

# **Ανοικτό Πανεπιστήμιο Κύπρου**

**Σχολή Θετικών και Εφαρμοσμένων Επιστημών**

## **Μεταπτυχιακή Διατριβή στα Πληροφοριακά Συστήματα**



**Ανάπτυξη Εξεταστικού Ημερολογίου, Εκτεταμένες Δοκιμές και  
Διαχείριση Πειραματικών Δεδομένων**

**Κωνσταντίνος Κουκίδης**

**Επιβλέπων Καθηγητής  
Δημήτριος Καλλές**

**Μάιος 2013**

# **Ανοικτό Πανεπιστήμιο Κύπρου**

## **Σχολή Θετικών και Εφαρμοσμένων Επιστημών**

**Ανάπτυξη Εξεταστικού Ημερολογίου, Εκτεταμένες Δοκιμές και  
Διαχείριση Πειραματικών Δεδομένων**

**Κωνσταντίνος Κουκίδης**

**Επιβλέπων Καθηγητής  
Δημήτριος Καλλές**

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε  
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών  
στα Πληροφοριακά Συστήματα

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών  
του Ανοικτού Πανεπιστημίου Κύπρου

**Μάιος 2013**

# Περίληψη

Στη παρούσα διπλωματική εργασία αρχικά περιγράφονται ενδεικτικά προβλήματα χρονοπρογραμματισμού που καλύπτουν ένα ευρύ φάσμα της ανθρώπινης δραστηριότητας. Για καθένα από τα προβλήματα αυτά καταγράφονται βασικοί κανόνες και οι περιορισμοί που θα πρέπει να ληφθούν υπόψη για την αποτελεσματική αντιμετώπισή τους. Επίσης περιγράφονται και κάποιες βασικές τεχνικές επίλυσης προβλημάτων χρονοπρογραμματισμού όπως είναι ο Δυναμικός Προγραμματισμός, ο Προγραμματισμός με Περιορισμούς και η Τοπική Αναζήτηση.

Έγινε ιδιαίτερη ανάλυση του Γενετικού Αλγορίθμου, ο οποίος αποτελεί τη βάση για το λογισμικό που δημιουργήθηκε και το οποίο θα δημιουργεί ένα εξεταστικό ημερολόγιο αυτόματα. Καταγράφεται η δομή του αλγορίθμου, τα πλεονεκτήματα που προσφέρει, τα κύρια χαρακτηριστικά του και ο τρόπος με τον οποίο εντοπίζεται η λύση σε ένα πρόβλημα.

Το πρόβλημα της κατάρτισης ενός εξεταστικού ημερολογίου αναλύθηκε και έγινε καταγραφή των ανελαστικών και των ελαστικών περιορισμών που πρέπει να λαμβάνει υπόψη του το λογισμικό. Έγιναν εκτεταμένες τροποποιήσεις και νέες προσθήκες σε υπάρχον λογισμικό ώστε να βελτιωθεί για να αντιμετωπίζει τις νέες απαιτήσεις του προβλήματος. Ακολούθησε εκτεταμένος αριθμός δοκιμών ώστε να ελεγχθεί το λογισμικό και να επιλεγούν οι κατάλληλοι παράμετροι του γενετικού αλγορίθμου ώστε το λογισμικό να γίνει ταχύτερο και αποτελεσματικότερο.

Ιδιαίτερα μελετήθηκαν τα Υπολογιστικά Συστήματα Πλέγματος (Grid) και προσαρμόστηκε κατάλληλα το λογισμικό που δημιουργήθηκε ώστε να μπορεί να εκτελεσθεί σε περιβάλλον παράλληλης επεξεργασίας αξιοποιώντας τις υποδομές του οργανισμού HellasGrid, ενώ εκτελέστηκε μια σειρά πειραμάτων και έγινε καταγραφή των αποτελεσμάτων που συμβάλανε στη διατύπωση ιδιαίτερων συμπερασμάτων.

**Λέξεις Κλειδιά:** Χρονοπρογραμματισμός Γεγονότων, Γενετικοί Αλγόριθμοι, Περιορισμοί, Παράλληλη Επεξεργασία, Υπολογιστικά Συστήματα Πλέγματος, Grid.

**Περιεχόμενο:** Κείμενο με Εικόνες, Βιβλιογραφικές Αναφορές, Παράρτημα, Λογισμικό.

## Summary

This thesis opens with a description of scheduling-timetabling problems that cover a wide spectrum of human activity, alongside basic and constraints that should be taken into consideration to address each one of these problems. Dynamic programming, constraint programming and local search techniques are then briefly described as regards their applicability to such problems and a more in-depth analysis is provided for genetic algorithms, which have been used as a solution workbench for this thesis. We then describe in detail present the genetic algorithm that solves the underlying examination timetabling problem and has been also implemented in software.

A complete analysis of the examination timetabling problem is also presented alongside the hard and soft constraints that must be taken into account. An existing software system has been extended to improve the way it handles up-to-date constraints of the underlying problem and extensive testing was carried out to fine-tune the genetic algorithm parameters and to improve software efficiency and effectiveness.

We have adopted the software and submitted it to a grid infrastructure though we eventually only utilised nodes featuring parallel processing (multi-core systems), via the HellasGrid system. We carried out a series of experiments and analysed the results to lead us to a series of observations regarding the behaviour of the implemented genetic algorithm.

**Keywords:** Job Scheduling, Genetic Algorithms, Constraints, Parallel Computing, Computational Grid Systems, Grid.

**Content:** Text and Images, References, Appendice, Software

## Ευχαριστίες

Ευχαριστώ ιδιαίτερα τον Επιβλέποντα Καθηγητή Δημήτριο Καλλέ για την εμπιστοσύνη που μου έδειξε αρχικά με την ανάθεση της παρούσας εργασίας, επίσης για την καθοδήγησή του σε σημεία καθοριστικά για την εξέλιξη της εργασίας καθώς και για την άμεση ανταπόκριση που είχε όποτε χρειάστηκα την βοήθειά του.

Ευχαριστίες αξίζουν επίσης στους ανθρώπους στο helpdesk @ grid.auth.gr για την άμεση ανταπόκρισή τους στην επίλυση των ελάχιστων προβλημάτων που προέκυψαν.

*Την Μεταπτυχιακή αυτή εργασία θα ήθελα να την αφιερώσω*

*στην σύζυγό μου Χρυσούλα*

*για την αμέριστη συμπαράσταση και την απaráμιλλη υπομονή που υπέδειξε*

*κατά τη διάρκεια του Μεταπτυχιακού Προγράμματος,*

*όπως και στα δυο αγγελούδια που από τα μέσα Φεβρουαρίου*

*άλλαξαν εντελώς τη ζωή μας.*

# Περιεχόμενα

Εισαγωγή .....	4
<b>Προβλήματα και Τεχνικές Επίλυσης Προβλημάτων .....</b>	<b>4</b>
<b>1.1 Προβλήματα Χρονοπρογραμματισμού .....</b>	<b>4</b>
1.1.1 Προγραμματισμός Προσωπικού .....	4
1.1.2 Εβδομαδιαία Ωρολόγια Προγράμματα για Σχολεία Δευτεροβάθμιας Εκπαίδευσης.....	4
1.1.3 Χρονικός Προγραμματισμός σε Συστήματα Παραγωγής Κατά Παραγγελία (job shop).....	4
1.1.4 Χρονικός Προγραμματισμός Δρομολογίων Τρένων .....	4
1.1.5 Χρονικός Προγραμματισμός Αθλητικών Δραστηριοτήτων .....	4
<b>1.2 Τεχνικές Επίλυσης Προβλημάτων .....</b>	<b>4</b>
1.2.1 Δυναμικός Προγραμματισμός (Dynamic Programming) .....	4
1.2.2 Προγραμματισμός με Περιορισμούς (Constraint Programming) .....	4
1.2.3 Τοπική Αναζήτηση (Local Search) .....	4
<b>Γενετικοί Αλγόριθμοι .....</b>	<b>4</b>
<b>2.1 Θεωρία της Εξέλιξης των Ειδών.....</b>	<b>4</b>
<b>2.2 Η Δομή του Αλγορίθμου .....</b>	<b>4</b>
<b>2.3 Πλεονεκτήματα Γενετικών Αλγορίθμων .....</b>	<b>4</b>
<b>2.4 Τα Κύρια Χαρακτηριστικά ενός Γενετικού Αλγορίθμου .....</b>	<b>4</b>
<b>2.5 Η Κωδικοποίηση .....</b>	<b>4</b>
<b>2.6 Οι Γενετικοί Τελεστές.....</b>	<b>4</b>
2.6.1 Αντικειμενική Συνάρτηση ή Συνάρτηση Καταλληλότητας .....	4
2.6.2 Επιλογή .....	4
2.6.3 Διασταύρωση .....	4
<b>2.7 Μετάλλαξη.....</b>	<b>4</b>
<b>2.8 Κριτήρια Τερματισμού.....</b>	<b>4</b>
<b>2.9 Παράδειγμα Γενετικών Αλγορίθμων .....</b>	<b>4</b>
<b>2.10 Εφαρμογές Γενετικών Αλγορίθμων .....</b>	<b>4</b>

<b>Το Πρόβλημα της Κατάρτισης Εξεταστικού Ημερολογίου με τη Χρήση Γενετικού Αλγορίθμου .....</b>	<b>4</b>
<b>3.1 Η Οπτική του Προβλήματος .....</b>	<b>4</b>
<b>3.2 Οι Περιορισμοί του Προβλήματος .....</b>	<b>4</b>
<b>3.3 Μαθηματικό Μοντέλο των Περιορισμών.....</b>	<b>4</b>
<b>3.4 Κωδικοποίηση του Προβλήματος σε Μορφή Χρωμοσώματος.....</b>	<b>4</b>
<b>3.5 Η Συνάρτηση Καταλληλότητας .....</b>	<b>4</b>
<b>3.6 Διάγραμμα Οντοτήτων Συσχετίσεων και τα Αρχεία Εισόδου .....</b>	<b>4</b>
<b>3.7 Το UML Διάγραμμα της Εφαρμογής.....</b>	<b>4</b>
<b>3.8 Το ECJ Toolkit .....</b>	<b>4</b>
<b>3.9 Εφαρμογή του Γενετικού Αλγόριθμου σε απλά Παραδείγματα .....</b>	<b>4</b>
3.9.1 Ορισμός του Προβλήματος.....	4
3.9.2 1ο Σενάριο.....	4
3.9.3 2ο Σενάριο:.....	4
3.9.4 3ο Σενάριο: .....	4
3.9.5 4ο Σενάριο .....	4
3.9.6 5ο Σενάριο:.....	4
<b>3.10 Εφαρμογή του Γενετικού Αλγόριθμου σε Μεγάλα Προβλήματα με τη Χρήση Πραγματικών Δεδομένων .....</b>	<b>4</b>
3.10.1 Παράλληλη Επεξεργασία σε Πολυπύρηνους Επεξεργαστές.....	4
3.10.2 Επιλογή Μεγέθους Πληθυσμού .....	4
3.10.3 Επιλογή Μορφής Διασταύρωσης .....	4
3.10.4 Υιοθέτηση των Προηγούμενων Συμπερασμάτων και Εκτέλεση του Αλγορίθμου για Περισσότερες Γενεές .....	4
<b>Πλέγμα Υπολογιστικών Συστημάτων–Grid Computing.....</b>	<b>4</b>
<b>4.1 Εισαγωγή – Τι είναι το Grid .....</b>	<b>4</b>
<b>4.2 Το πρόγραμμα EGEE – Enabling Grids for E-sciencE και ο οργανισμός European Grid Infrastructure .....</b>	<b>4</b>
<b>4.3 Παράγοντες που ευνόησαν την ανάπτυξη του Υπολογιστικού Πλέγματος ....</b>	<b>4</b>
<b>4.4 Η δομή του Grid .....</b>	<b>4</b>
<b>4.5 Βασικές αρχές του Υπολογιστικού Πλέγματος.....</b>	<b>4</b>
<b>4.6 Ο Οργανισμός HellasGrid.....</b>	<b>4</b>

<b>4.7 Εικονικοί Οργανισμοί - Virtual Organizations (VO)</b> .....	<b>4</b>
<b>4.8 Υποβολή εργασιών στο Grid</b> .....	<b>4</b>
<b>Εκτέλεση του Λογισμικού σε Περιβάλλον Υπολογιστικού Πλέγματος και Παράλληλης Επεξεργασίας</b> .....	<b>4</b>
<b>5.1 Παράλληλη Επεξεργασία</b> .....	<b>4</b>
5.1.1 Η Μέθοδος της Κατανεμημένης Αξιολόγησης (Distributed Evaluation).....	4
5.1.2 Η Μέθοδος των Νησιών (Islands Method) .....	4
5.1.3 Πολυνηματική Εκτέλεση των Διαδικασιών της Αναπαραγωγής και της Αξιολόγησης. ....	4
<b>5.2 Επιλογή Παράλληλης Μεθόδου</b> .....	<b>4</b>
<b>5.3 Παράλληλη Εκτέλεση του Προγράμματος σε Συγκεκριμένο Server (ui.afroditi.hellasgrid.gr)</b> .....	<b>4</b>
5.3.1 Σύγκριση Αποτελεσμάτων.....	4
5.3.2 Επιλογή Πιθανότητας Μετάλλαξης .....	4
5.3.3 Εκτέλεση του Προγράμματος για Μεγάλο Πλήθος Γενεών .....	4
<b>5.4 Εκτέλεση του Προγράμματος στο Grid</b> .....	<b>4</b>
<b>Επίλογος</b> .....	<b>4</b>
<b>Βιβλιογραφία</b> .....	<b>4</b>
<b>Παράρτημα Α</b> .....	<b>A4</b>
<b>Επισυναπτόμενο Λογισμικό</b> .....	<b>A4</b>
<b>A.1 Επισυναπτόμενος Κώδικας του Λογισμικού</b> .....	<b>A4</b>
<b>A.2 Οδηγίες για την Εκτέλεση του Λογισμικού</b> .....	<b>A4</b>



# Εισαγωγή

Η διαδικασία της κατάρτισης ενός αποδεκτού εξεταστικού ημερολογίου είναι ένα πρόβλημα που έχει απασχολήσει αρκετούς ερευνητές και έχει εκπονηθεί ένας μεγάλος αριθμός εργασιών υλοποιώντας αντίστοιχους αλγορίθμους, προσπαθώντας να παρουσιάσουν την βέλτιστη λύση.

Το πρόβλημα κατάρτισης ενός εξεταστικού ημερολογίου μπορεί να θεωρηθεί ως ένα πρόβλημα όπου ένας αριθμός εξετάσεων πρέπει να υλοποιηθεί μέσα στα πλαίσια καθορισμένων χρονοθυρίδων ή περιορισμένου χρονικού διαστήματος και παράλληλα να εκπληρώνεται ένας όσο το δυνατό μεγαλύτερος αριθμός συγκεκριμένων περιορισμών. Οι περιορισμοί αυτοί βέβαια διαφέρουν για κάθε ίδρυμα αλλά μπορούν να καταταχθούν σε δύο γενικές κατηγορίες ανάλογα με την απαίτηση υποχρεωτικής εκπλήρωσής τους. Οι ανελαστικοί περιορισμοί δεν επιτρέπουν καμία απόκλιση και προσδιορίζουν αν ένα εξεταστικό ημερολόγιο είναι αποδεκτό. Οι ελαστικοί περιορισμοί, η ικανοποίηση των οποίων δεν είναι υποχρεωτική αλλά επιθυμητή.

Έγινε ιδιαίτερη μελέτη ώστε το τελικό λογισμικό που δημιουργήθηκε να μπορεί να εκτελείται σε περιβάλλον υπολογιστικού πλέγματος με τεχνικές παράλληλου προγραμματισμού για να εκμεταλλευτεί πλήρως την αποτελεσματικότητα των γενετικών αλγορίθμων, οι οποίοι δίνουν λύσεις που προσεγγίζουν το βέλτιστο αλλά απαιτούν μεγάλη υπολογιστική δύναμη και συνήθως είναι δύσκολο να εκτελεστούν σε περιβάλλον προσωπικού υπολογιστή.

Η διάρθρωση της διπλωματικής διατριβής γίνεται σε πέντε κεφάλαια:

Στο **Πρώτο Κεφάλαιο** παρουσιάζονται ανάλογα προβλήματα χρονοπρογραμματισμού και θα περιγραφούν αντίστοιχες τεχνικές επίλυσης τέτοιων προβλημάτων που έχουν χρησιμοποιηθεί για τον προσδιορισμό βέλτιστης λύσης.

Στο **Δεύτερο Κεφάλαιο** γίνεται ενδελεχής παρουσίαση του γενετικού αλγορίθμου, ο οποίος χρησιμοποιήθηκε μέσω λογισμικού για την κατάρτιση ενός αποδεκτού εξεταστικού ημερολογίου στη συνέχεια.

