ δεν είναι ικανή, τότε το G περιέχει μια πανχρωμτική κόπια του F. Μια τέτοια κόπια βρίσκεται αν, με κάθε σύνολο χρωμάτων που βρίσκουμε, κρατάμε τουλάχιστον ένα αντίγραφο του αντίστοιχου υπογραφήματος που χρωματίζεται από αυτό.

Στη συνέχεια θα δούμε πώς βρίσκουμε το κάθε δέντρο ξεχωριστά. Έστω λοιπόν ότι $T = T_i$ είναι ένα κατευθυνόμενο δέντρο με $m = k_i$ κόμβους, όπου $1 \leq i \leq l$, κι έστω ότι r είναι ένας τυχαίος κόμβος του T. Σε χρόνο $2^{O(m)}$, βρίσκουμε για κάθε κόμβο $v \in V$ όλα τα σύνολα χρωμάτων που εμφανίζονται σε πανχρωματικές κόπιες του T στο G και στις οποίες ο v παίζει το ρόλο του r. Αν το T περιέχει μόνο ένα κόμβο τότε αυτό γίνεται πολύ εύκολα. Αν περιέχει περισσότερους τότε, έστω ότι $e_T = (r, r')$ είναι μια κατευθυνόμενη ακμή του T. Η αφαίρεση της e_T από το T, χωρίζει το δέντρο σε δύο κατευθυνόμενα υποδέντρα T' και T''. Αναδρομικά βρίσκουμε, για κάθε $v \in V$ όλα τα σύνολα χρωμάτων που εμφανίζονται σε πανχρωματικές κόπιες του T' στις οποίες ο v παίζει το ρόλο του

r και όλα τα σύνολα χρωμάτων που εμφανίζονται σε πανχρωματικές κόπιες του T'' στις οποίες ο v παίζει το ρόλο του r'. Για κάθε κατευθυνόμενη ακμή $e = (u, v) \in E$, αν το C' είναι το σύνολο χρωμάτων που εμφανίζεται στη συλλογή του u (που αντιστοιχεί στο T') και αν C'' είναι το σύνολο χρωμάτων που εμφανίζεται στη συλλογή του v (που αντιστοιχεί στο T'') και αν $C' \cap C'' = \emptyset$, τότε το $C' \cup C''$ προστίθεται στη συλλογή του u (που αντιστοιχεί στο T). Είναι εύκολο να δούμε ότι η πολυπλοκότητα αυτού του αλγορίθμου είναι $2^{O(m)}$, όπως απαιτείται.

Για να πετύχουμε το καλύτερο όριο στη μη κατευθυνόμενη περίπτωση, εκμεταλλευόμαστε το γεγονός ότι σε ένα γράφημα $G = (V, E)$ με τουλάχιστον $k \cdot |V|$ ακμές, περιέχεται σαν υπογράφημα οποιοδήποτε δάσος k κόμβων.$\square$

Όπως είπαμε και στην αρχή, σε αναλογία με την παράγραφο 3.2 χρησιμοποιώντας τη βασική ιδέα του παραπάνω αλγορίθμου σε συνδυασμό με τον ορισμό 3.6 μπορούμε να πάρουμε ένα νέο αλγόριθμο, ο οποίος ψάχνει όχι μόνο δέντρα και δάση αλλά οποιοδήποτε γράφημα με φραγμένο δεντροπλάτος. Έτσι λοιπόν έχουμε το παρακάτω θεώρημα.

Θεώρημα 4.4 *Έστω H ένα κατευθυνόμενο ή μη κατευθυνόμενο γράφημα k κόμβων με δεντροπλάτος t. Έστω $G = (V, E)$ ένα κατευθυνόμενο ή μη κατευθυνόμενο γράφημα. Ένα υπογράφημα του G ισομορφικό με το H, αν αυτό υπάρχει, μπορεί να βρεθεί σε αναμενόμενο χρόνο $2^{O(k)} \cdot V^{t+1}$ και σε χρόνο $2^{O(k)} \cdot V^{t+1} \log V$ στη χειρότερη περίπτωση.*

Κεφάλαιο 5

Divide and Color

Όπως μπορεί κάποιος να καταλάβει εύκολα, η μέθοδος Διαίρει και Χρωμάτιζε (Divide and Color) [28] είναι ένας συνδυασμός δύο μεθόδων, της πολύ γνωστής Διαίρει και Βασίλευε (Divide and Conquer) και της σχετικά νέας Χρωματικής Κωδικοποίησης (Color Coding). Η βασική ιδέα της νέας αυτής προσέγγισης είναι να χρησιμοποιούνται μόνο δύο χρώματα (άσπρο και μαύρο) και να λύνεται αναδρομικά ένα αναγόμενο στιγμιότυπο του προβλήματος σε κάθε ένα από τα αναγόμενα υπογραφήματα. Οι δύο λύσεις που θα προκύψουν με αυτό τον τρόπο πρέπει να συνδυαστούν σε μια λύση για το αρχικό στιγμιότυπο. Η μέθοδος αυτή έχει τρία πολύ βασικά πλεονεκτήματα :

1. Όταν χρωματίζουμε σε στάδια, ένας λάθος χρωματισμός μπορεί να διορθωθεί στο στάδιο που τρέχει, χωρίς να πρέπει να ξεκινήσουμε από την αρχή.

2. Δε χρειάζεται να λυθεί το πολύπλοκο πρόβλημα που προκύπτει από κάποιον αλγόριθμο δυναμικού προγραμματισμού.

3. Η αποτυχαιοποίηση μπορεί να γίνει πολύ πιο εύκολα στη νέα αυτή προσέγγιση [16].

Όλα τα παραπάνω θα γίνουν πιο ξεκάθαρα στη συνέχεια. Αρχικά, όπως και στην περίπτωση της Χρωματικής Κωδικοποίησης θα λύσουμε το πρόβλημα της εύρεσης απλού μονοπατιού μήκους $k-1$.

5.1 Εύρεση απλού μονοπατιού μήκους $k-1$

Αρχικά δίνουμε τον ορισμό του προβλήματος απόφασης Longest Path.

Ορισμός 5.1 *Το πρόβλημα Longest Path ορίζεται ως εξής. Μας δίνεται ένα γράφημα $G = (V, E)$ κι ένας ακέραιος k, που θεωρείται η παράμετρος. Η ερώτηση είναι αν υπάρχει απλό μονοπάτι μήκους $k-1$ στο G.*

Ο τρόπος προσέγγισης του προβλήματος, η μέθοδος δηλαδή που θα αναλύσουμε στη συνέχεια, είναι ότι αντί να λύσει κατευθείαν το πρόβλημα απόφασης Longest Path, χρησιμοποιεί μια εμφύτευση αυτού στο ακόλουθο στενά συνδεδεμένο πρόβλημα ψαξίματος.

Ορισμός 5.2 *Δεδομένου ενός γραφήματος $G = (V, E)$, γράφουμε $u \xrightarrow{k} v$ για δύο κόμβους $u, v \in V$ όταν υπάρχει ένα απλό μονοπάτι μήκους $k - 1$ από το u στο v. Το πρόβλημα Extended Longest Path ορίζεται ως εξής. Μας δίνεται ένα γράφημα $G = (V, E)$ κι ένας ακέραιος k, που θεωρείται η παράμετρος. Ψάχνουμε ένα σύνολο $\{(u, v) \in V^2 | u \xrightarrow{k} v\}$. Λέμε ότι ένας αλγόριθμος λύνει το Extended Longest Path με πιθανότητα λάθους p αν για κάθε γράφημα $G = (V, E)$ και κάθε $u, v \in V$ με $u \xrightarrow{k} v$, το αποτέλεσμά του στην είσοδο G, k περιέχει τα (u, v) με πιθανότητα τουλάχιστον p.*

Έχουμε λοιπόν τον παρακάτω αλγόριθμο L και το λήμμα που ακολουθεί.

