



Τμήμα Πληροφορικής & Τηλεπικοινωνιών Πανεπιστήμιο Ιωαννίνων

Αλγόριθμοι και Πολυπλοκότητα

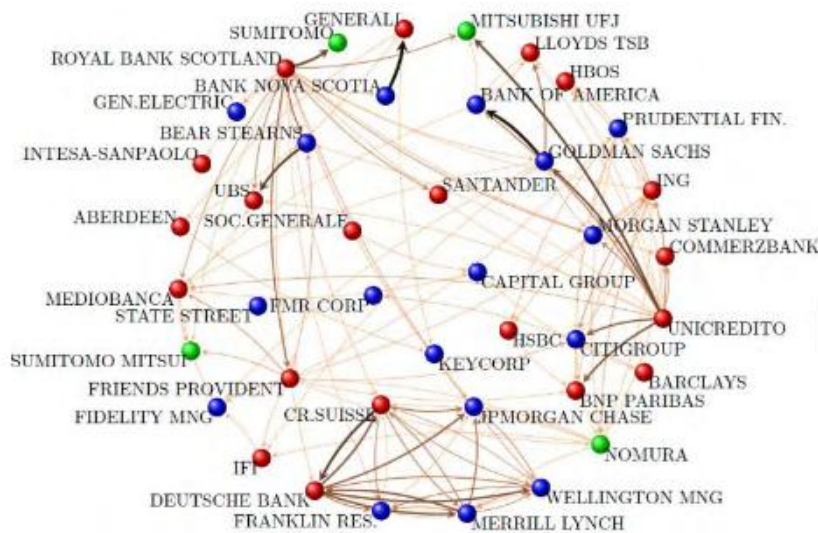
Τεχνική Αναφορά Εργασίας

“Αλγόριθμοι χρωματισμού γραφημάτων”

Παπαδημητρίου Μελίνα

A.M. 1534

Διδάσκων: Γκόγκος Χρήστος



Περίληψη

Το πρόβλημα του χρωματισμού γραφήματος είναι ένα NP-hard πρόβλημα συνδυαστικής βελτιστοποίησης. Αφορά την ανάθεση ενός χρώματος σε κάθε κορυφή ενός γραφήματος έτσι ώστε γειτονικές κορυφές να χρωματίζονται με διαφορετικό χρώμα, 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.

Περιεχόμενα

Περίληψη	2
ΕΙΣΑΓΩΓΗ (ΠΡΟΒΛΗΜΑΤΑ NP-COMLETE)	4
ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΧΡΩΜΑΤΙΣΜΟΥ ΓΡΑΦΗΜΑΤΩΝ	4
ΠΡΟΣΕΓΓΙΣΕΙΣ ΕΠΙΛΥΣΗΣ.....	5
ΔΕΔΟΜΕΝΑ ΠΡΟΒΛΗΜΑΤΟΣ (Toronto datasets)	5
ΠΙΝΑΚΑΣ ΣΤΑΤΙΣΤΙΚΩΝ ΣΤΟΙΧΕΙΩΝ ΠΡΟΒΛΗΜΑΤΩΝ.....	7
ΕΦΑΡΜΟΓΗ ΑΛΓΟΡΙΘΜΩΝ	8
FIRST FIT	8
DSATUR.....	8
Μαζική επίλυση προβλημάτων	8
ΑΠΟΤΕΛΕΣΜΑΤΑ	9
FIRST FIT	9
DSATUR.....	9
Βιβλιογραφία	11

ΕΙΣΑΓΩΓΗ (ΠΡΟΒΛΗΜΑΤΑ NP-COMPLETE)

Τα NP-πλήρη Προβλήματα (NP-complete Problems) ανήκουν στην κλάση NP.

Για να δείξουμε ότι ένα πρόβλημα P είναι NP-πλήρες θα πρέπει να δείξουμε ότι ανήκει στην κλάση NP κατασκευάζοντας έναν πολυωνυμικό επαληθευτή και έπειτα να δείξουμε ότι κάθε πρόβλημα στην κλάση NP ανάγεται στο P.