Στο **Τρίτο Κεφάλαιο** διερευνάται το πρόβλημα, κατηγοριοποιούνται οι περιορισμοί σε ανελαστικούς και σε ελαστικούς. Στη συνέχεια παρουσιάζεται γενικά το λογισμικό που δημιουργήθηκε και το οποίο θα δημιουργεί προγράμματα εξετάσεων, τα οποία θα παρουσιάζονται στην οθόνη μετά την ολοκλήρωση της διαδικασίας. Εκτέλεση εκτεταμένων δοκιμών με κατάλληλα δεδομένα ώστε να δοκιμασθούν όλες οι περιπτώσεις περιορισμών καθώς και οι ιδιαίτερες πτυχές του γενετικού αλγορίθμου.

Στο **Τέταρτο Κεφάλαιο** γίνεται παρουσίαση των Υπολογιστικών Συστημάτων Πλέγματος για την εκτέλεση εφαρμογών που απαιτούν μεγάλη υπολογιστική ισχύ.

Στο **Πέμπτο Κεφάλαιο** εκτελείται το λογισμικό σε περιβάλλον υπολογιστικού πλέγματος και παράλληλης επεξεργασίας με τη χρήση μεγάλων σε όγκο δεδομένων και την καταγραφή των αποτελεσμάτων.

# Κεφάλαιο 1

## Προβλήματα και Τεχνικές Επίλυσης Προβλημάτων.

Οι εξελίξεις στον τεχνολογικό, οικονομικό, κοινωνικό και πολιτικό τομέα επιδρούν άμεσα στις ανθρώπινες δραστηριότητες, συμβάλλοντας στην εμφάνιση νέων προβλημάτων αλλά και στην διαφοροποίηση παλαιότερων κάνοντας τις όποιες τεχνικές επίλυσης εφαρμόζονταν ανεπαρκείς. Έτσι επειδή ο κόσμος εξελίσσεται, αλλάζουν και τα προβλήματα που καλείται να αντιμετωπίσει μεταμορφώνοντας την ανάγκη για εξεύρεση αποτελεσματικών λύσεων σε μια διαρκή διαδικασία.

## 1.1 Προβλήματα Χρονοπρογραμματισμού

Παρακάτω παρουσιάζονται και αναλύονται πέντε ενδεικτικά προβλήματα χρονοπρογραμματισμού που καλύπτουν ένα ευρύ φάσμα των ανθρώπινων δραστηριοτήτων. Τα προβλήματα αυτά λόγω του ενδιαφέροντος που παρουσιάζουν αποτέλεσαν κατ' επανάληψη αντικείμενο έρευνας, ενώ τα αποτελέσματα των ερευνών αυτών έχουν σκοπό να βοηθήσουν στην αποτελεσματική αντιμετώπισή τους.

### 1.1.1 Προγραμματισμός Προσωπικού

Το πρόβλημα της κατανομής του προσωπικού σε βάρδιες και η ανάθεση εργασιών αποτελεί σημαντική διαχειριστική δραστηριότητα για πολλούς οργανισμούς όπως είναι τα νοσοκομεία, τα αστυνομικά τμήματα, οι πυροσβεστικοί σταθμοί, οι μεγάλες βιομηχανικές μονάδες, οι εταιρείες μεταφορών κτλ που λειτουργούν περισσότερες από 16 έως 24 ώρες την ημέρα και για 7 ημέρες την εβδομάδα. [03]

Ο χρονοπρογραμματισμός του προσωπικού είναι η διαδικασία κατά την οποία δημιουργείται ένα πρόγραμμα εργασίας για το προσωπικό έτσι ώστε ο οργανισμός να μπορεί να καλύψει τις απαιτήσεις του για την παραγωγή αγαθών ή την παροχή υπηρεσιών. Το πρώτο στάδιο αυτής της διαδικασίας περιλαμβάνει τον καθορισμό του συνολικού αριθμού των εργαζομένων που απαιτούνται, καθώς και τα προσόντα που πρέπει αυτοί να διαθέτουν, για να καλυφθούν πλήρως οι ανάγκες του οργανισμού. Σε δεύτερο στάδιο ο κάθε εργαζόμενος κατανέμεται σε βάρδιες ώστε να πληρούνται τα απαιτούμενα επίπεδα στελέχωσης σε διάφορες χρονικές στιγμές και μετά γίνεται η ανάθεση των συγκεκριμένων εργασιών που πρέπει να εκτελεσθούν στα πλαίσια της κάθε βάρδιας. Η εργατική νομοθεσία και οι κανονισμοί πρέπει να τηρούνται καθόλη τη διαδικασία.

Είναι εξαιρετικά δύσκολο να βρεθούν καλές λύσεις για αυτά τα προβλήματα χρονοπρογραμματισμού, όπου υπάρχουν πολλοί περιορισμοί και σύνθετες δυσκολίες. Ακόμη πιο δύσκολο είναι να ορισθούν βέλτιστες λύσεις που ελαχιστοποιούν το κόστος, που λαμβάνουν υπόψη τις απαιτήσεις του προσωπικού, που κατανέμουν τις βάρδιες δίκαια μεταξύ των εργαζομένων και τέλος που καλύπτουν πλήρως όλους τους περιορισμούς στον εργασιακό χώρο.

Παρόλο που έχει αναπτυχθεί ένας μεγάλος αριθμός εμπορικών προγραμμάτων για την αντιμετώπιση του προβλήματος του χρονοπρογραμματισμού των εργαζομένων και της στελέχωσης των οργανισμών είναι αδύνατο αυτά τα προγράμματα να ταιριάζουν απόλυτα σε όλες τις περιπτώσεις και να είναι κατάλληλα για όλους τους οργανισμούς ή τις εταιρείες ακόμη και του ίδιου κλάδου. Για παράδειγμα ένα πρόγραμμα που καλύπτει τις ανάγκες ενός νοσοκομείου που κάνει εφημερίες πολύ δύσκολα θα κάλυπτε απόλυτα τις ανάγκες ενός αεροδρομίου ή μιας μεταφορικής εταιρείας. Για αυτό το λόγο έχουν χρησιμοποιηθεί πολλά μαθηματικά μοντέλα και αλγόριθμοι ενώ αναρίθμητες έρευνες έχουν γίνει ώστε να δοθούν λύσεις πάνω στο πρόβλημα της κατάλληλης στελέχωσης των οργανισμών καλύπτοντας τις ιδιαιτερότητες της κάθε περίπτωσης ξεχωριστά.

Η **Διαδικασία της Στελέχωσης** ενός οργανισμού είναι ένα πρόβλημα που σύμφωνα με τη βιβλιογραφία[08] μπορεί να λυθεί εάν διαχωριστεί σε πέντε στάδια ξεκινώντας με την καταγραφή του φόρτου εργασίας που πρέπει να εκτελεσθεί, σε ορισμένο χρονικό διάστημα και τελειώνει με την ανάθεση των εργασιών στους εργαζόμενους.

**1<sup>ο</sup> Στάδιο, Καταγραφή των Καθηκόντων:** Αρχικά πρέπει να προβλεφθεί και να καταγραφεί ο φόρτος εργασίας και το είδος των αναγκών που έχει ο υπό εξέταση οργανισμός, λαμβάνοντας υπόψη τις περιπτώσεις: α) όπου ο φόρτος εργασίας είναι σταθερός και η ζήτηση για εργαζομένους σταθερή, β) όπου πρέπει να γίνει πρόβλεψη για τις ανάγκες που θα προκύψουν, η ζήτηση δηλαδή για εργαζομένους είναι μεταβαλλόμενη και γ) όπου ο φόρτος εργασίας ορίζεται από την κάλυψη βαρδιών.

**2<sup>ο</sup> Στάδιο, Προσδιορισμός των Εργαζομένων:** Εδώ γίνεται ο προσδιορισμός του ελάχιστου αριθμού των εργαζομένων που απαιτούνται από τον οργανισμό ώστε να καλύπτεται πλήρως ο φόρτος εργασίας και τα καθήκοντα όπως αυτά έχουν προσδιορισθεί.

**3<sup>ο</sup> Στάδιο, Ορισμός των Αργιών και των Ημερών Ξεκούρασης:** Στο στάδιο αυτό περιγράφεται το πώς θα κατανέμονται οι ημέρες αργίας και ξεκούρασης στο προσωπικό και αφορά κυρίως το προσωπικό που εργάζεται σε βάρδιες ή καλύπτει ευέλικτες μορφές εργασίας.

**4<sup>ο</sup> Στάδιο, Προγραμματισμός Βαρδιών:** Εδώ περιγράφεται ο τρόπος με τον οποίο θα στελεχωθούν οι βάρδιες από το προσωπικό, ώστε να καλύπτονται οι ανάγκες του οργανισμού. Όλοι οι κανόνες της εργατικής νομοθεσίας πρέπει να λαμβάνονται υπόψη όπως και οι κανονισμοί του οργανισμού για να αντιμετωπίζονται θέματα όπως για παράδειγμα οι αλληλουχία των

ημερών εργασίας και αργίας, τα διαλείμματα που πρέπει να γίνουν κατά τη διάρκεια της βάρδιας ή του ταξιδιού, όταν αναφερόμαστε σε προγραμματισμό πληρωμάτων (αεροσκαφών, πλοίων κτλ).

**5<sup>ο</sup> Στάδιο, Ανάθεση Καθηκόντων:** Σ' αυτό το στάδιο γίνεται ορισμός των καθηκόντων σε κάθε εργαζόμενο, ανάλογα με τις ημέρες που αυτός εργάζεται. Ορισμένα από τα καθήκοντα είναι δυνατόν να απαιτούν ιδιαίτερα προσόντα ή εμπειρία και ως εκ τούτου μπορεί να ανατεθούν σε συγκεκριμένους εργαζόμενους.

### **1.1.2 Εβδομαδιαία Ωρολόγια Προγράμματα για Σχολεία Δευτεροβάθμιας Εκπαίδευσης**

Το πρόβλημα της δημιουργίας Εβδομαδιαίου Ωρολογίου Προγράμματος για σχολεία της δευτεροβάθμιας εκπαίδευσης στη γενική του μορφή είναι μια προσπάθεια να εντοπίσει τη βέλτιστη ακολουθία από σχολικές δραστηριότητες που μπορούν να εκτελεσθούν σε συγκεκριμένες ώρες, για χρονικό διάστημα συνήθως μιας εβδομάδας, λαμβάνοντας υπόψη ένα μεγάλο αριθμό εμπλεκόμενων οντοτήτων και εξειδικευμένων περιορισμών. [02]

Το πρόβλημα αυτό το αντιμετωπίζουν σχεδόν όλα τα σχολεία της Δευτεροβάθμιας Εκπαίδευσης, στην αρχή κάθε νέας σχολικής χρονιάς. Συνήθως αυτή η εργασία ανατίθεται σε έναν εκπαιδευτικό, ο οποίος δημιουργεί αρχικά ένα πρόχειρο πρόγραμμα και με συνεχείς αλλαγές προσπαθεί να το βελτιώσει καλύπτοντας τους πολλούς περιορισμούς που ισχύουν ταυτόχρονα. Είναι μια διαδικασία χρονοβόρα και κουραστική και ώσπου να δημιουργηθεί ένα αποδεκτό πρόγραμμα είναι δυνατό να περάσουν μερικές εβδομάδες. Σε πολλές περιπτώσεις και ενώ έχει αρχίσει να εφαρμόζεται το σχολικό πρόγραμμα είναι δυνατό να αλλάξουν ορισμένες συνθήκες όπως για παράδειγμα να μετακινηθούν ορισμένοι καθηγητές, να συγχωνευθούν ή να διααιρεθούν κάποια τμήματα μαθητών και να χρειάζεται να γίνουν ξανά αλλαγές στο πρόγραμμα επιδεινώνοντας ακόμη περισσότερο την όλη διαδικασία.

Τα τελευταία χρόνια έχουν γίνει αρκετές ερευνητικές εργασίες γύρω από αυτό το πρόβλημα και έχουν εμφανισθεί σχετικά προγράμματα που βοηθούν στη δημιουργία εβδομαδιαίων

ωρολόγιων προγραμμάτων. Οι συχνές αλλαγές όμως της εκπαιδευτικής νομοθεσίας και τις παράλληλες μεταβολές στα αναλυτικά προγράμματα που εφαρμόζονται στα σχολεία της δευτεροβάθμιας εκπαίδευσης επιδεινώνουν το πρόβλημα ακόμη περισσότερο καθιστώντας το βοηθητικό λογισμικό πολλές φορές λιγότερο χρήσιμο ή και ακατάλληλο. Επίσης στα πλαίσια της επαγγελματικής εκπαίδευσης υπάρχουν τύποι σχολείων όπου εφαρμόζονται εξειδικευμένοι περιορισμοί για παράδειγμα ειδικά μαθήματα γίνονται σε αντίστοιχα εργαστήρια, τα εργαστήρια αυτά είναι λίγα ή και μοναδικά ή τα ίδια εργαστήρια τα χρησιμοποιούν πολλά σχολεία, για την ασφάλεια των μαθητών πρέπει να υπάρχουν στα εργαστήρια ταυτόχρονα περισσότεροι εκπαιδευτικοί σχετικής ειδικότητας κτλ. Σ' αυτούς τους ειδικούς περιορισμούς θα πρέπει να προσθέσουμε και τα εσπερινά σχολεία ή τα ειδικά σχολεία που εφαρμόζουν ειδικά αναλυτικά προγράμματα σπουδών.

Από τα παραπάνω εύκολα γίνεται αντιληπτός ο μεγάλος βαθμός δυσκολίας και πολυπλοκότητας του προβλήματος της δημιουργίας εβδομαδιαίου ωρολόγιου προγράμματος για σχολεία της δευτεροβάθμιας εκπαίδευσης. Ενώ όσον αφορά την επαγγελματική και την ειδική εκπαίδευση, θα μπορούσαν μελλοντικές ερευνητικές εργασίες να αντιμετωπίσουν το πρόβλημα κατάλληλα κυρίως με την δημιουργία εξειδικευμένου λογισμικού. Το λογισμικό αυτό θα μπορούσε να δώσει αποδεκτά και προ πάντων βελτιωμένα προγράμματα που θα ικανοποιούσαν όλους τους περιορισμούς, σε σύντομο χρονικό διάστημα.

**Ορισμός του Προβλήματος:** Για την ανάπτυξη του εβδομαδιαίου ωρολόγιου προγράμματος πρέπει να τηρηθούν δύο κατηγορίες κανόνων. Η πρώτη κατηγορία περιλαμβάνει τους ανελαστικούς κανόνες και περιορισμούς που πρέπει πάντα να τηρούνται και καθορίζονται κυρίως από την εκπαιδευτική νομοθεσία. Η δεύτερη κατηγορία περιλαμβάνει τους κανόνες που καθορίζουν την ποιότητα του αποτελέσματος και μπορούν να χαρακτηρισθούν είτε ως ανελαστικοί κανόνες είτε να ενσωματωθούν στους παράγοντες που επηρεάζουν αρνητικά το αποτέλεσμα. Η δεύτερη κατηγορία κανόνων καθορίζεται συνήθως από το σύλλογο των διδασκόντων λόγω κάποιων ειδικών συνθηκών ή αντιπροσωπεύουν κανόνες που βελτιώνουν την εκπαιδευτική διαδικασία. [02]

Ο κατάλογος με τους κανόνες που ισχύουν σε γενικές γραμμές στα σχολεία της δευτεροβάθμιας εκπαίδευσης και θα πρέπει να ληφθεί υπόψη για την δημιουργία του προγράμματος είναι ο ακόλουθος: [19]

- i. Τα μαθήματα που αντιστοιχούν σε κάθε τάξη σύμφωνα με το αναλυτικό πρόγραμμα πρέπει να διδάσκονται από καθηγητές της αντίστοιχης ή παραπλήσιας ειδικότητας.
- ii. Σύμφωνα με το πρόγραμμα που θα προκύψει, κάθε καθηγητής πρέπει να αναλάβει τόσες ώρες είναι και το υποχρεωτικό ωράριο που πρέπει να καλύψει.
- iii. Κανένας καθηγητής δεν πρέπει να διδάσκει δυο μαθήματα σε δύο διαφορετικές τάξεις την ίδια διδακτική ώρα.
- iv. Σε ορισμένα μαθήματα κυρίως εργαστηριακά, ανάλογα με τον αριθμό των μαθητών, είναι απαραίτητο να παραβρίσκονται στο μάθημα περισσότεροι καθηγητές.
- v. Οι διδακτικές ώρες πρέπει να είναι συνεχόμενες και πάντα να ξεκινούν από την πρώτη ώρα. Δεν επιτρέπονται κενές ώρες ανάμεσα σε διδακτικές ώρες παρά μόνο στο τέλος του σχολικού ωραρίου. Το σύννηθες σχολικό ωράριο αντιστοιχεί σε επτά διδακτικές ώρες.
- vi. Κάθε εκπαιδευτικός πρέπει να έχει ένα όσο το δυνατόν ισορροπημένο πρόγραμμα εργασίας και κάθε μέρα εργασίας του στο σχολείο να έχει τουλάχιστον μία ώρα διδασκαλίας.
- vii. Ορισμένα μαθήματα κυρίως θεωρητικά που γίνονται σε περισσότερες από μια ώρες να μοιράζονται στις ημέρες του εβδομαδιαίου προγράμματος ενώ άλλα κυρίως εργαστηριακά τεχνικά μαθήματα να γίνονται σε συνεχόμενα δίωρα ή τρίωρα ανάλογα για την διευκόλυνση της εκπαιδευτικής δραστηριότητας.
- viii. Μαθήματα που γίνονται για μια ώρα την εβδομάδα να μην γίνονται την πρώτη ή την τελευταία ώρα της ημέρας.
- ix. Οι προτιμήσεις των καθηγητών σχετικά με τις ώρες διδασκαλίας που έχουν αν θα πρέπει να γίνονται στις πρώτες ώρες ή στις τελευταίες ώρες του ημερήσιου ωρολόγιου προγράμματος πρέπει να λαμβάνονται σοβαρά υπόψη.
- x. Να υπάρχουν όσο το δυνατόν λιγότερα κενά ανάμεσα στις διδακτικές ώρες των καθηγητών αν και θα μπορούσαν να εκμεταλλευτούν τα κενά με την παροχή διοικητικής εργασίας.

