



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
UNIVERSITY OF WEST ATTICA

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΤΟΥ
ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΧΟΛΙΚΩΝ ΜΟΝΑΔΩΝ ΜΕ
ΧΡΗΣΗ ΤΕΧΝΙΚΩΝ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

Μπολάνο Χρήστος

A.M. 711151092

Εισηγητής: Χρήστος Τρούσσας, Επ. Καθηγητής

Αθήνα 2024

ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΤΟΥ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΧΟΛΙΚΩΝ ΜΟΝΑΔΩΝ ΜΕ ΧΡΗΣΗ ΤΕΧΝΙΚΩΝ
ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΤΟΥ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΧΟΛΙΚΩΝ ΜΟΝΑΔΩΝ ΜΕ ΧΡΗΣΗ ΤΕΧΝΙΚΩΝ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

Μπολάνο Χρήστος

A.M. 711151092

Εισηγητής:

Χρήστος Τρούσσας, Επ. Καθηγητής

Εξεταστική Επιτροπή:

Χρήστος Τρούσσας, Επ. Καθηγητής

Ακριβή Κρούσκα, Μέλος ΕΔΙΠ

Παναγιώτα Τσελέντη, Μέλος ΕΔΙΠ

Ημερομηνία Εξέτασης:

Ιούλιος, 2024

Περίληψη

Η δημιουργία ωρολογίων προγραμμάτων για εκπαιδευτικούς οργανισμούς είναι μια κρίσιμη και δύσκολη διαδικασία που επηρεάζει τόσο την απόδοση των μαθητών όσο και την ευημερία των εκπαιδευτικών. Ένα δίκαιο πρόγραμμα αποτελεί ένδειξη αποτελεσματικής διοίκησης, παρέχοντας ένα κλίμα αρμονίας και συνεργασίας.

Σκοπός αυτής της διπλωματικής εργασίας είναι η ανάπτυξη και αξιολόγηση ενός ολοκληρωμένου συστήματος για τη βέλτιστη παραγωγή των ωρολογίων προγραμμάτων στη δευτεροβάθμια εκπαίδευση. Πρώτος και κύριος στόχος της πλατφόρμας είναι η αυτοματοποιημένη δημιουργία του εβδομαδιαίου προγράμματος μαθημάτων λαμβάνοντας υπόψιν όλες τις παραμέτρους και τους περιορισμούς που μπορεί να έχει ένα εκπαιδευτικό σύστημα δευτεροβάθμιας εκπαίδευσης. Οι διαθέσιμοι καθηγητές, οι αίθουσες, καθώς και οι απαιτήσεις κάθε μαθήματος συμπεριλαμβανομένης της διάρκειας και του αριθμού των φοιτητών που θα συμμετάσχουν αποτελούν μόνο μερικούς από τους παράγοντες που επηρεάζουν το τελικό αποτέλεσμα.

Το βασικότερο κομμάτι αυτής της πλατφόρμας αποτελεί η ανάπτυξη ενός αλγορίθμου που μπορεί να παράξει ένα ωρολόγιο πρόγραμμα σε δευτερόλεπτα, ενσωματώνοντας πολύπλοκες παραμέτρους και περιορισμούς με ακρίβεια και αξιοπιστία.

Στο 1^ο Κεφάλαιο παρουσιάζεται το πεδίο εφαρμογής της διπλωματικής εργασίας, διερευνώντας το πλαίσιο στο οποίο αυτή εστιάζει. Στη συνέχεια, αναλύεται το αντικείμενο της διπλωματικής εργασίας, περιγράφοντας το συγκεκριμένο πρόβλημα που μελετάται και τους στόχους που έχουν τεθεί.

Στο 2^ο Κεφάλαιο παρουσιάζεται το θεωρητικό υπόβαθρο που αφορά το πρόβλημα του χρονοπρογραμματισμού στην εκπαίδευση, το οποίο χαρακτηρίζεται ως ένα NP-δύσκολο πρόβλημα. Αναλύονται τα προβλήματα ικανοποίησης περιορισμών καθώς και οι τεχνικές που χρησιμοποιήθηκαν στην προσπάθεια επίλυσής τους. Τέλος, γίνεται ανασκόπηση της υπάρχουσας βιβλιογραφίας σχετικά με τις τεχνικές και τα εργαλεία που χρησιμοποιούνται για τη δημιουργία ωρολογίων προγραμμάτων.

Στο 3^ο Κεφάλαιο περιγράφεται η μεθοδολογία που ακολουθήθηκε για την ανάπτυξη του συστήματος, παρουσιάζονται οι υπάρχουσες λύσεις και τέλος αναλύονται τα εργαλεία και οι βιβλιοθήκες που βοήθησαν στην ανάπτυξη αυτού του συστήματος.

Στο 4^ο Κεφάλαιο γίνεται η παρουσίαση του συστήματος, εστιάζοντας στις τεχνικές λεπτομέρειες της πλατφόρμας. Αρχικά, αναφέρονται οι ιδιαιτερότητες της ελληνικής δευτεροβάθμιας εκπαίδευσης. Έπειτα, εξηγούνται οι οντότητες του προβλήματος και αναλύεται ο κάθε περιορισμός του προβλήματος με την αντίστοιχη μαθηματική του διατύπωση.

Στο 5^ο Κεφάλαιο προχωράμε με την παρουσίαση του συστήματος, δίνοντας έμφαση στη διαδραστική διεπαφή της πλατφόρμας. Αναλύουμε το περιβάλλον χρήστη, δείχνοντας πώς η οπτική παρουσίαση καθοδηγεί τον χρήστη στη διαδικασία δημιουργίας και διαχείρισης των ωρολογίων προγραμμάτων. Παρουσιάζονται στιγμιότυπα οθόνης που απεικονίζουν τα κύρια χαρακτηριστικά και τις λειτουργίες της πλατφόρμας, όπως η εισαγωγή δεδομένων, η ρύθμιση παραμέτρων και η τελική απεικόνιση των προγραμμάτων. Επιπλέον, εξηγούμε πώς οι χρήστες μπορούν να επεξεργαστούν και να προσαρμόσουν τα προγράμματα ανάλογα με τις ανάγκες τους, διασφαλίζοντας τη βέλτιστη κατανομή πόρων και χρόνου.

Τέλος, στο 6^ο Κεφάλαιο συνοψίζονται τα κύρια ευρήματα της εργασίας και καταγράφονται τα συμπεράσματα.

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη	3
Κατάλογος σχημάτων	6
Κατάλογος εικόνων	7
Συντομογραφίες & Ακρωνύμια	8
Κεφάλαιο 1 Εισαγωγή	9
1.1 Πεδίο Εφαρμογής	9
1.2 Αντικείμενο της Διπλωματικής Εργασίας	10
1.3 Προτεινόμενη Λύση	11
Κεφάλαιο 2 Θεωρητικό υπόβαθρο	12
2.1 Προβλήματα Προγραμματισμού, NP-complete, NP-hard προβλήματα	12
2.2 Πρόβλημα Ικανοποίησης Περιορισμών	13
2.2.1 Ορισμός	13
2.2.2 Πεδίο ορισμού	14
2.2.3 Περιορισμοί	14
2.2.4 Αναζήτηση	14
2.2.5 Πρώιμος Έλεγχος	15
2.3 Ανασκόπηση βιβλιογραφίας	16
Κεφάλαιο 3 Μεθοδολογία	18
3.1 Επιλογή θέματος	18
3.2 Έρευνα	18
3.3 Υπάρχουσες λύσεις	19
3.3.1 Mimoso	19
3.3.2 aSc Timetables	21
3.3.3 Timetabler	22
3.3.4 UniTime	23
3.4 Επιλογή εργαλείου επίλυσης (Solver)	25
3.5 Εργαλεία ανάπτυξης	25
3.5.1 PyCharm	26
3.5.2 PostgreSQL	26
3.5.3 Docker	27
3.5.4 Git	27
Κεφάλαιο 4 Παρουσίαση του συστήματος	28
4.1 Χαρακτηριστικά του Ελληνικού ωρολογίου προγράμματος	28
4.1.1 Τύποι μαθημάτων	28
4.1.2 Διαθεσιμότητα των πόρων	29

4.2 Οντότητες του προβλήματος	29
4.2.1 Η οντότητα του Χρόνου – (Times).....	30
4.2.2 Η οντότητα των πόρων (Resources).....	30
4.2.3 Η Οντότητα γεγονός (Event).....	31
4.3 Μεταβλητές	31
4.4 Περιορισμοί (Constraints)	32
4.4.1 Ανάθεση πόρου (Assign Resource) C1	34
4.4.2 Ανάθεση χρόνου (Assign Time) C2	35
4.4.3 Διαχωρισμός γεγονότων (Split Events) C3	35
4.4.4 Επιθυμητοί πόροι (Prefer Resource) C4	37
4.4.5 Επιθυμητοί χρόνοι (Prefer Times) C5	37
4.4.6 Αποφυγή διαχωρισμού αναθέσεων (Avoid split Assignments) C6	38
4.4.7 Διασπορά γεγονότων (Spread Events) C7	39
4.4.8 Σύνδεσμος Γεγονότων (Link Events) C8	40
4.4.9 Σειρά γεγονότων (Order Events) C9	41
4.4.10 Αποφυγή Συγκρούσεων (Avoid Clashes Constraint) C10	42
4.4.11 Αποφυγή Μη Διαθέσιμων Χρόνων (Avoid unavailable times) C11	43
4.5 Objective function	44
Κεφάλαιο 5 Παραδείγματα χρήσης	45
5.1 Είσοδος.....	45
5.2 Αρχική οθόνη (Home)	46
5.3 Εισαγωγή δεδομένων.....	47
5.3.1 Χρόνοι (Times).....	47
5.3.2 Πόροι (Resource).....	50
5.3.3 Μαθήματα (Events)	54
5.4 Βελτιστοποίηση (Optimization).....	56
5.5 Πρόγραμμα (Timetable)	57
Κεφάλαιο 6 Συμπεράσματα.....	59
Βιβλιογραφία	60

Κατάλογος σχημάτων

Σχήμα 2.1 Κατηγορίες υπολογιστικής πολυπλοκότητας	13
Σχήμα 3.1 Οθόνη εισαγωγής δεδομένων του λογισμικού Mimosa	20
Σχήμα 3.2 Παράδειγμα αποτελέσματος του λογισμικού Mimosa	20
Σχήμα 3.3 Οθόνη εισαγωγής δεδομένων του λογισμικού aSc Timetables	21
Σχήμα 3.4 Παράδειγμα αποτελέσματος του λογισμικού aSc Timetables	22
Σχήμα 3.5 Οθόνη εισαγωγής δεδομένων του λογισμικού Timetabler	23
Σχήμα 3.6 Παράδειγμα αποτελέσματος του λογισμικού Timetabler	23
Σχήμα 3.7 Οθόνη εισαγωγής δεδομένων του λογισμικού UniTime	24
Σχήμα 3.8 Παράδειγμα αποτελέσματος του λογισμικού UniTime	24
Σχήμα 4.1 Παράδειγμα διαχωρισμού γεγονότων	36
Σχήμα 4.2 Παράδειγμα περιορισμού Επιθυμητοί χρόνοι	38
Σχήμα 4.3 Παράδειγμα περιορισμού Αποφυγή διαχωρισμού αναθέσεων	38
Σχήμα 4.4 Παράδειγμα περιορισμού Διασπορά γεγονότων	39
Σχήμα 4.5 Παράδειγμα περιορισμού Σύνδεσμος Γεγονότων	40
Σχήμα 4.6 Παράδειγμα περιορισμού Αποφυγή Συγκρούσεων	42
Σχήμα 4.7 Παράδειγμα περιορισμού Αποφυγή Μη Διαθέσιμων Χρόνων	43
Σχήμα 5.1 Παράδειγμα Ομάδας Χρόνων	48
Σχήμα 5.2 Παράδειγμα Ομάδας Πόρων 1	50
Σχήμα 5.3 Παράδειγμα Ομάδας Πόρων 2	50

Κατάλογος εικόνων

Εικόνα 1 Οθόνη Εγγραφής Χρήστη	45
Εικόνα 2 Οθόνη Εισόδου Χρήστη	45
Εικόνα 3 Αρχική Οθόνη	46
Εικόνα 4 Φόρμα Generate Times	47
Εικόνα 5 Φόρμα επεξεργασίας Ομάδας Χρόνων.....	48
Εικόνα 6 Φόρμα επεξεργασίας Χρόνου.....	49
Εικόνα 7 Φόρμα επεξεργασίας Ομάδας Πόρων	51
Εικόνα 8 Φόρμα Επεξεργασίας Πόρων	51
Εικόνα 9 Φόρμα επεξεργασίας Αίθουσας.....	52
Εικόνα 10 Φόρμα εισαγωγής Μη Διαθέσιμων Χρόνων για ένα Πόρο	53
Εικόνα 11 Φόρμα εισαγωγής Περιορισμός χρόνων απασχόλησης.....	53
Εικόνα 12 Φόρμα εισαγωγής Περιορισμός Χρόνου αδράνειας.....	53
Εικόνα 13 Φόρμα επεξεργασίας Ομάδας Γεγονότων	54
Εικόνα 14 Φόρμα Επεξεργασίας Γεγονότος.....	55
Εικόνα 15 Φόρμα εισαγωγείς Επιθυμητών χρόνων.....	55
Εικόνα 16 Φόρμα εισαγωγείς Επιθυμητών πόρων	56
Εικόνα 17 Παράθυρο διαδικασίας βελτιστοποίησης	57
Εικόνα 18 Παράδειγμα προγράμματος για το 4ο Εξάμηνο	58

Συντομογραφίες & Ακρωνύμια

AI - Artificial Intelligence
CGH - Column Generation Heuristic
CI - Computational Intelligence
CPU – Central Processing Unit
CP-Sat - Constraint Programming Satisfiability
CSP - Constraint Satisfaction Problems
CSOP - Constraint Satisfaction Optimization Problems
CSV – Comma Separated Value
IDE – Integrated Development Environments
MIP – Mixed Integer Programming
IP – Integer Programming
MP – Mixed Integer Programming
MILP – Mixed Integer Linear Programming
TSA – Two Stage Approach
CGH - Column Generation Heuristic
XHSTT - High School Timetable
XML - Extensible Markup Language

Κεφάλαιο 1 Εισαγωγή

Στο κεφάλαιο αυτό αναλύεται το πεδίο εφαρμογής του θέματος της παρούσας Διπλωματικής Εργασίας, το αντικείμενό της, καθώς και η δομή της. Στην ενότητα 1.1 τονίζεται η σημασία της ύπαρξης ενός ωρολογίου προγράμματος σε κάθε οργανισμό που απασχολεί προσωπικό, καθώς και το πόσο περίπλοκη είναι η δημιουργία ενός τέτοιου προγράμματος από έναν άνθρωπο. Στην ενότητα 1.2 περιγράφεται το αντικείμενο της παρούσας Διπλωματικής Εργασίας, το οποίο είναι η σχεδίαση και υλοποίηση ενός αλγόριθμου Υπολογιστικής Νοημοσύνης που θα αυτοματοποιεί τη δημιουργία ωρολογίων προγραμμάτων για σχολεία με τη χρήση Ηλεκτρονικού Υπολογιστή. Τέλος, στην ενότητα 1.3 παρουσιάζεται η προτεινομένη λύση.

1.1 Πεδίο Εφαρμογής

Η οργάνωση ενός χρονοδιαγράμματος για τον καθορισμό των ωρών απασχόλησης του προσωπικού αποτελεί έναν κρίσιμο παράγοντα για την εύρυθμη λειτουργία κάθε οργανισμού. Η αποτελεσματική διαχείριση του ανθρώπινου δυναμικού και η βέλτιστη κατανομή των εργασιών είναι αναγκαία για να διασφαλιστεί ότι οι καθημερινές δραστηριότητες θα εκτελούνται ομαλά και αποδοτικά.

Παραδείγματα τέτοιων οργανισμών είναι τα νοσοκομεία και τα σχολεία. Στα νοσοκομεία, για παράδειγμα, η δημιουργία ενός χρονοδιαγράμματος εργασίας είναι ιδιαίτερα περίπλοκη λόγω της ανάγκης για 24ωρη κάλυψη των ασθενών. Οι βάρδιες των γιατρών και των νοσηλευτών πρέπει να οργανώνονται με τέτοιο τρόπο ώστε να διασφαλίζεται ότι υπάρχει πάντα διαθέσιμο προσωπικό για την αντιμετώπιση των αναγκών των ασθενών. Επιπλέον, πρέπει να λαμβάνονται υπόψη οι ειδικές ανάγκες κάθε τμήματος, όπως η εντατική θεραπεία ή το τμήμα επειγόντων περιστατικών, καθώς και οι προσωπικές ανάγκες και προτιμήσεις του προσωπικού, όπως οι άδειες και οι ώρες ανάπαυσης.

Στα σχολεία, η οργάνωση των ωρών διδασκαλίας των εκπαιδευτικών και των διαλειμμάτων των μαθητών είναι εξίσου πολύπλοκη. Κάθε εκπαιδευτικός έχει ένα συγκεκριμένο ωράριο και ειδικότητες που πρέπει να καλυφθούν, ενώ παράλληλα πρέπει να εξασφαλίζεται ότι οι μαθητές έχουν επαρκή χρόνο για μάθηση και ξεκούραση. Επιπλέον, οι απαιτήσεις για ειδικά μαθήματα ή δραστηριότητες, όπως τα εργαστήρια προσθέτουν επιπλέον πολυπλοκότητα στη δημιουργία ενός αποτελεσματικού χρονοδιαγράμματος.

Η παραδοσιακή μέθοδος κατασκευής τέτοιων προγραμμάτων με το χέρι ήταν χρονοβόρα και επιρρεπής σε λάθη. Συχνά απαιτούσε μεγάλες ώρες εργασίας και επαναλαμβανόμενες διορθώσεις, γεγονός που μείωνε την αποδοτικότητα και την ακρίβεια των χρονοδιαγραμμάτων. Ένα ανθρώπινο λάθος θα μπορούσε να οδηγήσει σε σοβαρά προβλήματα, όπως η ανεπαρκής κάλυψη των βαρδιών στα νοσοκομεία ή η υπερφόρτωση των μαθημάτων για κάποιους εκπαιδευτικούς στα σχολεία. Αυτές οι ανακρίβειες μπορούσαν να επηρεάσουν αρνητικά τη λειτουργία του οργανισμού και να δημιουργήσουν άγχος και δυσαρέσκεια μεταξύ του προσωπικού.

Ωστόσο, με την πρόοδο της πληροφορικής, η αυτοματοποίηση της διαδικασίας μέσω κατάλληλου λογισμικού και αλγορίθμων έχει βελτιώσει σημαντικά την αποτελεσματικότητα και την ακρίβεια. Η χρήση εξειδικευμένου λογισμικού μπορεί να υπολογίσει και να βελτιστοποιήσει τους παράγοντες που εμπλέκονται, μειώνοντας το χρόνο και τις πιθανότητες σφάλματος. Αυτά τα συστήματα είναι σε θέση να διαχειρίζονται μεγάλες ποσότητες δεδομένων και να παρέχουν γρήγορα λύσεις που είναι βέλτιστες για τον οργανισμό. Το λογισμικό μπορεί να λαμβάνει υπόψη του πολυάριθμους παράγοντες, όπως τις προτιμήσεις του προσωπικού, τους διαθέσιμους πόρους, τις απαιτήσεις του οργανισμού και τις νομικές υποχρεώσεις.

Η κατασκευή ενός ωρολογίου προγράμματος για έναν οργανισμό αποτελεί ένα πρόβλημα βελτιστοποίησης, καθώς επιδιώκεται η εξεύρεση βέλτιστων λύσεων που θα πληρούν συγκεκριμένους περιορισμούς. Αυτό το είδος προβλήματος μελετάται από την Επιχειρησιακή Έρευνα (Operational Research – OR), έναν κλάδο των Μαθηματικών που εφαρμόζει προηγμένες αναλυτικές μεθόδους για τη λήψη βέλτιστων αποφάσεων. Συναφείς

επιστημονικοί κλάδοι είναι η Τεχνητή Νοημοσύνη (Artificial Intelligence – AI) και η Υπολογιστική Νοημοσύνη (Computational Intelligence – CI).

Η τεχνητή νοημοσύνη και η υπολογιστική νοημοσύνη προσφέρουν ακόμα πιο προχωρημένες λύσεις που διευκολύνουν την επίλυση αυτών των προβλημάτων βελτιστοποίησης. Μέσω της ανάλυσης δεδομένων και της χρήσης αλγορίθμων μηχανικής μάθησης, επιτρέπεται η αυτοματοποιημένη δημιουργία ποιοτικών χρονοδιαγραμμάτων που ανταποκρίνονται στις συγκεκριμένες ανάγκες κάθε οργανισμού. Αυτές οι τεχνολογίες μπορούν να αναγνωρίσουν πρότυπα, να προβλέψουν μελλοντικές ανάγκες και να προσαρμόσουν τα χρονοδιαγράμματα σε πραγματικό χρόνο, εξασφαλίζοντας έτσι την βέλτιστη αξιοποίηση του προσωπικού και των πόρων. Η χρήση της τεχνητής νοημοσύνης επιτρέπει επίσης την προσαρμογή των χρονοδιαγραμμάτων σε απρόβλεπτες αλλαγές, όπως η απουσία ενός υπαλλήλου ή μια έκτακτη ανάγκη, με ελάχιστο χρόνο και κόπο.

Εν κατακλείδι, η σύγχρονη τεχνολογία έχει φέρει επανάσταση στη διαδικασία δημιουργίας χρονοδιαγραμμάτων, κάνοντάς την πιο αποτελεσματική, ακριβή και προσαρμόσιμη στις σύνθετες απαιτήσεις των διαφόρων οργανισμών. Η αυτοματοποίηση και η χρήση της τεχνητής νοημοσύνης προσφέρουν εργαλεία που βελτιώνουν την καθημερινή λειτουργία των οργανισμών, επιτρέποντάς τους να επικεντρωθούν περισσότερο στην παροχή ποιοτικών υπηρεσιών και λιγότερο στη διαχείριση των χρονοδιαγραμμάτων.

1.2 Αντικείμενο της Διπλωματικής Εργασίας

Η κατασκευή ενός προγράμματος που ικανοποιεί όλους τους λειτουργικούς κανόνες και ανάγκες σε ένα εκπαιδευτικό οργανισμό, ταυτόχρονα εκπληρώνοντας όσο το δυνατόν περισσότερες από τις επιθυμίες και τις απαιτήσεις του προσωπικού και των φοιτητών, αποτελεί μια σημαντική αλλά εξαιρετικά δύσκολη εργασία για το εμπλεκόμενο προσωπικό. Σε περισσότερα ιδρύματα, αυτή η εργασία ανατίθεται στο διοικητικό προσωπικό, και η τρέχουσα πρακτική είναι να αντιγράφονται τα προγράμματα των προηγούμενων ετών με μικρές αλλαγές για να προσαρμοστούν σε νέες καταστάσεις. Ωστόσο, στην περίπτωση που οι αλλαγές συμβαίνουν συχνά, η επιδιόρθωση του προηγούμενου προγράμματος δεν είναι πάντα η καλύτερη πολιτική. Υπό αυτές τις συνθήκες και με βάση την πρόοδο που έχει επιτευχθεί στις τεχνολογίες υλικού και λογισμικού, η επιστημονική κοινότητα συνεχίζει να εργάζεται πάνω στο πρόβλημα με σκοπό την ανάπτυξη τυποποιημένων και αυτοματοποιημένων διαδικασιών για τη δημιουργία αποτελεσματικών και επιθυμητών προγραμμάτων.

Καταλαβαίνουμε λοιπόν, ότι δημιουργία ενός αποτελεσματικού ωρολογίου προγράμματος είναι μια πραγματική πρόκληση για κάθε οργανισμό και ειδικά για το προσωπικό που αναλαμβάνει αυτήν την αποστολή. Οι πολλές παράμετροι που πρέπει να ληφθούν υπόψη, μαζί με τους διάφορους περιορισμούς, καθιστούν τη διαδικασία περίπλοκη και χρονοβόρα. Η χρήση τεχνολογικών εργαλείων, όπως ο Ηλεκτρονικός Υπολογιστής, μπορεί να απλοποιήσει αυτήν τη διαδικασία, εφόσον εφαρμοστεί ένας κατάλληλος αλγόριθμος που λαμβάνει υπόψη τους όλους τους παράγοντες. Η ποιότητα των προγραμμάτων που παράγονται μπορεί να αξιολογηθεί με αντικειμενικότητα, επιτρέποντας τη σύγκριση της αποτελεσματικότητας διαφόρων αλγορίθμων.

Στη διεθνή βιβλιογραφία υπάρχουν αρκετοί δημοσιευμένοι αλγόριθμοι [1], οι οποίοι επιλύουν το πρόβλημα της κατασκευής ωρολογίων προγραμμάτων για σχολεία.

Το Αντικείμενο της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η ανάπτυξη μιας ολοκληρωμένης λύσης για την αυτόματη παραγωγή ενός έγκυρου, εφικτού και ποιοτικού ωρολογίου προγράμματος για το πρόβλημα ανάθεσης μαθημάτων (Course Timetabling) για χρήση από τις Πανεπιστημιακές και Πολυτεχνικές σχολές της Ελλάδας. Πιο συγκεκριμένα, υλοποιήθηκε ένας προχωρημένος αλγόριθμος που βασίζεται στη φιλοσοφία του επιλυτή CP-Sat Solver της Google ο οποίος πλαισιώθηκε από μια μοντέρνα διαδικτυακή εφαρμογή με στόχο να απλοποιήσει ο τρόπος με τον οποίο μπορούν να προγραμματιστούν μαθήματα.

1.3 Προτεινόμενη Λύση

Για την αντιμετώπιση των προβλημάτων που αναφέρθηκαν, προτείνεται η ανάπτυξη ενός πληροφοριακού συστήματος για τη διαχείριση και παραγωγή του ωρολογίου προγράμματος στο ακαδημαϊκό περιβάλλον. Το σύστημα αυτό θα παρέχει μιας ολοκληρωμένης λύσης για την αυτόματη παραγωγή ενός έγκυρου, εφικτού και ποιοτικού ωρολογίου προγράμματος για το πρόβλημα ανάθεσης μαθημάτων (Course Timetabling) για χρήση από τις Πανεπιστημιακές και Πολυτεχνικές σχολές της Ελλάδας. Οι διαχειριστές θα πρέπει εκτός από το να παράγουν σε γρήγορο χρόνο ένα ποιητικό πρόγραμμα, να μπορούν πολύ ευκολά να τροποποιήσουν το ενεργό πρόγραμμα της εκάστοτε σχολής ανάλογα με τις ανάγκες ή και τις απαιτήσεις που προκύπτουν κατά τη διάρκεια της ακαδημαϊκής χρονιάς. Κάθε μάθημα θα περιλαμβάνει πληροφορίες όπως η ώρα και η αίθουσα διδασκαλίας, ο καθηγητής, τις προαπαιτούμενες μαθητικές γνώσεις και η διαθεσιμότητα για τους φοιτητές. Για το σύστημα αυτό θα πρέπει να υλοποιηθεί ένας προχωρημένος αλγόριθμος που θα βασίζεται σε έναν μοντέρνο επιλυτή (Solver) ώστε να μπορεί να διαχειριστεί των μεγάλο αριθμό των απαιτήσεων ενός ελληνικού πανεπιστημίου.

Με την υιοθέτηση του προτεινόμενου συστήματος, αναμένεται η εξέλιξη της διαδικασίας της σύνταξης και διαχείρισης του ωρολογίου προγράμματος. Η αυτοματοποίηση των διαδικασιών θα οδηγήσει σε μείωση του χρόνου και των πόρων που απαιτούνται, ενώ θα επιτρέπει την ακριβέστερη και αποτελεσματικότερη διαχείριση των διδακτικών δραστηριοτήτων. Τέλος, η διαδικτυακή πρόσβαση στο ωρολόγιο πρόγραμμα θα διευκολύνει την ενημέρωση και την πρόσβαση των ενδιαφερομένων μερών, συμβάλλοντας έτσι στη βελτίωση της επικοινωνίας και της οργάνωσης στο πλαίσιο του ακαδημαϊκού περιβάλλοντος.

Κεφάλαιο 2 Θεωρητικό υπόβαθρο

2.1 Προβλήματα Προγραμματισμού, NP-complete, NP-hard προβλήματα

Η δημιουργία ενός αποδοτικού ωρολογίου προγράμματος αποτελεί ένα πολύπλοκο πρόβλημα που απαιτεί την αντιστοίχιση διαφόρων πόρων, όπως ανθρώπινο δυναμικό, χώρο και εξοπλισμό, σε συγκεκριμένες χρονικές περιόδους. Σε κάθε περίπτωση, πρέπει να ληφθούν υπόψη διάφοροι περιορισμοί, όπως οι ανάγκες των χρηστών, οι διαθέσιμοι πόροι, οι προτεραιότητες και οι περιορισμοί που προκύπτουν από τον ίδιο τον οργανισμό.

