Υλοποίηση και Πειραματική Μελέτη του Αλγόριθμου των Hochbaum-Nishizeki-Shmoys για Χρωματισμό Ακμών

Ζήσης Προκόπης Ταλαμάγκας

Διπλωματική Εργασία

Επιβλέπων: Χρήστος Νομικός

Ιωάννινα, Μάρτιος, 2023

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

Ευχαριστίες

Με την περάτωση της παρούσας διπλωματικής εργασίας θα ήθελα να ευχαριστήσω θερμά τον αναπληρωτή καθηγητή του τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής Ιωαννίνων κ. Χρήστο Νομικό, αρχικά για την ευκαιρία που μου έδωσε να συνεργαστούμε και έπειτα για το ενδιαφέρον που έδειχνε κάθε φορά που συναντούσα κάποιο πρόβλημα, παρέχοντας όποια βοήθεια μου ήταν αναγκαία όσο πιο άμεσα γινόταν.

Θα ήθελα επίσης να ευχαριστήσω την οικογένεια μου, τόσο για την συμπαράσταση, όσο και για τις πολύτιμες συμβουλές που μου προσέφεραν αυτά τα χρόνια των σπουδών μου.

Μάρτιος 2024

Ζήσης Προκόπης Ταλαμάγκας

Περίληψη

Στην παρούσα διπλωματική εργασία θα παρουσιάσουμε και θα υλοποιήσουμε μία λύση του προβλήματος edge-coloring σε πολυγραφήματα, βασισμένη στον αλγόριθμο που πρότειναν οι Dorit S. Hochbaum, Takao Nishizeki και David B. Shmous. Το συγκεκριμένο πρόβλημα ανήκει στην κατηγορία των NP–Δύσκολων προβλημάτων, γι' αυτό και χρησιμοποιούνται προσεγγιστικοί αλγόριθμοι για την αποδοτική επίλυση του.

Έχουν προταθεί αρκετοί αλγόριθμοι που ασχολούνται με αυτό το ζήτημα, ωστόσο όπως απέδειξε ο Holyer ,εκτός αν P = NP, δεν γίνεται να υπάρξει πολυωνυμικός προσεγγιστικός αλγόριθμος για τον χρωματισμό ενός πολυγραφήματος που πάντα χρησιμοποιεί λιγότερα από $\frac{4}{3}\chi$ χρώματα, όπου χ είναι ο βέλτιστος αριθμός χρωμάτων. Αυτό καθιστά την ανεύρεση αποδεδειγμένα καλών χρωματισμών ακμών εξαιρετικά δύσκολη. Ωστόσο οι Dorit S. Hochbaum, Takao Nishizeki και David B. Shmous δημιούργησαν έναν αλγόριθμο ο οποίος ποτέ δεν χρησιμοποιεί περισσότερα από $[\frac{9}{8}\chi + \frac{3}{4}]$. Επιπλέον αν $\chi \geq [\frac{9}{8}\Delta + \frac{3}{4}]$ τότε ο αλγόριθμος χρωματίζει βέλτιστα το γράφημα σε πολυωνυμικό χρόνο. Επιπρόσθετα ο αλγόριθμος ποτέ δεν χρησιμοποιεί παραπάνω από $\frac{4}{3}\chi$ χρώματα και εκτελείται σε χρόνο $O(|E|(|V| + \Delta))$ όπου E το σύνολο των ακμών και V το σύνολο των κορυφών.

Η υλοποίηση έγινε στη γλώσσα προγραμματισμού *Python*, και κατά την ολοκλήρωση του αλγορίθμου, θα χρησιμοποιήσουμε τη βιβλιοθήκη της *Python,* **networkx**, ώστε να παρουσιάσουμε την οπτική απεικόνιση της λύσης μας.

Τέλος, έχουμε πραγματοποιήσει πειραματική μελέτη που εξετάζει την επίδοση του αλγορίθμου μας, με γραφήματα κατάλληλα ώστε να εξετάσουμε φυσιολογικές (τυχαίες) αλλά και ακραίες περιπτώσεις.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Κρίσιμο μονοπάτι, NP-Δύσκολο Πρόβλημα, Κύκλος, Πολυγράφημα, ab-μονοπάτι, Υπογράφημα, Κοινό χρώμα.

Abstract

In this thesis, we will present and implement a solution to the edge-coloring problem in multigraphs, based on the algorithm developed by Dorit S. Hochbaum, Takao Nishizeki, and David B. Shmoys. This particular problem is categorized as NP-Hard, hence approximation algorithms are used for efficient resolution.

Several algorithms have been proposed to address this issue, but as a result of Holyer, unless P = NP, a polynomial approximation algorithm for multigraph coloring that always uses fewer than $\frac{4}{3}\chi$ colors, where χ is the optimal number of colors, cannot exist. This makes finding provably good edge colorings exceptionally challenging. However, Hochbaum, Nishizeki, and Shmoys created an algorithm that never uses more than $[9/8\ \chi + 3/4]$ colors. Moreover, if $\chi \geq [9/8\ \Delta + 3/4]$, then the algorithm colors the graph optimally in polynomial time. The algorithm also never uses more than $4/3\chi$ colors and runs in $O(|E|(|V| + \Delta))$ time, where E is the set of edges, and V is the set of vertices.

The implementation was done in Python, utilizing the networkx library for visual representation. Finally, an experimental study was conducted to evaluate our algorithm's performance, including normal (random) and extreme cases.

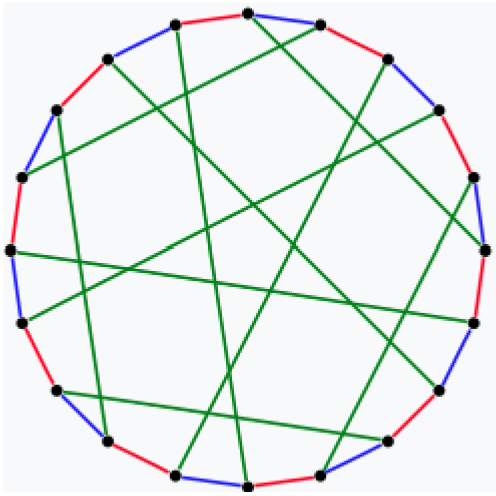
Περιεχόμενα

Κεφάλαιο 1. Εισαγωγή

1.1 Ορισμός και Εφαρμογές του Προβλήματος

Το αντικείμενο αυτής της Διπλωματικής Εργασίας, είναι η υλοποίηση και πειραματική μελέτη του αλγόριθμου που παρουσίασαν οι Dorit S. Hochbaum, Takao Nishizeki και David B. Shmous για το πρόβλημα edge-coloring σε πολυγραφήματα. Πριν όμως ξεκινήσουμε να εμβαθύνουμε στον τρόπο επίλυσης του προβλήματος μας, είναι αναγκαίο να εξηγήσουμε, τη φύση του προβλήματος και έπειτα, το λόγο που επιλέξαμε τον συγκεκριμένο τρόπο επίλυσης.

Στη Θεωρία Γραφημάτων ο χρωματισμός ακμών ενός γραφήματος είναι ένας από τους ποικίλους τρόπους χρωματισμού γραφήματος. Το πρόβλημα χρωματισμού ακμών διερευνά αν είναι εφικτό να χρωματίσουμε τις ακμές ενός δοθέντος γραφήματος, χρησιμοποιώντας μέχρι κ διαφορετικά χρώματα, όπου κ μια δεδομένη τιμή, ή με όσο το δυνατόν λιγότερα χρώματα. Ο ελάχιστος αριθμός χρωμάτων για τις ακμές ενός δοθέντος γραφήματος ονομάζεται χρωματικός δείκτης του γραφήματος. Για παράδειγμα, οι ακμές του γραφήματος της εικόνας μπορούν να χρωματιστούν με τρία χρώματα αλλά όχι με δύο, οπότε το εμφανιζόμενο γράφημα έχει χρωματικό δείκτη ίσο με τρία.



Σύμφωνα με το θεώρημα του Vizing, ο απαιτούμενος αριθμός χρωμάτων για το χρωματισμό ενός απλού γραφήματος είναι είτε *Δ* ή *Δ+1* όπου Δ ο βαθμός του *γραφήματος*. Σε κάποια γραφήματα, όπως στα διμερή ή στα επίπεδα υψηλού βαθμού, ο αριθμός των χρωμάτων είναι πάντα *Δ* , και για τα πολυγραφήματα ο αριθμός των χρωμάτων μπορεί να είναι ως και *3Δ/2*. Υπάρχουν πολυωνυμικοί αλγόριθμοι για το βέλτιστο χρωματισμό διμερών γραφημάτων και το χρωματισμό απλών μη διμερών γραφημάτων οι οποίοι χρησιμοποιούν μέχρι και *Δ+1* χρώματα. Ωστόσο, το γενικό πρόβλημα ανεύρεσης του βέλτιστου χρωματισμού ακμών είναι το NP-πλήρης και ακόμα και οι ταχύτεροι γνωστοί αλγόριθμοι χρειάζονται απαγορευτικά μεγάλο χρόνο για την αντιμετώπισή του. Έχουν μελετηθεί πολλές παραλλαγές του προβλήματος χρωματισμού ακμών, στις οποίες οι αναθέσεις χρωμάτων στα άκρα/ακμές πρέπει να πληρούν όρους/συνθήκες μη γειτνίασης (στις οποίες δεν μπορούν δύο κοντινές ακμές να έχουν το ίδιο χρώμα). Ο χρωματισμός ακμών έχει εφαρμογή σε προβλήματα προγραμματισμού και στην εκχώρηση συχνοτήτων για δίκτυα οπτικών ινών.

Στόχος εν κατακλείδι, είναι ο βέλτιστος χρωματισμός ακμών με όσο το δυνατόν λιγότερα χρώματα και προφανώς στο λιγότερο δυνατόν χρονικό διάστημα.

1.2 Κλάσεις Πολυπλοκότητας

Παραπάνω αναφέρονται κάποια προβλήματα κλάσης P και NP, χωρίς να έχει οριστεί όμως τι ακριβώς σημαίνουν αυτοί οι συμβολισμοί. Η θεωρία πολυπλοκότητας ταξινομεί ταπροβλήματα σε κλάσεις πολυπλοκότητας, έτσι ώστε τα προβλήματα στην ίδια κλάση να έχουν την ίδια δυσκολία. Δύο εξαιρετικά ενδιαφέρουσες κλάσεις πολυπλοκότηταςείναι η P και NP.

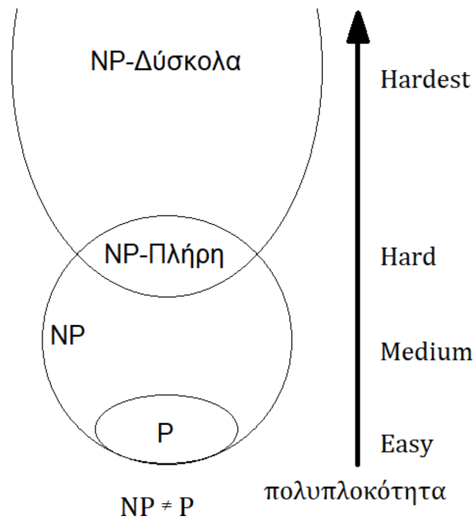
- **P**: Deterministic Polynomial Time ή αλλιώς Ντετερμινιστικός Πολυωνυμικός Χρόνος. Αποτελείται από όλα εκείνα τα προβλήματα απόφασης, τα οποία επιλύονται από μία ντετερμινιστική Μηχανή Turing σε πολυωνυμικό χρόνο, τα προβλήματα δηλαδή, που μπορούν να λυθούν στη χειρότερη περίπτωση σε πολυωνυμικό χρόνο $O(n^k)$, όπου k σταθερά.
- **NP**: Non-Deterministic Polynomial Time ή αλλιώς Μη-Ντετερμινιστικός Πολυωνυμικός Χρόνος. Η κλάση NP περιλαμβάνει όλα τα προβλήματα απόφασης που επιλύονται από μία μη ντετερμινιστική Μηχανή Turing σε πολυωνυμικό χρόνο. Όσον αφορά στα NP προβλήματα, είναι εύκολο να ελεγχθεί η ορθότητα μιας θετικής απάντησης σε ένα ερώτημα απόφασης με τη βοήθεια ενός πιστοποιητικού. Δεν ζητείται δηλαδή ο τρόπος εύρεσης της λύσης, αλλά για μία δεδομένη θεωρητικά σωστή απάντηση, η επαλήθευση τηςεγκυρότητας της σε πολυωνυμικό χρόνο.

