ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΣΤΗΝ ΕΠΙΣΤΗΜΗ ΥΠΟΛΟΓΙΣΤΩΝ

Διπλωματική Εργασία

Μεταπτυχιακού Διπλώματος Ειδίκευσης

Προβλήματα Βελτιστοποίησης σε Γράφους Διαστημάτων και Συγγενείς

Αγγελική Τάραμα

Επιβλέπων: Ιωάννης Μήλης

ΑΘΗΝΑ, ΙΟΥΝΙΟΣ 2007

Περιεχόμενα

Πρόλογος

Η Αλγοριθμική Θεωρία Γράφων είναι ένας από τους πιο σημαντικούς κλάδους της Θεωρητικής Πληροφορικής. Στην παρούσα εργασία, θα επικεντρωθούμε σε μια ιδιαίτερα ενδιαφέρουσα κλάση γράφων, στους γράφους διαστημάτων, καθώς και σε συγγενικούς τους γράφους, όπως είναι οι χορδικοί γράφοι και οι γράφοι κυκλικών τόξων. Ένας γράφος διαστημάτων είναι ουσιαστικά ο γράφος αλληλεπικάλυψης ενός συνόλου διαστημάτων στην ευθεία των ακεραίων.

Οι γράφοι διαστημάτων και οι συγγενείς τους έχουν πολλές πρακτικές εφαρμογές. Ενδεικτικά, στην αρχαιολογία, οι χρονικές ζώνες, στις οποίες ανήκει το κάθε αντικείμενο, αναπαρίστανται από διαστήματα, οπότε οι γράφοι διστημάτων χρησιμοποιούνται για την εύρεση της χρονολογικής σειράς των αντικειμένων [1]. Επιπλέον, οι γράφοι διαστημάτων είναι ιδιαίτερα χρήσιμοι στο χρονοπρογραμματισμό εργασιών. Δεδομένου ενός πλήθους εργασιών με διαφορετικούς χρόνους εκκίνησης και επεξεργασίας, το ζητούμενο είναι η εύρεση μιας σειράς για την επεξεργασία τους, η οποία θα πληρεί συγκεκριμένες προϋποθέσεις [2]. Τέλος, αυτή η κατηγορία γράφων χρησιμοποιείται στον κλάδο της βιολογίας, όπου ορισμένες ακολουθίες του DNA μοντελοποιούνται ως διαστήματα και το πρόβλημα περιλαμβάνει την κατασκευή DNA χαρτών [3], καθώς και την εύρεση γονιδίων [4]. Άλλες εφαρμογές των γράφων διαστημάτων και των συγγενών τους συναντώνται σε τεχνικές δέσμευσης πόρων στον τομέα των συστημάτων και των δικτύων υπολογιστών [5], [6], [7], [8].

Ο βασικός στόχος αυτής της εργασίας είναι η μελέτη εγγενώς δύσκολων, στη γενική τους περίπτωση, προβλημάτων, τα οποία, λόγω ορισμένων σημαντικών ιδιοτήτων που χαρακτηρίζουν τους γράφους διαστημάτων και τους συγγενείς τους, επιλύονται ευκολότερα σε αυτούς. Για το λόγο αυτό, η επίλυση αυτών των προβλημάτων σε αυτές τις ειδικές κατηγορίες γράφων έχει γίνει αντικείμενο εκτεταμένης έρευνας και έχει υπάρξει μια πληθώρα συνεισφορών στην περιοχή αυτή. Έτσι, ενώ ορισμένα από τα προβλήματα που μελετάμε είναι $\mathcal{NP}$-πλήρη για γενικούς γράφους, έχουν προταθεί αλγόριθμοι μέσω των οποίων τα ίδια προβλήματα επιλύονται πολυωνυμικά σε γράφους διαστημάτων ή σε συγγενείς τους. Για άλλα προβλήματα, τα οποία παραμένουν $\mathcal{NP}$-πλήρη και στις κατηγορίες γράφους που μελετάμε, υπάρχει ένα ευρύ φάσμα προσεγγιστικών αλγορίθμων για την προσέγγιση της βέλτιστης λύσης τους. Σε αυτή την εργασία αναπτύσσουμε το μεγαλύτερο μέρος του θεωρητικού αυτού έργου που αφορά την επίλυση προβλημάτων βελτιστοποίησης σε γράφους διαστημάτων και συγγενείς τους.

Τα περιεχόμενα αυτής της εργασίας αναπτύσσονται ως εξής. Στο Κεφάλαιο 1 εισάγουμε ορισμένες βασικές έννοιες της Θεωρίας Γράφων και δίνουμε τους αυστηρούς ορισμούς των κλάσεων γράφων που θα χρησιμοποιήσουμε.

Στο Κεφάλαιο 2 αρχικά ορίζουμε και αποδεικνύουμε βασικές ιδιότητες των γράφων διαστημάτων και των συγγενών τους, καθώς και το πώς συνδέονται αυτές οι κλάσεις γράφων μεταξύ τους. Στη συνέχεια παρουσιάζουμε αλγόριθμους για την επίλυση κλασικών προβλημάτων (μέγιστη κλίκα, χρωματισμός κορυφών, ανεξάρτητο σύνολο, κάλυψη από κλίκες) σε χορδικούς γράφους.

Στο Κεφάλαιο 3 εξετάζουμε το πρόβλημα του χρωματισμού ακμών ενός γράφου διαστημάτων, δηλαδή την ανάθεση χρωμάτων στις ακμές του, ούτως ώστε οι γειτονικές ακμές να λαμβάνουν διαφορετικά χρώματα.

Στο Κεφάλαιο 4 εξετάζουμε το πρόβλημα του χρωματισμού κορυφών με περιορισμούς πληθυκότητας σε γράφους διαστημάτων, δηλαδή την ανάθεση χρωμάτων στις κορυφές του γράφου, έτσι ώστε οι κορυφές που γειτνιάζουν να έχουν διαφορετικό χρώμα και κάθε χρώμα να χρωματίζει έως ένα συγκεκριμένο αριθμό κορυφών.

Στο Κεφάλαιο 5 μελετάμε το χρωματισμό κορυφών σε γράφους κυκλικών τόξων και στο Κεφάλαιο 6 μελετάμε τον άμεσο χρωματισμό κορυφών σε γράφους διαστημάτων και κυκλικών τόξων. Ο άμεσος χρωματισμός κορυφών διαφοροποιείται από τον απλό χρωματισμό κορυφών στο ότι δε γνωρίζει εκ των προτέρων την τοπολογία του γράφου.

Στο Κεφάλαιο 7 παρουσιάζουμε το πρόβλημα του μέγιστου χρωματισμού κορυφών σε γράφους διαστημάτων. Σε κάθε κορυφή του γράφου αποδίδουμε κάποιο βάρος και αναζητάμε ένα χρωματισμό έτσι ώστε το άθροισμα των μέγιστων βαρών των κορυφών του κάθε χρώματος να είναι ελάχιστο.

Τέλος, στο Κεφάλαιο 8 εξετάζουμε το πρόβλημα του χρονοπρογραμματισμού ενός συνόλου εργασιών με διάταξη διαστημάτων και στόχο την ελαχιστοποίηση του χρόνου ολοκλήρωσης των εργασιών σε ένα σύνολο ισοδύναμων μεταξύ τους επεξεργαστών.

Κεφάλαιο 1

Εισαγωγή

1.1 Βασικές έννοιες θεωρίας γράφων

Στις ενότητες αυτού του κεφαλαίου θα αναφέρουμε βασικές έννοιες της Θεωρίας Γράφων, ούτως ώστε να γίνουν ευχολότερα κατανοητά όσα ακολουθούν στα επόμενα κεφάλαια.

Αρχικά, ορίζουμε ότι ένας γράφος $G = (V, E)$ είναι ένα σύνολο κορυφών (ή κόμβων) που συνδέεται από ένα σύνολο ακμών. Το σύνολο των κορυφών αποτελείται από $|V| = n$ κορυφές, οι οποίες θεωρούνται διατεταγμένες με τυχαίο τρόπο. Το σύνολο των ακμών αποτελείται από $|E|$ ακμές, οι οποίες μπορεί να έχουν κατεύθυνση ή όχι. Αν οι ακμές έχουν κατεύθυνση, τότε λέμε ότι ο γράφος είναι κατευθυνόμενος (**directed**) και η ακμή που ξεκινάει από την κορυφή u και καταλήγει στην κορυφή v, $u, v \in V$, συμβολίζεται με το διατεταγμένο ζεύγος (u, v). Αν οι ακμές δεν έχουν κατεύθυνση, τότε λέμε ότι ο γράφος είναι μη κατευθυνόμενος (**undirected**) και η ακμή που συνδέει τις κορυφές $u, v \in V$ συμβολίζεται με το μη διατεταγμένο ζεύγος (u, v). Πλήρης (**complete**) ονομάζεται ο γράφος που έχει το μέγιστο δυνατό αριθμό ακμών, δηλαδή ο γράφος στον οποίο οποιοδήποτε ζεύγος κορυφών συνδέεται μέσω μιας ακμής.

Σε ένα γράφο $G = (V, E)$ ορίζουμε την έννοια του κύκλου. Κύκλος είναι μια διαδρομή (μέσω ακμών του G), η οποία ξεκινάει και καταλήγει στην ίδια κορυφή περνώντας το πολύ μία φορά από τις υπόλοιπες κορυφές, για οποιαδήποτε κορυφή του κύκλου. Ένας κύκλος, ο οποίος διασχίζει k κορυφές συμβολίζεται με C_k.

Ένας επαγώμενος (**induced**) υπογράφος $G[V']$ του $G = (V, E)$ είναι ένας γράφος με σύνολο κορυφών το $V' \subseteq V$ και σύνολο ακμών το υποσύνολο των ακμών του E που συνδέουν τις κορυφές του V'. Σημειώνουμε ότι μια κλάση γράφων $\mathcal{G}$ είναι κληρονομική (**hereditary**), αν για οποιονδήποτε γράφο $G \in \mathcal{G}$ κάθε (επαγώμενος) υπογράφος του ανήκει στην ίδια κλάση με τον G.

Ο βαθμός $d(v)$ μιας κορυφής v ενός μη κατευθυνόμενου γράφου είναι ο αριθμός των ακμών που προσπίπτουν σε αυτήν. Μέγιστος βαθμός, $\Delta(G)$, ενός μη κατευθυνόμενου γράφου G είναι ο μεγαλύτερος από τους βαθμούς των κορυφών του.

Σε έναν κατευθυνόμενο γράφο ορίζουμε τα εξής:

Ορισμός 1.1
Έστω κατευθυνόμενος γράφος $G = (V, E)$. Για κάθε $u \in V$, ορίζουμε:

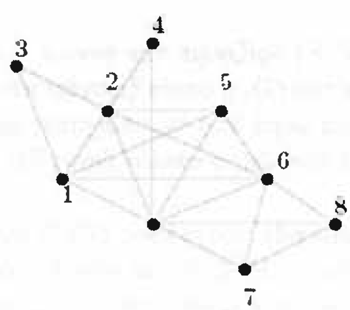
- *ως βαθμό εισόδου* (**indegree**) *και συμβολίζουμε με $d_G^-(u)$ τον αριθμό των προς τα μέσα ακμών που προσπίπτουν στην u και*
- *ως βαθμό εξόδου* (**outdegree**) *και συμβολίζουμε με $d_G^+(u)$ τον αριθμό των προς τα έξω ακμών που προσπίπτουν στην u.*

Αν ένας κατευθυνόμενος γράφος G είναι ακυκλικός (**acyclic**), τότε μπορούμε να ορίσουμε μια διάταξη για τις κορυφές του, έτσι ώστε κάθε κορυφή να βρίσκεται αριστερότερα από όλες τις κορυφές προς τις οποίες έχει ακμές. Αυτή η διάταξη ονομάζεται τοπολογική διάταξη (**topological sorting**) των κορυφών του G.

1.2 Γράφοι διαστημάτων και συγγενείς

Σε αυτή την παράγραφο θα ορίσουμε κάποιες βασικές κατηγορίες γράφων, τις οποίες θα χρησιμοποιήσουμε στη συνέχεια.

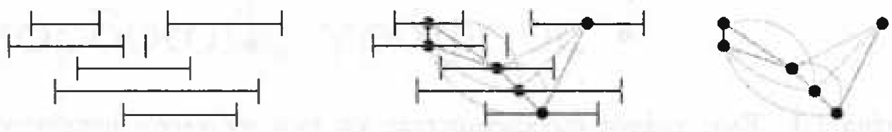
Ορισμός 1.2 (Χορδικοί γράφοι - Chordal graphs)
Ένας γράφος είναι χορδικός αν δεν περιέχει επαγώμενο κύκλο C_k για $k \geq 4$. Χορδή ονομάζεται κάθε ακμή που συνδέει δύο μη διαδοχικές κορυφές ενός κύκλου. Ισοδύναμα, ένας γράφος είναι χορδικός αν κάθε C_k ($k \geq 4$) του γράφου έχει χορδή.



Σχήμα 1.1: Ένας χορδικός γράφος.

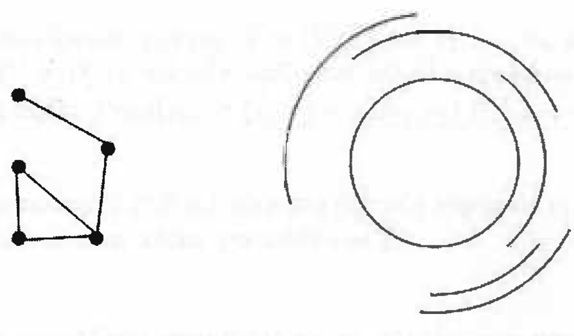
Αν θεωρήσουμε ως διάστημα (**interval**) κάθε ευθύγραμμο τμήμα στον άξονα των ακεραίων, τότε:

Ορισμός 1.3 (Γράφοι διαστημάτων - Interval graphs)
Δεδομένου ενός συνόλου διαστημάτων, ορίζουμε το γράφο διαστημάτων $G = (V, E)$, *ο οποίος έχει μια κορυφή* $u \in V$ *για κάθε διάστημα* I_u *και μια ακμή* $(u, v) \in E$ *ανν τα διαστήματα* I_u *και* I_v *αλληλεπικαλύπτονται, δηλαδή* $(u, v) \in E \Leftrightarrow I_u \cap I_v \neq \emptyset$. *Ισοδύναμα, ένας γράφος είναι γράφος διαστημάτων αν είναι ο γράφος αλληλεπικάλυψης ενός συνόλου διαστημάτων.*



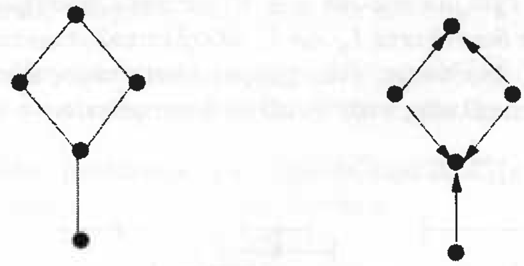
Σχήμα 1.2: Ένα σύνολο διαστημάτων και ο γράφος που ορίζεται από αυτό.

Ορισμός 1.4 (Γράφοι κυκλικών τόξων - Circular arc graphs)
Ένας γράφος ονομάζεται γράφος κυκλικών τόξων αν μπορούμε να αντιστοιχίσουμε κάθε κορυφή του με ένα τόξο στην περιφέρεια ενός κύκλου, έτσι ώστε δύο τόξα να αλληλεπικαλύπτονται ανν οι αντίστοιχες κορυφές γειτνιάζουν. Ισοδύναμα, ένας γράφος κυκλικών τόξων είναι ο γράφος αλληλεπικάλυψης ενός συνόλου τόξων.



Σχήμα 1.3: Ένας γράφος κυκλικών τόξων και το σύνολο των τόξων, από το οποίο αναπαρίσταται.

Ορισμός 1.5 (Γράφοι συγκρισιμότητας - Comparability graphs)
Ένας γράφος είναι γράφος συγκρισιμότητας αν έχει έναν ακυκλικό μεταβατικό προσανατολισμό.



Σχήμα 1.4: Ένας γράφος συγκρισιμότητας και ένας ακυκλικός μεταβατικός προσανατολισμός του.

Ορισμός 1.6 (Διμερείς γράφοι - Bipartite graphs)
Ένας γράφος $G = (V, E)$ ονομάζεται διμερής αν το σύνολο των κορυφών του V μπορεί να διασπαστεί σε δύο ανεξάρτητα μεταξύ τους σύνολα X και Y ($V = X \cup Y$), έτσι ώστε κάθε ακμή του E να συνδέει πάντα μια κορυφή του X με μια κορυφή του Y.

Κεφάλαιο 2

Κλασικά προβλήματα σε χορδικούς γράφους

2.1 Εισαγωγή

Στο κεφάλαιο αυτό θα παρουσιάσουμε αρχικά ορισμούς, λήμματα και θεωρήματα που χαρακτηρίζουν και συνδέουν μεταξύ τους τους χορδικούς γράφους και τους γράφους διαστημάτων. Στη συνέχεια θα παρουσιάσουμε αλγόριθμους για κλασικά προβλήματα σε γράφους, τα οποία λόγω κάποιων ιδιοτήτων που χαρακτηρίζουν τους χορδικούς γράφους, επιλύονται ευκολότερα σε αυτούς [9], [10]. Ειδικότερα θα μελετήσουμε τα προβλήματα της μέγιστης κλίκας, του χρωματισμού κορυφών, του ανεξάρτητου συνόλου και της κάλυψης από κλίκες.

2.2 Χορδικοί γράφοι και γράφοι διαστημάτων

Έστω μη κατευθυνόμενος γράφος $G = (V, E)$ και $v_1, \ldots, v_n$ μια διάταξη των κορυφών του. Γι' αυτή τη διάταξη ορίζονται σύνολα προγόνων $Pred(v_i)$ για κάθε κορυφή ως εξής: $Pred(v_i) = \{v_j : j < i, (v_i, v_j) \in E\}$, $i = 1, \ldots, n$.

Ορισμός 2.1
Τέλεια διάταξη απαλοιφής (τ.δ.α.) είναι μια διάταξη κορυφών $v_1, \ldots, v_n$, τέτοια ώστε το $Pred(v_i)$ να είναι κλίκα για κάθε $i = 1, \ldots, n$.

Ορισμός 2.2
(i) Μια κορυφή ονομάζεται **simplicial** *αν το σύνολο των γειτόνων της σχηματίζει κλίκα. Διαχωριστήρας* **(separator)** *είναι ένας διαμερισμός $V = S \cup A \cup B$ των κορυφών ενός γράφου, τέτοιος ώστε να μην υπάρχουν ακμές μεταξύ των A, B.*
(ii) Δεδομένων δύο μη γειτονικών κορυφών a, b, ένας (a, b)-διαχωριστήρας **((a,b)-seperator)** *είναι ένας διαχωριστήρας $V = S \cup A \cup B$, τέτοιος ώστε $a \in A$ και $b \in B$.*

*(iii) Ελαχιστικός (a,b)-διαχωριστήρας (**minimal (a,b)-seperator**) είναι ένας (a,b)-διαχωριστήρας $V = S \cup A \cup B$, τέτοιος ώστε κανένα υποσύνολο του S να μην είναι (a,b)-διαχωριστήρας.*

Θεώρημα 2.3
Ένας γράφος είναι χορδικός ανν κάθε ελαχιστικός διαχωριστήρας του είναι κλίκα.

Απόδειξη. $\Rightarrow$: Έστω G χορδικός γράφος, a,b δύο μη γειτονικές κορυφές του, S ένας ελαχιστικός (a,b)-διαχωριστήρας και x,y δυο τυχαίες κορυφές στο S. Ο υπογράφος $G[V-S]$ έχει τουλάχιστον δύο συνδεδεμένες συνιστώσες, έστω $G[A]$ και $G[B]$, οι οποίες περιέχουν αντίστοιχα τις a,b. Έστω p_1 το συντομότερο μονοπάτι ανάμεσα στις x,y μέσω κορυφών που ανήκουν στο A και αντίστοιχα p_2 μέσω κορυφών που ανήκουν στο B. Συνδυάζοντας τα p_1, p_2 δημιουργείται C_k με $k \geq 4$. Εφόσον ο G είναι χορδικός, κάθε κύκλος πρέπει να έχει χορδή. Αφού δεν υπάρχουν ακμές μεταξύ των κορυφών του A και του B, η (x,y) είναι ακμή του G. Άρα, το S είναι κλίκα.
$\Leftarrow$: Έστω G ένας γράφος, στον οποίο κάθε ελαχιστικός διαχωριστήρας είναι κλίκα. Θεωρούμε ότι ο γράφος δεν είναι χορδικός, δηλαδή περιέχει κύκλο $w, x, y, z_1, \ldots, z_k, w, k \geq 1$ χωρίς χορδή. Κάθε ελαχιστικός (w,y)-διαχωριστήρας πρέπει να περιέχει την x και τουλάχιστον μια z_i, $1 \leq i \leq k$ και επομένως η (w, z_i) είναι ακμή του G. □

Λήμμα 2.4
*Ένας χορδικός γράφος είτε είναι πλήρης είτε έχει τουλάχιστον δύο μη γειτονικές **simplicial** κορυφές.*

Απόδειξη. Έστω μη πλήρης χορδικός γράφος G. Θα αποδείξουμε ότι έχει τουλάχιστον δύο μη γειτονικές **simplicial** κορυφές. Η απόδειξη θα γίνει με επαγωγή στον αριθμό n των κορυφών. Για $n = 2$, ο G έχει δύο απομονωμένες κορυφές, οι οποίες είναι προφανώς **simplicial** . Για $n > 2$, υποθέτουμε ότι το λήμμα ισχύει για γράφους με λιγότερες από n κορυφές. Έστω a, b δύο μη γειτονικές κορυφές του G και S ένας ελαχιστικός (a,b)-διαχωριστήρας. Ο υπογράφος $G[V-S]$ έχει τουλάχιστον δύο συνδεδεμένες συνιστώσες, έστω $G[A]$ και $G[B]$, οι οποίες περιέχουν αντίστοιχα τις a,b. Ο $G[A \cup S]$ είναι ένας χορδικός γράφος με λιγότερες από n κορυφές και επομένως είτε είναι πλήρης (οπότε κάθε κορυφή του A είναι **simplicial**) είτε έχει τουλάχιστον δύο μη γειτονικές **simplicial** κορυφές, μία εκ των οποίων ανήκει στο A εφόσον το S είναι κλίκα (Θεώρημα 2.3). Αφού το A δε γειτνιάζει με κορυφές εκτός του συνόλου $A \cup S$, όλες οι **simplicial** κορυφές του $G[A \cup S]$ που ανήκουν στο A θα είναι επίσης **simplicial** κορυφές του G. Άρα, ο G έχει τουλάχιστον μια **simplicial** κορυφή που ανήκει στο A και ομοίως τουλάχιστον μια που ανήκει στο B. □

Θεώρημα 2.5
Ένας γράφος είναι χορδικός ανν έχει μια τέλεια διάταξη απαλοιφής.

Απόδειξη. Για το ευθύ, έστω χορδικός γράφος G με n κορυφές. Υποθέτουμε ότι το θεώρημα ισχύει για πλήθος κορυφών $< n$. Αν v είναι μια **simplicial** κορυφή του G, τότε ο γράφος $G(V - v)$ έχει μια τ.δ.α. β (από το βήμα της επαγωγής). Διατάσσοντας την v στο τέλος της β, προκύπτει μια τ.δ.α. για το G. Για το αντίστροφο, έστω $v_1, v_2, \ldots, v_n$ μια τ.δ.α. του G. Υποθέτουμε ότι ο G δεν είναι χορδικός και συνεπώς περιέχει $C_k, k \geq 4$ χωρίς χορδή. Έστω v η κορυφή του C_k που εμφανίζεται τελευταία στην τ.δ.α. Οι δύο γείτονές της στον C_k θα είναι και πρόγονοί της στην τ.δ.α., οι οποίοι θα πρέπει να σχηματίζουν κλίκα. Εντούτοις, ο κύκλος C_k στον οποίο ανήκουν δεν έχει χορδή. Άτοπο. □

Οι χορδικοί γράφοι είναι ακριβώς οι γράφοι που έχουν τέλειες διατάξεις απαλοιφής. Λόγω αυτής της ιδιότητας, μια πληθώρα εγγενώς δύσκολων προβλημάτων επιλύεται ευκολότερα σε χορδικούς γράφους, καθώς και σε υποσύνολα αυτών, όπως είναι οι γράφοι διαστημάτων.

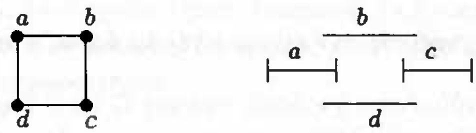
Θεώρημα 2.6
Κάθε γράφος διαστημάτων είναι χορδικός.

Απόδειξη. Έστω ένα σύνολο διαστημάτων. Σύμφωνα με το αριστερό τους άκρο, μπορούμε να διατάξουμε τις αντίστοιχες κορυφές. Έστω κορυφή v_i και s_i η αρχή του αντίστοιχου διαστήματος. Το σύνολο $Pred(v_i)$ των προγόνων της περιλαμβάνει όλες τις κορυφές, των οποίων τα διαστήματα αρχίζουν πριν το s_i, αλλά δεν τερματίζουν πριν από αυτό. Όλα αυτά τα διαστήματα αλληλεπικαλύπτονται (διότι περιέχουν το σημείο s_i) και επομένως το $Pred(v_i)$ είναι κλίκα. Συνεπώς, η διάταξη των κορυφών είναι μια τ.δ.α. και άρα ο γράφος είναι χορδικός (Θεώρημα 2.5). □

Θεώρημα 2.7
Τα παρακάτω είναι ισοδύναμα:
(1) Ο G είναι ένας γράφος διαστημάτων.
(2) Ο G δεν περιέχει C_4 και ο $\overline{G}$ είναι ένας γράφος συγκρισιμότητας.
(3) Μπορούμε να διατάξουμε τις μεγιστικές κλίκες του G, έτσι ώστε για κάθε κορυφή του γράφου οι κλίκες που την περιέχουν να εμφανίζονται διαδοχικά στην εν λόγω διάταξη.

Απόδειξη. (1) $\Rightarrow$ (2) Ένας γράφος, ο οποίος περιέχει C_4 δεν είναι γράφος διαστημάτων.

Ας υποθέσουμε ότι ο C_4 του σχήματος 1.5 μπορεί να αναπαρασταθεί από ένα σύνολο διαστημάτων. Η κορυφή a δε γειτνιάζει με την c και συνεπώς τα αντίστοιχα διαστήματα I_a και I_c δεν αλληλεπικαλύπτονται. Χωρίς βλάβη της γενικότητας, ας θεωρήσουμε ότι το I_a βρίσκεται στα αριστερά του I_c. Η b

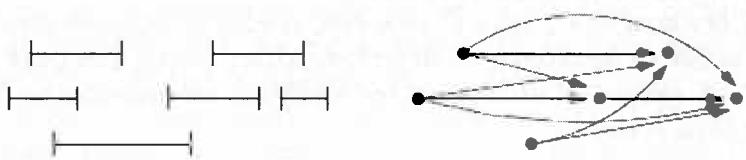


Σχήμα 2.1: Ο C_4 δεν είναι γράφος διαστημάτων.

γειτνιάζει με τις a και c και άρα το I_b θα πρέπει να καλύπτει ολόκληρη την περιοχή ανάμεσα στα I_a και I_c. Ομοίως για το διάστημα I_d της κορυφής d. Κατά συνέπεια, τα I_b και I_d θα πρέπει να αλληλεπικαλύπτονται. Άτοπο.
Αν ο G είναι ένας γράφος διαστημάτων, τότε ο $\overline{G}$ είναι ένας γράφος συγκρισιμότητας. Έστω u, v δύο μη γειτονικές κορυφές του G (άρα γειτνιάζουν στο $\overline{G}$). Τα διαστήματα I_u και I_v δεν παρουσιάζουν επικαλύψεις και επομένως είτε το I_u βρίσκεται στα αριστερά του I_v είτε το αντίθετο. Ορίζουμε έναν προσανατολισμό για τις ακμές του $\overline{G}$ ως εξής: για κάθε ζεύγος κορυφών u, v με $(u, v) \notin E(G)$, κατηύθυνε την ακμή ως $u \to v$, αν το I_u βρίσκεται στα αριστερά του I_v και ως $v \to u$ διαφορετικά (Σχήμα 2.2). Αυτός ο προσανατολισμός των ακμών του $\overline{G}$ είναι:

- ακυκλικός, δηλαδή δεν περιέχει κατευθυνόμενο κύκλο. Αυτό ισχύει, διότι κάθε ακμή προσανοτολίζεται από ένα διάστημα στα αριστερά προς ένα διάστημα στα δεξιά και ποτέ το αντίστροφο.
- μεταβατικός, δηλαδή αν $u \to v$ και $v \to w$, τότε και $u \to w$. Αυτό ισχύει, γιατί αν το I_u βρίσκεται στα αριστερά του I_v και το I_v στα αριστερά του I_w, τότε το I_u είναι επίσης στα αριστερά του I_w.

Άρα, ο συμπληρωματικός ενός γράφου διαστημάτων έχει έναν ακυκλικό και μεταβατικό προσανατολισμό ακμών, δηλαδή είναι ένας γράφος συγκρισιμότητας.



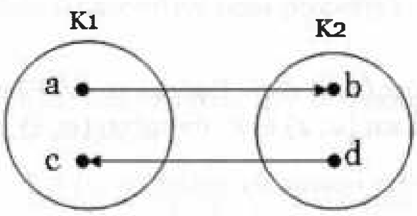
Σχήμα 2.2: Ένας προσανατολισμός των ακμών του συμπληρωματικού ενός γράφου διαστημάτων.

(2) $\Rightarrow$ (3) Για την απόδειξη αυτής της συνεπαγωγής, θα χρησιμοποιήσουμε το ακόλουθο λήμμα.

Λήμμα 2.8
Έστω K_1, K_2 δύο μη κενές μεγιστικές κλίκες του G και (V, F) ένας ακυκλικός μεταβατικός προσανατολισμός των ακμών του $\overline{G}$. Τότε:

a. *υπάρχει τόξο μεταξύ των K_1, K_2 που ανήκει στο F και*

b. *όλα τα τόξα μεταξύ των K_1, K_2 έχουν την ίδια κατεύθυνση.*



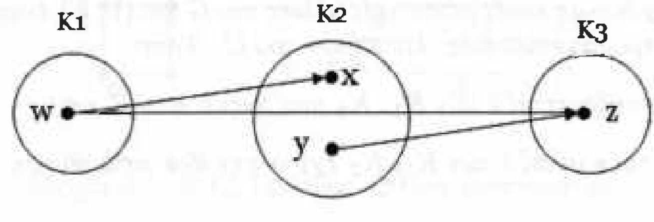
Σχήμα 2.3: Παράδειγμα για το λήμμα.

Απόδειξη. a. Αν δεν ήταν αληθές, τότε το σύνολο $B = K_1 \cup K_2$ θα ήταν κλίκα του G. Άτοπο, αφού τα K_1, K_2 είναι μεγιστικές κλίκες στο G.
b. Υποθέτουμε ότι η K_1 περιέχει τις κορυφές a, c, η K_2 τις b, d και $(a, b), (d, c) \in F$, όπως φαίνεται στο σχήμα 2.3. Θέλουμε να δείξουμε ότι αυτό δεν είναι εφικτό. Οι a, c είναι διακριτές, διότι αν $a \equiv c$, τότε $(d, b) \in F$ (λόγω της μεταβατικότητας του (V, F)). Άτοπο, διότι το σύνολο K_2 είναι κλίκα του G. Ομοίως για τις b, d. Επίσης, οι ακμές (a, d) και (b, c) δε δύναται να ανήκουν ταυτόχρονα στο E, διότι ο G δεν περιέχει C_4. Χωρίς βλάβη της γενικότητας, υποθέτουμε ότι $(a, d) \notin E$. Τότε,

- είτε $(a, d) \in F$, οπότε λόγω της μεταβατικής ιδιότητας συνεπάγεται ότι $(a, c) \in F$,
- είτε $(d, a) \in F$, οπότε ομοίως $(d, b) \in F$.