Η κλάση **P** περιλαμβάνει το σύνολο των προβλημάτων που μπορούν να επιλυθούν και να επιβεβαιωθούν σε πολυωνυμικό χρόνο. Αυτά τα προβλήματα μπορούν να λυθούν σε χρόνο $O(n^k)$ στη χειρότερη περίπτωση, όπου n είναι το μέγεθος της εισόδου και k μια σταθερά. Ενώ η κλάση προβλημάτων **P** αποτελεί υποσύνολο της κλάσης **NP**, καθώς εάν μπορούμε να λύσουμε ένα πρόβλημα σε πολυωνυμικό χρόνο, μπορούμε επίσης να επιβεβαιώσουμε μια προτεινόμενη λύση σε πολυωνυμικό χρόνο.

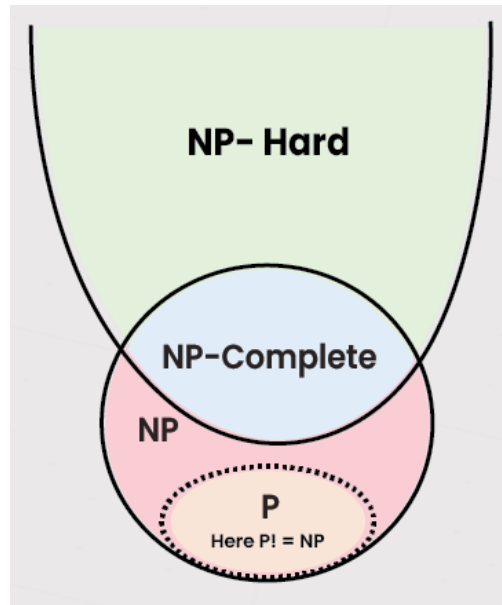
Η κλάση **NP** περιλαμβάνει προβλήματα που μπορούν να επιβεβαιωθούν σε πολυωνυμικό χρόνο $O(n^k)$, δεδομένης μιας λύσης τους. Ένα παράδειγμα προβλήματος σε αυτήν την κατηγορία είναι το Sudoku, όπου η επιβεβαίωση μιας λύσης σε ένα ημι-συμπληρωμένο ταμπλό μπορεί να γίνει σε πολυωνυμικό χρόνο. Επίσης, το πρόβλημα εύρεσης του αθροίσματος των στοιχείων ενός υποσυνόλου ανήκει στην ίδια κατηγορία. Στη γενική του διατύπωση, αυτό το πρόβλημα αφορά την εύρεση ενός συγκεκριμένου αθροίσματος από ένα σύνολο ακέραιων αριθμών. Αν και δεν υπάρχει αλγόριθμος επίλυσης σε πολυωνυμικό χρόνο, μπορούμε να επιβεβαιώσουμε την ορθότητα μιας προτεινόμενης λύσης σε πολυωνυμικό χρόνο.

Ένα πρόβλημα χαρακτηρίζεται ως **NP-Hard** όταν κάθε πρόβλημα **L** της κλάσης **NP** μπορεί να μετασχηματιστεί σε ένα πολυωνυμικό πρόβλημα **H**. Σε απλούστερους όρους, ένα πρόβλημα θεωρείται **NP-Hard** εάν τουλάχιστον είναι τόσο δύσκολο όσο τα πιο δύσκολα προβλήματα της κλάσης **NP**. Μια συχνή παρεξήγηση είναι ότι το **NP** στο **NP-Hard** αντιστοιχεί στην έννοια του "μη πολυωνυμικού", ενώ στην πραγματικότητα αναφέρεται στον όρο "μη ντετερμινιστικών πολυωνυμικών προβλημάτων αποδοχής". Επιπλέον, η υπόθεση ότι δεν υπάρχουν πολυωνυμικοί αλγόριθμοι για την επίλυση προβλημάτων της κλάσης **NP-Hard** δεν έχει αποδειχτεί.

Η κλάση **NP-Complete** περιλαμβάνει τα προβλήματα για τα οποία η κατάστασή τους παραμένει άγνωστη, καθώς δεν έχει ανακαλυφθεί κανένας αλγόριθμος επίλυσης σε πολυωνυμικό χρόνο. Τα **NP-Complete** προβλήματα είναι επίσης **NP-Hard**, δηλαδή μπορούν να μετασχηματιστούν σε άλλα **NP-Complete** προβλήματα σε πολυωνυμικό χρόνο. Στην ουσία, αν ένα πρόβλημα ανήκει στην κατηγορία **NP-Complete**, αυτό σημαίνει ότι μπορεί να επιβεβαιωθεί σε πολυωνυμικό χρόνο, ενώ επίσης εμπίπτει στην κατηγορία **NP**. Η κλάση **NP-Complete** αποτελεί την τομή των κλάσεων **NP** και **NP-Hard**. Εάν ένα πρόβλημα **NP-Complete** μπορεί να επιλυθεί σε πολυωνυμικό χρόνο, τότε μπορεί επίσης να επιλυθούν σε πολυωνυμικό χρόνο και τα υπόλοιπα προβλήματα της κλάσης **NP-Complete**.

Τα προβλήματα Προγραμματισμού (scheduling ή timetabling problems), στη γενική τους μορφή, ανήκουν στην κατηγορία **NP-complete** [2], πράγμα που σημαίνει ότι δεν υπάρχουν γνωστοί αλγόριθμοι πολυωνυμικού χρόνου που να μπορούν να τα επιλύσουν αποτελεσματικά. Η δυσκολία στην εύρεση μιας βέλτιστης λύσης αυξάνεται δραματικά όσο αυξάνεται το μέγεθος του προβλήματος, καθιστώντας την επίλυσή τους με μη ντετερμινιστικούς αλγορίθμους μια πρακτική ανάγκη. Αυτή η προσέγγιση προσφέρει πρακτικές λύσεις που μπορούν να υπολογιστούν σε λογικό χρόνο, παρά την εκθετική δυσκολία του προβλήματος.

Στο Σχήμα 2.1 παρουσιάζονται οι παραπάνω οι κατηγορίες μαζί με κάποια παραδείγματα προβλημάτων που εντάσσονται σε αυτές.



Σχήμα 2.1 Κατηγορίες υπολογιστικής πολυπλοκότητας

2.2 Πρόβλημα Ικανοποίησης Περιορισμών

2.2.1 Ορισμός

Πρόβλημα ικανοποίησης περιορισμών (CSP) ορίζεται ένα μαθηματικό πρόβλημα, που αποτελείται από ένα σύνολο μεταβλητών, των οποίων οι αναθέσεις οφείλουν να ικανοποιούν ένα σύνολο καθορισμένων περιορισμών.

Πιο συγκεκριμένα, ένα CSP είναι μία τριάδα (V, D, C) , όπου:

- $V = \{v_1, \dots, v_n\}$ είναι ένα σύνολο μεταβλητών απόφασης
- $D = \{D_1, \dots, D_n\}$ είναι ένα πεδίο τιμών για τις αντίστοιχες μεταβλητές V
- $C = \{C_1, \dots, C_m\}$ είναι ένα σύνολο περιορισμών

Κάθε μεταβλητή v_i μπορεί να πάρει τιμές από πεδίο ορισμού της D_i . Κάθε περιορισμός C_i αναφέρεται σε κάποιο υποσύνολο των μεταβλητών V και καθορίζει τους επιτρεπτούς συνδυασμούς τιμών γι' αυτό το υποσύνολο. Μια κατάσταση του προβλήματος ορίζεται με ανάθεση τιμών σε μερικές ή όλες τις μεταβλητές $\{v_i = x_i, \dots, v_j = x_j\}$. Μια ανάθεση τιμής που δεν παραβιάζει κανέναν περιορισμό ονομάζεται συνεπής ή νόμιμη ανάθεση. Πλήρης ανάθεση είναι μια ανάθεση που περιλαμβάνει όλες τις μεταβλητές, και λύση ενός προβλήματος ικανοποίησης περιορισμών είναι μια πλήρης ανάθεση τιμών που ικανοποιεί όλους τους περιορισμούς. Σε πολλές περιπτώσεις επίσης χρησιμοποιείται και μία Αντικειμενική Συνάρτηση (Objective Function) στις μεταβλητές του προβλήματος και δηλώνει την ποιότητα της λύσης. Στις περιπτώσεις αυτές σκοπός είναι να βρεθεί μια ανάθεση τιμών στις μεταβλητές που να ελαχιστοποιούν ή να μεγιστοποιούν την αντικειμενική συνάρτηση. Τέτοια προβλήματα αναφέρονται γενικά ως Προβλήματα Βελτιστοποίησης Ικανοποίησης Περιορισμών (Constraint Satisfaction Optimization Problems ή CSOP).

2.2.2 Πεδίο ορισμού

Το πιο συνηθισμένο είδος προβλημάτων ικανοποίησης περιορισμών χρησιμοποιεί μεταβλητές που είναι **διακριτές** (discrete) και έχουν πεπερασμένα πεδία ορισμού (finite domains). Ως τέτοια προβλήματα μπορούν να θεωρηθούν το πρόβλημα χρωματισμού χάρτη ή το πρόβλημα των 8 βασιλισσών. Αν το μέγιστο πεδίο ορισμού οποιασδήποτε μεταβλητής σε ένα πρόβλημα ικανοποίησης περιορισμών είναι d , τότε ο αριθμός των δυνατών πλήρων αναθέσεων τιμών είναι $O(d^n)$ – δηλαδή εκθετικός ως προς τον αριθμό των μεταβλητών.

Οι μεταβλητές μπορούν επίσης να έχουν **άπειρα** πεδία ορισμού (infinite domains) για παράδειγμα, το σύνολο των ακεραίων ή το σύνολο των συμβολοσειρών. Ένα παράδειγμα αποτελεί ο χρονοπρογραμματισμός οικοδομικών εργασιών σε ένα ημερολόγιο: η ημερομηνία έναρξης μιας εργασίας είναι μια μεταβλητή και οι δυνατές τιμές της είναι ακέραιοι αριθμοί ημερών από την τρέχουσα ημερομηνία. Με τα άπειρα πεδία ορισμού, δεν είναι δυνατόν να περιγράφονται οι περιορισμοί με απαρίθμηση όλων των επιτρεπτών συνδυασμών τιμών.

Τέλος, Τα προβλήματα ικανοποίησης περιορισμών με **συνεχή** πεδία ορισμού (continuous domains) είναι πολύ συνηθισμένα στον πραγματικό κόσμο και μελετώνται εκτεταμένα στο πεδίο της επιχειρησιακής έρευνας. Η γνωστότερη κατηγορία προβλημάτων ικανοποίησης περιορισμών με συνεχή πεδία ορισμού είναι τα προβλήματα γραμμικού προγραμματισμού (linear programming), όπου οι περιορισμοί πρέπει να είναι γραμμικές ανισότητες που σχηματίζουν μια κυρτή περιοχή. Τα προβλήματα γραμμικού προγραμματισμού μπορούν να λύνονται σε χρόνο πολυωνυμικό ως προς τον αριθμό των μεταβλητών.

2.2.3 Περιορισμοί

Στην παράγραφο αυτή, εξετάζονται τα είδη των περιορισμών που μπορούμε να συναντήσουμε στα προβλήματα ικανοποίησης περιορισμών.

Ο απλούστερος τύπος περιορισμού που υπάρχει είναι ο **μοναδιαίος περιορισμός** (unary constraint), ο οποίος περιορίζει την τιμή μίας μοναδικής μεταβλητής. Όλοι οι μοναδιαίοι περιορισμοί μπορούν να εξαλειφθούν κάνοντας μια προ-επεξεργασία στο πεδίο ορισμού της αντίστοιχης μεταβλητής προκειμένου να αφαιρέσουμε τις τιμές εκείνες που παραβιάζουν τον περιορισμό. Το επόμενο είδος περιορισμού είναι ο **δυναδικός περιορισμός** (binary constraint) και σχετίζεται με τις τιμές δυο μεταβλητών, για παράδειγμα $v_1 = v_2$. Τέλος, μπορούμε να έχουμε και **μεγαλύτερης τάξης περιορισμούς** (higher-order constraints), οι οποίοι αφορούν τρεις ή περισσότερες μεταβλητές.

Όλοι οι παραπάνω περιορισμοί ονομάζονται **απόλυτοι** (Hard) περιορισμοί, η παραβίαση των οποίων αποκλείει κάθε δυνατή λύση. Οι απόλυτοι περιορισμοί είναι ουσιαστικές συνθήκες που πρέπει να ικανοποιηθούν για να θεωρηθεί μια λύση έγκυρη. Εάν κάποιος σκληρός περιορισμός παραβιαστεί, η λύση θεωρείται μη εφικτή (infeasible). Ένας τέτοιος περιορισμός στο δικό μας παράδειγμα θα ήταν το γεγονός ότι μαθήματα του ίδιου εξάμηνου δεν επιτρέπεται να χρονοπρογραμματισμού την ίδια χρονική στιγμή (no-overlapping). Πολλές φορές όμως στον πραγματικό κόσμο υπάρχει ανάγκη για **ελαστικούς** (soft) περιορισμούς οι οποίοι υποδεικνύουν τις προτιμητέες λύσεις. Σε αντίθεση με τους απόλυτους περιορισμούς, οι ελαστικοί μπορούν να παραβιαστούν και η λύση να συνεχίσει να είναι έγκυρη. Για παράδειγμα στο παρών πρόβλημα κατάρτισης ωρολογίου προγράμματος, ένας ελαστικός περιορισμός θα μπορούσε να είναι αποφυγή των «δύσκολων μαθημάτων» τις πολύ πρωινές ώρες.

2.2.4 Αναζήτηση

Αναζήτηση με υπαναχώρηση. Ένας αλγόριθμος υπαναχώρησης είναι ένας βασικός, συστηματικός αλγόριθμος αναζήτησης για την επίλυση προβλημάτων ικανοποίησης περιορισμών. Πρόκειται για μια αναζήτηση πρώτα κατά βάθος που επιλέγει τιμές για μια μόνο μεταβλητή τη φορά και υπαναχωρεί όταν μια μεταβλητή δεν έχει άλλες νόμιμες τιμές που μπορούν να τις ανατεθούν. Κατασκευάζει μια ανάθεση τιμών σε ένα υποσύνολο μεταβλητών που

ικανοποιεί όλους τους περιορισμούς στις μεταβλητές του συνόλου αυτού. Αυτή η ανάθεση ονομάζεται μερική ανάθεση. Μια μερική ανάθεση επεκτείνεται προσθέτοντας νέες μεταβλητές μια προς μια, μέχρι να γίνει μια ολοκληρωμένη λύση. Όταν για μια μεταβλητή, καμία τιμή δεν ικανοποιεί όλους τους περιορισμούς με τη μερική λύση, η τιμή για την πιο πρόσφατα προστιθέμενη μεταβλητή στη μερική λύση αλλάζει. Αυτή η ενέργεια ονομάζεται υπαναχώρηση (backtracking). Παρόλο που η αναζήτηση με υπαναχώρηση είναι ένας απλός αλγόριθμος αναζήτησης πρώτα κατά βάθος, πολλά ζητήματα πρέπει να ληφθούν υπόψη για τη βελτίωση της αποτελεσματικότητας. Για παράδειγμα, η σειρά της επιλογής μεταβλητών και τιμών επηρεάζει αρκετά την αποτελεσματικότητα των αλγορίθμων.

Η αναζήτηση με υπαναχώρηση έχει μια δεδομένη πολιτική για το τι να κάνει όταν ένας κλάδος αναζήτησης αποτύχει: υποχωρεί στην προηγούμενη μεταβλητή και δοκιμάζει να της δώσει διαφορετική τιμή. Αυτό ονομάζεται χρονολογική υπαναχώρηση (chronological backtracking), επειδή επανέρχεται στο πιο πρόσφατο σημείο απόφασης.

Μια πιο έξυπνη αντιμετώπιση της υπαναχώρησης είναι να γυρίζουμε κατευθείαν πίσω σε μια από το σύνολο μεταβλητών που προκάλεσαν την αποτυχία. Το σύνολο αυτό ονομάζεται σύνολο συγκρούσεων. Γενικά, το σύνολο συγκρούσεων για μια μεταβλητή X είναι το σύνολο των μεταβλητών στις οποίες έχουν προηγουμένως ανατεθεί τιμές και οι οποίες συνδέονται με τη X μέσω περιορισμών. Η μέθοδος της υπαναχώρησης με άλμα (back jumping) υποχωρεί στην πιο πρόσφατη μεταβλητή του συνόλου συγκρούσεων. Αυτό μπορεί να υλοποιηθεί εύκολα με τροποποίηση της αναζήτησης με υπαναχώρηση έτσι ώστε να συσσωρεύει το σύνολο συγκρούσεων ενώ ελέγχει για μια νόμιμη τιμή για να την αποδώσει. Αν δε βρει καμία νόμιμη τιμή, θα πρέπει να επιστρέψει το πιο πρόσφατο στοιχείο του συνόλου συγκρούσεων μαζί με την ένδειξη αποτυχίας

2.2.5 Πρώμος Έλεγχος

Η σειρά επιλογής των μεταβλητών μπορεί να επηρεάσει αρκετά την αποτελεσματικότητα του αλγορίθμου. Μια κοινή τακτική για την επιλογή της μεταβλητής που θα προστεθεί στη μερική λύση είναι η επιλογή της μεταβλητής που περιέχεται στους περισσότερους περιορισμούς, η αλλαγή της οποίας είναι περισσότερο πιθανό να οδηγήσει σε αδιέξοδο. Αυτός ο ευρετικός μηχανισμός ονομάζεται αρχή της πρώτης αποτυχίας (first fail principle), αφού υποδεικνύει ότι η δουλειά που είναι πιο πιθανό να αποτύχει πρέπει να πραγματοποιηθεί πρώτη. Το ζήτημα που γεννιέται είναι πως μπορεί να μετρηθεί το κατά πόσο μια μεταβλητή μπορεί να οδηγήσει σε αδιέξοδο, δηλαδή πόσο περιορισμένη είναι.

Ένας απλός τρόπος για να εκτιμήσουμε πόσο περιορισμένη είναι μια συγκεκριμένη μεταβλητή είναι η χρήση του αριθμού των μεταβλητών που σχετίζονται μέσω περιορισμού με τη μεταβλητή. Έτσι στο Γράφο περιορισμών, μπορούμε να χρησιμοποιήσουμε τον αριθμό των ακμών που συνδέονται σε ένα κόμβο (μια μεταβλητή). Κατά συνέπεια, όσο μεγαλύτερος είναι ο αριθμός των ακμών, τόσο πιο πολύ περιορισμένη είναι η μεταβλητή. Ωστόσο, αυτή η μέτρηση είναι κάπως αδύναμη αφού είναι στατική, δηλαδή δεν λαμβάνει υπόψη την τωρινή κατάσταση της μερικής λύσης. Ένας καλύτερος τρόπος είναι η μέτρηση των ακμών που συνδέονται στους κόμβους που δεν έχουν ακόμα συμπεριληφθεί στη μερική λύση. Αυτή η μέτρηση είναι δυναμική, αφού το νούμερο αλλάζει δυναμικά κατά τη διάρκεια κάθε διεργασίας και μας δείχνει πόσο περιορισμένη είναι μια μεταβλητή σε σχέση με το μελλοντικό υποπρόβλημα. Μια εναλλακτική μέθοδος είναι η μέτρηση του αριθμού των τιμών που είναι συνεπείς με τη μερική λύση. Αυτή η μέθοδος συνήθως χρησιμοποιείται στον αλγόριθμο αναζήτησης που ονομάζεται πρώμος έλεγχος (forward checking). Πιο αναλυτικά, όταν ανατίθεται στην τρέχουσα μεταβλητή μια τιμή, ο πρώμος έλεγχος ψάχνει για κάθε μελλοντική μεταβλητή Y (στην οποία δεν έχει ανατεθεί ακόμα τιμή) και η οποία συνδέεται με κάποιο περιορισμό με τη X και διαγράφει από το πεδίο τιμών της Y κάθε τιμή που βρίσκεται σε ασυνέπεια με την επιλεγόμενη τιμή της X . Έτσι μπορεί εύκολα σε κάθε βήμα ο αλγόριθμος να επιλέξει τη μεταβλητή με το μικρότερο τρέχον πεδίο τιμών. Ο τρόπος με τον οποίο οι τρέχουσες αποφάσεις για τη μεταβλητή και την επιλογή τιμής για αυτήν αποτρέπουν μέρος της μελλοντικής αναζήτησης είναι η απλούστερη μορφή διάδοσης περιορισμών. Ο πρώμος έλεγχος διαδίδει το αποτέλεσμα της προσωρινά επιλεγμένης τιμής σε κάθε μια από τις μελλοντικές μεταβλητές με τις οποίες σχετίζεται σε κάποιο περιορισμό. Αν το πεδίο τιμών μιας από αυτές τις μελλοντικές

μεταβλητές γίνει κενό (όλες οι τιμές βρίσκονται σε ασυνέπεια) τότε η προσωρινά επιλεγμένη τιμή απορρίπτεται και η επόμενη υποψήφια επιλέγεται. Η διαδικασία αυτή ακολουθείται μέχρι την εύρεση λύσης ή την υπόδειξη ότι δεν υπάρχει λύση. Ο πρώτος έλεγχος ανιχνεύει πολλές ασυνέπειες, αλλά δεν τις ανιχνεύει όλες. Για το σκοπό αυτό εισήχθησαν οι αλγόριθμοι συνέπειας.

2.3 Ανασκόπηση βιβλιογραφίας

Οι πρώτες προσπάθειες για την επίλυση κάποιων παραλλαγών του προβλήματος με αποτελεσματικό τρόπο ξεκίνησαν την δεκαετία του 60. Συγκεκριμένα, οι Welsh και Powell [3] χρησιμοποίησαν την τεχνική χρωματισμού γραφημάτων για να λύσουν ένα πρόβλημα χρονοπρογραμματισμού, ενώ οι Schmidt και Strohleln [4] προσέγγισαν το πρόβλημα με τεχνικές προσομοίωσης. Μεταξύ των πρώτων προσεγγίσεων στον μαθηματικό προγραμματισμό, οι Lawrie [5] και Akkoynlu [6] παρουσιάζουν μοντέλα γραμμικού και ακέραιου προγραμματισμού για ορισμένες εκδόσεις του προβλήματος και έτσι καταφέρνουν να υπολογίσουν βέλτιστες λύσεις για ένα σχολικό και ένα πανεπιστημιακό πρόβλημα χρονοπρογραμματισμού. Πιο συγκεκριμένα, στο [5] η διαδικασία χρονοπρογραμματισμού ξεκινά με τη δημιουργία των λεγόμενων διατάξεων (layouts) για τις διάφορες ομάδες μαθητών και μια διατύπωση ακέραιου προγραμματισμού προσπαθεί να βρει ένα σύνολο βέλτιστων διατάξεων (layouts), για την κάθε ομάδα. Αντίθετα, στο [6] το πρόβλημα λύνεται ως πρόβλημα ανάθεσης (assignment problem) μεταξύ ενός συνόλου μαθημάτων και των χρονικών περιόδων ενός εβδομαδιαίου προγράμματος, με κεντρικό στόχο στην αποφυγή συγκρούσεων.

Στο [7] παρέχεται λύση για το πρόβλημα ανάθεσης καθηγητών, ένα πρόβλημα στενά συνδεδεμένο με το πρόβλημα χρονοπρογραμματισμού, χρησιμοποιώντας μοντέλα γραμμικού προγραμματισμού MIP. Το ίδιο πρόβλημα μελετήθηκε στο [8], πάλι με τη βοήθεια του μαθηματικού προγραμματισμού. Πιο πρόσφατα, στο [9] το πρόβλημα ανάθεσης καθηγητών διατυπώνεται ως πρόβλημα MIP και λύνεται ως ειδική περίπτωση του προβλήματος σταθερού κόστους μεταφοράς. Επίσης, στο [10] γίνεται διατύπωση γραμμικού προγραμματισμού για το πρόβλημα κατανομής αιθουσών, ένα υποπρόβλημα του πανεπιστημιακού χρονοπρογραμματισμού.

Στο [11] αποδεικνύεται η NP-πληρότητα του προβλήματος το οποίο αντιμετωπίζεται με την βοήθεια του ακέραιου προγραμματισμού IP και της θεωρίας γραφημάτων. Επεκτάσεις αυτού του έργου, ειδικά όσον αφορά το λεγόμενο πίνακα συγκρούσεων, παρουσιάζονται στο [12].

Πολλοί συγγραφείς διαιρούν τις απαιτήσεις σε δύο ομάδες: τις σκληρές (Hard), οι οποίες περιλαμβάνονται στους περιορισμούς και ορίζουν τον χώρο αναζήτησης, και τις μαλακές (Soft), οι οποίες περιλαμβάνονται στην αντικειμενική συνάρτηση [13]. Στο [14] εμφανίζεται μια νέα τεχνική με το πρόβλημα να σπάει σε υποπροβλήματα.

Εκτός από τις κλασικές προσεγγίσεις μαθηματικού προγραμματισμού, έχουν επίσης χρησιμοποιηθεί τεχνικές για συνδυαστικά προβλήματα (combinatorial problems). Μεταξύ αυτών, η αναζήτηση tabu (tabu search) στα [15] [16]. Ο λογικός προγραμματισμός περιορισμών παρουσιάζεται στα [17] [18].

Τέλος, οι γενετικοί αλγόριθμοι έχουν αξιοποιηθεί ως ένα αποτελεσματικό εργαλείο για την επίλυση προβλημάτων χρονοπρογραμματισμού. Για παράδειγμα, στα [19] [20] [21] [22] καθώς και στα [23] [24].

Στο [25] πρότειναν ένα μοντέλο Ακέραιου Προγραμματισμού (IP) με αποσύνθεση δύο σταδίων για την επίλυση του προβλήματος χρονοπρογραμματισμού σχολείου Lectio. Το μοντέλο βασίζεται στο προηγούμενο μοντέλο IP τους [26]. Χρησιμοποίησαν την αποσύνθεση για να χωρίσουν το μοντέλο σε δύο στάδια και αφαίρεσαν περιττές μεταβλητές για να μειώσουν τον χρόνο εκτέλεσης. Στο πρώτο στάδιο, τα γεγονότα ανατέθηκαν σε χρονικά διαστήματα. Στο δεύτερο στάδιο, τα γεγονότα ανατέθηκαν σε αίθουσες. Για να ξεπεραστεί η μη αναστρέψιμη απόφαση που λαμβάνεται στο πρώτο στάδιο, το μοντέλο του δεύτερου σταδίου ενσωματώθηκε μερικώς στο μοντέλο του πρώτου σταδίου, αξιοποιώντας το πρόβλημα ελάχιστου βάρους και μέγιστης αντιστοίχισης. Τα υπολογιστικά αποτελέσματα δείχνουν σημαντική βελτίωση στο μοντέλο τους σε σύγκριση με το προηγούμενο μοντέλο IP.

Στο [27] πρότειναν μια διατύπωση Μικτού Ακέραιου Γραμμικού Προγραμματισμού (MILP) για σύνολα δεδομένων μορφοποιημένα σύμφωνα με το ΧΗSTT. Η προσέγγισή τους αποτελείται από δύο βήματα. Στο πρώτο βήμα, το μοντέλο MIP κατασκευάστηκε μόνο με σκληρούς περιορισμούς. Χρησιμοποιήθηκε για να ελαχιστοποιήσει τις παραβιάσεις σκληρών περιορισμών μέχρι να λήξει το χρονικό όριο ή να βρεθεί μια εφικτή λύση. Αν βρισκόταν μια εφικτή λύση, τότε ενεργοποιούνταν το δεύτερο βήμα. Στο δεύτερο βήμα, η λύση βελτιωνόταν περαιτέρω στον εναπομείναντα χρόνο με την προσθήκη όλων των μαλακών περιορισμών στο μοντέλο. Η προσέγγιση αυτή κατάφερε να βρει βέλτιστες λύσεις για δύο περιπτώσεις και να αποδείξει τη βελτιστοποίηση τεσσάρων γνωστών λύσεων.

Στο [28] πρότειναν μια προσέγγιση δύο σταδίων (TSA) με μοντέλο Μικτού Ακέραιου Προγραμματισμού (MIP), καθώς και μια μέθοδο Column Generation Heuristic (CGH) για την αντιμετώπιση έξι προβλημάτων χρονοπρογραμματισμού λυκείων από το Κουβέιτ. Στο πρώτο στάδιο, το οποίο αναφέρεται ως μοντέλο 1, καθορίστηκαν οι εβδομαδιαίες χρονοθυρίδες για τις τάξεις. Στο δεύτερο στάδιο, το οποίο αναφέρεται ως μοντέλο 2, οι δάσκαλοι ανατέθηκαν στις τάξεις. Για την εναλλακτική μέθοδο, παρουσίασαν ένα ολοκληρωμένο μοντέλο MIP (αναφέρεται ως μοντέλο 3) για την επιλογή έγκυρων συνδυασμών εβδομαδιαίων προγραμμάτων για κάθε απασχολούμενο δάσκαλο. Λόγω του εκθετικού αριθμού μεταβλητών στο μοντέλο 3, σχεδιάστηκε μια CGH για την επίλυση της χαλάρωσης του γραμμικού προγραμματισμού, όπου δημιουργήθηκε μια ευρετική μέθοδος διαδοχικής σταθεροποίησης μεταβλητών για την παραγωγή μιας ποιοτικής εφικτής λύσης. Και οι δύο προσεγγίσεις παρήγαγαν λύσεις σε σχετικά σύντομους χρόνους CPU.

