



ΠΑΝΕΠΙΣΤΗΜΙΟ ΣΤΕΡΕΑΣ ΕΛΛΑΔΑΣ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ

## Αυτόματη Δημιουργία Ωρολογίων Προγραμμάτων με Χρήση Μεθόδων Υπολογιστικής Νοημοσύνης

Ελένη Ορφανίδου

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ  
Υπεύθυνος  
Βασίλειος Π. Πλαγιανάκος  
Επίκουρος Καθηγητής

Λαμία, 2012

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Κατ'αρχάς, θα ήθελα να ευχαριστήσω θερμά τον κ. Πλαγιανάκο, για την καθοδήγηση του και την βοήθεια του σε κάθε βήμα της εργασίας, όπως και για την κατανόηση του στα προβλήματα που αντιμετώπιζα κατά καιρούς. Επίσης, θα ήθελα να ευχαριστήσω το Μιχάλη Επιτροπάκη για τη συνεισφορά του σε ένα προβληματικό σημείο της εργασίας και τελευταίο αλλά όχι λιγότερο σημαντικό ένα μεγάλο ευχαριστώ στην οικογένεια μου για την συνεχή υποστήριξη της υλικά και συναισθηματικά κατά τη διάρκεια των σπουδών μου.

## ΣΥΝΟΨΗ

Η παρούσα πτυχιακή εργασία μελετά το πρόβλημα της δημιουργίας ωρολογίων προγραμμάτων που το συναντάμε σε εκπαιδευτικούς και μη οργανισμούς καθώς και την επίλυση του από τους γενετικούς αλγόριθμους που ανήκουν στο χώρο της Υπολογιστικής Νοημοσύνης. Επίσης, παρουσιάζεται το λογισμικό FET, που δημιουργεί αυτόματα ωρολόγια προγράμματα, παραμετροποιώντας το για τις απαιτήσεις του τμήματος Πληροφορικής με Εφαρμογές στη Βιοϊατρική. Τέλος, εξηγείται ο τρόπος κατασκευής μιας ωρολόγιας διαδικτυακής εφαρμογής για τη συλλογή των περιορισμών του παραπάνω τμήματος.

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:** Δημιουργία Ωρολογίων Προγραμμάτων, Υπολογιστική Νοημοσύνη, Γενετικοί Αλγόριθμοι, FET, Ωρολόγια Διαδικτυακή Εφαρμογή.

## ABSTRACT

This paper studies the timetabling problem, which is presented in educational organisations, as well as the genetic algorithms from Computational Intelligence who find solutions in this problem. Also, it presents the software FET, which generates timetables, and parameterizes the requirements for department of Informatics with Applications in Biomedical. Finally, it explains the way of construction one timetabling website which is collecting constraints for the above department.

**KEY WORDS:** Timetabling, Computational Intelligence, Genetic Algorithms, FET, Timetabling Website.

Στο δάσκαλο μου και στην οικογένεια μου  
για τη στήριξη και την υπομονή τους.

# ΠΕΡΙΕΧΟΜΕΝΑ

Ευχαριστίες.....	2
Σύνοψη.....	3
Abstract.....	3
Περιεχόμενα.....	5
Λίστα Πινάκων.....	7
Λίστα Εικόνων.....	8
Πρόλογος.....	10

## I Εισαγωγή

<b>1. Πρόβλημα δημιουργίας ωρολογίων προγραμμάτων.....</b>	<b>12</b>
1.1 Τι είναι η δημιουργία ωρολογίων προγραμμάτων (timetabling).....	12
1.2 Η χρήση μεθόδων Υπολογιστικής Νοημοσύνης.....	14
1.2.1 Τεχνητά Νευρωνικά Δίκτυα.....	15
1.2.2 Ασαφής Λογική.....	20
1.2.3 Εξελικτικοί Αλγόριθμοι.....	22

## II Η συμβολή των Γενετικών Αλγορίθμων

<b>2. Οι Γενετικοί Αλγόριθμοι.....</b>	<b>27</b>
2.1 Εισαγωγή στους Γενετικούς Αλγορίθμους.....	27
2.2 Βασικές Έννοιες.....	28
2.2.1 Γονίδια και Χρωμοσώματα.....	28
2.2.2 Γονείς.....	29
2.2.3 Πληθυσμοί και Γενιές.....	29
2.2.4 Παιδιά.....	31
2.2.5 Συνάρτηση Καταλληλότητας.....	31
2.3 Αποτελεσματικότητα και Αποδοτικότητα.....	32
2.4 Εφαρμογές.....	33
2.4.1 Εύρεση μέγιστης τιμής αριθμητικών συναρτήσεων.....	33
2.4.2 Συνδυαστική βελτιστοποίηση.....	33
2.5 Τα βήματα του βασικού Γενετικού Αλγορίθμου.....	34
2.6 Παράμετροι ενός γενετικού αλγορίθμου.....	34
2.6.1 Κωδικοποίηση ενός χρωμοσώματος.....	34
2.6.2 Διασταύρωση.....	35
2.6.3 Μετάλλαξη.....	37
2.6.4 Πιθανότητα διασταύρωσης και μετάλλαξης.....	37
2.6.5 Μέγεθος πληθυσμού.....	37
2.6.6 Επιλογή.....	38
<b>3. Οι Γενετικοί Αλγόριθμοι στη δημιουργία ωρολογίων προγραμμάτων.....</b>	<b>39</b>
3.1 Το κλασικό πρόβλημα χρονοπρογραμματισμού.....	39
3.2 Περιορισμοί.....	40
3.2.1 Περιορισμοί ακμών.....	40
3.2.2 Περιορισμοί διάταξης.....	40
3.2.3 Περιορισμοί γεγονότων που καλύπτουν πολλές σχισμές.....	41
3.2.4 Προκαθορισμένα γεγονότα και εξαιρέσεις.....	41
3.2.5 Περιορισμοί χωρητικότητας.....	41
3.2.6 Σκληροί και χαλαροί περιορισμοί.....	42
3.3 Γενετικοί αλγόριθμοι και ωρολόγια προγράμματα.....	42

3.3.1 Αναπαράσταση.....	42
3.3.2 Χειρισμός περιορισμών.....	43
3.3.3 Εκτέλεση αλγορίθμου.....	44

### **III Μελέτη της εφαρμογής FET καθώς και άλλων ωρολόγιων εφαρμογών**

<b>4. Η ωρολόγια εφαρμογή FET και η παραμετροποίηση της.....</b>	<b>48</b>
4.1 Περιγραφή της ωρολόγιας εφαρμογής FET.....	48
4.1.1 Χαρακτηριστικά του FET.....	48
4.2 Παραμετροποίηση της ωρολόγιας εφαρμογής FET για το τμήμα της Πληροφορικής με Εφαρμογές στη Βιοϊατρική.....	49
4.2.1 Χειμερινό Εξάμηνο 2010 – 2011.....	50
4.2.2 Εαρινό Εξάμηνο 2010 – 2011.....	81
4.2.3 Παρατηρήσεις.....	94
<b>5. Εναλλακτικές εφαρμογές ωρολόγιων προγραμμάτων.....</b>	<b>95</b>
5.1 aSc Timetables.....	95
5.2 Lantiv Timetabler.....	97
5.3 Untis.....	98

### **IV Ωρολόγια Διαδικτυακή Εφαρμογή**

<b>6. Ωρολόγια διαδικτυακή εφαρμογή.....</b>	<b>102</b>
6.1 Σκοπός της δημιουργίας ωρολόγιας διαδικτυακής εφαρμογής.....	102
6.2 Περιγραφή της ωρολόγιας διαδικτυακής εφαρμογής.....	102
6.2.1 Εισαγωγή νέων εγγραφών στη βάση δεδομένων.....	104
6.2.2 Ωρολόγια διαδικτυακή εφαρμογή.....	108
6.2.3 Παραγωγή αρχείου fet.....	114

### **V Συμπεράσματα**

<b>7. Συμπεράσματα.....</b>	<b>119</b>
7.1 Συμπεράσματα.....	119
7.2 Μελλοντικές Προεκτάσεις.....	119

<b>Παράρτημα.....</b>	<b>121</b>
<b>Βιβλιογραφία.....</b>	<b>149</b>

## ΛΙΣΤΑ ΠΙΝΑΚΩΝ

1.1 Παράδειγμα Μαθητές και Διαγωνισμοί.....	13
1.2 Παράδειγμα κενού ωρολογίου προγράμματος.....	13
1.3 Παράδειγμα πιθανού ωρολογίου προγράμματος.....	13
4.1 Δραστηριότητες Χειμερινού Εξαμήνου.....	53
4.2 Διδάσκοντες-Μη Διαθέσιμες ημέρες για το χειμερινό εξάμηνο.....	64
4.3 Μαθήματα και Χώροι Προτίμησης για το χειμερινό εξάμηνο.....	66
4.4 Δραστηριότητες και Χώροι Προτίμησης για το χειμερινό εξάμηνο.....	67
4.5 Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων - Α' Εξάμηνο.....	77
4.6 Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων – Γ' Εξάμηνο.....	78
4.7 Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων – Ε' Εξάμηνο.....	79
4.8 Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων – Ζ' Εξάμηνο.....	80
4.9 Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων – Ελεύθερη Επιλογή.....	81
4.10 Δραστηριότητες Εαρινού Εξαμήνου.....	82
4.11 Διδάσκοντες-Μη Διαθέσιμες ημέρες για το εαρινό εξάμηνο.....	85
4.12 Μαθήματα και Χώροι Προτίμησης για το εαρινό εξάμηνο.....	86
4.13 Δραστηριότητες και Χώροι Προτίμησης για το εαρινό εξάμηνο.....	86
4.14 Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων – Β' Εξάμηνο.....	90
4.15 Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων – Δ' Εξάμηνο.....	91
4.16 Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων – ΣΤ' Εξάμηνο.....	92
4.17 Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων – Η' Εξάμηνο.....	93
4.18 Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων – Ελεύθερη Επιλογή.....	94

## ΛΙΣΤΑ ΕΙΚΟΝΩΝ

1.1 Αναπαράσταση βιολογικού νευρώνα.....	16
1.2 Αναπαράσταση τεχνητού νευρώνα.....	17
1.3 Αναπαράσταση τεχνητού νευρωνικού δικτύου.....	18
1.4 Συνάρτηση συγγένειας για τα ασαφή σύνολα «κοντός», «μεσαίος», «ψηλός».....	21
1.5 Ψευδοκώδικας Εξελικτικού Αλγορίθμου.....	23
1.6 Εξελικτικός Κύκλος.....	23
2.1 Κωδικοποίηση Δένδρου.....	35
2.2 Κωδικοποίηση Τιμών.....	35
2.3 Διασταύρωση Ενός Σημείου.....	36
2.4 Πολυμεσική Διασταύρωση.....	36
2.5 Μετάλλαξη.....	37
4.1 Μενού→File→New.....	53
4.2 Δημιουργία αρχείου για το χειμερινό εξάμηνο.....	54
4.3 Μενού→Data→Basic→Institution information.....	54
4.4 Institution name-Όνομα πανεπιστημίου.....	54
4.5 Comments-Σχόλια.....	55
4.6 Μενού→Data→Basic→Days and hours.....	55
4.7 Days-Ημέρες.....	56
4.8 Hours-Ώρες.....	57
4.9 Μενού_Data.....	57
4.10 Teachers-Διδάσκοντες.....	58
4.11 Μενού→Data→Students.....	59
4.12 Ομάδες φοιτητών.....	59
4.13 Subjects-Μαθήματα Χειμερινού Εξαμήνου.....	60
4.14 Space-Χώροι Φοίτησης.....	60
4.15 Προσθήκη κτηρίου.....	61
4.16 Buildings-Κτήρια.....	61
4.17 Προσθήκη αίθουσας.....	62
4.18 Rooms-Αίθουσες και Εργαστήρια.....	62
4.19 Γραμμική Άλγεβρα-Αδάμ-Πρώτο Έτος.....	63
4.20 Μενού_Time.....	65
4.21 Breaks-Διαλείμματα.....	66
4.22 Μενού→Time→Teachers.....	68
4.23 Πότε η κα Αδάμ δεν είναι διαθέσιμη.....	69
4.24 Μενού_Space.....	69
4.25 Space→Rooms→A room's not available times.....	70
4.26 Διαθεσιμότητα Αίθουσας 1.....	70
4.27 Space→Teachers→A teacher has a home room.....	71
4.28 Αδάμ-Αίθουσα 1.....	71
4.29 Διδάσκοντες και αίθουσες προτίμησης.....	72
4.30 Space→Students→A set of students has a home room.....	72
4.31 Πρώτο Έτος_Αίθουσα 1.....	73
4.32 Δεύτερο Έτος_Αίθουσα 2.....	73
4.33 Τρίτο Έτος_Αίθουσα 3.....	74
4.34 Space→Subjects→A subject has a preferred room.....	74



4.35	Αριθμητική Ανάλυση Ε_Εργαστήριο 1.....	75
4.36	Μαθήματα και Προτεινόμενοι Χώροι Φοίτησης.....	75
4.37	Space→Activities→An activity has a preferred room.....	76
4.38	(1 ώρα-Παπανάγνου-Ανατομία & Φυσιολογία Ι-Δεύτερο Έτος)_Εργαστήριο Ανατομίας.....	76
4.39	Δημιουργία αρχείου για το εαρινό εξάμηνο.....	87
4.40	Διδάσκοντες του εαρινού εξαμήνου.....	88
4.41	Subjects-Μαθήματα Εαρινού Εξαμήνου.....	89
4.42	Βασιλάκη-Αίθουσα 1.....	89
5.1	Στιγμιότυπα λειτουργίας του wizard.....	96
5.2	Ωρολόγιο πρόγραμμα στο aSc Timetables.....	97
5.3	Ωρολόγιο πρόγραμμα στο Lantiv Timetabler.....	98
5.4	Μενού του Untis.....	99
5.5	Ωρολόγιο πρόγραμμα στο Untis.....	100
6.1	phpMyAdmin.....	103
6.2	Βάση Δεδομένων timetable και πίνακας nametable.....	103
6.3	Πεδία του πίνακα nametable.....	104
6.4	index.php.....	105
6.5	Φόρμα Συμπλήρωσης.....	105
6.6	config.php.....	106
6.7	insert.php.....	107
6.8	Εγγραφές της ΒΔ.....	107
6.9	Είσοδος στο Ωρολόγιο Πρόγραμμα.....	108
6.10	Ωρολόγιο Πρόγραμμα.....	109
6.11	Τέλος Ωρολογίου Προγράμματος.....	109
6.12	forma.php.....	110
6.13	forma2.php (Μέρος 1).....	111
6.14	forma2.php (Μέρος 2).....	111
6.15	forma2.php (Μέρος 3).....	112
6.16	forma3.php (Μέρος 1).....	113
6.17	forma3.php (Μέρος 2).....	113
6.18	generate.php (Μέρος 1).....	115
6.19	generate.php (Μέρος 2).....	116
6.20	generate.php (Μέρος 3).....	116
6.21	generate.php (Μέρος 4).....	117
6.22	Παραγωγή αρχείου fet.....	117

# ΠΡΟΛΟΓΟΣ

Συχνό πρόβλημα στα εκπαιδευτικά ιδρύματα και όχι μόνο είναι η δημιουργία ωρολογίων προγραμμάτων, καθώς απαιτείται αρκετός κόπος και χρόνος για να συλλεχθούν δεδομένα και να ρυθμιστούν περιορισμοί που αφορούν τους διδάσκοντες, τις ώρες διδασκαλίας και τους χώρους μάθησης. Για αυτόν τον λόγο αναπτύχθηκαν πολλές μέθοδοι που λύνουν το πρόβλημα αυτό γρήγορα και εύκολα όπως η ταμπού αναζήτηση, η εξομοιωμένη ανόπτηση, οι γενετικοί αλγόριθμοι, ο προγραμματισμός με περιορισμούς και άλλες.

Σκοπός της παρούσας πτυχιακής εργασίας είναι:

- να περιγράψει τις μεθόδους Υπολογιστικής Νοημοσύνης, να ξεχωρίσει μία από αυτές και να αναλύσει τον τρόπο με τον οποίο αυτή λύνει το πρόβλημα,
- να παρουσιάσει μία ωρολόγια εφαρμογή και να την παραμετροποιήσει για ένα εκπαιδευτικό ίδρυμα και
- να δημιουργήσει μία ωρολόγια διαδικτυακή εφαρμογή η οποία θα διευκολύνει την δημιουργία ωρολογίων προγραμμάτων.

Τέλος, η δομή της πτυχιακής εργασίας είναι:

- στο 1<sup>ο</sup> κεφάλαιο γίνεται αναφορά στο πρόβλημα της δημιουργίας ωρολογίων προγραμμάτων και στις βασικές κατηγορίες αλγορίθμων της Υπολογιστικής Νοημοσύνης,
- στο 2<sup>ο</sup> κεφάλαιο περιγράφονται οι Γενετικοί Αλγόριθμοι και τα χαρακτηριστικά τους,
- στο 3<sup>ο</sup> κεφάλαιο αναλύεται το πρόβλημα της δημιουργίας ωρολογίων προγραμμάτων με τη βοήθεια των Γενετικών Αλγορίθμων,
- στο 4<sup>ο</sup> κεφάλαιο μελετάται η ωρολόγια εφαρμογή FET και παραμετροποιείται για το τμήμα της Πληροφορικής με Εφαρμογές στη Βιοϊατρική του Πανεπιστημίου Στερεάς Ελλάδας,
- στο 5<sup>ο</sup> κεφάλαιο παρουσιάζονται κάποιες εναλλακτικές εφαρμογές ωρολογίων προγραμμάτων και συγκρίνονται με την εφαρμογή FET,
- στο 6<sup>ο</sup> κεφάλαιο περιγράφεται η ωρολόγια διαδικτυακή εφαρμογή που έχει δημιουργηθεί για τις απαιτήσεις του τμήματος και τέλος
- στο 7<sup>ο</sup> κεφάλαιο παρουσιάζονται τα συμπεράσματα του φοιτητή για την πτυχιακή εργασία.

**ΜΕΡΟΣ Ι**  
**Εισαγωγή**

# ΚΕΦΑΛΑΙΟ 1

## Πρόβλημα δημιουργίας ωρολογίων προγραμμάτων

*Τα προβλήματα που υπάρχουν σήμερα  
στον κόσμο δεν μπορούν να λυθούν  
με το επίπεδο σκέψης που τα δημιούργησε.*

*~Albert Einstein (1879-1955)*

### 1.1 Τι είναι η δημιουργία ωρολογίων προγραμμάτων (timetabling)

Η δημιουργία ωρολογίων προγραμμάτων είναι ένα είδος προβλήματος στο οποίο χρειάζεται να προγραμματιστούν διάφορες δραστηριότητες ή γεγονότα στον κατάλληλο χρόνο, δεδομένου διάφορων πόρων και σύμφωνα πάντα με κάποιους περιορισμούς. Για την ακρίβεια, είναι ένα τυπικό πρόβλημα χρονοπρογραμματισμού (scheduling problem), το οποίο αξίζει να σημειωθεί για την πολυπλοκότητα του καθώς κατατάσσεται στα NP πλήρη προβλήματα. Αυτό σημαίνει πως δεν έχει βρεθεί κανένας αλγόριθμος που να το λύνει σε πολυωνυμικό χρόνο.

Όσον αφορά τη δημιουργία ωρολογίων προγραμμάτων για την εκπαίδευση, προσπαθούμε να τοποθετήσουμε μέσα στο χρόνο ένα δεδομένο σύνολο από συναντήσεις μεταξύ μαθητών και καθηγητών, έτσι ώστε το πρόγραμμα να είναι αποδεκτό από όλους τους εμπλεκόμενους. Βασικός μας στόχος είναι να γίνεται σεβαστή η χωρητικότητα κάθε πόρου (μαθητές, καθηγητές, αίθουσες κτλ) σε κάθε χρονική στιγμή, για αυτό και δίνουμε μεγάλη προσοχή στους περιορισμούς. Οι περιορισμοί συνήθως διακρίνονται σε "μαλακούς" και "σκληρούς". Οι μεν σκληροί (hard) είναι αυστηρά επιβαλλόμενοι και χρειάζεται άμεσα να ικανοποιηθούν, οι δε μαλακοί (soft) είναι επιθυμητοί αλλά όχι εντελώς απαραίτητοι. Για παράδειγμα, σκληρός περιορισμός θα ήταν ότι κανένας καθηγητής δεν μπορεί να δίνει δύο διαλέξεις ταυτόχρονα, ενώ μαλακός περιορισμός θα ήταν ότι ο X καθηγητής επιθυμεί να ξεκινά τις διαλέξεις του μετά τις 10:00 το πρωί.

Επίσης, παρατηρούμε πως υπάρχουν δύο είδη ωρολόγιων προγραμμάτων, το ωρολόγιο πρόγραμμα εξετάσεων και το ωρολόγιο πρόγραμμα μαθημάτων. Ένα απλό παράδειγμα ενός προβλήματος ωρολόγιου προγράμματος εξετάσεων είναι το ακόλουθο: Υποθέστε ότι υπάρχουν 10 μαθητές (s1..s10), 6 εξετάσεις (e1..e6) και 8 χρονικά διαστήματα (t1..t8). Τα χρονικά διαστήματα από t1..t4 εμφανίζονται την 1η ημέρα και τα χρονικά διαστήματα από t5..t8 εμφανίζονται την 2η ημέρα. Κάθε ημέρα έχει 4 χρονικά διαστήματα, 2 το πρωί και 2 το απόγευμα. Κάθε μαθητής δίνει διαφορετικό αριθμό εξετάσεων όπως παρουσιάζεται στον πίνακα 1.1, όπου ο μαθητής 1 δίνει τις εξετάσεις 1,2,4,6, ο μαθητής 2 δίνει τις εξετάσεις 1,2,4,5 κ.ο.κ. Επίσης, παρουσιάζεται ένα κενό ωρολόγιο πρόγραμμα (timetable) στον πίνακα 1.2, όπου κάθε χρονικό διάστημα έχει το προσδιοριστικό του.

Μαθητής	Διαγωνισμός
s1	e1, e2, e4, e6
s2	e1, e2, e4, e5
s3	e1, e2, e4, e5
s4	e1, e2, e5
s5	e4
s6	e4, e5
s7	e2, e4
s8	e2, e4, e6
s9	e3, e4, e6
s10	e3, e4, e6

Πίνακας 1.1: Παράδειγμα Μαθητές και Διαγωνισμοί

Ημέρα	Προ Μεσημβρίας		Μετά Μεσημβρίας	
	09:30 – 11:00	11:30 – 13:00	14:00 – 15:30	16:00 – 17:30
Ημέρα 1	t1	t2	t3	t4
Ημέρα 2	t5	t6	t7	t8

Πίνακας 1.2: Παράδειγμα κενού ωρολόγιου προγράμματος

Σε αυτό το παράδειγμα σκληροί περιορισμοί είναι ότι ο μαθητής δεν μπορεί να δίνει δύο διαφορετικές εξετάσεις στο ίδιο χρονικό διάστημα και τρεις διαφορετικές εξετάσεις την ίδια ημέρα. Ενώ μαλακός περιορισμός είναι ότι ο ίδιος μαθητής δεν μπορεί να δίνει δύο συνεχόμενες εξετάσεις. Έτσι ένα πιθανό ωρολόγιο πρόγραμμα θα μπορούσε να είναι όπως στον πίνακα 1.3 όπου οι εξετάσεις e1, e2, e3, e4, e5 και e6 τοποθετούνται στα χρονικά διαστήματα t1, t5, t2, t4, t7 και t8 αντίστοιχα.

Ημέρα	Προ Μεσημβρίας		Μετά Μεσημβρίας	
	09:30 – 11:00	11:30 – 13:00	14:00 – 15:30	16:00 – 17:30
Ημέρα 1	e1	e3		e4
Ημέρα 2	e2		e5	e6

Πίνακας 1.3: Παράδειγμα πιθανού ωρολόγιου προγράμματος

## 1.2 Η χρήση μεθόδων Υπολογιστικής Νοημοσύνης

Ο όρος Υπολογιστική Νοημοσύνη (Computational Intelligence) αναφέρεται στον κλάδο της Επιστήμης των Υπολογιστών ο οποίος ασχολείται με μεθόδους που μιμούνται βιολογικές διεργασίες, όπως η διαδικασία της εξέλιξης των ειδών ή η λειτουργία του εγκεφάλου. Παραδείγματα τέτοιων μεθόδων είναι τα Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks), η Ασαφής Λογική (Fuzzy Logic) και οι Εξελικτικοί Αλγόριθμοι (Evolutionary Algorithms). Αυτές οι μέθοδοι καθώς και ο συνδυασμός αυτών χρησιμοποιούνται συχνά για να λύσουν προβλήματα βελτιστοποίησης όπως αυτό της δημιουργίας ωρολογίων προγραμμάτων (timetabling) και του χρονοπρογραμματισμού (scheduling).

Ο χώρος της βελτιστοποίησης στα εφαρμοσμένα μαθηματικά αναφέρεται στην αναζήτηση βέλτιστων παραμέτρων ενός - συνήθως περίπλοκου - συστήματος. Προβλήματα βελτιστοποίησης απαντώνται σε πολλά επιστημονικά πεδία όπως π.χ. στη φυσική, στη χημεία, στην οικονομία κ.α.

Στα μαθηματικά διατυπώνεται ένα πρόβλημα βελτιστοποίησης σαν πρόβλημα ελαχιστοποίησης ή μεγιστοποίησης μιας συνάρτησης μίας μεταβλητής ή πολλών μεταβλητών. Στη μελέτη συναρτήσεων πολλών μεταβλητών χρησιμοποιούνται κυρίως αριθμητικές μέθοδοι για έναν προσεγγιστικό ορισμό ελάχιστων (ή μέγιστων) σημείων, ενώ στην ελαχιστοποίηση (ή μεγιστοποίηση) συναρτήσεων μίας μεταβλητής μπορούν να χρησιμοποιηθούν αναλυτικές και αλγεβρικές μέθοδοι για τον ακριβή ορισμό ελάχιστων (ή μέγιστων).

Για τη βελτιστοποίηση προβλημάτων υπάρχουν διάφορα είδη αλγορίθμων. Η βελτιστοποίηση προβλημάτων είναι τομέας της γενικότερης θεωρίας αλγορίθμων. Η θεωρία αλγορίθμων έχει μεγάλη παράδοση και η ηλικία μερικών αλγορίθμων αριθμεί χιλιάδες χρόνια, όπως για παράδειγμα ο αλγόριθμος του Ευκλείδη για την εύρεση του μέγιστου κοινού διαιρέτη δύο αριθμών ή το λεγόμενο κόσκινο του Ερατοσθένη για την εύρεση των πρώτων αριθμών από 1 ως  $n$ .

Η λέξη αλγόριθμος (algorithm) προέρχεται από μια μελέτη του Πέρση μαθηματικού Abu Ja'far Mohammed ibn Musa al Khowarizmi, που έζησε περί το 825 μ.Χ. Πέντε αιώνες αργότερα η μελέτη αυτή μεταφράστηκε στα λατινικά και άρχισε με την φράση «Algoritmi dixit...» (δηλ. ο αλγόριθμος λέει...).

Ο όρος «αλγόριθμος» επέζησε επί χίλια χρόνια ως σπάνιος όρος, που είχε την έννοια της συστηματικής διαδικασίας αριθμητικών χειρισμών. Την σημερινή του αξία απέκτησε από την αρχή του 20ου αιώνα με την ανάπτυξη της ομώνυμης θεωρίας και φυσικά με την χρήση των ηλεκτρονικών υπολογιστών.

Ο όρος αλγόριθμος χρησιμοποιείται για να δηλώσει μεθόδους που εφαρμόζονται σε προγράμματα για την επίλυση προβλημάτων. Ωστόσο, ένας πιο αναλυτικός ορισμός της έννοιας αυτής είναι ο εξής: «Αλγόριθμος είναι ένα πεπερασμένο σύνολο εντολών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο οι οποίες αν ακολουθηθούν επιτυγχάνεται ένα επιθυμητό αποτέλεσμα.»

Από τον ορισμό προκύπτει ότι για να χαρακτηριστεί μια ακολουθία βημάτων – διαδικασία ως αλγόριθμος, θα πρέπει να έχει τα εξής χαρακτηριστικά:

- Είσοδος: Καμία, μία ή περισσότερες ποσότητες να δίνονται ως είσοδοι στον αλγόριθμο.
- Έξοδος: Ο αλγόριθμος να δημιουργεί τουλάχιστον μια ποσότητα ως αποτέλεσμα.
- Καθορισμός: Κάθε εντολή της διαδικασίας να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της.
- Περαιτότητα: Ο αλγόριθμος να τελειώνει μετά από πεπερασμένα βήματα εκτέλεσης των εντολών του. Μια διαδικασία που δεν τελειώνει μετά από πεπερασμένο αριθμό βημάτων, λέγεται απλώς υπολογιστική διαδικασία.

- **Αποτελεσματικότητα:** Κάθε μεμονωμένη εντολή του αλγορίθμου να είναι απλή. Αυτό σημαίνει ότι μια εντολή δεν αρκεί να έχει ορισθεί, αλλά πρέπει να είναι και εκτελέσιμη.

Σημαντικά χαρακτηριστικά στην μελέτη αλγορίθμων, είναι η εκτίμηση της επίδοσης και της αποδοτικότητάς τους. Ο απλούστερος τρόπος μέτρησης της επίδοσης είναι ο εμπειρικός ή αλλιώς ο λεγόμενος *pesteriori* (εκ των υστέρων). Δηλαδή υλοποιείται ο αλγόριθμος και δίνεται ως ένα σύνολο δεδομένων ώστε να υπολογισθεί ο απαιτούμενος χρόνος επεξεργασίας. Στην μέθοδο αυτή όμως υπάρχει το μειονέκτημα ότι είναι δύσκολο να προβλεφθεί η συμπεριφορά του αλγορίθμου για κάποιο άλλο σύνολο δεδομένων και συνεπώς να συναχθούν λάθος εκτιμήσεις για την επίδοσή του.

Για μια περισσότερο αντικειμενική εκτίμηση των αλγορίθμων έχει εισαχθεί ο όρος της πολυπλοκότητας του αλγορίθμου η οποία εκφράζεται με τον λεγόμενο συμβολισμό  $O$  ( $O$ -notation).

Έτσι δημιουργείται μια ταξινόμηση, με βάση την οποία η θεωρία αλγορίθμων προσπαθεί να βρει αποτελεσματικότερους αλγορίθμους. Η ταξινόμηση που γίνεται στην αφηρημένη θεωρία πολυπλοκότητας χωρίζει τους αλγορίθμους σε κλάσεις πολυπλοκότητας. Αν ένα νέο πρόβλημα αποδειχθεί ότι ανήκει σε μία κλάση για την οποία υπάρχει αποτελεσματικός αλγόριθμος, τότε μπορεί με μια απλή μετατροπή του να λυθεί και το νέο αυτό πρόβλημα.

Σήμερα πολλοί διαφορετικοί αλγόριθμοι είναι γνωστοί για τους οποίους θα κάνουμε δύο διαφορετικές κατηγοριοποιήσεις. Κατ'αρχάς μπορεί κανείς να χωρίσει τους αλγορίθμους σε προσεγγιστικούς αλγορίθμους (*approximation algorithms*) και μη προσεγγιστικούς/εξαντλητικούς. Ένας εξαντλητικός αλγόριθμος βρίσκει οπωσδήποτε βέλτιστη λύση για ένα δοσμένο πρόβλημα βελτιστοποίησης, αν μία τέτοια λύση υπάρχει. Για την εύρεση όμως αυτής της λύσης μπορεί να χρειαστεί πολύ μεγάλος υπολογιστικός χρόνος. Αντιθέτως ένας προσεγγιστικός αλγόριθμος βρίσκει κάποια λύση, αν υπάρχει, σε αποδεκτά πλαίσια χρόνου. Η λύση αυτή ενδέχεται να είναι βέλτιστη, αλλά αυτό δεν είναι εγγυημένο.

Μια διαφορετική κατηγοριοποίηση χωρίζει τους αλγορίθμους σε εποικοδομητικούς (*constructive*), απαριθμητικούς (*enumerative*) και σε επαναληπτικούς (*iterative*). Ένας εποικοδομητικός αλγόριθμος εφαρμόζει ένα απλό σύνολο κανόνων οι οποίοι όμως καθορίζουν ακριβώς το πώς, με ποιιά βήματα και με ποιές ενδιάμεσες λύσεις δημιουργείται μία κατάλληλη τελική λύση, η οποία «χτίζεται» μόνο από τα δεδομένα του προβλήματος. Ένας απαριθμητικός αλγόριθμος εφαρμόζει ένα σχήμα απαρίθμησης που καθορίζει πώς μπορούν να παραχθούν όλες οι λύσεις. Στη συνέχεια ο αλγόριθμος αρχίζει να παράγει αυτές τις λύσεις στην προσπάθεια αναζήτησης της πιο κατάλληλης. Τέλος ένας επαναληπτικός αλγόριθμος αναζητά μία κατάλληλη λύση εναλλάσσοντας επαναληπτικά τη σειρά δρομολόγησης των λειτουργιών μίας αρχικής δοσμένης λύσης.

Παρακάτω παρουσιάζονται οι τρεις βασικές κατηγορίες αλγορίθμων της Υπολογιστικής Νοημοσύνης.

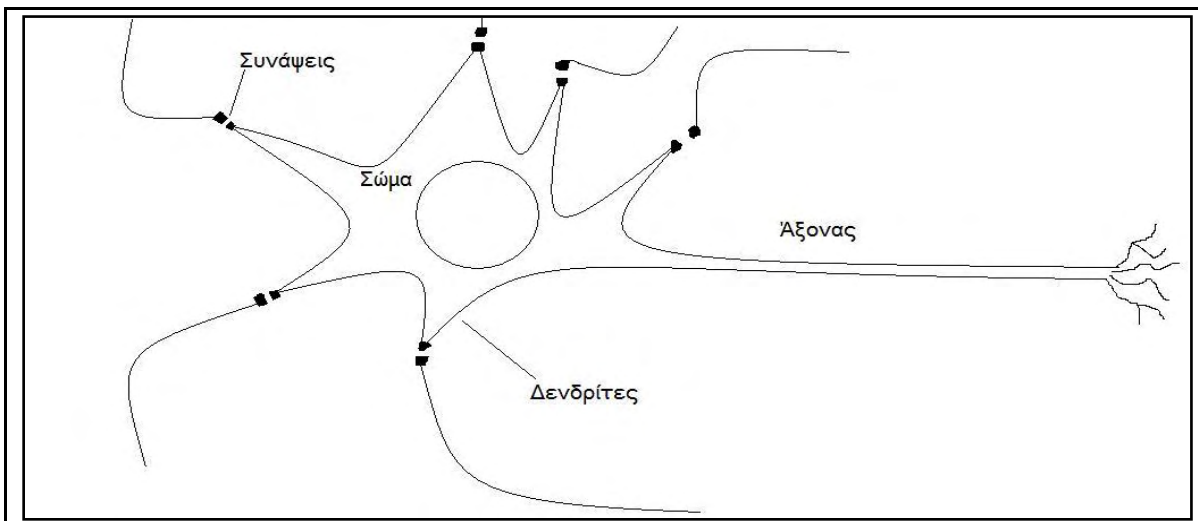
### 1.2.1 Τεχνητά Νευρωνικά Δίκτυα

Τα Τεχνητά Νευρωνικά Δίκτυα [32][33][38][48] είναι αλγόριθμοι οι οποίοι προσπαθούν να μιμηθούν τα νευρωνικά δίκτυα του ανθρώπινου εγκεφάλου και αποτελούνται από τεχνητούς νευρώνες διασυνδεδεμένους μεταξύ τους.

## Βιολογικός Νευρώνας

Όπως είναι γνωστό, η ικανότητα του ανθρώπου να μαθαίνει, να νιώθει, να σκέφτεται, να παίρνει αποφάσεις και να επιλύει προβλήματα βασίζεται στον εγκέφαλο του, ένα όργανο το οποίο αποτελείται από ένα δίκτυο δισεκατομμυρίων νευρώνων που επικοινωνούν μεταξύ τους με συνάψεις. Οι νευρώνες, οι βασικές μονάδες από τις οποίες αποτελείται ο εγκέφαλος, είναι κύτταρα που πραγματοποιούν απλούς υπολογισμούς και επικοινωνούν με παρόμοια κύτταρα.

Η δομή του κάθε νευρώνα αποτελείται από το σώμα, τους δενδρίτες και τον άξονα (εικόνα 1.1). Οι δενδρίτες είναι με το μέρος του νευρώνα που λαμβάνει ηλεκτροχημικά σήματα από τους άλλους νευρώνες μέσω των συνάψεων και φροντίζει να τα διαβιβάσει στο σώμα του νευρώνα μέσω του άξονα. Στο σώμα, η επεξεργασία των σημάτων έχει την μορφή της άθροισης όλων των σημάτων. Όταν αυτό το άθροισμα ξεπερνάει ένα όριο (το λεγόμενο “κατώφλι”) για κάποια δεδομένη χρονική στιγμή, το σώμα παράγει ένα παλμό, ο οποίος και μεταδίδεται στους γειτονικούς νευρώνες μέσω του άξονα, των δενδριτών και των συνάψεων.



Εικόνα 1.1: Αναπαράσταση βιολογικού νευρώνα

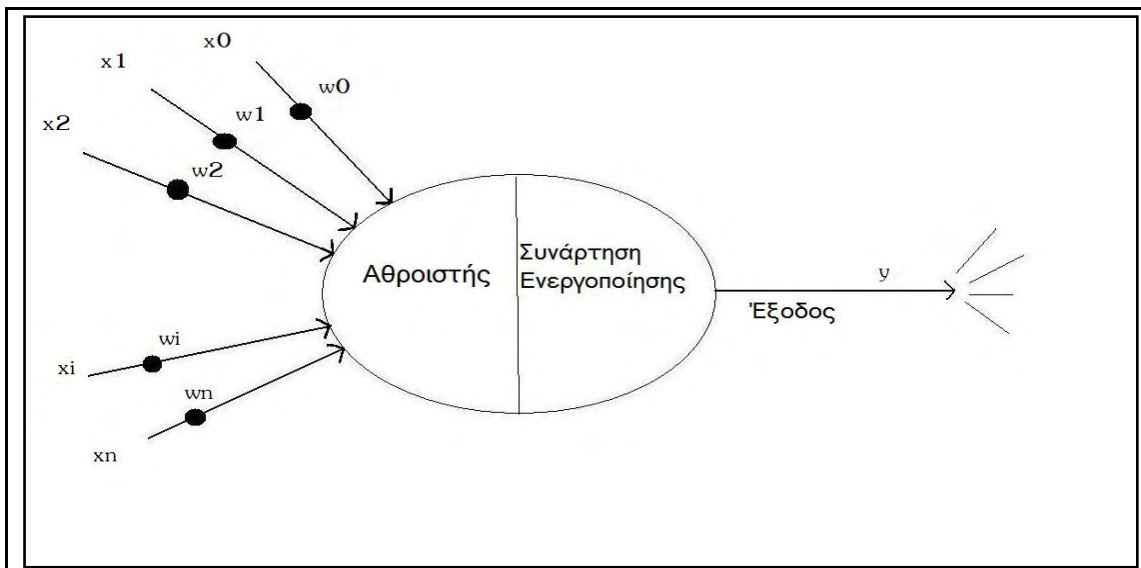
Έχει παρατηρηθεί ότι η δυνατότητα εκμάθησης κάθε νευρώνα οφείλεται στην ηλεκτροχημική αντίσταση που έχει η κάθε σύναψη. Ο νευρώνας έχει την δυνατότητα να αυξομειώνει την αγωγιμότητα των συνάψεων και να ελέγχει έτσι ποια ερεθίσματα από τους γειτονικούς νευρώνες θα “περνάνε” στον πυρήνα και ποια όχι.

Οι διαφορές στην αγωγιμότητα των συνάψεων και, επομένως, το φιλτράρισμα των ερεθισμάτων είναι ουσιαστικά αυτό που κάνει τον ανθρώπινο εγκέφαλο εκπαιδευόμενο και προσαρμοζόμενο. Θεωρούμε ότι μια γνώση έχει κατακτηθεί από τον εγκέφαλο όταν οι σχετικοί νευρώνες ενεργοποιούνται μόνο στα ανάλογα ερεθίσματα που σχετίζονται με τη γνώση αυτή. Αυτή ακριβώς τη συμπεριφορά προσπαθούν να εξομοιώσουν τα τεχνητά νευρωνικά δίκτυα.



## Τεχνητός Νευρώνας

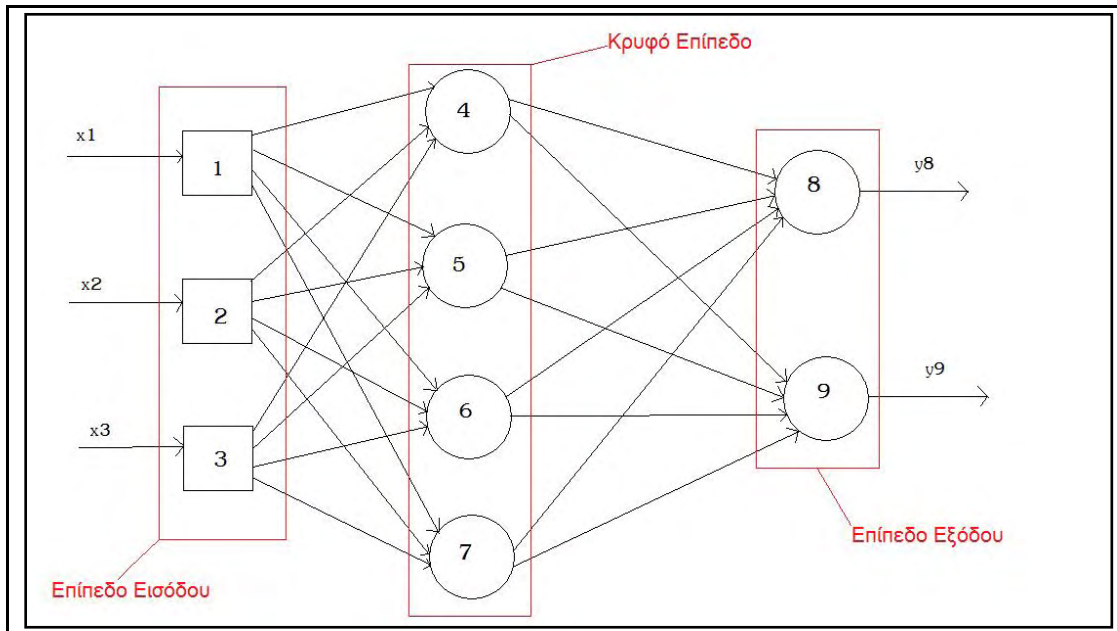
Κάθε τεχνητός νευρώνας είναι ένα υπολογιστικό μοντέλο και μοιάζει σε πολλά σημεία με έναν βιολογικό νευρώνα. Όπως απεικονίζεται στην εικόνα 1.2, ένας τεχνητός νευρώνας δέχεται κάποια σήματα εισόδου  $x_1, x_2, \dots, x_n$  τα οποία μεταβάλλονται από διάφορες τιμές βάρους  $w_1, w_2, \dots, w_n$ . Το βάρος μας παραπέμπει στο ρόλο που έχει η σύναψη στο βιολογικό νευρώνα και τα σήματα εισόδου στους ηλεκτρικούς παλμούς του εγκεφάλου. Το σώμα του τεχνητού νευρώνα χωρίζεται σε δύο μέρη, τον αθροιστή και τη συνάρτηση ενεργοποίησης. Ο αθροιστής προσθέτει τα επηρεασμένα από τα βάρη σήματα εισόδου παράγοντας την ποσότητα  $S$  και η συνάρτηση ενεργοποίησης διαμορφώνει την τελική τιμή του σήματος εξόδου  $y$  ανάλογα πάντα με την ποσότητα  $S$  και με μία τιμή κατωφλίου. Ακόμη, παρατηρείται πως ο τεχνητός νευρώνας μπορεί να έχει πολλές εξόδους, αλλά όλες θα έχουν την ίδια τιμή.



Εικόνα 1.2: Αναπαράσταση τεχνητού νευρώνα

## Τεχνητό Νευρωνικό Δίκτυο

Η εικόνα 1.3 παριστάνει ένα τεχνητό νευρωνικό δίκτυο με τους τεχνητούς νευρώνες οργανωμένους σε τρία επίπεδα. Το πρώτο από αυτά τα επίπεδα ονομάζεται επίπεδο εισόδου και χρησιμοποιείται για την εισαγωγή δεδομένων. Τα στοιχεία του ουσιαστικά δεν είναι τεχνητοί νευρώνες, καθώς δεν εκτελούν κανένα υπολογισμό. Στη συνέχεια, ακολουθούν ένα ή περισσότερα κρυφά επίπεδα στα οποία γίνονται οι περισσότερες πράξεις και στο τέλος υπάρχει το επίπεδο εξόδου, το οποίο παράγει το τελικό αποτέλεσμα.



Εικόνα 1.3: Αναπαράσταση τεχνητού νευρωνικού δικτύου

## Εκπαίδευση

Το κύριο χαρακτηριστικό των τεχνητών νευρωνικών δικτύων είναι η έμφυτη ικανότητα εκπαίδευσης. Ως εκπαίδευση μπορεί να οριστεί η σταδιακή βελτίωση της ικανότητας του δικτύου να επιλύει κάποιο πρόβλημα (π.χ. η σταδιακή προσέγγιση μίας συνάρτησης). Η εκπαίδευση επιτυγχάνεται μέσω μίας επαναληπτικής διαδικασίας κατά την οποία σταδιακά προσαρμόζονται οι παράμετροι του δικτύου (συνήθως τα βάρη) σε τιμές κατάλληλες, ώστε να επιλυθεί με επαρκή επιτυχία το προς εξέταση πρόβλημα. Αφού ένα δίκτυο εκπαιδευτεί, οι παράμετροί του συνήθως σταθεροποιούνται στις κατάλληλες τιμές και από εκεί κι έπειτα είναι σε λειτουργική κατάσταση. Το ζητούμενο είναι το λειτουργικό δίκτυο να χαρακτηρίζεται από μία ικανότητα γενίκευσης: αυτό σημαίνει πως δίνει ορθές εξόδους για εισόδους καινοφανείς και διαφορετικές από αυτές με τις οποίες εκπαιδεύτηκε.

Υπάρχουν τρία είδη εκπαίδευσης στα τεχνητά νευρωνικά δίκτυα. Πρώτο είδος εκπαίδευσης είναι η *εκπαίδευση με επίβλεψη* στην οποία με την καθοδήγηση κάποιου αλγορίθμου εκπαίδευσης και κάποιων σφαλμάτων αναπροσαρμόζονται οι τιμές των βαρών. Δεύτερο είδος εκπαίδευσης είναι η *βαθμολογημένη εκπαίδευση* στην οποία η έξοδος χαρακτηρίζεται "καλή" ή "κακή" με βάση κάποια αριθμητική κλίμακα και αναλόγως αλλάζουν οι τιμές των βαρών. Τέλος, το τρίτο είδος εκπαίδευσης είναι η *εκπαίδευση χωρίς επίβλεψη* στην οποία το δίκτυο έχει την τάση να αυτο-οργανώνεται με βάση τα διανύσματα εισόδου και μόνο, καθώς δεν υπάρχουν αντίστοιχα διανύσματα εξόδου. Στις περισσότερες εφαρμογές χρησιμοποιείται η εκπαίδευση με επίβλεψη, για την οποία υπάρχουν αρκετοί αλγόριθμοι. Παραδείγματα τέτοιων αλγορίθμων είναι ο αλγόριθμος που βασίζεται στον κανόνα Δέλτα και ο αλγόριθμος ανάστροφης μετάδοσης λάθους.

## Ιδιότητες

Οι βασικές ιδιότητες των τεχνητών νευρωνικών δικτύων είναι οι εξής [49]:

- Η ικανότητα τους να μαθαίνουν μέσω παραδειγμάτων
- Η δυνατότητα θεώρησης τους ως κατανεμημένη μνήμη (distributed memory) και ως μνήμη συσχέτισης (associative memory)
- Η μεγάλη τους ανοχή σε σφάλματα (fault-tolerant)
- Η ικανότητα τους για αναγνώριση προτύπων (pattern recognition)

## Εφαρμογές

Όλες οι εφαρμογές των τεχνητών νευρωνικών δικτύων έχουν δημιουργηθεί τα τελευταία χρόνια και μερικές από αυτές ήδη βρίσκονται ως έτοιμα προϊόντα στην αγορά και χρησιμοποιούνται ευρέως. Οι εφαρμογές αυτές περιλαμβάνουν αναγνώριση προτύπων, υπολογισμό συναρτήσεων, βελτιστοποίηση, πρόβλεψη, αυτόματο έλεγχο και πολλά άλλα θέματα. Μία απλή λίστα σε μερικές περιοχές με διάφορες εφαρμογές θα μπορούσε να περιλάβει επιγραμματικά και μόνον:

### Βιολογία

- Καλύτερη κατανόηση της λειτουργίας του εγκεφάλου
- Μοντέλα για την όραση

### Επιχειρήσεις

- Εκτίμηση για την ύπαρξη κοιτασμάτων πετρελαίου σε γεωλογικά πετρώματα
- Για την επιλογή του κατάλληλου προσωπικού σε σημαντικές θέσεις στην επιχείρηση

### Ιατρική

- Ανάγνωση και ανάλυση των ακτίνων X
- Κατανόηση των επιληπτικών κρίσεων
- Παρακολούθηση εγχείρησης
- Προβλέψεις για αντιδράσεις οργανισμών στην λήψη φαρμάκων
- Διάγνωση και θεραπεία από τα συμπτώματα
- Ανάλυση ομιλίας σε ακουστικά βαρηκοίας κωφών ατόμων

### Στρατιωτική τέχνη

- Αναγνώριση και παρακολούθηση στόχων
- Βελτιστοποίηση της χρήσης πόρων σε έλλειψη
- Κωδικοποίηση σημάτων ραντάρ
- Δημιουργία "έξυπνων" όπλων
- Για κατόπτευση

## Χρηματοοικονομικά

- Ανάλυση επικινδυνότητας δανείων
- Ανάγνωση χειρόγραφων κειμένων
- Αξιολόγηση επενδύσεων και ανάλυση χαρτοφυλακίων
- Αναγνώριση πλαστογραφιών

## Βιομηχανία

- Αυτοματοποίηση ρομπότ και συστημάτων ελέγχου
- Επιλογή ανταλλακτικών κατά τη συναρμολόγηση
- Έλεγχος στην γραμμή παραγωγής
- Επιθεώρηση της ποιότητας κατά την κατασκευή

## Περιβάλλον

- Πρόβλεψη καιρού
- Ανάλυση τάσεων και παρατηρήσεων

Όσον αφορά το πρόβλημα της δημιουργίας ωρολογίων προγραμμάτων, αυτή η προσέγγιση είναι αρκετά υποσχόμενη καθώς ερευνητές έχουν μελετήσει και χρησιμοποιήσει διάφορα μοντέλα τεχνητών νευρωνικών δικτύων, όπως για παράδειγμα τα τεχνητά νευρωνικά δίκτυα Potts.

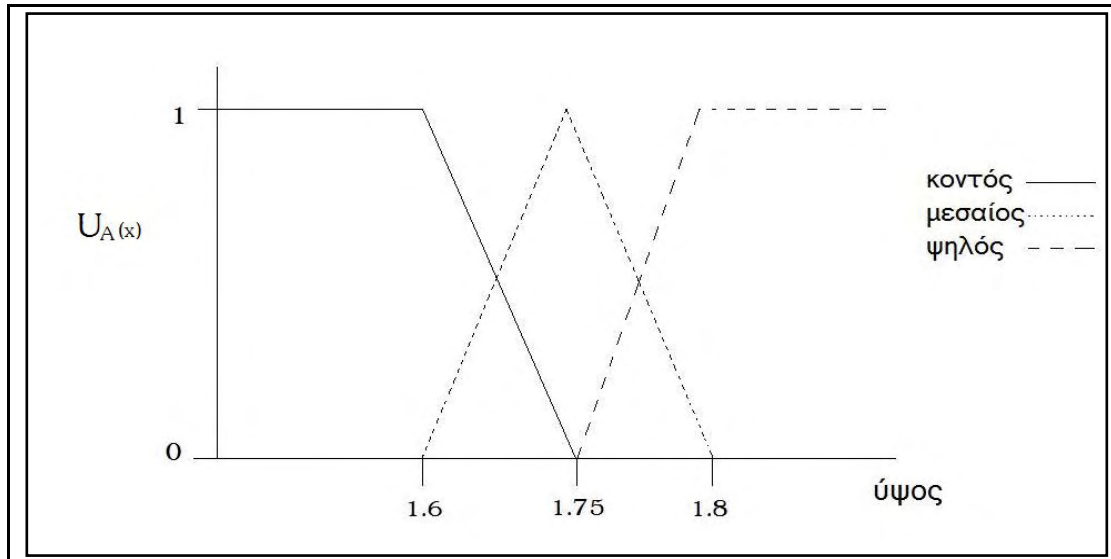
### **1.2.2 Ασαφής Λογική**

Η Ασαφής Λογική (Fuzzy Logic) [50][51][52][53][54][55] είναι μια επέκταση της κλασσικής αριστοτέλειας λογικής, προερχόμενη από τη θεωρία των ασαφών συνόλων. Μία πρόταση μπορεί να είναι αληθής με κάποιο βαθμό αληθείας και όχι απλά αληθής ή ψευδής. Με απλά λόγια, η ασαφής λογική λέει ότι τα πράγματα συχνά δεν είναι «άσπρο-μαύρο» αλλά «αποχρώσεις του γκρι». Η ιδέα αυτή απετέλεσε επανάσταση στη θεωρία της λογικής, γιατί ξέφυγε από το μοντέλο που κυριαρχούσε εδώ και 2500 χρόνια, δηλαδή το μοντέλο του «0-1», «αληθές-ψευδές».

