



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

ΤΕΙ ΗΠΕΙΡΟΥ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ & ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΛΟΓΙΣΤΙΚΗΣ &
ΧΡΗΜΑΤΟΟΙΚΟΝΟΜΙΚΗΣ

ΤΕΙ ΗΠΕΙΡΟΥ

Τμήμα Λογιστικής και Χρηματοοικονομικής.

Μεταπτυχιακό Πρόγραμμα Σπουδών (Μ.Π.Σ.)
στη “Λογιστική - Χρηματοοικονομική και Διοικητική Επιστήμη”.
Κατεύθυνση: Διοίκηση Επιχειρήσεων και Οργανισμών.

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΤΟ ΠΡΟΒΛΗΜΑ ΤΟΥ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ
ΣΧΟΛΙΚΩΝ ΜΟΝΑΔΩΝ ΩΣ ΥΠΟΠΕΡΙΠΤΩΣΗ ΤΟΥ
ΠΡΟΒΛΗΜΑΤΟΣ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ
ΠΡΟΣΩΠΙΚΟΥ**

Νικόλαος Α. Βερύκιος

Επιβλέπων: Δρ. ΧΡΗΣΤΟΣ Γ. ΓΚΟΓΚΟΣ
Επίκουρος Καθηγητής

ΠΡΕΒΕΖΑ ΙΟΥΛΙΟΣ 2017



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

ΤΕΙ ΗΠΕΙΡΟΥ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ & ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΛΟΓΙΣΤΙΚΗΣ &
ΧΡΗΜΑΤΟΟΙΚΟΝΟΜΙΚΗΣ

ΤΕΙ ΗΠΕΙΡΟΥ

Τμήμα Λογιστικής και Χρηματοοικονομικής.

Μεταπτυχιακό Πρόγραμμα Σπουδών (Μ.Π.Σ.)
στη “Λογιστική - Χρηματοοικονομική και Διοικητική Επιστήμη”.
Κατεύθυνση: Διοίκηση Επιχειρήσεων και Οργανισμών.

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΤΟ ΠΡΟΒΛΗΜΑ ΤΟΥ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ
ΣΧΟΛΙΚΩΝ ΜΟΝΑΔΩΝ ΩΣ ΥΠΟΠΕΡΙΠΤΩΣΗ ΤΟΥ
ΠΡΟΒΛΗΜΑΤΟΣ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ
ΠΡΟΣΩΠΙΚΟΥ**

Νικόλαος Α. Βερούκιος

Επιβλέπων: Δρ. ΧΡΗΣΤΟΣ Γ. ΓΚΟΓΚΟΣ
Επίκουρος Καθηγητής

ΠΡΕΒΕΖΑ ΙΟΥΛΙΟΣ 2017

**THE GREEK HIGH SCHOOL TIMETABLING PROBLEM
AS A SPECIAL CASE OF THE EMPLOYEE SCHEDULING
PROBLEM.**

Αυτή η σελίδα αφήνεται σκόπιμα κενή.

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

ΠΡΕΒΕΖΑ 14 Ιουλίου 2017

Επιτροπή αξιολόγησης της πτυχιακής εργασίας:

Επιβλέπων:

Δρ. Γκόγκος Χρήστος
Επίκουρος Καθηγητής

Μέλος:

Δρ. Καραμάνης Κωνσταντίνος
Αναπληρωτής Καθηγητής

Μέλος:

Δρ. Διακομιχάλης Μιχαήλ.
Αναπληρωτής Καθηγητής.
Διευθυντής Προγράμματος Μεταπτυχιακών
Σπουδών.

© Βερύκιος Νικόλαος, 2017.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

ΒΕΡΥΚΙΟΣ Α. ΝΙΚΟΛΑΟΣ

14 ΙΟΥΛΙΟΥ 2017

Στην οικογένειά μου.

ΠΕΡΙΛΗΨΗ

Κάθε σχολική χρονιά αρχίζει με τις αναθέσεις των μαθημάτων διδασκαλίας σε καθηγητές και τη δημιουργία ενός ωρολογίου προγράμματος. Το πρόγραμμα αυτό καλείται να είναι τέτοιο, ώστε να βοηθάει το μαθητή μοιράζοντας τις διδακτικές ώρες κατάλληλα, ώστε η κούραση να κατανέμεται ομοιόμορφα, τόσο στις ώρες κάθε ημέρας, όσο και στις ημέρες της εβδομάδας. Ταυτόχρονα, αυτό πρέπει να είναι κατασκευασμένο με ισοσταθμισμένη την κόπωση και για τους καθηγητές, τόσο μέσα στην ημέρα όσο και μέσα στην εβδομάδα με τα ελάχιστα αλλά και απαραίτητα κενά. Η ομοιόμορφη κατανομή μαθημάτων στους καθηγητές, είναι δείγμα σωστής διοίκησης εκ μέρους της διεύθυνσης, και προσφέρει κλίμα ηρεμίας και συνεργασίας για την εύρυθμη λειτουργία της κάθε σχολικής μονάδας και εμπιστοσύνης προς τη διεύθυνση.

Ως πρόβλημα χρονοπρογραμματισμού στην εκπαίδευση, θεωρείται η ανάθεση συγκεκριμένων ωρών και πόρων σε διδασκαλίες, ενώ ταυτόχρονα θα πρέπει να ικανοποιείται και ένα σύνολο από περιορισμούς. Το πρόβλημα χαρακτηρίζεται ως ένα δύσκολο NP πρόβλημα στις περισσότερες μορφοποιήσεις του και έχει γίνει ένα ελκυστικό πεδίο έρευνας στα πεδία της πληροφορικής και της επιχειρησιακής έρευνας.

Η εργασία αυτή περιγράφει τη μορφοποίηση του προβλήματος που χρησιμοποιήθηκε για την έρευνα και το γενικό πλαίσιο των μεθόδων που έχουν χρησιμοποιηθεί στην προσπάθεια επίλυσης του προβλήματος, όπως περιοχές της επιχειρησιακής έρευνας, του μαθηματικού προγραμματικού καθώς και ευρετικές και μεταευρετικές τεχνικές.

Στη συνέχεια παρουσιάζει την ελληνική πραγματικότητα και ιδιαιτερότητα του προβλήματος στην Δευτεροβάθμια Εκπαίδευση, και συνεχίζει με την νέα κωδικοποίηση σε ΧΗSTT, η οποία γενικεύει το πρόβλημα αλλά και δίνει την δυνατότητα για νέα έρευνα.

Η ΧΗSTT δίνει ώθηση στην έρευνα περιγράφοντας το πρόβλημα ολοκληρωμένο, με περισσότερους περιορισμούς και διευρύνεται ο χώρος για βελτιωμένες λύσεις. Στη συνέχεια αναφέρονται οι τελευταίες εξελίξεις προς την κατεύθυνση αυτή. Δίνεται ακόμα ένα πραγματικό παράδειγμα ενός Γενικού Λυκείου που παρουσιάζει τη δύσκολη, για προγραμματισμό, κατάσταση που καλείται να επιλύσει το πρόβλημα. Τέλος αναφέρονται αντιπροσωπευτικά λογισμικά τα οποία χρησιμοποιούνται ή δύνανται να χρησιμοποιηθούν στις ελληνικές σχολικές μονάδες, προτείνοντας κατευθύνσεις έρευνας στην προσπάθεια αναζήτησης της βέλτιστης λύσης.

Λέξεις κλειδιά: γενικό λύκειο, δευτεροβάθμια εκπαίδευση, χρονοπρογραμματισμός, προγραμματισμός διδασκαλιών, ωρολόγιο πρόγραμμα

ABSTRACT

Every school year starts with the assignment of lessons and the creation of a school curriculum, which should meet the students' needs in such a way that it helps them by distributing teaching hours so that the factor of tiredness/fatigue is evenly apportioned in the hours of a day and accordingly in the days of a week. The school curriculum should be made in such a way that it meets the needs of the teaching staff too so that the factor of tiredness is evenly allocated both in the hours of a day and the days of a week with some time off (45 -minute gap the least) among tuition hours - which is equally important. The even distribution of lessons to teachers is indicative of the right and fair administration and coordination as far as the headmaster is concerned and is also conducive to an environment of calmness and cooperation for the satisfactory functioning of each school unit. Last but not least this even distribution of hours promotes confidence on the face of the headmaster.

The main problem of a school curriculum is considered to be this assignment/distribution of specific teaching hours and resources while at the same time a set of limitations should be placed and observed. This problem is described as a NP hard problem in its formatting and has therefore become attractive ground for research and study in the field of computing as well as in the field of operational research.

This paper is about the formatting of the problem that was used in the research as well as the wider context of the approaches that have been employed in order to solve the problem, such as aspects of the mathematical programming and meta-heuristic techniques.

Furthermore, the paper presents the Greek reality and peculiarity of the problem in Secondary Education and moves on to the new codification of XHSTT which generalizes the problem but also offers fertile ground of possibilities for new research.

The XHSTT gives a boost to the research by describing the problem extensively, including more limitations (to be observed) and the field is expanded with improved solutions. Next, the latest developments towards this direction are mentioned, an example of a General High school is given, which describes the difficult situation of a school curriculum and how this should be resolved.

Finally, representative software is mentioned, which are used or have the potential to be used in Greek schools, which in turn suggest new directions in research opting for the best solution.

Keywords: high school, secondary education, timetabling, scheduling, XHSTT

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ	1
1. Προβλήματα και θεωρία	4
1.1 Προβλήματα Βελτιστοποίησης.....	4
1.2 Προβλήματα χρονοπρογραμματισμού.....	4
1.3 Χώρος προβλημάτων και πληρότητα (NP Completeness).....	5
1.4 P, NP, NP Πλήρη και NP Δύσκολα προβλήματα.....	5
1.5 Προβλήματα απόφασης και προβλήματα βελτιστοποίησης.....	6
2. Χρονοπρογραμματισμός σχολικών μονάδων.....	8
2.1 Κωδικοποίηση του προβλήματος.....	8
2.2 Το μαθηματικό μοντέλο του προβλήματος.....	10
3. Τεχνικές αντιμετώπισης του προβλήματος	14
3.1 Γραμμικός προγραμματισμός.....	14
3.2 Ακέραιος προγραμματισμός.....	14
3.3 Συνδυαστικά προβλήματα βελτιστοποίησης.....	15
3.4 Branch and Bound.....	16
3.5 Επιλυτές προβλημάτων Ικανοποιησιμότητας (SAT solvers).....	18
3.6 Προγραμματισμός με περιορισμούς (Constraint Programming).....	19
4. Ευρετικές και μεταευρετικές τεχνικές.....	22
4.1 Γειτονιά της λύσης.....	22
4.2 Ευρετικές (Heuristics) και μεταευρετικές (metaheuristics) τεχνικές.....	22
4.3 Tabu Search.....	23
4.4 Προσομοιωμένη απόπτηση (simulated annealing).....	24
4.5 Γενετικοί Αλγόριθμοι.....	25
4.5.1 Γενετικοί τελεστές.....	26
4.5.2 Οι τιμές ποινής.....	26
4.6 Ο χώρος των μεταευρετικών τεχνικών.....	26
5. Το πρόβλημα δημιουργίας ωρολογίου προγράμματος στην ελληνική δημόσια παιδεία δευτεροβάθμιας εκπαίδευσης.....	28
5.1 Το Γενικό Λύκειο.....	28
5.1.1 Ο χωρισμός των τάξεων σε ομάδες τμημάτων ανά ζώνη.....	32
5.2 Το Επαγγελματικό Λύκειο (ΕΠΑ.Λ.).....	33
5.3 Το Γυμνάσιο.....	33
5.4 Άλλες κατηγορίες σχολείων.....	33
6. Η περίπτωση ενός γενικού λυκείου.....	35
6.1 Τα δεδομένα του προβλήματος.....	35
6.2 Χρήση του asc TimeTables και εκτέλεση του προγράμματος.....	42
6.3 Αποτελέσματα εκτέλεσης του λογισμικού.....	43
6.4 Άλλες δυνατότητες του asc TimeTables.....	45

7.	Νέα μορφοποίηση του προβλήματος.....	46
7.1	Το περιβάλλον επικοινωνίας με τα λογισμικά.....	46
7.2	Είσοδος δεδομένων.....	46
7.3	Η έξοδος (εμφάνιση στην οθόνη και στον εκτυπωτή).....	47
7.4	Μελέτη επίλυσης του προβλήματος οπτικοποίησης της λύσης.	48
8.	Λογισμικά χρονοπρογραμματισμού για σχολικές μονάδες.....	52
8.1	Γλώσσες επικοινωνίας XML.....	52
8.1.1	Κανόνες ονομασίας.....	53
8.2	XML για περιγραφή προβλημάτων ωρολογίων προγραμμάτων και των λύσεων του προβλήματος (XHSTT).	54
8.3	Η XML του χρονοπρογραμματισμού.....	55
8.4	Η οντότητα της χρονικής διάρκειας – (Times).	56
8.4.1	Η οντότητα των πόρων (Resources).	56
8.4.2	Η Οντότητα γεγονός (Event).	58
8.5	Οι Περιορισμοί.....	60
8.6	Η Μορφοποίηση κατά Fonseca 2017.....	62
8.7	Το ανοιχτό λογισμικό KHE High School Timetabling Engine.....	63
8.7.1	Περιγραφή διαδικασίας εύρεσης λύσης.....	63
8.8	Το εμπορικό λογισμικό asc TIMETABLE.....	65
8.9	Το ανοιχτό λογισμικό FET.....	65
9.	Συμπεράσματα, προτάσεις, επίλογος.....	68
	ΕΠΙΛΟΓΟΣ.....	70
	ΠΑΡΑΡΤΗΜΑ Α.....	71
	Βιβλιογραφία.....	75

Πίνακες

Πίνακας 5-1: Ώρες Διδασκαλίας ανα τάξη στο Ελληνικό Γενικό Λύκειο	29
Πίνακας 5-2: Γ Λυκείου - Ώρες διδασκαλίας και τμήματα ανά κατηγορία μαθημάτων.	32
Πίνακας 6-1: Ώρες διδασκαλίας στο λύκειο ανα τάξη και κατηγορία μαθημάτων	35
Πίνακας 6-2: Ώρες και τμήματα της Γ Λυκείου ανά κατηγορία μαθημάτων.	35
Πίνακας 6-3: Ώρες και τμήματα της Β Λυκείου ανά κατηγορία μαθημάτων	36
Πίνακας 6-4: Ώρες και τμήματα της Α Λυκείου ανά κατηγορία μαθημάτων	36
Πίνακας 6-5: Ώρες και Γεγονότα που πρέπει να τοποθετηθούν στο Ωρολόγιο Πρόγραμμα για την Γ λυκείου.	37
Πίνακας 6-6: Ώρες και Γεγονότα που πρέπει να τοποθετηθούν στο Ωρολόγιο Πρόγραμμα για την Β λυκείου.	37
Πίνακας 6-7: Ώρες και Γεγονότα που πρέπει να τοποθετηθούν στο Ωρολόγιο Πρόγραμμα για την Α λυκείου.	37
Πίνακας 6-8: Ώρες και Γεγονότα που θα έπρεπε να τοποθετηθούν στο Ωρολόγιο Πρόγραμμα για την Β λυκείου, εάν είχαν δημιουργηθεί όλες οι κατηγορίες.	37
Πίνακας 6-9: Ώρες και Γεγονότα που θα έπρεπε να τοποθετηθούν στο Ωρολόγιο Πρόγραμμα για την Α λυκείου, εάν είχαν δημιουργηθεί όλες οι κατηγορίες.	38
Πίνακας 7-1: Παράδειγμα αρχείου τύπου ics για συγκεκριμένο γεγονός που παρουσιάζεται σε ημερολόγιο.	50

Πίνακας Σχημάτων

Σχήμα 4-1: Γενική κατηγοριοποίηση μεταερευνητικών τεχνικών βασισμένων στη διαδικασία αναζήτησης. [MLG13]	27
Σχήμα 7-1: Διαδικασία επίλυσης προβλήματος.	49

Πίνακας Εικόνων

Εικόνα 1-1 : Σχέση μεταξύ των P , NP προβλημάτων.	6
Εικόνα 5-1: Χωρισμός τάξης-τμήματος σε κατηγορίες (ζώνες διδασκαλίας)	32
Εικόνα 5-2: Ζώνες διδασκαλίας για την Γ λυκείου (α -προσανατολισμού, β -Ξένης γλώσσας, γ -μαθημάτων επιλογής).	32
Εικόνα 6-1: Προκαθορισμός Ζωνών διδασκαλίας της Γ λυκείου.	38
Εικόνα 6-2: Προκαθορισμός ζώνης διδασκαλίας μαθημάτων προσανατολισμού της B λυκείου.	39
Εικόνα 6-3: Περιορισμοί για καθηγητές στο <i>asc TimeTables</i> (α).	40
Εικόνα 6-4: Περιορισμοί για καθηγητές στο <i>asc TimeTables</i> (β).	40
Εικόνα 6-5: Περιορισμοί για ομάδες καθηγητών στο <i>asc TimeTables</i> .	40
Εικόνα 6-6: Παράθυρο διαλόγου για ορισμό περιορισμού σε ομάδα (δύο) μαθημάτων.	41
Εικόνα 6-7: Πτυσσόμενη λίστα με περιορισμούς. Όπως φαίνεται από την κατακόρυφη γραμμή ολίσθησης, φαίνονται σχεδόν μόνο οι μισοί περιορισμοί.	42
Εικόνα 6-8: Παράθυρο διαλόγου μετά την εύρεση λύσης, στο οποίο παρουσιάζονται στατιστικά στοιχεία σχετικά με περιορισμούς για τους καθηγητές. <i>asc TimeTables</i> .	42
Εικόνα 6-9: Στατιστικά δεύτερης επιτυχημένης προσπάθειας δημιουργίας ωρολογίου προγράμματος.	43
Εικόνα 6-10: Περιορισμοί που χαλάρωσαν από την εύρεση αποδεκτής λύσης.	44
Εικόνα 7-1: Αποτέλεσμα εισαγωγής ενός γεγονότος (του πίνακα 7.1) στο <i>iCal</i> .	49
Εικόνα Παράρτημα Α - 1: Έξοδος (<i>output</i>) του <i>xslt</i> προγράμματος. Παρουσιάζεται μόνο ένα τμήμα. Το πρόβλημα είναι από ένα φινλανδικό σχολείο, από το <i>dataset</i> για τον διαγωνισμό <i>high school timetabling</i> .	75

Συντομογραφίες:

<i>NP</i>	<i>Non Polynomial</i>
<i>HC</i>	<i>Hard Constraint</i>
<i>SC</i>	<i>Soft Constraint</i>
<i>COMP</i>	<i>Combinatorial Problems</i>
<i>CP</i>	<i>Constraint Programming</i>
<i>LP</i>	<i>Linear Programming</i>
<i>IP</i>	<i>Integer Programming</i>
<i>BIP</i>	<i>Binary Integer Programming</i>
<i>MIP</i>	<i>Mixed Integer Programming</i>
<i>MILP</i>	<i>Mixed Integer Linear Programming</i>
<i>TSP</i>	<i>Traveling Salesman Problem</i>
<i>B&B</i>	<i>Branch and Bound</i>
<i>SAT</i>	<i>Satisfiability</i>
<i>SA</i>	<i>Simulated Annealing</i>
<i>GA</i>	<i>Genetic Algorithm</i>
<i>ΓΠ</i>	<i>Γενικής Παιδείας</i>
<i>ΞΓ</i>	<i>Ξένες Γλώσσες</i>
<i>ME</i>	<i>Μαθήματα Επιλογής</i>
<i>ΠΡ</i>	<i>Προσανατολισμός</i>
<i>ΕΠΑ.Λ.</i>	<i>Επαγγελματικό Λύκειο</i>
<i>Δ.Ε.</i>	<i>Δευτεροβάθμια Εκπαίδευση</i>
<i>XML</i>	<i>eXtensive Markup Language</i>
<i>XHSTT</i>	<i>XML for High School TimeTable</i>

ΕΙΣΑΓΩΓΗ

Εταιρείες, πανεπιστήμια, νοσοκομεία, σχολεία αναγκάζονται σε τακτά χρονικά διαστήματα να επιλύουν προβλήματα χρονοπρογραμματισμού, δημιουργώντας ωρολόγια προγράμματα λειτουργίας. Η δημιουργία προγραμμάτων λειτουργίας (timetabling) είναι συχνά χρονοβόρα διαδικασία και απαιτεί σημαντική προσπάθεια, ώστε να ικανοποιούνται διάφορες απαιτήσεις και επιθυμίες. Οι πίνακες πτήσεων, για παράδειγμα, είναι κρίσιμοι για την εταιρεία, καθώς επίσης και για τα αεροδρόμια και τους επιβάτες, εξαιτίας των αυστηρών σχεδιασμών και των διάφορων αβέβαιων συνθηκών, στις οποίες τα επιβατικά αεροπλάνα λειτουργούν. Οι πίνακες διαδρομών των τρένων, εντός και εκτός πόλεων, καλούνται να αντιμετωπίσουν δύσκολες καταστάσεις, όπως είναι οι ώρες αιχμής καθώς και οι ώρες εκτός αιχμής, οι ζώνες εργασιών και κατασκευών καθώς και η διαθεσιμότητα των οδηγών. Είναι σημαντικό για τους φοιτητές των πανεπιστημίων και τους μαθητές των σχολείων να λειτουργούν με ένα πολύ καλό πρόγραμμα μαθημάτων, ώστε να είναι δυνατόν να επιτύχουν τα καλύτερα αποτελέσματά τους. Η αποδοτικότητα των νοσοκόμων στα νοσοκομεία επηρεάζεται σημαντικά από την ποιότητα του χρονοδιαγράμματος για τις βάρδιες εργασίας τους.

Τις τελευταίες δύο δεκαετίες από τις κατηγορίες των παραπάνω προβλημάτων αυτά που αφορούν εκπαιδευτικά ιδρύματα έχουν μεγάλη αύξηση ενδιαφέροντος. Λόγω της αύξησης των εκπαιδευτικών ιδρυμάτων σε όλο το κόσμο, ο σχεδιασμός ενός εκπαιδευτικού ωρολογίου προγράμματος μαθημάτων έχει γίνει πιο δύσκολος από ποτέ. Λόγω των πολλών αντικειμένων που προσπαθούν να καλύψουν τα γυμνάσια και τα λύκεια, στη βασική εκπαίδευση που προσφέρουν, υπάρχει μια αυξημένη δυσκολία δημιουργίας ωρολογίων προγραμμάτων.

Στη διαδικασία δημιουργίας προγραμμάτων, σημαντικό ρόλο έχουν οι περιορισμοί. Αρχικά, για να είναι δυνατόν να δημιουργηθεί ένα οποιοδήποτε πρόγραμμα, είναι υποχρεωτικό να ικανοποιούνται οι αυστηροί περιορισμοί. Όταν πλέον έχουμε μία εφικτή λύση, τότε μόνο μπορούμε να αναζητήσουμε μια νέα λύση, με καλύτερη ποιότητα. Λύση καλύτερης ποιότητας έχουμε, όταν ικανοποιούνται περισσότεροι χαλαροί περιορισμοί. Τόσο οι χαλαροί όσο και οι αυστηροί περιορισμοί, έχουν άμεση σχέση με το ίδιο το πρόβλημα. Για να ξεπεραστεί το πρόβλημα της καλύτερης ποιοτικά λύσης, έχει δημιουργηθεί διαγωνισμός δημιουργίας ωρολογίων προγραμμάτων και φυσικά αξιολόγηση των λύσεων, με σκοπό να μειωθεί το χάσμα ανάμεσα στην έρευνα και την πρακτική

δημιουργίας ωρολογίων προγραμμάτων, θέτοντας μια κοινή βάση στην οποία οι αλγόριθμοι μπορούν να ελεγχθούν και να αξιολογηθούν. Η συγκριτική αξιολόγηση του Toronto [Patat10] καθώς και οι δύο πρώτοι διεθνείς διαγωνισμοί παρουσίασαν ένα από τα πιο μελετημένα συστήματα αξιολόγησης δημιουργίας ωρολογίων προγραμμάτων.

Σε αυτή τη διπλωματική εργασία, ενδιαφερόμαστε να μελετήσουμε την κατάσταση που επικρατεί στη θεωρία και την πράξη και να ερευνήσουμε την ελληνική πραγματικότητα, ώστε να εφαρμοσθούν καλύτερες διαδικασίες δημιουργίας ωρολογίων προγραμμάτων.

Στο πρώτο κεφάλαιο γίνεται μια αναφορά στα προβλήματα χρονοπρογραμματισμού και τη σχέση τους με τα προβλήματα βελτιστοποίησης και απόφασης. Επίσης αναφέρεται η θέση του προβλήματος στο σύνολο των NP και δύσκολων NP προβλημάτων.

Στο επόμενο κεφάλαιο μελετάται το πρόβλημα από μαθηματικής άποψης και κωδικοποιείται με μαθηματικές σχέσεις. Οι αυστηροί και οι χαλαροί περιορισμοί μετατρέπονται σε μαθηματικές σχέσεις και στο τρίτο κεφάλαιο μελετώνται από την βιβλιογραφία μέθοδοι επίλυσης που χρησιμοποιούνται από τον μαθηματικό προγραμματισμό και την συνδυαστική βελτιστοποίηση.

Στο τέταρτο κεφάλαιο περιγράφονται οι βασικότερες ευρετικές και μεταευρετικές τεχνικές, οι οποίες αναζητούν είτε εξιδεικευμένα είτε γενικευμένα σε μικρές και μεγάλες γειτονιές, ποιοτικές λύσεις του προβλήματος, έχοντας σαν στόχο τη βέλτιστη.

Στα δύο επόμενα κεφάλαια γίνεται η μελέτη της κατάστασης που επικρατεί στις σχολικές μονάδες δευτεροβάθμιας εκπαίδευσης. Συγκεκριμένα, στο πέμπτο κεφάλαιο αναφέρονται όλες οι παράμετροι, που αφορούν το σύγχρονο ελληνικό γενικό λύκειο και οι οποίες θα βοηθήσουν στην κωδικοποίηση του προβλήματος και θα οδηγήσουν σε συμπεράσματα για την καλύτερη επίλυσή του. Στο έκτο κεφάλαιο καταγράφεται η προσπάθεια επίλυσης του προβλήματος για το ίδιο σχολείο με το εμπορικό λογισμικό ascTimeTable. Στη συνέχεια αναλύονται τα αποτελέσματα της λύσης του προβλήματος. Η λύση που δίνεται, αν και είναι λειτουργική, δεν είναι η καλύτερη δυνατή και φαίνεται η ανάγκη χρήσης νέας κωδικοποίησης του προβλήματος για την εύρεση καλύτερων λύσεων.

Το κεφάλαιο οκτώ αναφέρεται σε πιο σύγχρονη κωδικοποίηση, χρησιμοποιώντας την νέα περιγραφική γλώσσα XHSTT στην πλήρη της μορφή. Η κωδικοποίηση αυτή χρησιμοποιείται διεθνώς, με αποτέλεσμα να μας δίνεται η δυνατότητα να χρησιμοποιήσουμε οποιοδήποτε λογισμικό (αν και το ελεύθερο λογισμικό μας ενδιαφέρει περισσότερο) είναι συμβατό με XHSTT, ώστε το πρόβλημα κάθε σχολείου (και πανεπιστημίου) να μπορεί να

επιλυθεί με τον τρόπο αυτό. Οι μελέτες αυτές δίνουν ενθαρρυντικά αποτελέσματα και οι λύσεις που εντοπίζονται αξιολογούνται ως καλύτερες.

Στο κεφάλαιο εννέα, περιγράφεται πιο αναλυτικά η περιγραφική γλώσσα XHSTT, η οποία ήδη παίζει σημαντικό ρόλο σε όλο το παγκόσμιο ερευνητικό και μη τομέα, όπως έχει ήδη διαφανεί.

Τα σχολεία παλαιότερα έπρεπε να επιλύουν το πρόβλημα του χρονοπρογραμματισμού μόνα τους, χωρίς καμιά βοήθεια. Σήμερα η κατάσταση είναι πολύ καλύτερη. Υπάρχουν αρκετά εμπορικά προγράμματα τα οποία καλύπτουν τις περισσότερες ανάγκες, αλλά και μερικά ελεύθερα λογισμικά τα οποία όμως έχουν ακόμα μικρές δυνατότητες. Όμως, ελπίζουμε η έρευνα να δώσει καλύτερα ερευνητικά εργαλεία, ώστε το πρόβλημα να λύνεται με καλύτερα αποτελέσματα. Η δυσκολία του προβλήματος, όμως, δεν σταματάει εκεί. Ακόμα και η εισαγωγή των δεδομένων απαιτεί προχωρημένους χρήστες με καλή γνώση του προβλήματος. Η έρευνα συνεχίζεται και για τη βελτίωση του γραφικού περιβάλλοντος εισαγωγής δεδομένων, από την πλευρά του ελεύθερου και του ερευνητικού λογισμικού.

Είναι γνωστό σε όλες τις σχολικές μονάδες ότι το ωρολόγιο πρόγραμμα μαθημάτων είναι σημαντικό για τους μαθητές και πρέπει να τους βοηθάει στο μέγιστο βαθμό στη δημιουργική απασχόλησή τους. Πρέπει, επίσης, να λαμβάνει υπόψη τη δυσκολία των μαθημάτων, ώστε να είναι κατανοητή ομοιόμορφα και μέσα στην ημέρα καθώς και μέσα στις ημέρες της εβδομάδας. Το ίδιο, όμως, πρέπει να συμβαίνει και για τους διδάσκοντες, ώστε να είναι δίκαιο και για αυτούς. Ο κάθε διδάσκων, ως συνεργάτης του μαθητή (όπως η σύγχρονη κοινωνία απαιτεί), θέλει να διακρίνει ότι η διοίκηση του δίνει ένα δίκαιο ωρολόγιο πρόγραμμα σε σχέση με τους συναδέλφους του και να αισθάνεται ότι τον σέβονται.

ΚΕΦΑΛΑΙΟ Ι

ΠΡΟΒΛΗΜΑΤΑ ΚΑΙ ΘΕΩΡΙΑ

1.1 Προβλήματα Βελτιστοποίησης.

Πρόβλημα βελτιστοποίησης είναι εκείνο στο οποίο ζητάμε να βρούμε την καλύτερη λύση, μέσα από το σύνολο δυνατών λύσεων του προβλήματος. Όταν οι μεταβλητές επιτρέπεται να έχουν δεκαδικές τιμές, έχουμε τα συνεχή προβλήματα, ενώ όταν οι μεταβλητές λαμβάνουν διακριτές τιμές, τότε έχουμε τα συνδυαστικά. Τα συνεχή προβλήματα διαχωρίζονται σε δύο μεγάλες κατηγορίες, τα γραμμικά και τα μη γραμμικά. Ο γραμμικός προγραμματισμός επιλύει τα γραμμικά προβλήματα, ενώ για τα μη γραμμικά χρησιμοποιούνται μέθοδοι καθολικής βελτιστοποίησης καθώς και ευρετικές μέθοδοι. [AS05]

1.2 Προβλήματα χρονοπρογραμματισμού.

Τα συγκεκριμένα προβλήματα ανήκουν στην κατηγορία συνδυαστικής βελτιστοποίησης και από μόνα τους είναι μια ιδιαίτερη κατηγορία. Τέτοιες κατηγορίες προβλημάτων είναι ο χρονοπρογραμματισμός σχολικών μονάδων, ο χρονοπρογραμματισμός εξετάσεων, ο χρονοπρογραμματισμός πανεπιστημιακών εκπαιδευτικών ιδρυμάτων, χρονοπρογραμματισμός αθλητικών γεγονότων, νοσοκόμων κλπ. Η επίλυσή τους είναι σημαντική, γιατί είτε αποσκοπεί να φέρει κέρδος είτε μπορεί να επιφέρει βελτίωση των συνθηκών εργασίας του προσωπικού στην ομάδα εργασίας.