Στο [29] βελτίωσαν περαιτέρω το έργο των [27] χρησιμοποιώντας νέες τομές και αναδιατυπώσεις για το υπάρχον μοντέλο IP για την επίλυση του συνόλου δεδομένων ITC2011. Από την αρχική διατύπωση, οι Fonseca βελτίωσαν ορισμένες έγκυρες ανισότητες και μια εκτεταμένη διατύπωση βασισμένη στη ροή. Ως αποτέλεσμα, τα κλασματικά σημεία μειώθηκαν στη νέα διατύπωση. Προτάθηκαν ρουτίνες προεπεξεργασίας για την αφαίρεση περιττών περιορισμών και μεταβλητών. Έγινε αφαίρεση αρκετών μη απαραίτητων περιορισμών, ώστε να μειωθεί το μέγεθος των περιορισμών σε μια εναλλακτική διατύπωση. Οι προτεινόμενες τομές βελτίωσαν τη γραμμική χαλάρωση της διατύπωσης, οδηγώντας σε μείωση του κεντρικού χάσματος κατά μέσο όρο 32%. Αυτή η προσέγγιση αποτελεί την πλέον προηγμένη τεχνική για τις περιπτώσεις προβλημάτων στο ITC2011.

Στο [30] πρότειναν ένα μοντέλο Μικτού Ακέραιου Προγραμματισμού (MIP) για την επίλυση προβλημάτων χρονοπρογραμματισμού λυκείων στην Ελλάδα. Το μοντέλο υλοποιήθηκε χρησιμοποιώντας τα εργαλεία Gurobi και CPLEX. Δύο μεθοδολογίες εφαρμόστηκαν στη διατύπωση. Η πρώτη χρησιμοποιούσε ένα "μονολιθικό" μοντέλο που περιλάμβανε όλους τους σκληρούς και μαλακούς περιορισμούς. Η δεύτερη διέσπασε το πρόβλημα σε έξι υποπροβλήματα. Τα υπολογιστικά αποτελέσματα έδειξαν ότι η δεύτερη μέθοδος ήταν αποτελεσματική στην παραγωγή βέλτιστων λύσεων, ενώ η πρώτη μέθοδος δεν παρήγαγε ικανοποιητικά αποτελέσματα.

Κεφάλαιο 3 Μεθοδολογία

3.1 Επιλογή θέματος

Για τα περισσότερα πανεπιστήμια της χώρας μας, η σχεδίαση του προγράμματος πραγματοποιείται χειροκίνητα από το διοικητικό και γραμματειακό προσωπικό του κάθε τμήματος. Πιο συγκεκριμένα, η γραμματεία επικοινωνεί με κάθε μέλος του διδακτικού προσωπικού για να ενημερωθεί για τη διαθεσιμότητά του. Στη συνέχεια, μετά την απαραίτητη ενημέρωση από όλους τους καθηγητές και λαμβάνοντας υπόψη μια πληθώρα άλλων παραγόντων όπως η διαθεσιμότητα και η χωρητικότητα των αιθουσών, η ισότιμη κατανομή των μαθημάτων κατά τη διάρκεια της εβδομάδας κ.λπ., η γραμματεία καταρτίζει το προσχέδιο του προγράμματος. Το προσχέδιο αυτό, μετά την έγκριση από όλους τους εμπλεκόμενους, ανακοινώνεται στους φοιτητές του εκάστοτε τμήματος.

Από τα παραπάνω γίνεται φανερό ότι η σχεδίαση του προγράμματος μαθημάτων είναι μια πολύπλοκη και χρονοβόρα διαδικασία, η οποία απαιτεί αυξημένη δέσμευση ανθρώπινου δυναμικού. Αυτό το γεγονός καθιστά τη διαδικασία όχι μόνο ανώφελη αλλά και αναποτελεσματική για ένα εκπαιδευτικό ίδρυμα.

Η ανάγκη για μια πιο αποδοτική και ευέλικτη λύση στον χρονοπρογραμματισμό εκπαιδευτικών δραστηριοτήτων είναι επιτακτική. Για το λόγο αυτό, αποφάσισα να αναπτύξω μια πλατφόρμα θα χρησιμοποιεί προηγμένες τεχνικές πληροφορικής και αλγορίθμους για την αυτοματοποίηση της διαδικασίας δημιουργίας προγραμμάτων. Στόχος είναι να μειωθεί ο χρόνος και η προσπάθεια που απαιτείται από το διοικητικό προσωπικό, ενώ παράλληλα να διασφαλιστεί η ακρίβεια και η αποτελεσματικότητα του τελικού προγράμματος. Επιπλέον, η πλατφόρμα θα επιτρέπει την εύκολη προσαρμογή και αναθεώρηση του προγράμματος, σε περίπτωση αλλαγών ή απροόπτων.

Με την ανάπτυξη αυτής της πλατφόρμας, φιλοδοξώ να προσφέρω ένα εργαλείο που θα εξοικονομήσει πόρους και χρόνο στα πανεπιστήμια, βελτιώνοντας την οργάνωση και την ποιότητα της εκπαιδευτικής διαδικασίας.

3.2 Έρευνα

Η έρευνα σχετικά με τις μεθόδους επίλυσης προβλημάτων χρονοπρογραμματισμού αποτελεί μια συνεχή δραστηριότητα, η οποία συμβάλλει διαρκώς στη βελτίωση των διαδικασιών κατασκευής χρονοδιαγραμμάτων. Για τις ανάγκες της παρούσας εργασίας, κρίθηκε αναγκαία η διερεύνηση της βιβλιογραφίας σχετικά με την προσέγγιση που θα χρησιμοποιηθεί. Ως σημείο αναφοράς στην αναζήτηση άρθρων και εργασιών, ορίστηκε ο χρονοπρογραμματισμός σχολικών μονάδων.

Για την αναπαράσταση της περιγραφής ενός προβλήματος χρονοπρογραμματισμού, των δεδομένων του και των διαφόρων λύσεων του, ορίστηκε μια XML μορφή αναπαράστασης, που ονομάζεται XML για High School TimeTable δεδομένα (XHSTT) [31] [32] [33]. Αυτή η μορφή χρησιμοποιήθηκε για να αναπαρασταθούν, με κοινό τρόπο, προβλήματα χρονοπρογραμματισμού σχολείων δευτεροβάθμιας εκπαίδευσης από διάφορα κράτη του κόσμου, με στόχο την ύπαρξη μιας κοινής μεθόδου αξιολόγησης των λύσεων σε κοινά προβλήματα.

Οι διαφορετικές εκπαιδευτικές δομές των κρατών ώθησαν στη δημιουργία μιας γενικευμένης περιγραφής τέτοιων προβλημάτων καθώς και μιας γενικευμένης περιγραφής των λύσεων ανά τον κόσμο. Η μορφή XHSTT υιοθετήθηκε ευρέως από την επιστημονική κοινότητα και χρησιμοποιήθηκε ως επίσημος τρόπος αναπαράστασης προβλημάτων και λύσεων σε παγκόσμιους διαγωνισμούς επίλυσης προβλημάτων δημιουργίας ωρολογίων προγραμμάτων για σχολεία δευτεροβάθμιας εκπαίδευσης.

Με αποτέλεσμα, πολλά ερευνητικά λογισμικά χρησιμοποιούν πλέον αυτή τη μορφοποίηση. Αυτή η τυποποιημένη προσέγγιση επιτρέπει στους ερευνητές να αναπτύσσουν και να συγκρίνουν αποτελεσματικά αλγόριθμους και τεχνικές επίλυσης προβλημάτων χρονοπρογραμματισμού. Η συνεχής εξέλιξη των μεθόδων αυτών δεν μόνο

βελτιώνει τις λύσεις που παρέχονται αλλά και συμβάλλει στη δημιουργία πιο αποδοτικών και ευέλικτων συστημάτων διαχείρισης χρονοδιαγραμμάτων για τα εκπαιδευτικά ιδρύματα.

3.3 Υπάρχουσες λύσεις

3.3.1 Mimosa

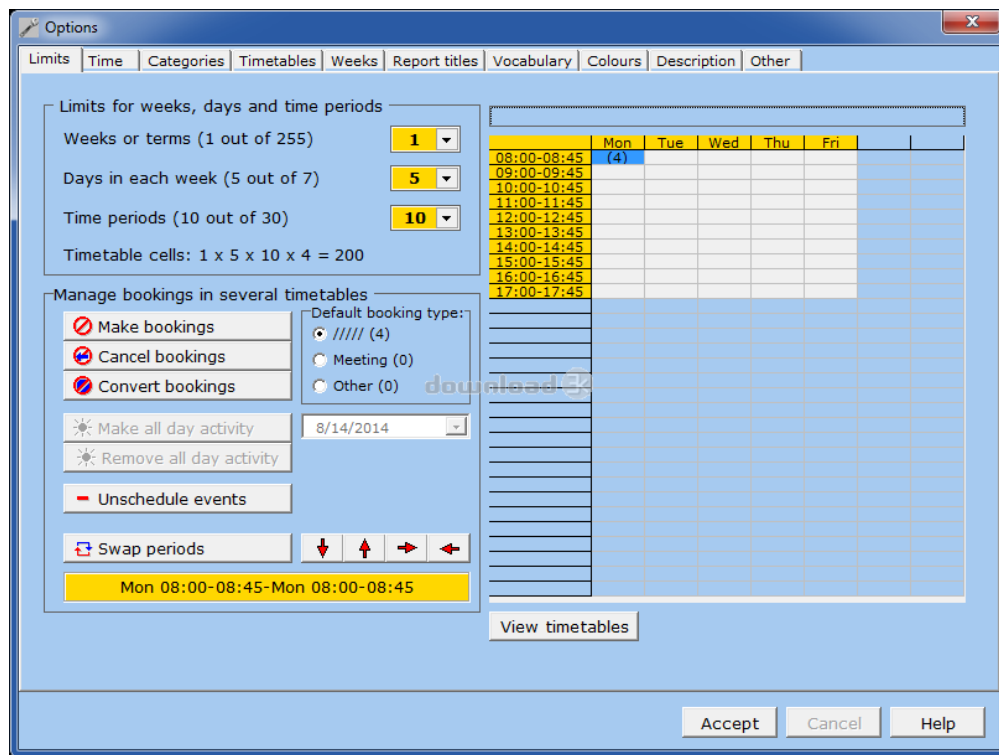
Το λογισμικό Mimosa είναι ένα σύστημα σχεδιασμένο για να διευκολύνει τη δημιουργία και διαχείριση ωρολογίων προγραμμάτων σε εκπαιδευτικά ιδρύματα. Το Mimosa είναι γνωστό για την ευελιξία και τη φιλικότητα προς τον χρήστη, προσφέροντας δυνατότητες που επιτρέπουν στους διαχειριστές να δημιουργούν προγράμματα που ικανοποιούν διάφορους περιορισμούς και απαιτήσεις. Το λογισμικό προσφέρει ένα φιλικό προς τον χρήστη περιβάλλον, καθιστώντας εύκολη τη δημιουργία και την επεξεργασία των προγραμμάτων από το διοικητικό προσωπικό. Είναι εξαιρετικά ευέλικτο και μπορεί να προσαρμοστεί για να καλύψει τις ανάγκες διαφόρων εκπαιδευτικών ιδρυμάτων, από σχολεία έως πανεπιστήμια, υποστηρίζοντας πολλούς τύπους μαθημάτων, διαλέξεων και εργαστηρίων, καθώς και διάφορες μορφές διδασκαλίας (π.χ., δια ζώσης, εξ αποστάσεως).

Το Mimosa λαμβάνει υπόψη μια μεγάλη γκάμα περιορισμών, όπως η διαθεσιμότητα των καθηγητών, των αιθουσών, και των μαθητών, καθώς και ειδικές απαιτήσεις μαθημάτων και τμημάτων. Διαχειρίζεται τόσο σκληρούς όσο και μαλακούς περιορισμούς, διασφαλίζοντας ότι τα προγράμματα είναι τόσο πρακτικά όσο και βελτιστοποιημένα. Προσφέρει λειτουργίες αυτοματοποιημένης δημιουργίας προγραμμάτων, μειώνοντας τον χρόνο και την προσπάθεια που απαιτούνται για τον χρονοπρογραμματισμό, με αλγόριθμους βελτιστοποίησης που εξασφαλίζουν αποδοτικά προγράμματα. Επιπλέον, το Mimosa εντοπίζει και επιλύει συγκρούσεις στα προγράμματα, όπως η ταυτόχρονη κράτηση της ίδιας αίθουσας ή η διπλή δέσμευση ενός καθηγητή.

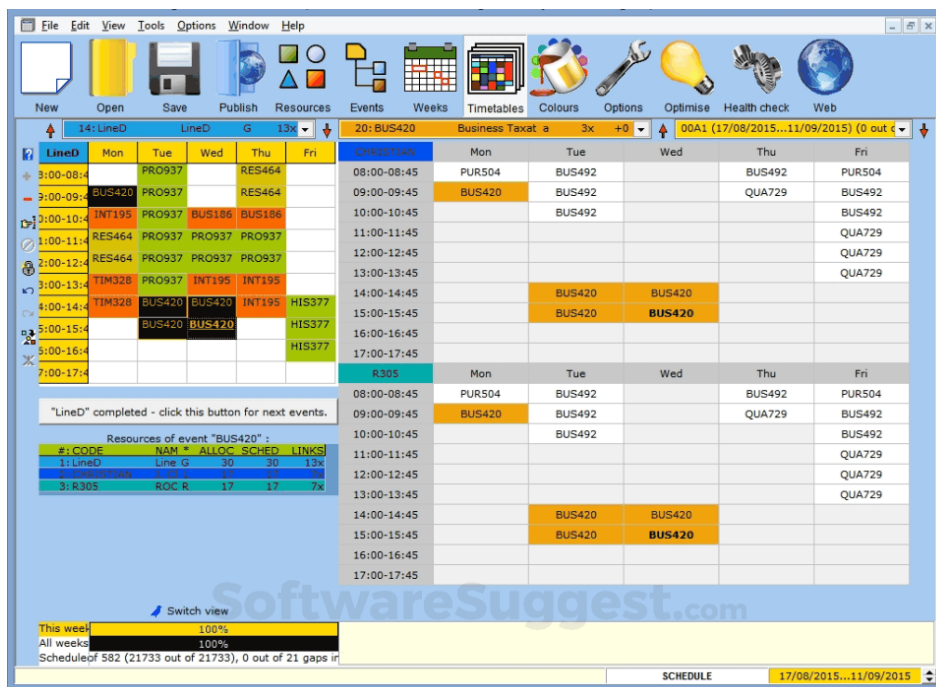
Το λογισμικό παρέχει δυνατότητες εκτύπωσης και εξαγωγής αναφορών, επιτρέποντας την εύκολη κοινοποίηση των προγραμμάτων στους ενδιαφερόμενους, όπως φοιτητές και καθηγητές. Χρησιμοποιείται από σχολεία, κολέγια και πανεπιστήμια για τη δημιουργία και διαχείριση των εβδομαδιαίων και ετήσιων ωρολογίων προγραμμάτων, βοηθώντας στη διαχείριση των αιθουσών διδασκαλίας και στην κατανομή των μαθημάτων με τρόπο που μεγιστοποιεί τη χρήση των διαθέσιμων πόρων. Διευκολύνει το διοικητικό προσωπικό στη διαδικασία του χρονοπρογραμματισμού, επιτρέποντας την εύκολη επεξεργασία και προσαρμογή των προγραμμάτων σύμφωνα με τις αλλαγές και τις ανάγκες που προκύπτουν. Παρέχει στους καθηγητές και στους φοιτητές πρόσβαση σε ενημερωμένα προγράμματα, διευκολύνοντας την οργάνωση και την παρακολούθηση των μαθημάτων.

Το Mimosa αποτελεί ένα ολοκληρωμένο εργαλείο για την επίλυση των προκλήσεων του χρονοπρογραμματισμού στην εκπαίδευση, συνδυάζοντας ευκολία στη χρήση, προσαρμοστικότητα και ισχυρές λειτουργίες αυτοματοποίησης και βελτιστοποίησης. Είναι σχεδιασμένο για να παρέχει περιορισμούς για έναν μεγάλο αριθμό διαφορετικών κανόνων και κανονισμών που υπάρχουν στο ακαδημαϊκό περιβάλλον. Συγκεκριμένα, επιτυγχάνει τη δημιουργία προγραμμάτων που είναι απαλλαγμένα από συγκρούσεις μεταξύ μαθημάτων, καθηγητών και αιθουσών, υποστηρίζοντας τον προγραμματισμό μαθημάτων που απαιτούν διαδοχικά χρονικά διαστήματα και συνεδρίες που επαναλαμβάνονται για να φιλοξενήσουν διαφορετικές ομάδες φοιτητών.

ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΤΟΥ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΧΟΛΙΚΩΝ ΜΟΝΑΔΩΝ ΜΕ ΧΡΗΣΗ ΤΕΧΝΙΚΩΝ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ



Σχήμα 3.1 Θθνή εισαγωγής δεδομένων του λογισμικού Mimoso



Σχήμα 3.2 Παράδειγμα αποτελέσματος του λογισμικού Mimoso

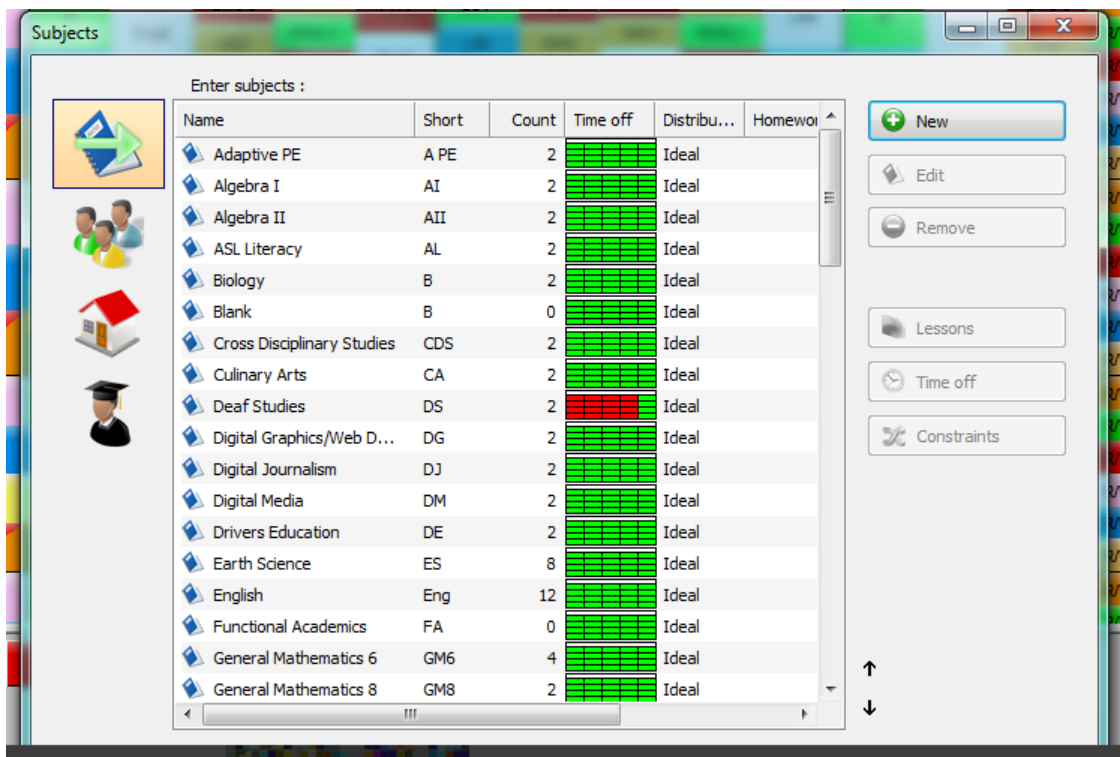
3.3.2 aSc Timetables

Το aSc Timetables είναι ένα λογισμικό χρονοπρογραμματισμού που προσφέρει μια πληθώρα δυνατοτήτων για τη διευκόλυνση της διαδικασίας δημιουργίας ωρολογίου προγράμματος. Η εφαρμογή επιτρέπει στον χρήστη να εισάγει δεδομένα σχετικά με τα μαθήματα, τα τμήματα, τους καθηγητές και τις διδασκαλίες, όπως η διάρκεια μιας διδασκαλίας, καθιστώντας το τελικό πρόγραμμα λεπτομερές και κατατοπιστικό. Επιπλέον, ο χρήστης μπορεί να εισάγει πληροφορίες σχετικά με τις διαθέσιμες αίθουσες, τα διαλείμματα και τη διάρκειά τους, καθώς και τους απαραίτητους πόρους για τη διεξαγωγή κάθε μαθήματος.

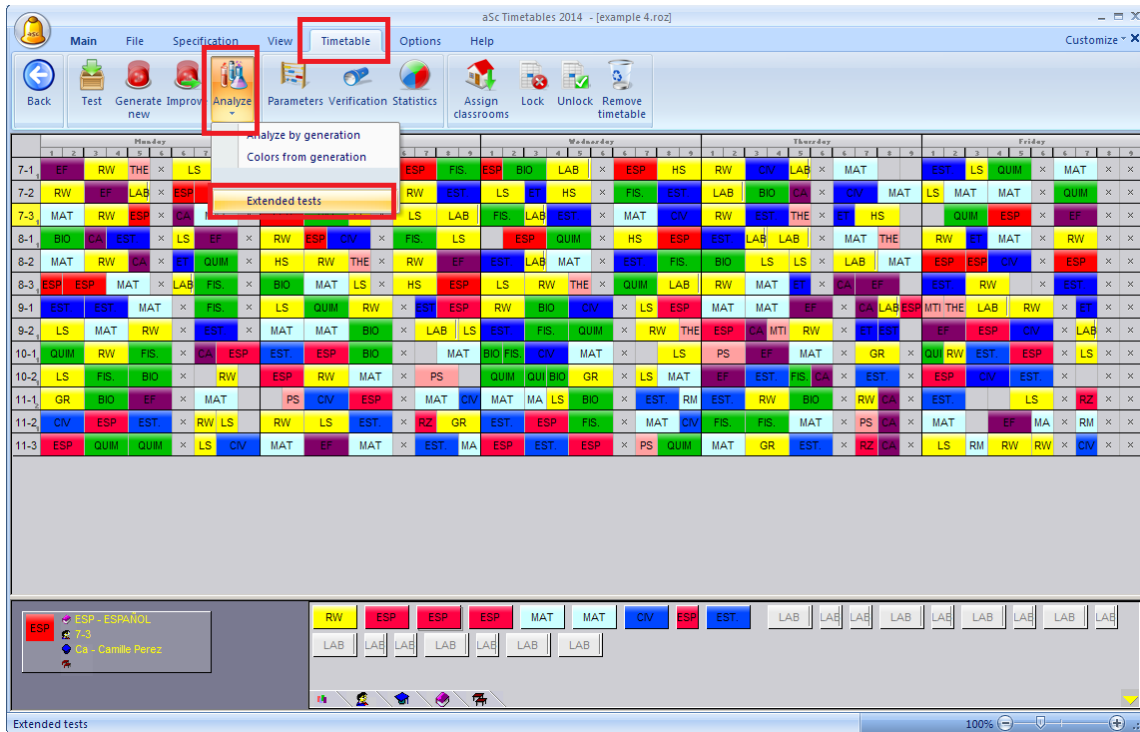
Το aSc Timetables παρέχει επίσης τη δυνατότητα δημιουργίας προγραμμάτων δύο εβδομάδων ή δέκα ημερών, προσφέροντας ευελιξία στον χρονοπρογραμματισμό. Οι χρήστες μπορούν να καθορίσουν περιορισμούς, όπως το να διδάσκει ένας καθηγητής μόνο την 7η ώρα ή να οριστεί ο ελάχιστος και ο μέγιστος αριθμός διδασκαλιών ανά ημέρα για έναν καθηγητή. Μόλις το πρόγραμμα δημιουργηθεί, μπορεί να εκτυπωθεί και να τροποποιηθεί η εμφάνισή του σύμφωνα με τις προτιμήσεις του χρήστη. Σε περιπτώσεις που τα δεδομένα είναι αποθηκευμένα σε ηλεκτρονική μορφή, η εφαρμογή υποστηρίζει την άμεση εισαγωγή τους.

Μία από τις σημαντικές λειτουργίες του aSc Timetables είναι η δυνατότητα αντικαταστάσεων, όπου ο χρήστης μπορεί να δηλώσει καθημερινά τους απόντες καθηγητές και η εφαρμογή θα εμφανίσει ποια μαθήματα χρειάζονται αντικατάσταση. Το λογισμικό παρέχει επίσης δυνατότητες δημοσίευσης του ωρολογίου προγράμματος και των αντικαταστάσεων μέσω της διαδικτυακής επέκτασης Timetable Online, η οποία είναι διαθέσιμη και για κινητά, καθώς και διατήρησης αντιγράφων ασφαλείας.

Τέλος, είναι σημαντικό να σημειωθεί ότι το aSc Timetables απαιτεί πληρωμή για τη δημιουργία του τελικού προγράμματος, παρόλο που προσφέρει μια δωρεάν δοκιμαστική έκδοση που περιλαμβάνει όλες τις δυνατότητες εκτός από την εξαγωγή δεδομένων. Αυτό καθιστά το aSc Timetables μια ολοκληρωμένη λύση για την αποτελεσματική διαχείριση και οργάνωση των ωρολογίων προγραμμάτων στα εκπαιδευτικά ιδρύματα.



Σχήμα 3.3 Οθόνη εισαγωγής δεδομένων του λογισμικού aSc Timetables



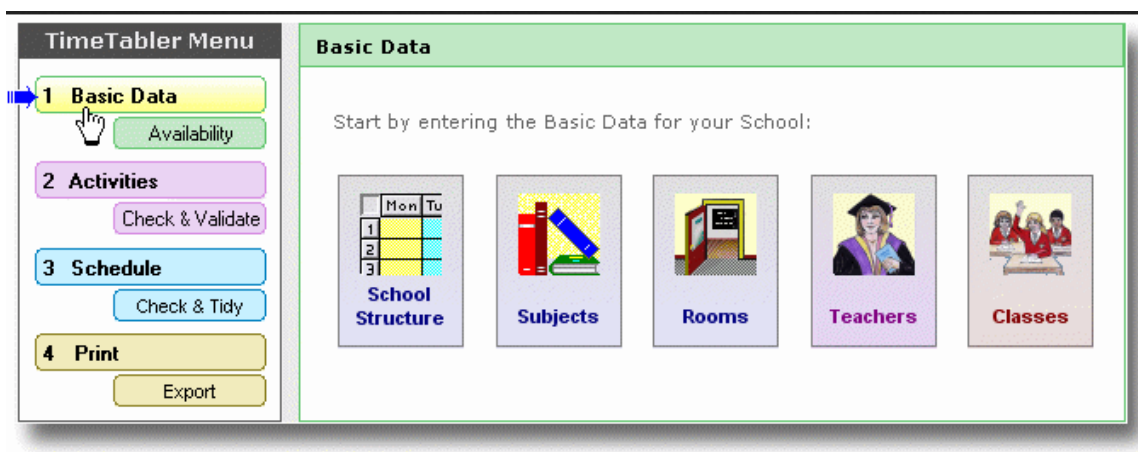
Σχήμα 3.4 Παράδειγμα αποτελέσματος του λογισμικού aSc Timetables

3.3.3 Timetabler

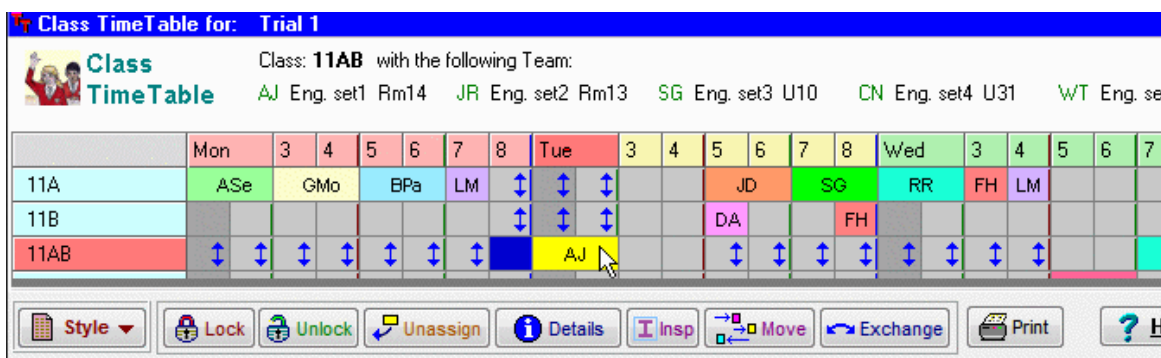
Το Timetabler είναι ένα λογισμικό χρονοπρογραμματισμού που χρησιμοποιείται παγκοσμίως από πολλούς εκπαιδευτικούς φορείς, προσφέροντας εκτενή καθοδήγηση για τους χρήστες. Με τη βοήθεια διαδραστικών βοηθητικών εικόνων, αναλυτικών βίντεο, σύντομου οδηγού χρήσης και γραμμής επικοινωνίας με το κέντρο υποστήριξης, οι χρήστες μπορούν να χρησιμοποιήσουν το πρόγραμμα χωρίς δυσκολία. Στο πρακτικό κομμάτι, το Timetabler εγγυάται έναν γρήγορο τρόπο εισαγωγής δεδομένων, αποφεύγοντας συγχύσεις και παράγοντας ένα πλήρως αποδοτικό ωρολόγιο πρόγραμμα.

