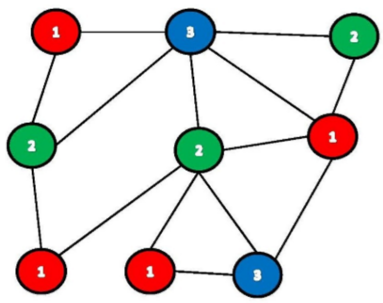
Εργασία στο μάθημα “Αλγόριθμοι και Πολυπλοκότητα”
Αλγόριθμοι χρωματισμού γραφημάτων V1.0

Γκόγκος Χρήστος
Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Πανεπιστημίου Ιωαννίνων

Άρτα, Οκτώβριος 2020

1 Εισαγωγή

Το πρόβλημα του χρωματισμού γραφήματος είναι ένα NP-hard πρόβλημα συνδυαστικής βελτιστοποίησης [Kar72]. Αφορά την ανάθεση ενός χρώματος σε κάθε κορυφή ενός γραφήματος έτσι ώστε γειτονικές κορυφές να χρωματίζονται με διαφορετικό χρώμα (όπως στο Σχήμα 1), ενώ παράλληλα χρησιμοποιείται ο ελάχιστος αριθμός διαφορετικών χρωμάτων. Στην παρούσα εργασία ζητείται η υλοποίηση τεσσάρων αλγορίθμων χρωματισμού γραφημάτων και η εφαρμογή τους σε γνωστά προβλήματα από τη βιβλιογραφία.



Σχήμα 1: Χρωματισμός κορυφών γραφήματος. Το σχήμα είναι από την ιστοσελίδα https://iq.opengenus.org

2 Περιγραφή προβλήματος

Το πρόβλημα χρωματισμού γραφήματος τυπικά ορίζεται ως εξής. Δεδομένου ενός μη κατευθυνόμενου απλού γραφήματος $G = (V, E)$ με ένα σύνολο κορυφών V και ένα σύνολο ακμών E, ζητείται η ανάθεση σε κάθε κορυφή $v \in V$ ενός ακεραίου $c(v) \in \{1, 2, ..., k\}$ έτσι ώστε το k να ελαχιστοποιείται και να ισχύει ότι $c(v) \neq c(u) \quad \forall \{v, u\} \in E$.

Το πρόβλημα συναντάται σε μεγάλο αριθμό πρακτικών εφαρμογών όπως ο χρονοπρογραμματισμός εκπαιδευτικών ιδρυμάτων (educational timetabling), ο χρονοπογραμματισμός αθλητικών γεγονότων (sports scheduling), η ανάθεση συχνοτήτων (frequency assignment), η ανάθεση καταχωρητών στους μεταγλωττιστές (compiler register allocation) και άλλα.

Πολλοί αλγόριθμοι χρωματισμού γραφημάτων έχουν προταθεί τα τελευταία 50 έτη. Στην παρούσα εργασία θα εξεταστούν τέσσερις αλγόριθμοι που ανήκουν στις λεγόμενες κατασκευαστικές τεχνικές (constructive techniques). Οι κατασκευαστικές τεχνικές δημιουργούν λύσεις βήμα προς βήμα, αναθέτοντας στη σειρά, σε κάθε κορυφή, ένα χρώμα, πιθανά εφαρμόζοντας οπισθοχώρηση κατά τη διαδικασία. Οι αλγόριθμοι που θα εξεταστούν είναι ο αλγόριθμος first fit, ο αλγόριθμος DSATUR, ο αλγόριθμος Recursive Largest First και ο αλγόριθμος backtracking DSATUR. Πληροφορίες για τους ανωτέρω αλγορίθμους μπορούν να βρεθούν στο άρθρο [LTMG12] καθώς και στις αναφορές του ίδιου άρθρου.

3 Δεδομένα προβλήματος

Το πρόβλημα χρονοπρογραμματισμού εξετάσεων αφορά φοιτητές που έχουν πραγματοποιήσει εγγραφές σε εξετάσεις μαθημάτων. Για κάθε εξέταση διατίθεται μια λίστα από φοιτητές και κάθε φοιτητής μπορεί να είναι εγγεγραμμένος σε μια ή περισσότερες εξετάσεις. Κάθε εξέταση θα πρέπει να τοποθετηθεί σε μια περίοδο εξέτασης και η λύση του προβλήματος συνίσταται στην ανάθεση όλων των εξετάσεων στο μικρότερο δυνατό αριθμό περιόδων έτσι ώστε να μην υπάρχουν **συγκρούσεις**, δηλαδή να μην υπάρχουν φοιτητές που θα έπρεπε να συμμετάσχουν σε εξετάσεις σε περισσότερα του ενός μαθήματα στην ίδια περίοδο.