Ας θεωρήσουμε ένα πρόβλημα χρονοπρογραμματισμού το οποίο θα μπορούσε να ανήκει σε μια οποιαδήποτε από τις παραπάνω κατηγορίες με 4 διαφορετικές παραμέτρους T , R , M και C . Οι παράμετροι θα είναι πεπερασμένα σύνολα και ας θεωρήσουμε ότι το T περιέχει χρονικές περιόδους και τα άλλα σύνολα περιέχουν πόρους, συναντήσεις (διδασκαλίες) και περιορισμούς αντίστοιχα. Τότε ως *πρόβλημα χρονοπρογραμματισμού* ορίζεται η δημιουργία συναντήσεων του M με ανάθεση των πόρων R , έτσι ώστε να επιτυγχάνεται η ικανοποίηση των περιορισμών C στον μέγιστο δυνατό βαθμό. [DBH10]

Ο χρονοπρογραμματισμός στα εκπαιδευτικά ιδρύματα μπορεί να χωριστεί στις κατηγορίες χρονοπρογραμματισμού *μαθημάτων δευτεροβάθμιας εκπαίδευσης* (γυμνασίων και λυκείων Δ.Ε.), *Πανεπιστημίων* (ΑΕΙ και ΤΕΙ) και *εξετάσεων Πανεπιστημίου*.

Οι παραπάνω κατηγορίες έχουν όμως κάποιες βασικές διαφορές μεταξύ τους και στη συνέχεια θα ασχοληθούμε μόνο με τον χρονοπρογραμματισμό σχολικών μονάδων Δ.Ε., στον οποίο ζητάμε να δημιουργήσουμε ένα εβδομαδιαίο πρόγραμμα μαθημάτων, δεδομένου ότι είναι γνωστές οι ώρες διδασκαλίας που απαιτούνται για κάθε μάθημα των καθηγητών σε κάθε τμήμα, ώστε να καλύπτεται το ωρολόγιο πρόγραμμα που καθορίζεται από το Υπουργείο Παιδείας.

1.3 Χώρος προβλημάτων και πληρότητα (NP Completeness).

Έχουν γραφτεί πολλά για ικανοποιητικής απόδοσης αλγόριθμους και επίλυση σύνθετων προβλημάτων, όπως του συντομότερου μονοπατιού, γράφου του Euler, spanning tree, κ.α. Όλα αυτά είναι επιτυχημένες λύσεις σχεδιασμού αλγόριθμων.

Είναι δυνατόν να επιλυθούν όλα τα υπολογιστικά προβλήματα από υπολογιστές; Υπάρχουν προβλήματα τα οποία δεν είναι δυνατόν να επιλυθούν από αλγόριθμο, ακόμα και σε απεριόριστο χρόνο. Τέτοιο παράδειγμα είναι το πρόβλημα του τερματισμού (Turing Halting Problem). Στο πρόβλημα τερματισμού ζητείται να ληφθεί απόφαση σχετικά με το εάν ένα πρόγραμμα θα ολοκληρώνει την εκτέλεσή του ή θα συνεχίσει να εκτελείται επ' άπειρο για κάποια πιθανή είσοδο, με την οποία θα μπορούσε να τροφοδοτηθεί. Ο Alan Turing απέδειξε ότι γενικός αλγόριθμος ο οποίος θα επιλύει το πρόβλημα τερματισμού για όλα τα πιθανά ζεύγη προγράμματος και εισόδου δεν υπάρχει. Κλειδί της απόδειξης είναι ότι η μηχανή Turing χρησιμοποιήθηκε ως μαθηματικός ορισμός του ζεύγους υπολογιστή - πρόγραμμα. [GL99]

Όμως υπάρχουν πολλά άλλα προβλήματα για τα οποία δεν έχει ακόμα βρεθεί γενικευμένη ικανοποιητική λύση: τα NP Complete προβλήματα, των οποίων η κατάσταση είναι άγνωστη. Για τα προβλήματα αυτής της κατηγορίας δεν έχει ακόμα βρεθεί αλγόριθμος ο οποίος να επιλύει το πρόβλημα σε πολυωνυμικό χρόνο, αλλά και δεν έχει αποδειχθεί ότι δεν υπάρχει μη πολυωνυμικός αλγόριθμος. Το ενδιαφέρον στα προβλήματα αυτά είναι ότι, εάν ένα από τα NP Complete προβλήματα επιλυθεί σε πολυωνυμικό χρόνο, τότε όλα μπορούν.

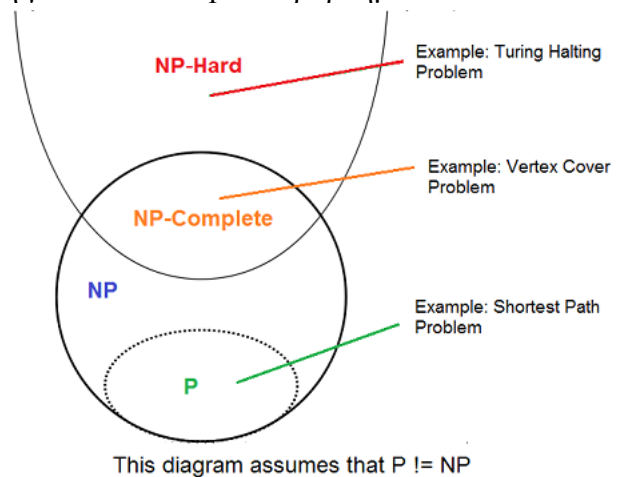
1.4 P, NP, NP Πλήρη και NP Δύσκολα προβλήματα.

Θεωρούμε ως P το σύνολο των προβλημάτων τα οποία μπορούν να επιλυθούν σε πολυωνυμικό χρόνο από μια αιτιοκρατική μηχανή Turing.

Θεωρούμε, επίσης, ως NP το σύνολο των προβλημάτων απόφασης, τα οποία μπορούν να επιλυθούν από μία **μη** αιτιοκρατική μηχανή Turing, σε πολυωνυμικό χρόνο. Το P είναι υποσύνολο του NP, λόγω του ότι κάθε πρόβλημα το οποίο μπορεί να επιλυθεί από μία αιτιοκρατική μηχανή Turing σε πολυωνυμικό χρόνο, είναι δυνατόν να επιλυθεί και από μία **μη** αιτιοκρατική μηχανή Turing σε πολυωνυμικό χρόνο. Άτυπα, NP είναι τα προβλήματα τα οποία μπορούν να επιλυθούν σε πολυωνυμικό χρόνο από ένα «τυχερό» αλγόριθμο, ο οποίος πάντα «μαντεύει» σωστά, δεδομένων των πιθανών επιλογών. Τα δυσκολότερα από τα NP προβλήματα λέγονται δύσκολα NP (NP hard) προβλήματα. [Val12]

Ένα πρόβλημα απόφασης L είναι NP Complete εάν:

- το L είναι NP (για δεδομένη λύση για τα NP complete προβλήματα είναι δυνατόν να επαληθευθεί σε σύντομο χρονικό διάστημα, αλλά δεν υπάρχει ικανοποιητικός αλγόριθμος εύρεσής της) και
- κάθε NP πρόβλημα μειώνεται σε (ανάγεται στο μικρότερο) L πρόβλημα, που επιλύεται δηλαδή σε πολυωνυμικό χρόνο.



Εικόνα 1-1 : Σχέση μεταξύ των P, NP προβλημάτων.

Ένα πρόβλημα είναι NP Hard, εάν ικανοποιεί την ιδιότητα 2, ενώ η πρώτη δεν είναι αναγκαία. Άρα, το NP Complete σύνολο είναι και αυτό ένα NP Hard σύνολο προβλημάτων [Wid06].

1.5 Προβλήματα απόφασης και προβλήματα βελτιστοποίησης.

Η NP πληρότητα εφαρμόζεται στη σφαίρα των προβλημάτων απόφασης. Δημιουργήθηκε με αυτό το τρόπο, λόγω του ότι είναι ευκολότερο να συγκριθεί η δυσκολία μεταξύ των προβλημάτων απόφασης παρά μεταξύ των προβλημάτων βελτιστοποίησης. Στην πραγματικότητα, όταν κάποιος μπορεί να επιλύσει ένα πρόβλημα απόφασης σε πολυωνυμικό χρόνο, τότε συχνά μπορεί να επιλύσει το αντίστοιχο πρόβλημα βελτιστοποίησης σε πολυωνυμικό χρόνο (χρησιμοποιώντας ένα πολυωνυμικό αριθμό από κλήσεις στο πρόβλημα απόφασης). Έτσι, η μελέτη της δυσκολίας των προβλημάτων απόφασης, είναι συχνά ισοδύναμη με τη μελέτη δυσκολίας των προβλημάτων βελτιστοποίησης [Wid06]. Για παράδειγμα, θεωρήστε το πρόβλημα κάλυψης κορυφών, στο

οποίο μας δίνεται ένας γράφος και καλούμαστε να βρούμε το ελάχιστο σύνολο κορυφών το οποίο καλύπτει όλες τις ακμές, κατά την έννοια ότι κάθε ακμή του γράφου έχει τουλάχιστον ένα από τα άκρα της να ανήκει στο σύνολο κορυφών που έχει επιλεγεί. Αυτό είναι ένα πρόβλημα βελτιστοποίησης.

ΚΕΦΑΛΑΙΟ ΙΙ

ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΧΟΛΙΚΩΝ ΜΟΝΑΔΩΝ.

2.1 Κωδικοποίηση του προβλήματος.

Η δευτεροβάθμια εκπαίδευση στα ελληνικά σχολεία διαρκεί συνολικά έξι χρόνια (3 υποχρεωτικά χρόνια στο γυμνάσιο και 3 προαιρετικά στο λύκειο). Κάθε τάξη χωρίζεται, αλφαβητικά, σε τμήματα, τα οποία διδάσκονται συγκεκριμένα μαθήματα, συνήθως από τους ίδιους καθηγητές του σχολείου. Οι διδασκαλίες γίνονται από Δευτέρα μέχρι και Παρασκευή σε 5, 6 ή 7 ώρες, συνεχόμενα χωρίς κενά. Το σύνολο των εβδομαδιαίων διδασκαλιών είναι προκαθορισμένο για κάθε τάξη, από το υπουργείο, όπως και τα αντικείμενα διδασκαλίας. Οι περισσότερες διδασκαλίες γίνονται ανά τμήμα και ανά αντικείμενο, από ένα καθηγητή της ειδικότητας του αντικειμένου διδασκαλίας. Ωστόσο, τα τμήματα μπορούν να διασπαστούν σε διαφορετικές ομάδες, με βάση συγκεκριμένα κριτήρια (για κάποιες ώρες την εβδομάδα), με σκοπό οι μαθητές των τμημάτων αυτών να παρακολουθήσουν διδασκαλίες αντικειμένων, ανάλογα με τις προτιμήσεις τους ή ακόμα και για διδασκαλία ίδιου αντικειμένου σε διαφορετικά επίπεδα (αρχάριοι, προχωρημένοι). Αυτό, συνήθως, αφορά τα γυμνάσια και τη διδασκαλία σε μαθήματα ξένων γλωσσών, η οποία πραγματοποιείται σε επίπεδα. Κάποια τμήματα χωρίζονται σε άλλα μικρότερα για διδασκαλίες οι οποίες λαμβάνουν χώρα σε μικρότερες αίθουσες, όπως τα εργαστήρια (χημείας, τεχνολογίας, πληροφορικής κ.α.). Σε αυτές τις περιπτώσεις, τις οποίες το τμήμα χωρίζεται και ενώ η μία ομάδα παρακολουθεί κάποια διδασκαλία σε εργαστήριο (π.χ. τεχνολογία), η άλλη ομάδα παρακολουθεί κάποια άλλη διδασκαλία (εργαστηριακή όπως η πληροφορική ή όχι όπως η γυμναστική). Σε αυτές τις περιπτώσεις, πρέπει να γίνει και άλλη μία διδασκαλία με τις ομάδες να εναλλάσσουν αντικείμενα διδασκαλίας (να παρακολουθήσουν τα αντικείμενα αντίστροφα).

Ο κάθε καθηγητής αναλαμβάνει να διδάξει ένα υποσύνολο μόνο από τις διδασκαλίες της ειδικότητας της κάθε τάξης, ενώ το πλήθος των διδακτικών ωρών του δεν είναι σταθερό και εξαρτάται από την παλαιότητά του, από κάποια θέση ευθύνης, που μπορεί να κατέχει καθώς και από άλλες παραμέτρους. Επίσης, είναι σχεδόν σίγουρο ότι σε κάθε σχολείο δεν είναι δυνατόν να καλυφθεί ακριβώς το ωράριο από τους υπάρχοντες καθηγητές του σχολείου και έτσι υπάρχουν κενά διδασκαλιών στο σχολείο αυτό. Γι' αυτό το λόγο υπάρχουν καθηγητές που έρχονται από άλλα σχολεία (στα οποία δεν συμπληρώνουν το υποχρεωτικό τους ωράριο) να διδάξουν κάποιες ώρες, για να συμπληρώσουν το δικό τους

υποχρεωτικό ωράριο διδασκαλίας, διδάσκοντας μαθήματα στα οποία υπάρχει έλλειψη σε κάποιο αντικείμενο στο σχολείο αυτό. Σε τέτοιες περιπτώσεις οι καθηγητές είναι διαθέσιμοι στο σχολείο αυτό συγκεκριμένες ημέρες, ή ακόμα και συγκεκριμένες ώρες κάποιων ημερών.

Στα προβλήματα που εξετάζουμε, οι αίθουσες χρησιμοποιούνται ως πόροι. Θεωρούμε αρχικά, ότι για κάθε διδασκαλία υπάρχει πάντα διαθέσιμη μία προκαθορισμένη αίθουσα (π.χ. για το σύνολο μαθητών Α1 υπάρχει η αίθουσα διδασκαλίας Α1) καθώς επίσης ότι υπάρχουν και κάποιες αίθουσες (πχ. εργαστήρια πληροφορικής, χημείας ή φυσικής) οι οποίες είναι δυνατόν να χρησιμοποιούνται για διδασκαλίες διαφορετικών τμημάτων. Αυτό δεν είναι απαραίτητη προϋπόθεση στη γενική του μορφή του προβλήματος.

Όπως είναι γενικά γνωστό, (Colomi et. al. 1992, Daskalaki- Birbas- Housos 2010, Post et. al. 2012, Τασσόπουλος 2016, κ.α.) στα προβλήματα βελτιστοποίησης συνδυασμών, υπάρχουν δύο ειδών περιορισμοί, οι ισχυροί (ανελαστικοί) και οι χαλαροί (ελαστικοί) περιορισμοί, οι οποίοι λαμβάνονται υπόψη για τη λύση του προβλήματος. Οι ισχυροί περιορισμοί, οι οποίοι δεν πρέπει να παραβιασθούν σε καμιά περίπτωση, και οι χαλαροί οι οποίοι μπορούν να παραβιασθούν και βαθμολογούνται, ώστε να αξιολογηθεί το αποτέλεσμα με μια βαθμολογία.

Οι ισχυροί περιορισμοί (Hard Constraints) αν και προφανείς για ένα άνθρωπο, δεν είναι προφανείς για ένα υπολογιστικό σύστημα και ορίζονται να είναι οι παρακάτω:

- HC1: Κάθε καθηγητής πρέπει να διδάσκει ένα συγκεκριμένο αριθμό ωρών σε κάθε τμήμα (βάσει των αναθέσεων του στην αρχή κάθε σχολικής χρονιάς).
- HC2: Κάθε καθηγητής πρέπει να διδάσκει μόνο τις ημέρες που θα βρίσκεται στο σχολείο.
- HC3: κάθε καθηγητής πρέπει να διδάσκει μόνο σε ένα τμήμα κάθε ώρα.
- HC4: κάθε καθηγητής πρέπει να διδάσκει τουλάχιστον μία (ή ένα ελάχιστο αριθμό διδασκαλιών) ώρα για κάθε ημέρα που βρίσκεται στο σχολείο.
- HC5: κάθε τμήμα δεν μπορεί να έχει περισσότερο από μία διδασκαλία κάθε χρονική περίοδο διδασκαλίας.
- HC6: Διδασκαλίες ίδιου αντικειμένου στο ίδιο τμήμα, δεν πρέπει να λαμβάνουν χώρα την ίδια ημέρα.
- HC7: Κενά μαθημάτων (όταν δεν καλύπτεται το επτάωρο) για κάθε τμήμα, είναι δυνατόν να υπάρχουν μόνο τις τελευταίες ώρες (συνεχόμενα) κάθε ημέρας.
- HC8: Ένας πόρος, δεν είναι δυνατόν να χρησιμοποιείται περισσότερες φορές από όσες είναι διαθέσιμος για κάθε χρονική περίοδο διδασκαλίας (πχ. δεν μπορεί να

έχουν διδασκαλία πληροφορικής στο μοναδικό εργαστήριο πληροφορικής περισσότερα από ένα τμήματα ανά ώρα διδασκαλίας).

Συνηθισμένοι χαλαροί περιορισμοί είναι οι παρακάτω:

- SC1: Κάθε καθηγητής καλό είναι να έχει τον ελάχιστο αριθμό κενών στο ωρολόγιο πρόγραμμά του. Ως κενό ορίζεται μία ώρα χωρίς μάθημα, ενώ υπάρχουν τουλάχιστον μία ώρα απασχόλησης πριν και μία μετά, την ίδια ημέρα.
- SC2: Κάθε καθηγητής πρέπει να έχει εβδομαδιαίο πρόγραμμα ισοσταθμισμένου φόρτου εργασίας.
- SC3: Οι προτιμήσεις των καθηγητών για διδασκαλίες τις πρώτες ή τις τελευταίες ώρες τις ημέρας, πρέπει να ικανοποιούνται στο μέγιστο βαθμό. Αυτός ο περιορισμός μπορεί να χρησιμοποιηθεί για να δώσει προτεραιότητα διδασκαλίας συγκεκριμένων αντικειμένων, να τοποθετηθούν τις πρώτες η τις τελευταίες ώρες τις ημέρας (πχ. τα μαθηματικά να διδάσκονται τις πρώτες ώρες της ημέρας).

Οι χαλαροί περιορισμοί μπορεί να είναι πολλοί και πολύπλοκοι. Παραπάνω αναφέρθηκαν μερικοί μόνο, οι οποίοι χρησιμοποιούνται για τη μορφοποίηση και κωδικοποίηση του προβλήματος με τον παραπάνω ορισμό. Το πρόβλημα παρακάτω κωδικοποιείται με επιπλέον περιορισμούς [CDM92, DBH10].

2.2 Το μαθηματικό μοντέλο του προβλήματος.

Τα βασικά σύνολα που χρησιμοποιούνται στον ορισμό του προβλήματος κατά τους Valouxis et al 2012 είναι:

- * T είναι το σύνολο των καθηγητών.
- * C είναι το σύνολο των τμημάτων όλων των τάξεων.
- * D είναι το σύνολο των ημερών του ωρολογίου προγράμματος, το οποίο είναι συνήθως οι 5 ημέρες της εβδομάδας.
- * H είναι το σύνολο των ωρών διδασκαλίας της ημέρας
- * E είναι το σύνολο των γεγονότων
- * R είναι το σύνολο των πόρων οι οποίοι μπορεί να χρησιμοποιούνται από περισσότερα τμήματα. Πόρος θεωρείται το εργαστήριο πληροφορικής, μια αίθουσα με προβολέα, ένα εργαστήριο φυσικής ή χημείας, κλπ.

Για τα παραπάνω σύνολα ορίζονται τα παρακάτω υποσύνολα:

- * Dt είναι το σύνολο των ημερών στις οποίες ο καθηγητής t είναι διαθέσιμος.
- * Et είναι το σύνολο των γεγονότων/συμβάντων/διδασκαλιών, στα οποία εμπλέκεται ο καθηγητής t
- * Wt είναι το σύνολο των ωρών που βρίσκεται καθημερινά (για όλη την εβδομάδα) ο καθηγητής t στο σχολείο και είναι διαθέσιμος για διδασκαλία στη σχολική μονάδα.

Επιτρεπτό ημερήσιο πρόγραμμα είναι ένας συνδυασμός από γεγονότα (ώρες διδασκαλίας) και κενά (για τους καθηγητές) και του οποίου το μήκος του ορίζεται ως $|H|$. Το σύνολο των *επιτρεπτών ημερησίων προγραμμάτων* ενός καθηγητή σχηματίζεται από την δημιουργία όλων των συνδυασμών των γεγονότων που επιλέγονται από το σύνολο E_t και των κενών ωρών. Γεγονότα τα οποία δεν επιτρέπεται να προγραμματιστούν την ίδια ημέρα (γεγονός/διδασκαλία του ίδιου μαθήματος) δεν μπορούν να δημιουργηθούν/συνυπάρξουν στον ίδιο συνδυασμό. Οι επιλεγόμενοι ημερήσιοι προγραμματισμοί όλων των καθηγητών πρέπει να καλύπτουν τη διδακτική εργασία για όλες τις ημέρες, έτσι ώστε το πρόβλημα να μπορεί να κατηγοριοποιηθεί ως ένα Σύνολο Λύσης/Κάλυψης του Προβλήματος.

Έστω x_{tdw} η δυαδική μεταβλητή η οποία θα έχει την τιμή 1 , εάν ο καθηγητής t την ημέρα d αναλαμβάνει το ημερήσιο πρόγραμμα w , ενώ σε αντίθετη περίπτωση, θα έχει την τιμή 0 . Έστω c_{tdw} το κόστος (βάρος) του ημερησίου προγράμματος w για τον καθηγητή t την ημέρα d . Το κόστος τότε συνίσταται από:

- * κόστος ποινής για κάθε κενό ανάμεσα από ώρες διδασκαλίας του ημερησίου προγράμματος.
- * κόστος ποινής για απόκλιση από το ελάχιστο και μέγιστο όριο από τις επιθυμητές ώρες διδασκαλίας.
- * κόστος για διδασκαλία σε ώρα ή οποία δεν ανήκει στις επιθυμητές ώρες διδασκαλίας του καθηγητή.

Το μαθηματικό πρόβλημα πλέον είναι :

$$\text{η ελαχιστοποίηση του αθροίσματος} \quad \sum_{t=1}^{|T|} \sum_{d=1}^{|D|} \sum_{w=1}^{|W_t|} c_{tdw} x_{tdw} \quad [1]$$

δεδομένων των περιορισμών:

$$\sum_{t=1}^{|T|} \sum_{d=1}^{|D|} a_{tdwe} x_{tdw} = 1, \quad t \in T, d \in D, w \in W \quad [2]$$

$$\sum_{t=1}^{|T|} \sum_{d=1}^{|D|} \sum_{w=1}^{|\bar{W}_t|} a_{tdwe} x_{tdw} = 1, \quad e \in E \quad [3]$$

$$\sum_{t=1}^{|T|} \sum_{w=1}^{|W_t|} a_{tdwhc} x_{tdw} = 1, \quad d \in D, h \in H, c \in C \quad [4]$$

$$\sum_{t=1}^{|T|} \sum_{w=1}^{|W_t|} a_{tdwhe} x_{tdw} = k_e y \quad d \in D, h \in H, e \in E \text{ with } T > 1 \quad [5]$$

$$\sum_{t=1}^{|T|} \sum_{w=1}^{|W_t|} a_{tdwhr} x_{tdw} = \text{Max} Q_r \quad d \in D, h \in H, r \in R \quad [6]$$

Η εξίσωση [1] είναι η αντικειμενική συνάρτηση του προβλήματος και παριστάνει το συνολικό κόστος της λύσης, το οποίο πρέπει να ελαχιστοποιηθεί. Είναι το άθροισμα του κόστους των ημερήσιων προγραμμάτων για όλους τους καθηγητές και για όλες τις ημέρες του εβδομαδιαίου προγράμματος.

Ο περιορισμός που περιγράφεται στην εξίσωση [2] δηλώνει ότι για κάθε καθηγητή και για κάθε ημέρα, μόνο ένα ημερήσιο πρόγραμμα μπορεί να επιλεγεί.

Η εξίσωση [3] μεταφράζεται στο ότι όλες οι αναθέσεις μαθημάτων πρέπει να προγραμματιστούν ακριβώς μία φορά η κάθε μία. Ο συντελεστής a_{tdwe} θα έχει την τιμή **1**, εάν ο ημερήσιος προγραμματισμός w του καθηγητή t την ημέρα d περιέχει το γεγονός e διαφορετικά θα έχει την τιμή **0**.

Η εξίσωση [4] εγγυάται ότι το ημερήσιο πρόγραμμα κάθε τμήματος δεν θα έχει κενά για κανένα τμήμα. Έτσι κάθε ώρα και για κάθε τμήμα θα υπάρχει μία ώρα διδασκαλίας. Όμοια, ο συντελεστής a_{tdwhc} θα έχει την τιμή **1**, εάν την ημέρα w για τον καθηγητή t την ημέρα d και την ώρα h υπάρχει διδασκαλία στο τμήμα c , διαφορετικά θα έχει τιμή **0**.

Η εξίσωση [5] περιγράφει και διαχειρίζεται τα γεγονότα στα οποία εμπλέκονται περισσότεροι από 1 καθηγητές, και τα κατασκευάζει με τρόπο, ώστε να λαμβάνουν χώρα την ίδια χρονική περίοδο (ημέρα και ώρα) για τους καθηγητές που εμπλέκονται. Έτσι ο

συντελεστής a_{tdwhe} θα έχει την τιμή **1**, εάν στο ημερήσιο πρόγραμμα του καθηγητή **t** την ημέρα **d** και ώρα **h** υπάρχει γεγονός **e** (κατάλληλη διδασκαλία), διαφορετικά θα έχει τιμή **0**.

Η εξίσωση [6] υπάρχει για να καλύψει το μοίρασμα των πόρων (ειδικές αίθουσες, κλπ.). Το $MaxQ_r$ είναι η απαιτούμενη-διαθέσιμη ποσότητα του πόρου **R** και η παράμετρος a_{tdwe} είναι η τιμή της απαιτούμενης ποσότητας του πόρου **r**, και θα είναι 1 όταν το ημερήσιο πρόγραμμα της ημέρας **w** του καθηγητή **t** την ημέρα **d** και ώρα **h** περιέχει ένα γεγονός/διδασκαλία το οποίο απαιτεί τον πόρο **R**, διαφορετικά θα είναι 0. ([DBH10])

Το μέγεθος του προβλήματος είναι πολύ μεγάλο για να βρεθεί η βέλτιστη λύση με τη δημιουργία όλων των πιθανών συνδυασμών τους και την αξιολόγησή τους.

ΚΕΦΑΛΑΙΟ ΙΙΙ

ΤΕΧΝΙΚΕΣ ΑΝΤΙΜΕΤΩΠΙΣΗΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

3.1 Γραμμικός προγραμματισμός.

Τα προβλήματα γραμμικού προγραμματισμού (Linear Programming) είναι μια κατηγορία προβλημάτων μαθηματικού προγραμματισμού, στα οποία προσπαθούμε να εντοπίσουμε ένα σύνολο τιμών για συνεχείς μεταβλητές (x_1, x_2, \dots, x_n), οι οποίες τιμές να μεγιστοποιούν ή ελαχιστοποιούν την αντικειμενική συνάρτηση (μια γραμμική συνάρτηση ως προς τις παραπάνω μεταβλητές) και να ικανοποιούν ταυτόχρονα ένα σύνολο από γραμμικούς περιορισμούς ισοτήτων (ή και ανισοτήτων). Τα προβλήματα αυτά αναφέρονται και ως προβλήματα βελτιστοποίησης με περιορισμούς. Ένα LP πρόβλημα εκφράζεται μαθηματικά ως εξής ([AS05]):

$$LP \text{ Maximize } z = c_j x_j$$
$$\text{δεδομένου ότι } \sum_j a_{ij} x_j \leq b_i \quad (i = 1, 2, \dots, m),$$
$$\text{και } x_j \geq 0 \quad (j = 1, 2, \dots, n).$$

3.2 Ακέραιος προγραμματισμός.

Ως πρόβλημα ακέραιου γραμμικού προγραμματισμού (Integer Programming) ορίζεται ένα πρόγραμμα γραμμικού προγραμματισμού στο οποίο τουλάχιστον μία από τις μεταβλητές περιορίζεται στο να λαμβάνει ακέραιες τιμές. Στις τελευταίες τρεις δεκαετίες χρησιμοποιήθηκε ο όρος Mixed Integer Programming (MIP), για προβλήματα LP με ακέραιους περιορισμούς σε μερικές, ή ακόμα και σε όλες τις μεταβλητές. Για να δοθεί έμφαση στα προβλήματα MIP, στα οποία όλες οι μεταβλητές πρέπει να έχουν ακέραιες τιμές, συνηθίζεται ο όρος *καθαρά προβλήματα IP* (pure IP).

Ο όρος «προγραμματισμός» (programming) στα παραπάνω προβλήματα δηλώνει σχεδιασμό ενεργειών, οι οποίες χρησιμοποιούν πόρους και απαιτήσεις, οι οποίες εκφράζονται με περιορισμούς. Οι πόροι μπορεί να εκφράζουν πρώτες ύλες, μηχανήματα, εξοπλισμό, εγκαταστάσεις, εργατικό δυναμικό, χρήματα, διαχείριση, δυνατότητες πληροφορικής και άλλα. Στα πραγματικά προβλήματα αυτοί οι πόροι είναι περιορισμένοι και πρέπει να μοιράζονται στη χρήση σε διάφορες ανταγωνιστικές δραστηριότητες. Ο πραγματικός σκοπός του LP είναι να δεσμεύει και να διανέμει τους πόρους, ώστε να ικανοποιούνται όλες οι απαιτήσεις στις ανταγωνιστικές δραστηριότητες, με τον βέλτιστο (καλύτερο δυνατό) τρόπο.

Ένα ακέραιο πρόβλημα (IP) του οποίου οι ακέραιες μεταβλητές περιορίζονται να έχουν τιμές 0 ή 1 χαρακτηρίζεται ως πρόβλημα *δυναδικού ακέραιου προγραμματισμού* (Binary Integer Programming) [CBD10].

3.3 Συνδυαστικά προβλήματα βελτιστοποίησης.

Ένα *συνδυαστικό πρόβλημα βελτιστοποίησης* (COMP) είναι ένα διακριτό πρόβλημα βελτιστοποίησης, στο οποίο επιδιώκουμε να βρούμε μια λύση από ένα πεπερασμένο σύνολο λύσεων, η οποία μεγιστοποιεί ή ελαχιστοποιεί μια αντικειμενική συνάρτηση. Τα προβλήματα αυτού του τύπου συνήθως προκύπτουν κατά την αναζήτηση επιλογής μιας λύσης, από ένα πεπερασμένο σύνολο εναλλακτικών λύσεων. Αυτές οι ποιοτικές εναλλακτικές λύσεις μπορούν να ποσοτικοποιηθούν με τη χρήση διακριτών μεταβλητών. Συνήθως, το σύνολο όλων των πιθανών λύσεων μπορεί να απαριθμηθεί και οι σχετικές αντικειμενικές τιμές τους μπορούν να αξιολογηθούν, ώστε να προσδιοριστεί η βέλτιστη λύση. Δυστυχώς, ο αριθμός των λύσεων με πλήρη απαρίθμηση είναι συνήθως υπερβολικά μεγάλος, ακόμη και για ένα πρόβλημα μέτριου μεγέθους [Gog12, DPPC00].

Τα COMP συνδέονται στενά με τα προβλήματα IP και, εάν όχι όλα, τα περισσότερα μπορούν να μετατραπούν σε προβλήματα δυναδικού προγραμματισμού (BIP).