Κάθε πρόβλημα κλάσης P, περιέχεται στην κλάση NP, δηλαδή NP ⊆ P. Δεν γνωρίζουμε αν P = NP, καθώς αυτό αποτελεί ένα σημαντικό, ανοιχτό ερώτημα στην επιστήμη των υπολογιστών, με την πλειοψηφία να υποστηρίζει πως P ≠ NP. Η ουσία του ερωτήματος είναι η εξής, εάν σε κάθε πρόβλημα που μπορούμε να επιβεβαιώσουμε γρήγορα την ύπαρξη λύσης του μέσα από έναν υπολογιστή, τότε μπορεί επίσης να επιλυθεί και γρήγορα από αυτόν. Με τον όρο γρήγορα, δηλώνουμε την ύπαρξη ενός αλγορίθμου για μια διαδικασία που τρέχει σε πολυωνυμικό χρόνο.

Υποσύνολο της κλάσης NP, αποτελούν τα **NP-Πλήρη** ή **NP-Complete** προβλήματα. Θα μπορούσαμε να τα χαρακτηρίσουμε ανεπίσημα ως τα δυσκολότερατης κλάσης NP. Καθώς αυτά ανήκουν στην κλάση NP, η ορθότητα κάθε λύσης μπορεί να επαληθευτεί σε πολυωνυμικό χρόνο, όπως αναφέραμε και προηγουμένως και ένας αλγόριθμος αναζήτησης ωμής βίας μπορεί να βρει μια λύση δοκιμάζοντας όλες τις πιθανές λύσεις.

NP-Δύσκολα ή **NP-Hard**, είναι τα προβλήματα εκείνα, τα οποία είναι τόσο δύσκολα όσο και τα δυσκολότερα προβλήματα της κλάσης NP. Αξίζει να σημειωθεί, πως τα NP-Δύσκολα προβλήματα, δε χρειάζεται απαραίτητα να ανήκουν στην κλάσηNP, όπως επίσης δεν είναι αναγκαίο να είναι και προβλήματα απόφασης. Ο ακριβής ορισμός είναι πως, ένα πρόβλημα Π1 είναι NP-Δύσκολο, αν υπάρχει ένα NP-Πλήρες πρόβλημα Π2, τέτοιο ώστε το Π2 να μπορεί να αναχθεί στο Π1 σε πολυωνυμικό χρόνο. Επειδή όμως κάθε NP-Πλήρες πρόβλημα μπορεί να αναχθεί σε οποιοδήποτε άλλο NP-Πλήρες πρόβλημα σε πολυωνυμικό χρόνο, όλα τα NP-Πλήρη προβλήματα μπορούν να αναχθούν σε οποιοδήποτε NP-Δύσκολο πρόβλημα σε πολυωνυμικό χρόνο. Τότε, αν υπάρχει λύση σε ένα NP-Δύσκολο πρόβλημα σε πολυωνυμικό χρόνο,υπάρχει λύση σε όλα τα NP προβλήματα σε πολυωνυμικό χρόνο. Είναι αναγκαίο να αναφέρουμε πως, το πρόβλημα που θα μελετήσουμε σε αυτή την διπλωματική εργασία, ανήκει στα κλάση των NP-Δύσκολων προβλημάτων, επομένως για να μπορέσουμε να το επιλύσουμε αρκετά αποδοτικά, είναι αναγκαία η χρήση προσεγγιστικών αλγορίθμων.

Το διάγραμμα τομής Euler για τις κλάσεις προβλημάτων παρακάτω, θα μας δώσει μια περισσότερο ξεκάθαρη απεικόνιση των διαφορών κάθε συνόλου προβλημάτων, θεωρώντας βέβαια πως NP ≠ P.



Εικόνα 1. Διάγραμμα τομής Euler μεταξύ των κλάσεων προβλημάτων

1.3 NP- Προβλήματα Βελτιστοποίησης

Στα μαθηματικά, στην επιστήμη των υπολογιστών και στα οικονομικά, ένα πρόβλημα βελτιστοποίησης, αποτελεί ένα πρόβλημα εύρεσης μιας βέλτιστης λύσης από όλες τις διαθέσιμες εφικτές λύσεις.

Τα προβλήματα βελτιστοποίησης είναι προβλήματα επιλογής της «καλύτερης» λύσης από ένα πεπερασμένο σύνολο. Ένα NP-πρόβλημα βελτιστοποίησης, Π, αποτελείται από:

- Ένα σύνολο εγκύρων στιγμιότυπων, $D\prod$,αποφασίσιμο σε πολυωνυμικό χρόνο. Όλοι οι αριθμοί στην είσοδο είναι φυσικοί αριθμοί. Το μέγεθος ενός στιγμιότυπου $I \in D\prod$, που συμβολίζεται με $|I|$, ορίζεται ως ο αριθμός των δυαδικών ψηφίων που απαιτούνται για την εγγραφή του I με την υπόθεση ότι όλοι οι αριθμοί που εμφανίζονται στο στιγμιότυπο, γράφονται σε δυαδικό σύστημα.
- Κάθε στιγμιότυπο $I \in D_\Pi$ έχει ένα σύνολο εφικτών λύσεων, $S\prod (I)$. Απαιτείται να ισχύει $S\prod (I) \neq \emptyset$ και για κάθε λύση $s \in S\prod (I)$ είναι πολυωνυμικά φραγμένη ως προς στο $|I|$. Επιπλέον υπάρχει ένας αλγόριθμος πολυωνιμικού χρόνου όπου όταν δίνεται ένα ζεύγος (I,s), απαντά αν $s \in S\prod (I)$.
- Υπάρχει αντικειμενική συνάρτηση υπολογίσιμη σε πολυωνιμικό χρόνο, $obj\prod$, που εκχωρεί έναν μη αρνητικό ρητό αριθμό σε κάθε ζεύγος (I,s), όπου το I είναι ένα στιγμιότυπο και είναι μια εφικτή λύση για το I. Στην αντικειμενική συνάρτηση δίνεται συχνά μια φυσική ερμηνεία, όπως κόστος, μήκος, βάρος κ.λπ.
- Τέλος το Π ότι είναι είτε πρόβλημα ελαχιστοποίησης είτε πρόβλημα μεγιστοποίησης.

Μια βέλτιστη λύση για ένα στιγμιότυπο ενός προβλήματος ελαχιστοποίησης (μεγιστοποίησης) είναι μια εφικτή λύση που επιτυγχάνει τη μικρότερη (μεγαλύτερη) τιμή της αντικειμενική συνάρτησης. Το $OPT\prod (I)$ υποδηλώνει την τιμή της αντικειμενικής συνάρτησης μιας βέλτιστης λύσης στο στιγμιότυπο I. Για λόγους συντομίας θα ονοματιστεί το $OPT\prod (I)$ σε OPT. Με κάθε πρόβλημα βελτιστοποίησης NP, μπορεί κανείς φυσικά να συσχετίσει ένα πρόβλημα απόφασης δίνοντας ένα όριο στη βέλτιστη λύση. Έτσι, η έκδοση απόφασης (decision version) του προβλήματος βελτιστοποίησης NP Π αποτελείται από ζεύγη (I, B), όπου το I είναι ένα στιγμιότυπο του Π και το Β είναι ένας ρητός αριθμός. Εάν το π είναι ένα πρόβλημα ελαχιστοποίησης (μεγιστοποίησης), τότε η απάντηση στην έκδοση απόφασης είναι "ναι" εάν υπάρχει μια εφικτή λύση στο I του κόστους $\leq B (\geq B)$. Αν είναι έτσι, θα οριστεί το (I, B) σαν ένα στιγμιότυπο "ναι" αλλιώς θα οριστεί ως ένα στιγμιότυπο “όχι”. Σαφώς, ένας αλγόριθμος πολυωνυμικού χρόνου για το Π, μπορεί να βοηθήσει στην επίλυση της έκδοσης απόφασης, υπολογίζοντας το κόστος μιας βέλτιστης λύσης και συγκρίνοντας το με το Β. Αντίθετα, η δυσκολία για ένα NP-πρόβλημα βελτιστοποίησης καθορίζεται δείχνοντας ότι η έκδοση απόφασης του είναι NP-hard.

1.4 *Προσεγγιστικοί αλγόριθμοι*

Τα NP-Δύσκολα προβλήματα βελτιστοποίησης παρουσιάζουν ένα πλούσιο σύνολο δυνατοτήτων, από το σημείο να μπορούν να επιτρέπουν την προσέγγιση σε οποιονδήποτε απαιτούμενο βαθμό, μέχρι ουσιαστικά να μην είναι δυνατή καθόλου η προσέγγιση της λύσης.

Οι προσεγγιστικοί αλγόριθμοι όπως προαναφέραμε, αφορούν κυρίως σε προβλήματα βελτιστοποίησης (ελαχιστοποίησης ή μεγιστοποίησης). Σε κάθε στιγμιότυπο αντιστοιχούν εφικτές (έγκυρες/αποδεκτές) λύσεις, που η κάθε μία, έχει μια τιμή μέσω μιας αντικειμενικής συνάρτησης (objective function), που συνήθως εκφράζει: κόστος, μήκος, βάρος, κέρδος, κ.λπ. Χρησιμοποιούμε επομένως έναν προσεγγιστικό αλγόριθμο, όταν αναζητούμε τη βέλτιστη λύση, μια εφικτή δηλαδή λύση με βέλτιστη τιμή.

Όπως τονίσαμε παραπάνω, υπάρχουν δύο ειδών προβλήματα, ελαχιστοποίησης (minimization) και μεγιστοποίησης (maximization). Στα προβλήματα ελαχιστοποίησης ανήκουν αναφορικά: ο χρωματισμός γραφήματος (Edge coloring), το πρόβλημα κοντινότερου μονοπατιού (Shortest Paths), το κυρίαρχο σύνολο

(Dominating Set), το Πρόβλημα του Πλανόδιου Πωλητή (TSP), και η Κάλυψη Κορυφών (Vertex Cover), ενώ στα προβλήματα μεγιστοποίησης έχουμε αντίστοιχα το Maximum Matching, Independent Set και Clique.

1.5 *Όριο Απόδοσης*

Όταν σχεδιάζουμε έναν αλγόριθμο προσέγγισης για ένα NP-Δύσκολο πρόβλημα βελτιστοποίησης, αντιμετωπίζουμε το εξής δίλημμα. Για να καθιερωθεί η εγγύηση προσέγγισης, το κόστος της λύσης που παράγεται από τον αλγόριθμο πρέπει να συγκριθεί με το κόστος μιας βέλτιστης λύσης. Ωστόσο, για τέτοια προβλήματα, όχι μόνο είναι NP-Δύσκολο να βρεθεί μια βέλτιστη λύση, αλλά είναι επίσης NP-Δύσκολο να υπολογιστεί και το κόστος μιας βέλτιστης λύσης. Επομένως, πώς μπορούμε να καθορίσουμε την εγγύηση μιας προσέγγισης; Η απάντηση σε αυτό το ερώτημα, δεν είναι τόσο απλή, καθώς αποτελεί βασικό πυλώνα για το σχεδιασμό αλγορίθμων προσέγγισης, και ο τρόπος ποικίλει ανά περίπτωση.

Βέβαια, μπορεί να μη γνωρίζουμε το κόστος της βέλτιστης λύσης, αλλά κάποιο ανώτατο φράγμα για αυτό. Στην παρούσα διπλωματική θα αναλύσουμε έναν αλγόριθμο που δημιούργησαν οι Dorit S. Hochbaum, Takao Nishizeki και David B. Shmous, ο οποίος ποτέ δεν χρησιμοποιεί περισσότερα από $[\frac{9}{8}\chi + \frac{3}{4}]$ χρώματα. Επιπλέον αν $\chi \geq [\frac{9}{8}\Delta + \frac{3}{4}]$ τότε ο αλγόριθμος χρωματίζει βέλτιστα το γράφημα σε πολυωνυμικό χρόνο. Επιπρόσθετα ο αλγόριθμος ποτέ δεν χρησιμοποιεί παραπάνω από $\frac{4}{3}\chi$ χρώματα και εκτελείται σε χρόνο $O(|E|(|V| + \Delta))$ όπου E το πλήθος των ακμών και V το σύνολο των κορυφών.