Οι παραπάνω κανόνες, από τον πρώτο (i) έως τον έβδομο (vii) είναι ανελαστικοί και θα πρέπει οπωσδήποτε να τηρούνται για να γίνει αποδεκτό ένα πρόγραμμα. Οι τρεις τελευταίοι κανόνες είναι ελαστικοί και καθορίζουν την ποιότητα του προγράμματος.



### **1.1.3 Χρονικός Προγραμματισμός σε Συστήματα Παραγωγής Κατά Παραγγελία (job shop)**

Χρονοπρογραμματισμός παραγωγής είναι η διαδικασία κατανομής πόρων κατά τη διάρκεια του χρόνου με σκοπό την ολοκλήρωση ενός συνόλου εργασιών. Ειδικότερα οι στόχοι που πρέπει να επιτευχθούν είναι η τήρηση των χρονικών προθεσμιών με την ελαχιστοποίηση το χρόνου υλοποίησης του προγράμματος, ελαχιστοποίηση του χρόνου που απασχολούνται οι μηχανές με την παράλληλη μεγιστοποίηση της αποδοτικότητάς τους. Ένα βελτιωμένο σύστημα παραγωγής ενισχύει την επιχειρηματική αποδοτικότητα, μειώνει τον χρόνο αδράνειας και προσφέρει προϊόντα και υπηρεσίες έγκαιρα στους καταναλωτές.

Τα συστήματα παραγωγής κατά παραγγελία (job shop) παράγουν προϊόντα για τα οποία οι ιδιότητες, οι προδιαγραφές όπως και οι ποσότητες που θα παραχθούν καθορίζονται από τους πελάτες και η κάθε παρτίδα παραγωγής είναι διαφορετική.

Το πρόβλημα του χρονικού προγραμματισμού γενικά σε συστήματα παραγωγής αποτελεί ένα μεγάλο πεδίο έρευνας και χαρακτηρίζεται από μεγάλο βαθμό πολυπλοκότητας όπου είναι πολύ δύσκολο και μερικές φορές αδύνατο να βρεθεί βέλτιστη λύση. Ο βαθμός πολυπλοκότητας αυξάνει εκθετικά όσο αυξάνεται ο αριθμός των μέσων που χρησιμοποιούνται για την παραγωγή, οι διαδικασίες παραγωγής, οι κανόνες παραγωγής ή τα κριτήρια για την αξιολόγηση της παραγωγής.

Ένα σύστημα παραγωγής κατά παραγγελία (job shop) αποτελείται από ένα σύνολο από μηχανές στις οποίες μπορεί να εκτελεσθεί ένα σύνολο από συγκεκριμένες λειτουργίες, ενώ κάθε εργασία είναι μια διατεταγμένη σειρά από λειτουργίες. Ο προγραμματισμός ενός τέτοιου συστήματος συνίσταται στην ανάθεση κάθε λειτουργίας του συνόλου των εργασιών που πρέπει να εκτελεσθούν σε συγκεκριμένη θέση της χρονικής κλίμακας σε μια συγκεκριμένη μηχανή με τέτοιο τρόπο ώστε να βελτιστοποιείται η απόδοση της παραγωγής [14]. Ένα τέτοιο σύστημα μπορεί να θεωρηθεί δυναμικό όταν καταφτάνουν κατά τη διαδικασία της παραγωγής συνεχώς νέες παραγγελίες οπότε το πλάνο της παραγωγής πρέπει να αναθεωρηθεί.

Σ' ένα δυναμικό σύστημα παραγωγής[07] πρέπει να ισχύουν κανόνες εκτέλεσης των παραγγελιών που θα ελαχιστοποιούν τον χρόνο παραγωγής και είναι διαφορετικοί από ένα μη δυναμικό σύστημα.

**Κανόνες προτεραιότητας:** Οι παραγγελίες μπορούν να εκτελούνται βάση προτεραιότητας, η οποία θα διαμορφώνεται για κάθε εργασία ανάλογα από τον Συνολικό Χρόνο Εκτέλεσης, τον Υπολειπόμενο Χρόνο Εκτέλεσης και Εκτέλεση Σύμφωνα με τη Σειρά Άφιξης.

**Περιορισμοί Παραγγελιών:** i. Όταν καταφτάνει μια νέα παραγγελία στο σύστημα κατευθείαν πηγαίνει στις μηχανές για την έναρξη της παραγωγής, ii. Κάθε παραγγελία έχει χαρακτηριστικά τα οποία είναι ανεξάρτητα από τα χαρακτηριστικά άλλων παραγγελιών, iii. Κάθε παραγγελία πρέπει να εκτελεστεί σε συγκεκριμένες μηχανές iv. Για την εκτέλεση κάθε συγκεκριμένης παραγγελίας απαιτείται ένα συγκεκριμένο χρονικό διάστημα. v. Μια παραγγελία μπορεί να αναστέλλει την εκτέλεσή της και να καθυστερεί όταν πρέπει να εκτελεστεί σε μια επόμενη μηχανή και αυτή η μηχανή είναι απασχολημένη.

**Περιορισμοί Μηχανών:** i. Ένα κέντρο μηχανών αποτελείται από μια ή περισσότερες πανομοιότυπες μηχανές. ii. Κάθε μηχανή στο σύστημα παραγωγής είναι ανεξάρτητη από τις άλλες μηχανές και έχει τον δικό της ρυθμό παραγωγής. iii. Κάθε μηχανή λειτουργεί συνεχώς και δεν επιτρέπονται διακοπές λόγω βλάβης ή για συντήρηση.

**Πολιτική Λειτουργίας:** i. Κάθε παραγγελία θεωρείται ως μια αδιαίρετη οντότητα έστω και αποτελείται από πολλά τμήματα. ii. Όταν μια παραγγελία γίνεται αποδεκτή από το σύστημα παραγωγής τότε πρέπει να εκτελεστεί και δεν επιτρέπεται η ακύρωσή της ή η αναβολή της. iii. Εάν μια εργασία ξεκινήσει να εκτελείται σε μια μηχανή δεν επιτρέπεται να σταματήσει για να εκτελεστεί στην ίδια μηχανή άλλη εργασία. iv. Η εκτέλεση μιας παραγγελίας γίνεται κάθε φορά σε μια μηχανή. v. Πριν από τις μηχανές υπάρχει περιθώριο να αναμένουν οι παραγγελίες την εκτέλεσή τους. vi. Μετά από την εκτέλεση μιας παραγγελίας σε μια μηχανή υπάρχει περιθώριο αναμονής.

#### 1.1.4 Χρονικός Προγραμματισμός Δρομολογίων Τρένων

Η δρομολόγηση των τρένων είναι το πρόβλημα σύμφωνα με το οποίο γίνεται ανάθεση των μέσων (γραμμών, σταθμών κλπ) που υπάρχουν σ' ένα σιδηροδρομικό δίκτυο, με σκοπό την ελαχιστοποίηση των καθυστερήσεων των τρένων και του αντίστοιχου κόστους καθυστέρησης που προκύπτει, ικανοποιώντας όλες τις απαιτήσεις δρομολόγησης. Η ικανοποίηση των κανόνων δρομολόγησης εξασφαλίζεται όταν τα τρένα χρησιμοποιούν σωστά τις γραμμές, τις συνδέσεις με άλλους προορισμούς, τους σταθμούς, τα σημεία αναμονής και τις διακλαδώσεις του δικτύου.[06]

Η διαχείριση των σιδηροδρομικών γραμμών αποκτά όλο και περισσότερη σημασία στα πλαίσια των ευρωπαϊκών συστημάτων μεταφορών. Οι λόγοι που συμβάλουν στην καλύτερη χρήση και στον αποτελεσματικό σχεδιασμό της σιδηροδρομικής υποδομής οφείλονται αφενός στο περιορισμένο σχετικά κεντρικό δίκτυο κίνησης των μέσων μεταφοράς σταθερής τροχιάς και αφετέρου στην μεγάλη πυκνότητα των δρομολογίων που εκτελούνται στην Κεντρική και Δυτική Ευρώπη κυρίως. Ενώ ο έντονος ανταγωνισμός που επικρατεί ανάμεσα στους φορείς που δρομολογούν τρένα συμβάλει ώστε η υφιστάμενη πίεση να αυξάνει συνέχεια και μελλοντικά η κατάσταση θα είναι ακόμη χειρότερη.

Η ανάπτυξη και χρήση αποτελεσματικών εργαλείων τα οποία βασίζονται στους υπολογιστές, συμβάλουν στην βελτίωση των σιδηροδρομικών μεταφορών και μπορούν να αντιμετωπίσουν τις μελλοντικές προκλήσεις που απορρέουν από τρεις βασικές αιτίες. Πρώτον, σύμφωνα με τις οδηγίες της Ευρωπαϊκής Ένωσης επιβάλλεται ο διαχωρισμός των δραστηριοτήτων των σιδηροδρομικών μεταφορών σε διαχειριστές υποδομής που θα είναι υπεύθυνοι για τον προγραμματισμό των δρομολογίων και τον έλεγχο της κίνησης σε πραγματικό χρόνο και σε φορείς που θα παρέχουν τις υπηρεσίες σιδηροδρομικών μεταφορών. Δεύτερο, ολοένα και περισσότερες ιδιωτικές εταιρείες παρέχουν υπηρεσίες μεταφορών εκμεταλλευόμενες το σιδηροδρομικό δίκτυο, αυξάνοντας έτσι τον ανταγωνισμό αλλά και τον συνωστισμό με την προσθήκη νέων δρομολογίων. Τρίτο, Ευρωπαϊκές Κυβερνήσεις αλλά και μέλη περιβαλλοντικών ομάδων πιέζουν ώστε οι μεταφορές να γίνονται μέσω του σιδηροδρόμου και όχι μέσω του οδικού δικτύου, εξαιτίας της μικτότερης επιβάρυνσης του περιβάλλοντος.

Αρκετές έρευνες έγιναν με σκοπό την ανάπτυξη αλγορίθμων οι οποίοι θα έδιναν τη δυνατότητα στους διαχειριστές της υποδομής να δημιουργούν πίνακες δρομολόγησης που θα επέτρεπαν στις εταιρείες σιδηροδρομικών μεταφορών που εκμεταλλεύονται τις σιδηροδρομικές μεταφορές και να δεσμεύουν τις υπάρχουσες σιδηροδρομικές γραμμές με έναν τρόπο βέλτιστο.

Ένας από τους αλγορίθμους που προέκυψαν ονομάζεται «Ικανότητα Διαχείρισης της Κυκλοφορίας» (Traffic Capacity Management – TCM)[04] και λειτουργεί ως εξής: Οι εταιρείες σιδηροδρομικών μεταφορών υποβάλλουν αιτήματα για διαδρομές που θέλουν να εκτελέσουν δεσμεύοντας συγκεκριμένες γραμμές προς τον φορέα της διαχείρισης της υποδομής και αυτός αφού συγκεντρώσει όλα τα αιτήματα, εκτελεί τον αλγόριθμο για τη βέλτιστη κατανομή των δρομολογίων με σκοπό να ικανοποιηθούν όλοι οι εμπλεκόμενοι. Στη συνέχεια δημιουργείται το πλάνο των δρομολογίων και διανέμεται στις εταιρείες ως πρόταση μαζί με τις αντίστοιχες χρεώσεις που θα προκύψουν από τη χρήση του σιδηροδρομικού δικτύου. Τα κυριότερα χαρακτηριστικά αυτής της διαδικασίας είναι τα εξής:

- Αρχικά στις εκάστοτε διαδρομές δίνεται ένας βαθμός, όπου στα πλαίσια ενός ιδανικού προγράμματος δρομολογίων αντιστοιχεί στις ιδανικές ώρες άφιξης και αναχώρησης μαζί με ένα χρονικό περιθώριο καθυστέρησης που μπορεί να γίνει ανεκτό.
- Για κάθε παράβαση του ιδανικού ωραρίου αφίξεων-αναχωρήσεων-καθυστερήσεων προκύπτει μια ανάλογη ποινή.
- Βέλτιστη κατανομή έχουμε όταν μεγιστοποιείται το άθροισμα που προκύπτει από τις διαφορές της αρχικής βαθμολογίας που αντιστοιχούσε στα ιδανικά δρομολόγια μείων τις ποινές που προκύπτουν από τις παραβάσεις των αρχικών ιδανικών προγραμμάτων δρομολόγησης.
- Κάποια μέρη από τις διαδρομές μπορούν να κλειδώσουν και να μην υπολογίζονται στη διαδικασία της δρομολόγησης είτε για λόγους προτεραιότητας είτε γιατί ανήκουν σε άλλες εταιρείες.
- Η διαδικασία του προγραμματισμού δρομολογίων πρέπει να λαμβάνει υπόψη τους κλασικούς περιορισμούς όπως είναι για παράδειγμα τα χρονικά κενά που πρέπει να υπάρχουν ανάμεσα στα δρομολόγια των τρένων, οι διαθέσιμες γραμμές στους σταθμούς ή ο τύπος των τρένων αλλά και τους περιορισμούς που προκύπτουν λόγω προγραμματισμένων εργασιών συντήρησης ή επισκευών για παράδειγμα και θα επηρεάσουν την εκτέλεση των δρομολογίων.
- Βλάβη στην αυτόματη σηματοδότηση για τη ρύθμιση της κυκλοφορίας είναι δυνατόν να υποβαθμίσει την εκτέλεση των προγραμματισμένων δρομολογίων και θα πρέπει σ' αυτές τις καταστάσεις να λογίζεται ότι θα χρησιμοποιηθεί ο χειροκίνητος τρόπος σηματοδότησης.

Ένας άλλος αλγόριθμος για τη αυτόματη δημιουργία του πίνακα δρομολογίων, ο ATTPS (Automatic Train Timetabling and Platforming System)[05] λαμβάνει σχετικά με το πρόβλημα, τα εξής ως είσοδο:

- Το ελάχιστο χρονικό κενό που πρέπει να υπάρχει ανάμεσα σε δύο τρένα μιας διαδρομής.
- Ο ελάχιστος χρόνος παραμονής ενός τρένου σ' ένα σταθμό.
- Η διάταξη του σταθμού, πλήθος παράλληλων γραμμών, θέσεις στάθμευσης τρένων, διακλαδώσεις γραμμών κτλ.
- Σχέδιο χρονοδιαγράμματος δρομολογίων. Μια λίστα με τα τρένα που θα ταξιδέψουν, όπου για κάθε τρένο θα υπάρχει ο τύπος του, οι γραμμές που θα χρησιμοποιήσει για τις αφίξεις και τις αναχωρήσεις και τους κατά προσέγγιση χρόνους που θα γίνουν οι αφίξεις και οι αναχωρήσεις.
- Οι πλατφόρμες φορτοεκφόρτωσης ή από-επιβίβασης που θα χρησιμοποιηθούν, ανάλογα θα υπολογιστεί και ένα κόστος.
- Το κόστος καθυστέρησης, ανάλογα με το τρένο και το δρομολόγιο αντιστοιχεί ένα ανάλογο κόστος για τυχόν καθυστερήσεις.

Αφού επεξεργαστούν τα δεδομένα εισόδου από τον αλγόριθμο ATTPS, παράγονται τα ακόλουθα αποτελέσματα:

- Πρόγραμμα με τα δρομολόγια των τρένων, ώρα αναχώρησης και αποβάθρα απ' όπου θα ξεκινήσει το κάθε τρένο. Το πρόγραμμα αυτό καλύπτει όλους τους περιορισμούς που τέθηκαν παραπάνω.
- Ανάλυση του προγράμματος που δόθηκε με πίνακες και διαγράμματα, όπου παρουσιάζονται οι αποκλίσεις από τους επιθυμητούς χρόνους άφιξης και αναχώρησης, στοιχεία για τις αλλαγές και τις ανταποκρίσεις των τρένων και στοιχεία για τη χρήση των αποβάθρων.