Τα περισσότερα προβλήματα συνδυαστικής βελτιστοποίησης δεν μπορούν να ενταχθούν σε κάποια γενική μέθοδο εύρεσης λύσης και έτσι χρησιμοποιούνται ευρετικές τεχνικές για την επίλυσή τους.

Μερικά πολύ γνωστά προβλήματα με ακέραιους συντελεστές (συνδυαστικά) ακολουθούν παρακάτω:

➤ Το *πρόβλημα ανάθεσης* μπορεί να εφαρμοσθεί, για παράδειγμα, για να γίνουν αναθέσεις n εργασιών σε n εργαζομένους με τον πιο αποδοτικό τρόπο, έτσι ώστε κάθε εργασία να ανατεθεί σε ένα μόνο εργαζόμενο και αντιστρόφως. Το πρόβλημα αυτό έχει επιλυθεί πλέον «πολύ καλά», επειδή κάθε βέλτιστη λύση, με *χαλάρωση* του LP, είναι φυσικά ακέραια. Επιπλέον, υπάρχουν ειδικοί αλγόριθμοι ανάθεσης (όπως ο *ουγγρικός*), οι οποίοι επιλύουν το πρόβλημα πολύ γρηγορότερα από την τυποποιημένη μέθοδο simplex.

➤ Το *πρόβλημα του περιοδευόντος πωλητή* (TSP). Στο πρόβλημα αυτό υπάρχει ένα δίκτυο από n πόλεις συνδεδεμένες μεταξύ τους και ζητάμε να εντοπίσουμε την διαδρομή που πρέπει να ακολουθήσει ένας πωλητής, ο οποίος πρέπει να περάσει από

όλες τις πόλεις, μόνο μία φορά, διανύοντας την μικρότερη δυνατή απόσταση (κόστος). Είναι δύσκολο (NP hard) να βρεθεί μια βέλτιστη λύση στο πρόβλημα αυτό λόγω της συνδυαστικής του φύσης.

➤ Το *διακριτό πρόβλημα σακιδίου* (0-1 knapsack problem). Υπάρχουν n αντικείμενα και κάθε i αντικείμενο έχει αξία v_i και όγκο c_i , και ζητάμε να εντοπίσουμε το υποσύνολο των συγκεκριμένων αντικειμένων, ώστε να ελαχιστοποιηθεί το συνολικό κόστος για το υποσύνολο των αντικειμένων και να μεγιστοποιηθεί το κέρδος της αξίας τους.

➤ Το *πρόβλημα κοπής υλικών* (Cutting Stock Problem). Έχουμε κάποιο υλικό (με κάποια επιφάνεια όπως χαρτί, ύφασμα κ.α.) και ζητείται να βρεθεί με πιο τρόπο πρέπει να γίνει η κοπή του, ώστε να καλύψουμε συγκεκριμένα σχέδια, ελαχιστοποιώντας το υλικό το οποίο στο τέλος δεν θα είναι εκμεταλλεύσιμο.

➤ Το *πρόβλημα ικανοποιησιμότητας* (satisfiability problem). Στο πρόβλημα αυτό δίνεται μια λογική (Boolean) έκφραση και ζητάμε να καθορίσουμε την τιμή των μεταβλητών της, έτσι ώστε η τιμή της έκφρασης να είναι τελικά αληθής.

➤ Ο *χρωματισμός κόμβων γράφου* (Graph colouring problem). Δίνεται ένας γράφος και ζητείται να χρωματίσουμε τους κόμβους του, έτσι ώστε κάθε κόμβος να έχει διαφορετικό χρώμα από όλους τους γειτονικούς κόμβους.

➤ Το *πρόβλημα Bin Packing*. Στο πρόβλημα αυτό δίνεται συγκεκριμένος αριθμός από πανομοιότυπα δοχεία, από τα οποία το κάθε ένα μπορεί να αντέξει συγκεκριμένο βάρος και ένα αριθμό από αντικείμενα γνωστού βάρους. Το πρόβλημα έγκειται στην τοποθέτηση των αντικειμένων στον μικρότερο αριθμό από δοχεία με τον περιορισμό αντοχής των αντικειμένων. Το πρόβλημα ανήκει στην κατηγορία των *NP-hard* προβλημάτων.

Κάθε ένα από τα παραπάνω προβλήματα συνιστά ουσιαστικά από μόνο του μια ευρεία κατηγορία προβλημάτων και όλα αυτά είναι δυνατόν να περιγραφούν με τον ακέραιο προγραμματισμό. Το μοντέλο του ακέραιου προγραμματισμού δεν είναι μοναδικό για κάθε πρόβλημα και για να βρεθεί αποδοτική λύση, εντός αποδεκτού χρονικού διαστήματος, απαιτείται μεγάλη εμπειρία.

3.4 Branch and Bound

Η μέθοδος Branch and Bound (διακλάδωση και οριοθέτηση) είναι μία γενικού σκοπού προσέγγιση, ικανή να επιλύσει καθαρά IP προβλήματα, μικτά IP (MILP) και

δυναμικά προβλήματα. Κάθε MILP και δυναμικό πρόβλημα μπορεί να μετατραπεί σε IP πρόβλημα και έτσι ένας αλγόριθμος ο οποίος μπορεί να επιλύσει ένα IP πρόβλημα, εύκολα μπορεί να προσαρμοστεί, για να επιλύσει και τα υπόλοιπα αναφερόμενα προβλήματα με προφανείς μετατροπές. Επίσης, κάθε πρόβλημα μεγιστοποίησης εύκολα μετατρέπεται σε πρόβλημα ελαχιστοποίησης και αντίστροφα. Έτσι μπορούμε να θεωρήσουμε ότι σε όλες τις περιπτώσεις το γενικό πρόβλημα είναι ένα IP πρόβλημα μεγιστοποίησης.

Θεωρητικά, κάθε καθαρό IP πρόβλημα με πεπερασμένα όρια στις ακέραιες μεταβλητές, θα μπορούσε να επιλυθεί με πλήρη απαρίθμηση όλων των πιθανών συνδυασμών που ικανοποιούν όλους τους περιορισμούς και οδηγούν στη μέγιστη αντικειμενική τιμή. Δυστυχώς όμως, ένα πρόβλημα με n ακέραιες μεταβλητές και m πιθανές τιμές για κάθε μία από αυτές, έχει m^n πιθανούς συνδυασμούς για να ελέγξουμε, ώστε να βρούμε τη βέλτιστη λύση. Συνεπώς αυτή η απλή διαδικασία είναι πρακτικά αδύνατη.

Κατά τους Friedman και Liberty ([Fri17,Lib08]), μια καλύτερη εναλλακτική τεχνική είναι να εφαρμόσουμε ένα πιο έξυπνο σχήμα απαρίθμησης, το οποίο μπορεί να καλύψει όλες τις πιθανές λύσεις, με ρητή αξιολόγηση μιας μικρής ομάδας λύσεων, αγνοώντας μια μεγάλη ομάδα από αδύνατες λύσεις. Μια τέτοια στρατηγική καλείται «*διαίρει και βασίλευε*» (divide and conquer). Η κύρια σκέψη της στρατηγικής είναι ότι διαιρεί το πρόβλημα σε άλλα μικρότερα προβλήματα, τα οποία συστηματικά δημιουργούνται και επιλύονται. Οι λύσεις αυτών των επιμέρους προβλημάτων τοποθετούνται κατάλληλα μαζί και δημιουργούν τη λύση του αρχικού προβλήματος.

Η Branch and Bound (B&B) μπορεί να θεωρηθεί σαν μία προσέγγιση «*διαίρει και βασίλευε*» ενός IP προβλήματος, στην οποία μια διαδικασία διακλαδίζει (branches) και μια άλλη οριοθετεί (Bounds, δηλαδή θέτει μέγιστο ή και ελάχιστο όριο). Καθ' όλη τη διάρκεια εφαρμογής του αλγόριθμου, μια σειρά από LP υποπροβλήματα δημιουργούνται και επιλύονται. Με τον τρόπο αυτό υπολογίζεται προοδευτικά ένα επάνω και ένα κάτω όριο. Οι τιμές για το επάνω και το κάτω όριο μεταβάλλονται σε κάθε επανάληψη της διαδικασίας και πλησιάζουν στην αντικειμενική τιμή, η οποία είναι η λύση του αρχικού προβλήματος.

Με την παραπάνω διαδικασία δημιουργείται ένα B&B δέντρο, το οποίο ουσιαστικά παρακολουθεί τη διακλάδωση και την οριοθέτηση. Ο κόμβος της ρίζας, ο οποίος παριστάνει τη χαλάρωση του προβλήματος αποτελεί την αρχική λύση του προβλήματος S_{LP} . Εάν η βέλτιστη λύση που βρέθηκε, ικανοποιεί τους ακέραιους περιορισμούς, τότε είναι και λύση του LP προβλήματος, διαφορετικά η LP αντικειμενική τιμή του προβλήματος γίνεται το αρχικό επάνω όριο της IP βέλτιστης αντικειμενικής τιμής και η ρίζα διαχωρίζεται σε δύο διακλαδώσεις (υποπροβλήματα). Αυτές οι διακλαδώσεις είναι έγκυρες περικοπές από την

όψη απλών ανισοτικών περιορισμών με τις εξής ιδιότητες: α) περικόπτουν το μη ακέραιο βέλτιστο σημείο LP και άλλες κλασματικές περιοχές β) οι δύο ακόλουθοι κόμβοι είναι αμοιβαία αποκλειόμενοι και η ένωσή τους περιλαμβάνει την ίδια ακέραια εφικτή λύση, όπως οι πρόγονοί τους (δηλαδή οι μη ακέραιες τιμές εξαλείφονται). Η λύση από την LP χαλάρωση σε ένα κόμβο, παρέχει πληροφορίες σχετικά με το α) εάν απαιτείται περαιτέρω διακλάδωση (ή εάν η διακλάδωση χρειάζεται να «κλαδευτεί») και β) εάν έχει βρεθεί ένα χαμηλότερο όριο (για προβλήματα μεγιστοποίησης) της αντικειμενικής τιμής του IP προβλήματος.

Υπάρχουν τρεις περιπτώσεις που δηλώνουν ότι ένας κόμβος μπορεί να κλαδευτεί:

1. όταν το υποπρόβλημα δεν έχει εφικτή λύση (κλάδεμα μη εφικτής λύσης),
2. όταν το υποπρόβλημα έχει μια ακέραια βέλτιστη λύση (κλάδεμα βέλτιστης λύσης) και
3. όταν το επάνω όριο του βέλτιστου του υποπροβλήματος είναι μικρότερο ή ίσο από το χαμηλότερο όριο του αρχικού προβλήματος (κλάδεμα παραβίασης ορίου).

Εάν ένας κόμβος κλαδεύεται λόγω βέλτιστης λύσης, αυτή μπορεί να χρησιμοποιηθεί για να αυξήσει το κάτω όριο της αντικειμενικής τιμής του αρχικού προβλήματος.

Όταν βρίσκεται μια ακέραια λύση ενός υποπροβλήματος, τότε είναι πιθανόν να ανήκει στην τελική λύση του αρχικού προβλήματος. Κατά τη διαδικασία επίλυσης με την B&B μέθοδο, η μέχρι στιγμής καλύτερη ακέραια λύση συνεχώς ενημερώνεται και βελτιώνεται [BMP08].

3.5 Επιλυτές προβλημάτων Ικανοποιησιμότητας (SAT solvers).

Το πρόβλημα της *ικανοποιησιμότητας* (SAT) είναι ένα πρόβλημα *απόφασης*. Σε αυτή την κατηγορία ζητείται να ελέγξουμε, εάν μπορούν να υπολογισθούν οι τιμές των μεταβλητών σε μία προτεινόμενη λογική έκφραση, ώστε η έκφραση να μπορεί τελικά να λάβει αληθή τιμή.

Η επέκταση της SAT είναι η *τμηματικού βάρους μεγιστοποίησης ικανοποιησιμότητας* (Partial Weighted maxSAT), στην οποία οι προτάσεις διαχωρίζονται σε *αυστηρές* και *χαλαρές προτάσεις* (clauses). Σε κάθε χαλαρή πρόταση αντιστοιχίζεται μια τιμή βάρους. Στόχος είναι να υπολογίσουμε τις αναθέσεις τιμών στις μεταβλητές, οι οποίες

θα ικανοποιούν τους αυστηρούς περιορισμούς και ταυτόχρονα θα ελαχιστοποιούν τους χαλαρούς περιορισμούς. [DM17].

3.6 Προγραμματισμός με περιορισμούς (Constraint Programming).

Στον CP, τα προβλήματα συνδυαστικής βελτιστοποίησης θεωρούν ως βασικό αντικείμενο προγραμματισμού τους περιορισμούς. Οι περιορισμοί είναι σχέσεις των μεταβλητών που καθορίζουν το χώρο λύσεων, καθορίζοντας περιορισμούς στις τιμές που επιτρέπεται να έχουν ταυτόχρονα οι μεταβλητές. Στον χρονοπρογραμματισμό, για παράδειγμα, ο «γνωστός» περιορισμός “alldiff” [vH01] απαιτεί ένα σύνολο μεταβλητών να δέχονται διακριτά ζεύγη τιμών. Το μοντέλο το οποίο προσπαθούμε να επιλύσουμε με CP δεν δέχεται καμιά πληροφορία σχετικά με το πώς θα επιλυθεί το κωδικοποιημένο πρόβλημα. Έτσι είναι δυνατόν να χρησιμοποιηθεί ο όρος *δηλωτικό μοντέλο* (declarative modelling) για να αποδώσουμε την απουσία του ελέγχου της πληροφορίας.

Για την επίλυση προβλημάτων με CP, ο προγραμματιστής χρησιμοποιεί διαδικασίες αναζήτησης με *χρονολογική υπαναχώρηση/οπισθοδρόμηση* (chronological backtracking) [BV92]. Με την υπαναχώρηση, αναπτύσσεται ουσιαστικά η *depth-first* λογική. Οι κόμβοι αναζήτησης αναπαριστούν τμηματικές εκχωρήσεις τιμών σε μεταβλητές αποφάσεων. Εάν μια τμηματική εκχώρηση μεταβλητών δεν μπορεί να επεκταθεί λόγω παραβίασης των περιορισμών, τότε το backtracking επιστρέφει για αναζήτηση λύσης στον πιο πρόσφατο κόμβο, ο οποίος έχει ένα κλάδο δένδρου διαθέσιμο για αναζήτηση. Η αποδοτικότητα του backtracking, ουσιαστικά εξαρτάται από την ποιότητα των στρατηγικών διακλάδωσης.

Για να μειωθεί η ένταση αναζήτησης, η χρονολογική οπισθοδρόμηση συνδυάζεται με διάδοση των περιορισμών (constraint propagation). Για ένα τυχαίο πρόβλημα η διάδοση των περιορισμών αποφασίζει βασικούς περιορισμούς, όπως π.χ. «η μεταβλητή x_i δεν μπορεί να έχει την τιμή a », και χρησιμοποιείται για να κλαδέψει (να μην επιτρέψει την αναζήτηση) το χώρο αναζήτησης του υποδέντρου. Η διάδοση των περιορισμών και η αναζήτηση, ουσιαστικά, λειτουργούν συμπληρωματικά. Δηλαδή, η διάδοση περιορισμών ενεργοποιείται έπειτα από κάθε δέσμευση, στη συνέχεια γίνεται διαγραφή τομέων των μεταβλητών (τομέας είναι το σύνολο τιμών μιας μεταβλητής ο οποίος, στο δένδρο καταλαμβάνει ένα κλάδο) και έτσι ενεργοποιείται άμεσα η οπισθοδρόμηση (backtracking), και στρατηγικές δυναμικής διακλάδωσης αναζητούν λύσεις σε άλλους χώρους του δένδρου. Έτσι, οι περιορισμοί έχουν ενεργό ρόλο στην επίλυση προβλημάτων.

Ο προγραμματισμός με περιορισμούς οδηγεί στη δημιουργία Constraint Solvers (επιλυτών με περιορισμούς). Οι επιλυτές αυτοί υπάρχουν και λειτουργούν ανεξάρτητα

μεταξύ τους, επικοινωνούν μόνο με μεταβλητές του τομέα τους και όταν ένας επιλυτής εφαρμόζεται σε ένα περιορισμό δεν μεταφέρεται καμία πληροφορία, εκτός από τον ίδιο τον περιορισμό και την κατάσταση των μεταβλητών του. Αυτός ο τμηματικός σχεδιασμός διευκολύνει τη χρήση *τεχνικών με άμεση σύνδεση και εφαρμογή* (plug and play) και επιτρέπει τη γρήγορη προσαρμογή και ενσωμάτωση νέων απαιτήσεων και εξιδεικευμένων επιλυτών με περιορισμούς.

Όταν σχεδιάζεται ένας κατάλληλος επιλυτής προβλημάτων, ο προγραμματιστής, ο οποίος χρησιμοποιεί περιορισμούς, καλείται να εξερευνήσει ένα πολυδιάστατο χώρο αναζήτησης. Ο επιλυτής καθορίζεται από τη διαδικασία επιλογής, τις στρατηγικές διακλάδωσης, τους επιλυτές περιορισμών, καθώς και από το πώς θα εκφράσει τους περιορισμούς του προβλήματος. Συνήθως ο χώρος των συνδυασμών είναι πολύ μεγάλος και έτσι μόνο ένα μικρό τμήμα από αυτόν είναι δυνατόν να εξερευνηθεί. Προς αυτή την κατεύθυνση θα βοηθήσει και η θεωρητική καθώς και η λειτουργική σκοπιά. Θεωρητικά ζητήματα είναι η ορθότητα και η πληρότητα της μελέτης, ενώ στα λειτουργικά ζητήματα περιλαμβάνονται θέματα, όπως ο χρόνος εκτέλεσης, η κατανάλωση μνήμης και η αξιοπιστία σε θέματα εύρεσης λύσης. Η πληρότητα διαπραγματεύεται την αποδοτικότητα και την αξιοπιστία. Αρκετές φορές θυσιάζονται λύσεις επιβάλλοντας επιπλέον περιορισμούς, έπειτα από προσεκτικές απλοποιήσεις του χώρου αναζήτησης, για να περιοριστεί ακόμη περισσότερο (ο χώρος αναζήτησης). Οι διευκρινήσεις των λειτουργικών θεμάτων συνήθως απαιτούν εμπειρική έρευνα [Nan12].

Τα τελευταία χρόνια ο CP έχει αναδειχθεί σε κομβική τεχνολογία για τη συνδυαστική βελτιστοποίηση στις βιομηχανικές εφαρμογές αλλά και σε άλλες περιοχές (όπως στο timetabling, με λιγότερη όμως έμφαση) και σε αυτό διαδραμάτισαν κομβικό ρόλο τρεις παράγοντες: Η εισαγωγή ολικών περιορισμών, το πληθωρικό μοντέλο και η διαθεσιμότητα ισχυρών προγραμματιστικών εργαλείων (π.χ. CHIP¹, ECLiPS^{e2}, SICStus Prolog³, ILOG Solver⁴, Mozart⁵).

Οι **ολικοί περιορισμοί** [AB93] είναι αφηρημένοι και προσεκτικά σχεδιασμένοι, οι οποίοι συνοπτικά και με φυσικό τρόπο επιτρέπουν σε προβλήματα από διαφορετικά πεδία εφαρμογών να γίνονται μοντέλα και να επιλύονται κατάλληλα. Ο “alldiff” περιορισμός

¹ <http://www.cosytec.com>

² <http://www.eclipseclp.org>

³ <http://sicstus.sics.se>

⁴ <http://www-01.ibm.com/software/info/ilog/>

⁵ <http://mozart.github.io>, τελευταία επίσκεψη για όλες τις ιστοσελίδες: 1/7/2017.

[vH01], ο οποίος αναφέρθηκε παραπάνω, είναι ένας διακεκριμένος και αντιπροσωπευτικός ολικός περιορισμός και έχει διάφορες εφαρμογές στον χρονοπρογραμματισμό εργασιών. Οι ολικοί περιορισμοί είναι σημαντικοί, διότι επιτρέπουν συμπαγή και κομψά μοντέλα περιορισμών, ενώ ταυτόχρονα διαχέουν στο μοντέλο σημασιολογικές ιδιότητες. [MT00 και DPPH00]

Η *πληθωρική μοντελοποίηση* αφορά στην επέκταση των CP μοντέλων με πληθωρικούς περιορισμούς. Ένας περιορισμός ονομάζεται *πληθωρικός*, όταν κάθε λύση του προβλήματος ικανοποιεί τον περιορισμό. Μπορεί να θεωρηθεί ως ένα είδος διάχυσης περιορισμών (constraint propagation) που υλοποιείται πριν την αναζήτηση των αποτελεσμάτων με άλλους περιορισμούς. Tsang [Tsa93] and Marriott & Stuckey [MS98].

ΚΕΦΑΛΑΙΟ IV

ΕΥΡΕΤΙΚΕΣ ΚΑΙ ΜΕΤΑΕΥΡΕΤΙΚΕΣ ΤΕΧΝΙΚΕΣ.

4.1 Γειτονιά της λύσης

Μια γενική μέθοδος επίλυσης προβλημάτων βελτιστοποίησης συνιστά η διαδικασία εύρεσης της τελικής λύσης από μια αρχική μέσω ενδιάμεσων λύσεων. Η *γειτονιά μιας λύσης* είναι το σύνολο των αποδεκτών λύσεων, στις οποίες μπορούμε να μεταβούμε από κάποια λύση τροποποιώντας ένα χαρακτηριστικό της αρχικής λύσης. Εάν τροποποιήσουμε δύο χαρακτηριστικά μιας αρχικής λύσης, εναλλάσσοντάς τα μεταξύ τους, τότε έχουμε γειτονιά διπλής εναλλαγής. Επίσης μπορούν να χρησιμοποιηθούν *γειτονιές με πολύ μεγάλο μέγεθος* (Very Large Scale Neighbourhoods). [Gog09]

4.2 Ευρετικές (Heuristics) και μεταευρετικές (metaheuristics) τεχνικές.

Θα μπορούσε να πει κάποιος ότι *ευρετική* (heuristic) είναι η διαδικασία η οποία έχει σκοπό να προσφέρει μια καλή, αλλά όχι απαραίτητα βέλτιστη λύση, εύκολα και γρήγορα, σε ένα συγκεκριμένο δύσκολο πρόβλημα. Ο Shapiro (2001) διακρίνει τους ευρετικούς αλγόριθμους σε εκείνους που χρησιμοποιούν «*εμπειρικούς κανόνες*» σχετικούς με το πρόβλημα, για να φθάσουν σε εφικτές λύσεις των προβλημάτων MIP και σε *μεθόδους γενικού σκοπού* για έξυπνη αναζήτηση χώρου εφικτών λύσεων. Ισχυρίζεται ότι οι τελευταίοι «μπορούν να συνδυαστούν με εξειδικευμένες ερωτήσεις για τη βελτίωση της αποτελεσματικότητάς τους» [CBD10]. Ένας όρος που χρησιμοποιείται μερικές φορές όταν ένας ευρετικός αλγόριθμος ελέγχει έναν άλλο σε χαμηλότερο επίπεδο δραστηριότητας είναι η *μεταευρετική*.

Η ευρετική τοπική αναζήτηση αρχίζει με μια δεδομένη ικανοποιητική λύση, προσπαθώντας να βελτιώσει την τιμή της αντικειμενικής συνάρτησης με περιορισμένες αλλαγές (αντιμεταθέσεις) σε μια ή περισσότερες διαστάσεις, και προσέχοντας, ώστε η λύση να είναι αποδεκτή. Συνεπώς, η ευρετική εφαρμόζει ένα σύνολο από κανόνες με σκοπό να επιλέξει ένα στοιχείο από ένα σύνολο. Μια προφανής ευρετική μέθοδος επίλυσης για το πρόβλημα του περιπλανώμενου πωλητή είναι: μέχρι να επισκεφθείς όλες τις πόλεις, πήγαινε στην κοντινότερη πόλη που δεν έχεις επισκεφθεί και τελικά επίστρεψε στην αρχική. Με τον τρόπο αυτό δημιουργείται μια λύση, η οποία πιθανότατα δεν είναι η βέλτιστη, και ίσως μάλιστα να απέχει πολύ από τη ζητούμενη βέλτιστη λύση.

Ο όρος heuristic χρησιμοποιείται για αλγόριθμους, οι οποίοι βρίσκουν λύσεις ανάμεσα σε άλλες πιθανές, αλλά δεν εγγυώνται ότι θα βρεθεί η καλύτερη και με αυτό τον

τρόπο θεωρούνται ως αλγόριθμοι προσέγγισης και όχι ως βέλτιστης λύσης. Οι αλγόριθμοι αυτοί συνήθως βρίσκουν λύση κοντά στη βέλτιστη, και μάλιστα τη βρίσκουν γρήγορα και εύκολα. Οι αλγόριθμοι αυτοί έχουν την ιδιομορφία ότι εξαρτώνται από το ίδιο το πρόβλημα και προσπαθούν να εκμεταλλευθούν πλήρως τις ιδιαιτερότητες του προβλήματος. Τα προβλήματα όμως είναι *άπληστα* και έτσι οι αλγόριθμοι αυτοί εγκλωβίζονται σε τοπικά ελάχιστα, με αποτέλεσμα να αποτυγχάνουν (γενικά) να εντοπίζουν τη βέλτιστη λύση.

Metaheuristics είναι ουσιαστικά η στοχαστική βελτιστοποίηση και χρησιμοποιεί μέχρι κάποιο βαθμό τεχνικές τυχαιότητας, ώστε να βρει βέλτιστη λύση σε δύσκολα προβλήματα. Οι μεταευρετικοί αλγόριθμοι λειτουργούν ανεξάρτητα από το πρόβλημα, δεν εκμεταλλεύονται τις ιδιαιτερότητες του προβλήματος και γι' αυτό θεωρούνται "*μαύρα κουτιά*". Στη γενικότητά τους δεν είναι άπληστοι. Στην πραγματικότητα επιδέχονται και αλλοίωση της λύσης (π.χ. Simulated annealing), ώστε να μελετήσουν πιο διεξοδικά το χώρο και έτσι να επιτευχθεί μια καλύτερη λύση και, αν είναι δυνατόν, να έχει βρεθεί και η βέλτιστη λύση.

Παρακάτω θα αναφερθούν δύο εξεζητημένες μέθοδοι τοπικής αναζήτησης (tabu search και simulated annealing) και στη συνέχεια θα αναφερθεί η γενικού σκοπού ευρετική μέθοδος αναζήτησης η οποία χρησιμοποιεί γενετικούς αλγόριθμους.

4.3 TABU SEARCH

Η αναζήτηση Tabu είναι ένας από τους πιο διαδεδομένους μεταευρετικούς αλγόριθμους για εύρεση μιας καλής λύσης σε προβλήματα συνδυαστικής βελτιστοποίησης. Για την εύρεση βέλτιστης λύσης αρχίζει κανείς από μια απλή και τυχαία λύση και επαναλαμβανόμενα κινείται από μια λύση σε μια άλλη, αναζητώντας διαφορετικά τμήματα του χώρου αναζήτησης (search space), με σκοπό την ελαχιστοποίηση της τιμής της αντικειμενικής συνάρτησης για την αξιολόγηση της λύσης. Σε κάθε επανάληψη υπάρχει μόνο μία *τρέχουσα λύση* και δημιουργείται και ελέγχεται ένας χώρος αναζήτησης, ο οποίος (σε αυτή την επανάληψη) ονομάζεται *γειτονιά* της τρέχουσας λύσης (neighbourhood). *Κίνηση* ονομάζεται μια *τροποποίηση*, η οποία καθιστά μια λύση "*γειτονιά της λύσης*". Για να προλάβουμε *κυκλικές κινήσεις* (κινήσεις που μας οδηγούν σε καταστάσεις που έχουμε ήδη ελέγξει) χρησιμοποιείται η *λίστα Tabu* (Tabu List) για να αποθηκεύει πληροφορίες για τις ήδη εφαρμοζόμενες κινήσεις οι οποίες ονομάζονται *κινήσεις Tabu* (Tabu Moves). Αυτές οι κινήσεις απαγορεύεται να χρησιμοποιηθούν όσο βρίσκονται ακόμα στη λίστα Tabu. Ωστόσο,

κάποιες κινήσεις ίσως είναι αρκετά καλές και ίσως βελτιώσουν θεαματικά την τρέχουσα λύση, οπότε αυτές θα πρέπει να γίνουν αποδεκτές στη λύση και τότε πρέπει να διαγραφεί η κατάστασή τους (διαγραφή από την Tabu List), λόγω του ότι ικανοποιεί τα *κριτήρια αισιοδοξίας*. Σε κάθε επανάληψη οι κινήσεις Tabu ελέγχονται, αν ικανοποιούν τα κριτήρια αισιοδοξίας και αν ναι, τότε η Tabu κατάστασή τους (ο χαρακτηρισμός τους ως Tabu) θα πρέπει να απορριφθεί άμεσα (να διαγραφούν από τη λίστα Tabu). Σε πολλές περιπτώσεις τα κριτήρια αισιοδοξίας που χρησιμοποιούνται, είναι αυτά τα οποία δίνουν στην αντικειμενική συνάρτηση τιμή μικρότερη από την τρέχουσα ([KNKN10, BKS03, Jam04]).

4.4 Προσομοιωμένη ανόπτηση (simulated annealing).

Οι SA αλγόριθμοι βασίζονται στην Τεχνική Monte-Carlo και μπορούν να χρησιμοποιηθούν σε προβλήματα βελτιστοποίησης. Η τεχνική SA είναι εξομοίωση της τεχνικής ψύξης ενός συνόλου υπέρθερμων ατόμων (μορίων) που δονούνται. Όταν τα άτομα βρίσκονται σε πολύ υψηλή θερμοκρασία, μπορούν να κινούνται ελεύθερα τριγύρω και τείνουν να κινούνται προς τυχαίες κατευθύνσεις. Ωστόσο, όταν η ύλη κρυώνει, οι δεσμοί μεταξύ των σωματιδίων αναγκάζουν τα σωματίδια να είναι μαζί. Όταν η ύλη έχει κρυώσει ολοκληρωτικά, τα άτομα βρίσκονται σε ακινησία. Εάν η ύλη κρυώνει γρήγορα, η πιθανότητα να έχουμε μια λύση χαμηλού κόστους είναι μικρότερη από όταν η ύλη κρυώνει αργά. Μια νέα κατάσταση των ατόμων είναι αποδεκτή, εάν η ενέργεια του συστήματος είναι μικρότερη. Ωστόσο, εάν η ενέργεια είναι υψηλότερη, τότε η κατάσταση είναι αποδεκτή μόνο εάν η πιθανότητα μιας τέτοιας αύξησης είναι μικρότερη από αυτή που αναμενόταν στη συγκεκριμένη θερμοκρασία. Αυτά τα κριτήρια αποδοχής της λύσης, βασίζονται στην τεχνική της φυσικής που ονομάζεται *ανόπτηση των μετάλλων*. [Abr91]

Πολλά προβλήματα βελτιστοποίησης μπορεί να θεωρηθούν σαν ένα σύνολο από αντικείμενα, τα οποία απαιτείται να προγραμματισθούν, ώστε να ελαχιστοποιηθεί η τιμή μιας αντικειμενικής συνάρτησης. Τα άτομα που μετακινούνται αντικαθίστανται από αντικείμενα που απαιτείται να χρονοπρογραμματισθούν. Μια αρχική κατάσταση δημιουργείται από τυχαίο προγραμματισμό των αντικειμένων και υπολογίζεται το αρχικό κόστος και η θερμοκρασία. Στη συνέχεια δημιουργούνται τυχαίες αναδιατάξεις αντικειμένων και υπολογίζεται η μεταβολή του κόστους. Εάν το κόστος μειώνεται, η νέα κατάσταση γίνεται αποδεκτή, ενώ στην αντίθετη περίπτωση υπολογίζεται η πιθανότητα σε αυτή την νέα κατάσταση θερμοκρασίας. Ένα από τα πλεονεκτήματα της μεθόδου (σε σχέση με τους αλγορίθμους hill

climbing, και όχι μόνο), είναι ότι έχουν λιγότερες πιθανότητες να εγκλωβιστούν σε κάποιο τοπικό ελάχιστο, λόγω του ότι το κόστος ενδέχεται και να αυξηθεί, και όχι μόνο να μειωθεί.