Η εφαρμογή προτείνει μια σειρά για τα μαθήματα, αφήνοντας στον χρήστη την τελική επιλογή, καθώς μπορούν να πραγματοποιηθούν αλλαγές ανά πάσα στιγμή. Με την ολοκλήρωση του προγράμματος, δίνεται η δυνατότητα εκτύπωσης σε διάφορες μορφές, όπως πρόγραμμα που απεικονίζει μόνο τις διαλέξεις, τους καθηγητές ή τις αίθουσες και τα ωράρια. Η τελική μορφή του προγράμματος μπορεί να τροποποιηθεί και να προστεθούν χρώματα για καλύτερη κατανόηση από όλους τους χρήστες. Επιπλέον, το πρόγραμμα μπορεί να εξαχθεί σε μορφές όπως MIS, CSV ή XML.

Το Timetabler προσφέρει δωρεάν δοκιμαστική έκδοση, αλλά η πλήρης πρόσβαση σε όλες τις δυνατότητες απαιτεί πληρωμή. Αυτό καθιστά το Timetabler μια ολοκληρωμένη και ευέλικτη λύση για τη δημιουργία και διαχείριση ωρολογίων προγραμμάτων σε εκπαιδευτικά ιδρύματα, συνδυάζοντας ευκολία στη χρήση με ισχυρές λειτουργίες και υποστήριξη.



Σχήμα 3.5 Οθόνη εισαγωγής δεδομένων του λογισμικού Timetabler



Σχήμα 3.6 Παράδειγμα αποτελέσματος του λογισμικού Timetabler

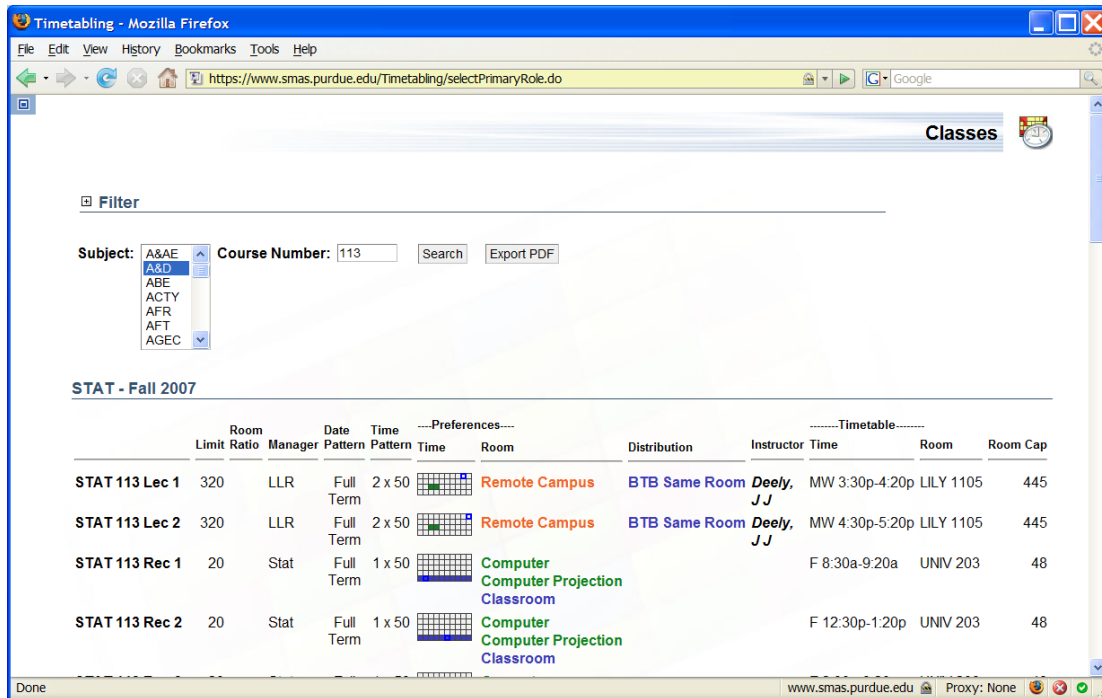
3.3.4 UniTime

Το UniTime αποτελεί ένα αυτόνομο και ολοκληρωμένο εκπαιδευτικό σύστημα προγραμματισμού, που υποστηρίζει την άμεση δημιουργία ωρολογίων προγραμμάτων - είτε μαθημάτων είτε εξετάσεων - με την βέλτιστη διαχείριση των διαθέσιμων αιθουσών και ανθρωπίνων πόρων. Το σύστημα επιτρέπει την ταυτόχρονη παρέμβαση διαφόρων εξουσιοδοτημένων χρηστών, όπως διοικητικό, γραμματειακό και εκπαιδευτικό προσωπικό, με στόχο την πιο άμεση και αποτελεσματική υλοποίηση του προγράμματος.

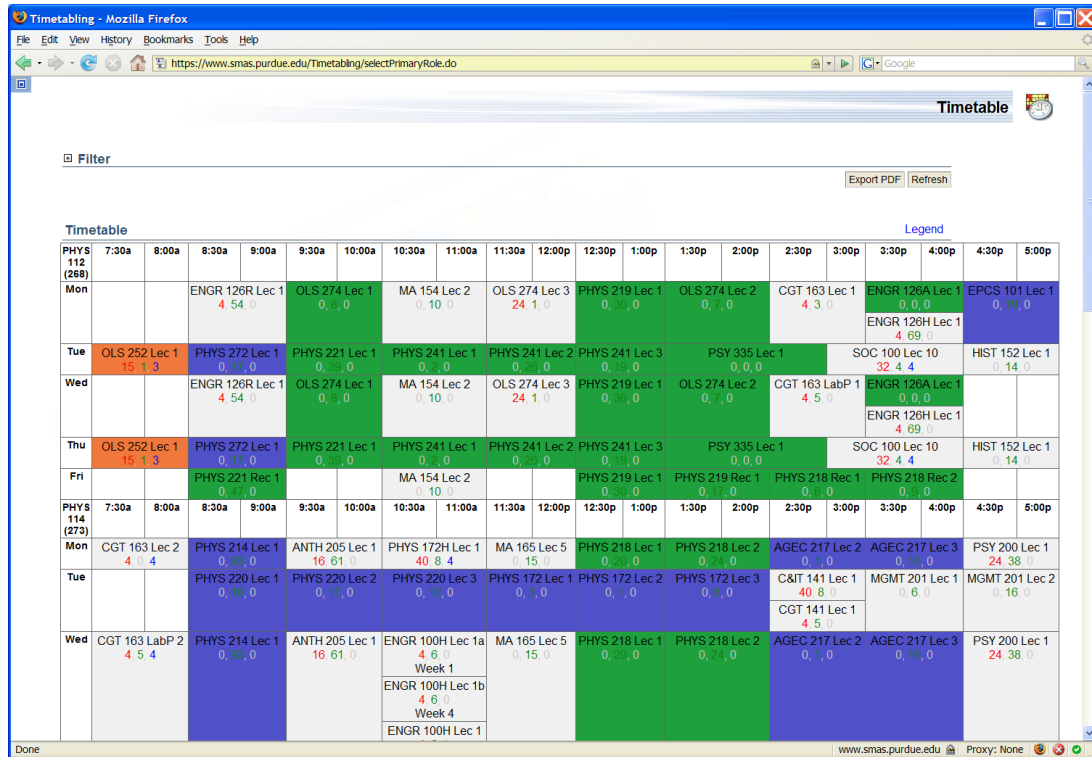
Αναπτύχθηκε από μέλη του προσωπικού και φοιτητές μεγάλων πανεπιστημίων της Ευρώπης και της Βόρειας Αμερικής, με στόχο τον καλύτερο προγραμματισμό των εκπαιδευτικών προγραμμάτων, που καθίσταται ιδιαίτερα επωφελής για τα εκπαιδευτικά ιδρύματα και τους σπουδαστές τους. Το UniTime διανέμεται δωρεάν με άδεια ανοικτού κώδικα και αποτελεί χρηματοδοτούμενο έργο του Ιδρύματος Arpego από το Μάρτιο του 2015.

Οι βασικοί ρόλοι της πλατφόρμας περιλαμβάνουν τόσο το εκπαιδευτικό όσο και το γραμματειακό/διοικητικό προσωπικό του εκπαιδευτικού ιδρύματος. Οι προτεινόμενες ώρες και ο τύπος της αίθουσας εισάγονται αρχικά στο σύστημα από τους καθηγητές και τους υπεύθυνους των μαθημάτων. Αφού όλα τα δεδομένα εισαχθούν στην πλατφόρμα, η γραμματεία μπορεί άμεσα να εξάγει ένα προσχέδιο του προγράμματος. Σε περίπτωση λάθους ή πιθανών βελτιώσεων, η πλατφόρμα υποστηρίζει την τροποποίηση του αυτοματοποιημένου προσχεδίου, παρέχοντας έτσι ευελιξία και ακρίβεια στη διαχείριση του ωρολογίου προγράμματος.

ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΤΟΥ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΧΟΛΙΚΩΝ ΜΟΝΑΔΩΝ ΜΕ ΧΡΗΣΗ ΤΕΧΝΙΚΩΝ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ



Σχήμα 3.7 Θθνή εισαγωγής δεδομένων του λογισμικού UniTime



Σχήμα 3.8 Παράδειγμα αποτελέσματος του λογισμικού UniTime

3.4 Επίλογή εργαλείου επίλυσης (Solver)

Όπως αναφέρθηκε και στο κεφάλαιο 2.3, υπάρχουν πολλές τεχνικές και διαφορετικοί αλγόριθμοι για την επίλυση των προβλημάτων χρονοπρογραμματισμού. Οι πιο δημοφιλείς βιβλιοθήκες για την επίλυση προβλημάτων με περιορισμούς είναι οι εξής:

- **IBM ILOG CPLEX:** Υποστηρίζει C++, Python, Java, .NET και είναι δωρεάν για ακαδημαϊκή χρήση.
- **Google OR-Tools:** Υποστηρίζει C++, Python, Java, .NET και διατίθεται υπό την Apache License 2.0.
- **OptaPlanner:** Υποστηρίζει Java και διατίθεται υπό την Apache License 2.0.
- **Gurobi:** Υποστηρίζει C, C++, Java, .NET, Python, MATLAB, και R και είναι δωρεάν για ακαδημαϊκή χρήση.

Η επιλογή της βιβλιοθήκης της Google, Google OR-Tools, ήταν η πιο ορθή για εμένα, μιας και είναι λογισμικό ανοιχτού κώδικα και υποστηρίζει την γλώσσα προγραμματισμού Python, με την οποία έχω τη μεγαλύτερη εξοικείωση. Το Google OR-Tools (Operations Research Tools) είναι ένα ισχυρό και ευέλικτο σύνολο λογισμικών βιβλιοθηκών και εργαλείων που έχουν αναπτυχθεί από την Google για την επίλυση σύνθετων προβλημάτων βελτιστοποίησης. Αποτελείται από αλγόριθμους και τεχνικές που επιτρέπουν την αποτελεσματική διαχείριση και επίλυση προβλημάτων όπως η δρομολόγηση οχημάτων, η χρονοπρογραμματισμός, η ανάθεση εργασιών και άλλα ζητήματα συνδυαστικής βελτιστοποίησης.

Όπως αναφέρθηκε προηγουμένως, υποστηρίζει πολλαπλές γλώσσες προγραμματισμού, όπως C++, Python, Java και .NET, παρέχοντας έτσι μεγάλη ευελιξία στους χρήστες να επιλέξουν το περιβάλλον που ταιριάζει καλύτερα στις ανάγκες τους. Η βιβλιοθήκη χρησιμοποιεί προηγμένους αλγόριθμους όπως ο αλγόριθμος Simplex για γραμμικό προγραμματισμό, οι αλγόριθμοι Branch and Bound και Branch and Cut για ακέραιο προγραμματισμό, καθώς και τεχνικές Metaheuristic όπως οι γενετικοί αλγόριθμοι. Αυτοί οι αλγόριθμοι επιτρέπουν την επίλυση τόσο μικρών όσο και μεγάλων προβλημάτων βελτιστοποίησης με υψηλή απόδοση.

Επιπλέον, το Google OR-Tools ενσωματώνει και υποστηρίζει τη χρήση εξωτερικών solvers, όπως το Gurobi, το CPLEX και το SCIP, επιτρέποντας στους χρήστες να αξιοποιούν εξειδικευμένα εργαλεία για τη βελτίωση της απόδοσης και της ακρίβειας των λύσεων τους. Η ευελιξία και η ισχύς του καθιστούν τη βιβλιοθήκη ιδανική για εφαρμογές σε διάφορους τομείς, όπως η εφοδιαστική αλυσίδα, η παραγωγή, οι μεταφορές και η διαχείριση πόρων. Το Google OR-Tools δεν περιορίζεται μόνο στην ακαδημαϊκή έρευνα, αλλά έχει ευρεία υιοθέτηση και στον βιομηχανικό τομέα, όπου χρησιμοποιείται για την επίλυση πραγματικών επιχειρηματικών προβλημάτων με σημαντικά οφέλη σε κόστος και αποδοτικότητα.

Η συνεχής ανάπτυξη και η υποστήριξη από την κοινότητα χρηστών και την ομάδα της Google διασφαλίζουν ότι παραμένει στην αιχμή της τεχνολογίας βελτιστοποίησης, προσφέροντας συνεχώς νέες δυνατότητες και βελτιώσεις. Είναι σημαντικό να αναφερθεί ότι από το 2013 μέχρι και σήμερα, τα OR-Tools παίρνουν το χρυσό μετάλλιο στον εθνικό διαγωνισμό προβλημάτων βελτιστοποίησης που ονομάζεται MiniZinc Challenge, μια διοργάνωση στην οποία μπορούν να συγκριθούν οι δυνατότητες των παραπάνω λύσεων σε πραγματικά προβλήματα.

Συνολικά, είναι ένα εξαιρετικά πολύτιμο εργαλείο για όποιον ασχολείται με την επίλυση προβλημάτων βελτιστοποίησης, παρέχοντας μια ολοκληρωμένη λύση που συνδυάζει ευελιξία, ισχύ και αποδοτικότητα.

3.5 Εργαλεία ανάπτυξης

Για την ανάπτυξη της πλατφόρμας διαχείρισης χρονοπρογραμματισμού εκπαιδευτικών δραστηριοτήτων, χρησιμοποιήθηκαν τα εξής εργαλεία: Το PyCharm, ένα δημοφιλές ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για Python που προσφέρει αυτόματη συμπλήρωση κώδικα και εργαλεία εντοπισμού σφαλμάτων. Το PostgreSQL, ένα ισχυρό σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων ανοιχτού κώδικα, γνωστό για την αξιοπιστία και την υψηλή απόδοσή του. Το Docker, μια πλατφόρμα που επιτρέπει την αυτοματοποίηση της ανάπτυξης εφαρμογών

μέσα σε κοντέινερ, εξασφαλίζοντας ομοιόμορφη λειτουργία σε διαφορετικά συστήματα. Και τέλος το Git, ένα σύστημα ελέγχου εκδόσεων ανοιχτού κώδικα που διευκολύνει την παρακολούθηση των αλλαγών στον κώδικα και τη συνεργασία μεταξύ προγραμματιστών.

3.5.1 PyCharm

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε το PyCharm. Το PyCharm είναι ένα πανίσχυρο εργαλείο για προγραμματιστές Python, προσφέροντας ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) που καλύπτει όλες τις φάσεις ενός έργου. Είναι κατάλληλο για τη συγγραφή και την επεξεργασία κώδικα, τον εντοπισμό σφαλμάτων, τη δημιουργία εκτελέσιμων αρχείων και την ανάπτυξη εφαρμογών. Υποστηρίζει διάφορα frameworks και βιβλιοθήκες Python, όπως Django και Flask.

Ανάμεσα στα βασικά χαρακτηριστικά του PyCharm που χρησιμοποιήθηκαν για την ανάπτυξη του κώδικα της εφαρμογής περιλαμβάνονται:

- **Επεξεργαστής Κώδικα και Έξυπνη Συμπλήρωση:** Το PyCharm παρέχει έναν προηγμένο επεξεργαστή κώδικα με δυνατότητες αυτόματης συμπλήρωσης και ευφυών προτάσεων. Το σύστημα IntelliSense προσφέρει υποδείξεις και συμβουλές για μεταβλητές, μεθόδους, κλάσεις και βιβλιοθήκες, καθιστώντας τη συγγραφή και την πλοήγηση στον κώδικα πιο αποδοτική.
- **Εργαλεία Εντοπισμού Σφαλμάτων:** Το PyCharm διαθέτει ισχυρά εργαλεία debugging, όπως σημεία διακοπής και παρατηρητές, επιτρέποντας την ανάλυση και τη διόρθωση σφαλμάτων μέσω της εκτέλεσης του κώδικα βήμα προς βήμα.
- **Διαχείριση Πακέτων:** Το PyCharm υποστηρίζει τη διαχείριση βιβλιοθηκών μέσω εργαλείων όπως το pip και το conda, διευκολύνοντας την εγκατάσταση και την ενημέρωση πακέτων.
- **Έλεγχος Έκδοσης:** Ενσωματωμένη υποστήριξη για Git και άλλα συστήματα ελέγχου έκδοσης, επιτρέποντας τη διαχείριση αποθετηρίων, τη δημιουργία διακλαδώσεων και τη συγχώνευση κώδικα.
- **Πρόσθετα και Plugins:** Το PyCharm διαθέτει ένα πλούσιο οικοσύστημα επεκτάσεων και plugins, αυξάνοντας τη λειτουργικότητα και την παραγωγικότητα κατά την ανάπτυξη.

Συνοψίζοντας, το PyCharm είναι ένα ευέλικτο και πλούσιο σε δυνατότητες IDE που επιτρέπει στους προγραμματιστές Python να δημιουργούν εφαρμογές με αποτελεσματικότητα και ευκολία, καλύπτοντας ένα ευρύ φάσμα αναγκών ανάπτυξης λογισμικού.

3.5.2 PostgreSQL

Το PostgreSQL είναι ένα ισχυρό, ανοικτού κώδικα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS), το οποίο έχει καθιερωθεί ως ένα από τα πλέον αξιόπιστα και προηγμένα συστήματα στη βιομηχανία πληροφορικής. Αναπτύχθηκε αρχικά στο Πανεπιστήμιο της Καλιφόρνιας, Μπέρκλεϊ, και η ανάπτυξή του συνεχίζεται από μια ευρεία κοινότητα προγραμματιστών σε όλο τον κόσμο. Το PostgreSQL υποστηρίζει ένα ευρύ φάσμα τύπων δεδομένων, όπως ακέραιοι αριθμοί, δεκαδικοί αριθμοί, κείμενο και τύποι δεδομένων χρήστη, παρέχοντας ευελιξία και προσαρμοστικότητα στις απαιτήσεις των εφαρμογών. Επιπλέον, προσφέρει προηγμένα χαρακτηριστικά όπως υποστήριξη για συναλλαγές (ACID compliance), πολύπλοκα ερωτήματα, ευρετήρια, αποθηκευμένες διαδικασίες και triggers. Η επεκτασιμότητα του PostgreSQL επιτρέπει την προσαρμογή και την επέκταση των λειτουργιών του μέσω των plugins και των ενοτήτων επέκτασης, διασφαλίζοντας ότι μπορεί να καλύψει τις ανάγκες διαφορετικών εφαρμογών και φορτίων εργασίας. Η ενσωματωμένη υποστήριξη για replication και high availability εξασφαλίζει την αξιοπιστία και τη συνεχή διαθεσιμότητα των δεδομένων. Επιπλέον, το PostgreSQL διαθέτει ισχυρά εργαλεία για την ασφάλεια και την διαχείριση των δικαιωμάτων πρόσβασης, προσφέροντας πλήρη έλεγχο και προστασία των

δεδομένων. Με την ευρεία υιοθέτησή του από την ακαδημαϊκή κοινότητα και τη βιομηχανία, το PostgreSQL αποτελεί μια στιβαρή επιλογή για την ανάπτυξη και τη διαχείριση βάσεων δεδομένων υψηλής απόδοσης.

3.5.3 Docker

Το Docker είναι μια πλατφόρμα λογισμικού που επιτρέπει στους προγραμματιστές να δημιουργούν, να αναπτύσσουν και να εκτελούν εφαρμογές μέσα σε κοντέινερ. Τα κοντέινερ είναι ελαφριές, απομονωμένες μονάδες λογισμικού που περιλαμβάνουν όλα τα απαραίτητα στοιχεία για την εκτέλεση μιας εφαρμογής, όπως κώδικα, runtime, βιβλιοθήκες και ρυθμίσεις. Αυτή η απομόνωση εξασφαλίζει ότι οι εφαρμογές θα λειτουργούν με τον ίδιο τρόπο ανεξαρτήτως του περιβάλλοντος στο οποίο εκτελούνται, εξαλείφοντας τα προβλήματα συμβατότητας μεταξύ ανάπτυξης και παραγωγής.

Η διαδικασία ξεκινά με το δημιουργία ενός Dockerfile, το οποίο είναι ένα αρχείο κειμένου που περιγράφει τα βήματα για τη δημιουργία ενός κοντέινερ. Το Dockerfile περιλαμβάνει εντολές για την εγκατάσταση των απαραίτητων εξαρτημάτων και τη ρύθμιση του περιβάλλοντος της εφαρμογής. Χρησιμοποιώντας την εντολή `'docker build'`, το Dockerfile μετατρέπεται σε ένα Docker image, το οποίο αποτελεί ένα στατικό στιγμιότυπο της εφαρμογής μαζί με το περιβάλλον της.

Στη συνέχεια, αυτό το Docker image μπορεί να αποθηκευτεί και να διανεμηθεί μέσω του Docker Hub, μιας δημόσιας ή ιδιωτικής αποθήκης για Docker images. Το Docker Hub λειτουργεί ως κεντρικός κόμβος όπου οι προγραμματιστές μπορούν να μοιράζονται και να αναζητούν images, διευκολύνοντας την επαναχρησιμοποίηση και τη συνεργασία.

Η ανάπτυξη της εφαρμογής πραγματοποιείται με την εντολή `'docker run'`, η οποία εκκινεί το Docker image ως ένα ενεργό κοντέινερ. Τα κοντέινερ είναι απομονωμένα μεταξύ τους και από το host σύστημα, διασφαλίζοντας έτσι την ασφάλεια και την σταθερότητα της εφαρμογής. Επιπλέον, τα κοντέινερ μπορούν να διαχειριστούν εύκολα μέσω του Docker Compose, το οποίο επιτρέπει τον ορισμό και την εκτέλεση πολυ-κοντέινερ εφαρμογών.

Η διαδικασία ανάπτυξης (deployment) με το Docker είναι εξαιρετικά αποδοτική, καθώς τα κοντέινερ μπορούν να αναπτυχθούν γρήγορα και με συνέπεια σε οποιοδήποτε περιβάλλον, είτε πρόκειται για τοπικό server, είτε για cloud υποδομή. Αυτό καθιστά τη διαδικασία συνεχούς ενσωμάτωσης και παράδοσης (CI/CD) πιο απρόσκοπτη, επιτρέποντας στους προγραμματιστές να παραδίδουν νέες εκδόσεις του λογισμικού με μεγαλύτερη συχνότητα και αξιοπιστία. Συνολικά, το Docker προσφέρει ευελιξία στην κλιμάκωση των εφαρμογών, μειώνει τις λειτουργικές δαπάνες και απλοποιεί τη διαχείριση λογισμικού, κάνοντάς το ένα αναντικατάστατο εργαλείο για τη σύγχρονη ανάπτυξη εφαρμογών.

3.5.4 Git

Το Git είναι ένα διανεμημένο σύστημα διαχείρισης εκδόσεων που έχει σχεδιαστεί για την αποτελεσματική διαχείριση και παρακολούθηση αλλαγών στον πηγαίο κώδικα κατά τη διάρκεια της ανάπτυξης λογισμικού. Δημιουργήθηκε από τον Linus Torvalds το 2005 και έκτοτε έχει καθιερωθεί ως το πρότυπο εργαλείο για την ανάπτυξη λογισμικού. Το Git επιτρέπει σε πολλούς προγραμματιστές να συνεργάζονται σε ένα έργο ταυτόχρονα, παρέχοντας ισχυρά εργαλεία για τη συγχώνευση αλλαγών και την επίλυση συγκρούσεων. Βασικό χαρακτηριστικό του Git είναι η διανεμημένη φύση του, καθώς κάθε προγραμματιστής διατηρεί ένα τοπικό αντίγραφο ολόκληρου του ιστορικού του έργου, επιτρέποντας τη γρήγορη πρόσβαση και την ανεξαρτησία από έναν κεντρικό διακομιστή. Το σύστημα παρέχει επίσης δυνατότητες όπως οι κλάδοι (branches) και οι ετικέτες (tags), οι οποίες διευκολύνουν τη διαχείριση των διαφορετικών εκδόσεων και των παραλλαγών του κώδικα. Επιπλέον, το Git είναι εξαιρετικά αποδοτικό και ασφαλές, με προηγμένους αλγόριθμους για την ανίχνευση και την αποθήκευση αλλαγών. Η ευρεία αποδοχή και υποστήριξή του από την κοινότητα ανάπτυξης λογισμικού το καθιστά ένα θεμελιώδες εργαλείο για τη σύγχρονη διαχείριση κώδικα και την ομαδική εργασία.

Κεφάλαιο 4 Παρουσίαση του συστήματος

Σε αυτό το κεφάλαιο θα εξηγήσουμε την διατύπωση που ακολουθήσαμε για την μορφοποίηση του προβλήματος δημιουργίας ενός ωρολογίου προγράμματος για ένα εκπαιδευτικό οργανισμό δευτεροβάθμιας εκπαίδευσης. Πολλοί περιορισμοί που περιλαμβάνονται στην διατύπωση μας βασίζονται στην αποτύπωση του Kristiansen [34]. Ωστόσο, υπάρχουν κάποιες βασικές διαφορές μεταξύ της μεθόδου επίλυσης του Kristiansen και αυτής που εφαρμόστηκε κατά τη διάρκεια της μεταπτυχιακής αυτής έρευνας. Η πρώτη είναι ότι ο Kristiansen διατυπώνει ένα ακέραιο πρόγραμμα για το γενικό πρόβλημα δημιουργίας ενός ωρολογίου προγράμματος για ένα εκπαιδευτικό οργανισμό, ενώ το δικό μας πρόγραμμα μπορεί να χρησιμοποιηθεί μόνο για περιπτώσεις που περιέχουν τους ίδιους τύπους περιορισμών με αυτούς της Ελλάδας (βλέπε 4.1). Ως δεύτερη διαφορά, λαμβάνουμε υπόψη μας μόνο ποινές για μαλακούς περιορισμούς (soft constraints) ενώ ο Kristiansen επιτρέπει ποινές και για σκληρούς περιορισμούς. Επίσης, στην διατύπωση μας ο περιορισμός με όνομα Ανάθεση χρόνου 4.4.2 μπορεί να εμφανιστεί και ως ελαστικός περιορισμός. Αυτό σημαίνει ότι οι λύσεις για τις οποίες δεν έχουν ανατεθεί όλα (ή ακόμα και κανένα) από τα γεγονότα θεωρούνται επίσης εφικτές. Με αυτόν τον τρόπο μπορούμε να δημιουργήσουμε προγράμματα ακόμα και στην περίπτωση που οι απαιτήσεις του χρήστη είναι μη εφικτές.

Το μοντέλο υλοποιήθηκε ώστε να παρέχει περιορισμούς για ένα μεγάλο αριθμό διαφορετικών κανόνων και κανονισμών που υπάρχουν στο ακαδημαϊκό περιβάλλον. Συγκεκριμένα, το μοντέλο επιτυγχάνει τη δημιουργία προγραμμάτων που είναι απαλλαγμένα από συγκρούσεις μεταξύ μαθημάτων, καθηγητών και αιθουσών. Επιπλέον, υποστηρίζει το προγραμματισμό μαθημάτων που απαιτούν διαδοχικά χρονικά διαστήματα, καθώς και μαθήματα που απαιτούν συνεδρίες που επαναλαμβάνονται αρκετές φορές για να φιλοξενήσουν διαφορετικές ομάδες φοιτητών. Είναι γνωστό ότι οι αιτήσεις για διαδοχικά χρονικά διαστήματα στα προβλήματα χρονοπρογραμματισμού αυξάνουν την πολυπλοκότητα και καθιστούν το πρόβλημα NP-Δύσκολο [35].