### **1.1.5 Χρονικός Προγραμματισμός Αθλητικών Δραστηριοτήτων**

Ο αθλητισμός στο σύνολό του έχει γίνει μια μεγάλη επιχείρηση, τις αθλητικές συναντήσεις τις παρακολουθεί ένας μεγάλος αριθμός θεατών σε όλο τον κόσμο και οι αθλητικοί σύλλογοι

πραγματοποιούν μεγάλες επενδύσεις για την ενίσχυση του έμφυλου δυναμικού, σε νέο εξοπλισμό, σε αθλητικά κέντρα κτλ.

Πολλές χώρες και πόλεις ανταγωνίζονται για την διοργάνωση αθλητικών εκδηλώσεων παγκόσμιας εμβέλειας όπως είναι οι Ολυμπιακοί Αγώνες, Παγκόσμια Πρωταθλήματα, Τελικοί Κυπέλων κτλ ενώ εθνικά επαγγελματικά πρωταθλήματα συγκεντρώνουν δραστηριότητες που αποτιμώνται σε πολλά χρήματα. Από την άλλη πλευρά τα ερασιτεχνικά πρωταθλήματα παρόλο που απαιτούν λιγότερες επενδύσεις, είναι αναγκαίος ο συντονισμός τους και η επίβλεψη των υλικοτεχνικών υποδομών τους λόγω του μεγάλου αριθμού των διοργανώσεων που πραγματοποιούνται και των πολλών συμμετεχόντων σ' αυτές.

Το βασικό πρόβλημα στον προγραμματισμό των αθλητικών εκδηλώσεων συνίσταται στον καθορισμό της ημερομηνίας, της ώρας και του χώρου που θα πραγματοποιηθεί το αθλητικό γεγονός [21].

Στον ομαδικό αθλητισμό οι αθλητικές συναντήσεις γίνονται μεταξύ των ομάδων κατά ζεύγη κατά τη διάρκεια μια χρονικής περιόδου σύμφωνα μ' ένα χρονοδιάγραμμα. Οι ομάδες που αγωνίζονται μεταξύ τους ανήκουν σε ένα πρωτάθλημα και συνήθως ο αριθμός τους είναι άρτιος αριθμός. Σε κάθε αγωνιστική ημέρα οι μισές ομάδες αγωνίζονται στην έδρα τους ενώ οι άλλες μισές είναι φιλοξενούμενοι και την επόμενη αγωνιστική εναλλάσσονται οι ρόλοι των γηπεδούχων με τους φιλοξενούμενους. Όταν όλες οι ομάδες αγωνισθούν μεταξύ τους μια φορά τότε ακολουθεί ο δεύτερος γύρος συναντήσεων όπου οι ομάδες πάλι αγωνίζονται μεταξύ τους με τη σειρά όπως στον πρώτο γύρο αλλά γίνεται εναλλαγή της έδρας. Οι αγωνιστικές συναντήσεις πρέπει να πραγματοποιούνται περιοδικά ανά τακτά χρονικά διαστήματα και πρέπει να αποφεύγονται τα διαστήματα όπου δεν πραγματοποιούνται αθλητικές συναντήσεις διότι έτσι ζημιώνονται οι ομάδες οικονομικά αλλά και αγωνιστικά[18].

Σε μερικές περιπτώσεις είναι δυνατόν να ισχύουν και άλλοι περιορισμοί πέρα από τους κλασικούς, με αποτέλεσμα οι φίλαθλοι να μπορούν να παρακολουθούν καλύτερα την διεξαγωγή των πρωταθλημάτων. Έτσι όταν δύο ομάδες έχουν ως έδρα την ίδια πόλη τότε είναι καλύτερο όταν η μια παίζει στην έδρα της, η άλλη ομάδα της ίδιας πόλης να αγωνίζεται μακριά ως φιλοξενούμενη και την επόμενη αγωνιστική να εναλλάσσονται. Οι περισσότερο δημοφιλείς ομάδες να αγωνίζονται σε διαφορετικές ημέρες ή ώρες και αυτό γιατί οι τηλεοπτικές μεταδόσεις θα έχουν περισσότερη τηλεθέαση.

## 1.2 Τεχνικές Επίλυσης Προβλημάτων

Παρακάτω παρουσιάζονται τρεις από τις πιο ενδεικτικές τεχνικές επίλυσης προβλημάτων που χρησιμοποιούνται διεθνώς από τους ειδικούς για την αντιμετώπιση κυρίως προβλημάτων χρονοπρογραμματισμού. Όπως τα προβλήματα χρονοπρογραμματισμού εξελίσσονται έτσι και οι τεχνικές επίλυσης πρέπει να εξελίσσονται και να παρακολουθούν εξελίξεις. Εάν δεν γίνεται αυτό τότε παύουν να είναι αποτελεσματικές και χρήσιμες και μοιραία αντικαθίστανται από άλλες περισσότερο αποτελεσματικές τεχνικές.

### 1.2.1 Δυναμικός Προγραμματισμός (Dynamic Programming)

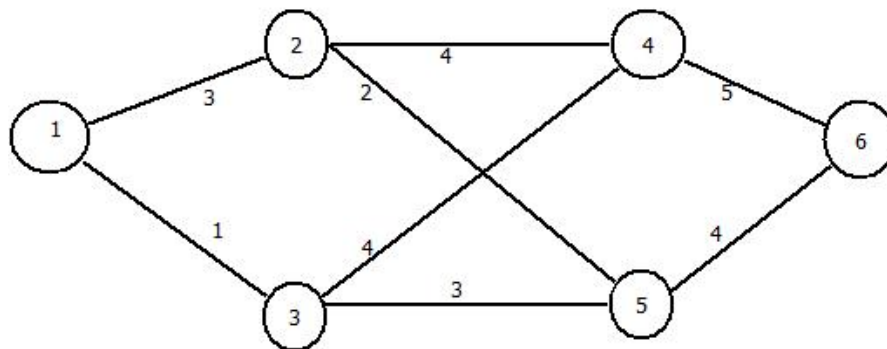
Ο όρος δυναμικός προγραμματισμός χρησιμοποιήθηκε αρχικά τη δεκαετία του 1940 από τον Richard Bellman για να περιγράψει τη διαδικασία επίλυσης προβλημάτων όπου μια σειρά από επιμέρους σωστές λύσεις οδηγούσαν στην καλύτερη λύση ενός προβλήματος. Ενώ το 1953 έδωσε έναν πιο εξευγενισμένο ορισμό με σύγχρονο νόημα λέγοντας ότι μικρότερα προβλήματα φωλιάζουν στο εσωτερικό μεγάλων αποφάσεων.

Ο αλγόριθμος του Δυναμικού Προγραμματισμού είναι μια μαθηματική μέθοδος βελτιστοποίησης αλλά και μια μέθοδος προγραμματισμού υπολογιστών. Αναφέρεται στην απλούστευση ενός πολύπλοκου προβλήματος με το σπάσιμό του σε απλούστερα, μικρά προβλήματα με επαναληπτικό τρόπο. Η βασική ιδέα του αλγορίθμου είναι ότι ένα βασικό πρόβλημα μπορεί να διασπαστεί σε μικρότερα προβλήματα, εντοπίζονται οι βέλτιστες λύσεις για αυτά τα επιμέρους προβλήματα και τέλος ο συνδυασμός αυτών των επιμέρους βέλτιστων λύσεων θα αποτελεί και βέλτιστη λύση για το συνολικό πρόβλημα. Δηλαδή ο δυναμικός προγραμματισμός εφαρμόζεται από τη βάση προς τη κορυφή (Bottom Up), επιλύοντας αρχικά μικρά προβλήματα και μετά συνθέτοντας τις βέλτιστες λύσεις που ήδη έχουν βρεθεί επιλύονται τα μεγαλύτερα ώσπου να λυθεί το αρχικό πρόβλημα.

Ο δυναμικός προγραμματισμός έχει εφαρμογές σε πολλούς τομείς της ανθρώπινης δραστηριότητας όπως σε προβλήματα κατανομής πόρων[16] σε ανταγωνιστικές

δραστηριότητες όπως είναι διαχείριση ανθρώπινου δυναμικού, σε προβλήματα πολλαπλών χρονικών περιόδων όπως είναι η διαχείριση αποθεμάτων και σε προβλήματα δικτύων όπως είναι η εύρεση συντομότερου μονοπατιού.

**Παράδειγμα:** Έστω ένα δίκτυο από πόλεις που τις αναπαριστούν οι κόμβοι που συνδέονται μεταξύ τους με δρόμους που αντιστοιχούν στις ακμές του δικτύου. Για κάθε δρόμο-ακμή υπάρχει η απόσταση που εκφράζεται με ένα θετικό ακέραιο για τους κόμβους πόλεις που ενώνει. Όπως φαίνεται στη παρακάτω εικόνα:

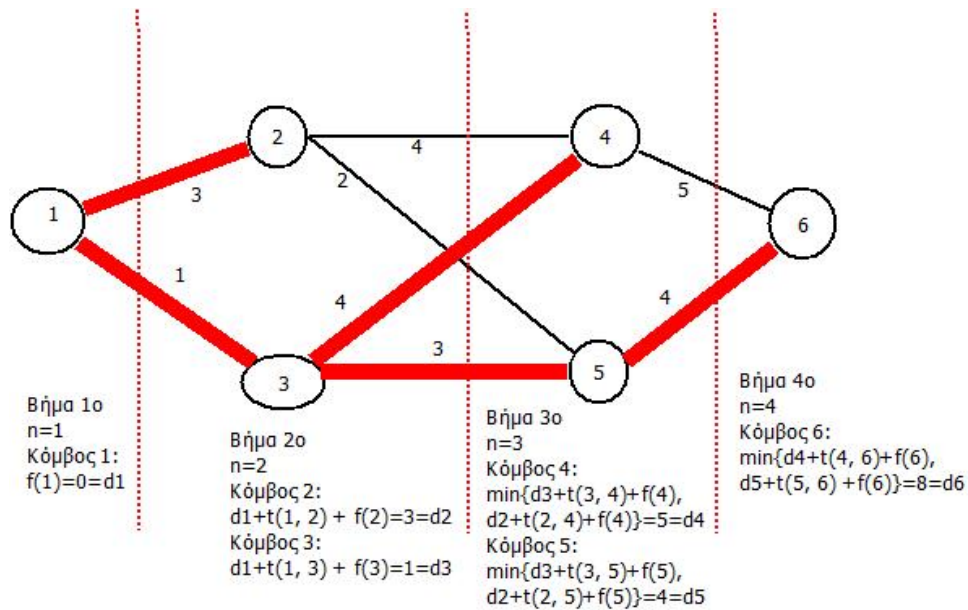


Εικόνα 1.1 Δίκτυο πόλεων

Ζητάμε την ελάχιστη διαδρομή από τον κόμβο 1 έως τον κόμβο 6. Υποθέτουμε σε κάθε κόμβο-πόλη υπάρχει μηδενική επιβάρυνση.

Για τον εντοπισμό της ελάχιστης διαδρομής από τον κόμβο 1 έως τον κόμβο 6 εργαζόμαστε[23] ως εξής:





Εικόνα 1.2: Εκτέλεση του αλγορίθμου σε κάθε στάδιο του προβλήματος

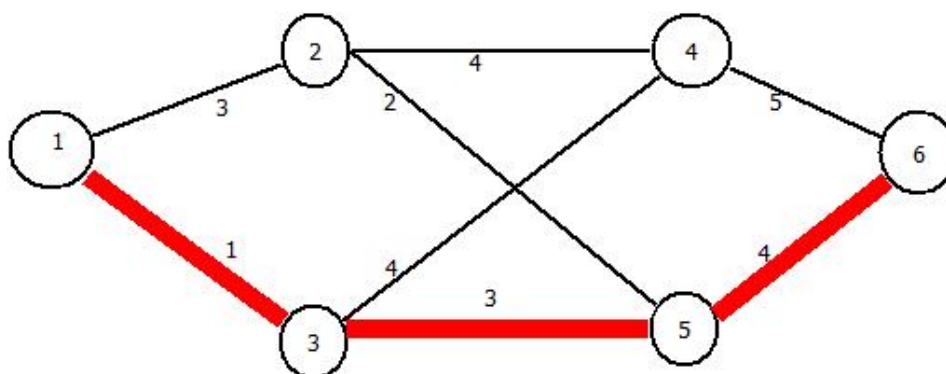
Το πρόβλημα χωρίζεται σε 4 μέρη και για κάθε μέρος υπολογίζουμε για τους κόμβους που βρίσκονται εκεί τη συντομότερη διαδρομή από τον κόμβο 1. Στο τέλος θα συνδυάσουμε τις βέλτιστες λύσεις από τα επιμέρους προβλήματα και θα προκύψει η βέλτιστη λύση για το συνολικό πρόβλημα. Έχουμε λοιπόν:

Βήμα 1<sup>ο</sup>: για την πόλη που βρίσκεται στον 1<sup>ο</sup> κόμβο και είναι η αφετηρία έχουμε μηδενική απόσταση, άρα  $f(1)=0=d_1$ .

Βήμα 2<sup>ο</sup>: Υπολογίζεται η συντομότερη διαδρομή από την πόλη-κόμβο 1 έως τις πόλεις-κόμβους 2 και 3. Για την πόλη 2 έχουμε  $d_1+t(1, 2)+f(2)=0+3+0=3=d_2$ . Για την πόλη 3 έχουμε  $d_1+t(1, 3)+f(3)=0+1+0=1=d_2$ , δηλαδή από την πόλη 1 το συντομότερο μονοπάτι προς τις πόλεις 2 και 3 έχει μήκος αντίστοιχα 3 και 1.

Βήμα 3<sup>ο</sup>: Υπολογίζεται η συντομότερη διαδρομή από την πόλη-κόμβο 1 έως τις πόλεις-κόμβους 4 και 5. Για την πόλη 4 έχουμε  $\min\{d_2+t(2, 4)+f(4), d_3+t(3, 4)+f(4)\}=5=d_4$ . Για την πόλη 5 έχουμε  $\min\{d_2+t(2, 5)+f(5), d_3+t(3, 5)+f(5)\}=4=d_5$ , δηλαδή από την πόλη 1 το συντομότερο μονοπάτι προς τις πόλεις 4 και 5 έχει μήκος αντίστοιχα 3 και 4.

Βήμα 4<sup>ο</sup>: Υπολογίζεται η συντομότερη διαδρομή από την πόλη-κόμβο 1 έως την πόλη-κόμβο 6 που είναι και πόλη προορισμός. Έχουμε  $\min\{d_4+t(4, 6)+f(6) , d_5+t(5, 6)+f(6)\}=8=d_6$ . Έτσι λοιπόν το συντομότερο μονοπάτι έχει απόσταση 8 και είναι η διαδρομή που περνά από τις πόλεις 1-3-5-6, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 1.3: Η συντομότερη διαδρομή από τον κόμβο 1 προς τον κόμβο 6

### 1.2.2 Προγραμματισμός με Περιορισμούς (Constraint Programming)

Ο προγραμματισμός με περιορισμούς έχει προσελκύσει τα τελευταία χρόνια την προσοχή μεγάλου αριθμού ερευνητών από πολλές επιστημονικές περιοχές λόγω της δυνατότητας που έχει στην επίλυση δύσκολων προβλημάτων που καλύπτουν πολλές πτυχές της ανθρώπινης δραστηριότητας. Δεν παρέχει απλά ένα ισχυρό θεωρητικό υπόβαθρο, αλλά χρησιμοποιείται ευρέως στον πραγματικό κόσμο και κυρίως για την μοντελοποίηση και την βελτιστοποίηση ετερογενών προβλημάτων, τα οποία απαιτούν την ικανοποίηση συγκεκριμένων παραμέτρων.

Από την άλλη πλευρά ο προγραμματισμός με περιορισμούς εξακολουθεί να είναι από τις λιγότερο γνωστές και κατανοητές τεχνολογίες. Οι περιορισμοί υπάρχουν σχεδόν σε όλους τους τομείς της ανθρώπινης δραστηριότητας, τυποποιούν τις εξαρτήσεις που υπάρχουν στον φυσικό κόσμο και τις μαθηματικές τους παραστάσεις με τρόπο φυσικό και διάφανο. Ένας περιορισμός είναι απλά μια λογική σχέση μεταξύ διαφόρων μεταβλητών που η κάθε μια παίρνει μια τιμή από

ένα συγκεκριμένο πεδίο ορισμού. Ένας περιορισμός μπορεί να περιορίσει τις τιμές που μπορούν να πάρουν κάποιες μεταβλητές παρέχοντας έτσι πληροφόρηση σχετικά με τις μεταβλητές που ερευνώνται. Οι περιορισμοί επίσης μπορεί να είναι ετερογενείς και να συσχετίζουν έτσι μεταβλητές μεταξύ τους με διαφορετικά πεδία ορισμού, για παράδειγμα, ένας περιορισμός θα μπορούσε να συνδέσει δύο μεταβλητές που η μια δέχεται αριθμούς και η άλλη αλφαριθμητικά. Το σημαντικότερο χαρακτηριστικό ενός περιορισμού είναι ο δηλωτικός του χαρακτήρας χωρίς να καθορίζεται μια διαδικασία που θα επιβάλλει αυτόν τον περιορισμό.