### **Παράδειγμα**

Ένας άνθρωπος Α έχει ύψος 1.95 και ένας άνθρωπος Β έχει ύψος 1.90. Η κλασσική λογική καθορίζει ότι η πρόταση "Ο άνθρωπος Α είναι ψηλότερος από τον άνθρωπο Β" είναι αληθής. Η ασαφής λογική όμως ορίζει ότι η παραπάνω πρόταση είναι αληθής μεν, αλλά με κάποιο βαθμό αληθείας, π.χ. αληθής κατά 20%. Ας θεωρήσουμε ότι δεν λαμβάνουμε υπ'όψιν μόνο το ύψος του ανθρώπου αλλά και άλλα χαρακτηριστικά τα οποία είναι εγγενώς υποκειμενικά όπως "η καλοσύνη", "η ομορφιά", "το πόσο εργατικός είναι" κλπ. Η κλασσική λογική δεν μπορεί να κωδικοποιήσει τα παραπάνω χαρακτηριστικά γιατί δεν υπάρχει σαφής ποσοτικοποίησή τους. Δηλαδή δεν μπορεί να πει ότι η "καλοσύνη" του είναι 4. Η ασαφής όμως λογική μπορεί να πραγματοποιήσει κάτι τέτοιο χρησιμοποιώντας λεκτικές μεταβλητές. Με τις λεκτικές μεταβλητές μπορούμε να έχουμε ποιοτικούς χαρακτηρισμούς αντί για ακριβείς τιμές. Για το συγκεκριμένο παράδειγμα, η ασαφής λογική

δημιουργεί τη λεκτική μεταβλητή " ύψος" την οποία και διαχωρίζει στις εξής κατηγορίες: "κοντός", "μεσαίος", "ψηλός". Κάθε κατηγορία είναι ένα ασαφές σύνολο όπως φαίνεται στην εικόνα 1.4 όπου  $U_A(x)$  είναι ο βαθμός αληθείας.



Εικόνα 1.4: Συνάρτηση συγγένειας για τα ασαφή σύνολα “κοντός”, “μεσαίος” και “ψηλός”

## Ασαφές Σύνολο

Ένα ασαφές σύνολο (fuzzy set)  $A$  ορίζεται ως ένα σύνολο διατεταγμένων ζευγών  $(x, U_A(x))$  όπου  $x \in X$  και  $U_A(x) \in [0,1]$ . Το σύνολο  $X$  αποτελεί ένα ευρύτερο σύνολο αναφοράς που περιλαμβάνει όλα τα αντικείμενα στα οποία μπορεί να γίνει αναφορά. Η τιμή  $U_A(x)$  όπως προαναφέραμε παραπάνω λέγεται βαθμός αληθείας, παίρνει τιμές στο διάστημα  $[0,1]$  και συμβολίζει το βαθμό συγγένειας του  $x$  στο  $A$ , δηλαδή, το κατά πόσο το στοιχείο  $x$  μπορεί να θεωρηθεί ότι ανήκει στο  $A$ . Τέλος, η συνάρτηση  $U_A$  ονομάζεται συνάρτηση συγγένειας. Στην πράξη, η συνάρτηση συγγένειας μπορεί να προέρχεται από:

- υποκειμενικές εκτιμήσεις
- προκαθορισμένες και απλοποιημένες μορφές
- συχνότητες εμφανίσεων και πιθανότητες
- φυσικές μετρήσεις και
- διαδικασίες μάθησης και προσαρμογής (συνήθως με νευρωνικά δίκτυα).

## Ασαφείς Μεταβλητές

Μία μεταβλητή της οποίας οι τιμές ορίζονται με ασαφή σύνολα ονομάζεται ασαφής μεταβλητή. Για παράδειγμα, τα ασαφή σύνολα κοντός, μεσαίος και ψηλός στην εικόνα 1.4, θα μπορούσαν να είναι το πεδίο τιμών της ασαφούς μεταβλητής ύψος. Για αυτό τον λόγο η μεταβλητή ύψος χαρακτηρίζεται και ως λεκτική μεταβλητή, ενώ τα κοντός, μεσαίος, ψηλός ως πρωταρχικές

λεκτικές τιμές. Οι ασαφείς μεταβλητές μπορεί να χρησιμοποιηθούν στην ασαφή συλλογιστική η οποία είναι έννοια που αφορά την εξαγωγή συμπερασμάτων με χρήση ασαφών κανόνων.

## Ασαφή Συστήματα

Τα ασαφή συστήματα μπορούν να λειτουργούν σε περιβάλλον ασάφειας και αβεβαιότητας και δίνουν αποτελέσματα που έχουν νόημα για τον άνθρωπο. Πλησιάζουν δηλαδή την ανθρώπινη λογική. Είναι ιδανικό εργαλείο για την λήψη αποφάσεων. Χαρακτηριστικό πλεονέκτημα της ασαφούς λογικής είναι ότι μπορεί και λειτουργεί, αλλά και αναλύει, συστήματα τα οποία είναι αρκετά πολύπλοκα. Ένας ασαφές σύστημα στην ουσία μπορεί να δημιουργηθεί κατά τέτοιο τρόπο ώστε να «αντικατοπτρίζει» τις γνώσεις ενός ειδικού πάνω στο θέμα που επιχειρεί να αντιμετωπίσει. Ουσιαστικά η γνώση του ειδικού κωδικοποιείται μέσα στο σύστημα ασαφούς λογικής.

## Εφαρμογές

Υπάρχουν αμέτρητες εφαρμογές για την ασαφή λογική. Οι πιο σημαντικές πραγματικές εφαρμογές της είναι οι παρακάτω [56][57][58]:

- το σύστημα Linkman,
- ο υπόγειος σιδηρόδρομος Sendai,
- η πρόβλεψη γενετικών γνωρισμάτων,
- ο έλεγχος θερμοκρασίας δωματίου,
- η αυτο-εστίαση σε μία κάμερα,
- οι ιατρικές διαγνώσεις,
- η μετεωρολογία,
- τα χρονοδιαγράμματα λεωφορείων,
- το χρηματιστήριο, κ.α.

Η προσέγγιση της *ασαφούς πολλαπλής ευρετικής διάταξης* (fuzzy multiple heuristic ordering) στη λύση προβλημάτων δημιουργίας ωρολογίων προγραμμάτων και χρονοπρογραμματισμού είναι επιτυχής.

### 1.2.3 Εξελικτικοί Αλγόριθμοι

Ένας Εξελικτικός Αλγόριθμος (Evolutionary Algorithm) [35][36][37] είναι μία μέθοδος αναζήτησης-βελτιστοποίησης που διατηρεί ένα πληθυσμό ατόμων τον οποίο εξελίσσει σύμφωνα με κάποιους κανόνες επιλογής και κάποιους τελεστές όπως ο *ανασυνδυασμός* (recombination) και η *μετάλλαξη* (mutation). Συνήθως παίρνει την έμπνευση της από τη βιολογία και τους φυσικούς μηχανισμούς εξέλιξης και για αυτόν τον λόγο μία υποψήφια λύση, η κωδικοποίηση και η αναπαράσταση της καλούνται άτομο, γενότυπος και φαινότυπος αντίστοιχα. Ένας απλός εξελικτικός αλγόριθμος παρουσιάζεται στην εικόνα 1.5 όπου  $P(G)$  είναι ο τρέχων πληθυσμός και  $P(G+1)$  ο νέος πληθυσμός.

```

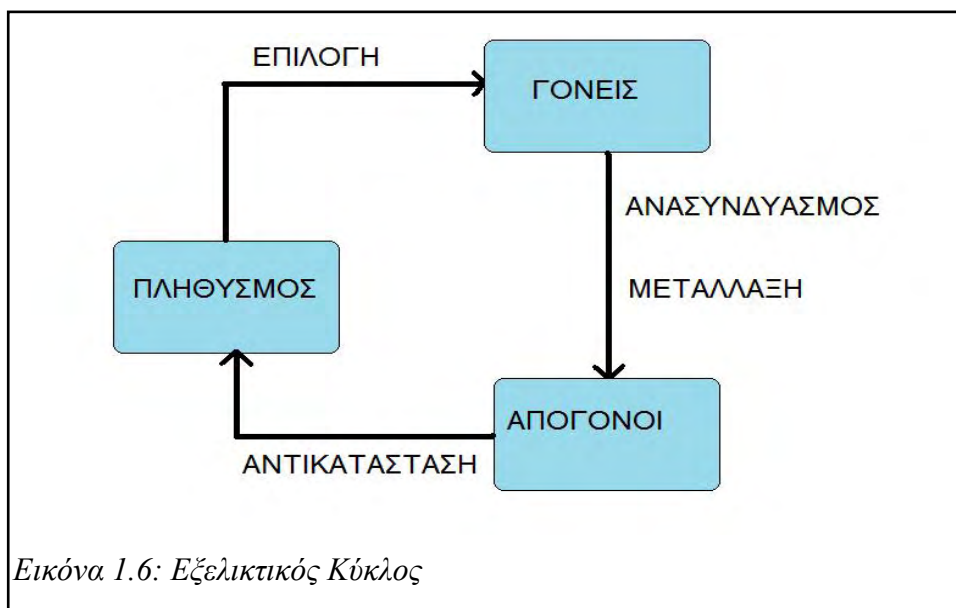
G:=0;
Αρχικοποίησε και αξιολόγησε [P(G)];
while not stop_condition do
  P'(G):=παραλλαγή [P(G)]
  Αξιολόγησε [P'(G)];
  P(G+1):= επέλεξε [P'(G), P(G)]
  G:=G+1;
end while

```

Εικόνα 1.5: Ψευδοκώδικας Εξελικτικού Αλγορίθμου

## Εξελικτικός Κύκλος

Ο εξελικτικός αλγόριθμος βαδίζει σύμφωνα με τον εξελικτικό κύκλο (εικόνα 1.6) όπου οι απόγονοι δημιουργούνται μέσω του ανασυνδυασμού των γονέων και μέσω της μετάλλαξης. Ακολουθεί η χρήση της *συνάρτησης ποιότητας* (fitness function) για την αξιολόγηση των απογόνων και τελικά η επιλογή των ατόμων του πληθυσμού που θα επιβιώσουν στην επόμενη γενιά. Αυτός ο κύκλος συνεχίζεται μέχρι να ικανοποιηθεί κάποια προκαθορισμένη συνθήκη τερματισμού. Αυτή θα μπορούσε να είναι η επίτευξη της βέλτιστης λύσης, κάποιο κριτήριο πλήρωσης χρόνου, ένας καθορισμένος αριθμός γενεών κ.α.



Εικόνα 1.6: Εξελικτικός Κύκλος

## Κύρια Συστατικά

Οι εξελικτικοί αλγόριθμοι έχουν διάφορα συστατικά, διαδικασίες ή τελεστές που πρέπει να διευκρινιστούν προκειμένου να καθοριστεί ένας συγκεκριμένος εξελικτικός αλγόριθμος. Τα σημαντικότερα συστατικά είναι τα εξής:

- Αναπαράσταση (καθορισμός των ατόμων). Αντικείμενα που σχηματίζουν πιθανές λύσεις μέσα στο αρχικό πλαίσιο προβλήματος καλούνται φαινότυποι, ενώ οι κωδικοποιήσεις αυτών των αντικειμένων καλούνται γενότυποι. Η κωδικοποίηση συνήθως γίνεται με δυαδική αναπαράσταση. Για παράδειγμα, σε ένα πρόβλημα βελτιστοποίησης ακέραιων αριθμών, θα χρησιμοποιούσαν ως φαινότυπο το 18 και ως γενότυπο το 10010.
- Συνάρτηση ποιότητας (αξιολόγησης). Είναι μία συνάρτηση που αναθέτει ένα ποιοτικό μέτρο στους γενότυπους και με αυτόν τον τρόπο γίνεται η αξιολόγηση τους.
- Πληθυσμός. Ο πληθυσμός κρατάει τις πιθανές λύσεις και για την ακρίβεια είναι ένα σύνολο που περιέχει γενότυπους.
- Μηχανισμός επιλογής γονέων. Ο ρόλος του μηχανισμού επιλογής γονέων είναι να διακρίνει τα καλύτερα άτομα με βάση την τιμή της ποιότητας τους και στη συνέχεια να τα επιλέγει ως γονείς.
- Τελεστές παραλλαγής, μετάλλαξη και ανασυνδυασμός. Η μετάλλαξη είναι ένας μοναδιαίος τελεστής παραλλαγής που εφαρμόζεται σε ένα γενότυπο και παράγει ένα απόγονο, δηλαδή μία καινούργια υποψήφια λύση. Ενώ ο ανασυνδυασμός ή η διασταύρωση είναι ένας δυαδικός τελεστής παραλλαγής που εφαρμόζεται σε δύο γενότυπους (γονείς) και παράγει έναν απόγονο.
- Μηχανισμός επιλογής επιζώντων. Ο ρόλος του μηχανισμού επιλογής επιζώντων είναι να διακρίνει τα καλύτερα άτομα με βάση την τιμή της ποιότητας τους και να ανανεώσει με αυτά τον πληθυσμό.

## Πλεονεκτήματα

Τα πλεονεκτήματα των εξελικτικών αλγορίθμων γενικά είναι τα παρακάτω:

- Είναι ευέλικτοι και αποτελεσματικοί και μπορούν να εφαρμοστούν σε διαφορετικών ειδών προβλήματα με την ανάλογη αναπαράσταση.
- Είναι σχετικά απλοί στη σύλληψη και την υλοποίηση τους.
- Το χαρακτηριστικό του έμμεσου παραλληλισμού (implicit parallelism) τους κάνει πολύ αποτελεσματικούς σε προβλήματα βελτιστοποίησης.
- Μπορούν να βρουν σχετικά γρήγορα ικανοποιητικές λύσεις σε προβλήματα με υπερβολικά μεγάλο αριθμό λύσεων.
- Μπορούν να χειριστούν δεδομένα με θόρυβο που μεταβάλλονται συνεχώς.
- Μπορούμε να χρησιμοποιούμε υβριδικές μορφές βελτιώνοντας την αποτελεσματικότητά τους.



## Μειονεκτήματα

Ενώ τα μειονεκτήματα τους είναι:

- Δεν εγγυώνται ότι η λύση του προβλήματος είναι η βέλτιστη.
- Αν ο αλγόριθμος εγκλωβιστεί σε τοπικό βέλτιστο δεν είναι σίγουρο ότι θα μπορεί πάντα να ξεφύγει από αυτό.
- Τέλος, δυσκολία υπάρχει στην επιλογή της κατάλληλης αναπαράστασης του κάθε προβλήματος και της επιλογής των παραμέτρων του αλγορίθμου επειδή οι διαδικασίες επιλογής, ανασυνδυασμού και μετάλλαξης διαφέρουν ανάλογα με τη φύση του προβλήματος.

## Εφαρμογές

Οι εξελικτικοί αλγόριθμοι μπορούν να δώσουν λύσεις σε πάρα πολλούς τομείς και οι εφαρμογές τους αυξάνονται συνεχώς. Παρακάτω αναφέρονται ενδεικτικά μερικοί τομείς που βρίσκουν εφαρμογή οι εξελικτικοί αλγόριθμοι.

- Χρονοπρογραμματισμός
- Επεξεργασίας εικόνας
- Νευρωνικά δίκτυα
- Προβλέψεις καιρού, σεισμών κ.α.
- Οικονομία

## Είδη Εξελικτικών Αλγορίθμων

Τα βασικότερα μοντέλα εξελικτικών αλγορίθμων είναι [59][60][61][62][63]:

- Γενετικοί Αλγόριθμοι (Genetic Algorithms)
- Εξελικτικός Προγραμματισμός (Evolutionary Programming)
- Αλγόριθμοι Μυρμηγκιών (Ant Algorithms)
- Γενετικός Προγραμματισμός (Genetic Programming)
- Εξελικτική Στρατηγική (Evolution Strategy)
- Βελτιστοποίηση με Σμήνος Σωματιδίων (Particle Swarm Optimization)
- Διαφοροεξελικτικοί Αλγόριθμοι (Differential Evolution)

Αν και οι βασικές αρχές τους είναι όμοιες κάθε ένα υλοποιεί τον εξελικτικό αλγόριθμο με διαφορετικό τρόπο. Κυρίως οι διαφορές αυτές αφορούν την αναπαράσταση της δομής τους, τους μηχανισμούς επιλογής, τους γενετικούς τελεστές και την μέτρηση της απόδοσης. Παρ'όλα αυτά, η πιο γνωστή μεθοδολογία είναι οι Γενετικοί Αλγόριθμοι. Στα επόμενα κεφάλαια θα αναφερθούμε ειδικά στους Γενετικούς Αλγορίθμους και στον τρόπο με τον οποίο λύνουν το πρόβλημα της δημιουργίας ωρολογίων προγραμμάτων.

**ΜΕΡΟΣ ΙΙ**  
**Η συμβολή των Γενετικών Αλγορίθμων**

# ΚΕΦΑΛΑΙΟ 2

## Οι Γενετικοί Αλγόριθμοι

Ο λογικός άνθρωπος προσαρμόζεται στον κόσμο,  
ο παράλογος επιμένει να προσπαθεί να προσαρμόσει τον κόσμο στα μέτρα του.  
Συνεπώς όλη η πρόοδος εξαρτάται από τον παράλογο άνθρωπο.

~George Bernard Shaw (1856-1950)

### 2.1 Εισαγωγή στους Γενετικούς Αλγόριθμους

Τις δεκαετίες του 1950–1960 πολλοί επιστήμονες της επιστήμης υπολογιστών πραγματοποίησαν μελέτες σε εξελικτικά συστήματα με την σκέψη ότι η εξέλιξη θα μπορούσε να χρησιμοποιηθεί ως εργαλείο βελτιστοποίησης για προβλήματα που απασχολούσαν μηχανικούς. Η ιδέα σε όλα αυτά τα συστήματα ήταν να αναπτύξουν έναν πληθυσμό «υποψηφίων» λύσεων σε ένα δεδομένο πρόβλημα χρησιμοποιώντας τελεστές παρόμοιους με αυτούς που χρησιμοποιεί η φύση κατά την διάρκεια γενετικών παραλλαγών και φυσικής επιλογής.

Η φύση έχει έναν πολύ ισχυρό μηχανισμό εξέλιξης των οργανισμών, που βασίζεται στον ακόλουθο κανόνα της φυσικής επιλογής: οι οργανισμοί που δε μπορούν να επιβιώσουν στο περιβάλλον τους πεθαίνουν, ενώ οι υπόλοιποι πολλαπλασιάζονται μέσω της αναπαραγωγής. Οι απόγονοι παρουσιάζουν μικρές διαφοροποιήσεις από τους προγόνους τους, ενώ συνήθως υπερισχύουν αυτοί που συγκεντρώνουν τα καλύτερα χαρακτηριστικά.

Αν το περιβάλλον μεταβάλλεται με αργούς ρυθμούς, τα διάφορα είδη μπορούν να εξελίσσονται σταδιακά ώστε να προσαρμόζονται σε αυτό. Αν όμως συμβούν ραγδαίες μεταβολές, αρκετά είδη οργανισμών θα εξαφανιστούν. Σποραδικά, συμβαίνουν τυχαίες μεταλλάξεις, από τις οποίες οι περισσότερες οδηγούν τα μεταλλαγμένα άτομα στο θάνατο, αν και είναι πιθανό, πολύ σπάνια όμως, να οδηγήσουν στη δημιουργία νέων «καλύτερων» οργανισμών.

Η θεωρία της εξέλιξης (evolution) έχει χρησιμοποιηθεί σε μία κατηγορία αλγορίθμων επίλυσης προβλημάτων, που ονομάζονται γενετικοί αλγόριθμοι (genetic algorithms).

Η εισαγωγή των αλγορίθμων αυτών έγινε το 1958 από τον Friedberg, ο οποίος επιχείρησε να συνδυάσει μικρά προγράμματα FORTRAN, ωστόσο τα προγράμματα που προέκυψαν τις περισσότερες φορές δεν ήταν εκτελέσιμα [1].

Οι γενετικοί αλγόριθμοι τελικά επινοήθηκαν από τον John Holland την δεκαετία του 1960 και αναπτύχθηκαν από τους φοιτητές του και τους συνεργάτες του στο πανεπιστήμιο του Μίσιγκαν [2].

Ο γενετικός αλγόριθμος είναι μια μέθοδος για μεταφορά από έναν πληθυσμό «χρωμοσωμάτων» σε έναν άλλο πληθυσμό χρησιμοποιώντας ένα είδος φυσικής επιλογής μαζί με γενετικούς τελεστές όπως:

- Διασταύρωσης (crossover), δηλαδή ανταλλαγή γενετικού υλικού μεταξύ δύο χρωμοσωμάτων από απλοειδείς γονείς.

- Μετάλλαξης (mutation), δηλαδή αναστροφή γενετικού υλικού σε ένα τυχαία επιλεγμένο γονίδιο.
- Αντιστροφής (inversion), δηλαδή ανακατανομή της διάταξης των γονιδίων σε ένα χρωμόσωμα.
- Επιλογής (selection), δηλαδή του διαχωρισμού των χρωμοσωμάτων που θα χρησιμοποιηθούν για την παραγωγή της επόμενης γενεάς.

Ο γενετικός αλγόριθμος εκτελεί μία αναζήτηση στο χώρο των υποψηφίων λύσεων, με στόχο την εύρεση κάποιας λύσης που μεγιστοποιεί τη συνάρτηση καταλληλότητας (αντικειμενική συνάρτηση κόστους). Η αναζήτηση αυτή είναι παράλληλη, καθώς σε κάθε υποψήφια λύση μπορεί να εκτελεστεί ξεχωριστή αναζήτηση. Η μέθοδος της αναζήτησης μπορεί να θεωρηθεί σαν αναρρίχηση λόφου, καθώς γίνονται μικρές αλλαγές στις υποψήφιες λύσεις του πληθυσμού και επιλέγονται πάντα οι καλύτερες, βάσει της συνάρτησης καταλληλότητας. Η αναζήτηση επικεντρώνεται στις περισσότερο κατάλληλες λύσεις, χωρίς όμως να αγνοούνται οι υπόλοιπες, καθώς υπάρχει πάντα ο κίνδυνος να παγιδευτεί η διαδικασία σε τοπικό μέγιστο.

Μέχρι τις αρχές της δεκαετίας του 1980 η μελέτη των γενετικών αλγορίθμων περιοριζόταν σε θεωρητικό επίπεδο, με λίγες πρακτικές εφαρμογές. Στην περίοδο αυτή οι γενετικοί αλγόριθμοι χρησιμοποιούνταν κυρίως για προβλήματα βελτιστοποίησης συναρτήσεων, χρησιμοποιώντας χρωμοσώματα συγκεκριμένου μήκους, δυαδικής κωδικοποίησης. Χαρακτηριστικές είναι οι μελέτες των Hollstien και De Jong την περίοδο αυτή.

Ο Hollstien πραγματοποίησε ανάλυση για το πώς επιδρούν οι τελεστές της επιλογής και της διασταύρωσης στην επίδοση του αλγορίθμου [3]. Ο De Jong προσπάθησε να προσδιορίσει τα στοιχεία εκείνα που εμποδίζουν τους αλγορίθμους να φτάσουν γρήγορα σε σύγκλιση [3].

Από την αρχή της δεκαετίας του '80, η επιστημονική κοινότητα που ασχολείται με τη θεωρία και εφαρμογή των γενετικών αλγορίθμων, έχει δημιουργήσει μία πληθώρα πρακτικών εφαρμογών που εκτείνονται σε πολλούς τομείς ερευνητικής και όχι μόνο δραστηριότητας. Βελτιώνοντας την απόδοση των γενετικών με τη ρύθμιση και βελτιστοποίηση των γενετικών τελεστών, αποδεικνύεται ότι οι γενετικοί αλγόριθμοι μπορούν να χρησιμοποιηθούν σε διάφορες κατηγορίες προβλημάτων και αποτελούν μία ισχυρή μέθοδο βελτιστοποίησης. Τα τελευταία χρόνια παρατηρείται συστηματική ανάπτυξη των γενετικών αλγορίθμων στις φυσικές επιστήμες, στη μηχανική και στον κόσμο των επιχειρήσεων καθώς χρησιμοποιούνται σε προβλήματα προγραμματισμού (scheduling), βελτιστοποίησης (optimization) κτλ.

## 2.2. Βασικές έννοιες

### 2.2.1 Γονίδια και χρωμοσώματα

Το γονίδιο αποτελεί τη βασική δομική μονάδα στην γενετική αλλά και στην μέθοδο βελτιστοποίησης με τη χρήση γενετικών αλγορίθμων. Τα γονίδια αποτελούν την κωδικοποιημένη παράσταση των παραμέτρων βελτιστοποίησης ενώ τα χρωμοσώματα περιέχουν το σύνολο των γονιδίων.

Πιο συγκεκριμένα, κάθε υποψήφια λύση παριστάνεται με μια συμβολοσειρά ενός πεπερασμένου αλφαβήτου. Κάθε στοιχείο της συμβολοσειράς είναι ένα γονίδιο και η ίδια η συμβολοσειρά είναι ένα χρωμόσωμα.

Στην φύση (DNA) το αλφάβητο έχει μήκος τέσσερα και αποτελείται από τα στοιχεία A,G,T και C ενώ στους γενετικούς αλγορίθμους χρησιμοποιείται συνήθως δυαδικό αλφάβητο. Ωστόσο υπάρχουν περιπτώσεις γενετικών αλγορίθμων που χρησιμοποιούν πιο σύνθετες μορφές αναπαράστασης (π.χ. πραγματικούς αριθμούς, αλφαριθμητικά στοιχεία κλπ).

Επίσης δυο βασικοί όροι που κληρονομούνται από τη βιολογία είναι ο φαινότυπος και ο γονότυπος. Φαινότυπος είναι όλα τα μορφολογικά, ηθολογικά και λοιπά χαρακτηριστικά που εκδηλώνει ένας οργανισμός σε μία δεδομένη στιγμή, δηλαδή το μέρος του γονοτύπου του οργανισμού το οποίο μπορούμε (άμεσα ή έμμεσα) να παρατηρήσουμε. Γονότυπος καλείται το σύνολο των γονιδίων ενός οργανισμού, δηλαδή το σύνολο των αλληλομόρφων που απαρτίζουν το DNA του. Με αυτόν τον ορισμό, είναι ένα μέγεθος το οποίο δεν είναι παρατηρήσιμο. Έτσι, δύο άτομα με τον ίδιο ακριβώς γονότυπο, αλλά μεγαλωμένοι σε διαφορετικά περιβάλλοντα, πιθανότατα θα διαφέρουν στον φαινότυπό τους.

## 2.2.2 Πληθυσμοί και γενιές

Ο πληθυσμός αποτελείται από μία ομάδα πιθανών λύσεων σε μορφή χρωμοσωμάτων. Χρησιμοποιείται από τον γενετικό αλγόριθμο για την εύρεση της βέλτιστης λύσης. Οι επαναλήψεις του γενετικού αλγόριθμου ονομάζονται γενιές.

Η αναπαραγωγή η οποία αποτελείται από την επιλογή (selection), την διασταύρωση (crossover) και την μετάλλαξη (mutation) συνεχίζεται, έως ότου να δημιουργηθεί μία νέα γενιά η οποία θα αντικαταστήσει την προηγούμενη. Εκείνα τα χρωμοσώματα της εκάστοτε γενιάς που έχουν τα καλύτερα χαρακτηριστικά, θα παράγουν περισσότερα αντίγραφα τους στην επόμενη γενιά με αποτέλεσμα την σύγκλιση του πληθυσμού προς την βέλτιστη λύση. Η διαδικασία ολοκληρώνεται όταν τα κριτήρια τερματισμού έχουν ικανοποιηθεί.

Με έναν αποδοτικό γενετικό αλγόριθμο, ο πληθυσμός θα πρέπει μετά από αρκετές επαναλήψεις να συγκλίνει προς το ολικό μέγιστο. Η σύγκλιση αυτή αφορά είτε το καλύτερο στοιχείο του πληθυσμού ή το μέσο όρο αυτού. Ένα γονίδιο θεωρείται ότι έχει συγκλίνει, όταν έχει την ίδια τιμή στο 95% των χρωμοσωμάτων. Ένας πληθυσμός θεωρείται ότι έχει συγκλίνει, όταν όλα τα γονίδια του έχουν συγκλίνει.

## 2.2.3 Γονείς

Η διαδικασία επιλογής γονέων σχετίζεται με την απόδοση πιθανοτήτων επιλογής προς αναπαραγωγή στα μέλη ενός πληθυσμού υποψηφίων λύσεων. Κατά τη διαδικασία αυτή, κάποιοι γονείς με υψηλή τιμή στη συνάρτηση καταλληλότητας ενδέχεται να επιλεγούν προς αναπαραγωγή περισσότερες από μία φορές, ενώ κάποιοι γονείς με χαμηλή καταλληλότητα ενδέχεται να μην επιλεγούν καθόλου.

Κατά τη διαδικασία επιλογής, αρχικά οι υποψήφιοι λύσεις αντιγράφονται σε μια δεξαμενή ζευγαρώματος (mating pool). Η δεξαμενή αυτή έχει μέγεθος ίσο με τον αρχικό πληθυσμό, και σε αυτήν αντιγράφονται μέλη του αρχικού πληθυσμού, με πιθανότητα ανάλογη της καταλληλότητας τους.

Έπειτα από την διαδικασία αρχικοποίησης του πληθυσμού, όπου δημιουργείται η πρώτη γενιά, επιλέγονται ζευγάρια χρωμοσωμάτων από τον πληθυσμό τα οποία ονομάζονται γονείς. Η επιλογή γίνεται με πιθανοτικά και σταθμισμένα κριτήρια από την συνάρτηση κόστους.

Για παράδειγμα, ένα τυπικό σενάριο επιλογής γονέων αποτελεί η μέθοδος «ρουλέτας», κατά την οποία σε κάθε χρωμόσωμα του πληθυσμού αντιστοιχίζεται σε ένα τμήμα της ρουλέτας. Το μέγεθος του τμήματος αυτού είναι αντίστοιχο της τιμής της σχετικής συνάρτησης κόστους του χρωμοσώματος. Η ρουλέτα περιστρέφεται κάθε φορά που αναζητούμε ένα γονέα. Χρωμοσώματα τα οποία κατέχουν μεγάλο τμήμα της ρουλέτας, έχουν μεγαλύτερη πιθανότητα να εκλεγούν και να περάσουν τα χαρακτηριστικά τους στην επόμενη γενιά.

Ο τρόπος επιλογής των γονέων που θα ζευγαρώσουν επηρεάζει σημαντικά την απόδοση των γενετικών αλγορίθμων. Δυο προβλήματα που εμφανίζονται συχνά στους γενετικούς αλγόριθμους και τα οποία μπορούν να αντιμετωπιστούν με τροποποίηση της διαδικασίας επιλογής, είναι η πρόωρη σύγκλιση (premature convergence) και η αργή σύγκλιση (slow convergence). Με τον όρο «σύγκλιση» εννοούμε την επικράτηση ενός χρωμοσώματος ή μικρών παραλλαγών του, σε μεγάλο ποσοστό στον πληθυσμό.

Κατά την πρόωρη σύγκλιση, ο πληθυσμός πολύ γρήγορα συγκλίνει γύρω από κάποιο χρωμόσωμα, το οποίο όμως αποτελεί τοπικό βέλτιστο. Το αποτέλεσμα είναι να μην μπορεί πλέον ο γενετικός αλγόριθμος να ξεφύγει από αυτό το τοπικό βέλτιστο, παρά μόνο με τη διαδικασία της μετάλλαξης, το οποίο όμως έχει πρακτικά πολύ μικρή πιθανότητα να συμβεί.

Το φαινόμενο αυτό εμφανίζεται σε περιπτώσεις που η συνάρτηση καταλληλότητας εμφανίζει πολύ απότομες μεταβολές και έντονα τοπικά βέλτιστα και μπορεί να αντιμετωπιστεί με δύο τρόπους. Ο πρώτος είναι η απεικόνιση της συνάρτησης καταλληλότητας σε μία νέα συνάρτηση, λιγότερο απότομη (fitness remapping). Ο δεύτερος είναι ο καθορισμός ελάχιστων και μέγιστων ορίων, όσον αφορά το πόσες φορές επιλέγεται ένα χρωμόσωμα προς αναπαραγωγή σε κάθε κύκλο ανανέωσης του πληθυσμού.

Η αργή σύγκλιση είναι ουσιαστικά το ακριβώς αντίθετο φαινόμενο της πρόωρης σύγκλισης. Κατά αυτήν, μετά από ένα μεγάλο αριθμό επαναλήψεων, ο πληθυσμός εξακολουθεί να μην συγκλίνει. Το φαινόμενο εμφανίζεται όταν η συνάρτηση καταλληλότητας έχει μικρές κλίσεις, με αποτέλεσμα τα μέγιστα και τα ελάχιστα της να έχουν μικρές διαφορές. Η λύση είναι και πάλι η απεικόνιση της συνάρτησης σε μια νέα, η οποία πρέπει να έχει πιο έντονες διακυμάνσεις.

Τέλος μια εναλλακτική προσέγγιση στο θέμα της εξέλιξης του πληθυσμού είναι η μη ανανέωση ολόκληρου του πληθυσμού σε κάθε γενιά, αλλά ένα μέρος αυτού. Ορίζεται ως χάσμα γενεών (generation gap) το ποσοστό των χρωμοσωμάτων κάθε γενιάς που ανανεώθηκε, προς το σύνολο των χρωμοσωμάτων. Στους τυπικούς γενετικούς αλγορίθμους ο συντελεστής αυτός ισούται με τη μονάδα. Ωστόσο η νέα τάση είναι η ανανέωση μέρους του πληθυσμού και μάλιστα, στην πιο ακραία μορφή μόνο δύο μελών. Η μέθοδος αυτή ονομάζεται μέθοδος μερικής ανανέωσης (Steady state replacement).

Η μέθοδος της μερικής ανανέωσης προσεγγίζει πιο πολύ στην πραγματικότητα, αφού εκεί συνυπάρχουν πάντα σε κάποιο βαθμό οι διαφορετικές γενεές. Μάλιστα δίνεται η δυνατότητα στους απογόνους να ανταγωνιστούν τους γονείς τους, επικρατώντας και πάλι ο καλύτερος.

Υπάρχουν δύο θέματα στην προσέγγιση της μερικής ανανέωσης: πώς θα επιλεγούν οι γονείς που θα ζευγαρώσουν και πώς θα επιλεγεί ισάριθμος αριθμός γονέων που θα αποχωρήσουν, ώστε να αφήσουν χώρο για τους απογόνους.

Οι προσεγγίσεις που ακολουθούνται είναι οι ακόλουθες τρεις:

- Επιλογή των γονέων προς αναπαραγωγή με πιθανότητα ανάλογη προς την καταλληλότητα τους και τυχαία επιλογή των γονέων που θα αποχωρήσουν.
- Επιλογή των γονέων προς αναπαραγωγή τυχαία και επιλογή των γονέων που θα αποχωρήσουν με πιθανότητα αντιστρόφως ανάλογη προς την καταλληλότητα τους.

- Επιλογή των γονέων προς αναπαραγωγή με πιθανότητα ανάλογη προς την καταλληλότητα τους και επιλογή των γονέων που θα αποχωρήσουν με πιθανότητα αντιστρόφως ανάλογη προς την καταλληλότητα τους.

#### 2.2.4 Παιδιά

Οι απόγονοι δημιουργούνται από την εφαρμογή στοχαστικών τελεστών στο επιλεγμένο ζεύγος γονέων, όπως η διασταύρωση και η μετάλλαξη. Η διασταύρωση (crossover) αφορά την τυχαία επιλογή του σημείου του χρωμοσώματος όπου θα γίνει η αντιστοίχιση και η συνένωση του γενετικού υλικού των δύο γονέων. Αποτέλεσμα είναι τα δύο παιδιά να φέρουν χαρακτηριστικά και των δύο γονέων. Η διασταύρωση πραγματοποιείται με πιθανότητα ( $P_{cross}$ ) που κυμαίνεται μεταξύ 0,6 και 0,8 και αποτελεί τον σημαντικότερο τελεστή.

Η μετάλλαξη αλλάζει την τιμή ενός τυχαία επιλεγμένου γονιδίου του χρωμοσώματος και αποτελεί ένα μηχανισμό ο οποίος διασφαλίζει ότι η επιλογή δεν συγκλίνει σε ένα τοπικό βέλτιστο. Επίσης διευρύνει την περιοχή αναζήτησης, καθώς εισάγει γενετικό υλικό το οποίο δεν υπάρχει στην προηγούμενη γενιά, ούτε είναι δυνατόν να προκύψει από τον τελεστή διασταύρωσης. Η μετάλλαξη δεν είναι τόσο σημαντική όσο η διασταύρωση και συνήθως πραγματοποιείται με μικρότερη πιθανότητα ( $P_{mutation}$ ) της τάξης του 0,05.

#### 2.2.5 Συνάρτηση Καταλληλότητας

Η αντικειμενική συνάρτηση η οποία καθορίζει τον στόχο βελτιστοποίησης, ονομάζεται συνάρτηση κόστους ή συνάρτηση καταλληλότητας. Η συνάρτηση καταλληλότητας δέχεται ως είσοδο ένα χρωμόσωμα και επιστρέφει έναν αριθμό που υποδηλώνει πόσο κατάλληλο είναι, δηλαδή αναθέτει μία τιμή σε κάθε χρωμόσωμα του πληθυσμού. Η τιμή αυτή αποτελεί κριτήριο για το πόσο ικανοποιητική είναι η λύση που αντιπροσωπεύει το κάθε χρωμόσωμα στην συγκεκριμένη φάση.

Συγκεκριμένα το πεδίο τιμών της συνάρτησης είναι συνήθως από το 0 έως το 1 (ανάλογα με την υλοποίηση αυτό μπορεί να διαφέρει). Τιμή ίση με 1 σημαίνει ότι το συγκεκριμένο χρωμόσωμα είναι τέλειο, δηλαδή ικανοποιεί όλες τις προϋποθέσεις του προβλήματος και αποτελεί αποδεκτή λύση.

Ανάλογα με το πρόβλημα, η κατασκευή της συνάρτησης καταλληλότητας μπορεί να είναι από απλή, έως εξαιρετικά πολύπλοκη. Η ιδανική συνάρτηση καταλληλότητας θα πρέπει να είναι συνεχής και μονότονη. Ωστόσο αυτό σπάνια συμβαίνει (εάν συνέβαινε, θα αρκούσε μια απλή αναζήτηση αναρρίχησης λόφου για να βρεθεί η βέλτιστη λύση), οπότε αυτό που επιζητείται είναι μια συνάρτηση καταλληλότητας που δεν έχει πολλά τοπικά μέγιστα ή ένα απομονωμένο ολικό μέγιστο.

Ο γενικός κανόνας στην κατασκευή της συνάρτησης καταλληλότητας είναι ότι αυτή πρέπει να αντικατοπτρίζει ρεαλιστικά την αξία του χρωμοσώματος. Ωστόσο η έννοια της αξίας ενός χρωμοσώματος δεν είναι πάντα ιδιαίτερα χρήσιμη. Για παράδειγμα, σε προβλήματα βελτιστοποίησης με περιορισμούς υπάρχουν πολλά χρωμοσώματα, δηλαδή πολλές υποψήφιες λύσεις που παραβιάζουν περιορισμούς, άρα η αξία τους είναι μηδενική.

Κλασσικό παράδειγμα είναι το πρόβλημα της δημιουργίας του ωρολογίου προγράμματος, όπου οι αίθουσες και οι διδάσκοντες πρέπει να κατανεμηθούν στα τμήματα, με περιορισμούς όπως ότι μια αίθουσα ή ένας διδάσκων δεν μπορεί να συνδεθεί την ίδια ώρα με δύο διαφορετικά τμήματα.

Σε αυτή την περίπτωση η συνάρτηση καταλληλότητας δεν πρέπει να εκφράζει την ποιότητα της λύσης, αλλά το πόσο κοντά αυτή βρίσκεται σε μια έγκυρη λύση. Έτσι υπολογίζεται ο αριθμός των περιορισμών που παραβιάζει η λύση και γίνεται μια εκτίμηση για την προσπάθεια που απαιτείται για να διορθωθεί.

Μια άλλη προσέγγιση που ακολουθείται πολλές φορές είναι αυτή της προσεγγιστικής συνάρτησης καταλληλότητας (approximate fitness function). Το κρίσιμο θέμα είναι η επιθυμητή ακρίβεια της συνάρτησης καταλληλότητας και το υπολογιστικό κόστος που θεωρείται αποδεκτό για μια συνάρτηση καταλληλότητας που δίνει ενδεχομένως άριστα αποτελέσματα.

Εάν λοιπόν υπάρχουν διαθέσιμες δύο συναρτήσεις καταλληλότητας, μια ακριβής και μια προσεγγιστική, όπου όμως η ακριβής έχει δεκαπλάσιο χρόνο υπολογισμού από την προσεγγιστική, ενδεχομένως να συμφέρει η χρήση της προσεγγιστικής, αφού αυτή θα επιτρέψει την πραγματοποίηση δεκαπλάσιας ποσότητας υπολογισμών (δηλαδή αναπαραγωγών) στον ίδιο χρόνο, σε σχέση με τη χρονοβόρα συνάρτηση.

## 2.3 Αποτελεσματικότητα και Αποδοτικότητα

Ο Holland το 1975 ήταν ο πρώτος που έδωσε μια τεκμηριωμένη εξήγηση για το πώς και γιατί οι γενετικοί αλγόριθμοι λειτουργούν, με το θεώρημα των σχημάτων (schema theorem). Για την κατανόηση του θεωρήματος θα δοθούν αρχικά μερικοί ορισμοί, θεωρώντας ότι το αλφάβητο στο οποίο παριστάνονται τα γονίδια είναι το δυαδικό  $\{0,1\}$ , τότε ένα σχήμα (schema) είναι ένα χρωμόσωμα που δημιουργείται από το αλφάβητο  $\{0, 1, \#\}$ , όπου το σύμβολο # υποδηλώνει οποιαδήποτε τιμή. Ένα χρωμόσωμα λέγεται ότι περιέχει ένα σχήμα, εάν οι δύο συμβολοσειρές ταυτίζονται για όλα τα διαφορετικά από το # στοιχεία του σχήματος. Για παράδειγμα, το χρωμόσωμα 11001 περιέχει τα σχήματα #10##, 1#00#, ##001 και 1###1.

Ορίζεται ως τάξη (order) ενός σχήματος το πλήθος των συμβόλων που είναι διάφορα από το #. Ορίζεται ως μήκος (length) ενός σχήματος η απόσταση των δύο πιο εξωτερικών σταθερών (μη-#) συμβόλων. Για τα προαναφερθέντα τέσσερα σχήματα, οι τάξεις τους είναι 2, 3, 3, 2 και τα μήκη τους 2, 4, 3, 5 αντίστοιχα.

Ο Holland θεωρεί ότι η υψηλή τιμή καταλληλότητας ενός χρωμοσώματος οφείλεται στο γεγονός ότι αυτό περιέχει «καλά» σχήματα. Επιλέγοντας για αναπαραγωγή τα χρωμοσώματα με υψηλή τιμή καταλληλότητας αυξάνουμε την πιθανότητα επικράτησης των καλών σχημάτων στον πληθυσμό. Το θεώρημα των σχημάτων καθορίζει ότι αποδίδοντας πιθανότητα αναπαραγωγής στα χρωμοσώματα ανάλογη προς την καταλληλότητα τους, τα «καλά» σχήματα τυγχάνουν εκθετικά αυξανόμενου αριθμού αναπαραγωγών στις επόμενες γενεές. Επίσης καθορίζει ότι ο αριθμός των σχημάτων τα οποία μεταφέρονται από γενεά σε γενεά είναι της τάξης του  $N^l$ , όπου  $N$  το μέγεθος του πληθυσμού. Η τελευταία ιδιότητα αναφέρεται και ως έμμεσος παραλληλισμός (implicit parallelism) και θεωρείται ως η εξήγηση για την αποτελεσματικότητα των γενετικών αλγορίθμων στην επίλυση προβλημάτων [15][16].

Το βασικότερο ερώτημα που έχει απασχολήσει την επιστημονική κοινότητα σε σχέση με τους γενετικούς αλγορίθμους είναι το πώς πρέπει να κωδικοποιηθεί ένα πρόβλημα, ώστε η αντιμετώπισή του να είναι αποδοτική. Μια καλή κωδικοποίηση πρέπει να έχει τις παρακάτω ιδιότητες:

- Τα σχετικά γονίδια να είναι γειτονικά στα χρωμοσώματα.
- Να υπάρχει μικρή αλληλεπίδραση μεταξύ των γονιδίων.



Σχετικά γονίδια είναι αυτά τα οποία συνδυασμένα επηρεάζουν τη συνάρτηση καταλληλότητας. Αυτό σημαίνει ότι η συνάρτηση καταλληλότητας δεν εξαρτάται από κάθε ένα γονίδιο ξεχωριστά, αλλά από συνδυασμούς αυτών. Εάν όλα τα γονίδια ήταν ανεξάρτητα μεταξύ τους (η ιδανική περίπτωση), τότε η επίλυση ενός προβλήματος θα αναγόταν στην επίλυση  $N$  ανεξάρτητων υποπροβλημάτων, όπου  $N$  ο αριθμός των γονιδίων.

Στην πράξη, αυτό που συμβαίνει είναι ότι υπάρχει μερική αλληλεπίδραση μεταξύ των γονιδίων. Σε αυτή την περίπτωση, μια αποδοτική κωδικοποίηση θα είχε τα σχετικά γονίδια κοντά, δημιουργώντας καλά σχήματα όσο το δυνατόν μικρότερης τάξης και μικρότερου μήκους. Η δημιουργία καλών σχημάτων μικρής τάξης και μήκους αυξάνει την πιθανότητα επιβίωσης τους κατά τη φάση της διασταύρωσης στη διαδικασία της αναπαραγωγής.

Σε περίπτωση που υπάρχει ισχυρή αλληλεπίδραση μεταξύ των γονιδίων μιας κωδικοποίησης, με αποτέλεσμα να μην είναι δυνατή η τοποθέτηση σχετικών γονιδίων σε κοντινές θέσεις και η δημιουργία μικρών σχημάτων, τότε η απόδοση του γενετικού αλγορίθμου είναι αμφίβολη. Σε αυτήν την περίπτωση πρέπει είτε να επιχειρηθεί εναλλακτική κωδικοποίηση, επιλέγοντας διαφορετικές μεταβλητές για την αναπαράσταση του φαινοτύπου ή να εγκαταλειφθεί η ιδέα της χρήσης γενετικών αλγορίθμων για την επίλυση του προβλήματος.

## 2.4 Εφαρμογές

Παρουσιάζονται στη συνέχεια μερικές αντιπροσωπευτικές εφαρμογές των γενετικών αλγορίθμων.

### 2.4.1 Εύρεση μέγιστης τιμής αριθμητικών συναρτήσεων

Πρόκειται για την πιο καλά μελετημένη εφαρμογή των γενετικών αλγορίθμων. Η εύρεση του μέγιστου μιας συνάρτησης δεν είναι καθόλου εύκολη υπόθεση για συναρτήσεις πολλών μεταβλητών, οι οποίες εμφανίζουν ασυνέχειες, θόρυβο, κλπ. Το πλεονέκτημα που εμφανίζει η εφαρμογή τους σε αυτά τα προβλήματα είναι ότι η συνάρτηση καταλληλότητας είναι δεδομένη.

### 2.4.2 Συνδυαστική βελτιστοποίηση

Πρόκειται για το κλασικό πρόβλημα κατανομής πόρων σε δραστηριότητες, με σκοπό τη μεγιστοποίηση του οφέλους ή την ελάττωση του κόστους. Τα προβλήματα αυτής της κατηγορίας παρουσιάζουν συνδυαστική έκρηξη του χώρου αναζήτησης, ως προς το μέγεθος του προβλήματος, με αποτέλεσμα ο έλεγχος όλων των υποψήφιων λύσεων να είναι αδύνατος.

Το πιο γνωστό πρόβλημα αυτής της κατηγορίας είναι αυτό του πλανόδιου πωλητή, όπου στόχος είναι η εύρεση της συντομότερης διαδρομής για την επίσκεψη ενός συνόλου πόλεων. Οι γενετικοί αλγόριθμοι μπορούν να δώσουν σε αυτό το πρόβλημα πολλές λύσεις κοντά στη βέλτιστη. Ένα άλλο πρόβλημα είναι η αποθήκευση κιβωτίων (bin packing) και αφορά την εύρεση του βέλτιστου τρόπου αποθήκευσης ενός αριθμού κιβωτίων σε περιορισμένο χώρο και έχει μεγάλη πρακτική σημασία στη βιομηχανία. Τέλος στην κατηγορία αυτών των εφαρμογών εντάσσονται και τα προβλήματα καταμερισμού – χρονοπρογραμματισμού εργασιών (Job shop & Flow shop scheduling).

## 2.5 Τα βήματα του βασικού γενετικού αλγορίθμου

1. Αρχή – παραγωγή τυχαίου πληθυσμού από  $n$  χρωμοσώματα (κατάλληλες λύσεις για το πρόβλημα)
2. Καταλληλότητα – εκτίμηση της καταλληλότητας κάθε χρωμοσώματος του πληθυσμού (για παράδειγμα εξέταση περιορισμών).
3. Νέος πληθυσμός – δημιουργία νέου πληθυσμού επαναλαμβάνοντας τα επόμενα βήματα μέχρι ο νέος πληθυσμός να ολοκληρωθεί
  - a. Επιλογή – επιλογή δύο χρωμοσωμάτων-γονέων από τον πληθυσμό σύμφωνα με την καταλληλότητα τους (όσο πιο κατάλληλο είναι ένα χρωμόσωμα τόσο μεγαλύτερη η πιθανότητα να επιλεγεί)
  - b. Διασταύρωση – με μία καθορισμένη πιθανότητα διασταύρωσης διασταυρώνονται οι δύο γονείς για να δημιουργηθεί ο νέος απόγονος. Αν δεν γίνει διασταύρωση, ο απόγονος είναι ακριβές αντίγραφο των γονέων.
  - c. Μετάλλαξη – με μια καθορισμένη πιθανότητα μετάλλαξης, μεταλλάσσεται ο νέος απόγονος.
  - d. Αποδοχή – τοποθέτηση νέου απογόνου στο νέο πληθυσμό
4. Αντικατάσταση – χρησιμοποίηση νέου παραγόμενου πληθυσμού για περαιτέρω εκτέλεση του αλγορίθμου
5. Εξέταση – αν η συνθήκη τερματισμού ικανοποιείται τότε σταματάμε και έχουμε την καλύτερη λύση του προβλήματος
6. Βρόχος – πηγαίνουμε στο βήμα 2

## 2.6 Παράμετροι ενός γενετικού αλγορίθμου

### 2.6.1 Κωδικοποίηση ενός χρωμοσώματος

Κάθε χρωμόσωμα θα πρέπει με κάποιο τρόπο να εμπεριέχει πληροφορία για τη λύση που παριστάνει. Το πώς κωδικοποιούμε τα χρωμοσώματα είναι η πρώτη ερώτηση που κάνουμε όταν ξεκινάμε να λύνουμε ένα πρόβλημα με γενετικούς αλγορίθμους. Η κωδικοποίηση εξαρτάται κυρίως από τον τύπο του προβλήματος που έχουμε να αντιμετωπίσουμε.

Υπάρχουν πολλοί τρόποι κωδικοποίησης όπως είναι:

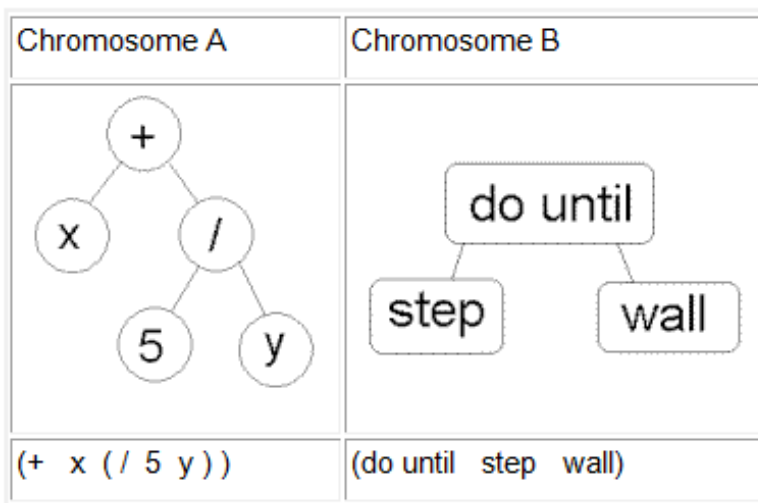
α) η δυαδική κωδικοποίηση (binary encoding) στην οποία το χρωμόσωμα αποτελείται από μια σειρά ψηφίων 0 ή 1

Chromosome A	10110010110010101110
Chromosome B	11111110000011000001

β) η κωδικοποίηση μετάθεσης (permutation encoding) όπου το χρωμόσωμα αποτελείται από αριθμούς που παριστάνουν μια θέση σε μια σειρά

Chromosome A      1 5 3 2 6 4 7 9 8  
 Chromosome B      8 5 6 7 2 3 1 4 9

γ) κωδικοποίηση δένδρου (tree encoding) όπου το χρωμόσωμα είναι ένα δένδρο κάποιων αντικειμένων



Εικόνα 2.1 Κωδικοποίηση Δένδρου

δ) η κωδικοποίηση τιμών (value encoding) στην οποία χρησιμοποιείται σειρά τιμών συνδεδεμένων με το πρόβλημα όπως πραγματικοί αριθμοί, χαρακτήρες ή οποιαδήποτε αντικείμενα

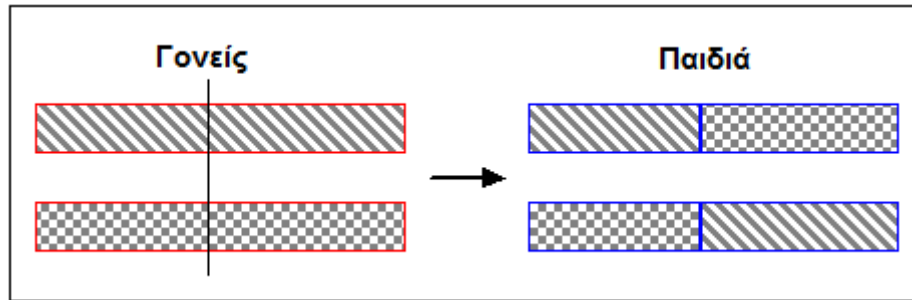
Chromosome A	1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B	A B D J E I F J D H D I E R J F D L D F L F E G T
Chromosome C	(back), (back), (right), (forward), (left)

Εικόνα 2.2 Κωδικοποίηση Τιμών

## 2.6.2 Διασταύρωση

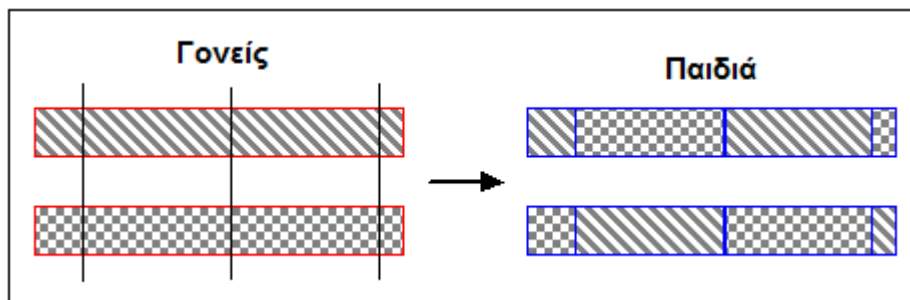
Αφού έχουμε αποφασίσει ποιο είδος κωδικοποίησης θα χρησιμοποιήσουμε, μπορούμε να προχωρήσουμε στη διαδικασία διασταύρωσης. Η διασταύρωση εφαρμόζεται σε επιλεγμένα γονίδια από τα χρωμοσώματα-γονείς και δημιουργεί νέο απόγονο.