Algorithm 5 Extended Longest Path (L)

Είσοδος: Ένα γράφημα $G = (V, E)$ και ένας ακέραιος k (η παράμετρος).
Έξοδος: Το σύνολο $\{(u, v) \in V^2 | u \xrightarrow{k} v\}$.

```
if k = 1 then
    Απάντησε {(v, v)|v ∈ V}
end if
for 3 · 2^k do
    Διάλεξε κάποιο V' ∈ 2^V με ομοιόμορφη πιθανότητα.
    G_1 := G[V'], G_2 := G[v − v'], R := ∅
    for all u, v, w, x ∈ V do
        if (u, v) ∈ L(G_1, ⌈k/2⌉) & {v, w} ∈ E & (w, x) ∈ L(G_2, ⌊k/2⌋) then
            R := R ∪ {(u, x)}
        end if
    end for
end for
Απάντησε R
```

Λήμμα 5.1 *Ο αλγόριθμος L λύνει το Extended Longest Path με πιθανότητα λάθους το πολύ $1/4$ σε χρόνο $O^*(4^k)$.*

Απόδειξη. Είναι εύκολο να δούμε ότι αν ο αλγόριθμος L επιστρέφει ένα ζευγάρι $\{u, v\}$, τότε όντως ισχύει $u \xrightarrow{k} v$. Αντίστροφα, αν το γράφημα G περιέχει ένα τέτοιο μονοπάτι τότε ο αλγόριθμος L δεν επιστρέφει το αντίστοιχο ζευγάρι $\{u, v\}$ με κάποια πιθανότητα p_k.

Υποθέτουμε ότι υπάρχει ένα απλό μονοπάτι k κόμβων (δηλαδή μήκους $k - 1$) από το u στο v στο G. Η πιθανότητα ότι οι πρώτοι $\lceil k/2 \rceil$ κόμβοι είναι μαύροι και οι υπόλοιποι $\lfloor k/2 \rfloor$ είναι άσπροι είναι 2^{-k} (αφού έχουμε k κόμβους και δύο χρώματα). Σε αυτή την περίπτωση, οι $L(G_1)$ και $L(G_2)$ δεν περιέχουν ζευγάρια που να επιτρέπουν στον αλγόριθμο να εισάγει το $\{u, v\}$ στο R με πιθανότητα το πολύ $2p_{\lceil k/2 \rceil}$ (αφού $p_{\lceil k/2 \rceil} + p_{\lfloor k/2 \rfloor} \leq 2p_{\lceil k/2 \rceil}$). Μετά από $3 \cdot 2^k$ επαναλήψεις, η πιθανότητα ότι το $\{u, v\} \notin R$ είναι τουλάχιστον

$$\left(1 - 2^{-k} + 2^{-k+1} p_{\lceil k/2 \rceil}\right)^{3 \cdot 2^k} \tag{5.1}$$

γιατί με πιθανότητα $1 - 2^{-k}$ ο χρωματισμός είναι κακός (αφού όπως είπαμε και στην αρχή με πιθανότητα 2^{-k} έχουμε καλό χρωματισμό) και με πιθανότητα το πολύ $2^{-k+1}p_{\lceil k/2\rceil}$ ο χρωματισμός είναι καλός αλλά ο αναδρομικός έλεγχος του άσπρου ή μαύρου υπομονοπατιού αποτυγχάνει (δηλαδή έχουμε $2^{-k}\cdot 2p_{\lceil k/2\rceil}$).

Στη συνέχεια δείχνουμε με επαγωγή στο k ότι $p_k \leq 1/4$. Προφανώς $p_1 = 0$ και αν υποθέσουμε ότι για $k \geq 2$ ισχύει για κάθε k μέχρι το p_k θα έχουμε την ανισότητα $p_k \leq (1-2^{-k}+2^{-k+1}/4)^{3\cdot 2^k} = (1-2^{-(k+1)})^{\frac{3}{2}\cdot 2^{k+1}} \leq e^{-3/2} < 1/4$ που μας δίνει το ζητούμενο.

Έστω τώρα ότι T_k είναι ο αριθμός των αναδρομικών κλήσεων που γίνονται από τον αλγόριθμο L. Παρατηρούμε ότι οι αναδρομικές κλήσεις που γίνονται στο εσωτερικό for loop έχουν τις ίδιες παραμέτρους για όλα τα u, v, w, x. Αρκεί λοιπόν να πραγματοποιήσουμε αυτές τις δύο κλήσεις μόνο μία φορά για κάθε χρωματισμό και να θυμόμαστε τα αποτελέσματα. Έτσι παίρνουμε την παρακάτω αναδρομή

$$T_k \leq 3\cdot 2^k \cdot (T_{\lceil k/2\rceil} + T_{\lfloor k/2\rfloor}) \leq 3\cdot 2^{k+1}T_{\lceil k/2\rceil}. \tag{5.2}$$

Χρησιμοποιώντας το απλό γεγονός ότι $k+\lceil k/2\rceil+\lceil\lceil k/2\rceil/2\rceil+\cdots+1 \leq 2k+\log k$, όπου το άθροισμα τελειώνει όταν κάποιος όρος γίνεται 1, παίρνουμε τελικά

$$T_k = O(3^{\log k}k^2 4^k) = O(k^{\log 3}k^2 4^k) = O^*(4^k). \tag{5.3}$$

Όλες οι άλλες διεργασίες που εκτελούνται κατά τη διάρκεια μιας κλήσης του αλγορίθμου παίρνουν μόνο πολυωνυμικό χρόνο. Έτσι λοιπόν, σύμφωνα με όλα τα παραπάνω, ο αλγόριθμος L βρίσκει ένα μονοπάτι μήκους $k-1$, αν αυτό υπάρχει, με πιθανότητα τουλάχιστον $3/4$. □

Αρχικός μας σκοπός όμως ήταν να λύσουμε το πρόβλημα Longest Path. Σύμφωνα με τα παραπάνω μπορούμε πολύ εύκολα να αποδείξουμε το ακόλουθο θεώρημα.

Θεώρημα 5.1 *Το πρόβλημα Longest Path μπορεί να λυθεί με εκθετικά μικρή πιθανότητα λάθους σε χρόνο $O^*(4^k)$.*

Απόδειξη. Τρέχοντας τον αλγόριθμο L ένα γραμμικό αριθμό φορών μπορούμε να λύσουμε το γενικότερο πρόβλημα Extended Longest Path με εκθετικά μικρή πιθανότητα λάθους, ενώ παράλληλα να διατηρήσουμε το χρόνο τρεξίματος σε $O^*(4^k)$. □

5.2 H-Graph Edge Packing

Στη συνέχεια είναι ενδιαφέρον να δούμε τη μέθοδο Διαίρει και Χρωμάτιζε και σε κάποια άλλα γραφοθεωρητικά προβλήματα. Ένα από αυτά είναι και το H-Graph Edge Packing το οποίο ορίζεται ως εξής.

Ορισμός 5.3 *Στο H-Graph Edge Packing μας δίνονται γραφήματα $G = (V, E)$ και $H = (V[H], E[H])$ κι ένας ακέραιος αριθμός k. Το ερώτημα είναι αν υπάρχουν k κόπιες του H στο G που δεν έχουν καμία κοινή ακμή μεταξύ τους.*

Όπως κάναμε και στην προηγούμενη παράγραφο, θα δώσουμε πρώτα τον αλγόριθμο για το πρόβλημα και μετά θα περάσουμε στην ανάλυσή του. Πριν όμως το κάνουμε αυτό είναι σκόπιμο να αναφέρουμε ότι, με μικρές τροποποιήσεις, όλα όσα ακολουθούν μπορούν να χρησιμοποιηθούν και για το παρόμοιο πρόβλημα H-Graph Packing που ορίζεται ως εξής.