1.6 *Γνωστοί Αλγόριθμοι για το Edge-Coloring*

Ο χρωματισμός των ακμών στα γραφήματα είναι ένα καλά μελετημένο πρόβλημα στην επιστήμη των υπολογιστών και τη συνδυαστική με αρκετούς γνωστούς αλγορίθμους που χρησιμοποιούνται για να βρεθεί ένα βέλτιστο ή σχεδόν βέλτιστο χρωματισμό. Ακολουθούν μερικοί από τους αξιοσημείωτους αλγορίθμους:

- Αλγόριθμος Άπληστου Χρωματισμού (Greedy Coloring Algorithm): Αυτή είναι μία απλή προσέγγιση όπου επιλέγετε διαδοχικά μία ακμή και της αναθέτει το μικρότερο διαθέσιμο χρώμα που δεν έχει χρησιμοποιηθεί από τις γειτονικές της ακμές. Παρόλο που είναι απλός και γρήγορος, ο άπληστος χρωματισμός δεν εγγυάται τον ελάχιστο αριθμό χρωμάτων για όλα τα γραφήματα.
- Θεώρημα και Αλγόριθμοι του Vizing: Το θεώρημα του Vizing δηλώνει ότι κάθε απλό, μη κατευθυνόμενο γράφημα μπορεί να χρωματιστεί στις ακμές του με Δ ή Δ+1 χρώματα, όπου Δ είναι ο μέγιστος βαθμός του γραφήματος. Βασιζόμενο σε αυτό το θεώρημα, ο αλγόριθμος του Vizing χρησιμοποιείται για να βρει έναν τέτοιο χρωματισμό, αν και η πραγματική υλοποίηση μπορεί να είναι περίπλοκη και να απαιτεί πολλούς υπολογιστικούς πόρους.
- Μέθοδοι T-Coloring και T-Join: Χρησιμοποιούνται για χρωματισμό ακμών σε πολυγραφήματα, είναι περίπλοκοι και συχνά απαιτούν τη δημιουργία βοηθητικών γραφημάτων. Αλγόριθμοι όπως οι Hopcroft-Karp ή Ford-Fulkerson.

Κάθε ένας από αυτούς τους αλγορίθμους έχει διαφορετικές προτεραιότητες σε όρους πολυπλοκότητας υλοποίησης, αποδοτικότητας χρόνου εκτέλεσης, και ποιότητας της λύσης (δηλαδή τον αριθμό των χρωμάτων που χρησιμοποιείται). Η επιλογή αλγορίθμου συχνά εξαρτάται από τις συγκεκριμένες ιδιότητες του γραφήματος με το οποίο εργάζεστε (π.χ. γενικά γραφήματα, διμερή γραφήματα, πολυγραφήματα) και τις απαιτήσεις της εφαρμογής σας (π.χ. ακριβής λύση έναντι ταχύτητας).

1.7 *Στόχος και Υλοποίηση*

Ο στόχος που κληθήκαμε να επιτύχουμε είναι, να υλοποιήσουμε έναν αλγόριθμο που θα μπορεί να φέρει εις πέρας τον χρωματισμό πολυγραφήματων έχοντας ως πάνω όριο πλήθος χρωμάτων ($\frac{9}{8}\chi + \frac{3}{4}$) το οποίο είναι μικρότερο από οποιονδήποτε άλλο υφιστάμενο αλγόριθμο. Η μεγάλη δυσκολία στην κατασκευή αυτού του αλγορίθμου κρύβεται πίσω στις πολλές περιπτώσεις που πρέπει να κατασκευαστούν ώστε να μπορέσει να επιτευχθεί το άνω όριο που έχουμε καθώς και στον καινοτόμο σχεδιασμό διαφόρων μεθόδων που θα αναλύσουμε μετέπειτα. Το μεγαλύτερο βάρος του αλγόριθμου βασίζεται στην “μέθοδο” recolor η οποία θα είναι υπεύθυνη για τον επαναχρωματισμό των ακμών. Ο τρόπος με τον οποίο χειρίζεται ο αλγόριθμος τον επαναχρωματισμό των ακμών είναι και το κλειδί ώστε να επιτευχθεί και το άνω όριο. Ο επαναχρωματισμός χωρίζετε σε πέντε (5) βασικές περιπτώσεις τις οποίες και θα αναλύσουμε εκτενέστερα στην συνέχεια. Προφανώς μια ακμή μπορεί να χρωματιστεί παραπάνω από μία(1) φορά ώστε να επιτευχθεί το επιθυμητό αποτέλεσμα.

Αφού εκτελεστεί ο αλγόριθμος με επιτυχία θα υπάρχει και μια οπτικοποίηση του γραφήματος, μέσω της βιβλιοθήκης network.py της Python όπου κανένα χρώμα δεν θα πρέπει να υπάρχει σε παραπάνω από μία (1) ακμή που προσπίπτει στην ίδια κορυφή και το συνολικό πλήθος των χρωμάτων δεν θα πρέπει να είναι μεγαλύτερο από $[\frac{9}{8}\chi + \frac{3}{4}]$, όπου χ το πλήθος των χρωμάτων.

Κεφάλαιο 2. Περιγραφή Αλγορίθμου Χρωματισμού Γραφήματος

2.1 *Εισαγωγή*

Στο κεφάλαιο αυτό περιγράψουμε τον αλγόριθμο που παρουσιάζεται στην εργασία από τους Dorit S. Hochbaum, Takao Nishizeki και David B. Shmous. Πριν ξεκινήσουμε όμως θα χρειαστεί να ορίσουμε κάποιους συμβολισμούς.

2.1.1 *Απαραίτητοι ορισμοί*

Το **Δ(G)** συμβολίζει τον μέγιστο βαθμό μια κορυφής δηλαδή το μέγιστο πλήθος ακμών που προσπίπτουν σε μία κορυφή. Τα **n** και **m** συμβολίζουν το πλήθος των κορυφών **(|V|)** και ακμών **(|E|)** αντίστοιχα. Το **c** λέμε ότι είναι ένα χρώμα που λείπει από μια κορυφή **u** αν δεν έχει ανατεθεί σε καμία ακμή που προσπίπτει στο **u.** Το σύνολο όλων τον χρωμάτων που λείπουν από το **u** θα συμβολίζεται με **M(u)**. Μία ακμή θα λέγεται **c-ακμή** όταν χρωματίζεται με το **c**. **ab-υπογράφημα (subgraph)** ορίζεται ένα υπογράφημα του **G** (αρχικό μας γράφημα) όταν περιέχει μόνο τις ακμές που είναι χρωματισμένες με a ή b και θα συμβολίζεται **G[a, b]**. Κάθε συνεκτική συνιστώσα του G[a, b] είναι είτε <u>κύκλος</u> είτε <u>μονοπάτι</u> με όλες τις ακμές του να είναι χρωματισμένες με a, b **εναλλάξ** και θα το ονομάζουμε **ab-μονοπάτι (path)** και **ab-κύκλος (cycle)**. Αν το a ανήκει στο M(x) δηλαδή στα χρώματα που λείπουν από το x τότε το **Apath(x, a, b)** μας δίνει ένα ab-μονοπάτι που ξεκινάει από την κορυφή x. Αν το a ανήκει στο M(x) και το b ανήκει στο M(y) τότε το ab-μονοπάτι μεταξύ των x και y, ***αν υπάρχει,*** λέγεται **ab-κρίσιμο μονοπάτι(ab-critical path)**. Κάθε κρίσιμο μονοπάτι περιέχει περιττό αριθμό κορυφών. Τέλος αν **H** είναι υπογράφημα του G τότε V(H) και E(H) θα είναι το σύνολο των κορυφών και των ακμών του H αντίστοιχα.

2.2 Κάτω Όριο Χρωμάτων

Ας ξεκινήσουμε θέτοντας μερικά απλά κάτω όρια για τον αλγόριθμο μας σχετικά με το <u>ιδανικό</u> πλήθος χρωμάτων (**χ’**) που θα χρησιμοποιηθεί για τον χρωματισμό του γραφήματος.

- ΓΕΓΟΝΟΣ 1 : $\chi'(G) \geq \Delta(G)$
- ΓΕΓΟΝΟΣ 2 : Θα θεωρούμε $\mu_\kappa(G)$ να είναι το μέγιστο πλήθος ακμών σε οποιοδήποτε υπογράφημα με $2\kappa + 1$ κορυφές. Τότε το $\chi'(G) \geq \lceil \mu_\kappa(G) / \kappa \rceil$. Αυτό το πόρισμα προκύπτει από το γεγονός ότι κάθε χρώμα μπορεί να χρησιμοποιηθεί το περισσότερο σε κ ακμές του υπογραφήματος. Για αυτό το λόγο θα ήταν βολικό να ορίσουμε σαν $\boldsymbol{\tau(G) = \max_{\kappa=1,2,3} \lceil \mu_\kappa(G) / \kappa \rceil}$
- Από το προηγούμενο πόρισμα προκύπτει ότι $\chi'(G) \geq \tau(G)$ (FACT 3)
- ΛΗΜΜΑ 1 : Ας υποθέσουμε ότι η ρουτίνα recolor έχει την παρακάτω ιδιότητα: αν ένα καινούργιο χρώμα προστεθεί στο σύνολο των χρωμάτων, ένα υπογράφημα 3, 5 ή 7 κορυφών έχει αναγνωρισθεί όπου το νούμερο των χρωμάτων που είχαν χρησιμοποιηθεί πριν ήταν αυστηρά λιγότερο από το τ(G). Τότε η μέθοδος color(ο χρωματισμός) χρησιμοποιεί το **πολύ** $\boldsymbol{\min(\lfloor (9\chi' + 6)/8 \rfloor, \frac{4}{3}\chi')}$ χρώματα. Επιπλέον αν $\boldsymbol{\chi'} \geq \boldsymbol{\lfloor (9\chi' + 6)/8 \rfloor}$, τότε η μέθοδος color οδηγεί σε βέλτιστο χρωματισμό. Στην συνέχεια θα αναλυθεί η ρουτίνα recolor και ο τρόπος εφαρμογής της στον αλγόριθμο.

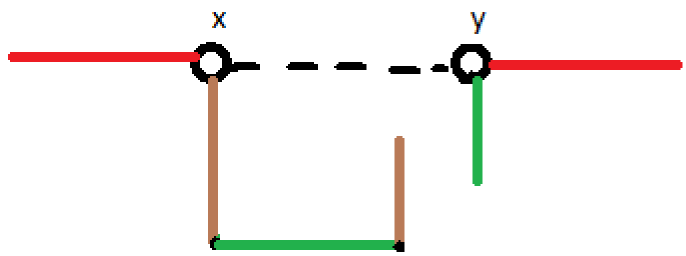
Το τελευταίο bullet μπορεί να αποδεικνύετε ακολούθως: Ας υποθέσουμε ότι το πλήθος των χρωμάτων αν ποτέ αυξηθεί από την ρουτίνα recolor τότε το πλήθος των χρωμάτων που είχαν χρησιμοποιηθεί πριν ήταν αυστηρά μικρότερο του τ(G). Δοθέντος αυτού, ο αλγόριθμος έχει την πολύ αυστηρή ιδιότητα όταν ένα χρώμα προστίθενται (πέρα από τα ήδη ⌊**9Δ +6)/8**⌋) τότε ο χρωματισμός είναι ιδανικός (bullet 3). Έτσι, η μοναδική άλλη περίπτωση είναι ότι ⌊**9Δ +6)/8**⌋ χρώματα έχουν χρησιμοποιηθεί αλλά από το bullet 1, αυτό σημαίνει ότι στην χειρότερη ⌊**9χ' +6)/8**⌋ χρώματα έχουν χρησιμοποιηθεί. Επιπλέον αν **Δ ≤ 2** τότε χρωματίζουμε βέλτιστα και αν **Δ ≥ 3** τότε ⌊**9Δ +6)/8**⌋ ≤ **4Δ/3** ≤ **4χ'/3.** Επομένως, για να αποδείξουμε την αναφερόμενη επίδοση του αλγορίθμου <u>αρκεί να αποδείξουμε ότι η ρουτίνα recolor πληροί την ιδιότητα του bullet 4</u>.