Ο προγραμματισμός με περιορισμούς[01] είναι η μελέτη των πληροφοριακών συστημάτων μέσω κάποιων περιορισμών και ως βασική ιδέα για την λύση των προβλημάτων έχει την καταγραφή των περιορισμών, που στην ουσία περιγράφουν το πρόβλημα και στη συνέχεια εξεύρεση λύσης που θα ικανοποιεί όλους τους περιορισμούς που τέθηκαν.

Ο προγραμματισμός με περιορισμούς διαχωρίζεται σε δύο κλάδους, στην **Ικανοποίηση των Περιορισμών (constraint satisfaction)** και στην **Επίλυση των Περιορισμών (constraint solving)**. Αν και οι δύο μοιράζονται την ίδια ορολογία εντούτοις είναι διαφορετική η προέλευσή τους και η τεχνολογία που χρησιμοποιείται για την εξεύρεση των λύσεων. Ο πρώτος κλάδος (constraint satisfaction) ασχολείται με προβλήματα που χρησιμοποιούν πεπερασμένα πεδία ορισμού και αυτό συμβαίνει στη συντριπτική πλειοψηφία των βιομηχανικών εφαρμογών. Ο δεύτερος κλάδος (constraint solving) περιγράφει και αυτός τα προβλήματα ως σύνολα με περιορισμούς και επιλύει τους περιορισμούς, αλλά αυτοί οι περιορισμοί ορίζονται κυρίως στο άπειρο ή σε πιο σύνθετες περιοχές. Ο δεύτερος κλάδος αντί για συνδυαστικές μεθόδους για την επίλυση των περιορισμών χρησιμοποιεί μαθηματικές τεχνικές όπως είναι τα αυτόματα διαφορικά (automatic differentiation), οι σειρές Taylor ή η μέθοδος Newton.

**Προβλήματα Ικανοποίησης Περιορισμών (Constraint Satisfaction Problems)** είναι προβλήματα αναζήτησης που έχουν απλή δομή και επιδέχονται μια απλή τυπική αναπαράσταση. Για αυτά τα προβλήματα υπάρχουν αλγόριθμοι αναζήτησης που εκμεταλλεύονται αυτή την απλή αναπαράσταση και χρησιμοποιώντας γενικούς ευρετικούς μηχανισμούς επιτυγχάνουν την επίλυση μεγάλων προβλημάτων.

**Ικανοποίηση Περιορισμών:** Ένα Πρόβλημα Ικανοποίησης Περιορισμών ορίζεται ως εξής:

Ένα σύνολο μεταβλητών  $X = \{x_1, x_2, \dots, x_n\}$

Για κάθε μεταβλητή  $x_i$  ένα πεδίο ορισμού  $D_i$

Ένα σύνολο από περιορισμούς, οι οποίοι καθορίζουν τις τιμές που μπορούν να πάρουν οι μεταβλητές ταυτόχρονα.

Λύση για αυτά τα προβλήματα μπορεί να βρεθεί με την ανάθεση μιας τιμής από το πεδίο ορισμού της κάθε μεταβλητής, με τέτοιο τρόπο ώστε όλοι οι περιορισμοί να ικανοποιούνται ταυτόχρονα. Μ' αυτόν τον τρόπο μπορούν να βρεθούν μια οποιαδήποτε λύση, όλες οι λύσεις ή η βέλτιστη λύση ή έστω μια καλή λύση η οποία θα ικανοποιεί μια αντικειμενική συνάρτηση που ορίζεται σε σχέση με κάποιες ή όλες τις μεταβλητές.

**Παράδειγμα[26]:** Έστω ότι πρέπει να εισαχθούν τέσσερα υλικά A, B, Γ και Δ μέσα σε ένα βιομηχανικό μύλο για την παραγωγή ενός τροφίμου. Λόγω της φύσης του προβλήματος η σειρά με την οποία θα εισαχθούν τα υλικά θα είναι η εξής: το υλικό A πρέπει να εισαχθεί μετά από το Δ, το Γ πριν από το B, και το B πριν από το A.

Μεταβλητές: A, B, Γ, Δ. Πεδία Ορισμού:  $D_A = D_B = D_\Gamma = D_\Delta = \{1,2,3,4\}$

Περιορισμοί:  $A \neq B, A \neq \Gamma, A \neq \Delta, B \neq \Gamma, B \neq \Delta, \Gamma \neq \Delta \rightarrow$  κανένα υλικό δεν θα εισαχθεί στο μύλο μαζί με άλλο ταυτόχρονα.

$A > \Delta$  Το υλικό A μετά το Δ

$\Gamma < B$  Το υλικό Γ πριν από το B

$B < A$  Το υλικό B πριν από το A

Το πρόβλημα έχει τρεις λύσεις, τις ακόλουθες:

$A=4, B=2, \Gamma=1, \Delta=3$ , δηλαδή η σειρά είναι: Γ, B, Δ, A

$A=4, B=3, \Gamma=1, \Delta=2$ , δηλαδή η σειρά είναι: Γ, Δ, B, A

$A=4, B=3, \Gamma=2, \Delta=1$ , δηλαδή η σειρά είναι: Δ, Γ, B, A

## **Επίλυση των Περιορισμών:**

Για να φτάσει κάποιος σε λύση που να ικανοποιεί όλους τους περιορισμούς μπορεί να εργαστεί ακολουθώντας τους παρακάτω αλγορίθμους οι οποίοι έχουν κατηγοριοποιηθεί ανάλογα με το πώς εργάζονται.

**Συστηματική Αναζήτηση:** Μπορούμε να φτάσουμε σε μια λύση εάν δοκιμάζουμε κάθε φορά μια τιμή σε μια μεταβλητή από το πεδίο ορισμού και ελέγχουμε εάν παραβιάζεται κάποιος περιορισμός. Η Συστηματική Αναζήτηση επιτυγχάνεται με δύο αλγορίθμους. Ο πρώτος ονομάζεται Γέννα και Δοκίμαζε (Generate and Test – GT) και δημιουργεί συστηματικά συνδυασμούς τιμών για τις μεταβλητές και δοκιμάζει εάν ικανοποιούν τους περιορισμούς. Ο δεύτερος αλγόριθμος ονομάζεται Οπισθοδρόμηση (Backtracking – BT) ο οποίος δίνει συστηματικά τιμές σε μεταβλητές και ελέγχει εάν ισχύουν οι περιορισμοί, εάν δεν ικανοποιείται κάποιος περιορισμός οπισθοδρομεί στη τελευταία ανάθεση που ικανοποιούσε τους περιορισμούς και δίνει άλλη τιμή. Αν εξαντληθούν οι τιμές μιας μεταβλητής οπισθοδρομεί και κάνει ανάθεση τιμών σε άλλη μεταβλητή. Μεγάλος μειονέκτημα τους είναι ο εκθετικός χρόνος αναζήτησης.

**Στοχαστικοί και Ευρετικοί Αλγόριθμοι:** Τέτοιοι αλγόριθμοι είναι ο αλγόριθμος Αναρρίχησης Λόφου (Hill Climbing), ο Ευριστικός αλγόριθμος των Ελαχίστων Συγκρούσεων (Min Conflicts Heuristic) και Αναζήτηση Tabu (Tabu Search – TS). Αυτοί οι αλγόριθμοι βελτιώνουν την διαδικασία αναζήτησης τιμών για τις μεταβλητές. Αρχικά επιλέγουν κάποιες τιμές για τις μεταβλητές από το πεδίο ορισμού τους, τις δοκιμάζουν και επιλέγουν τις τιμές για τις οποίες η εκάστοτε μεταβλητή παραβιάζει τους λιγότερους περιορισμούς και σταματά η εκτέλεσή τους όταν δεν παραβιάζεται κανείς περιορισμός.

**Έλεγχος Συνέπειας:** Η βασική ιδέα των αλγορίθμων αυτής της κατηγορίας είναι ότι γίνεται απαλοιφή από τα αρχικά πεδία τιμών των μεταβλητών εκείνων των τιμών οι οποίες δεν μπορούν να συμμετέχουν στην τελική λύση διότι παραβιάζουν κάποιους περιορισμούς και έτσι μειώνονται κατά πολύ οι έλεγχοι που απομένουν να γίνουν. Τέτοιοι αλγόριθμοι είναι ο Γράφος Περιορισμών (Constraint Graph), Συνέπειας Κόμβου (Node Consistency – NC), Συνέπειας Τόξου (Arc Consistency – AC), Συνέπειας Μονοπατιού (Path Consistency – PC).

**Διάδοση Περιορισμών:** Οι αλγόριθμοι αυτής της κατηγορίας ελέγχουν την συνέπεια των τιμών που θα αποδοθούν στις μεταβλητές πριν αρχίσει η αναζήτηση και οι τιμές που παραβιάζουν τους

περιορισμούς δεν χρησιμοποιούνται. Αλγόριθμοι που εργάζονται με αυτόν τον τρόπο είναι ο Προς τα Εμπρός Έλεγχος (Forward Checking – FC), Έγκαιρης μερικής Εξέτασης (Partial Look Ahead – PLA), Έγκαιρης Πλήρους Εξέτασης (Full Look Ahead – FLA), Διατήρησης Συνέπειας Τόξου (Maintaining Arc Consistency – MAC).

Ο **Λογικός Προγραμματισμός** (Logic Programming – LP)[01] είναι ένα ιδιαίτερο κομμάτι του προγραμματισμού με περιορισμούς και η βασική του ιδέα είναι ότι ο χρήστης δηλώνει το ΤΙ θα πρέπει να λυθεί και όχι το ΠΩΣ θα το λύσει. Ο συνδυασμός των περιορισμών και του λογικού προγραμματισμού είναι πολύ φυσικός και το αποτέλεσμα αυτού του συνδυασμού είναι ο **Λογικός Προγραμματισμός με Περιορισμούς** (Constraint Logic Programming - CLP), ένα περιβάλλον όπου δηλώνονται οι περιορισμοί με σκοπό να λυθούν τα προβλήματα μέσω αυτών των περιορισμών.

Ο προγραμματισμός με περιορισμούς έχει εφαρμοστεί με επιτυχία σε μια μεγάλη ομάδα προβλημάτων όπως είναι ο χρονοπρογραμματισμός και ανάθεση προσωπικού σε εργασίες, προγραμματισμός βιομηχανικής παραγωγής, διαχείριση δικτύων, ανάθεση πόρων και προγραμματισμού δρομολογίων.

### **1.2.3 Τοπική Αναζήτηση (Local Search)**

Σε πολλά προβλήματα αυτό που ενδιαφέρει κυρίως είναι η εύρεση μιας λύσης και όχι η διαδρομή που ακολουθήθηκε έως την εξεύρεση της λύσης. Αυτή η κλάση προβλημάτων περιλαμβάνει πολλές και σημαντικές εφαρμογές όπως είναι ο χρονοπρογραμματισμός παραγωγικής διαδικασίας (Job Shop Scheduling), διάταξη χώρου παραγωγής, δίκτυα μεταφορών, δίκτυα τηλεπικοινωνιών και η διαχείριση χαρτοφυλακίων.

Οι αλγόριθμοι της τοπικής αναζήτησης[22] λειτουργούν χρησιμοποιώντας μια τρέχουσα κατάσταση σ' ένα πρόβλημα και προσπαθούν να την βελτιώσουν εξετάζοντας γειτονικές σε αυτήν καταστάσεις. Εκτός από το να βρεθεί μια λύση οι αλγόριθμοι αυτοί είναι καλοί και στην επίλυση προβλημάτων βελτιστοποίησης στα οποία ο σκοπός είναι να βρεθεί η καλύτερη λύση σύμφωνα με μια αντικειμενική συνάρτηση. Οι αλγόριθμοι της τοπικής αναζήτησης

χρησιμοποιούν την τεχνική της επαναληπτικής βελτίωσης και η λειτουργία τους βασίζεται στα ακόλουθα βήματα:

- Επιλέγεται μια λύση και αφού αποτιμηθεί ονομάζεται τρέχουσα λύση.
- Εφαρμόζεται ένας μετασχηματισμός στη τρέχουσα λύση και παράγεται μια νέα λύση.
- Εάν η νέα λύση είναι καλύτερη από την τρέχουσα τότε η νέα λύση γίνεται τρέχουσα διαφορετικά απορρίπτεται.
- Επαναλαμβάνονται τα προηγούμενα βήματα έως ότου κανένας μετασχηματισμός να μην μπορεί να βελτιώσει άλλο την τρέχουσα λύση.

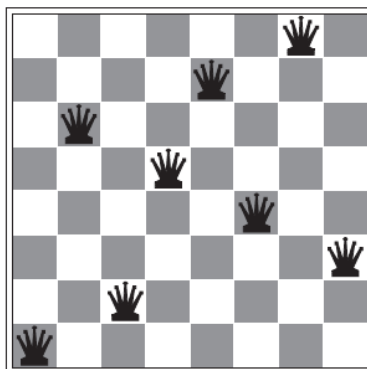
**Αναζήτηση με Αναρρίχηση Λόφων:** είναι ο απλούστερος αλγόριθμος τοπικής αναζήτησης, είναι ένας βρόγχος που μετακινείται συνεχώς προς την κατεύθυνση της αυξανόμενης τιμής – όταν αναζητάμε μέγιστη τιμή – ενώ τερματίζει ο αλγόριθμος όταν κανένας γειτονικός κόμβος δεν έχει μεγαλύτερη τιμή. Πολύ γρήγορα εντοπίζει μια λύση αν και το πιο πιθανό είναι να βρει ένα τοπικό μέγιστο ή ελάχιστο παρά ένα καθολικό μέγιστο ή ελάχιστο.

**Παράδειγμα:** Έστω το κλασικό πρόβλημα των 8 βασίλισσών, όπου σε μια σκακιέρα 8x8 είναι τοποθετημένες 8 βασίλισσες τοποθετημένες αρχικά τυχαία, μία σε κάθε στήλη, θέλουμε με τις κατάλληλες μετακινήσεις βασίλισσών να βρεθούμε σε κατάσταση όπου καμιά βασίλισσα δεν θα απειλείται από κάποια άλλη, τότε θα υπάρχει βέλτιστη κατάσταση για το συγκεκριμένο πρόβλημα.

18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	♣	13	16	13	16
♣	14	17	15	♣	14	16	16
17	♣	16	18	15	♣	15	♣
18	14	♣	15	15	14	♣	16
14	14	13	17	12	14	12	18

Εικόνα 1.4: 8 Βασίλισσες τοποθετημένες τυχαία πάνω στη σκακιέρα

Στην παραπάνω εικόνα, στο κάθε τετράγωνο φαίνονται οι απειλές που αντιστοιχούν αν μετακινούνταν μια βασίλισσα, της ίδιας στήλης, στη συγκεκριμένη θέση. Στην παρούσα φάση η αντικειμενική συνάρτηση θα μας δείχνει πόσα ζεύγη βασιλισσών αλληλοαπειλούνται και είναι ίσο με 17. Μετακινώντας τις βασίλισσες της κάθε στήλης σε τετράγωνο της ίδιας στήλης αν υπάρχει με λιγότερες απειλές από αυτές τις αντικειμενικής συνάρτησης και αναπροσαρμόζοντας κάθε φορά και την τιμή της αντικειμενικής συνάρτησης, θα μπορούσαμε να φτάσουμε στην κατάσταση που φαίνεται στην παρακάτω εικόνα.



Εικόνα 1.5: Αντικειμενική συνάρτηση ίσον με 1

Στην προηγούμενη εικόνα η αντικειμενική συνάρτηση είναι ίση με 1 δηλαδή υπάρχει ένα ζεύγος από βασίλισσες που απειλούν η μία την άλλη. Από την προηγούμενη διαδικασία είναι φανερό ότι έχει βρεθεί ένα τοπικό ελάχιστο και όχι το καθολικό ελάχιστο που θα συνέβαινε εάν η αντικειμενική συνάρτηση είχε τιμή 0. Ο αλγόριθμος της αναζήτησης με αναρρίχηση λόφων είναι πιο πιθανό να εντοπίσει τοπικό ελάχιστο-μέγιστο παρά καθολικό ελάχιστο-μέγιστο. Στο συγκεκριμένο παράδειγμα στο 86% των εκτελέσεων του αλγορίθμου εντοπίζεται τοπικό ελάχιστο και στο 14% καθολικό ελάχιστο με αντικειμενική συνάρτηση ίση με 0. Είναι όμως ταχύτατος αφού με 4 κινήσεις εντοπίζει το καθολικό ελάχιστο και με 3 παγιδεύεται σε τοπικό ελάχιστο.