Ο απλούστερος τρόπος για να γίνει η διασταύρωση είναι να επιλεγεί τυχαία πάνω στα χρωμοσώματα ένα σημείο διασταύρωσης και να αντιγραφούν όλα τα γονίδια πριν από το συγκεκριμένο σημείο από τον πρώτο γονέα και όλα τα γονίδια μετά το σημείο διασταύρωσης από τον δεύτερο γονέα στον απόγονο. Η διασταύρωση αυτή λέγεται **Διασταύρωση Ενός Σημείου** και φαίνεται στο παρακάτω σχήμα.



Εικόνα 2.3 Διασταύρωση Ενός Σημείου

Υπάρχουν και άλλοι τρόποι να εφαρμόσουμε διασταύρωση όπως το να επιλέξουμε περισσότερα του ενός σημεία διασταύρωσης. Στην **Πολυσημειακή Διασταύρωση** επιλέγονται ομοιόμορφα και τυχαία  $m$  σημεία και στη συνέχεια οι μεταβλητές μεταξύ των διαδοχικών σημείων διασταύρωσης ανταλλάσσονται μεταξύ των δύο γονέων για να παραγάγουν δύο νέους απογόνους.



Εικόνα 2.4 Πολυσημειακή Διασταύρωση

Ένα ακόμα είδος διασταύρωσης είναι η **Ομοιόμορφη Διασταύρωση**. Σύμφωνα με αυτήν, δύο γονείς λαμβάνουν μέρος σε μια διαδικασία ανταλλαγής γενετικού υλικού από την οποία προκύπτουν δύο παιδιά. Η διαδικασία λαμβάνει χώρα με βάση μια συμβολοσειρά που λέγεται φόρμα (template) ή μάσκα (mask). Αναλόγως με την τιμή του δυαδικού ψηφίου της μάσκας σε κάθε θέση καθορίζεται για κάθε παιδί ο γονέας από τον οποίο θα προέλθει το γενετικό υλικό για τη συγκεκριμένη θέση. Η μάσκα επιλέγεται με τυχαίο τρόπο. Η τιμή του δυαδικού ψηφίου της μάσκας σε κάθε θέσης της, προσδιορίζει το γονέα από τον οποίο θα προέλθει το δυαδικό ψηφίο για το παιδί. Όταν το δυαδικό ψηφίο της είναι 1, τότε το πρώτο παιδί παίρνει την αντίστοιχη τιμή του γονέα 1, ενώ όταν είναι 0 παίρνει την τιμή του γονέα 0. Το δεύτερο παιδί παίρνει το δυαδικό ψηφίο του άλλου γονέα κάθε φορά.

Η διασταύρωση μπορεί γενικώς να γίνει αρκετά πολύπλοκη διαδικασία και αυτό εξαρτάται κυρίως από την κωδικοποίηση των χρωμοσωμάτων. Ο σωστός τρόπος διασταύρωσης αν εφαρμοστεί στο σωστό πρόβλημα μπορεί να βελτιώσει αισθητά την απόδοση του γενετικού αλγορίθμου.

### 2.6.3 Μετάλλαξη

Αφού ολοκληρωθεί η διαδικασία της διασταύρωσης, ακολουθεί η μετάλλαξη. Η μετάλλαξη έχει σκοπό να εμποδίσει τον αλγόριθμο από το να παγιδευτεί μέσα σε τοπικά ακρότατα του προβλήματος σε κάθε επανάληψή του. Με τη μετάλλαξη μεταβάλλεται τυχαία ο απόγονος που προήλθε από τη διασταύρωση.

Η τεχνική της μετάλλαξης (όπως και της διασταύρωσης) εξαρτάται βασικά από την κωδικοποίηση του χρωμοσώματος. Στην δυαδική κωδικοποίηση, για παράδειγμα, μπορούμε να μετατρέψουμε μερικά τυχαία επιλεγμένα ψηφία από 0 σε 1 ή από 1 σε 0.

Original offspring 1	1101111000011110
Original offspring 2	1101100100110110
Mutated offspring 1	1100111000011110
Mutated offspring 2	1101101100110110

Εικόνα 2.5 Μετάλλαξη

Στην κωδικοποίηση μετάθεσης η μετάλλαξη μπορεί να γίνει με ανταλλαγή δύο γονιδίων του ίδιου χρωμοσώματος.

### 2.6.4 Πιθανότητα διασταύρωσης και μετάλλαξης

Πιθανότητα διασταύρωσης: πόσο συχνά συμβαίνει διασταύρωση. Αν δεν υπάρξει διασταύρωση, οι απόγονοι είναι ακριβή αντίγραφα των γονέων. Αν συμβεί διασταύρωση, οι απόγονοι εμπεριέχουν κομμάτια και από τους δύο γονείς από τους οποίους προήλθαν. Αν η πιθανότητα διασταύρωσης είναι 100%, τότε όλοι οι απόγονοι έχουν δημιουργηθεί μετά από διασταύρωση. Αν η συγκεκριμένη πιθανότητα είναι 0% τότε όλη η επόμενη γενιά είναι φτιαγμένη από ακριβή αντίγραφα χρωμοσωμάτων από την προηγούμενη γενιά (χωρίς αυτό να σημαίνει ότι η νέα γενιά είναι ακριβώς η ίδια με την προηγούμενη).

Πιθανότητα μετάλλαξης: πόσο συχνά κομμάτια (γονίδια) χρωμοσωμάτων μεταλλάσσονται. Αν δεν υπάρξει μετάλλαξη, οι απόγονοι παράγονται ακριβώς μετά τη διασταύρωση (ή κατευθείαν αντιγράφονται, αν δεν υπάρξει ούτε διασταύρωση) χωρίς καμία αλλαγή. Αν συμβεί μετάλλαξη, ένα ή περισσότερα γονίδια ενός χρωμοσώματος αλλάζουν. Αν η πιθανότητα μετάλλαξης είναι 100%, όλο το χρωμόσωμα αλλάζει, αν είναι 0%, τίποτα δεν αλλάζει. Η μετάλλαξη δεν θα πρέπει να συμβαίνει πολύ συχνά, διότι τότε ο γενετικός αλγόριθμος θα μετατραπεί σε τυχαία αναζήτηση, χάνοντας τα χαρακτηριστικά και τις ιδιότητές του.

### 2.6.5 Μέγεθος πληθυσμού

Ένα ακόμα βασικό ερώτημα είναι πόσα χρωμοσώματα αποτελούν τον πληθυσμό (σε μια γενιά); Αν υπάρχουν πολύ λίγα χρωμοσώματα, ο γενετικός αλγόριθμος έχει λίγες δυνατότητες να εφαρμόσει διασταύρωση και μόνο ένα μικρό κομμάτι του πεδίου αναζήτησης εξερευνείται. Από την

άλλη πλευρά, αν υπάρχουν πάρα πολλά χρωμοσώματα, ο γενετικός γίνεται υπολογιστικά απαιτητικότερος. Έρευνες δείχνουν ότι μετά από κάποιο όριο (το οποίο εξαρτάται κυρίως από την κωδικοποίηση και το είδος του προβλήματος) δεν είναι σκόπιμο να χρησιμοποιήσουμε πολύ μεγάλους πληθυσμούς γιατί ο αλγόριθμος λύνει το πρόβλημα σχετικά αργά.

## 2.6.6 Επιλογή

Από το σύνολο των χρωμοσωμάτων που προκύπτουν σε κάθε επανάληψη του αλγορίθμου, επιλέγονται κάποια και εφαρμόζεται σε αυτά διασταύρωση ώστε να προκύψουν νέοι απόγονοι. Υπάρχουν διάφορες μέθοδοι επιλογής, μερικές από τις οποίες παρουσιάζονται στη συνέχεια:

### Μέθοδος roulette wheel selection

Οι γονείς επιλέγονται σύμφωνα με την καταλληλότητά τους. Όσο καλύτερα είναι τα χρωμοσώματα, τόσο περισσότερες πιθανότητες έχουν να επιλεγούν. Ας φανταστούμε μια «ρουλέτας» (roulette wheel) όπου όλα τα χρωμοσώματα του πληθυσμού είναι τοποθετημένα. Το μέγεθος του τομέα που καταλαμβάνει κάθε χρωμόσωμα είναι ανάλογο της τιμής της συνάρτησης καταλληλότητας για το συγκεκριμένο χρωμόσωμα. Το μέγεθος του τομέα είναι ακόμα ανάλογο και με την πιθανότητα επιλογής κάθε χρωμοσώματος. Είναι ξεκάθαρο ότι τα χρωμοσώματα με τη μεγαλύτερη καταλληλότητα επιλέγονται περισσότερες φορές.

### Μέθοδος rank selection

Ο προηγούμενος τύπος επιλογής έχει προβλήματα όταν υπάρχουν μεγάλες διαφορές μεταξύ των τιμών των καταλληλοτήτων των χρωμοσωμάτων του πληθυσμού. Για παράδειγμα, αν το καλύτερο χρωμόσωμα έχει καταλληλότητα που είναι το 90% του αθροίσματος όλων των καταλληλοτήτων τότε τα άλλα χρωμοσώματα έχουν πολύ λίγες πιθανότητες να επιλεγούν. Η rank selection κατατάσσει τον πληθυσμό πρώτα με βάση την καταλληλότητα του κάθε χρωμοσώματος και έπειτα δίνει νέες τιμές στις καταλληλότητες με βάση πλέον την κατάταξη. Το χειρότερο θα έχει καταλληλότητα 1, το δεύτερο χειρότερο 2 και το καλύτερο N (N ο αριθμός των χρωμοσωμάτων στον πληθυσμό). Με αυτό τον τρόπο όλα τα χρωμοσώματα έχουν πιθανότητα να επιλεγούν. Βέβαια, με αυτή τη μέθοδο η σύγκλιση είναι βραδύτερη, αφού το καλύτερο χρωμόσωμα δε διαφέρει τόσο πολύ από τα υπόλοιπα.

### Η μέθοδος steady-state selection

Είναι μέθοδος επιλογής όχι για γονείς αλλά για επόμενη γενιά. Η βασική της ιδέα είναι να επιλέγει τα περισσότερα από τα χρωμοσώματα για την επόμενη γενιά και τα λιγότερο κατάλληλα να αντικαθιστά με κάποιους απογόνους.

# ΚΕΦΑΛΑΙΟ 3

## Οι Γενετικοί Αλγόριθμοι στη δημιουργία ωρολογίων προγραμμάτων

Η υπομονή είναι πικρή, αλλά ο καρπός της είναι γλυκός.

~Jean Jacques Rousseau (1712-1778)

### 3.1 Το κλασικό πρόβλημα χρονοπρογραμματισμού

Στις επόμενες παραγράφους του κεφαλαίου θα γίνει μια προσπάθεια ανάλυσης του προβλήματος χρονοπρογραμματισμού με τη βοήθεια των γενετικών αλγορίθμων. Στο πρόβλημα αυτό έχει γίνει εκτενής έρευνα [17][18][19][20][21][22].

Τα προβλήματα χρονοπρογραμματισμού όπως το δικό μας στη γενική περίπτωση είναι προβλήματα NP. Δηλαδή είναι δυσεπίλυτα προβλήματα, για τα οποία οι αλγόριθμοι επίλυσης (αν υπάρχουν) δεν τρέχουν σε πολυωνυμικό χρόνο. Για το λόγο αυτό, χρησιμοποιούμε γενετικούς αλγόριθμους, οι οποίοι δεν βρίσκουν τη βέλτιστη λύση, αλλά προσπαθούν να τη προσεγγίσουν. Η πολυπλοκότητα σε ένα πρόβλημα με τη χρήση γενετικού αλγορίθμου μπορεί να υπολογιστεί πειραματικά.

Τα απλούστερα προβλήματα δημιουργίας ωρολογίων προγραμμάτων μπορούν να θεωρηθούν ως προβλήματα ανάθεσης  $n$  γεγονότων (για παράδειγμα, εξετάσεις, σεμινάρια, προγράμματα, συναντήσεις) σε  $s$  διαθέσιμες χρονοσχισμές. Πιο τυπικά, έχουμε τον ακόλουθο ορισμό ενός απλού προβλήματος χρονοδιαγράμματος:

Έστω  $E$  το πεπερασμένο σύνολο  $n$  γεγονότων  $\{e_1, e_2, \dots, e_n\}$

Έστω  $T$  το πεπερασμένο σύνολο χρονοσχισμών  $\{t_1, t_2, \dots, t_s\}$ .

Οι χρονοσχισμές είναι οι χρονικές περίοδοι στις οποίες μπορεί να τοποθετηθεί ένα γεγονός. Στο παράδειγμα δημιουργίας ενός σχολικού προγράμματος, ένα γεγονός είναι η διδασκαλία ενός συγκεκριμένου μαθήματος ενώ η χρονοσχισμή είναι η σχολική ώρα.

Αν μια ανάθεση είναι ένα διατεταγμένο ζεύγος  $(\alpha, \beta)$ , τέτοιο ώστε  $\alpha \in E$  και  $\beta \in T$ , με την απλή ερμηνεία ότι «το γεγονός  $\alpha$  πραγματοποιείται στη χρονική στιγμή  $\beta$ », το πρόβλημα παραγωγής χρονοδιαγράμματος ανάγεται στο πρόβλημα του προσδιορισμού των  $n$  αναθέσεων, μια για κάθε γεγονός, οι οποίες παράγουν ένα καλό ωρολόγιο πρόγραμμα.

Η δημιουργία ωρολογίων προγραμμάτων συνοδεύεται από την παρουσία πολλών περιορισμών και στόχων που πρέπει να πληρούνται. Αυτοί οι περιορισμοί και οι στόχοι μπορεί να καταστήσουν πολλές (ή και όλες) τις  $s^V$  δυνατές διαφορετικές λύσεις «κακές» ή άκυρες. Ο πιο σημαντικός συνολικός περιορισμός (που ισχύει σε όλα τα προγράμματα) είναι ότι δεν πρέπει να υπάρχουν συγκρούσεις: τα γεγονότα τα οποία αναμένονται να μοιράζονται κοινούς μαθητές ή

εκπαιδευτικούς δεν θα πρέπει να είναι προγραμματισμένα να συμβούν ταυτόχρονα. Το γεγονός ότι ένα πρόσωπο δεν μπορεί να είναι σε δύο μέρη ταυτόχρονα είναι κεντρικής σημασίας για τα περισσότερα προβλήματα παραγωγής χρονοδιαγραμμάτων. Στη συνέχεια θα παρουσιάσουμε αναλυτικά τα διάφορα είδη περιορισμών που συναντά κανείς στα προβλήματα αυτά.

## 3.2 Περιορισμοί

### 3.2.1 Περιορισμοί ακμών

Οι περιορισμοί επί των οποίων δύο γεγονότα δεν μπορούν να μοιράζονται την ίδια χρονική σχισμή ονομάζονται ακμές (edges) [17]. Ένας χαρακτηριστικός τρόπος θεώρησης αυτών των περιορισμών είναι κατ'αναλογία με το πρόβλημα των χρωματικών γράφων. Στο πλαίσιο αυτής της αναλογίας, θεωρούμε το σύνολο των γεγονότων ως τις κορυφές ενός γράφου και το σύνολο των χρονοσχισμών  $T$  τα χρώματα (ή «ετικέτες») που μπορούν να χρησιμοποιηθούν για να ζωγραφίσει κανείς τις κορυφές του γράφου. Ο περιορισμός υπό εξέταση μπορεί στη συνέχεια να παρουσιαστεί ως μια ακμή μεταξύ των κορυφών  $e_i$  και  $e_j$  και το πρόβλημα δημιουργίας ωρολογίου προγράμματος αντιστοιχίζεται στο πρόβλημα χρωματισμού κάθε κορυφής με τέτοιο τρόπο ώστε να μην υπάρχουν κορυφές που να συνδέονται μεταξύ τους με μια ακμή και να είναι του ίδιου χρώματος.

Μέθοδοι που βασίζονται στη θεωρία γράφων μπορούν να χρησιμοποιηθούν για να λύσουν ένα απλό πρόβλημα δημιουργίας ωρολογίων προγραμμάτων. Δυστυχώς, στην πραγματικότητα τα προβλήματα είναι πιο σύνθετα και δεν μπορούν να καλυφθούν με τη χρήση γράφων. Στη συνέχεια παρουσιάζονται περιορισμοί, οι οποίοι μπορούν να ικανοποιηθούν με τη χρήση γενετικών αλγορίθμων.

### 3.2.2 Περιορισμοί διάταξης

Σε πολλές περιπτώσεις μπορεί να γίνει μερική διάταξη των γεγονότων στο  $E$ . Παραδείγματος χάριν, μπορεί να θέλει κανείς ένα εργαστηριακό μάθημα να γίνεται αργότερα από το αντίστοιχο θεωρητικό για προφανείς λόγους. Χωρίς την ύπαρξη περιορισμών διάταξης, ένα απλό πρόβλημα δημιουργίας ωρολογίων προγραμμάτων έχει την ιδιότητα ότι κάθε λύση να παραμένει λύση όταν υπόκειται σε αντιμετάθεση χρονοσχισμών.

Για παράδειγμα, έστω ότι το ακόλουθο πρόγραμμα είναι λύση σε ένα πρόβλημα με μόνο περιορισμούς ακμών:

( (e1,t1), (e2,t1), (e3,t1), (e4,t2), (e5,t2) )

Τότε, μπορεί κανείς να αντιμεταθέσει τα  $t_1$  και  $t_2$ , και το παραγόμενο πρόγραμμα να εξακολουθεί να είναι λύση:

( (e1,t2), (e2,t2), (e3,t2), (e4,t1), (e5,t1) )



Αυτό συμβαίνει γιατί αυτό που έχει σημασία είναι οι χρονοσχισμές για συγκεκριμένες ομάδες από γεγονότα να είναι διαφορετικές, η ίδια η χρονοσχισμή δεν έχει σημασία. Η εισαγωγή περιορισμών διάταξης καταργεί αυτή την ιδιότητα και καθιστά το πρόβλημα πιο δύσκολο.

### 3.2.3 Περιορισμοί γεγονότων που καλύπτουν πολλές σχισμές

Σε ένα χρονοδιάγραμμα εξετάσεων, είναι συχνά επιθυμητό οι εξετάσεις για κάθε μαθητή να είναι όσο το δυνατόν περισσότερο απλωμένες μέσα στη βδομάδα. Μια πιθανή εκδήλωση αυτής της επιθυμίας είναι να διευκρινιστεί ότι κανένας φοιτητής δε θα έχει πάνω από 2 εξετάσεις σε μια μέρα. Στη δημιουργία προγράμματος διαλέξεων, ένα κοινό παράδειγμα είναι ότι ένας φοιτητής δεν θα πρέπει να έχει να παρακολουθήσει 4 ή περισσότερες διαλέξεις μέσα σε μια μέρα και ιδανικά οι διαλέξεις θα πρέπει να είναι ομοιόμορφα κατανεμημένες κατά τη διάρκεια της εβδομάδας. Επίσης, διάφορες διαλέξεις στο ίδιο αντικείμενο θα πρέπει κατά προτίμηση να γίνονται σε διαφορετικές ημέρες. Ένα άλλο παράδειγμα θα μπορούσε να είναι ότι ορισμένες συνεδρίες σε εργαστήρια δεν μπορεί να είναι προγραμματισμένες για την ίδια ημέρα ώστε να υπάρχει χρόνος καθαρισμού και ετοιμασίας του εξοπλισμού ανάμεσα στην προηγούμενη και την επόμενη τάξη.

Οι εν λόγω περιορισμοί είναι σημαντικοί, αλλά μπορεί να είναι πολύ δύσκολο να ληφθούν υπόψη σε γενικές γραμμές και συνήθως αγνοούνται. Σε ορισμένες περιπτώσεις, κανείς χειρίζεται αυτούς του περιορισμούς τεχνητά ορίζοντας ορισμένα «εικονικά» γεγονότα τα οποία πρέπει να συμβαίνουν μεταξύ αυτών που πρόκειται να κατανεμηθούν.

### 3.2.4 Προκαθορισμένα γεγονότα και εξαιρέσεις

Μερικές φορές, μπορεί να υπάρχει απαίτηση εκ των προτέρων σε ένα συγκεκριμένο γεγονός να μην μπορεί να αποδοθεί κάποια χρονοσχισμή η οποία ανήκει σε ένα δεδομένο σύνολο χρονοσχισμών ή ανάποδα, ότι ορισμένες αναθέσεις πρέπει να είναι προκαθορισμένες. Σε ορισμένες περιπτώσεις, τέτοιες προδιαγραφές ή αποκλεισμοί είναι αυστηροί περιορισμοί, συνεπώς δεν μπορούμε να δεχτούμε ένα χρονοδιάγραμμα που δεν τους ικανοποιεί. Για τις περιπτώσεις αυτές, μια πιθανή προσέγγιση είναι να αφαιρεθεί κάθε γεγονός με προκαθορισμένη εκχώρηση από το  $E$ , και για να λυθεί το πρόβλημα του χρονοδιαγράμματος με τα υπόλοιπα γεγονότα και χρονοσχισμές. Κατά μία έννοια όταν συμβεί το παραπάνω ο χώρος αναζήτησης μειώνεται. Ωστόσο, σε γενικές γραμμές η προσθήκη προκαθορισμένων γεγονότων και αποκλεισμών οδηγεί σε περαιτέρω επιπλοκή.

Για παράδειγμα ένα γεγονός  $ef$  που έχει καθοριστεί σε μια χρονοσχισμή  $tf$ , μπορεί να μην επιτρέπεται να αφαιρεθεί από το  $E$ , αφού άλλα γεγονότα εξακολουθούν να συνδέονται με αυτό με κάποιους περιορισμούς. Για το λόγο αυτό, μια πιο γενική και ευέλικτη προσέγγιση είναι να αποφύγει κανείς την «αφαίρεση» του γεγονότος και αντί αυτού, να ορίσει μια ποινή για την παραβίαση ενός περιορισμού ή προδιαγραφής. Το μέγεθος της εν λόγω ποινής ορίζει εξάλλου και το πόσο απαιτείται η προδιαγραφή αυτή στην πραγματικότητα.

### 3.2.5 Περιορισμοί χωρητικότητας

Για να επαληθευτεί ότι δεν υπάρχει υπέρβαση στην χωρητικότητα μιας αίθουσας, θα πρέπει

πρώτα να γνωρίζουμε τη συνολική χωρητικότητα της αίθουσας, τη διαθεσιμότητα καθώς και το πλήθος των μαθητών που ανήκουν στη συγκεκριμένη τάξη. Σε ορισμένες περιπτώσεις, για παράδειγμα, αρκετές εξετάσεις μπορούν να διεξαχθούν ταυτόχρονα και να μοιράζονται το ίδιο μεγάλο δωμάτιο, αν δεν υπάρχουν κοινοί μαθητές. Κατά την επαλήθευση του προγράμματος, το σύστημα πρέπει να ελέγξει τη λίστα των περιορισμών και να υπολογίσει την ποινή για κάθε παραβίαση περιορισμού.

### 3.2.6 Σκληροί και μαλακοί περιορισμοί

Κάνοντας διάκριση μεταξύ των σκληρών περιορισμών που δεν πρέπει να παραβιάζονται και των μαλακών περιορισμών, ή προτιμήσεων, οι οποίες θα ήταν ευχάριστο να ικανοποιηθούν αλλά εάν όμως είναι απαραίτητο να παραβιαστούν, είναι από τις μεγαλύτερες δυσκολίες στην κατασκευή ωρολογίων προγραμμάτων. Αυτό ισχύει και για τους ανθρώπους, οι οποίοι συχνά αρχίζουν προσπαθώντας να λύσουν ένα συγκεκριμένο πρόβλημα θέτοντας συγκεκριμένα σύνολα από αυστηρούς και χαλαρούς περιορισμούς. Εάν αυτή η προσπάθεια αποτύχει, μπορεί να μετατρέψουν κάποιους από τους αυστηρούς περιορισμούς σε χαλαρούς, ή ακόμη και να διαγράψουν εξ ολοκλήρου μερικούς, μετά από εκ νέου διαπραγμάτευση με τους εμπλεκόμενους. Έτσι θα προχωρούν στη λύση ενός νέου προβλήματος, για το οποίο το έργο που έχει ήδη γίνει για τη λύση του αρχικού προβλήματος μπορεί να είναι σε μεγάλο βαθμό άχρηστο.

Η πιο συνηθισμένη προσέγγιση που βασίζεται σε γενετικούς αλγορίθμους για την αντιμετώπιση των περιορισμών, χρησιμοποιεί διαφορετικά επίπεδα ποινής για την παραβίαση διαφορετικών ειδών περιορισμών, έτσι ώστε να μπορεί να αποφύγει κανείς τη διάκριση μεταξύ αυστηρών και χαλαρών περιορισμών. Ως εκ τούτου, ένα χρονοδιάγραμμα το οποίο παραβαίνει μονάχα έναν ή δύο αυστηρούς περιορισμούς μπορεί τελικά να θεωρηθεί ως καλύτερη λύση από ότι ένα με μεγάλο αριθμό παραβιάσεων χαλαρών περιορισμών.

## 3.3 Γενετικοί αλγόριθμοι και ωρολόγια προγράμματα

### 3.3.1 Αναπαράσταση

Κατά τη χρήση γενετικών αλγορίθμων για την επίλυση ενός προβλήματος βελτιστοποίησης, το πρώτο σημαντικό βήμα είναι να επιλεγεί ο τρόπος αναπαράστασης μιας λύσης αυτού του προβλήματος ως ένα χρωμόσωμα, στο οποίο μπορεί στη συνέχεια να υποβληθεί μετάλλαξη. Η αναπαράσταση του χρωμοσώματος είναι συνήθως μια διατεταγμένη λίστα αριθμών. Αν για παράδειγμα υπάρχουν δέκα γεγονότα που θα προγραμματιστούν και οκτώ χρονοσχισμές, τότε ένα χρωμόσωμα σε αυτό το πρόβλημα είναι μια λίστα με μήκος δέκα (κάθε θέση σε αυτήν τη λίστα αντιπροσωπεύει ένα συγκεκριμένο γεγονός), όπου κάθε γονίδιο μπορεί να είναι ένας αριθμός από το ένα έως οκτώ. Για παράδειγμα, το χρωμόσωμα: 3 7 2 1 7 7 4 6 5 8, αντιπροσωπεύει ένα χρονοδιάγραμμα όπου το γεγονός 1 βρίσκεται στην χρονοσχισμή 3, το γεγονός 2 είναι στην 7, κλπ. Το κίνητρο για αυτή την αναπαράσταση είναι ότι μπορεί να χρησιμοποιηθεί οποιοδήποτε λογισμικό επεξεργασίας και λειτουργίας γενετικών αλγορίθμων χωρίς να χρειάζεται τροποποίηση.

Άλλη συνηθισμένη αναπαράσταση είναι αυτή κατά την οποία το χρωμόσωμα μοιάζει περισσότερο με το πραγματικό χρονοδιάγραμμα: οι θέσεις στην αριθμητική ακολουθία είναι οι

χρονοσχιμές ενώ οι τιμές που μπορεί να πάρει ένα «γονίδιο» είναι σύνολα από γεγονότα που θα μπορούσαν να συμβούν εκείνη τη στιγμή. Αυτού του είδους η εκπροσώπηση οδηγεί όμως σε ένα πρόβλημα που καλείται το πρόβλημα αντικατάστασης ετικέτας [12]. Το πρόβλημα είναι ότι κατά τη διασταύρωση μπορεί εύκολα να οδηγηθεί ο αλγόριθμος στην παραγωγή χρονοδιαγραμμάτων από τα οποία έχουν εξέλθει πολλά από τα γεγονότα που πρέπει να προγραμματιστούν. Ένας αλγόριθμος πρέπει να χρησιμοποιηθεί για να τροποποιήσει στη συνέχεια το χρωμόσωμα και να εισάγει εκ νέου όλα τα απαραίτητα γεγονότα. Η ανάλογη κατάσταση στο πρώτο είδος αναπαράστασης μας είναι ότι ένα χρωμόσωμα μπορεί να αντιπροσωπεύει χρονοδιαγράμματα στα οποία δεν είναι γεμάτες όλες οι διαθέσιμες χρονοσχιμές. Στη περίπτωση αυτή βέβαια, το ωρολόγιο πρόγραμμα είναι έγκυρο και πιθανώς χρήσιμο. Ως εκ τούτου, κατά τη χρήση αυτής της αναπαράστασης, ο γενετικός αλγόριθμος δεν χρειάζεται να χρησιμοποιεί επιπλέον τον αλγόριθμο αντικατάστασης ετικετών ο οποίος είναι υπολογιστικά χρονοβόρος.

Όταν θα πρέπει να χρησιμοποιηθούν περισσότερες από μία διάσταση, για παράδειγμα χρόνος και χώρος, για το πρόβλημα, η αναπαράσταση συνήθως αποτελείται από ένα χρωμόσωμα με μήκος δύο φορές μεγαλύτερο από ότι στην προηγούμενη αναπαράσταση. Το πρώτο μισό του χρωμοσώματος είναι μια λίστα με μήκος  $v$ , όπου  $v$  ο αριθμός των γεγονότων που θα προγραμματιστούν και κάθε στοιχείο της λίστας είναι ένας αριθμός μεταξύ 1 και  $s$ , όπου  $s$  ο αριθμός των διαθέσιμων χρονοσχιμών. Το δεύτερο μισό του χρωμοσώματος είναι μια λίστα με μήκος  $v$  (ο αριθμός των γεγονότων που πρέπει να προγραμματιστούν) και κάθε στοιχείο της είναι ένας αριθμός μεταξύ 1 και  $c$  όπου  $c$  ο αριθμός των διαθέσιμων αιθουσών.

Κατά τον ίδιο τρόπο, είναι πολύ εύκολο να επεκταθεί κανείς σε περισσότερες διαστάσεις. Για παράδειγμα, σε ένα χρωμόσωμα με μήκος  $3v$ ,  $v$  θέσεις δηλώνουν χρονοσχιμές για  $v$  γεγονότα,  $v$  θέσεις ορίζουν που θα πραγματοποιηθούν τα  $v$  γεγονότα και οι τελευταίες  $v$  θέσεις ορίζουν καθηγητές για τα  $v$  γεγονότα.

### 3.3.2 Χειρισμός περιορισμών

#### Χειρισμός εξαιρέσεων

Το μέγεθος του χώρου αναζήτησης μπορεί να μειωθεί, λαμβάνοντας υπόψη ότι κάποια γεγονότα θα πρέπει να εξαιρεθούν από ορισμένες χρονοσχιμές. Ένα αρχείο εξαιρέσεων μπορεί να αποκλείσει όλες τις μη επιτρεπόμενες χρονοσχιμές για κάθε γεγονός με σκοπό την ελαχιστοποίηση των μη αποδεκτών χρωμοσωμάτων και με αποτέλεσμα την ελαχιστοποίηση του συνολικού χώρου αναζήτησης. Με άλλα λόγια, ο χώρος αναζήτησης είναι μειωμένος, επειδή το σύνολο των χρονοσχιμών για τα γεγονότα αυτά είναι μικρότερο από τη μέγιστη. Μπορεί κανείς να διατυπώσει την παραπάνω κατάσταση ως εξής:

Έστω ότι έχουμε ένα σύνολο από γεγονότα  $E$ , για κάθε γεγονός  $e_i$  υπάρχει ένα σύνολο από επιτρεπτές χρονοσχιμές  $T_i$ . Οι έγκυρες χρονοσχιμές ανήκουν στο σύνολο:

$$S = \{ \{ (e_1, t_1) (e_2, t_2) \dots (e_n, t_n) \}; t_1 \in T_1, \dots, t_n \in T_n \}$$

Αν  $S_1 \in S$  και  $S_2 \in S$ , τότε ισχύει ότι η διασταύρωση των  $S_1$  και  $S_2$  ανήκει επίσης στο  $S$ , για όλες τις διασταυρώσεις που δεν αλλάζουν τη θέση ενός γονιδίου στο χρωμόσωμα. Επίσης, η μετάλλαξη του  $S_1 \in S$  θα παράξει ένα νέο μέλος του  $S$  αρκεί να οριστεί κατάλληλα, για παράδειγμα, για τη μετάλλαξη του  $k$ -οστού γονιδίου πρέπει να επιλέξουμε ένα τυχαίο μέλος του  $T_k$ .

Ο χειρισμός τέτοιων εξαιρέσεων είναι σχετικά απλός στους γενετικούς αλγόριθμους. Με τη χρήση ενός αρχείου εξαιρέσεων μπορεί να οριστεί το επιτρεπόμενο εύρος τιμών κάθε γονιδίου ξεχωριστά.

## Χειρισμός περιορισμών χωρητικότητας

Ένα άλλο κοινό πρόβλημα είναι το πρόβλημα της ανάθεσης γεγονότων σε τοποθεσίες και είναι συνήθως δυνατό ( και εύκολο) να γίνει η ανάθεση αιθουσών σε ένα συγκεκριμένο χρονοδιάγραμμα. Συνήθως στα σχολικά ωρολόγια προγράμματα η αίθουσα δε λαμβάνεται υπόψη, καθώς κάθε τάξη χρησιμοποιεί αποκλειστικά μία αίθουσα. Στη γενική περίπτωση όμως, εκτός από τα σύνολα  $E$  και  $T$ , υπάρχει ένα πεπερασμένο σύνολο  $L$  με τις πιθανές τοποθεσίες στις οποίες τα γεγονότα μπορούν να λάβουν χώρα. Η εκχώρηση αυτή είναι μία τριάδα  $(e,t,l)$ , και το πρόβλημα γίνεται ακόμη πιο περίπλοκο λόγω της έξτρα διάστασης. Όταν χρειάζεται να τεθεί ένας περιορισμός που αφορά χωρητικότητα, η διαδικασία είναι αντίστοιχη με αυτή στην περίπτωση εξαιρέσεων.

Έστω ότι έχουμε ένα σύνολο από γεγονότα  $E$ , για κάθε γεγονός  $e_i$  υπάρχει ένα σύνολο από επιτρεπτές αίθουσες  $L_i$ . Οι έγκυρες αίθουσες ανήκουν στο σύνολο:

$$S = \{ \{ (e_1, l_1) (e_2, l_2) \dots (e_n, l_n) \}; l_1 \in L_1, \dots, l_n \in L_n \}$$

Αν  $S_1 \in S$  και  $S_2 \in S$ , τότε ισχύει ότι η διασταύρωση των  $S_1$  και  $S_2$  ανήκει επίσης στο  $S$ , για όλες τις διασταυρώσεις που δεν αλλάζουν τη θέση ενός γονιδίου στο χρωμόσωμα. Επίσης, η μετάλλαξη του  $S_1 \in S$  θα παράξει ένα νέο μέλος του  $S$  αρκεί να οριστεί κατάλληλα, για παράδειγμα, για τη μετάλλαξη του  $k$ -οστού γονιδίου πρέπει να επιλέξουμε ένα τυχαίο μέλος του  $L_k$ .

## Χειρισμός προδιαγραφών

Ο χρήστης μπορεί να προδιαγράψει κάποιο γεγονός να συμβεί σε συγκεκριμένη χρονοσχισμή σε κάποιο αρχείο. Το αρχείο αυτό είναι παρόμοιο με το αρχείο εξαιρέσεων. Για παράδειγμα, η προδιαγραφή το γεγονός  $ek$  να συμβεί στη χρονοσχισμή  $T_j$  θα μπορούσε να οριστεί ως κανόνας εξαιρέσεων:  $ek$  δεν πρέπει να πάρει τιμές  $\{ t_1, \dots, t_{j-1}, t_{j+1}, \dots, t_s \}$ .

### 3.3.3 Εκτέλεση αλγορίθμου

#### Αρχικοποίηση

Κατά την αρχικοποίηση επιλέγεται ένα τυχαίο χρωμόσωμα από το χώρο των επιτρεπτών χρωμοσωμάτων. Στη συνέχεια τα γονίδια στο χρωμόσωμα είναι όλα έγκυρα και υπόκεινται βηματικά σε επιλογή, διασταύρωση, μετάλλαξη και εκτίμηση από γενιά σε γενιά.

## Συνάρτηση καταλληλότητας

Δεδομένου ενός χώρου  $P$  από πιθανές λύσεις στο πρόβλημα, υπάρχουν τρεις επιθυμητές ιδιότητες για τη συνάρτηση καταλληλότητας  $f(p)$  με  $p \in P$ :

- η  $f(p)$  θα θέλαμε να είναι αύξουσα συνάρτηση της ποιότητας του  $p$  ως υποψήφια λύση στο πρόβλημα, έτσι ώστε οι ιδανικές λύσεις να βρίσκονται στα ολικά μέγιστα της  $f$ ,
- η τιμή της  $f$  θα πρέπει να αντιστοιχεί σε πληροφορία για την ποιότητα του  $p$  ως λύση,
- η  $f$  να αλλάζει με τέτοιο τρόπο ώστε να φαίνεται ότι καθώς αλλάζει το  $p$ , τείνει προς την ιδανική λύση.

Η ποιότητα της λύσης  $p$  μπορεί να μην αλλάζει ομαλά καθώς τα γονίδια του  $p$  αλλάζουν. Για παράδειγμα σε εφαρμογές παραγωγής χρονοδιαγραμμάτων [18], το χρωμόσωμα μπορεί να μην αποτελεί τη λύση αντίθετα μπορεί να αποτελεί μια ακολουθία βημάτων προκειμένου να χτιστεί η λύση. Μια μικρή αλλαγή σε ένα αρχικό μέλος της ακολουθίας μπορεί να οδηγήσει στην κατασκευή μιας σημαντικά διαφορετικής υποψήφιας λύσης.

Στο πρόβλημα κατασκευής ωρολογίων προγραμμάτων, μια λύση η οποία ικανοποιεί όλους τους περιορισμούς είναι ιδανική λύση. Επιπλέον, είναι λογικό να ξεχωρίσει κανείς τη καταλληλότητα διαφορετικών προγραμμάτων με βάση το πλήθος και το είδος των περιορισμών που καταπατούνται. Μια απλή λύση είναι να χρησιμοποιήσει κανείς μια συνάρτηση αντιστρόφως ανάλογη στο πλήθος των περιορισμών που καταπατούνται. Για παράδειγμα, αν  $V(p)$  το πλήθος των περιορισμών που δεν ικανοποιήθηκαν στη λύση  $p$ , η συνάρτηση καταλληλότητας μπορεί να γραφεί:  
 $f(p) = 1/(1+V(p))$  (3.1).

Έτσι αν έχουν ικανοποιηθεί όλοι οι περιορισμοί, η  $f(p)$  έχει τη μέγιστη τιμή της 1. Το πρόβλημα όμως είναι ότι η συνάρτηση αυτή δεν έχει καμιά πληροφορία για το είδος των κανόνων που καταπατήθηκαν και δε ξεχωρίζει τους περιορισμούς με βάση τη σημαντικότητά τους.

Συνεπώς, η λύση είναι να ορίσει κανείς διαφορετικό βάρος ανά είδος περιορισμού και η ποινή στη συνάρτηση καταλληλότητας να είναι ανάλογη της σημασίας κάθε περιορισμού. Στην απλούστερη μορφή της, μια τέτοια συνάρτηση καταλληλότητας, με  $n$  είδη περιορισμών, με την ποινή/βάρος για τον τύπο  $i$  να είναι  $w_i$ , και το πλήθος των περιορισμών που δεν ικανοποιήθηκαν για το είδος  $i$  να είναι  $c_i(p)$ , στη λύση  $p$  δίνεται ως εξής [11]:

$$f(p) = \frac{1}{1 + \sum_{i=1}^n w_i c_i(p)}$$

(3.2)

## Μετάλλαξη

Η μετάλλαξη μπορεί να γίνεται τυχαία, επιλέγοντας και αλλάζοντας τυχαία κάποια γονίδια από το χρωμόσωμα κάθε φορά. Αν όμως χρησιμοποιούνται κάποιοι κανόνες ώστε η μετάλλαξη να γίνεται πιο στοχευμένα στην επιθυμητή λύση, ο γενετικός αλγόριθμος μπορεί να είναι πιο αποδοτικός. Πιο συγκεκριμένα, προτείνονται στη δημοσίευση [17] τέσσερις κανόνες μετάλλαξης:

- Επιλογή γονιδίου που αντιστοιχεί σε γεγονός το οποίο έχει τη μέγιστη ποινή στη συγκεκριμένη λύση και τυχαία αλλαγή της χρονοσχισμής που του έχει ανατεθεί
- Επιλογή γονιδίου που αντιστοιχεί σε γεγονός το οποίο έχει τη μέγιστη ποινή στη συγκεκριμένη λύση και επιλογή τιμής τέτοια ώστε να έχει την ελάχιστη δυνατή ποινή.

- Τυχαία επιλογή γονιδίου από αυτά που έχουν τις μεγαλύτερες ποινές στη συγκεκριμένη λύση και τυχαία αλλαγή της χρονοσχισμής που του έχει ανατεθεί
- Τυχαία επιλογή γονιδίου από αυτά που έχουν τις μεγαλύτερες ποινές στη συγκεκριμένη λύση και τυχαία επιλογή νέας χρονοσχισμής από αυτές που έχουν τις μικρότερες ποινές.

Οι κανόνες αυτοί μπορούν να υπολογιστούν κατά τον υπολογισμό της συνάρτησης κόστους. Αυξάνουν την πολυπλοκότητα του αλγόριθμου, αλλά σε γενικές γραμμές μειώνουν σημαντικά τον αριθμό επαναλήψεων που θα χρειαστούν μέχρι να βρεθεί μια ιδανική λύση.

### **Συνθήκη Τερματισμού**

Η συνθήκη τερματισμού προκύπτει από το συνδυασμό ενός μέγιστου αριθμού και από τη συνάρτηση καταλληλότητας. Αν η συνάρτηση καταλληλότητας δώσει μία τιμή πάνω από ένα προκαθορισμένο κατώφλι, ο αλγόριθμος τερματίζει.

### **ΜΕΡΟΣ ΙΙΙ**

## **Μελέτη της εφαρμογής FET καθώς και άλλων ωρολόγιων εφαρμογών**

# ΚΕΦΑΛΑΙΟ 4

## Η ωρολόγια εφαρμογή FET και η παραμετροποίηση της

Αν Α ισούται με την επιτυχία, τότε η εξίσωση έχει ως εξής:  $A=X+\Psi+Z$ , όπου X είναι εργασία, Ψ είναι παιχνίδι και Z είναι να κρατάς το στόμα σου κλειστό.

~Albert Einstein (1879-1955)

### 4.1 Περιγραφή της ωρολόγιας εφαρμογής FET

Το FET (Free Timetabling Software) είναι ένα δωρεάν λογισμικό ανοικτού κώδικα (open source) το οποίο δημιουργεί αυτόματα ωρολόγια προγράμματα για ένα σχολείο ή ένα πανεπιστήμιο. Χρησιμοποιεί έναν γρήγορο και αποδοτικό αλγόριθμο που έχει ως βάση τους Γενετικούς Αλγορίθμους που περιγράψαμε σε προηγούμενα κεφάλαια. Για την ακρίβεια το FET λύνει ένα περίπλοκο ωρολόγιο πρόγραμμα σε μέγιστο χρόνο 5 έως 20 λεπτά, ενώ για απλούστερα ωρολόγια προγράμματα ενδέχεται να κάνει ελάχιστα δευτερόλεπτα. Τέλος, για ασυνήθιστα δύσκολα προβλήματα ο χρόνος επίλυσης μπορεί να είναι αρκετά μεγάλος, ίσως και μερικές ώρες.

#### 4.1.1 Χαρακτηριστικά του FET

Το FET έχει τα εξής χαρακτηριστικά:

- Είναι ελεύθερο λογισμικό ανοικτού κώδικα (open source). Ο καθένας μπορεί ελεύθερα να το χρησιμοποιήσει, να το αντιγράψει, να το τροποποιήσει και να το αναδιανέμει όπως επιτρέπει η GNU GPL.
- Είναι μεταφρασμένο σε Αγγλικά(ΗΠΑ), Αραβικά, Καταλανικά, Γερμανικά, Ισπανικά, Περσικά, Γαλλικά, Ελληνικά, Δανέζικα, Γαλικίστικα, Εβραϊκά, Ουγγρικά, Ινδονησιακά, Ιταλικά, Λιθουανέζικα, Μακεδονικά, Μαλαισιακά, Ολλανδικά, Πολωνέζικα, Πορτογαλικά (Βραζιλίας), Ρουμανικά, Ρωσικά, Σλοβακικά, Σερβικά, Τούρκικα, Ουκρανικά και ατελώς μεταφρασμένο για ορισμένες ακόμα γλώσσες.
- Ο αλγόριθμος που παράγει τα ωρολόγια προγράμματα, μπορεί να λειτουργεί εντελώς αυτόματα ή ημιαυτόματα επιτρέποντας στο χρήστη να ορίσει κάποια μαθήματα/ώρες/αίθουσες.
- Είναι ανεξάρτητη εφαρμογή που μπορεί να τρέχει σε Linux, Windows, Mac και σε οποιοδήποτε σύστημα που υποστηρίζει η Qt.
- Για το αρχείο εισόδου χρησιμοποιείται μία ευέλικτη μορφή XML επιτρέποντας την επεξεργασία από ένα πρόγραμμα ή με το χέρι.



- Το αποτέλεσμα των ωρολογίων προγραμμάτων εξάγεται σε HTML, XML και CSV μορφή.
- Πολύ ευέλικτη δομή φοιτητών (μαθητών) οργανωμένη σε έτη, ομάδες και υποομάδες. Ακόμη μπορούμε να ορίσουμε άτομα (φοιτητές) ως ξεχωριστά σύνολα.
- Κάθε περιορισμός έχει εκατοστιαία αναλογία βάρους (από 0.0% έως 100.0%) παρ'όλα αυτά κάποιοι ειδικοί περιορισμοί μπορούν να έχουν μόνο 100% ποσοστό βάρους.
- Υπάρχουν τοπικοί (space) και χρονικοί (time) περιορισμοί.
- Τέλος, υπάρχουν τα παρακάτω όρια για τον αλγόριθμο.
  - Μέγιστος συνολικός αριθμός ωρών (περιόδων) ανά ημέρα: 60
  - Μέγιστος αριθμός εργασιμων ημερών ανά εβδομάδα: 35
  - Μέγιστος συνολικός αριθμός διδασκόντων: 6000
  - Μέγιστος συνολικός αριθμός συνόλων των φοιτητών: 30000
  - Μέγιστος συνολικός αριθμός μαθημάτων: 6000
  - Σχεδόν απεριόριστο αριθμό ετικετών δραστηριότητας (activity tags)
  - Μέγιστος αριθμός δραστηριοτήτων: 30000
  - Μέγιστος αριθμός αιθουσών: 6000
  - Μέγιστος αριθμός κτηρίων: 6000
  - Ρυθμιζόμενη διάρκεια για κάθε δραστηριότητα (κάθε δραστηριότητα μπορεί να έχει διάρκεια 1 ώρα, 2 ώρες, 3 ώρες ή περισσότερο)
  - Δυνατότητα προσθήκης πολλαπλών διδασκόντων και φοιτητών για κάθε δραστηριότητα
  - Μέγιστος αριθμός χρονικών περιορισμών: 60000
  - Μέγιστος αριθμός τοπικών περιορισμών: 60000

## 4.2 Παραμετροποίηση της ωρολόγιας εφαρμογής FET για το τμήμα της Πληροφορικής με Εφαρμογές στη Βιοϊατρική

Για να ξεκινήσει η παραμετροποίηση της ωρολόγιας εφαρμογής FET και να δημιουργηθούν ωρολόγια προγράμματα για το χειμερινό και το εαρινό εξάμηνο αρκεί να συλλέξουμε πληροφορίες για το τμήμα της Πληροφορικής με Εφαρμογές στη Βιοϊατρική. Οι πληροφορίες πρέπει να αφορούν τους διδάσκοντες, τα μαθήματα, τα έτη και τους χώρους φοίτησης και τους περιορισμούς χρόνου και τόπου. Έτσι, λοιπόν, χωρίζουμε τα παρακάτω δεδομένα σε δύο κατηγορίες οι οποίες είναι το χειμερινό εξάμηνο και το εαρινό εξάμηνο του έτους 2010-2011 και τις καταχωρούμε στην εφαρμογή FET για να παράγουμε τα αντίστοιχα ωρολόγια προγράμματα.

## 4.2.1 Χειμερινό Εξάμηνο 2010-2011

Το χειμερινό εξάμηνο περιλαμβάνει τα παρακάτω δεδομένα τα οποία θα καταχωρηθούν στην εφαρμογή FET.

**Διδάσκοντες:** Αδάμ, Αλετράς, Αναγνωστόπουλος, Αποστολόπουλος, Βάβουλας, Βασιλάκη, Βασιλακόπουλος, Βαϊόπουλος, Γεωργίου, Γκανέτσος, Γούγα, Δελήμπασης, Ιακωβίδης, Κακαρούντας, Καραγεώργος, Καρυδάκης, Κοσμόπουλος, Μάρκου, Μαγκλογιάννης, Μπάγκος, Μπράλιου, Παπανάγνου, Παππά, Παρασκευοπούλου, Πλαγιανάκος, Σίσκου, Σανδαλίδης, Σπαθούλας, Χατζηδήμου.

**Έτη φοίτησης:** Πρώτο Έτος, Δεύτερο Έτος, Τρίτο Έτος, Τέταρτο Έτος

**Αίθουσες:** Αίθουσα 1, Αίθουσα 2, Αίθουσα 3, Εργαστήριο 1, Εργαστήριο 2, Εργαστήριο 3, Εργαστήριο Ανατομίας.

**Μαθήματα Α' Εξαμήνου:** Γραμμική Άλγεβρα, Εισαγωγή στον Προγραμματισμό, Φυσική, Βιοχημεία, Εισαγωγή στην Πληροφορική, Αγγλικά I, Μαθηματική Ανάλυση I.

**Μαθήματα Γ' Εξαμήνου:** Αρχές Ηλεκτρονικής, Πιθανότητες & Στοιχεία Στατιστικής, Βιολογία II, Αριθμητική Ανάλυση, Αγγλικά III, Ανατομία & Φυσιολογία I, Αρχιτεκτονική Υπολογιστών.

**Μαθήματα Ε' Εξαμήνου:** Ενσωματωμένα Συστήματα Υπολογιστών, Γενετική & Ασθένειες- Πληθυσμιακή Γενετική, Εισαγωγή στις Τηλεπικοινωνίες, Στοιχεία Θεωρίας Πληροφορίας & Κωδικών, Ψηφιακή Επεξεργασία Βιοσημάτων, Βιοπληροφορική I, Γραφική με Υπολογιστές, Τηλεϊατρική, Βάσεις Δεδομένων, Τεχνολογία Λογισμικού.

**Μαθήματα Ζ' Εξαμήνου:** Οργάνωση και Διοίκηση Συστημάτων Υγείας, Θεωρία Υπολογισμού, Επικοινωνία Ανθρώπου Υπολογιστή, Συστήματα Στήριξης Ιατρικών Αποφάσεων, Συστήματα Ιατρικής Απεικόνισης, Πληροφοριακά Συστήματα Υγείας II, Τεχνητή Νοημοσύνη, Τεχνολογίες Εφαρμογών Διαδικτύου, Θεωρία Γράφων.

**Ελεύθερα Μαθήματα:** Κοινωνιολογία, Παιδαγωγικά, Εισαγωγή στην Πολιτική Οικονομία, Δημόσιες Σχέσεις.

Η καταχώρηση των παραπάνω δεδομένων καθώς και άλλων γίνεται με την έναρξη της εφαρμογής FET. Ανοίγουμε την εφαρμογή (εικόνα 4.1) και δημιουργούμε ένα καινούργιο αρχείο με το όνομα “xeimerino.fet” το οποίο θα αφορά το χειμερινό εξάμηνο του έτους 2010-2011 (εικόνα 4.2). Ύστερα, από το μενού επιλέγουμε το θέμα Data στο οποίο θα καταχωρήσουμε όλα τα δεδομένα του χειμερινού εξαμήνου.

Αρχικά προσθέτουμε το όνομα του πανεπιστημίου (Πανεπιστήμιο Στεράς Ελλάδας) επιλέγοντας Basic-->Institution information-->Institution name (εικόνες 4.3 και 4.4). Συνεχίζουμε με την καταχώρηση του ονόματος του τμήματος (Πληροφορική με Εφαρμογές στη Βιοϊατρική) επιλέγοντας Basic-->Institution information-->Comments (εικόνα 4.5). Ύστερα επιλέγουμε Basic-->Days and hours-->Days per week για να δηλώσουμε τις ημέρες που θέλουμε να έχει το ωρολόγιο πρόγραμμα (εικόνες 4.6 και 4.7). Και για να δηλώσουμε τις ώρες για την κάθε ημέρα επιλέγουμε Basic-->Days and hours -->Hours(periods) per day (εικόνες 4.6 και 4.8).

Για την καταχώρηση των διδασκόντων επιλέγουμε από το μενού Data-->Teachers (εικόνα 4.9 και 4.10). Για την καταχώρηση των ετών φοίτησης επιλέγουμε από το μενού Data-->Students-->Years (forms, classes) (εικόνα 4.11) και καταχωρούμε όλες τις ομάδες φοιτητών (εικόνα 4.12). Για την καταχώρηση των μαθημάτων του χειμερινού εξαμήνου επιλέγουμε από το μενού Data-->

Subjects (εικόνα 4.9). Μαθήματα του χειμερινού εξαμήνου είναι τα μαθήματα του Α', Γ', Ε', Ζ' εξαμήνου καθώς και τα αντίστοιχα των ελευθέρων μαθημάτων (εικόνα 4.13). Αφού έχουν καταχωρηθεί τα παραπάνω στοιχεία προσθέτουμε τους χώρους φοίτησης επιλέγοντας από το μενού Data-->Space-->Buildings και Data-->Space-->Rooms (εικόνες 4.14, 4.15, 4.16, 4.17, 4.18).