Ορισμός 5.4 *Στο H-Graph Packing μας δίνονται γραφήματα $G = (V, E)$ και $H = (V[H], E[H])$ κι ένας ακέραιος αριθμός k. Το ερώτημα είναι αν υπάρχουν k κόπιες του H στο G που δεν έχουν κανένα κοινό κόμβο μεταξύ τους.*

Algorithm 6 H-Graph Edge Packing (P)

Είσοδος: Ένα γράφημα $G = (V, E)$ και ένας ακέραιος k (η παράμετρος).
Έξοδος: Απάντηση ΝΑΙ ή ΟΧΙ.

```
if k ≤ 1 then
    if Υπάρχουν k ή περισσότερες κόπιες του H χωρίς κοινή ακμή then
        Απάντησε ΝΑΙ
    else
        Απάντησε ΟΧΙ
    end if
end if
for 3 · 2^((h−1)k) do
    Διάλεξε κάποιο E' ∈ 2^E με ομοιόμορφη πιθανότητα.
    G1 := (V, E'), G2 := (V, E − E')
    if P(G1, ⌈k/2⌉) & P(G2, ⌊k/2⌋) then
        Απάντησε ΝΑΙ
    end if
end for
Απάντησε ΟΧΙ
```

Λήμμα 5.2 *Ο Αλγόριθμος P λύνει το πρόβλημα H-Graph Edge Packing με πιθανότητα λάθους $1/4$ σε χρόνο $O^*(4^{(h-1)k})$, όπου $h = |E[H]|$.*

Απόδειξη. Υποθέτουμε ότι το γράφημα H έχει $h > 1$ ακμές. Στο for loop του αλγορίθμου P το E χωρίζεται σε δύο τμήματα, τα E' και $E - E'$. Ας πούμε ότι οι ακμές στο E' είναι μαύρες και οι ακμές στο $E - E'$ είναι άσπρες. Φιξάρουμε μια λύση, δηλαδή k κόπιες του H στο G οι οποίες δεν έχουν καμία κοινή ακμή μεταξύ τους. Για κάθε κόπια του H, η πιθανότητα ότι όλες οι ακμές της έχουν το ίδιο χρώμα είναι 2^{-h+1} (δηλαδή 2^{-h} για κάθε ένα χρώμα). Η πιθανότητα όλες οι κόπιες να είναι μονοχρωματικές είναι $2^{-(h-1)k}$ και η πιθανότητα να είναι ακριβώς $\lceil k/2 \rceil$ κόπιες μαύρες και $\lfloor k/2 \rfloor$ κόπιες άσπρες είναι τουλάχιστον $2^{-(h-1)k}/k$.

Έστω τώρα ότι p_k είναι η πιθανότητα με την οποία ο αλγόριθμος P δεν βρίσκει τη φιξαρισμένη λύση. Τότε σύμφωνα και με τα παραπάνω θα έχουμε

$$p_k \leq \left(1 - \frac{2^{-(h-1)k}}{k} + \frac{2^{-(h-1)k}}{k}(p_{\lceil k/2 \rceil} + p_{\lfloor k/2 \rfloor})\right)^{3 \cdot 2^{(h-1)k}} \tag{5.4}$$

δηλαδή με πιθανότητα $1 - \frac{2^{-(h-1)k}}{k}$ ο χρωματισμός είναι κακός και με πιθανότητα $\frac{2^{-(h-1)k}}{k}(p_{\lceil k/2 \rceil} + p_{\lfloor k/2 \rfloor})$ ο χρωματισμός είναι καλός αλλά ο αλγόριθμος αποτυγχάνει. Στη συνέχεια δείχνουμε επαγωγή στο k ότι $p_k \leq 1/4$. Προφανώς για $k \leq 1$ έχουμε ότι $p_k = 0$. Τώρα αν για $k > 1$ ισχύει για όλα μέχρι το k στο p_k θα έχουμε

$$p_k \leq \left(1 - \frac{2^{-(h-1)k}}{k} + \frac{2^{-(h-1)k}}{k}\right)^{3 \cdot 2^{(h-1)k}} \leq \left(1 - 2^{-(h-1)k-1}\right)^{3 \cdot 2^{(h-1)k}} < e^{-3/2}.$$

Η ανάλυση για το χρόνο που τρέχει ο αλγόριθμος είναι παρόμοια με αυτή του αλγορίθμου L στο λήμμα 5.2 και παίρνουμε τελικά το επιθυμητό αποτέλεσμα που είναι $O^*(4^{(h-1)k})$. $\square$

Θεώρημα 5.2 *Τα προβλήματα H-Graph Edge Packing και H-Graph Packing μπορούν να λυθούν με εκθετικά μικρή πιθανότητα λάθους σε χρόνο* $O^*(2^{2(h-1)k})$, *όπου* $h = |E[H]|$ *και* $h = |V[H]|$ *αντίστοιχα.*

Απόδειξη. Για το H-Graph Edge Packing απλά επαναλαμβάνουμε τον αλγόριθμο P ένα γραμμικό αριθμό φορών με σκοπό να μεγαλώσουμε την πιθανότητα επιτυχίας. Το H-Graph Packing μπορεί να λυθεί παρόμοια και, όπως είναι φυσικό, αυτή τη φορά χρωματίζουμε κόμβους αντί για ακμές. $\square$

Κεφάλαιο 6

Βελτιωμένοι αλγόριθμοι για προβλήματα μονοπατιού

6.1 Εισαγωγή

Στο κεφάλαιο αυτό θα αναφερθούμε σε ένα paper των Chen, Lu, Sze και Zhang [6] στο οποίο παρουσιάζονται κάποιοι νέοι βελτιωμένοι αλγόριθμοι για προβλήματα μονοπατιών (Path) και κατ΄ επέκταση για προβλήματα ταιριάσματος (Matching) και συσσώρευσης (Packing).

Στο paper αναπτύσσονται νέες και βελτιωμένες τεχνικές για πιθανοτικούς και ντετερμινιστικούς αλγόριθμους που λύνουν προβλήματα των παραπάνω κατηγοριών. Οι πιθανοτικοί αλγόριθμοι βασίζονται σε μια νέα διαίρει και βασίλευε (divide and conquer) τεχνική, που οδηγεί στους βελτιωμένους αλγόριθμους για αυτά τα προβλήματα. Πιο συγκεκριμένα, για το πρόβλημα k-Path ο πιθανοτικός αλγόριθμος τρέχει σε $O(4^k mk^{3.42})$ χρόνο και $O(nklogk+m)$ χώρο, βελτιώνοντας τους προηγούμενους καλύτερους πιθανοτικούς αλγόριθμους για το πρόβλημα που τρέχουν σε χρόνο $O(5.44^k km)$ και χώρο $O(2^k kn+m)$.

Επιπλέον, για να επιτευχθούν οι βελτιωμένοι ντετερμινιστικοί αλγόριθμοι, πετυχαίνουν ένα κατώτερο όριο $\Omega(2.718^k)$ κι ένα βελτιωμένο ανώτερο όριο $O(6.4^k n)$ στον αριθμό των χρωματισμών που χρησιμοποιούνται για να χρωματίσουν το γράφημα με k χρώματα. Αναπτύσουν δηλαδή μια νέα τεχνική χρωματισμού, χωρίς να χρησιμοποιούν τις k-τέλειες οικογένειες συναρτήσεων κατακερματισμού και, με τον τρόπο αυτό, οδηγούνται σε ντετερμινιστικό αλγόριθμο που λύνει το k-Path πρόβλημα σε χρόνο $O(12.8^k nm)$ βελτιώνοντας έτσι τον προηγούμενο καλύτερο ντετερμινιστικό αλγόριθμο για το πρόβλημα [2].

Πριν περάσουμε στην ανάλυση των αλγορίθμων είναι σημαντικό να σημειώσουμε το εξής. Όπως ήδη έχουμε παρατηρήσει από τα προηγούμενα κεφάλαια, πολλά NP-δύσκολα προβλήματα έχουν να κάνουν με την εύρεση ενός υποσυνόλου S_k με k στοιχεία σε ένα δεδομένο σύνολο S έτσι ώστε το υποσύνολο S_k να ικανοποιεί συγκεκριμένες απαιτήσεις. Οι διάφορες τεχνικές χρωματισμού ενός γραφήματος με k χρώματα που χρησιμοποιούνται φαίνεται να παρέχουν ένα ικανοποιητικό στήσιμο των στοιχείων του S έτσι ώστε όλα τα στοιχεία του υποσυνόλου S_k να έχουν διαφορετικά χρώματα. Το γεγονός αυτό μειώνει κατά πολύ το ψάξιμο για την εύρεση της λύσης του προβλήματος. Από αυτή την οπτική γωνία, η μελέτη των τεχνικών χρωματισμού με k χρώματα ενός γραφήματος φαίνεται να έχει πολύ ενδιαφέρον για την επίλυση πολλών NP-hard προβλημάτων.