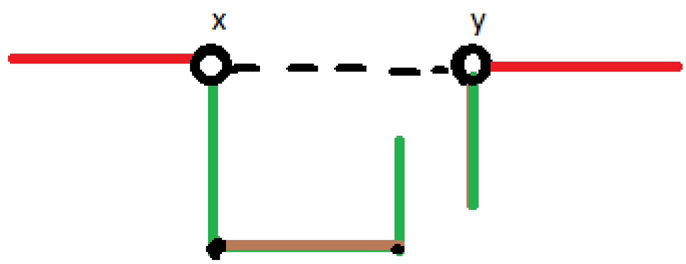
2.3 Recolor

Η ρουτίνα recolor είναι αυτή που θα μας απασχολήσει καθώς είναι και αυτή που κάνει όλη την δουλειά και όλο τον χρωματισμό. Προφανώς υπάρχουν και άλλες μέθοδοι, οι οποίες όμως έχουν κυρίως βοηθητικό χαρακτήρα στην recolor και θα αναλυθούν όταν έρθει η ώρα. Η ρουτίνα recolor χωρίζεται σε 5 κύριες κατηγορίες (cases) που αναλύονται ακολούθως και που στον αλγόριθμο τρέχουν επαναληπτικά η μία κάτω από την άλλη (αν δεν ισχύει το case 1 τσεκάρει το case 2 κτλπ).

2.3.1 Περίπτωση 1

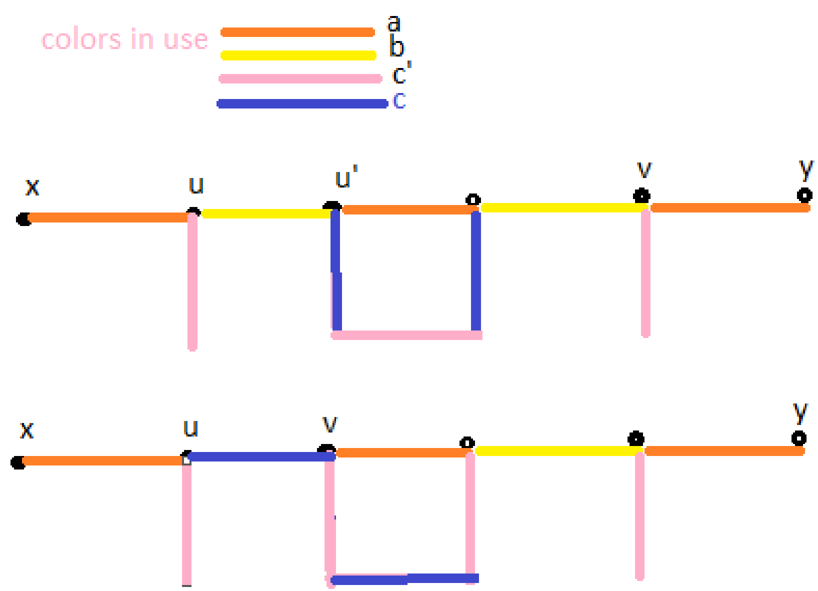
Η πρώτη περίπτωση (case 1) καλύπτει τις περιπτώσεις όπου δύο (2) κορυφές x, y έχουν κάποιο χρώμα (από τα διαθέσιμα) το οποίο λείπει και στις δύο (**common_missing_color**) ή το G' δεν έχει ab-κρίσιμο μονοπάτι. Αρχικά εξετάζει αν δύο (2) κορυφές x, y έχουν κάποιο χρώμα το οποίο λείπει και στις δύο και αν ναι τότε χρωματίζει την ακμή xy με αυτό το χρώμα. Αν δεν υπάρχει τότε ψάχνει για ένα ab-μονοπάτι που ξεκινάει από το x όπου a ∈ M(x)και το b ∈ M(y). Το μονοπάτι αυτό δεν γίνεται να τελειώνει στο y επειδή σε αυτήν την περίπτωση θα είχαμε ab-κρίσιμο μονοπάτι (το οποίο έχουμε υποθέσει ότι δεν υπάρχει). Αφού βρεθεί το μονοπάτι τότε εναλλάσσει τα χρώματα του. Αυτό έχει σαν αποτέλεσμα να ελευθερωθεί το ένα από τα δύο χρώματα και να μπορεί η ακμή xy να χρωματιστεί με αυτό.





2.3.2 Περίπτωση 2

Η δεύτερη περίπτωση (case 2) ελέγχει αν το ab-κρίσιμο μονοπάτι Q περιέχει δύο (2) κορυφές u και v στις οποίες λείπει ένα κοινό χρώμα c. Αρχικά θα ψάξει για δύο (2) διαδοχικές κορυφές στις οποίες λείπει ένα κοινό χρώμα c. Αν βρεθούν τότε χρωματίζει την ακμή uv με c και συνεχίζει στην επόμενη (ξαναμπαίνει στην λούπα για την επόμενη ακμή ξεκινώντας με case1). Αν δεν βρεθούν δύο διαδοχικές κορυφές τότε ψάχνει για δύο κορυφές σε όλο το μονοπάτι στις οποίες λείπει ένα κοινό χρώμα c και σε όλες τις άλλες κορυφές θα υπάρχει. Χρησιμοποιώντας την αμέσως επόμενη κορυφή από την u και το missing_color της (έστω c') ψάχνει ένα μονοπάτι που ξεκινάει από την u και είναι c'c (αντί για ab που έχουμε δει ως τώρα) και εναλλάσσει τα χρώματα των ακμών αυτού του μονοπατιού. Αν το μονοπάτι τελειώνει στην κορυφή u τότε θέτει u=u' αλλιώς θέτει v= u'. Έτσι έχει καταφέρει να φέρει τις δύο κορυφές u,v πιο κοντά κατά τουλάχιστον μία ακμή και δημιουργώντας ένα καινούργιο κοινό χρώμα που λείπει για αυτές. Έτσι έχει καταφέρει να μην υπάρχει πλέον ένα ab-κρίσιμο μονοπάτι και καλεί την περίπτωση 1.



2.3.3 Περίπτωση 3

ΛΗΜΜΑ : 2. Πριν συνεχίσουμε στην τρίτη (3) περίπτωση ας υποθέσουμε ότι **Δ(G) ≥ 3** και ότι όλες οι κορυφές του G' εκτός της xy είναι χρωματισμένες με τουλάχιστον **⌊9Δ +6)/8⌋** χρώματα. Έστω S ένα υποσύνολο κορυφών τέτοιο ώστε δύο κορυφές να μην έχουν κοινό χρώμα που λείπει. Τότε α) Αν το x, y ∈ S τότε το |S| **≤** 8

β) Αν το x ∈ S τότε το |S| ≤ 16

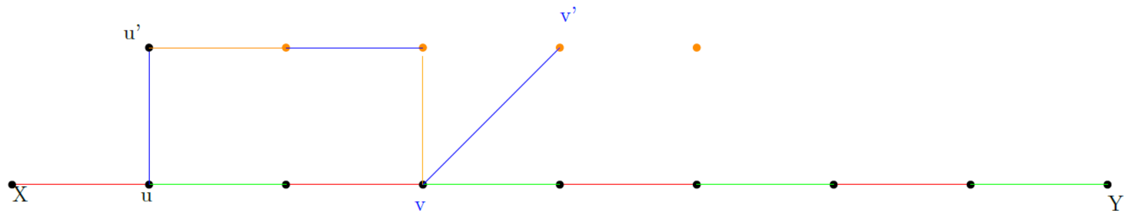
Συνοψίζοντας αυτό σημαίνει ότι αν ένα κρίσιμο μονοπάτι Q περιέχει εννέα (9) ή παραπάνω κορυφές, τότε το Q περιέχει δύο (2) κορυφές με κοινό χρώμα που λείπει μεταξύ των πρώτων 17 κορυφών και έτσι προκύπτει η περίπτωση 2. **Για αυτό αν η περίπτωση 1 ή η περίπτωση 2 δεν προκύψουν τότε το μονοπάτι θα περιέχει είτε εφτά (7) είτε πέντε (5) είτε τρεις (3) κορυφές.**

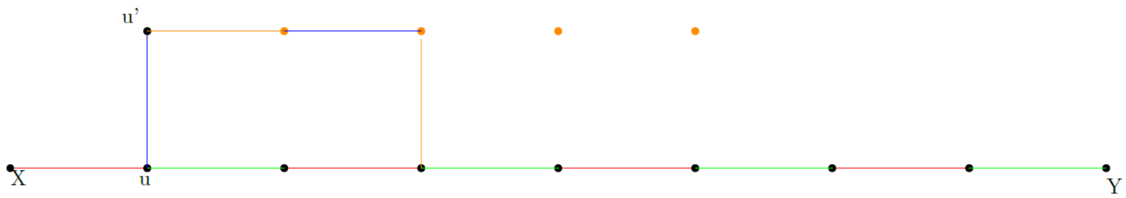
Πρωτού συνεχίσουμε στην ανάλυση της περίπτωσης τρία (3) θα χρειαστεί να κατασκευάσουμε πρώτα μερικές βοηθητικές υπορουτίνες. Η πρώτη υπορουτίνα που θα χρειαστούμε είναι η **twopath**.

Η twopath παίρνει δύο κρίσιμα μονοπάτια Q και R έστω ab και fg μονοπάτια αντίστοιχα όπου a ∈ M(x), b ∈ M(y), g ∈ M(x), f ∈ M(y) και a ≠ f ή b ≠ f με x άκρο του Q και y άκρο του R. Αν το a δεν είναι διάφορο του f τότε εναλλάσσουμε τους ρόλους των x y και ξαναρχίζουμε την twopath. Αρχικά θα επιλεχθούν οι δύο (2) κορυφές αμέσως μετά την x για το κάθε μονοπάτι, έστω u' για το Q και v' για το R και c' c'' τα missing color για τις αντίστοιχες κορυφές. Αφού ψάξει για δύο μονοπάτια, ένα που θα ξεκινάει από την u' και θα έχει χρώματα c'c και ένα που θα ξεκινάει από την v' με χρώματα c''c, θα τσεκάρει αν u ≠ u' <u>και</u> το πρώτο μονοπάτι δεν τελειώνει στην κορυφή u **ή** v ≠ v' <u>και</u> το δεύτερο δεν τελειώνει στην κορυφή v. Αν καλύπτονται οι προαναφερθέν συνθήκες τότε εναλλάσσει τα χρώματα του μονοπατιού που ξεκινάει από την u' με χρώματα c'c και καλεί την περίπτωση δύο (case2). Αν δεν καλύπτονται οι συνθήκες τότε ανάλογα με το αν το u ≠ u' τότε εναλλάσσει τα χρώματα του μονοπατιού που ξεκινάει από την u' με χρώματα c'c αλλιώς αν το v ≠ v' τότε εναλλάσσει τα χρώματα του μονοπατιού που ξεκινάει από την v' με χρώματα c''c. Έτσι το c ∈ M(u') ∩ M(u). Τώρα ψάχνει ένα καινούργιο μονοπάτι που ξεκινάει από την κορυφή x με χρώματα a c. Αν αυτό τελειώνει στην κορυφή v' τότε εναλλάσσει τα χρώματα του μονοπατιού, χρωματίζει την ακμή xu' του Q με το χρώμα c και την άχρωμη ακμή xy με το χρώμα b. Αν δεν τελειώνει στην κορυφή v' τότε εναλλάει τα χρώματα του μονοπατιού και καλεί την περίπτωση δύο (case2). Αυτό σημαίνει ότι το a, c ≠ f, g, ότι το R είναι ακόμα κρίσιμο μονοπάτι και ότι το R περιέχει δύο (2) κορυφές x και v' με κοινό missing color c.