Άτοπο. □

Επιστρέφοντας στην απόδειξη του θεωρήματος, ορίζουμε μια σχέση σύγκρισης για τις κλίκες του G ως εξής: για δύο μεγιστικές κλίκες K_1 και K_2, θα ισχύει $K_1 < K_2$ ανν υπάρχει ακμή μεταξύ τους με κατεύθυνση από την K_1 προς την K_2. Θα αποδείξουμε ότι η παραπάνω σχέση είναι μεταβατική και συνεπώς διατάσσει πλήρως τις κλίκες του G. Έστω ότι $K_1 < K_2$ και $K_2 < K_3$. Δηλαδή $\exists (w, x) \in F : w \in K_1,\ x \in K_2$ και $\exists (y, z) \in F : y \in K_2,\ z \in K_3$, όπως φαίνεται στο σχήμα 2.4. Θα δείξουμε ότι $K_1 < K_3$. Διακρίνουμε τις εξής τρεις περιπτώσεις.



Σχήμα 2.4: Παράδειγμα για το θεώρημα.

1. Αν $(x,z) \notin E$, τότε $(x,z) \in F$. Εφόσον το (V,F) είναι ένας μεταβατικός προσανατολισμός και $(w,x) \in F$, θα πρέπει $(w,z) \in F$ και άρα $K_1 < K_3$.
2. Αν $(w,y) \notin E$, τότε ομοίως θα πρέπει $(w,z) \in F$ και άρα $K_1 < K_3$.
3. Αν $(x,z),(w,y) \in E$, τότε θα πρέπει $(w,z) \notin E$ (εφόσον ο G δεν περιέχει C_4). Άρα, $(w,z) \in F$ και επομένως $K_1 < K_3$.

Άρα, η σχέση σύγκρισης που ορίσαμε διατάσσει πλήρως τις κλίκες του G. Πράγματι, έστω $x \in V$ και $K_i < K_j < K_k$, τέτοια ώστε $x \in K_i$, $x \in K_k$ και $x \notin K_j$. Τότε:

$$(x \notin K_j) \Rightarrow \exists y \in K_j : (x,y) \notin E$$
$$(K_i < K_j) \Rightarrow (x,y) \in F$$
$$(K_j < K_k) \Rightarrow (y,x) \in F$$

Άτοπο.

(3) $\Rightarrow$ (1) Έστω $K_1, K_2, \ldots, K_k$ μια πλήρης διάταξη των μεγιστικών κλικών του G. Για κάθε κορυφή v ορίζουμε το διάστημα $I_v = [min\{i : v \in K_i\}, max\{j : v \in K_j\}]$. Θα δείξουμε ότι αυτά τα διαστήματα αναπαριστούν ένα γράφο διαστημάτων. Έστω (v,w) μια ακμή που περιέχεται στην κλίκα K_h, δηλαδή $v,w \in K_h$. Επομένως, τα I_v, I_w επικαλύπτονται, διότι περιέχουν αμφότερα τον ακέραιο h. Αντίστροφα, υποθέτουμε ότι τα I_v, I_w επικαλύπτονται. Έστω ότι περιέχουν αμφότερα τον ακέραιο h. Εφόσον η διάταξη των κλικών είναι διαδοχική, έπεται ότι οι κορυφές v και w περιέχονται στην K_h. Άρα, $(v,w) \in E$. □

Πόρισμα 2.9
Ένας γράφος G είναι γράφος διαστημάτων ανν ο G είναι χορδικός και ο $\overline{G}$ είναι γράφος συγκρισιμότητας.

Απόδειξη. Αν ο G είναι γράφος διαστημάτων, τότε είναι χορδικός (Θεώρημα 2.6) και ο $\overline{G}$ είναι γράφος συγκρισιμότητας (Θεώρημα 2.7). Το αντίστροφο ισχύει από το Θεώρημα 2.7, δεδομένου ότι οι χορδικοί γράφοι δεν περιέχουν C_4. □

Σημειώνουμε ότι ο πίνακας κλικών ενός γράφου είναι ένας πίνακας M με στοιχεία τα $0, 1$, τέτοιος ώστε οι γραμμές του να αντιστοιχούν στις μεγιστικές κλίκες του γράφου, οι στήλες του στις κορυφές του και $M_{ij} = 1$ ανν η v_j ανήκει στην i-οστή μεγιστική κλίκα. Από το Θεώρημα 2.7, αν οι γραμμές είναι διατεταγμένες σύμφωνα με τη σχέση σύγκρισης των κλικών, τότε οι μονάδες κάθε στήλης θα εμφανίζονται συνεχόμενα. Αυτή η ιδιότητα ονομάζεται ιδιότητα των διαδοχικών μονάδων (**consecutive ones property**) .

Θεώρημα 2.10
Ένας γράφος είναι γράφος διαστημάτων ανν ο πίνακας κλικών του έχει την ιδιότητα των διαδοχικών μονάδων.

Απόδειξη. Είναι άμεση απόρροια της τρίτης ισοδυναμίας του Θεωρήματος 2.7. □

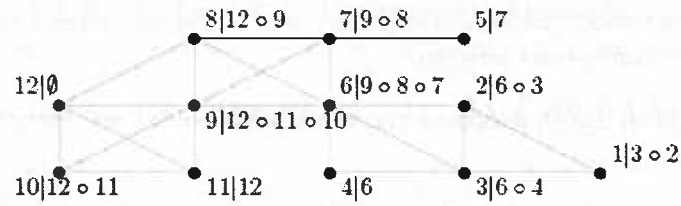
Στη συνέχεια θα μελετήσουμε προβλήματα σε χορδικούς γράφους, δηλαδή σε γράφους που έχουν μια τ.δ.α.. Αρχικά, θα παρουσιάσουμε αλγόριθμους για την κατασκευή μιας τ.δ.α. [11], καθώς και για την εύρεση των προγόνων της κάθε κορυφής. Στη συνέχεια θα μελετήσουμε το πρόβλημα της μέγιστης κλίκας (**maximum clique**), του ελάχιστου χρωματισμού κορυφών (**minimum vertex coloring**), του μέγιστου ανεξάρτητου συνόλου (**maximum independent set**) και της ελάχιστης κάλυψης από κλίκες (**minimum covering by cliques**) **kai ja parousi'asoume tous ant'istoiqous alg'orijmous pou pr'oteine o Fanica Gavril** στο [9]. Σημειώνουμε ότι οι αλγόριθμοι που θα παρουσιάσουμε για τα παραπάνω προβλήματα εφαρμόζονται ομοίως και για γράφους διαστημάτων, δεδομένου ότι κάθε γράφος διαστημάτων είναι και χορδικός (Θεώρημα 2.6).

2.3 Τέλεια διάταξη απαλοιφής και εύρεση προγόνων

Ένας μη πλήρης χορδικός γράφος $G = (V, E)$ έχει τουλάχιστον δύο μη γειτονικές **simplicial** κορυφές (Θεώρημα 2.4) και κάθε επαγώμενος υπογράφος του είναι επίσης χορδικός. Επομένως, επιλέγοντας επανειλημμένα μια **simplicial** κορυφή και διαγράφοντάς την από το γράφο, μπορούμε να κατασκευάσουμε μια τ.δ.α.. Σε αυτή την ιδέα βασίζεται ο Αλγόριθμος **LexBFS**, ο οποίος κατασκευάζει μια τ.δ.α. σε γραμμικό χρόνο [11]. Στον Αλγόριθμο **LexBFS**, μια ετικέτα L_a είναι λεξικογραφικά μεγαλύτερη από μια άλλη L_b, αν η L_a εμφανίζεται μετά την L_b σε έναν τηλεφωνικό κατάλογο. Ορίζουμε ότι η ετικέτα n είναι μικρότερη από την n ο $(n-1)$. Η μικρότερη δυνατή ετικέτα είναι η κενή.

Αλγόριθμος LexBFS

1. for $i = 1, \ldots, |V|$ do
2. Ανάθεσε στην κορυφή v_i την ετικέτα $L(v_i) = \emptyset$;
3. for $i = |V|, \ldots, 1$ do
4. Επέλεξε ως v_i την κορυφή που δεν έχει επιλεγεί ακόμα και έχει τη λεξικογραφικά μεγαλύτερη ετικέτα από όλες τις μη επιλεγμένες κορυφές;
5. for κάθε μη επιλεγμένη κορυφή $w \in \Gamma(v_i)$
6. Θέσε $L(w) = L(w) \circ i$;



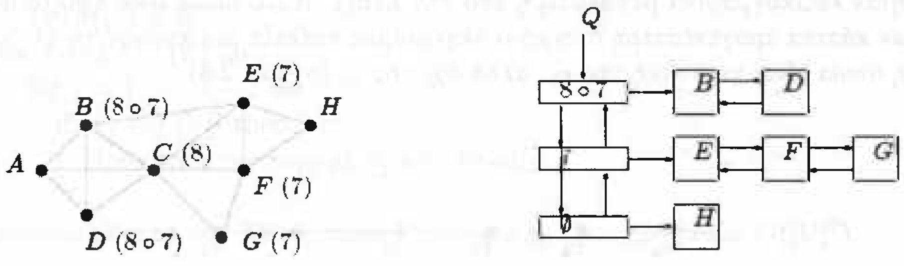
Σχήμα 2.5: Παράδειγμα εφαρμογής του Αλγόριθμου LexBFS σε γράφο. Όταν μια κορυφή έχει την ετικέτα $i|j$ σημαίνει ότι επιλέχτηκε ως v_i και εκείνη τη στιγμή είχε την ετικέτα j.

Για να υλοποιήσουμε αποτελεσματικά τον Αλγόριθμο LexBFS, θα χρησιμοποιήσουμε μια λίστα από κουβάδες (list of buckets) Q. Κάθε κορυφή του γράφου που βρίσκεται στη Q ανήκει σε μια λίστα (κουβά) P_l. Η λίστα P_l περιέχει όλες τις κορυφές v με ετικέτα $L(v) = l$. Η Q είναι ταξινομημένη κατά φθίνουσα λεξικογραφική σειρά και δεν περιέχει ποτέ άδειο κουβά (σχήμα 2.6).

Η διαδικασία αρχικοποιείται θέτοντας την ετικέτα κάθε κορυφής v ως $L(v) = \emptyset$ ($O(|V|)$ πολυπλοκότητα). Επομένως, η Q περιέχει αρχικά μόνο τον κουβά $P_\emptyset$, στον οποίο ανήκουν όλες οι κορυφές του γράφου. Κάθε κορυφή γνωρίζει ποιος κουβάς την περιέχει, καθώς και την ακριβή της θέση σε αυτόν. Κάθε κουβάς γνωρίζει τη θέση του στη Q. Όλες οι λίστες της δομής είναι διπλά συνδεδεμένες.

Χρησιμοποιώντας τη δομή Q, η εύρεση της κορυφής v_i με τη λεξικογραφικά μεγαλύτερη ετικέτα πραγματοποιείται σε χρόνο $O(1)$, διότι αυτή θα βρίσκεται πάντα στην πρώτη θέση του πρώτου κουβά της Q (η Q είναι ταξινομημένη). Στη συνέχεια διαγράφουμε την v_i από τη Q και τροποποιούμε την ετικέτα του κάθε γειτονά της (συνολικά $deg(v_i)$ γείτονες), ενημερώνοντας κατάλληλα τη δομή (όλες οι $w \in \Gamma(v_i)$ αλλάζουν κουβά, εφόσον έχει αλλάξει η ετικέτα τους). Δεδομένου ότι οι λίστες είναι διπλά συνδεδεμένες, η εύρεση, διαγραφή και μετακίνηση στοιχείων ανάμεσα στους κουβάδες γίνονται σε $O(1)$ χρόνο. Άρα, το συνολικό κόστος ανά κορυφή v_i είναι $O(1 + deg(v_i))$. Έτσι, εφόσον

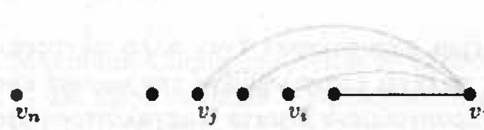
$\sum_{i=1}^{|V|} deg(v_i) = 2|E|$, η πολυπλοκότητα του Αλγόριθμου **LexBFS** είναι $O(|V| + |E|)$.



Σχήμα 2.6: Παράδειγμα της δομής (λίστα από κουβάδες) για ένα γράφο, όπου η κορυφή A επιλέχτηκε πρώτη ως v_8 και στη συνέχεια η C ως v_7. Σημειώνουμε ότι η δομή περιέχει τώρα όλες τις μη επιλεγμένες κορυφές, κάθε μια εκ των οποίων είναι συνδεδεμένη στη λίστα που αντιστοιχεί στην ετικέτα της.

Θεώρημα 2.11
Ένας γράφος είναι χορδικός αν η διάταξη που κατασκευάζει ο Αλγόριθμος LexBFS είναι μια τέλεια διάταξη απαλοιφής.

Απόδειξη. Έστω χορδικός γράφος $G = (V, E)$ και $v_n, \ldots, v_1$ η διάταξη που κατασκευάζει ο Αλγόριθμος **LexBFS**. Η απόδειξη θα γίνει με επαγωγή στον αριθμό των κορυφών $|V| = n$. Για $n = 1$ είναι τετριμμένο. Για $n > 1$, υποθέτουμε ότι το Θεώρημα ισχύει για γράφους με λιγότερες από n κορυφές. Άρα, αρκεί να δείξουμε ότι η n-οστή κορυφή που επιλέχτηκε (v_1) είναι **simplicial** . Έστω ότι η v_1 δεν είναι **simplicial**, δηλαδή έχει δύο προγόνους v_i και v_j, οι οποίοι δε γειτνιάζουν ($(v_i, v_j) \notin E$). Χωρίς βλάβη της γενικότητας, υποθέτουμε ότι $j > i$, δηλαδή η v_j βρίσκεται στα αριστερά της v_i στην τ.δ.α., όπως φαίνεται στο σχήμα 2.7.



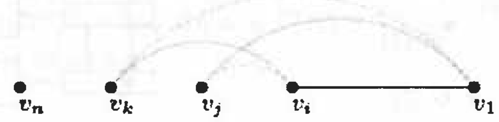
Σχήμα 2.7: Δύο πρόγονοι της v_1 που δε γειτνιάζουν.

Όταν επιλέχτηκε η v_j, ο δείκτης j προστέθηκε στην $L(v_1)$, αλλά όχι στην $L(v_i)$ (αφού η v_i δεν είναι γείτονας της v_j),δηλαδή:

$$L(v_1) = \ldots j \ldots$$

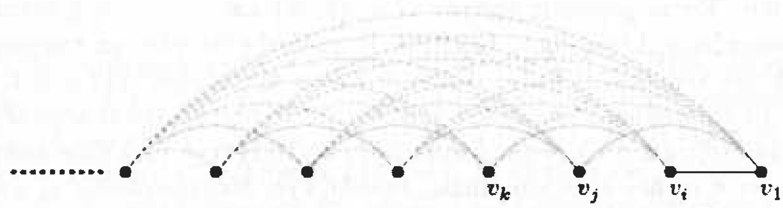
$$L(v_i) = \ldots \cancel{j} \ldots$$

Εφόσον η v_i επιλέχτηκε από τον αλγόριθμο πριν την v_1, η $L(v_i)$ θα πρέπει να ήταν λεξικογραφικά μεγαλύτερη από την $L(v_1)$. Αυτό όμως είναι εφικτό μόνο αν κάποια προγενέστερη στιγμή ο αλγόριθμος επέλεξε μια κορυφή v_k $(k > j)$, η οποία είναι γειτονική της v_1, αλλά όχι της v_i (σχήμα 2.8).



Σχήμα 2.8: Η ύπαρξη της κορυφής v_k.

Η ακμή $(v_j, v_k) \notin E$, διότι διαφορετικά ο G θα είχε C_4 $(v_1, v_i, v_k, v_j, v_1)$ χωρίς χορδή. Ομοίως, η $L(v_i)$ περιέχει το δείκτη k, ενώ η $L(v_j)$ όχι. Δεδομένου ότι η v_j επιλέχτηκε πριν την v_i, θα πρέπει να υπάρχει μια κορυφή πριν την v_k, η οποία γειτνιάζει με την v_j, αλλά όχι με την v_i. Η παραπάνω διαδικασία μπορεί να συνεχιστεί επ' άπειρον (σχήμα 2.9), αλλά ο G είναι πεπερασμένος. Άτοπο. Άρα η v_1 είναι **simplicial** . □



Σχήμα 2.9: Η διαδικασία μπορεί να συνεχιστεί επ' άπειρον.

Οι **Rose** και **Tarjan** παρουσίασαν έναν άλλο αλγόριθμο για την εύρεση μιας τ.δ.α., ο οποίος επιλέγει επανειλημμένα την κορυφή εκείνη, για την οποία έχει ήδη επιλεγεί σε προηγούμενα βήματα ο μεγαλύτερος αριθμός γειτόνων. Ο αλγόριθμος και η απόδειξή του βρίσκονται στο [12].

Δεδομένης μιας τ.δ.α. $v_1, \ldots, v_{|V|}$ των κορυφών ενός χορδικού γράφου $G = (V, E)$, ο Αλγόριθμος **Predecessors** βρίσκει το σύνολο των προγόνων $Pred(v_i)$ έκαστης κορυφής v_i.

Αλγόριθμος Predecessors

1. for $i = 1, \ldots, |V|$ do
2. $\quad Pred(v_i) = \emptyset$;
3. for $i = 2, \ldots, |V|$ do
4. $\quad$ for $j = 1, \ldots, i-1$ do
5. $\quad\quad$ if $(v_j, v_i) \in E$ then
6. $\quad\quad\quad$ Πρόσθεσε την κορυφή v_j στο $Pred(v_i)$;

Η πολυπλοκότητα του Αλγόριθμου **Predecessors** είναι προφανώς $O(|V|^2)$.

2.4 Μέγιστη κλίκα

Κλίκα (clique) ενός γράφου $G = (V, E)$ ονομάζεται κάθε υποσύνολο των κορυφών του V, στο οποίο κάθε κορυφή συνδέεται (μέσω ακμής) με κάθε άλλη. Το πρόβλημα της μέγιστης κλίκας (**maximum clique**) ισοδυναμεί με την εύρεση του μέγιστου πλήρως συνδεδεμένου υποσυνόλου των κορυφών του G. Το πλήθος των κορυφών της μέγιστης κλίκας συμβολίζεται με $\omega(G)$.

Έστω χορδικός γράφος $G = (V, E)$ και $v_1, \ldots, v_{|V|}$ μια τ.δ.α. του. Κάθε σύνολο της μορφής $v_i \cup Pred(v_i)$ είναι κλίκα. Αντίστροφα, αν το σύνολο S είναι κλίκα, τότε $S = v_i \cup Pred(v_i)$, όπου v_i είναι η μεγαλύτερη από τις κορυφές του S στην τ.δ.α.. Επομένως, το πλήθος των κλικών ενός πεπερασμένου χορδικού γράφου είναι το πολύ ίσο με τον αριθμό των κορυφών του $|V|$. Ο Αλγόριθμος MC βρίσκει τη μέγιστη κλίκα του G.

Αλγόριθμος MC

1. for $i = 1, \ldots, |V|$ do
2. $\quad$ Βρες την κορυφή v_t με το μεγαλύτερο αριθμό προγόνων $Pred(v_t)$;
3. Επέστρεψε το $v_t \cup Pred(v_t)$;

Ο Αλγόριθμος **Maximum Clique** υπολογίζει το πλήθος των προγόνων της κάθε κορυφής του G και κατά συνέπεια η πολυπλοκότητά του στη χειρότερη περίπτωση είναι $|V| \; max_{i \in V} |Pred(v_i)|$.

2.5 Ελάχιστος χρωματισμός κορυφών

Στο χρωματισμό κορυφών (**vertex coloring**), το πρόβλημα είναι η ανάθεση χρωμάτων (ή ακεραίων) σε κάθε κορυφή, έτσι ώστε κορυφές που γειτνιάζουν να έχουν διαφορετικό χρώμα. Το αντίστοιχο πρόβλημα βελτιστοποίησης (ελάχιστος χρωματισμός) αναφέρεται στην εύρεση του ελάχιστου αριθμού χρωμάτων (χρωματικός αριθμός $\chi(G)$) που χρωματίζει σωστά τις κορυφές του G. Ισχύει πάντα $\chi(G) \geq \omega(G)$.

Ο Αλγόριθμος VC υπολογίζει το $\chi(G)$.

Αλγόριθμος VC

1. for $i = 1, \ldots, |V|$ do
2. Χρωμάτισε την κορυφή v_i με το μικρότερο χρώμα που δεν έχει χρησιμοποιηθεί ανάμεσα στους προγόνους της;

Κάθε κορυφή v_i έχει $Pred(v_i)$ προγόνους και επομένως θα χρωματιστεί με ένα εκ των $Pred(v_i)+1$ χρωμάτων. Άρα, ο Αλγόριθμος VC θα χρησιμοποιήσει το πολύ $max_i\{Pred(v_i) + 1\}$ χρώματα. Έστω i^* ο δείκτης για τον οποίο επιτυγχάνεται το μέγιστο, οπότε $\chi(G) \leq Pred(v_{i^*}) + 1$. Εφόσον η διάταξη των κορυφών είναι μια τ.δ.α., το σύνολο $v_{i^*} \cup Pred(v_{i^*})$ είναι κλίκα και συνεπώς $\omega(G) \geq Pred(v_{i^*}) + 1$. Όμως, $\chi(G) \geq \omega(G)$ και επομένως $\chi(G) = \omega(G) = Pred(v_{i^*}) + 1$. Άρα, ο Αλγόριθμος VC καταλήγει στο βέλτιστο χρωματισμό και η πολυπλοκότητά του είναι $O(|V|^2)$.

Αν ο G είναι ένας γράφος διαστημάτων, μπορούμε να υλοποιήσουμε τον Αλγόριθμο VC χρησιμοποιώντας λίστες και να επιτύχουμε έτσι καλύτερη πολυπλοκότητα. Κάθε κορυφή v_i αντιστοιχεί σ' ένα διάστημα I_i με αρχή s_i και τέλος f_i. Με κατάλληλο αλγόριθμο ταξινομούμε τα άκρα όλων των διαστημάτων (συνολικά $2|V|$) και χρησιμοποιούμε το δείκτη k για να σαρώσουμε διαδοχικά όλα τα άκρα σύμφωνα με την ταξινόμηση που κατασκευάσαμε. Διατηρούμε δύο λίστες χρωμάτων: τη διαθέσιμη λίστα, η οποία περιέχει τα χρώματα που μπορούν να χρησιμοποιηθούν και τη μη διαθέσιμη, η οποία περιέχει τα χρώματα που δεν μπορούν να χρησιμοποιηθούν τη δεδομένη χρονική στιγμή που ο δείκτης k διαβάζει κάποιο άκρο s_i ή f_i. Η διαδικασία ανάθεσης χρωμάτων στα διαστήματα είναι η εξής:

- Αν ο δείκτης k διαβάζει κάποιο s_i, τότε αναθέτουμε το διάστημα I_i (αντίστοιχα την κορυφή v_i) στο χρώμα που βρίσκεται στην πρώτη θέση της διαθέσιμης λίστας και μεταφέρουμε το εν λόγω χρώμα από τη διαθέσιμη στη μη διαθέσιμη λίστα.

- Αν ο δείκτης k διαβάζει κάποιο f_i, τότε ελευθερώνουμε το χρώμα στο οποίο έχει ανατεθεί το I_i, δηλαδή το μεταφέρουμε από τη μη διαθέσιμη στην αρχή της διαθέσιμης λίστας (το I_i είχε ήδη ανατεθεί σε κάποιο χρώμα, διότι εφόσον $s_i < f_i$, ο δείκτης k διάβασε πρώτα το s_i και μετά το f_i).

Δουλεύοντας πάντα στην αρχή της διαθέσιμης λίστας, επιτυγχάνουμε το βέλτιστο χρωματισμό, διότι επιλέγουμε πάντα ένα χρώμα, το οποίο είχε ήδη ανατεθεί σε κάποιο διάστημα (όταν αυτό είναι εφικτό) προτού χρησιμοποιήσουμε ένα νέο. Έτσι, η πολυπλοκότητα του **Vertex Coloring** είναι $O(|V| +$

$|V|\log|V|)$ ($O(|V|\log|V|)$ για την ταξινόμηση των άκρων και $O(|V|)$ για τη σάρωσή τους με το δείκτη k).

2.6 Μέγιστο ανεξάρτητο σύνολο και ελάχιστη κάλυψη από κλίκες

Ανεξάρτητο σύνολο (**independent set**) ονομάζεται κάθε υποσύνολο του V, όπου για δύο οποιεσδήποτε κορυφές του δεν υπάρχει ακμή που να τις συνδέει. Το πρόβλημα του μέγιστου ανεξάρτητου συνόλου (**maximum independent set**) ισοδυναμεί με την εύρεση του μέγιστου συνόλου κορυφών, τέτοιου ώστε για οποιεσδήποτε δύο κορυφές να μην υπάρχει ακμή ανάμεσά τους. Το πλήθος των κορυφών του μέγιστου ανεξάρτητου συνόλου συμβολίζεται με $a(G)$.

Το πρόβλημα της ελάχιστης κάλυψης από κλίκες (**minimum covering by cliques**) αναφέρεται στην εύρεση του ελάχιστου αριθμού κλικών $k(G)$ που καλύπτουν τις κορυφές του G. Ισχύει πάντα $k(G) \geq a(G)$.

Ο Αλγόριθμος **ISCC** βρίσκει το μέγιστο ανεξάρτητο σύνολο και την ελάχιστη κάλυψη από κλίκες του G.

Αλγόριθμος ISCC

1. $y_1 = v_{|V|}$;
2. $S_1 = y_1 \cup Pred(y_1)$;
3. $j = 2$;
4. **for** $i = |V| - 1, \ldots, 1$ **do**
5.   **if** $v_i \notin Pred(y_1) \cup \ldots \cup Pred(y_{j-1})$ **then**
6. $y_j = v_i$;
7. $S_j = y_j \cup Pred(y_j)$;
8. $j = j + 1$;
9. $t = j$;
10. Επέστρεψε το $\{y_1, \ldots, y_t\}$, $\{S_1, \ldots, S_t\}$;

Το σύνολο $\{y_1, \ldots, y_t\}$ που παράγει ο Αλγόριθμος ISCC στην έξοδο είναι ένα ανεξάρτητο σύνολο μεγέθους t, διότι διαφορετικά αν $(y_i, y_j) \in E$ και $i < j$,τότε $y_i \in Pred(y_j)$. Άτοπο. Άρα, $a(G) \geq t$. Επίσης, κάθε σύνολο $S_i = y_i \cup Pred(y_i)$ είναι κλίκα και επομένως το $\{S_1, \ldots, S_t\}$ είναι μια κάλυψη από κλίκες μεγέθους t, διότι το $\{y_1, \ldots, y_t\} \cup Pred(y_1) \cup \ldots \cup Pred(y_t)$ είναι το σύνολο όλων των κορυφών του G. Όμως, $k(G) \geq a(G)$ και συνεπώς $a(G) = k(G) = t$. Άρα, το $\{y_1, \ldots, y_t\}$ είναι το μέγιστο ανεξάρτητο σύνολο και το $\{S_1, \ldots, S_t\}$ η ελάχιστη κάλυψη από κλίκες.

Στο k-οστό βήμα, ο αλγόριθμος ψάχνει τη μεγαλύτερη κορυφή που είναι μικρότερη από την y_{k-1} και δεν περιέχεται στο $Pred(y_1) \cup \ldots \cup Pred(y_{k-1})$. Επομένως εκτελεί το πολύ $(n-k+1)(k-1)$ βήματα και άρα η πολυπλοκότητά του είναι:

$$\sum_{k=1}^{t}(n-k+1)(k-1)=\frac{t(t-1)(3n-2t+1)}{6}$$

Κεφάλαιο 3

Χρωματισμός ακμών σε γράφους διαστημάτων

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα παρουσιάσουμε το πρόβλημα του χρωματισμού ακμών (**edge coloring**) σε γράφους διαστημάτων. Ο χρωματισμός των ακμών ενός γράφου G είναι μια διαδικασία, η οποία χρωματίζει τις ακμές του, έτσι ώστε ακμές που γειτνιάζουν να λαμβάνουν διαφορετικά χρώματα. Ο ελάχιστος αριθμός χρωμάτων που επαρκεί για το χρωματισμό των ακμών του G ονομάζεται χρωματικός δείκτης (**chromatic index**) και συμβολίζεται με $\chi'(G)$. Με $\Delta(G)$ συμβολίζουμε το μέγιστο βαθμό του G.

Το πρόβλημα της εύρεσης του χρωματικού δείκτη ενός γράφου G είναι $\mathcal{NP}$-πλήρες ακόμα και για γράφους βαθμού 3 ([13]). Παρόλο που ο **V.G. Vizing** απέδειξε στο [14] ότι ο $\chi'(G)$ ισούται είτε με $\Delta(G)$ είτε με $\Delta(G)+1$, το πρόβλημα απόφασης ανάμεσα σε αυτούς τους δύο ακεραίους είναι $\mathcal{NP}$-πλήρες. Αν $\chi'(G) = \Delta(G)$, τότε λέμε ότι ο γράφος G ανήκει στην κατηγορία 1 **class 1**, ενώ αν $\chi'(G) = \Delta(G)+1$, τότε λέμε ότι ο G ανήκει στην κατηγορία 2 **class 2**. Εντούτοις, ο χρωματικός δείκτης υπολογίζεται σε πολυωνυμικό χρόνο για ορισμένες κατηγορίες γράφων (πλήρεις, διμερείς, κ.τ.λ.). Σε αυτό το κεφάλαιο θα παρουσιάσουμε έναν πολυωνυμικό αλγόριθμο του **V.A. Bojarshinov** [15] για τον υπολογισμό του $\chi'(G)$ σε γράφους διαστημάτων με περιττό μέγιστο βαθμό.

3.2 Γράφοι διαστημάτων με περιττό μέγιστο βαθμό

Από το Θεώρημα 2.7, γνωρίζουμε ότι ένας γράφος $G = (V, E)$ είναι γράφος διαστημάτων ανν οι μεγιστικές κλίκες του $K_1, K_2, \ldots, K_r$ μπορούν να διαταχθούν γραμμικά, έτσι ώστε για κάθε κορυφή $v \in V$: αν $v \in K_i$, $v \in K_j$ και $1 \leq i < j \leq r$, τότε $v \in K_l$, $l = i+1, i+2, \ldots, j-2, j-1$. Μια τέτοια διάταξη ονομάζεται χαρακτηριστική σειρά μεγιστικών κλικών (**characteristic**

maximal cliques sequence). Η ένωση των συνόλων κορυφών των μεγιστικών κλικών που περιέχουν την κορυφή v ονομάζεται συστάδα (**cluster**) της v και συμβολίζεται με $\mathcal{C}(v)$.

Στο Θεώρημα 3.2 θα αποδείξουμε ότι ο Αλγόριθμος **EC** του **V.A. Bojarshinov** υπολογίζει σε πολυωνυμικό χρόνο το $\chi'(G)$ ενός γράφου διαστημάτων G. Για την απόδειξή του θα χρησιμοποιήσουμε το παρακάτω λήμμα.