6.2 Πιθανοτικός αλγόριθμος για το k-Path πρόβλημα

Το πρώτο αποτέλεσμα που δίνεται σε αυτή την εργασία είναι ένας πιθανοτικός αλγόριθμος για το k-Path πρόβλημα. Το k-Path πρόβλημα ορίζεται ως εξής.

Ορισμός 6.1 *Δεδομένου ενός γραφήματος $G = (V, E)$ κι ενός ακεραίου k είτε κατασκεύασε ένα απλό μονοπάτι μήκους $k - 1$ (k κόμβων) είτε ανάφερε ότι ένα τέτοιο μονοπάτι δεν υπάρχει.*

Πριν περάσουμε στην ανάλυση του αλγορίθμου, θα μιλήσουμε λίγο γενικά για τη νέα αυτή μέθοδο. Υποθέτοντας λοιπόν ότι ψάχνουμε για ένα υποσύνολο S_k με k στοιχεία ενός συνόλου S, όπως ακριβώς περιγράψαμε και στο τέλος της εισαγωγής, αυτό που κάνουμε πρώτα είναι να χωρίσουμε το υποσύνολο S_k σε δύο μισά ίσου μεγέθους και στη συνέχεια να ελέγξουμε αναδρομικά για ένα υποσύνολο με $k/2$ στοιχεία σε κάθε ένα από τα δύο μισά. Η μέθοδος αυτή παρά την απλότητά της οδηγεί σε βελτιωμένο πιθανοτικό αλγόριθμο για το πρόβλημα. Τα νέα αποτελέσματα θα παρουσιαστούν στο τέλος της παραγράφου.

Ο πιθανοτικός αλγόρισμος Find-Path(G', P', k) $[FP(G', P', k)]$ για το k-Path πρόβλημα φαίνεται στο τέλος της παραγράφου.

Στο σημείο αυτό είναι αναγκαίο να δώσουμε κάποιες πληροφορίες για την καλύτερη κατανόηση του αλγορίθμου. Ένα απλό μονοπάτι στο γράφημα $G = (V, E)$ είναι ένα (u, k)-path αν περιέχει ακριβώς k κόμβους και αν η μία άκρη του μονοπατιού είναι ο κόμβος u. Πιο συγκεκριμένα, για κάθε κόμβο u, ένα $(u, 1)$-path περιέχει ακριβώς ένα κόμβο, τον u. Όταν ο κόμβος u είναι τυχαίος, ένα (u, k)-path θα καλείται απλά ως k-path . Ο αλγόριθμος $FP(G', P', k)$ στο υπογράφημα G' και στο σύνολο P' των k'-paths , όπου κανένας κόμβος στα μονοπάτια του P' δεν είναι στο G', επιστρέφει τελικά ένα σύνολο P μονοπατιών το καθένα από τα οποία είναι μια σύνδεση ενός k'-path στο P' κι ενός k-path στο G'. Αν κανένα τέτοιο μονοπάτι δεν υπάρχει τότε ο αλγόριθμος επιστρέφει ένα κενό σύνολο. Δηλαδή, ο αλγόριθμος $FP(G', \emptyset, k)$ επιστρέφει ένα σύνολο από k-paths στο γράφημα G'.

Στη συνέχεια αποδεικνύουν το παρακάτω θεώρημα χρησιμοποιώντας τη σχέση ότι για κάθε ακέραιο $k > 1$ ισχύει $\lceil \log k \rceil = \lceil \log(\lceil k/2 \rceil) \rceil + 1$.

Θεώρημα 6.1 *Έστω ένα γράφημα $G = (V, E)$ με n κόμβους και m ακμές κι ένας ακέραιος $k \geq 1$, αν το γράφημα G περιέχει ένα (u, k)-path για ένα κόμβο u, τότε με πιθανότητα μεγαλύτερη του $1 - 1/e > 0.632$, το σύνολο P που επιστρέφεται από τον αλγόριθμο $FP(G, \emptyset, k)$ περιέχει ένα (u, k)-path . Ο αλγόριθμος αυτός τρέχει σε χρόνο $O(4^k m k^{3.42})$ και σε χώρο $O(nk \log k + m)$.*

Όπως είναι εύκολο να καταλάβουμε, χρησιμοποιώνται το θεώρημα 6.1 μπορούμε να αναπτύξουμε πιθανοτικούς αλγόριθμους με αυθαίρετα μικρό όριο λάθους για το k-Path πρόβλημα. Για παράδειγμα, για να πετύχουμε ένα όριο λάθους της τάξης του 0.0001 μπορούμε να τρέξουμε τον αλγόριθμο του θεωρήματος 6.1 t φορές, όπου το t ικανοποιεί τη σχέση $(1/e)^t \leq 0.0001$, κάτι που συμβαίνει για $t = 10$.

Algorithm 7 Find-Path(G', P', k) [$FP(G', P', k)$]

Είσοδος: Ένα υπογράφημα G' του G, ένα σύνολο P' με k'-paths στο G που δεν περιέχει κόμβους του G', ένας ακέραιος $k \geq 1$.
Έξοδος: Ένα σύνολο P μονοπατιών όπου το καθένα είναι μια κονκατενέισιον ενός k'-path στο P' κι ένα k-path στο G'.

$P = \emptyset$
if $k = 1$ **then**
  if $P' = \emptyset$ **then return** Όλα τα 1-path στο G'
  else
    for για κάθε (u, k')-path p στο G και κάθε κόμβο w στο G' **do**
      if (u, w) είναι ακμή του G **then**
        σύνδεσε τα p και w για να φτιάξεις ένα $(w, k'+1)$-path p'
        πρόσθεσε το p' στο P αν δεν υπάρχει τέτοιο μονοπάτι στο P
      end if
    end for
  end if
end if
return P
for $2.51 \cdot 2^k$ **do**
  Τυχαία χώρισε τους κόμβους του G' σε δύο μέρη V_L και V_R
  $G_L := (V_L, E'),\ G_R := (V_R, E - E')$
  $P_L = FP(G_L, P', \lceil k/2 \rceil)$
  if $P_L \neq \emptyset$ **then**
    $P_R = FP(G_R, P_L, k - \lceil k/2 \rceil)$
    for για κάθε $(u, k'+k)$-path p στο P_R **do**
      πρόσθεσε το p στο P αν δεν υπάρχει τέτοιο μονοπάτι στο P
    end for
  end if
end for
return P

6.3 Ανώτερα και κατώτερα όρια για την $\tau(n,k)$

Για να αναπτυχθούν βελτιωμένοι ντετερμινιστικοί αλγόριθμοι για τα προβλήματα μονοπατιού, το κλειδί είναι να μελετηθεί μεθοδικά και συστηματικά η πολυπλοκότητα των χρωματικών κωδικοποιήσεων (με k χρώματα). Να θυμηθούμε ότι ένας k-χρωματισμός ενός συνόλου S είναι μια συνάρτηση από το S επί του $\{0,\ldots,k-1\}$. Μια συλλογή $\mathcal{C}$ με k-χρωματισμούς του S είναι ένα σχήμα k-χρωματικής κωδικοποίησης του S αν για κάθε υποσύνολο S_k του S με k στοιχεία υπάρχει ένας χρωματισμός F στο $\mathcal{C}$ έτσι ώστε δεν υπάρχουν στοιχεία στο S_k που να είναι χρωματισμένα με το ίδιο χρώμα σύμφωνα με το χρωματισμό F.

Στη συνέχεια της παραγράφου θα δηλώνουμε με $\tau(n,k)$ το ελάχιστο μέγεθος ενός σχήματος χρωματικής κωδικοποίησης ενός συνόλου με n στοιχεία χρησιμοποιώντας k χρώματα. Αυτά που θα μελετηθούν είναι τα ανώτερα και κατώτερα όρια για τη συνάρτηση $\tau(n,k)$.