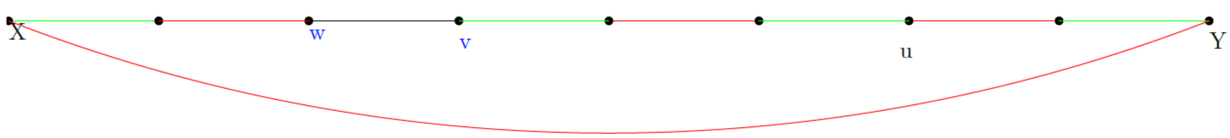
Η επόμενη υπορουτίνα που θα χρειαστούμε είναι η **leave**. Πριν ξεκινήσουμε να αναλύουμε την leave θα χρειαστεί να επισημάνουμε κάποιες ορολογίες. Αρχικά, μία ακμή του G' θα εννοούμε ότι **φεύγει** από το ένα άλλο υπογράφημα αν η μία της κορυφή ανήκει στο υπογράφημα και η άλλη δεν ανήκει. Από το **ΓΕΓΟΝΟΣ 2** αν **q < |E(H|/((|V(H)| - 1)/2)**, τότε **q < τ(G)** και μπορούμε να παρουσιάσουμε ένα νέο χρώμα q+1 για την (x,y). Τότε μπορούμε να υποθέσουμε ότι **q ≥** $\frac{|\boldsymbol{E}(\boldsymbol{H})|}{(|\boldsymbol{V}(\boldsymbol{H})|-\mathbf{1})/\mathbf{2}}$. Αφού το |V(H)| είναι περιττός, κάθε από τα q χρώματα ανατίθεται το πολύ σε (|V(H)| - 1)/2 ακμές του υπογραφήματος. Αφού το υπογράφημα περιέχει μία άχρωμη ακμή (x, y) τότε υπάρχει χρώμα c που ανατίθεται το πολύ σε (|V(H)| - 1)/2 - 1 ακμές. Έτσι το υπογράφημα περιέχει τρεις (3) ή περισσότερες κορυφές όπου δεν περιέχουν c-ακμές και πρέπει να σε το πολύ μία από τις κορυφές να λείπει το χρώμα c στο G'. Έτσι το υπογράφημα πρέπει να περιέχει τουλάχιστον δύο (2) c-ακμές που να φεύγουν από το υπογράφημα και αν το c δεν λείπει από καμιά κορυφή τότε υπάρχουν τουλάχιστον τρεις (3) c-ακμές που φεύγουν από το υπογράφημα. Σε αυτές τις περιπτώσεις η υπορουτίνα leave επεξεργάζεται τον χρωματισμό του G' έτσι ώστε μία ακμή με χρώμα που λείπει από το x ή το y θα φεύγει από το υπογράφημα.

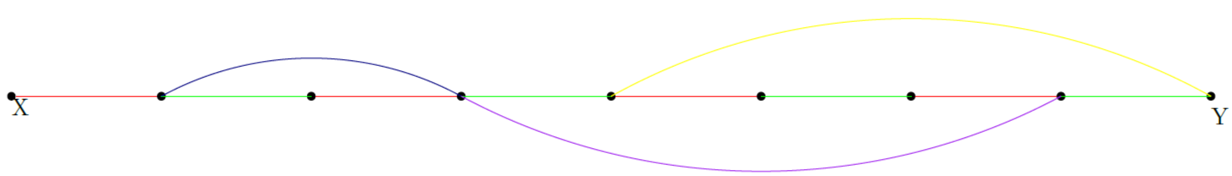
Αρχικά η leave θα χρειαστεί να ελέγξει αν **q ≥** $\frac{|\boldsymbol{E}(\boldsymbol{H})|}{(|\boldsymbol{V}(\boldsymbol{H})|-\mathbf{1})/\mathbf{2}}$. Ο επόμενος έλεγχος που θα κάνει είναι αν το G' δεν έχει καμιά ακμή να φεύγει από το υπογράφημα H όπου είναι χρωματισμένη με missing color μιας κορυφής του H. Αν ισχύουν τα παραπάνω τότε έστω (u,u') μια c-ακμή που φεύγει από το H όπου u ∈ H, u' ∉ H , f missing color του u διαφορετικό από a,b,c και (v, v') η τελευταία c-ακμή, από μονοπάτι που ξεκινάει από την κορυφή u και έχει χρώματα f, c τέτοια ώστε το v ∈ H και το v' ∉ H. Τότε αν u = v εναλλάσσει τα χρώματα του μονοπατιού και έτσι το c ∈ M(u) και υπάρχει μία c-ακμή που φεύγει από το H. Αν το u ≠ v τότε έστω g missing color του v διαφορετικό από a, b, c. Αν δεν υπάρχει τέτοιο χρώμα τότε είναι πιο εύκολη περίπτωση και μπορούμε να καλέσουμε την περίπτωση δύο (case2) αφού καμιά f,g ακμή δεν φεύγει από το H, τότε το μονοπάτι από το v με χρώματα g, f δεν φεύγει από το H και πρέπει να τελειώνει στο u. Έναλλάζει τα χρώματα των μονοπατιών που ξεκινάνε από την κορυφή v και τα δύο και έχουν χρώματα g, f (f ∈ M(v)) και χρώματα f, c όπου c ∈ M(v) και υπάρχει c-ακμή που φεύγει από το H. Ξαναορίζει πλέον το χρώμα c και έχοντας τις κορυφές u,v ∈ H και u' ∉ H τέτοιο ώστε c ∈ M(v) και μία c-ακμή (u, u') που φεύγει από το H. Αν u ≠ x, y τότε έστω ότι οι κορυφές x, v, u, y εμφανίζονται με αυτήν την σειρά (μπορεί να υπάρχουν και άλλες ενδιάμεσα τους) στο μονοπάτι Q και w μία κορυφή πριν την v. Διαγράφει τώρα το χρώμα από την ακμή (w, v) του Q και εναλλάσσει τα χρώματα του υπομονοπατιού του Q από τις κορυφές x μέχρι w. Χρωματίζει τώρα την ακμή (x, y) με το χρώμα b. Πλέον η ακμή (w, v) δεν έχει χρώμα και το c ∈ M(v) και υπάρχει μία c-ακμή (u, u') που φεύγει από το H. Ξαναορίζει πλέον την ακμή (w, v) να είναι η ακμή του G' που δεν έχει χρώμα. Μετά από αυτήν την διαδικασία πλέον το χρώμα c λείπει από το καινούργιο x ή y και μία c-ακμή φεύγει από H.



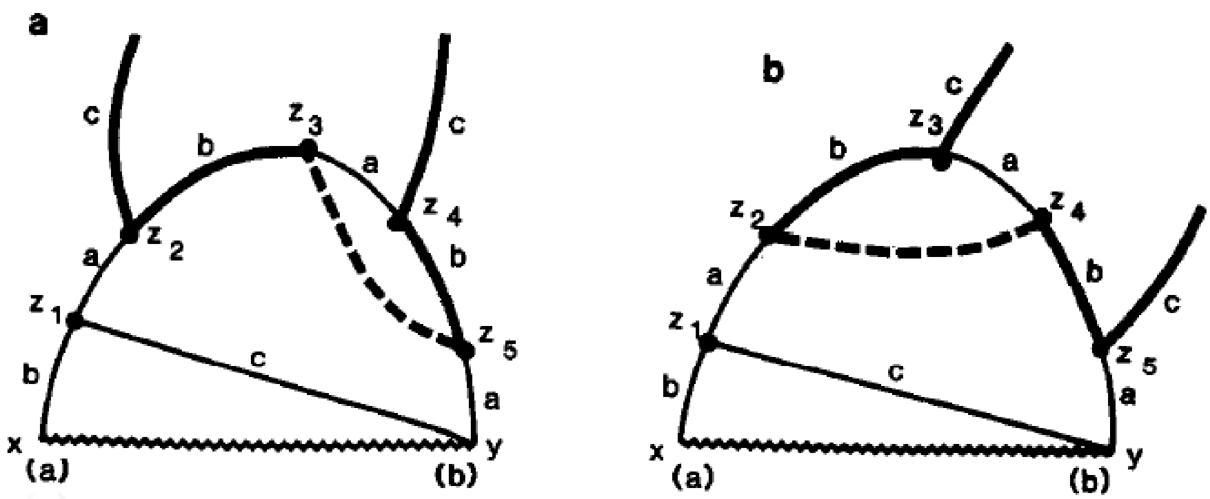


Αφού ορίσαμε τις υπορουτίνες **leave** και **twopath** μπορούμε πλέον να προχωρήσουμε στην **seven (περίπτωση 3),** που είναι η περίπτωση που το μονοπάτι έχει εφτά (7) κορυφές, έχοντας όσα χρειαζόμαστε. Το πρώτο που πρέπει να γίνει στην seven είναι να κατασκευαστεί το υπογράφημα H το οποίο θα περιέχει όλες τις κορυφές του Q καθώς και όσες ακμές υπάρχουν μεταξύ των κορυφών του μονοπατιού (πχ έστω οι κορυφές του μονοπατιού 1,2,3,4,5 με αυτήν την σειρά. Το H θα περιέχει και επιπλέον ακμές όπως (1, 3), (2, 4), (1, 5) κτλπ).