4.1 Χαρακτηριστικά του Ελληνικού ωρολογίου προγράμματος

Πρέπει να αναφέρουμε ότι το πρόβλημα δημιουργίας ωρολογίου προγράμματος για τα ιδρύματα δευτεροβάθμιας εκπαίδευσης στην Ελλάδα είναι πιο δύσκολο από ότι για τις περισσότερες άλλες χώρες. Οι βασικοί λόγοι είναι οι εξής:

- Ένα μάθημα που προσφέρεται από οποιοδήποτε τμήμα του πανεπιστημίου μπορεί να αποτελείται μόνο από διαλέξεις, ή διαλέξεις και ασκήσεις, ή διαλέξεις, ασκήσεις και/ή εργαστήριο.
- Οι διαλέξεις διδάσκονται από καθηγητές, εκπαιδευτικούς ή άλλο εκπαιδευτικό προσωπικό και επιλέγουν οι ίδιοι πως θα χωρίσουν τις διδακτικές ώρες για το κάθε μάθημα.
- Δύο ή περισσότεροι καθηγητές μπορεί να αναλάβουν τις διαλέξεις για ένα συγκεκριμένο μάθημα, στην περίπτωση αυτή πρέπει να αποφασίσουν τον τρόπο διαμοιρασμού του φορτίου διδασκαλίας.
- Τα εργαστήρια λαμβάνουν χώρα σε ειδικά εξοπλισμένες αίθουσες.
- Μαθητές στις ανώτερες τάξεις έχουν τη δυνατότητα να επιλέξουν μαθήματα από άλλες κατευθύνσεις.
- Πολλοί εκπαιδευτικοί μερικής απασχόλησης οι οποίοι είναι διαθέσιμοι μόνο κατά συγκεκριμένες ημέρες ή μέρη ημέρας.

4.1.1 Τύποι μαθημάτων

Σε πολλά Ελληνικά πανεπιστήμια, τα μαθήματα χαρακτηρίζονται είτε ως υποχρεωτικά είτε ως επιλογής. Τα υποχρεωτικά μαθήματα είναι αυτά που ένα συγκεκριμένο τμήμα θεωρεί ως βασικά για την εκπαίδευση των φοιτητών του. Κατά τη διάρκεια των πρώτων ετών, τα περισσότερα μαθήματα είναι υποχρεωτικά· επομένως σχεδιάζονται για όλους τους φοιτητές του ίδιου εξάμηνου. Αντιθέτως, τα επιλογής είναι λιγότερες σε αριθμό. Ωστόσο, στο πρόγραμμα δεν πρέπει ποτέ να επικαλύπτονται με κανένα υποχρεωτικό ή άλλο επιλογής μάθημα.

Κατά τα υψηλότερα έτη σπουδών, αντίστοιχα, τα υποχρεωτικά μαθήματα είναι λιγότερα σε αριθμό, ενώ οι επιλογές αυξάνονται εκθετικά. Επιπλέον, είναι πολύ συνηθισμένο για ορισμένα τμήματα να οργανώνουν τους φοιτητές τους σε διάφορες κατευθύνσεις και κάθε κατεύθυνση έχει το δικό της σύνολο υποχρεωτικών και επιλογής μαθημάτων. Με αποτέλεσμα, ο χρονοπρογραμματισμός να γίνεται ακόμα πιο περίπλοκος, δεδομένου ότι μερικά μαθήματα μπορεί να επικαλύπτονται επειδή δεν υπάρχει κοινό ενδιαφέρον μεταξύ των κατευθύνσεων, ενώ αυτά που έχουν κοινό ενδιαφέρον να μην πρέπει να συγκρούονται και το πρόγραμμά τους πρέπει να είναι βολικό για όλους τους ενδιαφερόμενους φοιτητές.

4.1.2 Διαθεσιμότητα των πόρων

Στο πλαίσιο του προγραμματισμού ενός πανεπιστημιακού προγράμματος, οι κύριοι πόροι που λαμβάνονται υπόψη είναι οι άνθρωποι και οι αίθουσες. Η διαθεσιμότητα των ανθρώπινων πόρων, που αφορά τους διδάσκοντες και τους εκπαιδευτικούς, καθορίζεται από τις διαθέσιμες ώρες διδασκαλίας που έχουν οι ίδιοι οι άνθρωποι. Από την άλλη πλευρά, η διαθεσιμότητα των αιθουσών καθορίζεται από το σύστημα που εφαρμόζει το πανεπιστήμιο. Κάποιες αίθουσες μπορεί να είναι διαθέσιμες κατά τη διάρκεια όλων των περιόδων, ενώ άλλες μπορεί να έχουν περιορισμένη διαθεσιμότητα. Για παράδειγμα, σε πολλά ελληνικά πανεπιστήμια, διάφορα τμήματα μπορεί να μοιράζονται τις ίδιες αίθουσες, οπότε μια αίθουσα μπορεί να είναι διαθέσιμη για ένα τμήμα για μερικές ημέρες της εβδομάδας και για άλλα τμήματα τις υπόλοιπες ημέρες. Αυτή η διάκριση μεταξύ των διαθέσιμων αιθουσών είναι σημαντική για τον χρονοπρογραμματισμό, καθώς επηρεάζει τη διαδικασία επιλογής αιθουσών για τη διεξαγωγή των μαθημάτων. Επιπλέον, οι περιορισμοί σχετικά με τη διαθεσιμότητα των αιθουσών μπορούν να οδηγήσουν σε δυσκολίες και περιορισμούς κατά τον σχεδιασμό του προγράμματος, καθώς ενδέχεται να πρέπει να ληφθούν υπόψη οι διαφορετικές διαθεσιμότητες για τις διάφορες ημέρες της εβδομάδας και τις διάφορες περιόδους της ημέρας.

4.2 Οντότητες του προβλήματος

Ένα γεγονός $e \in E$ έχει διάρκεια $D_e \in \mathbb{N}$ και απαιτήσεις από πόρους $er \in ER_e$. Υπάρχει περίπτωση για ένα πόρο $r \in R$ να είναι προκαθορισμένο ότι θα ικανοποίηση μια απαίτηση ενός γεγονότος $er \in ER_e$. Η παράμετρος $PA_{er} \in \{1, 0\}$ παίρνει τιμή 1 εάν η απαίτηση του γεγονότος er είναι προκαθορισμένη, και 0 στην διαφορετική περίπτωση, επίσης η παράμετρος $PR_{er,r} \in \{1, 0\}$ παίρνει τιμή 1 εάν ο πόρος r είναι προκαθορισμένος για την απαίτηση του γεγονότος er , και 0 στην διαφορετική περίπτωση. Υπο-γεγονός se ορίζεται το απόσπασμα ενός γεγονότος $e \in E$. Το κάθε γεγονός έχει διάρκεια $D_{se} \leq D_e$ και κληρονομεί τις απαιτήσεις πόρων από το κεντρικό γεγονός.

Οντότητα Ορισμός

T Χρόνος: συμβολίζει μια χρονοθυρίδα στο πρόβλημα μας.

TG Ομάδα χρόνων: ένα σύνολο από χρόνους T

R Πόρος: είναι μια οντότητα η οποία πρέπει να παραβρίσκεται σε ένα γεγονός. Ως πόρο ορίζουμε μια αίθουσα, έναν καθηγητή ή έναν μαθητή.

RG Ομάδα πόρων: Ένα σύνολο από πόρους R

E Γεγονός: είναι μια οντότητα η οποία περιγράφει ότι κάποιοι πόροι R πρέπει να συναντηθούν για ένα συγκεκριμένο χρονικό διάστημα

EG Ομάδα γεγονότων: Ένα σύνολο από γεγονότα E

C Περιορισμός: περιγράφει έναν κανόνα που θα πρέπει να σεβαστεί ο επιλυτής.

Ορίζουμε ως SE το σύνολο όλων των υπο-γεγονότων στο πρόβλημα μας και ως $se \in SE_e$ το σύνολο των υπο-γεγονότων για ένα συγκεκριμένο γεγονός e . D_e το άθροισμα όλων των διαρκειών των υπο-γεγονότων για ένα

συγκεκριμένο γεγονός $e \in E$. Επίσης, οι χρόνοι T είναι ταξινομημένοι με χρονολογική σειρά και ως, p_t συμβολίζουμε τον δείκτη του ενός χρόνου $t \in T$. Επιπλέον, ως $t \in T_{tg}$ συμβολίζουμε τους χρόνους οι οποίοι ανήκουν στο σύνολο χρόνων tg για κάθε $tg \in TG$. Η παρακάτω εξίσωση (1) συμβολίζει όλο το σύνολο χρονοθυρίδων που μπορεί να λάβει μέρος ένα υπο-γεγονός $se \in SE$ εάν ξεκινήσει μια χρονική στιγμή $t \in T$.

$$T_{se,t}^{start} = \{t' \in T \mid p_t - D_{se} + 1 \leq p_{t'} \leq p_t\} \quad (1)$$

Τέλος, ο κάθε περιορισμός $c \in C$ αφορά συγκεκριμένα γεγονότα, πόρους ή ομάδες γεγονότων. Επομένως οι συμβολισμοί: $e \in E_c, r \in R_c$ and $eg \in EG_c$ απεικονίζουν σε αντιστοιχία, τις οντότητες στις οποίες εφαρμόζεται ο περιορισμός.

4.2.1 Η οντότητα του Χρόνου – (Times)

Η οντότητα χρόνος (Times) περιέχει πληροφορίες για την ελάχιστη μμονάδα διδασκαλίας (την διδακτική ώρα, χρονοθυρίδα ή timeslot) και ένα σύνολο από αυτές ορίζει την ομάδα χρόνων (Time Group). Ως ομάδα χρόνων μπορεί να οριστεί ένα δίωρο διδασκαλίας σε μια συγκεκριμένη ημέρα ή όλα τα σύνολα των χρονοθυρίδων της 5^{ης} ώρας από κάθε ημέρα διδασκαλίας. Για παράδειγμα στο σύστημα μας το:

```
"Time": [
  {
    "Id": "Mo_6",
    "Name": "Monday 6",
    "Day": "Δευτέρα",
    "TimeGroups": ["BeforeLaunch"]
  },
]
```

Ορίζει μια χρονική περίοδο Time, με αναγνωριστικό MO_6, την ημέρα Δευτέρα που ανήκει στην χρονική ομάδα “BeforeLaunch”.

4.2.2 Η οντότητα των πόρων (Resources).

Ο πόρος περιέχει πληροφορίες σχετικά με τους πόρους που έχει το εκπαιδευτικό ίδρυμα. Οι πόροι χωρίζονται σε τρεις μεγάλες κατηγορίες, οι οποίες είναι ο καθηγητής (Teacher), η αίθουσα διδασκαλίας (Room) και το τμήμα διδασκαλίας (Class). Για παράδειγμα το:

```
"Resource": [
  {
    "Id": "gr_R001",
    "Name": "Αμφιθέατρο",
    "Capacity": 100,
    "ResourceType": "Room",
    "ResourceGroups": ["ΜεγάληΑίθουσα"]
  }
]
```

Ορίζει έναν πόρο τύπου αίθουσα (Room) με αναγνωριστικό gr_R001 και χωρητικότητα 100 θέσεων. Επίσης, αυτός ο πόρος ανήκει στην ομάδα πόρων (ResourceGroup) Μεγάλη Αίθουσα. Αντίστοιχες ομάδες (Groups) ορίζονται, για τους καθηγητές και τα τμήματα.

4.2.3 Η Οντότητα γεγονός (Event).

Όπως αναφέραμε και στην εισαγωγή αυτού του κεφαλαίου, ένα γεγονός (event) καθορίζει ότι συγκεκριμένοι πόροι (resources) πρέπει να δεσμευτούν για συγκεκριμένο αριθμό από χρονικές στιγμές (times) που ονομάζονται διάρκεια (duration). Το κάθε γεγονός θα πρέπει να ανήκει σε ένα και μόνο ένα συγκεκριμένο μάθημα (Course).). Για παράδειγμα το:

```

"Event": [
  {
    "Id": "Event_C001",
    "Name": "ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ Θ",
    "Duration": 2,
    "Course": "ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ",
    "Resources": [
      {
        "Id": "Semester1",
        "ResourceType": "Class"
      },
      {
        "Id": ["gr_T001", "gr_T002"],
        "ResourceType": "Teacher"
      },
      {
        "ResourceType": "Room"
      }
    ],
    "EventGroups": ["gr_Events"]
  }
]

```

Ορίζει ένα γεγονός (Event) με αναγνωριστικό Event_C001 που ανήκει στο μάθημα (Course) ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ και έχει τις εξής απαιτήσεις: χρειάζεται έναν πόρο τύπου καθηγητή (Teacher) αναμεσα στους δυο πιθανούς καθηγητές με αναγνωριστικά gr_T001, gr_T002. Επίσης, χρειάζεται έναν πόρο τύπου αίθουσα (Room) και εντάσσεται στα μαθήματα του 1^{ου} εξαμήνου (Semester 1).

4.3 Μεταβλητές

Η βασική μεταβλητή απόφασης για το πρόβλημα μας είναι η παρακάτω:

$$x_{se,t,er,r} = \begin{cases} 1, & \text{Εαν το υπο - γεγονός } se \text{ ξεκίνησε την χρονική στιγμή } t \text{ και ο} \\ & \text{πόρος } r \text{ ανατέθηκε στην απαίτηση πορου } er \\ 0, & \text{Ενναλακτικά} \end{cases}$$

Επιπλέον, χρησιμοποιήθηκαν και οι παρακάτω βοηθητικές μεταβλητές:

- $y_{se,t}$ = 1 εάν το υπογεγονός se ξεκίνησε την χρονική στιγμή t ; 0 ενναλακτικά.
- $w_{se,er,r}$ = 1 εάν στο υπογεγονός se ανατέθηκε ο πόρος r για να ικανοποίηση την απαίτηση er ; 0 ενναλακτικά.
- $u_{t,r}$ = Οι φορές που χρησιμοποιήθηκε ο πόρος r την χρονική στιγμή t .
- $q_{r,tg}$ = 1 εάν ο πόρος r είναι απασχολημένος την χρονική στιγμή t ; 0 ενναλακτικά.
- $p_{r,tg}$ = 1 εάν ο πόρος r είναι απασχολημένος τουλάχιστον μια φορά στο σύνολο χρόνων tg ; 0 ενναλακτικά.
- $o_{e,t}$ = 1 εάν τουλάχιστον ένα υπογεγονός του γεγονότος e ανατέθηκε στην χρονοθυρίδα t ; 0 ενναλακτικά.
- $l_{eg,t}$ = 1 εάν τουλάχιστον ένα γεγονός του συνόλου γεγονότων eg ανατέθηκε στην χρονοθυρίδα t ; 0 ενναλακτικά.
- $k_{eg,r}$ = 1 εάν ο πόρος r σε τουλάχιστον ένα γεγονός του συνόλου γεγονότων eg ; 0 ενναλακτικά.

$$\begin{aligned}
 h_{r,tg,t} &= 1 \text{ εάν ο πόρος } r \text{ έχει κενό χρόνο στην χρονοθυρίδα } t \text{ στο σύνολο χρόνων } tg; 0 \text{ εναλλακτικά.} \\
 h_e^{first} &= \text{Ο τακτικός αριθμός της πρώτης χρονοθυρίδας που ανατέθηκε σε οπουδήποτε υπογεγονός του γεγονότος } e. \\
 h_e^{last} &= \text{Ο τακτικός αριθμός της τελευταίας χρονοθυρίδας που ανατέθηκε σε οπουδήποτε υπογεγονός του γεγονότος } e.
 \end{aligned}$$

4.4 Περιορισμοί (Constraints)

Εκτός από τους περιορισμούς που αναγράφονται στο XHSTT, χρειάστηκε να προστεθούν κάποιοί βασικοί περιορισμοί ώστε να διασφαλιστεί η συνεπεία του μοντέλου.

Αρχικά, θα πρέπει το κάθε υπογεγονός να ξεκινήσει μόνο σε μια χρονική περίοδο και το πλήθος των πόρων που θα ικανοποιήσουν αυτό το υπογεγονός να είναι ακριβώς ίσο με το πλήθος των συνολικών απαιτήσεων του κεντρικού γεγονότος.

$$\sum_{t \in T} \sum_{r \in R} x_{se,t,er,r} = 1 \quad \begin{aligned} &\forall se \in SE \\ &\forall er \in ER_{se} \end{aligned} \quad (2)$$

Η ακόλουθη μεταβλητή περιορισμού $y_{se,t}$ μαζί με τον περιορισμό (2) εξασφαλίζουν ότι ένα υπογεγονός δεν μπορεί να εξαπλωθεί σε μη συνεχόμενες χρονικές περιόδους. Ορίζουμε ως $|ER|_{se}$ το πλήθος των απαιτήσεων από πόρους που μπορεί να έχει ένα υπογεγονός $se \in SE$.

$$\sum_{er \in ER} \sum_{r \in ER_{er}} x_{se,t,er,r} = |ER_{se}| \times y_{se,t} \quad \begin{aligned} &\forall se \in SE \\ &\forall t \in T \end{aligned} \quad (3)$$

Ο ορισμός για την μεταβλητή $u_{t,r}$ εμφανίζεται στον περιορισμό (4)

$$\sum_{se \in SE} \sum_{er \in ER_{se}} \sum_{t' \in T_{se,t}^{start}} x_{se,t',er,r} = u_{t,r} \quad \begin{aligned} &\forall t \in T \setminus \{t_D\} \\ &\forall r \in R \end{aligned} \quad (4)$$

Ο ορισμός για την μεταβλητή $w_{se,er,r}$ προκύπτει από τον παρακάτω περιορισμό:

$$\sum_{t \in T} x_{se,t,er,r} = w_{se,er,r} \quad \begin{aligned} &\forall se \in SE \\ &\forall er \in ER_{se} \\ &\forall r \in R \end{aligned} \quad (5)$$

Ο παρακάτω περιορισμός περιγράφει ότι: Σε ένα υπογεγονός δεν μπορεί να ξεκινήσει σε χρονική περίοδο στην οποία δεν έπονται αρκετές χρονοθυρίδες ώστε να ικανοποιήσουν την διάρκεια του.

$$y_{se,t} = 0 \quad \begin{aligned} &\forall se \in SE \\ &\forall t \in T \setminus \{TD\}, p_t + D_{se} - 1 > |T| \end{aligned} \quad (6)$$

Η συνολική διάρκεια όλων των υπο-γεγονότων πρέπει να ίση με την συνολική διάρκεια του βασικού γεγονότος:

$$\sum_{se \in SE_e} D_{se} = D_e \quad \forall e \in E \quad (7)$$

Ένα μεγάλο πλήθος περιορισμών απαιτούν η απόκλιση $V \in \mathbb{N}$ πρέπει να είναι εντός ενός ανώτατου ορίου $\bar{B}_c \in \mathbb{N}$ και ενός κατωτάτου ορίου $\underline{B}_c \in \mathbb{N}$. Έτσι ώστε η ποινή να είναι ίση με τιμή την οποία η απόκλιση ξεπερνάει το ανώτατο όριο \bar{B}_c ή υπολείπεται του κατωτάτου ορίου \underline{B}_c . To simplify notation for these cases, we introduce the function $U_{\underline{B}_c, \bar{B}_c} V$ which is defines as follows:

$$s \geq U_{\underline{B}_c, \bar{B}_c} V \Rightarrow \begin{cases} s \geq V - \bar{B}_c \\ s \geq \underline{B}_c - V \end{cases} \quad (8)$$

Λέμε ότι ένας πόρος είναι απασχολημένος μια χρονική περίοδο αν ικανοποιεί τουλάχιστον ένα γεγονός εκείνη την χρονική περίοδο και ότι είναι απασχολημένος σε μια ομάδα χρόνων TG αν ικανοποιεί μια ή περισσότερες χρονικές περιόδους $t \in TG$. Έστω ότι η μεταβλητή $q_{r,t} \in \{0,1\}$ παίρνει τιμή 1 εάν ο πόρος $r \in R$ είναι απασχολημένος την χρονική περίοδο $t \in T$, 0 εναλλακτικά. Παρομοίως, η μεταβλητή $p_{r,tg} \in \{0,1\}$ παίρνει τιμή 1 εάν ο πόρος $r \in R$ είναι απασχολημένος κάποια χρονική περίοδο μέσα στην ομάδα χρόνων $tg \in TG$, και 0 στην διαφορετική περίπτωση.

$$|SE| \times q_{r,t} \geq u_{t,r} \quad \begin{matrix} \forall r \in R \\ \forall t \in T \setminus \{t_D\} \end{matrix} \quad (9)$$

$$q_{r,t} \leq u_{t,r} \quad \begin{matrix} \forall r \in R \\ \forall t \in T \setminus \{t_D\} \end{matrix} \quad (10)$$

$$p_{r,tg} \geq q_{r,t} \quad \begin{matrix} \forall r \in R \\ \forall tg \in TG \\ \forall t \in T_{tg} \end{matrix} \quad (11)$$

$$p_{r,tg} \leq \sum_{t \in T_{tg}} q_{r,t} \quad \begin{matrix} \forall r \in R \\ \forall tg \in TG \end{matrix} \quad (12)$$

Οι περιορισμοί (9) και (11) καθορίζουν τα κατώτατα όρια για τις μεταβλητές $q_{r,t}$ και $p_{r,tg}$, δηλαδή εξασφαλίζουν ότι θα πάρουν τιμή 1 στην περίπτωση που ο πόρος είναι πραγματικά απασχολημένος στην χρονική περίοδο / ομάδα χρόνων. Από την άλλη, οι περιορισμοί (10) και (12) εξασφαλίζουν ότι η τιμή των μεταβλητών $q_{r,t}$ και $p_{r,tg}$ θα είναι 0 στην περίπτωση ο πόρος δεν είναι απασχολημένος.

4.4.1 Ανάθεση πόρου (Assign Resource) C^1

Συνήθως, κάθε γεγονός χρειάζεται πόρους που είναι απαραίτητοι για τη διεξαγωγή του. Ενίοτε αυτοί οι πόροι είναι προκαθορισμένοι. Για παράδειγμα, συνήθως είναι γνωστό ποιοι φοιτητές έχουν ανατεθεί σε ποιο γεγονός. Ωστόσο, σε ορισμένες περιπτώσεις των προβλημάτων HSTT υπάρχουν αποφάσεις για την ανάθεση πόρων που πρέπει να ληφθούν από τον σχεδιαστή. Ο περιορισμός Assign Resource ελέγχει εάν έχουν ληφθεί αυτές οι αποφάσεις. Προφανώς, δεν είναι μόνο σημαντικό ότι σε κάθε γεγονός έχουν ανατεθεί οι απαιτούμενοι πόροι, αλλά έχει σημασία και ποιοι πόροι ανατέθηκαν. Για παράδειγμα, το μάθημα ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ χρειάζεστε καθηγητές που μπορούν να διδάξουν το μάθημα.

Αυτός ο περιορισμός ελέγχει εάν οι απαιτήσεις πόρων ενός γεγονότος ικανοποιούνται. Μια απαίτηση πόρου θα πρέπει να είναι διαθέσιμη σε όλη την διάρκεια ενός γεγονότος. Είναι σημαντικό να αναφερθεί ότι αυτός ο περιορισμός ελέγχει κατά πόσο τελικά ανατέθηκε ένας πόρος συγκεκριμένου τύπου στο γεγονός, δεν ελέγχει ποιος τελικά είναι αυτός (βλέπε 4.4.4).

Η απόκλιση $s_{c,er}^1$ για κάθε γεγονός αυτού του περιορισμού υπολογίζεται από την ακόλουθη εξίσωση:

$$D_e - \sum_{se \in SE_e} \sum_{r \in R_{er} \setminus \{r_D\}} D_{se} \times w_{se,er,r} = s_{c,er}^1 \quad \begin{array}{l} \forall c \in C^1 \\ \forall e \in E_c \\ \forall er \in ER_e: role_{er} = role_c \end{array} \quad (13)$$

Στο δικό μας μοντέλο ο παραπάνω περιορισμός παρουσιάζεται μόνο στην αυστηρή μορφή του (Hard Constraint). Άρα, δεν νοείται λύση για την οποία σε ένα οποιοδήποτε γεγονός δεν έχουν ικανοποιηθεί όλες οι απαιτήσεις πόρων.

```

"Event": [
  {
    "Id": "Event_C001",
    "Name": "ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ Θ",
    "Duration": 2,
    "Course": "ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ",
    "Resources": [
      {
        "Id": "Semester1",
        "ResourceType": "Class"
      },
      {
        "Id": ["gr_T001", "gr_T002"],
        "ResourceType": "Teacher"
      },
      {
        "ResourceType": "Room"
      }
    ]
  },
  "EventGroups": ["gr_Events"]
]

```

Στο παραπάνω παράδειγμα βλέπουμε ότι το γεγονός με αναγνωριστικό Event_C001 έχει απαίτηση από ένα καθηγητή, και μια απαίτηση από μια αίθουσα.

4.4.2 Ανάθεση χρόνου (Assign Time) C^2

Αυτός ο περιορισμός ελέγχει εάν σε ένα γεγονός έχει ανατεθεί σε κάποια χρονική περίοδο. Όπως και στην Ανάθεση Πόρου (4.4.1) δεν ελέγχουμε ποια χρονική στιγμή ανατέθηκε στο γεγονός αλλά κατά πόσο τελικά αυτό το γεγονός κατάφερε να τοποθετηθεί μέσα στο πρόγραμμα.

Η απόκλιση $s_{c,er}^2$ για κάθε γεγονός αυτού του περιορισμού υπολογίζεται από την ακόλουθη εξίσωση:

$$D_e - \sum_{t \in T \setminus \{t_D\}} \sum_{se \in SE_e} D_{se} \times y_{se,t} = s_{c,er}^2 \quad \begin{array}{l} \forall c \in C^2 \\ \forall e \in E_c \end{array} \quad (14)$$

Σε αντίθεση με την Ανάθεση Πόρου (4.4.1) αυτός ο περιορισμός μπορεί να εμφανιστεί και ως ελαστικός (Soft Constraint). Ο λόγος που έδωσα την δυνατότητα αυτή είναι διότι ήθελα να καλύψω τις περιπτώσεις που το μοντέλο μας δεν καταφέρει να χρονοπρογραμματίσει όλα τα γεγονότα.

```

"Event": [
  {
    "Id": "Event_C001",
    "Name": "ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ Θ",
    "Duration": 2,
    "Course": "ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ",
    "Resources": [
      {
        "Id": "Semester1",
        "ResourceType": "Class"
      },
      {
        "Id": ["gr_T001", "gr_T002"],
        "ResourceType": "Teacher"
      },
      {
        "ResourceType": "Room"
      }
    ],
    "EventGroups": ["gr_Events"]
  }
]

```

Στο παραπάνω παράδειγμα βλέπουμε ότι το γεγονός με αναγνωριστικό Event_C001 έχει συνολική διάρκεια δυο χρονοθυρίδων. Αρά, στο τελικό πρόγραμμα που θα παράξει το μοντέλο θα πρέπει το γεγονός αυτό να δεσμεύει ακριβώς δυο χρονοθυρίδες.

4.4.3 Διαχωρισμός γεγονότων (Split Events) C^3

Ο περιορισμός αυτός ορίζει όρια στον αριθμό των υπο-γεγονότων που μπορεί να προέλθουν από ένα γεγονός και στην διάρκεια τους. Οι παράμετροι $\underline{B}_c^{amt} \in \mathbb{N}$ and $\overline{B}_c^{amt} \in \mathbb{N}$ συμβολίζουν τον ελάχιστο και μέγιστο αριθμό από υπο-γεγονότα που μπορούν να παραχθούν από ένα γεγονός. Έστω $\underline{B}_c^{dur} \in \mathbb{N}$ και $\overline{B}_c^{dur} \in \mathbb{N}$ η ελάχιστη και μέγιστη διάρκεια που μπορεί να έχει ένα γεγονός. Το κόστος παραβίασης αυτού του περιορισμού προκύπτει από τον αριθμό

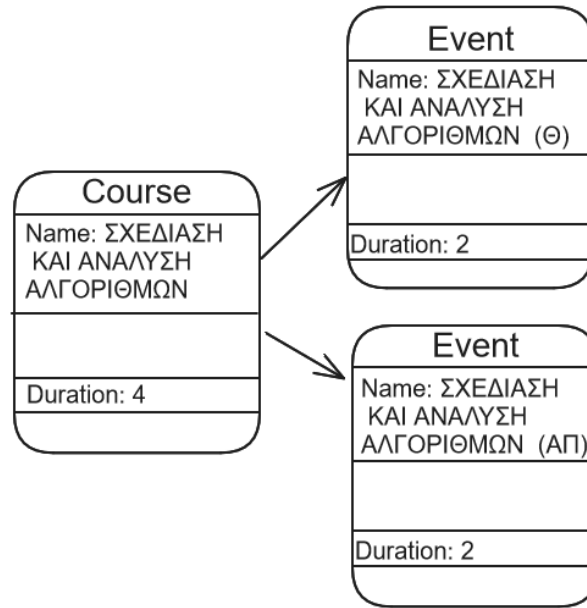
των υπο-γεγονότων που η διάρκεια είναι μικρότερη από \underline{B}_c^{dur} και μεγαλύτερη από \overline{B}_c^{dur} ($s_{c,e}^{3a}$), και την διαφορά του τελικού πλήθους υπο-γεγονότων από ελάχιστη \underline{B}_c^{amt} ή μέγιστη \overline{B}_c^{amt} ($s_{c,e}^{3b}$) ποσότητα.

$$\sum_{\substack{se \in SE_e: \\ (\underline{B}_c^{dur} > D_{se}) \vee (\overline{B}_c^{dur} < D_{se})}} u_{se} = s_{c,e}^{3a} \quad \begin{array}{l} \forall c \in C^3 \\ \forall e \in E_c \end{array} \quad (15)$$

$$\overline{B}_c^{amt} - \sum_{se \in SE_e} u_{se} \leq s_{c,e}^{3b} \quad \begin{array}{l} \forall c \in C^3 \\ \forall e \in E_c \end{array} \quad (16)$$

$$\sum_{se \in SE_e} u_{se} - \overline{B}_c^{amt} \leq s_{c,e}^{3b} \quad \begin{array}{l} \forall c \in C^3 \\ \forall e \in E_c \end{array} \quad (17)$$

Με αποτέλεσμα η απόκλιση για αυτόν τον περιορισμό $c \in C^3$ να δίνεται από: $s_{c,e}^{3a} + s_{c,e}^{3b}$. Όμως, στο δικό μας μοντέλο θεωρούμε ότι ο καθηγητής είναι υπεύθυνος για τον αριθμό και την διάρκεια των γεγονότων που αποτελείται ένα μάθημα.



