



# Αλγόριθμοι Χρωματισμού Γραφημάτων

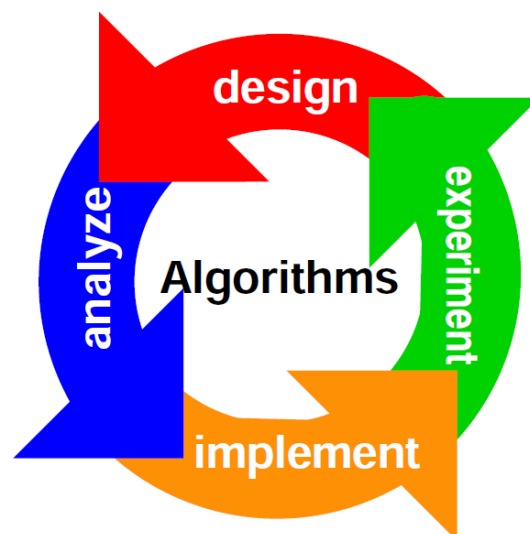
## Αλγόριθμοι και Πολυπλοκότητα

Κωνσταντίνος Σακκάς

AM: 15799 (παλιό), 1549 (νέο)

[ksakkas@uoi.gr](mailto:ksakkas@uoi.gr)

<http://myweb.dit.uoi.gr/ksakkas/mysite/>





## Περίληψη:

Το πρόβλημα του χρωματισμού γραφήματος είναι ένα NP-hard πρόβλημα συνδυαστικής βελτιστοποίησης.

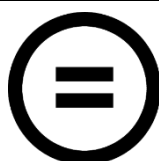
Αφορά την ανάθεση ενός χρώματος σε κάθε κορυφή ενός γραφήματος έτσι ώστε γειτονικές κορυφές να χρωματίζονται με διαφορετικό χρώμα, ενώ παράλληλα χρησιμοποιείται ο ελάχιστος αριθμός διαφορετικών χρωμάτων.

Στην παρούσα εργασία ζητείται η υλοποίηση τεσσάρων αλγορίθμων χρωματισμού γραφημάτων και η εφαρμογή τους σε γνωστά προβλήματα από τη βιβλιογραφία.



## Περιεχόμενα

Περίληψη:	2
1. ΕΙΣΑΓΩΓΗ (ΠΡΟΒΛΗΜΑΤΑ NP-COMPLETE)	4
1.1 Η Κλάση P	4
1.2 Η Κλάση NP	4
1.3 P vs. NP	5
1.4 NP-Complete (NP-Πλήρες)	5
2. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΧΡΩΜΑΤΙΣΜΟΥ ΓΡΑΦΗΜΑΤΩΝ	6
3. ΠΡΟΣΕΓΓΙΣΕΙΣ ΕΠΙΛΥΣΗΣ	7
3.1 ΔΕΔΟΜΕΝΑ ΠΡΟΒΛΗΜΑΤΟΣ (Toronto datasets)	7
3.2 ΠΙΝΑΚΑΣ ΣΤΑΤΙΣΤΙΚΩΝ ΣΤΟΙΧΕΙΩΝ ΠΡΟΒΛΗΜΑΤΩΝ	8
4. ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ	10
4.1 FIRST FIT	10
4.2 DSATUR	10
4.3 RLF	10
4.4 Backtracking Dsaturn	10
4.5 Μαζική επίλυση προβλημάτων	11
5. ΑΠΟΤΕΛΕΣΜΑΤΑ (LINK)	12
5.1 Στατιστικά Αρχείων .stu	12
5.2 FIRST FIT	13
5.3 DSATUR	14
Αναφορές	15



GDPR

Copyright © 2021 Κωνσταντίνος Σακκάς. Με την επιφύλαξη παντός  
δικαιώματος



## 1. ΕΙΣΑΓΩΓΗ (ΠΡΟΒΛΗΜΑΤΑ NP-COMplete)

### 1.1 Η Κλάση P

Η Κλάση P (polynomial): Περιλαμβάνει τα προβλήματα που επιλύονται (αποφασίζονται) από κάποιο αλγόριθμο εντός πολυωνυμικού χρόνου.

