Αλγόριθμοι και προχωρημένες δομές δεδομένων

Βασιλικη νασκαρη

2021

Περιεχόμενα

[Εισαγωγή 1](#_Toc82982622)

[Περιγραφή προβλήματος 1](#_Toc82982623)

[Τεχνικές επίλυσης προβλήματος 2](#_Toc82982624)

[Αποτελέσματα 4](#_Toc82982625)

# Εισαγωγή

 Εξ ορισμού, αναφερόμενοι σε*P* προβλήματα εννοούμε τα προβλήματα που μπορούν να λυθούν σε πολυωνυμικό χρόνο , δηλαδή θεωρούμε δεδομένη την ύπαρξη ενός αλγορίθμου για τη λύση του προβλήματος έτσι ώστε ο αρθμός των βημάτων του αλγορίθμου να περιορίζεται από μια συνάρτηση n, όπου το n αντιστοιχεί στο μήκος εισόδου του προβλήματος. *NP*προβλήματα χαρακτηρίζονται για τα οποία υπάρχει πολυωνυμικός non-deterministic αλγόριθμος. Ως εκ τούτου, μπορούμε να ισχυριστούμε ότι ένα πρόβλημα P ανήκει στην κλάση NP εάν έχει την ιδιότητα της πολυωνυμικής επαληθευσιμότητας.

     Η διαφορά των ανωτέρω προβλημάτων έγκειται στο γεγονός ότι η κλάση P αποτελείται από όλα τα προβλήματα απόφασης τα οποία μπορούν να επιλυθούν από μια προσδιοριστική ακολουθιακή μηχανή σε χρόνο ο οποίος είναι πολυωνυμικός στο μέγεθος της εισόδου, ενώ η κλάση NP αποτελείται από όλα τα προβλήματα απόφασης των οποίων οι λύσεις μπορούν να επιβεβαιωθούν σε πολυωνυμικό χρόνο με δεδομένο τις σωστές πληροφορίες, ή ισοδύναμα, των οποίων οι λύσεις μπορούν να βρεθούν σε πολυωνυμικό χρόνο σε μια μη-προσδιοριστή μηχανή.

     Με σκοπό την προσέγγιση του ερωτήματος εάν P=NP, θεωρείται πολύ χρήσιμη η έννοια των NP-Complete προβλημάτων. Με τον όρο NP-Complete προβλήματα αναφερόμαστε σε ένα σύνολο προβλημάτων σε καθένα από τα οποία μπορεί να υποβιβαστεί ένα οποιοδήποτε άλλο NP-πρόβλημα σε πολυωνυμικό χρόνο, και των οποίων η λύση εξακολουθεί να μπορεί να επιβεβαιωθεί σε πολυωνυμικό χρόνο. Αυτό σημαίνει ότι οποιοδήποτε NP πρόβλημα μπορεί να αναχθεί σε ένα οποιοδήποτε NP-Complete πρόβλημα.

    Στην παρούσα εργασία, το πρόβλημα χρωματισμού του γράφου κατά το οποίο παράγεται η αρχική λύση ανήκει στην κατηγορία των NP-hard προβλημάτων, ενώ η διαδιακασία βελτιστοποίησης που πραγματοποιείται με τη χρήση του μεταευρετικού αλγορίθμου της προσομοιωμένης ανόπτησης αποτελεί ένα NP-Complete πρόβλημα. [1]

# Περιγραφή προβλήματος

Το πρόβλημα του χρονοπρογραμματισμού εξετάσεων αφορά εξετάσεις, σπουδαστές και χρονικές περιόδους σε κάθε μία από τις οποίες μπορούν προγραμματιστούν μία η περισσότερες εξετάσεις. Το πρόβλημα ανήκει στην κατηγορία των προβλημάτων NP-Complete. Το πρόβλημα αφορά την δημιουργία εφικτού προγράμματος εξετάσεων ώστε να ικανοποιείται ο περιορισμός του προγραμματισμού σε διαφορετικές χρονικές περιόδους, εξετάσεων, οι οποίες έχουν κοινούς φοιτητές. Το σύνολο δεδομένων το οποίο χρησιμοποιήθηκε, είναι το σύνολο δεδομένων carter, το οποίο περιέχει 13 προβλήματα πραγματικού χρόνου. Τα αρχεία δεδομένων αποτελούνται από την εξής δομή. Οι γραμμές του αρχείου δεδομένων αναπαριστούν τους φοιτητές του προβλήματος και αποτελούνται από σύνολα εξετάσεων στις οποίες έχει πραγματοποιήσει εγγραφή ο φοιτητής. Στον πίνακα \_\_ παρουσιάζονται τα κύρια δεδομένα των προβλημάτων του συνόλου δεδομένων carter. Για κάθε πρόβλημα διατίθενται τα εξής δεδομένα:

* Αριθμός εξετάσεων.
* Αριθμός φοιτητών.
* Διαθέσιμες χρονικές περίοδοι ανά πρόβλημα.
* Συντελεστής πυκνότητας.