Λήμμα 3.1
Έστω $\mathcal{K}_n$ ο πλήρης γράφος με n κορυφές. Τότε:

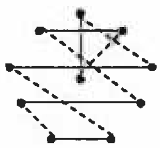
$$\chi'(\mathcal{K}_n) = \begin{cases} \Delta(\mathcal{K}_n), & n = 2m \\ \Delta(\mathcal{K}_n) + 1, & n = 2m+1 \end{cases}$$

Απόδειξη. Έστω ο πλήρης γράφος $\mathcal{K}_{2m}$. Θα δείξουμε ότι $\chi'(\mathcal{K}_{2m}) = \Delta(\mathcal{K}_{2m}) = 2m-1$. Τέλειο ταίριασμα (**matching**) ονομάζεται ένα σύνολο μη γειτονικών ακμών, το οποίο καλύπτει όλες τις κορυφές ενός γράφου G. Δηλαδή κάθε κορυφή του G προσπίπτει σε ακριβώς μια ακμή ενός τέλειου ταιριάσματος. Εφόσον οι ακμές ενός ταιριάσματος δε γειτνιάζουν, μπορούν να χρωματστούν με ένα χρώμα. Άρα, για να αποδείξουμε το ζητούμενο, αρκεί να δείξουμε ότι ο $\mathcal{K}_{2m}$ έχει $2m-1$ τέλεια ταιριάσματα (ξένα μεταξύ τους), κάθε ένα εκ των οποίων περιέχει m ακμές. Αυτό είναι εφικτό ανν οι ακμές του $\mathcal{K}_{2m}$ χωρίζονται σε $2m-1$ ξένα τέλεια ταιριάσματα. Πράγματι, το πλήθος των ακμών του $\mathcal{K}_{2m}$ είναι $|E| = \frac{2m(2m-1)}{2} = 2m^2 - m$, οπότε ο $\mathcal{K}_{2m}$ έχει $\frac{2m^2-m}{m} = 2m-1$ τέλεια ταιριάσματα. Το Σχήμα 3.1 υποδεικνύει τον τρόπο εύρεσης των $2m-1$ τέλειων ταιριασμάτων του $\mathcal{K}_{2m}$. Από τις κορυφές του γράφου, διατάσσουμε τις $2m-1$ κυκλικά και τοποθετούμε μια στο κέντρο του νοητού κύκλου. Το ταίριασμα που προκύπτει (Σχήμα 3.1) περιέχει m ακμές, εκ των οποίων η μία έχει ως ένα άκρο της την κεντρική κορυφή και οι υπόλοιπες είναι ακμές που συνδέουν τις απέναντι κορυφές. Περιστρέφοντας τον κύκλο όπως υποδεικνύουν οι διακεκομμένες γραμμές, δημιουργείται ένα νέο ταίριασμα μεγέθους m. Οι $2m-1$ δυνατές περιστροφές του κύκλου έχουν ως αποτέλεσμα τη δημιουργία των $2m-1$ μη επικαλυπτόμενων τέλειων ταιριασμάτων του $\mathcal{K}_{2m}$. Εφόσον για κάθε ταίριασμα χρησιμοποιούμε ένα μόνο χρώμα, έπεται ότι $\chi'(\mathcal{K}_{2m}) = \Delta(\mathcal{K}_{2m}) = 2m-1$.

Έστω ο πλήρης γράφος $\mathcal{K}_{2m+1}$. Ομοίως, το πλήθος των ακμών του είναι $|E| = \frac{(2m+1)2m}{2} = 2m^2+m$, οπότε ο αριθμός των μη επικαλυπτόμενων τέλειων (μεγέθους m) ταιριασμάτων που μπορούν να δημιουργηθούν είναι $\frac{2m^2+m}{m} = 2m+1$. Άρα, ο γράφος $\mathcal{K}_{2m+1}$ θα χρωματιστεί βέλτιστα με $2m+1$ χρώματα, δηλαδή $\chi'(\mathcal{K}_{2m+1}) = \Delta(\mathcal{K}_{2m+1}) = 2m+1$. □

Θεώρημα 3.2
Έστω γράφος διαστημάτων G με $\Delta(G) = 2m+1$. Ο G μπορεί να χρωματιστεί με $2m+1$ χρώματα σε χρόνο $O(|V|+|E|+\Delta^2)$.

Απόδειξη. Έστω γράφος διαστημάτων $G = (V, E)$ και $K_1, K_2, \ldots, K_r$ μια χαρακτηριστική σειρά των μεγιστικών κλικών του. Χωρίς βλάβη της γενικότη-



Σχήμα 3.1: Ο πλήρης γράφος $\mathcal{K}_{10}$ έχει 9 τέλεια ταιριάσματα και επομένως $\chi'(\mathcal{K}_{10}) = 9$.

τας, υποθέτουμε ότι υπάρχει κορυφή $v' \in K_1$, τέτοια ώστε $deg(v') = \Delta(G) = 2m+1$ (διαφορετικά, μπορούμε να προσθέσουμε στην K_1 τον απαιτούμενο αριθμό ψεύτικων κορυφών). Εφόσον κάθε ακμή του G ανήκει σε τουλάχιστον μια μεγιστική κλίκα, ο χρωματισμός των ακμών του ισοδυναμεί με το χρωματισμό των ακμών όλων των μεγιστικών κλικών του, υπό τις εξής προϋποθέσεις:
(i) Αν μια ακμή ανήκει σε πολλές κλίκες, τότε θα έχει το ίδιο χρώμα σε κάθε μια από αυτές.
(ii) Αν δύο ακμές προσπίπτουν στην ίδια κορυφή, τότε θα έχουν διαφορετικό χρώμα.

Ορίζουμε το σύνολο των κορυφών που ανήκουν σε δύο διαδοχικές μεγιστικές κλίκες ως: $Trans(k, k+1) = \{v | v \in K_k \cap K_{k+1}\}$. Ο Αλγόριθμος EC χρωματίζει τις ακμές όλων των μεγιστικών κλικών του G ικανοποιώντας τις παραπάνω προϋποθέσεις.

Αλγόριθμος Edge Coloring

1. Θεώρησε μια κλίκα με $2m+2$ κορυφές;
2. Αρίθμησε αυθαίρετα τις κορυφές της κλίκας;
3. Χρωμάτισε τις ακμές της κλίκας με $2m+1$ χρώματα;
4. Φτιάξε έναν $(2m+2) \times (2m+2)$ πίνακα $A = (a_{ij})$;
5. Γέμισε τον A ως εξής: αν η ακμή (i, j) έχει χρώμα c, τότε $a_{ij} := c$, $c \in \{1, 2, \ldots, 2m+1\}$;
6. Θεώρησε το $\mathcal{C}(v')$;
7. Αρίθμησε αυθαίρετα τις κορυφές του $\mathcal{C}(v')$;
8. Χρωμάτισε τις ακμές του $\mathcal{C}(v')$ ως εξής: αν $a_{ij} = c$, τότε χρωμάτισε την ακμή (i, j) με c;
9. Θέσε $k = 1$;
10. while $k < r$ do
11. Έστω $K_1, K_2, \ldots, K_k$ οι κλίκες των οποίων οι ακμές έχουν ήδη χρωματιστεί;
12. Υπολόγισε το $Trans(k, k+1)$;
13. Επέλεξε την κορυφή v^* ως την πρώτη κορυφή του $Trans(k, k+1)$;

14. Θεώρησε το $\mathcal{C}(v^*)$ και αρίθμησε όλες τις μη αριθμημένες κορυφές του;
15. Χρωμάτισε τις μη χρωματισμένες ακμές του $\mathcal{C}(v^*)$ ως εξής: αν $a_{ij} = c$, τότε χρωμάτισε την ακμή (i, j) με c;

Ορθότητα και πολυπλοκότητα του Αλγόριθμου EC

Στις Γραμμές 1-5, ο Αλγόριθμος EC χρωματίζει μια κλίκα μεγέθους $2m+2$ χρησιμοποιώντας $2m+1$ χρώματα (Λήμμα 3.1) και αποθηκεύει τον εν λόγω χρωματισμό στον πίνακα A. Στις Γραμμές 6-8, οι ακμές του $\mathcal{C}(v')$ χρωματίζονται ορθώς με χρήση του πίνακα A, δεδομένου ότι στο $\mathcal{C}(v')$ θα περιέχονται το πολύ $2m+2$ κορυφές.

Υποθέτουμε ότι μετά τα πρώτα i βήματα, ο Αλγόριθμος EC κατασκευάζει σωστά ένα μερικό χρωματισμό των ακμών του γράφου. Έστω $K_1, K_2, \ldots, K_k$ οι κλίκες των οποίων οι ακμές χρωματίστηκαν στα πρώτα i βήματα και έστω $K_l, K_{l+1}, \ldots, K_k, K_{k+1}, \ldots, K_{k+c}$ οι κλίκες που περιέχουν την κορυφή $v^* \in Trans(k, k+1)$ (η οποία επιλέχτηκε από τον Αλγόριθμος EC στο i+1 βήμα). Θα αποδείξουμε ότι ο μερικός χρωματισμός μετά το i+1 βήμα είναι επίσης σωστός.

Έστω ότι ο μερικός χρωματισμός που κατασκευάστηκε μετά το χρωματισμό του $\mathcal{C}(v^*)$ δεν είναι σωστός και θα καταλήξουμε σε άτοπο. Υποθέτουμε ότι υπάρχουν δύο ακμές $e_1 = (a, b)$ και $e_2 = (b, c)$ που έχουν το ίδιο χρώμα. Από την υπόθεση, οι $K_1, K_2, \ldots, K_k$ είχαν χρωματιστεί σωστά, οπότε μια τουλάχιστον εκ των $\{e_1, e_2\}$ δεν ανήκει στο σύνολο αυτών των κλικών. Χωρίς βλάβη της γενικότητας, υποθέτουμε ότι η e_2 χρωματίστηκε για πρώτη φορά στο i+1 βήμα. Η κορυφή b ανήκει σε τουλάχιστον μια εκ των $K_k, K_{k+1}, \ldots, K_{k+c}$ και εφόσον ο Αλγόριθμος EC επέλεξε την v^* και όχι την b, έπεται ότι η a και συνεπώς η ακμή e_1 ανήκουν στο $\mathcal{C}(v^*)$.

Κάποιες από τις κορυφές που ανήκουν στο $\mathcal{C}(v^*)$ είναι ήδη αριθμημένες σε προηγούμενα βήματα. Εφόσον $deg(v^*) \leq 2m+1$, συνεπάγεται ότι το πλήθος των κορυφών στο $\mathcal{C}(v^*)$ θα είναι το πολύ $2m+2$. Επομένως, όλες οι υπόλοιπες κορυφές μπορούν να αριθμηθούν, έτσι ώστε ο αριθμός οποιασδήποτε από αυτές να μην υπερβαίνει το $2m+2$ και να μην υπάρχουν δύο κορυφές με τον ίδιο αριθμό. Άρα, οι a, b και c λαμβάνουν διαφορετικούς αριθμούς και συνεπώς οι ακμές e_1 και e_2 έχουν διαφορετικό χρώμα. Άτοπο και άρα ο μερικός χρωματισμός μετά το i+1 βήμα είναι σωστός.

Τα σύνολα $Trans(i, i+1)$ και οι κορυφές v^* μπορούν να υπολογιστούν καθώς κατασκευάζουμε τη χαρακτηριστική σειρά των μεγιστικών κλικών, η οποία απαιτεί $O(|V| + |E|)$ βήματα ([16],[17]). Ο χρωματισμός μιας κλίκας με $\Delta(G) + 1$ κορυφές και η κατασκευή του πίνακα A από τις πληροφορίες του χρωματισμού χρειάζεται $O((\Delta(G))^2)$ βήματα. Άρα, η πολυπλοκότητα του Αλγόριθμου EC είναι $O(|V| + |E| + \Delta^2)$. □

Το Θεώρημα 3.2 δεν ισχύει για γράφους διαστημάτων με άρτιο μέγιστο βαθμό, δηλαδή δεν μπορούμε να αποδείξουμε ότι ένας γράφος διαστημάτων G με $\Delta(G) = 2m$ χρωματίζεται βέλτιστα με $\Delta(G)$ χρώματα. Ας θεωρήσουμε τον πλήρη γράφο $\mathcal{K}_{2m+1}$. Τότε, στις αντίστοιχες Γραμμές 1-5 του Αλγόριθμου EC θα έπρεπε να χρωματίσουμε μια κλίκα μεγέθους $2m + 1$. Σύμφωνα όμως με το Λήμμα 3.1 αυτή χρωματίζεται βέλτιστα με $2m + 1$ χρώματα και έτσι για το χρωματισμό του $\mathcal{K}_{2m+1}$ θα χρησιμοποιούσαμε $\Delta(G) + 1$ χρώματα. Δε γνωρίζουμε όμως αν πράγματι χρειάζονται $\Delta(G) + 1$ χρώματα ή αν ο βέλτιστος χρωματισμός μπορεί να επιτευχθεί με χρήση $\Delta(G)$ χρωμάτων. Έτσι, ένας αντίστοιχος αλγόριθμος δε μπορεί να εφαρμοστεί στην περίπτωση γράφων διαστημάτων με άρτιο μέγιστο βαθμό.

Κεφάλαιο 4

Χρωματισμός κόμβων με περιορισμούς πληθυκότητας σε γράφους διαστημάτων

4.1 Εισαγωγή

Σε αυτή την ενότητα θα μελετήσουμε το πρόβλημα του βέλτιστου χρωματισμού κορυφών ενός γράφου διαστημάτων G, υπό τον περιορισμό ότι κάθε χρώμα χρησιμοποιείται για το χρωματισμό το πολύ k κορυφών, όπου k είναι ένας θετικός ακέραιος. Στην ανάλυση που ακολουθεί αναφερόμαστε στο σύνολο των διαστημάτων $\mathcal{I}$ αντί του γράφου G και συμβολίζουμε με $\chi(\mathcal{I}, k)$ το πλήθος των κλάσεων χρωμάτων του βέλτιστου χρωματισμού του $\mathcal{I}$, όπου κάθε χρώμα χρησιμοποιείται το πολύ k φορές. Είναι προφανές ότι $\chi(\mathcal{I}, k) \geq \max\{\chi(\mathcal{I}), \lceil \frac{n}{k} \rceil\}$, όπου n είναι το πλήθος των διαστημάτων του $\mathcal{I}$. Οι **Bodlaender** και **Jansen** ([18]) απέδειξαν ότι το πρόβλημα του k-χρωματισμού είναι $\mathcal{NP}$-πλήρες σε γράφους διαστημάτων για $k \geq 4$, ενώ για $k = 3$ το πρόβλημα παραμένει ανοιχτό.

Ορίζουμε ως ταίριασμα (**matching**) ένα σύνολο διαστημάτων, τα οποία ανά δύο δεν αλληλεπικαλύπτονται. Το μέγιστο σύνολο τέτοιων διαστημάτων ονομάζεται μέγιστο ταίριασμα (**maximum matching**). Τέλειο ταίριασμα (**perfect matching**) είναι ένα μέγιστο ταίριασμα, στο οποίο περιέχονται όλα τα διαστήματα του $\mathcal{I}$.

4.2 Ένας πολυωνυμικός αλγόριθμος

Στην ενότητα αυτή θα παρουσιάσουμε έναν απλό αλγόριθμο, τον Αλγόριθμο **Matching**, για την επίλυση του χρωματισμού κορυφών με περιορισμούς πληθυκότητας για $k = 2$. Σημειώνουμε ότι ο Αλγόριθμος **Matching** μπορεί να εφαρμοστεί σε έναν οποιονδήποτε γράφο G και όχι μόνο σε γράφους διαστημάτων.

Αλγόριθμος Matching

Θεώρησε το συμπληρωματικό γράφο του G, έστω $\overline{G}$;
Βρες ένα τέλειο ταίριασμα T στον $\overline{G}$;
Θέσε $col = 0$;
for κάθε ακμή του T **do**
    Θέσε $col = col + 1$;
    Χρωμάτισε τις κορυφές που προσπίπτουν στην ακμή με το χρώμα col;

Ο Αλγόριθμος **Matching**, δεδομένου ενός γράφου G, βρίσκει ένα τέλειο ταίριασμα T στον $\overline{G}$, δηλαδή ένα σύνολο ξένων μεταξύ τους ακμών του $\overline{G}$, οι οποίες καλύπτουν όλες τις κορυφές του γράφου. Οι κορυφές που προσπίπτουν σε κάθε ακμή του T δε γειτνιάζουν στον G, οπότε τις χρωματίζουμε με το ίδιο χρώμα και έτσι επιλύουμε το πρόβλημα του χρωματισμού κορυφών με περιορισμούς πληθυκότητας για $k = 2$. Η πολυπλοκότητα εύρεσης ενός τέλειου ταιριάσματος είναι $O(n^{2.5})$ [19].

4.3 Ένας γραμμικός αλγόριθμος

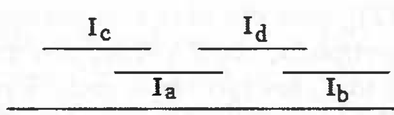
Σε αυτή την παράγραφο, θα παρουσιάσουμε ένα γραμμικό αλγόριθμο για την επίλυση του 2-χρωματισμού σε γράφους διαστημάτων, ο οποίος προτάθηκε από τον **Gardi** [20]. Θεωρούμε ένα σύνολο διαστημάτων $\mathcal{I} = \{I_1, \ldots, I_n\}$ και υποθέτουμε ότι γνωρίζουμε τη διάταξη τους σύμφωνα με το αριστερό ($l(I_u)$) και το δεξιό τους ($r(I_u)$) άκρο. Συμβολίζουμε με $\mathcal{C} = \{C_1, \ldots, C_{\chi(\mathcal{I})}\}$ το βέλτιστο χρωματισμό του $\mathcal{I}$ και υπενθυμίζουμε ότι κάθε σύνολο του $\mathcal{C}$, δηλαδή κάθε κλάση χρωμάτων, είναι ένα ανεξάρτητο σύνολο. Εφόσον ο G είναι ένας γράφος διαστημάτων, ισχύει $\chi = \omega$. Έτσι, συμβολίζουμε με $K_{max} = \{K_1, \ldots, K_{\chi(\mathcal{I})}\}$ τη μέγιστη κλίκα του $\mathcal{I}$ (όπου $K_1, \ldots, K_{\chi(\mathcal{I})}$) είναι διαστήματα του $\mathcal{I}$ και θεωρούμε ότι διάστημα K_i περιέχεται στο $C_i \in \mathcal{C}$, $\forall i = 1, \ldots, \chi(\mathcal{I})$.

Λήμμα 4.1
Αν κάποιο $C_i \in \mathcal{C}$ περιέχει μόνο ένα διάστημα, τότε αυτό το διάστημα ανήκει στη μέγιστη κλίκα K_{max} του $\mathcal{I}$.

Απόδειξη. Προφανές, διότι για κάθε γράφο διαστημάτων ισχύει $\chi = \omega$, οπότε η K_{max} πρέπει να περιέχει ακριβώς ένα διάστημα από κάθε σύνολο του $\mathcal{C}$. Αν το μοναδικό διάστημα του C_i δεν ανήκει στη K_{max} του $\mathcal{I}$, τότε οδηγούμαστε σε αντίφαση. □

Λήμμα 4.2
Αν ο αριθμός n των διαστημάτων του $\mathcal{I}$ είναι άρτιος και κάθε σύνολο του $\mathcal{C}$ περιέχει τουλάχιστον δύο διαστήματα, τότε $\chi(\mathcal{I}, 2) = \frac{n}{2}$.

Απόδειξη. Από την υπόθεση του λήμματος, κανένα σύνολο του $\mathcal{C}$ δεν περιέχει μόνο ένα διάστημα. Στη γενική περίπτωση, κάποια από τα σύνολα του $\mathcal{C}$ περιέχουν άρτιο αριθμό διαστημάτων και κάποια περιττό. Εφόσον ο αριθμός n των διαστημάτων του $\mathcal{I}$ είναι άρτιος, το πλήθος των συνόλων που έχουν περιττό μέγεθος είναι άρτιο. Έστω C_i, C_j δύο σύνολα περιττού μεγέθους. Θα δείξουμε ότι υπάρχουν δύο διαστήματα, ένα στο C_i και ένα στο C_j, τα οποία δεν αλληλεπικαλύπτονται. Υποθέτουμε, χωρίς βλάβη της γενικότητας, ότι $I_a, I_b \in C_i$, όπου $r(I_a) \leq l(I_b)$ και $I_c, I_d \in C_j$, όπου $r(I_c) \leq l(I_d)$. Αν τα I_a, I_d δεν επικαλύπτονται, τότε αυτά είναι τα ζητούμενα διαστήματα. Αν επικαλύπτονται, τότε υποθέτουμε, χωρίς βλάβη της γενικότητας, ότι $l(I_d) \leq r(I_a)$. Επειδή $r(I_a) \leq l(I_b)$ και $r(I_c) \leq l(I_d)$, συνεπάγεται ότι $r(I_c) \leq l(I_b)$. Δηλαδή τα I_b, I_c δεν επικαλύπτονται και επομένως είναι τα ζητούμενα διαστήματα (Σχήμα 4.1).



Σχήμα 4.1: Παράδειγμα για το Λήμμα 4.2.

Έτσι, σε κάθε ζεύγος συνόλων περιττού μεγέθους μπορούμε να βρούμε δύο διαστήματα (ένα σε κάθε σύνολο), τα οποία δεν αλληλεπικαλύπτονται και να τα αφαιρέσουμε. Τα σύνολα που προκύπτουν (μετά την απομάκρυνση των δύο διαστημάτων) είναι άρτιου μεγέθους και το ζεύγος διαστημάτων που αφαιρέσαμε αποτελεί ένα ανεξάρτητο σύνολο μεγέθους δύο. Μετά το πέρας αυτής της διαδικασίας, όλα τα σύνολα είναι ανεξάρτητα, περιέχουν τουλάχιστον δύο διαστήματα και είναι άρτιου μεγέθους. Έτσι, μπορούμε να διασπάσουμε κάθε ένα εξ' αυτών σε ξένα ανεξάρτητα σύνολα μεγέθους δύο. Άρα, $\chi(\mathcal{I}, 2) = \frac{n}{2}$. □

Λήμμα 4.3
Έστω $\theta(\mathcal{I})$ το πλήθος των συνόλων του $\mathcal{C}$, τα οποία περιέχουν ακριβώς ένα διάστημα. Τότε, $\chi(\mathcal{I}, 2) = \lceil \frac{n+\theta(\mathcal{I})}{2} \rceil$.

Απόδειξη. Εφόσον υπάρχουν $\theta(\mathcal{I})$ ανεξάρτητα σύνολα που περιέχουν ακριβώς ένα διάστημα, έπεται ότι υπάρχουν $\theta(\mathcal{I})$ διαστήματα, των οποίων το χρώμα είναι μοναδικό. Για τα υπόλοιπα $n - \theta(\mathcal{I})$ διαστήματα (τα οποία περιέχονται σε ανεξάρτητα σύνολα μεγέθους τουλάχιστον δύο), έχουμε ότι (Λήμμα 4.2),

- αν ο $n - \theta(\mathcal{I})$ είναι άρτιος, τότε θα χρησιμοποιήσουμε $\frac{n-\theta(\mathcal{I})}{2}$ κλάσεις χρωμάτων (μεγέθους το πολύ δύο) για το χρωματισμό τους, ενώ

- αν ο $n-\theta(\mathcal{I})$ είναι περιττός, τότε θα χρησιμοποιήσουμε $\frac{n-\theta(\mathcal{I})}{2}-1$ κλάσεις χρωμάτων (μεγέθους το πολύ δύο) για το χρωματισμό τους.

Συνεπώς, για το χρωματισμό των $n - \theta(\mathcal{I})$ διαστημάτων, απαιτούνται $\lceil \frac{n-\theta(\mathcal{I})}{2} \rceil$ ανεξάρτητα σύνολα μεγέθους τουλάχιστον δύο. Άρα, $\chi(\mathcal{I},2) = \lceil \frac{n-\theta(\mathcal{I})}{2} \rceil + \theta(\mathcal{I}) = \lceil \frac{n+\theta(\mathcal{I})}{2} \rceil$ □

Σύμφωνα με το Λήμμα 4.3, για να επιλύσουμε βέλτιστα το πρόβλημα του 2-χρωματισμού αρκεί να βρούμε ένα χρωματισμό του συνόλου $\mathcal{I}$, τέτοιο ώστε το πλήθος $\theta(\mathcal{I})$ των ανεξάρτητων συνόλων μεγέθους ένα να είναι όσο το δυνατόν μικρότερο. Από το Λήμμα 4.1, γνωρίζουμε ότι το μοναδικό διάστημα κάθε συνόλου του $\mathcal{C}$ μεγέθους ένα περιέχεται στη μέγιστη κλίκα K_{max} του $\mathcal{I}$. Έτσι, υπολογίζουμε το μέγιστο ταίριασμα T ανάμεσα στα διαστήματα της K_{max} και στα υπόλοιπα διαστήματα. Θεωρούμε ότι το T αναπαρίσταται από ένα διάνυσμα μεγέθους $|\chi(\mathcal{I})|$, όπου στη θέση i του διανύσματος είτε έχει αποθηκευτεί ο δείκτης του διαστήματος του $\mathcal{I} \setminus K_{max}$ που αποτελεί το ταίρι του $K_i \in K_{max}$ είτε η τιμή 0 αν το K_i δεν έχει τέτοιο ταίρι. Έχοντας υπολογίσει το T, ελαχιστοποιούμε το $\theta(\mathcal{I})$ με τον εξής τρόπο: σε κάθε $C_i = \{I_u\}$ (δηλαδή σε κάθε σύνολο του $\mathcal{C}$ μεγέθους ένα) προσθέτουμε το διάστημα $I_v \in C_j$ (όπου το C_j είναι ένα σύνολο του $\mathcal{C}$ μεγέθους τουλάχιστον δύο) αν τα I_u, I_v αποτελούν ταίριασμα. Ο Αλγόριθμος Gar2 βρίσκει το ελάχιστο $\theta(\mathcal{I})$.

Αλγόριθμος Gar2(C,T)

$Q = \emptyset$;
for κάθε $C_i \in \mathcal{C}$ **do**
  if το C_i είναι μεγέθους ένα **then**
    $Q = Q \cup C_i$;
while $Q \neq \emptyset$ **do**
  $Q = Q \setminus C_i$;
  Έστω k ο δείκτης που είναι αποθηκευμένος στη θέση i του T ;
  if $k \neq 0$ **then**
    Έστω j ο δείκτης του συνόλου του $\mathcal{C}$, στο οποίο ανήκει το I_k;
    $C_j = C_j \setminus \{I_k\}$;
    $C_i = C_i \cup \{I_k\}$;
    if το C_j είναι μεγέθους ένα **then** $Q = Q \cup \{C_j\}$;

Ο Αλγόριθμος Gar1 επιλύει το πρόβλημα του 2-χρωματισμού και επιστρέφει το S, το οποίο περιέχει τις κλάσεις χρωμάτων (τα ανεξάρτητα σύνολα μεγέθους τουλάχιστον δύο) του βέλτιστου χρωματισμού. Για τον Αλγόριθμο **Garl** , θα χρειαστούμε τον παρακάτω ορισμό:

Ορισμός 4.4
Ένας διμερής γράφος $G_B = (X \cup Y, E)$ ονομάζεται Y-κυρτός αν οι κορυφές του Y μπορούν να διαταχθούν έτσι ώστε όλες οι κορυφές που γειτνιάζουν στην ίδια κορυφή του X να παρουσιάζονται διαδοχικά σε αυτή την διάταξη.

Σημειώνουμε ότι οι **Steiner** και **Yeomans** ([21]) πρότειναν έναν αλγόριθμο, ο οποίος δεδομένου ενός κυρτού διμερή γράφου διαστημάτων $G_B = (X \cup Y, E)$, βρίσκει ένα μέγιστο ταίριασμα ανάμεσα στα διαστήματα του X και του Y. Δηλαδή για κάθε διάστημα $I_x \in X$ ελέγχει αν υπάρχει κάποιο διάστημα $I_y \in Y$, έτσι ώστε τα I_x, I_y να μην επικαλύπτονται. Ο αλγόριθμός τους έχει πολυπλοκότητα $O(|X|)$.

Αλγόριθμος Gar1

Βήμα 1
Υπολόγισε ένα βέλτιστο χρωματισμό $\mathcal{C} = \{C_1, \ldots, C_{\chi(\mathcal{I})}\}$ του $\mathcal{I}$;
if όλα τα σύνολα του $\mathcal{C}$ είναι μεγέθους το πολύ δύο **then goto** Βήμα 3;
if όλα τα σύνολα του $\mathcal{C}$ είναι μεγέθους τουλάχιστον δύο **then goto** Βήμα 3;

Βήμα 2
Υπολόγισε τη μέγιστη κλίκα $K_{max} = \{K_1, \ldots, K_{\chi(\mathcal{I})}\}$ του $\mathcal{I}$;
Κατασκεύασε το διμερή γράφο $G_B = (K_{max} \cup (\mathcal{I} \setminus K_{max}), E')$, όπου
$E' = \{(I_v, I_u) | I_v \in K_{max}, I_u \in \mathcal{I} \setminus K_{max}\}$ και $I_v \cap I_u = \emptyset$;
Υπολόγισε το μέγιστο ταίριασμα T ανάμεσα στα σύνολα K_{max} και $\mathcal{I} \setminus K_{max}$ του G_B;
$\mathcal{C} = Gar2(\mathcal{C}, T)$;

Βήμα 3
$S = \emptyset$;
for κάθε σύνολο $C_i \in \mathcal{C}$ μεγέθους ένα **do**
    $\mathcal{C} = \mathcal{C} \setminus C_i$;
    $S = S \cup C_i$;
if ο αριθμός των διαστημάτων του $\mathcal{C}$ είναι περιττός **then**
    Αφαίρεσε ένα διάστημα από ένα οποιοδήποτε σύνολο του $\mathcal{C}$ περιττού μεγέθους και πρόσθεσέ το σε ένα νέο ανεξάρτητο σύνολο στο S;
Υπολόγισε ένα τέλειο ταίριασμα (σύμφωνα με τη διαδικασία του Λήμματος 4.2) και πρόσθεσέ το στο S;
Επέστρεψε το S;

Πολυπλοκότητα

Για να υπολογίσουμε την πολυπλοκότητα του Αλγόριθμου **Gar1**, θα υπολογίσουμε την πολυπλοκότητα του κάθε βήματός του.

Βήμα 1: Η πολυπλοκότητα του Βήματος 1 έγκειται στο κόστος εύρεσης του βέλτιστου χρωματισμού $\mathcal{C}$ του συνόλου διαστημάτων $\mathcal{I}$, το οποίο είναι $O(n)$ (σύμφωνα με τον Αλγόριθμο **VC** του Κεφαλαίου 2 και δεδομένου ότι διαθέτουμε την ταξινόμηση των διαστημάτων του $\mathcal{I}$ σύμφωνα με το αριστερό και δεξιό άκρο τους).