4.5 Γενετικοί Αλγόριθμοι

Οι *γενετικοί αλγόριθμοι* (GA) εξομοιώνουν την κληρονομικότητα των ζωντανών οργανισμών και έχουν διαδοθεί ευρέως ως μέθοδος επίλυσης προβλημάτων βελτιστοποίησης και χρησιμοποιούνται σε προβλήματα πολλών ειδών. Αφενός έχουν την δύναμη να ξεφεύγουν από τα τοπικά ελάχιστα, όπου μπορεί να εγκλωβισθεί ένας αλγόριθμος και να αποτύχει και αφετέρου μπορεί να δώσει περισσότερες από μία βέλτιστες λύσεις στο πρόβλημα και έτσι ο χρήστης να έχει την δυνατότητα να επιλέγει. Οι γενετικοί αλγόριθμοι βελτιστοποίησης χρησιμοποιούν αιτιοκρατικές (deterministic) και στοχαστικές (stochastic) μεθόδους και έτσι μπορούν να δώσουν λύση σε (πολύ) NP-δύσκολα προβλήματα, όπως είναι αυτό του χρονοπρογραμματισμού (timetabling). Οι δύο κυριότερες λειτουργίες είναι το *crossover* (σπάσιμο συνόρων) και η *mutation* (μετάλλαξη). [GPV]

Στην περίπτωση του χρονοπρογραμματισμού η *συνέπεια* των ξεχωριστών γονιδίων είναι πολύ σημαντική. Με τη «συνέπεια» εννοούμε ότι η κάθε πληροφορία θα εμφανίζεται μόνο μία φορά στο γονότυπο. Έτσι αντί για διασπαστικές λειτουργίες θα χρησιμοποιούνται μόνο λειτουργίες σύνθεσης. Στη φύση, αυτό λαμβάνει χώρα με τρόπο, ώστε κάθε πληροφορία να υπάρχει σε προκαθορισμένη θέση στα χρωμοσώματα και κατά τη διάρκεια της διάσπασης μόνο, αυτά τα τμήματα μπορεί να αλλάζουν. Το σύστημα είναι αρκετά πλεοναστικό για να διορθώνεται, εάν κάποια τμήματα λείπουν ή υπάρχουν σε διπλότυπα, ή ακόμα και όταν οι αντιστοιχίσεις μεταξύ των γονιδίων δεν είναι αμοιβαίες. Παρά ταύτα, η συμβατότητα καταστρέφεται μερικές φορές, ακόμα και στην φύση, αλλά αυτό, στην περίπτωσή μας δεν πρέπει να το επιτρέψουμε.

Σε κάθε ωρολόγιο πρόγραμμα ουσιαστική σημασία έχει μόνο το ποια διδασκαλία θα λάβει χώρα σε κάθε χρονική μονάδα. Η σειρά έχει ενδιαφέρον μόνο στην αναπαράσταση του πίνακα. Με αυτή τη σκέψη μπορούμε να χρησιμοποιήσουμε μια γραμμική δομή για την αναπαράσταση δεδομένων, είτε με βάση τη στήλη των καθηγητών για κάθε χρονική στιγμή (κατακόρυφη αναπαράσταση), είτε οριζόντια, με βάση την χρονική μονάδα διδασκαλίας για όλους τους καθηγητές (οριζόντια αναπαράσταση). Δεν υπάρχει ουσιαστική διαφορά μεταξύ των δύο διαφορετικών τρόπων και το αποτέλεσμα είναι ότι θα χρησιμοποιηθεί αναπαράσταση συνόλων (set representation), και “κενά” σύνολα θα χρησιμοποιούνται για να δηλώσουν ότι κάποια χρονική περίοδο, συγκεκριμένος καθηγητής δεν θα διδάσκει.

4.5.1 Γενετικοί Τελεστές.

Η *διάσχιση (crossover)* είναι ένας από τους πιο ουσιώδεις *τελεστές*. Ο σκοπός τους είναι η πραγματοποίηση της αιτιοκρατικής αναζήτησης, και προσπαθεί να αναβαθμίσει το χώρο του προβλήματος, τη σωζόμενη γνώση. Ο τελεστής αυτός από δύο δεδομένα άτομα παράγει ένα νέο, του οποίου η γενετική υπόστρωση, είναι ο συνδυασμός γονιδίων των δύο προηγούμενων ατόμων. Πρακτικά από τα δύο αρχικά άτομα, δημιουργούνται τέσσερα νέα άτομα και συνολικά το γενετικό υπόστρωμα των τεσσάρων απογόνων καλύπτει το γενετικό υπόστρωμα των γονιών. Έτσι η πιθανότητα για καλύτερο αποτέλεσμα είναι ευνοϊκή.

Ο ρόλος της μετάλλαξης στους GA είναι η επιβεβαίωση της *μεταευρετικής αναζήτησης* (heuristic search). Προσπαθεί να φτάσει στα άτομα που βρίσκονται στην μέχρι εκείνη τη στιγμή ανεξερεύνητη περιοχή του χώρου του προβλήματος. Η βασική περίπτωση της μετάλλαξης είναι η παρακάτω: διάλεξε ένα σημείο και στη συνέχεια σε ένα άτομο αντιμετάθεσε τμήμα τυχαίου μήκους από το επιλεγμένο σημείο, με τμήμα ίδιου μήκους αρχίζοντας από διαφορετικό σημείο. Μια πιο αποτελεσματική λύση, σύμφωνα με ερευνητικά αποτελέσματα [GDM92, GPV], δημιουργείται όταν τμήματα με ένα αντικείμενο αντιμετατίθενται πολλές φορές. Σε κάποιες περιπτώσεις η λειτουργία της αντιστροφής αποδεικνύεται πιο αποτελεσματική, όπως αναφέρεται σε βιβλιογραφία. [GPV, FDH14]

4.5.2 Οι τιμές ποινής.

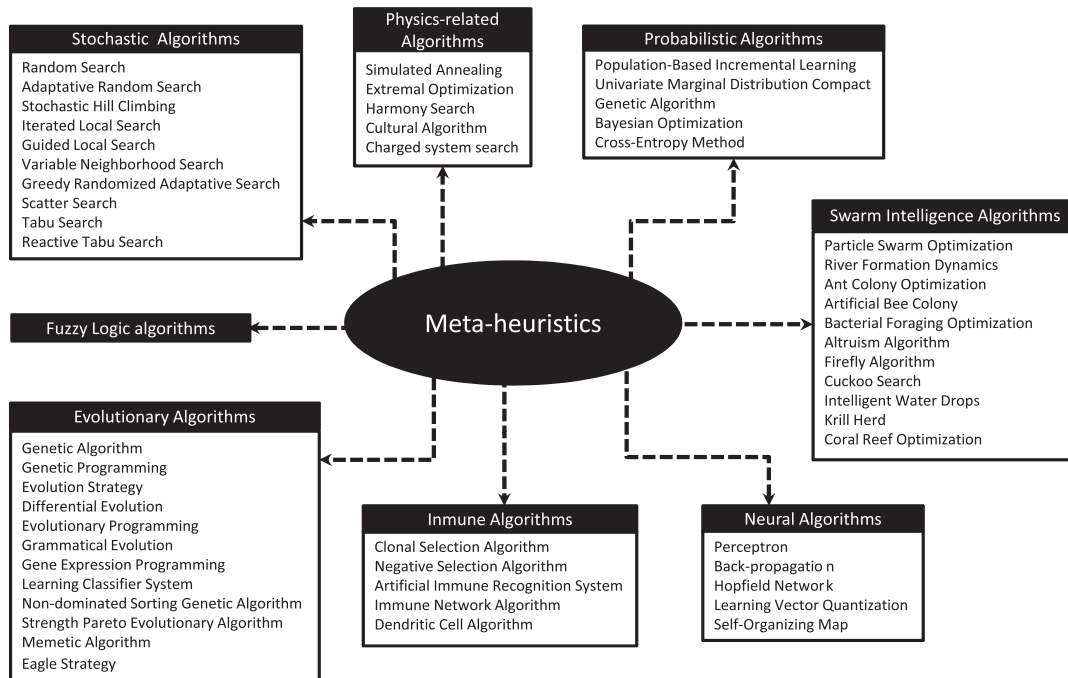
Οι ποινές δηλώνουν σε ποιο βαθμό μια κατάσταση μπορεί να οδηγήσει σε μη εφαρμόσιμο ωρολόγιο πρόγραμμα. Η μονάδα μέτρησης είναι αυθαίρετη, οπότε δεν έχει αξία η τιμή της ποινής τόσο, όσο ο λόγος της τιμής προς την μονάδα. Κάθε παραβίαση ενός περιορισμού μπορεί να έχει διαφορετική τιμή τιμωρίας και εξαρτάται από τον ίδιο τον περιορισμό.

4.6 Ο χώρος των μεταευρετικών τεχνικών.

Για την επίλυση προβλημάτων συνδυαστικής βελτιστοποίησης υπάρχουν πολλές τεχνικές. Οι τεχνικές αυτές είναι ο μαθηματικός προγραμματισμός, η τεχνητή νοημοσύνη, η υπολογιστική νοημοσύνη και οι μεταευρετικές τεχνικές. Δεν φαίνεται όμως να έχει ξεχωρίσει κάποια μέθοδος ή κάποια κατηγορία μεθόδων, η οποία να αντιμετωπίζει τα προβλήματα στο μεγαλύτερό τους μέρος με τρόπο αποτελεσματικότερο από τις υπόλοιπες κατηγορίες.

Το συμπέρασμα αυτό δεν ισχύει για υποσύνολα του συνόλου των προβλημάτων, έτσι ενδέχεται πράγματι να υπάρχει κάποια αποδοτικότερη μέθοδος για συγκεκριμένες κατηγορίες προβλημάτων σε σχέση με κάποιες άλλες. Το σχήμα που ακολουθεί ([MLG13]) παρουσιάζει το χώρο με τις μεταερευτικές τεχνικές, κατηγοριοποιημένο με την ιδέα που μπορεί να ακολουθήσει κάποιος για να εντάξει το πρόβλημα προς έρευνα σε ένα ή περισσότερα πεδία.

D. Manjarres et al. / Engineering Applications of Artificial Intelligence 26 (2013) 1818–1831



Σχήμα 4-1: Γενική κατηγοριοποίηση μεταερευτικών τεχνικών βασισμένων στη διαδικασία αναζήτησης. [MLG13]

ΚΕΦΑΛΑΙΟ V

Το πρόβλημα δημιουργίας ωρολογίου προγράμματος στην ελληνική δημόσια παιδεία δευτεροβάθμιας εκπαίδευσης.

Τα τελευταία χρόνια έχουν γίνει πολλές εξειδικευμένες έρευνες και μελέτες περιπτώσεων σε σχολικές μονάδες σε αρκετά κράτη (όπως στην Αυστραλία [Kin16], Βραζιλία [FS13,FDH14,FHC16a,FHC16b], Ελλάδα [Val12], Ιταλία [FHC16b], Ολλανδία [DBH10,FDH14], Ισπανία [FDH14, DBH10], Ηνωμένο Βασίλειο [DBH10,FDH14] και Δανία [Kri15, Sti14],[Pil14]) που αφορούν το timetabling. Οι μελέτες αυτές έχουν δείξει ότι το πρόβλημα αυτό διαφέρει ανάλογα με το κράτος, το εκπαιδευτικό σύστημα και τη φιλοσοφία του κάθε εκπαιδευτικού συστήματος [Post12a].

Τα δεδομένα για το ωρολόγιο πρόγραμμα στην Ελληνική Δευτεροβάθμια Εκπαίδευση (Δ.Ε.) είναι πολυσύνθετα. Είναι γνωστό ότι σε κάθε εβδομάδα οι διδασκαλίες λαμβάνουν χώρα μόνο τις 5 εργάσιμες ημέρες και κάθε ημέρα τα σχολεία λειτουργούν το πολύ 7 ώρες (συνήθως).

Στους πόρους ανήκουν οι αίθουσες που θα χρησιμοποιηθούν (και μάλιστα αρκετές φορές). Έχει μεγάλη σημασία πόσες ειδικές αίθουσες, όπως εργαστήρια φυσικής, χημείας, βιολογίας, πληροφορικής, πολυμέσων, ξένων γλωσσών υπάρχουν, λόγω του ότι κάθε αίθουσα μπορεί να χρησιμοποιηθεί μόνο από ένα τμήμα κάθε φορά. Στη συνέχεια, θα αναφέρουμε τα μαθήματα που διδάσκονται σε κάθε τάξη και τους καθηγητές με τις αναθέσεις διδασκαλίας μαθημάτων που θα διδάσκουν. Τέλος σημαντικό ρόλο παίζει και το πλήθος των τμημάτων των μαθητών που συνθέτουν την κάθε τάξη.

Η Δευτεροβάθμια Εκπαίδευση περιλαμβάνει δύο βαθμίδες, το Γυμνάσιο και το Λύκειο.

5.1 Το Γενικό Λύκειο.

Στο γενικό λύκειο, σε όλες τις τάξεις Α, Β και Γ, φοιτούν γενικά αρκετοί μαθητές με το σύνηθες φαινόμενο κάθε τάξη να διασπάται σε ένα αριθμό τμημάτων (συνήθως μέγιστο όριο είναι το 27 έως σπανίως και 35 μαθητές ανά τμήμα). Έτσι κάθε τάξη διαχωρίζεται σε αρκετά τμήματα, στα οποία διδάσκονται κοινά μαθήματα όλοι οι μαθητές της κάθε τάξης, με κάθε τμήμα να ακολουθεί διαφορετικό πρόγραμμα.

Κάθε χρόνο το Υπουργείο Παιδείας (ΥΠ) ανακοινώνει τα μαθήματα τα οποία οι μαθητές κάθε τάξης πρέπει να διδάσκονται. Παρατηρώντας τους πίνακες αυτούς έχουμε τα

δεδομένα ότι οι μαθητές κάθε τάξης υποχρεώνονται να παρακολουθήσουν τα μαθήματα γενικής παιδείας (ΓΠ) και επιπλέον μαθήματα άλλων ομάδων. Τέτοιες ομάδες αποτελούν 1) τα μαθήματα ξένων γλωσσών (ΞΓ), τα οποία ο μαθητής οφείλει να παρακολουθεί για ολόκληρη την περίοδο των 3 τάξεων και δεν δικαιούται να αλλάξει γλώσσα διδασκαλίας, 2) τα μαθήματα επιλογής (ΜΕ), από τα οποία ο μαθητής μπορεί να επιλέγει διαφορετικά μαθήματα κάθε έτος σπουδών (συνήθως είναι ένα μάθημα στην Α και ένα στην Γ λυκείου), 3) τα μαθήματα προσανατολισμού (ΠΡ και παλαιότερα Κατευθύνσεων), τα οποία ουσιαστικά είναι μαθήματα προετοιμασίας των μαθητών για εισαγωγικές εξετάσεις στην Τριτοβάθμια Εκπαίδευση και αφορούν τις 2 τελευταίες τάξεις (με 5 και 15 ώρες διδασκαλίας για τις τάξεις Β και Γ αντίστοιχα) και τέλος 4) η Ερευνητική Εργασία (Project η PR) το οποίο διδάσκεται για 2 και 1 ώρες στην Α και Β λυκείου αντίστοιχα.

Όταν υπάρχουν όλες οι κατηγορίες προγραμμάτων για κάθε τάξη, τότε οι συνδυασμοί είναι πολύπλοκοι. Παρακάτω ακολουθεί ένας πίνακας με τις κατηγορίες μαθημάτων και τις ώρες διδασκαλίας ανά τάξη για κάθε τέτοια κατηγορία:

Κατηγορία μαθημάτων	Τάξη Α (διδ. ώρες)	Β (διδ. ώρες)	Γ (διδ. ώρες)
Γενικής παιδείας	29	27	13
Προσανατολισμού	0	(2 + 3=) 5	(5+3+3+2+2=)15
Ξένων Γλωσσών	2	2	2
Επιλογής	2	0	2
Ερευνητικής Εργασίας	2	1	0
Σύνολο	35	35	32

Πίνακας 5-1: Ώρες Διδασκαλίας ανά τάξη στο Ελληνικό Γενικό Λύκειο

Τα μαθήματα κάθε κατηγορίας, κάθε τάξης, πρέπει να διδάσκονται σε ζώνες. Δεν είναι δυνατόν να γίνονται διδασκαλίες παρά μόνο για μια ζώνη ανά τάξη, σε κάθε διδακτική ώρα. Δηλαδή είναι αδύνατο να γίνεται διδασκαλία κάποιου μαθήματος επιλογής την ίδια ώρα που διδάσκεται κάποιο μάθημα ξένης γλώσσας, ή μάθημα γενικής παιδείας. Έτσι δημιουργούνται οι ζώνες διδασκαλίας για τις ομάδες μαθημάτων κάθε τάξης. Για παράδειγμα, οι 15 ώρες διδασκαλίας για τα μαθήματα προσανατολισμού θα γίνονται πάντα ώρες κατά τις οποίες δεν διδάσκεται τίποτε άλλο, παρά μόνο τα συγκεκριμένα μαθήματα προσανατολισμού, σύμφωνα με τον χωρισμό των μαθητών σε τμήματα προσανατολισμού. Σε αρκετά σχολεία η διεύθυνση δεσμεύει έτσι ώστε αυτές οι ώρες εκ των προτέρων να είναι οι 3 πρώτες ώρες κάθε ημέρας, (για διάφορους λόγους) και για αυτό τις ονομάζουμε **ζώνες**.

Σε περίπτωση που δεν επιθυμούμε να δεσμεύσουμε συγκεκριμένες ώρες, μπορούμε να ορίσουμε χαλαρούς περιορισμούς, ώστε τα μαθήματα της κάθε ζώνης να είναι τουλάχιστον συνεχόμενα στις διδασκαλίες κάθε ημέρας. Δεν είναι επιθυμητό να εναλλάσσονται συνεχώς μαθήματα διάφορων ζωνών. Η εναλλαγή μαθημάτων από διάφορες ζώνες συνεπάγεται συνεχής αλλαγή αίθουσας για το μεγαλύτερο πλήθος των μαθητών.

Εάν, επίσης, παρατηρήσουμε τις ομάδες μαθημάτων, θα προσέξουμε ότι κάποιες από αυτές αφορούν μαθήματα της ίδιας ειδικότητας και απαιτείται να διδάσκονται ταυτόχρονα, δηλαδή να δημιουργείται ζώνη διδασκαλίας μόνο για μαθήματα μιας ειδικότητας. Για παράδειγμα, στις ξένες γλώσσες δημιουργείται αναγκαστικά ζώνη, με συνέπεια να απαιτείται να διδάσκονται όλα τα τμήματα παράλληλα ξένες γλώσσες, και εάν υπάρχουν π.χ. 4 τμήματα αγγλικών και 1 κάποιας άλλης γλώσσας (π.χ. γαλλικών) – φαινόμενο αρκετά συνηθισμένο – τότε πρέπει να βρίσκονται στην σχολική μονάδα 4 καθηγητές αγγλικών και ένας ακόμα γαλλικών. Παρόμοιο φαινόμενο δημιουργείται για το μάθημα επιλογής της Α λυκείου, στην περίπτωση που δημιουργούνται πολλά τμήματα για διδασκαλία πληροφορικής και μόνο ένα τμήμα το οποίο διδάσκεται κάποιο άλλο μάθημα (π.χ. καλλιτεχνικά).

Το φαινόμενο αυτό δημιουργεί ιδιαίτερα αυξημένο πρόβλημα στη σχολική μονάδα, όταν για κάποιο αντικείμενο διδασκαλίας, παρευρίσκεται στη σχολική μονάδα μόνο ένας καθηγητής ο οποίος έχει την δυνατότητα να το διδάξει. Στην περίπτωση αυτή, πρέπει να μετακινηθούν στο σχολείο αυτό καθηγητές από άλλες σχολικές μονάδες, τις ίδιες ώρες, για να καλύψουν τις διδασκαλίες. Το πρόβλημα ενδέχεται να επιδεινώνεται, εάν αυτό συμβαίνει σε περισσότερες τάξεις. Στις περιπτώσεις αυτές δημιουργούνται πολλά προβλήματα για τους καθηγητές αυτών των ειδικοτήτων (ξένων γλωσσών, πληροφορικής, κ.α.). Στα σχολεία αυτά θα υπάρχει η απαίτηση αρκετοί καθηγητές (ίσως και περισσότεροι από 4) να μετακινούνται ταυτόχρονα για διδασκαλία για λίγες μόνο ώρες (συνήθως όχι περισσότερες από 6 ώρες ανά εβδομάδα), και μάλιστα για τόσες ημέρες όσες είναι οι εβδομαδιαίες ώρες διδασκαλίας του μαθήματος (συνήθως δύο ημέρες την εβδομάδα).

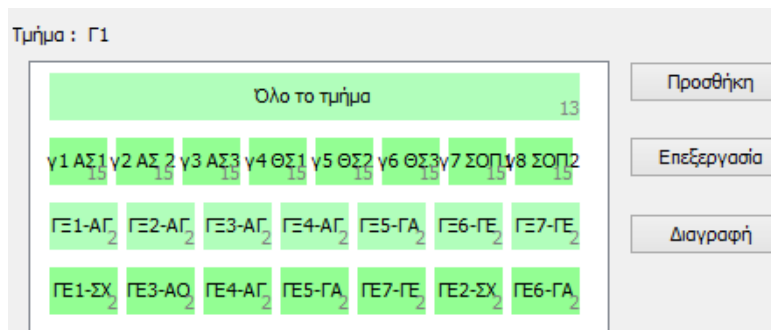
Αντίστοιχο πρόβλημα δημιουργείται για το μάθημα της φυσικής 3 ωρών, στο προσανατολισμό θετικών σπουδών της Β λυκείου όπου, λόγω του ότι διδάσκεται σε ζώνη διάρκειας 5 διδακτικών ωρών ανά εβδομάδα, κανένας καθηγητής δεν μπορεί να διδάξει σε περισσότερα από ένα τμήμα. Έτσι απαιτούνται τόσοι καθηγητές φυσικής, όσα τμήματα προσανατολισμού θετικών σπουδών έχουμε. Εάν ένας καθηγητής Φυσικής αναλάμβανε 2 τμήματα για 3ωρη διδασκαλία το κάθε ένα, τότε θα απαιτούνταν να διδάξει 6 ώρες φυσική

στο διάστημα των 5 ωρών που απαιτεί το πρόγραμμα για τα μαθήματα προσανατολισμού της Β τάξης του Λυκείου (γεγονός αδύνατο!).

Ακραία παραδείγματα (που συμβαίνουν συχνά, ακόμα και στο σχολείο του συγγραφέα) είναι ότι απαιτούνται α) 4 καθηγητές αγγλικών για παράλληλη διδασκαλία 4 τμημάτων (με ένα μόνιμο καθηγητή της ειδικότητας εντός σχολείου και τρεις επιπλέον καθηγητές από άλλα σχολεία) β) 4 καθηγητές φυσικής, λόγω της ύπαρξης 4 τμημάτων θετικής κατεύθυνσης, εκ των οποίων οι 2 καθηγητές φυσικής να είναι του ιδίου σχολείου, και οι ανάγκες να καλύπτονται από 2 επιπλέον καθηγητές από άλλα σχολεία (για 3 ώρες διδασκαλία ανά εβδομάδα). Σε τέτοιες περιπτώσεις ενδέχεται κάποιος καθηγητής φυσικής του σχολείου να μην συμπληρώνει τις απαραίτητες ώρες διδασκαλίας στο σχολείο του με αποτέλεσμα καθηγητές της ίδιας ειδικότητας να ανταλλάσσονται μεταξύ των σχολείων λόγω αυτών των ζωνών διδασκαλίας. Η κατάσταση αυτή έχει αρνητικές συνέπειες και για τα άλλα σχολεία, από τα οποία προέρχονται οι καθηγητές, εκτός του ότι κάτι αντίστοιχο θα συμβαίνει και στα σχολεία προέλευσης των καθηγητών. Συνήθως άμεση συνέπεια είναι η αδυναμία εύρεσης ικανοποιητικής λύσης του προβλήματος.

Η ύπαρξη ζωνών Μαθημάτων Επιλογής (ΜΕ) δημιουργεί επίσης πολλά προβλήματα, σύμφωνα με το υπαρκτό πρόγραμμα σπουδών. Τα μαθήματα επιλογής διδάσκονται παραδοσιακά τις τελευταίες ώρες, όπου οι μαθητές συνήθως είναι κουρασμένοι. Για τα αντικείμενα διδασκαλίας τους ενδέχεται συχνά, να μην υπάρχει καθηγητής της ειδικότητας στη σχολική μονάδα (π.χ. σχέδιο γραμμικό ή ελεύθερο). Τότε θα πρέπει να διατεθεί καθηγητής για 4 ώρες, οι οποίες θα πρέπει να τοποθετηθούν στις 4 τελευταίες ώρες για 2 δίωρα ή για 4 διδασκαλίες της μιας ώρας σε 4 ημέρες (δηλαδή να αναγκάζεται να έρχεται καθηγητής στην σχολική μονάδα για να διδάσκει την 7^η ώρα, για 4 από τις 5 ημέρες του εβδομαδιαίου προγράμματος). Αυτό εκ των πραγμάτων δημιουργεί πολύ κακές λύσεις και δεν επιτρέπουμε στο λογισμικό να μας βοηθήσει στον υπολογισμό μιας αξιόλογης λύσης. Στην περίπτωση που η λύση περιέχει δύο δίωρα, τότε αυτά θα πρέπει να αποτελούν ένα συνεχόμενο δίωρο για κάθε τμήμα, σε δύο διαφορετικές ημέρες (και έτσι παραβιάζεται άλλος ένας περιορισμός). Εάν επιχειρήσουμε να τοποθετήσουμε στο δίωρο του καθηγητή 2 διαφορετικά τμήματα, τότε αυτό θα παραβιάσει αυστηρό περιορισμό δημιουργώντας από ένα κενό διδασκαλίας σε κάθε τμήμα την 6 ώρα. Στην περίπτωση που δημιουργηθούν 3 τμήματα τότε, εκτός των άλλων απαιτείται και δεύτερος καθηγητής. Δηλαδή για κάθε δύο τμήματα απαιτείται ένας επιπλέον καθηγητής.

5.1.1 Ο χωρισμός των τάξεων σε ομάδες τμημάτων ανά ζώνη.



Εικόνα 5-1: Χωρισμός τάξης-τμήματος σε κατηγορίες (ζώνες διδασκαλίας)

Στη διπλανή εικόνα φαίνεται ο χωρισμός μιας τάξης (ή καλύτερα κάθε τμήματός της) σε τμήματα ανά ζώνη. Αρχικά η Γ' τάξη λυκείου έχει χωρισθεί σε 6 τμήματα (Γ1 – Γ6) που θεωρείται και ο αρχικός

“κανονικός” χωρισμός. Στη συνέχεια η τάξη πρέπει να χωρισθεί σε άλλες 3 ζώνες/κατηγορίες μαθημάτων. Για να επιτευχθεί αυτό, χωρίζεται κάθε τμήμα στις ίδιες νέες ομάδες-τμήματα, και έτσι στο κάθε νέο τμήμα συμμετέχουν μαθητές από όλη την Γ' λυκείου, σύμφωνα με κάθε νέο χωρισμό. Τελικά η Γ τάξη του λυκείου χωρίζεται ως εξής:

Γ Λυκείου	Ώρες διδασκαλίας	Πλήθος τμημάτων	Τμήματα
Γεν. Παιδείας	13	6	Γ1 - Γ6
Ομάδων Προσανατολισμού	15	8	γ1 - γ8
Ξένων Γλωσσών	2	7	ΓΕΓλ1 - ΓΕΓλ7 (1-4 αγγλ, 5 γαλλ, 6-7 γερμ)
Επιλογής	2	7	ΓΕ1 - ΓΕ7
Σύνολο	32		

Πίνακας5-2: Γ Λυκείου - Ώρες διδασκαλίας και τμήματα ανά κατηγορία μαθημάτων.

Στον παρακάτω πίνακα φαίνονται, με το πράσινο χρώμα, οι χρονικές θέσεις της εβδομάδας (σε πίνακα 5 ημερών οριζοντίως και 7 ωρών κατακόρυφα) στις οποίες η κάθε

ΓΚ Μαθηματικά	ΜαθΓΚ	25	
Γ Ξενη Γλώσσα	Γ ΞΓ	14	
ΓΕ ΕΠΙΛΟΓΗΣ	Γ ΕΠΙΑ	14	

Εικόνα 5-2: Ζώνες διδασκαλίας για την Γ λυκείου (α-προσανατολισμού, β-Ξένης γλώσσας, γ-μαθημάτων επιλογής)

μία από τις τρεις τελευταίες κατηγορίες μαθημάτων, επιτρέπεται να διδαχθεί. Αυτό επιτυγχάνεται δηλώνοντας ότι όλα τα

μαθήματα της κάθε κατηγορίας επιτρέπεται να διδάσκονται τις ίδιες συγκεκριμένες ώρες

της αντίστοιχης κατηγορίας). Έτσι, στη θέση που είναι τα μαθηματικά (σε 15 ώρες – πράσινο χρώμα) θα τοποθετηθούν όλα τα μαθήματα προσανατολισμού, στη δεύτερη γραμμή φαίνεται ποιες ακριβώς 2 ώρες διδάσκονται οι ξένες γλώσσες και στην τρίτη γραμμή φαίνεται ότι υπάρχουν 2 διαθέσιμες θέσεις (δίωρου) για το δίωρο μάθημα ζώνης επιλογής. Οι υπόλοιπες ώρες (κοινές ώρες με κόκκινο χρώμα) “αφήνονται ελεύθερες” για τα μαθήματα γενικής παιδείας της Γ λυκείου.

Ο κάθε μαθητής με το τρόπο αυτό ανήκει σε 4 διαφορετικά τμήματα, δηλαδή σε ένα από κάθε ζώνη. Για παράδειγμα ένας μαθητής Γ τάξης που ανήκει στο Γ1, μπορεί να ανήκει στο γ3 (3^ο τμήμα) προσανατολισμού, στο ΓΞ5 (5^ο τμήμα) ξένης γλώσσας και στο ΓΕ2 (2^ο τμήμα) της ζώνης μαθημάτων επιλογής.

5.2 Το Επαγγελματικό Λύκειο (ΕΠΑ.Λ.).

Στο επαγγελματικό λύκειο αντί για τους προσανατολισμούς υπάρχουν οι ειδικότητες, οι οποίες είναι περισσότερες, αλλά ακολουθούν την ίδια φιλοσοφία με τις ζώνες του γενικού λυκείου. Για την μελέτη του προβλήματος αυτό δεν σημαίνει κάτι καινούργιο. Στην πράξη, για το λογισμικό, θα υπάρχουν οι ίδιες δομές, εκτός από το ότι το πλήθος των τμημάτων για τις ειδικότητες θα είναι πιθανότατα μεγαλύτερο και με επίσης περισσότερα εργαστήρια. Έτσι δεν θα αναφερθούμε περισσότερο στο θέμα.

5.3 Το Γυμνάσιο.