Ως συντελεστής πυκνότητας ορίζεται η πιθανότητα να υπάρχει σύγκρουση μεταξύ δύο εξετάσεων. Σημειώνεται ότι στο πρόβλημα δεν λαμβάνονται περιορισμοί όπως οι αίθουσες στις οποίες θα πραγματοποιηθούν οι εξετάσεις ή ο μέγιστος αριθμός φοιτητών ανά εξέταση καθώς και άλλοι περιορισμοί που συγκεντρώνουν αρκετές πιθανότητες να υφίστανται σε ένα πρόβλημα χρονοπρογραμματισμού εξετάσεων, το οποίο περιλαμβάνει εξετάσεις ενός τμήματος πανεπιστημίου.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ΠΡΟΒΛΗΜΑ | ΕΞΕΤΑΣΕΙΣ | ΦΟΙΤΗΤΕΣ | ΠΕΡΙΟΔΟΙ | ΣΥΝΤΕΛΕΣΤΗΣ ΠΥΚΝΟΤΗΤΑΣ |
| car-f-92 | 543 | 18419 | 32 | 0.14 |
| car-s-91 | 682 | 16925 | 35 | 0.13 |
| ear‐f‐83 | 190 | 1125 | 24 | 0.27 |
| hec‐s‐92 | 81 | 2823 | 18 | 0.42 |
| kfu‐s‐93 | 461 | 5349 | 20 | 0.06 |
| lse‐f‐91 | 381 | 2726 | 18 | 0.06 |
| pur‐s‐93 | 2419 | 30029 | 42 | 0.03 |
| rye‐s‐93 | 486 | 11483 | 23 | 0.07 |
| sta‐f‐83 | 139 | 611 | 13 | 0.14 |
| tre‐s‐92 | 261 | 4360 | 23 | 0.18 |
| uta‐s‐92 | 622 | 21266 | 35 | 0.13 |
| ute‐s‐92 | 184 | 2749 | 10 | 0.08 |
| yor‐f‐83 | 181 | 941 | 21 | 0.29 |

# Τεχνικές επίλυσης προβλήματος

Για την επίλυση του προβλήματος χρησιμοποιήθηκαν δύο τεχνικές. Η πρώτη τεχνική που χρησιμοποιήθηκε είναι η διαδικασία του χρωματισμού γραφών, η οποία αναλαμβάνει τη δημιουργία αρχικής λύσης. Τοποθετώντας τις εξετάσεις του προβλήματος στις διαθέσιμες χρονικές περιόδους, λαμβάνοντας υπόψη τον περιορισμό της διεξαγωγής εξετάσεων στις οποίες συμμετέχουν κοινοί φοιτητές σε διαφορετική περίοδο.

Η διαδικασία που χρησιμοποιήθηκε για την δημιουργία του αρχικού προγράμματος είναι ένας αλγόριθμος άπληστου χρωματισμού, ο DSATUR.

Η διαδικασία του αλγορίθμου εκκινεί με την επιλογή της κορυφής με το μεγαλύτερο βαθμό. Η κορυφή αυτή χρωματίζεται με το πρώτο χρώμα και στη συνέχεια ενημερώνονται οι τιμές κορεσμού των γειτονικών κορυφών. Ως τιμή κορεσμού για μία κορυφή ορίζουμε το πλήθος των κορυφών στις όποιες έχει εφαρμοστεί κάποιο χρώμα στο συγκεκριμένο σημείο του αλγορίθμου. Στη συνέχεια, επιλέγεται η κορυφή με το μικρότερο επίπεδο κορεσμού ανάμεσα από τις γειτονικές κορυφές. Σε αυτήν, ο αλγόριθμος αναζητά το πρώτο διαθέσιμο χρώμα, το οποίο στη συνέχεια χρησιμοποιείται για το χρωματισμό της κορυφής. Αφού πραγματοποιηθεί ο χρωματισμός της κορυφής, ενημερώνονται οι τιμές κορεσμού των γειτονικών κορυφών της. Οι τιμές αυτές αυξάνονται κατά μία μονάδα. Η διαδικασία αυτή εκτελείται επαναληπτικά για όλες τις κορυφές του γραφήματος. Στον πίνακα 1, παρουσιάζονται τα αποτελέσματα χρωματισμού του συνόλου δεδομένων, έπειτα από εφαρμογή του αλγορίθμου DSATUR. [2]