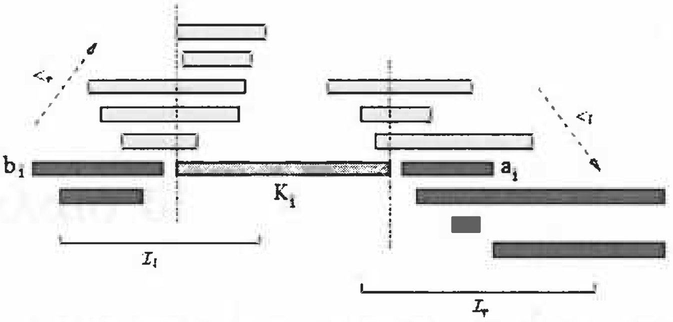
Βήμα 2: Η πολυπλοκότητα του Βήματος 2 εξαρτάται από την ακόλουθη ιδιότητα:

Λήμμα 4.5
Ο διμερής γράφος $G_B = (K_{max} \cup (\mathcal{I} \setminus K_{max}), E')$ *είναι* $(\mathcal{I} \setminus K_{max})$*-κυρτός.*

Απόδειξη. Διασπάμε το $\mathcal{I} \setminus K_{max}$ σε δύο ξένα υποσύνολα $\mathcal{I}_l$ και $\mathcal{I}_r$, όπου $\mathcal{I}_l$ είναι το σύνολο των διαστημάτων που βρίσκονται στα αριστερά των διαστημάτων του K_{max} και $\mathcal{I}_r$ το σύνολο των διαστημάτων που βρίσκονται στα δεξιά των διαστημάτων του K_{max}. Τα σύνολα $\mathcal{I}_l$, $\mathcal{I}_r$ είναι ξένα μεταξύ τους, γιατί αν ένα διάστημα του $\mathcal{I} \setminus K_{max}$ ανήκει και στο $\mathcal{I}_l$ και στο $\mathcal{I}_r$, τότε θα πρέπει να ανήκει και στο K_{max}. Διατάσσουμε τα διαστήματα του $\mathcal{I}_l$ κατά αύξουσα σειρά σύμφωνα με το δεξιό τους άκρο και τα διαστήματα του $\mathcal{I}_r$ κατά αύξουσα σειρά σύμφωνα με το αριστερό τους άκρο. Με αυτό τον τρόπο κατασκευάζουμε μια διάταξη για όλα τα διαστήματα του $\mathcal{I} \setminus K_{max}$. Για κάθε $K_i \in K_{max}$ ορίζουμε τους δείκτες $a_i = \min\{u | r(K_i) \leq l(I_u)$ και $I_u \in \mathcal{I}_r\}$ και $b_i = \min\{u | r(I_u) \leq l(K_i)$ και $I_u \in \mathcal{I}_l\}$. Προφανώς, τα διαστήματα K_i και I_u $(u \in \{a_i, b_i\})$ δεν επικαλύπτονται και αν λάβουμε υπόψην τη διάταξη που ορίσαμε για τα διαστήματα του $\mathcal{I} \setminus K_{max}$, καταλήγουμε στο συμπέρασμα ότι ο G_B είναι $(\mathcal{I} \setminus K_{max})$-κυρτός (Σχήμα 4.2). □

Ο διμερής γράφος G_B είναι $(\mathcal{I} \setminus K_{max})$-κυρτός, οπότε μπορούμε να βρούμε ένα μέγιστο ταίριασμά του σε $O(n)$ χρόνο, χρησιμοποιώντας τον αλγόριθμο των **Steiner** και **Yeomans** ([21]). Ο αλγόριθμός τους απαιτεί ως είσοδο τη διάταξη των διαστημάτων του $\mathcal{I} \setminus K_{max}$ και τους δείκτες a_i και b_i για κάθε $K_i \in K_{max}$. Θα αποδείξουμε ότι τα παραπάνω μπορούν να υπολογιστούν με $O(n)$ κόστος.

Η κλίκα $K_{max} = \{K_1, \ldots, K_{\chi(\mathcal{I})}\}$ έχει οριστεί έτσι ώστε το διάστημα K_i να περιέχεται στο σύνολο C_i του $\mathcal{C}$. Έτσι, μπορούμε να διαγράψουμε τα διαστήματα της κλίκας σε $O(n)$ χρόνο και δεδομένου ότι έχουμε στη διάθεσή μας τη διάταξη των διαστημάτων του $\mathcal{I}$ κατά αύξουσα σειρά σύμφωνα με το



Σχήμα 4.2: Η απόδειξη του Λήμματος 4.5.

αριστερό και το δεξιό τους άκρο, μπορούμε να αποκτήσουμε τη διάταξη του $\mathcal{I} \setminus K_{max}$ (όπως αυτή ορίζεται στο Λήμμα 4.3) με $O(n)$ κόστος.

Για την εύρεση των δεικτών a_i αρκεί να διατάξουμε τα διαστήματα του K_{max} κατά αύξουσα σειρά σύμφωνα με το δεξιό άκρο τους (τετριμμένο εφόσον διαθέτουμε την ταξινόμηση των διαστημάτων του $\mathcal{I}$) και να σαρώσουμε τα διαστήματα του $\mathcal{I}_r$ (διότι ισχύει $a_{i-1} \leq a_i$, $\forall i = 2, \ldots, \chi(\mathcal{I})$). Με συμμετρικό τρόπο υπολογίζουμε τους δείκτες b_i. Άρα, η πολυπλοκότητα του Βήματος 2 είναι $O(n)$.

Βήμα 3: Η πολυπλοκότητα του Βήματος 3 εξαρτάται από το κόστος εύρεσης του τέλειου ταιριάσματος στη Γραμμή 16. Αυτό το σύνολο βρίσκεται βάσει της απόδειξης του Λήμματος 4.2 (εφόσον το $\mathcal{C}$ αποτελείται από σύνολα μεγέθους τουλάχιστον δύο και ο συνολικός αριθμός των διαστημάτων του είναι άρτιος), οπότε απαιτεί $O(n)$ χρόνο.

Το κόστος εκτέλεσης του Αλγόριθμου **Gar2** είναι $O(n)$, οπότε ο Αλγόριθμος **Gar1** έχει πολυπλοκότητα $O(n)$.

Κεφάλαιο 5

Χρωματισμός κορυφών σε γράφους κυκλικών τόξων

5.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα μελετήσουμε το πρόβλημα του χρωματισμού κορυφών ενός γράφου κυκλικών τόξων. Υπενθυμίζουμε ότι ένας γράφος G ονομάζεται γράφος κυκλικών τόξων, αν οι κορυφές του μπορούν να αναπαρασταθούν ως τόξα στην περιφέρεια ενός κύκλου, ούτως ώστε δύο τόξα να αλληλεπικαλύπτονται ανν οι αντίστοιχες κορυφές γειτνιάζουν. Υποθέτουμε ότι ο G αναπαρίσταται από το σύνολο των κυκλικών τόξων $F = \{A_1, A_2, \ldots, A_n\}$. Ορίζουμε ως φόρτο του F, και συμβολίζουμε με $L(F)$, το μέγιστο αριθμό τόξων, τα οποία περιέχουν ένα κοινό σημείο του κύκλου. Στο χρωματισμό των κορυφών του G προσπαθούμε να αναθέσουμε τον ελάχιστο αριθμό χρωμάτων στα τόξα του, ούτως ώστε δύο τόξα που επικαλύπτονται να μην έχουν το ίδιο χρώμα. Προφανώς, $\chi(G) \geq L$.

Θα δείξουμε, αρχικά, ότι ο χρωματισμός των κορυφών ενός γράφου κυκλικών τόξων είναι $\mathcal{NP}$-πλήρες πρόβλημα [22] και στη συνέχεια θα παρουσιάσουμε προσεγγιστικούς αλγόριθμους [8],[23] για αυτό το πρόβλημα.

5.2 $\mathcal{NP}$-πληρότητα

Το $\mathcal{NP}$-πλήρες πρόβλημα, το οποίο θα ανάγουμε στο πρόβλημα του χρωματισμού των κορυφών ενός γράφου κυκλικών τόξων, είναι μια ειδική περίπτωση του προβλήματος των μη τεμνόμενων μονοπατιών (Μ.Τ.Μ.).

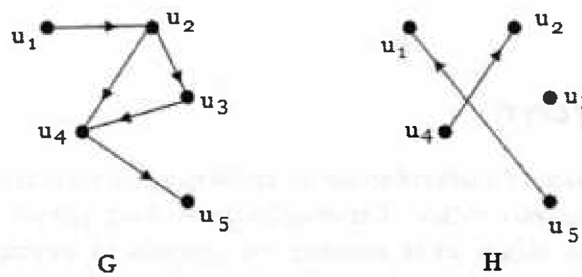
Το γενικό πρόβλημα των Μ.Τ.Μ. ορίζεται ως εξής:

Στιγμιότυπο: Κατευθυνόμενοι γράφοι $G = (V, E)$ και $H = (V, E')$ (δηλαδή οι G, H σχηματίζονται από το ίδιο σύνολο κορυφών V, αλλά από διαφορετικά σύνολα ακμών).

Ερώτηση: Υπάρχει για κάθε ακμή του H $e \in E'$ μονοπάτι P_e στο G, έτσι ώστε:

(i) τα μονοπάτια να μην περιέχουν καμμία κοινή ακμή και
(ii) κάθε μονοπάτι P_e σε συνδυασμό με την ακμή e να σχηματίζουν κατευθυνόμενο κύκλο

Ο γράφος G ονομάζεται γράφος παροχής (**supply graph**) και ο H γράφος απαίτησης (**demand graph**). Οι **Even, Itai** και **Shamir** απέδειξαν στο ([24]) ότι το γενικό πρόβλημα των Μ.Τ.Μ. είναι $\mathcal{NP}$-πλήρες ακόμα και αν ο γράφος παροχής G είναι ακυκλικός. Θα αποδείξουμε αρχικά ότι μια ειδική περίπτωση του προβλήματος Μ.Τ.Μ. είναι $\mathcal{NP}$-πλήρης. Για το σκοπό αυτό θα χρειαστούμε τους δύο παρακάτω ορισμούς. Σημειώνουμε ότι ένας κατευθυνόμενος γράφος ονομάζεται **Eulerian** αν υπάρχει κύκλος που διέρχεται μία ακριβώς φορά από κάθε ακμή του (κύκλος **Euler**).



Σχήμα 5.1: Οι G, H είναι δύο γράφοι, οι οποίοι αποτελούνται από το ίδιο σύνολο κορυφών, αλλά από διαφορετικά σύνολα ακμών. Ο G είναι ένας ακυκλικός γράφος παροχής και ο H ένας γράφος απαίτησης.

Ορισμός 5.1
Ένας κατευθυνόμενος γράφος $G = (V, E)$ είναι **Eulerian** *ανν $\forall u \in V$, ισχύει $d^-_G(u) = d^+_G(u)$.*

Ορισμός 5.2
Έστω κατευθυνόμενοι γράφοι $G = (V, E)$ και $H = (V, E')$. Ορίζουμε το άθροισμα των G, H ως το γράφο $G + H$ που προκύπτει αν συνδέσουμε τις κορυφές του V χρησιμοποιώντας αμφότερα τα σύνολα ακμών των G και H, δηλαδή $G + H = (V, E \cup E')$.

Όταν οι γράφοι G, H είναι τέτοιοι ώστε ο $G + H$ να είναι **Eulerian**, το αντίστοιχο πρόβλημα ονομάζεται **Eulerian** πρόβλημα των μη τεμνόμενων μονοπατιών. Στο παρακάτω θεώρημα αποδεικνύουμε ότι και αυτό το πρόβλημα είναι $\mathcal{NP}$-πλήρες [25].

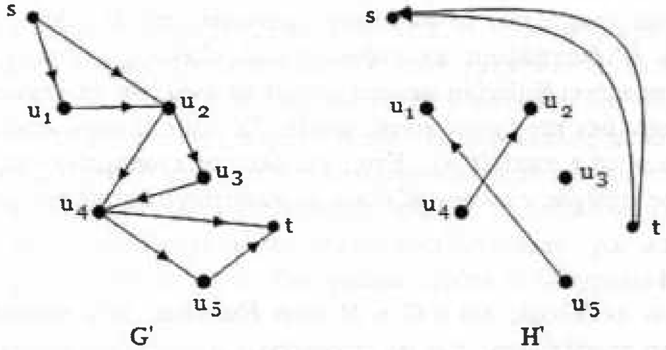
Θεώρημα 5.3
Το **Eulerian** *πρόβλημα των μη τεμνόμενων μονοπατιών είναι $\mathcal{NP}$-πλήρες.*

Απόδειξη. Έστω κατευθυνόμενοι γράφοι $G = (V, E)$ και $H = (V, E')$ και έστω ότι ο G είναι ακυκλικός (Σχήμα 5.1). Θεωρούμε το $\mathcal{NP}$-πλήρες πρόβλημα των μη τεμνόμενων μονοπατιών που ορίζεται από αυτούς.

Έστω G' ο τροποποιημένος γράφος που προκύπτει από το G αν προσθέσουμε τις κορυφές s, t και συμπληρώνουμε με ακμές, ως εξής:

- αν $d^-_{G+H}(u) < d^+_{G+H}(u)$, δηλαδή αν ο βαθμός εισόδου μιας κορυφής u στο $G+H$ είναι μικρότερος από το βαθμό εξόδου της, τότε προσθέτουμε $d^+_{G+H}(u) - d^-_{G+H}(u)$ ακμές (s, u) στο γράφο G', ενώ
- αν $d^-_{G+H}(u) > d^+_{G+H}(u)$, δηλαδή αν ο βαθμός εισόδου μιας κορυφής u στο $G+H$ είναι μεγαλύτερος από το βαθμό εξόδου της, τότε προσθέτουμε $d^-_{G+H}(u) - d^+_{G+H}(u)$ ακμές (u, t) στο γράφο G'.

Επίσης, έστω H' ο τροποποιημένος γράφος που προκύπτει από τον H αν προσθέσουμε στον H τις κορυφές s, t και $d^+_{G'}(s)$ ακμές (t, s). Η διαδικασία μετατροπής των G, H σε G', H' αναπαρίσταται στο Σχήμα 5.2.



Σχήμα 5.2: Οι τροποποιημένοι γράφοι G', H' που προκύπτουν αντίστοιχα από τους G, H του Σχήματος 5.1.

Θα δείξουμε ότι $d^+_{G'}(s) = d^-_{G'}(t)$. Έστω $u_1, u_2, \ldots, u_n$ οι κορυφές του G. Υποθέτουμε, χωρίς βλάβη της γενικότητας, ότι $n = 2m$, $m \in \mathbb{Z}$, και ότι για τις κορυφές u_i, όπου ο δείκτης i είναι περιττός, ισχύει $d^-_{G+H}(u_i) < d^+_{G+H}(u_i)$, ενώ για τις κορυφές u_i, όπου ο δείκτης i είναι άρτιος, ισχύει $d^-_{G+H}(u_i) > d^+_{G+H}(u_i)$. Έτσι, ισχύει:

$$d^+_{G'}(s) = \left(d^+_{G+H}(u_1) - d^-_{G+H}(u_1)\right) + \ldots + \left(d^+_{G+H}(u_{2m-1}) - d^-_{G+H}(u_{2m-1})\right)$$

$$d^-_{G'}(t) = \left(d^-_{G+H}(u_2) - d^+_{G+H}(u_2)\right) + \ldots + \left(d^-_{G+H}(u_{2m}) - d^+_{G+H}(u_{2m})\right)$$

Για να αποδείξουμε τον ισχυρισμό, έστω

$$d^+_{G'}(s) = d^-_{G'}(t) \Rightarrow$$

$$\Rightarrow d^+_{G+H}(u_1) + \ldots + d^+_{G+H}(u_{2m}) = d^-_{G+H}(u_1) + \ldots + d^-_{G+H}(u_{2m}) \Rightarrow$$

$$\Rightarrow \sum_{i=1}^{2m} d^+_{G+H}(u_i) = \sum_{i=1}^{2m} d^-_{G+H}(u_i)$$

το οποίο είναι αληθές.

Οι G' και H' (που κατασκευάσαμε από τους G και H αντίστοιχα) είναι τέτοιοι ώστε στον $G' + H'$ ο βαθμός εισόδου κάθε κορυφής να ισούται με το βαθμό εξόδου της. Έτσι, από τον Ορισμό 5.1, συνεπάγεται ότι ο $G' + H'$ είναι **Eulerian.** Θα αποδείξουμε ότι το πρόβλημα των Μ.Τ.Μ. λύνεται στο ζεύγος γραφών (G', H') ανν έχει λύση στους (G, H).

Έστω ότι διαθέτουμε μια λύση για τους γράφους G', H'. Τότε, εφόσον για κάθε ακμή $e \in E \cap E'$ το αντίστοιχο μονοπάτι P_e δε χρησιμοποιεί καμμία από τις νέες ακμές του G' (διότι αυτές θα χρησιμοποιηθούν για τα μονοπάτια που αντιστοιχούν στις ακμές (t, s) του E'), προκύπτει άμεσα μια λύση του προβλήματος στους γράφους G, H. Έστω τώρα μια λύση για τους G, H. Για κάθε ακμή του H $e \in E$, διαγράφουμε από το γράφο $G' + H'$ την εν λόγω ακμή και τις ακμές που απαρτίζουν το αντίστοιχο μονοπάτι της P_e. Από τον **Eulerian** γράφο $G' + H'$ διαγράψαμε κατευθυνόμενους κύκλους, οπότε ο γράφος που προέκυψε παραμένει **Eulerian** και άρα μπορεί να αναλυθεί σε ξεχωριστούς κύκλους, οι οποίοι δεν περιέχουν κοινές ακμές. Σε κάθε τέτοιον κύκλο περιέχεται ακριβώς μια φορά η ακμή (t, s). Έτσι, για όλες τις εναπομείναντες ακμές (t, s) του E', ο αντίστοιχος κύκλος ορίζει και το αντίστοιχο ζητούμενο μονοπάτι. □

Λήμμα 5.4
Αν ο G είναι ακυκλικός και ο $G + H$ είναι **Eulerian**, *τότε οποιαδήποτε λύση του* **Eulerian** *προβλήματος των μη τεμνόμενων μονοπατιών χρησιμοποιεί κάθε ακμή του γράφου G.*

Απόδειξη. Έστω μια λύση για το πρόβλημα των Μ.Τ.Μ.. Αν διαγράψουμε από το γράφο $G+H$ όλους τους κατευθυνόμενους κύκλους που ανήκουν στη λύση, ο γράφος που προκύπτει παραμένει **Eulerian.** Έχουμε διαγράψει όμως όλες τις ακμές του H, οπότε αυτός ο γράφος που προκύπτει είναι υπογράφος του G και επομένως ακυκλικός. Ένας **Eulerian** ακυκλικός γράφος, όμως, είναι ένας γράφος χωρίς ακμές. Αυτό σημαίνει ότι χρησιμοποιήσαμε όλες τις ακμές του G στη λύση. □

Για να αποδείξουμε ότι το πρόβλημα του χρωματισμού κορυφών σε γράφους κυκλικών τόξων είναι $\mathcal{NP}$-πλήρες [22], κάνουμε αναγωγή από το **Eulerian** πρόβλημα των Μ.Τ.Μ..

Θεώρημα 5.5
Το πρόβλημα του χρωματισμού κορυφών ενός γράφου κυκλικών τόξων είναι $\mathcal{NP}$-πλήρες.

Απόδειξη. Έστω κατευθυνόμενοι γράφοι $G = (V, E)$ και $H = (V, E')$, εκ των οποίων ο G είναι ακυκλικός. Οι G, H είναι τέτοιοι ώστε ο γράφος $G + H$ να είναι **Eulerian**. Εφόσον ο G είναι ακυκλικός, μπορούμε να διατάξουμε τις κορυφές του σε τοπολογική σειρά, έστω $u_1, u_2, \ldots, u_n$. Κατασκευάζουμε ένα γράφο κυκλικών τόξων G' ακολουθώντας την εξής διαδικασία: θεωρούμε έναν τυχαίο κύκλο και ορίζουμε πάνω του διαδοχικά τα σημεία $0, 1, 2, \ldots, n$, ακολουθώντας δεξιόστροφη φορά. Για κάθε ακμή $(u_i, u_j) \in E$ του G, $1 \leq i < j \leq n$, γράφουμε τόξο στην περιφέρεια του κύκλου από το σημείο i στο j. Για κάθε ακμή $(u_l, u_m) \in E'$ του H, γράφουμε αντίστοιχα τόξο από το l στο m. Υποθέτουμε ότι κάθε τέτοιο τόξο διέρχεται από το σημείο 0, δηλαδή $1 \leq m < l \leq n$. Πράγματι, αυτό ισχύει (για να έχει λύση το πρόβλημα Μ.Τ.Μ.), διότι διαφορετικά αν $m > l$, τότε η κορυφή u_m θα βρίσκεται δεξιότερα της u_l στην τοπολογική διάταξη (που υποθέσαμε για το G) και επομένως δε θα υπάρχει τρόπος να φτάσουμε από την u_m στην u_l χρησιμοποιώντας ακμές του G.

Έστω $|E'| = k$. Θα αποδείξουμε ότι ο G' χρωματίζεται με k χρώματα ανν υπάρχει λύση για το **Eulerian** πρόβλημα των Μ.Τ.Μ. στους γράφους G, H.

Έστω μια λύση του προβλήματος των Μ.Τ.Μ. για τους γράφους G, H. Από το Λήμμα 5.4, γνωρίζουμε ότι αυτή η λύση χρησιμοποιεί όλες τις ακμές του G, οπότε ο $G + H$ μπορεί να διασπαστεί σε k κατευθυνόμενους κύκλους, οι οποίοι δεν περιέχουν κοινές ακμές. Κάθε τέτοιος κύκλος απαρτίζεται από μια ακμή (u_l, u_m) του H και το αντίστοιχο μονοπάτι από την u_m στην u_l μέσω ακμών του G. Από τον τρόπο κατασκευής του G', τα τόξα που αντιστοιχούν στις ακμές ενός τέτοιου κύκλου δεν αλληλεπικαλύπτονται. 'ρα, αυτά τα τόξα μπορούν να χρωματιστούν με το ίδιο χρώμα, οπότε ο G' χρωματίζεται με k χρώματα.

Έστω τώρα ότι ο G' χρωματίζεται με k χρώματα. Αφού οι ακμές του H είναι k στο πλήθος, υπάρχουν ακριβώς k τόξα, τα οποία διέρχονται από το σημείο 0. Επομένως, καθένα εξ' αυτών λαμβάνει διαφορετικό χρώμα. Ο $G + H$ είναι ένας **Eulerian** γράφος, οπότε για κάθε κορυφή του u, ισχύει $d^-_{G+H}(u) = d^+_{G+H}(u)$. Αντίστοιχα, στην αναπαράσταση κυκλικών τόξων του G', ο αριθμός των τόξων που ξεκινούν από το σημείο i ισούται με τον αριθμό των τόξων που τερματίζουν στο i, $1 \leq i \leq n$. 'ρα, από κάθε σημείο του κύκλου διέρχονται ακριβώς k τόξα, τα οποία λαμβάνουν διαφορετικό χρώμα. Για κάθε ακμή του H $(u_l, u_m) \in E'$, θεωρούμε όλα τα τόξα, τα οποία έχουν το ίδιο χρώμα με το τόξο που αντιστοιχει στην εν λόγω ακμή. Αυτά καθορίζουν ένα μονοπάτι από την κορυφή u_m στην u_l, το οποίο, σε συνδυασμό με την ακμή (u_l, u_m), μας δίνει τη ζητούμενη λύση (διότι εφόσον από κάθε σημείο του κύκλου διέρχονται ακριβώς k τόξα, υπάρχουν k κατευθυνόμενοι κύκλοι χωρίς κοινές ακμές). □

5.3 Προσεγγιστικοί Αλγόριθμοι

5.3.1 Ένας 2-προσεγγιστικός αλγόριθμος

Ο Αλγόριθμος ES είναι ένας απλός 2-προσεγγιστικός αλγόριθμος για το χρωματισμό των κορυφών ενός γράφου κυκλικών τόξων G που δίδεται στο [26]. Σημειώνουμε ότι μια μεγιστική (**maximal**) κλίκα είναι μια κλίκα που δεν αποτελεί υποσύνολο κάποιας άλλης μεγαλύτερης.

Αλγόριθμος ES

1. Χρωμάτισε μια μεγιστική κλίκα του G;
2. Διέγραψέ την από τον G;
3. Έστω G' ο γράφος διαστημάτων που προκύπτει μετά τη διαγραφή;
4. Χρωμάτισε βέλτιστα τον G';

Θεώρημα 5.6
Ο Αλγόριθμος ES είναι ένας 2-προσεγγιστικός αλγόριθμος για το χρωματισμό των κορυφών ενός γράφου κυκλικών τόξων G.

Απόδειξη. Έστω K_{max} η μέγιστη κλίκα του G. Ισχύει $\chi(G) \geq K_{max}$. Ο Αλγόριθμος ES χρησιμοποιεί το πολύ K_{max} χρώματα για το χρωματισμό μιας μεγιστικής κλίκας του G και ακριβώς $\chi(G')$ χρώματα για να χρωματίσει βέλτιστα το γράφο διαστημάτων G' (Κεφάλαιο 2). Έτσι, ο συνολικός αριθμός των χρωμάτων που χρησιμοποιεί για το χρωματισμό των κορυφών του G είναι το πολύ:

$$K_{max} + \chi(G') \leq K_{max} + \chi(G) \leq 2\,\chi(G)$$

□

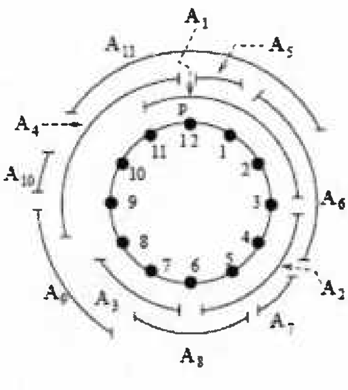
5.3.2 Ο Αλγόριθμος Tucker

Ο Tucker παρουσίασε στο [8] έναν αλγόριθμο για τον ορθό χρωματισμό των τόξων του F και κατ' επέκταση για το χρωματισμό των κορυφών του G. Πιθανολόγησε ότι ο αλγόριθμός του έχει λόγο προσέγγισης $\frac{3}{2}$, χωρίς όμως να καταφέρει να το αποδείξει.

Υποθέτουμε ότι τα τόξα του F είναι ανοιχτά, δηλαδή $A_i = (s_i, f_i)$, όπου s_i, $f_i \in \mathbb{Z}^+$ είναι αντίστοιχα το αριστερό και δεξιό άκρο του A_i κατά τη δεξιόστροφη φορά. Χωρίς βλάβη της γενικότητας, θεωρούμε ότι τα κυκλικά τόξα είναι τέτοια ώστε τα άκρα τους να ορίζουν m ισαπέχοντα σημεία πάνω στον κύκλο, όπου $m \leq 2n$ (Σχήμα 5.3). Εφόσον υποθέσαμε ότι τα κυκλικά τόξα είναι ανοιχτά, θεωρούμε ότι κανένα τόξο δεν περιέχει ως σημεία τα άκρα του.

Αλγόριθμος Tucker

1. Διάλεξε ένα σημείο p του κύκλου, τέτοιο ώστε το σύνολο των τόξων S_p που περιέχουν το p να είναι μεγέθους $L(F)$;
2. Έστω $A_1 = (s_1, f_1)$ το τόξο του S_p, του οποίου το αριστερό άκρο απέχει τη μικρότερη απόσταση από το p σε σχέση με τα υπόλοιπα τόξα του S_p;
3. Θέσε $col = 1$;
4. Ανάθεσε το χρώμα col στο τόξο A_1;
5. **for** $i = 2, \ldots, n$ **do**
6.   Θεώρησε ως $A_i = (s_i, f_i)$ το πρώτο (κατά τη δεξιόστροφη φορά) μη χρωματισμένο τόξο, για το οποίο ισχύει $s_i > f_{i-1}$;
7.   **if** υπάρχει τόξο A_j, $j < i$, που επικαλύπτεται με το A_i και είναι χρωματισμένο με χρώμα col **then**
8.   Θέσε $col = col + 1$;
9.   Χρωμάτισε το A_i με το χρώμα col;
10.   **else** χρωμάτισε το A_i με το χρώμα col;

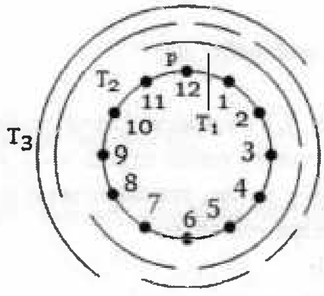


Σχήμα 5.3: Παράδειγμα εκτέλεσης του Αλγορίθμου Tucker. Το σύνολο F αποτελείται από 11 τόξα. Μετά την ολοκλήρωση του αλγορίθμου, τα τόξα A_1, A_2, A_3 έχουν χρωματιστεί με το χρώμα 1, τα A_4, A_5, A_6, A_7, A_8 με το χρώμα 2, ενώ τα A_9, A_{10}, A_{11} με το χρώμα 3.

Θα αναλύσουμε τον Αλγόριθμο Tucker σύμφωνα με το [27], ούτως ώστε να δούμε πόσο καλά προσεγγίζει τη βέλτιστη λύση του προβλήματος χρωματισμού κορυφών ενός γράφου κυκλικών τόξων.

Λέμε ότι ο Αλγόριθμος Tucker έχει ολοκληρώσει k βήματα, αν έχει διασχίσει $k+1$ φορές το σημείο s_1 κατά την εκτέλεσή του (όπου s_1 είναι το αριστερό άκρο του πρώτου τόξου A_1 που χρωματίζεται). Έστω S_k το σύνολο των τόξων που χρωματίστηκαν κατά τη διάρκεια του k-οστού βήματος και F_k το σύνολο των τόξων που έχουν χρωματιστεί από την αρχή έως και το βήμα k, δηλαδή $F_k = \bigcup_{i=1}^{k} S_i$. Ορίζουμε ως κυκλική κάλυψη l του F τον ελάχιστο αριθμό τόξων, των οποίων η ένωση περιέχει όλα τα σημεία του κύκλου. Λέμε ότι αυτά τα τόξα καλύπτουν πλήρως τον κύκλο.

Στη συνέχεια θα αποδείξουμε μια σειρά από λήμματα χρήσιμα για την ανάλυση του Αλγόριθμου Tucker.



Σχήμα 5.4: Παράδειγμα για το Λήμμα 5.9, όπου $T_3 = T_3'$.

Λήμμα 5.7
Έστω T_i το πρώτο τόξο του F που χρωματίστηκε με χρώμα i, $i = 1, 2, \ldots$. Τότε, για κάθε $i > 1$, τα τόξα T_i και T_{i-1} αλληλεπικαλύπτονται.

Απόδειξη. Προφανές, διότι το T_i είναι το πρώτο τόξο που συναντάμε (κινούμενοι δεξιόστροφα πάνω στον κύκλο), στο οποίο δεν μπορούμε να αναθέσουμε το χρώμα $i-1$. 'ρα, το T_i θα πρέπει να επικαλύπτεται με κάποιο τόξο χρώματος $i-1$. Λόγω της δεξιόστροφης κίνησης του Αλγόριθμου Tucker, το πρώτο τόξο χρώματος $i-1$ που επικαλύπτει το T_i είναι το T_{i-1}. □

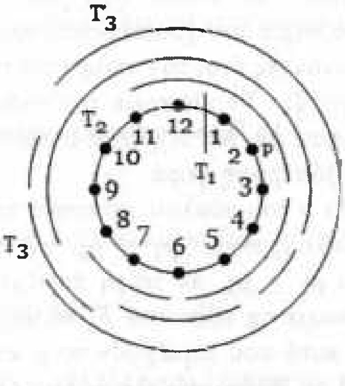
Λήμμα 5.8
Για κάθε βήμα $k \geq 1$, ισχύει $L(F \setminus F_k) \leq L(F) - k$, όπου $L(F \setminus F_k)$ είναι ο φόρτος του συνόλου τόξων $F \setminus F_k$.