Μια παραλλαγή του προηγούμενου αλγορίθμου, από τις πολλές που έχουν προταθεί, είναι η **αναρρίχηση των λόφων με τυχαίες επανεκκινήσεις**, επαναλαμβάνει την εκτέλεση του αλγορίθμου ξεκινώντας κάθε φορά με διαφορετικές αρχικές τιμές, έως ότου εντοπιστεί το καθολικό ελάχιστο ή μέγιστο. Στο παράδειγμα με τις βασίλισσες θα αρκούσαν 7



επαναλαμβανόμενες εκτελέσεις (με 14% πιθανότητα επιτυχίας κάθε φορά) για να εντοπιστεί το καθολικό ελάχιστο.

**Αναζήτηση με Προσομοιωμένη Ανόπτηση:** Ένας αλγόριθμος αναζήτησης με αναρρίχηση των λόφων επιλέγει πάντα κινήσεις για την επίλυση προβλημάτων – εάν υπάρχουν - αυτές που δίνουν μια καλύτερη λύση από μια προηγούμενη και έτσι παγιδεύεται σε τοπικά μέγιστα-ελάχιστα. Στην μεταλλουργία ανόπτηση είναι η διαδικασία σκλήρυνσης των μετάλλων, ενώ αρχικά θερμαίνονται και είναι πιο εύπλαστα, στη συνέχεια ψύχονται για να γίνουν σκληρότερα. Στην αναζήτηση με προσομοιωμένη ανόπτηση, σε κάθε επανάληψη, επιλέγεται μια τυχαία κίνηση και αν βελτιώνει την κατάσταση τότε η κίνηση γίνεται αποδεκτή, διαφορετικά η κίνηση γίνεται αποδεκτή με μια πιθανότητα - μικρότερη του 1 -. Η πιθανότητα μειώνεται εκθετικά ανάλογα με το πόσο κακή είναι η κίνηση, επίσης η πιθανότητα επιλογής μειώνεται ακόμη περισσότερο ανάλογα με τον αριθμό των κινήσεων που έχουν γίνει (ή πιο σωστά καθώς κατεβαίνει η «θερμοκρασία»). Δηλαδή οι «κακές» κινήσεις είναι πιο πιθανό να επιλεγούν στην αρχή.

**Τοπική Ακτινική Αναζήτηση:** Ο αλγόριθμος αυτός αντί να παρακολουθεί μια κατάσταση, παρακολουθεί ένα συγκεκριμένο αριθμό καταστάσεων. Αρχικά επιλέγονται οι καταστάσεις που θα παρακολουθούνται και σε κάθε βήμα παράγονται οι διάδοχες καταστάσεις. Αν σε κάποια διάδοχη κατάσταση εντοπιστεί ολικό μέγιστο-ελάχιστο ο αλγόριθμος σταματά διαφορετικά επιλέγονται οι καλύτερες διάδοχες καταστάσεις και ο αλγόριθμος συνεχίζει με την παραγωγή νέων διαδόχων καταστάσεων. Σε κάθε βήμα γίνεται ανταλλαγή πληροφοριών μεταξύ των καταστάσεων που εξετάζονται με αποτέλεσμα η έρευνα να επικεντρώνεται στις καταστάσεις που παρουσιάζουν καλύτερη πρόοδο.

**Γενετικός Αλγόριθμος:** Στον αλγόριθμο αυτό επιλέγονται αρχικά κάποιες καταστάσεις ή άτομα που ονομάζονται πληθυσμός, κάθε κατάσταση βαθμολογείται σύμφωνα με μια συνάρτηση αξιολόγησης, εάν δεν προκύψει η βέλτιστη κατάσταση-άτομο τότε απορρίπτονται οι χειρότερες καταστάσεις-άτομα και αντικαθίστανται τυχαία από αντίγραφα καλύτερων καταστάσεων-ατόμων και με τον τυχαίο συνδυασμό δύο γονικών καταστάσεων-ατόμων προκύπτουν οι διάδοχες καταστάσεις-άτομα και ο αλγόριθμος επαναλαμβάνεται μέχρι τον εντοπισμό της βέλτιστης κατάστασης-ατόμου. Ο αλγόριθμος αυτός εξετάζεται λεπτομερώς στο δεύτερο κεφάλαιο της παρούσας διατριβής και είναι ο αλγόριθμος που θα χρησιμοποιηθεί για την δημιουργία εξεταστικού ημερολογίου στα επόμενα κεφάλαια.

# Κεφάλαιο 2

## Γενετικοί Αλγόριθμοι

Οι Γενετικοί Αλγόριθμοι εμφανίσθηκαν για πρώτη φορά στις αρχές της δεκαετίας του 1950 από βιολόγους οι οποίοι θέλησαν να προσομοιώσουν με τη χρήση υπολογιστών συστήματα βιολογίας. Ενώ το 1975 ο John Holland με τους φοιτητές του ανέπτυξε τους Γενετικούς Αλγορίθμους με τη σημερινή τους μορφή. Ο αλγόριθμος που διατύπωσε ο Holland μμείται τη φύση για την επίλυση προβλημάτων Τεχνικής Νοημοσύνης.

Τα τελευταία χρόνια γίνεται μεγάλος αριθμός προσπαθειών για την ανάπτυξη μεθόδων βελτιστοποίησης που βασίζονται στους Γενετικούς Αλγόριθμους αλλά και στις άλλες εξελικτικές στρατηγικές όπως είναι ο Γενετικός Προγραμματισμός, οι Εξελικτικοί Αλγόριθμοι κτλ. Αυτό οφείλεται κυρίως στις αδυναμίες που έχουν οι κλασικοί μέθοδοι αναζήτησης και βελτιστοποίησης, αλλά και στη ραγδαία εξέλιξη των υπολογιστικών συστημάτων που έχει ως αποτέλεσμα την ανάπτυξη λογισμικού που μπορεί να εκμεταλλευτεί πλήρως τα πλεονεκτήματα των Γενετικών Αλγορίθμων.

Η ανθρωπότητα καθόλη τη διάρκεια της εξέλιξής της προσπαθεί να πετύχει την τελειότητα σε πολλούς τομείς. Βασική επιδίωξη είναι η επίτευξη της μέγιστης επιτυχίας με την ελάχιστη δυνατή προσπάθεια, κάνοντας την βελτιστοποίηση παρούσα σε κάθε πτυχή της καθημερινότητας.

## 2.1 Θεωρία της Εξέλιξης των Ειδών

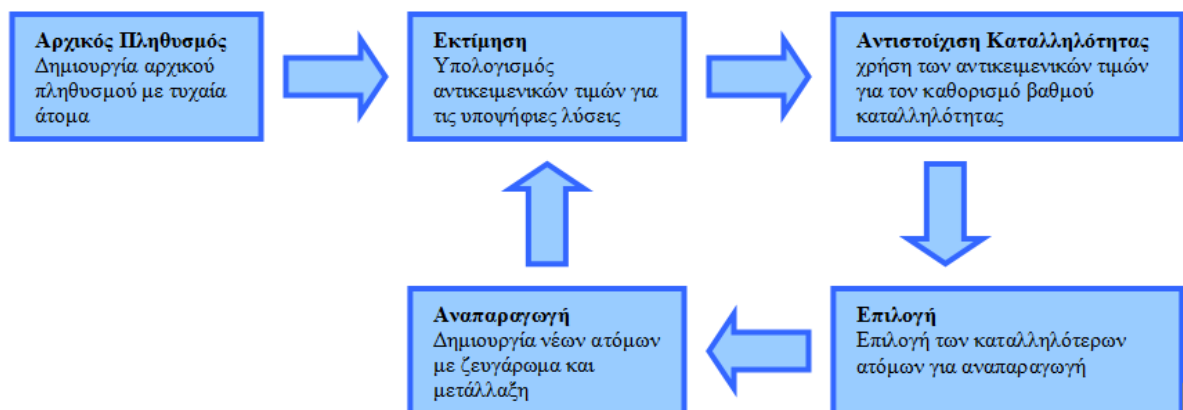
Οι Γενετικοί Αλγόριθμοι στηρίζονται στην Αρχή της Εξέλιξης των Ειδών που διατυπώθηκε από τον Δαρβίνο στα μέσα του 19<sup>ου</sup> αιώνα. Τα βασικά στοιχεία της Αρχής της εξέλιξης των Ειδών είναι:

1. Δεν υπάρχει κάποιος αντικειμενικός κανόνας που να διαχωρίζει τους οργανισμούς ενός βιολογικού είδους σε ανώτερους και κατώτερους. Το περιβάλλον και οι συνθήκες διαβίωσης συμβάλουν στην αλλαγή των χαρακτηριστικών των οργανισμών με κύριο σκοπό την επιβίωσή τους αλλά και την διαιώνιση του είδους τους. Έτσι βελτίωση των συνθηκών διαβίωσης αυξάνει τις πιθανότητες αύξησης του αριθμού των απογόνων οι οποίοι θα κληρονομήσουν τα χαρακτηριστικά των γονιών τους. Αυτά τα χαρακτηριστικά είναι περισσότερο πιθανό θα επικρατήσουν στις επόμενες γενεές έναντι των χαρακτηριστικών των άλλων μελών του ίδιου είδους που δεν ευνοήθηκαν από τις συνθήκες διαβίωσης.
2. Τα χαρακτηριστικά των μελών ενός είδους κωδικοποιούνται σε πολύπλοκα οργανικά μόρια τα οποία ονομάζονται Χρωμοσώματα (chromosomes). Τα χρωμοσώματα αποτελούνται από μικρότερα μέρη τα γονίδια (genes). Το σύνολο της γενετικής πληροφορία που είναι αποθηκευμένο στα γονίδια ονομάζεται γενότυπος (genotype). Τα χαρακτηριστικά ενός οργανισμού κληρονομούνται με την αποκωδικοποίηση του γενότυπου. Το σύνολο των χαρακτηριστικών ενός οργανισμού που είναι ορατά και της συμπεριφορά του, που καθορίζονται από το γενότυπο, συνιστούν τον φαινότυπο (phenotype).
3. Βασικές λειτουργίες για την διαιώνιση των ειδών είναι η αναπαραγωγή (reproduction) και η μετάλλαξη (mutation). Κατά τη διαδικασία της αναπαραγωγής συνήθως δύο μέλη ενός οργανισμού, μέσω κάποιων ανταγωνιστικών διαδικασιών επιλέγονται και

ανταλλάσσουν γενετικό υλικό. Οι απόγονοι κληρονομούν τα χαρακτηριστικά των γονέων τους ή κάποιων άλλων προγόνων τους. Η μετάλλαξη συμβαίνει σε πολύ αραιά χρονικά διαστήματα και οφείλεται σε ειδικές περιβαλλοντικές συνθήκες ή σε λόγους που επηρεάζουν την επιβίωση ενός είδους. Για παράδειγμα η εμφάνιση ενός γρήγορου θηρευτή σε μια περιοχή θα προκαλέσει μετά από ένα μεγάλο χρονικό διάστημα την μετάλλαξη του γονιδίου της ταχύτητας στα θύματα του ώστε να γίνουν γρηγορότερα και να μπορούν να επιβιώνουν.

## 2.2 Η Δομή του Αλγορίθμου

Ο τρόπος λειτουργίας των Εξελικτικών Αλγορίθμων γενικά παρουσιάζεται στην παρακάτω εικόνα[24]:



Εικόνα 2.1: Ο κύκλος λειτουργίας των Εξελικτικών Αλγορίθμων

Η λειτουργία των Εξελικτικών Αλγορίθμων θα μπορούσε να περιγραφεί με τα παρακάτω βήματα:

1. Αρχικά, ένας πληθυσμός διαφορετικών ατόμων με τυχαίο γονιδίωμα δημιουργείται.
2. Υπολογίζονται οι τιμές των Αντικειμενικών Συναρτήσεων  $f \in F$  για κάθε υποψήφια λύση.
3. Η χρησιμότητα των επιμέρους χαρακτηριστικών των υποψήφιων λύσεων αξιολογείται με τις αντικειμενικές συναρτήσεις και καθορίζεται μια τιμή καταλληλότητας για τις

υποψήφιας. Η διαδικασία καθορισμού καταλληλότητας θα μπορούσε να καθορίσει τη σειρά επιλογής μιας υποψήφιας λύσης.

4. Μια διαδικασία επιλογής αποκλείει τις υποψήφιας λύσεις με χαμηλή τιμή καταλληλότητας και τοποθετεί τις υπόλοιπες με υψηλή τιμή καταλληλότητας σε μια δεξαμενή από όπου θα επιλεγούν τα άτομα για ζευγάρωμα.
5. Κατά τη φάση της αναπαραγωγής, δημιουργούνται οι απόγονοι με μεταβολή ή συνδυασμό των γενοτύπων των ατόμων που ζευγαρώνουν. Στη συνέχεια οι απόγονοι ενσωματώνονται στον πληθυσμό.
6. Εάν η συνθήκη για τον τερματισμό του αλγορίθμου ικανοποιείται, ο αλγόριθμος τερματίζεται διαφορετικά συνεχίζει από το 2<sup>ο</sup> βήμα.

Οι Γενετικοί Αλγόριθμοι ακολουθούν την ίδια διαδικασία όπως περιγράφηκε παραπάνω δηλαδή διαθέτουν έναν πληθυσμό από υποψήφιας λύσεις, οι οποίες αναπαράγονται και μεταλλάσσονται και αυτό συνεχίζεται έως ότου εκπληρωθεί μια συνθήκη τερματισμού. Γενική μορφή του Γενετικού Αλγορίθμου [29]:

procedure Genetic

begin

t=0

initialize ( P(t) )

evaluate ( P(t) )

while (not termination ( P (t) )) do

t=t+1

subP(t)=select(P(t))

crossover (subP(t))

mutate (subP(t))

$P(t+1)=\text{recombine}(P(t), \text{subP}(t))$

evaluate(P(t+1))

end while

end Genetic

Στον παραπάνω αλγόριθμο με  $P(t)$  συμβολίζεται ο πληθυσμός των υποψήφιων λύσεων στη γενιά  $t$ . Η αρχικοποίηση του πληθυσμού, που γίνεται από τη συνάρτηση initialize, εξαρτάτε κάθε φορά από τη φύση του προβλήματος και το μέγεθος του πληθυσμού παραμένει σταθερό καθόλη τη διάρκεια εκτέλεσης του αλγορίθμου. Ο αλγόριθμος επαναλαμβάνεται εφόσον δεν ισχύσει η συνθήκη τερματισμού, που μπορεί να είναι είτε η υπέρβαση του μέγιστου αριθμού επαναλήψεων είτε η εύρεση βέλτιστης λύσης. Σε κάθε επανάληψη επιλέγεται ένας μέρος του πληθυσμού - subP(t) - και σ' αυτόν εφαρμόζονται οι γενετικοί τελεστές της διασταύρωσης και της μετάλλαξης. Στη συνέχεια συνδυάζονται το μέρος του πληθυσμού που υπέστη διασταύρωση και μετάλλαξη και ο τρέχων πληθυσμός, ώστε να παραχθεί η επόμενη  $P(t+1)$  γενιά.

## 2.3 Πλεονεκτήματα Γενετικών Αλγορίθμων

Οι Γενετικοί Αλγόριθμοι παρουσιάζουν αρκετά πλεονεκτήματα και γι' αυτό το λόγο επιλέγονται από του ερευνητές για την επίλυση δύσκολων προβλημάτων. Τα κυριότερα πλεονεκτήματα είναι [27]:

1. Λύνουν δύσκολα προβλήματα γρήγορα και αξιόπιστα. Τοπικά μέγιστα-ελάχιστα δεν αποτελούν εμπόδια στην εύρεση βέλτιστων λύσεων, οι γενετικοί αλγόριθμοι είναι σε θέση να τα ξεπεράσουν και να εντοπίσουν βέλτιστη λύση σ' ένα πρόβλημα.
2. Μπορούν να συνεργαστούν εύκολα με άλλα συστήματα ή να συμμετέχουν σε υβριδικές μορφές με άλλες μεθόδους. Επειδή οι Γενετικοί Αλγόριθμοι είναι απλοί και εύκολοι στην υλοποίησή τους μπορούν αρκετά εύκολα να εργασθούν σε συνδυασμό με άλλα συστήματα αναζήτησης λύσεων, για παράδειγμα θα μπορούσε να εκτελείται ένας

Γενετικός Αλγόριθμος και μετά από κάθε επανάληψη να εκτελείται ένας άλλος αλγόριθμος τοπικής αναζήτησης.