Στο γυμνάσιο η κατάσταση σύμφωνα με το αναλυτικό ωρολόγιο πρόγραμμα και του τελευταίου έτους αλλά και των προηγούμενων ετών είναι πιο απλή από το λύκειο. Οι μαθητές κάθε τάξης διαχωρίζονται σε τμήματα στα οποία διδάσκονται το μεγαλύτερο πλήθος των μαθημάτων τους. Τα μαθήματα στα οποία διαφέρει η κατάσταση είναι οι ξένες γλώσσες (όπου οι μαθητές δύνανται να διαχωρίζονται σε τμήματα με διαφορετικό τρόπο – επίπεδα) και η πληροφορική με την τεχνολογία, όπου κάθε τμήμα σπάει σε 2 μικρότερα τμήματα και έτσι έχουμε 2 καθηγητές να διδάσκουν στα δύο υπο-τμήματα. Οι καταστάσεις αυτές εύκολα εντάσσονται στο πολύπλοκο σύστημα των λυκείων και έτσι δεν θα ασχοληθούμε ειδικότερα με το πρόβλημα των γυμνασίων.

5.4 Άλλες κατηγορίες σχολείων.

Εκτός από τις αναφερόμενες κατηγορίες σχολείων, υπάρχουν ακόμα και τα μουσικά σχολεία, τα εκκλησιαστικά, τα σχολεία ειδικής αγωγής, και τα μειονοτικά. Ομοίως και αυτά

δεν έχουν κάτι πιο πολύπλοκο στη δομή τους, το οποίο να μην μπορεί να περιγραφεί από τη δομή που αναφέρθηκε παραπάνω. Ιδιαίτερα στα μουσικά σχολεία, τα μαθήματα διδάσκονται για 8 ή 9 ώρες κάθε ημέρα και οι τελευταίες ώρες αφιερώνονται στα μουσικά μαθήματα (σε ζώνη διδασκαλίας μουσικών μαθημάτων). Επίσης σε κάποια πρακτικά μουσικά μαθήματα η διδασκαλία είναι εξατομικευμένη (ανά μαθητή), όπως για το πιάνο, το βιολί, κ.α.

ΚΕΦΑΛΑΙΟ VI

Η ΠΕΡΙΠΤΩΣΗ ΕΝΟΣ ΓΕΝΙΚΟΥ ΛΥΚΕΙΟΥ.

6.1 Τα δεδομένα του προβλήματος.

Να αναφέρουμε αρχικά ότι το λογισμικό που χρησιμοποιήθηκε στο συγκεκριμένο σχολείο είναι το asc TimeTable, έκδοσης 2017. Η σχολική μονάδα μελέτης είναι γενικό λύκειο και έχει τρεις τάξεις (Α, Β και Γ). Κάθε τάξη πρέπει να διδαχθεί τα αντικείμενα διδασκαλίας της σε μια εβδομάδα των 5 ημερών με το πολύ 7 ώρες διδασκαλίας καθημερινά. Από τα δεδομένα του Υπουργείου και τις επιλογές των μαθητών έχουμε τα παρακάτω δεδομένα.

Κατηγορία μαθημάτων	Τάξη		
	Α	Β	Γ
Γενικής παιδείας	29	27	13
Προσανατολισμού	0	(2 + 3=) 5	(5+3+3+2+2=)15
Ξένων Γλωσσών	2	2	2
Επιλογής	2	0	2
Ερευνητικής Εργασίας	2	1	0
Σύνολο	35	35	32

Πίνακας 6-1: Ώρες διδασκαλίας στο λύκειο ανά τάξη και κατηγορία μαθημάτων

Στη συνέχεια, βάση του παραπάνω πίνακα, θα χρειαστεί να γνωρίζουμε τα συγκεκριμένα τμήματα που θα υπάρχουν σε κάθε τάξη και τα οποία περιέχονται στους παρακάτω πίνακες.

Γ Λυκείου	Ώρες διδασκαλίας ανά τμήμα	Πλήθος τμημάτων	Σύνολο ωρών διδασκαλίας	Τμήματα
Γεν. Παιδείας (κατηγορία K1)	13	6	78	Γ-K1-1 μέχρι Γ-K1-6
Ομάδων Προσανατολισμού (κατηγορία K2)	15	8	120	Γ-K2-1 μέχρι Γ-K2-8
Ξένων Γλωσσών (κατηγορία K3)	2	7	14	Γ-K3-1 μέχρι Γ-K3-7
Επιλογής (κατηγορία K4)	2	7	14	Γ-K4-1 μέχρι Γ-K4-7
Project (πιθανή κατηγορία K5)	Δεν προβλέπεται στην Γ λυκείου			
Σύνολο	32	28	226	

Πίνακας 6-2: Ώρες και τμήματα της Γ Λυκείου ανά κατηγορία μαθημάτων.

B Λυκείου	Ώρες διδασκαλίας ανά τμήμα	Πλήθος τμημάτων	Σύνολο ωρών διδασκαλίας	Τμήματα
Γεν. Παιδείας (κατηγορία K1)	29	4	116	B-K1-1 μέχρι B-K1-4
Ομάδων Προσανατολισμού (κατηγορία K2)	6 (μαζί με το Project)	5	30	B-K2-1 μέχρι B-K2-5
Ξένων Γλωσσών (κατηγορία K3)	Συμπεριλαμβάνονται στα γενικής παιδείας λόγω μη δημιουργίας τμημάτων δεύτερης ξένης γλώσσας. <u>Κόστος</u> : κάποιοι μαθητές υποχρεούνται να παρακολουθήσουν Ξένη Γλώσσα που δεν επιθυμούν ενώ μέχρι την Α λυκείου παρακολουθούσαν διαφορετική Ξένη Γλώσσα.			
Επιλογής (κατηγορία K4)	Δεν προβλέπεται στην Β λυκείου.			
Project (1 ώρα/εβδ) (κατηγορία K2)	Συμπεριλαμβάνεται στην ομάδα K2, με σκοπό να απαιτηθεί ένας καθηγητής ειδικότητας ΠΕ04.01 λιγότερος για μετακίνηση προς τη σχολική μονάδα, με κόστος αύξησης μιας διδακτικής ώρας ανά εβδομάδα.			
Σύνολο	35	9	146	

Πίνακας 6-3: Ώρες και τμήματα της Β Λυκείου ανά κατηγορία μαθημάτων

A Λυκείου	Ώρες διδασκαλίας ανά τμήμα	Πλήθος τμημάτων	Σύνολο ωρών διδασκαλίας	Τμήματα
Γεν. Παιδείας (κατηγορία K1)	35	4	140	A-K1-1 μέχρι A-K1-4
Ομάδων Προσανατολισμού (κατηγορία K2)	Δεν προβλέπεται στην Α λυκείου			
Ξένων Γλωσσών (κατηγορία K3)	Όλοι σχεδόν επέλεξαν Αγγλικά. Συμπεριλαμβάνονται στα γενικής παιδείας λόγω μη δημιουργίας τμημάτων δεύτερης ξένης. <u>Κόστος</u> : κάποιοι μαθητές υποχρεούνται να παρακολουθήσουν Ξένη Γλώσσα που δεν επιθυμούν ενώ μέχρι την Α λυκείου παρακολουθούσαν διαφορετική Ξένη Γλώσσα.			
Επιλογής (κατηγορία K4)	Συμπεριλαμβάνεται στην κατηγορία K1 διότι οι μαθητές έτυχε να επιλέξουν σχεδόν όλοι το ίδιο μάθημα.			
Project (1ώρα/εβδ)	Συμπεριλαμβάνεται στην ομάδα K1 για λόγους εύρυθμης λειτουργίας του σχολείου και του ωρολογίου προγράμματος.			
Σύνολο	35	4	140	

Πίνακας 6-4: Ώρες και τμήματα της Α Λυκείου ανά κατηγορία μαθημάτων

Τα παραπάνω δεδομένα και σύμφωνα με την ανάλυση για τις ζώνες (η οποία έγινε σε προηγούμενο κεφάλαιο), συνεπάγονται τη δημιουργία των ζωνών μαθημάτων που παρουσιάζονται στους παρακάτω πίνακες.

		Ωρες	Τμήματα	Συν. Ωρες	
Γ Λυκείου	ΓΠ	13	6	78	ZΩNH Γ1
	ΠΡ	15	8	120	ZΩNH Γ2
	ΞΓ	2	7	14	ZΩNH Γ3
	ΕΠ	2	7	14	ZΩNH Γ4
Σύνολα		32	28	226	

Πίνακας 6-5: Ωρες και Γεγονότα που πρέπει να τοποθετηθούν στο Ωρολόγιο πρόγραμμα για την Γ λυκείου.

		Ωρες	Τμήματα	Συν. Ωρες	
Β Λυκείου	ΓΠ	29	4	116	ZΩNH Β1
	ΠΡ	6	5	30	ZΩNH Β2
	ΞΓ	Δεν δημιουργήθηκαν			ZΩNH Β3
	ΡΡ	Δεν δημιουργήθηκαν			ZΩNH Β4
Σύνολα		35	9	146	

Πίνακας 6-6: Ωρες και Γεγονότα που πρέπει να τοποθετηθούν στο Ωρολόγιο πρόγραμμα για την Β λυκείου.

		Ωρες	Τμήματα	Συν. Ωρες	
Α Λυκείου	ΓΠ	35	4	140	ZΩNH Α1
	ΕΠ	Δεν δημιουργήθηκαν			ZΩNH Α2
	ΞΓ	Δεν δημιουργήθηκαν			ZΩNH Α3
Σύνολα		35	4	140	

Πίνακας 6-7: Ωρες και Γεγονότα που πρέπει να τοποθετηθούν στο Ωρολόγιο πρόγραμμα για την Α λυκείου.

Έτσι οι συνολικές ώρες διδασκαλίες θα είναι 512 και τα τμήματα 41, με την Γ λυκείου να λειτουργεί σε 4 διαφορετικές ζώνες ενώ η Β σε 2 και η Α σε 1 ενιαία ζώνη. Θα ελεγχθεί και η περίπτωση που τα τμήματα θα ήταν σε πλήρη εξέλιξη επειδή α) αυτό είναι το λογικό και το πρόβλημα και δεύτερο θα φανούν τα προβλήματα που δημιουργούνται στις περισσότερες σχολικές μονάδες και είναι αξιολογικά για μελέτη. Έτσι στην περίπτωση που δημιουργούνταν τα τμήματα θα είχαμε τα διαφορετικά δεδομένα όπως παρουσιάζονται στις Β και Α λυκείου:

		Ωρες	Τμήματα	Συν. Ωρες	
Β Λυκείου	ΓΠ	27	4	108	ZΩNH Β1
	ΠΡ	5	5	25	ZΩNH Β2
	ΞΓ	2	5	10	ZΩNH Β3
	ΡΡ	1	5	5	ZΩNH Β4
Σύνολα		35	19	148	

Πίνακας 6-8: Ωρες και Γεγονότα που θα έπρεπε να τοποθετηθούν στο Ωρολόγιο πρόγραμμα για την Β λυκείου, εάν είχαν δημιουργηθεί όλες οι κατηγορίες.

		Ωρες	Τμήματα	Συν. Ωρες	
A Λυκείου	ΓΠ	29	4	116	ZΩNH A1
	ΕΠ	2	5	10	ZΩNH A2
	ΞΓ	2	5	10	ZΩNH A3
	PR	2	5	10	ZΩNH A4
Σύνολα		35	19	146	

Πίνακας 6-9: Ωρες και Γεγονότα που θα έπρεπε να τοποθετηθούν στο Ωρολόγιο πρόγραμμα για την Α λυκείου εάν είχαν δημιουργηθεί όλες οι κατηγορίες.

Με τα δεδομένα αυτά, οι ώρες θα αυξάνονταν κατά 8 (από 512 αρχικά θα έφταναν τις 520) ενώ τα τμήματα από 41 θα έφταναν τα 66 (αυτή η αύξηση των τμημάτων θα επηρέαζε αρνητικά το πρόβλημα).

Οι καθηγητές του σχολείου είναι περιορισμένοι, ενώ οι αίθουσες επαρκούν για όλες τις διδασκαλίες και στην περίπτωση που δεν επαρκούν, το πρόβλημα επιλύεται με ανθρώπινη παρέμβαση.

Στη συγκεκριμένη σχολική μονάδα, οι ώρες διδασκαλίας για κάθε ζώνη προκαθορίζονται από τη διεύθυνση του σχολείου στις αποτυπωμένες με χρώμα περιοχές του διπλανού πίνακα, με αυστηρούς περιορισμούς.

	1	2	3	4	5	6	7
Δ							
Τ					ΞΓΛ	ΕΠΙΛΟΓΗΣ	
Τ	ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΙ						
Π					ΞΓΛ	ΕΠΙΛΟΓΗΣ	
Π							

Εικόνα 6-1: Προκαθορισμός Ζωνών διδασκαλίας της Γ λυκείου.

Η απόφαση ώστε η ξένη γλώσσα να διδάσκεται την 5^η ώρα και να είναι συνεχόμενη με τα μαθήματα επιλογής (6^η και 7^η ώρα) είναι απόφαση στρατηγικής από την διεύθυνση και λήφθηκε γιατί στα μαθήματα επιλογής, διδάσκεται και δεύτερη ξένη γλώσσα, με αποτέλεσμα, οι περισσότεροι καθηγητές που έρχονται στη σχολική μονάδα να χρησιμοποιούνται τις τρεις τελευταίες ώρες για 2 ημέρες την εβδομάδα. Όσον αφορά στα μαθήματα γενικής παιδείας, οι ώρες που απομένουν να διδαχθούν είναι είτε μόνο την πρώτη ώρα (Τρίτη και Πέμπτη) είτε τις 4 τελευταίες ώρες για τις τρεις ημέρες την εβδομάδα. Επίσης να αναφέρουμε ότι για την διδασκαλία των ξένων γλωσσών απαιτούνται 4 καθηγητές αγγλικής, 2 της γερμανικής και 1 της γαλλικής γλώσσας, με αποτέλεσμα να δεσμεύονται 6 επιπλέον καθηγητές από 6 διαφορετικά σχολεία για την συγκεκριμένη ώρα, εκ των οποίων μόνο οι τρεις από αυτούς θα χρησιμοποιηθούν τις επόμενες ώρες, στα μαθήματα επιλογής (μιας που μόνο ένας καθηγητής διδασκαλίας αγγλικών υπάρχει στη σχολική μονάδα). Στα μαθήματα επιλογής απαιτούνται και επιπλέον καθηγητές (π.χ. σχέδιο) από γυμνάσια, γιατί καθηγητές σχεδίου δεν είναι τοποθετημένοι σε

λύκεια, μιας και δεν υπάρχουν αντίστοιχα μαθήματα, που να δικαιολογούν την ειδικότητα αυτή.

Στη Β λυκείου το πλάνο μαθημάτων αποτυπώνεται στον παρακάτω πίνακα.

	1	2	3	4	5	6	7
Δ							
Τ							
Τ							
Π							
Π							

Εικόνα6-2: Προκαθορισμός ζώνης διδασκαλίας μαθημάτων προσανατολισμού της Β λυκείου.

Παρατηρούμε ότι διατέθηκαν η 2^η και 3^η ώρα καθημερινά για την διδασκαλία των μαθημάτων προσανατολισμού (μαζί με το project). Τελικά τοποθετήθηκαν σε 3 ημέρες (Τρίτη, Τετάρτη και Παρασκευή), και αυτό ήταν επιθυμητό. Όμως και αυτό δεν ήταν τυχαίο. Καθηγητής ο

οποίος διατέθηκε στο σχολείο για 6 ώρες μαθήματος προσανατολισμού ήταν διαθέσιμος μόνο αυτές τις 3 ημέρες, και κατά συνέπεια το αποτέλεσμα ήταν προδιαγραμμαμένο και όχι απόφαση βέλτιστης λύσης ενός αλγόριθμου.

Όσον αφορά την Α λυκείου, η κατάσταση είναι πιο εύκολη, διότι τα μαθήματα των δύο ζωνών διδάσκονται από τον ίδιο καθηγητή και μάλιστα ως μαθήματα γενικής παιδείας. Επίσης, το ίδιο συνέβη και στην ξένη γλώσσα. Οι λιγοστοί μαθητές, οι οποίοι επιθυμούσαν να διδαχθούν διαφορετική ξένη γλώσσα, δεν ήταν αρκετοί για να δημιουργηθεί τμήμα δεύτερης ξένης γλώσσας και έτσι λειτούργησε και το μάθημα αυτό ως γενικής παιδείας με μία μόνο ξένη γλώσσα. Το τελικό αποτέλεσμα ήταν να μην δημιουργηθεί καμία ζώνη μαθημάτων για την τάξη αυτή.

Παρακάτω θα αναφέρουμε και τους χαλαρούς περιορισμούς που προστέθηκαν στο λογισμικό.

A. Όσον αφορά τις ώρες διδασκαλίας των μαθημάτων:

Τα μαθήματα των ζωνών που αναφέρθηκαν παραπάνω έχουν όλα περιορισθεί κατάλληλα. Επιπλέον:

1. Το μάθημα της Ελληνικής Γλώσσας απαγορεύθηκε να διδάσκεται την τελευταία (7^η) ώρα, και
2. Η Φυσική Αγωγή απαγορεύθηκε να διδάσκεται την (1^η ώρα).

Β. Όσον αφορά τους καθηγητές οι περιορισμοί ήταν οι παρακάτω (και φαίνονται στο διπλανό σχήμα):

1. υποχρεωτικοί λόγω μετακίνησης από άλλο σχολείο:

Οι 14 καθηγητές μετακινούνται από διαφορετικά σχολεία, για συγκεκριμένες ημέρες ή ακόμα και ώρες

Αρκετοί από τους περιορισμούς πλεονάζουν. Πρώτα από όλα για τους καθηγητές ξένων γλωσσών της Γ, ένας και μόνο περιορισμός αρκεί. Οι υπόλοιποι καθηγητές θα τοποθετηθούν τις ίδιες ώρες.

2. κάποιες επιθυμίες καθηγητών ανεξάρτητα από το αν υπάρχει υπηρεσιακός λόγος.

Υπάρχουν 6 καθηγητές, οι οποίοι επιθυμούν συγκεκριμένες ώρες να απουσιάζουν από το σχολείο και 1 ακόμα (ο πρώτος) που για υπηρεσιακούς λόγους έχει συγκεκριμένο (προκαθορισμένο) ωράριο διδασκαλίας.

Γ. Οι γενικοί περιορισμοί οι οποίοι σχετίζονται με τα κενά των μαθημάτων ή των καθηγητών και τέθηκαν στο συγκεκριμένο σχολείο φαίνονται παρακάτω:

	Μαθήματα	Εφαρμογή σε:	Περιγραφή	Παράμετροι	Παράμετροι
1	Λογιστ, Γλώσσα	Όλα τα τμήματα	Όχι την ίδια ημέρα.		
2	Όλα	ΝΕ, ΣΝ, ΣΑ, ΣΙ, ΣΚ, ΚΕ, ΛΛ, ΠΣ, ΡΘ, ΓΑ, ΜΑ, ΡΣ, ΒΑ	Μέγιστος αριθμός κενών ανά εβδομάδα	Όλα τα τμήματα	
3	Όλα	Όλοι οι Εκπαιδευτικοί	Μέγιστος αριθμός περιόδων ανά ημέρα	Όλα τα τμήματα	
4	Όλα	ΚΑ, ΚΚ, ΝΕ, ΠΑ, ΣΝ, ΣΑ, ΣΙ, ΣΚ, ΤΛ, ΚΕ, ΛΛ, ΠΣ, ΡΘ, ΓΑ, ΓΒ, ΚΔ, ΜΑ, ΜΔ, ΡΣ, ΚΔ, ΒΑ	Ελάχιστος αριθμός περιόδων ανά ημέρα	Όλα τα τμήματα	
5	Όλα	ΝΕ, ΣΝ, ΣΙ, ΣΚ, ΚΕ, ΛΛ, ΠΣ, ΡΘ, ΓΑ, ΓΒ, ΜΑ, ΣΑ	Μέγιστος αριθμός ημερών με διδασκαλία την ίδια περίοδο	Όλα τα τμήματα	

Εικόνα 6-5: Περιορισμοί για ομάδα καθηγητών στο asc TimeTables.

	ΠΕ01-02	ΣΒ	4	
	ΠΕ02-04	ΠΚ	4	
	ΠΕ02-06	ΣΜ	9	
	ΠΕ04.01-2	ΚΔ	8	
	ΠΕ04.01-3	ΜΜ	6	
	ΠΕ04.02-2	ΓΜ	4	
	ΠΕ04.04-2	ΖΜ	14	
	ΠΕ05-1	ΒΘ	6	
	ΠΕ06-01	ΚΕ	6	
	ΠΕ06-02	ΜΔ	2	
	ΠΕ06-04	ΜΚ	2	
	ΠΕ07-01	ΑΘ	2	
	ΠΕ07-02	ΧΙ	4	
	ΠΕ11-02	ΠΠ	6	

Εικόνα6-3: Περιορισμοί για καθηγητές στο asc TimeTables (α).

	ΠΕ02-01	ΠΣ	5	
	ΠΕ02-02	ΚΚ	20	
	ΠΕ02-05	ΠΑ	20	
	ΠΕ02-10	ΣΚ	18	
	ΠΕ02-11	ΤΛ	20	
	ΠΕ04.02-1	ΓΒ	21	
	ΠΕ06-03	ΜΔ	18	

Εικόνα 6-4 : Περιορισμοί για καθηγητές στο asc TimeTables.

1. Ο πρώτος περιορισμός αναφέρει ότι τα δύο φιλολογικά μαθήματα (λόγω της φύσης τους) δεν πρέπει να διδάσκονται την ίδια ημέρα, για κανένα από τα 14 τμήματα. (εικόνα 8)

2. Ο επόμενος περιορισμός αναφέρει ότι ο μέγιστος αριθμός κενών είναι 3 για συγκεκριμένους 13 καθηγητές.

3. Ο 3^{ος} περιορισμός, δηλώνει ότι δεν πρέπει κανένας καθηγητής να διδάσκει πάνω από 5 ώρες ανά ημέρα.

4. Στη συνέχεια δηλώνεται ότι οι περισσότεροι καθηγητές πρέπει να διδάσκουν τουλάχιστον 3 ώρες κάθε ημέρα.

5. Ο τελευταίος περιορισμός που τέθηκε με τον τρόπο αυτό, προσπαθεί να επιβάλλει διδασκαλία κατά την 7^η, ώρα το πολύ μία ημέρα την εβδομάδα για κάθε εκπαιδευτικό.

Δ. Η επόμενη κατηγορία περιορισμών αφορά 2 καταστάσεις: α) ο πρώτος περιορισμός περιορίζει τα κενά που επιτρέπονται για κάθε καθηγητή συνολικά για όλη την εβδομάδα, και για κάποιους από αυτούς τέθηκε ο αριθμός 2 και β) ο δεύτερος περιορισμός περιορίζει την ημερήσια διδασκαλία να είναι από 3 μέχρι 5 ώρες ανά ημέρα, ο οποίος ζητήθηκε επίσης να εφαρμοσθεί σε μερικούς μόνο καθηγητές.

Στην παρακάτω εικόνα φαίνεται πληθώρα από περιορισμούς.

Εικόνα6-6: Παράθυρο διαλόγου για ορισμό περιορισμού σε ομάδα (δύο) μαθημάτων.

Οι κάρτες πρέπει να είναι στις επιλεγμένες θέσεις (#60)
 Ελάχιστος αριθμός καρτών σε μία περίοδο (#16)
 Ελάχιστος αριθμός καρτών σε μία περίοδο (Επιτρέπεται κενή περίοδος) (#36)
 Μέγιστος αριθμός καρτών (εκπαιδευτικών, τμημάτων κλπ) ταυτόχρονα σε μία θέση (#15)
 Ελάχιστος αριθμός ημερών ανά εβδομάδα (#9)
 Μέγιστος αριθμός ημερών ανά εβδομάδα (#0)
 Ελάχιστος αριθμός περιόδων ανά ημέρα (κενή μέρα είναι OK) (#3)
 Ελάχιστος αριθμός περιόδων ανά ημέρα (#29)
 Μέγιστος αριθμός περιόδων ανά ημέρα (#10)
 Ελάχιστος αριθμός περιόδων ανά εβδομάδα (#12)
 Ελάχιστος αριθμός περιόδων ανά εβδομάδα (κενή εβδομάδα είναι OK) (#38)
 Μέγιστος αριθμός περιόδων ανά εβδομάδα (#13)
 Μέγιστες διδασκαλίες ανά εβδομάδες/εξάμηνα (#31)
 Μέγιστο πλήθος εβδομάδων/εξαμήνων σε μία περίοδο (#57)
 Ελάχιστος αριθμός περιόδων+κενών ανά ημέρα (κενή μέρα είναι OK) (#18)
 Μέγιστος αριθμός περιόδων+κενών ανα ημέρα (#19)
 Μέγιστος αριθμός ελεύθερων ημερών μεταξύ καρτών ανά εβδομάδα (#20)
 Μέγιστος αρ. ελεύθερων εβδομάδων μεταξύ των καρτών ανά έτος (#48)
 Μέγιστος αριθμός κενών ανά ημέρα (#4)
 Μέγιστος αριθμός κενών ανά εβδομάδα (#5)
 Μέγιστος αριθμός κενών ανά εβδομάδα (Israel) (#28)
 Μέγιστος αριθμός συνεχόμενων ημερών (#23)
 Μέγιστος αριθμός συνεχόμενων περιόδων (#6)
 Το πολύ 3 συνεχόμενες περίοδοι, ή 2 διπλές (#55)
 Μέγιστο πλήθος συνεχόμενων διδασκαλιών (#56)
 Μέγιστος αριθμός ημερών με διδασκαλία την ίδια περίοδο (#7)
 Μέγιστος αριθμός ημερών με διδασκαλία την ίδια περίοδο (για όλες τις εβδομάδες) (#8)
 Μέγιστος Αριθμός διαφορετικών περιόδων(ωρών) ανα εβδομάδα (#17)
 Μέγιστος αριθμός αιθουσών σε μία περίοδο (#39)

Εικόνα6-7: Πτυσσόμενη λίστα με περιορισμούς. Όπως φαίνεται από την κατακόρυφη γραμμή ολίσθησης, φαίνονται σχεδόν μόνο οι μισοί περιορισμοί.

6.2 Χρήση του asc TimeTables και εκτέλεση του προγράμματος.

Το asc TimeTables (όπως κάθε σοβαρό λογισμικό) πριν αρχίσει την προσπάθεια για τη δημιουργία ωρολογίου προγράμματος, ελέγχει εάν τα δεδομένα που έχουν εισαχθεί είναι συμβατά και αναζητείται να βρεθεί πιθανή ασυμβατότητα, σε οποιοδήποτε σύνολο

Στατιστικά		Όνομα	Κενά	Ερ...	Κόπωση	Ωρες ανά ημέρα
Εκπαιδευτικοί:	39	ΠΕ02-03	3	0	5	5/3/3/3/4
Ασυμπλήρωτοι	0	ΠΕ02-05	3	0	5	4/3/4/5/4
Τμήματα:	14	ΠΕ02-06	3	0	3	0/0/0/4/5
Κάρτες:	486	ΠΕ02-07	3	0	4	4/4/4/3/3
Εκκενρείς κάρτες	0	ΠΕ02-08	3	0	4	5/3/5/4/3
Κάρτες σε λάθος θέση:	0	ΠΕ04.01-1	3	0	4	5/2/3/4/6
Συνολικός αριθμός κενών:	82	ΠΕ04.04-2	3	0	3	5/2/5/0/2
Μ.Ο. κενών	2,102564	ΠΕ01-01	4	0	4	3/3/5/3/4
Μέγιστα κενά σε	7	ΠΕ02-02	4	0	4	3/3/5/4/5
Εκπαιδευτικοί με το μέγ. αριθ. κενών:	1	ΠΕ03-03	4	0	4	5/3/5/3/4
Μέγιστος αριθμός κενών:	2	ΠΕ06-03	4	0	5	5/3/3/4/3
Εκπαιδευτικοί με το μέγιστο	11	ΠΕ12	4	0	4	2/4/3/3/4
		ΠΕ02-10	5	0	4	5/3/4/3/3
		ΠΕ02-11	6	0	5	3/4/4/4/5
		ΠΕ04.04-1	6	0	5	5/2/6/3/5
		ΠΕ19	7	0	3	4/3/4/2/3

Δημιουργία: 01: 09, Saturday, September 21
 Τελευταία ενημέρωση: 11: 29, Tuesday, June 06

Εικόνα 6-8: Παράθυρο διαλόγου, μετά την εύρεση λύσης, στο οποίο παρουσιάζονται στατιστικά στοιχεία σχετικά με περιορισμούς για τους καθηγητές. asc TimeTables.

(καθηγητών, διδακτικών ωρών, τμημάτων, αιθουσών, κ.α.) η οποία θα οδηγούσε σε αδύνατη κατάσταση.

Το πρόγραμμα εκτελέστηκε με την επιλογή να επιτρέψει να «χαλαρώσουν» περιορισμοί, εάν δεν είναι δυνατόν να βρεθεί βέλτιστη λύση, ώστε να δημιουργηθεί ωρολόγιο πρόγραμμα με κόστος την χαλάρωση περιορισμών.

Η εκτέλεση της έκδοσης 2017, με τους παραπάνω περιορισμούς διήρκεσε 4 λεπτά και λίγα δευτερόλεπτα, σε ένα υπολογιστικό σύστημα με επεξεργαστή Dual Core 2, μνήμη 2 GB, και λειτουργικό σύστημα windows XP, sp3.

6.3 Αποτελέσματα εκτέλεσης του λογισμικού.

Το λογισμικό μετά την εισαγωγή των δεδομένων, εκτελέστηκε και έδωσε λύση σε εύλογο χρονικό

διάστημα. Η

αξιολόγηση του

ωρολογίου

προγράμματος γίνεται

από το ίδιο το λογισμικό

με διάφορους τρόπους,

αλλά χωρίς να δίνει την

τιμή της αντικειμενικής

συνάρτησης. Η

αξιολόγηση αναφέρει

πόσα συνολικά κενά

υπάρχουν, αθροιστικά

για όλους τους

καθηγητές, και αναλύονται ανά καθηγητή. Επίσης, αναφέρεται η κόπωση ανά καθηγητή και ο μέγιστος αριθμός κενών που εμφανίστηκε σε κάποιον καθηγητή. Στη συγκεκριμένη περίπτωση οι 7 ώρες κενά σε μία εβδομάδα αναφέρεται σε υποδιευθυντή ο οποίος δεν έχει περιορισμό λόγω της παραμονής του στο σχολείο καθ' όλη τη διάρκεια λειτουργίας του. Επίσης οι κάρτες (events) που τοποθετήθηκαν είναι 486 γιατί οι υπόλοιπες 26 είχαν ήδη τοποθετηθεί (και κλειδωθεί ώστε να μην αλλάξουν) από τον χρήστη του προγράμματος, και

Στατιστικά		Όνομα	Κενά	Ερ...	Κόπωση	Ώρες ανά ημέρα
Εκπαιδευτικοί:	39	ΠΕ02-10	3	0	4	5/3/4/3/3
Ασυμπλήρωτοι	0	ΠΕ03-02	3	0	5	5/3/5/4/4
Τμήματα:	14	ΠΕ04.01-1	3	0	4	3/3/5/3/6
Κάρτες:	486	ΠΕ04.01-2	3	0	4	0/4/0/0/4
Εκκρεμείς κάρτες	0	ΠΕ11-01	3	0	4	4/3/5/2/4
Κάρτες σε λάθος θέση:	0	ΠΕ12	3	0	4	3/2/3/4/4
		ΠΕ13	3	0	5	5/5/5/3/3
Συνολικός αριθμός κενών:	84	ΠΕ01-01	4	0	3	4/3/5/3/3
Μ.Ο. κενών	2.153846	ΠΕ02-02	4	0	4	3/4/5/3/5
Μέγιστα κενά σε	4	ΠΕ02-06	4	0	3	0/0/0/4/5
Εκπαιδευτικοί με το μέγ. αριθ. κενών:	9	ΠΕ02-11	4	0	5	5/3/4/3/5
Μέγιστος αριθμός κενών:	2	ΠΕ03-03	4	0	3	5/3/5/4/3
Εκπαιδευτικοί με το μέγιστο	14	ΠΕ04.04-1	4	0	5	5/3/5/3/5
		ΠΕ04.04-2	4	0	3	5/1/5/0/3
		ΠΕ06-03	4	0	4	4/3/3/5/3
		ΠΕ19	4	0	5	2/2/3/5/4
Δημιουργία	01: 09, Saturday, September 21					
Τελευταία ενημέρωση:	11: 42, Tuesday, June 06					

Εικόνα 6-9: Στατιστικά δεύτερης επιτυχημένης προσπάθειας δημιουργίας ωρολογίου προγράμματος.