Απόδειξη. Προφανές, γιατί μετά την ολοκλήρωση του k-οστού βήματος, έχουμε διανύσει k φορές την περιφέρεια του κύκλου (οπότε ισχύει η ισότητα) και έχουμε περάσει $k+1$ φορές το αριστερό άκρο s_1 του A_1 (οπότε ενδεχομένως ισχύει η ανισότητα). □

Λήμμα 5.9
Μετά την ολοκλήρωση του βήματος k, $1 \leq k \leq l-2$, ο Αλγόριθμος Tucker έχει χρησιμοποιήσει το πολύ $k+1$ χρώματα για το χρωματισμό των τόξων του F_k.

Απόδειξη. Η απόδειξη θα γίνει με επαγωγή στον αριθμό των βημάτων k. Για $k = 1$, ο ισχυρισμός προφανώς ισχύει. Έστω $k > 1$. Ορίζουμε ως T_k και T'_k, αντίστοιχα, το πρώτο και το τελευταίο τόξο του S_{k-1}, το οποίο χρωματίστηκε με χρώμα k. Σημειώνουμε ότι τα T_k και T'_k ενδέχεται να είναι το ίδιο τόξο. Υποθέτουμε ότι το λήμμα ισχύει για όλα τα βήματα μέχρι και το $k-1$. Θα δείξουμε ότι ισχύει και για το k-οστό βήμα.

Το T_{k+1} είναι το πρώτο τόξο του συνόλου S_k, το οποίο χρωματίστηκε με χρώμα $k+1$, οπότε από το Λήμμα 5.7 επικαλύπτεται με το T_k. Υποθέτουμε ότι το T'_{k+1} είναι το τελευταίο τόξο του S_k και έστω ότι δεν μπορεί να χρωματιστεί με το χρώμα $k+1$. Έτσι, έστω ότι χρωματίζουμε το T'_{k+1} με χρώμα $k+2$ (οπότε χρησιμοποιούμε περισσότερα από $k+1$ χρώματα για το χρωματισμό των τόξων του F_k). Από το Λήμμα 5.7, τα T_{k+1} και T'_{k+1} αλληλεπικαλύπτονται (διότι το T_{k+1} είναι εξ' ορισμού το πρώτο τόξο που χρωματίζεται με χρώμα $k+1$ και το T'_{k+1} το πρώτο που χρωματίζεται με χρώμα $k+2$). Επομένως, τα τόξα $T_2, T_3, \ldots, T_k, T_{k+1}, T'_{k+1}$ ($k+1$ στο πλήθος) καλύπτουν πλήρως τον κύκλο, δηλαδή $l = k+1$. Όμως, $k \leq l-2 \Rightarrow l \geq k+2$, οπότε οδηγούμαστε σε αντίφαση. 'ρα, μπορούμε να χρωματίσουμε το T'_{k+1} με χρώμα $k+1$ και έτσι αποδεικνύεται το ζητούμενο. □



Σχήμα 5.5: Παράδειγμα για το Λήμμα 5.9, όπου υποθέτουμε ότι δεν μπορούμε να χρωματίσουμε το τόξο $T'_{k+1} = T'_3$ με το χρώμα 3 και οδηγούμαστε σε άτοπο.

Θεώρημα 5.10
Αν $l \geq 5$, τότε ο Αλγόριθμος Tucker χρησιμοποιεί το πολύ $\lceil \frac{(l-1)}{(l-2)} L \rceil$ χρώματα για το χρωματισμό ενός γράφου κυκλικών τόξων G.

Απόδειξη. Θα αποδείξουμε το ζητούμενο με επαγωγή στο L. Για $L = 1$, ισχύει προφανώς, καθότι αρκούν δύο χρώματα για το χρωματισμό του F. Υποθέτουμε ότι το ζητούμενο ισχύει για κάθε φόρτο του F έως $L-1$. Θα δείξουμε ότι ισχύει και για L.

Από το Λήμμα 5.8, μετά την ολοκλήρωση του βήματος $k = l - 2$, ισχύει $L(F \setminus F_{l-2}) \leq L - (l-2) < L$. Έτσι, εμπίπτουμε στην επαγωγική υπόθεση, οπότε τα επαρκούντα χρώματα για το χρωματισμό των τόξων του $F \setminus F_{l-2}$ είναι:

$$\frac{(l-1)}{(l-2)} \cdot L(F \setminus F_{l-2}) \leq \frac{(l-1)}{(l-2)} \cdot \Big(L - (l-2)\Big) = \frac{(l-1)}{(l-2)} \cdot L - (l-1)$$

Από το Λήμμα 5.9, ο Αλγόριθμος **Tucker** χρησιμοποιεί το πολύ $l-1$ χρώματα για το χρωματισμό των τόξων του F_{l-2}. Συνεπώς, ο συνολικός αριθμός των χρωμάτων που επαρκούν για το χρωματισμό των τόξων του $F = (F \setminus F_{l-2}) \cup F_{l-2}$ είναι:

$$\frac{(l-1)}{(l-2)} \cdot L - (l-1) + (l-1) = \frac{(l-1)}{(l-2)} \cdot L \leq \lceil \frac{(l-1)}{(l-2)} \rceil \cdot L$$

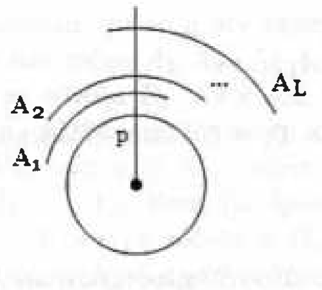
οπότε αποδεικνύεται το ζητούμενο. □

5.3.3 Ο Αλγόριθμος Karapetian

Ο **Iskandar A. Karapetian** παρουσίασε στο [23] έναν αλγόριθμο με λόγο προσέγγισης $\frac{3}{2}$ για το πρόβλημα του χρωματισμού κορυφών ενός γράφου κυκλικών τόξων G, επιβεβαιώνοντας έτσι τον ισχυρισμό του **Tucker** ότι το πρόβλημα προσεγγίζεται με λόγο $\frac{3}{2}$. Υποθέτουμε ότι κάθε τόξο A_i είναι κλειστό, δηλαδή $A_i = [A_{s_i}, A_{f_i}]$, όπου A_{s_i} και A_{f_i} είναι αντίστοιχα η αρχή και το πέρας του τόξου A_i κατά τη δεξιόστροφη φορά.

Επιλέγουμε ένα σημείο p του κύκλου, το οποίο περιέχεται σε L τόξα. Για δύο σημεία p_1, p_2 του κύκλου, ισχύει $p_1 \leq p_2$, αν το p_1 περιέχεται στο δεξιόστροφο τόξο $[p, p_2]$ και $p_1 \geq p_2$, αν το p_1 περιέχεται στο αριστερόστροφο τόξο $[p, p_2]$. Αναδιατάσσουμε τα τόξα του F, ούτως ώστε τα πρώτα L τόξα, $A_1, A_2, \ldots, A_L$, να είναι αυτά που περιέχουν το p και $\forall i, j = 1, \ldots, L$, όπου $i < j$, να ισχύει $A_{f_i} < A_{f_j}$. Έστω σύνολο τόξων Q και τόξο $g \in Q$, τέτοιο ώστε $g_s \leq q_s$, $\forall q \in Q$. Τότε, ορίζουμε την αρχή του συνόλου Q ως $Q_s = g_s$. Σε πλήρη αντιστοιχία, $Q_f = g_f$, όπου $g \in Q$, τέτοιο ώστε $g_f \geq q_f$, $\forall q \in Q$. Επίσης, ορίζουμε ως $\cap Q$ το τόξο αλληλεπικάλυψης όλων των τόξων του Q για οποιοδήποτε σύνολο τόξων Q.

Ο Αλγόριθμος **Karapetian** χρησιμοποιεί τους Αλγόριθμους **Clockwise**(F, g) και **Counter-Clockwise**(F, g). Ο Αλγόριθμος **Clockwise**(F, g), δεδομένου ενός συνόλου τόξων F και ενός τόξου $g \in F$, σαρώνει τον κύκλο κατά τη δεξιόστροφη φορά (ξεκινώντας από το g) και επιστρέφει σύνολο τόξων S, του οποίου τα τόξα δεν επικαλύπτονται. Ο Αλγόριθμος **Counter-Clockwise**(F, g) είναι όμοιος με τον Αλγόριθμο **Clockwise**(F, g), με μόνη διαφορά ότι σαρώνει τον κύκλο κατά την αριστερόστροφη φορά.



Σχήμα 5.6: Τα πρώτα L τόξα του F περιέχουν το σημείο p και είναι διατεταγμένα σύμφωνα με το δεξιό άκρο τους.

Αλγόριθμος S = Clockwise(F,g)

$S = \{g\}$;
$Q = \{q \in F :$ το q δεν αλληλεπικαλύπτεται με το $g\}$;
while $Q \neq \emptyset$
    Επέλεξε $h \in Q$, τέτοιο ώστε $h_s = Q_s$;
    $S = S \cup \{h\}$;
    $Q = Q \setminus \{q \in Q :$ το q αλληλεπικαλύπτεται με το $h\}$;

Αλγόριθμος S = Counter-Clockwise(F,g)

$S = \{g\}$;
$Q = \{q \in F :$ το q δεν αλληλεπικαλύπτεται με το $g\}$;
while $Q \neq \emptyset$
    Επέλεξε $h \in Q$, τέτοιο ώστε $h_f = Q_f$;
    $S = S \cup \{h\}$;
    $Q = Q \setminus \{q \in Q :$ το q αλληλεπικαλύπτεται με το $h\}$;

Εκ κατασκευής, τα τόξα του συνόλου S που κατασκευάζει ο Αλγόριθμος Clockwise(F, g) δεν επικαλύπτονται και συνεπώς μπορούν να λάβουν το ίδιο χρώμα. Ομοίως για τον Αλγόριθμο Counter-Clockwise(F, g).

Αλγόριθμος Karapetian(F)

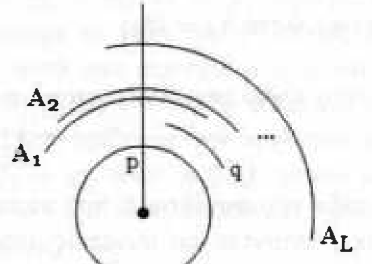
$Q = F \setminus \{A_1, A_2, \ldots, A_L\}$;
for $i = 1, \ldots, \lfloor \frac{L}{2} \rfloor$;
    Επέλεξε $q_i \in Q$, τέτοιο ώστε $q_{s_i} = Q_s$;
    Χρωμάτισε το σύνολο $B_i = Clockwise(Q, q_i)$ με το χρώμα $L + i$;

5. $\quad Q = Q \setminus B_i$;
6. $Q = Q \cup \{A_1, A_2, \ldots, A_L\}$;
7. **for** $i = L, \ldots, 1$;
8. $\quad$ Χρωμάτισε το σύνολο $D_i = \mathit{Counter\text{-}Clockwise}(Q, A_i)$ με το χρώμα i;
9. $\quad Q = Q \setminus \{D_i\}$;

Εφόσον τα τόξα του συνόλου S (αμφότερων των Αλγορίθμων **Clockwise**(F, g) και **Counter-Clockwise**(F, g)) μπορούν να λάβουν το ίδιο χρώμα, συνεπάγεται ότι στις Γραμμές 4 και 8 του Αλγόριθμου **Karapetian** μπορούμε να χρωματίσουμε κάθε ένα από τα σύνολα B_i και D_i με ένα χρώμα. Θα δείξουμε ότι μετά την ολοκλήρωση του Αλγόριθμου **Karapetian** το σύνολο Q είναι κενό, οπότε επαρκούν $L + \lfloor \frac{L}{2} \rfloor = \lfloor \frac{3}{2} L \rfloor$ χρώματα για το χρωματισμό των τόξων του F.

Λήμμα 5.11
Αν μετά την ολοκλήρωση του βρόχου στη Γραμμή 2 του Αλγόριθμου **Karapetian** *υπάρχει τόξο $q \in Q$, τότε $\forall i = 1, \ldots, \lfloor \frac{L}{2} \rfloor$ υπάρχει τόξο $b_i \in B_i$, τέτοιο ώστε το b_i να περιέχει το σημείο q_s.*

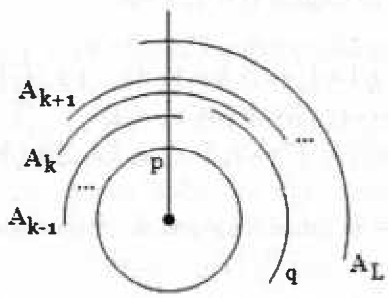
Απόδειξη. Εφόσον το τόξο q ανήκει στο Q μετά την ολοκλήρωση του βρόχου της Γραμμής 2, έπεται ότι, αν και ήταν υποψήφιο σε κάθε βήμα i για να εισέλθει στο σύνολο B_i (Γραμμή 4), τελικά δεν επιλέχτηκε. Αυτό σημαίνει ότι σε κάθε σύνολο B_i υπάρχει τόξο b_i, το οποίο αλληλεπικαλύπτεται με το q. Επίσης, ισχύει ότι $b_{s_i} \leq q_s$, διότι διαφορετικά (αν $b_{s_i} > q_s$) ο Αλγόριθμος **Clockwise** θα επέλεγε το τόξο q και όχι το b_i. Συνεπώς, το b_i περιέχει το σημείο q_s. □



Σχήμα 5.7: Παράδειγμα για το Λήμμα 5.12 στην περίπτωση όπου $k = 1$.

Λήμμα 5.12
Αν μετά την ολοκλήρωση του βρόχου στη Γραμμή 7 του Αλγόριθμου **Karapetian** *υπάρχει τόξο $q \in Q$, τότε υπάρχει δείκτης k, $2 \leq k \leq L$, τέτοιος ώστε για τα άκρα των τόξων q, A_{k-1} και A_k να ισχύει $q_s \geq A_{f_{k-1}}$ και $q_s \leq A_{f_k}$.*

Απόδειξη. Αρχικά, σημειώνουμε ότι το q δεν περιέχει το σημείο p, διότι διαφορετικά θα ήταν ένα εκ των τόξων $A_1, A_2, \ldots, A_L$, οπότε θα είχε προφανώς προστεθεί σε κάποιο από τα σύνολα D_i. Αν $k=1$, τότε $q_s \geq p$ και $q_s \leq A_{f_1}$, οπότε τα τόξα $q, A_1, \ldots, A_L$ αλληλεπικαλύπτονται και άρα ο φόρτος του F είναι $L+1$. 'τοπο. Αν $k > L$, τότε $q_s \geq A_{f_L}$, οπότε το q δεν επικαλύπτεται με κανένα από τα τόξα $A_1, A_2, \ldots, A_L$. Έτσι, (με όμοια απόδειξη όπως αυτή του Λήμματος 5.11), $\forall i = 1, \ldots, L$ υπάρχει τόξο $d_i \in D_i$, τέτοιο ώστε το d_i να περιέχει το αριστερό άκρο q_f του q. Αυτό σημαίνει ότι τα τόξα $q, d_1, \ldots, d_L$ αλληλεπικαλύπτονται. 'τοπο, διότι ο φόρτος του F είναι L. Συνεπώς, $2 \leq k \leq L$. □



Σχήμα 5.8: Παράδειγμα για το Λήμμα 5.12. Τα τόξα $q, A_k, \ldots, A_L$ αλληλεπικαλύπτονται.

Λήμμα 5.13
Αν μετά την ολοκλήρωση του βρόχου στη Γραμμή 7 του Αλγόριθμου **Karapetian** *υπάρχει τόξο $q \in Q$, για το οποίο $q_s \leq A_{f_k}$ και $q_s \geq A_{f_{k-1}}$, τότε για κάθε $h \in (\cup_{i=k}^{L} D_i) \setminus \{A_k, \ldots, A_L\}$, ισχύει $h_s \leq q_f$.*

Απόδειξη. Έστω ότι υπάρχει τόξο $h \in (\cup_{i=k}^{L} D_i) \setminus \{A_k, \ldots, A_L\}$, τέτοιο ώστε $h_s \geq q_f$ και θα καταλήξουμε σε άτοπο. Εφόσον $q_s \geq A_{f_{k-1}}$, το q δεν αλληλεπικαλύπτεται με τα A_i, $i = 1, \ldots, k-1$. Όμως, μετά την ολοκλήρωση του Βήματος 7 του Αλγόριθμου **Karapetian,** **isq'uei 'oti** $q \in Q$, οπότε υπάρχουν τόξα $d_i \in D_i$, $i = 1, \ldots, k-1$, τα οποία περιέχουν το σημείο q_f. Ορίζουμε τα σύνολα

$$U = \{d_i \in D_i : q_f \in d_i,\ h_s \in d_i,\ 1 \leq i \leq k-1\}$$

και

$$V = \{d_i \in D_i : q_f \in d_i,\ h_s \notin d_i,\ 1 \leq i \leq k-1\} \cup \{q\}$$

Από τον ορισμό των συνόλων U και V, ισχύει ότι $V = (\{d_1, \ldots, d_{k-1}\} \setminus U) \cup \{q\}$. Θα δείξουμε ότι

$$k - 1 + \lfloor \frac{L}{2} \rfloor - L < |U| \leq \lfloor \frac{L}{2} \rfloor$$

Το h δε χρωματίστηκε μετά την ολοκλήρωση του βρόχου της Γραμμής 2 του Αλγόριθμου **Karapetian**, οπότε από το Λήμμα 5.11 υπάρχουν τουλάχιστον $\lfloor \frac{L}{2} \rfloor$ τόξα (τουλάχιστον ένα από κάθε σύνολο B_i), τα οποία περιέχουν το σημείο h_s. Έτσι, το συνολικό πλήθος των τόξων που περιέχουν το h_s είναι $|U|+1+\lfloor \frac{L}{2} \rfloor$. Επειδή ο φόρτος του F είναι ίσος με L, ισχύει:

$$|U|+1+\lfloor \frac{L}{2} \rfloor \leq L \Rightarrow |U| \leq \lfloor \frac{L}{2} \rfloor$$

Υποθέτουμε ότι $k-1+\lfloor \frac{L}{2} \rfloor - L \geq L$ και θα καταλήξουμε σε άτοπο. Όλα τα τόξα του V περιέχουν το σημείο q_f και ικανοποιούν τις συνθήκες του Λήμματος 5.11, οπότε ο φόρτος στο σημείο $(\cap V)_s$ είναι

$$|V|+\lfloor \frac{L}{2} \rfloor = k-1-|U|+1+\lfloor \frac{L}{2} \rfloor \geq k-1-(k-1+\lfloor \frac{L}{2} \rfloor - L)+1+\lfloor \frac{L}{2} \rfloor = L+1$$

'τοπο, διότι ο φόρτος του F είναι L. 'ρα, $k-1+\lfloor \frac{L}{2} \rfloor - L \geq |U|$. Ορίζουμε τα σύνολα

$$I = \{i : d_i \in D_i | h_s \in d_i,\ k \leq i \leq L\}$$

και

$$J = \{i : d_i \in D_i | h_s \notin d_i,\ k \leq i \leq L\}$$

Προφανώς, $J = \{k, \ldots, L\} \setminus I$. Το πλήθος των τόξων που περιέχουν το h_s είναι $|I|+|U|+1+\lfloor \frac{L}{2} \rfloor$. Ισχύει, $|I|+|U|+1+\lfloor \frac{L}{2} \rfloor \leq L$, οπότε $|I| \leq \lfloor \frac{L}{2} \rfloor - |U|$ και $|J| \geq L-(k-1)-\lfloor \frac{L}{2} \rfloor + |U|$. Υποθέτουμε ότι για κάθε $i \in J$ υπάρχει τόξο z_i που περιέχει το σημείο q_f. Ορίζουμε το σύνολο

$$Z = \{z_i : i \in J\} \cup V$$

Όλα τα τόξα του Z περιέχουν το σημείο q_f και ικανοποιούν τις συνθήκες του Λήμματος 5.11, οπότε ο φόρτος του σημείου $(\cap Z)_s$ είναι:

$$|Z|+\lfloor \frac{L}{2} \rfloor = |J|+|V|+\lfloor \frac{L}{2} \rfloor \geq L-(k-1)-\lfloor \frac{L}{2} \rfloor+|U|+k-1-|U|+1+\lfloor \frac{L}{2} \rfloor = L+1$$

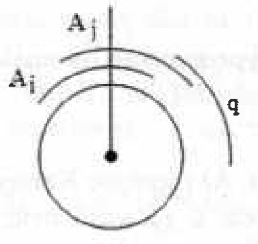
'τοπο, και άρα το σημείο q_f δεν περιέχεται σε όλα τα $z_i, i \in J$.

Έστω $j \in J$ ο μικρότερος δείκτης, για τον οποίο κανένα τόξο του D_j δεν περιέχει το q_f. Ορίζουμε

$$J' = \{i : d_i \in D_i | h_s \notin d_i,\ k \leq i < j\}$$

Έτσι, τα τόξα $d_i \in D_i$, $i \in J'$ περιέχουν το q_f. Εφόσον $q_f \notin D_j$, $A_{f_j} \leq q_f$, οπότε $A_{f_i} \leq A_{f_j} \leq q_f$. Δηλαδή, για κάθε $i \in J'$, υπάρχει τόξο $d_i \in D_i$ που περιέχει το q_f, αλλά αυτό δεν είναι το A_i.

Θεωρούμε το σύνολο $V \cup \{d_i : i \in J'\}$. Παρατηρούμε ότι αποτελείται από τόξα, τα οποία περιέχουν το σημείο q_f, αλλά όχι το h_s (δηλαδή δεν καλύπτουν το διάστημα ανάμεσα στα q_f και h_s). Έτσι, κάθε τόξο του συνόλου $V \cup \{d_i :$



Σχήμα 5.9: Παράδειγμα για το Λήμμα 5.13. Το τόξο A_j δεν περιέχει το σημείο q_f, οπότε ισχύει $A_{f_i} \le A_{f_j} \le q_f$.

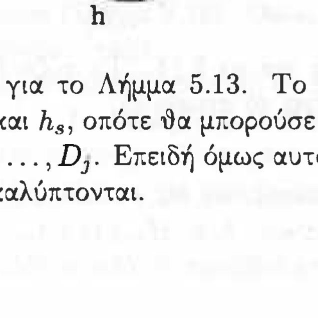
$i \in J'\}$ είναι υποψήφιο για να συμπεριληφθεί τουλάχιστον σε κάποιο από τα σύνολα $D_L, \ldots, D_j$, εκτός και αν αλληλεπικαλύπτεται με το A_j (σημειώνουμε ότι λόγω της διάταξης $A_1, \ldots, A_L$, αν ένα τόξο αλληλεπικαλυπτεται με το A_j, τότε θα αλληλεπικαλύπτεται και με τα $A_{j+1}, \ldots, A_L$).

Όμως, δεδομένου ότι κανένα τόξο δεν έχει εισέλθει σε κάποιο εκ των συνόλων $D_L, \ldots, D_j$, συνεπάγεται ότι καθένα εξ' αυτών αλληλεπικαλύπτεται με το A_j. Έτσι, ο φόρτος στο σημείο $(\cap(V \cup \{d_i : i \in J'\}))_s$ είναι:

$$|V \cup \{d_i : i \in J'\}| + |\{A_j, \ldots, A_L\}| + \lfloor \frac{L}{2} \rfloor = |V| + |J| + \lfloor \frac{L}{2} \rfloor \ge$$

$$\ge k - 1 - |U| + 1 + L - (k-1) - \lfloor \frac{L}{2} \rfloor + |U| + \lfloor \frac{L}{2} \rfloor = L + 1$$

Άτοπο. Συνεπώς, δεν υπάρχει $h \in (\cup_{i=k}^{L} D_i) \setminus \{A_k, \ldots, A_L\}$, τέτοιο ώστε $h_s \ge q_f$. □



Σχήμα 5.10: Παράδειγμα για το Λήμμα 5.13. Το τόξο d_i δεν καλύπτει το διάστημα ανάμεσα στα q_f και h_s, οπότε θα μπορούσε να είχε συμπεριληφθεί σε κάποιο από τα σύνολα $D_L, \ldots, D_j$. Επειδή όμως αυτό δε συμβαίνει, συνεπάγεται ότι τα d_i, A_j αλληλεπικαλύπτονται.

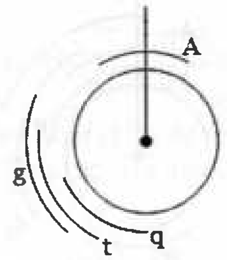
Θεώρημα 5.14
Ο Αλγόριθμος **Karapetian** *χρησιμοποιεί το πολύ $\lfloor \frac{3}{2} L \rfloor$ χρώματα για το χρωματισμό ενός γράφου κυκλικών τόξων.*

Απόδειξη. Παρατηρούμε ότι ο Αλγόριθμος **Karapetian** χρησιμοποιεί $\lfloor \frac{L}{2} \rfloor$ χρώματα στις Γραμμές 2 και 4 και L χρώματα στις Γραμμές 7 και 8 για να χρωματίσει τα τόξα του F. Ως εκ τούτου, αν μετά την ολοκλήρωσή του το σύνολο Q είναι κενό, τότε το ζητούμενο έχει αποδειχθεί. Υποθέτουμε ότι μετά το τέλος του Αλγόριθμου **Karapetian** $Q \neq \emptyset$ και θα καταλήξουμε σε άτοπο. Έστω $q \in Q$. Από το Λήμμα 5.12, υπάρχει δείκτης k, τέτοιος ώστε $q_s \geq A_{f_{k-1}}$ και $q_s \leq A_{f_k}$. Εφόσον $q_s \geq A_{f_{k-1}}$, έπεται ότι υπάρχουν τόξα $d_i \in D_i$, $i = 1, \ldots, k-1$, τα οποία περιέχουν το σημείο q_f.

Ορίζουμε ως g_i το δεύτερο τόξο που προστίθεται στο σύνολο D_i (το πρώτο είναι το A_i). Κατασκευάζουμε σύνολα τόξων $H_{k-1}, \ldots, H_L$ ως εξής:

1. $H_{k-1} = \{d_1, \ldots, d_{k-1}\} \cup \{q\}$;
2. **for** $i = k-1, \ldots, L-1$
3. **if** το A_{i+1} αλληλεπικαλύπτεται με όλα τα τόξα του H_i **then**
4. $H_{i+1} = H_i \cup \{A_{i+1}\}$;
5. **else** $H_{i+1} = H_i \cup \{g_{i+1}\}$;

Θα δείξουμε ότι τα τόξα του H_i, $i = k-1, \ldots, L$ αλληλεπικαλύπτονται. Υποθέτουμε ότι αυτό δεν είναι αληθές και έστω j ο μικρότερος δείκτης, για τον οποίο τα τόξα του H_j δεν αλληλεπικαλύπτονται. Προφανώς, $j > k-1$ (διότι όλα τα τόξα του H_{k-1} αλληλεπικαλύπτονται, δεδομένου ότι όλα περιέχουν το q_f) και $H_j = H_{j-1} \cup \{g_j\}$ (διότι εκ κατασκευής των συνόλων, αν $H_j = H_{j-1} \cup \{A_j\}$, τότε το A_j θα επικάλυπτεται με όλα τα τόξα του H_{j-1}). Υπάρχει, συνεπώς, τόξο $h \in H_{j-1}$, το οποίο δεν επικαλύπτεται με το g_j.



Σχήμα 5.11: Παράδειγμα για το 5.14. Το τόξο t περιέχει το q_f, $t_f \leq g_f$, $g_s \leq q_f$, οπότε το g περιέχει το σημείο q_f.

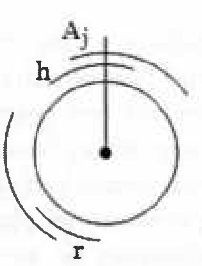
Από τον τρόπο κατασκευής των H_i, σε κάθε βήμα προσθέτουμε είτε κάποιο A_i είτε κάποιο g_i. Επομένως, $h \in H_{j-1} \cap \{A_k, \ldots, A_{j-1}\}$ ή $h \in (H_{j-1} \cap \{g_k, \ldots, g_{j-1}\}) \cup H_{k-1}$. Θα δείξουμε ότι όλα τα τόξα του $H_{j-1} \cap \{g_k, \ldots, g_{j-1}\}$

περιέχουν το σημείο q_f. Τότε, επειδή όλα τα τόξα του H_{k-1} περιέχουν το σημείο q_f, όλα τα τόξα του $(H_{j-1} \cap \{g_k, \ldots, g_{j-1}\}) \cup H_{k-1}$ θα αλληλεπικαλύπτονται. Έτσι, το h θα ανήκει στο $H_{j-1} \cap \{A_k, \ldots, A_{j-1}\}$.

Ορίζουμε την ακολουθία δεικτών $i_1, \ldots, i_m$, τέτοια ώστε

$$H_{j-1} \cap \{g_k, \ldots, g_{j-1}\} = \{g_{i_1}, \ldots, g_{i_m}\}$$

Υποθέτουμε ότι τα τόξα $g_{i_1}, \ldots, g_{i_r}$ περιέχουν το σημείο q_f. Θα δείξουμε ότι και το $g_{i_{r+1}}$ περιέχει το q_f. Επειδή $g_{i_{r+1}} \in H_{j-1}$, ισχύει ότι $H_{i_{r+1}} = H_{i_{r+1}-1} \cup \{g_{i_{r+1}}\}$. Ως εκ τούτου, υπάρχει τόξο $t \in H_{k-1} \cup \{g_{i_1}, \ldots, g_{i_r}\}$, το οποίο δεν επικαλύπτεται με το $A_{i_{r+1}}$ (διαφορετικά θα ίσχυε $H_{i_{r+1}} = H_{i_{r+1}-1} \cup \{A_{i_{r+1}}\}$) και, λόγω της επαγωγής, περιέχει το q_f. Συνεπάγεται, λοιπόν, ότι το t ήταν υποψήφιο για να εισαχθεί στο $D_{i_{r+1}}$. Ο αλγόριθμος, όμως, επέλεξε το $g_{i_{r+1}}$ ως δεύτερο τόξο του $D_{i_{r+1}}$, οπότε $t_f \leq g_{f_{i_{r+1}}}$. Από το Λήμμα 5.13, $g_{s_{i_{r+1}}} \leq q_f$ και έτσι το $g_{i_{r+1}}$ περιέχει το q_f (Σχήμα 5.11). Όλα, επομένως, τα τόξα του $H_{j-1} \cap \{g_k, \ldots, g_{j-1}\}$ αλληλεπικαλύπτονται, οπότε $h \in H_{j-1} \cap \{A_k, \ldots, A_{j-1}\}$.



Σχήμα 5.12: Παράδειγμα για το 5.14. Παρατηρούμε ότι τα τόξα h, r δεν επικαλύπτονται, αλλά εφόσον ανήκουν αμφότερα στο H_{j-1} οδηγούμαστε σε αντίφαση.

Δεδομένου ότι $H_j = H_{j-1} \cup \{g_j\}$, γνωρίζουμε ότι υπάρχει τόξο $r \in H_{j-1} \setminus \{A_1, \ldots, A_{j-1}\}$, το οποίο δεν επικαλύπεται με το A_j (αλλιώς $H_j = H_{j-1} \cup \{A_j\}$). Το r είναι υποψήφιο για να εισέλθει στο D_j, αλλά εφόσον ο αλγόριθμος επέλεξε το g_j, θα πρέπει $r_f \leq g_{f_j}$. Έχουμε ότι $h \in H_{j-1} \cap \{A_k, \ldots, A_{j-1}\}$, δηλαδή $h = A_i$ για κάποιο $i \in \{k, \ldots, j-1\}$. Έτσι, $h_f \leq A_{f_j}$ και επειδή ορίσαμε το δείκτη j τέτοιον ώστε τα h, g_j να μην επικαλύπτονται, συνεπάγεται ότι τα h, r δεν επικαλύπτονται (Σχήμα 5.12). Όμως, $h \in H_{j-1}$ και $r \in H_{j-1}$, δηλαδή τα h, g_j επικαλύπτονται. 'τοπο.