Στην ουσία έχουμε συνδέσει την έννοια της αποδοτικότητας με την κλάση P. Η σύνδεση αυτή βασίζεται στην παρατήρηση ότι ένας πολυωνυμικός αλγόριθμος για ένα πρόβλημα βασίζεται σε κάποια βαθύτερη ιδιότητα του προβλήματος που επιτρέπει τη γρήγορη επίλυσή του αποφεύγοντας την λύση της εξαντλητικής αναζήτησης (brute force) που για τυπικά προβλήματα είναι εκθετικής πολυπλοκότητας. Για παράδειγμα, η εύρεση ενός ελάχιστου ζευγνύοντος δένδρου γίνεται σε πολυωνυμικό χρόνο, αφού μπορεί να εφαρμοσθεί η μέθοδος της απληστίας. (Δούρος, 2012) (Ζισσιμόπουλος, -)

### 1.2 Η Κλάση NP

Περιλαμβάνει τα προβλήματα που μπορούν να επαληθευτούν εντός πολυωνυμικού χρόνου.

Η κλάση πολυπλοκότητας NP αντιστοιχεί σε όλα τα προβλήματα για τα οποία υπάρχει πολυωνυμικός αντισταθμιστικός (non-deterministic) αλγόριθμος. Υπάρχει όμως ένας ισοδύναμος ορισμός που είναι εξαιρετικά πιο διαεισθητικός και δείχνει ακριβώς τη σημασία αυτής της κλάσης πολυπλοκότητας. Ένα πρόβλημα P ανήκει στην κλάση NP αν έχει την ιδιότητα της πολυωνυμικής επαληθευσιμότητας. Για παράδειγμα, το πρόβλημα της ύπαρξης Hamiltonian κυκλώματος (HAMiltonian Circle - HAMC) ορίζεται ως εξής:

Δοθέντος ενός μη-κατευθυνόμενου γραφήματος  $G = (V, E)$ , να βρεθεί αν υπάρχει Hamiltonian κύκλωμα  $W$  στο  $G$ . (Δούρος, 2012)  
(Ζισσιμόπουλος, -)



## 1.3 P vs. NP

Το πρόβλημα P vs. NP είναι ΤΟ ανοιχτό πρόβλημα στο χώρο της θεωρητικής Πληροφορικής.

99,99999...9% η σωστή απάντηση είναι ότι  $P \neq NP$

Κανένας όμως μέχρι σήμερα δεν έχει καταφέρει να το αποδείξει και να πείσει την επιστημονική κοινότητα ότι το απέδειξε σωστά.

0,000...0001% είναι η πιθανότητα η σωστή απάντηση να είναι  $P = NP$ .  
(Δούρος, 2012)

## 1.4 NP-Complete (NP-Πλήρες)

Ανήκει στην κλάση NP

– Όλα τα υπόλοιπα προβλήματα της κλάσης NP ανάγονται (πολυωνυμικά) σε αυτό.

«Ένα πρόβλημα A ανάγεται σε ένα πρόβλημα B»

- Το A ανήκει στην κλάση NP
- Το B ανήκει στην κλάση NPComplete
- Το A δεν είναι πιο δύσκολο από το B

Για να δείξουμε, λοιπόν, ότι ένα πρόβλημα Π 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 Π. Το δεύτερο μέρος, όμως, είναι αρκετά πολύπλοκο και για αυτό το λόγο αν αποδείξουμε ότι ισχύει για ένα πρόβλημα, έπειτα κάνοντας αναγωγές από αυτό θα αποδεικνύουμε ότι και άλλα προβλήματα είναι NP-πλήρη. Το γνωστό θεώρημα των Cook-Levin δίνει το πρώτο τέτοιο πρόβλημα που είναι NP-πλήρες.  
(Ζισσιμόπουλος, -) (Δούρος, 2012) (Βικιπαίδεια, 2020)



## 2. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΧΡΩΜΑΤΙΣΜΟΥ ΓΡΑΦΗΜΑΤΩΝ

Το πρόβλημα χρωματισμού ακμών διερευνά αν είναι εφικτό να χρωματίσουμε τις ακμές ενός δοθέντος γραφήματος χρησιμοποιώντας μέχρι  $k$  διαφορετικά χρώματα, όπου  $k$  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.