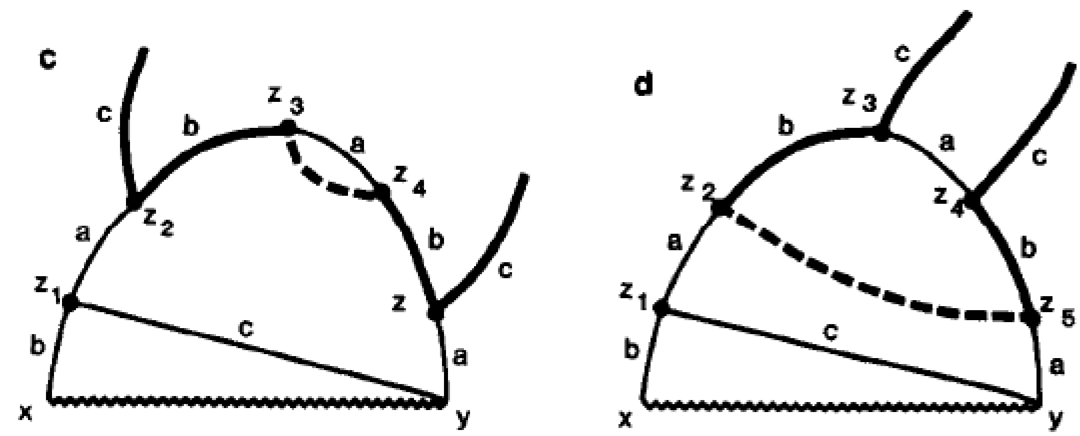




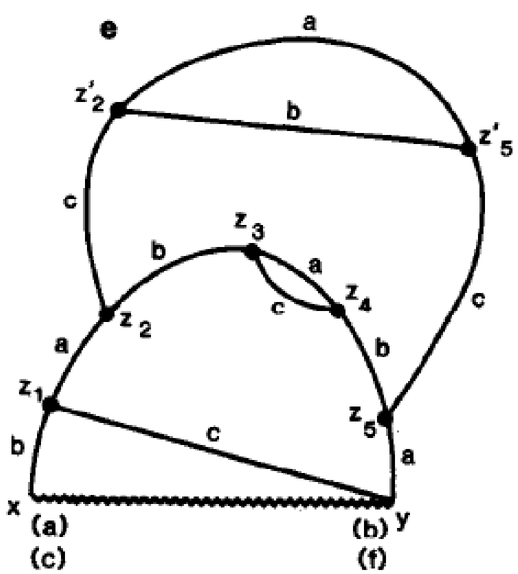
Αφού κατασκευαστεί το H τότε ο πρώτος έλεγχος που θα χρειαστεί να γίνει είναι αν το **$q < |E(H|/((|V(H)| -1)/2)$** και αν ισχύει τότε το πλήθος χρωμάτων αυξάνεται κατά ένα και δίνει το χρώμα αυτό στην ακμή (x, y). Αν δεν ισχύει ο προηγούμενος έλεγχος τότε αυτό σημαίνει ότι υπάρχουν τουλάχιστον δύο ή περισσότερες ακμές που φεύγουν από το H και είναι χρωματισμένες με το ίδιο χρώμα. Το πρώτο πράγμα που κάνει η seven είναι να καλέσει την leave. Αμέσως μετά το κάλεσμα υπάρχει ένα χρώμα $c \in M(x)$ και μία c-ακμή που φεύγει από το H. Το c δεν λείπει από καμιά κορυφή πέρα από το x και τουλάχιστον δύο c-ακμές φεύγουν από το H. Έστω το μονοπάτι Q με κορυφές (x, z1, z2, …, z5, y). Αν δεν υπάρχει κάποιο cb-κρίσιμο μονοπάτι τότε καλεί την περίπτωση ένα (case1) αλλιώς σημαίνει ότι υπάρχει ένα cb-κρίσιμο μονοπάτι έστω R. Αν το R περιέχει δύο κορυφές με κοινό missing color τότε καλεί την περίπτωση δύο (case2) αλλιώς σημαίνει ότι περιέχει τουλάχιστον εφτά (7) κορυφές το R. Αν υπάρχουν δύο κορυφές w, v, με κοινό missing color όπου το w είναι κορυφή στο Q και το v στο R τότε καλεί την twopath. Αλλιώς αφού $|\mathbf{V(Q \cup R)}| \leq \mathbf{8}$ από Lemma 2 το R δεν φεύγει από το H, άρα είναι αδύνατο να προστεθεί μία κορυφή που να φεύγει από το H αφού το χρώμα b είναι κοινό χρώμα και στο Q και στο R. Έτσι το $\mathbf{u \neq x, y, z1}$. Πριν προχωρήσουμε στην επομένη περίπτωση πρέπει να κατασκευαστεί ένα μονοπάτι με διαφορετικό τρόπο από αυτόν που χρησιμοποιούσαμε μέχρι τώρα. Ξεκινώντας από z2 κορυφή (μπορεί να είναι και άλλη) θα ψάξει για το bc μονοπάτι και αφού το βρει θα ψάξει ξεκινώντας πάλι από την z2 για το cb μονοπάτι και θα τα ενώσει αυτά τα δύο σε ένα. Αν το μονοπάτι που προέκυψε είναι διαφορετικό από το R, δεν είναι κύκλος δύο κορυφών και περιέχει ακριβώς μία από τις δύο ακμές (z2, z3) ή (z4, z5) τότε το μονοπάτι που φτιάχτηκε έχει δύο c-ακμές που φεύγουν από το H. Εναλλάσσει τα χρώματα του μονοπατιού και έτσι το ab-κρίσιμο μονοπάτι, αν υπάρχει, έχει εννέα ή παραπάνω κορυφές. Αν δεν υπάρχει το ab-κρίσιμο μονοπάτι τότε καλεί την περίπτωση ένα (case1) αλλιώς την περίπτωση δύο (case2) και σημαίνει ότι το ab-κρίσιμο μονοπάτι έχει δύο ή παραπάνω κορυφές με κοινό missing color. Αν το μονοπάτι δεν πληροί τις προδιαγραφές που αναφέραμε πριν, τότε τσεκάρει υπάρχει ac-μονοπάτι (κατασκευασμένο με τον τρόπο που περιγράψαμε πριν) όπου δεν είναι κύκλος δύο (2) κορυφών και που περιέχει ακριβώς μία (1) από τις τρεις (3) a-ακμές. Αν ισχύουν οι συνθήκες αυτές τότε εναλλάσσει τα χρώματα του μονοπατιού και έτσι το ab-κρίσιμο μονοπάτι, αν υπάρχει, περιέχει εννέα (9) ή παραπάνω κορυφές. Αν δεν υπάρχει τότε καλεί την περίπτωση ένα (case1) αλλιώς καλεί την περίπτωση δύο (case2). Αν το μονοπάτι που κατασκευάστηκε πάλι δεν πληροί τις συνθήκες αυτές, τότε μπορεί να δει κανείς ότι η ακμή (y, z1) πρέπει να χρωματιστεί με το χρώμα c. Έστω S ένα bc-μονοπάτι ή κύκλος που περιέχει τις κορυφές z2, z3, z4, z5. Ανάλογα με τις σειρά που υπάρχουν στο S χωρίζονται σε δύο (2) υποπεριπτώσεις. Στην **περίπτωση 3.1(case3.1)** αν οι κορυφές είναι με την σειρά z2, z3, z5, z4 ή z3, z2, z4, z5 τότε το $c \notin C(z2, z4)$ ή $c \notin C(z3, z5)$. Έτσι δύο c-ακμές φεύγουν από το H είτε στο z2 και στο z4 είτε στο z3 και το z5. Σε αυτήν την περίπτωση εναλλάσσουν τα χρώματα του S και του ac-κύκλου (z1, z2, …, z5, y) και αν δεν υπάρχει cb-κρίσιμο μονοπάτι τότε καλείτε η περίπτωση ένα (case1) αλλιώς υπάρχει cb-κρίσιμο μονοπάτι με τουλάχιστον εννέα κορυφές και καλείτε η περίπτωση δύο (case2).



Στην **περίπτωση 3.2 (case 3.2)** όπου οι κορυφές θα πρέπει να εμφανίζονται είτε με την σειρά z2, z3, z4, z5 ή z3, z2, z5, z4 τότε μπορούμε να δούμε δύο μακρινές c-ακμές (z2, z2') και (z5, z5') όπου φεύγουν από την H. Υποθέτουμε ότι c ∈ (z2, z5). Τότε στο H οι δύο c-ακμές φεύγουν από το z3 και το z4 και το (z1, z2, z5, y) είναι ένα ac-κύκλος. Οπότε το μονοπάτι του G'[a, c] περιέχει την a-ακμή (z3, z4) δεν είναι ούτε κύκλος δύο σημείων ούτε περιέχει τις άλλες δύο(2) a-ακμές του Q. Οπότε έστω (z2, z2') και (z5, z5') οι c-ακμές που φεύγουν από το H. Αν a ∉ C(z2', z5') ή [b ∈ M(z2') ∪ M(z5')] ή [b ∉ C(z2', z5')] τότε εναλλάσσει τα χρώματα στο S και ο καινούργιος χρωματισμός ικανοποιεί ένα από τα παρακάτω. Είτε δεν υπάρχει ab-κρίσιμο μονοπάτι (βέλτιστη περίπτωση), είτε το καινούργιο ab-κρίσιμο μονοπάτι Q' έχει εννέα ή παραπάνω κορυφές (από το a ∉ C(z2', z5')), είτε η κορυφή x έχει κοινό missing color c με τις z2' ή z5' (από το b ∈ M(z2') ∪ M(z5')), είτε το μονοπάτι του G'[a,c] που περιέχει την a-ακμή (z2', z5') δεν είναι ούτε κύκλος δύο (2) σημείων ούτε περιέχει κάποια από τις a-ακμές του καινούργιο ab-μονοπατιού Q' = (x, z1, z2, z2', z5', z5, y) (από το a ∈ C(z2', z5'), b ∈ M(z2') ∪ M(z5')), b ∉ C(z2', z5')) τότε καλεί την recolor.

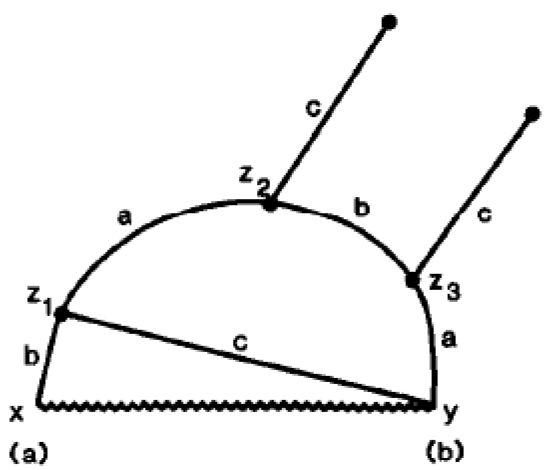


Αλλιώς το a, b ∈ C(z2', z5') και αυτό έχει σαν αποτέλεσμα ο κύκλος δύο σημείων να χρησιμοποιηθεί στην παρακάτω περίπτωση όπου a, c ∈ C(z3, z4). Έστω f ≠ b να είναι οποιαδήποτε missing color του y. Αν δεν υπάρχει af ή cf-κρίσιμο μονοπάτι ή το af ή cf-κρίσιμο μονοπάτι περιέχει δύο κορυφές με κοινό missing color τότε καλούμε την recolor για h, f (h= a ή c). Αλλιώς αν δύο (2) από τα τέσσερα (4) ab- af- cb- cf-κρίσιμα μονοπάτια περιέχουν δύο κορυφές με κοινό missing color τότε καλεί twopath για αυτά τα δύο. Αλλιώς αφού af και cf-κρίσιμα μονοπάτια δεν φεύγουν από το H τότε από το Lemma2, μία f-ακμή ούτε φεύγει από το H στην κορυφή z5 ούτε ενώνει την z5 με το x ή z1. Έτσι μία f-ακμή ενώνει το z5 είτε με το z2 ή z3ή z4. Αν η f-ακμή ενώνει το z5 με το z2 τότε εναλλάσσει τα χρώματα του μονοπατιού G'[c, f] που περιέχει την f-ακμή (z5, z2). Αν δεν υπάρχει af-κρίσιμο μονοπάτι ή το af-κρίσιμο μονοπάτι περιέχει δύο κορυφές με κοινό missing color τότε καλεί την recolor με a,f αλλιώς το ab και το af-κρίσιμο μονοπάτι περιέχουν εννέα (9) ή παραπάνω κορυφές όπου δύο κορυφές έχουν κοινό missing color και καλείτε η twopath για αυτά. Αν η f-ακμή ενώνει την z5 είτε με την z3 ή z4, τότε εναλλάσσει τα χρώματα του μονοπατιού S και το καινούργιο ab-κρίσιμο μονοπάτι (x, z1, z2, z2', z5', z5, y) και το af-κρίσιμο μονοπάτι περιέχουν δύο κορυφές με κοινό missing color και καλείτε η twopath για αυτά.

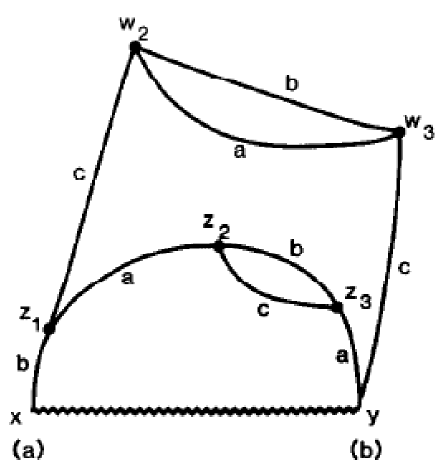


2.3.4 *Περίπτωση 4*

Η **περίπτωση τέσσερα (case 4)** εξετάζει την περίπτωση που το ab-κρίσιμο μονοπάτι Q (x, z1, z2, z3, y) περιέχει πέντε κορυφές και χωρίζεται σε τρεις υποκατηγορίες. Αρχικά όπως και η περίπτωση τρία (case3) ο πρώτος έλεγχος που θα χρειαστεί να γίνει είναι αν το **q < |E(H|/((|V(H)| -1)/2)** και αν ισχύει τότε το πλήθος χρωμάτων αυξάνεται κατά ένα και δίνει το χρώμα αυτό στην ακμή (x, y). Αν δεν ισχύει ο προηγούμενος έλεγχος τότε αυτό σημαίνει ότι υπάρχουν τουλάχιστον δύο ή περισσότερες ακμές που φεύγουν από το H και είναι χρωματισμένες με το ίδιο χρώμα. Το πρώτο πράγμα που κάνει η five είναι να καλέσει την leave. Αμέσως μετά το κάλεσμα υπάρχει ένα χρώμα c ∈ M(x) και μία c-ακμή που φεύγει από το H.