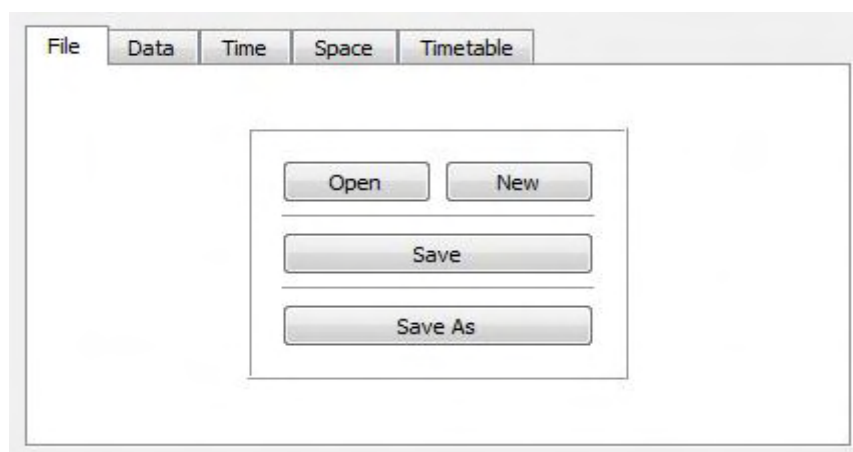
Στη συνέχεια καταχωρούμε τις δραστηριότητες του τμήματος επιλέγοντας από το μενού Data-->Activities (εικόνα 4.9). Όταν λέμε δραστηριότητες εννοούμε τα μαθήματα που γίνονται από κάποιους διδάσκοντες με συγκεκριμένο ωράριο και έτος φοίτησης. Παρακάτω παραθέτω τις δραστηριότητες του χειμερινού εξαμήνου (πίνακας 4.1) τις οποίες και καταχωρούμε κάθε μία όπως παρουσιάζεται στην εικόνα 4.19.

<b>ΜΑΘΗΜΑ</b>	<b>ΔΙΔΑΣΚΩΝ</b>	<b>ΔΙΑΡΚΕΙΑ</b>	<b>ΕΤΟΣ ΦΟΙΤΗΣΗΣ</b>
Γραμμική Άλγεβρα	Αδάμ	2 Δίωρα	Πρώτο Έτος
Εισαγωγή στον Προγραμματισμό Εργαστήριο	Βαϊόπουλος	1 Δίωρο	Πρώτο Έτος
Εισαγωγή στον Προγραμματισμό Εργαστήριο	Σπαθούλας	1 Δίωρο	Πρώτο Έτος
Εισαγωγή στον Προγραμματισμό Εργαστήριο	Βάβουλας	1 Δίωρο	Πρώτο Έτος
Εισαγωγή στον Προγραμματισμό Εργαστήριο	Αποστολόπουλος	1 Δίωρο	Πρώτο Έτος
Φυσική	Γκανέτσος	1 Τρίωρο και 1 Δίωρο	Πρώτο Έτος
Εισαγωγή στον Προγραμματισμό	Μάρκου	1 Τρίωρο	Πρώτο Έτος
Βιοχημεία	Μπράλιου	2 Δίωρα	Πρώτο Έτος
Εισαγωγή στην Πληροφορική Εργαστήριο	Σπαθούλας Αποστολόπουλος	2 Δίωρα	Πρώτο Έτος
Αγγλικά Ι	Παππά	1 Δίωρο	Πρώτο Έτος
Μαθηματική Ανάλυση Ι	Αδάμ	1 Δίωρο και 1 Τρίωρο	Πρώτο Έτος
Εισαγωγή στην Πληροφορική	Αναγνωστόπουλος	1 Δίωρο	Πρώτο Έτος
Αρχές Ηλεκτρονικής	Γκανέτσος	1 Τρίωρο	Δεύτερο Έτος
Αρχές Ηλεκτρονικής Εργαστήριο	Βάβουλας Βαϊόπουλος	3 Δίωρα	Δεύτερο Έτος
Πιθανότητες και Στοιχεία Στατιστικής	Γεωργίου	2 Δίωρα	Δεύτερο Έτος
Βιολογία ΙΙ	Μπάγκος	2 Δίωρα	Δεύτερο Έτος
Αριθμητική Ανάλυση	Πλαγιανάκος	1 Δίωρο	Δεύτερο Έτος
Αγγλικά ΙΙΙ	Παππά	1 Δίωρο	Δεύτερο Έτος
Ανατομία και Φυσιολογία	Παπανάγνου	1 Ώρα και 1 Τρίωρο	Δεύτερο Έτος

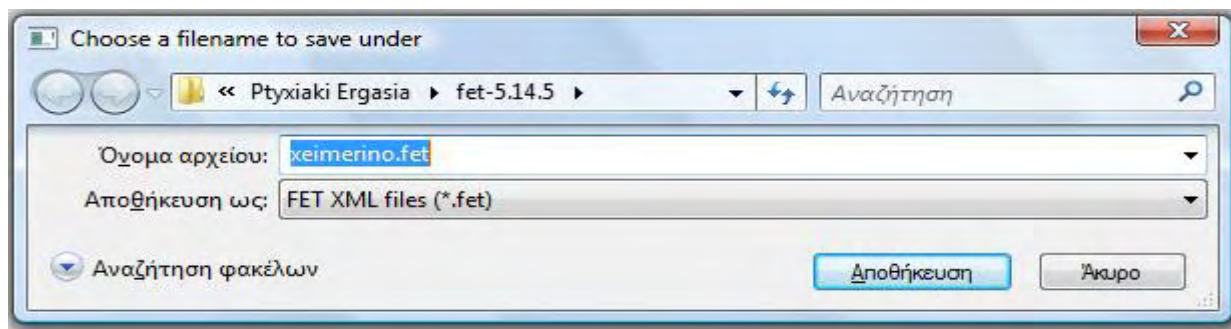
I			
Αριθμητική Ανάλυση Εργαστήριο	Βάβουλας	2 Δίωρα	Δεύτερο Έτος
Αριθμητική Ανάλυση Εργαστήριο	Βαϊόπουλος	1 Δίωρο	Δεύτερο Έτος
Αρχιτεκτονική Υπολογιστών	Κακαρούντας	2 Δίωρα	Δεύτερο Έτος
Βάσεις Δεδομένων Εργαστήριο	Βαϊόπουλος	1 Δίωρο	Τρίτο Έτος
Βάσεις Δεδομένων Εργαστήριο	Βάβουλας	1 Δίωρο	Τρίτο Έτος
Ενσωματωμένα Συστήματα Υπολογιστών	Αλετράς	1 Δίωρο	Τρίτο Έτος
Ενσωματωμένα Συστήματα Υπολογιστών	Καραγεώργος Χατζηδήμου	1 Δίωρο	Τρίτο Έτος
Γενετική και Ασθένειες Πληθυσμιακή Γενετική	Μπράλιου	2 Δίωρα	Τρίτο Έτος
Εισαγωγή στις Τηλεπικοινωνίες	Σανδαλίδης	1 Δίωρο	Τρίτο Έτος
Εισαγωγή στις Τηλεπικοινωνίες Εργαστήριο	Βάβουλας	1 Δίωρο	Τρίτο Έτος
Εισαγωγή στις Τηλεπικοινωνίες Εργαστήριο	Βαϊόπουλος	1 Δίωρο	Τρίτο Έτος
Ψηφιακή Επεξεργασία Βιοσημάτων	Δελήμπασης	1 Δίωρο	Τρίτο Έτος
Ψηφιακή Επεξεργασία Βιοσημάτων Εργαστήριο	Δελήμπασης	1 Δίωρο	Τρίτο Έτος
Βιοπληροφορική I	Μπάγκος	2 Δίωρα	Τρίτο Έτος
Γραφική με Υπολογιστές	Δελήμπασης	2 Δίωρα	Τρίτο Έτος
Στοιχεία Θεωρίας Πληροφορίας και Κωδικών	Σανδαλίδης	2 Δίωρα	Τρίτο Έτος
Τηλεϊατρική	Μαγκλογιάννης	2 Δίωρα	Τρίτο Έτος
Βάσεις Δεδομένων	Βασιλακόπουλος	2 Δίωρα	Τρίτο Έτος
Τεχνολογία Λογισμικού	Κοσμόπουλος	2 Δίωρα	Τρίτο Έτος
Οργάνωση και Διοίκηση Συστημάτων Υγείας	Σίσκου	2 Δίωρα	Τέταρτο Έτος
Θεωρία Υπολογισμού	Μάρκου	2 Δίωρα	Τέταρτο Έτος

Επικοινωνία Ανθρώπου Υπολογιστή	Καρυδάκης	2 Δίωρα	Τέταρτο Έτος
Συστήματα Στήριξης Ιατρικών Αποφάσεων	Ιακωβίδης	2 Δίωρα	Τέταρτο Έτος
Συστήματα Ιατρικής Απεικόνισης	Αλετράς	2 Δίωρα	Τέταρτο Έτος
Πληροφοριακά Συστήματα Υγείας II	Μαγκλογιάννης	2 Δίωρα	Τέταρτο Έτος
Τεχνητή Νοημοσύνη	Πλαγιανάκος	2 Δίωρα	Τέταρτο Έτος
Τεχνολογίες Εφαρμογών Διαδικτύου	Αναγνωστόπουλος	2 Δίωρα	Τέταρτο Έτος
Θεωρία Γράφων	Βασιλακόπουλος	2 Δίωρα	Τέταρτο Έτος
Κοινωνιολογία	Γούγα	1 Δίωρο	Ελεύθερη Επιλογή
Παιδαγωγικά	Παρασκευοπούλου	1 Δίωρο	Ελεύθερη Επιλογή
Εισαγωγή στην Πολιτική Οικονομία	Βασιλάκη	1 Δίωρο	Ελεύθερη Επιλογή
Δημόσιες Σχέσεις	Βασιλάκη	1 Δίωρο	Ελεύθερη Επιλογή

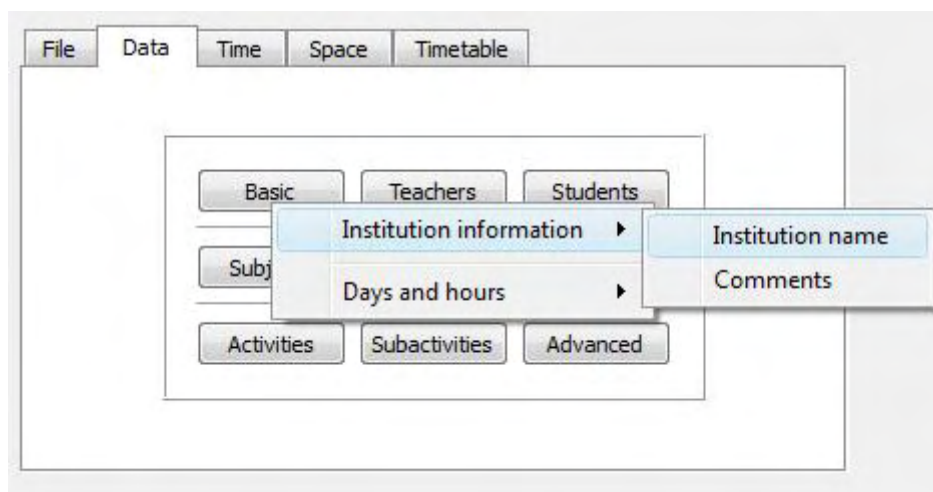
Πίνακας 4.1: Δραστηριότητες Χειμερινού Εξαμήνου



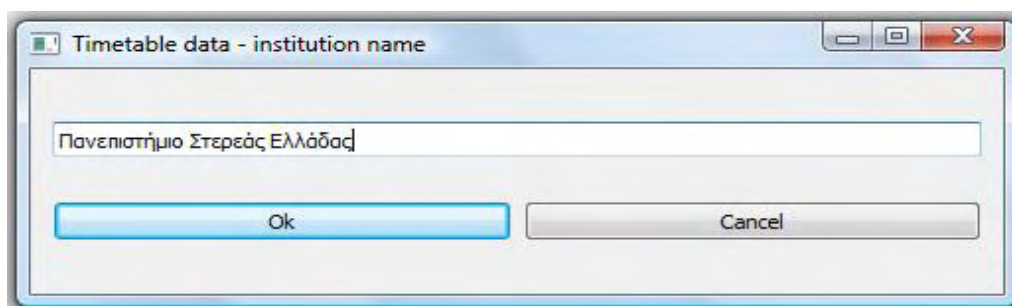
Εικόνα 4.1: Μενού →File →New



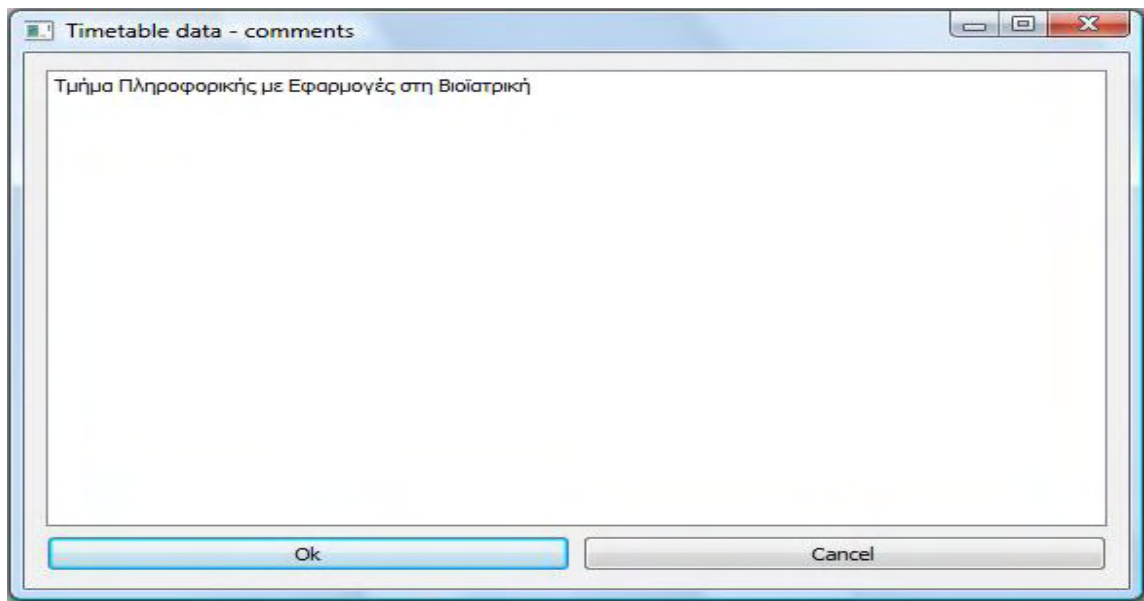
Εικόνα 4.2: Δημιουργία αρχείου για το χειμερινό εξάμηνο



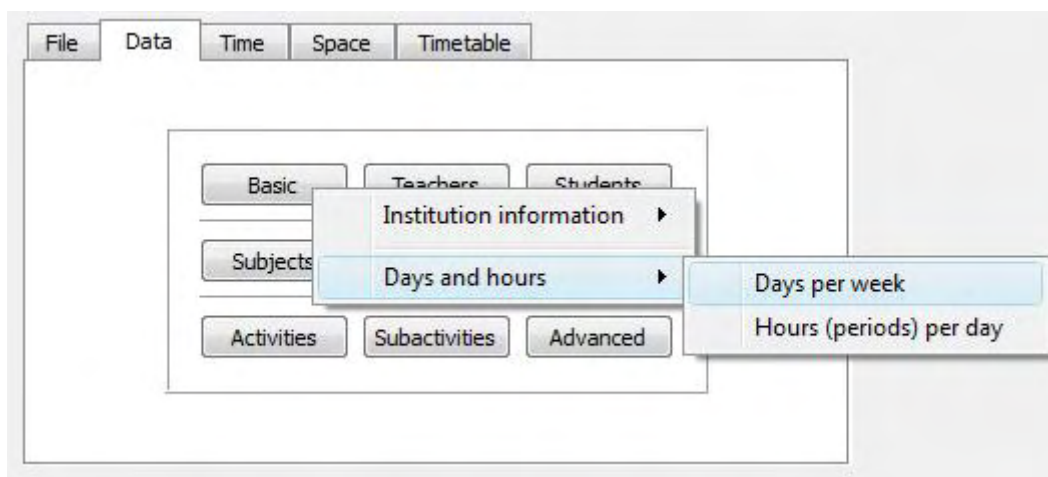
Εικόνα 4.3: Μενού →Data →Basic →Institution information



Εικόνα 4.4: Institution name-Όνομα πανεπιστημίου



Εικόνα 4.5: Comments-Σχόλια



Εικόνα 4.6: Μενού →Data →Basic →Days and hours

The days of the week

Number of working days per week

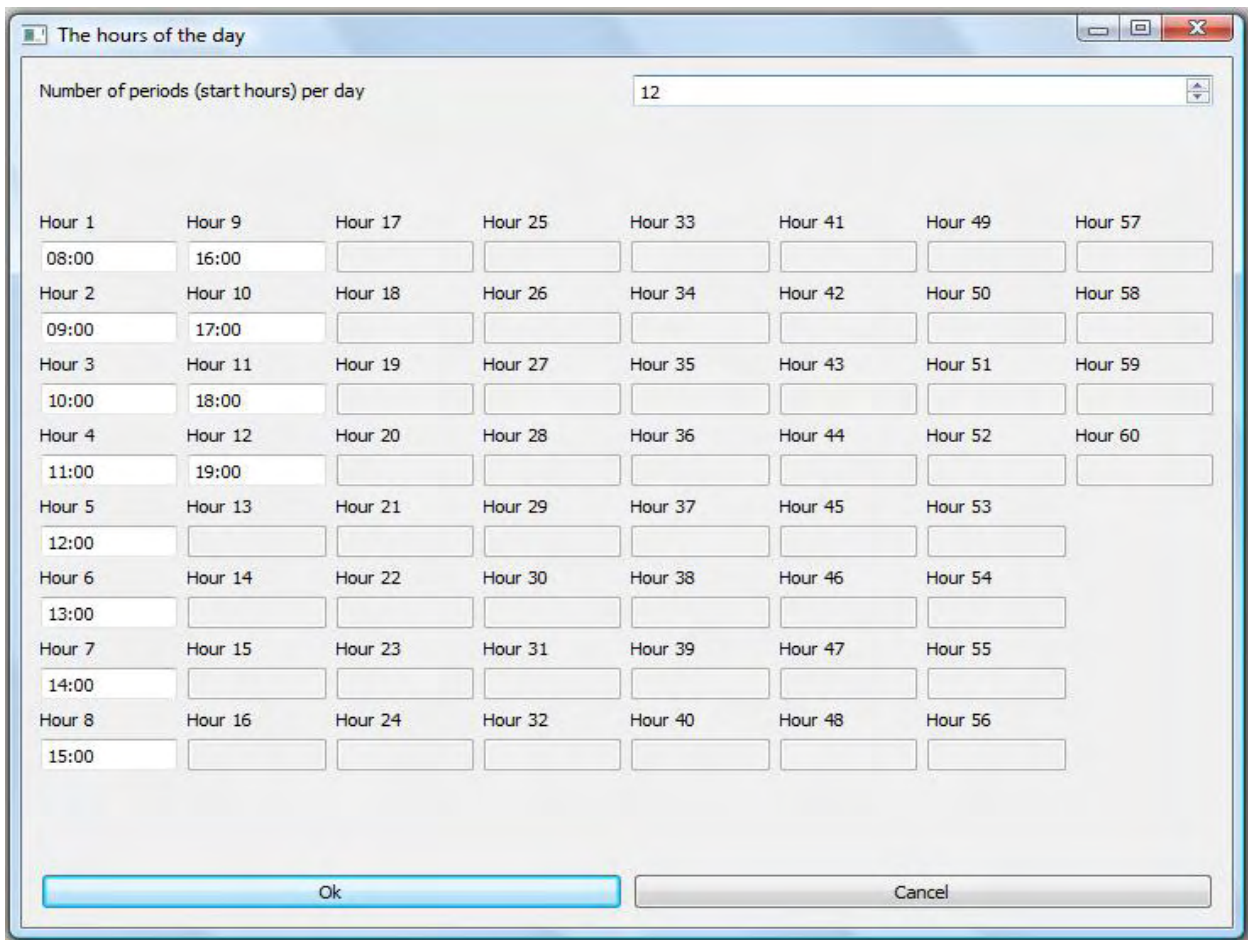
5

Day 1	Day 8	Day 15	Day 22	Day 29
Δευτέρα				
Day 2	Day 9	Day 16	Day 23	Day 30
Τρίτη				
Day 3	Day 10	Day 17	Day 24	Day 31
Τετάρτη				
Day 4	Day 11	Day 18	Day 25	Day 32
Πέμπτη				
Day 5	Day 12	Day 19	Day 26	Day 33
Παρασκευή				
Day 6	Day 13	Day 20	Day 27	Day 34
Day 7	Day 14	Day 21	Day 28	Day 35

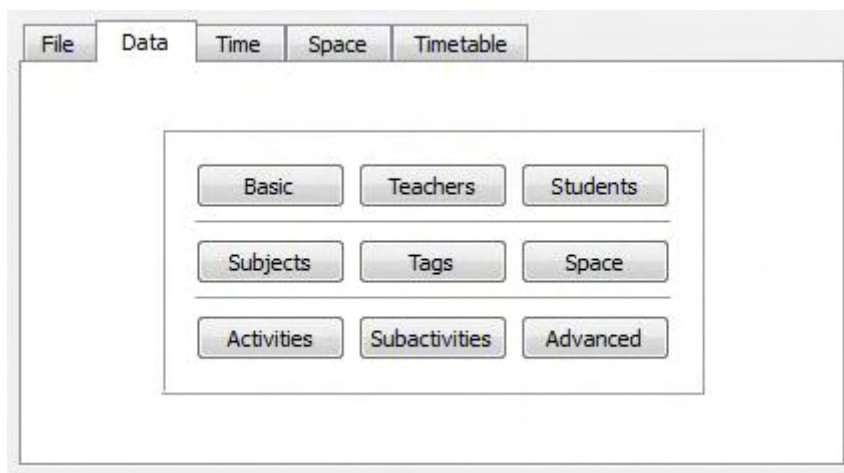
Ok Cancel

Εικόνα 4.7: Days-Ημέρες

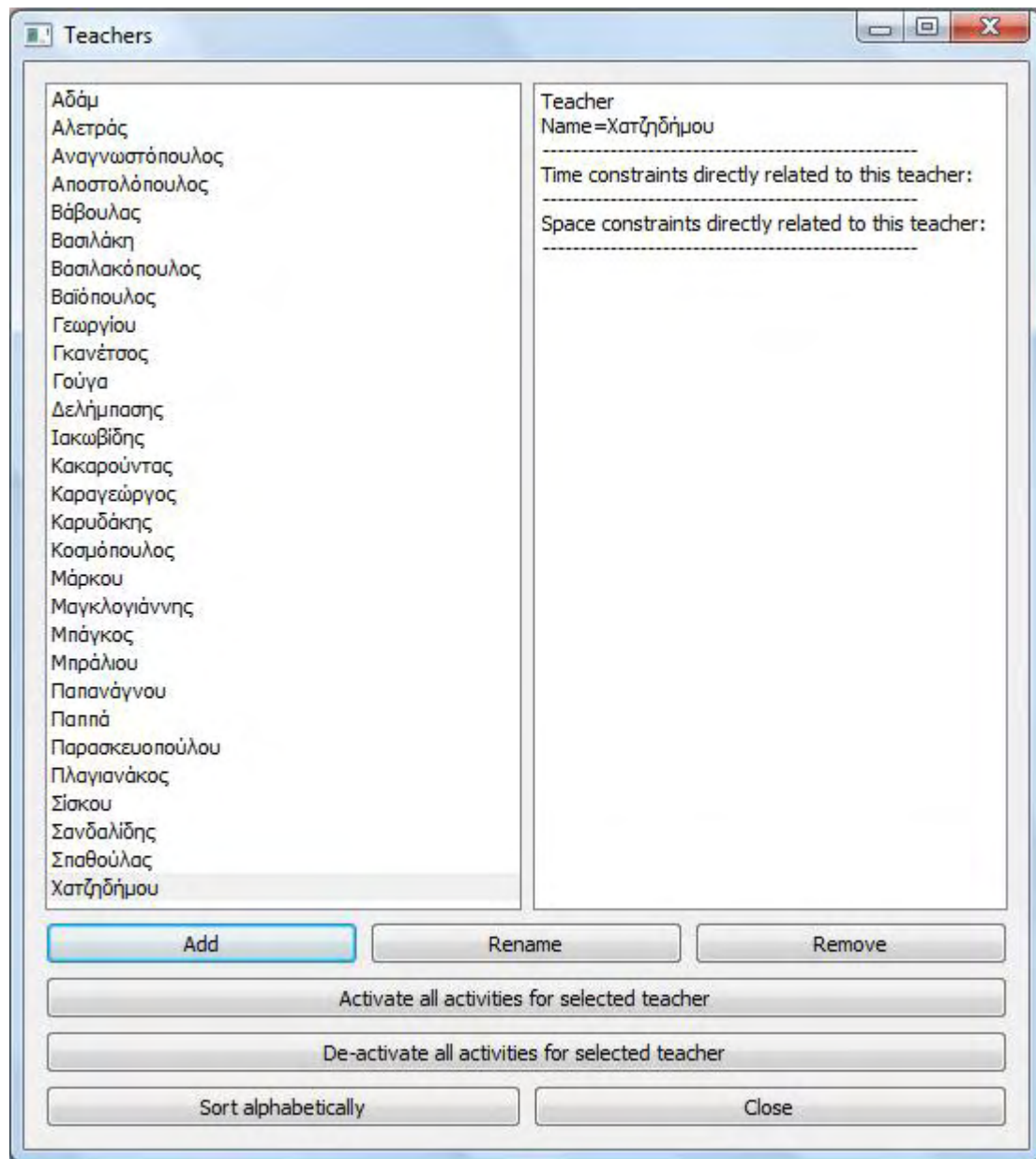




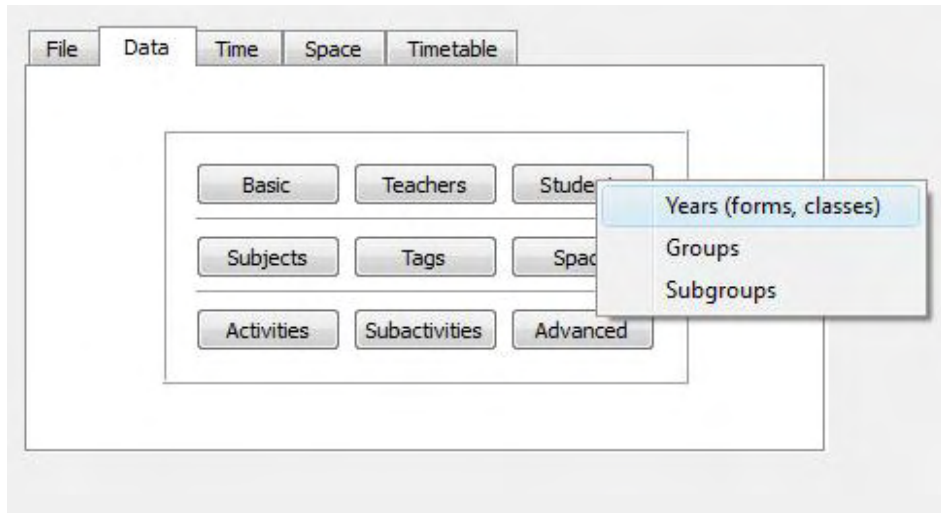
Εικόνα 4.8: Hours-Ωρες



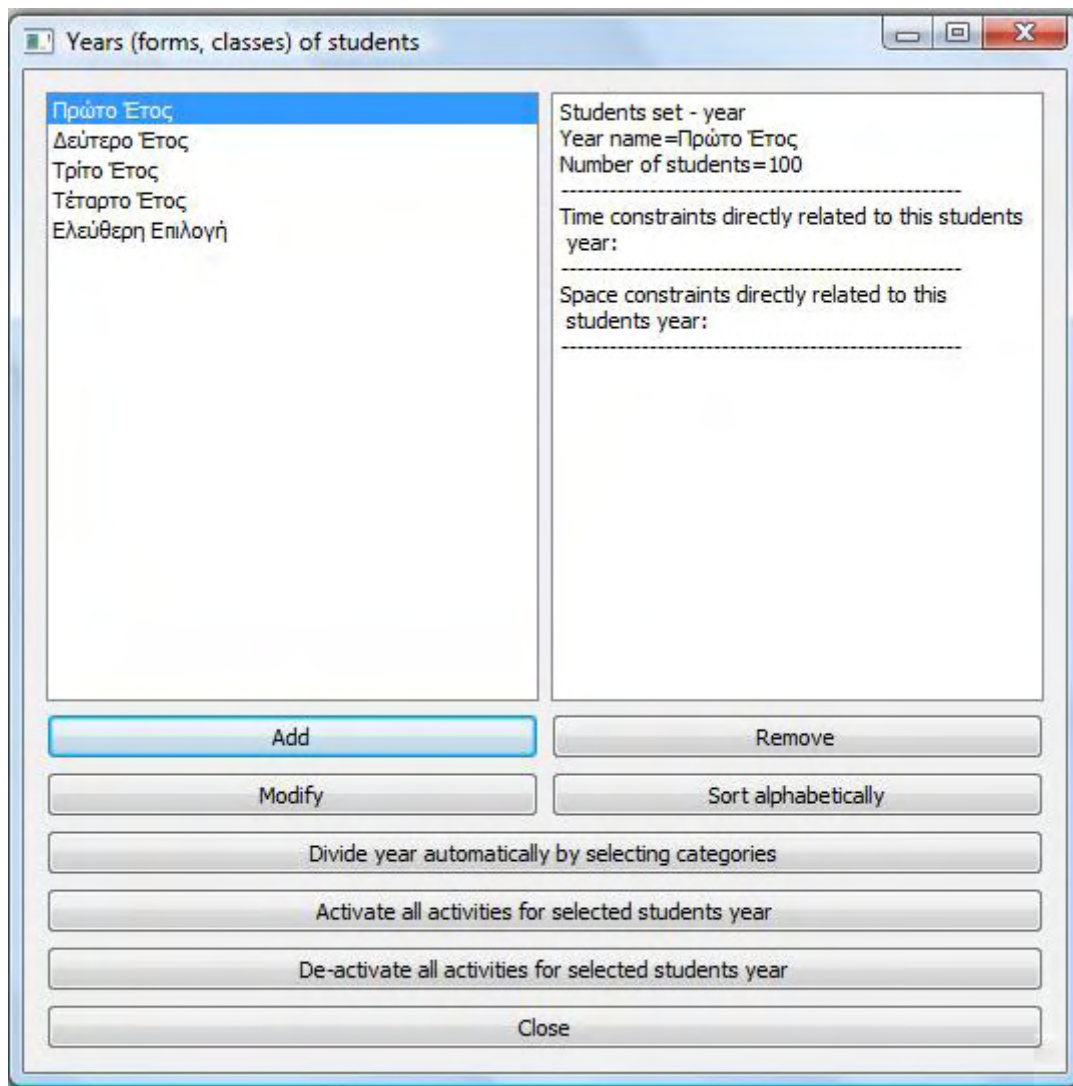
Εικόνα 4.9: Μενού Data



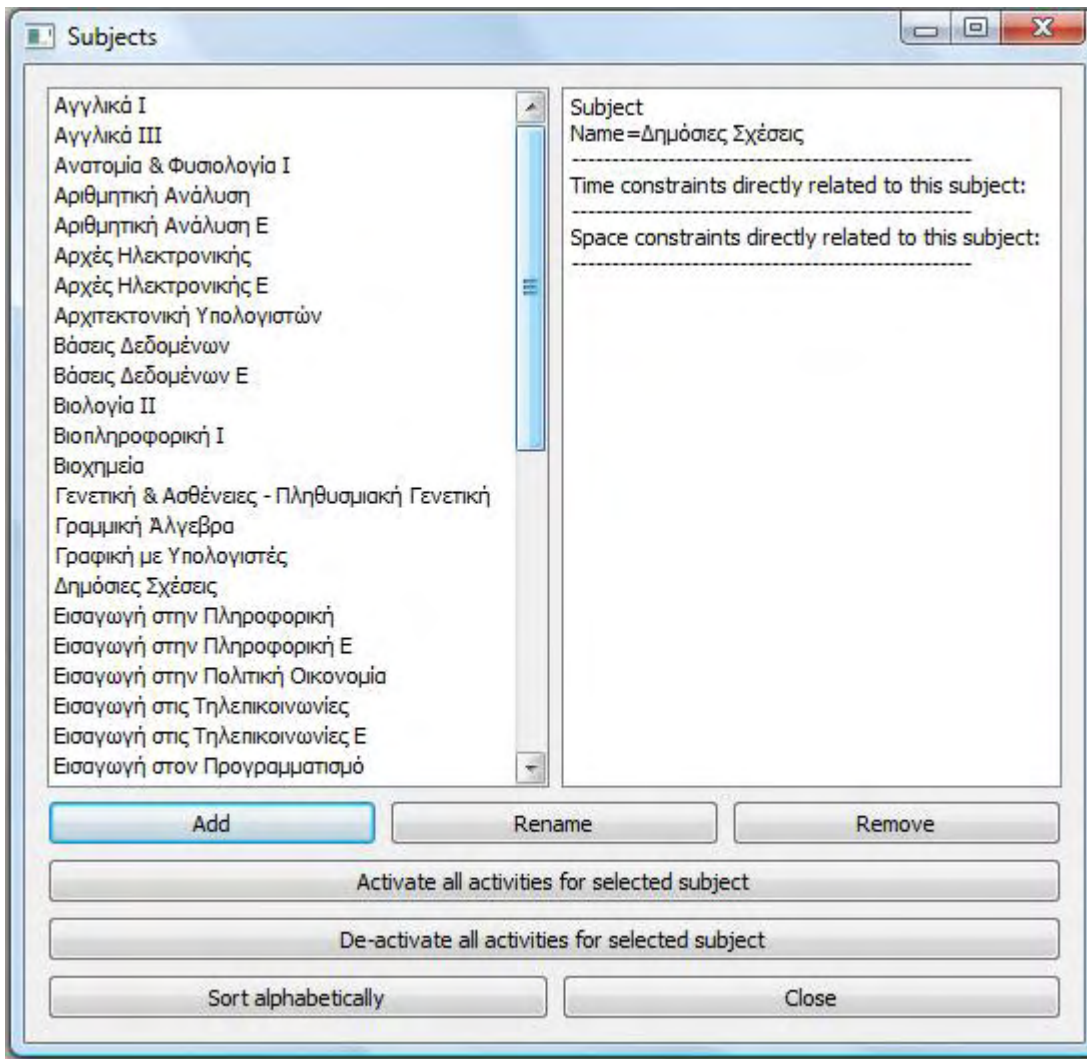
Εικόνα 4.10: Teachers-Διδάσκοντες



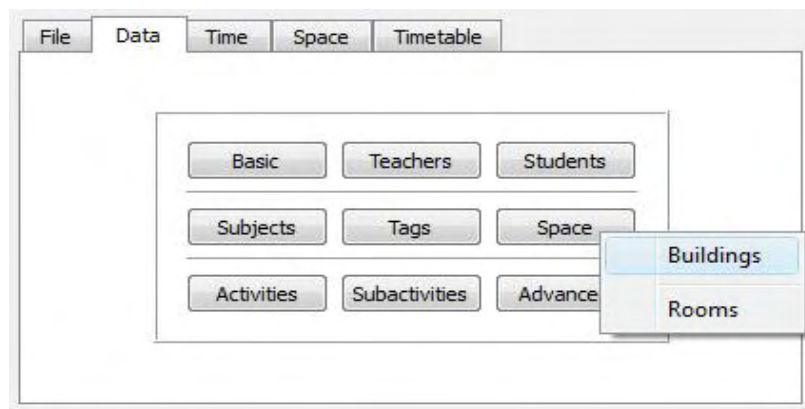
Εικόνα 4.11: Μενού →Data →Students



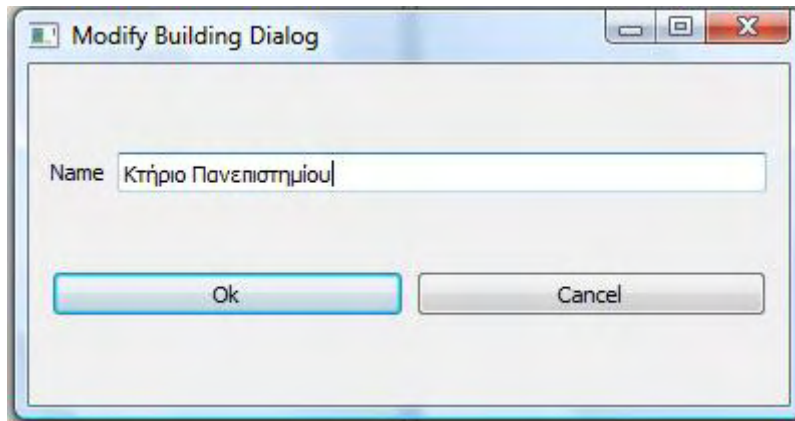
Εικόνα 4.12: Ομάδες φοιτητών



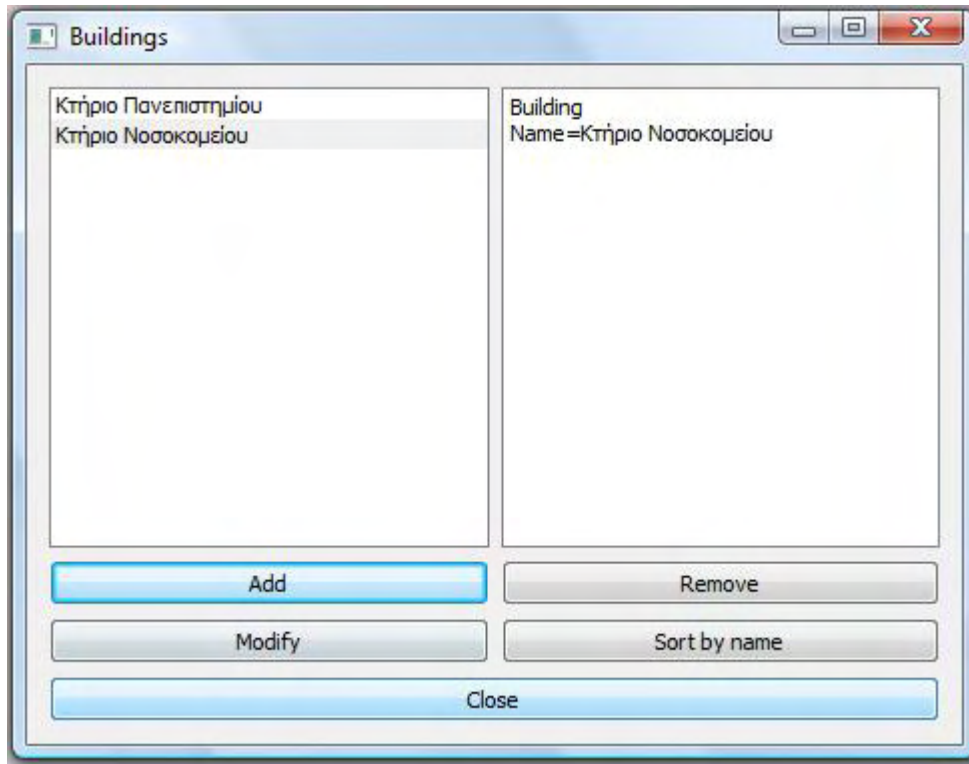
Εικόνα 4.13: Subjects-Μαθήματα Χειμερινού Εξαμήνου



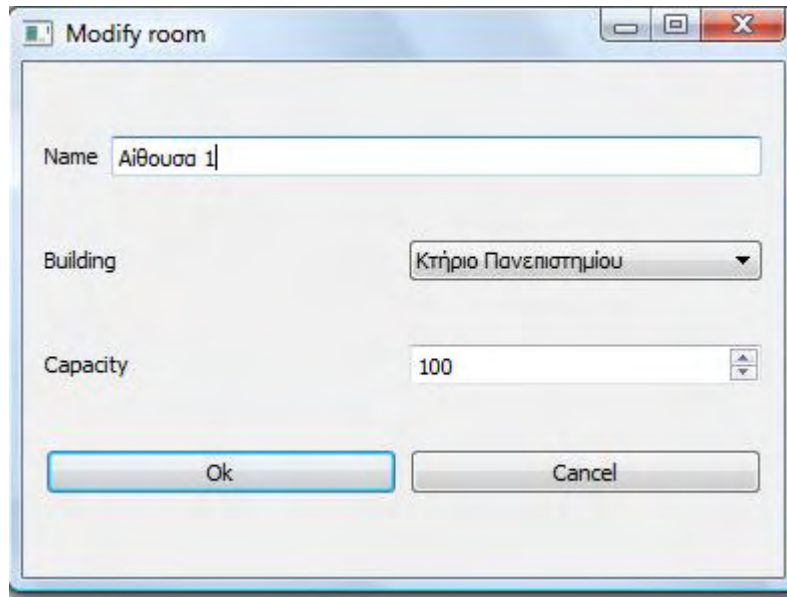
Εικόνα 4.14: Space-Χώροι φοίτησης



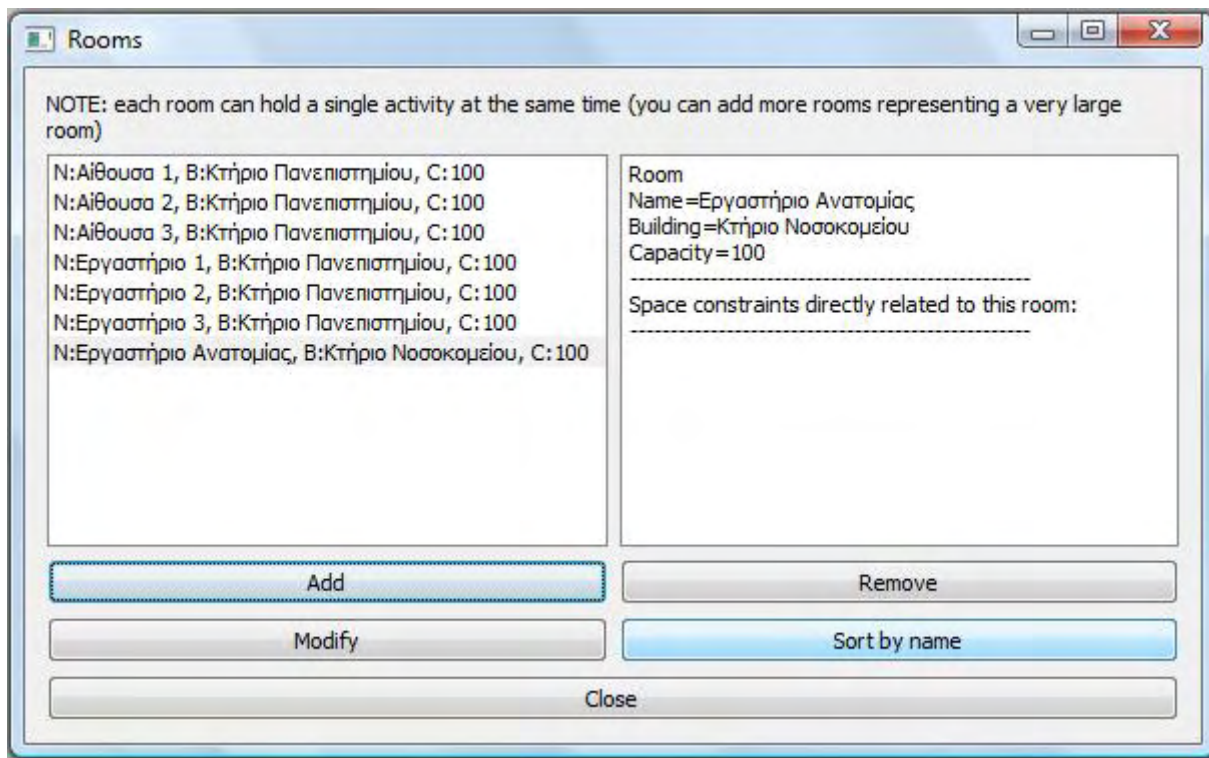
Εικόνα 4.15: Προσθήκη κτηρίου



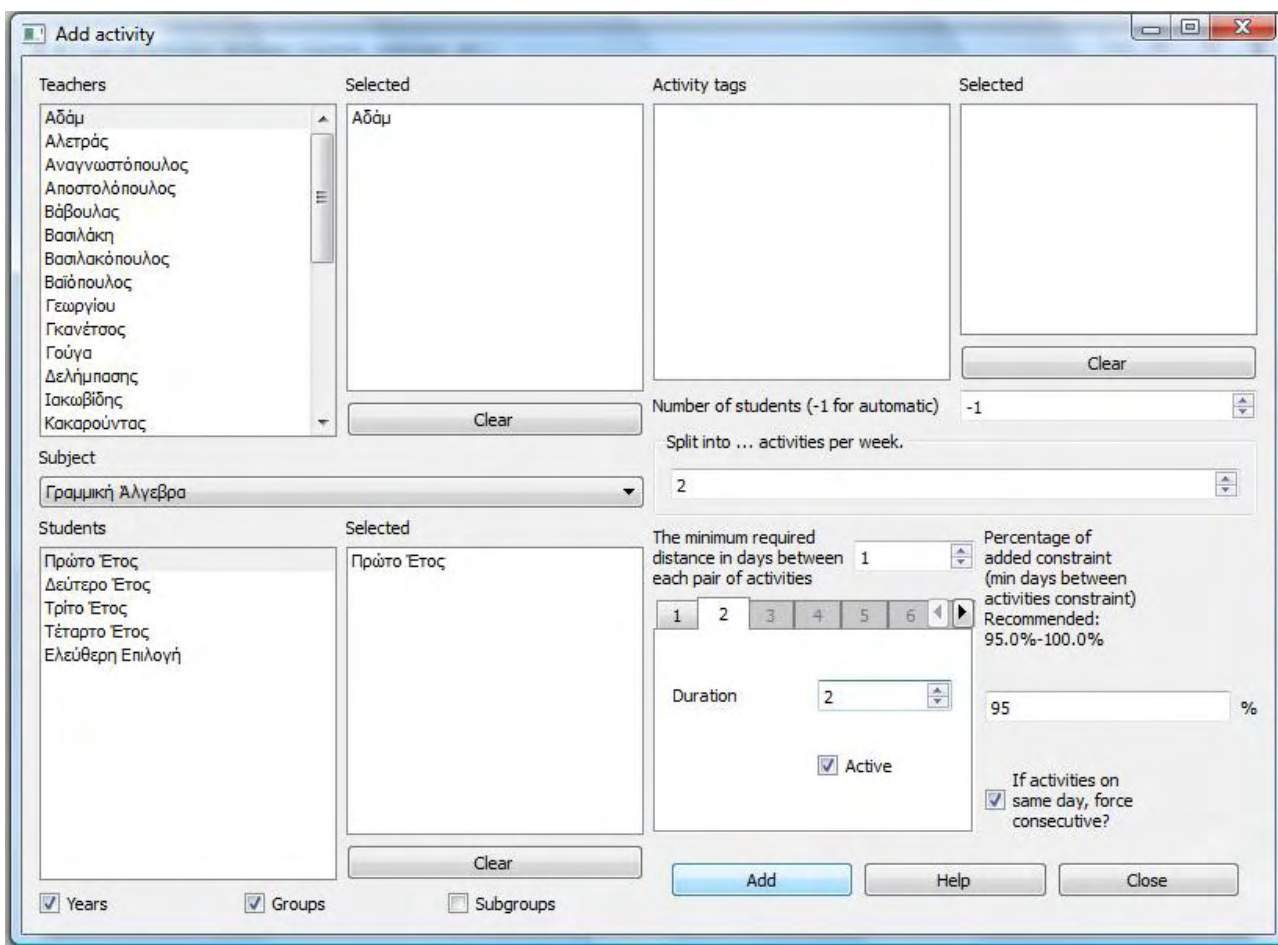
Εικόνα 4.16: Buildings-Κτήρια



Εικόνα 4.17: Προσθήκη αίθουσας



Εικόνα 4.18: Rooms-Αίθουσες και Εργαστήρια



Εικόνα 4.19: Γραμμική Άλγεβρα-Αδάμ-Πρώτο Έτος

Για τους περιορισμούς χρόνου και τόπου πρέπει να έχουμε στη διάθεση μας πληροφορίες που να αφορούν τις απαιτήσεις του τμήματος όπως πότε ο κάθε διδάσκοντας είναι διαθέσιμος να διδάσκει ή τις προτιμήσεις χώρων σχετικά με τις δραστηριότητες ή τα μαθήματα.

Για αρχή ξεκινάμε με την καταχώρηση των περιορισμών χρόνου όπου επιλέγουμε από το μενού Time->Break (εικόνα 4.20) για να καταχωρίσουμε τα διαλείμματα που γίνονται μέσα στο χρονικό διάστημα που ορίσαμε. Στην περίπτωση μας δείχνει πότε το τμήμα έχει ώρες που δε διδάσκονται μαθήματα και στη συγκεκριμένη περίπτωση είναι αφιερωμένες στη συνέλευση των φοιτητών (εικόνα 4.21). Μετά επιλέγουμε από το μενού το Time-->Teachers-->A teacher's not available times για να καταχωρήσουμε πότε ο κάθε διδάσκων δεν είναι διαθέσιμος (εικόνες 4.20 και 4.22). Ένα παράδειγμα πώς καταχωρούμε αυτούς τους περιορισμούς χρόνου είναι η εικόνα 4.23 όπου η κα Αδάμ είναι διαθέσιμη να διδάσκει Τετάρτη, Πέμπτη και Παρασκευή και όχι Δευτέρα και Τρίτη. Για την ακρίβεια μπορούμε να καταχωρήσουμε και ποιες ώρες δεν είναι διαθέσιμη αλλά στην παρούσα εργασία θα περιοριστούμε μόνο στις ημέρες της εβδομάδας. Έτσι, με αυτόν τον τρόπο καταχωρούμε τους υπόλοιπους περιορισμούς για τους διδάσκοντες όπως παρουσιάζεται στον πίνακα 4.2 και τελειώνουμε με τους περιορισμούς χρόνου.

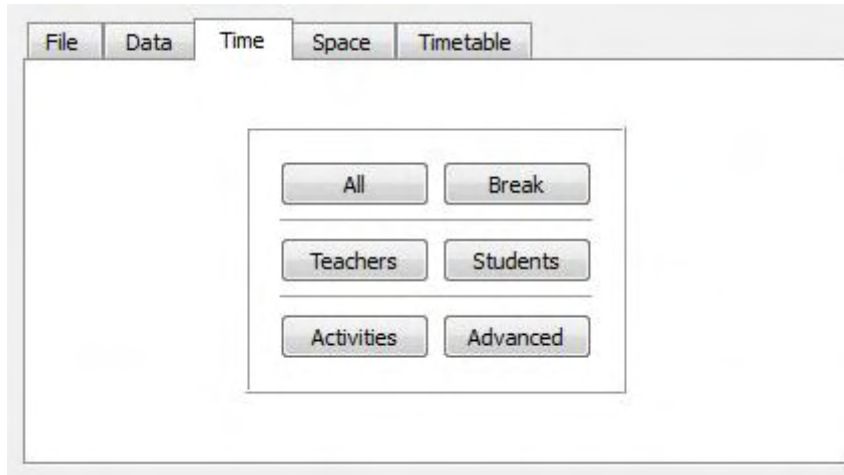
ΔΙΔΑΣΚΩΝ	ΜΗ ΔΙΑΘΕΣΙΜΕΣ ΗΜΕΡΕΣ
Αδάμ	Δευτέρα, Τρίτη
Αλετράς	Τετάρτη, Πέμπτη
Αναγνωστόπουλος	Δευτέρα, Τρίτη
Αποστολόπουλος	Δευτέρα, Παρασκευή
Βάβουλας	Πέμπτη
Βασιλάκη	Δευτέρα, Τρίτη, Τετάρτη, Πέμπτη
Βασιλακόπουλος	Δευτέρα, Τρίτη
Βαϊόπουλος	Τετάρτη, Πέμπτη
Γεωργίου	Τετάρτη, Πέμπτη, Παρασκευή
Γκανέτσος	Τρίτη, Τετάρτη, Πέμπτη
Γούγα	Τρίτη, Τετάρτη, Πέμπτη, Παρασκευή
Δελήμπασης	Δευτέρα, Παρασκευή
Ιακωβίδης	Δευτέρα, Παρασκευή
Κακαρούντας	Δευτέρα, Τρίτη, Τετάρτη
Καραγεώργος	Τρίτη, Τετάρτη, Πέμπτη, Παρασκευή
Καρυδάκης	Τετάρτη, Πέμπτη, Παρασκευή
Κοσμόπουλος	Δευτέρα, Τρίτη, Τετάρτη
Μάρκου	Πέμπτη, Παρασκευή
Μαγκλογιάννης	Δευτέρα, Παρασκευή
Μπάγκος	Δευτέρα, Παρασκευή
Μπράλιου	Πέμπτη
Παπανάγνου	Δευτέρα, Τρίτη, Παρασκευή
Παππά	Δευτέρα, Τετάρτη, Πέμπτη, Παρασκευή
Παρασκευοπούλου	Δευτέρα, Τρίτη, Πέμπτη, Παρασκευή
Πλαγιανάκος	Δευτέρα, Παρασκευή
Σίσκου	Τρίτη, Τετάρτη, Πέμπτη, Παρασκευή
Σανδαλίδης	Πέμπτη, Παρασκευή
Σπαθούλας	Τετάρτη, Παρασκευή
Χατζηδήμου	Τρίτη, Τετάρτη, Πέμπτη, Παρασκευή

Πίνακας 4.2: Διδάσκοντες-Μη διαθέσιμες ημέρες για το χειμερινό εξάμηνο

Στη συνέχεια ασχολούμαστε με τους περιορισμούς τόπου όπου χρειάζεται να μελετήσουμε τις απαιτήσεις και προτιμήσεις που υπάρχουν για τους χώρους φοίτησης. Έτσι, λοιπόν ο πρώτος περιορισμός τόπου που καταχωρούμε αφορά τη διαθεσιμότητα της αίθουσας 1 και επιλέγουμε από το μενού Space-->Rooms-->A room's not available times (εικόνες 4.24 και 4.25). Η αίθουσα 1 είναι διαθέσιμη όλες τις μέρες της εβδομάδας εκτός της Πέμπτης από τις 12:00 μέχρι τις 14:00 (εικόνα 4.26) όπου πραγματοποιείται η συνέλευση των φοιτητών. Κατόπιν επιλέγουμε από το μενού Space-->Teachers-->A teacher has a home room (εικόνες 4.24 και 4.27) για να καταχωρήσουμε αν κάποιος διδάσκοντας επιθυμεί να έχει μία αίθουσα προτίμησης για να παραδίδει τα μαθήματα του (εικόνα 4.28). Καταχωρούμε για τέσσερις διδάσκοντες την αίθουσα προτίμησης τους όπως παρουσιάζεται



στην εικόνα 4.29. Μετά επιλέγουμε από το μενού Space-->Students-->A set of students has a home room για να καταχωρήσουμε τις αίθουσες προτίμησης των ετών φοίτησης (εικόνες 4.24 και 4.30). Για την ακρίβεια μόνο τα τρία πρώτα έτη φοίτησης μπορούν να διαλέξουν μία αίθουσα και όχι το τέταρτο έτος επειδή δεν επαρκούν οι αίθουσες. Δηλαδή υπάρχουν τρεις αίθουσες αλλά τέσσερα έτη φοίτησης. Στις εικόνες 4.31, 4.32 και 4.33 παρουσιάζεται η καταχώρηση των αιθουσών για τα τρία πρώτα έτη.