3. Είναι εύκολα επεκτάσιμοι και εξελίξιμοι και οι προγραμματιστές μπορούν εύκολα να προβούν σε αλλαγές πάντα προς όφελος της απόδοσης.
4. Εφαρμόζονται σε μια μεγάλη ποικιλία προβλημάτων περισσότερο από κάθε άλλη μέθοδο βελτιστοποίησης.
5. Δεν απαιτούν περιορισμούς στις συναρτήσεις που επεξεργάζονται και είναι ανθεκτικοί σε σφάλματα. Για παράδειγμα ένα μηχάνημα που καταγράφει τους καρδιακούς παλμούς μπορεί να εισάγει στη μέτρηση ένα σφάλμα στη μέτρηση 0,1% και ο Γενετικός αλγόριθμος να οδηγήσει σε αξιόπιστη λύση.
6. Μπορούν εύκολα να παραλληλιστούν. Είναι τέτοια η φύση των Γενετικών Αλγορίθμων που θα μπορούσαν να εκμεταλλευτούν πλήρων μια παράλληλη μηχανή. Για παράδειγμα θα μπορούσε η αντικειμενική συνάρτηση να εκτελείται σε διαφορετικούς υπολογιστές ή ο κάθε υπολογιστής να εκτελεί τον αλγόριθμο για ένα μέρος του πληθυσμού.
7. Μπορούν να καλύψουν μεγάλους χώρους στην αναζήτηση βέλτιστης λύσης σε μικρό χρονικό διάστημα. Αυτό γίνεται διότι ο πληθυσμός που εξετάζεται είναι αντιπροσωπευτικό δείγμα και πρακτικά η αναζήτηση καλύπτει πολύ μεγαλύτερους πληθυσμούς.
8. Είναι η μοναδική μέθοδος που κάνει ταυτόχρονα αναζήτηση καλύτερων λύσεων και εκμετάλλευση της ήδη επεξεργασμένης πληροφορίας.

## 2.4 Τα Κύρια Χαρακτηριστικά ενός Γενετικού Αλγορίθμου

Τα ιδιαίτερα χαρακτηριστικά που έχουν οι Γενετικοί Αλγόριθμοι τους κάνουν να πλεονεκτούν έναντι των παραδοσιακών μεθόδων στην αναζήτηση λύσεων πάνω σε δύσκολα προβλήματα καθώς και στη βελτιστοποίησή τους.

Οι Γενετικοί Αλγόριθμοι εξετάζουν κωδικοποιημένες τιμές που προέρχονται από το συνολικό εύρος τιμών που μπορούν να πάρουν οι μεταβλητές και όχι τις ίδιες τις μεταβλητές του προβλήματος. Η κωδικοποίηση γίνεται για να μπορεί να γίνει η επεξεργασία των δεδομένων εύκολη αλλά και να επιτρέπεται έτσι η παράλληλη επεξεργασία τους. Για παράδειγμα εάν έπρεπε να υπολογίσουμε την μεγιστοποίηση της παραγωγής με ταυτόχρονη ελαχιστοποίηση του κόστους που έχουν 10 μηχανές, θα μπορούσαμε να κωδικοποιήσουμε τη λειτουργία ή όχι μιας μηχανής με το δυαδικό ψηφίο 1 ή 0 αν είναι ανενεργή. Έτσι θα είχαμε μια σειρά από 10 δυαδικά ψηφία όπως την 1100111110, που δηλώνει ότι οι μηχανές τρία, τέσσερα και δέκα είναι ανενεργές.

Οι Γενετικοί Αλγόριθμοι για τον εντοπισμό ενός ολικού μέγιστου ή ελάχιστου κάνουν ταυτόχρονα αναζήτηση σε πολλά διαφορετικά σημεία και έτσι δεν έχουν τον κίνδυνο να παγιδευτούν σε τοπικά ελάχιστα ή μέγιστα.

Χρησιμοποιείται μόνο η αντικειμενική συνάρτηση για την λύση ενός προβλήματος και καμιά άλλη πληροφορία κατά τη διάρκεια εκτέλεσης του αλγορίθμου. Αυτό έχει ως αποτέλεσμα την ανεξαρτησία του αλγορίθμου αλλά μπορεί να μην ληφθούν υπόψη και πληροφορίες που θα βοηθούσανε στην καλύτερη αντιμετώπιση του προβλήματος.

Στους Γενετικούς Αλγορίθμους κατά τη διαδικασία της επιλογής των ατόμων που θα ζευγαρώσουν, στη διαδικασία της διασταύρωσης των ζευγαριών και στη φάση της μετάλλαξης, λαμβάνονται αποφάσεις βάση κάποιων πιθανοτήτων όπου εμπεριέχεται το στοιχείο της τύχης. Η τύχη αυτή όμως οδηγεί σε καλύτερες λύσεις και όχι σε τυφλές αναζητήσεις.



## 2.5 Η Κωδικοποίηση

Το πιο σημαντικό μέρος ενός Γενετικού Αλγορίθμου είναι η κωδικοποίηση του προβλήματος σε μορφή χρωμοσωμάτων, σύμφωνα με πολλούς ερευνητές. Η κωδικοποίηση θα πρέπει να γίνεται με τέτοιο τρόπο ώστε:

- Να μην χάνεται καμιά πληροφορία κατά την κωδικοποίηση των δεδομένων.
- Να μπορεί να γίνει διάκριση μεταξύ ενός καλού χρωμοσώματος και ενός κακού.

Η κωδικοποίηση επίσης αφορά ένα μέρος του πληθυσμού που εξετάζεται και από το οποίο θα προκύψει η λύση του προβλήματος. Για το λόγο αυτό θα πρέπει να γίνει με ένα μαθηματικό τρόπο ώστε η επεξεργασία να μπορεί να γίνει εύκολα από υπολογιστή.

Η δυαδική Κωδικοποίηση είναι η πρώτη μορφή κωδικοποίησης που εφαρμόστηκε και είναι η απλούστερη όλων. Μπορεί το χρωμόσωμα να είναι ένας ακέραιος αριθμός και τα γονιδιά του να είναι δυαδικά ψηφία ή όλο το χρωμόσωμα να είναι δυαδικά ψηφία ή το χρωμόσωμα να είναι αλφαριθμητικά σύμβολα και να μετατρέπονται σε δυαδικά ψηφία. Η κωδικοποίηση γίνεται ανάλογα με το πρόβλημα και επειδή από αυτήν εξαρτάται η λύση του προβλήματος απαιτεί ιδιαίτερη προσοχή.

## 2.6 Οι Γενετικοί Τελεστές

Στον πυρήνα των Γενετικών Αλγορίθμων υπάρχουν οι τρεις γενετικοί τελεστές της επιλογής, της διασταύρωσης και της μετάλλαξης. Η σειρά που θα έχουν μέσα στο αλγόριθμο και η ύπαρξή τους εξαρτάται από το πρόβλημα, ενώ η εύρεση βέλτιστης τιμής για την Αντικειμενική Συνάρτηση ή Συνάρτηση Καταλληλότητας (fitness function) αποτελεί στον στόχο για τους Γενετικούς Αλγορίθμους.

### 2.6.1 Αντικειμενική Συνάρτηση ή Συνάρτηση Καταλληλότητας

Η αντικειμενική συνάρτηση παίρνει ως είσοδο μια συμβολοσειρά ενός χρωμοσώματος και επιστρέφει μια τιμή καταλληλότητας που είναι ανάλογη με το πόσο καλά αυτό το χρωμόσωμα λύνει το πρόβλημα. Η τιμή καταλληλότητας επίσης παίζει ρόλο για το αν το άτομο που αντιπροσωπεύεται με το συγκεκριμένο χρωμόσωμα θα συνεχίσει την επιβίωσή του και αν θα επιλεγεί για αναπαραγωγή.

Η αντικειμενική συνάρτηση θα πρέπει να αντανακλά άμεσα το πρόβλημα και να είναι τέτοια που να διαφοροποιείται για κάθε χρωμόσωμα. Έτσι όσο πιο πολύ διαφοροποιείται η τιμή της από χρωμόσωμα σε χρωμόσωμα τόσο καλύτερη είναι και τόσο πιο πολύ θα βοηθήσει τον Γενετικό Αλγόριθμο να αντιμετωπίσει αποτελεσματικότερα το πρόβλημα. Είναι σημαντικό όμως να είναι εύκολα υπολογίσιμη για να μην επιβραδύνει την εξέλιξη του αλγορίθμου.

### 2.6.2 Επιλογή

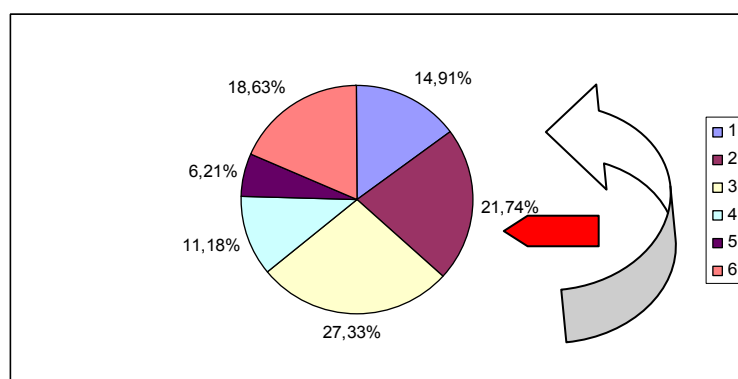
Όπως γίνεται στη φύση όπου τα καταλληλότερα άτομα ζευγαρώνουν μεταξύ τους ώστε να αναπαραχθούν έτσι και στους Γενετικούς Αλγορίθμους δημιουργείται μια λίστα με υποψήφια άτομα για ζευγάρωμα. Αυτή η λίστα λέγεται δεξαμενή ζευγαρώματος (mating pool) και δεν περιέχει μόνο τα καταλληλότερα άτομα για ζευγάρωμα αλλά μπορεί να περιέχει και λιγότερο κατάλληλα άτομα. Αυτό γίνεται για να διευκολυνθεί η εξέλιξη του αλγορίθμου διότι κάποιες φορές από λιγότερο κατάλληλα άτομα προκύπτουν καταλληλότεροι απόγονοι. Για την επιλογή ακολουθούνται οι παρακάτω μέθοδοι:

**Επιλογή Ρουλέτας:** Σύμφωνα με αυτή τη μέθοδο, κάθε άτομο που έχει κωδικοποιηθεί το χρωμόσωμά του σε μια συμβολοσειρά, μέλος του πληθυσμού που εξετάζεται με τον αλγόριθμο, αντιπροσωπεύεται σε ένα μέρος της ρουλέτας αναλογικά σύμφωνα με την τιμή καταλληλότητας. Για παράδειγμα έστω ότι έχουμε έναν πληθυσμό των 6 συμβολοσειρών (χρωμοσώματα ατόμων) όπως φαίνεται στον παρακάτω πίνακα:

Αριθμός Χρωμοσώματος	Χρωμόσωμα	Τιμή Καταλληλότητας	Απόδοση %
1	1100110101	240	14,91 %
2	1011110001	350	21,74 %
3	1110100101	440	27,33 %
4	00111000011	180	11,18 %
5	1111001100	100	6,21 %
6	0101001110	300	18,63 %
Σύνολο		1610	100,00 %

Πίνακας 2.1 Τιμές καταλληλότητας και απόδοση

Υποθέτουμε ότι στη στήλη «Τιμή Καταλληλότητας» εμφανίζεται η τιμή της συνάρτησης καταλληλότητας για το κάθε χρωμόσωμα και στη στήλη «Απόδοση %» εμφανίζεται το ποσοστό του κάθε χρωμοσώματος ως προς το συνολικό ποσοστό του πληθυσμού. Η τελευταία στήλη του πίνακα απεικονίζεται στην επιφάνεια μιας ρουλέτας όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 2.2 Σχηματική αναπαράσταση της ρουλέτας

Για την διαδικασία της αναπαραγωγής στρίβουμε τη ρουλέτα 6 φορές και όποιο χρωμόσωμα επιλέγεται αντιγράφεται στη δεξαμενή της αναπαραγωγής (mating pool). Επειδή κάποια η επιλογή βασίζεται σε κάποιες πιθανότητες, μπορεί να επιλεγεί δύο φορές ένα χρωμόσωμα και άλλο να μην επιλεγεί καθόλου ή μπορεί να επιλεγεί το χρωμόσωμα με τη χειρότερη τιμή και να μην επιλεγεί αυτό με την καλύτερη. Λογικά όμως τα χρωμοσώματα με καλύτερη τιμή στη συνάρτηση καταλληλότητας έχουν περισσότερες πιθανότητες να επιλεγούν σε σχέση με τα υπόλοιπα.

Αυτή η μέθοδος έχει τρία μειονεκτήματα. Το πρώτο είναι ότι τα χρωμοσώματα με υψηλή τιμή στη συνάρτηση καταλληλότητας θα τείνουν να έχουν περισσότερα αντίγραφα στη δεξαμενή ζευγαρώματος και τα χρωμοσώματα με χειρότερες τιμές καταλληλότητας να μην έχουν κανένα αντίγραφο. Το δεύτερο είναι ότι υπάρχει κίνδυνος να παγιδευτεί η διαδικασία σε πρόωρη σύγκλιση. Εάν υπάρχει χρωμόσωμα με τιμή καταλληλότητας αρκετά πάνω από το μέσο όρο αλλά να απέχει αρκετά από τη βέλτιστη λύση του προβλήματος, υπάρχει κίνδυνος να έχει πολλά αντίγραφα στη δεξαμενή ζευγαρώματος και να παγιδευτεί έτσι ο αλγόριθμος σε τοπικό μέγιστο-ελάχιστο. Το τρίτο είναι ο κίνδυνος της στασιμότητας, δηλαδή σε κάποια φάση τα χρωμοσώματα μέσα στη δεξαμενή ζευγαρώματος να μοιάζουν αρκετά και να μην μπορεί η διαδικασία να εξελιχτεί.

**Επιλογή Tournament:** Η μέθοδος αυτή είναι η πιο απλή, γρήγορη και οδηγεί σε καλά αποτελέσματα. Διαλέγει από τα υποψήφια χρωμοσώματα, με τυχαίο τρόπο ομάδες (μεγαλύτερες του 2) χρωμοσωμάτων και επιλέγει από την κάθε ομάδα, το χρωμόσωμα με την καλύτερη τιμή καταλληλότητας και το αντιγράφει στην δεξαμενή ζευγαρώματος.

**Ελιτίστικη Επιλογή:** Η μέθοδος αυτή εξασφαλίζει ότι τουλάχιστον ένα αντίγραφο από τα χρωμοσώματα με την καλύτερη τιμή καταλληλότητας θα περάσει στη δεξαμενή ζευγαρώματος. Αυτή η μέθοδος χρησιμοποιείται συμπληρωματικά με τις προηγούμενες μεθόδους επιλογής. Σε κάποιες περιπτώσεις είναι δυνατόν να παγιδευτεί σε τοπικό ελάχιστο-μέγιστο.

### 2.6.3 Διασταύρωση

Όταν ολοκληρωθεί η διαδικασία της επιλογής θα έχει δημιουργηθεί η λίστα που ονομάζεται δεξαμενή διασταύρωσης (mating pool) που θα περιέχει τα άτομα που προορίζονται να γονιμοποιηθούν μεταξύ τους ανά ζεύγη. Η διαδικασία της διασταύρωσης επιλέγει από τη δεξαμενή διασταύρωσης τα άτομα που υπάρχουν εκεί για να διασταυρωθούν δηλαδή να ανταλλάξουν γενετικό υλικό. Ο τρόπος που θα επιλεγούν τα ζευγάρια και το πώς θα γίνει η ανταλλαγή του γενετικού υλικού διαφέρει από υλοποίηση σε υλοποίηση.

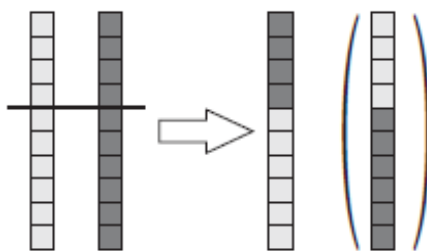
Η **πιθανότητα διασταύρωσης**: είναι η πιθανότητα ( $p_c$ ) που καθορίζει το ποσοστό παλαιών των χρωμοσωμάτων που πρέπει να αντικατασταθούν από νέα. Για παράδειγμα αν έχουμε έναν

πληθυσμό 100 χρωμοσωμάτων και η πιθανότητα αντικατάστασης είναι  $p_c = 0,1$  αυτό θα σημαίνει ότι πρέπει να αντικατασταθεί το 10% του πληθυσμού των χρωμοσωμάτων, δηλαδή 10 χρωμοσώματα θα αντικατασταθούν από νέα που θα προκύψουν από τη διασταύρωση, ενώ τα υπόλοιπα θα αντιγραφούν στον νέο πληθυσμό. Η αντιγραφή χρωμοσωμάτων από γενιά σε γενιά ονομάζεται κλωνοποίηση (cloning).