Σχήμα 4.1 Παράδειγμα διαχωρισμού γεγονότων

Στην παραπάνω εικόνα βλέπουμε ότι οι ώρες διδασκαλίας για το μάθημα ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ είναι 4 και διαχωρίστηκαν σε δυο υπο-γεγονότα με 2 διδακτικές ώρες το καθένα.

4.4.4 Επιθυμητοί πόροι (Prefer Resource) C^4

Αυτός ο περιορισμός περιγράφει ότι κάποιο γεγονός μπορεί να έχει προτίμηση σε συγκεκριμένους πόρους. Το κόστος παραβίασης αυτού του περιορισμού προκύπτει από το άθροισμα της διάρκειας όλων των υπο-γεγονότων των οποίων δεν ανατέθηκε τελικά κάποιος από τους επιθυμητούς πόρους.

```

"Event": [
  {
    "Id": "Event_C001",
    "Name": "ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ Θ",
    "Duration": 2,
    "Course": "ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ",
    "Resources": [
      {
        "Id": "Semester1",
        "ResourceType": "Class"
      },
      {
        "Id": ["gr_T001", "gr_T002"],
        "ResourceType": "Teacher"
      },
      {
        "ResourceType": "Room"
      }
    ],
    "EventGroups": ["gr_Events"]
  }
]

```

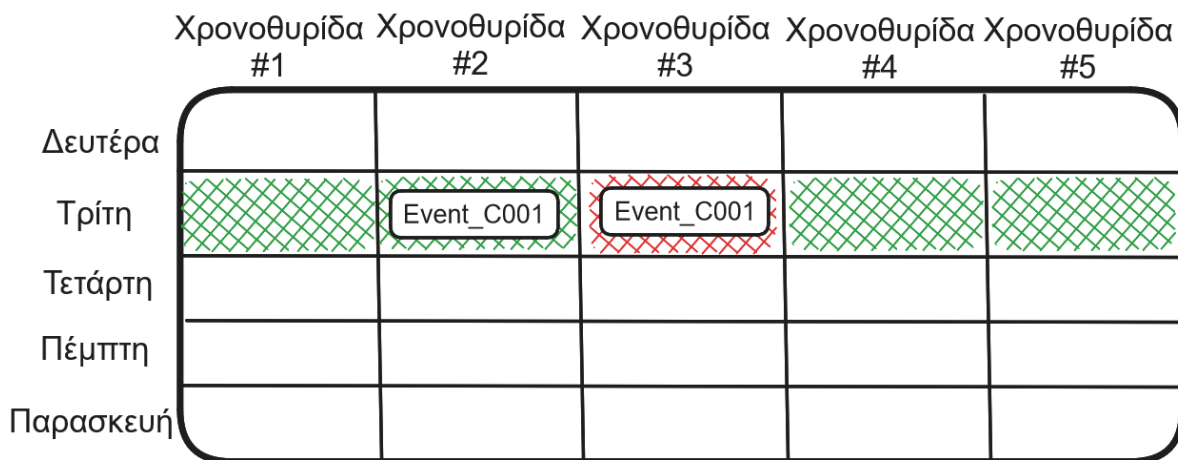
Στο παραπάνω παράδειγμα βλέπουμε ότι το γεγονός με αναγνωριστικό Event_C001 έχει απαίτηση από ένα καθηγητή. Επίσης, έχει δηλωθεί ότι ο τελικός καθηγητής που θα ανατεθεί στο γεγονός θα πρέπει αναγκαστικά να είναι είτε ο καθηγητής με αναγνωριστικό gr_T001 είτε με αναγνωριστικό gr_T002. Από την άλλη, το γεγονός αυτό δεν έχει προτίμηση σε συγκεκριμένη αίθουσα.

Έστω R_c είναι οι επιθυμητοί πόροι για ένα γεγονός αυτό του περιορισμού. Η απόκλιση $s_{c,er}^4$ για κάθε γεγονός αυτού του περιορισμού υπολογίζεται από την ακόλουθη εξίσωση:

$$\sum_{se \in SE_e} \sum_{\substack{r \notin R_D, r \notin R_c \\ r \notin R_D, r \notin R_c}} D_{se} \times w_{se,er,r} = s_{c,er}^4 \quad \begin{array}{l} \forall c \in C^4 \\ \forall e \in E_c \\ \forall er \in ER_E, PA_{er} = 0 \wedge role_{er} = role_c \end{array} \quad (18)$$

4.4.5 Επιθυμητοί χρόνοι (Prefer Times) C^5

Όπως και στο περιορισμό 4.4.4, μερικά γεγονότα μπορεί να έχουν προτίμηση να προγραμματιστούν σε συγκεκριμένες περιόδους. Το κόστος παραβίασης αυτού του περιορισμού προκύπτει από το άθροισμα της διάρκειας όλων των υπο-γεγονότων για τα οποία δεν ανατέθηκε τελικά κάποια επιθυμητή περίοδος.



Σχήμα 4.2 Παράδειγμα περιορισμού Επιθυμητοί χρόνοι

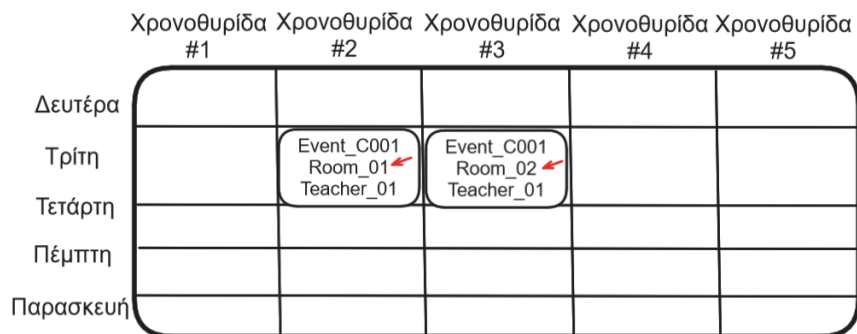
Στο παραπάνω παράδειγμα βλέπουμε ότι το γεγονός με αναγνωριστικό Event_C001 έχει προτίμηση να προγραμματιστεί μόνο την ημέρα Τρίτη και στο διάστημα [#1, #2, #4, #5]. Ωστόσο, το γεγονός προγραμματίστηκε και στη χρονοθυρίδα #3 με αποτέλεσμα να μην σεβαστεί ο συγκεκριμένος περιορισμός.

Έστω T_c είναι οι επιθυμητοί χρόνοι για ένα γεγονός αυτό του περιορισμού. Η απόκλιση $s_{c,er}^5$ για κάθε γεγονός αυτού του περιορισμού υπολογίζεται από την ακόλουθη εξίσωση:

$$\sum_{se \in SE_e} \sum_{t \notin T_D, t \notin T_c \wedge D_c = D_{se}} D_{se} \times y_{se,t} = s_{c,er}^5 \quad \begin{array}{l} \forall c \in C^5 \\ \forall e \in E_c \end{array} \quad (19)$$

4.4.6 Αποφυγή διαχωρισμού αναθέσεων (Avoid split Assignments) C^6

Όταν ένα γεγονός χωρίζεται σε υπο-γεγονότα υπάρχει πιθανότητα στο κάθε υπογεγονός να ανατεθεί διαφορετικός πόρος. Αυτός ο περιορισμός τιμωρεί την ανάθεση διαφορετικών πόρων ίδιας κατηγορίας στην επιλεγμένη ομάδα γεγονότων. Η μεταβλητή $k_{eg,r} \in \{0,1\}$ παίρνει τιμή 1 εάν στο γεγονός e ανατέθηκε ο πόρος r με σεβασμό στο παρόν περιορισμό.



Σχήμα 4.3 Παράδειγμα περιορισμού Αποφυγή διαχωρισμού αναθέσεων

Συνεχίζοντας με το ίδιο παράδειγμα του κεφαλαίου 4.4.4, βλέπουμε ότι το γεγονός με αναγνωριστικό Event_C001 έχει απαίτηση από μια αίθουσα χωρίς προτίμηση σε κάποια συγκεκριμένη αίθουσα. Αν εφαρμοζόταν αυτός ο περιορισμός για τον πόρο αίθουσα τότε θα έπρεπε σε κάθε χρονοθυρίδα που προγραμματίστηκε αυτό το γεγονός να λάμβανε μέρος στην ίδια αίθουσα.

Η απόκλιση $s_{c,er}^6$ για κάθε γεγονός αυτού του περιορισμού υπολογίζεται από την ακόλουθη εξίσωση:

$$\sum_{\substack{er \in ER_e, PA_{er}=0 \\ role_c=role_{er}}} w_{se,er,r} \leq k_{eg,r} \quad \begin{array}{l} \forall c \in C^6 \\ \forall r \in R \\ \forall eg \in EG_c \\ \forall e \in E_{eg} \\ \forall se \in SE_e \end{array} \quad (20)$$

$$\sum_{r \in R} k_{eg,r} - 1 \leq s_{c,eg}^7 \quad \begin{array}{l} \forall c \in C^6 \\ \forall eg \in EG_c \end{array} \quad (21)$$

4.4.7 Διασπορά γεγονότων (Spread Events) C^7

Η διασπορά γεγονότων έχει απόκλιση για την κάθε χρονική ομάδα $tg \in TG_c \in C^8$. Εστω ότι $\underline{B}_{c,tg}$ και $\overline{B}_{c,tg}$ είναι ο ελάχιστος και ο μεγιστος αριθμος υπο-γεγονότων που μπορούν να προγραμματιστούν σε μια χρονική ομάδα tg του περιορισμού c^7 . Η αποκλιση $s_{c,eg,tg}^8$ για την κάθε χρονική ομάδα δίνεται από τα υπο γεγονότα τα οποία είναι κάτω $\underline{B}_{c,tg} \in \mathbb{N}$ ή πάνω από $\overline{B}_{c,tg} \in \mathbb{N}$.

	Χρονοθυρίδα #1	Χρονοθυρίδα #2	Χρονοθυρίδα #3	Χρονοθυρίδα #4	Χρονοθυρίδα #5
Δευτέρα	Event 1				
Τρίτη					
Τετάρτη			Event 2	Event 3	
Πέμπτη					
Παρασκευή		Event 4			

Σχήμα 4.4 Παράδειγμα περιορισμού Διασπορά γεγονότων

Αυτός ο περιορισμός θέτει όρια στις αναθέσεις μιας ομάδας γεγονότων. Για παράδειγμα, υποθέστε ότι αυτός ο περιορισμός εφαρμόζεται σε μια ομάδα γεγονότων, δηλαδή την EventGroupA που αποτελείται από το Event1, Event2, Event3 και Event4. Υποθέστε ότι ο αριθμός των γεγονότων περιορίζεται από το εύρος 0-1 για κάθε ημέρα της εβδομάδας. Τότε στο Σχήμα 4.4 Παράδειγμα περιορισμού Διασπορά γεγονότων, βλέπουμε ότι υπάρχει ένα γεγονός προγραμματισμένο τη Δευτέρα, κανένα την Τρίτη, δύο την Τετάρτη, κανένα την Πέμπτη και ένα την Παρασκευή. Δηλαδή, για κάθε ημέρα, υπάρχουν 0-1 γεγονότα, εκτός από την Τετάρτη. Την Τετάρτη υπάρχουν δύο

γεγονότα προγραμματισμένα, δηλαδή το Event3 και το Event4. Για την ομάδα γεγονότων EventGroupA, αυτός ο περιορισμός παράγει αρκετές αποκλίσεις, μία απόκλιση για κάθε ημέρα. Η απόκλιση υπολογίζεται από τη διαφορά μεταξύ του αριθμού των προγραμματισμένων γεγονότων και του καθορισμένου εύρους 0-1.

Έτσι, η ομάδα EventGroupA έχει απόκλιση μηδέν για τη Δευτέρα, την Τρίτη, την Πέμπτη και την Παρασκευή. Παράγει μια απόκλιση για την Τετάρτη. Αυτές οι πέντε αποκλίσεις συνοψίζονται από τη συνάρτηση κόστους σε ένα κόστος.

$$\underline{B}_{c,tg} - \sum_{se \in SE_e} \sum_{e \in E_{eg}} \sum_{t \in T_{tg}} y_{se,t} \leq s_{c,e,eg,tg}^8 \quad \begin{array}{l} \forall c \in C^8 \\ \forall eg \in EG_c \\ \forall tg \in TG_c \end{array} \quad (22)$$

$$\sum_{se \in SE_e} \sum_{e \in E_{eg}} \sum_{t \in T_{tg}} y_{se,t} - \bar{B}_{c,tg} \leq s_{c,e,eg,tg}^8 \quad \begin{array}{l} \forall c \in C^8 \\ \forall eg \in EG_c \\ \forall tg \in TG_c \end{array} \quad (23)$$

4.4.8 Σύνδεσμος Γεγονότων (Link Events) C^8

Ο περιορισμός γεγονότων περιγράφει ότι ορισμένα γεγονότα θα πρέπει να προγραμματιστούν την ίδια χρονική στιγμή. Για κάθε γεγονός που ανήκει σε μια ομάδα γεγονότων χτίζουμε το σύνολο των χρονών που αυτό προγραμματίστηκε. Η απόκλιση το σύνολο των χρονοθυρίδων που δεν προγραμματίστηκαν ταυτόχρονα όλα τα γεγονότα. Η μεταβλητή $o_{e,t} \in \{0,1\}$ παίρνει τιμή 1 εάν εστω και ένα υπογεγονος του γεγονοτος $e \in E_c$ προγραμματίστηκε στην χρονική στιγμή $t \in T$ και 0 εναλλακτικά. Η μεταβλητή $l_{eg,t} \in \{0,1\}$ παίρνει τιμή 1 ένα εστω και ένα γεγονός της ομάδας γεγονοτων $eg \in EG_c$ ανατεθηκε στην χρονική στιγμή $t \in T$, και 0 εναλλακτικά. Οι περιορισμοί (24) και (26) εξασφαλίζουν ότι οι παραπάνω μεταβλητές παίρνουν σωστές τιμές. Η χαλαρότητα του περιορισμού ορίζεται στην εξίσωση (27). Ο περιορισμός (25) είναι απαραίτητος για να περιορίσει την μεταβλητή $o_{e,t}$ να παρει τιμή 1 στις περιπτώσεις που το γεγονός δεν ανατεθηκε σε καμία χρονική στιγμή.



Σχήμα 4.5 Παράδειγμα περιορισμού Σύνδεσμος Γεγονότων

Για παράδειγμα στο Σχήμα 4.5 ο περιορισμός εφαρμόζεται στην ομάδα γεγονότων 'gr_Event7and8', η οποία αποτελείται από τα γεγονότα Event7 και Event8.

Ο περιορισμός καθορίζει ότι αυτά τα γεγονότα πρέπει να συνδεθούν. Δηλαδή, θα πρέπει να προγραμματιστούν την ίδια ώρα. Βλέπουμε το Event7 προγραμματίζεται την Τρίτη στην χρονοθυρίδα #2, #3, ενώ το Event8 προγραμματίζεται την Τρίτη στην χρονοθυρίδα #3, #4.

Η απόκλιση της ομάδας γεγονότων είναι ο αριθμός των ωρών στις οποίες δεν έχουν προγραμματιστεί όλα τα γεγονότα της ομάδας, αλλά τουλάχιστον ένα από αυτά. Σε αυτήν την περίπτωση, την Τρίτη χρονοθυρίδα #3, και το Event7 και το Event8 έχουν προγραμματιστεί. Αυτό δεν δίνει καμία απόκλιση. Ωστόσο, την Τρίτη χρονοθυρίδα #2, προγραμματίζεται μόνο το Event7. Την Τρίτη, χρονοθυρίδα 4, προγραμματίζεται μόνο το Event8. Επομένως, η απόκλιση για την ομάδα γεγονότων είναι 2.

$$\sum_{t' \in T_{se,t}^{start}} y_{se,t'} \leq o_{e,t} \quad \begin{array}{l} \forall e \in E \\ \forall se \in SE_e \\ \forall t \in T \setminus \{t_D\} \end{array} \quad (24)$$

$$\sum_{se \in SE_e} \sum_{t' \in T_{se,t}^{start}} y_{se,t'} \geq o_{e,t} \quad \begin{array}{l} \forall e \in E \\ \forall se \in SE_e \\ \forall t \in T \setminus \{t_D\} \end{array} \quad (25)$$

$$l_{eg,t} \geq o_{e,t} \quad \begin{array}{l} \forall eg \in EG \\ \forall e \in E_{eg} \\ \forall t \in T \setminus \{t_D\} \end{array} \quad (26)$$

$$l_{eg,t} - o_{e,t} \leq s_{c,eg,t}^8 \quad \begin{array}{l} \forall c \in c^8 \\ \forall eg \in EG_c \\ \forall t \in T \setminus \{t_D\} \end{array} \quad (27)$$

4.4.9 Σειρά γεγονότων (Order Events) C^9

Ο περιορισμός σειρά γεγονότων ορίζει ότι δυο γεγονότα θα πρέπει να προγραμματιστούν με τέτοιο τρόπο ώστε όταν ξεκινήσει το δεύτερο να έχει ήδη ολοκληρωθεί το πρώτο. Έστω οι παράμετροι $\underline{B}_c \in \mathbb{N}$ και $\overline{B}_c \in \mathbb{N}$ συμβολίζουν τον ελαχιστο και τον μεγιστο αριθμο χρονοθυριδων που διαχωριζουν τα δυο γεγονοτα. Έστω επίσης, $(e, e^{\wedge}) \in C$ συμβολίζεται ένα ζευγος γεγονοτων που ανηκει σε αυτό τον περιορισμο. Η μεταβλητή $h_e^{last} \in \mathbb{N}$ είναι ο κανονικος αριθμος της μεγαλιτερης χρονικα τιμης που ανατεθηκε σε ένα οποιοδηποτε υπογεγονος από το γεγονος e . Κατά αντιστοιχια η μεταβλητη $h_e^{first} \in \mathbb{N}$ είναι ο κανονικος αριθμος με την μικροτερη χρονικα τιμη που ανατεθηκε σε ένα οποιοδηποτε υπογεγονος από το γεγονος e^{\wedge} . Τέλος, η απόκλιση δίνεται από το ποσό κατά το οποίο η διαφορά μεταξύ αυτών των δύο αριθμών υπερβαίνει το \overline{B}_c ή υπολείπεται του \underline{B}_c .

$$p_t \times y_{se,t} + D_{se} \leq h_e^{last} \quad \begin{array}{l} \forall c \in C^9 \\ \forall e \in E_c \\ \forall se \in SE_e \\ \forall t \in T \setminus \{t_D\} \end{array} \quad (28)$$

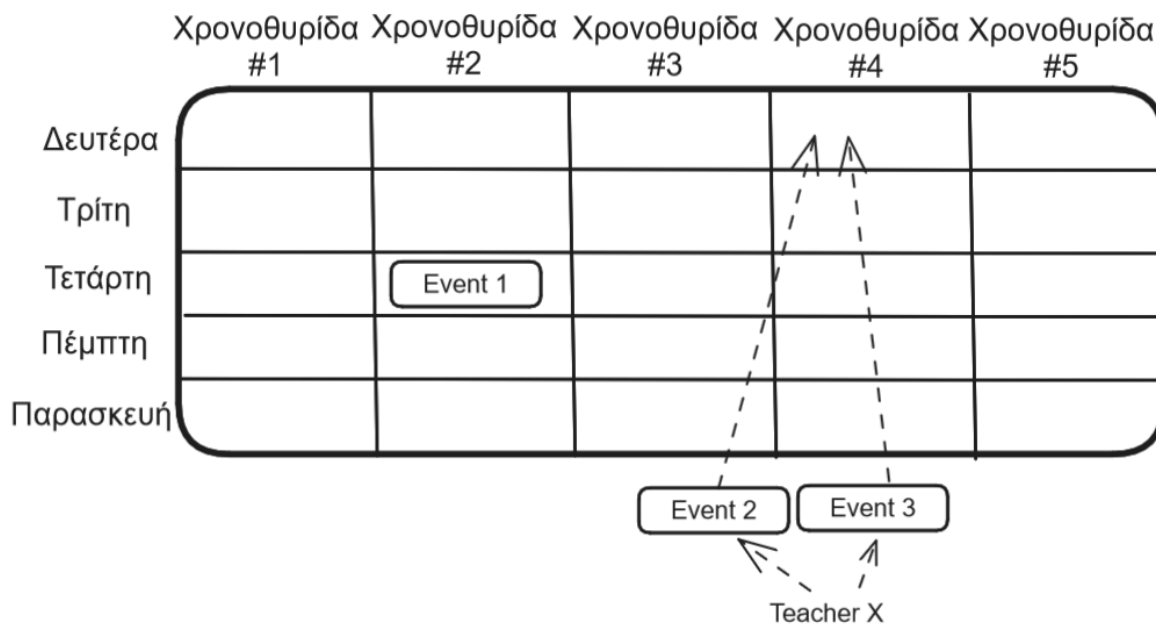
$$|T| - (|T| - p_t) \times y_{se,t} \geq h_e^{first} \quad \begin{array}{l} \forall c \in C^9 \\ \forall e \in E_c \\ \forall se \in SE_e \\ \forall t \in T \setminus \{t_D\} \end{array} \quad (29)$$

$$\underline{B}_c - (h_e^{first} - h_e^{last}) \leq s_{c,e,e^\wedge}^9 \quad \begin{array}{l} \forall c \in C^9 \\ \forall (e, e^\wedge) \in (E, E)_c \end{array} \quad (30)$$

$$(h_e^{first} - h_e^{last}) - \bar{B}_c \leq s_{c,e,e^\wedge}^9 \quad \begin{array}{l} \forall c \in C^9 \\ \forall (e, e^\wedge) \in (E, E)_c \end{array} \quad (31)$$

4.4.10 Αποφυγή Συγκρούσεων (Avoid Clashes Constraint) C^{10}

Αυτός ο περιορισμός καθορίζει ότι ορισμένοι πόροι δεν πρέπει να έχουν συγκρούσεις στα προγράμματά τους. Δηλαδή, δεν πρέπει να ανατίθενται ταυτόχρονα σε δύο ή περισσότερα γεγονότα. Αυτός ο περιορισμός παράγει ένα σύνολο αποκλίσεων για κάθε πόρο. Για να εκτιμηθεί η απόκλιση αυτού του πόρου, για κάθε χρονική στιγμή υπολογίζεται ο αριθμός των εμφανίσεων του δεδομένου πόρου και αφαιρείται ο αριθμός ένα.



Σχήμα 4.6 Παράδειγμα περιορισμού Αποφυγή Συγκρούσεων

Δίνουμε ένα παράδειγμα αυτού του περιορισμού στο Σχήμα 4.6. Ο περιορισμός εφαρμόζεται στους δασκάλους στην ομάδα 'gr_Teachers'. Υποθέτουμε ότι ο Δάσκαλος X από την ομάδα 'gr_Teachers' ανατίθεται σε μια απαίτηση πόρου του Event 2 και σε μία του Event 3. Υποθέτουμε ότι το Event 2 και το Event 3 ανατίθενται και τα δύο στην χρονοθυρίδα #4 της Δευτέρας. Τότε αυτός ο περιορισμός παράγει μια απόκλιση, επειδή ο Δάσκαλος X δεν πρέπει να παρακολουθεί το Event 2 και το Event 3 την ίδια ώρα. Η απόκλιση είναι ίση με τον αριθμό των γεγονότων στα οποία ο πόρος έχει προγραμματιστεί. Έτσι, σε αυτήν την περίπτωση, η απόκλιση είναι 1. Εάν ένας πόρος έχει περισσότερες από 1 ώρες στις οποίες παρακολουθεί περισσότερα από 1 γεγονότων, η απόκλιση του πόρου είναι το άθροισμα των αποκλίσεων σε κάθε ώρα.

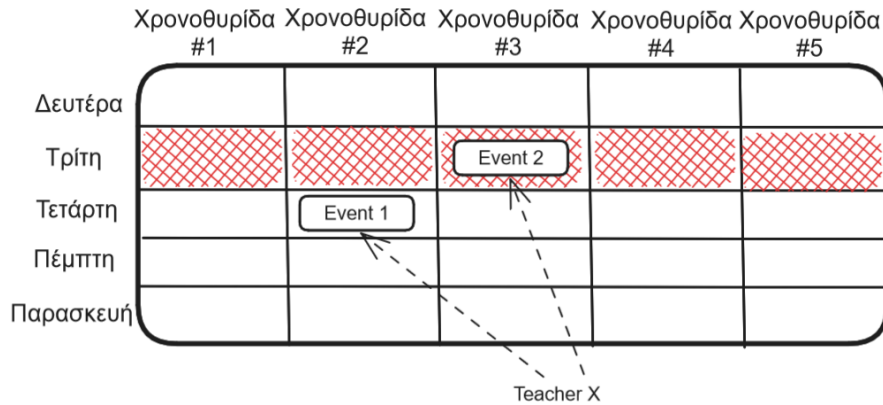
$$u_{t,r} - 1 \leq s_{c,r,t}^{10} \quad \begin{array}{l} \forall c \in C^{10} \\ \forall r \in R_c \\ \forall t \in T \setminus \{t_D\} \end{array} \quad (32)$$

Στην πραγματικότητα όμως αυτός ο περιορισμός εμφανίζεται μόνο ως μη ελαστικός (Hard) . Άρα, οι περιορισμοί (4), (9), (10) και (32) μπορούν να αντικατασταθούν από τον περιορισμό:

$$\sum_{se \in SE} \sum_{er \in ER_{se}} \sum_{\hat{t} \in T_{se,t}^{start}} x_{se,\hat{t},er,r} = q_{r,t} \quad \begin{array}{l} \forall t \in T \setminus \{t_D\} \\ \forall r \in R \end{array} \quad (33)$$

4.4.11 Αποφυγή Μη Διαθέσιμων Χρόνων (Avoid unavailable times) C^{11}

Ο περιορισμός Αποφυγή Μη Διαθέσιμων Χρόνων καθορίζει ότι ορισμένοι πόροι δεν είναι διαθέσιμοι για όλα τα γεγονότα σε ορισμένες χρονικές στιγμές. Αυτό σημαίνει ότι τα γεγονότα δεν πρέπει να προγραμματίζονται σε αυτές τις μη διαθέσιμες ώρες. Για παράδειγμα, στο , ο καθηγητής X έχει ανατεθεί στην απαίτηση πόρου των γεγονότων Event 1 και Event 2. Το Event 1 προγραμματίστηκε στη χρονοθυρίδα #2 της Τετάρτης και το Event 2 προγραμματίστηκε στη χρονοθυρίδα #3 της Τρίτης. Ωστόσο, βλέπουμε ότι ο καθηγητής X δεν είναι διαθέσιμος την Τρίτη. Η απόκλιση, δηλαδή η απόσταση από την ιδανική κατάσταση, είναι ο αριθμός των ωρών κατά τις οποίες ένας πόρος έχει ανατεθεί σε γεγονότα αλλά είναι μη διαθέσιμος. Σε αυτήν την περίπτωση, η απόκλιση είναι 1 ώρα, καθώς ο καθηγητής X έχει ανατεθεί σε μια ώρα την Τρίτη που δεν μπορεί να παρευρεθεί. Αυτή η απόκλιση πρέπει να ελαχιστοποιηθεί για την βέλτιστη παραγωγή του προγράμματος.