Οι συγγραφείς του paper αρχικά αναπτύσσουν ένα κατώτατο όριο κι αποδεικνύουν ότι $\tau(n,k) > 2.718^k$. Για τη βελτίωση του ανώτατου ορίου της συνάρτησης $\tau(n,k)$ προτείνουν μια διαδικασία κατακερματισμού τεσσάρων επιπέδων χρησιμοποιώντας διάφορες τεχνικές. Στο πρώτο επίπεδο χρησιμοποιείται μια τεχνική πυρηνοποίησης που είναι βασισμένη στη συνάρτηση κατακερματισμού που μελετάται στο [40]. Τα επόμενα δύο επίπεδα της διαδικασίας χρησιμοποιούν μια συνάρτηση κατακερματισμού που είναι σχεδόν βέλτιστη όσον αφορά στην ελαχιστοποίηση συγκρούσεων. Το τελευταίο επίπεδο στηρίζεται σε μια μη τετριμμένη αναδρομική φόρμουλα της $\tau(n,k)$ και σε προσεκτικές κατασκευές σχημάτων χρωματικής κωδικοποίησης με k χρώματα, για μικρές τιμές του k. Οι παραπάνω τεχνικές δίνουν ένα νέο σχήμα χρωματικής κωδικοποίησης με k χρώματα μεγέθους $O(6.4^k n)$. Το τελευταίο είναι πολύ καλύτερο από το προηγούμενο ανώτατο όριο για την $\tau(n,k)$, που ήταν μεγαλύτερο από 4000^k.

Ορισμός 6.2 *Ένας k-χρωματισμός ενός συνόλου S είναι μια συνάρτηση από το S στο Z_k. Μια συλλογή $\mathcal{F}$ από k-χρωματισμούς του S είναι ένα σχήμα χρωματικής κωδικοποίησης του S με k χρώματα αν για κάθε υποσύνολο W του S με k στοιχεία, υπάρχει ένας k-χρωματισμός f_W στο $\mathcal{F}$ που είναι αντιστοιχεία στο W. Το μέγεθος του σχήματος χρωματικής κωδικοποίησης $\mathcal{F}$ με k χρώματα είναι ίσο με τον αριθμό των k-χρωματισμών στο $\mathcal{F}$.*

Στον παραπάνω ορισμό με Z_k δηλώνουμε το σύνολο ακεραίων $\{0,1,\ldots,k-1\}$. Έστω τώρα ακέραιοι n,k με $n \geq k$. Σύμφωνα με τα παραπάνω το $\tau(n,k)$ είναι το ελάχιστο μέγεθος του σχήματος χρωματικής κωδικοποίησης με k χρώματα για το σύνολο Z_n. Στη συνέχεια αποδεικνύουν τα εξής.

Θεώρημα 6.2 *Αν το Z_{k^2} έχει ένα σχήμα χρωματικής κωδικοποίησης με k χρώματα μεγέθους r, τότε το Z_n έχει ένα σχήμα χρωματικής κωδικοποίησης με k χρώματα μεγέθους το πολύ $2nr$.*

Πόρισμα 6.1 $\tau(n,k) \leq 2n \cdot \tau(k^2,k)$.

Θεώρημα 6.3 *Για κάθε πραγματικό αριθμό $\epsilon > 0$, η τιμή της $\tau(n,k)$ είναι μεγαλύτερη από $(e-\epsilon)^k$ όταν το n είναι αρκετά μεγάλο.*

Μέχρι τώρα το καλύτερο ανώτερο όριο για την $\tau(n,k)$ ήταν της τάξης του $n(2^c)^k$, όπου $c \geq 12$, κάτι που είναι μεγαλύτερο από 4000^k αν αγνοήσουμε τους πολυωνυμικούς παράγοντες [2] . Από την άλλη πλευρά όμως, το Θεώρημα 6.3 δείχνει ότι υπάρχει μια σταθερά c_1, με $c_1 > 2.718$, τέτοια ώστε να ισχύει $\tau(n,k) = \Omega(c_1^k)$. Με τον τρόπο αυτό αποδεικνύουν το κατώτερο όριο. Στο σημείο αυτό όμως, αντιλαμβανόμαστε ότι ο κυρίαρχος παράγοντας του μεγέθους της $\tau(n,k)$ ενός σχήματος χρωματικής κωδικοποίησης με k χρώματα για το Z_k φαίνεται να είναι της μορφής c_0^k για μια σταθερά c_0, όπου $2.718 < c_0 < 4000$. Οπότε για να βρεθεί το ανώτερο όριο της $\tau(n,k)$ το ενδιαφέρον πρέπει να επικεντρωθεί στην εύρεση της ακριβούς τιμής της σταθεράς c_0.

Στη συνέχεια θα αναφερθούμε επιγραμματικά στην κατασκευή του σχήματος χρωματικής κωδικοποίησης που θα αναπτύξουν για την εύρεση του ανώτερου ορίου της $\tau(n,k)$. Αρχικά μελετούν κάποιες γενικές ιδιότητες που έχουν τα σχήματα χρωματικής κωδικοποίησης. Στη συνέχεια αναπτύσσουν το νέο σχήμα χρωματικής κωδικοποίησης με k χρώματα για το σύνολο Z_n και, εκμεταλλευόμενοι το Πόρισμα 6.1 , επικεντρώνονται στην περίπτωση όπου $n = k^2$. Ο αλγόριθμος χρωματισμού που χρησιμοποιούν σχετίζεται άμεσα με ένα σύνολο παραμέτρων που ικανοποιεί κάποιες πολύ συγκεκριμένες ιδιότητες. Αυτό που έχει σημασία εδώ είναι το παρακάτω θεώρημα.

Θεώρημα 6.4 *Ο συνολικός αριθμός των συνδυασμών των παραμέτρων που ικανοποιούν τις επιθυμητές ιδιότητες είναι φραγμένος από την* $O(6.383^k k^{\log k-4})$ *κι αυτοί οι συνδυασμοί μπορούν να αριθμηθούν συστηματικά.*

Έπειτα αποδεικνύουν μια σειρά θεωρημάτων και πορισμάτων, κυρίως τεχνικού και κατασκευαστικού περιεχομένου, και καταλήγουν στο εξής.

Θεώρημα 6.5 *Για κάθε υποσύνολο W με k στοιχεία στο Z_n, υπάρχει ένας συνδυασμός παραμέτρων που ικανοποιεί τις επιθυμητές ιδιότητες όπου ο αλγόριθμος χρωματισμού παράγει ένα k-χρωματισμό για το Z_n που είναι αντιστοιχία στο W.*

Συνδυάζοντας τα Θεωρήματα 6.4 , 6.5 και τρέχοντας τον αλγόριθμο χρωματισμού με όλους τους πιθανούς συνδυασμούς των παραμέτρων που ικανοποιούν τις επιθυμητές ιδιότητες, παίρνουμε ένα σχήμα χρωματικής κωδικοποίησης με k χρώματα μεγέθους $O(6.383^k k^{\log k-4})$. Ο χρόνος τερματισμού του αλγόριθμου χρωματισμού είναι $O(k^2)$ για κάθε χρωματισμό. Αν θυμηθούμε ότι μελετάμε την περίπτωση όπου ισχύει $n = k^2$ έχουμε το παρακάτω θεώρημα.

Θεώρημα 6.6 *Αν το k διαιρείται από το 4, τότε ισχύει $\tau(k^2,k) \leq O(6.383^k k^{\log k-4})$ κι ένα σχήμα χρωματικής κωδικοποίησης με k χρώματα μεγέθους $O(6.383^k k^{\log k-4})$ για το σύνολο Z_{k^2} μπορεί να κατασκευαστεί σε χρόνο $O(6.383^k k^{\log k-2})$.*

Προφανώς όμως, εμάς μας ενδιαφέρει η περίπτωση όπου τα n και k είναι τυχαία. Χρησιμοποιώντας το Θεώρημα 6.2 μπορούμε να επεκτείνουμε το Θεώρημα 6.6 και να καταλήξουμε στο παρακάτω πολύ σημαντικό αποτέλεσμα που μας δίνει και το ανώτερο όριο για την $\tau(n,k)$ για γενικές τιμές των n και k.