Ως δεδομένα του προβλήματος θα χρησιμοποιηθούν τα δεδομένα του προβλήματος χρονοπρογραμματισμού εξετάσεων Toronto τα οποία είναι διαθέσιμα προς μεταφόρτωση στη διεύθυνση https://github.com/chgogos/datasets/blob/main/UETT/toronto.zip. Τα δεδομένα Toronto αποτελούνται από 13 προβλήματα και πληροφορίες για κάθε πρόβλημα παρουσιάζονται στον Πίνακα 1. Τα αρχεία δεδομένων (κατάληξη .stu) διαθέτουν για κάθε σπουδαστή μια γραμμή που περιέχει τους αριθμούς των μαθημάτων στα οποία είναι εγγεγραμμένος χωρισμένους μεταξύ τους με κενά. Η πρώτη γραμμή του αρχείου αντιστοιχεί στον πρώτο σπουδαστή, η δεύτερη γραμμή στο δεύτερο σπουδαστή κ.ο.κ. Για παράδειγμα το αρχείο car-f-92.stu περιέχει 18419 σειρές δεδομένων και ξεκινά με τις ακόλουθες σειρές:

```
0170
0156
0281
0006
0154 0156
0383
0534 0535 0536
0275
0091 0160 0164
...
```

που σημαίνουν ότι ο φοιτητής 1 έχει εγγραφεί στο μάθημα 0170, ο φοιτητής 2 έχει εγγραφεί στο μάθημα 0156, ο φοιτητής 3 έχει εγγραφεί στο μάθημα 0281, ο φοιτητής 4 έχει εγγραφεί στο μάθημα 0006, ο φοιτητής 5 στα μαθήματα 0154 0156 κ.ο.κ

Πίνακας 1: Δεδομένα προβλημάτων

Πρόβλημα	Αρχείο Δεδομένων	Εξετάσεις	Φοιτητές	Εγγραφές
car-f-92	car-f-92.stu	543	18419	55522
car-s-91	car-s-91.stu	682	16925	56877
ear-f-83	ear-f-83.stu	190	1125	8109
hec-s-92	hec-s-92.stu	81	2823	10632
kfu-s-93	kfu-s-93.stu	461	5349	25113
lse-f-91	lse-f-91.stu	381	2726	10918
pur-s-93	pur-s-93.stu	2419	30029	120681
rye-s-93	rye-s-93.stu	486	11483	45051
sta-f-83	sta-f-83.stu	139	611	5751
tre-s-92	tre-s-92.stu	261	4360	14901
uta-s-92	uta-s-92.stu	622	21266	58979
ute-s-92	ute-s-92.stu	184	2749	11793
yor-f-83	yor-f-83.stu	181	941	6034

Θεωρώντας κάθε εξέταση ως κόμβο ενός γραφήματος και κάθε ακμή ανάμεσα σε δύο κόμβους να υποδηλώνει την ύπαρξη κοινών φοιτητών ανάμεσα στις δύο εξετάσεις που βρίσκονται στα άκρα της ακμής, το πρόβλημα μπορεί να θεωρηθεί ως πρόβλημα χρωματισμού γραφήματος όπου κάθε χρώμα είναι και μια περίοδος εξέτασης.

4 Ζητούμενα

Να κατασκευάσετε ένα πρόγραμμα που θα ελέγχεται από ένα μενού (ή ένα GUI=Graphical User Interface) με τις ακόλουθες επιλογές:

1. Στατιστικά στοιχεία προβλημάτων
2. Επίλυση προβλήματος με first fit
3. Επίλυση προβλήματος με DSATUR
4. Επίλυση προβλήματος με RLF
5. Επίλυση προβλήματος με backtracking DSATUR
6. Μαζική επίλυση προβλημάτων

Σε όσες επιλογές ζητείται η επίλυση προβλήματος θα παράγεται ένα αρχείο λύση (με επέκταση .sol) που θα είναι αρχείο κειμένου και το οποίο σε κάθε γραμμή θα περιέχει τον αριθμό κορυφής (εξέτασης), ξεκινώντας από την πρώτη κορυφή, ακολουθούμενη από ένα κενό και τον αριθμό χρώματος (περίοδο εξέτασης) στην οποία έχει τοποθετηθεί στη συγκεκριμένη λύση. Σε περίπτωση που χρησιμοποιούνται k χρώματα οι τιμές των χρωμάτων θα είναι από το 0 μέχρι και το $k-1$. Για παράδειγμα η έξοδος ενός προβλήματος θα μπορούσε να είναι η ακόλουθη:

1 11
2 11
3 9
4 6
5 0
...

που σημαίνει ότι η κορυφή 1 έχει λάβει το χρώμα 11, η κορυφή 2 έχει λάβει το χρώμα 11, η κορυφή 3 έχει λάβει το χρώμα 9, η κορυφή 4 έχει λάβει το χρώμα 6, η κορυφή 5 έχει λάβει το χρώμα 0 κ.ο.κ.