Αφού το c δεν λείπει σε καμία από τις τέσσερεις κορυφές z1, z2, z3 και y τότε υπάρχουν δύο ή τέσσερεις c-ακμές που φεύγουν από την H. Αν δεν υπάρχει cb-κρίσιμο μονοπάτι τότε καλείτε η περίπτωση ένα (case1). Αν υπάρχει cb-κρίσιμο μονοπάτι έστω R τότε αν το R έχει δύο κορυφές με κοινό missing color τότε καλείτε η περίπτωση δύο αλλιώς το R έχει το πολύ εφτά (7) κορυφές. Αν το R έχει εφτά (7) τότε καλείτε η περίπτωση τρία (case3) αλλιώς αν το R έχει τρεις (3) κορυφές (R = x, z1, y) δύο c-ακμές φεύγουν από το H στο z2 και στο z3 για c ∈ M(x) όπως δείχνει και η εικόνα από πάνω. Τότε εναλλάσσουν τα χρώματα του bc-μονοπατιού ή κύκλου που περιέχει τις z2 και z3 κορυφές και καλείτε η recolor. Το νέο ab-κρίσιμο μονοπάτι, αν υπάρχει, έχει τουλάχιστον εφτά (7) κορυφές. Αν το R έχει πέντε (5) κορυφές τότε αν υπάρχουν δύο κορυφές με κοινό missing color στο σύνολο V(Q ∪ R) τότε καλείτε η twopath αλλιώς έστω μονοπάτι R (x, z1, z2, z3, y) και w2, w3 ∉ H και (z1, w2) και (y, w3) δύο c-ακμές που φεύγουν από το H. Έστω T υπογράφημα του G’ με την ίδια λογική που φτιάχτηκε και το H. Το T περιέχει εφτά (7) κορυφές. Αν δύο c-ακμές φεύγουν από το H στην z2 και στην z3 τότε εναλλάσσει τα χρώματα του bc-μονοπατιού ή κύκλου που περιέχουν την z2 και z3 και καλεί την recolor (το ab-κρίσιμο μονοπάτι ή κύκλος περιέχει εφτά ή παραπάνω ακμές). Αν μία a-ακμή ενώνει το z2 με το z3 τότε αν a ∉ C(w2, w3) δύο a-ακμές φεύγουν από το T στην w2 και στην w3. Εναλλάσσει τα χρώματα ab-μονοπατιού ή κύκλου που περιέχει τις w2 και w3 και καλεί την recolor (το bc-κρίσιμο μονοπάτι ή κύκλος περιέχει εφτά ή παραπάνω ακμές).



Αν μία a-ακμή ενώνει τις κορυφές w2 και w3 τότε τσεκάρει αν **q < |E(T|/((|V(T)| -1)/2) (q < τ(G))** και αν ισχύει τότε το πλήθος χρωμάτων αυξάνεται κατά ένα και δίνει το χρώμα αυτό στην ακμή (x, y). Αν δεν ισχύει ο προηγούμενος έλεγχος τότε αυτό σημαίνει ότι το G' έχει δύο ακμές που φεύγουν από το T και είναι χρωματισμένες με το ίδιο χρώμα και χωρίζεται σε πέντε (5) υποπεριπτώσεις (case 4.1 μέχρι case 4.5). Στην **περίπτωση 4.1 (case 4.1)** αν η f-ακμή φεύγει από το T για f ∈ M(y) τότε αν δεν υπάρχει af ή cf-κρίσιμο μονοπάτι ή το af ή cf-κρίσιμο μονοπάτι έχει δύο κορυφές με κοινό missing color ή το af ή cf-κρίσιμο μονοπάτι περιέχει εφτά κορυφές τότε καλείτε η recolor για τα h,f. Αλλιώς αν δύο (2) από τα τέσσερα (4) ab-, cb-, af- και cf-κρίσιμα μονοπάτια έχουν δύο κορυφές με κοινό missing color τότε καλείτε την twopath αλλιώς το af και cf-κρίσιμα μονοπάτια έχουν το πολύ πέντε κορυφές και δεν φεύγουν από το T. Έτσι καμία f-ακμή δεν φεύγει από το T από τις κορυφές x, z3 ή w3. Άρα δύο f-ακμές φεύγουν από το T στις κορυφές z1, z2 ή w2 και το af-κρίσιμο μονοπάτι δεν περιέχει την a-ακμή (z1, z2) αλλιώς το af-κρίσιμο μονοπάτι θα έφευγε από το T. Εναλλάσσει τότε τα χρώματα του af-μονοπατιού ή κύκλου που περιέχει τα z1 και z2. Έτσι το ab-κρίσιμο μονοπάτι έχει εφτά ή παραπάνω κορυφές και καλείτε η recolor.

Στην **περίπτωση 4.2 (case 4.2)** αφορά όταν υπάρχει μία f-ακμή φεύγει από το T με f ∈ M(z1) τότε διαγράφει το χρώμα b από την ακμή (x, z1) και χρωματίζει την ακμή (x, y) με το b. Πλέον η ακμή (x, z1) είναι η νέα άχρωμη ακμή έστω (x', y'). Τα a, c ∈ M(x'), b ∈ M (y') και μία f-ακμή φεύγει από το T για f ∈ M(y') Έτσι πλέον η περίπτωση 4.1 είναι εφαρμόσιμη τώρα, και καλείτε η recolor για το x', y'.

Η **περίπτωση 4.3 (case 4.3)** αφορά όταν υπάρχει μία f-ακμή φεύγει από το T με f ∈ M(x) τότε αν δεν υπάρχει fb-κρίσιμο μονοπάτι ή το fb-κρίσιμο μονοπάτι έχει δύο κορυφές με κοινό missing color ή το fb-κρίσιμο μονοπάτι περιέχει εφτά (7) κορυφές τότε καλείτε η recolor για τα f, b. Αν δεν ισχύει καμία από τις παραπάνω περιπτώσεις τότε υπάρχει fb-κρίσιμο μονοπάτι με πέντε (5) κορυφές. Αν δύο (2) από τα τρία (3) ab-, cb- και bfκρίσιμα μονοπάτια έχουν δύο κορυφές με κοινό missing color τότε καλείτε η twopath, αλλιώς σημαίνει ότι δεν υπάρχει fb-κρίσιμο μονοπάτι που να φεύγει από το T στις κορυφές y ή z1. Έτσι οι δύο f-ακμές φεύγουν είτε από τις κορυφές z2, z3, w2 ή w3. Ας υποθέσουμε ότι η μία ακμή θα φεύγει είτε από την κορυφή z2 ή την z3. Τότε εναλλάσσει τα χρώματα του bf-μονοπατιού που περιέχει την f-ακμή που φεύγει από το T. Με αυτό επιτεύχθηκε το ab-μονοπάτι, αν υπάρχει, να περιέχει εφτά (7) ή παραπάνω κορυφές και καλείτε η recolor.

Η **περίπτωση 4.4 (case 4.4)** αφορά όταν υπάρχει μία f-ακμή φεύγει από το T με το f να είναι missing color μιας εκ των κορυφών z2, z3, w2 ή w3. Ας υποθέσουμε ότι το f ∈ M(w3) τότε αν δεν υπάρχει fb-κρίσιμο μονοπάτι ή το fb-κρίσιμο μονοπάτι έχει δύο κορυφές με κοινό missing color ή το fb-κρίσιμο μονοπάτι περιέχει εφτά (7) κορυφές τότε καλείτε η recolor για τα f, b. Αν και είναι σαφές ότι το z3 και το w3 είναι εντελώς ισοδύναμα, δεν είναι τόσο εύκολο να δούμε ότι το w2 δεν είναι θεμελιωδώς διαφορετικό από το w3. Ωστόσο, Αυτό μπορεί να επιτευχθεί με αποχρωματισμό της ακμής b (x, z1) και χρωματισμού (x, y) με b. Αυτό ανταλλάσσει τους ρόλους των w3 και w2. Έτσι αν το μονοπάτι (έστω S) που ξεκινάει από την κορυφή y με χρώματα b, f δεν τελειώνει στην w3 ή το S περιέχει δύο κορυφές με κοινό missing color ή το S έχει ακριβώς εφτά (7) κορυφές ή το S και το R έχουν δύο κορυφές με κοινό missing color τότε αφαιρεί το χρώμα c από την ακμή (y, w3) και χρωματίζει με c την ακμή (x, y) και καλείτε η recolor για τα (y, w3) με b, f. Αλλιώς, αν μία f-ακμή φεύγει από το T στην κορυφή w2 ή y τότε |V(T ∪ S)| ≥ 9 και το T ∪ S περιέχει δύο κορυφές u και v με κοινό missing color το g, όπου u ανήκει στο S − T και το v ανήκει στο T − S. Το v = z2 ή z3 αφού το S και το R δεν έχουν δύο κορυφές με κοινό missing color g ≠ a, b, c, f. Έστω h ≠ b είναι missing color του y (h ≠ a, b, c, f, g). Αν το μονοπάτι που ξεκινάει από την κορυφή y με χρώματα h, g δεν τελειώνει στην κορυφή v τότε εναλλάσσουμε τα χρώματα του μονοπατιού και καλείτε η περίπτωση δύο(case 2) αφού πλέον οι δύο κορυφές v, y του μονοπατιού Q έχουν κοινό missing color

g. Αλλιώς αν το μονοπάτι τελειώνει στην κορυφή v τότε εναλλάσσει τα χρώματα του μονοπατιού και πλέον οι κορυφές u και y έχουν κοινό missing color το g και αποχρωματίζει την ακμή (y, w3) που έχει χρώμα c και χρωματίζει με αυτό την (x, y) και καλεί την περίπτωση δύο (case2) για την ακμή (y, w3) με χρώματα (b, f). Αν η f-ακμή φεύγει από το T στην κορυφή x ή z1 τότε το bf-μονοπάτι (S) περιέχει ακριβώς τρεις (3) ή πέντε (5) κορυφές και δεν περιέχει ούτε τη x ούτε την z1. Εναλλάσσει τότε τα χρώματα του S, αποχρωματίζει το χρώμα c της ακμής (y, w3) και χρωματίζει με αυτό την (x, y). Το fc-κρίσιμο μονοπάτι, αν υπάρχει, έχει εφτά (7) ή παραπάνω κορυφές και καλείτε η recolor για την ακμή (y, w3) και τα χρώματα f, c. Τέλος αν μία f-ακμή φεύγει από το T είτε στην κορυφή z2 ή την z3 τότε στην ουσία φεύγουν δύο (2) f-ακμές από το T και από την z2 και από την z3. Μία f-ακμή ενώνει την w2 με μία από τις κορυφές x, z1 ή y. Αν το f ∈ C(w2, y) τότε f ∈ C(x, z1). Εναλλάσσει τα χρώματα του S και πλέον το af-κρίσιμο μονοπάτι, αν υπάρχει, έχει εφτά (7) ή παραπάνω κορυφές και καλείτε η recolor.