Συμπεραίνουμε, συνεπώς, ότι τα τόξα του H_i, $k-1 \leq i \leq L$ αλληλεπικαλύπτονται και επειδή $|H_L| = L + 1$ καταλήγουμε σε άτοπο. 'ρα, η αρχική μας υπόθεση ότι $Q \neq \emptyset$ δεν είναι αληθής. Έτσι, μετά το τέλος του αλγόριθμου $Q = \emptyset$, οπότε αρχούν $\lfloor \frac{3}{2} L \rfloor$ χρώματα για το χρωματισμού ενός γράφου κυκλικών τόξων. $\square$

Κεφάλαιο 6

Άμεσος χρωματισμός κορυφών

6.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα μελετήσουμε το πρόβλημα του άμεσου χρωματισμού κορυφών (**online vertex coloring**) σε γράφους διαστημάτων και κυκλικών τόξων. Η διαφορά του άμεσου χρωματισμού από τον απλό χρωματισμό κορυφών που παρουσιάσαμε στα Κεφάλαια 2 και 5 έγκειται στο γεγονός ότι ένας αλγόριθμος που υλοποιεί τον άμεσο χρωματισμό (άμεσος αλγόριθμος) δε γνωρίζει εκ των προτέρων την τοπολογία (κορυφές και αντίστοιχες ακμές) του γράφου. Αντιθέτως, οι κορυφές (που πρόκειται να χρωματίσει) παρουσιάζονται σε αυτόν διαδοχικά, με αυθαίρετη σειρά και ο αλγόριθμος πρέπει να αναθέτει άμεσα το κατάλληλο χρώμα.

Ο στόχος αυτού του κεφαλαίου είναι να αναλύσουμε την απόδοση άμεσων αλγορίθμων για γράφους διαστημάτων και κυκλικών τόξων. Η απόδοση ενός άμεσου αλγόριθμου ορίζεται ως ο λόγος του αριθμού χρωμάτων που χρησιμοποιεί προς τον ελάχιστο αριθμό χρωμάτων που θα χρησιμοποιούσε ένας αλγόριθμος, ο οποίος θα είχε εξ' αρχής δεδομένη την τοπολογία του γράφου και θα αποφάσιζε για το χρώμα της κάθε κορυφής λαμβάνοντας υπόψην το σύνολο του γράφου. Ο λόγος αυτός είναι γνωστός ως λόγος ανταγωνισμού (**competitive ratio**).

Για τον άμεσο χρωματισμό κορυφών σε ένα γράφο διαστημάτων, θα αναλύσουμε τους άμεσους αλγορίθμους **First-Fit** [5] και **Kierstead-Trotter** [28], ενώ για το αντίστοιχο πρόβλημα σε γράφους κυκλικών τόξων, θα μελετήσουμε και πάλι τον Αλγόριθμο **First-Fit** [6] καθώς και έναν άμεσο αλγόριθμο [29], ο οποίος βασίζεται στον Αλγόριθμο **Kierstead-Trotter**.

Έστω $\mathcal{I}$ ένα σύνολο διαστημάτων και F ένα σύνολο κυκλικών τόξων. Ο Αλγόριθμος **First-Fit (FF)** χρωματίζει τα διαστήματα του $\mathcal{I}$ (ή τα τόξα του F) ως εξής: το πρώτο διάστημα (ή τόξο) που εμφανίζεται ανατίθεται στην κλάση χρωμάτων 1, ενώ κάθε διάστημα (ή τόξο) που επακολουθεί ανατίθεται στη μικρότερη δυνατή κλάση χρωμάτων, ούτως ώστε ο μερικός χρωματισμός να είναι ορθός. Πιο συγκεχριμένα, έστω ότι ο Αλγόριθμος **FF** έχει χρωματίσει το υποσύνολο $\mathcal{I}' \subseteq \mathcal{I}$ (ή $F' \subseteq F$), έχοντας χρησιμοποιήσει τις κλάσεις χρωμάτων $1, \ldots, k$. Όταν το επόμενο διάστημα (ή τόξο) παρουσιαστεί, ο Αλγόριθμος **FF** βρίσκει το μικρότερο δείκτη i, $1 \leq i \leq k+1$, για τον οποίο το διάστημα (ή τόξο) δεν επικαλύπτει κανένα στοιχείο του $\mathcal{I}'$ (ή F'). Για την ορθότητα της διαδικασίας, σημειώνουμε ότι ένας τέτοιος δείκτης υπάρχει πάντα, δεδομένου ότι κανένα στοιχείο του $\mathcal{I}'$ (ή F') δεν έχει ανατεθεί στην κλάση χρωμάτων $k+1$. Τέλος, ο Αλγόριθμος **FF** αναθέτει το διάστημα (ή τόξο) στην κλάση χρωμάτων i και η διαδικασία επαναλαμβάνεται έως ότου χρωματιστούν όλα τα στοιχεία του $\mathcal{I}$ (ή F).

6.2 Γράφοι διαστημάτων

Σε αυτή την ενότητα θα παρουσιάσουμε δύο άμεσους αλγόριθμους, τον **First-Fit** και τον **Kierstead-Trotter**, για το χρωματισμό κορυφών ενός γράφου διαστημάτων G.

6.2.1 Ο Αλγόριθμος First-Fit

Έστω ότι ο Αλγόριθμος **FF** χρωματίζει το γράφο διαστημάτων G χρησιμοποιώντας m κλάσεις χρωμάτων. Θα δείξουμε ότι υπάρχει ένα στοιχειώδες διάστημα, το οποίο περιέχεται σε τουλάχιστον $\frac{m}{8}$ διαστήματα του $\mathcal{I}$, δηλαδή ότι υπάρχουν τουλάχιστον $\frac{m}{8}$ διαστήματα που αλληλεπικαλύπτονται. Επομένως ο γράφος διαστημάτων G περιέχει μια κλίκα μεγέθους τουλάχιστον $\frac{m}{8}$. Έτσι, οποιοσδήποτε χρωματισμός του G θα πρέπει να χρησιμοποιεί τουλάχιστον $\frac{m}{8}$ χρώματα, δηλαδή $\chi(G) \geq \frac{m}{8}$ και επειδή ο βέλτιστος αριθμός χρωμάτων είναι $opt = \chi(G)$, προκύπτει ότι $\frac{m}{opt} \leq 8$.

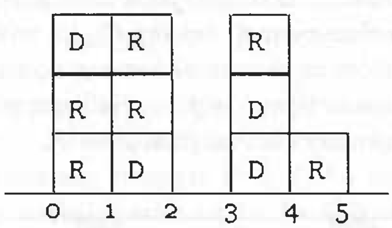
Θεωρούμε, χωρίς βλάβη της γενικότητας, ότι κάθε τόξο $I \in \mathcal{I}$ είναι της μορφής $I = [i, j]$, $0 \leq i < j \leq N$, όπου οι i, j, N είναι ακέραιοι αριθμοί. Συμβολίζουμε με $\mathcal{E}$ το σύνολο των στοιχειωδών διαστημάτων $[i-1, i]$, $1 \leq i \leq N$. Συνεπώς, κάθε διάστημα $I \in \mathcal{I}$ είναι η ένωση διαδοχικών στοιχειωδών διαστημάτων και επομένως οποιαδήποτε δύο διαστήματα του $\mathcal{I}$ αλληλεπικαλύπτονται αν περιέχουν αμφότερα τουλάχιστον ένα κοινό στοιχειώδες διάστημα. Θα κατασκευάσουμε ένα σύνολο στηλών, το οποίο αντιστοιχεί στο χρωματισμό του Αλγόριθμου **FF**. Μια στήλη αντιστοιχεί σε ένα μοναδικό στοιχειώδες διάστημα e και ονομάζεται στήλη e. Ενδέχεται να υπάρχουν στοιχειώδη διαστήματα, στα οποία δεν αντιστοιχούν στήλες. Κάθε στήλη e έχει ένα θετικό ύψος t και μπορεί να αναπαρασταθεί ως ένα ορθογώνιο $\{(x, y) | l(e) \leq x \leq r(e), 1 \leq y \leq t\}$, όπου $l(e), r(e)$ είναι αντίστοιχα το αριστερό και το δεξιό άκρο του στοιχειώδους διαστήματος e. Μια στήλη e ύψους t λέμε ότι είναι ενεργή στα ύψη $1, \ldots, t$ και ανενεργή στα ύψη $t+1, \ldots, \infty$. Κάθε τετράγωνο $\{(x, y) | l(e) \leq x \leq r(e), i-1 \leq y \leq i\}$ της στήλης e, όπου

$1 \leq i \leq t$, περιέχει ένα σύμβολο εκ των R, D, F. Άρα, κάθε στήλη e ύψους t έχει ένα από τα παραπάνω σύμβολα σε κάθε ενεργό ύψος της.

Μια στήλη e ύψους t έχει το σύμβολο R σε κάποιο ύψος $1 \leq i \leq t$ ανν κάποιο διάστημα $I \in \mathcal{I}$, το οποίο έχει ανατεθεί στην $i-$οστή κλάση χρωμάτων, περιέχει το στοιχειώδες διάστημα e. Ένα στοιχειώδες διάστημα e ενδέχεται να περιέχεται σε ένα $I \in \mathcal{I}$ που έχει ανατεθεί στην $i-$οστή κλάση χρωμάτων, αλλά είτε να μην υπάρχει στήλη που να αντιστοιχεί στο e είτε η στήλη e να είναι ανενεργή στο ύψος i. Ο στόχος της διαδικασίας κατασκευής στηλών είναι να δείξουμε ότι υπάρχει μια στήλη με τουλάχιστον $\frac{m}{8}$ R σύμβολα. Έτσι, ο G θα έχει μια κλίκα μεγέθους τουλάχιστον $\frac{m}{8}$.

Περιγραφικά, η διαδικασία κατασκευής των στηλών αρχικοποιείται στο πρώτο βήμα με ένα σύνολο στηλών, οι οποίες είναι ενεργές μόνο στο ύψος 1. Για τα επόμενα βήματα $i \geq 2$, η διαδικασία επιλέγει ένα υποσύνολο των ενεργών στηλών στο ύψος $i-1$ ως το σύνολο των ενεργών στηλών στο ύψος i. Αν αυτό είναι κενό, τότε η διαδικασία τερματίζεται.

Έστω C_i το σύνολο των ενεργών στηλών στο ύψος i. Για κάθε σύνολο στηλών υπάρχει μια φυσική διάταξη, η οποία ταυτίζεται με τη φυσική διάταξη των αντίστοιχων στοιχειωδών διαστημάτων. Έτσι, $\forall e \in C_i$, ο αριστερός (δεξιός) γείτονας της e στο C_i είναι η στήλη που προηγείται (επακολουθεί) της e στη φυσική διάταξη. Αν δεν υπάρχει τέτοια στήλη, τότε ο αριστερός (δεξιός) γείτονας της e στο C_i δεν ορίζεται.



Σχήμα 6.1: Ένα πιθανό αποτέλεσμα της διαδικασίας κατασκευής στηλών. Οι στήλες αντιστοιχούν στα στοιχειώδη διαστήματα [0,1], [1,2], [3,4] και [4,5]. Η στήλη [1,2] έχει ύψος 3 και περιέχει τα σύμβολα D, R και R στα ύψη 1, 2 και 3 αντίστοιχα. Σημειώνουμε ότι $C_1 = \{[0,1],[1,2],[3,4],[4,5]\}$, $C_2 = \{[0,1],[1,2],[3,4]\}$ και $C_3 = \{[0,1],[1,2],[3,4]\}$.

Τυπικά, ο τρόπος δημιουργίας των συνόλων C_i είναι ο εξής. Στο πρώτο βήμα ($i=1$), το σύνολο C_1 περιέχει μόνο τις στήλες που αντιστοιχούν σε στοιχειώδη διαστήματα, τα οποία περιέχονται σε κάποιο $I \in \mathcal{I}$ που έχει ανατεθεί στην κλάση χρωμάτων 1. Αυτές οι στήλες λαμβάνουν το σύμβολο R στο ύψος 1. Για $i \geq 2$, οι παρακάτω τρεις κανόνες καθορίζουν ποιες στήλες του C_{i-1} επιλέγονται ως στήλες του C_i.

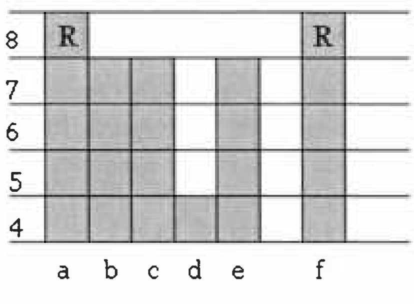
1. Για κάθε $e \in C_{i-1}$, αν το στοιχειώδες διάστημα e περιέχεται σε κάποιο

$I \in \mathcal{I}$ που έχει ανατεθεί στην κλάση χρωμάτων i, τότε η στήλη e περιλαμβάνεται στο C_i με το σύμβολο R.

2. Για κάθε $e \in C_{i-1}$ που δεν επιλέχτηκε από τον κανόνα 1, αν η e είναι στο C_{i-1} ο αριστερός (ή ο δεξιός) γείτονας κάποιας στήλης e' που προστέθηκε στο C_i από τον κανόνα 1, τότε η e περιλαμβάνεται στο C_i με το σύμβολο D. Σημειώνουμε ότι η e παραμένει ο αριστερός (ή δεξιός) γείτονας της e' στο C_i.

3. Για κάθε $e \in C_{i-1}$ που δεν επιλέχτηκε από τους κανόνες 1 και 2, έστω e' ο αριστερός γείτονας της e στο C_{i-1}. Έστω ότι η e' είναι ο αριστερός γείτονας της e στα σύνολα $C_j, \ldots, C_{i-1}$ και δεν είναι στα $C_1, \ldots, C_{j-1}$. Αν ο αριθμός των R συμβόλων της e στα ύψη $j, \ldots, i-1$ είναι μεγαλύτερος από $(i-j)/4$, τότε η e περιλαμβάνεται στο C_i με το σύμβολο F. Αν ο αριστερός γείτονας της e στο C_{i-1} δεν υπάρχει, τότε ελέγχουμε το δεξιό της γείτονα, έστω e''. Έστω ότι η e'' είναι ο δεξιός γείτονας της e στα σύνολα $C_k, \ldots, C_{i-1}$ και δεν είναι στα $C_1, \ldots, C_{k-1}$. Αν ο αριθμός των R συμβόλων της e στα ύψη $k, \ldots, i-1$ είναι μεγαλύτερος από $(i-k)/4$, τότε η e περιλαμβάνεται στο C_i με το σύμβολο F. Αν ούτε ο δεξιός γείτονας της e στο C_{i-1} δεν υπάρχει, τότε η e δεν περιλαμβάνεται στο C_i.

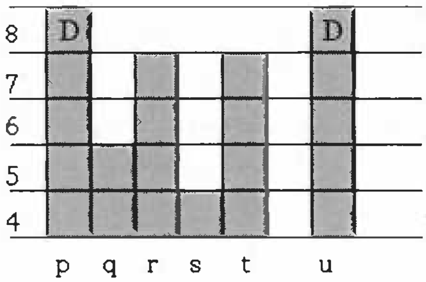
Η κατασκευαστική διαδικασία ολοκληρώνεται στο βήμα (ύψος) m', μετά το οποίο καμμία στήλη δεν είναι ενεργή, δηλαδή $C_{m'+1} = \emptyset$. Σημειώνουμε ότι αν $m' > m$ (υπενθυμίζουμε ότι m είναι οι κλάσεις χρωμάτων που χρησιμοποίησε ο Αλγόριθμος FF), τότε στα ύψη $m+1, \ldots, m'$ εφαρμόζεται μόνο ο κανόνας 3, οπότε τα σύμβολα που προστίθενται είναι μόνο F.



Σχήμα 6.2: Ένα στιγμιότυπο της δημιουργίας του C_8 μετά την εφαρμογή του κανόνα 1. Οι στήλες b, e θα συμπεριληφθούν στο C_8 με D σύμβολα λόγω του κανόνα 2.

Λήμμα 6.1
Έστω $1 \leq i < j \leq m$. *Για κάθε διάστημα* $I \in \mathcal{I}$ *που έχει ανατεθεί στην*

κλάση χρωμάτων j, υπάρχει τουλάχιστον μια στήλη $e \in C_i$, τέτοια ώστε το I να περιέχει το στοιχειώδες διάστημα e.



Σχήμα 6.3: Ένα στιγμιότυπο μετά την εφαρμογή του κανόνα 2. Η στήλη p είναι ο αριστερός γείτονας της r στα C_6, C_7 και η t είναι ο δεξιός γείτονας της r στα C_5, C_6, C_7. Η r θα προστεθεί στο C_8 (με το σύμβολο F) αν το πλήθος των R συμβόλων της στα ύψη 6,7 είναι μεγαλύτερο από $\frac{2}{4}$ ή αν το πλήθος των R συμβόλων της στα ύψη 5,6,7 είναι μεγαλύτερο από $\frac{3}{4}$.

Απόδειξη. Η απόδειξη θα γίνει με επαγωγή στο δείκτη i. Η βασική ιδιότητα του Αλγόριθμου **First-Fit** είναι ότι αν ένα διάστημα $I \in \mathcal{I}$ έχει ανατεθεί στην κλάση χρωμάτων k, τότε $\forall 1 \leq i \leq k-1$ υπάρχει ένα διάστημα I', το οποίο έχει ανατεθεί στην κλάση i και επικαλύπτεται με το I. Λόγω αυτού, το λήμμα ισχύει για $i = 1$. Για το επαγωγικό βήμα, υποθέτουμε ότι το λήμμα ισχύει για $i-1$, όπου $i \geq 2$. Έστω I ένα διάστημα του $\mathcal{I}$ που έχει ανατεθεί στην κλάση χρωμάτων $j > i$. Επομένως, υπάρχει $I' \in \mathcal{I}$, το οποίο έχει ανατεθεί στην κλάση χρωμάτων i και επικαλύπτεται με το I. Από την επαγωγική υπόθεση, υπάρχει μη κενό υποσύνολο C' (διαδοχικών στηλών) του C_{i-1}, τέτοιο ώστε το I' να περιέχει τα αντίστοιχα διαδοχικά στοιχειώδη διαστήματα. Αντίστοιχα, υπάρχει όμοιο σύνολο C για το I. Από τον κανόνα 1, όλες οι στήλες του C' προστίθενται στο C_i (εφόσον το I' έχει χρωματιστεί με χρώμα i). Έτσι, αν $C \cap C' \neq \emptyset$, τότε το I περιέχει τουλάχιστον ένα στοιχειώδες διάστημα, του οποίου η αντίστοιχη στήλη ανήκει στο C_i, οπότε το λήμμα αποδείχτηκε. Αν $C \cap C' = \emptyset$, δεν μπορεί να υπάρχει στήλη $e \in C_{i-1}$ μεταξύ των στηλών του C και C', διότι τότε τα I, I' δε θα αλληλεπικαλύπτονται (εφόσον το στοιχειώδες διάστημα e θα βρίσκεται ανάμεσά τους). Ως εκ τούτου, υπάρχει στήλη $e' \in C$, η οποία είναι είτε ο αριστερός είτε ο δεξιός γείτονας κάποιας στήλης του C' στο C_{i-1}. Η στήλη e' προστίθεται στο C_i από τον κανόνα 2 (αν δεν έχει ήδη προστεθεί από τον κανόνα 1), οπότε η απόδειξη ολοκληρώνεται. □

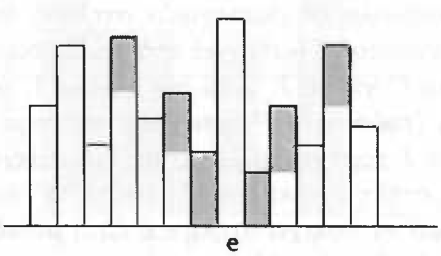
Το Λήμμα 6.1 υποδεικνύει ότι τα σύνολα $C_1, \ldots, C_m$ είναι μη κενά και έτσι $m' \geq m$. Για μια στήλη e ύψους j και $1 \leq i \leq j$, ορίζουμε ως $\rho_e(i,j)$, $\delta_e(i,j)$ και $\phi_e(i,j)$ τον αριθμό των συμβόλων R, D και F αντίστοιχα ανάμεσα στα ύψη i και j (συμπεριλαμβανομένων αυτών). Ορίζουμε επίσης $\rho_e(j) = \rho_e(1, j)$, $\delta_e(j) = \delta_e(1, j)$, $\phi_e(j) = \phi_e(1, j)$ και $\rho_e(0) = \delta_e(0) = \phi_e(0) = 0$.

Λήμμα 6.2
Για $\forall e \in C_i$, όπου $1 \leq i \leq m'$, ισχύει ότι $\rho_e(i) \geq \frac{1}{4}(\rho_e(i) + \phi_e(i))$.

Απόδειξη. Η απόδειξη θα γίνει με επαγωγή στο δείκτη i. Για $i = 0, 1$, το λήμμα ισχύει, διότι $\rho_e(0) \geq \frac{1}{4}(\rho_e(0)+\phi_e(0)) \Rightarrow 0 \geq 0$ και $\rho_e(1) \geq \frac{1}{4}(\rho_e(1)+\phi_e(1)) \Rightarrow \rho_e(1) \geq \frac{1}{4}\rho_e(1)$, που είναι αληθή. Υποθέτουμε ότι το λήμμα ισχύει για όλα τα ύψη έως το $i-1$ και θα αποδείξουμε ότι ισχύει και για το ύψος i. Αν η στήλη e δεν έχει το σύμβολο F στο ύψος i, τότε η επαγωγή ολοκληρώνεται, διότι $\phi_e(i) = 0$ και $\rho_e(i) \geq \frac{1}{4}\rho_e(i)$. Υποθέτουμε, λοιπόν, ότι η e έχει το σύμβολο F στο ύψος i, το οποίο έχει προστεθεί στο C_i από τον κανόνα 3. Άρα, υπάρχει ένα $1 \leq j \leq i-1$, τέτοιο ώστε $\rho_e(j, i-1) > (i-j)/4$ και έτσι $\rho_e(j, i) > (i-j+1)/4$. Από το βήμα της υπόθεσης έπεται ότι:

$$\begin{aligned}
\rho_e(i) &= \rho_e(j-1) + \rho_e(j, i) \\
&\geq \frac{1}{4}(\rho_e(j-1) + \phi_e(j-1)) + \frac{1}{4}(i-j+1) \\
&\geq \frac{1}{4}(\rho_e(j-1) + \phi_e(j-1)) + \frac{1}{4}(\rho_e(j, i) + \phi_e(j, i)) \\
&= \frac{1}{4}(\rho_e(i) + \phi_e(i))
\end{aligned}$$

όπου $\rho_e(j, i) + \phi_e(j, i) \leq i - j + 1$, διότι το πλήθος των R και F συμβόλων μεταξύ των υψών j και i είναι το πολύ ίσο με τον αριθμό των τετραγώνων ανάμεσα σε αυτά τα δύο ύψη, δηλαδή $i - j + 1$. □



Σχήμα 6.4: Οι στήλες στο τέλος της διαδικασίας κατασκευής. Ο αριθμός των D συμβόλων της στήλης e φράσσεται από τον αριθμό των R συμβόλων σε όλα τα σκιασμένα ορθογώνια.

Θεώρημα 6.3
Έστω m το πλήθος των χρωμάτων που χρησιμοποίησε ο Αλγόριθμος FF για

το χρωματισμό του συνόλου διαστημάτων $\mathcal{I}$. Υπάρχει μια κλίκα μεγέθους τουλάχιστον $\frac{m}{8}$ στον αντίστοιχο γράφο διαστημάτων.

Απόδειξη. Θα δείξουμε ότι υπάρχει μια στήλη $e \in C_{m'}$, τέτοια ώστε $\rho_e(m') \geq \frac{m}{8}$. Έστω e μια στήλη του $C_{m'}$, η οποία έχει αριστερό και δεξιό γείτονα στο $C_{m'}$. Η περίπτωση όπου κάποιος εκ των δύο γειτόνων δεν ορίζεται είναι ευκολότερη. Έστω $f_1, \ldots, f_a$ οι αριστεροί γείτονες της e, όπου f_i είναι ο αριστερός γείτονας της e στα $C_{t_{i-1}+1}, \ldots, C_{t_i}$, $i = 1, \ldots, a$, όπου $t_0 = 0 < t_1 < \ldots < t_a = m'$. Ομοίως, έστω $g_1, \ldots, g_b$ οι δεξιοί γείτονες της e, όπου g_i είναι ο δεξιός γείτονας της e στα $C_{n_{i-1}+1}, \ldots, C_{n_i}$, $i = 1, \ldots, b$, όπου $n_0 = 0 < n_1 < \ldots < n_b = m'$. Ισχύει ότι

$$\delta_e(m') \leq \sum_{i=1}^{a} \rho_{f_i}(t_{i-1}+1, t_i) + \sum_{i=1}^{b} \rho_{g_i}(n_{i-1}+1, n_i)$$

Η παραπάνω ανισωτική σχέση ισχύει, διότι για να προστεθεί το σύμβολο D στο ύψος i της στήλης e θα πρέπει ένας εκ των γειτόνων της να έχει το σύμβολο R στο ίδιο ύψος (κανόνας 2). Εφόσον κάθε f_i και g_i γίνεται ανενεργό πέραν του ύψους t_i και n_i αντίστοιχα, από τον κανόνα 3 ισχύει ότι

$$\rho_{f_i}(t_{i-1}+1, t_i) \leq \frac{1}{4}(t_i - t_{i-1})$$

και

$$\rho_{g_i}(n_{i-1}+1, n_i) \leq \frac{1}{4}(n_i - n_{i-1})$$

Συνεπώς,

$$\delta_e(m') \leq \sum_{i=1}^{a} \frac{1}{4}(t_i - t_{i-1}) + \sum_{i=1}^{b} \frac{1}{4}(n_i - n_{i-1}) \leq \frac{1}{4}(t_a + n_b) = \frac{m'}{2}$$

Εφόσον

$$\rho_e(m') + \delta_e(m') + \phi_e(m') = m'$$

ισχύει

$$\rho_e(m') + \phi_e(m') = m' - \delta_e(m') \geq m' - \frac{m'}{2} = \frac{m'}{2}$$

Εφαρμόζοντας το Λήμμα 6.2 έχουμε

$$\begin{aligned}
\rho_e(m') &\geq \frac{1}{4}(\rho_e(m') + \phi_e(m')) \\
\rho_e(m') &\geq \frac{1}{4}\frac{m'}{2} \\
\rho_e(m') &\geq \frac{m'}{8}
\end{aligned}$$

Εφόσον δεν υπάρχουν R σύμβολα πέραν του ύψους m, $\rho_e(m) = \rho_e(m')$. Άρα,

$$\rho_e(m) = \rho_e(m') \geq \frac{m'}{8} \geq \frac{m}{8}$$

Ομοίως, μπορούμε να δείξουμε ότι $\rho_e(m) \geq \frac{m}{8}$ στις περιπτώσεις όπου η στήλη e δεν έχει αριστερό ή δεξιό γείτονα. □

Θεώρημα 6.4
Ο Αλγόριθμος FF χρησιμοποιεί το πολύ $8\chi(G)$ κλάσεις χρωμάτων για τον άμεσο χρωματισμό των κορυφών ενός γράφου διαστημάτων G.

Απόδειξη. Άμεση απόρροια του Θεωρήματος 6.3, εφόσον ο G περιέχει μια κλίκα μεγέθους τουλάχιστον $\frac{m}{8}$ και ο Αλγόριθμος FF χρησιμοποίησε m κλάσεις χρωμάτων για τον άμεσο χρωματισμό του. □

Τέλος, σημειώνουμε ότι στο [30] έχει αποδειχθεί ότι ο Αλγόριθμος FF χρησιμοποιεί τουλάχιστον $4.4\chi(G)$ κλάσεις χρωμάτων για τον άμεσο χρωματισμό των κορυφών ενός γράφου διαστημάτων.

6.2.2 Ο Αλγόριθμος Kierstead-Trotter

Ο Αλγόριθμος Kierstead-Trotter (KT) επιλύει το πρόβλημα του άμεσου χρωματισμού κορυφών ενός γράφου διαστημάτων G. Κατασκευάζει σύνολα κορυφών $S_1, S_2, \ldots$ και όταν μια κορυφή u παρουσιάζεται, την τοποθετεί στο S_i, όπου i είναι ο μικρότερος δείκτης, τέτοιος ώστε το $S_1 \cup S_2 \cup \ldots \cup (S_i \cup \{u\})$ να μην περιέχει κλίκα μεγέθους $i+1$.

Αλγόριθμος KT

1. $k = 1$;
2. $S_k = \emptyset$;
3. for $j = 1, \ldots, n$ do
4. Βάλε την u_j στο S_i, όπου $1 \leq i \leq k$ είναι ο μικρότερος δείκτης, τέτοιος ώστε το $S_1 \cup S_2 \cup \ldots \cup (S_i \cup \{u\})$ να μην περιέχει κλίκα μεγέθους $i+1$;
5. if δεν υπάρχει τέτοιο i then θέσε $k = k+1$ και βάλε την u_j στο S_k;
6. Χρώματισε την u_j ως εξής: αν η u_j ανήκει στο S_1, τότε χρωμάτισέ την με το χρώμα 1. Διαφορετικά χρωμάτισέ την με ένα από τα τρία χρώματα που επαρκούν για το χρωματισμό των κορυφών του συνόλου S_i, $i \neq 1$;

Στη συνέχεια θα αποδείξουμε ότι ο αριθμός των χρωμάτων που χρησιμοποιεί ο Αλγόριθμος KT είναι το πολύ $3\chi(G) - 2$ (Θεώρημα 6.6). Για αυτό το σκοπό θα χρησιμοποιήσουμε (χωρίς να το αποδείξουμε) το παρακάτω λήμμα, το οποίο απέδειξαν οι Kierstead και Trotter στο [28].

Λήμμα 6.5
Έστω $S_1, S_2, \ldots$ τα σύνολα που κατασκευάζει ο Αλγόριθμος ΚΤ. Τότε, κάθε υπογράφος $G[S_i]$ προκύπτει από την ένωση μη τεμνόμενων μονοπατιών, δηλαδή από την ένωση μονοπατιών που δε διέρχονται από κοινές κορυφές.

Θεώρημα 6.6
Ο Αλγόριθμος ΚΤ χρησιμοποιεί το πολύ $3\chi(G) - 2$ κλάσεις χρωμάτων για τον άμεσο χρωματισμό των κορυφών του G.