αφορούσαν μαθήματα επιλογής και ξένων γλωσσών, σύμφωνα με όσα αναφέρθηκαν παραπάνω.

Το πρόγραμμα επανεκτελέστηκε και έδωσε παρόμοια αποτελέσματα, με ελάχιστες διαφορές. Κάθε φορά που επανεκτελείται το πρόγραμμα αρχίζει με διαφορετική αρχική λύση και έτσι πιθανότατα θα δίνει συνεχώς διαφορετική λύση.

Στη εικόνα 6-10, παρουσιάζονται οι περιορισμοί οι οποίοι χαλάρωσαν για την δημιουργία ωρολογίου προγράμματος οι οποίοι κρίνονται ως αρκετοί και οι βασικότεροι θα αναλυθούν σύντομα παρακάτω. Ο πρώτος περιορισμός που χαλάρωσε αφορά το μάθημα των αρχαίων προσανατολισμού, το οποίο διδάσκεται 3 ώρες την

Αντικείμενο	Περιγραφή
4x	Κατανομή καρτών στην εβδομάδα
(ΣΚ) ΠΕ02 Σ...	Μέγιστος αριθμός ημερών με διδασκαλία την ίδια περίοδο
7x	Μέγιστος αριθμός κενών ανά εβδομάδα
(ΣΚ) ΠΕ02 Σ...	Μέγιστος αριθμός κενών ανά εβδομάδα: 5>4
(ΣΚ) ΠΕ02 Σ...	Μέγιστος αριθμός κενών ανά εβδομάδα: 5>3
(ΤΛ) ΠΕ02 Τ...	Μέγιστος αριθμός κενών ανά εβδομάδα: 6>4
(ΠΣ) ΠΕ03 Π...	Μέγιστος αριθμός κενών ανά εβδομάδα: 4>3
(ΜΑ) ΠΕ04....	Μέγιστος αριθμός κενών ανά εβδομάδα: 6>3
(ΜΑ) ΠΕ04....	Μέγιστος αριθμός κενών ανά εβδομάδα: 6>4
(ΒΝ) ΠΕ19 Β...	Μέγιστος αριθμός κενών ανά εβδομάδα: 7>4
(ΓΑ) ΠΕ04.0...	Ελάχιστος αριθμός περιόδων ανά ημέρα: 2<3
(ΓΑ) ΠΕ04.0...	Μέγιστος αριθμός περιόδων ανά ημέρα: 6>5
(ΓΒ) ΠΕ04.0...	Μέγιστος αριθμός ημερών με διδασκαλία την ίδια περίοδο
(ΜΑ) ΠΕ04....	Μέγιστος αριθμός περιόδων ανά ημέρα: 6>5
(ΜΑ) ΠΕ04....	Ελάχιστος αριθμός περιόδων ανά ημέρα: 2<3
(ΜΑ) ΠΕ04....	Μέγιστος αριθμός ημερών με διδασκαλία την ίδια περίοδο
(ΜΔ) ΠΕ06 ...	Μέγιστος αριθμός συνεχόμενων περιόδων: 5>4
(ΚΔ) ΠΕ11 Κ...	Ελάχιστος αριθμός περιόδων ανά ημέρα, Μέγιστος αριθμός π...
(ΚΔ) ΠΕ11 Κ...	Ελάχιστος αριθμός περιόδων ανά ημέρα: 2<3
(ΒΑ) ΠΕ13 Β...	Ελάχιστος αριθμός περιόδων ανά ημέρα, Μέγιστος αριθμός π...
(ΒΑ) ΠΕ13 Β...	Μέγιστος αριθμός συνεχόμενων περιόδων: 6>5
(ΒΑ) ΠΕ13 Β...	Μέγιστος αριθμός περιόδων ανά ημέρα: 6>5

Εικόνα 6-10: Περιορισμοί που χαλάρωσαν από την εύρεση αποδεκτής λύση

εβδομάδα, τις 3 ημέρες που διδάσκονται τα μαθήματα προσανατολισμού. Το 1^ο τμήμα προσανατολισμού προέρχεται από τα 4 τμήματα γενικής παιδείας και διδάσκεται το μάθημα των Αρχαίων Ελληνικών σε ένα μονόωρο και ένα δίωρο μάθημα σε 2 ημέρες αντί για 3 μονόωρα σε κάθε μία από 3 ημέρες. Ο επόμενος περιορισμός αναφέρει ότι ο καθηγητής ΣΚ διδάσκει 2 φορές την εβδομάδα την 7^η ώρα, ενώ έχει ζητηθεί για όλους να μην ξεπερνάει τη μία διδασκαλία σε 7^η ώρα. Στη συνέχεια παρατηρούμε ότι 7 καθηγητές θα έχουν περισσότερα κενά την εβδομάδα από το ελάχιστο που ζήτησαν. Αμέσως παρακάτω χαλαρώνουν 2 περιορισμοί για τον καθηγητή ΓΑ. Αυτοί οι περιορισμοί αφορούν τις ώρες διδασκαλίας ανά ημέρα, οι οποίες κάποια ημέρα είναι 2 (με ελάχιστο ζητούμενο 3) και κάποια άλλη ημέρα είναι 6 (με ζητούμενο μέγιστο 5). Το ίδιο πρόβλημα παρουσιάζεται και στον καθηγητή ΜΑ καθώς και σε άλλους καθηγητές παρακάτω.

Η πολυπλοκότητα του προβλήματος είναι μεγάλη και έτσι δεν είναι δυνατό να προβλέψουμε τους λόγους, για τους οποίους εμφανίζονται τα προβλήματα αυτά. Από προσωπική ενασχόληση και εμπειρία ετών, μπορεί να αναφερθεί ότι βασική αιτία χαλάρωσης των περιορισμών συνιστά ο τρόπος της κατανομής των μαθημάτων μεταξύ των καθηγητών της ίδιας ειδικότητας (δεδομένων βέβαια των προβλημάτων που αναφέρθηκαν). Οι καθηγητές κάθε ειδικότητας συναποφασίζουν για την κατανομή διδασκαλίας των μαθημάτων που αφορούν την ίδια ειδικότητα. Θα βοηθούσε πάρα πολύ να επέλεγαν οι καθηγητές μαθήματα διδασκαλίας από κάθε ζώνη (γενικής παιδείας, προσανατολισμού, κ.λ.π) και όχι μόνο μαθήματα προσανατολισμού ή μόνο γενικής παιδείας.

Κάποια λογισμικά (όπως και το ascTimeTable), μετά την δημιουργία της λύσης, αναφέρουν συμβουλές που θα βελτίωναν τη λύση. Αυτό βέβαια σημαίνει ότι οι καθηγητές θα έπρεπε να αλλάξουν τις αναθέσεις που ανέλαβαν, χωρίς αυτό να σημαίνει ότι το θέλουν ή ότι το επιθυμούν και θα γίνει.

6.4 Άλλες δυνατότητες του asc TimeTables.

Οι δυνατότητες παρουσίασης και εκτύπωσης του πίνακα ωρολογίου προγράμματος από το λογισμικό είναι πολλές και παραμετροποιήσιμες. Επίσης, εισάγει δεδομένα προβλημάτων και εξάγει λύσεις σε πολλές διαφορετικές μορφοποιήσεις για διάφορα κράτη του κόσμου. Ακόμα έχει την δυνατότητα να εισάγει δεδομένα προβλημάτων από το δωρεάν λογισμικό FET. Η μεταφορά δεδομένων πραγματοποιείται σε XML (όχι όμως τη XHSTT).

Το εμπορικό αυτό λογισμικό έχει ένα πάρα πολύ καλό γραφικό περιβάλλον, στο οποίο εάν ο χρήστης κατανοεί το πρόβλημα που έχει να λύσει, τότε το ascTimetables του δίνει την δυνατότητα σε ελάχιστο χρόνο να έχει καταχωρήσει το πρόβλημα σωστά και κατά συνέπεια να μπορέσει με το γρήγορο και ικανοποιητικό αλγόριθμό του να έχει γρήγορα μια πολύ καλή λύση, ακόμα και για μεγάλα σχολεία.

Από την άλλη σκοπιά όμως πρέπει να τονίσουμε ότι η δομή του προβλήματος των λυκείων της Ελλάδας είναι ιδιόμορφη και περιορίζει πάρα πολύ τον χώρο αναζήτησης. Το αποτέλεσμα είναι με ελάχιστους περιορισμούς να έχουμε ταχύτατα μια πολύ καλή λύση, ενώ στην περίπτωση που θέσουμε πολλούς περιορισμούς να μην έχουμε καθόλου λύση. Στην περίπτωση που δεν υπάρχει λύση, το ascTimetables αναφέρει τους περιορισμούς που θεωρεί υπαίτιους για την μη εύρεση λύσης.

ΚΕΦΑΛΑΙΟ VII

ΝΕΑ ΜΟΡΦΟΠΟΙΗΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Η μορφοποίηση μαθηματικού προγραμματισμού του προβλήματος αναφέρθηκε στο κεφάλαιο 2. Όμως η μορφοποίηση αυτή έγινε πριν την χρήση του XHSTT. Η εμφάνιση της XML έκδοσης περιγραφής προβλημάτων γενικεύει το πρόβλημα. Ο G. Fonseca το 2017, επαναπροσδιορίζει το μαθηματικό μοντέλο του προβλήματος, με νέα μορφοποίηση [Fon17], ώστε να εκμεταλλεύεται πλήρως την XHSTT, την οποία ονομάζει *F2* (Formulation 2) και με τον τρόπο αυτό είναι δυνατόν να περιγράψουμε προβλήματα χρονοπρογραμματισμού δευτεροβάθμιας εκπαίδευσης, τριτοβάθμιας εκπαίδευσης, χρονοπρογραμματισμού εξετάσεων και άλλων προβλημάτων [Fon17]. Αυτό γίνεται λόγω της γενίκευσης που ακολουθήθηκε, ώστε να είναι δυνατή η περιγραφή του προβλήματος στη γενικότερη μορφή του. Οι γενικεύσεις αυτές από τη μια γενίκευσαν το πρόβλημα και έδωσαν την δυνατότητα χρήσης νέων περιορισμών, ενώ από την άλλη φυσικά το γενικευμένο πρόβλημα είναι πιο σύνθετο στη λύση του.

Ο G. Fonseca στη διδακτορική του διατριβή [Fon17] καθορίζει πλήρως πλέον το πρόβλημα με πιο γενικευμένο μαθηματικό μοντέλο, στο οποίο αλλάζουν αρκετές παράμετροι.

7.1 Το περιβάλλον επικοινωνίας με τα λογισμικά.

Η επικοινωνία με το κάθε λογισμικό (γενικά) γίνεται με δύο τρόπους: σε περιβάλλον εντολών (Command Line Interface) και σε γραφικό περιβάλλον (Graphical User Interface).

Ουσιαστικά η επίλυση ενός προβλήματος δεν απαιτεί γραφικά. Μπορούμε να έχουμε τα δεδομένα σε κάποιο αρχείο κειμένου, το λογισμικό να επιλύσει το πρόβλημα και να μας δώσει την απάντηση σε αρχείο κειμένου. Το γραφικό περιβάλλον βοηθάει στην είσοδο δεδομένων και στην οπτικοποίηση των αποτελεσμάτων.

7.2 Είσοδος δεδομένων.

Τα δεδομένα για το πρόβλημα του χρονοπρογραμματισμού είναι πολλά, και οι σχέσεις που τα συνδέουν στον ορισμό του προβλήματος είναι πολύπλοκες. Από τους

περιορισμούς, μόνο οι χαλαροί θα καταχωρισθούν σαν δεδομένα στο λογισμικό, ενώ οι αυστηροί είναι ενσωματωμένοι στο λογισμικό γιατί πρέπει να ικανοποιούνται εξ ορισμού.

Τα λογισμικά των ερευνητών δέχονται τα δεδομένα τους σε μορφή XML αρχείων (π.χ. XHSTT) ενώ κάποια από αυτά δέχονται απλά αρχεία κειμένου. Όμως, όπως ο κοινός χρήστης ενός εμπορικού λογισμικού (ο οποίος ουσιαστικά απαιτεί ένα εύκολο τρόπο εισαγωγής), έτσι και ο ερευνητής θα ήθελε ένα εύκολο τρόπο εισαγωγής στοιχείων, τουλάχιστον για οικονομία χρόνου. Εξάλλου οι υπολογιστές κατάφεραν να γίνουν εμπορικοί από την στιγμή που αναπτύχθηκε και εξελίχθηκε το γραφικό περιβάλλον επικοινωνίας ανθρώπου – υπολογιστή (Human Computer Interaction).

Τα δεδομένα μπορεί να πληκτρολογηθούν κατευθείαν σε ένα συντάκτη (editor), σε αρχείο κειμένου, ή σε XML αρχείο. Προφανώς ένα XML αρχείο είναι από μόνο του περιγραφικό, σε αντίθεση με ένα αρχείο κειμένου, στο οποίο πρέπει να γνωρίζει κάποιος (είτε το λογισμικό, είτε ο άνθρωπος) μια προσυμφωνημένη γραμμογράφηση. Είναι γενικά αποδεκτό σήμερα ότι τα δεδομένα ανταλλάσσονται με μορφή XML, η οποία είναι κατανοητή και από τα λογισμικά και από τον άνθρωπο.

Ανάπτυξη στο θέμα έχει γίνει περιορισμένα [Fon17, FDH14] και αναμένεται να έχει αρκετή πρόοδο στο κοντινό μέλλον. Το περιβάλλον ανάπτυξης που έχει ξεκινήσει είναι σε περιβάλλον WEB, ενώ σε περιβάλλον προσωπικού υπολογιστή με γραφικό περιβάλλον δεν υπάρχει καμία αναφορά.

Τα εμπορικά λογισμικά, τα οποία χρησιμοποιούνται ευρέως, συνήθως χρησιμοποιούν (για αποθήκευση) ένα δικό τους XML αρχείο, το οποίο όμως δεν συμφωνεί με κανένα λογισμικό άλλης εταιρείας. Αυτό συμβαίνει και για τα δεδομένα του προβλήματος καθώς επίσης και για τους περιορισμούς και για τις λύσεις. Έτσι δεν είναι δυνατό να υπάρχει «επικοινωνία» μεταξύ των λογισμικών για τη μεταφορά κάποιου προβλήματος. Επίσης, τα διαδεδομένα εμπορικά προγράμματα έχουν κατά κανόνα ικανοποιητικό, γραφικό περιβάλλον εισαγωγής δεδομένων.

7.3 Η έξοδος (εμφάνιση στην οθόνη και στον εκτυπωτή).

Ο σκοπός του χρονοπρογραμματισμού είναι η δημιουργία ενός ωρολογίου προγράμματος. Η λύση δίνεται αρχικά στις δομές δεδομένων ενός προγράμματος HY και πρέπει να εξαχθεί με τη μορφή πίνακα, με όλες τις διδασκαλίες για όλους τους καθηγητές και ώρες τις εβδομάδας.

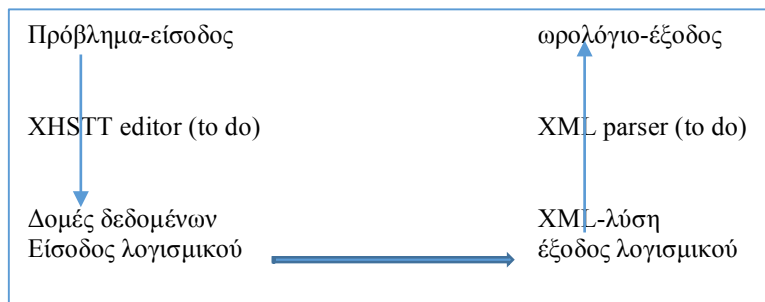
Τα λογισμικά, όπως στην είσοδο έτσι και στην έξοδο, μπορούν να εξάγουν το πρόγραμμα σε αρχείο κειμένου. Το αρχείο αυτό μπορεί να είναι είτε απλό αρχείο με προκαθορισμένη γραμμογράφιση, είτε αρχείο XML. Στη συνέχεια όμως, τουλάχιστον για πρακτικούς σκοπούς, το ωρολόγιο πρόγραμμα πρέπει να μετατραπεί σε πίνακα εμφανίσιμο με τυποποιημένες προδιαγραφές και να είναι διαθέσιμο σε επιπλέον μορφές, όπως pdf, xls (λογιστικού φύλλου) ή ακόμα και ics (ημερολογίου).

Η βασική μορφή εξόδου θα είναι πιθανότατα μια XML μορφή (είτε η XHSTT είτε κάποια μορφή του δημιουργού του λογισμικού). Οι δημιουργοί των λογισμικών έχουν φροντίσει και έχουν απεικονίσει τη λύση τους σε πίνακες σε γραφικό περιβάλλον και σε άλλες μορφές, όπως αρχεία pdf, xls κ.α.

Τα λογισμικά όμως που χρησιμοποιούν την XHSTT μορφή, λαμβάνουν το XML αρχείο με την περιγραφή του προβλήματος και επιστρέφουν το αρχείο, αφού του έχουν ενσωματώσει τη λύση (και το κόστος της λύσης). Ένα αρχείο για γυμνάσιο με 9 τμήματα συνολικά σε τρεις τάξεις (3 τμήματα ανά τάξη) δημιουργεί αρχείο του προβλήματος μεγέθους 255 KB και αρχείο εξόδου (που περιέχει και τη λύση) μεγέθους 308 KB. Η λύση σε XHSTT είναι κατανοητή, αλλά εύλογο είναι ότι θα πρέπει να παρουσιαστεί σε μορφή πίνακα πλήρους ωρολογίου προγράμματος. Λογισμικό για αυτό το βήμα δεν υπάρχει διαθέσιμο. Μέχρι σήμερα μόνο τα ερευνητικά ιδρύματα ασχολούνται με την XHSTT και δεν έχουν διαθέσει αντίστοιχο λογισμικό, ενώ τα εμπορικά προγράμματα παρέχουν ανεξάρτητες λύσεις για το πρόβλημα.

7.4 Μελέτη επίλυσης του προβλήματος οπτικοποίησης της λύσης.

Όπως αναφέρθηκε, η περιγραφή του προβλήματος με χρήση XHSTT μπορεί να συμπεριλάβει τη λύση του προβλήματος (σε XML μορφή). Για να δούμε τον ζητούμενο πίνακα εξόδου, είτε στην οθόνη είτε σε αρχείο, απαιτείται να γραφεί επιπλέον λογισμικό το οποίο θα έχει ως είσοδο το XHSTT αρχείο και θα εμφανίζει τον πίνακα του ωρολογίου προγράμματος. Αυτό το πρόγραμμα θα είναι ουσιαστικά ένας XML parser. Η είσοδος του είναι ένα δένδρο XML, το οποίο ο parser πρέπει να διατρέξει και να δημιουργήσει τον πίνακα.



Σχήμα 7-1: Διαδικασία επίλυσης προβλήματος.

Η απεικόνιση της λύσης μπορεί να γίνει με διάφορες τεχνολογίες. Σε κάθε περίπτωση όμως χρειάζεται η συγγραφή ενός XML parser. Οι διάφορες λύσεις που θα εμφανισθούν θα έχουν ως είσοδο το συγκεκριμένο αρχείο XHSTT, άρα αναγκαστικά θα περιέχουν τμήμα προγράμματος, το οποίο θα διατρέχει το XML δένδρο, θα συλλέγει τις απαραίτητες πληροφορίες και θα εξάγει τα δεδομένα με διάφορες μορφές.

A) Η λύση με XSLT περιγραφική γλώσσα XML.

Η υλοποίηση μπορεί να γίνει σε οποιαδήποτε γλώσσα προγραμματισμού γενικής χρήσης, αλλά θα μπορούσε επίσης να χρησιμοποιηθεί και η XSLT. Η γλώσσα αυτή είναι μια περιγραφική γλώσσα XML, η οποία όμως δεν χρησιμοποιεί δομές δεδομένων. Στο πλαίσιο της συγκεκριμένης έρευνας δημιουργήθηκε ένα XSLT πρόγραμμα (παράρτημα Α), το οποίο έχει ακριβώς αυτή τη δυνατότητα, να παρουσιάζει δηλαδή την XML λύση ενός προβλήματος στον χώρο εργασίας ενός φυλλομετρητή. Το πρόγραμμα λειτουργεί σχετικά αργά λόγω του ότι η XSLT είναι μια script γλώσσα, η οποία δεν έχει δομές δεδομένων και για αυτό έχει ελάχιστες δυνατότητες, ώστε να χρησιμοποιηθεί για την παρουσίαση ενός ωρολογίου προγράμματος με κάποιες επιπλέον δυνατότητες, πέρα από ένα βασικό πίνακα δύο διαστάσεων με επικεφαλίδες.

B) Η λύση calendar για εμφάνιση σε ημερολόγιο.

Το ημερολόγιο παρουσιάζει γεγονότα, έτσι θα είναι μία πολύ καλή λύση, ο parser να εξάγει για κάθε διδασκαλία και ένα γεγονός calendar (αρχεία μορφής ημερολογίου) και να το στέλνει για παρουσίαση σε διάφορα ημερολόγια. Τα ημερολόγια που μπορούν να χρησιμοποιηθούν για παρουσίαση μπορεί να είναι το calendar από το προσωπικό κινητό ή τον προσωπικό υπολογιστή του κάθε διδάσκοντα, μέχρι και το calendar του κάθε τμήματος, το calendar του σχολείου στον υπολογιστή του διευθυντή, καθώς επίσης και του κάθε μαθητή. Ακόμα τα γεγονότα αυτά θα μπορούσαν να στέλνονται και σε ένα web calendar του σχολείου, για να είναι διαθέσιμα στην ιστοσελίδα του σχολείου.

Στη περίπτωση που αναφερόμαστε σε ημερολόγιο συγκεκριμένων γεγονότων, θα πρέπει να γνωρίζουμε και την ακριβή ημερομηνία έναρξης του προγράμματος, ώστε τα γεγονότα να δημιουργούνται για πραγματικές ημερομηνίες.

Η υλοποίηση θα γίνεται από τον parser, ο οποίος θα μετατρέπει κάθε γεγονός (event) που περιέχεται στη λύση ενός XHSTT αρχείου, σε αρχείο τύπου ics το οποίο είναι γνωστό στις εφαρμογές calendar και αυτόματα μπορεί να το εμφανίσει κατάλληλα.

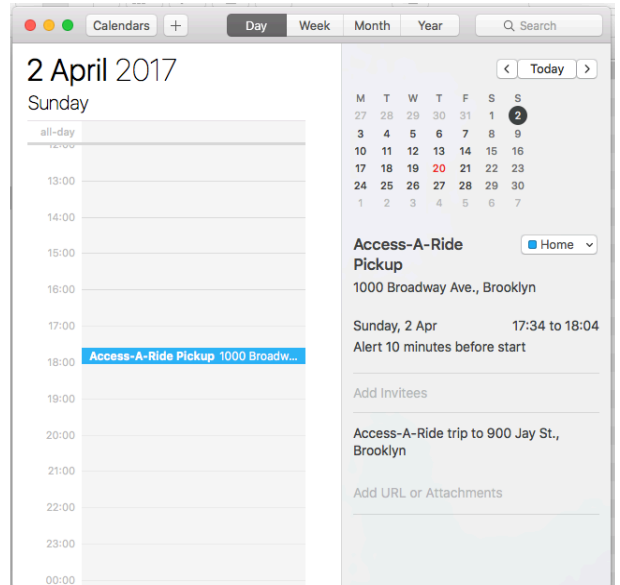
```
BEGIN:VCALENDAR
VERSION:2.0
CALSCALE:GREGORIAN
BEGIN:VEVENT
SUMMARY:Access-A-Ride Pickup
DTSTART;TZID=America/New_York:20170402T103400
DTEND;TZID=America/New_York:20170402T110400
LOCATION:1000 Broadway Ave.\, Brooklyn
DESCRIPTION: Access-A-Ride trip to 900 Jay St.\, Brooklyn
STATUS:CONFIRMED
SEQUENCE:3
BEGIN:VALARM
TRIGGER:-PT10M
DESCRIPTION:Pickup Reminder
ACTION:DISPLAY
END:VALARM
END:VEVENT
BEGIN:VEVENT
SUMMARY:Access-A-Ride Pickup
DTSTART;TZID=America/New_York:20170404T200000
DTEND;TZID=America/New_York:20170405T203000
LOCATION:900 Jay St.\, Brooklyn
DESCRIPTION: Access-A-Ride trip to 1000 Broadway Ave.\, Brooklyn
STATUS:CONFIRMED
SEQUENCE:3
BEGIN:VALARM
TRIGGER:-PT10M
DESCRIPTION:Pickup Reminder
ACTION:DISPLAY
END:VALARM
END:VEVENT
END:VCALENDAR
```

Πίνακας 7-1: Παράδειγμα αρχείου τύπου ics για συγκεκριμένο γεγονός που παρουσιάζεται σε ημερολόγιο

Στην διπλανή εικόνα (7-1) παρουσιάζεται το αποτέλεσμα που έχει στο ημερολόγιο iCal σε Mac OS X, 10.11.6 το αρχείο του πίνακα 7-1.

C) WEB BASED εμφάνιση.

Μια τέτοια εφαρμογή μπορεί να υλοποιηθεί σε κάθε γλώσσα προγραμματισμού, η οποία έχει πρόσβαση σε ιστοσελίδα. Παραδείγματα γλωσσών που θα μπορούσαν να διαβάσουν το XHSTT αρχείο και να εμφανίσουν το ωρολόγιο πρόγραμμα είναι η Java (applet), Javascript, Php, python, κ.α. Για όλες όμως, τα δεδομένα θα προκύπτουν από κάποιο parser. Η έξοδος θα είναι HTML η οποία μπορεί να εμφανιστεί σε οποιοδήποτε φυλλομετρητή. Η υλοποίηση μπορεί να γίνεται είτε στον server είτε στον client φυλλομετρητή.



Εικόνα 7-1: Αποτέλεσμα εισαγωγής ενός γεγονότος (του πίνακα 7.1) στο iCal.

ΚΕΦΑΛΑΙΟ VIII

ΛΟΓΙΣΜΙΚΑ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΓΙΑ ΣΧΟΛΙΚΕΣ ΜΟΝΑΔΕΣ

8.1 Γλώσσες επικοινωνίας XML.

Η XML (eXtensible Markup Language) είναι μια γλώσσα σήμανσης (markup language), την οποία μπορεί να χρησιμοποιήσει οποιοσδήποτε με τις δικές του ετικέτες. Η γλώσσα δημιουργήθηκε από το World Wide Web Consortium, για να ξεπεράσει τους περιορισμούς της HTML γλώσσας, η οποία χρησιμοποιείται σε ιστοσελίδες, και όσα αναφέρονται εδώ βασίζονται στις ανακοινώσεις του παραπάνω consortium.

Η HTML γλώσσα απλά ενημερώνει τον φυλλομετρητή πώς να εμφανίσει κάτι (μέγεθος γραμμάτων, έντονα, κ.λ.π.), ενώ η XML επιτρέπει στον χρήστη να χρησιμοποιεί δικές του ετικέτες, ώστε να δίνει νόημα στα δεδομένα και να απευθύνεται με αυτόν τον τρόπο και προς τον άνθρωπο. Οι μηχανές (HY) έτσι κι αλλιώς μπορούν εύκολα να επεξεργαστούν τις γλώσσες με ετικέτες (Markup Languages).

Η XML δημιουργήθηκε για να περιγράψει δομές για ανταλλαγή δεδομένων μεταξύ ανόμοιων συστημάτων και επιπλέον έχει ενσωματωμένο μηχανισμό για την επικύρωση των δεδομένων, ώστε να εγγυάται την έγκυρη δομή τους.

Η XML χρησιμοποιεί τρεις όρους για να περιγράψει τμήματα ενός XML κειμένου: tag, elements και attributes. Δείγμα ενός XML κειμένου είναι το παρακάτω:

```
<address>
  <name>
    <title>Mr.</title>
    <first-name>Robert</first-name>
    <last-name>De Niro</last-name>
  </name>
  <street>1631 Main Street</street>
  <city state="DV">Anycity</city>
  <postal-code>63577</postal-code>
</address>
```

- **tag** είναι το κείμενο μεταξύ των δύο αγκυλών (< και >). Υπάρχουν τα αρχικά tag (<name>) και τα tag τέλους (</name>).
- τα **στοιχεία (elements)** είναι τα αρχικά tag, τα tag τέλους, και οτιδήποτε υπάρχει μεταξύ τους. Στο παραπάνω παράδειγμα, το element <name> περιέχει τρία element παιδιά, τα <title>, <first-name> και <last-name>. Στην ουσία είναι ετικέτες, περιέχουν τιμές και είναι οργανωμένα σε δένδρο.

- οι **ιδιότητες (attribute)** είναι ένα ζεύγος όνομα-τιμή, μέσα σε ένα αρχικό tag. Στο ίδιο παράδειγμα, το state είναι η ιδιότητα του στοιχείου city. Οι ιδιότητες, δίνουν νόημα στα δεδομένα και μας βοηθούν για μια πιο κατανοητή περιγραφή. Με τον τρόπο αυτό τα δεδομένα ουσιαστικά γίνονται αυτοπεριγραφικά. Ο σκοπός των δεδομένων είναι, ούτως ή άλλως, να περιέχουν όσο περισσότερες πληροφορίες γίνεται.

Σε XML κείμενα υπάρχουν ακόμα τα:

- comments (σχόλια), τα οποία μπορεί να υπάρχουν οπουδήποτε στο κείμενο. Σχόλια αρχίζουν με `<!--` και τελειώνουν με `-->`.
`<!--comments, σχόλια -->`
- Processing instructions (εκτελέσιμες διεργασίες): μια εκτελέσιμη διεργασία προσδιορίζεται, για να εκτελεστεί κάποιος συγκεκριμένος εκτελέσιμος κώδικας.
`<?cocoon-process type="sql"?>`
- Entities - οντότητες, οι οποίες μας επιτρέπουν να χρησιμοποιούμε στο κείμενό μας κάποιο ορισμένο αλφαριθμητικό. Για παράδειγμα η οντότητα `<!ENTITY euro "€">` μας επιτρέπει να χρησιμοποιούμε παντού στο κείμενό μας το euro αντί του #8364 που είναι το σύμβολο του ευρώ.

8.1.1 Κανόνες ονομασίας.