Θεώρημα 6.7 *Για οποιουσδήποτε ακεραίους n και k, όπου $n \geq k$, $\tau(n,k) = O(6.4^k n)$ κι ένα σχήμα χρωματικής κωδικοποίησης με k χρώματα μεγέθους $O(6.4^k n)$ για το σύνολο Z_n μπορεί να κατασκευαστεί σε χρόνο $O(6.4^k n)$.*

6.4 Ντετερμινιστικός αλγόριθμος για το k-Path πρόβλημα

Αυτό που έχει επιτευχθεί μέχρι τώρα είναι η κατασκευή ενός σχήματος χρωματικής κωδικοποίησης που έχει σημαντικά βελτιωμένο μέγεθος σε σχέση με τα ήδη γνωστά, σύμφωνα με το Θεώρημα 6.7 . Χρησιμοποιώντας αυτό το νέο σχήμα μπορούμε αμέσως να πάρουμε έναν αρκετά βελτιωμένο ντετερμινιστικό αλγόριθμο για το k-Path πρόβλημα.

Έστω λοιπόν ένα γράφημα G με n κόμβους και m ακμές. Υποθέτουμε ότι το G περιέχει ένα μονοπάτι P μήκους $k-1$ κι ότι έχουμε ένα χρωματισμό k χρωμάτων των κόμβων του G που είναι μονομορφισμός ως προς τους k κόμβους του μονοπατιού P. Από προηγούμενο κεφάλαιο γνωρίζουμε ότι ένας απλός αλγόριθμους δυναμικού προγραμματισμού βρίσκει σε αυτό το χρωματισμένο γράφημα ένα πανχρωματικό μονοπάτι μήκους $k-1$, αν αυτό υπάρχει, σε χρόνο $O(c^k n^{O(1)})$.

Έστω τώρα, λοιπόν, ότι $\mathcal{F}$ είναι το σχήμα χρωματικής κωδικοποίησης k χρωμάτων που χρησιμοποιούμε για το χρωματισμό του γραφήματος και το οποίο προέρχεται από το Θεώρημα 6.7 . Τότε ένας από τους χρωματισμούς στο $\mathcal{F}$ θα είναι μονομορφισμός ως προς τους k κόμβους του μονοπατιού P μήκους $k-1$ αν αυτό υπάρχει. Οπότε αν χρωματίσουμε τους κόμβους του γραφήματος με κάθε έναν από τους χρωματισμούς στο $\mathcal{F}$ κι εφαρμόσουμε τον παραπάνω αλγόριθμο δυναμικού προγραμματισμού σε κάθε ένα από τα χρωματισμένα γραφήματα με k χρώματα, τότε θα μπορέσουμε να κατασκευάσουμε ένα πανχρωματικό μονοπάτι στο γράφημα G, αν βέβαια αυτό υπάρχει.

Συνδυάζοντας όλα τα παραπάνω, παίρνουμε έναν ντετερμινιστικό αλγόριθμο για το k-Path πρόβλημα που τρέχει σε χρόνο $O(2^k m \cdot 6.4^k n)$ δηλαδή σε $O(12.8^k nm)$. Είναι ξεκάθαρο ότι ο χρόνος αυτός είναι σημαντικά βελτιωμένος από το χρόνο $O(c^k n^{O(1)})$ του αλγορίθμου του Κεφαλαίου 3, που χρησιμοποιεί τις k-τέλειες οικογένειες συναρτήσεων κατακερματισμού.

6.5 Περισσότερα αποτελέσματα και παρατηρήσεις

Κλείνοντας το Κεφάλαιο είναι ενδιαφέρον να αναφέρουμε ότι, χρησιμοποιώντας τις παραπάνω τεχνικές, μπορούμε να πάρουμε βελτιωμένους πιθανοτικούς και ντετερμινιστικούς αλγόριθμους και για άλλα προβλήματα όπως είναι τα 3-D Matcing και 3-Set Packing.

Ορισμός 6.3 *Το πρόβλημα 3-D Matching ορίζεται ως εξής. Δεδομένου ενός συνόλου S από τριάδες κι ενός ακεραίου k, είτε βρες ένα υποσύνολο S_k με k τριάδες που δεν έχουν κοινά στοιχεία είτε ανάφερε ότι τέτοιο υποσύνολο δεν υπάρχει.*

Ορισμός 6.4 *Το πρόβλημα 3-Set Packing ορίζεται ως εξής. Δεδομένης μιας συλλογής από τρισύνολα κι ενός ακεραίου k, είτε βρες μια υποσυλλογή από k τρισύνολα που δεν έχουν κοινά σημεία είτε ανάφερε ότι τέτοια υποσυλλογή δεν υπάρχει.*

Έτσι λοιπόν καταλήγουμε στο επόμενο πολύ σημαντικό Θεώρημα.

Θεώρημα 6.8 *Τα προβλήματα 3-D Matcing και 3-Set Packing μπορούν να λύθούν μϵ πιθανοτικούς αλγορίθμους σϵ χρόνο $O(2.52^{3k}n)$ και χώρο $O(nk \log k)$.*

Να σημειώσουμε ότι οι προηγούμενοι αμέσως καλύτεροι αλγόριθμοι για τα προβλήματα [29] τρέχουν σε χρόνο $O(10.88^{3k}n^{O(1)})$ και χώρο $O(2^{3k}n^{O(1)})$, όπου ο χώρος είναι εκθετικός ως προς το k.

Με την ίδια λογική, παίρνουμε και νέους βελτιωμένους ντετερμινιστικούς αλγόριθμους για αυτά τα προβλήματα. Έτσι λοιπόν, αν χρησιμοποιήσουμε το σχήμα χρωματικής κωδικοποίησης μεγέθους $O(6.4^{3k}n)$ του Θεωρήματος 6.7 , χρησιμοποιώντας δηλαδή $3k$ χρώματα, μπορούμε να πάρουμε ντετερμινιστικό αλγόριθμο για το 3-D Matching που τρέχει σε χρόνο $O(12.8^{3k}n^2)$. Ο ίδιος αλγόριθμος με ελάχιστες τροποποιήσεις δίνει αλγόριθμο για το 3-Set Packing που τρέχει κι αυτός σε χρόνο $O(12.8^{3k}n^2)$. Να αναφέρουμε ότι οι προηγούμενοι καλύτεροι ντετερμινιστικοί αλγόριθμοι για αυτά τα προβλήματα τρέχουν σε χρόνο $O(c^k n^{O(1)})$, όπου $c > 32000$ [29]. Κλείνοντας τις συγκρίσεις, είναι σκόπιμο να αναφέρουμε ότι το νέο αυτό σχήμα χρωματικής κωδικοποίησης, συνδυαζόμενο με άλλες τεχνικές που έχουν αναπτυχθεί για την επίλυση αυτών των προβλημάτων, δίνει και πάλι νέα καλύτερα αποτελέσματα.

Στο μέλλον είναι πιθανή η περαιτέρω βελτίωση του ανώτερου ορίου για το μέγεθος του $\tau(n, k)$ δηλαδή του σχήματος χρωματικής κωδικοποίησης με k χρώματα του συνόλου Z_n. Μάλιστα οι συγγραφείς του paper υποστηρίζουν ότι, με μια πιο προσεκτική ανάλυση, το μέγεθος μπορεί να μειωθεί σε $O(6.1^k n^{O(1)})$ [30] . Από την άλλη πλευρά όμως, το κατώτερο όριο όπως προκύπτει από το Θεώρημα 6.3 έχει θέσει ένα ισχυρό φράγμα για περαιτέρω βελτιώσεις προς αυτή την κατεύθυνση.

Κεφάλαιο 7

Επίλογος

7.1 Γενική Συζήτηση

Η συνειδητοποίηση ότι όλα τα προβλήματα απόφασης δεν έχουν πάντα αποδοτικούς αλγορίθμους (εκτός κι αν ισχύει P=NP) έγινε στα τέλη της δεκαετίας του εξήντα [8, 13, 26, 27] . Από τότε οι επιστήμονες της Θεωρητικής Πληροφορικής έχουν κατηγοριοποιήσει τα προβλήματα σε προσβάσιμα δηλαδή σε αυτά που είναι γνωστό ότι έχουν αλγόριθμο πολυωνυμικού χρόνου και σε δύσβατα δηλαδή σε αυτά που δεν έχουν.