Σχήμα 4.7 Παράδειγμα περιορισμού Αποφυγή Μη Διαθέσιμων Χρόνων

Η απόκλιση $s_{c,r}^{11}$ είναι το σύνολο των χρονικών στιγμών που ανατέθηκαν σε ένα μη διαθέσιμο πόρο. Συμβολίζουμε με $t \in T_c$ ένα μη διαθέσιμο χρόνο για τον περιορισμό $c \in C^{11}$:

$$\sum_{t \in T_c} q_{r,t} = s_{c,r}^{11} \quad \begin{array}{l} \forall c \in C^{11} \\ \forall r \in R_c \end{array} \quad (34)$$

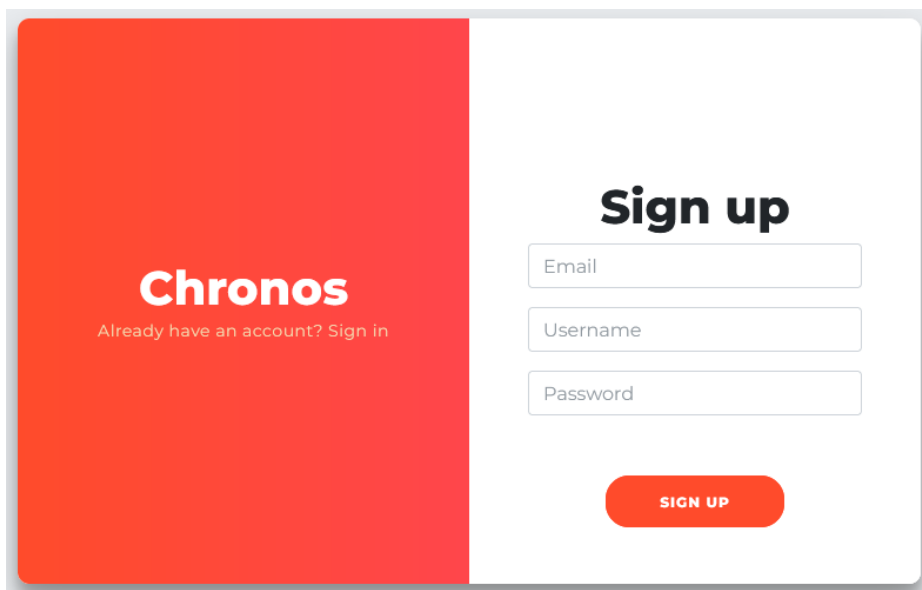
4.5 Objective function

Κάθε περιορισμός συμβάλλει στο κόστος. Κάθε περιορισμός ορίζει μια απόκλιση, ένα βάρος και μια συνάρτηση κόστους. Η απόκλιση είναι το ποσό κατά το οποίο παραβιάζεται ένας περιορισμός. Για κάθε σημείο εφαρμογής, υπολογίζεται μια απόκλιση. Το σημείο εφαρμογής είναι το στοιχείο που προκαλεί μια πιθανή παραβίαση. Ένα σημείο εφαρμογής μπορεί να είναι, για παράδειγμα, ένα γεγονός, ένας πόρος, μια ομάδα χρόνου ή μια ομάδα γεγονότων. Μια συνάρτηση κόστους, που παρέχεται με τον περιορισμό, ορίζει πώς θα αντιστοιχιστούν αυτές οι αποκλίσεις στο σύνολο των μη αρνητικών ακεραίων. Αυτή μπορεί να είναι είτε γραμμική είτε μη γραμμική συνάρτηση. Το βάρος ορίζει τη σημαντικότητα του περιορισμού. Όσο μεγαλύτερο είναι το βάρος, τόσο πιο σημαντικό είναι να ικανοποιηθεί ένας περιορισμός. Για κάθε σημείο εφαρμογής, το κόστος είναι το βάρος πολλαπλασιασμένο με τη συνάρτηση κόστους επί την(τις) απόκλιση(εις).

$$obj = \text{Min} \sum_{j=1}^{11} f(s_{c,r}^j) \quad (35)$$

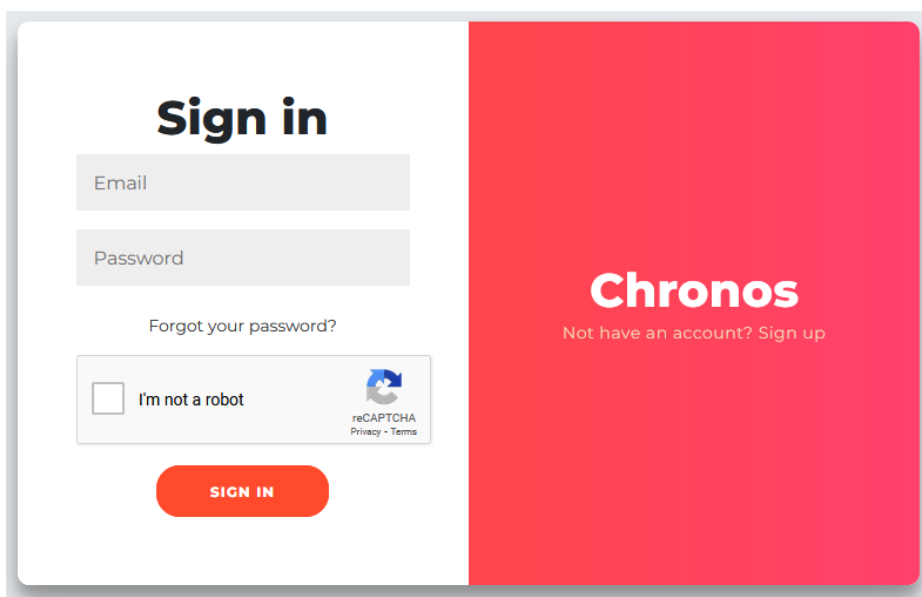
Κεφάλαιο 5 Παραδείγματα χρήσης

5.1 Είσοδος



The screenshot shows the 'Sign up' page for Chronos. On the left, there is a red vertical bar with the 'Chronos' logo in white and the text 'Already have an account? Sign in' below it. On the right, the page has a white background with the heading 'Sign up' in bold black text. Below the heading are three input fields: 'Email', 'Username', and 'Password'. At the bottom right, there is a red rounded button with the text 'SIGN UP' in white.

Εικόνα 1 Οθόνη Εγγραφής Χρήστη



The screenshot shows the 'Sign in' page for Chronos. On the left, there is a white background with the heading 'Sign in' in bold black text. Below the heading are two input fields: 'Email' and 'Password'. Below the 'Password' field is the text 'Forgot your password?'. Below that is a reCAPTCHA widget with a checkbox and the text 'I'm not a robot'. At the bottom left, there is a red rounded button with the text 'SIGN IN' in white. On the right, there is a red vertical bar with the 'Chronos' logo in white and the text 'Not have an account? Sign up' below it.

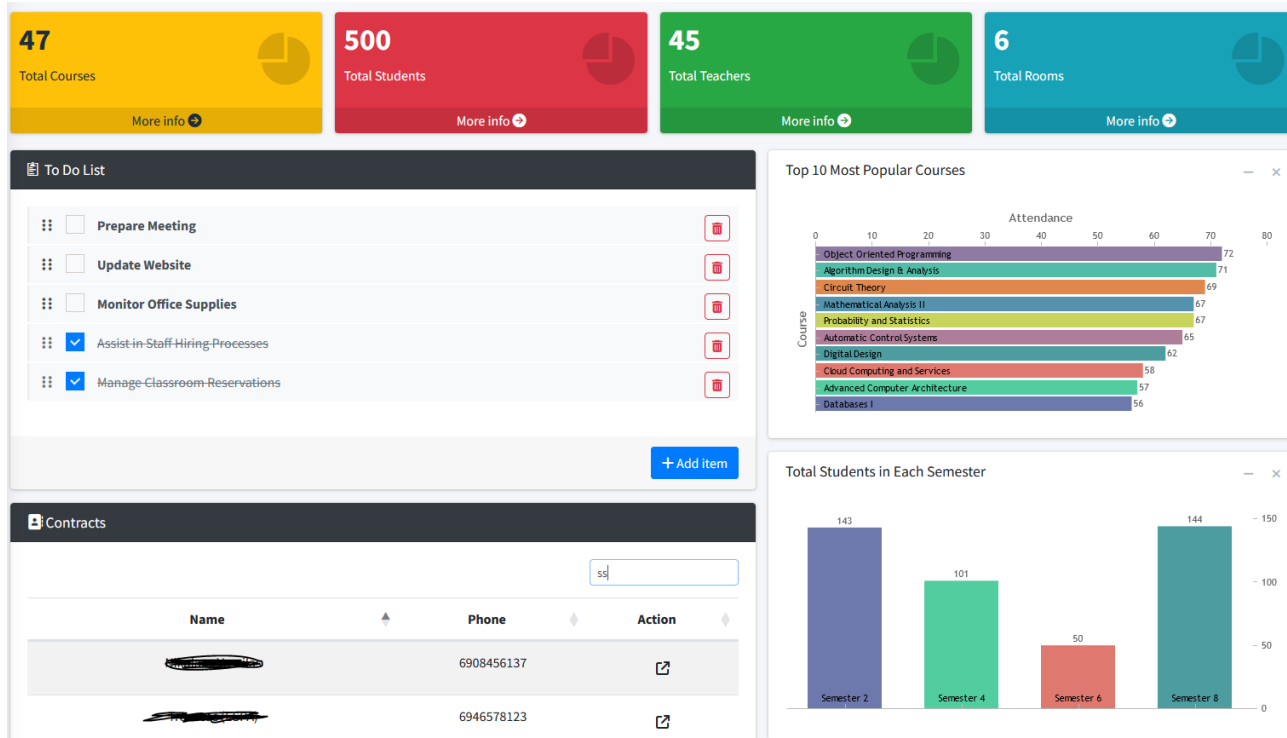
Εικόνα 2 Οθόνη Εισόδου Χρήστη

5.2 Αρχική οθόνη (Home)

Η αρχική οθόνη του συστήματος, παρέχει στον χρήστη μια ολοκληρωμένη επισκόπηση των βασικών δεδομένων και στατιστικών του τμήματος. Μόλις ο χρήστης εισέλθει στο σύστημα, μπορεί να ενημερωθεί άμεσα για τον συνολικό αριθμό δασκάλων, μαθητών, μαθημάτων και αιθουσών. Επιπλέον, η αρχική οθόνη περιλαμβάνει διαγράμματα που απεικονίζουν σημαντικά δεδομένα. Ένα από τα διαγράμματα δείχνει τον μέγιστο αριθμό μαθητών ανά εξάμηνο, δίνοντας μια οπτική αναπαράσταση της κατανομής των μαθητών κατά τη διάρκεια του ακαδημαϊκού έτους. Ένα άλλο σημαντικό γράφημα παρουσιάζει τα δημοφιλέστερα μαθήματα. Αυτό το γράφημα βοηθά στον εντοπισμό μαθημάτων με υψηλή ή χαμηλή συμμετοχή ώστε διαχειριστής να μπορεί να κάνει καταλληλότερη κατανομή των πόρων.

Πέρα από τα γραφήματα, η αρχική οθόνη διαθέτει έναν κατάλογο με τα τηλέφωνα των καθηγητών και των μαθητών, προσφέροντας εύκολη και γρήγορη πρόσβαση στα στοιχεία επικοινωνίας τους. Αυτό διευκολύνει την επικοινωνία μεταξύ του διοικητικού προσωπικού, των καθηγητών και των μαθητών, ενισχύοντας την αποτελεσματικότητα και τη συνεργασία.

Τέλος, υπάρχει ένα σημειωματάριο στην αρχική οθόνη, το οποίο επιτρέπει στους χρήστες να κρατούν σημειώσεις και να διαχειρίζονται τις καθημερινές τους υποχρεώσεις. Το σημειωματάριο αυτό είναι ένα ευέλικτο εργαλείο που μπορεί να χρησιμοποιηθεί για την καταγραφή σημαντικών υπενθυμίσεων, την οργάνωση των εργασιών και τη βελτίωση της προσωπικής παραγωγικότητας.



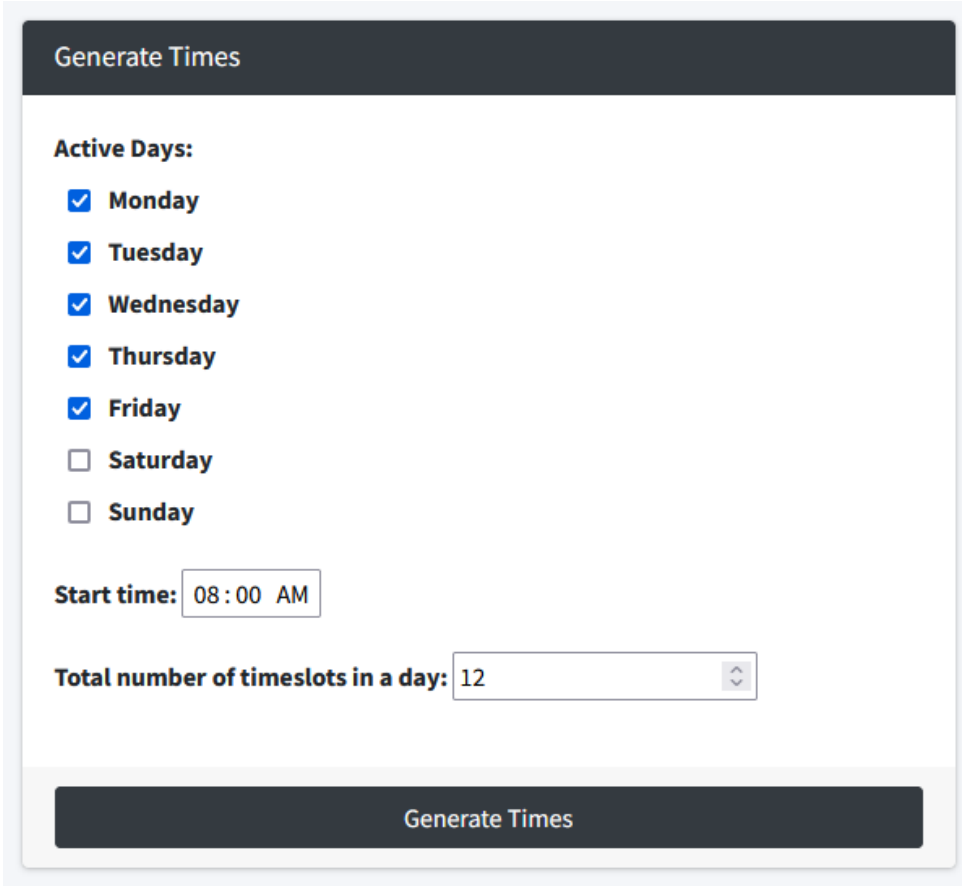
Εικόνα 3 Αρχική Οθόνη

5.3 Εισαγωγή δεδομένων

5.3.1 Χρόνοι (Times)

Generate times

Για να ορίσουμε τις πιθανές ώρες που θα γίνονται τα μαθήματα μέσα στην εβδομάδα θα πρέπει να μετάβουμε από το μενού στο Times → Generate times. Σε αυτή τη φόρμα θα πρέπει να δηλώσουμε τις ημέρες της εβδομάδας που μπορούν να διεξαχθούν μαθήματα στο πανεπιστήμιο μας, την ώρα έναρξης των μαθημάτων αλλά και πόσες διδακτικές ώρες περιέχει η κάθε ημέρα.



Generate Times

Active Days:

- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday
- Sunday

Start time:

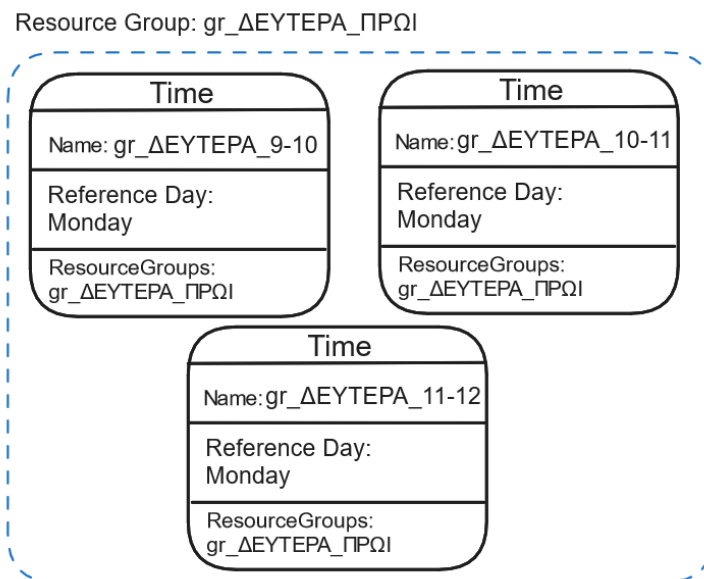
Total number of timeslots in a day:

Generate Times

Εικόνα 4 Φόρμα Generate Times

Ομάδα χρόνων (Time Groups)

Μέσω των ομάδων χρόνων μπορούμε να οργανώσουμε τους χρόνους σε group. Για παράδειγμα αν θα θέλαμε ορισμένα μαθήματα να μην διδάσκονται την Δευτέρα το πρωί τότε θα έπρεπε να φτιάξουμε την παρακάτω ομάδα χρόνων και έπειτα να ενεργοποιήσουμε τον κατάλληλο περιορισμό για την συγκεκριμένη ομάδα χρόνων.



Σχήμα 5.1 Παράδειγμα Ομάδας Χρόνων

Για να προσθέσουμε και να επεξεργαστούμε τις ομάδες χρόνων θα πρέπει μέσα από το μενού να πάμε στο: Time → Manage Time Group.

Για να προσθέσουμε μια καινούργια ομάδα χρόνων, πατάμε το κουμπί Add New στην σελίδα Manage Time Group και συμπληρώνουμε τα πεδία:

- Name: Όνομα αίθουσας
- Description: Περιγραφή
- IsDay: Δηλώνουμε αν η ομάδα χρόνων που φτιάχνουμε είναι ημερολογιακή ημέρα ή όχι.

Αφού ολοκληρώσουμε τη διαδικασία πατάμε Save.

The screenshot shows the 'Edit Time Group' form. It has a dark header with the text 'Edit Time Group'. Below the header, there are three main sections: 'Name:' with a text input field containing 'Monday', 'Description:' with an empty text input field, and 'Is Day:' with a checked checkbox. At the bottom of the form, there is a dark button labeled 'Edit'.

Εικόνα 5 Φόρμα επεξεργασίας Ομάδας Χρόνων

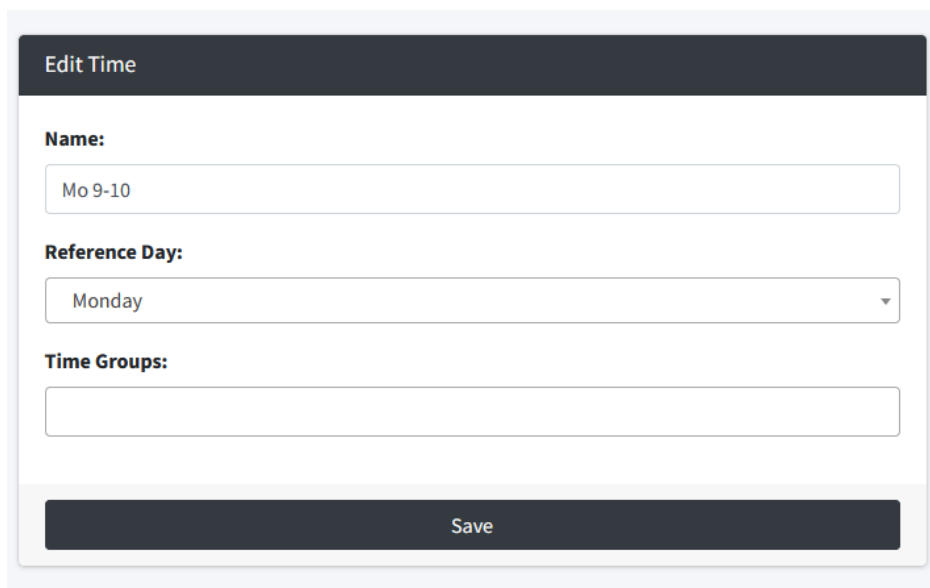
Χρόνος (Time)

Για να προσθέσουμε και να επεξεργαστούμε τους χρόνους θα πρέπει μέσα από το μενού να πάμε στο: Time → Manage Time.

Για να προσθέσουμε ένα καινούργιο χρόνο, πατάμε το κουμπί Add New στην σελίδα Manage Time και συμπληρώνουμε τα πεδία:

- Name: Όνομα αίθουσας
- Reference Day: Η ημέρα στην οποία ανήκει ο συγκεκριμένος χρόνος
- TimeGroups: Δηλώνουμε τις ομάδες χρόνων στις οποίες ανήκει ο συγκεκριμένος χρόνος.

Αφού ολοκληρώσουμε τη διαδικασία πατάμε Save.



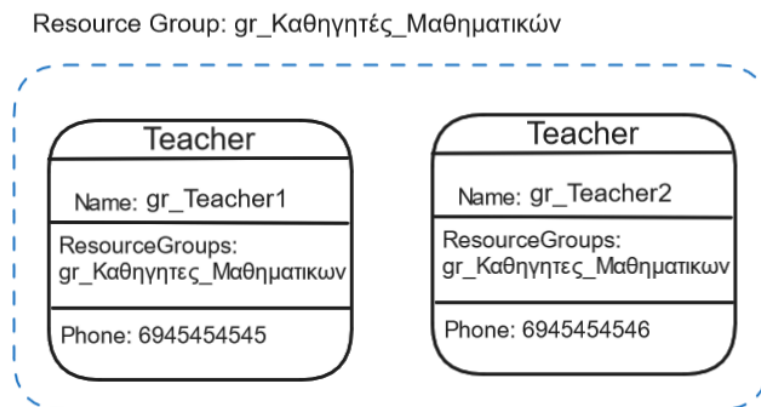
The image shows a web form titled "Edit Time". It contains three main input areas: a text box for "Name" with the value "Mo 9-10", a dropdown menu for "Reference Day" set to "Monday", and an empty text box for "Time Groups". A "Save" button is located at the bottom of the form.

Εικόνα 6 Φόρμα επεξεργασίας Χρόνου

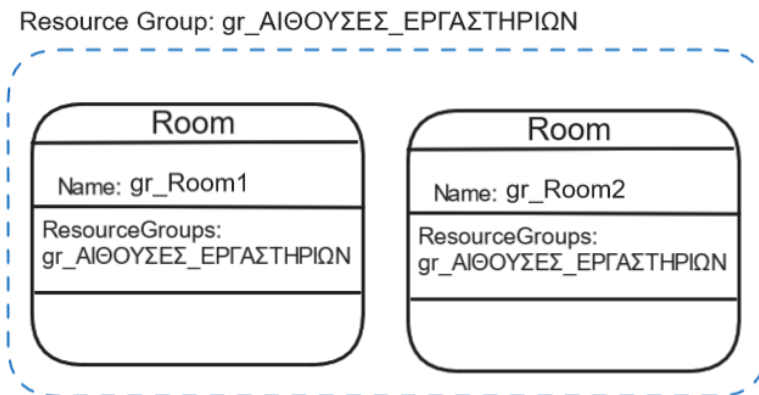
5.3.2 Πόροι (Resource)

Ομάδες πόρων (Resource Groups)

Μέσω των ομάδων πόρων μπορούμε να οργανώσουμε τους πόρους σε group. Για παράδειγμα αν θα θέλαμε ένα μάθημα να διδάσκεται από συγκεκριμένους καθηγητές, θα έπρεπε να φτιάξουμε την παρακάτω ομάδα και να την δηλώσουμε ως πόρο ικανοποίησης στο παραπάνω μάθημα.



Σχήμα 5.2 Παράδειγμα Ομάδας Πόρων 1



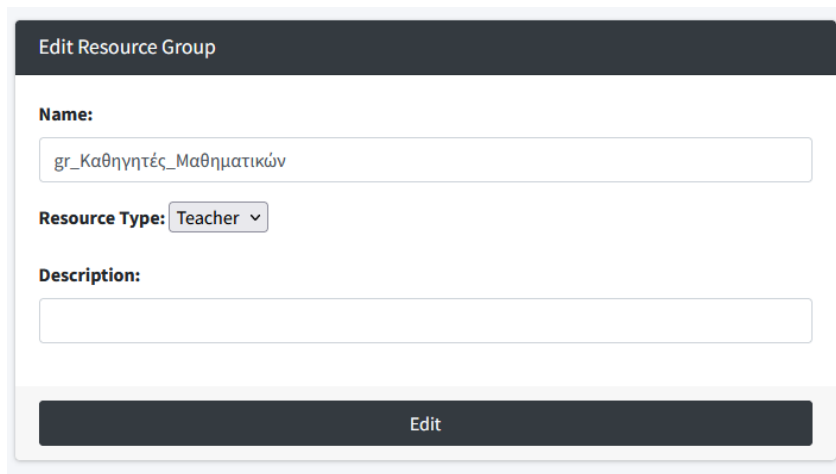
Σχήμα 5.3 Παράδειγμα Ομάδας Πόρων 2

Για να προσθέσουμε και να επεξεργαστούμε τις ομάδες πόρων θα πρέπει μέσα από το μενού να μετάβουμε στο: Resource → Manage Resource Groups.

Για να προσθέσουμε μια καινούργια ομάδες πόρων, πατάμε το κουμπί Add New στην σελίδα Manage Resource Groups και συμπληρώνουμε τα πεδία:

- Name: Όνομα ομάδας πόρων
- Resource Type: Τύπος ομάδας (Καθηγητές, Αίθουσες, Τμήμα, Μαθητές)
- Description: Περιγραφή

Αφού ολοκληρώσουμε τη διαδικασία πατάμε Save.



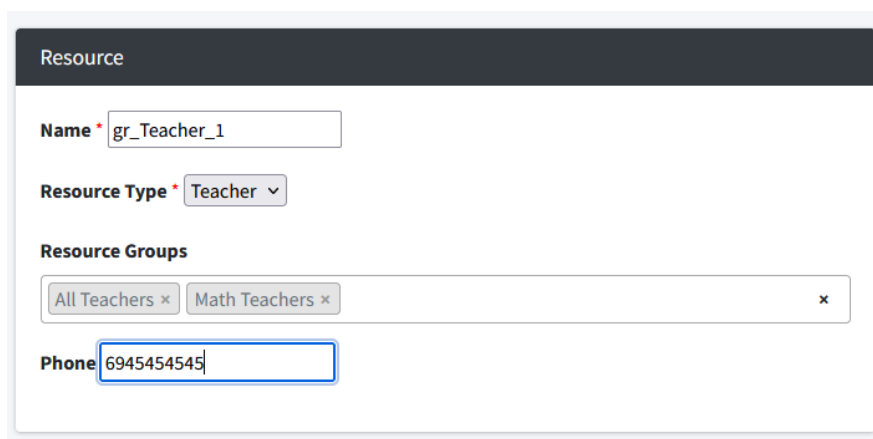
Εικόνα 7 Φόρμα επεξεργασίας Ομάδας Πόρων

Καθηγητές (Teachers)

Για να προσθέσουμε και να επεξεργαστούμε τους καθηγητές θα πρέπει μέσα από το μενού να πάμε στο: Resource → Manage Resources.

Για να προσθέσουμε έναν καινούργιο καθηγητή, πατάμε το κουμπί Add New στην σελίδα Manage Resources και συμπληρώνουμε τα πεδία:

- Name: Όνομα αίθουσας
- Resource type: Τύπος πόρου
- Resource Groups: Οι ομάδες πόρων που μπορεί να ανήκει η συγκεκριμένος καθηγητής.
- Phone: Αριθμός τηλεφώνου.



Εικόνα 8 Φόρμα Επεξεργασίας Πόρων

Αφού ολοκληρώσουμε τη διαδικασία πατάμε Save. Σε αυτό το σημείο θα μπορούσαμε να ορίσουμε τις προτιμήσεις για τον κάθε καθηγητή αλλά θα το κάνουμε στη συνέχεια βλέπε 1063032560.533.-558945776Περιορισμοί πόρων (Resource Constraints).

Αίθουσες (Rooms)

Για να προσθέσουμε και να επεξεργαστούμε τις αίθουσες θα πρέπει μέσα από το μενού να πάμε στο: Resource → Manage Resources.

Για να προσθέσουμε μια καινούργια αίθουσα, πατάμε το κουμπί Add New στην σελίδα Manage Resources και συμπληρώνουμε τα πεδία:

- Name: Όνομα αίθουσας
- Resource type: Τύπος πόρου
- Resource Groups: Οι ομάδες που ανήκει η συγκεκριμένη αίθουσα.

Αφού ολοκληρώσουμε τη διαδικασία πατάμε Save. Σε αυτό το σημείο θα μπορούσαμε να ορίσουμε τις προτιμήσεις για τον κάθε καθηγητή αλλά θα το κάνουμε στη συνέχεια βλέπε 1063032560.533.-558945776Περιορισμοί πόρων (Resource Constraints).

Εικόνα 9 Φόρμα επεξεργασίας Αίθουσας

Περιορισμοί πόρων (Resource Constraints)

Σε αυτό το σημείο ορίζουμε τις προτιμήσεις ενός Πόρου. Αυτές οι προτιμήσεις θα ισχύουν αυτόματα και για κάθε μάθημα που έχει ο συγκεκριμένος Πόρος.

Αποφυγή Μη Διαθέσιμων Χρόνων (Unavailable Times)

Στην παρακάτω φόρμα μπορεί ο κάθε πόρος να δηλώσει τις ομάδες χρόνων ή τους χρόνους που δεν είναι διαθέσιμος. Περισσότερες λεπτομέρειες για αυτό τον περιορισμό μπορείτε να βρείτε στο κεφάλαιο 4.4.11.

ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΤΟΥ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΧΟΛΙΚΩΝ ΜΟΝΑΔΩΝ ΜΕ ΧΡΗΣΗ ΤΕΧΝΙΚΩΝ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

Name *	Required	TimeGroups	Time	Delete
Unavailable on Monday & Tu13-14	<input checked="" type="checkbox"/>	Monday x	x	<input type="checkbox"/>

Εικόνα 10 Φόρμα εισαγωγής Μη Διαθέσιμων Χρόνων για ένα Πόρο

Περιορισμός χρόνων απασχόλησης (Limit Busy Times)

Στην παρακάτω φόρμα μπορούμε να δηλώσουμε το ανώτατο και το κατώτο όριο απασχόλησης ενός πόρου σε μια χρονική περίοδο. Στο παράδειγμα που ακολουθεί βλέπουμε ότι συγκεκριμένος πόρος σε καθημερινή βάση είναι διαθέσιμος από καμία έως το πολύ 8 διδακτικές ώρες.