4.1 Στατιστικά στοιχεία προβλημάτων

Θα εμφανίζονται τα ακόλουθα στατιστικά στοιχεία για καθένα από τα 13 προβλήματα του Toronto dataset:

1. Αριθμός κορυφών.
2. Πυκνότητα. Για τον υπολογισμό της πυκνότητας θα πρέπει να κατασκευαστεί ο πίνακας συγκρούσεων. Ο πίνακας συγκρούσεων είναι ένας δισδιάστατος πίνακας c στον οποίο κάθε στοιχείο $c_{ij} = 1$ αν η εξέταση i βρίσκεται σε σύγκρουση με την εξέταση j ενώ ισχύει ότι $c_{ij} = 0$ σε άλλη περίπτωση. Η πυκνότητα συγκρούσεων υπολογίζεται διαιρώντας τον αριθμό των στοιχείων του πίνακα συγκρούσεων που έχουν την τιμή 1 με το συνολικό πλήθος των στοιχείων του πίνακα.
3. Για τους βαθμούς (degrees) των κορυφών η ελάχιστη τιμή (min), η διάμεσος τιμή (median), η μέγιστη τιμή (max), η μέση τιμή (mean) καθώς και ο συντελεστής διακύμανσης (CV=coefficient of variation) που ορίζεται ως η τυπική απόκλιση προς τη μέση τιμή.

Τα αποτελέσματα θα πρέπει να είναι παρόμοια με τις τιμές του Table 1 από το [LTMG12] στη σελίδα 17.

4.2 Επίλυση προβλήματος με first fit

Θα ζητείται η επιλογή ενός από τα διαθέσιμα προβλήματα και θα επιλύεται με τη μέθοδο first fit. Ο αλγόριθμος first fit είναι ένας άπληστος (greedy) αλγόριθμος που λαμβάνει κάθε κορυφή και την αναθέτει στο μικρότερο αριθμό χρώματος που δεν προκαλεί σύγκρουση, δημιουργώντας νέα χρώματα όταν χρειάζεται. Οι κορυφές μπορούν αρχικά να ταξινομηθούν σε φθίνουσα σειρά βαθμού, όπως έχει προταθεί στο [WP67] και ο χρωματισμός των κορυφών να γίνει από την κορυφή με τον υψηλότερο βαθμό προς την κορυφή με το χαμηλότερο βαθμό.

4.3 Επίλυση προβλήματος με DSATUR

Θα ζητείται η επιλογή ενός από τα διαθέσιμα προβλήματα και θα επιλύεται με τον αλγόριθμο DSATUR. Ο αλγόριθμος DSATUR [Bré79] καθορίζει δυναμικά την επόμενη κορυφή που θα χρωματιστεί επιλέγοντας ανάμεσα στις κορυφές που δεν είναι χρωματισμένες εκείνη που κάθε φορά έχει το μεγαλύτερο αριθμό διαφορετικών χρωμάτων σε γειτονικές κορυφές.

4.4 Επίλυση προβλήματος με RLF

Θα ζητείται η επιλογή ενός από τα διαθέσιμα προβλήματα και θα επιλύεται με τον αλγόριθμο Recursive Largest First που περιγράφεται στο [LTMG12] καθώς και στο [Lei79].

4.5 Επίλυση προβλήματος με backtracking Dsatur

Θα ζητείται η επιλογή ενός από τα διαθέσιμα προβλήματα και θα επιλύεται με τη μέθοδο backtracking DSATUR που περιγράφεται στο [LTMG12].

4.6 Μαζική επίλυση προβλημάτων

Θα επιλύονται όλα τα προβλήματα Toronto (13) με όλους τους διαθέσιμους τρόπους επίλυσης (4), δηλαδή θα παράγονται $13 \times 4 = 52$ σειρές αποτελεσμάτων που θα αναφέρουν το όνομα του προβλήματος, τη μέθοδο επίλυσης και τον αριθμό χρωμάτων που χρειάστηκαν.

5 Παραδοτέα εργασίας

Τα παραδοτέα της εργασίας θα είναι τα ακόλουθα:

1. Τεχνική αναφορά για την εργασία, στα πρότυπα σύντομου επιστημονικού άρθρου (5 σελίδες).
2. Github project που θα εμφανίζεται ως Github page και το οποίο θα περιέχει:
   (α') Σύντομη περιγραφή της εφαρμογής
   (β') Τον πλήρη κώδικα της εφαρμογής
   (γ') Οδηγίες εγκατάστασης
   (δ') Οδηγίες εκτέλεσης

6 Παρατηρήσεις

- Η υλοποίηση του κώδικα θα πρέπει να γίνει κατά προτίμηση στην γλώσσα προγραμματισμού C++. Εναλλακτικά μπορεί να χρησιμοποιηθεί η Java ή η Python.
- Η υλοποίηση επιπλέον αλγορίθμων χρωματισμού γραφημάτων δίνει bonus στη βαθμολογία της εργασίας.

Αναφορές

[Bré79] Daniel Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.

[Kar72] Richard M Karp. Complexity of computer computations, chapter reducibility among combinatorial problems. *Plenum Press. Survey of State-of-the-Art*, 23:85–104, 1972.

[Lei79] Frank Thomson Leighton. A graph coloring algorithm for large scheduling problems. *Journal of research of the national bureau of standards*, 84(6):489–506, 1979.

[LTMG12] Rhyd Lewis, J Thompson, C Mumford, and J Gillard. A wide-ranging computational comparison of high-performance graph colouring algorithms. *Computers & Operations Research*, 39(9):1933–1950, 2012.

[WP67] Dominic JA Welsh and Martin B Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 10(1):85–86, 1967.