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. 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, 2021) (Wikipedia, 2020)



### 3. ΠΡΟΣΕΓΓΙΣΕΙΣ ΕΠΙΛΥΣΗΣ

#### 3.1 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) 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 , 2021)

## 3.2 ΠΙΝΑΚΑΣ ΣΤΑΤΙΣΤΙΚΩΝ ΣΤΟΙΧΕΙΩΝ ΠΡΟΒΛΗΜΑΤΩΝ

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

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 ( $CV = \text{coefficient of variation}$ ) 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. (Γκόγκος, 2021)



## 4. ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ

### 4.1 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.

### 4.2 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.

### 4.3 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].

### 4.4 Backtracking Dsaturn

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



### 4.5 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. (Γκόγκος, 2021)



## 5. ΑΠΟΤΕΛΕΣΜΑΤΑ ([LINK](#))

### 5.1 Στατιστικά Αρχείων .stu

**STATISTICS TABLE BY FILE**

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



## 5.2 FIRST FIT

### FIRST FIT RESULTS BY FILE

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



### 5.3 DSATUR

#### DSATUR RESULTS BY FILE

FILENAME	COLORS USED	ALGORITHM EXECUTED
rye_s_93.stu	23	DSATUR
hec_s_92.stu	20	DSATUR
sta_f_83.stu	13	DSATUR
uta_s_92.stu	35	DSATUR
ute_s_92.stu	10	DSATUR
yor_f_83.stu	24	DSATUR
car_f_92.stu	34	DSATUR
car_s_91.stu	35	DSATUR
ear_f_83.stu	25	DSATUR
kfu_s_93.stu	21	DSATUR
lse_f_91.stu	18	DSATUR
pur_s_93.stu	38	DSATUR
tre_s_92.stu	24	DSATUR



## Αναφορές

- Δούρος, Β. (2012, Ιούνιος 11). *Κλάση NP, NP-Complete*. Ανάκτηση από <http://www2.aueb.gr/>:  
[http://www2.aueb.gr/users/douros/algorithms/tutorials\\_2012/14\\_frontistiro\\_complete.pdf](http://www2.aueb.gr/users/douros/algorithms/tutorials_2012/14_frontistiro_complete.pdf)
- Wikipedia. (2020, Δεκέμβριος 19). *Graph coloring*. Ανάκτηση από [en.wikipedia.org/](https://en.wikipedia.org/):  
[https://en.wikipedia.org/wiki/Graph\\_coloring](https://en.wikipedia.org/wiki/Graph_coloring)
- Βικιπαίδεια. (2020, Οκτώβριος 18). *Πρόβλημα  $P=NP$* . Ανάκτηση από [el.wikipedia.org/](https://el.wikipedia.org/):  
[https://el.wikipedia.org/wiki/%CE%A0%CF%81%CF%8C%CE%B2%CE%BB%CE%B7%CE%BC%CE%B1\\_P%3DNP](https://el.wikipedia.org/wiki/%CE%A0%CF%81%CF%8C%CE%B2%CE%BB%CE%B7%CE%BC%CE%B1_P%3DNP)
- Γκόγκος, Χ. (2021). *Αλγόριθμοι και Πολυπλοκότητα*. Ανάκτηση από [github.com/chgogos/alco](https://github.com/chgogos/alco): <https://github.com/chgogos/alco>
- Ζισσιμόπουλος, Β. (-). *Η κλάση NP*. Ανάκτηση από <http://cgi.di.uoa.gr/>:  
<http://cgi.di.uoa.gr/~vassilis/ac/12L30-NPcompleteness.pdf>