Name *	Required	Minimum *	Maximum *	TimeGroups *	Delete
AtMost8HoursPerDay	<input checked="" type="checkbox"/>	0	8	Monday x Tuesday x Wednesday x Thursday x Friday x	<input type="checkbox"/>

Εικόνα 11 Φόρμα εισαγωγής Περιορισμός χρόνων απασχόλησης

Περιορισμός Χρόνου αδράνειας (Limit Idle Times)

Στην παρακάτω φόρμα μπορούμε να δηλώσουμε το ανώτατο και το κατώτο όριο αδράνειας ενός πόρου σε μια χρονική περίοδο. Στο παράδειγμα που ακολουθεί βλέπουμε ότι συγκεκριμένος πόρος σε καθημερινή βάση είναι μπορεί να είναι αδρανείς από καμία έως το πολύ 4 διδακτικές ώρες.

Name *	Required	Minimum *	Maximum *	TimeGroups *	Delete
AtMost4HoursPerDay	<input checked="" type="checkbox"/>	0	4	Monday x Tuesday x Wednesday x Thursday x Friday x	<input type="checkbox"/>

Εικόνα 12 Φόρμα εισαγωγής Περιορισμός Χρόνου αδράνειας

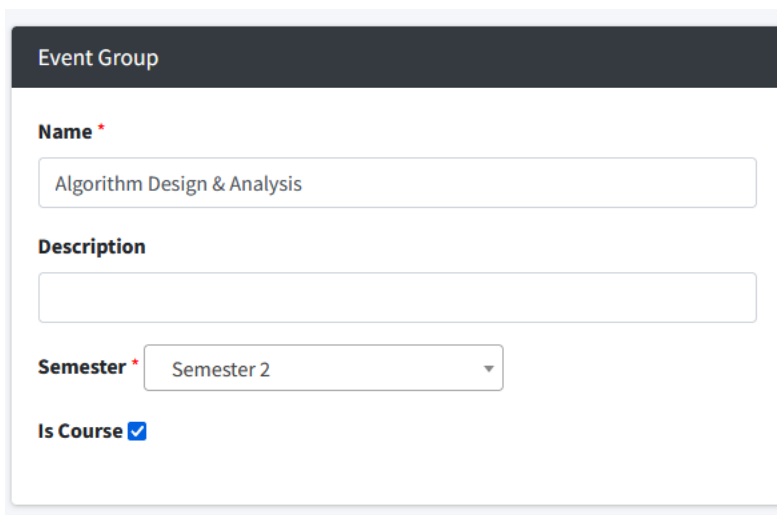
5.3.3 Μαθήματα (Events)

Ομάδα γεγονότων (Event Group)

Για να ορίσουμε τις πιθανές ώρες που θα γίνονται τα μαθήματα μέσα στην εβδομάδα θα πρέπει να μετάβουμε από το μενού στο Event → Manage Event Group. Πατάμε το κουμπί Add new και αυτό θα μας οδηγήσει στην φόρμα εισαγωγής νέου μαθήματος με τα εξής πεδία.

- Name: Όνομα μαθήματος
- Description: Περιγραφή μαθήματος
- Semester: Το εξάμηνο στο οποίο ανήκει αυτό το μάθημα.
- Is Course: Δυαδική μεταβλητή που δηλώνουμε αν η καταχώριση που κάνουμε αφορά ένα καινούργιο μάθημα ή μια γενική ομάδα γεγονότων.

Αφού ολοκληρώσουμε τη διαδικασία πατάμε Save.



Εικόνα 13 Φόρμα επεξεργασίας Ομάδας Γεγονότων

Γεγονός (Event)

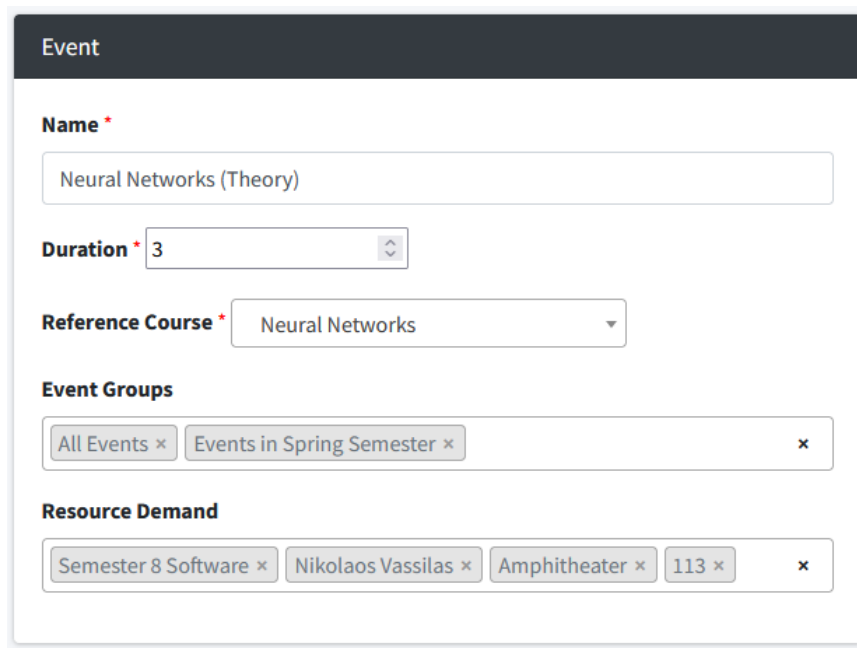
Για να προσθέσουμε ένα μάθημα δεν αρκεί να φτιάξουμε μόνο το αντίστοιχο event group. Είναι απαραίτητο να δημιουργήσουμε ένα ή περισσότερα γεγονότα για αυτό το μάθημα. Αν το μάθημα αποτελείται μόνο από μια διάλεξη συνεχόμενης διάρκειας τότε θα πρέπει μόνο ένα γεγονός. Από την άλλη, αν το μάθημα αποτελείται από δυο ή περισσότερες διαλέξεις συνεχόμενης διάρκειας τότε θα πρέπει να φτιάξουμε τόσα γεγονότα όσα και οι διαλέξεις του μαθήματος. Για παράδειγμα το μάθημα Βάσης Δεδομένων I χωρίζεται σε 2 διδακτικές ώρες εργαστήριου και 2 διδακτικές ώρες θεωρίας. Άρα, θα πρέπει στο σύστημα να φτιαχτούν 2 Event Για το μάθημα Βάσης Δεδομένων I.

Παρακάτω βλέπουμε πως μπορούμε να δημιουργήσουμε και να επεξεργαστούμε ένα Γεγονός. Μεταβαίνουμε από το μενού στο Event → Manage Events. Για να προσθέσουμε ένα καινούργιο γεγονός, πατάμε το κουμπί Add New στην σελίδα Manage Events και συμπληρώνουμε τα πεδία:

- Name: Όνομα γεγονότος
- Duration: Διάρκεια γεγονότος
- Reference Course: Το μάθημα στο οποίο ανήκει αυτό το γεγονός

- **Resource Demand:** Οι απαιτήσεις που μπορεί να έχει αυτό το γεγονός από πόρους ή ομάδες πόρων. (Εξάμηνο, Αίθουσα, Καθηγητής)

Αφού ολοκληρώσουμε τη διαδικασία πατάμε Save. Σε αυτό το σημείο θα μπορούσαμε να ορίσουμε τις προτιμήσεις για τον κάθε Γεγονός αλλά θα το κάνουμε στη συνέχεια βλέπε 1063032560.533.-558945776Περιορισμοί γεγονότων (Event Constraints).



The screenshot shows a form titled "Event" with the following fields and values:

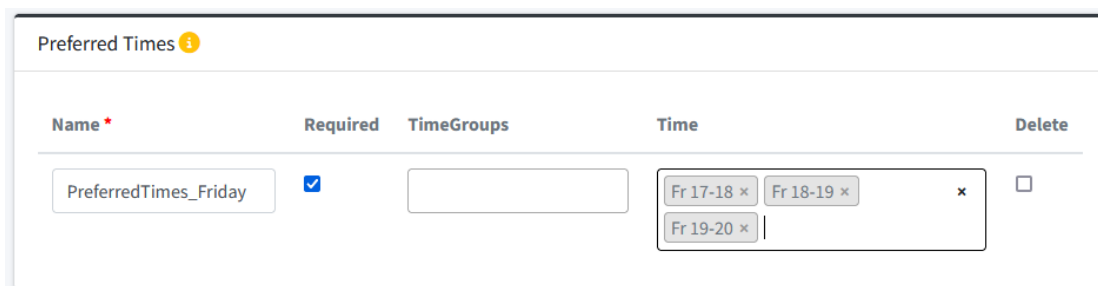
- Name ***: Neural Networks (Theory)
- Duration ***: 3
- Reference Course ***: Neural Networks
- Event Groups**: All Events ×, Events in Spring Semester ×
- Resource Demand**: Semester 8 Software ×, Nikolaos Vassilas ×, Amphitheater ×, 113 ×

Εικόνα 14 Φόρμα Επεξεργασίας Γεγονότος

Περιορισμοί γεγονότων (Event Constraints)

Επιθυμητοί χρόνοι (Preferred Times)

Στην παρακάτω φόρμα μπορεί το κάθε Γεγονός να δηλώσει τις ομάδες χρόνων ή τους χρόνους που επιθυμεί να προγραμματιστεί. Περισσότερες λεπτομέρειες για αυτό τον περιορισμό μπορείτε να βρείτε στο κεφάλαιο 4.4.5. Το παρακάτω παράδειγμα μας δείχνει ότι συγκεκριμένο γεγονός επιθυμεί να προγραμματιστεί την ημέρα Παρασκευή στο διάστημα 17:00 με 20:00.



The screenshot shows a form titled "Preferred Times" with a table containing the following data:

Name *	Required	TimeGroups	Time	Delete
PreferredTimes_Friday	<input checked="" type="checkbox"/>		Fr 17-18 × Fr 18-19 × Fr 19-20 ×	<input type="checkbox"/>

Εικόνα 15 Φόρμα εισαγωγής Επιθυμητών χρόνων

Επιθυμητοί Πόροι (Preferred Resources)

Στην παρακάτω φόρμα μπορεί το κάθε Γεγονός να δηλώσει τις ομάδες πόρων ή τους πόρους που επιθυμεί να το ικανοποιήσουν. Περισσότερες λεπτομέρειες για αυτό τον περιορισμό μπορείτε να βρείτε στο κεφάλαιο 4.4.4. Το παρακάτω παράδειγμα μας δείχνει ότι συγκεκριμένο γεγονός επιθυμεί να πραγματοποιηθεί στην αίθουσα 113.

Name *	Required	Resource	ResourceGroups	Delete
Preferred_Room-113	<input checked="" type="checkbox"/>	113 x x		<input type="checkbox"/>

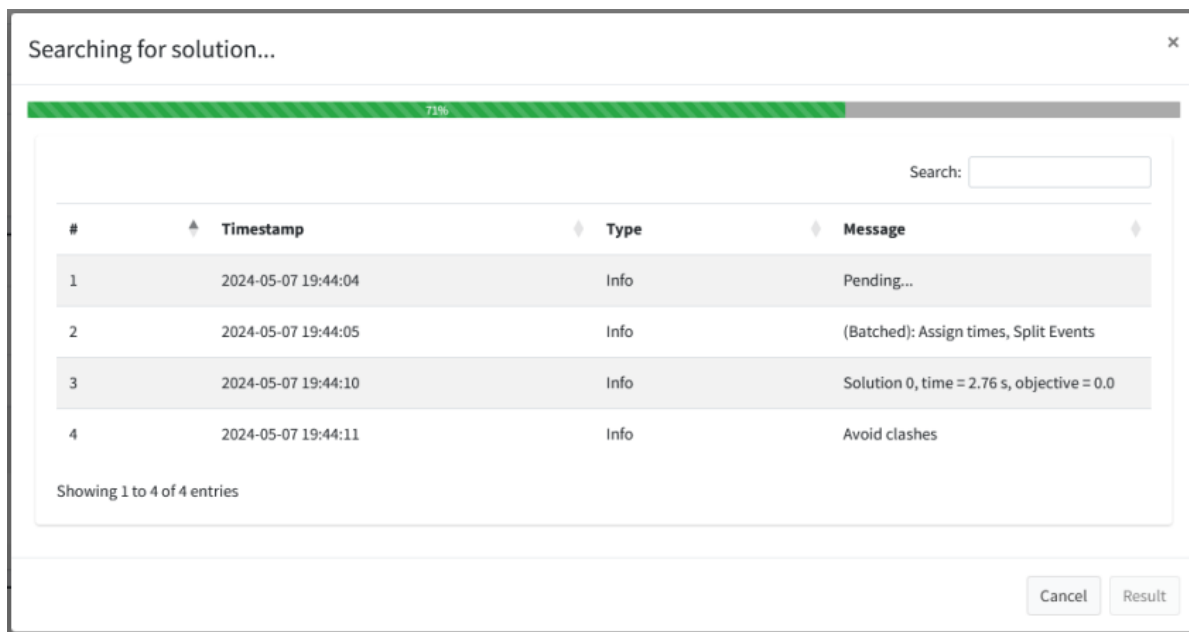
Εικόνα 16 Φόρμα εισαγωγής Επιθυμητών πόρων

5.4 Βελτιστοποίηση (Optimization)

Αφού ολοκληρώσουμε την εισαγωγή των δεδομένων το επόμενο βήμα είναι να πράξουμε το ωρολόγιο πρόγραμμα με την βοήθεια του Optimizer που αναπτύχθηκε στο πλαίσιο αυτής της διπλωματικής. Ανάλογα με τις ανάγκες του Ιδρύματος, ο Optimizer μπορεί να βοηθήσει με δημιουργία ενός καινούργιου προγράμματος με βάση τα δεδομένα που έχουν εισαχθεί στο σύστημα ή να ελέγξει αν το υπάρχον ωρολόγιο πρόγραμμα είναι εφικτό. Συνοπτικά η διαδικασία επίλυσης είναι η παρακάτω:

1. **Φόρτωση (Load)** όλων των δεδομένων που έχουν εισαχθεί σχετικά με τα μαθήματα, τις αίθουσες, και τις προτιμήσεις στη μνήμη του Optimizer. Κατά τη διάρκεια της διαδικασίας αυτής, διεξάγονται έλεγχοι για ασυνέπειες στην εισαγωγή δεδομένων ή για πληροφορίες που λείπουν σχετικά με τα μαθήματα και είναι απαραίτητα για το συνέχεια της διαδικασίας.
2. **Έλεγχος (Check)** εφικτότητας. Σε αυτό το σημείο ο Optimizer αγνοεί όλες τους περιορισμούς που έχουν εισαχθεί εκτός από τον περιορισμό αποφυγής συγκρούσεων. Με αυτό το τρόπο μπορούμε πολύ γρήγορα να ενημερώσουμε τον χρήστη αν το πρόγραμμα το οποίο πάει να παράξει είναι εφικτό ή αν θα πρέπει να διόρθωση κάποιο από τα μαθήματα που εισήγαγε.
3. **Βελτιστοποίηση (Optimization)**. Ο Optimizer ξεκινάει να επιλύει το πρόβλημα προσθέτοντας ένα περιορισμό τη φορά. Με αυτό το τρόπο ο χρήστης είναι σε θέση γνωρίζει ποιος περιορισμός προκάλεσε την μη εφικτότητα του προβλήματος.
4. **Εξαγωγή Αποτελέσματος**. Από όλες τις πιθανές λύσεις που βρήκε ο Optimizer, επιλέγουμε την βέλτιστη.

Για να εκτελέσουμε βελτιστοποίηση και να δημιουργήσουμε ένα καινούργιο πρόγραμμα θα πρέπει να μετάβουμε από το μενού στο Optimization → Run Optimization. Στην παρακάτω εικόνα, μπορούμε να δούμε την οθόνη που εμφανίζεται κατά την εκτέλεση της βελτιστοποίησης, όπου παρέχονται πληροφορίες σχετικά με τα διαφορετικά στάδια της διαδικασίας.



Εικόνα 17 Παράθυρο διαδικασίας βελτιστοποίησης

5.5 Πρόγραμμα (Timetable)

Για να δούμε και να επεξεργαστούμε και τέλος να οριστικοποιήσουμε το παραγόμενο πρόγραμμα από την διαδικασία της βελτιστοποίησης θα πρέπει να μετάβουμε από το μενού στο παράθυρο Timetable. Σε αυτό το παράθυρο θα εμφανιστούν τα μαθήματα για όλα τα εξάμηνα του τρέχων ακαδημαϊκού έτους. Ο διαχειριστής μπορεί μετακινήσει κάποιο ή κάποια μαθήματα και στη συνέχεια να πατήσει το κουμπί Validate ώστε να ξανά τρέξει βελτιστοποίηση και να ελέγξει αν το επεξεργασμένο πρόγραμμα είναι και εφικτό.

Για να δούμε, να επεξεργαστούμε και να οριστικοποιήσουμε το παραγόμενο πρόγραμμα από τη διαδικασία της βελτιστοποίησης, θα πρέπει να μετάβουμε από το μενού στη περιοχή Timetable. Σε αυτή τη περιοχή, θα εμφανιστούν όλα τα μαθήματα για τα εξάμηνα του τρέχοντος ακαδημαϊκού έτους, επιτρέποντας στον διαχειριστή να δει με λεπτομέρεια τον προγραμματισμό.

Ο διαχειριστής μπορεί να μετακινήσει μαθήματα, αν αυτό κριθεί απαραίτητο. Για παράδειγμα, μπορεί να αλλάξει την ώρα ή την αίθουσα διδασκαλίας ενός μαθήματος ή και τον διδάσκοντα, ώστε να καλύψει συγκεκριμένες ανάγκες ή προτιμήσεις που μπορεί να έχουν προκύψει.

Αν γίνουν αλλαγές στο πρόγραμμα, ο διαχειριστής έχει την επιλογή να ελέγξει τις αλλαγές του πατώντας το κουμπί Validate. Αυτό το βήμα είναι κρίσιμο καθώς προκαλεί την εκ νέου εκτέλεση της διαδικασίας βελτιστοποίησης. Ο στόχος είναι να ελεγχθεί αν το επεξεργασμένο πρόγραμμα παραμένει εφικτό. Κατά τη διαδικασία του ελέγχου, το σύστημα θα ελέγξει όλες τις νέες ρυθμίσεις για να διασφαλίσει ότι δεν υπάρχουν συγκρούσεις ή άλλες παραβιάσεις περιορισμών.

ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΤΟΥ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΧΟΛΙΚΩΝ ΜΟΝΑΔΩΝ ΜΕ ΧΡΗΣΗ ΤΕΧΝΙΚΩΝ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

Semester 4 class 1					
	Monday	Tuesday	Wednesday	Thursday	Friday
09:00	09:00 - 11:00 Technical Writing Theory				
11:00	11:00 - 14:00 Signals and Systems Theory [Class 1]	Teacher: Ioannis Voyiatzis Room: 114	11:00 - 14:00 Operating Systems II Theory [Class 1]		12:00 - 16:00 Software Development Methodologies Theory [Class 1]
13:00		14:00 - 17:00 Digital Circuit Design Theory [Class 1]	14:00 - 17:00 Computer Networks II Theory [Class 1]	13:00 - 16:00 Databases I Theory [Class 1]	
15:00					
17:00					
19:00					
21:00					

Εικόνα 18 Παράδειγμα προγράμματος για το 4ο Εξάμηνο

Κεφάλαιο 6 Συμπεράσματα

Στην παρούσα διπλωματική εργασία αναπτύχθηκε ένα ολοκληρωμένο σύστημα για την κατάρτιση ωρολογίου προγράμματος στη δευτεροβάθμια εκπαίδευση, με τη χρήση σύγχρονων τεχνικών και εργαλείων από τον τομέα της πληροφορικής και της επιχειρησιακής έρευνας. Ο σχεδιασμός του συστήματος βασίστηκε στη θεωρία των προβλημάτων ικανοποίησης περιορισμών, επιτρέποντας την εύκολη έκφραση και διαχείριση των ποικίλων περιορισμών που χαρακτηρίζουν το εκπαιδευτικό περιβάλλον.

Η αυτόματη κατάρτιση ωρολογίου προγράμματος αποτελεί ένα εξαιρετικά δύσκολο πρόβλημα, καθώς η εκδοχή του προβλήματος που καλούμαστε να λύσουμε ανήκει στην κατηγορία των NP-Δύσκολων [35] προβλημάτων. Για την επίλυσή του, χρησιμοποιήθηκε ένας επιτυτής τελευταίας τεχνολογίας ο οποίος αναπτύχθηκε από την Google και ονομάζεται CP-SAT Solver. Ο αλγόριθμος που δημιουργήθηκε στο πλαίσιο αυτής της διπλωματικής εργασίας εξασφαλίζει ποιοτικές και γρήγορες λύσεις, ακόμα και για πολύπλοκα σύνολα δεδομένων. Επιτρέπει την παραγωγή ωρολογίων προγραμμάτων μέσα σε δευτερόλεπτα, αντιμετωπίζοντας επιτυχώς τους περιορισμούς και τις απαιτήσεις του ελληνικού εκπαιδευτικού συστήματος.

Οι σημαντικότερες δυνατότητες του συστήματος αυτού, είναι η άμεση δημιουργία εβδομαδιαίων προγραμμάτων μαθημάτων, η διαχείριση των περιορισμών που αφορούν τη διαθεσιμότητα των καθηγητών, των αιθουσών και τις ανάγκες των μαθημάτων και η εύκολη προσαρμογή του προγράμματός σε νέες καταστάσεις. Η ευελιξία αυτή καθιστά το σύστημα εξαιρετικά χρήσιμο για τους διαχειριστές, προσφέροντας μια σαφή και αποδοτική λύση για την οργάνωση του ακαδημαϊκού χρόνου.

Παρόλο που το σύστημα παρουσιάζει σημαντικές βελτιώσεις, υπάρχουν περιθώρια για περαιτέρω ανάπτυξη. Μια μελλοντική βελτίωση θα μπορούσε να είναι η αυτοματοποιημένη παραγωγή του προγράμματος εξεταστικής περιόδου. Η προσθήκη αυτής της δυνατότητας θα εξασφαλίσει την αποτελεσματική διαχείριση και των εξεταστικών προγραμμάτων, λαμβάνοντας υπόψη τις απαιτήσεις και τους περιορισμούς που σχετίζονται με τις εξετάσεις. Αλτώςτε, αυτά τα δυο προβλήματα είναι πάρα πολύ όμοια. Επιπλέον, θα ήταν χρήσιμο να δημιουργηθεί ένας μηχανισμός που θα επιτρέπει την αυτόματη δημοσίευση του τελικού ωρολογίου προγράμματος στο διαδικτυακό χώρο της σχολικής μονάδας, με τη μορφή ανακοίνωσης. Αυτό θα εξασφάλιζε ότι οι μαθητές και οι καθηγητές θα έχουν άμεση πρόσβαση στο ενημερωμένο πρόγραμμα.

Η υλοποίηση της παρούσας λύσης αναδεικνύει τη δυναμική των σύγχρονων τεχνολογιών στην επίλυση σύνθετων προβλημάτων και την ενίσχυση της εκπαιδευτικής διαδικασίας, βελτιώνοντας την καθημερινότητα μαθητών και καθηγητών. Η περαιτέρω ανάπτυξη και βελτιστοποίηση του συστήματος μπορεί να συμβάλει ακόμα περισσότερο στη δημιουργία ενός ευέλικτου και αποδοτικού εκπαιδευτικού περιβάλλοντος.

Βιβλιογραφία

- [1] N. Pillay, «A survey of school timetabling research,» *Annals of Operations Research*, pp. 261-293, 2014.
- [2] J. Ullman, «NP-Complete Scheduling Problems,» *Journal of Computer and System*, pp. 384-393, 1975.
- [3] D. Welsh and M. Powell, "An upper bound to the chromatic number of a graph and its application to timetabling problem," *The Computer Journal*, p. 85–86, 1967.
- [4] G. Schmidt και T. Strohlein, «Timetable construction—An annotated bibliography,» *The Computer Journal* 23, p. 307–316, 1979.
- [5] N. Lawrie, «An integer linear programming model of a school timetabling problem,» *The Computer Journal* 12, p. 307–316, 1969.
- [6] E. Akkoyunlu, «A linear algorithm for computing the optimum university timetable,» *The Computer Journal* 16, p. 347–350, 1973.
- [7] J. Breslaw, «A linear programming solution to the faculty assignment problem,» *Socio-Economic Planning Science* 10, pp. 227-230, 1976.
- [8] R. McClure και C. Wells, «A mathematical programming model for faculty course assignment,» *Decision Science* 15, p. 409–420, 1984.
- [9] Hultberg και Cardoso, «The teacher assignment problem: A special case of the fixed charge transportation problem,» *European Journal of Operational Research* 101, pp. 463-473, 1997.
- [10] K. Gosselin και M. Truchon, «Allocation of classrooms by linear programming,» *Journal of Operational Research Society* 37, p. 561–569, 1986.
- [11] D. Werra, «An introduction to timetabling,» *European Journal of Operational Research* 19, p. 151–162, 1985.
- [12] A. Tripathy, « School timetabling – A case in large binary integer linear programming,» *Management Science* 30, pp. 1473-1489, 1984.
- [13] H. Eiselt και G. Laporte, «Combinatorial optimization problems with soft and hard requirements,» *Journal of Operational Research* 38, p. 785–795.
- [14] Aubin και Ferland, «A large scale timetabling problem,» *Computers and Operational Research* 18, p. 67–77, 1989.
- [15] D. Costa, «A tabu search algorithm for computing an operational timetable,» *European Journal of Operational Research* 76, p. 98–110, 1994.
- [16] A. Hertz, «Find a feasible course schedule using tabu search,» *Discrete Applied Mathematics* 35, p. 255–270, 1992.
- [17] Deris, Omatu, Ohta και Samat, «University timetabling by constraint-based reasoning: A case study,» *Journal*

of the Operational Research Society 48, p. 1178–1190, 1997.

- [18] Kang και White, «A logic approach to the resolution of constraints in timetabling,» *European Journal of Operational*, p. 306–317, 1992.
- [19] Burke, Elliman και Weare, «A genetic algorithm for university timetabling,» *University of Leeds*, 1994.
- [20] Burke, Elliman και Weare, «A genetic algorithm based university timetabling system,» *22nd East–West International Conference on Computer Technologies in Education*, p. 35–40, 1994.
- [21] Carrasco και Rato, «A multiobjective genetic algorithm for the class/teacher timetabling problem,» *Practice and Theory of Timetabling III*, p. 3–17, 2001.
- [22] A. Drexl και F. Salewski, «Distribution requirements and compactness constraints in school timetabling,» *European Journal of Operational Research* 102, p. 193–214, 1997.
- [23] B. Paechter, «Optimising a presentation timetable using evolutionary algorithms,» *AISB Workshop on Evolutionary Computation*, p. 264–276, 1994.
- [24] Paechter, Cumming, Luchian και Petriuc, «Two solutions to the general timetable problem using evolutionary methods,» *The Proceedings of the IEEE Conference on Evolutionary Computing*, 1994.
- [25] M. Sørensen και F. H. Dahms, «A two-stage decomposition of high school timetabling applied to cases in Denmark,» *Computers & Operations Research* 43, pp. 36-49, 2014.
- [26] M. Sorensen και T. J. R. Stidsen, «Comparing solutions approaches for a complete model of high school timetabling,» *DTU Management Engineering Report*, pp. 1-45, 2013.
- [27] S. Kristiansen, M. Sørensen και T. R. Stidsen, «Integer programming for the generalized high school timetabling problem,» *Journal of Scheduling* 18, pp. 377-392, 2015.
- [28] S. M. Al-Yakoob και D. SheraliH, «Mathematical models and algorithms for a high school timetabling problem,» *Computers & Operations Research* 61, pp. 56-68, 2015.
- [29] G. H. Fonseca, H. G. Santos και E. G. Carrano, «Integer programming techniques for educational timetabling,» *European Journal of Operational Research*, pp. 28-39, 2017.
- [30] I. X. Tassopoulos, C. A. Iliopoulou και G. N. Beligiannis, «Solving the greek school timetabling problem by a mixed integer programming model,» *The Journal of the Operational Research Society* 71, pp. 117 - 132, 2020.
- [31] P. Gerhard, S. Ahmadi και F. Geertsema, «Cyclic Transfers in School Timetabling,» *Springer Link* 34, pp. 133-154, 2012.
- [32] G. Post, L. Di Gaspero, J. Kingston, B. McCollum και A. Schaerf, «Practice and Theory of Automated Timetabling,» *The Third International Timetabling Competition*, pp. 29-31, 2012.
- [33] G. Post, J. Kingston, S. Ahmadi, S. Daskalaki, C. Gogos, J. Kyngas, C. Nurmi, N. Musliu, N. Pillay, H. Santos και A. Schaerf, «An XML archive for high school timetabling,» *Ann Oper Res*, 2014.
- [34] S. Kristiansen, M. Sørensen και T. Stidsen, «Integer programming for the generalized high school timetabling problem,» *Journal of Scheduling*, pp. 1-16, 2014.

[35] Eikelder και Willemen, «Some Complexity Aspects of Secondary School,» p. 18–27, 2001.

[36] J. H. Kingston, «High School Timetable File Format Specification: Constraints,» [Ηλεκτρονικό]. Available:
<http://jeffreykingston.id.au/cgi-bin/hseval.cgi?op=spec&part=constraints>. [Πρόσβαση March 2024].