Δυστυχώς, πολλά ή μάλλον τα περισσότερα από τα προβλήματα που αντιμετωπίζονται σε καθημερινή βάση ανήκουν στη δεύτερη κατηγορία. Η ανάγκη να λυθούν αυτά τα προβλήματα έχει παράγει έναν πολύ μεγάλο αριθμό ιδεών ως προς το πώς πρέπει να αντιμετωπιστούν. Μία από τις νεότερες ιδέες είναι αυτή της Παραμετρικής Πολυπλοκότητας.

Η ουσία της παραμετρικής πολυπλοκότητας βρίσκεται στο γεγονός ότι τα περισσότερα προβλήματα δεν είναι τόσο γενικά όσο η αφηρημένη περιγραφή του προβλήματος που βρίσκουμε συνήθως σε εκτενείς συλλογές προβλημάτων [23, 24]. Θα μπορούσαμε, για παράδειγμα, να γνωρίζουμε κάτι για τα στιγμιότυπα εισόδου όπως ας πούμε τα γραφήματα να έχουν φραγμένο δεντροπλάτος ή θα μπορούσαμε να έχουμε κάποιες συγκεκριμένες απαιτήσεις για τη λύση, όπως για παράδειγμα μια λύση είναι ενδιαφέρουσα αν δεν είναι πολύ μεγάλη. Επιπλέον πληροφορίες όπως οι προηγούμενες είναι δυνατόν να βελτιώσουν την επιλυσιμότητα ενός προβλήματος και οδηγούν στην ιδέα της παραμέτρου.

Στην κλασσική θεωρία πολυπλοκότητας τα NP-πλήρη προβλήματα είναι ισοδύναμα μεταξύ τους ως προς τη δυσκολία. Η εισαγωγή της παραμέτρου αλλάζει ακριβώς αυτό το γεγονός. Μπορεί να δειχθεί ότι μια παραμετροποίηση ενός προβλήματος αυτής της κατηγορίας μπορεί να έχει έναν αποδοτικό ως προς το χρόνο αλγόριθμο, αρκεί η παράμετρος να είναι μικρή. Για άλλα προβλήματα, η παραμετροποίηση δεν παίζει κανένα ρόλο ως προς τη βελτίωση της κατάστασης. Ο διαχωρισμός αυτός των κλασσικών προβλημάτων που ανήκουν στο NP είναι από μόνος του ενδιαφέρων κι έχει γίνει αρκετή δουλειά στη θεωρητική πληροφορική [10–12, 19, 20, 25].

Στην εργασία αυτή ασχοληθήκαμε με τις σημαντικότερες τεχνικές σχεδιασμού παραμετρικών αλγορίθμων που έχουν να κάνουν με τη μέθοδο της χρωματικής κωδικοποίησης. Υπάρχουν όμως κι άλλες τεχνικές αντιμετώπισης αυτών των προβ-

λημάτων. Ο κάθε ενδιαφερόμενος μπορεί να απευθυνθεί, για μια ουσιαστική εισαγωγή στο χώρο, σε μία εκ των δύο σημαντικών δουλειών που ακολουθούν πάνω στο σχεδιασμό παραμετρικών αλγορίθμων. Η πρώτη είναι των Downey και Fellows το "Parameter Complexity" [10] και η άλλη του Rolf Niedermeier το "Invitation to Fixed-Parameter Algorithms" [33]. Το πρώτο βιβλίο βγήκε σχετικά νωρίς και για το λόγο αυτό δεν περιέχει όλες τις νεότερες ιδέες. Επίσης, δεν ασχολείται ιδιαίτερα με το να κατηγοριοποιήσει συγκεκριμένες τεχνικές. Παρόλα αυτά παραμένει ακόμη μια αξιόλογη δουλειά που δίνει όλα τα σημαντικά εφόδια για τη συνέχεια. Από την άλλη, η δουλειά του Niedermeier ασχολείται αποκλειστικά με τις αλγοριθμικές τεχνικές. Ξεχωρίζει τις "Bounded Search Trees" και "Kernelization" ως τις δύο βασικές τεχνικές και δίνει λεπτομερείς εξηγήσεις και για τις δύο. Στη συνέχεια αναφέρει κι επεξηγεί ως άλλες αλγοριθμικές τεχνικές τις Integer Linear Programming, Dynamic Programming, Color Coding και Tree Decompositions
.

Πάνω σε αυτές τις κατηγορίες έχουν αναπτυχθεί τα τελευταία χρόνια πολλές διαφορετικά ονομαζόμενες τεχνικές που μπορούν να εφαρμοστούν για το σχεδιασμό παραμετρικών αλγορίθμων. Ενδεικτικά παραπέμπουμε στα [2,7,9,10,15,17,33, 34,36–38]. Έτσι λοιπόν θα μπορούσαμε να πούμε ότι δημιουργείται μια λανθασμένη γενική εικόνα, σε ό,τι αφορά τις διάφορες τεχνικές. Ο αριθμός των πραγματικά ξεχωριστών διαφορετικών τεχνικών είναι αρκετά μικρότερος. Πολλές από αυτές βασίζονται στην ίδια βασική ιδέα. Προφανώς, οι λεπτομέρειες σε κάθε περίπτωση θα ποικίλλουν από πρόβλημα σε πρόβλημα.

7.2 Συμπεράσματα και ανοιχτά προβλήματα

Η τεχνική της χρωματικής κωδικοποίησης δείχνει να παρέχει μια νέα σπουδαία προσέγγιση για το σχεδιασμό αποδοτικών αλγορίθμων που αντιμετωπίζουν NP-δύσκολα προβλήματα. Πιο συγκεκριμένα, στα προβλήματα που αφορούν στην εύρεση k συγκεκριμένων στοιχείων που έχουν κάποια ιδιότητα από ένα σύνολο n στοιχείων. Η προπαρασκευαστική διαδικασία τού να χρωματίζεις τα n στοιχεία με k χρώματα, έτσι ώστε τα k στοιχεία που ψάχνουμε να είναι χρωματισμένα με k διαφορετικά χρώματα (πανχρωματικά), φαίνεται να μειώνει σημαντικά το χώρο ψαξίματος για την εύρεση των αλγορίθμων που να λύνουν το εκάστοτε πρόβλημα [5].

Για παράδειγμα, ας υποθέσουμε ότι έχουμε χρωματίσει τους n κόμβους ενός γραφήματος G με k χρώματα έτσι ώστε οι k κόμβοι μια κλίκας μεγέθους k να είναι χρωματισμένη με διαφορετικά χρώματα. Τότε, το να βρούμε την κλίκα μεγέθους k μπορεί να γίνει εύκολα σε χρόνο $O((n/k)^k)$. Βλέπουμε ότι ο χρόνος αυτός είναι αρκετά πιο γρήγορος από το να ψάχνουμε την κλίκα μεγέθους k σε ένα γράφημα που δεν έχουμε χρωματίσει.

Τα τελευταία χρόνια σε πολλές εργασίες έχουν παρουσιαστεί νέες τεχνικές που έχουν ως βάση το χρωματισμό των γραφημάτων για το σχεδιασμό αποδοτικότερων αλγορίθμων για συγκεκριμένα προβλήματα. Παρόλα αυτά σε ένα μεγάλο αριθμό εργασιών αναφέρεται ότι δεν έχουν εκμεταλλευθεί όλες οι δυνατότητες αυτών των τεχνικών με αποτέλεσμα να υπάρχει ανοιχτό το ενδεχόμενο για ακόμη καλύτερα αποτελέσματα.

Ως συμπλήρωση των παραπάνω, κλείνοντας, θα αναφέρουμε δύο από τα πιο γνωστά ανοιχτά προβλήματα που είχαν θέσει οι Alon, Yuster και Zwick [2] και τα οποία αναζητούν ακόμη την επίλυσή τους.