Η XML χρησιμοποιείται σε διάφορα πεδία με πάρα πολλές εφαρμογές. Απλά θα αναφερθούμε σε κάποια πεδία από αυτά:

- Η MathML χρησιμοποιείται για μαθηματικούς τύπους. Με τον τρόπο αυτό μπορούν να περιγραφούν πολύπλοκοι μαθηματικοί τύποι και συναρτήσεις με ρίζες, παραγώγους, ολοκληρώματα κ.α., να μεταφέρονται μέσω διαδικτύου και να γίνονται «κατανοητοί» από διάφορες εφαρμογές και ταυτόχρονα να είναι επεξεργάσιμα δεδομένα.
- Στην χημεία χρησιμοποιείται η CML (Chemistry Markup Language) για κείμενα χημείας. Έτσι υπάρχει μια ανεξάρτητη πλατφόρμα για διαμοιρασμό και επεξεργασία μοριακών δεδομένων.
- Στον ηλεκτρονικό κόσμο των επιχειρήσεων η XRBL (eXtensible Business Reporting Language) έχει γίνει οδηγός των ηλεκτρονικών επιχειρήσεων με αντίστοιχο τρόπο.

- Η εξελικτική βιολογία και συγκριτική γονιδιωματική χρησιμοποιεί την PhyloXML φυλογενετικά δένδρα και δίκτυα καθώς και άλλα σχετικά δεδομένα.
- Η Systems Biology Markup Language (SBML) χρησιμοποιεί αναπαράσταση βασισμένη σε XML για υπολογιστικά μοντέλα και μοντέλα επικοινωνίας βιολογικών διεργασιών.

Η XML χρησιμοποιείται σε πολλές επιστήμες στην αναπαράσταση γνώσης και δεδομένων και τη μεταφορά των δεδομένων μεταξύ εφαρμογών ή/και ιστοσελίδων στο διαδίκτυο.

8.2 XML για περιγραφή προβλημάτων ωρολογίων προγραμμάτων και των λύσεων του προβλήματος (XHSTT).

Για την αναπαράσταση της περιγραφής ενός προβλήματος χρονοπρογραμματισμού καθώς επίσης των δεδομένων του και τελικά και των διαφόρων λύσεών του, ορίστηκε μια XML μορφή αναπαράστασης, που λέγεται Xml για High School TimeTable δεδομένα (XHSTT) [Post12a, Post12b, Post12c]. Η μορφή αυτή χρησιμοποιήθηκε για να αναπαρασταθούν, με κοινό τρόπο, προβλήματα χρονοπρογραμματισμού σχολείων δευτεροβάθμιας εκπαίδευσης από διάφορα κράτη του κόσμου, ώστε να υπάρξει και κοινή μέθοδος αξιολόγησης των λύσεων σε κοινά προβλήματα. Οι διαφορετικές εκπαιδευτικές δομές των κρατών ώθησε στη δημιουργία μιας γενικευμένης περιγραφής τέτοιων προβλημάτων καθώς και γενικευμένης περιγραφής των λύσεων ανά τον κόσμο.

Αυτή η μορφή αναπαράστασης χρησιμοποιήθηκε κατά κόρον στην επιστημονική κοινότητα. Επίσης, χρησιμοποιήθηκε σαν επίσημος τρόπος αναπαράστασης προβλημάτων και λύσεων σε παγκόσμιους διαγωνισμούς επίλυσης προβλημάτων δημιουργίας ωρολογίων προγραμμάτων για σχολεία δευτεροβάθμιας εκπαίδευσης. Έτσι πολλά ερευνητικά λογισμικά χρησιμοποιούν πλέον αυτή τη μορφοποίηση.

Ο Dr. Kingston [Kin16], δημιούργησε λογισμικό με άδεια GNU και ανοικτό κώδικα στο source Forge, όπου επίσης δέχεται προβλήματα με μορφοποίηση XHSTT και τελικά μέσα εκεί ενσωματώνει και τη λύση που προτείνει. Ο αλγόριθμος εύρεσης λύσης χρησιμοποιήθηκε από τον Fonseca [Fon17] ως αρχική λύση σε μεταερευνητικούς αλγόριθμους.

8.3 Η XML του χρονοπρογραμματισμού.

Όπως αναφέρθηκε παραπάνω, η XML έκδοση για τα προβλήματα χρονοπρογραμματισμού δευτεροβάθμιας και τριτοβάθμιας εκπαίδευσης είναι η XHSTT (XML for High School Time Tables). Πρωτοεμφανίστηκε επίσημα το 2011 από μία ομάδα ερευνητών και αρχικός της σκοπός ήταν να υπάρχει ένας κοινός τρόπος παρουσίασης των προβλημάτων, των περιορισμών και των λύσεων, ώστε να είναι δυνατόν να γίνει μια συγκριτική αξιολόγηση (benchmark) και να αξιολογούνται τα ερευνητικά λογισμικά, με σκοπό την εξέλιξη της έρευνας και της βελτίωσής τους. Έτσι από τότε μέχρι σήμερα έχει δημιουργηθεί ένα αποθετήριο⁶ από 50 περίπου προβλήματα⁷. Κάθε πρόβλημα μπορεί να επιλυθεί από κάποιο νέο ερευνητικό λογισμικό ή από μία νέα βελτιωμένη έκδοση κάποιου υπάρχοντος λογισμικού. Με τον τρόπο αυτό έχουν δημιουργηθεί αρκετά λογισμικά και τα περισσότερα από αυτά βελτιώνονται συνεχώς. Για αρκετά από αυτά τα προβλήματα, για τα οποία αρχικά είχε βρεθεί κάποια ικανοποιητική λύση, έχει βρεθεί πλέον και η βέλτιστη λύση, από βελτιωμένες εκδόσεις λογισμικών. Ένα πρόβλημα θα έχει επιλυθεί πλήρως εάν η συνάρτηση ελαχιστοποίησης έχει τιμή 0 κατά την λύση του προβλήματος. Επίσης είναι δυνατόν να υπάρχουν περισσότερες από μία βέλτιστες λύσεις με τιμή μηδέν. Εάν η τιμή δεν είναι μηδέν δεν είναι γνωστό, εάν υπάρχει βέλτιστη λύση με κόστος 0 και συνεχίζεται η αναζήτηση καλύτερης λύσης. Η XHSTT έγινε αποδεκτή από την επιστημονική κοινότητα και χρησιμοποιείται πολύ από τα ερευνητικά ιδρύματα στον τομέα του timetabling.

Το 2007 διεξήχθη ο δεύτερος διεθνής διαγωνισμός timetabling, ο οποίος όμως αφορούσε χρονοπρογραμματισμό για πανεπιστημιακά ιδρύματα. Εκεί η Ελληνική συμμετοχή με τον Δρ. Χ. Γκόγκο κατέκτησε τη δεύτερη θέση παγκοσμίως. Τα επόμενα χρόνια οι διαγωνισμοί που έγιναν αφορούσαν ωρολόγια προγράμματα δευτεροβάθμιας εκπαίδευσης και η διαδικασία αξιολόγησης είναι ανοικτή συνεχώς, όπως αναφέρθηκε παραπάνω.

Το 2012 το πρόβλημα προσδιορίστηκε χρησιμοποιώντας ουσιαστικά την XHSTT. Ο Dr. Kingston δημιούργησε νέα έκδοση του επιλυτή⁸ KHE το 2014 (ο οποίος επιλυτής περιγράφεται παρακάτω) ο οποίος βελτιώνεται συχνά (τελευταία έκδοση 6/2/2016). Επίσης ο Fonseca [2017] χρησιμοποίησε τη συγκεκριμένη μορφοποίηση, για να επαναπροσδιορίσει

⁶ <https://www.utwente.nl/ctit/hstt/archives/>

⁷ <https://www.utwente.nl/ctit/hstt/datasets/>

⁸ <http://sydney.edu.au/engineering/it/~jeff/khe/> (τελευταία πρόσβαση 1/7/2017)

και να γενικεύσει το πρόβλημα. Ο ίδιος ισχυρίζεται ότι μπορεί να περιγράψει όλα τα προβλήματα χρονοπρογραμματισμού και εξετάσεων δευτεροβάθμιας και τριτοβάθμιας εκπαίδευσης και πιθανότατα και γενικότερα προβλήματα [Fon2017].

Οι βασικές έννοιες (όπως της διδασκαλίας, των πόρων κλπ) συνεχίζουν να υπάρχουν αλλά σε πιο γενικευμένη μορφή.

8.4 Η οντότητα της χρονικής διάρκειας – (Times).

Η οντότητα χρόνος (Times) περιέχει πληροφορίες για την ελάχιστη μονάδα διδασκαλίας (την διδακτική ώρα, χρονοθυρίδα ή timeslot) και ένα σύνολο από αυτές ορίζει τη χρονική ομάδα διδασκαλίας (Time Group). Η χρονική ομάδα διδασκαλίας μπορεί να ορισθεί είτε ως το δίωρο διδασκαλίας της γλώσσας ή ως ένα συνεχόμενο δίωρο διδασκαλίας αρχαίων της Α λυκείου, ίσως ακόμα οι 5 ώρες διδασκαλίας αρχαίων της Α λυκείου ή ακόμα και ένα επτάωρο διδασκαλίας για μία ημέρα διδασκαλίας οποιουδήποτε τμήματος, καθώς και το σύνολο των 5 ωρών της 7^{ης} ώρας από κάθε ημέρα διδασκαλίας. Έτσι στην XHSTT το:

```
<Time Id="Mon4">
  <Name>Mon4</Name>
  <Day Reference="Δευτέρα"/>
  <TimeGroups>
    <TimeGroup Reference="BeforeLaunch"/>
  </TimeGroups>
</Time>
```

ορίζει ένα χρόνο Time, με Id (αναγνωριστικό) Mon4 (εννοείται η 4^η ώρα), την ημέρα Δευτέρα που είναι στο “TimeGroups” με όνομα «BeforeLaunch».

8.4.1 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). Η γενική σύνταξη των πόρων είναι:

```
Resources
+ResourceTypes
+ResourceGroups
*Resource
```

Η σύνταξη του ResourceTypes είναι:

ResourceTypes *ResourceType

Στον συμβολισμό που χρησιμοποιείται για όλη τη μορφοποίηση, το + δηλώνει ότι η κατηγορία που ακολουθεί είναι προαιρετική, ενώ το * δηλώνει ότι η ακόλουθη κατηγορία μπορεί είτε να μην εμφανισθεί, είτε να εμφανισθεί πολλές φορές.

Οι πιθανές τιμές για το ResourceType είναι οι τρεις προαναφερθείσες: Teacher, Room και Class. Για κάθε τύπο μπορούμε να ορίσουμε ομάδες (ResourceGroup). Ένα παράδειγμα τέτοιου ορισμού ακολουθεί:

```
ResourceGroups
  *ResourceGroup
```

```
ResourceGroup Id
  Name
  ResourceType
```

όπου

```
ResourceType Reference
```

Έτσι ένα παράδειγμα είναι το παρακάτω:

```
<ResourceGroup Id="ΕργαστήριοHY">
  <Name> ΕργαστήριοHY </Name>
  <ResourceType Reference="Αίθουσα"/>
</ResourceGroup>
```

ορίζει την ομάδα ResourceGroup με το όνομα ΕργαστήριοHY, και αναφέρεται ως αίθουσα διδασκαλίας.

Κάθε παιδί της κατηγορίας Resources ορίζει μία οντότητα (entity) πόρου, η οποία συμμετέχει σε γεγονότα (events), δηλαδή παραδόσεις μαθημάτων (Resources δεν είναι σίγουρα το Times).

Η σύνταξη του είναι:

```
Resource Id
  Name
  ResourceType
  +ResourceGroups
```

Ενώ η σύνταξη του ResourceGroups είναι:

```
ResourceGroups
  *ResourceGroup
```

και η σύνταξη του ResourceGroup Reference είναι:

ResourceGroup Reference

Όλα μαζί τα παραπάνω συνδυάζονται για να ορίσουν ένα πόρο (resource) όπως στο παράδειγμα:

```
<Name>Αίθουσα03</Name>
  <ResourceType Reference="Αίθουσα"/>
  <ResourceGroups>
    <ResourceGroup Reference="ΜεγάληΑίθουσα"/>
    <ResourceGroup Reference="ΑίθουσαΗΥ"/>
  </ResourceGroups>
</Resource>
```

στο οποίο η Αίθουσα03 ορίζεται ότι είναι Αίθουσα, η οποία ανήκει στις κατηγορίες “ΜεγάληΑίθουσα” και “ΑίθουσαΗΥ” (δηλαδή αφενός είναι μεγάλη αίθουσα, αφετέρου αίθουσα με υπολογιστές).

Αντίστοιχες ομάδες (Groups) ορίζονται, εκτός από τους πόρους της αίθουσας, για τους καθηγητές και τα τμήματα.

Κάποιοι πόροι είναι προκαθορισμένοι από τα δεδομένα του προβλήματος, ενώ οι υπόλοιποι, οι οποίοι δεν έχουν τιμές, είναι αυτοί στους οποίους πρέπει να ανατεθεί τιμή από τον επιλυτή. Κάθε πόρος έχει σημαντικό ρόλο στα events (γεγονότα) και χρησιμοποιείται για να συνδεθεί με περιορισμούς.

8.4.2 Η Οντότητα γεγονός (Event).

Ένα γεγονός (event) καθορίζει ότι συγκεκριμένοι πόροι (resources) πρέπει να συνδυαστούν για συγκεκριμένο αριθμό (ακέραιο) από χρονικές στιγμές (times) που ονομάζονται *διάρκεια*. Ο συγκεκριμένος αριθμός της διάρκειας και των πόρων μπορεί να καθορίζεται στο event ή ακόμα και να παραμένει απροσδιόριστος, ώστε να υπολογισθεί από τον επιλυτή του προβλήματος βάση των περιορισμών.

Το Events είναι παιδί (στο δένδρο) του Instance (δεδομένα προβλήματος για δημιουργίας ωρολογίου προγράμματος) με σύνταξη:

```
Events
  +EventGroups
  *Event
```

όπου

EventGroups
*Course
*EventGroup

και

EventGroup Id
Name

Η σύνταξη του Event είναι

Event Id +Color
Name
Duration
+Workload
+Course
+Time
+Resources
+ResourceGroups
+EventGroups

Τα Id και Name αναθέτουν ένα αναγνωριστικό και ένα όνομα με τον συνηθισμένο πλέον τρόπο. Η προαιρετική ιδιότητα Color μπορεί να χρησιμοποιηθεί ως χρώμα φόντου κατά την εκτύπωση και η μορφή του είναι, όπως για κάθε web χρώμα, της μορφής #RRGGBB (Red, Green, Blue). Η ιδιότητα διάρκεια (Duration) πρέπει να είναι ένας ακέραιος με τιμή τουλάχιστον 1.

Η προαιρετική κατηγορία workload ορίζει το φόρτο εργασίας του event. Εάν παραληφθεί, τότε έχει την (default) τιμή 1, ενώ σε αντίθετη περίπτωση πρέπει να έχει τουλάχιστον την ακέραια τιμή 0 για να υπολογισθεί η τιμή από τον επιλυτή.

Η προαιρετική κατηγορία Course καθορίζει ότι το γεγονός event ανήκει στο αναφερόμενο διδακτέο μάθημα Course, και η σύνταξή του είναι:

Course Reference

Όπου:

EventGroups
*EventGroup

με

EventGroup Reference

και καθορίζει ότι το event βρίσκεται στο αναφερόμενο event groups.

Η προαιρετική κατηγορία χρόνος (Time) χρησιμοποιείται μόνο όταν το γεγονός έχει γνωστή αρχή και καθορίζει ότι το γεγονός αρχίζει την χρονική στιγμή που προκαθορίζεται κατά την περιγραφή της. Η σύνταξή της είναι:

Time Reference

Τέλος, οι κατηγορίες Resources και ResourceGroups είναι και αυτές προαιρετικές και καθορίζουν τους πόρους που απαιτεί το συγκεκριμένο γεγονός event. Τα Resources έχουν σύνταξη:

Resources

*Resource

Κάθε Resource αναφέρεται σε ένα μόνο πόρο και χρησιμοποιείται μόνο όταν δεν αναφέρεται στον τυπικό (default) συνηθισμένο πόρο, και κάθε resource θα αναφέρεται ως event resource, για να διαφοροποιείται από το resource για καθηγητή ή αίθουσα (teacher και room resource). Η σύνταξη είναι παρόμοια:

Resource +Reference

+Role

+ResourceType

+Workload

και το νόημά του καθορίζεται από το αν η ιδιότητα Reference υπάρχει ή όχι.

Όταν η ιδιότητα Reference δεν έχει ορισθεί, τότε απαιτούμε από τον επιλυτή να αναθέσει ένα πόρο στο event resource, και τότε οι κατηγορίες Role και ResourceType είναι υποχρεωτικό να ορισθούν. Το Role δεν έχει ούτε ιδιότητες ούτε παιδιά. Το κυρίως σώμα του χρησιμοποιείται ως αναγνωριστικό, για να ταυτοποιήσει κάποιο πόρο για ένα event resource. Δύο event resources για κάποιο event είναι αδύνατο να έχουν την ίδια τιμή στο Role, αλλά φυσιολογικό για event resources μέσα σε διαφορετικά events να έχουν την ίδια τιμή για το Role. Η κατηγορία ResourceType καθορίζει τον τύπο του πόρου που πρέπει να εκχωρηθεί εδώ, ενώ σε αντίθετη περίπτωση η λύση θα απορριφθεί από τον επιλυτή.

8.5 Οι περιορισμοί

Οι περιορισμοί ορίζουν τις συνθήκες που απαιτούνται και τις συνθήκες οι οποίες είναι επιθυμητές να ικανοποιούνται σε ένα ωρολόγιο πρόγραμμα (timetable). Όπως έχει αναφερθεί, αυστηροί περιορισμοί είναι εκείνοι στους οποίους η ικανοποίησή τους είναι υποχρεωτική, ενώ χαλαροί είναι εκείνοι οι οποίοι είναι επιθυμητό να ικανοποιούνται. Κάθε περιορισμός πρέπει υποχρεωτικά να είναι είτε αυστηρός είτε χαλαρός. Η παράβαση κάποιου περιορισμού συνεπάγεται απόκλιση από τη ζητούμενη βέλτιστη λύση, και εάν ο περιορισμός που παραβιάζεται είναι αυστηρός, τότε δεν είναι δυνατόν να δεχθούμε την κατάσταση αυτή, ενώ εάν παραβιάζεται κάποιος χαλαρός περιορισμός, τότε η παραβίαση αυτή προσθέτει ένα βάρος στην αντικειμενική συνάρτηση. Υπάρχουν τρεις τύποι για τη συνάρτηση κόστους: η γραμμική, η τετραγωνική και η βηματοειδής. Για περιορισμούς με

γραμμικό κόστος, το κόστος για τις παραβιάσεις ισούται με το πλήθος των παραβιάσεων πολλαπλασιασμένο με το κόστος της μιας παραβίασης. Για τους περιορισμούς τετραγωνικού κόστους, στην αντικειμενική συνάρτηση προστίθεται το γινόμενο του κόστους με το τετράγωνο του πλήθους των περιορισμών αυτών. Τέλος, για τους περιορισμούς βηματικού κόστους, το κόστος προστίθεται μόνο μία φορά, ανεξάρτητα από το πόσοι τέτοιοι περιορισμοί έχουν παραβιασθεί.

Στην περιγραφή με XHSTT υπάρχουν πλέον περιορισμοί 16 διαφορετικών τύπων, οι οποίοι περιγράφονται (συνοπτικά) παρακάτω γιατί αυτό θεωρείται σημαντικό:

1. **Ανάθεση χρόνου** (Assign Time): καθορίζει την αναγκαιότητα ανάθεσης αρκετών χρονικών μονάδων για ένα σύνολο από γεγονότα (events - διδασκαλίες).
2. **Ανάθεση πόρου** (Assign Resource): καθορίζει την απαίτηση ανάθεσης ενός πόρου σε ένα σύνολο από χρονικές στιγμές (timeslots).
3. **Επιθυμητές ώρες** (Prefer Times): δηλώνει ότι κάποια γεγονότα έχουν προτεραιότητα για ένα σύνολο από χρονικές στιγμές (timeslots).
4. **Επιθυμητοί πόροι** (Prefer Resources): δηλώνει ότι κάποια γεγονότα έχουν προτεραιότητα για ένα συγκεκριμένο σύνολο από πόρους (resources).
5. **Σύνδεση γεγονότων** (Link Events): καθορίζει ότι ένα σύνολο από γεγονότα πρέπει να λάβουν χώρα την ίδια χρονική περίοδο.
6. **Σειρά γεγονότων** (Order Events): Αναγκάζει ένα γεγονός να λάβει χώρα, τουλάχιστον (ή το πολύ) ένα προκαθορισμένο αριθμό από χρονικές στιγμές μετά από ένα άλλο.
7. **Διασπορά γεγονότων** (Spread events): δηλώνει ότι ένα σύνολο από γεγονότα πρέπει να διαμοιρασθεί σε τουλάχιστον (ή το πολύ) ένα συγκεκριμένο αριθμό από ημέρες.
8. **Αποφυγή διαχωρισμού αναθέσεων** (Avoid split assignments): Καθορίζει ότι οι ίδιοι πόροι, πρέπει να ανατίθενται για όλες τις συναντήσεις ενός γεγονότος (διδασκαλίας).
9. **Κατανομή διαχωρισμένων διδασκαλιών** (Distribute split events): δηλώνει ότι ένα σύνολο από γεγονότα πρέπει να διαχωρισθεί σε συναντήσεις μεταξύ ενός ελαχίστου και ενός μεγίστου αριθμού από συναντήσεις δεδομένης διάρκειας.

10. **Διαχωρισμός γεγονότων (Split events)**: καθορίζει όρια στο πλήθος μη διαδοχικών συναντήσεων που δημιουργούνται για ένα γεγονός, καθώς και στη διάρκειά αυτών.
11. **Αποφυγή συγκρούσεων (Avoid clashes)**: ορίζει ότι συγκεκριμένοι πόροι πρέπει να ανατεθούν χωρίς σύγκρουση (δηλαδή, να μην ανατεθούν οι ίδιοι πόροι σε περισσότερα από ένα γεγονός ανά χρονική στιγμή/ώρα διδασκαλίας).
12. **Αποφυγή μη διαθέσιμων χρονικών περιόδων (Avoid unavailable times)**: καθορίζει τους βεβαιωμένους πόρους που δεν είναι διαθέσιμοι, να είναι διαθέσιμοι σε οποιαδήποτε διδασκαλία (γεγονός), σε συγκεκριμένες χρονικές περιόδους.
13. **Περιορισμός φόρτου εργασίας (Limit workload)**: περιορίζει το φόρτο εργασίας για συγκεκριμένο πόρο, μεταξύ μιας ελάχιστης και μιας μέγιστης τιμής.
14. **Περιορισμός των κενών (Limit Idle Times)**: ορίζει ότι ο αριθμός των κενών (ώρα χωρίς διδασκαλία) ανά ημέρα πρέπει να βρίσκεται σε μια περιοχή ενός ελαχίστου και ενός μέγιστου, για κάποιους πόρους.
15. **Περιορισμός απασχολούμενων ωρών (Limit busy times)**: δηλώνει ότι ο αριθμός των απασχολούμενων ωρών για μία ημέρα πρέπει να βρίσκεται ανάμεσα από ένα ελάχιστο και ένα μέγιστο για κάποιο σύνολο από πόρους.
16. **Σύνολο απασχολούμενων περιόδων (Cluster Busy Times)**: καθορίζει τα όρια του αριθμού των ημερών στις οποίες μπορεί να λάβουν χώρα δραστηριότητες.

Η αντικειμενική συνάρτηση (new **Formulation 2**) $f(\cdot)$ υπολογίζεται με όρους παραβίασης περιορισμών. Κάθε παραβίαση περιορισμού έχει κόστος ανάλογα με το βάρος του (στο πρόβλημα ελαχιστοποίησης). Η τιμή της αντικειμενικής συνάρτησης είναι πλέον συνάρτηση δύο μεταβλητών (H, S) , όπου H είναι το κόστος παραβίασης του περιορισμού, και S το πλήθος των παραβιάσεων για τους χαλαρούς περιορισμούς.

8.6 Η Μορφοποίηση κατά Fonseca 2017

Οι αλλαγές που φέρνει η χρήση της XML είναι πολλές. Το ίδιο το πρόβλημα που περιγράφεται στο κεφάλαιο 3 γίνεται πιο πολύπλοκο και οι εξισώσεις που περιγράφουν το πρόβλημα πλέον γίνονται πολυσύνθετες. Οι νέες μαθηματικές εξισώσεις [Fon17] που περιγράφουν το πρόβλημα είναι αντίστοιχα πιο ισχυρές και πιο σύνθετες και το μαθηματικό μοντέλο μπορεί να περιγράψει προβλήματα πιο πολύπλοκα από τα υπαρκτά προβλήματα

κάθε χώρας και σε κάθε βαθμίδα εκπαίδευσης, για ωρολόγιο πρόγραμμα καθώς και για πρόγραμμα εξετάσεων.

Οι περισσότεροι ερευνητές του χρονοπρογραμματισμού μέχρι και σήμερα συνηθίζουν να ασχολούνται με περιορισμένες λύσεις, για προβλήματα που αφορούν ιδρύματα μόνο για δευτεροβάθμια ή για τριτοβάθμια εκπαίδευση. Η χρήση της XHSTT όμως αναγκάζει σε αναζήτηση λύσης του γενικευμένου προβλήματος και επιτρέπει τη χρήση τεχνικών προ-επεξεργασίας, για καλύτερα αποτελέσματα και βοηθάει στην αναζήτηση νέων τεχνικών αναζήτησης λύσης του προβλήματος.

8.7 Το ανοιχτό λογισμικό KHE High School Timetabling Engine

Ο Dr. Kingston αρχικά ανέπτυξε ένα πρόγραμμα αξιολόγησης για τους προαναφερόμενους διαγωνισμούς. Στη συνέχεια όμως ανέπτυξε λογισμικό το οποίο επιλύει προβλήματα που περιγράφονται γενικευμένα με XHSTT και επίσης αξιολογεί τις λύσεις. Κάθε λύση ενσωματώνεται στο εξαγόμενο αρχείο και έτσι τελικά κάθε πρόβλημα μπορεί να περιέχει πολλές διαφορετικές αξιολογημένες λύσεις από πολλούς διαφορετικούς επιλυτές (λογισμικά).

Το KHE υλοποιήθηκε να λειτουργεί με βάση την XHSTT και αξιοποιεί κάθε δυνατότητά της. Έχει γραφεί εγχειρίδιο για τους προγραμματιστές, ώστε να μπορούν, εάν το επιθυμούν, να χρησιμοποιήσουν κάθε τμήμα του. Το εγχειρίδιο περιγράφει όλες τις διαδικασίες – συναρτήσεις με τις παραμέτρους, που μπορεί ένα προγραμματιστής να χρησιμοποιήσει. Αυτό μάλιστα γίνεται οργανωμένα και ανά οντότητα. Έτσι ένας προγραμματιστής έχει τη δυνατότητα να χρησιμοποιήσει έτοιμες συναρτήσεις, από το KHE, για να «διαβάσει» (είσοδος) ένα XML αρχείο, ή να «δημιουργήσει» ένα αρχείο (έξοδος) μιας λύσης έχοντας όμως το δικό του λογισμικό δημιουργίας λύσης, εάν το επιθυμεί.

8.7.1 Περιγραφή διαδικασίας εύρεσης λύσης.

Ενδιαφέρον παρουσιάζει η δομή του αλγόριθμου, ο οποίος χωρίζεται σε τρία τμήματα.

Στην αρχή υπάρχει η *φάση δημιουργίας δομών δεδομένων* (structural phase), κατά την οποία δημιουργείται η αρχική λύση. Η λύση αυτή είναι χτισμένη χωρίς αναθέσεις χρόνων και πόρων. Στο βήμα αυτό, τα γεγονότα χωρίζονται σε υπογεγονότα (sub-events), των οποίων η διάρκεια εξαρτάται από τον περιορισμό διαχωρισμού γεγονότων και τα οποία ομαδοποιούνται σε σύνολα που ονομάζονται *κόμβοι* (nodes). Τα υπογεγονότα τα οποία προκύπτουν από το ίδιο γεγονός είναι εμφωλεμένα κάτω από τον ίδιο κόμβο. Στον ίδιο

κόμβο βρίσκονται και τα υπογεγονότα τα οποία προκύπτουν από τους περιορισμούς διασποράς γεγονότων (περιορισμός 7) και αποφυγής διαχωρισμού αναθέσεων (περιορισμός 8). Γεγονότα τα οποία συνδέονται με περιορισμό σύνδεσης γεγονότων έχουν τις συναντήσεις/διδασκαλίες συνδεδεμένες με τρόπο, ώστε ο χρόνος ανάθεσης σε κάποια από αυτές τις συναντήσεις εκτείνεται επίσης στις άλλες συνδεδεμένες συναντήσεις. Κάθε συνάντηση περιέχει επίσης ένα σύνολο από χρονικά διαστήματα και ονομάζεται *time domain* (πεδίο του χρόνου) και το οποίο καθορίζει τις χρονικές περιόδους διδασκαλίας που μπορούν να ανατεθούν στην απλούστερη διδασκαλία. Κάθε διδασκαλία περιέχει επίσης ένα task για κάθε απαιτούμενο πόρο κατάλληλου τύπου και καλείται *resource domain* (πεδίο πόρων). Οι πόροι του πεδίου αυτού βασίζονται στους περιορισμούς επιθυμητών πόρων (4). Στη φάση αυτή ανατίθενται στις μεταβλητές χρόνοι και πόροι, οι οποίοι έχουν εκ των προτέρων ανατεθεί από τον δημιουργό του προβλήματος από τα γεγονότα στα υπογεγονότα.

Η δεύτερη φάση λέγεται *φάση ανάθεσης χρόνων (time assignment)* στην οποία ανατίθεται ο χρόνος στον οποίο πρέπει να γίνει κάθε συνάντηση (κάθε γεγονός/event). Η φάση αυτή αρχίζει με τη δημιουργία ενός επιπέδου (layer) για κάθε πόρο του περιορισμού «αποφυγής συγκρούσεων», το οποίο είναι ένα σύνολο από κόμβους που περιέχουν συναντήσεις προανατιθέμενες στον δεδομένο πόρο. Έπειτα, τα επίπεδα ταξινομούνται, έτσι ώστε τα επίπεδα με τις λιγότερες διαθέσιμες επιλογές να είναι πρώτα. Έτσι οι ώρες διδασκαλίας ανατίθενται στις συναντήσεις από τα επίπεδα με βάση τη σειρά επιπέδων. Αυτή η ανάθεση γίνεται με το ελάχιστο κόστος αντιστοίχισης μεταξύ συναντήσεων του τρέχοντος επιπέδου και των χρόνων ανάθεσης. Κάθε ακμή του γράφου περιέχει κόστος ίσο με την επίπτωση που κοστίζει η ανάθεση με βάση την αντικειμενική συνάρτηση.

Στην τελική φάση που ονομάζεται *φάση ανάθεσης πόρων (resource assignment phase)*, ανατίθενται οι πόροι του προβλήματος. Για κάθε τύπο πόρων εκτελείται μία από τις δύο παρακάτω διαδικασίες: (i) εάν η ανάθεση του πόρου περιορίζεται από τον περιορισμό «αποφυγής διαχωρισμού αναθέσεων», καλείται ο αλγόριθμος «συσκευασίας πόρων» (*resource packing*), (ii) σε κάθε άλλη περίπτωση καλείται ένας ευρετικός αλγόριθμος.

Το λογισμικό δημιουργήθηκε στη γλώσσα C και ο πηγαίος κώδικας διατίθεται δωρεάν με την άδεια GNU version 3. Ο κώδικας μεταφράζεται σε όλες τις διαδεδομένες πλατφόρμες (mac os x, linux και windows) και εκτελείται σε περιβάλλον απλού τερματικού χωρίς γραφικά. Η νεότερη έκδοση αναρτήθηκε στις 6/2/2016 (τελευταίος έλεγχος έγινε τον Ιούλιο 2017), η οποία μεταφράζεται και εκτελείται χωρίς προλήματα.