Εικόνα 4.20: Μενού Time

Modify break times

The important thing is that a break does not induce gaps for teachers or students

Weight percentage (necessary: 100%) 100

	Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή
08:00					
09:00					
10:00					
11:00					
12:00				X	
13:00				X	
14:00					
15:00					
16:00					
17:00					
18:00					
19:00					

Set all allowed X=not allowed, empty=allowed

Set all break Ok Cancel

Εικόνα 4.21: Breaks-Διαλείμματα

Επίσης υπάρχει η επιλογή στο μενού (εικόνες 4.24 και 4.34) Space-->Subjects-->A subject has a preferred room όπου μπορούμε να καταχωρήσουμε μαθήματα που διαλέγουν συγκεκριμένους χώρους φοίτησης. Στην εικόνα 4.35 απεικονίζεται πως καταχωρούμε ένα μάθημα με την αντίστοιχη αίθουσα προτίμησης του και στην εικόνα 4.36 η υπόλοιπη καταχώρηση κάποιων μαθημάτων με τους χώρους προτίμησης τους (πίνακας 4.3).

ΜΑΘΗΜΑ	ΧΩΡΟΣ ΠΡΟΤΙΜΗΣΗΣ
Αριθμητική Ανάλυση (Εργαστήριο)	Εργαστήριο 1
Εισαγωγή στον Προγραμματισμό (Εργαστήριο)	Εργαστήριο 1
Εισαγωγή στην Πληροφορική (Εργαστήριο)	Εργαστήριο 1
Αρχές Ηλεκτρονικής (Εργαστήριο)	Εργαστήριο 2
Ενσωματωμένα Συστήματα Υπολογιστών	Εργαστήριο 2



Εισαγωγή στις Τηλεπικοινωνίες (Εργαστήριο)	Εργαστήριο 1
Ψηφιακή Επεξεργασία Βιοσημάτων (Εργαστήριο)	Εργαστήριο 1
Βάσεις Δεδομένων (Εργαστήριο)	Εργαστήριο 1
Τεχνολογίες Εφαρμογών Διαδικτύου	Αίθουσα 2
Επικοινωνία Ανθρώπου Υπολογιστή	Αίθουσα 2
Εισαγωγή στην Πολιτική Οικονομία	Αίθουσα 1
Δημόσιες Σχέσεις	Αίθουσα 1
Παιδαγωγικά	Αίθουσα 1
Κοινωνιολογία	Αίθουσα 3
Συστήματα Ιατρικής Απεικόνισης	Αίθουσα 3

Πίνακας 4.3: Μαθήματα και Χώροι Προτίμησης για το χειμερινό εξάμηνο

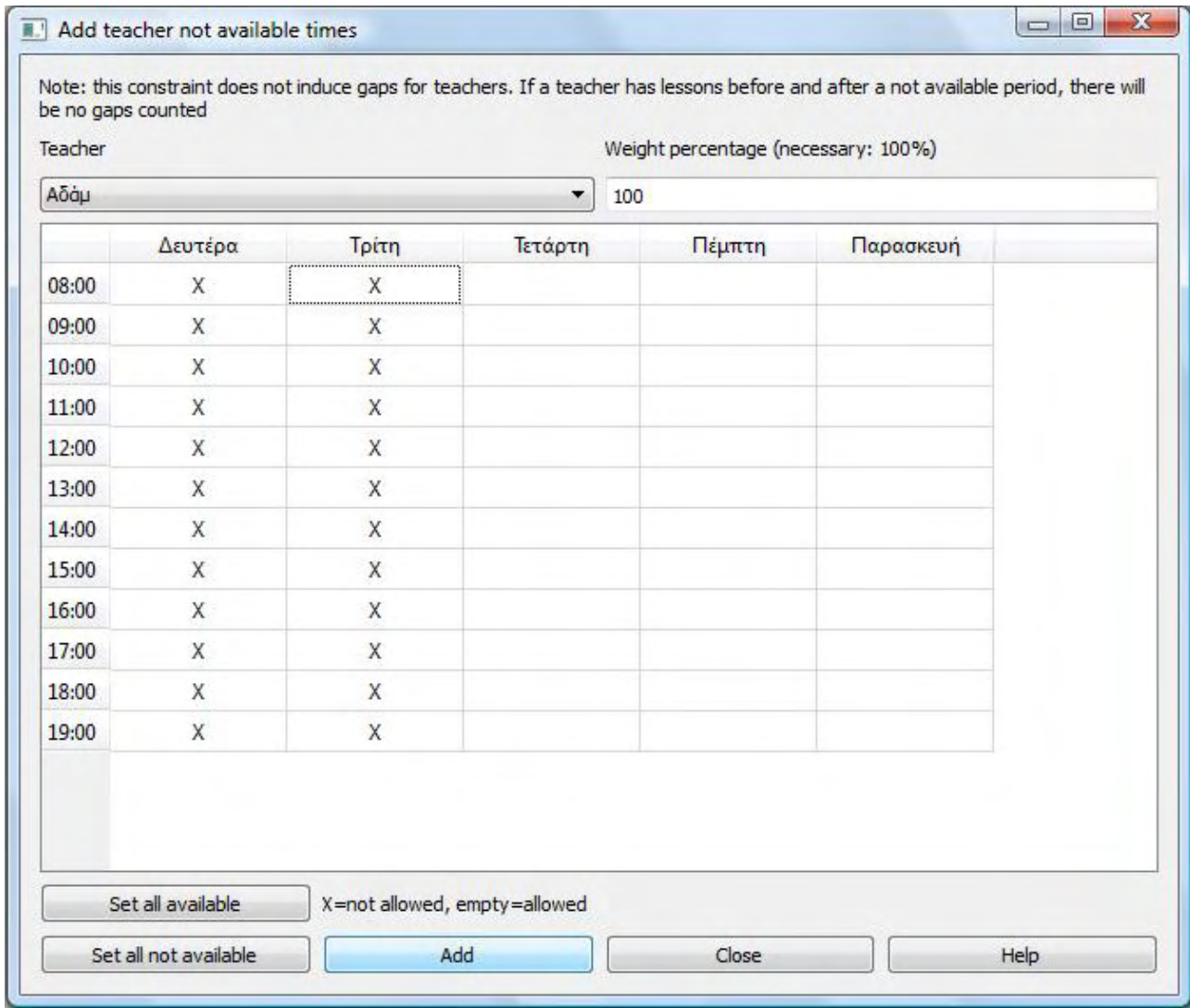
Ακόμη στο μενού η επιλογή Space-->Activities-->An activity has a preferred room (εικόνα 4.37) απευθύνεται στην προτίμηση που έχει κάποια δραστηριότητα για ένα χώρο. Συγκεκριμένα για τη μία ώρα Ανατομίας & Φυσιολογίας I του δευτέρου έτους επιλέγουμε να διδάσκεται στο εργαστήριο Ανατομίας (εικόνα 4.38) και για κάποιες δραστηριότητες του τέταρτου έτους που έμειναν ακάλυπτες από χώρο επιλέγουμε αίθουσες και εργαστήρια όπως παρουσιάζεται παρακάτω στον πίνακα 4.4.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ	ΧΩΡΟΣ ΠΡΟΤΙΜΗΣΗΣ
2/4-Σίσκου-Οργάνωση & Διοίκηση Συστημάτων Υγείας	Αίθουσα 1
2/4-Σίσκου-Οργάνωση & Διοίκηση Συστημάτων Υγείας	Αίθουσα 2
2/4-Μαγκλογιάννης-Πληροφοριακά Συστήματα Υγείας II	Εργαστήριο 1
2/4-Μαγκλογιάννης-Πληροφοριακά Συστήματα Υγείας II	Αίθουσα 2
2/4-Ιακωβίδης-Συστήματα Στήριξης Ιατρικών Αποφάσεων	Αίθουσα 1
2/4-Ιακωβίδης-Συστήματα Στήριξης Ιατρικών Αποφάσεων	Αίθουσα 3
2/4-Πλαγιανάκος-Τεχνητή Νοημοσύνη	Αίθουσα 1
2/4-Πλαγιανάκος-Τεχνητή Νοημοσύνη	Εργαστήριο 1
2/4-Βασιλακόπουλος-Θεωρία Γράφων	Αίθουσα 1
2/4-Βασιλακόπουλος-Θεωρία Γράφων	Αίθουσα 3

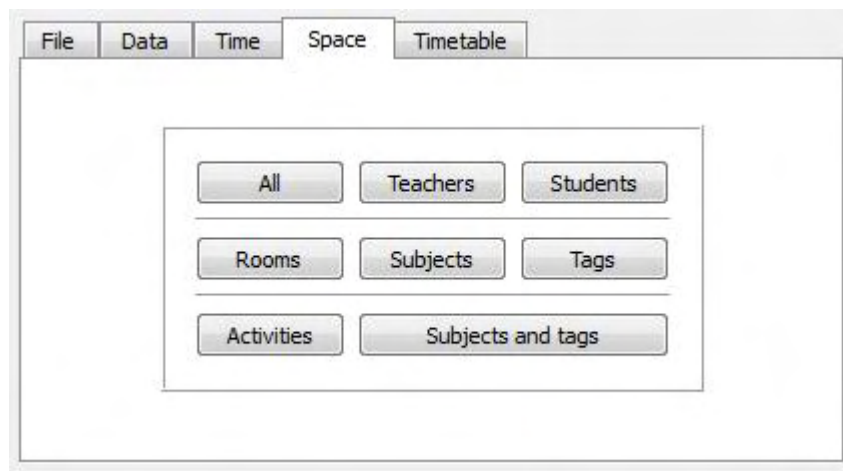
Πίνακας 4.4: Δραστηριότητες και Χώροι Προτίμησης για το χειμερινό εξάμηνο

A teacher's not available times	
	Max days per week for a teacher
	Min days per week for a teacher
	Max gaps per day for a teacher
	Max gaps per week for a teacher
	Max hours daily for a teacher
	Max hours daily with an activity tag for a teacher
	Min hours daily for a teacher
	Max hours continuously for a teacher
	Max hours continuously with an activity tag for a teacher
	A teacher works in an hourly interval max days per week
<hr/>	
	Max days per week for all teachers
	Min days per week for all teachers
	Max gaps per day for all teachers
	Max gaps per week for all teachers
	Max hours daily for all teachers
	Max hours daily with an activity tag for all teachers
	Min hours daily for all teachers
	Max hours continuously for all teachers
	Max hours continuously with an activity tag for all teachers
	All teachers work in an hourly interval max days per week

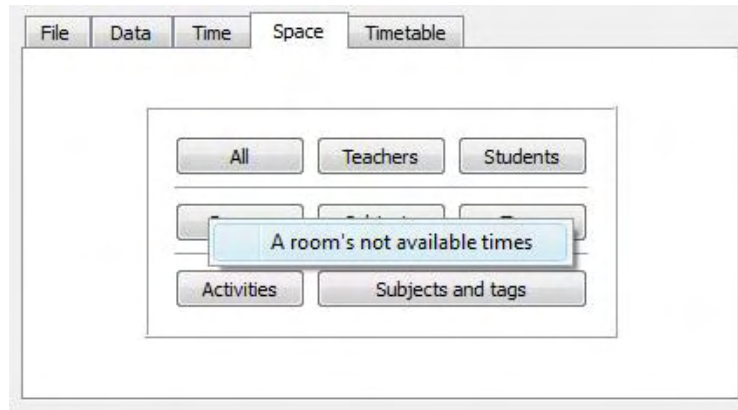
Εικόνα 4.22: Μενού  $\rightarrow$ Time  $\rightarrow$ Teachers



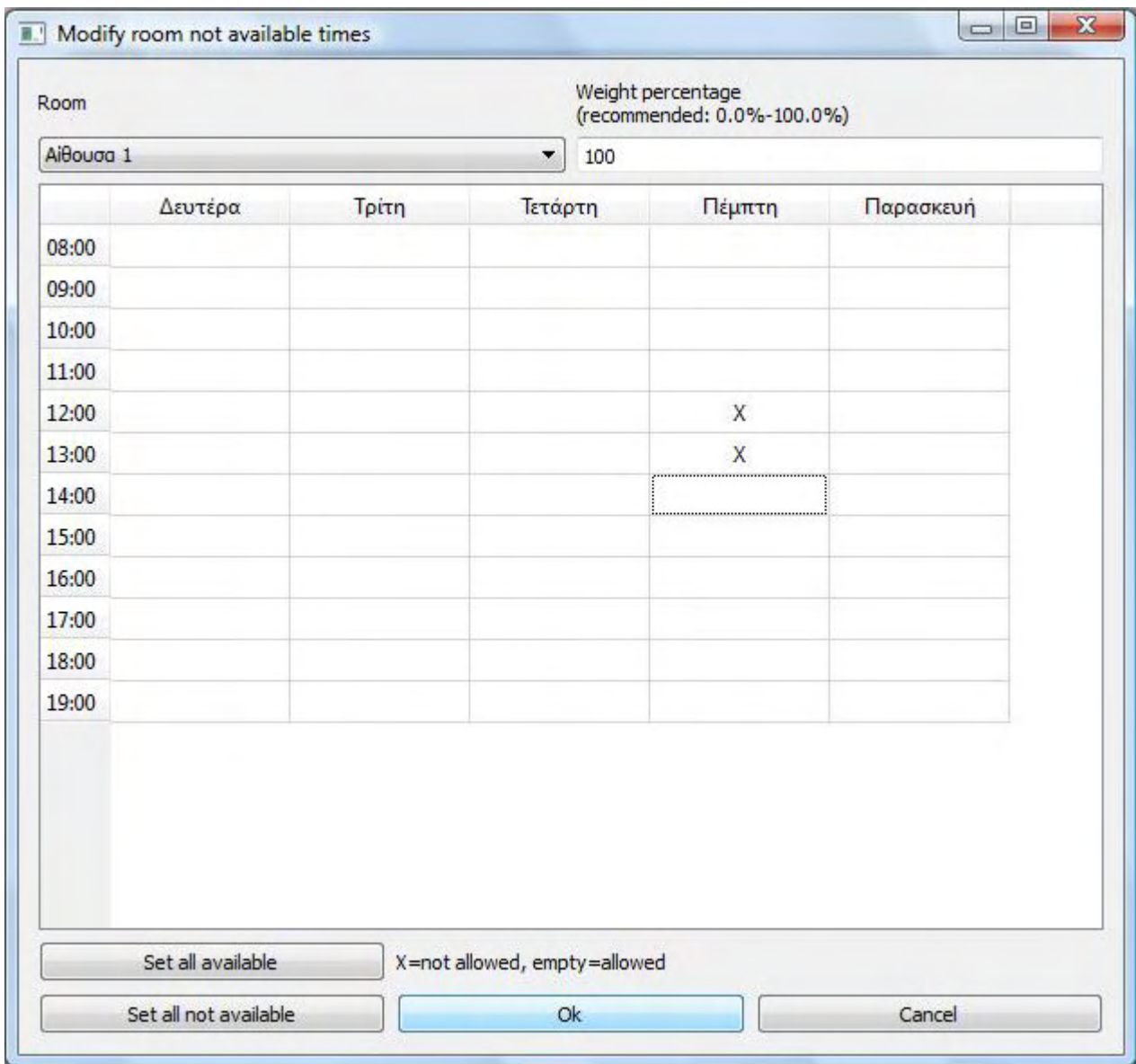
Εικόνα 4.23: Πότε η κα Αδάμ δεν είναι διαθέσιμη



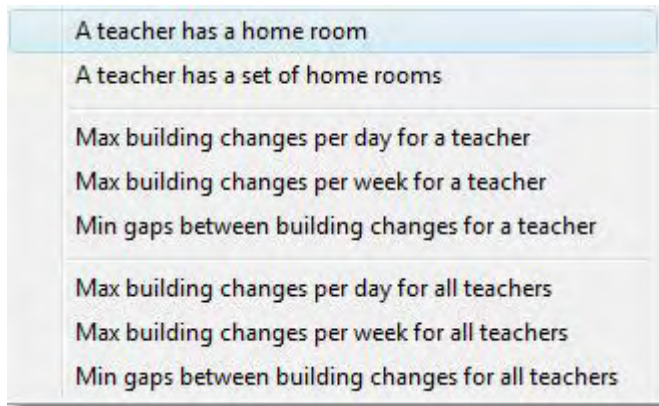
Εικόνα 4.24: Μενού Space



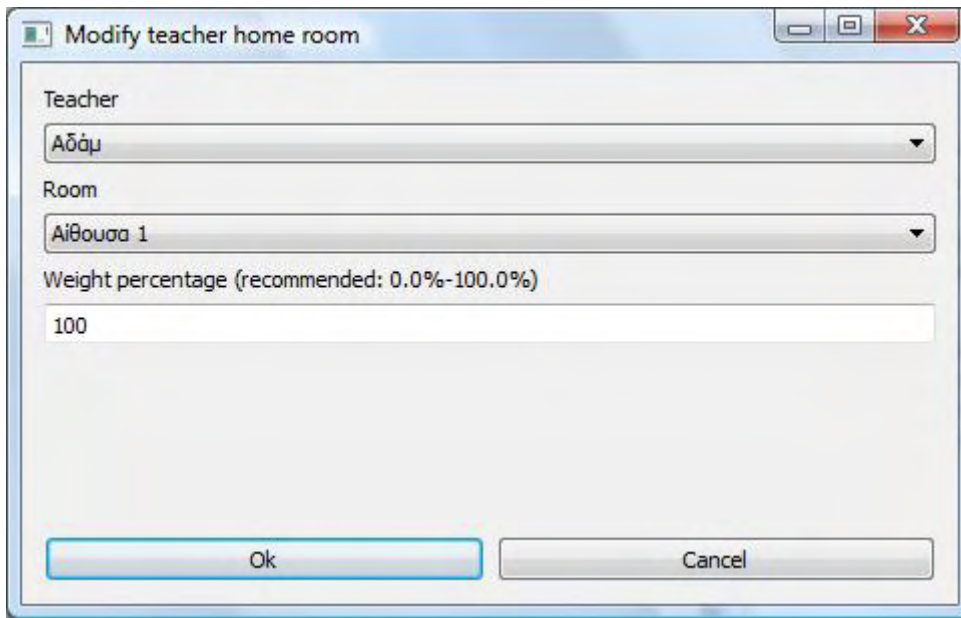
Εικόνα 4.25: Space → Rooms → A room's not available times



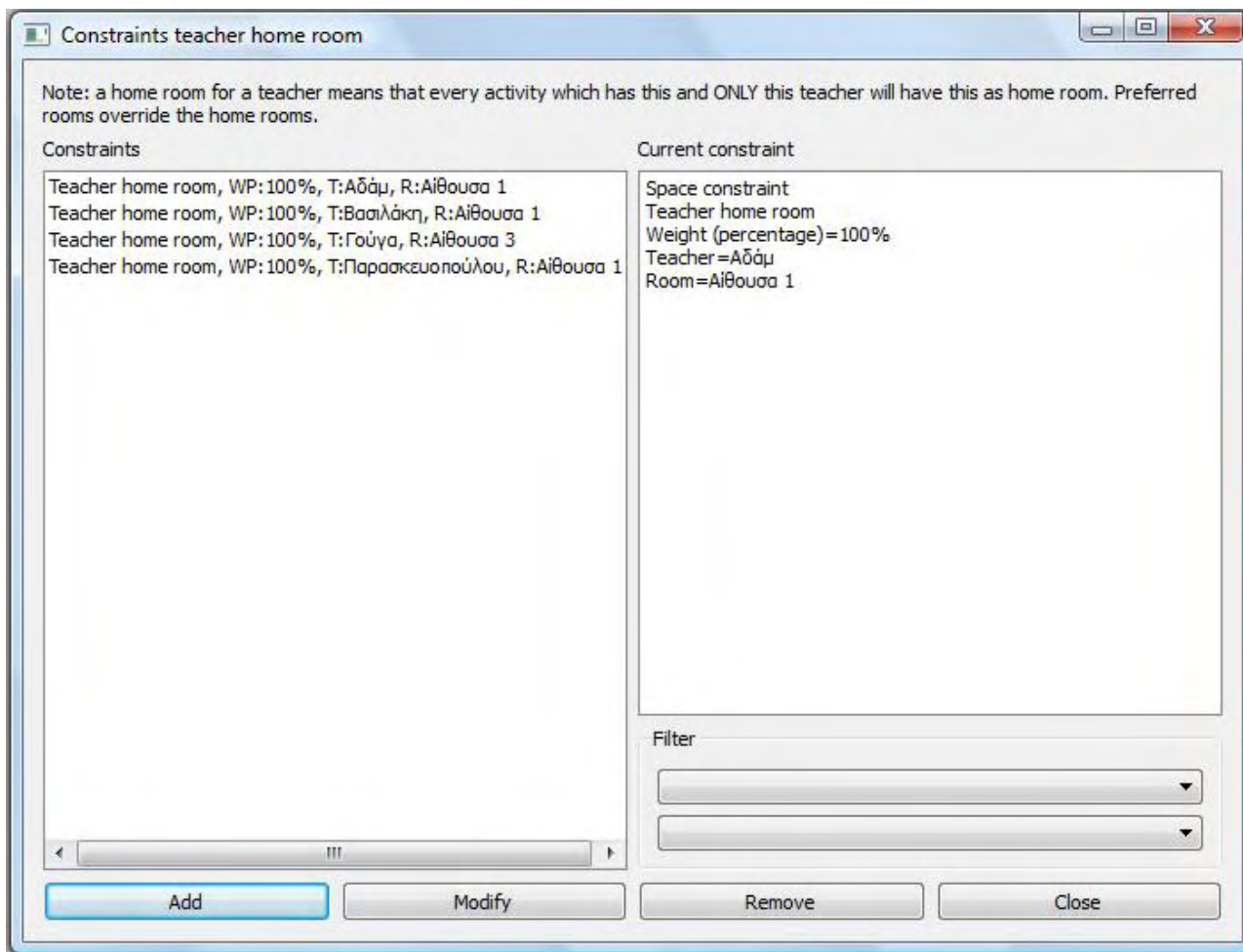
Εικόνα 4.26: Διαθεσιμότητα Αίθουσας 1



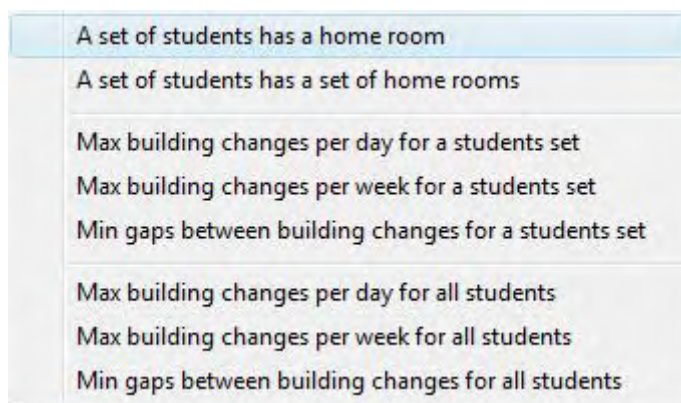
*Εικόνα 4.27: Space → Teachers → A teacher has a home room*



*Εικόνα 4.28: Αδάμ-Αίθουσα 1*

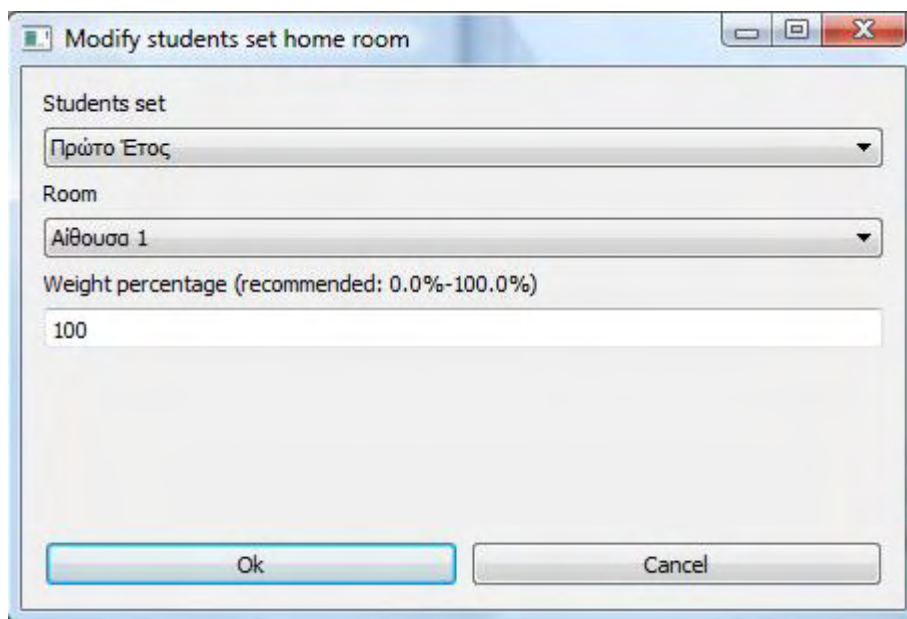


Εικόνα 4.29: Διδάσκοντες και αίθουσες προτίμησης

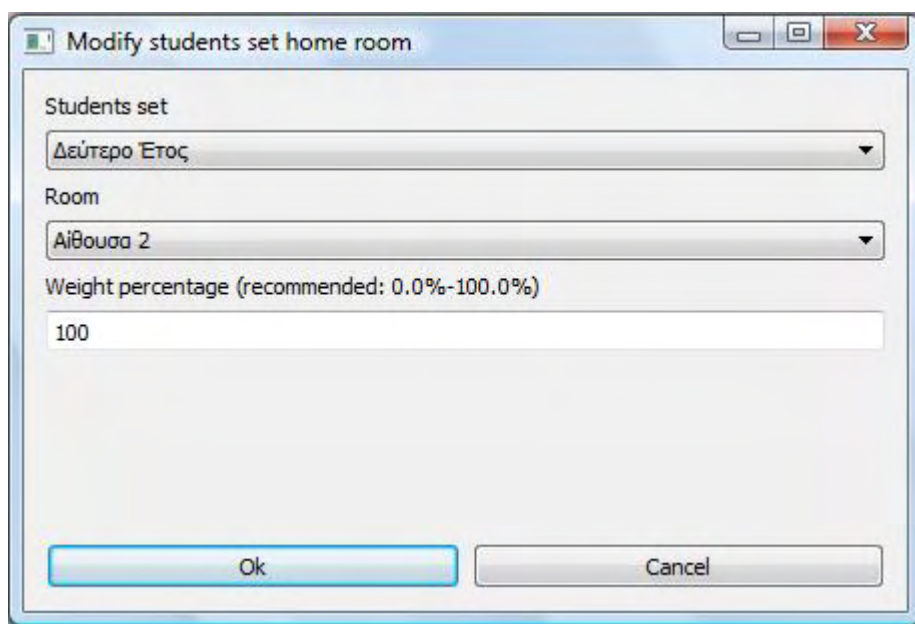


Εικόνα 4.30: Space → Students → A set of students has a home room

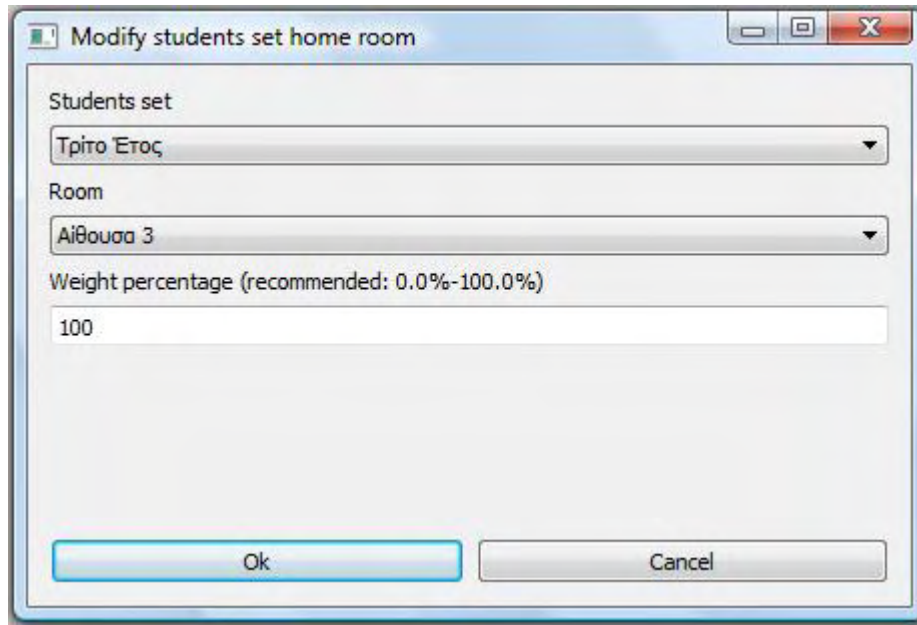




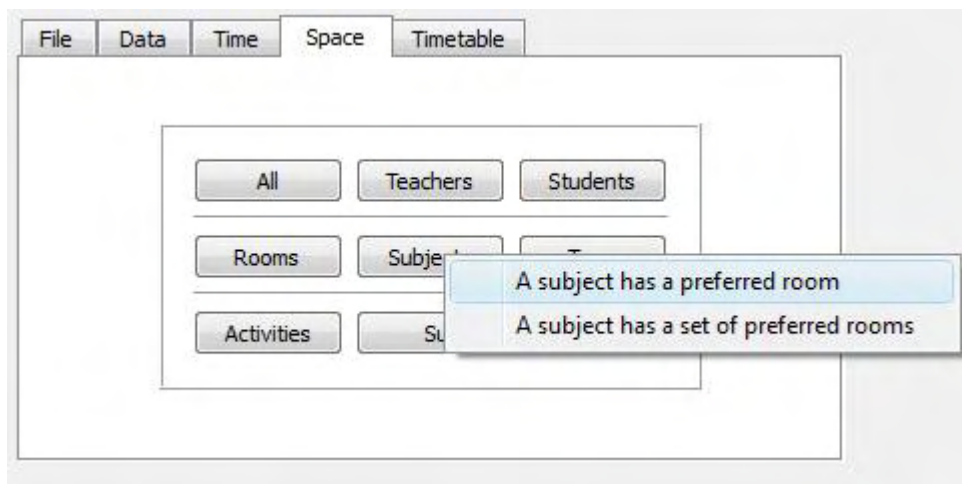
*Εικόνα 4.31: Πρώτο Έτος Αίθουσα 1*



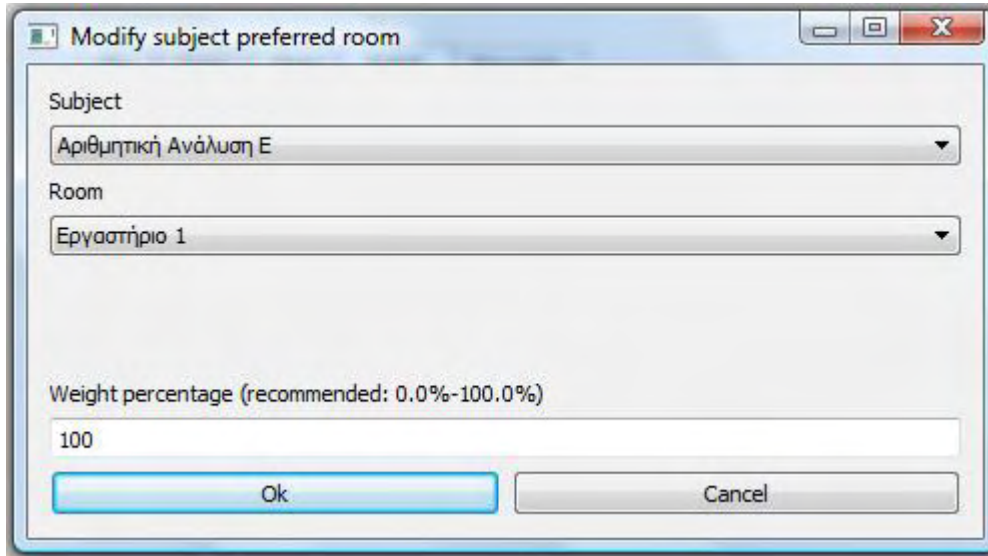
*Εικόνα 4.32: Δεύτερο Έτος\_Αίθουσα 2*



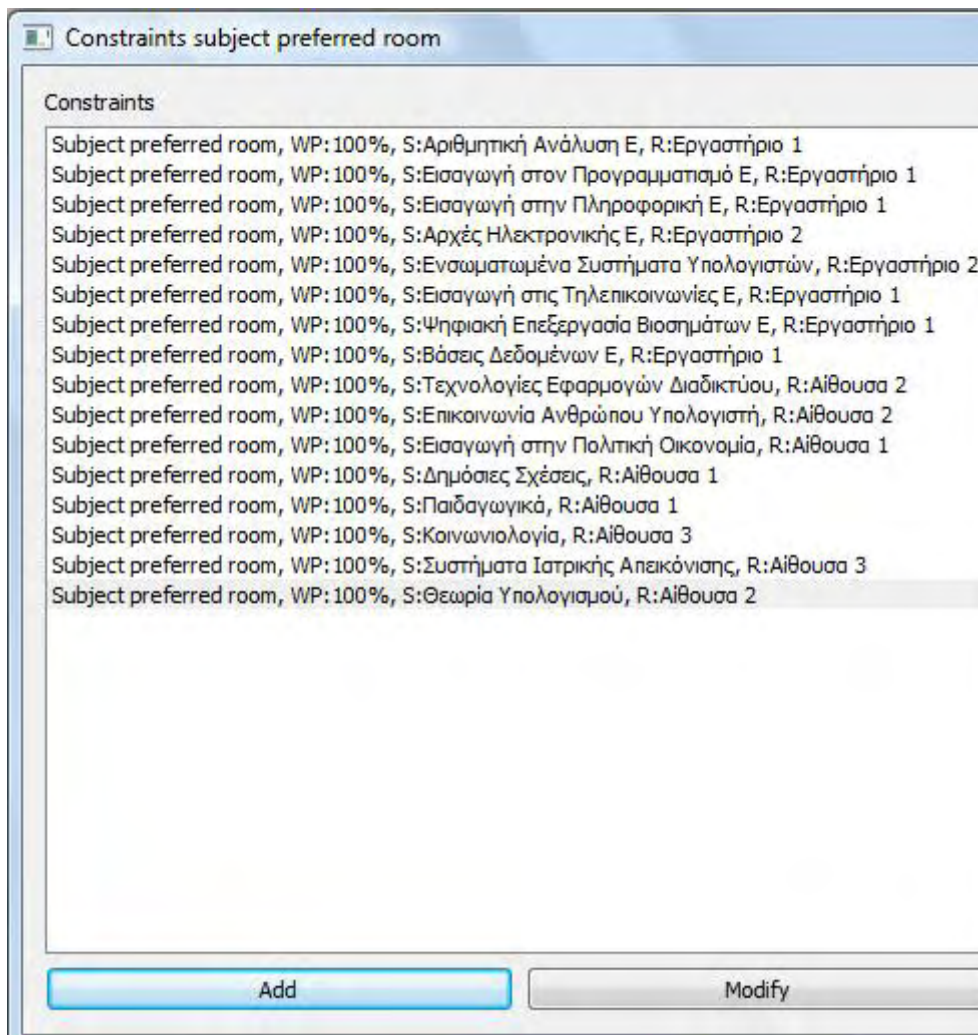
Εικόνα 4.33: Τρίτο Έτος Αίθουσα 3



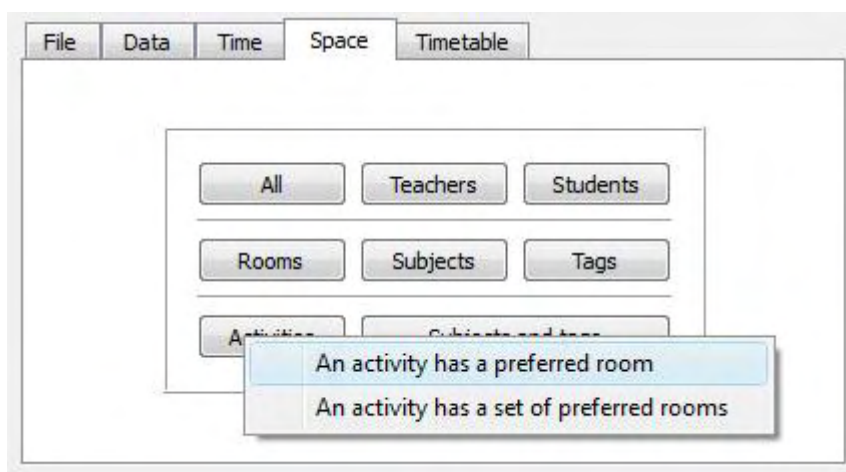
Εικόνα 4.34: Space → Subjects → A subject has a preferred room



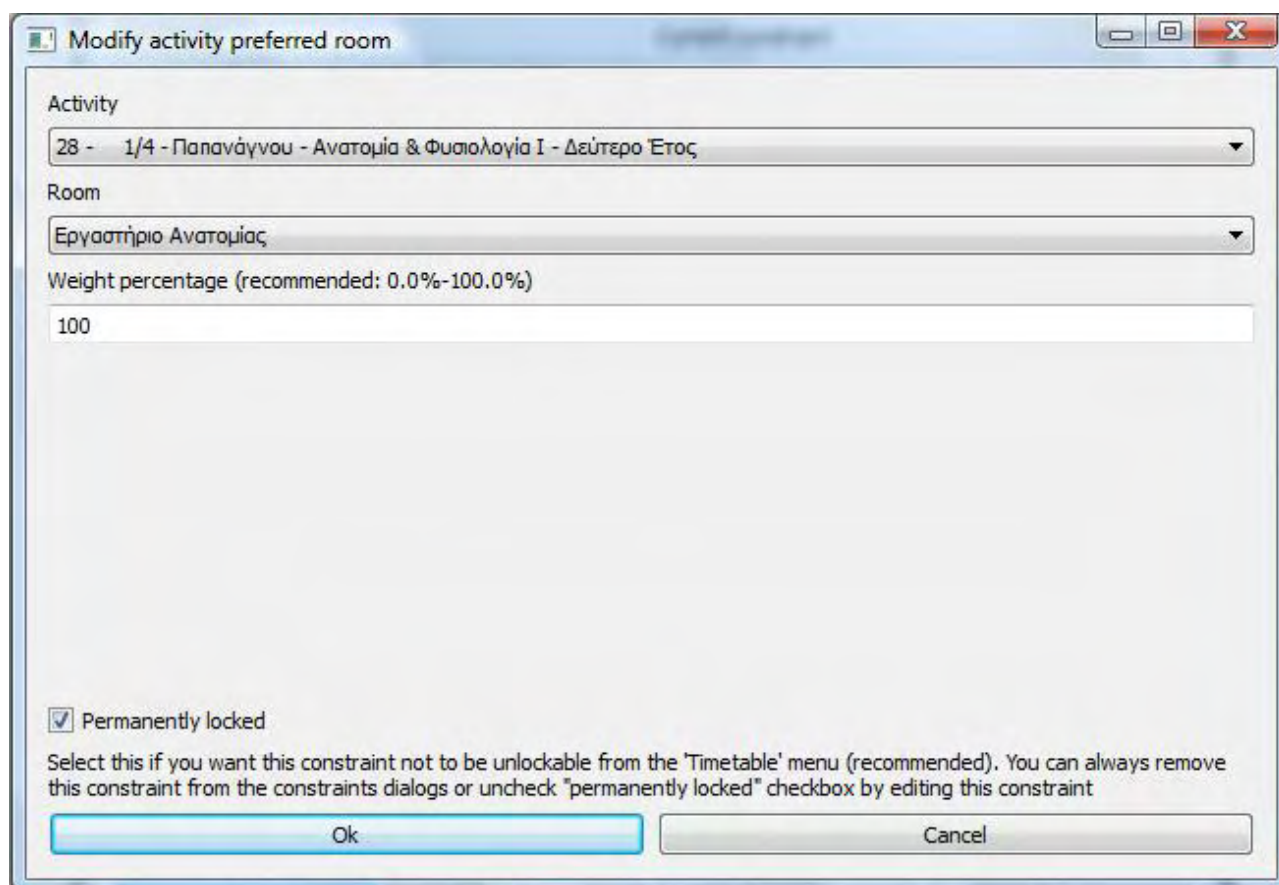
Εικόνα 4.35: Αριθμητική Ανάλυση Ε Εργαστήριο 1



Εικόνα 4.36: Μαθήματα και προτεινόμενοι Χώροι Φοίτησης



Εικόνα 4.37: Space →Activities →An activity has a preferred room



Εικόνα 4.38: (1 ώρα-Παπανάγνου-Ανατομία & Φυσιολογία I-Δεύτερο Έτος) Εργαστήριο Ανατομίας

Τέλος αφού έχουμε καταχωρήσει τους περιορισμούς χρόνου και τύπου μπορούμε να δημιουργήσουμε το ωρολόγιο πρόγραμμα. Παρακάτω παρουσιάζονται τα αποτελέσματα του χειμερινού ωρολόγιου προγράμματος που μας έδωσε η εφαρμογή FET για τα τέσσερα έτη καθώς και για τα μαθήματα ελεύθερης επιλογής (πίνακες 4.5, 4.6, 4.7, 4.8, 4.9).

<b>Πρώτο Έτος Automatic Group</b>					
	<b>Δευτέρα</b>	<b>Τρίτη</b>	<b>Τετάρτη</b>	<b>Πέμπτη</b>	<b>Παρασκευή</b>
<b>08:00</b>		---	---	Εισαγωγή στην	---
<b>09:00</b>	Φυσική Γκανέτσος Αίθουσα 1	Εισαγωγή στην	Μαθηματική Ανάλυση I Αδάμ Αίθουσα 1	Πληροφορική Ε	---
<b>10:00</b>		Σπαθούλας, Αποστολόπουλος Εργαστήριο 1		Αποστολόπουλος Εργαστήριο 1	
<b>11:00</b>	---	---		Μαθηματική Ανάλυση I Αδάμ Αίθουσα 1	Εισαγωγή στην Πληροφορική
<b>12:00</b>	---	---	---	-X-	Αναγνωστόπο υλος Αίθουσα 1
<b>13:00</b>	---	Εισαγωγή στον	---	-X-	
<b>14:00</b>	---	Προγραμματισμό Ε Αποστολόπουλος Εργαστήριο 1	---	---	Φυσική Γκανέτσος Αίθουσα 1
<b>15:00</b>	---			Εισαγωγή στον	---
<b>16:00</b>	Εισαγωγή στον	Βιοχημεία	Γραμμική	Προγραμματισμό Ε	Εισαγωγή
<b>17:00</b>	Βαϊόπουλος	Μπράλιου	Άλγεβρα	Σπαθούλας	στον
	Εργαστήριο 1	Αίθουσα 1	Αδάμ	Εργαστήριο 1	Προγραμματι σμό Ε
<b>18:00</b>		---	Εισαγωγή στον	---	Βάβουλας
<b>19:00</b>	Βιοχημεία	Αγγλικά I	Προγραμματισμό	Γραμμική	Εργαστήριο 1
	Μπράλιου	Παππά	Μάρκου	Άλγεβρα	---
	Αίθουσα 1	Αίθουσα 1	Αίθουσα 1	Αδάμ	---
				Αίθουσα 1	
Timetable generated with FET 5.14.5 on 12/14/11 12:28 AM					

Πίνακας 4.5: Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων-Α Εξάμηνο

Δεύτερο Έτος Automatic Group					
	Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή
<b>08:00</b>	Αριθμητική	---	Αριθμητική	---	---
<b>09:00</b>	Ανάλυση Ε Βάβουλας Εργαστήριο 1	Πιθανότητες & Στοιχεία Στατιστικής	Ανάλυση Ε Βάβουλας Εργαστήριο 1	Βιολογία II Μπάγκος Αίθουσα 2	---
<b>10:00</b>	Αρχές Ηλεκτρονικής Ε	Γεωργίου Αίθουσα 2	Ανατομία & Φυσιολογία I	---	Αρχιτεκτονική Υπολογιστών Κακαρούντας Αίθουσα 2
<b>11:00</b>	Βάβουλας, Βαϊόπουλος Εργαστήριο 2	Αγγλικά III Παππά Αίθουσα 2	Παπανάγνου Αίθουσα 2	---	---
<b>12:00</b>				-X-	Αίθουσα 2
<b>13:00</b>	Αρχές Ηλεκτρονικής Ε	Αρχές Ηλεκτρονικής Ε	---	-X-	---
<b>14:00</b>	Γκανέτσος Αίθουσα 2	Βάβουλας, Βαϊόπουλος Εργαστήριο 2	---	Αρχιτεκτονική Υπολογιστών Κακαρούντας Αίθουσα 2	Αρχές Ηλεκτρονικής Ε Βάβουλας, Βαϊόπουλος Εργαστήριο 2
<b>15:00</b>	---	---	---		
<b>16:00</b>	---	---	Βιολογία II	---	---
<b>17:00</b>	---	Αριθμητική Ανάλυση Ε	Μπάγκος Αίθουσα 2	Αρχιτεκτονική Υπολογιστών Κακαρούντας Αίθουσα 2	---
<b>18:00</b>	Πιθανότητες & Στοιχεία	Βαϊόπουλος Εργαστήριο 1	Αριθμητική Ανάλυση	Ανατομία & Φυσιολογία I	Αρχιτεκτονική Υπολογιστών Κακαρούντας Αίθουσα 2
<b>19:00</b>	Στατιστικής Γεωργίου Αίθουσα 2	---	Πλαγιανάκος Αίθουσα 2	Παπανάγνου Εργαστήριο Ανατομίας	
Timetable generated with FET 5.14.5 on 12/14/11 12:28 AM					

Πίνακας 4.6: Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων-Γ Εξάμηνο

<b>Τρίτο Έτος Automatic Group</b>					
	<b>Δευτέρα</b>	<b>Τρίτη</b>	<b>Τετάρτη</b>	<b>Πέμπτη</b>	<b>Παρασκευή</b>
<b>08:00</b>	Στοιχεία	---	---		
<b>09:00</b>	Θεωρίας Πληροφορίας & Κωδικών Σανδαλίδης Αίθουσα 3	---	Βάσεις Δεδομένων Βασιλακόπουλος Αίθουσα 3	Βάσεις Δεδομένων Βασιλακόπουλος Αίθουσα 3	Εισαγωγή στις Τηλεπικοινωνίες Ε Βαϊόπουλος Εργαστήριο 1
<b>10:00</b>	Γενετική & Ασθένειες -	---			---
<b>11:00</b>	Πληθυσμιακή Γενετική Μπράλιου Αίθουσα 3	Βάσεις Δεδομένων Ε Βαϊόπουλος Εργαστήριο 1	---	Γραφική με Υπολογιστές Δελήμπασης Αίθουσα 3	Βάσεις Δεδομένων Ε Βάβουλας Εργαστήριο 1
<b>12:00</b>	Ενσωματωμένα			-X-	
<b>13:00</b>	Συστήματα Υπολογιστών Καραγεώργος, Χατζηδήμου Εργαστήριο 2	Ψηφιακή Επεξεργασία Βιοσημάτων Δελήμπασης Αίθουσα 3	Βιοπληροφορική I Μπάγκος Αίθουσα 3	-X-	---
<b>14:00</b>	---		Εισαγωγή στις Τηλεπικοινωνίες Ε	Βιοπληροφορική I Μπάγκος Αίθουσα 3	Τεχνολογία Λογισμικού Κοσμόπουλος Αίθουσα 3
<b>15:00</b>	---	Στοιχεία Θεωρίας Πληροφορίας & Κωδικών Σανδαλίδης Αίθουσα 3	Βάβουλας Εργαστήριο 1		
<b>16:00</b>	Ενσωματωμένα		Τηλεϊατρική Μαγκλογιάννης Αίθουσα 3	Τηλεϊατρική Μαγκλογιάννης Αίθουσα 3	Γενετική & Ασθένειες - Πληθυσμιακή Γενετική Μπράλιου Αίθουσα 3
<b>17:00</b>	Συστήματα Υπολογιστών Αλετράς Εργαστήριο 2	Γραφική με Υπολογιστές Δελήμπασης Αίθουσα 3			
<b>18:00</b>	Εισαγωγή στις		Ψηφιακή Επεξεργασία Βιοσημάτων Ε	Τεχνολογία Λογισμικού Κοσμόπουλος Αίθουσα 3	---
<b>19:00</b>	Τηλεπικοινωνίες Σανδαλίδης Αίθουσα 3	---	Δελήμπασης Εργαστήριο 1		---
Timetable generated with FET 5.14.5 on 12/14/11 12:28 AM					

Πίνακας 4.7: Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων-Ε Εξάμηνο

Τέταρτο Έτος Automatic Group					
	Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή
08:00		Συστήματα		---	---
09:00	Θεωρία Υπολογισμού Μάρκου Αίθουσα 2	Στήριξης Ιατρικών Αποφάσεων Ιακωβίδης Αίθουσα 3	Πληροφοριακά Συστήματα Υγείας II Μαγκλογιάννης Αίθουσα 2	---	Θεωρία Γράφων Βασιλακόπουλος Αίθουσα 1
10:00	Οργάνωση & Διοίκηση	---		---	
11:00	Συστημάτων Υγείας Σίσκου Αίθουσα 2	Τεχνητή Νοημοσύνη Πλαγιανάκος Αίθουσα 1	Τεχνητή Νοημοσύνη Πλαγιανάκος Εργαστήριο 1	---	---
12:00	Οργάνωση & Διοίκηση		---	-X-	---
13:00	Συστημάτων Υγείας Σίσκου Αίθουσα 1	---	Τεχνολογίες Εφαρμογών Διαδικτύου Αναγνωστόπουλος Αίθουσα 2	-X-	---
14:00	Συστήματα	---		Συστήματα	---
15:00	Ιατρικής Απεικόνισης Αλετράς Αίθουσα 3	Θεωρία Υπολογισμού Μάρκου Αίθουσα 2	---	Στήριξης Ιατρικών Αποφάσεων Ιακωβίδης Αίθουσα 1	---
16:00	Επικοινωνία		---	---	Τεχνολογίες
17:00	Ανθρώπου Υπολογιστή Καρυδάκης Αίθουσα 2	---	---	---	Εφαρμογών Διαδικτύου Αναγνωστόπουλος Αίθουσα 2
18:00	---	Επικοινωνία			
19:00	---	Ανθρώπου Υπολογιστή Καρυδάκης Αίθουσα 2	Θεωρία Γράφων Βασιλακόπουλος Αίθουσα 3	Πληροφοριακά Συστήματα Υγείας II Μαγκλογιάννης Εργαστήριο 1	Συστήματα Ιατρικής Απεικόνισης Αλετράς Αίθουσα 3
Timetable generated with FET 5.14.5 on 12/14/11 12:28 AM					

Πίνακας 4.8: Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων-Z Εξάμηνο



Ελεύθερη Επιλογή Automatic Group					
	Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή
08:00	---	---	---	---	---
09:00	---	---	---	---	---
10:00	---	---	---	---	---
11:00	---	---	---	---	---
12:00	---	---	---	-X-	---
13:00	---	---	Παιδαγωγικά	-X-	---
14:00	---	---	Παρασκευοπούλου Αίθουσα 1	---	---
15:00	---	---	---	---	Δημόσιες Σχέσεις Βασιλάκη Αίθουσα 1
16:00	Κοινωνιολογία Γούγα	---	---	---	Αίθουσα 1
17:00	Αίθουσα 3	---	---	---	Εισαγωγή στην Πολιτική Οικονομία Βασιλάκη Αίθουσα 1
18:00	---	---	---	---	---
19:00	---	---	---	---	---
Timetable generated with FET 5.14.5 on 12/14/11 12:28 AM					

Πίνακας 4.9: Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων-Ελεύθερη Επιλογή

#### 4.2.2 Εαρινό Εξάμηνο 2010-2011

Το εαρινό εξάμηνο περιλαμβάνει τα παρακάτω δεδομένα τα οποία θα καταχωρηθούν στην εφαρμογή FET.

**Διδάσκοντες:** Αδάμ, Αλετράς, Αναγνωστόπουλος, Αποστολόπουλος, Ασημάκης, Βάβουλας, Βασιλάκη, Βασιλακόπουλος, Βαϊόπουλος, Δελήμπασης, Λουκόπουλος, Μάρκου, Μαγκλογιάννης, Μπάγκος, Μυλωνάς, Νικολόπουλος, Παπαγεωργίου, Παπαδοπούλου, Παπανάγνου, Παππά, Παρασκευοπούλου, Πλαγιανάκος, Σανδαλίδης, Σπαθούλας, Φούρλας, Χατζηδήμου.

**Έτη φοίτησης:** Πρώτο Έτος, Δεύτερο Έτος, Τρίτο Έτος, Τέταρτο Έτος

**Αίθουσες:** Αίθουσα 1, Αίθουσα 2, Αίθουσα 3, Εργαστήριο 1, Εργαστήριο 2, Εργαστήριο 3, Εργαστήριο Ανατομίας.

**Μαθήματα Β' Εξαμήνου:** Λογική Σχεδίαση, Αντικειμενοστραφής Προγραμματισμός, Εισαγωγή στη Βιοϊατρική Τεχνολογία, Διακριτά Μαθηματικά, Αγγλικά II, Μαθηματική Ανάλυση II, Βιολογία I.