◊ Υπάρχει αλγόριθμος (ντετερμινιστικός ή πιθανοτικός) πολυωνυμικού χρόνου που να αποφασίζει αν ένα δοσμένο γράφημα $G = (V, E)$ περιέχει ένα μονοπάτι μήκους $\log^2 V$;

◊ Είναι το πρόβλημα απόφασης για το αν ένα δοσμένο γράφημα $G = (V, E)$ περιέχει ένα τρίγωνο το ίδιο δύσκολο με τον Boolean πολλαπλασιαμό δύο $V \times V$ πινάκων ;

Βιβλιογραφία

[1] N. Alon και M. Naor. Derandomization, witnesses for boolean matrix multiplication and construction of perfect hash functions. *Algorithmica*, 1994.

[2] N. Alon, R. Yuster και U. Zwick. Color-coding. *Journal of the Association for Computing Machinery*, σελίδες 4:844–856, 1995.

[3] B. Bollobás. *Combinatorics.* Cambridge University Press, Cambridge, 1986.

[4] B. Bollobás. *Extremal graph theory.* Dover Publications Inc., Mineola, NY, 2004. Επανέκδοση του αρχικού 1978.

[5] J. Chen, X. Huang, I. Kanj και G. Xia. Linear FPT reductions and computational lower bounds. Στο *Proceedings of 36th ACM Symposium on Theory of Computing*, σελίδες 212–221. 2004.

[6] J. Chen, S. Lu, S. Sze και F. Zhang. Improved algorithms for path, matching and packing problems. 2006.

[7] B. Chor, M.R. Fellows και D. Juedes. Linear kernels in linear time or how to save k colors in $o(n^2)$ steps. Στο *Proceedings of WG2004*, Lecture Notes in Computer Science. 2004.

[8] S. Cook. The complexity of theorem proving procedures. Στο *Proceedings of 3rd annual ACM Symposium on Theory of Computing*, Association for Computing Machinery, σελίδες 151–158. New York, 1971.

[9] F. Dehne, M.R. Fellows, F. Rosamond και P. Shaw. Greedy localization and iterative compression and modeled crown reductions. Στο *Proceedings of IWPEC2004*, τόμος 3162 στο *Lecture Notes in Computer Science*, σελίδες 271–281. 2004.

[10] R. G. Downey και M. R. Fellows. *Parameterized complexity.* Monographs in Computer Science. Springer-Verlag, New York, 1999.

[11] R.G. Downey και M.R. Fellows. Fixed-parameter tractability and completeness. Στο *Proceedings of the Twenty-first Manitoba Conference on Numerical Mathematics and Computing (Winnipeg, MB, 1991)*, τόμος 87, σελίδες 161–178, 1992.

[12] R.G. Downey και M.R. Fellows. Parameterized complexity after (almost) ten years: review and open questions. Στο *Combinatorics, computation & logic '99 (Auckland)*, τόμος 21 στο *Aust. Comput. Sci. Commun.*, σελίδες 1–33. Springer, Singapore, 1999.

[13] J. Edmonds. Paths, trees and flowers. *Can J. Math.*, 17(3):449–467, 1965.

[14] M. Fellows, F.V. Fomin, D. Lokshtanov, F. Rosamond, S. Saurabh, S. Szeider και C. Thomassen. On the complexity of some colorful problems parameterized by treewidth. Στο *Combinatorial optimization and applications*, τόμος 4616 στο *Lecture Notes in Computer Science*, σελίδες 366–377. Springer, Berlin, 2007.

[15] M.R. Fellows. Blow-ups win/wins and crown rules : Some new directions in fpt. Στο *Proceedings of WG 2003*, τόμος 2880 στο *Lecture Notes in Computer Science*, σελίδες 1–12. Springer-Verlag, Berlin, 2003.

[16] M.R. Fellows. Faster fixed-parameter tractable algorithms for matching and packing problems. Στο *Proceedings of 12th ESA*, τόμος 3221 στο *Lecture Notes in Computer Science*, σελίδες 311–322. Springer, Berlin, 2004.

[17] M.R. Fellows, C. McMartin, F. Rosamond και U. Stege. Coordinatized kernels and catalytic reductions. *Foundations of Software Technology and Theoretical Computer Science*, 2000.

[18] H. Fernau. Parametric duality: Kernel sizes and algorithmics. Τεχνική Αναφορά υπ. αρίθμ. TP04-027, Electronic Colloquium on Computational Complexity, 2003.

[19] J. Flum και M. Grohe. Fixed-parameter tractability, definability, and model-checking. *SIAM J. Comput.*, 31(1):113–145, 2001.

[20] J. Flum και M. Grohe. Describing parameterized comlplexity classes. Στο *Proceedings of 19th STACS*, τόμος 2285 στο *Lecture Notes in Computer Science*, σελίδες 113–145. 2002.

[21] J. Flum και M. Grohe. *Parameterized complexity theory.* Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2006.

[22] M.L. Fredman, J. Komlos και E. Szemeredi. Storing a sparse table with o(1) worst case access time. *Journal of the Association for Computing Machinery*, σελίδες 31:538–544, 1984.

[23] M.R. Garey και D.S. Johnson. *Computers and intractability. A guide to the theory of NP-completeness.* W. H. Freeman and Co., San Francisco, Calif., 1979.

[24] M.R. Garey, D.S. Johnson και L. Stockmeyer. Some simplified np-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.

[25] M. Grohe. Descriptive and parameterized comlplexity. Στο *13th CLS*, τόμος 1683 στο *Lecture Notes in Computer Science*, σελίδες 14–31. 1999.

[26] R. M. Karp. On the computational complexity of combinatorial problems. *Networks*, 5(1):45–68, 1975.

[27] R.M. Karp. Reducibility among combinatorial problems. Complexity of Computer Computations, σελίδες 85–103. Plenum Press, New York, 1972.

[28] J. Kneis, D. Molle, S. Richter και P. Rossmanith. Divide and color. τόμος 4721 στο *Lecture Notes in Computer Science*, σελίδες 58–67. Springer, Berlin, 2006.

[29] I. Koutis. A faster parameterized algorithm for set packing. *Information processing letters*, (94):7–9, 2005.

[30] Y. Liu, S. Lu, J. Chen και S. Sze. Greedy localization, color-coding and dynamic programming: Improved algorithms for matching and packing problems. Στο *Proceedings of 2nd IWPEC*, τόμος 4169 στο *Lecture Notes in Computer Science*. Springer, 2006.

[31] D.W. Matula. A min–max theorem for graphs with application to graph coloring. *SIAM Reviews*, 10:481–482, 1968.

[32] D.W. Matula και L.L. Beck. Smallest-last ordering and clustering and graph coloring algorithms. *Journal of the Association for Computing Machinery*, σελίδες 30:417–427, 1983.

[33] R. Niedermeier. *Invitation to fixed-parameter algorithms*, τόμος 31 στο *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, 2006.

[34] R. Niedermeier και P. Rossmanith. A general method to speed up fixed parameter algorithms. τόμος 73 στο *Information processing letters*, σελίδες 125–129. 2000.

[35] C.H. Papadimitriou και M. Yannakakis. On limited nondeterminism and the complexity of the V–C dimension. *Journal of Computer and System Sciences*, 53:161–170, 1996.

[36] E. Prieto. *Systematic Kernelization in FPT algorithm design.* Διδακτορική Διατριβή , 2007.

[37] E. Prieto και C. Sloper. Either/or : Using vertex cover structure in designing fpt algorithms - the case of k-internal spanning tree. Στο *Proceedings of WADS 2003*, τόμος 2748 στο *Lecture Notes in Computer Science*, σελίδες 465–483. 2003.

[38] E. Prieto και C. Sloper. Looking at the stars. Στο *Proceedings of IWPEC 2004*, τόμος 3162 στο *Lecture Notes in Computer Science*, σελίδες 138–149. 2004.

[39] N. Robertson και P. Seymour. Graph minors II Algorithmic aspects of tree-width. *Journal of Algorithms*, σελίδες 7:309–322, 1986.

[40] J.P. Schmidt και A. Siegel. The spatial complexity of oblivious k-probe hash functions. *SIAM Journal on Computing*, σελίδες 19(5):775–786, 1990.