Απόδειξη. Έστω $S_1, S_2, \ldots, S_t$ τα σύνολα που κατασκευάζει ο Αλγόριθμος ΚΤ. Μια κορυφή u τοποθετείται στο S_t, διότι το $S_1 \cup S_2 \cup \ldots \cup (S_{t-1} \cup \{u\})$ περιέχει μια κλίκα μεγέθους t. Άρα, $\chi(G) \geq t$. Εξ' ορισμού, το σύνολο S_1 είναι ένα ανεξάρτητο σύνολο (αφού δεν περιέχει κλίκα μεγέθους 2) και επομένως οι κορυφές του μπορούν να χρωματιστούν με ένα χρώμα. Εφόσον κάθε υπογράφος $G[S_i]$ είναι η ενώση μη τεμνόμενων μονοπατιών (Λήμμα 6.5), έπεται ότι οι κορυφές των συνόλων S_i, $2 \leq i \leq t$ χρωματίζονται άμεσα με το πολύ 3 χρώματα. Έτσι, οι κλάσεις χρωμάτων που χρησιμοποιεί ο Αλγόριθμος ΚΤ είναι το πολύ $3t - 2 \leq 3\chi(G) - 2$. □

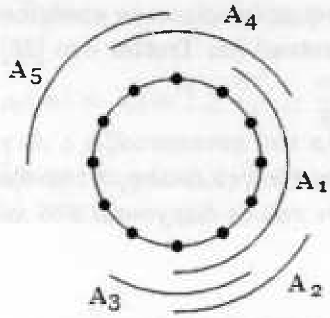
6.3 Γράφοι κυκλικών τόξων

Σε αυτή την παράγραφο θα παρουσιάσουμε δύο άμεσους αλγόριθμους για το χρωματισμό κορυφών ενός γράφου κυκλικών τόξων G. Σημειώνουμε ότι σε αυτήν την ενότητα θεωρούμε δεδομένη την αναπαράσταση του γράφου με τόξα. Υπενθυμίζουμε ότι $L(F) = L$ είναι ο φόρτος του F. Θεωρούμε, χωρίς βλάβη της γενικότητας, ότι κάθε τόξο $A_i \in F$ αντιστοιχεί στην ένωση διαδοχικών στοιχειωδών τόξων του κύκλου και ορίζουμε ως μήκος H_i του A_i το πλήθος των στοιχειωδών τόξων του κύκλου που περιέχει το A_i (Σχήμα 6.5). Έστω $H = \max_{i=1,\ldots,n} H_i$. Λέμε ότι δύο τόξα του F αλληλεπικαλύπτονται αν περιέχουν αμφότερα τουλάχιστον ένα κοινό στοιχειώδες τόξο του κύκλου.

6.3.1 Ο Αλγόριθμος First-Fit

Σύμφωνα με το [6], αν χρησιμοποιήσουμε τον Αλγόριθμο **FF** για να χρωματίσουμε το G, τότε αρχούν $(4L - 3)\log_3 H + L$ χρώματα.

Ορισμός 6.7
Ορίζουμε ως $C(H_i)$ το μεγαλύτερο δείχτη της κλάσης χρωμάτων, στην οποία έχει ανατεθεί κάποιο τόξο του F μήκους το πολύ H_i.



Σχήμα 6.5: Έστω $F = \{A_1, A_2, A_3, A_4, A_5\}$. Έχουμε ότι $H_1 = 5$, $H_2 = H_3 = 2$ και $H_4 = H_5 = 3$

Λήμμα 6.8
Έστω $H_t \in \{H_1, \ldots, H_n\}$. Κάθε τόξο μήκους το πολύ $3H_t$ θα ανατεθεί το πολύ στην κλάση χρωμάτων με δείκτη $C(H_t) + 4L - 3$, δηλαδή $C(3H_t) \leq C(H_t) + 4L - 3$.

Απόδειξη. Από τον Ορισμό 6.7, κάθε τόξο μήκους το πολύ H_t έχει ανατεθεί το πολύ στην κλάση χρωμάτων με δείκτη $C(H_t)$. Έστω τόξο A μήκους k, όπου $H_t < k \leq 3H_t$, δηλαδή $H_t \geq \frac{k}{3}$. Θα δείξουμε ότι το A μπορεί να ανατεθεί σε μια κλάση χρωμάτων, της οποίας ο δείκτης ανήκει στο σύνολο $\widetilde{C} = \{C(H_t) + 1, \ldots, C(H_t) + 4L - 3\}$. Έστω $\widetilde{F} = \{A_1, \ldots, A_m\}$ το σύνολο των τόξων, τα οποία αλληλεπικαλύπτονται με το A και έχουν ανατεθεί σε κάποια κλάση χρωμάτων, της οποίας ο δείκτης ανήκει στο $\widetilde{C}$. Έστω $h_1, h_2, \ldots, h_k$ τα στοιχειώδη τόξα του A. Ονομάζουμε τα $h_1, h_{\lceil \frac{k}{3} \rceil}, h_{\lceil \frac{2k}{3} \rceil}$ και h_k κρίσιμα τόξα του A.

Τα τόξα του $\widetilde{F}$ έχουν ανατεθεί σε κάποια κλάση χρωμάτων με δείκτη μεγαλύτερο του $C(H_t)$, οπότε $H_i \geq H_t \geq \frac{k}{3}$, $\forall A_i \in \widetilde{F}$. Άρα, κάθε $A_i \in \widetilde{F}$ περιέχει τουλάχιστον ένα κρίσιμο τόξο. Έστω $n_1, n_{\lceil \frac{k}{3} \rceil}, n_{\lceil \frac{2k}{3} \rceil}$ και n_k ο αριθμός των τόξων του $\widetilde{F}$, τα οποία περιέχουν αντίστοιχα τα κρίσιμα τόξα $h_1, h_{\lceil \frac{k}{3} \rceil}, h_{\lceil \frac{2k}{3} \rceil}$ και h_k. Εφόσον ο φόρτος του F είναι L και το A περιέχει τα κρίσιμα τόξα, ισχύει:

$$|\widetilde{F}| \leq n_1 + n_{\lceil \frac{k}{3} \rceil} + n_{\lceil \frac{2k}{3} \rceil} + n_k \leq 4(L - 1)$$

Όμως, το πλήθος των κλάσεων χρωμάτων με δείκτη στο $\widetilde{C}$ είναι $4L - 3 > 4L - 4$, οπότε υπάρχει τουλάχιστον μια διαθέσιμη κλάση χρωμάτων με δείκτη στο $\widetilde{C}$ για το χρωματισμό του A. Έτσι, το A θα ανατεθεί το πολύ στην κλάση χρωμάτων με δείκτη $C(H_t) + 4L - 3$. □

Θεώρημα 6.9
Ο Αλγόριθμος FF χρησιμοποιεί το πολύ $(4L-3)\log_3 H + L$ κλάσεις χρωμάτων για το χρωματισμό του G, δηλαδή $C(H) \leq (4L-3)\log_3 H + L$.

Απόδειξη. Το ζητούμενο αποδεικνύεται εύκολα με χρήση του Λήμματος 6.8.

$$C(H) \leq (4L-3) + C(\frac{H}{3}) \leq 2(4L-3) + C(\frac{H}{3^2}) \leq \ldots \leq$$

$$\leq (4L-3)\log_3 H + C(1) = (4L-3)\log_3 H + L$$

□

Σημειώνουμε, επίσης, ότι στο [6] έχει αποδειχθεί ότι ο Αλγόριθμος FF χρησιμοποιεί τουλάχιστον $0.5L log_2 H + L$ κλάσεις χρωμάτων για τον άμεσο χρωματισμό των κορυφών ενός γράφου κυκλικών τόξων.

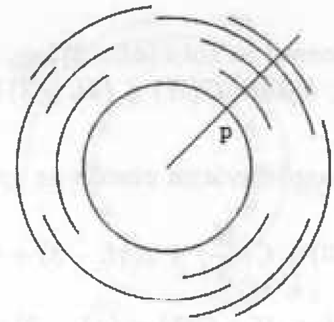
6.3.2 Ο Αλγόριθμος TRI

Ο Αλγόριθμος TRI ([29]) επιλύει το πρόβλημα του άμεσου χρωματισμού κορυφών του G, ακολουθώντας την εξής διαδικασία: ορίζει ένα σημείο p πάνω στον κύκλο και θεωρεί την ακτίνα r_p του κύκλου που διέρχεται από το p. Όταν ένα νέο τόξο παρουσιάζεται, ο Αλγόριθμος TRI ελέγχει αν η ακτίνα r_p το τέμνει ή όχι. Αν το τέμνει, τότε το αναθέτει σε μια νέα κλάση χρωμάτων (σημειώνουμε ότι όλα τα τόξα που τέμνονται από την r_p σχηματίζουν κλίκα, οπότε ορθώς λαμβάνουν διαφορετικό χρώμα). Προφανώς, το σύνολο των τόξων που δεν τέμνονται από την r_p σχηματίζει ένα γράφο διαστημάτων. Έτσι, αν η r_p δεν τέμνει το εκάστοτε τόξο που παρουσιάζεται, τότε χρησιμοποιούμε τον Αλγόριθμο KT της Παραγράφου 6.2.2 για το χρωματισμό του.

Αλγόριθμος TRI

Θεώρησε ένα σημείο p του κύκλου και την ακτίνα r_p;
Θέσε $col = 0$;
for $i = 1, \ldots, n$
    if η r_p τέμνει το A_i then
        $col = col + 1$;
        Ανάθεσε το χρώμα col στο A_i;
    else χρωμάτισε το A_i σύμφωνα με τον Αλγόριθμο KT;

Θεώρημα 6.10
Ο Αλγόριθμος TRI χρησιμοποιεί το πολύ $4\ \chi(G) - 2$ κλάσεις χρωμάτων για τον άμεσο χρωματισμό κορυφών του γράφου κυκλικών τόξων G.



Σχήμα 6.6: Κάθε τόξο, το οποίο τέμνεται από την ακτίνα του κύκλου που διέρχεται από το σημείο p, ανατίθεται σε μια νέα κλάση χρωμάτων, ενώ τα υπολοίπα χρωματίζονται σύμφωνα με τον Αλγόριθμο KT.

Απόδειξη. Έστω K_{max} η μέγιστη κλίκα του G. Ισχύει $\chi(G) \geq K_{max}$. Ο Αλγόριθμος TRI χρησιμοποιεί το πολύ K_{max} χρώματα για να χρωματίσει τα τόξα που τέμνονται από την ακτίνα r_p. Έστω G' ο γράφος διαστημάτων που προκύπτει από το σύνολο των υπόλοιπων τόξων. Σύμφωνα με τον Αλγόριθμος KT, ο Αλγόριθμος TRI θα χρησιμοποιήσει το πολύ $3\,\chi(G') - 2$ χρώματα για το χρωματισμό του G'. Συνολικά, το πλήθος των χρωμάτων που θα χρησιμοποιηθούν είναι το πολύ

$$K_{max} + 3\,\chi(G') - 2 \leq 4\,\chi(G) - 2$$

□

Κεφάλαιο 7

Μέγιστος χρωματισμός κορυφών σε γράφους διαστημάτων

7.1 Εισαγωγή

Έστω γράφος $G = (V, E)$ και έστω $w : V \to \mathbb{N}$ μια συνάρτηση βάρους, η οποία αναθέτει ένα θετικό ακέραιο βάρος σε κάθε κορυφή $u \in V$. Το πρόβλημα του μέγιστου χρωματισμού κορυφών (**max vertex coloring**) ([5]) αναζητά έναν ορθό χρωματισμό του G, έστω $C_1, C_2, \ldots, C_k$, έτσι ώστε να ελαχιστοποιείται το άθροισμα $\sum_{i=1}^{k} w_i$, όπου $w_i = max_{u \in C_i} w(u)$. Η ελάχιστη τιμή αυτού του αθροίσματος ονομάζεται κόστος ή βάρος του (μέγιστου) χρωματισμού. Σε αυτό το κεφάλαιο, θα αποδείξουμε αρχικά ότι το πρόβλημα του μέγιστου χρωματισμού κορυφών σε γράφους διαστημάτων είναι $\mathcal{NP}$ πλήρες και στη συνέχεια θα παρουσιάσουμε τρεις προσεγγιστικούς αλγόριθμους για την επίλυσή του.

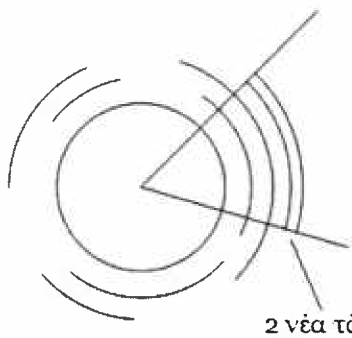
7.2 $\mathcal{NP}$-πληρότητα

Θα αποδείξουμε ότι ο μέγιστος χρωματισμός κορυφών σε γράφους διαστημάτων είναι $\mathcal{NP}$ πλήρες πρόβλημα ([5]) κάνοντας αναγωγή από το πρόβλημα του χρωματισμού κορυφών σε γράφους κυκλικών τόξων (το οποίο αποδείξαμε ότι είναι $\mathcal{NP}$ πλήρες στο Θεώρημα 5.5).

Θεωρούμε ένα γράφο κυκλικών τόξων $G = (V, E)$ και το αντίστοιχο σύνολο κυκλικών τόξων $F = \{A_1, A_2, \ldots, A_n\}$, όπου τα τόξα A_i είναι ανοιχτά. Έστω $k \in \{1, \ldots, n\}$ και έστω p σημείο του κύκλου, το οποίο περιέχεται σε ακριβώς k κυκλικά τόξα (δηλαδή η ακτίνα του κύκλου που διέρχεται από το p τέμνει ακριβώς k τόξα). Αν δεν υπάρχει τέτοιο p, τότε τροποποιούμε το γράφο G ως εξής: θεωρούμε μια επίκεντρη γωνία, τέτοια ώστε να μην περιέχει άκρα τόξων. Έστω $l < k$ το πλήθος των τόξων που διέρχονται από αυτήν. Προσθέτουμε $k - l$ τόξα στην περιφέρεια του κύκλου, τέτοια ώστε τα άκρα τους να είναι σημεία των ημιευθειών που ορίζουν την επίκεντρη γωνία (Σχήμα 7.1). Έτσι, επιλέγουμε ως p ένα οποιοδήποτε σημείο του κύκλου που περιέχεται στην επίκεντρη γωνία.

Σημειώνουμε ότι ο τροποποιημένος γράφος $\widetilde{G}$ (ο οποίος προκύπτει μετά την προσθήκη των $k-l$ τόξων) χρωματίζεται με k χρώματα ανν ο G χρωματίζεται με k χρώματα. Πράγματι, ένας ορθός χρωματισμός k χρωμάτων του G, επεκτείνεται εύκολα στο $\widetilde{G}$, αν χρωματίσουμε τα προστιθέμενα τόξα χρησιμοποιώντας τα $k-l$ διαθέσιμα χρώματα. Αν ο $\widetilde{G}$ χρωματίζεται με k χρώματα, τότε διαγράφοντας τα $k-l$ προστιθέμενα τόξα και διατηρώντας ως έχει το χρωματισμό στα υπόλοιπα, λαμβάνουμε ένα χρωματισμό k χρωμάτων για το G.

Σε κάθε περίπτωση, συνεπώς, μπορούμε να επιλέξουμε ένα σημείο p, το οποίο να περιέχεται σε k τόξα.



Σχήμα 7.1: Η κυκλική αναπαράσταση ενός γράφου κυκλικών τόξων. Αν υποθέσουμε ότι $k=4$, τότε προσθέτουμε $k-l=2$ τόξα.

Θεώρημα 7.1
Το πρόβλημα του μέγιστου χρωματισμού κορυφών σε γράφους διαστημάτων είναι $\mathcal{NP}$ πλήρες.

Απόδειξη. Επιλέγουμε σημείο p του κύκλου και υποθέτουμε ότι τα k τόξα που περιέχουν το p είναι τα $A_1, A_2, \ldots, A_k$. Έστω $A_i = (s_i, f_i)$ το τόξο του F που αντιστοιχεί στην κορυφή $u_i \in V$, όπου s_i, f_i είναι αντίστοιχα η αρχή και το πέρας του A_i κατά τη δεξιόστροφη φορά.

Χωρίζουμε κάθε A_i, $i = 1, \ldots, k$, σε δύο τόξα l_i και r_i, όπου $l_i = (s_i, p)$ και $r_i = (p, f_i)$. Ορίζουμε τα σύνολα $S_l = (l_i)_{i\in\{1,\ldots,k\}}$ και $S_r = (r_i)_{i\in\{1,\ldots,k\}}$, καθώς και το σύνολο διαστημάτων $\mathcal{I} = S_l \cup S_r \cup (A_i)_{i\in\{k+1,\ldots,n\}}$. Θεωρούμε το γράφο διαστημάτων $G' = (\mathcal{I}, E')$ του $\mathcal{I}$. Αναθέτουμε τα εξής βάρη στις κορυφές του $\mathcal{I}$: $w(l_i) = w(r_i) = i$ και $w(u_i) = 1$. Θα δείξουμε ότι ο G χρωματίζεται με k χρώματα ανν ο G' έχει μέγιστο χρωματισμό κόστους $\frac{k(k+1)}{2}$.

Έστω $\mathcal{C} = (C_1, C_2, \ldots, C_k)$ ένας χρωματισμός του G. Χωρίς βλάβη της γενικότητας, υποθέτουμε ότι η κορυφή u_i ανήκει στο C_i, $i = 1, \ldots, k$. Ορίζουμε το σύνολο $\mathcal{C}' = (C'_1, C'_2, \ldots, C'_k)$, όπου $C'_i = C_i \setminus \{u_i\} \cup \{l_i, r_i\}$. Το $\mathcal{C}'$ είναι ένας χρωματισμός k χρωμάτων του G' με κόστος:

$$\sum_{i=1}^{k} w(l_i) = \sum_{i=1}^{k} i = \frac{k(k+1)}{2}$$

Τα σύνολα S_l και S_r σχηματίζουν αμφότερα από μια κλίκα συνολικού βάρους $\frac{k(k+1)}{2}$. Άρα, το κόστος του μέγιστου χρωματισμού του G' θα είναι μεγαλύτερο ή ίσο της τιμής $\frac{k(k+1)}{2}$. Υποθέτουμε ότι ο G' έχει ένα μέγιστο χρωματισμό κορυφών, έστω $\mathcal{C}' = (C'_1, C'_2, \ldots, C'_t)$, κόστους $\frac{k(k+1)}{2}$. Θα πρέπει να ισχύει $t = k$ και $\{l_i, r_i\} \in C'_i$, $i = 1, \ldots, k$, διότι διαφορετικά το κόστος του χρωματισμού αυξάνεται. Θέτουμε λοιπόν $C_i = C'_i \setminus \{l_i, r_i\} \cup \{u_i\}$, οπότε ο $\mathcal{C} = (C_1, C_2, \ldots, C_k)$ είναι ένας χρωματισμός k χρωμάτων του G. □

7.3 Προσεγγιστικοί αλγόριθμοι

Σε αυτή την ενότητα θα παρουσιάσουμε τρεις προσεγγιστικούς αλγόριθμους για το πρόβλημα του μέγιστου χρωματισμού κορυφών σε γράφους διαστημάτων.

7.3.1 Μέγιστος χρωματισμός μέσω άμεσων αλγορίθμων

Έστω $\mathcal{A}$ ένας αλγόριθμος, ο οποίος επιλύει το πρόβλημα του άμεσου χρωματισμού κορυφών σε γράφους διαστημάτων. Δεδομένου του $\mathcal{A}$, ένας γενικός προσεγγιστικός αλγόριθμος για την επίλυση του μέγιστου χρωματισμού κορυφών στο γράφο διαστημάτων $G = (V, E)$ είναι ο Αλγόριθμος PRV ([5]).

Αλγόριθμος PRV(A)

1. Ταξινόμησε τις κορυφές του G κατά φθίνουσα σειρά σύμφωνα με τα βάρη τους. Έστω $u_1, \ldots, u_n$ αυτή η ταξινόμηση;
2. Παρουσίασε τις κορυφές στον Αλγόριθμο $\mathcal{A}$ με τη σειρά $u_1, \ldots, u_n$;
3. Επέστρεψε το χρωματισμό που κατασκεύασε ο Αλγόριθμος $\mathcal{A}$;

Στο Θεώρημα 7.2 θα δείξουμε ότι το πλήθος των χρωμάτων που χρησιμοποιεί ο άμεσος αλγόριθμος $\mathcal{A}$ και το βάρος του μέγιστου χρωματισμού που παράγεται από τον Αλγόριθμο PRV σχετίζονται. Έτσι, αν χρησιμοποιήσουμε τους άμεσους αλγόριθμους του Κεφαλαίου 6, μπορούμε να κατασκευάσουμε προσεγγιστικούς αλγόριθμους σταθερού παράγοντα για το μέγιστο χρωματισμό κορυφών σε γράφους διαστημάτων.

Θεώρημα 7.2
Έστω $\mathcal{G}$ μια κληρονομική κλάση γράφων και έστω $\mathcal{A}$ ένας άμεσος αλγόριθμος,

ο οποίος χρησιμοποιεί το πολύ k χρώματα για το χρωματισμό οποιουδήποτε $G \in \mathcal{G}$, όπου $k \leq \rho \cdot \chi(G)$ και $\rho > 0$ ακέραια σταθερά. Τότε, για οποιαδήποτε συνάρτηση βάρους $w : V \to \mathbb{N}$, ο χρωματισμός που κατασκευάζει ο Αλγόριθμος PRV έχει βάρος το πολύ $\rho \cdot opt$, όπου opt είναι το βάρος του βέλτιστου μέγιστου χρωματισμού του γράφου G.

Απόδειξη. Έστω $\mathcal{C}^* = \{C_1^*, C_2^*, \ldots, C_t^*\}$ ο βέλτιστος μέγιστος χρωματισμός του G και έστω $w_i^* = max_{u \in C_i^*} w(u)$. Χωρίς βλάβη της γενικότητας, υποθέτουμε ότι $w_1^* \geq w_2^* \geq \ldots \geq w_t^*$. Προφανώς, $opt = \sum_{i=1}^{t} w_i^*$. Η βέλτιστη λύση χρησιμοποιεί τουλάχιστον $\chi(G)$ χρώματα, οπότε $t \geq \chi(G)$. Έστω $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ ο χρωματισμός που κατασκευάζει ο Αλγόριθμος PRV και έστω $w_i = max_{u \in C_i} w(u)$. Χωρίς βλάβη της γενικότητας, θεωρούμε ότι $w_1 \geq w_2 \geq \ldots \geq w_k$. Ισχύει ότι $k \leq \rho \cdot \chi(G) \leq \rho \cdot t$. Για λόγους ευκολίας συμβολισμού, ορίζουμε τα σύνολα $C_{k+1}, C_{k+2}, \ldots, C_{c \cdot \chi(G)} = \emptyset$ και θέτουμε $w_i = 0$, $k < i \leq \rho \cdot \chi(G)$.

Θα δείξουμε ότι $w_i^* \geq w_j$, $1 \leq i \leq t$ και $\rho \cdot (i-1) < j \leq \rho \cdot i$. Δεδομένου ότι ορίσαμε ως w_1^* το μέγιστο βάρος όλων των κορυφών του G, ο ισχυρισμός ισχύει για $i = 1$, δηλαδή $w_1^* \geq w_j$, $0 < j \leq \rho$. Για $i \geq 2$, ορίζουμε το υποσύνολο κορυφών $V_i = \{u | w(u) > w_i^*\}$. Το V_i περιλαμβάνει τουλάχιστον τις κορυφές με βάρη $w_1^*, \ldots, w_{i-1}^*$ και επομένως ο βέλτιστος μέγιστος χρωματισμός $\mathcal{C}^*$ του G περιορισμένος στις κορυφές του V_i αποτελεί ένα χρωματισμό $(i-1)$ χρωμάτων για τον υπογράφο $G[V_i]$. Επομένως, ο $\mathcal{A}$ χρησιμοποιεί το πολύ $\rho \cdot (i-1)$ χρώματα για να χρωματίσει τις κορυφές του $G[V_i]$. Κατά τον Αλγόριθμο PRV, οι κορυφές παρουσιάζονται στον $\mathcal{A}$ κατά φθίνουσα σειρά των βαρών τους, οπότε όλες οι κορυφές του V_i χρωματίζονται νωρίτερα από οποιαδήποτε κορυφή βάρους w_i^*. Επομένως, το μέγιστο βάρος όλων των κορυφών που βρίσκονται στις κλάσεις χρωμάτων C_j, $\rho \cdot (i-1) < j \leq \rho \cdot i$ είναι w_i^*, δηλαδή $w_i^* \geq w_j$, $1 \leq i \leq t$ και $\rho \cdot (i-1) < j \leq \rho \cdot i$.

Εύκολα τώρα καταλήγουμε στο ζητούμενο αποτέλεσμα, διότι ο χρωματισμός που κατασκευάζει ο Αλγόριθμος PRV έχει κόστος:

$$\sum_{l=1}^{\rho \cdot \chi(G)} w_l = \sum_{i=1}^{\chi(G)} \sum_{j=\rho \cdot (i-1)+1}^{\rho \cdot i} w_j = \sum_{i=1}^{\chi(G)} \rho \cdot w_j \leq \rho \sum_{i=1}^{\chi(G)} w_i^* \leq \rho \cdot opt.$$

□

Σημειώνουμε ότι οι γράφοι διαστημάτων είναι μια κληρονομική κλάση γράφων. Όπως αποδείξαμε στο Θεώρημα 6.4, ο Αλγόριθμος FF χρησιμοποιεί το πολύ $8 \cdot \chi(G)$ κλάσεις χρωμάτων για τον άμεσο χρωματισμό των κορυφών ενός γράφου διαστημάτων G. Χρησιμοποιώντας λοιπόν τον Αλγόριθμο FF στη θέση του Αλγόριθμου $\mathcal{A}$ στον Αλγόριθμο PRV και θέτοντας $\rho = 8$ στο Θεώρημα 7.2, προκύπτει ότι:

Πόρισμα 7.3
Ο Αλγόριθμος PRV(FF) είναι ένας 8-προσεγγιστικός αλγόριθμος για το πρόβλημα του μέγιστου χρωματισμού κορυφών ενός γράφου διαστημάτων G.

Στο Θεώρημα 6.6, αποδείξαμε ότι ο Αλγόριθμος KT χρησιμοποιεί το πολύ $3\ \chi(G) - 2$ κλάσεις χρωμάτων για τον άμεσο χρωματισμό ενός γράφου διαστημάτων G, δηλαδή $k \leq 3\ \chi(G) - 2 < 3\ \chi(G)$. Χρησιμοποιώντας λοιπόν τον Αλγόριθμο KT στη θέση του Αλγόριθμου $\mathcal{A}$ στον Αλγόριθμο PRV και θέτοντας $\rho = 3$ στο Θεώρημα 7.2, προκύπτει ότι:

Πόρισμα 7.4
Ο Αλγόριθμος PRV(KT) είναι ένας 3-προσεγγιστικός αλγόριθμος για το πρόβλημα του μέγιστου χρωματισμού κορυφών ενός γράφου διαστημάτων G.

7.3.2 Ένας 2-προσεγγιστικός αλγόριθμος

Μπορούμε να τροποποιήσουμε κάποια βήματα του Αλγόριθμου KT και έτσι να ελαττώσουμε τον προσεγγιστικό παράγοντα από 3 σε 2. Υπενθυμίζουμε ότι κάθε υπογράφος $G[S_i]$ σχηματίζεται από την ένωση μη τεμνόμενων μονοπατιών και επομένως υπάρχει άμεσος αλγόριθμος που χρωματίζει τον $G[S_i]$ με το πολύ 3 χρώματα. Συνεπώς, αν χρωματίσουμε τις κορυφές του $G[S_i]$ βλέποντας ολόκληρο τον υπογράφο και όχι άμεσα, θα χρησιμοποιήσουμε το πολύ 2 κλάσεις χρωμάτων. Έτσι, χρησιμοποιώντας ως $\mathcal{A}$ τον τροποποιημένο Αλγόριθμο KT, προκύπτει ο Αλγόριθμος PRV2.

Αλγόριθμος PRV2

1. Ταξινόμησε τις κορυφές του G κατά φθίνουσα σειρά σύμφωνα με τα βάρη τους. Έστω $u_1, \ldots, u_n$ αυτή η ταξινόμηση;
2. $k = 1$;
3. $S_k = \emptyset$;
4. for $j = 1, \ldots, n$ do
5. Βάλε την u_j στο S_i, όπου $1 \leq i \leq k$ είναι ο μικρότερος δείκτης, τέτοιος ώστε το $S_1 \cup S_2 \cup \ldots \cup (S_i \cup \{u\})$ να μην περιέχει κλίκα μεγέθους $i + 1$;
6. if δεν υπάρχει τέτοιο i then θέσε $k = k + 1$ και βάλε την u_j στο S_k;
7. Χρωμάτισε τις κορυφές του S_1 με το χρώμα 1;
8. for $i = 2, \ldots, k$ do
9. Χρωμάτισε τις κορυφές του S_i με τα χρώματα $2i - 1$ και $2i - 2$;

Θεώρημα 7.5
Ο Αλγόριθμος PRV2 είναι ένας 2-προσεγγιστικός αλγόριθμος για το πρόβλημα του μέγιστου χρωματισμού κορυφών ενός γράφου διαστημάτων G.

Απόδειξη. Έστω $C^* = \{C_1^*, C_2^*, \ldots, C_t^*\}$ ο βέλτιστος μέγιστος χρωματισμός για το G και έστω $w_i^* = max_{u \in C_i^*} w(u)$. Υποθέτουμε, χωρίς βλάβη της γενικότητας, ότι $w_1^* \geq w_2^* \geq \ldots \geq w_t^*$. Ισχύει $opt = \sum_{i=1}^{t} w_i^*$ και $t \geq \chi(G)$.

Έστω $S_1, S_2, \ldots, S_k$ τα σύνολα κορυφών που κατασκευάζει ο Αλγόριθμος PRV2. Μια κορυφή u τοποθετείται στο S_k, διότι το $S_1 \cup S_2 \cup \ldots \cup (S_{k-1} \cup u)$ περιέχει κλίκα μεγέθους k. Συνεπώς, $\chi(G) \geq k$, οπότε $k \leq t$. Έστω $w_i = max_{u \in S_i} w(u)$. Θα δείξουμε ότι $w_i^* \geq w_i$, $1 \leq i \leq k$.

Αμφότερα τα σύνολα C_1^* και S_1 περιέχουν την κορυφή με το μέγιστο βάρος w_1^*, οπότε είναι προφανές ότι $w_1^* = w_1$. Υποθέτουμε ότι i είναι ο μικρότερος δείκτης για τον οποίο $w_i^* < w_i$ και θα καταλήξουμε σε άτοπο. Εφόσον $w_i^* < w_i$, όλες οι κορυφές, των οποίων το βάρος είναι μεγαλύτερο ή ίσο του w_i, βρίσκονται σε κάποια κλάση C_j^* του βέλτιστου χρωματισμού, όπου $j < i$. Όλες αυτές οι κορυφές (οι οποίες ανήκουν στα $C_1^*, \ldots, C_{i-1}^*$) χρωματίζονται βέλτιστα με $i-1$ χρώματα, οπότε σχηματίζουν κλίκα μεγέθους το πολύ $i-1$. Αυτό σημαίνει ότι οι κορυφές του $S_1 \cup \ldots \cup S_{i-1}$ (δηλαδή όλες οι κορυφές βάρους $\geq w_i$) σχηματίζουν κλίκα μεγέθους το πολύ $i-1$ και επομένως η κορυφή βάρους w_i θα μπορούσε να είχε τοποθετηθεί σε κάποιο εκ των $S_1, \ldots, S_{i-1}$ και όχι στο S_i. Άτοπο και άρα $w_i^* \geq w_i$, $1 \leq i \leq k$.

Χρωματίζοντας τις κορυφές των συνόλων $S_1, S_2, \ldots, S_k$, λαμβάνουμε ένα χρωματισμό για το G κόστους το πολύ $w_1 + 2\ \sum_{i=2}^{k} w_i$ (αφού για κάθε σύνολο, εκτός του S_1, χρησιμοποιούμε δύο χρώματα). Ισχύει:

$$w_1 + 2\sum_{i=2}^{k} w_i \leq w_1^* + 2\sum_{i=2}^{k} w_i^* \leq 2 \cdot opt$$

□

Κεφάλαιο 8

Χρονοπρογραμματισμός εργασιών με διάταξη διαστημάτων

8.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα μελετήσουμε ειδικές περιπτώσεις του κλασσικού προβλήματος χρονοπρογραμματισμού ενός μερικώς διατεταγμένου συνόλου εργασιών σε ένα σύνολο επεξεργαστών, έτσι ώστε ο χρόνος ολοκλήρωσής τους να είναι ο ελάχιστος δυνατός.

