

# Τεχνική Αναφορά για το Μάθημα της Υπολογιστικής Πολυπλοκότητας

## Παρουσίαση της κατηγορίας προβλημάτων ικανοποίησης περιορισμών

Οι αλγοριθμικές προσεγγίσεις σε *np-hard* προβλήματα με απαιτήσεις ικανοποίησης περιορισμών, στοχεύουν στην κατανομή πόρων με ορισμένους λογικές δεσμεύσεις, αναζητώντας μια μέγιστη εφικτή λύση. Το ενδιαφέρον στα εν λόγω προβλήματα, είναι η πρόκληση εύρεσης μέγιστης ιδανικής λύσης, σε δεδομένο χρόνο και δεδομένους υπολογιστικούς πόρους. Προφανώς οι βασικοί δείκτες της λύσης είναι η επιθυμητή ακρίβεια, η ισοροπία ανάμεσα σε υπολογιστική ακρίβεια της λύσης απο άποψη ικανοποίησης περιορισμών, σε σχέση με τον χρόνο και τους υπολογιστικού πόρους.

Την τελευταία δεκαετία, με την ευρεία εισαγωγή μοντέλων παραλληλου προγραμματισμού, καθώς και την ευρεία χρήση τεχνικών μηχανικής κ βαθιάς μάθησης, οι κλασσικές προσεγγίσεις επίλυσης τέτοιων προβλημάτων, έχουν αναθεωρηθεί, και έχει ανατροφοδοτηθεί η αναζήτηση μόντερνων λύσεων στα κλασσικά προβλήματα.

Βεβαίως, σε κάθε περίπτωση, το βασικό ζητούμενο κάθε προσέγγισης, είναι να πετύχει γραμμική πολυπλοκότητα όσον αφορά και τον χρόνο και τον χώρο(μνήμη). Ακόμη και αν η

γραμματική είναι ανέφικτή, πρέπει πάση θυσία να αποφύγουμε περιπτώσεις πολυωνυμικής πολυπλοκότητας

Το πρόβλημα του προγραμματισμού των εξετάσεων είναι ένα απο αυτά τα προβλήματα, και το ζητούμενο της εργασίας.

Ουσιαστικά, είναι το πρόβλημα της ανάθεσης ενός συνόλου δοκιμών σε καθορισμένο αριθμό δεδομένων χρονικών διαστημάτων και υπό συγκεκριμένους περιορισμούς.

Ο στόχος για την επίλυση του προβλήματος είναι να προγραμματίσετε εξετάσεις για το λιγότερο χρονικό διάστημα χωρίς αλληλεπικάλυψη και να προσπαθήσετε να μεγιστοποιήσετε τις προτεραιότητες, όπως η αύξηση του χρονικού διαστήματος μεταξύ διοίκησης και εκπαιδευτών και οι δύο διαδοχικές εξετάσεις των μαθητών όσο το δυνατόν περισσότερο, όταν ο αριθμός του χρόνου οι περίοδοι καθορίζονται .

Υπάρχουν περιορισμοί που πρέπει να ληφθούν υπόψη για την επίλυση του προβλήματος. Το χρονικό διάστημα που πρέπει να απομείνει μεταξύ των εξετάσεων, η αποφυγή συγκρούσεων, ο μέγιστος αριθμός εξετάσεων που μπορεί να εισαγάγει ένας μαθητής ανά ημέρα κτλ κτλ

Οι περιορισμοί που καθορίζουν το χώρο λύσης και δεν πρέπει να παραβιαστούν και είναι οι λεγόμενοι σκληροί. οι περιορισμοί που μπορούν να παραβιαστούν με μια

συγκεκριμένη τιμωρία και να συμβάλουν στην αντικειμενική λειτουργία ορίζονται ως μαλακοί περιορισμοί .

Ο βαθμός στον οποίο μπορούν να ικανοποιηθούν οι απαλοί περιορισμοί κατά την προετοιμασία των διαγραμμάτων εξέτασης θα αυξήσει την επιτυχία όσον αφορά την ικανοποίηση των απαιτήσεων των γραφημάτων.

Οι σκληροί περιορισμοί που πρέπει να ληφθούν υπόψη στη μελέτη είναι: Πολλαπλές εμφανίσεις του ίδιου μαθήματος στην ίδια περίοδο πχ. Οι μαλακοί περιορισμοί που λαμβάνονται υπόψη στη μελέτη είναι: Η απόσταση μεταξύ περιόδων για τον ίδιο φοιτητή.

### **Τεχνικές Προσέγγισεις με αλγοριθμική βάση**

Οι κατηγορίες προσεγγίσεων , κατα βάσιν με την χρονολογική σειρά που εμφανίστηκαν είναι οι εξής:

**1.Μαθηματικές τεχνικές**

**2.Γραφικός προγραμματισμός**

**3.Ευρετικοί αλγόριθμοι (Heuristics)**

**4.Τεχνητή νοημοσύνη ( A.I )**

**5.Νευρωνικά δίκτυα**

**6.Αλγόριθμος κατωφλίου (Threshold)**

**7.Γενετικοί αλγόριθμοι**

**8.Η έρευνα ταμπού (Tabu search)**

Προφανώς, παρουσιάστηκαν και υβριδικές προσεγγίσεις, που συνδυάζαν, μια η παραπάνω τεχνικές απο διαφορετικές κατηγορίες

### Παρουσίαση του δικού μας αλγόριθμου

Η υλοποίηση μας, έγινε με την γλώσσα προγραμματισμού c.

Λειτουργεί ως εξής:

Παίρνει ένα αρχείο εισαγωγής κειμένου που περιέχει αριθμό φοιτητή και μαθήματα για κάθε άτομο.

Το πρόγραμμα αναλύει το αρχείο εισαγωγής και καθορίζει έναν \* κατάλληλο προγραμματισμό εξετάσεων \* έτσι ώστε για κάθε άτομο καμία εξέταση δεν θα είναι αλληλεπικαλυπτόμενη . Ένα κοινό πρόβλημα στα πανεπιστήμια αφού κανένα άτομο δεν μπορεί να συμμετάσχει σε δύο εξετάσεις ταυτόχρονα .

Χρησιμοποιεί αλγόριθμο χρωματισμού γραφημάτων . Μόλις διαβάσει το αρχείο εισαγωγής, δημιουργεί κορυφές για κάθε μάθημα. Στη συνέχεια, δημιουργεί άκρα μεταξύ κορυφών με βάση τα μαθήματα κάθε ατόμου. Έτσι, εάν μια κορυφή έχει ένα άκρο μεταξύ άλλης κορυφής, τότε αυτές οι κορυφές δεν θα βρίσκονται στο ίδιο πρόγραμμα εξετάσεων, επειδή θα είναι αλληλεπικαλυπτόμενες . Αφού δημιουργήσει τις άκρες, το πρόγραμμα εξάγει έναν \* πίνακα συγκρούσεων \* για να δείξει τις σχέσεις μεταξύ των μαθημάτων. Με βάση αυτόν τον πίνακα γειτνίασης(συγκρούσεων) το πρόγραμμα επισκέπτεται κάθε κορυφή χρησιμοποιώντας τον αλγόριθμο DFS και χρώματα

κάθε κορυφή κατάλληλα. Στη συνέχεια, βάσει αυτών των χρωμάτων, δημιουργεί προγράμματα εξετάσεων .