8.8 Το εμπορικό λογισμικό asc TIMETABLE

Εάν όχι το πιο διαδεδομένο, τότε ένα από τα πιο διαδεδομένα προγράμματα δημιουργίας ωρολογίων προγραμμάτων είναι το ascTIMETABLE. Τα εμπορικά λογισμικά χρησιμοποιούν και αυτά XML γλώσσα και εσωτερικά στο λογισμικό τους αλλά και για επικοινωνία με άλλα βοηθητικά προγράμματα, αλλά και για εμφάνιση των λύσεων τους σε ιστοσελίδες.

Η XML όμως του κάθε εμπορικού λογισμικού είναι διαφορετική από την XML οποιουδήποτε άλλου. Έτσι δεν χρησιμοποιείται ουσιαστικά από κανένα άλλο λογισμικό. Συνεπώς, ενώ τα ερευνητικά λογισμικά μπορούν να ανταλλάσσουν προβλήματα, περιορισμούς, λύσεις και να έχουν επιπλέον κοινή μέθοδο αξιολόγησης, τα εμπορικά δεν είναι δυνατό να έχουν κάτι από αυτά, τουλάχιστον αυτοματοποιημένα.

Επίσης, δεν αναφέρονται παρά μόνο ελάχιστα για τους αλγόριθμους που χρησιμοποιούνται για την επίλυση του προβλήματος. Η συγκεκριμένη εταιρεία αναφέρει στην ιστοσελίδα της ότι «έχει δημιουργήσει και χρησιμοποιεί ένα καινοτόμο αλγόριθμο που βασίζεται στην οπισθοδρόμηση και χρησιμοποιεί πολλές μεταερευνητικές τεχνικές και παραλληλία σε επεξεργαστές με πολλά νήματα».

Το σημαντικότερο που μπορεί να αναφερθεί για το λογισμικό αυτό είναι ότι το γραφικό περιβάλλον εισαγωγής δεδομένων καθώς και παρουσίασης των αποτελεσμάτων είναι ένα από τα πιο ολοκληρωμένα στο χρονοπρογραμματισμό σχολικών μονάδων.

8.9 Το ανοιχτό λογισμικό FET

Το FET (Free Timetabling Software) δημιούργησε ο Liviu Lalescu από τη Ρουμανία για πρώτη φορά το 2002 χρησιμοποιώντας ένα γενετικό αλγόριθμο με μικρές δυνατότητες. Το 2007 δημιουργήθηκε νέος αλγόριθμος, ο οποίος είχε την δυνατότητα να επιλύσει δύσκολα προβλήματα γρήγορα. Ο ίδιος αλγόριθμος, με βελτιώσεις, χρησιμοποιείται μέχρι και σήμερα. Αναφέρεται ότι χρησιμοποιεί ευρετικές τεχνικές (heuristics), προσπαθώντας να προσομοιώσει τη χειροκίνητη διαδικασία επίλυσης του προβλήματος. Ο ίδιος ονόμασε τον αλγόριθμό του "recursive swapping" και σχετίζεται με τον αλγόριθμο "ejection chain".

Ο αλγόριθμος τοποθετεί τις δραστηριότητες με τη σειρά, αρχίζοντας από τις πιο “δύσκολες”. Εάν δεν μπορεί να βρει λύση, τότε παραπέμπει σε πιθανά λάθη αναθέσεως μαθημάτων. Όταν απαιτείται τοποθέτηση νέας δραστηριότητας, αντιμετωπίζει δραστηριότητες αναδρομικά, εάν είναι δυνατό, με σκοπό να δημιουργηθεί χώρος για αυτήν, ή ακόμα (σε ακραίες περιπτώσεις) κάνει οπισθοδρόμηση (backtracks) αλλάζοντας τη σειρά δραστηριοτήτων.

Ο αλγόριθμος, σύμφωνα με τα στοιχεία που δίνει ο συγγραφέας του λογισμικού, περιγράφεται συνοπτικά παρακάτω:

1. Ταξινόμηση των δραστηριοτήτων, αρχίζοντας από τις πιο δύσκολες. Ο δημιουργός ισχυρίζεται ότι το βήμα αυτό επιταχύνει τον αλγόριθμο μέχρι και 10 φορές.
2. Προσπάθησε να τοποθετήσεις τη δραστηριότητα $A[i]$ σε ένα επιτρεπόμενο χρονικό διάστημα, ακολουθώντας την παραπάνω σειρά. Αναζήτησε μια διαθέσιμη χρονική μονάδα για διδασκαλία στην οποία μπορεί να τοποθετηθεί η δραστηριότητα, εάν το επιτρέπουν οι περιορισμοί. Εάν υπάρχουν πολλές θέσεις, τότε διάλεξε μία στην τύχη, ενώ εάν δεν υπάρχουν, κάνε αναδρομικά αντιμετωπίσεις.
 - a. Για κάθε χρονική μονάδα $T[j]$ κατάγραψε τι συμβαίνει, εάν θέσουμε το $A[i]$ στο $T[j]$. Δημιούργησε μια λίστα με δραστηριότητες οι οποίες έρχονται σε αντίθεση με την παραπάνω τοποθέτηση (π.χ. η δραστηριότητα $A[k]$ απαιτεί να τοποθετηθεί στην ίδια χρονική στιγμή $T[j]$ και έχει τον ίδιο καθηγητή ή το ίδιο τμήμα διδασκαλίας όπως και η $A[i]$). Δημιούργησε μια λίστα με τις συγκρουόμενες δραστηριότητες για κάθε χρονική μονάδα διδασκαλίας $T[j]$.
 - b. Διάλεξε ένα χρονικό διάστημα $T[j]$ που να έχει τον ελάχιστο αριθμό από συγκρουόμενες δραστηριότητες. Υποθέτουμε ότι αυτή η λίστα έχει 3 δραστηριότητες: $A[p]$, $A[q]$ και $A[r]$.
 - c. Τοποθέτησε την $A[i]$ στο $T[j]$ και μην καταχωρείς τα $A[p]$, $A[q]$ και $A[r]$.
 - d. Αναδρομικά προσπάθησε να καταχωρίσεις τα $A[p]$, $A[q]$ και $A[r]$ (εάν το επίπεδο της αναδρομής δεν είναι τόσο βαθύ, έστω 14, και εάν ο συνολικός αριθμός των αναδρομικών κλήσεων του βήματος (2) από την αρχική προσπάθεια τοποθέτησης του $A[i]$ δεν είναι πολύ μεγάλο, έστω 2^n) όπως και στο βήμα (2).
 - e. Εάν τα $A[p]$, $A[q]$ και $A[r]$ τοποθετήθηκαν με επιτυχία, τελείωσε το βήμα 2 και επέστρεψε με επιτυχία, διαφορετικά προσπάθησε να τοποθετήσεις άλλα χρονικά

διαστήματα, δηλαδή πήγαινε στο βήμα 2b και προσπάθησε με το επόμενο χρονικό διάστημα.

- f. Εάν όλα (ή ένας λογικά μεγάλος αριθμός) από χρονικά διαστήματα δοκιμάστηκαν ανεπιτυχώς, επέστρεψε με αποτυχία.
- g. Εάν βρισκόμαστε στο επίπεδο 0, δεν επιτεύχθηκε η τοποθέτηση του A[i] τοποθέτησέ το, όπως στα βήματα 2b και 2c, αλλά χωρίς αναδρομή. Τώρα πρέπει να τοποθετηθούν ακόμα $3 - 1 = 2$ δραστηριότητες. Πήγαινε στο βήμα 2. (Στο σημείο αυτό τοποθετούνται μερικές διαδικασίες για να αποφευχθεί η κυκλική χρήση διαδικασιών).

Για πολλά προγράμματα, και ιδιαίτερα ανοικτού κώδικα, γράφονται διάφορα πρόσθετα εργαλεία. Για το FET υπάρχει ένα εργαλείο το οποίο δέχεται την έξοδο του FET με το δημιουργημένο ωρολόγιο πρόγραμμα και δημιουργεί γεγονότα που μπορούν να τοποθετηθούν σε ημερολόγιο τύπου iCal, όπως είναι το google calendar και το iCalendar της Apple. Επίσης υπάρχουν και κάποια «εργαλεία» ακόμη, ώστε να διευκολύνουν τη χρήση του προγράμματος. Αυτό συμβαίνει γιατί το FET μειονεκτεί ως προς το γραφικό περιβάλλον εισαγωγής των δεδομένων με αποτέλεσμα προγραμματιστές δημιουργίας λογισμικού ανοικτού κώδικα να δημιουργούν βοηθητικά εργαλεία, όπου χρειάζεται.

Το FET χαρακτηρίζεται ως ένα πολύ καλό λογισμικό για σχετικά μικρά σχολεία, μιας και οι δυνατότητές του είναι σχετικά μικρές, σύμφωνα με τους περιορισμούς που αναφέρει ο δημιουργός στην ιστοσελίδα υποστήριξης της εφαρμογής αυτής. Αυτό σημαίνει ότι μπορεί να χρησιμοποιηθεί σε δημοτικά σχολεία και σε μικρά γυμνάσια σε όλη την Ελλάδα.

ΚΕΦΑΛΑΙΟ 9

Συμπεράσματα, προτάσεις, επίλογος.

Πολλοί αλγόριθμοι έχουν υλοποιηθεί τα τελευταία χρόνια με σκοπό την εύρεση βέλτιστης ή έστω ικανοποιητικής λύσης χρονοπρογραμματισμού σχολικών μονάδων. Αρκετοί από αυτούς δίνουν ικανοποιητικά αποτελέσματα σε ικανοποιητικό χρόνο εκτέλεσης. Οι περισσότερες τεχνικές χρησιμοποιούν συνδυασμούς από μαθηματικές τεχνικές και μεταευρετικές μεθόδους και συνεχώς βελτιώνονται. Ο χώρος αναζήτησης είναι τεράστιος και οι διαδικασίες λύσεις περιέχουν μεταευρετικές τεχνικές με αναζήτηση σε περιορισμένο χώρο, χωρίς να υπάρχει εγγύηση εύρεσης της βέλτιστης λύσης. Ο χώρος έχει γίνει ελκυστικός για έρευνα και η ύπαρξη της νέας μορφοποίησης δίνει άλλη διάσταση στο πρόβλημα.

Πέρα από τις σκέψεις που έχουν υλοποιηθεί, ο χώρος είναι ανοιχτός για συνεχή έρευνα και νέες υλοποιήσεις. Οι εφαρμογές που υπάρχουν είναι συνήθως αυτόνομες και εκτελούνται σε τοπικούς υπολογιστές.

Κάποιες εφαρμογές χρησιμοποιούν προγραμματιστικά εργαλεία και επιλυτές εύρεσης λύσεων είτε ακέραιου προγραμματισμού (ανοικτού κώδικα όπως το GLPK⁹, ή εμπορικά όπως το gurobi¹⁰) είτε μεταευρετικούς αλγόριθμους. Ο Rajero et. Al. 2011 [PCLF11], αναφέρει μια συγκριτική αξιολόγηση για 33 λογισμικά βελτιστοποίησης, με 271 διαφορετικά κριτήρια, ομαδοποιημένα σε 6 περιοχές. Τα αναφερόμενα από τον Rajero et.al 2011 λογισμικά βιβλιοθήκες μπορούν να χρησιμοποιηθούν ως επιλυτές του προβλήματος χρονοπρογραμματισμού, αρκεί βέβαια να γενικεύσουμε και να κωδικοποιήσουμε το πρόβλημα μας κατάλληλα, ώστε να είναι κατανοητό από τον κάθε επιλυτή/λογισμικό.

Μια σκέψη είναι να δημιουργηθεί για κάθε επιλυτή κάποιο web service, το οποίο θα εκτελείται σε κάποιον ισχυρό εξυπηρετητή. Ο επιλυτής θα τροφοδοτείται με το XHSTT πρόβλημα (περιγραφή και περιορισμούς). Η επιστροφή από τον επιλυτή θα περιέχει την λύση καθώς και αξιολόγηση για το κατά πόσο έχουν παραβιασθεί κάποιοι περιορισμοί και το αντίστοιχο «κόστος παραβίασης». Ακόμη και εάν οι επιλυτές δεν δέχονται ως είσοδο αρχεία XHSTT, μπορεί να δημιουργηθεί κάποιο άλλο web service, το οποίο θα έχει ως

⁹ <https://www.gnu.org/software/glpk/>

¹⁰ <http://www.gurobi.com/products/gurobi-optimizer/>

είσοδο αρχεία του τύπου αυτού και έξοδο κατάλληλη, ώστε να αποτελούν είσοδο για τους επιλυτές.

Τα ωρολόγια προγράμματα όλων των σχολείων, ούτως ή άλλως, αποθηκεύονται σε κεντρική βάση δεδομένων (στο Υπουργείο Παιδείας). Εκεί μπορεί να προστεθεί η δυνατότητα να αποθηκευτούν ως ΧΗSTT (η να μετατρέπονται εύκολα σε αυτή τη μορφή). Αυτό μπορεί να επιτρέψει την εύκολη μεταφορά δεδομένων μεταξύ των σχολείων. Κάθε σχολείο θα μπορεί να ξέρει για τους καθηγητές που δεν απασχολεί πλήρως, ποιες ημέρες και ώρες είναι διαθέσιμοι, και ποιες ημέρες και ώρες απασχολούνται και σε ποιες σχολικές μονάδες.

Επίσης εάν κάποιες σχολικές μονάδες χρησιμοποιούν τις ίδιες ή πολύ κοντινές εγκαταστάσεις και απασχολούν κοινούς καθηγητές τότε θα μπορούσαν να δημιουργούν κοινούς περιορισμούς, οι οποίοι θα περιγράφουν τις διαθέσιμες ημέρες και ώρες που θα αφορούν κάθε καθηγητή που μετακινείται σε περισσότερες από μια σχολικές μονάδες. Έτσι το web service για τη δημιουργία του κάθε timetable θα μπορούσε να διορθώσει και να βελτιώσει τη λύση για κάθε σχολείο κατάλληλα. Ακόμα κάποιοι κατάλληλοι επιλυτές θα μπορούσαν να προτείνουν μια καλή λύση, σε σχέση με το ποιες μέρες και ποιες ώρες θα πρέπει να διατίθεται ο κάθε καθηγητής στο σχολείο στο οποίο καλείται να συμπληρώσει το ωράριό του.

ΕΠΙΛΟΓΟΣ

Τα ωρολόγια προγράμματα ανά το κόσμο έχουν μεγάλες διαφορές. Όσον αφορά τα ελληνικά δεδομένα, και ιδιαίτερα στα λύκεια, υπάρχει μια ιδιομορφία σε μεγάλο βαθμό. Αυτή είναι η ύπαρξη ζωνών. Με τη μελέτη της βιβλιογραφίας, έχει γίνει φανερό ότι δεν υπάρχει αλγόριθμος ο οποίος να εκμεταλλεύεται την ύπαρξη ζωνών. Η ιδιομορφία αυτή μπορεί να περιορίσει αρκετά το χώρο αναζήτησης και θεωρητικά φαίνεται να είναι πιο εύκολο να ελεγχθούν αρχικά οι περιορισμοί και να μειωθεί αρκετά, σε πρώτη φάση, ο τεράστιος χώρος αναζήτησης, τοποθετώντας πόρους (κυρίως καθηγητές) σε ομάδες χρόνου (Time Groups). Σε δεύτερη φάση αφήνεται η αναζήτηση της βέλτιστης λύσης στο περιορισμένο πλέον χώρο. Οι τεχνικές αναζήτησης που μπορεί να χρησιμοποιηθούν στις δύο φάσεις, θα είναι είτε από τον μαθηματικό προγραμματισμό είτε κάποια από τις ευρετικές ή μεταευρετικές τεχνικές. Μεταξύ των δύο φάσεων μπορεί να υπάρχει και οπισθοδρόμηση.

Το πρόβλημα της εύρεσης λύσης με πλήρως αυτοματοποιημένη διαδικασία για κάθε πρόβλημα, είναι πολύ δύσκολο να αποδώσει βέλτιστη λύση, εάν τελικά αυτό είναι δυνατόν. Τα timetables λογισμικά είναι δυνατόν να βελτιώνονται συνεχώς.

ΠΑΡΑΡΤΗΜΑ Α

XSLT Πρόγραμμα το οποίο εμφανίζει XHSTT timetable στην οθόνη φυλλομετρητή.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <!-- ***** IT USES functions (templates) in order to pass parameters. **** -->
  <!-- **** template FindRow (actually print a row) *****8** -->
  <xsl:template name="findRow">
    <xsl:param name="myRow" select="$myRow"/>
    <xsl:value-of select="x" />
    <xsl:for-each
select="/HighSchoolTimetableArchive/Instances/Instance/Times/Time">
      <td>
        <xsl:call-template name="findColoumn">
          <xsl:with-param name="myRow2" select="$myRow"/><!-- Teacher's Name -
->
          <xsl:with-param name="myCol" select="@Id"/><!-- Hour= Coloumn's Name
-->
        </xsl:call-template>
      </td>
    </xsl:for-each>
  </xsl:template>
  <!-- ***** template FindComlounn to print just an element of a row
***** -->
  <xsl:template
match="/HighSchoolTimetableArchive/SolutionGroups/SolutionGroup/Solution/Events/Eve
nt" name="findColoumn">
    <!-- scan all events to find the one (if any) which fits to the specified ROW-
COLOUMN -->
    <xsl:param name="myRow2"/><!-- Teacher's Name -->
    <xsl:param name="myCol"/><!-- Hour= Coloumn'd Name -->
    <xsl:for-each
select="/HighSchoolTimetableArchive/SolutionGroups/SolutionGroup/Solution/Events/Eve
nt"><!-- Solution Events with Reference,Duration,Time -->
      <xsl:if test="$myCol=Time/@Reference">
        <!--xsl:value-of select="Time/@Reference"/-->
        <xsl:call-template name="findRelation">
          <xsl:with-param name="TableTeacher" select="$myRow2"/>
          <xsl:with-param name="TableDayHour" select="$myCol"/>
          <xsl:with-param name="mySolutionAnathesi" select="@Reference"/>
          <xsl:with-param name="mySolutionHour" select="Time/@Reference"/>
        </xsl:call-template>
      </xsl:if>
    </xsl:for-each>
  </xsl:template>
  <!-- ***** template findRelation ***** -->
  <!-- *** Scan a solution-event (with ID), a teacher and an hour to check if it has to be
printed *** -->
```

```

<!-- ** and IF the course(Lecture) description Contains the TableTeacher IT IS A
SOLUTION -**** -->
<!-- ** DISPLAY the result . ***-->
<xsl:template

match="/HighSchoolTimetableArchive/Instances/Instance/Events/Event/EventGroups/Cour
se" name="findRelation">
  <xsl:param name="TableTeacher"/><!-- the Teacher -->
  <xsl:param name="TableDayHour"/><!-- The Lecture Time -->
  <xsl:param name="mySolutionAnathesi"/><!-- What Course (referenced from
Solution) -->
  <xsl:param name="mySolutionHour"/><!-- Which teacher (referenced from Solution)
-->
  <xsl:value-of select="$mySolutionAnathesi"/>,,
  <xsl:for-each
select="/HighSchoolTimetableArchive/Instances/Instance/Events/Event/EventGroups/Cours
e">
  <xsl:value-of select="Resources/Resource/@Reference"/>
  <xsl:if test="$TableTeacher/@Id=Resources/Resource/@Reference and
  $TableDayHour=$mySolutionHour and $mySolutionAnathesi=@Id">
    <xsl:call-template name="myextraction">
      <xsl:with-param name="MyResult1" select="Name"/>
      <xsl:with-param name="MyLogical1" select="$TableTeacher=substring-
before(substring-after(Name,'Teachers:[]'),'') and $TableDayHour=$mySolutionHour"/>
      <xsl:with-param name="mySolutionTEACHER" select="$TableTeacher"/>
    </xsl:call-template>
  </xsl:if>
</xsl:for-each>
</xsl:template>
<!-- ***** -->
<xsl:template name="myextraction">
  <xsl:param name="MyResult1"/>
  <xsl:param name="MyLogical1"/>
  <xsl:param name="mySolutionTEACHER"/>
  <!-- FINAL PRINT OUT -->
  <xsl:value-of select="substring-before(substring-after($MyResult1,'Classes:[]'),'')"/>,
  <!--xsl:value-of select="substring-before(substring-
after($MyResult1,'Teachers:[]'),'')"/-->
  <!--xsl:value-of select="substring-before(substring-
after($MyResult1,'Subject:'),'Classes')"/-->
</xsl:template>
<!-- ***** -->
<!-- ***** MAIN ***** -->
<!-- ***** -->
<xsl:template match="/">
  <html>
  <body>
    <table border="1">
      <tr bgcolor="#9acd32">
        <!-- ***** Header ***** -->

```



```

        <th></th>
        <xsl:for-each
select="/HighSchoolTimetableArchive/Instances/Instance/Times/Time">
        <th>
            <xsl:value-of select="@Id"/>
        </th>
        </xsl:for-each>
    </tr>
    <!-- for each teacher -->
    <xsl:for-each
select="/HighSchoolTimetableArchive/Instances/Instance/Resources/Resource">
        <xsl:if test="ResourceType/@Reference='Teacher'">
            <tr>
                <td>
                    <xsl:value-of select="Name"/>
                </td>
                <xsl:call-template name="findRow">
                    <xsl:with-param name="myRow" select="."/>
                </xsl:call-template>
            </tr>
        </xsl:if>
    </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
<!-- ***** -->

```

The image shows a Safari browser window displaying a large data table. The browser's address bar shows 'localhost'. The table has columns for days of the week (Mo_1 to Th_1) and rows for schools (ANT, HAA, HAN, HIL, HIM, JAA). Each cell in the table contains an event ID, such as 'Event_C001_1', 'Event_C002_1', etc. The table is organized into a grid with 24 columns and 6 rows of data.

Εικόνα Παράρτημα Α – 1: Έξοδος (output) του xslt προγράμματος. Παρουσιάζεται μόνο ένα τμήμα. Το πρόβλημα είναι από ένα φινλανδικό σχολείο, από το dataset για τον διαγωνισμό high school timetabling.

BIBΛΙΟΓΡΑΦΙΑ

- [Abr91] D. Abramson. Constructing School Timetables using Simulated Annealing: Sequential and Parallel Algorithms. *Management Science*, Vol 37, No 1, Jan 1991, pp 98 - 113.
- [AB93] Abderrahmane Aggoun and Nicolas Beldiceanu. Extending CHIP in order to solve complex scheduling and placement problems. *Mathematical and Computer Modelling*, 17(7):57–73, 1993.
- [AS05] Alper Atamturk, Martin W. P. Savelsbergh. Integer-Programming Software Systems. *Annals of Operations Research* 140, 67–124, 2005. Springer Science + Business Media, Inc. Manufactured in The Netherlands.
- [BV92] Maurice Bruynooghe and Raf Venken. Backtracking. In Stuart C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, pages 84–88. Wiley, 1992. Volume 1, second edition.
- [BKS03] E.K. Burke, G. Kendall and E. Soubeiga. A Tabu-Search Hyperheuristic for Timetabling and Rostering. *Journal of Heuristics*, 9: 451–470, 2003.
- [BMP08] E. K. Burke, J. Marecek, A.J. Parkes, H. Rudova. A Branch-and-cut Procedure for the Udine Course Timetabling Problem. June 2008.
- [CBD10] Der-San Chen, Robert G. Batson, Yu Dang. 2010. *Applied Integer Programming. Modeling and Solution*. Ch14, p353-389. Wiley. A John Wiley & Sons, Inc. Publication.
- [CDM92] Colomi A., Dorigo M., Maniezzo V. (1992) Genetic Algorithms: A New Approach to the Timetable Problem. In: Akgül M., Hamacher H.W., Tüfekçi S. (eds) *Combinatorial Optimization*. NATO ASI Series (Series F: Computer and Systems Sciences), vol 82. Springer, Berlin, Heidelberg
- [DBH10] S. Daskalaki, T. Birbas, E. Housos. An integer programming formulation for a case study in university timetabling. Elsevier, *European Journal of Operational Research* 153 (2004) 117–135 Jeffrey H. Kingston. “Solving the General High School Timetabling Problem”, (PATAT 2010)
- [DM17] Emir Demirovic and Nysret Musliu. MaxSAT-Based Large Neighborhood Search for High School Timetabling. *Journal Computers and Operations Research archive*, Volume 78 Issue C, February 2017, Pages 172-180, Elsevier Science Ltd.

- [DPPC00] Ulrich Dorndorf, Erwin Pesch, and Toa`n Phan-Huy. Constraint propagation techniques for the disjunctive scheduling problem. *Artificial Intelligence*, 122(1–2):189–240, 2000
- [Fon17] George H.G. Fonseca. *Formulations and Algorithms for Timetabling*. Ph.D. thesis.
- [FS13] George H. G. Fonseca, Haroldo G. Santos. Memetic Algorithms to the High School Timetabling Problem. *IEEE Xplore*, 15 July 2013
- [FDH14] George H.G. Fonseca · Thaise D. Delfino · Haroldo G. Santos. A Web-Software to handle XHSTT Timetabling Problems. 10th International Conference of the Practice and Theory of Automated Timetabling PATAT 2014, 26-29 August 2014, York, United Kingdom
- [FHC16a] George H.G. Fonseca, Haroldo G. Santos, Eduardo G. Carrano. Integrating matheuristics and metaheuristics for timetabling. *ELSEVIER, Computers & Operations Research* 74 (2016) 108–117
- [FHC16b] George H.G. Fonseca · Haroldo G. Santos · Eduardo G. Carrano · Thomas J.R. Stidsen. Modelling and Solving University Course Timetabling Problems Through XHSTT. 11th International Conference on the Practice and Theory of Automated Timetabling. Udine, Italy, Aug 2016 (PATAP 2016)
- [Fri17] Joshua S. Friedman. Automated timetabling for small colleges and high schools using huge integer programs. Cornell University Library, *Artificial Intelligence*. 4 Jan 2017. arXiv:1612.08777v2 [cs.AI].
- [GL99] Gacs P. Boston University, Lovasz, L, Yale University. *Complexity of Algorithms, Lecture Notes*, Spring 1999
- [Gog12] Γκόγκος Ν. Χρήστος, 2009. Αλγόριθμοι συνδυαστικής βελτιστοποίησης με έμφαση σε μεταερευτικές τεχνικές. Διδακτορική διατριβή. Πανεπιστήμιο Πατρών. Τμήμα Ηλεκτρολόγων μηχανικών και Τεχνολογίας Υπολογιστών.
- [GPV] Sandor Gyori, Zoltan Petres, Annamaria R. Varkonyi-Koczy. *Genetic Algorithms in Timetabling. A New Approach*. Budapest University of Technology and Economics Department of Measurement and Information Systems Mu`egyetem rkp. 9., Budapest, Hungary, H-1521
- [Jam04] Marcone Jamilson, Freitas Souza, Nelson Maculan, Luis Satoru Ochi, A GRASP-TABU search algorithm for solving school timetabling problems. *Springer Link*, 2004, series 1, volume 86, pp 659-672.

- [Kin16] Jeffrey H. Kingston. The KHE High School Timetabling Engine. Version of 6 February 2016.
- [KNKN10] Khang Nguyen Tan Tran Minh, Nguyen Dang Thi Thanh, Khon Trieu Trang, Nuong Tran Thi Hue, “Using Tabu Search for Solving a High School Timetabling Problem“, *Advances in Intelligent Information and Database Systems* pp 305-313. Part of the *Studies in Computational Intelligence* book series (SCI, volume 283). Springer-Verlag Berlin Heidelberg 2010
- [Kri15] S. Kristiansen, M. Sørensen, T. R. Stidsen. Integer Programming for the Generalized (High) School Timetabling Problem, *ACM, Journal of Scheduling* archive, Volume 18 Issue 4, August 2015, Pages 377-392.
- [Lib08] Leo Liberti. *Introduction to Global Optimization*. LIX, École Polytechnique, Palaiseau F-91128, France. 2008.
- [MLG13] D. Manjarres, I. Landa-Torres, S. Gil-Lopez, J. Del Ser, M.N. Bilbao, S. Salcedo-Sanz, Z.W. Geem. A survey on applications of the harmony search algorithm. *Elsevier, Engineering Applications of Artificial Intelligence* 26 (2013) 1818–1831.
- [MS98] Kim Marriott and Peter Stuckey. *Programming with Constraints: An Introduction*. MIT Press, 1998.
- [MT00] Kurt Mehlhorn and Sven Thiel. Faster algorithms for bound- consistency of the sortedness and the alldifferent constraint. In Rina Dechter, editor, *Sixth International Conference on Principles and Practice of Constraint Programming*, LNCS 1894, pages 306–319. Springer, 2000.
- [Nan12] Anirudha Nanda, Manisha P. Pai, and Abhijeet Gole. An Algorithm to Automatically Generate Schedule for School Lectures Using a Heuristic Approach. *International Journal of Machine Learning and Computing*, Vol. 2, No. 4, Aug 2012
- [PATAT10] *Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling*. 10 - 13 August 2010 (Queen’s University of Belfast). Edited by: Barry McCollum, Queen’s University Belfast, UK
Edmund, Burke, University of Nottingham, UK
George White, University of Ottawa, Canada.
Published by Queen’s University Belfast.

- [PCLF11] J.A. Pajero, A. Ruiz-Cortes, S. Lozano and P. Fernandez. Metaheuristic optimization frameworks: a survey and benchmarking. Springer-Verlag 2011. Soft Comput DOI 10.1007/s00500-011-0754-8
- [Pil14] Nelishia Pillay. A survey of school timetabling research. Annals of Operations Research, July 2014, Volume 218, Issue 1, pp 261–293
- [Post12a] Gerhard Post, Samad Ahmadi, and Frederik Geertsema. Cyclic Transfers in School Timetabling. Springer Link, January 2012, Volume 34, Issue 1, pp 133–154
- [Post12b] Gerhard Post, Luca Di Gaspero, Jeffrey H. Kingston, Barry McCollum, and Andrea Schaerf. The Third International Timetabling Competition. Practice and Theory of Automated Timetabling (PATAT 2012), 29-31 August 2012, Son, Norway.
- [Post12c] Gerhard Post, Jeffrey H. Kingston, Samad Ahmadi, Sophia Daskalaki, Christos Gogos, Jari Kyngas, Cimmo Nurmi, Nysret Musliu, Nelishia Pillay, Haroldo Santos, Andrea Schaerf. XHSTT: An XML archive for high school timetabling problems in different countries. Ann Oper Res (2014) 218:295–301 DOI 10.1007/s10479-011-1012-2.
- [Sha01] Jeremy F. Shapiro. Modeling the Supply Chain (Suxbury Applied). South-Western College Pub. 2001.
- [Sti14] K. S. Stidsen, T.R. & Herold, M. B. (2014). Solving Multiple Timetabling Problems at Danish High Schools. Department of Management Engineering, Technical University of Denmark.
- [Tsa93] Edward Tsang. Foundations of Constraint Satisfaction. Academic Press, 1993
- [Val12] Christos Valouxis, Christos Gogos., Panayiotis Alefragis and Efthymios Housos. “Decomposing the High School TimeTable Problem”. Practice and Theory of Automated Timetabling. PATAT 2012, 29-31 August 2012, Son Norway:209-221.
- [vH01] W. J. van Hoeve. The alldifferent constraint: A survey. In 6th Annual Workshop of the ERCIM Working Group on Constraints, 2001.
- [Wid06] Avi Wigderson, P, NP and mathematics – a computational complexity perspective, 2006