Η δεύτερη τεχνική επίλυσης που χρησιμοποιήθηκε είναι ο αλγόριθμος της προσομοιωμένης ανόπτησης, ο οποίος παρέχει τη δυνατότητα βελτιστοποίησης της αρχικής λύσης. Ο αλγόριθμος επιτρέπει τη δημιουργία εναλλακτικών λύσεων με βάση τη λύση η οποία έχει κατασκευαστεί με τη χρήση του αλγορίθμου DSATUR, οι οποίες αξιολογούνται με βάση τον συνολικό αριθμό περιόδων που χρησιμοποιούν. Σε περίπτωση που κάποια από τις λύσεις χρησιμοποιεί μικρότερο αριθμό περιόδων, αντικαθιστά την αρχική λύση.

Ο αλγόριθμος της προσομοιωμένης ανόπτησης χρησιμοποιεί βασίζεται σε τρείς παραμέτρους. Η βασικότερη παράμετρος, είναι η θερμοκρασία. Όσο η θερμοκρασία κυμαίνεται σε υψηλά επίπεδα τόσο περισσότερες αυξάνονται οι πιθανότητες να πραγματοποιηθεί τροποποίηση στην τρέχουσα λύση. Η δεύτερη παράμετρος, είναι ο συντελεστής μείωσης της θερμοκρασίας με τιμή εύρους 0.8-0.9999, η οποία χρησιμοποιείται για να μεταβάλει τη θερμοκρασία. Επίσης, με βάση μια τυχαία δεκαδική τιμή εύρους 0 έως 1, ο αλγόριθμος αποδέχεται λύσεις οι οποίες χρησιμοποιούν ίδιο ή μεγαλύτερο αριθμό περιόδων. Τέλος, κάθε φορά που η θερμοκρασία προσεγγίζει την τιμή 0.1, πραγματοποιείται επαναρχικοποίηση της θερμοκρασίας σε μια τιμή εύρους από 50 έως 100% της αρχικής θερμοκρασίας. [3]

Ο αλγόριθμος ολοκληρώνεται μετά την ολοκλήρωση ενός συγκεκριμένου χρονικού ορίου που ορίζεται από το χρήστη. Στον πίνακα 2, παρουσιάζονται τα αποτελέσματα του αλγορίθμου για το σύνολο δεδομένων.

# Αποτελέσματα

Πίνακας 1. Αποτελέσματα αλγορίθμου DSATUR

|  |  |
| --- | --- |
| **ΠΡΟΒΛΗΜΑ** | **ΠΕΡΙΟΔΟΙ** |
| car-f-92 | 30 |
| car-s-91 | 31 |
| ear‐f‐83 | 23 |
| hec‐s‐92 | 19 |
| kfu‐s‐93 | 19 |
| lse‐f‐91 | 19 |
| pur‐s‐93 | 35 |
| rye‐s‐93 | 22 |
| sta‐f‐83 | 13 |
| tre‐s‐92 | 23 |
| uta‐s‐92 | 31 |
| ute‐s‐92 | 10 |
| yor‐f‐83 | 20 |

Πίνακας 2. Αποτελέσματα αλγορίθμου Simulated Annealing

|  |  |
| --- | --- |
| **ΠΡΟΒΛΗΜΑ** | **ΠΕΡΙΟΔΟΙ** |
| car-f-92 | 30 |
| car-s-91 | 31 |
| ear‐f‐83 | 23 |
| hec‐s‐92 | 17 |
| kfu‐s‐93 | 19 |
| lse‐f‐91 | 17 |
| pur‐s‐93 | 34 |
| rye‐s‐93 | 21 |
| sta‐f‐83 | 13 |
| tre‐s‐92 | 20 |
| uta‐s‐92 | 31 |
| ute‐s‐92 | 10 |
| yor‐f‐83 | 20 |

# Βιβλιογραφία

1)Hardesty, Larry. “Explained: P vs. NP.” *MIT News | Massachusetts Institute of Technology*, news.mit.edu/2009/explainer-pnp.

2) “DSatur.” *Wikipedia*, Wikimedia Foundation, 8 July 2021, en.wikipedia.org/wiki/DSatur.

3) Brownlee, Jason. “Simulated Annealing From Scratch in Python.” *Machine Learning Mastery*, 28 Feb. 2021, machinelearningmastery.com/simulated-annealing-from-scratch-in-python/.