Η **περίπτωση 4.5 (case 4.5)** αφορά την περίπτωση που υπάρχουν τρεις (3) f-ακμές που φεύγουν από το T και το f δεν λείπει από καμία κορυφή του T. Έστω (u, u’) μία f-ακμή που φεύγει από το T τέτοια ώστε το u ∈ T – (x, y) , το u’ ∉ T και το g οποιοδήποτε missing color του u. Έτσι το g ≠ a, b, c, f και καμία g-ακμή δεν φεύγει από το T διαφορετικά θα ίσχυε μία από τις προηγούμενες τέσσερες (4) περιπτώσεις (case4.1-4.4). Έστω S το μονοπάτι που ξεκινάει από την κορυφή u με χρώματα g, f και (v, v’) η τελευταία f-ακμή που φεύγει από το S τέτοια ώστε το v ∈ T και το v’ ∉ T. Αν v = u τότε εναλλάσσει τα χρώματα του S και αφού το f ∈ M(u) τότε μία f-ακμή φεύγει από το T και αναγκαστικά μία από τις προηγούμενες τέσσερεις (4) περιπτώσεις ισχύουν και καλείτε η recolor. Στην περίπτωση όπου το v ≠ u αν είναι και διαφορετικό του x τότε έστω h οποιοδήποτε missing color του v διαφορετικό από a, b, c και δεν υπάρχει h-ακμή που να φεύγει από το T αλλιώς θα ίσχυαν οι προηγούμενες περιπτώσεις. Παίρνει το μονοπάτι που ξεκινάει από v με χρώματα h, g και τελειώνει στο u και δεν φεύγει από το T καθώς και το g και το h λείπουν μόνο στο u και v και δεν υπάρχουν g-h- ακμές που να φεύγουν από το T (αλλιώς θα ίσχυαν οι προηγούμενες περιπτώσεις). Εναλλάσσει τα χρώματα του μονοπατιού και πλέον το g ∈ M(v) και μετά από αυτό εναλλάσσει και τα χρώματα του μονοπατιού που ξεκινάει από v με χρώματα g, f. Αφού το f ∈ M(v) και μία f-ακμή φεύγει από το u μία από τις προηγούμενες περιπτώσεις προκύπτουν και καλείτε η recolor για h, b. Αν κανένα από τα μονοπάτια (x, a, f) και (x, c, f) περιέχουν μία κορυφή του T διαφορετική του x, τότε αν μία f-ακμή έστω (s, s') φεύγει από το T σε μία από τις κορυφές z2, z3, w2 ή w3. Έστω s =w2 ή w3 τότε εναλλάσσει τα χρώματα του μονοπατιού (x, a, f) . Αν δεν υπάρχει το fb-κρίσιμο μονοπάτι ή το fb-κρίσιμο μονοπάτι δεν έχει εφτά (7) ή παραπάνω κορυφές τότε καλείτε η recolor αλλιώς αφού το fb-κρίσιμο μονοπάτι περιέχει το πολύ πέντε (5) κορυφές και το μονοπάτι δεν περιέχει καμιά από τις w2, w3, s'. Εναλλάσσει τα χρώματα του fb-μονοπατιού ή κύκλου που περιέχει τα w2, w3, s'. Πλέον το καινούργιο cb-κρίσιμο μονοπάτι, αν υπάρχει, έχει τουλάχιστον εφτά (7) κορυφές και καλείτε η recolor. Αν υπάρχουν τρεις ακριβώς (3) f-ακμές που να φεύγουν από το T (x, x'), (y, y') και (z1, z1') τότε αν το b ∉ C(z1', y) τότε εναλλάσσει τα χρώματα του μονοπατιού (x, a, f) και το καινούργιο bf-κρίσιμο μονοπάτι, αν υπάρχει, έχει τουλάχιστον εφτά (7) κορυφές και καλείτε η recolor. Αλλιώς έστω P fb-μονοπάτι (x, z1, z1', y,' y) και δύο (2) από τις εννέα (9) κορυφές του P, Q ή R μονοπατιού έχουν κοινό missing color. Αν P έχει δύο από τις κορυφές με κοινό missing color τότε εναλλάσσει τα χρώματα του (x, a, f) και καλείτε η recolor αλλιώς τότε το P και το R έχουν δύο κορυφές με κοινό missing color και εναλλάσσει τα χρώματα του μονοπατιού (x, a, f) και καλείτε η recolor.

2.3.5 *Περίπτωση 5*

Η **περίπτωση πέντε (case5)** εξετάζει την περίπτωση που το ab-κρίσιμο μονοπάτι Q (x, z1, y) περιέχει τρεις (3) κορυφές. Αρχικά όπως και η περίπτωση τρία (case3) και πέντε (case5) ο πρώτος έλεγχος που θα χρειαστεί να γίνει είναι αν το **q < |E(H|/((|V(H)| -1)/2)** και αν ισχύει τότε το πλήθος χρωμάτων αυξάνεται κατά ένα και δίνει το χρώμα αυτό στην ακμή (x, y). Αν δεν ισχύει ο προηγούμενος έλεγχος τότε αυτό σημαίνει ότι υπάρχουν τουλάχιστον δύο ή περισσότερες ακμές που φεύγουν από το H και είναι χρωματισμένες με το ίδιο χρώμα. Το πρώτο πράγμα που κάνει η three είναι να καλέσει την *leave* έστω ότι c ∈ M(x) και δύο c-ακμές φεύγουν από το H στο y και z. Εναλλάσσει τα χρώματα του ac-μονοπατιού ή κύκλου που περιέχει τα y και z. Έτσι το νέο ab-κρίσιμο μονοπάτι, αν υπάρχει, έχει πέντε ή παραπάνω κορυφές και καλείτε η recolor.

Κεφάλαιο 3. Απαραίτητα Στοιχεία Ώστε να Εκτελεστεί ο Αλγόριθμος

Ο αλγόριθμος είναι γραμμένος στη γλώσσα προγραμματισμού ***Python***. Για την εκτέλεση του αλγορίθμου θα χρειαστεί και ένα text αρχείο.

3.1 *Απαραίτητα Αρχεία στον Φάκελο*

Για να τρέξει ο αλγόριθμος, πρέπει στον ίδιο φάκελο με το project της main να βρίσκεται και ένα text αρχείο που θα περιέχει το γράφημα με την εξής μορφή : δύο αριθμούς που θα αντικατοπτρίζουν τις κορυφές χωρισμένους με κενό και κάθε ακμή θα είναι η μία κάτω από την άλλη. Πχ με το παρακάτω παράδειγμα δημιουργείτε το γράφημα με τις ακμές ((1, 2), (2, 1), (2, 2))

1 2

2 1

2 2

3.2 *Βιβλιοθήκες*

Για να τρέξει ο αλγόριθμος, πρέπει πέρα από το text αρχείο να υπάρχουν και οι εξής βιβλιοθήκες της python: η network as nx, η matplotlib.pyplot as plt, η random και η από την (from) matplotlib.patches import FancyArrowPatch. Δεν θα χρειαστεί κάτι άλλο πέρα από αυτά για να τρέξει ο αλγόριθμος.

3.3 *Λεξικά*

Απαραίτητα εργαλεία για να λειτουργήσει σωστά ο αλγόριθμος είναι η δημιουργία δύο(2) λεξικών τα οποία θα συνδέονται μεταξύ τους και θα αλληλοεπιρεάζονται (bi-directional mapping) με το γράφημα το οποίο θα είναι και ο συνδετικός κρίκος ανάμεσα τους. Έτσι αλλάζοντας κάτι στο γράφημα θα αλλάζει αυτόματα και στα λεξικά. Τα δύο αυτά λεξικά θα περιέχουν το σαν κλειδί τις ακμές και σαν τιμή το χρώμα της εκάστοτε ακμής και ανάποδα (**edge_to_color_dict** και **color_to_edge_dict**). Τα λεξικά θα είναι τα πρώτα που θα αρχικοποιηθούν με όλες τις ακμές του γραφήματος (χωρίς χρώμα), πριν αρχίσει οποιαδήποτε άλλη λειτουργία του αλγορίθμου καθώς όλες οι αλλαγές θα γίνονται πάνω σε αυτά.

Κεφάλαιο 4. Πειράματα και αποτελέσματα

4.1 Πειράματα

Η πειραματική διαδικασία ήταν αρκετά απλή. Χρησιμοποίησα ένα σκριπτ για την δημιουργία γραφημάτων και ύστερα απλά έτρεξα τον αλγόριθμο για τα γραφήματα αυτά. Επίσης έγιναν πειράματα και σε απλά γραφήματα αλλά και σε πολυγραφήματα. Τα παραδείγματα που υπάρχουν από κάτω είναι σε διαφορετικά πλήθη ακμών, κορυφών, πυκνοτήτων αλλά και είδος γραφημάτων.

	50- 500	50-200
Βαθμός	25 | 28 | 25 | 26	13 | 13 | 14 | 15
Πλήθος χρωμάτων	28 | 28 | 27 | 28	13 | 14 | 14 | 15

	100- 1000	100-500
Βαθμός	28 | 33 | 30 | 30	20 | 20 | 19 | 21
Πλήθος χρωμάτων	29 | 33 | 30 | 30	20 | 20 | 19 | 21

	200- 2000	200-1000
Βαθμός	31 | 29 | 35 | 33	18 | 17 | 21 | 20
Πλήθος χρωμάτων	32 | 31 | 35 | 33	19 | 17 | 21 | 20

	400- 4000	400-2000
Βαθμός	33 | 32 | 33 | 37	18 | 21 | 21 | 19
Πλήθος χρωμάτων	33 | 33 | 34 | 37	19 | 21 | 22 | 20

	500- 5000	500-2500
Βαθμός	36 | 37 | 35 | 32	20 | 20 | 22 | 25
Πλήθος χρωμάτων	36 | 37 | 35 | 33	20 | 20 | 22 | 25

Πολυγραφήματα:

	50- 500	50-200
Βαθμός	29 | 30 | 28 | 29	16 | 14 | 12 | 16
Πλήθος χρωμάτων	29 | 30 | 28 | 29	16 | 14 | 13 | 16

	100- 1000	100-500
Βαθμός	31 | 31 | 34 | 30	19 | 19 | 17 | 19
Πλήθος χρωμάτων	31 | 31 | 34 | 30	19 | 19 | 17 | 19

	200- 2000	200-1000
Βαθμός	35 | 31 | 34 | 31	18 | 17 | 19 | 17
Πλήθος χρωμάτων	36 | 31 | 34 | 31	18 | 17 | 19 | 18

	400- 4000	400-2000
Βαθμός	35 | 36 | 32 | 35	19 | 23 | 21 | 21
Πλήθος χρωμάτων	35 | 36 | 32 | 35	20 | 23 | 21 | 21

	500- 5000	500-2500
Βαθμός	34 | 35 | 33 | 33	20 | 22 | 20 | 19
Πλήθος χρωμάτων	34 | 35 | 33 | 33	20 | 22 | 20 | 20

4.2 Πειραματικά αποτελέσματα

Έπειτα από τόσα πειράματα μπορεί κανείς να συμπεράνει ότι ο αλγόριθμος φέρνει σχεδόν σε κάθε περίπτωση τα βέλτιστα δυνατά αποτελέσματα.

Κεφάλαιο 5. Συμπεράσματα

Έχοντας δει όλα τα παραπάνω μπορούμε με ασφάλεια να οδηγηθούμε στα εξής συμπεράσματα:

1. Χρησιμοποιώντας τον αλγόριθμο που παρουσιάσαμε μπορούμε να πετύχουμε ένα βέλτιστο πλήθος χρωμάτων που θα χρησιμοποιηθεί στο κόστος όμως του χρόνου.
2. Πολλές από τις περιπτώσεις που ελέγχονται από τον αλγόριθμο είναι τόσο ειδικές που είναι πολύ δύσκολο να ισχύουν και μόνο αν το γράφημα είναι αρκετά "κακοφτιαγμένο" πρόκειται οι περισσότεροι από του ελέγχους να ισχύουν. Για αυτό και οι περισσότερες υπορουτίνες δεν θα χρησιμοποιηθούν ούτε σε αρκετά μεγάλα γραφήματα.
3. Ο αλγόριθμος φέρνει σχεδόν σε κάθε περίπτωση τα βέλτιστα δυνατά αποτελέσματα.

Αναφορές

[1] R. COLE AND J. HOPCROFT, On edge coloring bipartite graphs, SIAM J. Comput. 11, (2+93), 540-546.

[2] S. FIORINI AND R. J. WILSON, “Edge-colouring of Graphs”, Pitman, London, 1977.

[3] H. N. GABOW AND O. KARIV, Algorithms for edge coloring bipartite graphs and multi-graphs, SIAM J. comput. 11, (1982), 117-129.

[4] M. R. GAREY AND D. S. JOHNSON, “Computers and Intractability: A Guide to the Theory of NP-Completeness,” Freeman, San Francisco, 1979

[5] M. K. GOLDBERG, Edge-coloring of multigraphs; Recoloring technique, J. Graph Theory 8, (1984), 121-137

[6] M. K. GOLDBERG, An approximate algorithm for the edge0coloring problem, in “Proceedings of the 15th Southeastern Conference on Graph Theory, combinatorics, and Computing,” to appear.

[7] T. GONZALEZ AND S. SAHNI, Open shop scheduling to minimize finish time, J. Assoc. Comput. Mach. 23, (1976), 665-679

[8] I. J. HOLYER, The NP-completeness of edge colourings, SIAM J. Comput. 10, (1980), 718-720

[9] T. NISHIZEKI AND M. SATO, An approximation algorithm for edge-coloring multigraphs, preprint, 1983

[10] C. E. SHANNON, A theorem on colouring lines of a network, J. Math. Phys. 28, (1949), 148-151

[11] V. G. VIZING, On an estimate of the chromatic class of a p-graph, Diskret. Anal. 3, (1964), 25-30. [Russion]