**Μαθήματα Δ' Εξαμήνου:** Λειτουργικά Συστήματα, Σήματα και Συστήματα, Δομές Δεδομένων & Αλγόριθμοι, Δίκτυα Υπολογιστών, Αγγλικά IV, Ανατομία & Φυσιολογία II, Βιοστατιστική I.

**Μαθήματα ΣΤ' Εξαμήνου:** Τεχνολογία Ιατρικού Εξοπλισμού, Θεωρία Γλωσσών, Ψηφιακές Επικοινωνίες, Ασύρματα Επικοινωνιακά Συστήματα, Αναγνώριση Προτύπων, Πληροφοριακά

Συστήματα Υγείας I, Ανάλυση Συστημάτων, Βιοπληροφορική II.

**Μαθήματα Η' Εξαμήνου:** Ειδικά Θέματα Βιοπληροφορικής & Βιοηθικής, Συστήματα Αυτομάτου Ελέγχου και Εφαρμογές στη Βιοϊατρική, Διασυνδεδεμένα Συστήματα Υπολογιστών, Προσομοίωση Βιολογικών Συστημάτων, Ανάλυση Βιοϊατρικών Εικόνων, Εξόρυξη Δεδομένων και Ανακάλυψη Γνώσης, Εφαρμοσμένα Μαθηματικά, Ασφάλεια Συστημάτων Υπολογιστών.

**Ελεύθερα Μαθήματα:** Δημόσιο Δίκαιο, Θεσμοί και Διοίκηση της Ε.Ε, Διδακτική της Πληροφορικής, Επιχειρηματικότητα και Καινοτομία.

Όπως στο χειμερινό εξάμηνο έτσι και στο εαρινό εξάμηνο πρέπει να δημιουργήσουμε ένα αρχείο στην εφαρμογή FET έστω με το όνομα “earino.fet” (εικόνα 4.39) και να καταχωρήσουμε όλες τις παραπάνω πληροφορίες επιλέγοντας από το μενού το θέμα Data (εικόνα 4.9).

- Για την καταχώρηση των βασικών πληροφοριών όπως το όνομα του πανεπιστημίου, το όνομα του τμήματος, τις ώρες και τις μέρες της εβδομάδος επιλέγουμε Data-->Basic (εικόνες 4.3, 4.4, 4.5, 4.6, 4.7, 4.8) όπως ακριβώς κάναμε και στο χειμερινό εξάμηνο.
- Για την καταχώρηση των διδασκόντων επιλέγουμε από το μενού Data-->Teachers (εικόνα 4.40).
- Για την καταχώρηση των ομάδων των φοιτητών που χωρίζονται σε έτη επιλέγουμε από το μενού Data-->Students (εικόνες 4.11 και 4.12).
- Για την καταχώρηση των μαθημάτων επιλέγουμε από το μενού Data-->Subjects (εικόνα 4.41).
- Για την καταχώρηση των χώρων φοίτησης επιλέγουμε από το μενού Data-->Space (εικόνες 4.14, 4.15, 4.16, 4.17, 4.18).
- Για την καταχώρηση των δραστηριοτήτων του εαρινού εξαμήνου επιλέγουμε από το μενού Data-->Activities, για την ακρίβεια καταχωρούμε τις παρακάτω δραστηριότητες του πίνακα 4.10.

ΜΑΘΗΜΑ	ΔΙΔΑΣΚΩΝ	ΔΙΑΡΚΕΙΑ	ΕΤΟΣ ΦΟΙΤΗΣΗΣ
Λογική Σχεδίαση	Ασημάκης	1 Δίωρο	Πρώτο Έτος
Λογική Σχεδίαση Εργαστήριο	Βάβουλας Βαϊόπουλος	3 Δίωρα	Πρώτο Έτος
Αντικειμενοστραφής Προγραμματισμός Εργαστήριο	Αποστόλοπουλος	1 Δίωρο	Πρώτο Έτος
Αντικειμενοστραφής Προγραμματισμός Εργαστήριο	Βαϊόπουλος	1 Δίωρο	Πρώτο Έτος
Αντικειμενοστραφής Προγραμματισμός Εργαστήριο	Βάβουλας	1 Δίωρο	Πρώτο Έτος
Αντικειμενοστραφής Προγραμματισμός	Μάρκου	1 Τρίωρο	Πρώτο Έτος
Εισαγωγή στη Βιοϊατρική Τεχνολογία	Αλετράς	1 Τρίωρο	Πρώτο Έτος

Εισαγωγή στη Βιοϊατρική Τεχνολογία Εργαστήριο	Αλετράς Χατζηδημού	1 Τετράωρο	Πρώτο Έτος
Διακριτά Μαθηματικά	Πλαγιανάκος	2 Δίωρα	Πρώτο Έτος
Αγγλικά II	Παππά	1 Δίωρο	Πρώτο Έτος
Μαθηματική Ανάλυση II	Αδάμ	1 Δίωρο 1 Τρίωρο	Πρώτο Έτος
Βιολογία I	Μπάγκος	2 Δίωρα	Πρώτο Έτος
Λειτουργικά Συστήματα	Λουκόπουλος	2 Δίωρα	Δεύτερο Έτος
Λειτουργικά Συστήματα Εργαστήριο	Βαϊόπουλος	1 Δίωρο	Δεύτερο Έτος
Σήματα & Συστήματα	Ασημάκης	2 Δίωρα	Δεύτερο Έτος
Δομές Δεδομένων & Αλγόριθμοι Εργαστήριο	Λουκόπουλος	2 Δίωρα	Δεύτερο Έτος
Δίκτυα Υπολογιστών	Αναγνωστόπουλος	2 Δίωρα	Δεύτερο Έτος
Αγγλικά IV	Παππά	1 Δίωρο	Δεύτερο Έτος
Ανατομία & Φυσιολογία II	Παπανάγνου	1 Δίωρο 1 Τρίωρο	Δεύτερο Έτος
Δομές Δεδομένων & Αλγόριθμοι	Βασιλακόπουλος	2 Δίωρα	Δεύτερο Έτος
Βιοστατιστική I Εργαστήριο	Νικολόπουλος	1 Τετράωρο	Δεύτερο Έτος
Βιοστατιστική I	Νικολόπουλος	1 Δίωρο	Δεύτερο Έτος
Τεχνολογία Ιατρικού Εξοπλισμού	Αλετράς	1 Τρίωρο	Τρίτο Έτος
Τεχνολογία Ιατρικού Εξοπλισμού Εργαστήριο	Αλετράς	1 Δίωρο	Τρίτο Έτος
Θεωρία Γλωσσών	Μάρκου	2 Δίωρα	Τρίτο Έτος
Ψηφιακές Επικοινωνίες	Σανδαλίδης	2 Δίωρα	Τρίτο Έτος
Ασύρματα Επικοινωνιακά Συστήματα	Σανδαλίδης	2 Δίωρα	Τρίτο Έτος
Αναγνώριση Προτύπων	Πλαγιανάκος	2 Δίωρα	Τρίτο Έτος
Πληροφοριακά Συστήματα Υγείας I	Μαγκλογιάννης	2 Δίωρα	Τρίτο Έτος
Ανάλυση Συστημάτων	Παπαγεωργίου	1 Δίωρο	Τρίτο Έτος
Ανάλυση Συστημάτων Εργαστήριο	Σπαθούλας	1 Δίωρο	Τρίτο Έτος
Ανάλυση Συστημάτων Εργαστήριο	Παπαγεωργίου	1 Δίωρο	Τρίτο Έτος
Βιοπληροφορική II	Μπάγκος	2 Δίωρα	Τρίτο Έτος
Ειδικά Θέματα Βιοπληροφορικής & Βιοηθική	Παπαδοπούλου	1 Δίωρο	Τέταρτο Έτος

Ειδικά Θέματα Βιοπληροφορικής & Βιοηθική	Μπάγκος	1 Δίωρο	Τέταρτο Έτος
Συστήματα Αυτομάτου Ελέγχου & Εφαρμογές στη Βιοϊατρική	Φούρλας	2 Δίωρα	Τέταρτο Έτος
Διασυνδεδεμένα Συστήματα Υπολογιστών	Αναγνωστόπουλος	2 Δίωρα	Τέταρτο Έτος
Προσομοίωση Βιολογικών Συστημάτων	Δελήμπασης	1 Τρίωρο	Τέταρτο Έτος
Ανάλυση Βιοϊατρικών Εικόνων Εργαστήριο	Δελήμπασης	1 Τρίωρο	Τέταρτο Έτος
Ανάλυση Βιοϊατρικών Εικόνων	Μαγκλογιάννης	1 Τρίωρο	Τέταρτο Έτος
Εξόρυξη Δεδομένων & Ανακάλυψη Γνώσης	Βασιλακόπουλος	2 Δίωρα	Τέταρτο Έτος
Εφαρμοσμένα Μαθηματικά	Αδάμ	2 Δίωρα	Τέταρτο Έτος
Ασφάλεια Συστημάτων Υπολογιστών	Μυλωνάς	2 Δίωρα	Τέταρτο Έτος
Δημόσιο Δίκαιο	Βασιλάκη	1 Δίωρο	Ελεύθερη Επιλογή
Θεσμοί & Διοίκηση της Ε.Ε	Βασιλάκη	1 Δίωρο	Ελεύθερη Επιλογή
Διδακτική της Πληροφορικής	Παρασκευοπούλου	1 Δίωρο	Ελεύθερη Επιλογή
Επιχειρηματικότητα & Καινοτομία	-----	1 Δίωρο	Ελεύθερη Επιλογή

Πίνακας 4.10: Δραστηριότητες Εαρινού Εξαμήνου

- Για τους περιορισμούς χρόνου που αφορούν τα διαλείμματα επιλέγουμε από το μενού Time-->Break και καταχωρούμε πότε οι φοιτητές έχουν διαλείμματα και διακοπές από τα μαθήματα. Στη δική μας περίπτωση όπως ανάφερα και στο χειμερινό εξάμηνο οι ώρες αυτές σχετίζονται με τη συνέλευση των φοιτητών (εικόνες 4.20 και 4.21).
- Για τους περιορισμούς χρόνου που αναφέρονται στις μη διαθέσιμες ώρες των διδασκόντων επιλέγουμε από το μενού Time-->Teachers-->A teacher's not available times και καταχωρούμε τις πληροφορίες του πίνακα 4.11.
- Για τους περιορισμούς τόπου που σχετίζονται με τις διαθεσιμότητες των αιθουσών επιλέγουμε από το μενού Space-->Rooms-->A room's not available times και καταχωρούμε τις μη διαθέσιμες ώρες για την αίθουσα 1 (εικόνες 4.24 και 4.25) καθώς σε αυτήν πραγματοποιείται η συνέλευση των φοιτητών.
- Για τους περιορισμούς τόπου σχετικά με τις προτιμήσεις των διδασκόντων για τους χώρους φοίτησης επιλέγουμε από το μενού Space-->Teachers-->A teacher has a home room και καταχωρούμε την κα Βασιλάκη να επιλέγει την αίθουσα 1 (εικόνα 4.42).
- Για τους περιορισμούς τόπου που αναφέρονται στις προτιμήσεις των ομάδων των φοιτητών για τους χώρους φοίτησης επιλέγουμε από το μενού Space-->Students-->A set of students has

a home room και καταχωρούμε για τα τρία πρώτα έτη φοίτησης τις τρεις αίθουσες που υπάρχουν (εικόνες 4.31, 4.32, 4.33).

<b>ΔΙΔΑΣΚΩΝ</b>	<b>ΜΗ ΔΙΑΘΕΣΙΜΕΣ ΗΜΕΡΕΣ</b>
Αδάμ	Δευτέρα, Τρίτη
Αλετράς	Πέμπτη, Παρασκευή
Αναγνωστόπουλος	Δευτέρα, Παρασκευή
Αποστολόπουλος	Τρίτη, Τετάρτη, Πέμπτη, Παρασκευή
Ασημάκης	Τετάρτη, Πέμπτη, Παρασκευή
Βάβουλας	Τετάρτη
Βασιλάκη	Τρίτη, Τετάρτη, Πέμπτη, Παρασκευή
Βασιλακόπουλος	Δευτέρα, Τρίτη
Βαϊόπουλος	Τρίτη, Τετάρτη
Δελήμπασης	Δευτέρα, Πέμπτη, Παρασκευή
Λουκόπουλος	Δευτέρα, Πέμπτη, Παρασκευή
Μάρκου	Πέμπτη, Παρασκευή
Μαγκλογιάννης	Δευτέρα, Παρασκευή
Μπάγκος	Δευτέρα, Τρίτη
Μυλωνάς	Δευτέρα, Τρίτη, Τετάρτη
Νικολόπουλος	Δευτέρα, Τρίτη, Τετάρτη, Πέμπτη
Παπαγεωργίου	Δευτέρα, Τρίτη, Παρασκευή
Παπαδοπούλου	Τρίτη, Τετάρτη, Πέμπτη, Παρασκευή
Παπανάγνου	Δευτέρα, Τρίτη, Παρασκευή
Παππά	Δευτέρα, Τετάρτη, Πέμπτη, Παρασκευή
Παρασκευοπούλου	Δευτέρα, Τρίτη, Πέμπτη, Παρασκευή
Πλαγιανάκος	Δευτέρα, Παρασκευή
Σανδαλίδης	Πέμπτη, Παρασκευή
Σπαθούλας	Δευτέρα, Τρίτη, Τετάρτη, Παρασκευή
Φούρλας	Δευτέρα, Τετάρτη, Παρασκευή
Χατζηδήμου	Δευτέρα, Πέμπτη, Παρασκευή

*Πίνακας 4.11: Διδάσκοντες-Μη Διαθέσιμες Ημέρες για το εαρινό εξάμηνο*

- Για τους περιορισμούς τόπου που αφορούν τις προτιμήσεις των μαθημάτων για τους διαθέσιμους χώρους επιλέγουμε από το μενού Space-->Subjects-->A subject has a preferred room και καταχωρούμε τις κατάλληλες πληροφορίες από τον πίνακα 4.12.
- Για τους περιορισμούς τόπου που αναφέρονται στις προτιμήσεις των δραστηριοτήτων για τους διαθέσιμους χώρους επιλέγουμε από το μενού Space-->Activities-->An activity has a preferred room και καταχωρούμε τις παρακάτω πληροφορίες από τον πίνακα 4.13.

Τέλος, αφού έχουμε καταχωρήσει και τους περιορισμούς χρόνου και τόπου μπορούμε να δημιουργήσουμε το ωρολόγιο πρόγραμμα. Παρακάτω παρουσιάζονται τα αποτελέσματα του εαρινού ωρολογίου προγράμματος που μας έδωσε η εφαρμογή FET για τα τέσσερα έτη καθώς και για τα μαθήματα ελεύθερης επιλογής (πίνακες 4.14, 4.15, 4.16, 4.17, 4.18).

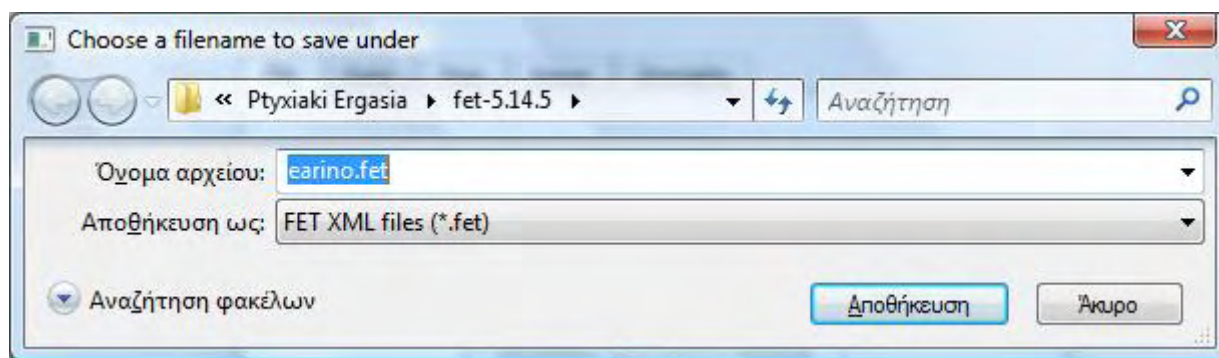
<b>ΜΑΘΗΜΑ</b>	<b>ΧΩΡΟΣ ΠΡΟΤΙΜΗΣΗΣ</b>
Λογική Σχεδίαση (Εργαστήριο)	Εργαστήριο 2
Αντικειμενοστραφής Προγραμματισμός (Εργαστήριο)	Εργαστήριο 1
Εισαγωγή στη Βιοϊατρική Τεχνολογία (Εργαστήριο)	Εργαστήριο 1
Λειτουργικά Συστήματα (Εργαστήριο)	Εργαστήριο 1
Βιοστατιστική I (Εργαστήριο)	Εργαστήριο 1
Ανάλυση Συστημάτων (Εργαστήριο)	Εργαστήριο 1
Τεχνολογία Ιατρικού Εξοπλισμού (Εργαστήριο)	Εργαστήριο 1
Ανάλυση Βιοϊατρικών Εικόνων (Εργαστήριο)	Εργαστήριο 1
Διδακτική της Πληροφορικής	Εργαστήριο 2
Επιχειρηματικότητα & Καινοτομία	Αίθουσα 1
Προσομοίωση Βιολογικών Συστημάτων	Εργαστήριο 1

Πίνακας 4.12: Μαθήματα και Χώροι Προτίμησης για το εαρινό εξάμηνο

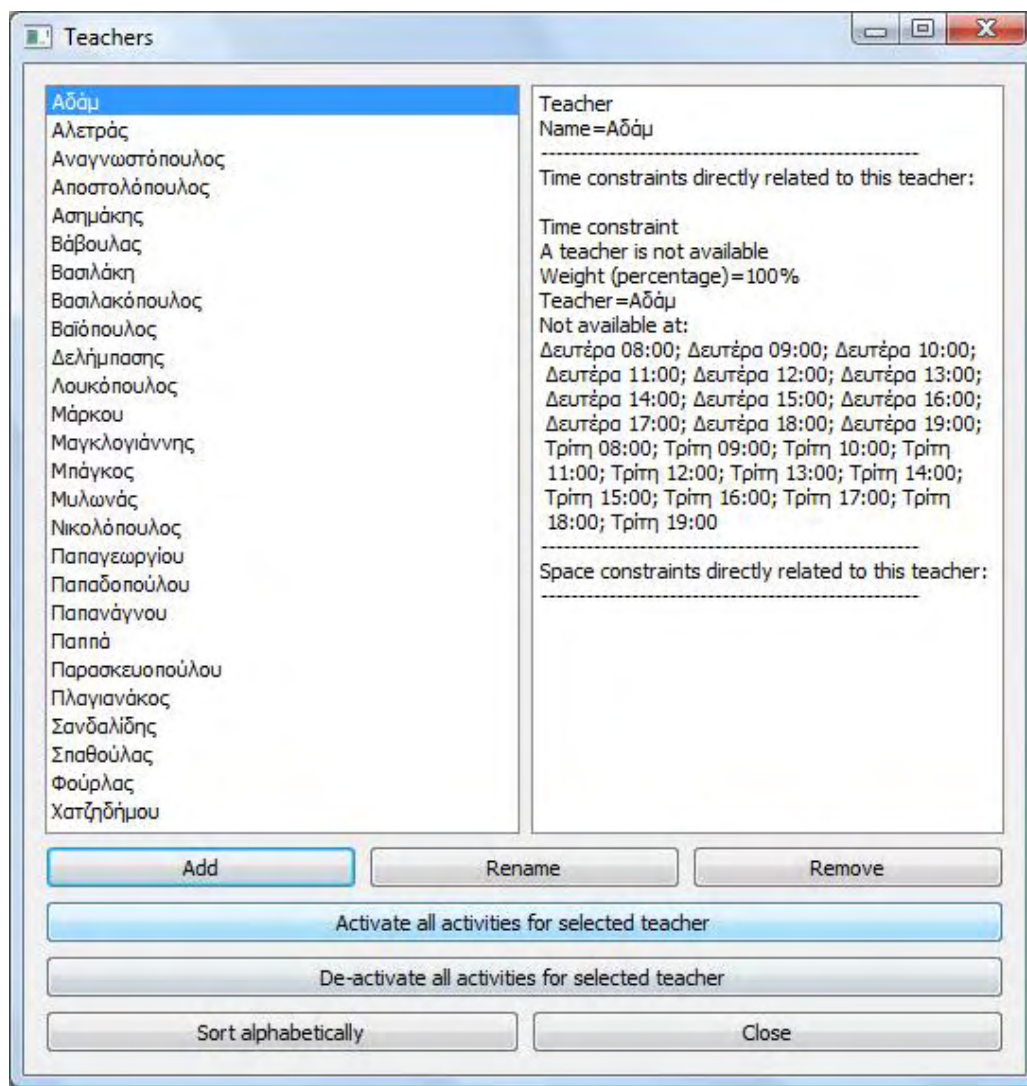
<b>ΔΡΑΣΤΗΡΙΟΤΗΤΑ</b>	<b>ΧΩΡΟΣ ΠΡΟΤΙΜΗΣΗΣ</b>
2/4-Σήματα & Συστήματα-Ασημάκης	Εργαστήριο 1
2/4-Δομές Δεδομένων & Αλγόριθμοι (Ε)-Λουκόπουλος	Εργαστήριο 1
2/5-Ανατομία & Φυσιολογία II-Παπανάγνου	Εργαστήριο Ανατομίας
2/4-Συστήματα Αυτομάτου Ελέγχου & Εφαρμογές στη Βιοϊατρική-Φούρλας	Εργαστήριο 1
2/4-Συστήματα Αυτομάτου Ελέγχου & Εφαρμογές στη Βιοϊατρική-Φούρλας	Αίθουσα 1
2-Ειδικά Θέματα Βιοπληροφορικής & Βιοηθική-Παπαδοπούλου	Αίθουσα 2
2-Ειδικά Θέματα Βιοπληροφορικής & Βιοηθική-Μπάγκος	Αίθουσα 3
2/4-Διασυνδεδεμένα Συστήματα Υπολογιστών-Αναγνωστόπουλος	Αίθουσα 1
2/4-Διασυνδεδεμένα Συστήματα Υπολογιστών-Αναγνωστόπουλος	Αίθουσα 2
1-Προσομοίωση Βιολογικών Συστημάτων (Ε)-Δελήμπασης	Αίθουσα 2
2/4-Εφαρμοσμένα Μαθηματικά-Αδάμ	Αίθουσα 1
2/4-Εφαρμοσμένα Μαθηματικά-Αδάμ	Αίθουσα 2

2/4-Εξόρυξη Δεδομένων & Ανακάλυψη Γνώσης-Βασιλακόπουλος	Αίθουσα 1
2/4-Εξόρυξη Δεδομένων & Ανακάλυψη Γνώσης-Βασιλακόπουλος	Αίθουσα 2
2/4-Ασφάλεια Συστημάτων Υπολογιστών-Μυλωνάς	Αίθουσα 1
2/4-Ασφάλεια Συστημάτων Υπολογιστών-Μυλωνάς	Αίθουσα 3
3-Ανάλυση Βιοϊατρικών Εικόνων-Μαγκλογιάννης	Αίθουσα 1

Πίνακας 4.13: Δραστηριότητες και Χώροι Προτίμησης για το εαρινό εξάμηνο

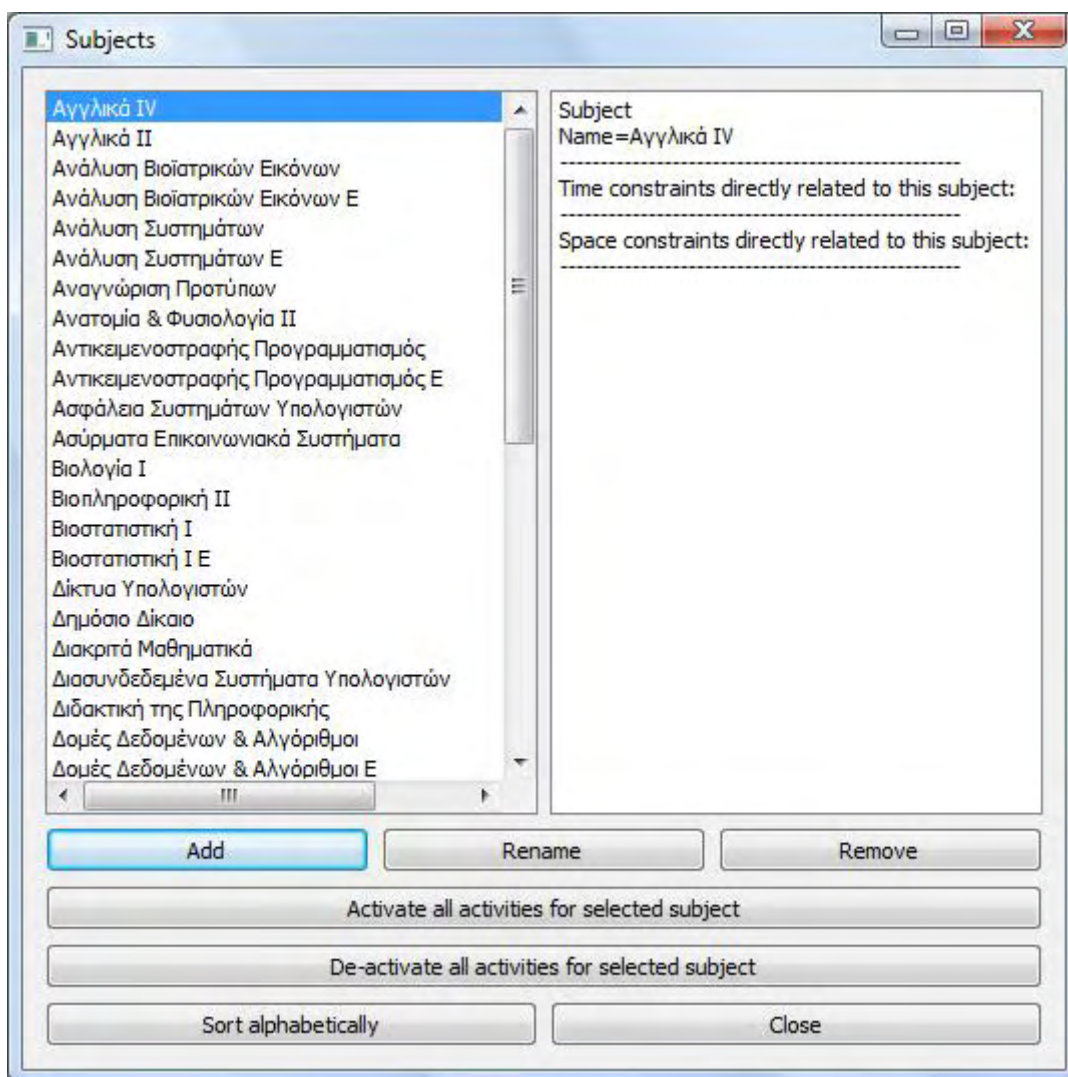


Εικόνα 4.39: Δημιουργία αρχείου για το εαρινό εξάμηνο

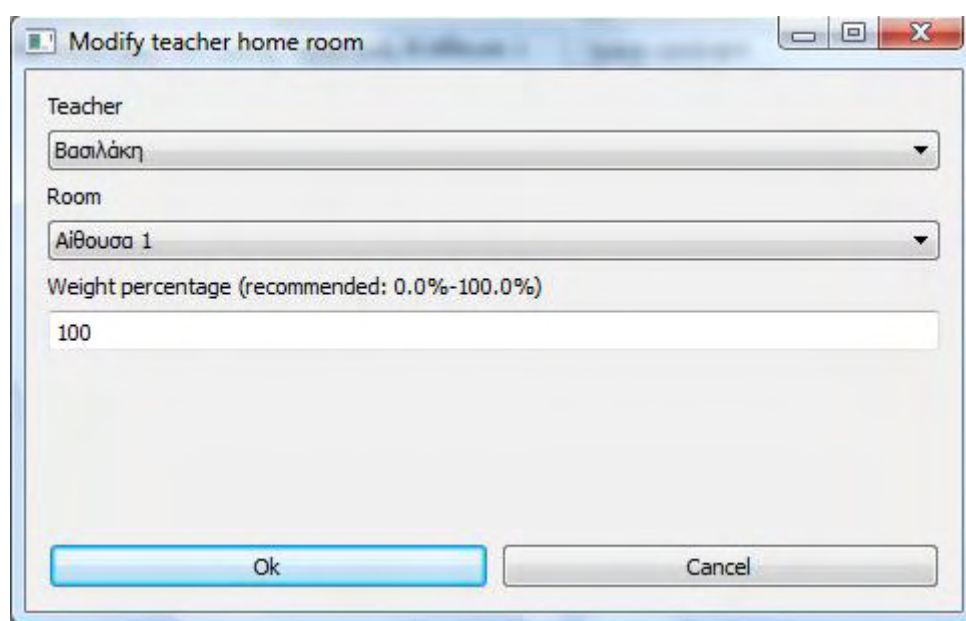


Εικόνα 4.40: Διδάσκοντες του εαρινού εξαμήνου





Εικόνα 4.41: Subjects-Μαθήματα Εαρινού Εξαμήνου



Εικόνα 4.42: Βασιλάκη-Αίθουσα 1

<b>Πρώτο Έτος Automatic Group</b>					
	<b>Δευτέρα</b>	<b>Τρίτη</b>	<b>Τετάρτη</b>	<b>Πέμπτη</b>	<b>Παρασκευή</b>
<b>08:00</b>	Λογική Σχεδίαση	Εισαγωγή στη	Μαθηματική	---	---
<b>09:00</b>	Ασημάκης Αίθουσα 1	Βιοϊατρική Τεχνολογία	Ανάλυση II Αδάμ Αίθουσα 1	---	---
<b>10:00</b>	Αντικειμενοστραφής Προγραμματισμός Ε	Αλετράς Αίθουσα 1	---	Αντικειμενοστραφής Προγραμματισμός Ε	---
<b>11:00</b>	Αποστολόπουλος Εργαστήριο 1	Διακριτά Μαθηματικά	---	Βαϊόπουλος Εργαστήριο 1	Μαθηματική Ανάλυση II
<b>12:00</b>	Αντικειμενοστραφής Προγραμματισμός Ε	Πλαγιανάκος Αίθουσα 1	---	-X-	Αδάμ Αίθουσα 1
<b>13:00</b>	Βάβουλας Εργαστήριο 1	Αγγλικά II Παππά Αίθουσα 1	---	-X-	---
<b>14:00</b>			---		
<b>15:00</b>	Λογική Σχεδίαση Ε Βάβουλας, Βαϊόπουλος Εργαστήριο 2	---	Εισαγωγή στη Βιοϊατρική Τεχνολογία Ε Αλετράς, Χατζηδήμου Εργαστήριο 1	Βιολογία I Μπάγκος Αίθουσα 1	Λογική Σχεδίαση Ε Βάβουλας, Βαϊόπουλος Εργαστήριο 2
<b>16:00</b>	---	Εισαγωγή στη	---	Διακριτά	---
<b>17:00</b>		Βιοϊατρική Τεχνολογία Ε	Βιολογία I	Μαθηματικά Πλαγιανάκος Αίθουσα 1	---
<b>18:00</b>	Αντικειμενοστραφής Προγραμματισμός Μάρκου Αίθουσα 1	Αλετράς, Χατζηδήμου Εργαστήριο 1	Μπάγκος Αίθουσα 1	Λογική Σχεδίαση Ε Βάβουλας,	---
<b>19:00</b>		---	---	Βαϊόπουλος Εργαστήριο 2	---
Timetable generated with FET 5.14.5 on 12/21/11 6:36 PM					

Πίνακας 4.14: Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων-Β Εξάμηνο

Δεύτερο Έτος Automatic Group					
	Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή
<b>08:00</b>	Λειτουργικά Συστήματα Ε	---	Ανατομία & Φυσιολογία ΙΙ	Δίκτυα	---
<b>09:00</b>	Βαϊόπουλος Εργαστήριο 1	Δομές Δεδομένων & Αλγόριθμοι Ε Λουκόπουλος Εργαστήριο 1	Παπανάγνου Εργαστήριο Ανατομίας	Υπολογιστών Αναγνωστόπουλος Αίθουσα 2	Δομές Δεδομένων & Αλγόριθμοι Βασιλακόπουλος Αίθουσα 2
<b>10:00</b>	---	---	---	---	---
<b>11:00</b>	---	---	---	---	---
<b>12:00</b>	---	Αγγλικά ΙV Παππά Αίθουσα 2	Λειτουργικά Συστήματα Λουκόπουλος Αίθουσα 2	-X-	---
<b>13:00</b>	---	Δίκτυα	---	-X-	Βιοστατιστική Ι Ε Νικολόπουλος Εργαστήριο 1
<b>14:00</b>	---	Υπολογιστών Αναγνωστόπουλος Αίθουσα 2	---	---	---
<b>15:00</b>	---	Λειτουργικά	---	Ανατομία & Φυσιολογία ΙΙ	---
<b>16:00</b>	Σήματα και Συστήματα Ασημάκης	Συστήματα Λουκόπουλος Αίθουσα 2	---	Παπανάγνου Αίθουσα 2	Βιοστατιστική Ι Νικολόπουλος Αίθουσα 2
<b>17:00</b>	Εργαστήριο 1	Σήματα και Συστήματα	Δομές Δεδομένων & Αλγόριθμοι Ε	Δομές Δεδομένων & Αλγόριθμοι	---
<b>18:00</b>	---	Ασημάκης Αίθουσα 2	Λουκόπουλος Αίθουσα 2	Βασιλακόπουλος Αίθουσα 2	---
<b>19:00</b>	---	---	---	---	---
Timetable generated with FET 5.14.5 on 12/21/11 6:36 PM					

Πίνακας 4.15: Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων-Δ Εξάμηνο

<b>Τρίτο Έτος Automatic Group</b>					
	<b>Δευτέρα</b>	<b>Τρίτη</b>	<b>Τετάρτη</b>	<b>Πέμπτη</b>	<b>Παρασκευή</b>
<b>08:00</b>	---	Ασύρματα	Ανάλυση	Ανάλυση	---
<b>09:00</b>	Ψηφιακές Επικοινωνίες Σανδαλίδης Αίθουσα 3	Επικοινωνιακά Συστήματα Σανδαλίδης Αίθουσα 3	Συστημάτων Ε Παπαγεωργίου Εργαστήριο 1	Συστημάτων Παπαγεωργίου Αίθουσα 3	---
<b>10:00</b>		Ψηφιακές	Βιοπληροφορική	---	Βιοπληροφορική II Μπάγκος Αίθουσα 3
<b>11:00</b>	Θεωρία Γλωσσών Μάρκου Αίθουσα 3	Επικοινωνίες Σανδαλίδης Αίθουσα 3	II Μπάγκος Αίθουσα 3	---	
<b>12:00</b>	Αίθουσα 3	Πληροφοριακά	Τεχνολογία	-X-	---
<b>13:00</b>	Ασύρματα Επικοινωνιακά Συστήματα Σανδαλίδης Αίθουσα 3	Συστήματα Υγείας I Μαγκλογιάννης Αίθουσα 3	Ιατρικού Εξοπλισμού Ε Αλετράς Εργαστήριο 1	-X-	---
<b>14:00</b>	Αίθουσα 3	---	---	---	---
<b>15:00</b>	---	Αναγνώριση	Πληροφοριακά	Ανάλυση	---
<b>16:00</b>	Τεχνολογία Ιατρικού Εξοπλισμού Αλετράς Αίθουσα 3	Προτύπων Πλαγιανάκος Αίθουσα 3	Συστήματα Υγείας I Μαγκλογιάννης Αίθουσα 3	Συστημάτων Ε Σπαθούλας Εργαστήριο 1	---
<b>17:00</b>		Θεωρία	---	---	---
<b>18:00</b>		Γλωσσών Μάρκου Αίθουσα 3	Αναγνώριση Προτύπων Πλαγιανάκος	---	---
<b>19:00</b>	---	---	Αίθουσα 3	---	---
Timetable generated with FET 5.14.5 on 12/21/11 6:36 PM					

Πίνακας 4.16: Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων-ΣΤ Εξάμηνο

Τέταρτο Έτος Automatic Group					
	Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή
08:00	---	---	Εξόρυξη Δεδομένων & Ανακάλυψη		---
09:00	---	---	Γνώσης Βασιλακόπουλος Αίθουσα 2	Ανάλυση Βιοϊατρικών Εικόνων	Ασφάλεια Συστημάτων Υπολογιστών
10:00	---	---	Προσομοίωση Βιολογικών Συστημάτων Ε Δελήμπασης Αίθουσα 2	Μαγκλογιάννης Αίθουσα 1	Μυλωνάς Αίθουσα 1
11:00	---	---	---	---	---
12:00	---	---	---	-X-	Ειδικά Θέματα
13:00	Ειδικά Θέματα Βιοπληροφορικής & Βιοηθική Παπαδοπούλου	Ανάλυση Βιοϊατρικών Εικόνων Ε Δελήμπασης Εργαστήριο 1	Εφαρμοσμένα Μαθηματικά Αδάμ Αίθουσα 1	-X-	Βιοπληροφορικής & Βιοηθική Μπάγκος Αίθουσα 3
14:00	Αίθουσα 2	---	Διασυνδεδεμένα Συστήματα	Ασφάλεια Συστημάτων	---
15:00	---	Διασυνδεδεμένα Συστήματα Υπολογιστών Αναγνωστόπουλος	Υπολογιστών Αναγνωστόπουλος Αίθουσα 2	Υπολογιστών Μυλωνάς Αίθουσα 3	Εξόρυξη Δεδομένων & Ανακάλυψη Γνώσης Βασιλακόπουλος
16:00	---	ος Αίθουσα 1	---	---	Αίθουσα 1
17:00	---	---	---	Συστήματα Αυτομάτου Ελέγχου & Εφαρμογές στη Βιοϊατρική Φούρλας Εργαστήριο 1	---
18:00	---	Συστήματα Αυτομάτου Ελέγχου & Εφαρμογές στη Βιοϊατρική Φούρλας Αίθουσα 1	Προσομοίωση Βιολογικών Συστημάτων Δελήμπασης Εργαστήριο 1	---	Εφαρμοσμένα Μαθηματικά Αδάμ Αίθουσα 2
19:00	---	---	---	---	---
Timetable generated with FET 5.14.5 on 12/21/11 6:36 PM					

Πίνακας 4.17: Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων-Η Εξάμηνο

Ελεύθερη Επιλογή Automatic Group					
	Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή
08:00	---	---	---	---	---
09:00	---	---	---	---	---
10:00	Δημόσιο Δίκαιο	---	Διδακτική της	---	---
11:00	Βασιλάκη Αίθουσα 1	---	Πληροφορικής Παρασκευοπούλου Εργαστήριο 2	---	---
12:00	Θεσμοί και Διοίκηση της Ε.Ε	---	---	-X-	---
13:00	Βασιλάκη Αίθουσα 1	---	---	-X-	---
14:00	Επιχειρηματικότητα & Καινοτομία	---	---	---	---
15:00	Αίθουσα 1	---	---	---	---
16:00	---	---	---	---	---
17:00	---	---	---	---	---
18:00	---	---	---	---	---
19:00	---	---	---	---	---
Timetable generated with FET 5.14.5 on 12/21/11 6:36 PM					

Πίνακας 4.18: Εβδομαδιαίο Ωρολόγιο Πρόγραμμα Μαθημάτων-Ελεύθερη Επιλογή

#### 4.2.3 Παρατηρήσεις

- Για την καταχώρηση των δραστηριοτήτων στην ωρολόγια εφαρμογή FET χρειάζεται εκ των προτέρων να γνωρίζουμε για την κάθε δραστηριότητα ποιος διδάσκων την έχει αναλάβει καθώς και από πόσα δίωρα ή τρίωρα αποτελείται. Μία δραστηριότητα μπορεί να αφορά είτε τη θεωρία του μαθήματος είτε το εργαστήριο του.
- Η καταχώρηση των διδασκόντων, των ομάδων φοίτησης και των μαθημάτων θα ήταν προτιμότερο να είναι ίδια και για τα δύο εξάμηνα του έτους. Δηλαδή το αρχείο εισόδου καλό είναι να περιλαμβάνει όλους τους διδάσκοντες, όλες τις ομάδες φοίτησης και όλα τα μαθήματα του προγράμματος σπουδών με σκοπό να κερδίζεται χρόνος αφού η παραγωγή ενός ωρολογίου προγράμματος στηρίζεται στις δραστηριότητες που είναι δηλωμένες.
- Για τους περιορισμούς χρόνου ενός τμήματος κύριο ρόλο διαδραματίζει η καταχώρηση των δεδομένων σχετικά με τις διαθεσιμότητες των διδασκόντων. Δηλαδή πρέπει να γνωρίζουμε για τον κάθε διδάσκοντα ποιες ημέρες και ώρες μπορεί να διδάσκει έτσι ώστε να δημιουργηθεί ένα ωρολόγιο πρόγραμμα που ταιριάζει σε όλους τους διδάσκοντες.
- Για τους περιορισμούς τόπου ενός τμήματος προτιμότερο είναι να καταχωρούμε πρώτα για κάθε έτος φοίτησης το χώρο προτίμησης του και ύστερα τις απαιτήσεις των δραστηριοτήτων για τους χώρους φοίτησης.

# ΚΕΦΑΛΑΙΟ 5

## Εναλλακτικές εφαρμογές ωρολογίων προγραμμάτων

Μην κρίνεις την ημέρα από τη σοδειά που θέρисες,  
αλλά από τους σπόρους που έσπειρες.

~Robert Louis Stevenson (1850-1894)

Στο κεφάλαιο αυτό θα γίνει η παρουσίαση 3 εναλλακτικών εφαρμογών για την εξαγωγή ωρολογίων προγραμμάτων. Θα παρουσιαστούν τα κύρια χαρακτηριστικά των εφαρμογών σε σύγκριση με την εφαρμογή FET.

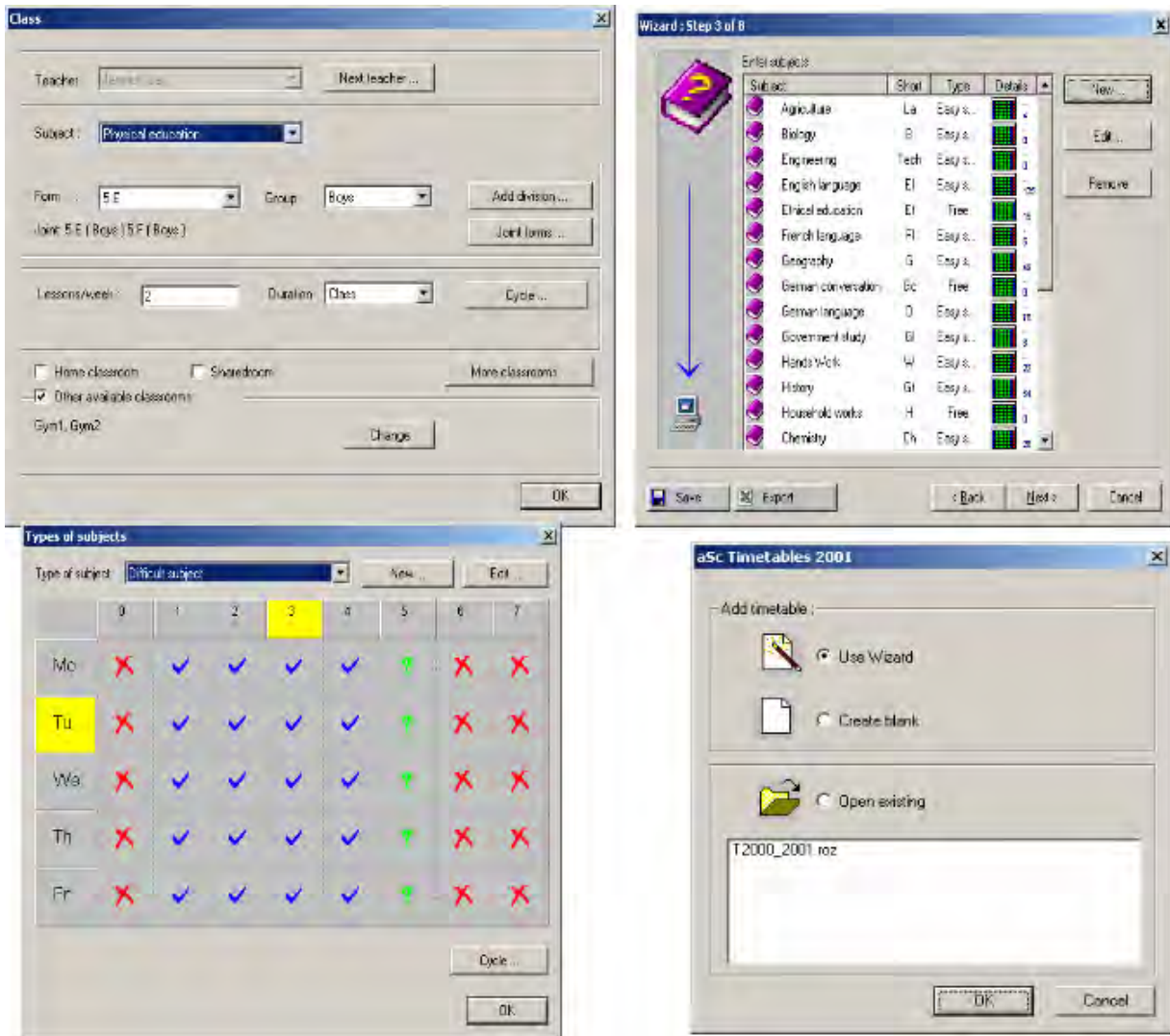
### 5.1 aSc TimeTables

Πρόκειται για ένα κλειστό λογισμικό που είναι διαθέσιμο μονάχα σε Windows. Προσφέρεται υπό τη μορφή δοκιμής για ένα χρονικό διάστημα, στη συνέχεια όμως απαιτείται να αγοραστεί η πλήρης έκδοση.

Το aSc TimeTables είναι ένα πρόγραμμα δημιουργίας ωρολογίων προγραμμάτων που υποστηρίζει μαθήματα, διδασκαλίες, αίθουσες, και δασκάλους με διαφορετικά συμβόλαια. Μετά την εισαγωγή δεδομένων, που γίνεται με αντίστοιχο τρόπο με αυτόν στην εφαρμογή FET, το aSc TimeTables παράγει προγράμματα με βάσει τα παρακάτω κριτήρια:

- Την ελαχιστοποίηση των κενών στα προγράμματα των εκπαιδευτικών: τον περιορισμό των κενών μέσα σε μια μέρα καθώς και τον περιορισμό των ημερών μέσα στη βδομάδα στις οποίες εκπαιδευτικοί έχουν διδασκαλία.
- Οι διδασκαλίες ενός μαθήματος πρέπει να διανεμηθούν ομαλά σε ολόκληρη την εβδομάδα.
- Τοποθέτηση διδασκαλιών σε αίθουσες διδασκαλίας που επιτρέπεται να γίνουν.

Πέρα από τα παραπάνω κριτήρια, ο χρήστης μπορεί να ορίσει εξαιρέσεις και προδιαγραφές σε κάποιες διδασκαλίες. Η εισαγωγή των δεδομένων μπορεί να γίνει με τη βοήθεια ενός βοηθού (wizard). Τα ωρολόγια προγράμματα μπορούν να εξαχθούν σε μορφή HTML είτε σε excel για περαιτέρω επεξεργασία.

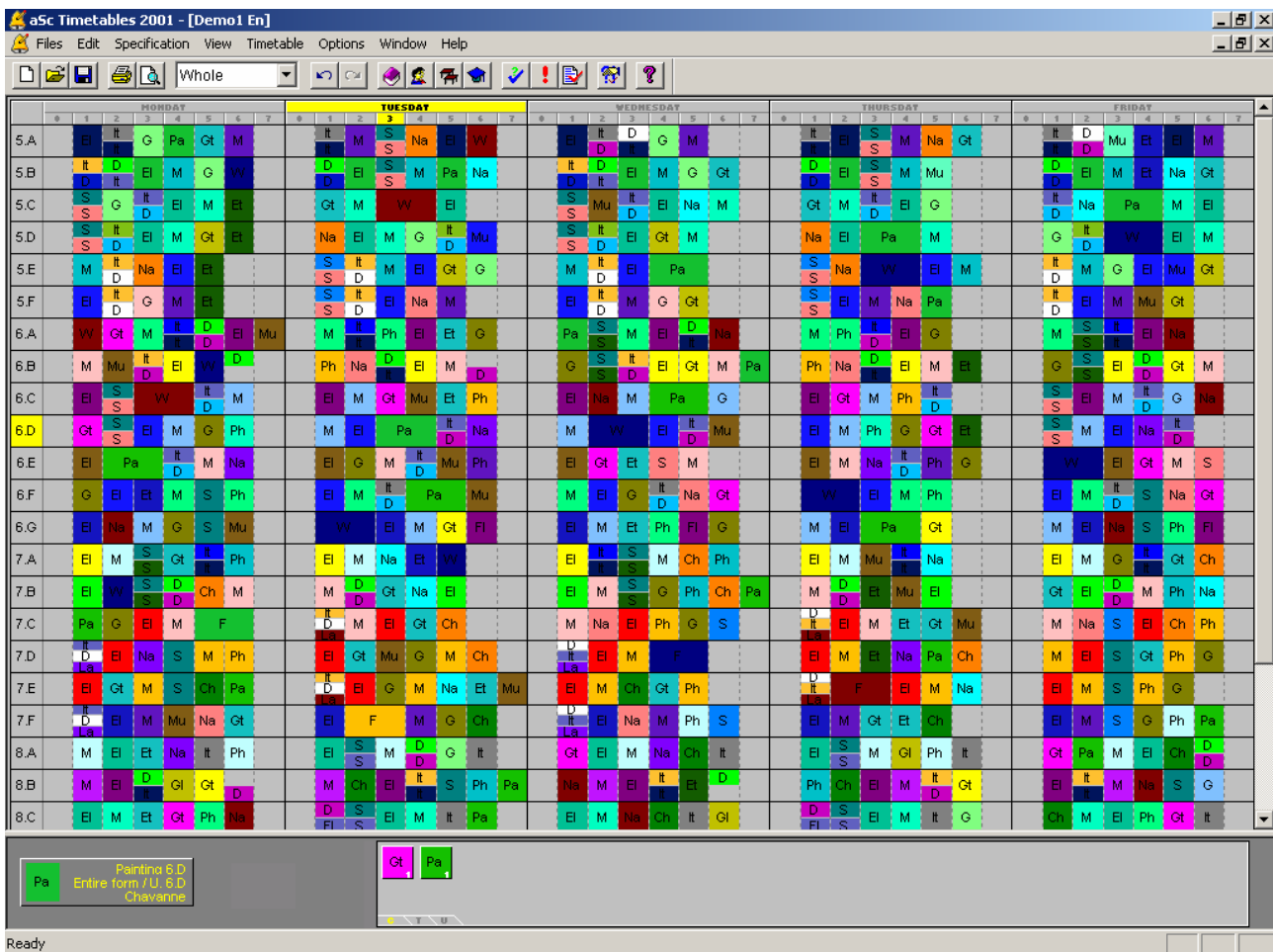


Εικόνα 5.1: Στιγμιότυπα λειτουργίας του wizard

Τα πλεονεκτήματά του σε σχέση με το FET είναι ότι υποστηρίζει συμβόλαια ανά καθηγητή, δημιουργεί προγράμματα που είναι περισσότερο φιλικά στους καθηγητές λόγω των κριτηρίων που προαναφέρθηκαν και έχει και ένα βοηθό για την εισαγωγή δεδομένων, είναι δηλαδή περισσότερο φιλικό στο χρήστη.

Τα μειονεκτήματά του είναι ότι πρόκειται για εμπορικό λογισμικό, δεν υποστηρίζει την ελληνική γλώσσα, δεν υποστηρίζει τη χρήση περισσότερων κτιρίων, είναι διαθέσιμο μονάχα σε windows και δεν έχει τη μεγάλη λίστα από περιορισμούς που είναι διαθέσιμοι στο FET.





Εικόνα 5.2: Ωρολόγιο πρόγραμμα στο aSc Timetables

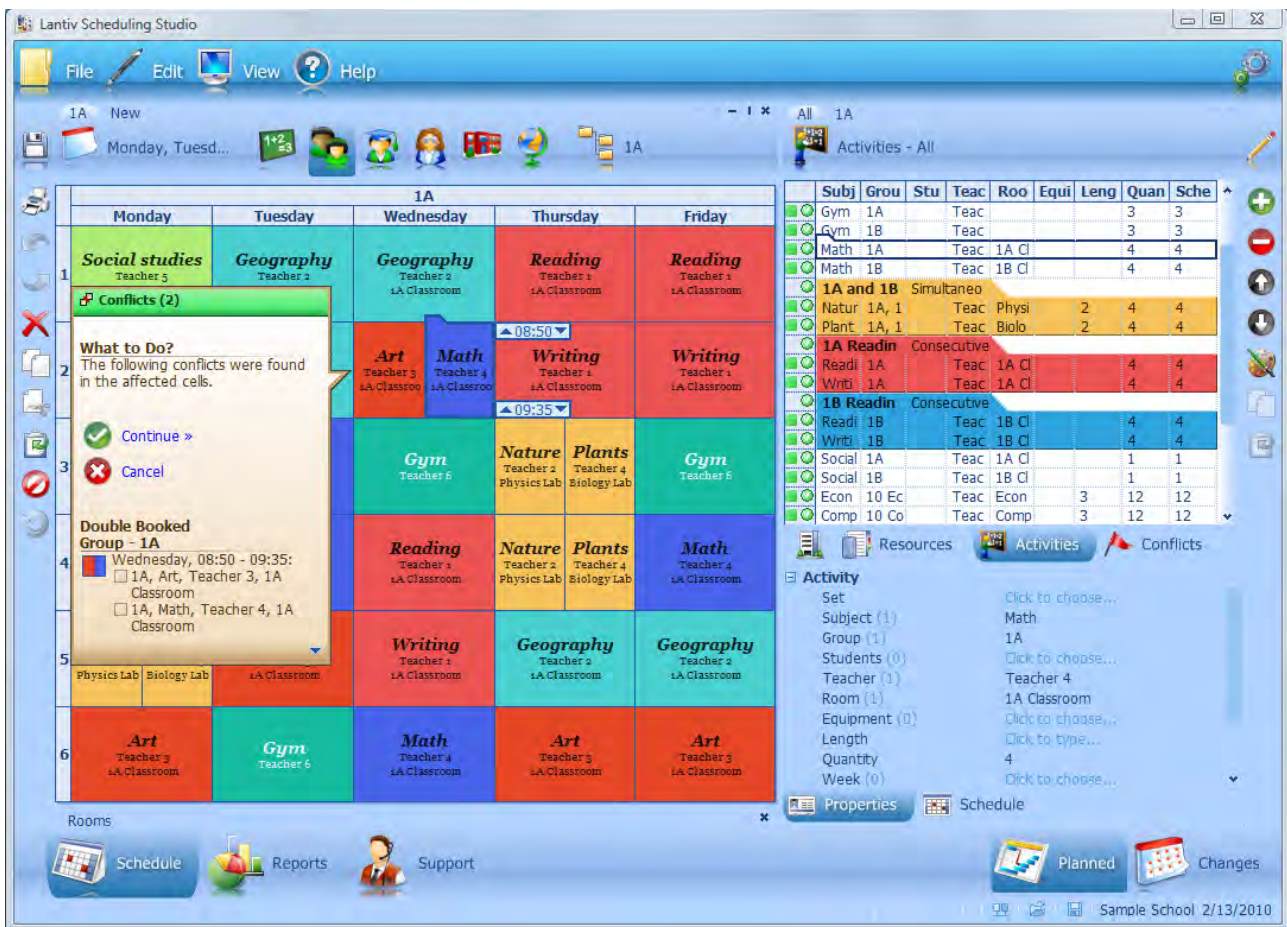
## 5.2 Lantiv TimeTabler

Πρόκειται για ένα κλειστό λογισμικό που είναι διαθέσιμο μονάχα σε Windows. Προσφέρεται υπό τη μορφή δοκιμής για ένα χρονικό διάστημα, στη συνέχεια όμως απαιτείται να αγοραστεί η πλήρης έκδοση.