Ορισμός 8.1
Ένα μερικώς διατεταγμένο σύνολο - μ.δ. σύνολο (**partially ordered set - p.o. set**) *$P = (V, \prec)$ είναι ένα σύνολο V εφοδιασμένο με μια σχέση μερικής διάταξης $\prec$. Μια σχέση $\prec$ είναι σχέση μερικής διάταξης ενός συνόλου V, αν έχει τις εξής τρεις ιδιότητες:*
(i) Αυτοπάθεια: $v \prec v$, $\forall v \in V$.
(ii) Αντισυμμετρικότητα: αν $v \prec u$ και $u \prec v$, τότε $v = u$.
(iii) Μεταβατικότητα: αν $v \prec w$ και $w \prec u$, τότε $v \prec u$.

Ένα μ.δ. σύνολο εργασιών $P = (V, \prec)$ αντιστοιχεί σε έναν κατευθυνόμενο ακυκλικό γράφο $G = (V, E)$, του οποίου κορυφές είναι οι εργασίες του συνόλου V και αν $v \prec u$, τότε $(v \to u) \in E$.

Στο κεφάλαιο αυτό υποθέτουμε ότι οι επεξεργαστές είναι πλήρως συνδεδεμένοι μεταξύ τους και ότι κάθε εργασία απαιτεί ακριβώς μια μονάδα χρόνου για να εκτελεστεί σε οποιονδήποτε επεξεργαστή (**Unit Execution Time - UET**). Η σειρά εκτέλεσης των εργασιών υπόκειται στη μεταξύ τους σχέση μερικής διάταξης: αν $v \prec u$, όπου $v, u \in V$, τότε η εργασία u μπορεί να αρχίσει την εκτέλεσή της μόνο μετά την ολοκλήρωση της εργασίας v (η u χρησιμοποιεί τα αποτελέσματα του υπολογισμού της v).

Στη συνέχεια θα εξετάσουμε δύο διαφορετικά μοντέλα εκτέλεσης των εργασιών σε σχέση με τον απαιτούμενο χρόνος επικοινωνίας ανάμεσα σε ζευγάρια εργασιών $v, u \in V$ τέτοια ώστε $v \prec u$ και καθεμιά από τις οποίες εκτελείται σε διαφορετικό επεξεργαστή. Έστω ότι η εργασία v ξεκινά την εκτέλεσή της στον επεξεργαστή i τη χρονική στιγμή t.

Στο πρώτο μοντέλο εκτέλεσης το κόστος επικοινωνίας θεωρείται μηδενικό και η εργασία u μπορεί να ξεκινήσει την εκτέλεσή της σε οποιονδήποτε επεξεργαστή τη χρονική στιγμή $t_1 \geq t+1$.

Στο δεύτερο μοντέλο εκτέλεσης το κόστος επικοινωνίας θεωρείται μοναδιαίο (**Unit Communication Time - UCT**) και η εργασία u μπορεί να ξεκινήσει την εκτέλεσή της:

- στον επεξεργαστή i τη χρονική στιγμή $t_1 \geq t+1$, ή
- σε οποιονδήποτε άλλο επεξεργαστή j, $j \neq i$, τη χρονική στιγμή $t_1 \geq t+2$.

Είναι γνωστό [31] ότι το πρόβλημα του χρονοπρογραμματισμού ενός γενικού μ.δ. συνόλου εργασιών με μοναδιαίους χρόνους εκτέλεσης είναι $\mathcal{NP}$-πλήρες, ακόμα και στην περίπτωση μηδενικού κόστους επικοινωνίας.

Στο κεφάλαιο αυτό θα παρουσιάσουμε δύο πολυωνυμικούς αλγόριθμους, οι οποίοι επιλύουν το παραπάνω πρόβλημα, για τα δύο παραπάνω μοντέλα εκτέλεσης, στην περίπτωση εργασιών με μερική διάταξη διαστημάτων.

Στην επόμενη παράγραφο ορίζεται η μερική διάταξη διαστημάτων και αποδεικνύεται μία χρήσιμη για την συνέχεια ιδιότητά της.

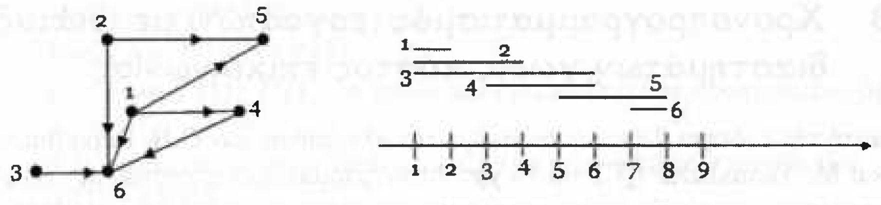
8.2 Μερική διάταξη διαστημάτων

Ορισμός 8.2
Ένα μ.δ. σύνολο $P = (V, \prec)$ ονομάζεται μ.δ. σύνολο διαστημάτων **(interval order)**, *αν για κάθε $v, u \in V$ μπορούμε να ορίσουμε διαστήματα $[l(v), r(v)]$ και $[l(u), r(u)]$ στην ευθεία των ακεραίων (όπου $l(v)$ και $r(v)$ είναι αντίστοιχα το αριστερό και δεξιό άκρο του διαστήματος που αντιστοιχεί στο v), έτσι ώστε $r(v) < l(u)$ ανν $v \prec u$.*

Η παρακάτω ιδιότητα των συνόλων μερικής διάταξης διαστημάτων είναι βασική για τους αλγόριθμους χρονοπρογραμματισμού που θα παρουσιαστούν στη συνέχεια. Για μία εργασία $v \in V$ ορίζουμε ως $S(v) = \{u \in V | v \prec u\}$ το σύνολο των απογόνων (**successors**) της v στο μ.δ. σύνολο P.

Λήμμα 8.3
Αν το $P = (V, \prec)$ είναι ένα μ.δ. σύνολο διαστημάτων, τότε $\forall v, u \in V$ είτε $S(v) \subseteq S(u)$ είτε $S(u) \subseteq S(v)$.

Απόδειξη. Προφανώς, το λήμμα ισχύει αν $S(v) = \emptyset$ ή $S(u) = \emptyset$. Έτσι, υποθέτουμε ότι υπάρχουν $v', u' \in V$, τέτοια ώστε $v \prec v'$ και $u \prec u'$ και τα σύνολα



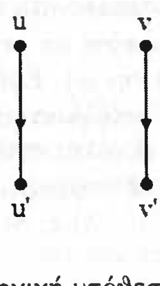
Σχήμα 8.1: Ένα σύνολο διάταξης διαστημάτων και ο γράφος διαστημάτων που ορίζεται από αυτό.

$S(v), S(u)$ να περιέχουν από ένα τουλάχιστον στοιχείο. Εφόσον το $P = (V, \prec)$ είναι ένα σύνολο διάταξης διαστημάτων, έπεται ότι για κάθε $v, u \in V$ μπορούμε να ορίσουμε διαστήματα $[l(v), r(v)]$ και $[l(u), r(u)]$, έτσι ώστε $r(v) < l(u)$ ανν $v \prec u$ (Ορισμός 8.2). Αυτό όμως δεν είναι εφικτό για το σύνολο διάταξης διαστημάτων του Σχήματος 8.2, όπου τα $S(v)$ και $S(u)$ περιέχουν από ακριβώς ένα στοιχείο. Επομένως τουλάχιστον ένα από τα $S(v), S(u)$ θα πρέπει να περιέχει περισσότερα του ενός στοιχεία. Εξετάζοντας όλες τις πιθανές περιπτώσεις, έχουμε ότι αν $v \prec u$ ή $v' \prec u'$ ή $v' \prec u$, τότε, λόγω της μεταβατικής ιδιότητας, ισχύει ότι $v \prec u'$. Ομοίως, αν $u \prec v$ ή $u' \prec v$ ή $u' \prec v$, τότε $u \prec v'$. Άρα, θα ισχύει σίγουρα είτε $v \prec u'$ είτε $u \prec v'$. Διακρίνουμε, λοιπόν, τις εξής δύο περιπτώσεις:

1. αν $v \prec v'$, $u \prec u'$ και $v \prec u'$, τότε $S(v) = \{v', u'\}$ και $S(u) = \{u'\}$, οπότε $S(u) \subseteq S(v)$

2. αν $v \prec v'$, $u \prec u'$ και $u \prec v'$, τότε $S(v) = \{v'\}$ και $S(u) = \{u', v'\}$, οπότε $S(v) \subseteq S(u)$

Άρα, το ζητούμενο αποδείχτηκε. □



Σχήμα 8.2: Η αρχική υπόθεση του Λήμματος 8.3

8.3 Χρονοπρογραμματισμός εργασιών με διάταξη διαστημάτων χωρίς κόστος επικοινωνίας

Σε αυτή την ενότητα θα παρουσιάσουμε τον αλγόριθμο των C.H. Papadimitriou και M. Yannakakis ([7]) για το χρονοπρογραμματισμό εργασιών με διάταξη διαστημάτων, μοναδιαίο χρόνο εκτέλεσης και μηδενικό χρόνο επικοινωνίας σε έναν αυθαίρετο αριθμό επεξεργαστών.

Αλγόριθμος PY

1. Ταξινόμησε τις $v \in V$ κατά φθίνουσα σειρά σύμφωνα με το $|S(v)|$;
2. Τοποθέτησε στον πρώτο (χρονικά) διαθέσιμο επεξεργαστή την πρώτη διαθέσιμη εργασία σύμφωνα με την παραπάνω σειρά ταξινόμησης (σε περίπτωση ισοπαλίας διάλεξε αυθαίρετα έναν επεξεργαστή);

Θεώρημα 8.4
Ο Αλγόριθμος PY επιλύει βέλτιστα το πρόβλημα του χρονοπρογραμματισμού εργασιών με διάταξη διαστημάτων.

Απόδειξη. Υποθέτουμε ότι ο Αλγόριθμος PY δεν είναι βέλτιστος και θα καταλήξουμε σε άτοπο. Έστω $P = (V, \prec)$ το μικρότερο μ.δ. σύνολο για το οποίο ο Αλγόριθμος PY δεν είναι βέλτιστος. Έστω ένας βέλτιστος χρονοπρογραμματισμός. Ορίζουμε ως $T(i)$ και $T'(i)$ τα υποσύνολα των εργασιών του V, τα οποία εκτελούνται τη χρονική στιγμή i σύμφωνα με τον PY και το βέλτιστο χρονοπρογραμματισμό αντίστοιχα. Ισχύει ότι $T'(1) \not\subseteq T(1)$, γιατί διαφορετικά, αν $T'(1) \subseteq T(1)$, τότε ο Αλγόριθμος PY δε θα ήταν βέλτιστος για το μ.δ. σύνολο $P = (V \backslash T(1), \prec)$. Όμως, αυτό είναι αδύνατο, διότι υποθέσαμε ότι το $P = (V, \prec)$ είναι το μικρότερο μ.δ. για το οποίο ο PY δεν είναι βέλτιστος.

Αφού $T'(1) \not\subseteq T(1)$, έστω $v \in T'(1) \setminus T(1)$. Η v χρονοπρογραμματίζεται βέλτιστα τη χρονική στιγμή $t = 1$, οπότε το $S(v)$ είναι ένα μεγιστικό σύνολο (διότι διαφορετικά αν το $S(v)$ ήταν υποσύνολο ενός άλλου μεγαλύτερου - π.χ. του $S(w) = S(v) \cup v$ - δε θα μπορούσε να εκτελεστεί τη χρονική στιγμή 1, γιατί θα έπρεπε να εκτελεστεί μετά την w). Εφόσον ο Αλγόριθμος PY χρονοπρογραμματίζει τις εργασίες του V σύμφωνα με την ταξινόμηση της Γραμμής 1 και το $S(v)$ είναι ένα μεγιστικό σύνολο, συνεπάγεται ότι η v θα μπορούσε να ανήκει στο $T(1)$. Όμως αυτό δε συμβαίνει, οπότε θα πρέπει να υπάρχει $u \in T(1) \setminus T'(1)$, έτσι ώστε $u \prec v$. Άρα, $S(v) \subseteq S(u)$. Κατασκευάζουμε σύνολα $T^{(k)}(1)$ ως εξής:

1. $T^{(0)}(1) = T'(1)$;
2. **for** $k = 1, \ldots, |T'(1) \setminus T(1)|$ **do**

3. $T^{(k)}(1) = (T^{(k-1)}(1) \setminus \{v\}) \cup \{u\}$;
4. Έστω $v \in T^{(k)}(1) \setminus T(1)$;
5. Έστω $u \in T(1) \setminus T'(1)$, το οποίο δεν έχει επιλεγεί σε προηγούμενο βήμα;

6. **if** δεν υπάρχει τέτοιο u **then** επέλεξε ένα οποιοδήποτε στοιχείο του $T(1) \setminus T'(1)$;

Όταν ολοκληρωθεί η παραπάνω διαδικασία, έχουμε ότι $T^{(k)}(1) = T(1)$. Εφόσον $S(v) \subseteq S(u)$, κάθε $T^{(k)}(1)$ είναι εφικτό και βέλτιστο και επομένως το $T(1)$ είναι βέλτιστο. Άτοπο, διότι το $P = (V, \prec)$ είναι το μικρότερο σύνολο για το οποίο ο Αλγόριθμος PY δεν είναι βέλτιστος. Άρα, ο Αλγόριθμος PY είναι βέλτιστος. □

8.4 Χρονοπρογραμματισμός εργασιών με διάταξη διαστημάτων και κόστος επικοινωνίας

Σε αυτή την ενότητα θα παρουσιάσουμε τον αλγόριθμο των **Hesham H. Ali** και **Hesham El-Rewini** ([32],[33])) για το χρονοπρογραμματισμό εργασιών με διάταξη διαστημάτων, μοναδιαίο χρόνο εκτέλεσης και μοναδιαίο χρόνο επικοινωνίας σε έναν αυθαίρετο αριθμό επεξεργαστών M. Με $T(i,t)$ συμβολίζουμε την εργασία, η οποία πρόκειται να εκτελεστεί στον επεξεργαστή i τη χρονική στιγμή t, ενώ με t_v^i συμβολίζουμε το νωρίτερο εφικτό χρόνο εκτέλεσης της εργασίας v στον επεξεργαστή i. Ισχύει ότι $t_v^i = \min\{t | T(i,t) = 0,\ t \geq t_{thr}\}$, όπου $t_{thr} = \max\{t_{u,i} + 1, t_{u,j} + 2\}$, $\forall u \prec v$, όπου $t_{u,i}$ είναι ο πραγματικός χρόνος εκτέλεσης της εργασίας u στον επεξεργαστή i.

Αλγόριθμος AR

1. Ταξινόμησε τις $v \in V$ κατά φθίνουσα σειρά σύμφωνα με το $|S(v)|$;
2. **for** κάθε $v \in V$ σύμφωνα με την παραπάνω ταξινόμηση **do**
3.   **for** $i = 1, \ldots, M$ **do**
4.   Υπολόγισε το t_v^i;
5.   Όρισε $t_{min} = \min\{t_v^i | 1 \leq i \leq M\}$;
6.   **if** υπάρχει μόνο ένας επεξεργαστής j, για τον οποίο $t_v^j = t_{min}$ **then**
7.   Ανάθεσε την v στον j τη χρονική στιγμή t_{min};
8.   **else**
9.   Βρες τον επεξεργαστή q, για τον οποίο η εργασία u που έχει ανατεθεί τη χρονική στιγμή $t_{min} - 1$ έχει το ελάχιστο $|S(u)|$;
10.   Ανάθεσε την v στον q τη χρονική στιγμή t_{min};

Θεώρημα 8.5
Ο Αλγόριθμος AR επιλύει βέλτιστα το πρόβλημα του χρονοπρογραμματισμού εργασιών με διάταξη διαστημάτων και μοναδιαία κόστη εκτέλεσης και επικοινωνίας.

Απόδειξη. Θα αποδείξουμε ότι ο Αλγόριθμος **AR** είναι βέλτιστος χρησιμοποιώντας επαγωγή στον αριθμό των εργασιών. Ο Αλγόριθμος **AR** είναι προφανώς βέλτιστος στην περίπτωση της μιας εργασίας. Υποθέτουμε ότι χρονοπρογραμματίζει βέλτιστα τις k πρώτες εργασίες. Έστω x η k+1-οστή εργασία (σύμφωνα με την ταξινόμηση της Γραμμής 1), η οποία εκτελείται στον επεξεργαστή i τη χρονική στιγμή t. Από τον Αλγόριθμο **AR**, έχουμε ότι $t_x^i \leq t_x^j$, $1 \leq j \leq M$. Υποθέτουμε ότι υπάρχει ένας βέλτιστος χρονοπρογραμματισμός, ο οποίος συμφωνεί με τον Αλγόριθμο **AR** στο χρονοπρογραμματισμό των πρώτων k εργασιών, αλλά αναθέτει την x στον q τη στιγμή t_1. Θα εξετάσουμε δύο περιπτώσεις, όπου θα δείξουμε ότι μπορούμε να εκτελέσουμε την x στον i τη στιγμή t (όπως ο Αλγόριθμος **AR**), διατηρώντας βέλτιστο και εφικτό το χρονοπρογραματισμό. Έτσι, θα αποδείξουμε ότι ο Αλγόριθμος **AR** είναι βέλτιστος.

- Περίπτωση 1η: Υποθέτουμε ότι ο βέλτιστος χρονοπρογραμματισμός αναθέτει την x στον q τη στιγμή $t_1 > t$ (δηλαδή $x = T(q, t_1)$), όπου δεν ισχύει απαραίτητα $q \neq i$. Εφόσον η $k + 1$-οστή εργασία που επιλέγει ο Αλγόριθμος **AR** είναι η x, έπεται ότι $|S(x)| \geq |S(T(i,t))|$, οπότε από το Λήμμα 8.3 είναι $S(T(i,t)) \subseteq S(x)$ και συνεπώς η $T(i,t)$ μπορεί να εκτελεστεί στον q τη στιγμή t_1. Επειδή οι δύο χρονοπρογραμματισμοί συμφωνούν στις πρώτες k εργασίες και ο Αλγόριθμος **AR** αναθέτει την x στον i τη στιγμή t, ισχύουν $T(j, t-1) \not\prec x$, $1 \leq j \leq M$, $j \neq i$ και $T(j, t_2) \not\prec x$, $1 \leq j \leq M$, $t \leq t_2 \leq t_1$. Άρα, η x μπορεί να εκτελεστεί στον i τη στιγμή t. Επομένως, η αντιμετάθεση των εργασιών $T(i,t)$ και x είναι εφικτή.

- Περίπτωση 2η: Υποθέτουμε ότι ο βέλτιστος χρονοπρογραμματισμός αναθέτει την x στον $q, q \neq i$ τη στιγμή t (δηλαδή $x = T(q,t)$). Θα δείξουμε ότι μπορούμε να αντιμεταθέσουμε όλες τις εργασίες που εκτελούνται στους q και i για όλες τις χρονικές στιγμές $\geq t$. Εφόσον ο Αλγόριθμος **AR** αναθέτει την x στον i τη στιγμή t, έπεται ότι $|S(T(i, t-1))| \leq |S(T(q, t-1))|$, διότι διαφορετικά θα επέλεγε κάποιον άλλο επεξεργαστή για την εκτέλεσή της. Επίσης, $T(q, t-1) \not\prec x$, οπότε η x (και όσες έπονται χρονικά της x στον q) μπορεί να εκτελεστεί στον i τη στιγμή t. Από το Λήμμα 8.3 και το γεγονός ότι $|S(T(i, t-1))| \leq |S(T(q, t-1))|$, συνεπάγεται ότι $S(T(i, t-1)) \subseteq S(T(q, t-1))$. Ο βέλτιστος χρονοπρογραμματισμός είναι εφικτός, οπότε $T(q, t-1) \not\prec T(i,t) \Rightarrow T(i, t-1) \not\prec T(i,t)$. Έτσι, η $T(i,t)$ μπορεί να εκτελεστεί στον q τη στιγμή t. Άρα, η ζητούμενη αντιμετάθεση είναι εφικτή.

□

Επίλογος

Σε αυτή την εργασία παρουσιάσαμε αλγόριθμους που επιλύουν βέλτιστα ή προσεγγίζουν τη βέλτιστη λύση συγκεκριμένων προβλημάτων βελτιστοποίησης σε γράφους διαστημάτων και σε συγγενείς τους. Είδαμε ότι κάποια από αυτά τα προβλήματα, ενώ είναι $\mathcal{NP}$-πλήρη σε γενικούς γράφους, επιλύονται πολυωνυμικά σε γράφους διαστημάτων ή σε συγγενείς τους. Το γεγονός αυτό σε συνδυασμό με το ότι αυτές οι συγκεκριμένες κατηγορίες γράφων παρουσιάζουν ιδιαίτερο πρακτικό ενδιαφέρον σε εφαρμογές του κλάδου της Επιστήμης Υπολογιστών, έχουν αποτελέσει εφαλτήριο για την ανάπτυξη έντονης ερευνητικής δραστηριότητας σε αυτό τον τομέα.

Αν και έως σήμερα έχει σημειωθεί σημαντική πρόοδος και με το πέρασμα του χρόνου έχουν επιλυθεί αρκετά προβλήματα και έχουν προταθεί νέοι αλγόριθμοι και τεχνικές ή ακόμα και νέα προβλήματα με ερευνητικό ενδιαφέρον, υπάρχουν παρόλα ταύτα αρκετά ανοιχτά προβλήματα.

Στο Κεφάλαιο 3 εξετάσαμε το πρόβλημα του χρωματισμού ακμών, ένα εν γένει $\mathcal{NP}$-πλήρες πρόβλημα. Η $\mathcal{NP}$-πληρότητα αυτού του προβλήματος έγκειται στο γεγονός ότι, αν και γνωρίζουμε ότι ο ελάχιστος αριθμός χρωμάτων που επαρκούν για το χρωματισμό των ακμών ενός γράφου G είναι είτε $\Delta(G)$ είτε $\Delta(G)+1$, το πρόβλημα απόφασης ανάμεσα σε αυτούς τους δύο ακεραίους είναι $\mathcal{NP}$-πλήρες. Εντούτοις, ο αλγόριθμος που παρουσιάσαμε επιλύει το εν λόγω πρόβλημα βέλτιστα και σε πολυωνυμικό χρόνο για γράφους διαστημάτων με περιττό μέγιστο βαθμό. Ανοιχτό, όμως, παραμένει το πρόβλημα του χρωματισμού των ακμών ενός γράφου διαστημάτων με άρτιο μέγιστο βαθμό.

Στο Κεφάλαιο 4 μελετήσαμε το πρόβλημα του χρωματισμού κορυφών με περιορισμούς πληθυκότητας σε γράφους διαστημάτων. Ο **Gardi** έδειξε ότι για $k=2$ το πρόβλημα επιλύεται σε γραμμικό χρόνο, ενώ οι **Bodlaender** και **Jansen** απέδειξαν ότι για $k \geq 4$ το πρόβλημα είναι $\mathcal{NP}$-πλήρες. Το πρόβλημα παραμένει όμως ανοιχτό για $k=3$, δηλαδή δε γνωρίζουμε τι συμβαίνει αν θέλουμε να χρωματίσουμε τις κορυφές ενός γράφου διαστημάτων χρησιμοποιώντας κάθε χρώμα το πολύ 3 φορές. Για το αντίστοιχο πρόβλημα σε γράφους κυκλικών τόξων, σημειώνουμε ότι τα γνωστά αποτελέσματα είναι ακριβώς όμοια. Πιο συγκεκριμένα, για $k=2$ το πρόβλημα επιλύεται πολυωνυμικά [34], για $k \geq 4$ είναι $\mathcal{NP}$-πλήρες [18], ενώ για $k=3$ παραμένει ανοιχτό.

Στο Κεφάλαιο 6 αποδείξαμε ότι ο αριθμός χρωμάτων που χρησιμοποιεί ο Αλγόριθμος **FF** για το χρωματισμό ενός γράφου διαστημάτων είναι το πολύ

$8\chi(G)$ [5] και σύμφωνα με το [30] τουλάχιστον $4.4\chi(G)$. Αποτελεί ανοιχτό ερώτημα το κατά ποσόν μπορούμε να καλύψουμε αυτό το κενό ανάμεσα στο $4.4\chi(G)$ και στο $8\chi(G)$. Ομοίως, ο Αλγόριθμος **FF** χρησιμοποιεί το πολύ $(4L-3)\lceil log_3 H\rceil + L$ και τουλάχιστον $0.5Llog_2H + L$ χρώματα για το χρωματισμό ενός γράφου κυκλικών τόξων [6], οπότε η κάλυψη αυτού του κενού αποτελεί ανοιχτό πρόβλημα. Επίσης, έχει αποδειχτεί ότι κανένας άμεσος αλγόριθμος δεν μπορεί να επιτύχει λόγο ανταγωνισμού μικρότερο του $\frac{4}{3}$ για το χρωματισμό ενός γράφου κυκλικών τόξων [29]. Ο Αλγόριθμος **TRI** έχει λόγο ανταγωνισμού 4, οπότε παραμένει ανοιχτό το πρόβλημα της εύρεσης ενός μικρότερου λόγου ανταγωνισμού ή ενός καλύτερου κάτω φράγματος.

Τέλος, στο Κεφάλαιο 8 μελετήσαμε το πρόβλημα του χρονοπρογραμματισμού ενός συνόλου εργασιών με διάταξη διαστημάτων, υποθέτοντας ότι το κόστος επικοινωνίας c ανάμεσα σε δύο εργασίες που εκτελούνται σε διαφορετικούς επεξεργαστές είναι είτε μηδενικό είτε μοναδιαίο. Ανοιχτό παραμένει το πρόβλημα όπου ο χρόνος επικοινωνίας είναι μεγαλύτερος της μονάδας, δηλαδή δε γνωρίζουμε τι συμβαίνει για $c \geq 2$.

Βιβλιογραφία

[1] Jeremy P. Spinrad. *Efficient Graph Representations.* American Mathematical Society, 2003.

[2] T. H. Cormen, C. E. Leiserson, Ronald L. Rivest, and C. Stein. *Introduction to Algorithms, 2nd edition.* MIT Press, McGraw-Hill Book Company, 2000.

[3] M.S. Waterman and J.R. Griggs. Interval graphs and maps of DNA. *Bulletin of Mathematical Biology*, 48(2):189–195, 1986.

[4] T. Biedl, B. Brejova, E. Demaine, A. Hamel, A. Lopez-Ortiz, and T. Vinar. Finding hidden independent sets in interval graphs. *Theoretical Computer Science*, 310(1–3):287–307, 2004.

[5] Sriram V. Pemmaraju, Rajiv Raman, and Kasturi Varadarajan. Buffer minimization using max coloring. In *Proceedings ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 562–571, 2004.

[6] Ori Gerstel, Galen Sasaki, Shay Kutten, and Rajiv Ramaswami. Worst-case analysis of dynamic wavelength allocation in optical networks. *IEEE/ACM Transactions on Networking*, 7(6):833–846, 1999.

[7] C.H. Papadimitriou and M. Yannakakis. Scheduling interval-ordered tasks. *SIAM Journal on Computing*, 8(3):405–409, 1979.

[8] Alan Tucker. Coloring a family of circular arcs. *SIAM Journal on Applied Mathematics*, 29(3):494–502, 1974.

[9] Fanica Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. *SIAM Journal on Computing*, 1(2):180–187, 1972.

[10] Martin Charles Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, 1980.

[11] Donald J. Rose, R. Endre Tarjan, , and George S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing*, 5(2):266–283, 1976.

[12] Donald J. Rose and R. Endre Tarjan. Algorithmic aspects of vertex elimination. In *Proceedings of 7th annual ACM Symposium on Theory of Computing (STOC)*, pages 245–254, 1975.

[13] I. Holyer. The NP-completeness of some edge-partition problems. *SIAM Journal on Computing*, 10(4):713–717, 1981.

[14] V.G. Vizing. On evaluation of the chromatic number of a p-graph. *Discrete Analysis, Collection of Works of Sobolev Institute of Mathematics*, 3:3–24, 1964.

[15] V.A. Bojarshinov. Edge and total coloring of interval graphs. *Discrete Applied Mathematics*, 114(1–3):23–28, 2001.

[16] T.H. Ma and W.L. Hsu. Fast and simple algorithms for recognizing chordal comparability graphs and interval orders. *SIAM Journal on Computing*, 28(3):1004–1020, 1999.

[17] T.H. Ma. *Algorithms on Special Classes of Graphs and Partially Ordered Sets.* PhD thesis, Department of Computer Science, Republic of China, Nankang, Taipei, 1993.

[18] Hans L. Bodlaender and Klaus Jansen. Restrictions of graph partition problems. Part I. *Theoretical Computer Science*, 148(1):93–109, 1995.

[19] S. Even and O. Kariv. An $O(n^{2.5})$ algorithm for maximum matching in graphs. In *Proceedings 16th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 100–112, 1975.

[20] Frederic Gardi. Mutual exclusion scheduling with interval graphs or related classes: Complexity and algorithms. *4OR: A Quarterly Journal of Operations Research*, 4(1):87–90, 2006.

[21] G. Steiner and J.S. Yeomans. A linear time algorithm for maximum matchings in convex, bipartite graphs. *Computers and Mathematics with Applications*, 31(12):91–96, 1996.

[22] Daniel Marx. A short proof of the NP-completeness of circular arc coloring. citeseer.ist.psu.edu/672361.html, 2003.

[23] I. Karapetian. On coloring arc graphs. *Dokladi (Reports) of the Academy of Science of the Armenian Soviet Socialist Republic*, 70:306–311, 1980.

[24] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM Journal on Computing*, 5(4):691–703, 1976.

[25] Jens Vygen. NP-completeness of some edge-disjoint paths problems. *Discrete Applied Mathematics*, 61(1):83–90, 1995.

[26] Guy Even and Shimon Shahar. Scheduling of a smart antenna: Capacitated coloring of unit circular-arc graphs. In *Proceedings Combinatorial and Algorithmic Aspects of Networking (CAAN)*, pages 58–71, 2006.

[27] Mario Valencia-Pabon. Revisiting tucker's algorithm to color circular arc graphs. *SIAM Journal on Computing*, 32(4):1067–1072, 2003.

[28] H.A. Kierstead and W.T. Trotter. An extremal problem in recursive combinatorics. *Congressus Numerantium*, 33:143–153, 1981.

[29] M.V. Marathe, H.B. Hunt, and S.S. Ravi. Efficient approximation algorithms for domatic partition and on-line coloring of circular arc graphs. *Discrete Applied Mathematics*, 64(2):135–149, 1996.

[30] M. Chrobak and M. Slusarek. On some packing problems related to dynamic storage allocation. *Theoretical Informatics and Applications*, 22(4):487–499, 1988.

[31] J. D. Ullman. Polynomial complete scheduling problems. In *Proceedings of the fourth ACM Symposium on Operating System Principles (SOSP)*, pages 96–101, 1973.

[32] Hesham H. Ali and Hesham El-Rewini. The time complexity of scheduling interval orders with communication is polynomial. *Parallel Processing Letters*, 3(1):53–58, 1993.

[33] Hesham H. Ali and Hesham El-Rewini. An optimal algorithm for scheduling interval ordered tasks with communication on n processors. *Journal of Computer and System Sciences*, 51(2):301–306, 1995.

[34] A.V. Golberg and A.V. Karzanov. Maximum skew-symmetric flows. In *Proceedings of the Third Annual European Symposium on Algorithms (ESA)*, pages 155–170, 1995.