Το δεύτερο μέρος όμως, είναι αρκετά πολύπλοκο και για αυτό το λόγο αν αποδείξουμε ότι ισχύει ένας αλγόριθμος πολυωνυμικού χρόνου για οποιοδήποτε από αυτά θα σήμαινε την ύπαρξη ενός αλγορίθμου πολυωνυμικού χρόνου για όλα. (Δούρος, 2012)

ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΧΡΩΜΑΤΙΣΜΟΥ ΓΡΑΦΗΜΑΤΩΝ

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 $c(v) \in \{1, 2, \dots, k\}$ έτσι ώστε το k να ελαχιστοποιείται και να ισχύει ότι $c(v) \neq c(u) \forall \{v, u\} \in E$. Το πρόβλημα συναντάται σε μεγάλο αριθμό πρακτικών εφαρμογών όπως ο χρονοπρογραμματισμός εκπαιδευτικών ιδρυμάτων (educational timetabling), ο χρονοπρογραμματισμός 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. (Wikipedia, 2020)

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. (Γκόγκος, 2020)

ΠΡΟΣΕΓΓΙΣΕΙΣ ΕΠΙΛΥΣΗΣ

ΔΕΔΟΜΕΝΑ ΠΡΟΒΛΗΜΑΤΟΣ (Toronto datasets)

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) που 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 (Γκόγκος, github, 2020)

ΠΙΝΑΚΑΣ ΣΤΑΤΙΣΤΙΚΩΝ ΣΤΟΙΧΕΙΩΝ ΠΡΟΒΛΗΜΑΤΩΝ

FILENAME	DENSITY	MAX	MEDIAN	MIN	MEAN	CV
ear_f_83.stu	0.26554	134	45	4	50.4526	56.2615
hec_s_92.stu	0.4155	62	32	9	33.65	36.55
sta_f_83.stu	0.143	61	16	7	19.87	67.61
ute_s_92.stu	0.08448	58	13	2	15.54	69.32
kfu_s_93.stu	0.05546	247	18	0	25.57	120.1
yor_f_83.stu	0.2873	117	51	7	52	35.32
car_f_92.stu	0.1377	381	63	0	74.79	75.41
car_s_91.stu	0.1282	472	77	0	87.43	70.96
lse_f_91.stu	0.06243	134	16	0	23.78	93.28
pur_s_93.stu	0.02948	857	47	0	71.32	129.5
rye_s_93.stu	0.07512	274	24	0	36.51	111.9
tre_s_92.stu	0.18	145	45	0	46.98	59.73
uta_s_92.stu	0.1254	303	65	1	77.97	73.73

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 διδιάστατος 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 ($CV = \text{coefficient of variation}$) που ορίζεται ως η τυπική απόκλιση προς τη μέση τιμή.

ΕΦΑΡΜΟΓΗ ΑΛΓΟΡΙΘΜΩΝ

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.

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.

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.

ΑΠΟΤΕΛΕΣΜΑΤΑ

FIRST FIT

FILENAME	COLORS USED
sta_f_83.stu	13
rye_s_93.stu	28
hec_s_92.stu	22
uta_s_92.stu	43
ute_s_92.stu	13
yor_f_83.stu	27
car_f_92.stu	44
car_s_91.stu	48
ear_f_83.stu	29
kfu_s_93.stu	25
lse_f_91.stu	22
pur_s_93.stu	54
tre_s_92.stu	29

DSATUR

FILENAME	COLORS USED
rye_s_93.stu	23
hec_s_92.stu	20
sta_f_83.stu	13
uta_s_92.stu	35
ute_s_92.stu	10
yor_f_83.stu	24
car_f_92.stu	34

car_s_91.stu	35
ear_f_83.stu	25
kfu_s_93.stu	21
lse_f_91.stu	18
pur_s_93.stu	38
tre_s_92.stu	24

Βιβλιογραφία

Δούρος, Β. (2012, Ιούνιος 11). <http://www2.aueb.gr/>. Ανάκτηση από Κλάση NP, NP-Complete:
http://www2.aueb.gr/users/douros/algorithms/tutorials_2012/14_frontistirio_complete.pdf

Wikipedia. (2020, Δεκέμβριος 19). *Wikipedia*. Ανάκτηση από Graph_coloring:
https://en.wikipedia.org/wiki/Graph_coloring

Γκόγκος, Χ. (2020). *github*. Ανάκτηση από datasets:
<https://github.com/chgogos/datasets/blob/main/UETT/toronto.zip>

Γκόγκος, Χ. (2020). *Github*. Ανάκτηση από Αλγόριθμοι και Πολυπλοκότητα:
<https://github.com/chgogos/alco>