Το Lantiv TimeTabler είναι ένα πρόγραμμα δημιουργίας ωρολογίων προγραμμάτων για διάφορες εφαρμογές (σχολεία, Πανεπιστήμια, Νοσοκομεία, ιδιωτικές εταιρίες). Επειδή στόχος του είναι η λύση του προβλήματος χρονοπρογραμματισμού σε διάφορους τομείς, έχει τη δυνατότητα μετατροπής του λεξιλογίου που χρησιμοποιείται από το χρήστη με βάση την εφαρμογή που τον ενδιαφέρει. Για παράδειγμα, σε ένα σχολείο, ένας επιβλέπων θα έχει το όνομα "καθηγητής", ενώ στο νοσοκομείο μπορεί να είναι «γιατρός» ή «νοσοκόμα».

Τα πλεονεκτήματά του σε σχέση με το FET είναι ότι έχει και διαδικτυακή έκδοση, δηλαδή ο χειρισμός του μπορεί να γίνει από οπουδήποτε μέσω ενός web browser, ενώ τα δεδομένα και τα προγράμματα αποθηκεύονται σε βάση δεδομένων του εξυπηρετητή.

Τα μειονεκτήματά του είναι ότι πρόκειται για εμπορικό λογισμικό, δεν υποστηρίζει την ελληνική γλώσσα, ενώ δεν έχει τόσο μεγάλη εξειδίκευση (πληθώρα περιορισμών) στο πρόβλημα της δημιουργίας ωρολογίων προγραμμάτων για σχολεία.



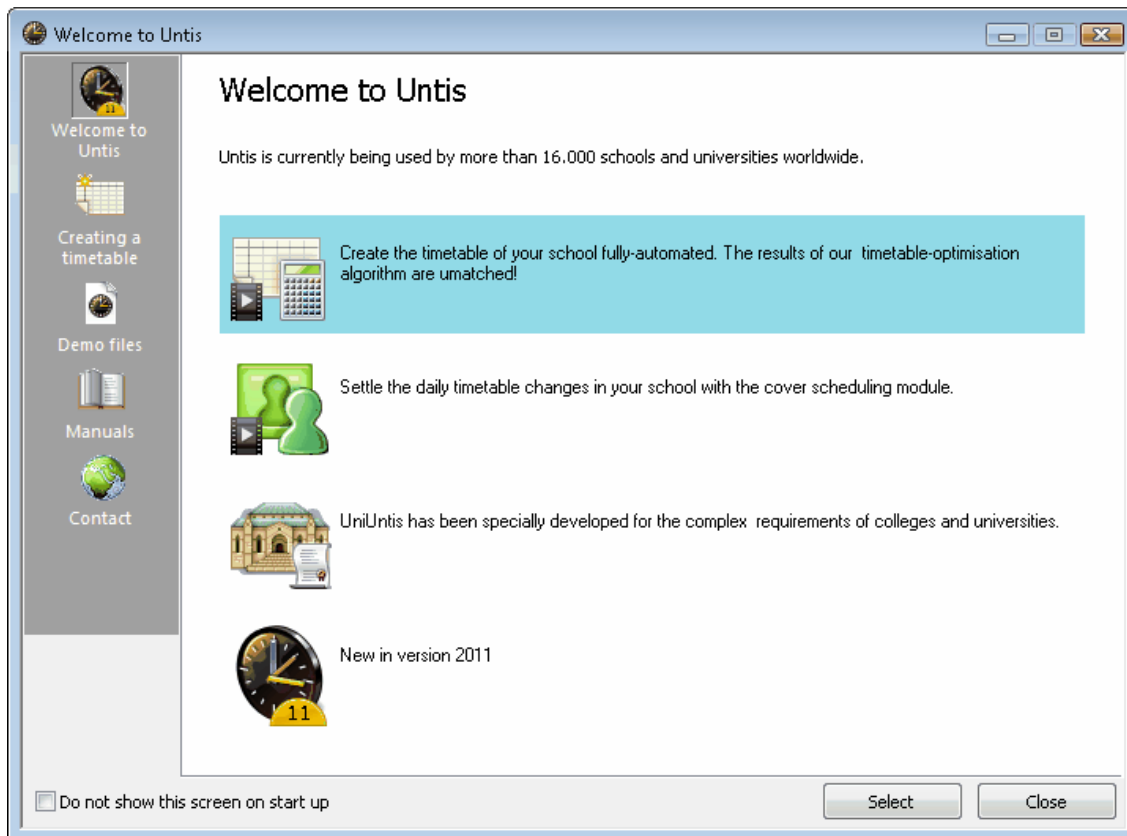
Εικόνα 5.3: Ωρολόγιο πρόγραμμα στο Lantiv TimeTabler

## 5.3 Untis

Πρόκειται για ένα κλειστό λογισμικό όπου υπάρχει και η δωρεάν δοκιμή για ένα χρονικό διάστημα. Το Untis είναι μία εφαρμογή που δημιουργεί ωρολόγια προγράμματα για κάθε τύπο σχολείου. Είναι μεταφρασμένο σε περισσότερες από 20 γλώσσες και κατέχει 30 χρόνια εμπειρίας σε αυτόν τον τομέα. Επίσης βασίζεται σε ένα ισχυρό αλγόριθμο ο οποίος υλοποιεί ένα ωρολόγιο πρόγραμμα σε σύντομο χρονικό διάστημα.

Τα πλεονεκτήματά του σε σχέση με το FET είναι ότι έχει και διαδικτυακή έκδοση, δηλαδή ο χειρισμός του μπορεί να γίνει από οπουδήποτε μέσω ενός web browser και μπορεί να χειριστεί κατάλληλα χρονοδιαγράμματα φοιτητών ή μαθητών. Τα μειονεκτήματά του είναι ότι πρόκειται για εμπορικό λογισμικό και δεν υποστηρίζει την ελληνική γλώσσα.

Παρακάτω παρουσιάζονται στιγμιότυπα από το Untis όπου φαίνεται πόσο φιλικό είναι προς τον χρήστη (εικόνες 5.4 και 5.5).



Εικόνα 5.4: Μενού του Untis

Untis 2011 Timetable 2011/2012 Test school DEMO 2 - Mozilla Firefox


file:///C:/Users/admin/Desktop/Cla1\_1a.htm

Untis 2011 Timetable 2011/2012 Tes...

### 1a Class 1a (Gauss)

	Mo	Tu	We	Th	Fr
1	<b>Geography and Econom.</b> Hugo R1a	<b>Mathematics</b> Arist R1a	<b>English</b> Arist R1a	<b>Mathematics</b> Arist R1a	<b>Mathematics</b> Arist R1a
2	<b>German</b> Rub R1a	<b>English</b> Arist R1a	<b>Girls PE.</b> Arist SH2	<b>German</b> Rub R1a	<b>Religious Education</b> Nobel R1a
3	<b>Biology</b> Cer R1a	<b>Art</b> Calla R1a	<b>Music</b> Calla R1a	<b>English</b> Arist R1a	<b>English</b> Arist R1a
4	<b>Girls PE.</b> Arist SH2		<b>German</b> Rub R1a		<b>German</b> Rub R1a
5	<b>Mathematics</b> Arist R1a	<b>German</b> Rub R1a	<b>Mathematics</b> Arist	<b>Religious Education</b> Nobel R1a	
6					
7				<b>Biology</b> Cer R1a	
8		<b>Girls PE.</b> Arist SH2		<b>Music</b> Calla R1a	
9		<b>Design.</b> Ander WS		<b>Geography and Econom.</b> Hugo R1a	
10					
11					

Gruber&PettersSoftware


Untis 2011
Gruber & Petters

Εικόνα 5.5: Ωρολόγιο πρόγραμμα στο Untis

**ΜΕΡΟΣ IV**  
**Ωρολόγια Διαδικτυακή Εφαρμογή**

# ΚΕΦΑΛΑΙΟ 6

## Ωρολόγια Διαδικτυακή Εφαρμογή

Υπάρχουν δύο πράγματα που στοχεύουμε στη ζωή μας:  
πρώτον ν'αποκτήσουμε αυτά που θέλουμε  
και δεύτερον να τ'απολαύσουμε.  
Μόνο οι σοφότεροι του ανθρώπινου  
είδους επιτυγχάνουν το δεύτερο.

~Logan Pearsall Smith (1865-1946)

### 6.1 Σκοπός της δημιουργίας Ωρολόγιας Διαδικτυακής Εφαρμογής

Σε προηγούμενα κεφάλαια παρουσιάσαμε την εφαρμογή FET και τις λειτουργίες της και μάθαμε πως για να δημιουργηθεί ένα ωρολόγιο πρόγραμμα χρειάζεται πρώτα να καταχωρηθούν πληροφορίες σχετικά με το τμήμα, το πρόγραμμα σπουδών και τις διαθεσιμότητες των διδασκόντων και των αιθουσών. Στη διαδικασία της επεξεργασίας των παραπάνω δεδομένων παρατηρήθηκε πως απαιτείται αρκετός χρόνος για τη συλλογή πληροφοριών σχετικά με τις διαθεσιμότητες των διδασκόντων καθώς οι διδάσκοντες πρέπει να επικοινωνούν κάθε φορά με τον υπεύθυνο για τα ωρολόγια προγράμματα και να δηλώνουν πότε είναι διαθέσιμοι να διδάσκουν το μάθημα τους. Έτσι λοιπόν, δημιουργήθηκε η ανάγκη μίας διαδικτυακής εφαρμογής στην οποία θα μπαίνουν οι διδάσκοντες με ένα όνομα και ένα κωδικό και θα καταχωρούν ποια χρονικά διαστήματα είναι διαθέσιμοι μέσα στην εβδομάδα. Τα δεδομένα αυτά θα στέλνονται σε μία βάση δεδομένων και κατόπιν θα δημιουργείται ένα αρχείο το οποίο θα τρέχει στην εφαρμογή FET. Με αυτόν τον τρόπο αποτρέπεται η σύγχυση και η καθυστέρηση και δημιουργείται εγκαίρως το απαιτούμενο ωρολόγιο πρόγραμμα. Στην επόμενη ενότητα περιγράφουμε την Ωρολόγια Διαδικτυακή Εφαρμογή.

### 6.2 Περιγραφή της Ωρολόγιας Διαδικτυακής Εφαρμογής

Για την υλοποίηση της εφαρμογής αυτής χρειάστηκε πρώτα να εγκαταστήσουμε τον Apache Server, την PHP και την MySQL. Για την ακρίβεια επιλέξαμε να εγκαταστήσουμε την εφαρμογή XAMPP η οποία περιέχει όλα τα παραπάνω. Με τον Apache Server μπορούμε να εκτελέσουμε τον κώδικα php και έτσι να παράγουμε σε πραγματικό χρόνο το τελικό περιεχόμενο το οποίο στέλνεται σε ένα πρόγραμμα περιήγησης web (web browser) σε μορφή κώδικα html. Με αυτόν τον τρόπο βελτιώσαμε την ωρολόγια διαδικτυακή εφαρμογή και φέραμε εις πέρας τις απαιτήσεις της.

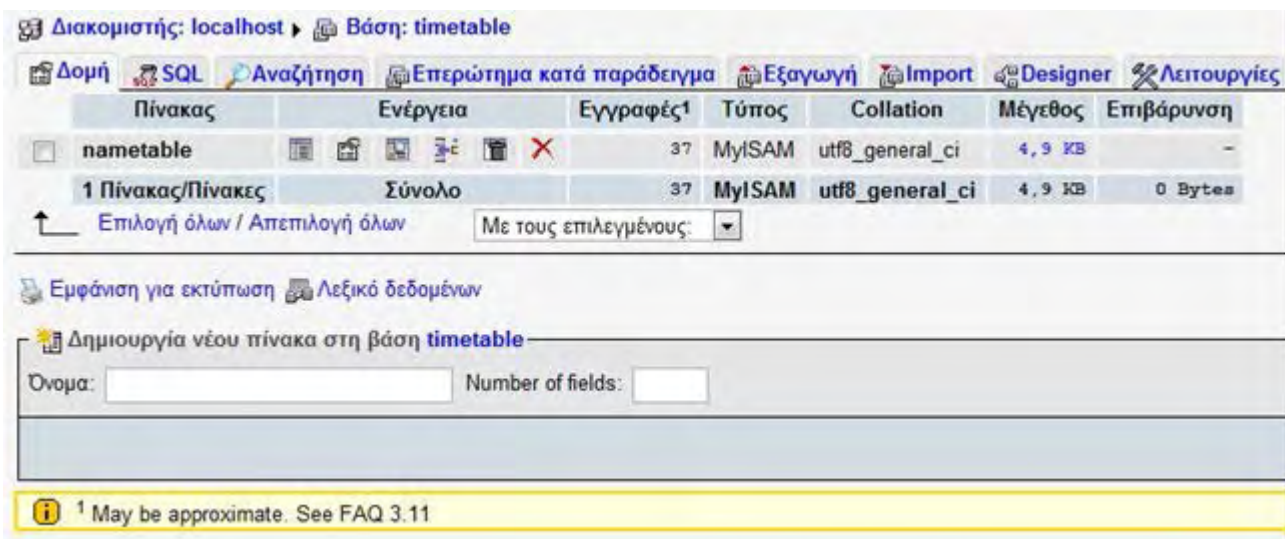
Μία από τις απαιτήσεις της ήταν η σύνδεση της με μία βάση δεδομένων στην οποία θα καταχωρούνται και θα αντλούνται πληροφορίες. Για αυτόν τον λόγο ήταν σημαντική η χρήση της MySQL καθώς με αυτήν μπορούσαμε να δημιουργήσουμε μία βάση δεδομένων που να πληρεί τις προϋποθέσεις της. Για τη διαχείριση της MySQL χρησιμοποιήσαμε το πρόγραμμα phpMyAdmin του XAMPP με το οποίο μπορούμε να εκτελούμε γρήγορα και εύκολα εντολές sql. Για την είσοδο μας στο phpMyAdmin χρειάζεται να ανοίξουμε ένα web browser και να γράψουμε την παρακάτω διεύθυνση: **127.0.0.1/phpmyadmin/**.

Σε αυτό το πρόγραμμα (εικόνα 6.1) δημιουργήσαμε μία καινούργια βάση δεδομένων με το όνομα timetable και έναν πίνακα με το όνομα nametable (εικόνα 6.2). Ο πίνακας nametable αποτελείται από πεδία όπως το όνομα του διδάσκοντα (lastname), το όνομα χρήστη (username), ο κωδικός (password), οι διαθεσιμότητες για τα πρωινά της Δευτέρας (monday\_a), της Τρίτης (tuesday\_a), της Τετάρτης (wednesday\_a), της Πέμπτης (thursday\_a) και της Παρασκευής (friday\_a) και οι διαθεσιμότητες για τα απογεύματα της Δευτέρας (monday\_p), της Τρίτης (tuesday\_p), της Τετάρτης (wednesday\_p), της Πέμπτης (thursday\_p) και της Παρασκευής (friday\_p) (εικόνα 6.3).

Οι διαθεσιμότητες των χρονικών αυτών διαστημάτων είναι πεδία τύπου μικρού ακέραιου αριθμού (tinyint) και παίρνουν τιμές 0 ή 1 ανάλογα με το εάν ο διδάσκοντας είναι διαθέσιμος να διδάσκει ή όχι. Δηλαδή αν η διαθεσιμότητα για το πρωί της Τρίτης έχει την τιμή 1 σημαίνει πως ο διδάσκοντας έχει την ευκαιρία να διδάσκει σε αυτό το χρονικό διάστημα ενώ αν έχει την τιμή 0 δεν μπορεί να διδάσκει. Επιπλέον, το επώνυμο του διδάσκοντα, το όνομα χρήστη και ο κωδικός είναι είναι πεδία τύπου χαρακτήρα (varchar) και έχουν κωδικοποίηση utf8\_general\_ci.



Εικόνα 6.1 phpMyAdmin



Εικόνα 6.2 Βάση Δεδομένων timetable και πίνακας nametable

Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/> id	int(11)			Όχι	Κανένα	auto_increment	[Icons]
<input type="checkbox"/> lastname	varchar(30)	utf8_general_ci		Όχι	Κανένα		[Icons]
<input type="checkbox"/> monday_a	tinyint(4)			Όχι	Κανένα		[Icons]
<input type="checkbox"/> monday_p	tinyint(4)			Όχι	Κανένα		[Icons]
<input type="checkbox"/> tuesday_a	tinyint(4)			Όχι	Κανένα		[Icons]
<input type="checkbox"/> tuesday_p	tinyint(4)			Όχι	Κανένα		[Icons]
<input type="checkbox"/> wednesday_a	tinyint(4)			Όχι	Κανένα		[Icons]
<input type="checkbox"/> wednesday_p	tinyint(4)			Όχι	Κανένα		[Icons]
<input type="checkbox"/> thursday_a	tinyint(4)			Όχι	Κανένα		[Icons]
<input type="checkbox"/> thursday_p	tinyint(4)			Όχι	Κανένα		[Icons]
<input type="checkbox"/> friday_a	tinyint(4)			Όχι	Κανένα		[Icons]
<input type="checkbox"/> friday_p	tinyint(4)			Όχι	Κανένα		[Icons]
<input type="checkbox"/> username	varchar(30)	utf8_general_ci		Όχι	Κανένα		[Icons]
<input type="checkbox"/> password	varchar(8)	utf8_general_ci		Όχι	Κανένα		[Icons]

Εικόνα 6.3 Πεδία του πίνακα nametable

## 6.2.1 Εισαγωγή νέων εγγραφών στη βάση δεδομένων

Πριν προχωρήσουμε στην περιγραφή των βασικών κωδικών php της ωρολόγιας διαδικτυακής εφαρμογής θα αναφέρουμε 2 αρχεία που δημιουργήσαμε τα οποία βοηθούν τον διαχειριστή να εισάγει εγγραφές στη βάση δεδομένων με μία φόρμα συμπλήρωσης στοιχείων των διδασκόντων. Το πρώτο αρχείο (index.php) παρουσιάζει τη φόρμα συμπλήρωσης που με την υποβολή της εκτελείται το δεύτερο αρχείο (insert.php). Το δεύτερο αρχείο περιέχει κώδικα php και στέλνει τα δεδομένα της φόρμας στη βάση δεδομένων και εμφανίζει στον browser μία λίστα με τα επίθετα των διδασκόντων και τα αντίστοιχα ονόματα χρηστών.

### Επεξήγηση του πρώτου αρχείου (index.php)

Ο κώδικας του αρχείου index.php (εικόνα 6.4) όπως αναφέραμε παραπάνω έχει γραφεί σε html και εμφανίζει στο browser μία φόρμα συμπλήρωσης (εικόνα 6.5).

- Οι γραμμές 11 και 31 είναι ετικέτες που δημιουργούν τη φόρμα. Η φράση action="insert.php" είναι μία ιδιότητα που στέλνει τα δεδομένα της φόρμας στο αρχείο insert.php ενώ η φράση method="post" είναι μία ιδιότητα που δείχνει ότι θα στείλει τα δεδομένα σε αυτό το αρχείο με τη μέθοδο post.
- Οι γραμμές 12-29 δημιουργούν ένα πίνακα που περιέχει τα στοιχεία της φόρμας.
- Και οι ετικέτες <input .../> δημιουργούν κενά πεδία για την εισαγωγή δεδομένων.



```

1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2  "http://www.w3.org/TR/html4/loose.dtd">
3
4  <html>
5  <head>
6  <meta http-equiv="content-type" content="text/html; charset=UTF8" >
7  <TITLE>ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ</TITLE>
8  </head>
9  <body>
10 <div id="container" align="center">ΦΟΡΜΑ ΣΥΜΠΛΗΡΩΣΗΣ<br>
11 <form action="insert.php" method="post"><br>
12 <table width="500" align="center">
13 <tr>
14 <td align="right"><b>ID: </b></td>
15 <td><input type="text" name="id" /></td>
16 </tr>
17 <tr>
18 <td align="right"><b>Επίθετο: </b></td>
19 <td><input type="text" name="lastname" /></td>
20 </tr>
21 <tr>
22 <td align="right"><b>Όνομα Χρήστη: </b></td>
23 <td><input type="text" name="username" /></td>
24 </tr>
25 <tr>
26 <td align="right"><b>Κωδικός: </b></td>
27 <td><input type="text" name="password" /></td>
28 </tr>
29 </table><br>
30 <p align="middle"><input type="submit" /></p>
31 </form>
32 </div>
33 </body>
34 </html>

```

Εικόνα 6.4 index.php

ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ

127.0.0.1/index.php

ΦΟΡΜΑ ΣΥΜΠΛΗΡΩΣΗΣ ΣΤΟΙΧΕΙΩΝ

ID:

Επίθετο:

Όνομα Χρήστη:

Κωδικός:

Εικόνα 6.5 ΦΟΡΜΑ ΣΥΜΠΛΗΡΩΣΗΣ ΣΤΟΙΧΕΙΩΝ

## Επεξήγηση του δεύτερου αρχείου (insert.php)

Ο κώδικας του αρχείου insert.php (εικόνα 6.7) όπως αναφέραμε παραπάνω έχει γραφεί σε γλώσσα php και html και πραγματοποιεί σε πρώτη φάση την αποστολή των δεδομένων της φόρμας στη βάση δεδομένων και κατόπιν εμφανίζει στον browser όλα τα επίθετα και τα ονόματα χρηστών των διδασκόντων (εικόνα 6.8).

- Η γραμμή 5 είναι μία δήλωση php (include “config.php”;) η οποία εκτελεί το αρχείο config.php (εικόνα 6.6). Αυτό χρειάζεται για την σύνδεση με τη βάση δεδομένων.
- Οι γραμμές 6 (<?php...) και 35 (?>) δηλώνουν την αρχή και το τέλος του κώδικα php.
- Οι γραμμές 7-16 αρχικοποιούν μεταβλητές που αφορούν τις διαθεσιμότητες των διδασκόντων με τιμή 1 με στόχο ο κάθε διδάσκοντας που εγγράφεται να εμφανίζει πως είναι διαθέσιμος όλη την εβδομάδα.
- Οι γραμμές 18-23 είναι μία εντολή sql (INSERT INTO...VALUES) η οποία προσθέτει νέες εγγραφές στη βάση δεδομένων. Για την ακρίβεια εισάγει δεδομένα όπως αριθμός καταχώρησης (id), επίθετο (lastname), διαθεσιμότητες, όνομα χρήστη (username), κωδικός (password).
- Η γραμμή 29 (mysql\_query()) εκτελεί την εντολή sql (SELECT \* FROM nametable) η οποία επιλέγει όλες τις στήλες από τον πίνακα nametable. Αυτό πραγματοποιείται επειδή θέλουμε να ανακτήσουμε δεδομένα από τη βάση δεδομένων μας.
- Οι γραμμές 30-33 (while) είναι μία επαναληπτική δομή με την οποία όσο ισχύει η συνθήκη (\$row=mysql\_fetch\_array(\$result)) τυπώνει στην οθόνη τον αριθμό id, το επίθετο και το όνομα χρήστη του διδάσκοντα. Η συνάρτηση mysql\_fetch\_array() επιστρέφει μία σειρά από ένα σύνολο εγγραφών. Με αυτόν τον τρόπο εμφανίζεται στο browser μία λίστα με όλες τις εγγραφές της βάσης δεδομένων.

## Επεξήγηση του αρχείου config.php

Ο κώδικας του αρχείου config.php (εικόνα 6.6) είναι php κώδικας που χρησιμοποιείται από άλλα αρχεία για την σύνδεση τους με μία βάση δεδομένων.

- Η γραμμή 2 είναι η δήλωση μίας μεταβλητής που αποθηκεύει το όνομα του διακομιστή που στην περίπτωση μας είναι η τοπική υποδοχή (localhost).
- Η γραμμή 3 είναι η δήλωση μίας μεταβλητής που αποθηκεύει το όνομα του διαχειριστή.
- Η γραμμή 4 είναι η δήλωση μίας μεταβλητής που αποθηκεύει το κωδικό του διαχειριστή.
- Η γραμμή 5 είναι η δήλωση μίας μεταβλητής που αποθηκεύει το όνομα της βάσης δεδομένων.
- Η γραμμή 6 είναι μία συνάρτηση που δημιουργεί μία σύνδεση με τη MySQL.
- Η γραμμή 7 είναι μία συνάρτηση που επιλέγει μία βάση δεδομένων.

```
1  <?
2  $dbhost = "localhost";
3  $dbuser = "eleni";
4  $dbpass = "password";
5  $dbname = "timetable";
6  $db = mysql_connect($dbhost,$dbuser,$dbpass) or die('wrong');
7  mysql_select_db($dbname,$db);
8  ?>
```

Εικόνα 6.6 config.php

```

1 <html>
2 <head>
3   <title>ΕΓΓΡΑΦΕΣ ΤΗΣ ΒΔ</title>
4 </head>
5 <? include "config.php";?>
6 <?php
7   $_POST[monday_a]=1;
8   $_POST[monday_p]=1;
9   $_POST[tuesday_a]=1;
10  $_POST[tuesday_p]=1;
11  $_POST[wednesday_a]=1;
12  $_POST[wednesday_p]=1;
13  $_POST[thursday_a]=1;
14  $_POST[thursday_p]=1;
15  $_POST[friday_a]=1;
16  $_POST[friday_p]=1;
17
18  $sql="INSERT INTO nametable (id, lastname, monday_a, monday_p,
19  tuesday_a, tuesday_p, wednesday_a, wednesday_p, thursday_a, thursday_p,
20  friday_a, friday_p, username, password) VALUES ('$_POST[id]', '$_POST[lastname]',
21  '$_POST[monday_a]', '$_POST[monday_p]', '$_POST[tuesday_a]', '$_POST[tuesday_p]',
22  '$_POST[wednesday_a]', '$_POST[wednesday_p]', '$_POST[thursday_a]', '$_POST[thursday_p]',
23  '$_POST[friday_a]', '$_POST[friday_p]', '$_POST[username]', '$_POST[password]')";
24
25  if (!mysql_query($sql,$db))
26  {
27    die('Error: ' . mysql_error());
28  }
29  $result = mysql_query("SELECT * FROM nametable");
30  while($row = mysql_fetch_array($result))
31  {
32    print($row['id'] . " ::: " . $row['lastname'] . " --> " . $row['username'] . "<br>");
33  }
34  mysql_close($db);
35  ?>
36 </html>

```

Εικόνα 6.7 insert.php

ΕΓΓΡΑΦΕΣ ΤΗΣ ΒΔ

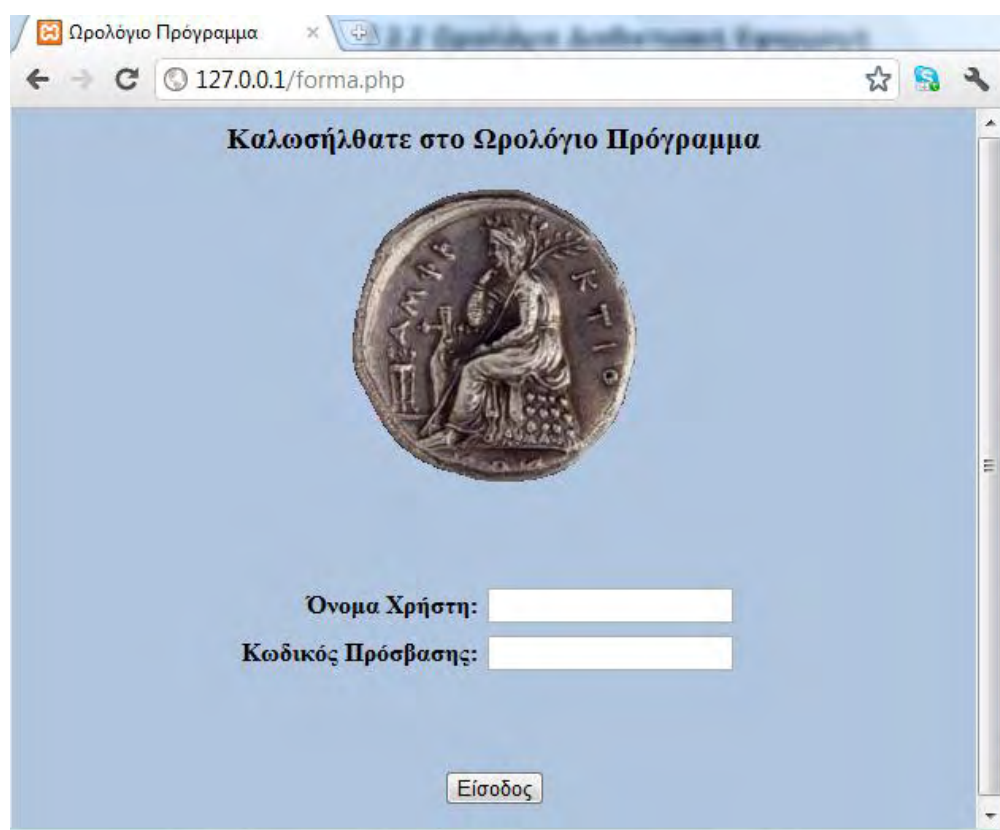
127.0.0.1/insert.php

1:::Αδάμ-->Adam  
2:::Αλετράς-->Aletras  
3:::Αναγνωστόπουλος-->Anagnostopoulos  
4:::Αποστολόπουλος-->Apostolopoulos  
5:::Ασημάκης-->Ashmakhs  
6:::Βάβουλας-->Baboulas  
7:::Βασιλάκη-->Basilakh  
8:::Βασιλακόπουλος-->Basilakopoulos  
9:::Βαϊόπουλος-->Baïopoulos  
10:::Γεωργίου-->Georgiou  
11:::Γεωργακάλου-->Georgakalou  
12:::Γκανέτσος-->Gkanetsos  
13:::Γούγα-->Gouga  
14:::Δελήμπασης-->Delhmpashs  
15:::Ιακωβίδης-->Iakwbidhs  
16:::Κακαρούντας-->Kakarountas  
17:::Καραγεώργος-->Karagewrgos  
18:::Καρυδάκης-->Karudakhs

Εικόνα 6.8 Εγγραφές της ΒΔ

## 6.2.2 Ωρολόγια Διαδικτυακή Εφαρμογή

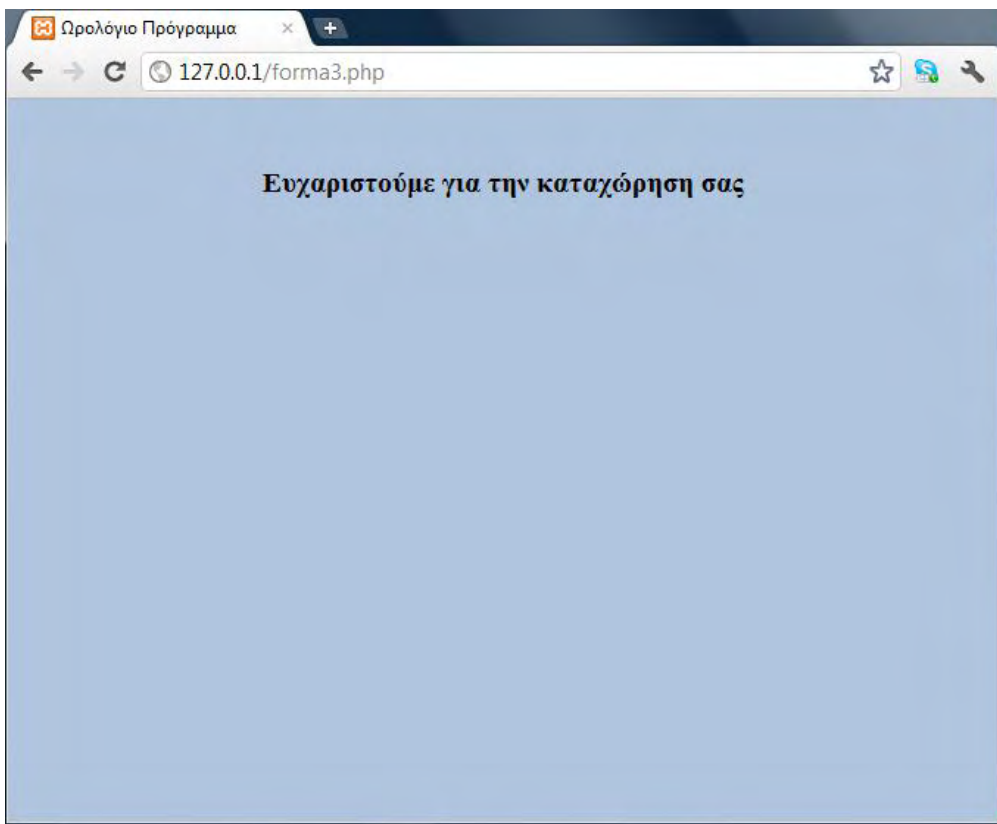
Για να υλοποιηθεί η ωρολόγια διαδικτυακή εφαρμογή χρειάστηκε να δημιουργηθούν 3 αρχεία php. Το πρώτο αρχείο (forma.php) παράγει μία σελίδα στον browser στην οποία ο διδάσκοντας πρέπει να εισάγει ένα όνομα χρήστη και έναν κωδικό για να συνδεθεί με την εφαρμογή (εικόνα 6.9). Το δεύτερο αρχείο (forma2.php) καλωσορίζει τον διδάσκοντα με το επώνυμο του, τον ρωτά πότε είναι διαθέσιμος και εμφανίζει έναν πίνακα με χρονικά διαστήματα (εικόνα 6.10). Έτσι ο διδάσκοντας μπορεί να επιλέξει τα χρονικά διαστήματα που είναι διαθέσιμος και να πατήσει αποστολή. Το τρίτο αρχείο (forma3.php) στέλνει τα δεδομένα που καταχώρησε ο διδάσκοντας στη βάση δεδομένων και εμφανίζει στο browser ένα ευχαριστήριο μήνυμα (6.11).



Εικόνα 6.9 Είσοδος στο Ωρολόγιο Πρόγραμμα



Εικόνα 6.10 Ωρολόγιο Πρόγραμμα



Εικόνα 6.11 Τέλος Ωρολογίου Προγράματος

## Επεξήγηση του αρχείου forma.php

- Οι γραμμές 18-32 δημιουργούν μία φόρμα η οποία περιέχει κενά συμπλήρωσης ενός ονόματος χρήστη και ενός κωδικού και ένα κουμπί υποβολής για την αποστολή αυτών των δεδομένων στο διακομιστή. Με την υποβολή των δεδομένων εκτελείται πάντα το αρχείο forma2.php και λαμβάνει τα δεδομένα με τη μέθοδο post.

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3
4 <html>
5 <head>
6 <style type="text/css">
7   body
8   {
9     background-color:#b0c4de;
10  }
11 </style>
12 <title>Ωρολόγιο Πρόγραμμα</title>
13 </head>
14
15 <body>
16 <p align="center"><h3 align="center"> Καλωσήλθατε στο Ωρολόγιο Πρόγραμμα</h3></p>
17 <p align="middle"></p>
18 <form action="forma2.php" method="post">
19 <br><br>
20 <table width="500" align="center">
21 <tr>
22 <td align="right"><b>Όνομα Χρήστη:</b></td>
23 <td><input type="text" name="username"></td>
24 </tr>
25 <tr>
26 <td align="right"><b>Κωδικός Πρόσβασης:</b></td>
27 <td><input type="password" name="password" ></td>
28 </tr>
29 </table>
30 <br><br>
31 <p align="middle"><input type="submit" value="Είσοδος" /></p>
32 </form>
33 </body>
34 </html>
```

Εικόνα 6.12 forma.php

## Επεξήγηση του αρχείου forma2.php

- Η γραμμές 4-9 ελέγχουν αν τα στοιχεία όνομα χρήστη και κωδικός που ελήφθησαν από την προηγούμενη φόρμα εισόδου (forma.php) υπάρχουν στη βάση δεδομένων. Αν υπάρχουν τότε η μεταβλητή `$_SESSION['view']` παίρνει την τιμή 1.
- Οι γραμμές 25-100 είναι κώδικας που αφορά τη φόρμα «διαθεσιμότητας» την οποία καλείται να συμπληρώσει ο διδάσκοντας.
- Η γραμμή 26 ορίζει μία παράγραφο η οποία θα βρίσκεται στο κέντρο της φόρμας και η γραμματοσειρά της θα έχει χρώμα μαύρο. Το περιεχόμενο της παραγράφου είναι: «Καλωσήλθατε “επίθετο διδάσκοντα”». Το επίθετο του διδάσκοντα τυπώνεται από την εντολή `echo $myrow[1]` όπου `$myrow[1]` η μεταβλητή που παίρνει τιμή από το πεδίο `lastname` του πίνακα `nametable`.
- Οι γραμμές 30-95 είναι κώδικας που αφορά τον πίνακα της φόρμας.
- Οι γραμμές 42-46 αφορούν τη δεύτερη στήλη και τη δεύτερη σειρά του πίνακα της φόρμας και ελέγχουν αν η διαθεσιμότητα του πρωινού της Δευτέρας (`monday_a`) από τη βάση

δεδομένων έχει την τιμή 1. Αν έχει την τιμή 1 τότε εμφανίζει ένα σημαδεμένο κουτάκι (checkbox), αν όχι τότε ένα κενό κουτάκι. Ίδια ερμηνεία έχουν οι γραμμές 47-93 που αφορούν τις υπόλοιπες διαθεσιμότητες της εβδομάδας.

```

1  |<? @session_start();?>
2  |<?php
3  |   include "config.php";
4  |   $query = "SELECT * FROM nametable";
5  |   $result = mysql_query($query, $db);
6  |   while($myrow=mysql_fetch_row($result)){
7  |       if($myrow[12]==$_POST[username]){
8  |           if($myrow[13]==$_POST[password]){
9  |               $_SESSION['view']=1;
10 |       }
11 |   }
12 |   <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
13 |   "http://www.w3.org/TR/html4/loose.dtd">
14 |   <html>
15 |   <head>
16 |   <style type="text/css">
17 |   <body
18 |   {
19 |       background-color:#b0c4de;
20 |   }
21 |   </style>
22 |   <meta content="text/html; charset=utf-8" />
23 |   <title>Προλόγιο Πρόγραμμα</title>
24 |   </head>
25 |   <body>
26 |   <form action="forma3.php" method="post" />
27 |   <p align="center"><font color="black"><h3 align="center">Καλωσήλθατε <? echo $myrow[1]; ?></h3></font></p>
28 |   <br><br><br><br>
29 |   <input type="hidden" value="<?echo $myrow[1];?>" name="lastname" />
30 |   <input type="hidden" value="<?echo $_SESSION['view'];?>" name="view" />
31 |   <table border="1" align="center" width="40%" height="20%" bgcolor="#7e9ec9" >
32 |   <caption><font color="black"><h3 align="center">Πότε είστε διαθέσιμος/η;</h3></font></caption>

```

Εικόνα 6.13 forma2.php (Μέρος 1)

```

32 |   <tr>
33 |       <td>         </td>
34 |       <td>Δευτέρα</td>
35 |       <td>Τρίτη</td>
36 |       <td>Τετάρτη</td>
37 |       <td>Πέμπτη</td>
38 |       <td>Παρασκευή</td>
39 |   </tr>
40 |   <tr>
41 |       <td>Πρωί</td>
42 |       <td>
43 |           <? if($myrow[2]==1){ ?>
44 |               <input type="checkbox" name="monday_a" checked="yes">
45 |           <? }else echo "<input type=\"checkbox\" name=\"monday_a\""; ?>
46 |       </td>
47 |       <td>
48 |           <? if($myrow[4]==1){ ?>
49 |               <input type="checkbox" name="tuesday_a" checked="yes">
50 |           <? }else echo "<input type=\"checkbox\" name=\"tuesday_a\""; ?>
51 |       </td>
52 |       <td>
53 |           <? if($myrow[6]==1){?>
54 |               <input type="checkbox" name="wednesday_a" checked="yes">
55 |           <? }else echo "<input type=\"checkbox\" name=\"wednesday_a\""; ?>
56 |       </td>
57 |       <td>
58 |           <? if($myrow[8]==1){?>
59 |               <input type="checkbox" name="thursday_a" checked="yes">
60 |           <? }else echo "<input type=\"checkbox\" name=\"thursday_a\""; ?>
61 |       </td>
62 |       <td>
63 |           <? if($myrow[10]==1){?>
64 |               <input type="checkbox" name="friday_a" checked="yes">
65 |           <? }else echo "<input type=\"checkbox\" name=\"friday_a\""; ?>
66 |       </td>
67 |   </tr>

```

Εικόνα 6.14 forma2.php (Μέρος 2)

```

68 <tr><td>Απόγευμα</td>
69 <td>
70 <? if($myrow[3]==1){?>
71 <input type="checkbox" name="monday_p" checked="yes">
72 <? }else echo "<input type=\"checkbox\" name=\"monday_p\">";?>
73 </td>
74 <td>
75 <? if($myrow[5]==1){?>
76 <input type="checkbox" name="tuesday_p" checked="yes">
77 <? }else echo "<input type=\"checkbox\" name=\"tuesday_p\">";?>
78 </td>
79 <td>
80 <? if($myrow[7]==1){?>
81 <input type="checkbox" name="wednesday_p" checked="yes">
82 <? }else echo "<input type=\"checkbox\" name=\"wednesday_p\">";?>
83 </td>
84 <td>
85 <? if($myrow[9]==1){?>
86 <input type="checkbox" name="thursday_p" checked="yes">
87 <? }else echo "<input type=\"checkbox\" name=\"thursday_p\">";?>
88 </td>
89 <td>
90 <? if($myrow[11]==1){?>
91 <input type="checkbox" name="friday_p" checked="yes">
92 <? }else echo "<input type=\"checkbox\" name=\"friday_p\">";?>
93 </td>
94 </tr>
95 </table><br><br>
96 <p align="center">Όπου πρωί είναι οι ώρες από τις 08:00 μέχρι τις 13:00</p>
97 <p align="center">Όπου απόγευμα είναι οι ώρες από τις 14:00 μέχρι τις 19:00</p><br>
98 <p align="center"><input type="submit" value="Αποστολή"></p>
99 </form>
100 </body>
101 </html>
102 <? }else echo "Λάθος Κωδικός Πρόσβασης"; }>
103 @session_destroy();
104 ?>

```

Εικόνα 6.15 forma2.php (Μέρος 3)

### Επεξήγηση του αρχείου forma3.php

- Οι γραμμές 20-49 ελέγχουν αν η φόρμα που στάλθηκε είχε τα κουτάκια σημαδεμένα ή όχι. Αν η μεταβλητή που αφορά ένα κουτάκι (π.χ. \$\_POST[monday\_a]) έχει τιμή on τότε το κουτάκι ήταν σημαδεμένο και η μεταβλητή παίρνει την τιμή 1, διαφορετικά παίρνει την τιμή 0.
- Οι γραμμές 50-55 δηλώνουν μία πρόταση sql η οποία ανανεώνει τα δεδομένα του πίνακα nametable που αφορούν τις διαθεσιμότητες των διδασκόντων.



```

1  <?
2  |   if($_POST[view]==1){
3  |       include "config.php";
4  |   }?>
5  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
6  |   "http://www.w3.org/TR/html4/loose.dtd">
7  <html>
8  <head>
9  <style type="text/css">
10 |   body
11 |   {
12 |       background-color:#b0c4de;
13 |   }
14 | </style>
15 | <meta content="text/html; charset=utf-8" />
16 | <title>Αρολόγιο Πρόγραμμα</title>
17 | </head>
18 <body>
19 <?php
20 |   if($_POST[monday_a]==on){
21 |       $_POST[monday_a]=1;
22 |   }else $_POST[monday_a]=0;
23 |   if($_POST[monday_p]==on){
24 |       $_POST[monday_p]=1;
25 |   }else $_POST[monday_p]=0;
26 |   if($_POST[tuesday_a]==on){
27 |       $_POST[tuesday_a]=1;
28 |   }else $_POST[tuesday_a]=0;
29 |   if($_POST[tuesday_p]==on){
30 |       $_POST[tuesday_p]=1;
31 |   }else $_POST[tuesday_p]=0;
32 |   if($_POST[wednesday_a]==on){
33 |       $_POST[wednesday_a]=1;
34 |   }else $_POST[wednesday_a]=0;
35 |   if($_POST[wednesday_p]==on){
36 |       $_POST[wednesday_p]=1;
37 |   }else $_POST[wednesday_p]=0;

```

Εικόνα 6.16 forma3.php (Μέρος 1)

```

38 |   if($_POST[thursday_a]==on){
39 |       $_POST[thursday_a]=1;
40 |   }else $_POST[thursday_a]=0;
41 |   if($_POST[thursday_p]==on){
42 |       $_POST[thursday_p]=1;
43 |   }else $_POST[thursday_p]=0;
44 |   if($_POST[friday_a]==on){
45 |       $_POST[friday_a]=1;
46 |   }else $_POST[friday_a]=0;
47 |   if($_POST[friday_p]==on){
48 |       $_POST[friday_p]=1;
49 |   }else $_POST[friday_p]=0;
50 |   $sql="UPDATE nametable SET monday_a='$_POST[monday_a]',
51 |   monday_p='$_POST[monday_p]', tuesday_a='$_POST[tuesday_a]',
52 |   tuesday_p='$_POST[tuesday_p]', wednesday_a='$_POST[wednesday_a]',
53 |   wednesday_p='$_POST[wednesday_p]', thursday_a='$_POST[thursday_a]',
54 |   thursday_p='$_POST[thursday_p]', friday_a='$_POST[friday_a]',
55 |   friday_p='$_POST[friday_p]' WHERE lastname='$_POST[lastname]' ";
56 |   $result=mysql_query($sql) or die(mysql_error());
57 |   echo "<br><p align=\"center\"><h3 align=\"center\">Ευχαριστούμε για την καταχώρηση σας </h3></p>";
58 |   ?>
59 | </body>
60 | </html>
61 <?
62 | }else {
63 |     echo "Something is wrong";
64 | }
65 | ?>

```

Εικόνα 6.17 forma3.php (Μέρος 2)

### 6.2.3 Παραγωγή αρχείου fet

Στην προηγούμενη υποενότητα περιγράψαμε την ωρολόγια διαδικτυακή εφαρμογή, που δεν είναι άλλο από μία δυναμική ιστοσελίδα που βοηθάει τους διδάσκοντες να δηλώνουν εύκολα και γρήγορα ποια χρονικά διαστήματα της εβδομάδας είναι διαθέσιμοι να διδάσκουν. Όμως με την κατασκευή μόνο αυτής της ιστοσελίδας δεν λύνεται το πρόβλημα της σύγχυσης καθώς τα δεδομένα που θα συγκεντρώνει κάθε φορά η βάση δεδομένων θα χρειάζονται επεξεργασία και μετατροπή σε χρήσιμη πληροφορία έτσι ώστε να χρησιμοποιούνται αργότερα από την εφαρμογή FET. Για τον λόγο αυτόν δημιουργήσαμε ένα ακόμη αρχείο php που συλλέγει τα κατάλληλα δεδομένα από τη βάση δεδομένων και παράγει ένα αρχείο xml ή αλλιώς ένα αρχείο εισόδου fet. Τα δεδομένα αυτά είναι το επώνυμο και οι διαθεσιμότητες του κάθε διδάσκοντα και χρησιμεύουν για την καταχώρηση των περιορισμών χρόνου.

Για την παραγωγή ενός αρχείου xml χρειάστηκε να εξετάσουμε τη δομή του xml αρχείου εισόδου του FET όπως δημιουργείται από την χειροκίνητη καταχώρηση. Βρήκαμε το κομμάτι αυτού του αρχείου που χρειάζεται αντικατάσταση με αυτήν των νέων πληροφοριών και το αφαιρέσαμε. Έτσι δημιουργήσαμε κώδικα php που παράγει αυτό το κομμάτι xml που αφαιρέθηκε με τη διαφορά ότι θα περιέχει τις απαιτούμενες πληροφορίες από τη βάση δεδομένων και θα ενώνεται με τα υπόλοιπα κομμάτια του αρχείου fet (που παραμένουν ίδια) σε ένα καινούργιο αρχείο fet. Στη συνέχεια θα τυπώνονται τα περιεχόμενα αυτού του αρχείου στην οθόνη του browser (εικόνα 6.22).

Τέλος, αφού παραχθεί το αρχείο fet, τότε μπορούμε να το χρησιμοποιήσουμε στο FET και να δημιουργήσουμε το απαιτούμενο ωρολόγιο πρόγραμμα.

#### Επεξήγηση του αρχείου generate.php

- Η γραμμή 8 του κώδικα είναι μία πρόταση sql που επιλέγει όλα τα πεδία του πίνακα nametable.
- Η γραμμή 9 είναι μία συνάρτηση που εκτελεί προτάσεις sql και στην περίπτωση μας εκτελεί την παραπάνω πρόταση.
- Η γραμμή 10 είναι μία συνάρτηση η οποία επιτρέπει την πρόσβαση και την τροποποίηση ενός αρχείου, στην περίπτωση μας που δεν υπάρχει το αρχείο gxml.txt τότε το δημιουργεί.
- Η γραμμή 11 είναι μία συνάρτηση η οποία ανοίγει το αρχείο gxml.txt και επιτρέπει την γραφή δεδομένων σε αυτό. Αν δεν ανοίξει το αρχείο τότε εμφανίζει μήνυμα λάθους.
- Οι γραμμές 12-104 περιλαμβάνουν τη δομή ελέγχου while η οποία εκτελείται όσο υπάρχουν εγγραφές στον πίνακα nametable της βάσης δεδομένων timetable.
- Οι γραμμές 13-14 αφορούν τη συνάρτηση fwrite() που γράφει στο αρχείο gxml.txt (\$fp) κώδικα xml.
- Η γραμμή 15 είναι η ίδια συνάρτηση με την παραπάνω και γράφει στο ίδιο αρχείο τον παρακάτω κώδικα xml: <Teacher> “Επώνυμο Διδάσκοντα” </Teacher>.
- Οι γραμμές 16-18 παίρνει τις τιμές όλων των πεδίων που αφορούν τις διαθεσιμότητες, τις προσθέτει και τις αποθηκεύει στην μεταβλητή \$sum.
- Η γραμμή 19 αποθηκεύει στην μεταβλητή \$value το αποτέλεσμα της αφαίρεσης (10-\$sum).
- Η γραμμή 20 αποθηκεύει στην μεταβλητή \$hours το αποτέλεσμα του πολλαπλασιασμού (\$value\*6).
- Δηλαδή οι γραμμές 16-20 υπολογίζουν πόσες ώρες δεν είναι διαθέσιμος ο διδάσκοντας (\$hours).
- Οι γραμμές 22-29 περιλαμβάνουν μία δομή ελέγχου if που ελέγχει αν η διαθεσιμότητα για το πρωί της Δευτέρας είναι 0 (μηδέν). Αν είναι μηδέν τότε γράφει στο αρχείο gxml.txt (\$fp) το κομμάτι xml που αντιστοιχεί για τις ώρες του πρωινού της Δευτέρας (23-28).

- Οι γραμμές 30-101 ελέγχουν με τη δομή ελέγχου if τις διαθεσιμότητες των υπόλοιπων χρονικών διαστημάτων όπως στις γραμμές 22-29 και γράφουν xml κώδικα στο αρχείο gxml.txt όπου είναι αναγκαίο.
- Η γραμμή 105 είναι μία συνάρτηση (fclose()) που κλείνει το αρχείο gxml.txt.
- Η γραμμή 106 δημιουργεί το αρχείο result.fet.
- Η γραμμή 107 ανοίγει (με την fopen()) το αρχείο result.fet για γράψιμο.
- Οι γραμμές 108-110 ανοίγουν τρία αρχεία (start.txt, gxml.txt, end.txt) για γράψιμο και ανάγνωση.
- Οι γραμμές 111-113 αποθηκεύουν τα παραπάνω τρία αρχεία σε κάποιες μεταβλητές.
- Οι γραμμές 114-117 περιλαμβάνουν τη δομή ελέγχου while η οποία εκτελείται εφόσον το αρχείο start.txt δεν έχει φτάσει στο τέλος του (EOF).
- Η γραμμή 115 διαβάζει το αρχείο start.txt με τη συνάρτηση fread() και το αποθηκεύει στην μεταβλητή \$text.
- Η γραμμή 116 γράφει (fwrite()) τα δεδομένα του αρχείου start.txt στο αρχείο result.fet.
- Η γραμμή 118 κλείνει (fclose()) το αρχείο start.txt.
- Οι γραμμές 119-128 εκτελούν κώδικα παρόμοιο με αυτό των γραμμών 114-118 και έτσι τα υπόλοιπα δύο αρχεία (gxml.txt, end.txt) προστίθενται στο αρχείο result.fet.
- Η γραμμή 131 τυπώνει το περιεχόμενο του αρχείου result.fet με τη βοήθεια δύο συναρτήσεων. Η file\_get\_contents() διαβάζει ένα αρχείο σε μία σειρά και η htmlspecialchars() μετατρέπει κάποιους χαρακτήρες (όπως "<", ">") σε html οντότητες.

```

1  | <? include "config.php"; ?>
2  | <html>
3  | <head>
4  | <meta http-equiv="content-type" content="text/html; charset=UTF8" >
5  | <TITLE>Αναπαραγωγή αρχείου</TITLE>
6  | </head>
7  | <?php
8  | $sql="SELECT * FROM nametable";
9  | $result=mysql_query($sql,$db) or die(mysql_error());
10 | touch("gxml.txt");
11 | $fp=fopen("gxml.txt", "w") or die("Couldn't open gxml.txt");
12 | while($newArray=mysql_fetch_array($result)){
13 |     fwrite($fp, "\n<ConstraintTeacherNotAvailableTimes>\n\n");
14 |     fwrite($fp, "<Weight_Percentage>100</Weight_Percentage>\n");
15 |     fwrite($fp, "<Teacher>". $newArray["lastname"]. "</Teacher>\n");
16 |     $sum=$newArray["monday_a"]+$newArray["monday_p"]+$newArray["tuesday_a"]
17 |     +$newArray["tuesday_p"]+$newArray["wednesday_a"]+$newArray["wednesday_p"]
18 |     +$newArray["thursday_a"]+$newArray["thursday_p"]+$newArray["friday_a"]+$newArray["friday_p"];
19 |     $value=10-$sum;
20 |     $hours=$value*6;
21 |     fwrite($fp, "<Number_of_Not_Available_Times>". $hours. "</Number_of_Not_Available_Times>\n");
22 |     if($newArray["monday_a"]==0){
23 |         fwrite($fp, "<Not_Available_Time><Day>Δευτέρα</Day><Hour>08:00</Hour></Not_Available_Time>\n");
24 |         fwrite($fp, "<Not_Available_Time><Day>Δευτέρα</Day><Hour>09:00</Hour></Not_Available_Time>\n");
25 |         fwrite($fp, "<Not_Available_Time><Day>Δευτέρα</Day><Hour>10:00</Hour></Not_Available_Time>\n");
26 |         fwrite($fp, "<Not_Available_Time><Day>Δευτέρα</Day><Hour>11:00</Hour></Not_Available_Time>\n");
27 |         fwrite($fp, "<Not_Available_Time><Day>Δευτέρα</Day><Hour>12:00</Hour></Not_Available_Time>\n");
28 |         fwrite($fp, "<Not_Available_Time><Day>Δευτέρα</Day><Hour>13:00</Hour></Not_Available_Time>\n");
29 |     }
30 |     if($newArray["monday_p"]==0){
31 |         fwrite($fp, "<Not_Available_Time><Day>Δευτέρα</Day><Hour>14:00</Hour></Not_Available_Time>\n");
32 |         fwrite($fp, "<Not_Available_Time><Day>Δευτέρα</Day><Hour>15:00</Hour></Not_Available_Time>\n");
33 |         fwrite($fp, "<Not_Available_Time><Day>Δευτέρα</Day><Hour>16:00</Hour></Not_Available_Time>\n");
34 |         fwrite($fp, "<Not_Available_Time><Day>Δευτέρα</Day><Hour>17:00</Hour></Not_Available_Time>\n");
35 |         fwrite($fp, "<Not_Available_Time><Day>Δευτέρα</Day><Hour>18:00</Hour></Not_Available_Time>\n");
36 |         fwrite($fp, "<Not_Available_Time><Day>Δευτέρα</Day><Hour>19:00</Hour></Not_Available_Time>\n");
37 |     }

```

Εικόνα 6.18 generate.php (Μέρος 1)

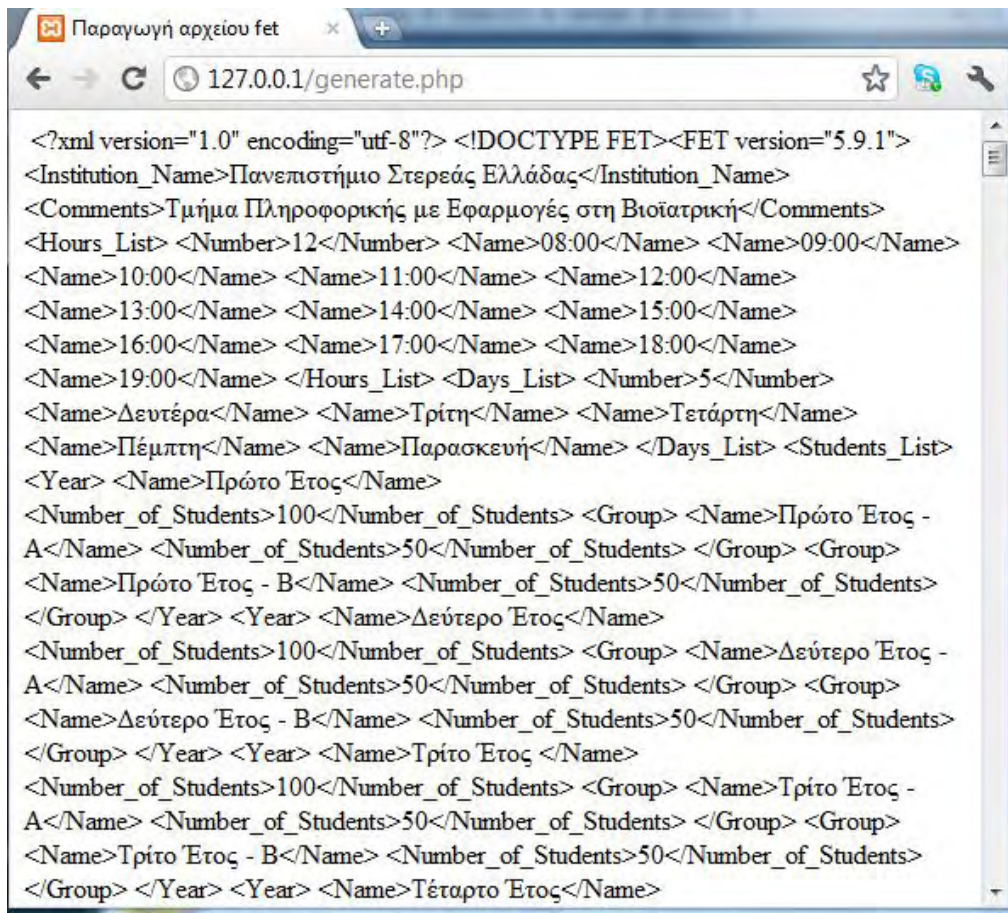


```

107 $fw=fopen("result.fet", "w") or die("Couldn't open result.fet");
108 ($f1=fopen("start.txt", "r+")) or die("Couldn't open file, sorry");
109 ($f2=fopen("gxml.txt", "r+")) or die("Couldn't open file, sorry");
110 ($f3=fopen("end.txt", "r+")) or die("Couldn't open file, sorry");
111 $file1="start.txt";
112 $file2="gxml.txt";
113 $file3="end.txt";
114 while(!feof($f1)){
115     $stext=fread($f1, filesize($file1));
116     fwrite($fw, $stext);
117 }
118 fclose($f1);
119 while(!feof($f2)){
120     $stet=fread($f2, filesize($file2));
121     fwrite($fw,$stet);
122 }
123 fclose($f2);
124 while(!feof($f3)){
125     $stex=fread($f3, filesize($file3));
126     fwrite($fw,$stex);
127 }
128 fclose($f3);
129 fclose($fw);
130 $file="result.fet";
131 echo htmlspecialchars(file_get_contents($file));
132 ?>
133 </html>

```

Εικόνα 6.21 generate.php (Μέρος 4)



Εικόνα 6.22 Παραγωγή Αρχείου fet

**ΜΕΡΟΣ V**  
**Συμπεράσματα**

# ΚΕΦΑΛΑΙΟ 7

## Συμπεράσματα

Σκέφτομαι συνέχεια για μήνες και χρόνια.  
Τις 99 φορές το συμπέρασμα είναι λάθος.  
Την εκατοστή φορά είναι σωστό.

~Albert Einstein (1879-1955)

### 7.1 Συμπεράσματα

Ολοκληρώνοντας την παρούσα εργασία και έχοντας ασχοληθεί με την παραμετροποίηση της εφαρμογής FET καθώς και με τη κατασκευή της ωρολόγιας διαδικτυακής εφαρμογής, μπορούμε να εξάγουμε κάποια συμπεράσματα όπως:

1. Η παραμετροποίηση της FET για ένα εκπαιδευτικό ίδρυμα είναι απλή υπόθεση, αρκεί ο υπεύθυνος για τα ωρολόγια προγράμματα να έχει στη διάθεση του όλες τις πληροφορίες που σχετίζονται με τους διδάσκοντες, τα μαθήματα και τους χώρους μάθησης.
2. Κατά τη διάρκεια της παραμετροποίησης πρέπει να προσεχθεί η καταχώρηση των περιορισμών χώρου σε περιπτώσεις που ο αριθμός των αιθουσών ή των εργαστηρίων είναι μικρός και δεν επαρκεί για όλες τις ομάδες φοίτησης.
3. Τέλος, η ωρολόγια διαδικτυακή εφαρμογή που κατασκευάστηκε θεωρώ πως είναι μία καλή βάση που μπορεί να εξυπηρετήσει τόσο τους διδάσκοντες όσο και τον υπεύθυνο για τα ωρολόγια προγράμματα αλλά χρειάζονται ακόμη πολλές βελτιώσεις.

### 7.2 Μελλοντικές Προεκτάσεις

Κάποιες μελλοντικές προεκτάσεις για την εφαρμογή FET θα μπορούσαν να είναι οι παρακάτω:

- Να προσαρμοστεί το περιβάλλον χρήσης της στην ελληνική γλώσσα για την τριτοβάθμια εκπαίδευση καθώς μέχρι στιγμής γίνεται προσπάθεια μόνο για την πρωτοβάθμια και δευτεροβάθμια εκπαίδευση [65].
- Να παραμετροποιηθεί για τη δημιουργία ωρολογίων προγραμμάτων εξετάσεων του τμήματος. Έτσι θα μπορούσαμε παρατηρήσουμε ποιες διαφορές υπάρχουν με τα ωρολόγια προγράμματα των εξαμήνων.

Κάποιες μελλοντικές προεκτάσεις για την ωρολόγια διαδικτυακή εφαρμογή θα μπορούσαν να είναι οι παρακάτω:

- Να δημιουργηθεί πίνακας της εβδομάδας με όλες τις ώρες αναλυτικά έτσι ώστε ο διδάσκοντας να μπορεί να επιλέξει ακριβώς ποιες ώρες είναι διαθέσιμος να διδάσκει.
- Να προστεθούν στην ιστοσελίδα χαρακτηριστικά όπως είναι η τρέχουσα ημερομηνία, η προθεσμία υποβολής της διαθεσιμότητας και email (ηλεκτρονική διεύθυνση) επικοινωνίας με τον υπεύθυνο των ωρολογίων προγραμμάτων.

- Να προστεθούν στη βάση δεδομένων επιπλέον πεδία όπως η ηλεκτρονική διεύθυνση του διδάσκοντα, το μικρό όνομα του διδάσκοντα σε περίπτωση που υπάρχει συνωνυμία και τα μαθήματα διδασκαλίας.
- Τέλος να δημιουργηθεί πίνακας στην ιστοσελίδα ο οποίος θα περιέχει τους χώρους μάθησης προκειμένου να επιλέξει ο κάθε διδάσκοντας το χώρο προτίμησης του. Αυτό θα ήταν καλό να γίνει αν θέλουμε να προσθέσουμε τέτοιους περιορισμούς τύπου. Σε αυτήν την περίπτωση θα πρέπει να δημιουργηθεί ο κατάλληλος κώδικας ο οποίος θα παράγει το αντίστοιχο αρχείο εισόδου fet.



## ΠΑΡΑΡΤΗΜΑ

Στο Παράρτημα αυτό παρουσιάζουμε το XML αρχείο που δημιούργησε η διαδικτυακή εφαρμογή και το οποίο δόθηκε σαν είσοδος στο πρόγραμμα FET.

### result.fet

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE FET><FET version="5.9.1">
<Institution_Name>Πανεπιστήμιο Στερεάς Ελλάδας</Institution_Name>
<Comments>Τμήμα Πληροφορικής με Εφαρμογές στη Βιοϊατρική</Comments>

<Hours_List>
<Number>12</Number>
<Name>08:00</Name>
<Name>09:00</Name>
<Name>10:00</Name>
<Name>11:00</Name>
<Name>12:00</Name>
<Name>13:00</Name>
<Name>14:00</Name>
<Name>15:00</Name>
<Name>16:00</Name>
<Name>17:00</Name>
<Name>18:00</Name>
<Name>19:00</Name>
</Hours_List>

<Days_List>
<Number>5</Number>
<Name>Δευτέρα</Name>
<Name>Τρίτη</Name>
<Name>Τετάρτη</Name>
<Name>Πέμπτη</Name>
<Name>Παρασκευή</Name>
</Days_List>

<Students_List>
<Year>
<Name>Πρώτο Έτος</Name>
<Number_of_Students>100</Number_of_Students>
<Group>
<Name>Πρώτο Έτος - Α</Name>
<Number_of_Students>50</Number_of_Students>
</Group>
<Group>
<Name>Πρώτο Έτος - Β</Name>
<Number_of_Students>50</Number_of_Students>
</Group>
</Year>
<Year>
<Name>Δεύτερο Έτος</Name>
<Number_of_Students>100</Number_of_Students>
<Group>
<Name>Δεύτερο Έτος - Α</Name>
<Number_of_Students>50</Number_of_Students>
</Group>
<Group>
```

<Name>Δεύτερο Έτος - Β</Name>  
<Number\_of\_Students>50</Number\_of\_Students>  
</Group>  
</Year>  
<Year>  
<Name>Τρίτο Έτος </Name>  
<Number\_of\_Students>100</Number\_of\_Students>  
<Group>  
<Name>Τρίτο Έτος - Α</Name>  
<Number\_of\_Students>50</Number\_of\_Students>  
</Group>  
<Group>  
<Name>Τρίτο Έτος - Β</Name>  
<Number\_of\_Students>50</Number\_of\_Students>  
</Group>  
</Year>  
<Year>  
<Name>Τέταρτο Έτος</Name>  
<Number\_of\_Students>100</Number\_of\_Students>  
<Group>  
<Name>Τέταρτο Έτος - Α</Name>  
<Number\_of\_Students>50</Number\_of\_Students>  
</Group>  
<Group>  
<Name>Τέταρτο Έτος - Β</Name>  
<Number\_of\_Students>50</Number\_of\_Students>  
</Group>  
</Year>  
<Year>  
<Name>Ελεύθερη Επιλογή</Name>  
<Number\_of\_Students>100</Number\_of\_Students>  
</Year>  
</Students\_List>

<Teachers\_List>  
<Teacher>  
<Name>Αδάμ</Name>  
</Teacher>  
<Teacher>  
<Name>Αλετράς</Name>  
</Teacher>  
<Teacher>  
<Name>Αναγνωστόπουλος</Name>  
</Teacher>

<Teacher>  
<Name>Αποστολόπουλος</Name>  
</Teacher>  
<Teacher>  
<Name>Ασημάκης</Name>  
</Teacher>  
<Teacher>  
<Name>Βάβουλας</Name>  
</Teacher>  
<Teacher>  
<Name>Βασιλάκη</Name>  
</Teacher>  
<Teacher>  
<Name>Βασιλακόπουλος</Name>

</Teacher>  
<Teacher>  
<Name>Βαϊόπουλος</Name>  
</Teacher>  
<Teacher>  
<Name>Γεωργίου</Name>  
</Teacher>  
<Teacher>  
<Name>Γεωργακάλου</Name>  
</Teacher>  
<Teacher>  
<Name>Γκανέτσος</Name>  
</Teacher>  
<Teacher>  
<Name>Γούγα</Name>  
</Teacher>  
<Teacher>  
<Name>Δελήμπασης</Name>  
</Teacher>  
<Teacher>  
<Name>Ιακωβίδης</Name>  
</Teacher>  
<Teacher>  
<Name>Κακαρούντας</Name>  
</Teacher>  
<Teacher>  
<Name>Καραγεώργος</Name>  
</Teacher>  
<Teacher>  
<Name>Καρυδάκης</Name>  
</Teacher>  
<Teacher>  
<Name>Κοσμόπουλος</Name>  
</Teacher>  
<Teacher>  
<Name>Λουκόπουλος</Name>  
</Teacher>  
<Teacher>  
<Name>Μάρκου</Name>  
</Teacher>  
<Teacher>  
<Name>Μαγκλογιάννης</Name>  
</Teacher>  
<Teacher>  
<Name>Μπάγκος</Name>  
</Teacher>  
<Teacher>  
<Name>Μπράλιου</Name>  
</Teacher>  
<Teacher>  
<Name>Μυλωνάς</Name>  
</Teacher>  
<Teacher>  
<Name>Νικολόπουλος</Name>  
</Teacher>  
<Teacher>  
<Name>Παπαγεωργίου</Name>  
</Teacher>  
<Teacher>  
<Name>Παπαδοπούλου</Name>

```

</Teacher>
<Teacher>
<Name>Παπανάγνου</Name>
</Teacher>
<Teacher>
<Name>Παππά</Name>
</Teacher>
<Teacher>
<Name>Παρασκευοπούλου</Name>
</Teacher>
<Teacher>
<Name>Πλαγιανάκος</Name>
</Teacher>
<Teacher>
<Name>Σίσκου</Name>
</Teacher>
<Teacher>
<Name>Σανδαλίδης</Name>
</Teacher>
<Teacher>
<Name>Σπαθούλας</Name>
</Teacher>
<Teacher>
<Name>Φούρλας</Name>
</Teacher>
<Teacher>
<Name>Χατζηδήμου</Name>
</Teacher>
</Teachers_List>

<Subjects_List>
<Subject>
<Name>Αγγλικά IV</Name>
</Subject>
<Subject>
<Name>Αγγλικά I</Name>
</Subject>
<Subject>
<Name>Αγγλικά II</Name>
</Subject>
<Subject>
<Name>Αγγλικά III</Name>
</Subject>
<Subject>
<Name>Ανάλυση Βιοϊατρικών Εικόνων</Name>
</Subject>
<Subject>
<Name>Ανάλυση Βιοϊατρικών Εικόνων Ε</Name>
</Subject>
<Subject>
<Name>Ανάλυση Συστημάτων</Name>
</Subject>
<Subject>
<Name>Ανάλυση Συστημάτων Ε</Name>
</Subject>
<Subject>
<Name>Αναγνώριση Προτύπων</Name>
</Subject>
<Subject>
<Name>Ανατομία και Φυσιολογία Ι</Name>

```

</Subject>  
<Subject>  
<Name>Ανατομία και Φυσιολογία I ΝΟΣ</Name>  
</Subject>  
<Subject>  
<Name>Ανατομία και Φυσιολογία ΙΙ</Name>  
</Subject>  
<Subject>  
<Name>Ανατομία και Φυσιολογία ΙΙ ΝΟΣ</Name>  
</Subject>  
<Subject>  
<Name>Αντικειμενοστρεφής Προγραμματισμός</Name>  
</Subject>  
<Subject>  
<Name>Αντικειμενοστρεφής Προγραμματισμός Ε</Name>  
</Subject>  
<Subject>  
<Name>Αριθμητική Ανάλυση</Name>  
</Subject>  
<Subject>  
<Name>Αριθμητική Ανάλυση Ε</Name>  
</Subject>  
<Subject>  
<Name>Αρχές Ηλεκτρονικής</Name>  
</Subject>  
<Subject>  
<Name>Αρχές Ηλεκτρονικής Ε</Name>  
</Subject>  
<Subject>  
<Name>Αρχιτεκτονική Υπολογιστών</Name>  
</Subject>  
<Subject>  
<Name>Ασφάλεια Συστημάτων Υπολογιστών</Name>  
</Subject>  
<Subject>  
<Name>Ασύρματα Επικοινωνιακά Συστήματα</Name>  
</Subject>  
<Subject>  
<Name>Βάσεις Δεδομένων </Name>  
</Subject>  
<Subject>  
<Name>Βάσεις Δεδομένων Ε</Name>  
</Subject>  
<Subject>  
<Name>Βιολογία Ι</Name>  
</Subject>  
<Subject>  
<Name>Βιολογία ΙΙ</Name>  
</Subject>  
<Subject>  
<Name>Βιοπληροφορική Ι</Name>  
</Subject>  
<Subject>  
<Name>Βιοπληροφορική ΙΙ</Name>  
</Subject>  
<Subject>  
<Name>Βιοστατιστική Ι</Name>  
</Subject>  
<Subject>  
<Name>Βιοστατιστική Ι Ε</Name>

</Subject>  
<Subject>  
<Name>Βιοχημεία</Name>  
</Subject>  
<Subject>  
<Name>Γενετική και Ασθένειες-Πληθυσμιακή Γενετική</Name>  
</Subject>  
<Subject>  
<Name>Γραμμική Άλγεβρα</Name>  
</Subject>  
<Subject>  
<Name>Γραφική με Υπολογιστές</Name>  
</Subject>  
<Subject>  
<Name>Δίκτυα Υπολογιστών</Name>  
</Subject>  
<Subject>  
<Name>Δημόσιες Σχέσεις</Name>  
</Subject>  
<Subject>  
<Name>Δημόσιο Δίκαιο</Name>  
</Subject>  
<Subject>  
<Name>Διακριτά Μαθηματικά</Name>  
</Subject>  
<Subject>  
<Name>Διασυνδεδεμένα Συστήματα Υπολογιστών</Name>  
</Subject>  
<Subject>  
<Name>Διδακτική της Πληροφορικής </Name>  
</Subject>  
<Subject>  
<Name>Δομές Δεδομένων και Αλγόριθμοι</Name>  
</Subject>  
<Subject>  
<Name>Δομές Δεδομένων και Αλγόριθμοι E</Name>  
</Subject>  
<Subject>  
<Name>Ειδικά Θέματα Βιοπληροφορικής και Βιοηθική</Name>  
</Subject>  
<Subject>  
<Name>Εισαγωγή στη Βιοϊατρική Τεχνολογία</Name>  
</Subject>  
<Subject>  
<Name>Εισαγωγή στη Βιοϊατρική Τεχνολογία E</Name>  
</Subject>  
<Subject>  
<Name>Εισαγωγή στην Πληροφορική </Name>  
</Subject>  
<Subject>  
<Name>Εισαγωγή στην Πληροφορική E</Name>  
</Subject>  
<Subject>  
<Name>Εισαγωγή στην Πολιτική Οικονομία</Name>  
</Subject>  
  
<Subject>  
<Name>Εισαγωγή στις Τηλεπικοινωνίες</Name>  
</Subject>  
<Subject>

<Name>Εισαγωγή στις Τηλεπικοινωνίες E</Name>  
</Subject>  
<Subject>  
<Name>Εισαγωγή στον Προγραμματισμό</Name>  
</Subject>  
<Subject>  
<Name>Εισαγωγή στον Προγραμματισμό E</Name>  
</Subject>  
<Subject>  
<Name>Ενσωματωμένα Συστήματα Υπολογιστών</Name>  
</Subject>  
<Subject>  
<Name>Εξόρυξη Δεδομένων και Ανακάλυψη Γνώσης</Name>  
</Subject>  
<Subject>  
<Name>Επικοινωνία Ανθρώπου Υπολογιστή</Name>  
</Subject>  
<Subject>  
<Name>Επιχειρηματικότητα και Καινοτομία</Name>  
</Subject>  
<Subject>  
<Name>Εφαρμοσμένα Μαθηματικά</Name>  
</Subject>  
<Subject>  
<Name>Θεσμοί και Διοίκηση της Ε.Ε</Name>  
</Subject>  
<Subject>  
<Name>Θεωρία Γλωσσών</Name>  
</Subject>  
<Subject>  
<Name>Θεωρία Γράφων</Name>  
</Subject>  
<Subject>  
<Name>Θεωρία Υπολογισμού</Name>  
</Subject>  
<Subject>  
<Name>Κοινωνιολογία</Name>  
</Subject>  
<Subject>  
<Name>Λειτουργικά Συστήματα</Name>  
</Subject>  
<Subject>  
<Name>Λειτουργικά Συστήματα E</Name>  
</Subject>  
<Subject>  
<Name>Λογική Σχεδίαση</Name>  
</Subject>  
<Subject>  
<Name>Λογική Σχεδίαση E</Name>  
</Subject>  
<Subject>  
<Name>Μαθηματική Ανάλυση I</Name>  
</Subject>  
<Subject>  
<Name>Μαθηματική Ανάλυση II</Name>  
</Subject>  
<Subject>  
<Name>Οργάνωση και Διοίκηση Συστημάτων Υγείας</Name>  
</Subject>  
<Subject>

<Name>Παιδαγωγικά</Name>  
</Subject>  
<Subject>  
<Name>Πιθανότητες και Στοιχεία Στατιστικής</Name>  
</Subject>  
<Subject>  
<Name>Πληροφοριακά Συστήματα Υγείας I</Name>  
</Subject>  
<Subject>  
<Name>Πληροφοριακά Συστήματα Υγείας II</Name>  
</Subject>  
<Subject>  
<Name>Προσομοίωση Βιολογικών Συστημάτων</Name>  
</Subject>  
<Subject>  
<Name>Προσομοίωση Βιολογικών Συστημάτων E</Name>  
</Subject>  
<Subject>  
<Name>ΣΑΕ και Εφαρμογές στη Βιοϊατρική</Name>  
</Subject>  
<Subject>  
<Name>Σήματα και Συστήματα</Name>  
</Subject>  
<Subject>  
<Name>Στοιχεία Θεωρίας Πληροφορίας και Κωδικών</Name>  
</Subject>  
<Subject>  
<Name>Συστήματα Ιατρικής Απεικόνισης</Name>  
</Subject>  
<Subject>  
<Name>Συστήματα Στήριξης Ιατρικών Αποφάσεων</Name>  
</Subject>  
<Subject>  
<Name>Τεχνητή Νοημοσύνη</Name>  
</Subject>  
<Subject>  
<Name>Τεχνολογία Ιατρικού Εξοπλισμού</Name>  
</Subject>  
<Subject>  
<Name>Τεχνολογία Ιατρικού Εξοπλισμού E</Name>  
</Subject>  
<Subject>  
<Name>Τεχνολογία Λογισμικού</Name>  
</Subject>  
<Subject>  
<Name>Τεχνολογίες Εφαρμογών Διαδικτύου</Name>  
</Subject>  
<Subject>  
<Name>Τηλεϊατρική</Name>  
</Subject>  
<Subject>  
<Name>Φυσική</Name>  
</Subject>  
<Subject>  
<Name>Ψηφιακές Επικοινωνίες</Name>  
</Subject>  
<Subject>  
<Name>Ψηφιακή Επεξεργασία Βιοσημάτων</Name>  
</Subject>  
<Subject>



```
<Name>Ψηφιακή Επεξεργασία Βιοσημάτων E</Name>
</Subject>
</Subjects_List>
```

```
<Activity_Tags_List>
</Activity_Tags_List>
```

```
<Activities_List>
<Activity>
<Teacher>Λουκόπουλος</Teacher>
<Subject>Λειτουργικά Συστήματα</Subject>
<Duration>2</Duration>
<Total_Duration>4</Total_Duration>
<Id>1</Id>
<Activity_Group_Id>1</Activity_Group_Id>
<Active>true</Active>
<Students>Δεύτερο Έτος - B</Students>
</Activity>
<Activity>
<Teacher>Λουκόπουλος</Teacher>
<Subject>Λειτουργικά Συστήματα</Subject>
<Duration>2</Duration>
<Total_Duration>4</Total_Duration>
<Id>2</Id>
<Activity_Group_Id>1</Activity_Group_Id>
<Active>true</Active>
<Students>Δεύτερο Έτος - B</Students>
</Activity>
<Activity>
<Teacher>Βαϊόπουλος</Teacher>
<Subject>Λειτουργικά Συστήματα E</Subject>
<Duration>2</Duration>
<Total_Duration>2</Total_Duration>
<Id>3</Id>
<Activity_Group_Id>0</Activity_Group_Id>
<Active>true</Active>
<Students>Δεύτερο Έτος - B</Students>
</Activity>
<Activity>
<Teacher>Ασημάκης</Teacher>
<Subject>Σήματα και Συστήματα</Subject>
<Duration>2</Duration>
<Total_Duration>4</Total_Duration>
<Id>4</Id>
<Activity_Group_Id>4</Activity_Group_Id>
<Active>true</Active>
<Students>Δεύτερο Έτος - B</Students>
</Activity>
<Activity>
<Teacher>Ασημάκης</Teacher>
<Subject>Σήματα και Συστήματα</Subject>
<Duration>2</Duration>
<Total_Duration>4</Total_Duration>
<Id>5</Id>
<Activity_Group_Id>4</Activity_Group_Id>
<Active>true</Active>
<Students>Δεύτερο Έτος - B</Students>
</Activity>
<Activity>
<Teacher>Λουκόπουλος</Teacher>
```

```

<Subject>Δομές Δεδομένων και Αλγόριθμοι Ε</Subject>
<Duration>2</Duration>
<Total_Duration>4</Total_Duration>
<Id>6</Id>
<Activity_Group_Id>6</Activity_Group_Id>
<Active>>true</Active>
<Students>Δεύτερο Έτος - Β</Students>
</Activity>
<Activity>
<Teacher>Λουκόπουλος</Teacher>
<Subject>Δομές Δεδομένων και Αλγόριθμοι Ε</Subject>
<Duration>2</Duration>
<Total_Duration>4</Total_Duration>
<Id>7</Id>
<Activity_Group_Id>6</Activity_Group_Id>
<Active>>true</Active>
<Students>Δεύτερο Έτος - Β</Students>
</Activity>
<Activity>
<Teacher>Αναγνωστόπουλος</Teacher>
<Subject>Δίκτυα Υπολογιστών</Subject>
<Duration>2</Duration>
<Total_Duration>4</Total_Duration>
<Id>8</Id>
<Activity_Group_Id>8</Activity_Group_Id>
<Active>>true</Active>
<Students>Δεύτερο Έτος - Β</Students>
</Activity>
<Activity>
<Teacher>Αναγνωστόπουλος</Teacher>
<Subject>Δίκτυα Υπολογιστών</Subject>
<Duration>2</Duration>
<Total_Duration>4</Total_Duration>
<Id>9</Id>
<Activity_Group_Id>8</Activity_Group_Id>
<Active>>true</Active>
<Students>Δεύτερο Έτος - Β</Students>
</Activity>
<Activity>
<Teacher>Παππά</Teacher>
<Subject>Αγγλικά IV</Subject>
<Duration>2</Duration>
<Total_Duration>2</Total_Duration>
<Id>10</Id>
<Activity_Group_Id>0</Activity_Group_Id>
<Active>>true</Active>
<Students>Δεύτερο Έτος - Β</Students>
</Activity>
<Activity>
<Teacher>Παπανάγνου</Teacher>
<Subject>Ανατομία και Φυσιολογία ΙΙ ΝΟΣ</Subject>
<Duration>2</Duration>
<Total_Duration>2</Total_Duration>
<Id>11</Id>
<Activity_Group_Id>0</Activity_Group_Id>
<Active>>true</Active>
<Students>Δεύτερο Έτος - Β</Students>
</Activity>
<Activity>
<Teacher>Βασιλακόπουλος</Teacher>

```

```

<Subject>Δομές Δεδομένων και Αλγόριθμοι</Subject>
<Duration>2</Duration>
<Total_Duration>4</Total_Duration>
<Id>12</Id>
<Activity_Group_Id>12</Activity_Group_Id>
<Active>>true</Active>
<Students>Δεύτερο Έτος - Β</Students>
</Activity>
<Activity>
<Teacher>Βασιλακόπουλος</Teacher>
<Subject>Δομές Δεδομένων και Αλγόριθμοι</Subject>
<Duration>2</Duration>
<Total_Duration>4</Total_Duration>
<Id>13</Id>
<Activity_Group_Id>12</Activity_Group_Id>
<Active>>true</Active>
<Students>Δεύτερο Έτος - Β</Students>
</Activity>
<Activity>
<Teacher>Παπανάγνου</Teacher>
<Subject>Ανατομία και Φυσιολογία ΙΙ</Subject>
<Duration>3</Duration>
<Total_Duration>3</Total_Duration>
<Id>14</Id>
<Activity_Group_Id>0</Activity_Group_Id>
<Active>>true</Active>
<Students>Δεύτερο Έτος - Β</Students>
</Activity>
<Activity>
<Teacher>Νικολόπουλος</Teacher>
<Subject>Βιοστατιστική Ι</Subject>
<Duration>2</Duration>
<Total_Duration>2</Total_Duration>
<Id>15</Id>
<Activity_Group_Id>0</Activity_Group_Id>
<Active>>true</Active>
<Students>Δεύτερο Έτος - Β</Students>
</Activity>
<Activity>
<Teacher>Νικολόπουλος</Teacher>
<Subject>Βιοστατιστική Ι Ε</Subject>
<Duration>2</Duration>
<Total_Duration>4</Total_Duration>
<Id>16</Id>
<Activity_Group_Id>16</Activity_Group_Id>
<Active>>true</Active>
<Students>Δεύτερο Έτος - Β</Students>
</Activity>
<Activity>
<Teacher>Νικολόπουλος</Teacher>
<Subject>Βιοστατιστική Ι Ε</Subject>
<Duration>2</Duration>
<Total_Duration>4</Total_Duration>
<Id>17</Id>
<Activity_Group_Id>16</Activity_Group_Id>
<Active>>true</Active>
<Students>Δεύτερο Έτος - Β</Students>
</Activity>
</Activities_List>

```

```

<Buildings_List>
<Building>
<Name>Κτήριο Νοσοκομείου</Name>
</Building>
<Building>
<Name>Κτήριο Πανεπιστημίου</Name>
</Building>
</Buildings_List>

<Rooms_List>
<Room>
<Name>Αίθουσα 1</Name>
<Building>Κτήριο Πανεπιστημίου</Building>
<Capacity>150</Capacity>
</Room>
<Room>
<Name>Αίθουσα 2</Name>
<Building>Κτήριο Πανεπιστημίου</Building>
<Capacity>150</Capacity>
</Room>
<Room>
<Name>Αίθουσα 3</Name>
<Building>Κτήριο Πανεπιστημίου</Building>
<Capacity>150</Capacity>
</Room>
<Room>
<Name>Εργαστήριο 1</Name>
<Building>Κτήριο Πανεπιστημίου</Building>
<Capacity>150</Capacity>
</Room>
<Room>
<Name>Εργαστήριο 2</Name>
<Building>Κτήριο Πανεπιστημίου</Building>
<Capacity>150</Capacity>
</Room>
<Room>
<Name>Εργαστήριο 3</Name>
<Building>Κτήριο Πανεπιστημίου</Building>
<Capacity>150</Capacity>
</Room>
<Room>
<Name>Εργαστήριο Ανατομίας</Name>
<Building>Κτήριο Νοσοκομείου</Building>
<Capacity>150</Capacity>
</Room>
</Rooms_List>

<Time_Constraints_List>
<ConstraintBasicCompulsoryTime>
<Weight_Percentage>100</Weight_Percentage>
</ConstraintBasicCompulsoryTime>
<ConstraintMinNDaysBetweenActivities>
<Weight_Percentage>95</Weight_Percentage>
<Consecutive_If_Same_Day>true</Consecutive_If_Same_Day>
<Number_of_Activities>2</Number_of_Activities>
<Activity_Id>1</Activity_Id>
<Activity_Id>2</Activity_Id>
<MinDays>1</MinDays>
</ConstraintMinNDaysBetweenActivities>
<ConstraintMinNDaysBetweenActivities>

```

<Weight\_Percentage>95</Weight\_Percentage>  
 <Consecutive\_If\_Same\_Day>true</Consecutive\_If\_Same\_Day>  
 <Number\_of\_Activities>2</Number\_of\_Activities>  
 <Activity\_Id>4</Activity\_Id>  
 <Activity\_Id>5</Activity\_Id>  
 <MinDays>1</MinDays>  
 </ConstraintMinNDaysBetweenActivities>  
 <ConstraintMinNDaysBetweenActivities>  
 <Weight\_Percentage>95</Weight\_Percentage>  
 <Consecutive\_If\_Same\_Day>true</Consecutive\_If\_Same\_Day>  
 <Number\_of\_Activities>2</Number\_of\_Activities>  
 <Activity\_Id>6</Activity\_Id>  
 <Activity\_Id>7</Activity\_Id>  
 <MinDays>1</MinDays>  
 </ConstraintMinNDaysBetweenActivities>  
 <ConstraintMinNDaysBetweenActivities>  
 <Weight\_Percentage>95</Weight\_Percentage>  
 <Consecutive\_If\_Same\_Day>true</Consecutive\_If\_Same\_Day>  
 <Number\_of\_Activities>2</Number\_of\_Activities>  
 <Activity\_Id>8</Activity\_Id>  
 <Activity\_Id>9</Activity\_Id>  
 <MinDays>1</MinDays>  
 </ConstraintMinNDaysBetweenActivities>  
 <ConstraintMinNDaysBetweenActivities>  
 <Weight\_Percentage>95</Weight\_Percentage>  
 <Consecutive\_If\_Same\_Day>true</Consecutive\_If\_Same\_Day>  
 <Number\_of\_Activities>2</Number\_of\_Activities>  
 <Activity\_Id>12</Activity\_Id>  
 <Activity\_Id>13</Activity\_Id>  
 <MinDays>1</MinDays>  
 </ConstraintMinNDaysBetweenActivities>  
 <ConstraintMinNDaysBetweenActivities>  
 <Weight\_Percentage>95</Weight\_Percentage>  
 <Consecutive\_If\_Same\_Day>true</Consecutive\_If\_Same\_Day>  
 <Number\_of\_Activities>2</Number\_of\_Activities>  
 <Activity\_Id>16</Activity\_Id>  
 <Activity\_Id>17</Activity\_Id>  
 <MinDays>1</MinDays>  
 </ConstraintMinNDaysBetweenActivities>

<ConstraintTeacherNotAvailableTimes>  
 <Weight\_Percentage>100</Weight\_Percentage>  
 <Teacher>Αδάμ</Teacher>  
 <Number\_of\_Not\_Available\_Times>24</Number\_of\_Not\_Available\_Times>  
 <Not\_Available\_Time><Day>Δευτέρα</Day><Hour>08:00</Hour></Not\_Available\_Time>  
 <Not\_Available\_Time><Day>Δευτέρα</Day><Hour>09:00</Hour></Not\_Available\_Time>  
 <Not\_Available\_Time><Day>Δευτέρα</Day><Hour>10:00</Hour></Not\_Available\_Time>  
 <Not\_Available\_Time><Day>Δευτέρα</Day><Hour>11:00</Hour></Not\_Available\_Time>  
 <Not\_Available\_Time><Day>Δευτέρα</Day><Hour>12:00</Hour></Not\_Available\_Time>  
 <Not\_Available\_Time><Day>Δευτέρα</Day><Hour>13:00</Hour></Not\_Available\_Time>  
 <Not\_Available\_Time><Day>Δευτέρα</Day><Hour>14:00</Hour></Not\_Available\_Time>  
 <Not\_Available\_Time><Day>Δευτέρα</Day><Hour>15:00</Hour></Not\_Available\_Time>  
 <Not\_Available\_Time><Day>Δευτέρα</Day><Hour>16:00</Hour></Not\_Available\_Time>  
 <Not\_Available\_Time><Day>Δευτέρα</Day><Hour>17:00</Hour></Not\_Available\_Time>  
 <Not\_Available\_Time><Day>Δευτέρα</Day><Hour>18:00</Hour></Not\_Available\_Time>  
 <Not\_Available\_Time><Day>Δευτέρα</Day><Hour>19:00</Hour></Not\_Available\_Time>  
 <Not\_Available\_Time><Day>Τρίτη</Day><Hour>08:00</Hour></Not\_Available\_Time>  
 <Not\_Available\_Time><Day>Τρίτη</Day><Hour>09:00</Hour></Not\_Available\_Time>  
 <Not\_Available\_Time><Day>Τρίτη</Day><Hour>10:00</Hour></Not\_Available\_Time>  
 <Not\_Available\_Time><Day>Τρίτη</Day><Hour>11:00</Hour></Not\_Available\_Time>

































<Not\_Available\_Time><Day>Τετάρτη</Day><Hour>18:00</Hour></Not\_Available\_Time>  
<Not\_Available\_Time><Day>Τετάρτη</Day><Hour>19:00</Hour></Not\_Available\_Time>  
<Not\_Available\_Time><Day>Πέμπτη</Day><Hour>14:00</Hour></Not\_Available\_Time>  
<Not\_Available\_Time><Day>Πέμπτη</Day><Hour>15:00</Hour></Not\_Available\_Time>  
<Not\_Available\_Time><Day>Πέμπτη</Day><Hour>16:00</Hour></Not\_Available\_Time>  
<Not\_Available\_Time><Day>Πέμπτη</Day><Hour>17:00</Hour></Not\_Available\_Time>  
<Not\_Available\_Time><Day>Πέμπτη</Day><Hour>18:00</Hour></Not\_Available\_Time>  
<Not\_Available\_Time><Day>Πέμπτη</Day><Hour>19:00</Hour></Not\_Available\_Time>  
<Not\_Available\_Time><Day>Παρασκευή</Day><Hour>14:00</Hour></Not\_Available\_Time>  
<Not\_Available\_Time><Day>Παρασκευή</Day><Hour>15:00</Hour></Not\_Available\_Time>  
<Not\_Available\_Time><Day>Παρασκευή</Day><Hour>16:00</Hour></Not\_Available\_Time>  
<Not\_Available\_Time><Day>Παρασκευή</Day><Hour>17:00</Hour></Not\_Available\_Time>  
<Not\_Available\_Time><Day>Παρασκευή</Day><Hour>18:00</Hour></Not\_Available\_Time>  
<Not\_Available\_Time><Day>Παρασκευή</Day><Hour>19:00</Hour></Not\_Available\_Time>  
</ConstraintTeacherNotAvailableTimes>  
</Time\_Constraints\_List>

<Space\_Constraints\_List>  
<ConstraintBasicCompulsorySpace>  
<Weight\_Percentage>100</Weight\_Percentage>

</ConstraintBasicCompulsorySpace>  
</Space\_Constraints\_List>

</FET>

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Fogel, David B. (editor) (1998). Evolutionary Computation: The Fossil Record. New York: IEEE Press. ISBN 0780334817.
- [2] Holland, John H (1975), Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor
- [3] Melanie Mitchel, An introduction to genetic algorithms, The MIT press, 1998
- [4] Cho Edwin K. P. Chong, Stanislaw H. Zak, An Introduction to Optimization, 2nd Edition, Wiley 2001
- [5] Din Ding – Zhu Du, Panos M. Pardalos, Weili Wu, Μαθηματική Θεωρία Βελτιστοποίησης, Εκδόσεις Νέων Τεχνολογιών, 2005
- [6] Noc Jorge Nocedal, Stephen J. Wright, Numerical Optimization. Springer 1999
- [7] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. 1990. Introduction to Algorithms. McGraw-Hill, New York
- [8] M. R. Garey, and D.S. Johnson. 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco
- [9] D. E. Knuth. 1997. The art of Computer Programming (Volumes 1-4). Addison-Wesley Professional
- [10] S. Hochbaum. 1997. Approximation algorithms for NP-hard problems. PWS Publishing Company, Boston
- [11] Παρασκευάς Η. Λεραντζής, “ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΠΑΡΑΓΩΓΗΣ ΜΕ ΧΡΗΣΗ ΓΕΝΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ”, Διπλωματική εργασία ΕΜΠ 2004
- [12] Eiben, A. E. et al (1994). "Genetic algorithms with multi-parent recombination". PPSN III: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: 78–87. ISBN 3-540-58484-6.
- [13] Ting, Chuan-Kang (2005). "On the Mean Convergence Time of Multi-parent Genetic Algorithms Without Selection". Advances in Artificial Life: 403–412
- [14] Goldberg, David E. (1989). Genetic Algorithms in Search Optimization and Machine Learning. Addison Wesley. pp. 41.
- [15] Alberto Bertoni , Marco Dorigo, “Implicit Parallelism in Genetic Algorithms” , Artificial Intelligence 61 (1993) , pp. 307-314
- [16] Genetic Algorithms by John H. Holland  
<http://www2.econ.iastate.edu/tesfatsi/holland.gaintro.htm>

- [17] H. Fang. Genetic Algorithms in Timetabling and Scheduling. PhD thesis, U niv. of Edimburg, 1994.
- [18] Abramson, D. A. and Abela, J. "A Parallel Genetic Algorithm for Solving the School Timetabling Problem", IJCAI workshop on Parallel Processing in AI, Sydney, August 91. Also appearing in 15 Australian Computer Science Conference, Hobart, Feb 1992, pp 1 - 11.
- [19] Burke Edmund , Elliman David , Weare Rupert (1993), A Genetic Algorithm for University Timetabling.
- [20] Sigl, B.; Golub, M.; Mornar, V., "Solving timetable scheduling problem using genetic algorithms", Proceedings of the 25th International Conference on Information Technology Interfaces, 2003. ITI 2003.
- [21] H.S.C.Lee, Timetabling Highly Constrained Systems via Genetic Algorithms, Masters Thesis, University of Philippines, Diliman, Quezon City, 2000
- [22] A.Schaerf, A Survey of Automated Timetabling, Artificial Intelligence Review 13(2), 87-127, 1999
- [23] S.Tongchim, Coarse-Grained Parallel Genetic Algorithm for Solving the Timetable Problem, Proc.of the 3rd Annual Nat.Symp.on Computational Science and Engineering. Bangkok, Thailand, 1999
- [24] H. Fang, P. Ross, D. Corne, "A promising Genetic Algorithm approach to job-shop scheduling, rescheduling, and open shop scheduling problems". In Proceedings of the Fifth International Conference on Genetic Algorithms, p.p 375-382, San Mateo, 1993.
- [25] FET: <http://www.lalescu.ro/liviu/fet/>
- [26] aSc TimeTables: <http://www.asctimetables.com/>
- [27] Lantiv TimeTabler: <http://www.lantiv.com/>
- [28] Dilip Datta, Kalyanmoy Deb, Carlos M. Fonseca, "Solving Class Timetabling Problem of IIT Kanpur using Multi-Objective Evolutionary Algorithm", KanGAL Report Number 2006006.
- [29] Κοκορέτσης Γεώργιος, "Αυτόματη Κατασκευή Ωρολογίου Προγράμματος για το Ενιαίο Λύκειο σε περιβάλλον Λογικού Προγραμματισμού με Περιορισμούς", Διπλωματική Εργασία, Διαπανεπιστημιακό Πρόγραμμα Μεταπτυχιακών Σπουδών στη Λογική και Θεωρία Αλγορίθμου και Υπολογισμού, Αθήνα 2004.
- [30] Σουρλίγκα Σοφία, "ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ ΜΕ ΤΗ ΧΡΗΣΗ ΓΕΝΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ", Μεταπτυχιακό Δίπλωμα Ειδίκευσης Μαθηματικά των Υπολογιστών και των Αποφάσεων, Πάτρα 2010.
- [31] Julie C. Meloni, "Μάθετε PHP, MySQL και Apache", Εκδόσεις Γκιούρδας.

- [32] Βλαχάβας Ιωάννης, Κεφαλάς Πέτρος, Βασιλειάδης Νικόλαος, Κόκκορας Φώτης, Σακελλαρίου Ηλίας, “Τεχνητή Νοημοσύνη”, Γ' Έκδοση, . Εκδόσεις Γκιούρδας.
- [33] Πάνος Αργυράκης, Νευρωνικά Δίκτυα & Εφαρμογές, Τόμος Β'.
- [34] <http://www.cs.sandia.gov/opt/survey/ea.html>
- [35] [http://en.wikipedia.org/wiki/Evolutionary\\_algorithm](http://en.wikipedia.org/wiki/Evolutionary_algorithm)
- [36] [http://aetos.it.teithe.gr/~adamidis/Ptyx/Kaklamanos\\_ptyx.pdf](http://aetos.it.teithe.gr/~adamidis/Ptyx/Kaklamanos_ptyx.pdf)
- [37] A.E. Eiben- J.E. Smith, Introduction to Evolutionary Computing.
- [38] [www.neural.uom.gr/Documents/projects/Thesis19.pdf](http://www.neural.uom.gr/Documents/projects/Thesis19.pdf)
- [39] Hishammuddin Asmuni, Edmund K. Burke and Jonathan M. Garibaldi, “Fuzzy Multiple Heuristic Ordering for Course Timetabling”.
- [40] <http://www.php.net/manual/en/index.php>
- [41] <http://www.w3schools.com/>
- [42] <http://www.school-timetabling.com/content/manual/UK/index.html>
- [43] [http://www.grupet.at/en/produkte/untis/uebersicht\\_untis.php](http://www.grupet.at/en/produkte/untis/uebersicht_untis.php)
- [44] <http://www.gnomikologikon.gr/>
- [45] [http://www.greek-language.gr/greekLang/modern\\_greek/tools/lexica/triantafyllides/index.html](http://www.greek-language.gr/greekLang/modern_greek/tools/lexica/triantafyllides/index.html)
- [46] <http://dictionary.sensagent.com/%CE%BA%CE%B1%CF%84%CF%8C%CF%80%CE%B9%CE%BD/el-el/>
- [47] <http://www.freestuff.gr/forums/index.php>
- [48] [http://el.wikipedia.org/wiki/%CE%9D%CE%B5%CF%85%CF%81%CF%89%CE%BD%CE%B9%CE%BA%CF%8C\\_%CE%B4%CE%AF%CE%BA%CF%84%CF%85%CE%BF](http://el.wikipedia.org/wiki/%CE%9D%CE%B5%CF%85%CF%81%CF%89%CE%BD%CE%B9%CE%BA%CF%8C_%CE%B4%CE%AF%CE%BA%CF%84%CF%85%CE%BF)
- [49] <http://dSPACE.lib.uom.gr/bitstream/2159/2337/7/PswmosPanagiwthsMSc2007.pdf>
- [50] Παπαδάκης Στέλιος, Αδαμίδης Παναγιώτης, “Άσαφή Συστήματα, Θεωρία και Εργαστηριακές Ασκήσεις”, Τ.Ε.Ι. Θεσσαλονίκης, Τμήμα Πληροφορικής, Μάϊος 2004.
- [51] Ε. Ανευλαβής, “Η άσαφής λογική στη Κλινική Ιατρική”, Κωνσταντοπούλειο ΓΠΙΝΝ «Η Αγία Όλγα», Ν.Ιωνία.
- [52] <http://www.mediamax.gr/articles/inDetail.cfm?EntryID=31>

- [53] [http://www.livepedia.gr/index.php/%CE%91%CF%83%CE%B1%CF%86%CE%AE%CF%82\\_%CE%9B%CE%BF%CE%B3%CE%B9%CE%BA%CE%AE](http://www.livepedia.gr/index.php/%CE%91%CF%83%CE%B1%CF%86%CE%AE%CF%82_%CE%9B%CE%BF%CE%B3%CE%B9%CE%BA%CE%AE)
- [54] [http://www.hep.upatras.gr/class/download/eid\\_the\\_psi\\_epe\\_sim\\_eik/%CE%91%CF%83%CE%B1%CF%86%CE%AE%CF%82%20%CE%BB%CE%BF%CE%B3%CE%B9%CE%BA%CE%AE.pdf](http://www.hep.upatras.gr/class/download/eid_the_psi_epe_sim_eik/%CE%91%CF%83%CE%B1%CF%86%CE%AE%CF%82%20%CE%BB%CE%BF%CE%B3%CE%B9%CE%BA%CE%AE.pdf)
- [55] [http://itia.ntua.gr/getfile/1109/7/documents/FuzzyLogic\\_HydroinfLesson.pdf](http://itia.ntua.gr/getfile/1109/7/documents/FuzzyLogic_HydroinfLesson.pdf)
- [56] [http://fuzzy-logic-automobile.blogspot.com/2010/07/blog-post\\_23.html](http://fuzzy-logic-automobile.blogspot.com/2010/07/blog-post_23.html)
- [57] <http://aibook.csd.auth.gr/include/slides/Chap14.pdf>
- [58] <http://www.dbnet.ece.ntua.gr/courses/AI/2002/notes/Chap04.pdf>
- [59] [http://hydra.it.teithe.gr/~adamidis/IntelSys/EA\\_notes.pdf](http://hydra.it.teithe.gr/~adamidis/IntelSys/EA_notes.pdf)
- [60] Krzysztof Socha, Michael Sampels, and Max Manfrin, “Ant Algorithms for the University Course Timetabling Problem with Regard to the State-of-the-Art”, IRIDIA, Université Libre de Bruxelles, CP 194/6, Av. Franklin D. Roosevelt 50, 1050 Bruxelles, Belgium.
- [61] Αστέριος Κακλαμάνος, Εισηγητής: Παναγιώτης Αδαμίδης, “ΔΗΜΙΟΥΡΓΙΑ ΕΠΕΝΔΥΤΙΚΟΥ ΜΟΝΤΕΛΟΥ ΜΕ ΧΡΗΣΗ ΕΞΕΛΙΚΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ”, Τ.Ε.Ι. Θεσσαλονίκης, Σχολή Τεχνολογικών Εφαρμογών, Τμήμα Πληροφορικής, Θεσσαλονίκη 2002.
- [62] Φοίβος Χ. Παλαιογιάννης, “Μεθοδολογία Βελτιστοποίησης Σμήνους Σωματιδίων για την εύρεση της βέλτιστης θέσης και των βέλτιστων ρυθμίσεων ευέλικτων συστημάτων μεταφοράς”, Διπλωματική Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Ηλεκτρικής Ισχύος, Αθήνα, Οκτώβριος 2010.
- [63] Μιχαήλ Γ. Επιτροπάκης, “Εκπαίδευση Τεχνητών Νευρωνικών Δικτύων, με την χρήση Εξελικτικών Αλγορίθμων, σε σειριακά και καταναμημένα συστήματα.”, Πανεπιστήμιο Πατρών Σχολή Θετικών Επιστημών Τμήμα Μαθηματικών Πάτρα, Μάιος 2008.
- [64] [http://en.wikipedia.org/wiki/Mathematical\\_optimization#Optimization\\_algorithms](http://en.wikipedia.org/wiki/Mathematical_optimization#Optimization_algorithms)
- [65] <http://lalescu.ro/liviu/fet/forum/index.php?topic=941.0>