Εάν η πιθανότητα διασταύρωσης είναι πολύ μικρή υπάρχει ο κίνδυνος να εγκλωβιστούμε σε τοπικό ελάχιστο-μέγιστο, ενώ εάν η πιθανότητα διασταύρωσης είναι πολύ μεγάλη τότε υπάρχει ο κίνδυνος να ξεπεράσουμε το ολικό μέγιστο-ελάχιστο και να αποκλίνουμε από το στόχο μας. Συνήθως επιλέγεται αρχικά μεγάλη πιθανότητα διασταύρωσης για να επιταχυνθεί ο αλγόριθμος και μετά σταδιακά ελαττώνεται η πιθανότητα διασταύρωσης για να επικεντρωθούμε στο ολικό μέγιστο-ελάχιστο.

Τα χρωμοσώματα που θα αντικατασταθούν μπορεί να είναι τα χειρότερα όσον αφορά την τιμή καταλληλότητας ή να αντικατασταθούν κάποια με τυχαίο τρόπο. Σε κάποιες περιπτώσεις τα νέα χρωμοσώματα αντικαθιστούν τους γονείς τους ή κάποια άλλα παλιά χρωμοσώματα μόνο εάν έχουν καλύτερη τιμή καταλληλότητας ενώ σε κάποιες άλλες περιπτώσεις τα παλιά χρωμοσώματα συνυπάρχουν μαζί με τα νέα αυξάνοντας όμως έτσι τον πληθυσμό.

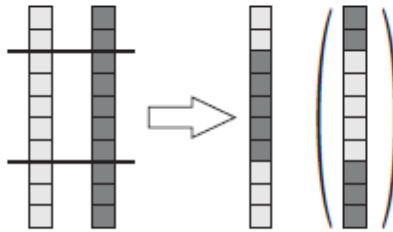
**Διασταύρωση Ενός Σημείου:** Είναι η πιο απλή και η πιο συνηθισμένη μορφή διασταύρωσης δύο χρωμοσωμάτων. Επιλέγεται τυχαία μια θέση μέσα στα χρωμοσώματα που θα διασταυρωθούν και μετά γίνεται ανταλλαγή του πάνω τμήματος του πρώτου χρωμοσώματος με το κάτω τμήμα του δευτέρου χρωμοσώματος.



Εικόνα 2.3 Διασταύρωση ενός σημείου

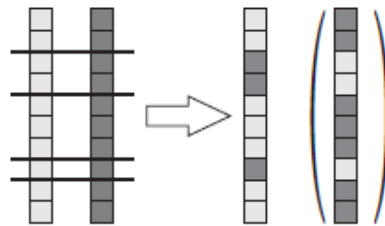
Όπως φαίνεται και στην παραπάνω εικόνα, ο πρώτος απόγονος που προέκυψε έχει το κάτω τμήμα του πρώτου γονέα και το πάνω τμήμα του δεύτερου γονέα, ενώ ο δεύτερος απόγονος έχει κληρονομήσει τα αντίθετα τμήματα από τα χρωμοσώματα των γονέων του.

**Διασταύρωση Δύο Σημείων:** Επιλέγονται τυχαία δύο σημεία στα χρωμοσώματα των γονέων, το πάνω και κάτω τμήμα του χρωμοσώματος του πρώτου γονέα και το ενδιάμεσο τμήμα από το χρωμόσωμα του δεύτερου γονέα κληρονομούνται από τον πρώτο απόγονο, ενώ ο δεύτερος απόγονος κληρονομεί τα αντίθετα τμήματα, όπως φαίνεται και στην παρακάτω εικόνα.



Εικόνα 2.4 Διασταύρωση δύο σημείων

**Διασταύρωση Πολλών Σημείων:** Επιλέγονται τυχαία πολλά σημεία στα χρωμοσώματα των γονέων και οι απόγονοι κληρονομούν αντίστροφα εναλλάξ τα αντίστοιχα τμήματα χρωμοσωμάτων των δύο γονέων όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 2.5 Διασταύρωση πολλών σημείων

## 2.7 Μετάλλαξη

Μετάλλαξη είναι η διαδικασία κατά την οποία γίνεται μια μικρή αλλαγή στο γενετικό υλικό. Η μετάλλαξη γίνεται τυχαία, με μικρή πιθανότητα, την πιθανότητα μετάλλαξης - mutation probability ( $p_m$ ) σε ένα ψηφίο του χρωμοσώματος, το οποίο αντιστρέφεται, από 0 σε 1 και το αντίστροφο. Η μετάλλαξη γίνεται κυρίως για να ανακτήσουμε τυχόν χαμένη πληροφορία και είναι σημαντικό η πιθανότητα να γίνει μετάλλαξη να είναι πολύ μικρή (μια στα χίλια ψηφία περίπου) γιατί διαφορετικά ο Γενετικός Αλγόριθμος θα εκφυλιστεί σε τυχαία αναζήτηση.

Έστω ότι θέλουμε να έχουμε μετάλλαξη με πιθανότητα ένα ψηφίο ανά δέκα χρωμοσώματα. Για κάθε ένα ψηφίο, για όλα τα χρωμοσώματα του πληθυσμού θα έπρεπε να γίνουν τα εξής: θα παίρναμε ένα συντελεστή μεταλλάξεως με τιμή στο διάστημα  $[0, 1]$ , έστω  $x$  ο συντελεστής μεταλλάξεως και  $x \leftarrow \text{random}(0,1)$  και εάν ίσχυε:  $x < p_m = \frac{1}{10M}$ , όπου  $M$  το πλήθος των ψηφίων του χρωμοσώματος, τότε θα συνέβαινε μετάλλαξη και το συγκεκριμένο ψηφίο του χρωμοσώματος για το οποίο γινόταν ο έλεγχος θα γινόταν από 1 σε 0 ή το αντίστροφο.

## 2.8 Κριτήρια Τερματισμού

Ένας αλγόριθμος πρέπει να βρίσκει λύση μετά από πεπερασμένο αριθμό επαναλήψεων, αν και στη φύση η εξέλιξη επαναλαμβάνεται συνεχώς. Για τον αποδοτικό τερματισμό ενός Γενετικού αλγορίθμου θα πρέπει να ισχύουν κάποια έξυπνα κριτήρια. Συνήθως χρησιμοποιούνται τα παρακάτω κριτήρια:

- Δεν επιτρέπεται η συνέχιση των επαναλήψεων που δεν δίνουν καλύτερα αποτελέσματα.
- Να διακόπτεται ο αλγόριθμος όταν υπάρχει βεβαιότητα για την εύρεση ολικού μέγιστου-ελάχιστου. Σε πολλές περιπτώσεις γνωρίζουμε εκ των προτέρων τη βέλτιστη λύση.
- Εάν η μέση απόσταση όλων των χρωμοσωμάτων είναι μικρότερη από κάποιο προκαθορισμένο όριο ή η διαφορά του χειρότερου από το καλύτερο χρωμόσωμα είναι μικρότερη επίσης από κάποιο προκαθορισμένο όριο.
- Όταν ξεπεράσουμε ένα συγκεκριμένο αριθμό επαναλήψεων, παρόλο που η ποιότητα των αποτελεσμάτων που θα έχουμε προσεγγίσει μέχρι εκείνη τη στιγμή θα είναι αμφίβολη.

## 2.9 Παράδειγμα Γενετικών Αλγορίθμων

Έστω ότι έχουμε 8 λογικές μεταβλητές τις a, b, c, d, e, f, g και h. Φυσικά αυτές οι μεταβλητές παίρνουν τιμές true/false και θέλουμε να διαπιστώσουμε εάν και πότε ικανοποιείται η παρακάτω έκφραση προτασιακής λογικής, η οποία περιέχει 5 περιορισμούς:

$$(\neg c \vee f) \wedge (\neg a \vee h) \wedge (e \vee d) \wedge (\neg a \vee \neg f \vee \neg h) \wedge (\neg b \vee d)$$

Ο γενότυπος του προβλήματος θα αποτελείται από τις λογικές τιμές των 8 μεταβλητών (true/false ή αντίστοιχα 1/0). Η συμβολοσειρά που θα κωδικοποιεί το χρωμόσωμα θα αποτελείται από μια ακολουθία 8 δυαδικών ψηφίων (bits) που θα αναπαριστούν τις τιμές των λογικών μεταβλητών a, b, c, d, e, f, g και h. Έτσι κάθε μεταβλητή θα είναι ένα γονίδιο μέσα στο χρωμόσωμα, που θα παίρνει τιμές 1/0.

Η συνάρτηση καταλληλότητας θα είναι το πλήθος των περιορισμών που ικανοποιούνται από μια συγκεκριμένη λύση. Για παράδειγμα το χρωμόσωμα 11100111 σημαίνει ότι οι μεταβλητές a, b, c, f, g και h έχουν την τιμή 1 (true) ενώ οι μεταβλητές d και e έχουν την τιμή 0 (false), ενώ η συνάρτηση καταλληλότητας έχει τιμή 2 αφού ικανοποιούνται μόνο 2 (οι δύο πρώτοι) περιορισμοί. Είναι προφανές ότι το χρωμόσωμα που θα έχει τιμή καταλληλότητας το 5 θα είναι η λύση στο πρόβλημα.

Για το συγκεκριμένο πρόβλημα υπάρχουν συνολικά  $2^8 = 256$  χρωμοσώματα που μπορούν να αναπαρασταθούν με δυαδικές συμβολοσειρές των 8 bits, όσες και οι μεταβλητές του προβλήματος, ενώ ικανοποίηση των περιορισμών και λύση στο πρόβλημα θα μπορούσαν να δώσουν περισσότερα από ένα χρωμοσώματα.

Αρχικά επιλέγουμε τα μέλη που θα απαρτίζουν τον αρχικό πληθυσμό στην πρώτη γενιά. Έστω λοιπόν ότι επιλέγονται τα μέλη με χρωμοσώματα 10101100, 10100111, 01100100 και 01000111.

1<sup>η</sup> Γενιά:

α/α Συμβολοσειράς	Συμβολοσειρά	Αξιολόγηση - Τιμή Καταλληλότητας	Αθροιστική Καταλληλότητα
1	10101100	4	4
2	10100111	3	4+3=7



3	01100100	3	7+3=10
4	01000111	2	10+2=12

Πίνακας 2.2 Τιμές για την 1η γενιά

Όπως φαίνεται στον παραπάνω πίνακα από τον αρχικό πληθυσμό υπολογίζεται η τιμή καταλληλότητας για το κάθε χρωμόσωμα, δηλαδή πόσοι περιορισμοί ικανοποιούνται. Επειδή κανένα χρωμόσωμα δεν έχει τιμή καταλληλότητας ίσον με 5, αυτό σημαίνει ότι δεν βρέθηκε λύση που να ικανοποιεί και τους 5 περιορισμούς. Άρα κριτήριο τερματισμού του αλγορίθμου στο συγκεκριμένο παράδειγμα είναι η τιμή καταλληλότητας να είναι ίση με 5. Από τις τιμές καταλληλότητας θα υπολογιστεί η αθροιστική καταλληλότητα η οποία θα χρησιμοποιηθεί στη συνέχεια στη διαδικασία της επιλογής.

Για την διαδικασία της επιλογής με τη μέθοδο της ρουλέτας θα επιλεγούν 4 χρωμοσώματα όπου θα τοποθετηθούν στη δεξαμενή διασταύρωσης. Αρχικά επιλέγονται τυχαία 4 αριθμοί στο διάστημα  $[0, 12]$ , έστω ότι επιλέγονται οι 3, 6, 2, 8. οι αριθμοί αυτοί σύμφωνα με τη στήλη της αθροιστικής καταλληλότητας αντιστοιχούν κατά σειρά στο πρώτο, στο δεύτερο, στο πρώτο και στο τρίτο χρωμόσωμα. Επιλέγονται δηλαδή δύο φορές το πρώτο χρωμόσωμα, μία το δεύτερο και μια το τρίτο, ενώ δεν επιλέγεται το τέταρτο χρωμόσωμα.

Η δεξαμενή ζευγαρώματος έχει τώρα τα 4 χρωμοσώματα: 10101100, 10100111, 10101100 και 01100100. Όπως φαίνεται στον παρακάτω πίνακα επιλέγεται να γίνει διασταύρωση μεταξύ πρώτης και δεύτερης συμβολοσειράς μετά το πέμπτο στοιχείο των συμβολοσειρών και μεταξύ τρίτης και τέταρτης συμβολοσειράς μετά το τρίτο στοιχείο των συμβολοσειρών αντίστοιχα.

α/α Συμβολοσειράς	Συμβολοσειρά	Νέα Συμβολοσειρά	Νέα Συμβολοσειρά μετά τη Μετάλλαξη
1	10101 100	10101111	10101111
2	10100 111	10100100	10100100
3	101 01100	10100100	10100110
4	011 00100	01101100	01101100

Πίνακας 2.3 Αναπαραγωγή και μετάλλαξη

Στη στήλη Νέα Συμβολοσειρά εμφανίζονται τα μέλη της επόμενης γενιάς και έστω ότι η πιθανότητα μετάλλαξης για το συγκεκριμένο πρόβλημα είναι  $p_m = \frac{1}{80} = 0,125$ . Για το κάθε bit της στήλης Νέα Συμβολοσειρά υπολογίζεται τυχαίος αριθμός στο διάστημα  $[0, 1]$ , αν ο τυχαίος αριθμός είναι μικρότερος από την πιθανότητα μετάλλαξης τότε θα συμβεί μετάλλαξη στο συγκεκριμένο bit και θα αντιστραφεί η τιμή του. Στο παράδειγμά μας έστω ότι πρέπει να γίνει

μετάλλαξη για το 23<sup>ο</sup> bit που βρίσκεται στη τρίτη συμβολοσειρά, το έβδομο στη σειρά, όπως φαίνεται και στον πίνακα παραπάνω, το συγκεκριμένο bit αλλάζει τιμή από 0 σε 1. Μετά τη μετάλλαξη ο νέος πληθυσμός βρίσκεται πλέον στη δεύτερη γενιά και ο Γενετικός Αλγόριθμος θα επαναληφθεί εκ νέου.

2<sup>η</sup> Γενιά:

α/α Συμβολοσειράς	Συμβολοσειρά	Αξιολόγηση - Τιμή Καταλληλότητας	Αθροιστική Καταλληλότητα
1	10101111	4	4
2	10100100	3	4+3=7
3	10100110	3	7+3=10
4	01101100	4	10+4=14

Πίνακας 2.4 Τιμές για την 2η γενιά

Για την δεύτερη γενιά γίνεται αξιολόγηση και υπολογίζεται η τιμή καταλληλότητας και αντίστοιχα η αθροιστική καταλληλότητα όπως φαίνεται στον πίνακα παραπάνω. Έτσι ο Γενετικός Αλγόριθμος επαναλαμβάνεται έως ότου βρεθεί χρωμόσωμα που να έχει τιμή καταλληλότητας ίσο με 5.

## 2.10 Εφαρμογές Γενετικών Αλγορίθμων

Αρχικά οι Γενετικοί Αλγόριθμοι ήταν αντικείμενο ανάπτυξης και μελέτης σε πανεπιστήμια και ερευνητικά κέντρα, τα τελευταία χρόνια όμως δημιουργούνται λόγω της τεχνολογικής εξέλιξης τεράστιες ανάγκες για ανάπτυξη αποδοτικών εφαρμογών στο χώρο της βελτιστοποίησης. Οι Γενετικοί Αλγόριθμοι πλέον εμπλέκονται άμεσα σε κάθε πτυχή της ανθρώπινης δραστηριότητας και της τεχνολογικής ανάπτυξης.

Συστήματα που βασίζονται σε Γενετικούς Αλγορίθμους αναπτύσσονται συνεχώς και λειτουργούν με επιτυχία σε τομείς όπως στη Μηχανολογία για την ανάπτυξη νέων μηχανών, στην Αεροναυπηγική για έλεγχο και ανάπτυξη νέων κατασκευών, στα Γραφικά Υπολογιστών, στη Σχεδίαση μέσω Υπολογιστών (CAD), στη Τεχνολογία Λογισμικού, στην Ρομποτική, στον Χρονοπρογραμματισμό, στην Οικονομία, στη Μουσική, στην Υγεία και γενικά σε όλους τους τομείς της ανθρώπινης δραστηριότητας όπου η τεχνολογική εξέλιξη και η βελτιστοποίηση είναι τα κύρια ζητούμενα.

# Κεφάλαιο 3

## Το Πρόβλημα της Κατάρτισης Εξεταστικού Ημερολογίου με τη Χρήση Γενετικού Αλγορίθμου

Ένα από τα πιο διάσημα προβλήματα που απασχόλησε κατά το παρελθόν και συνεχίζει να απασχολεί αρκετούς ερευνητές είναι το πρόβλημα της κατάρτισης ενός ημερολογίου εξετάσεων για ιδρύματα τριτοβάθμιας εκπαίδευσης. Αυτή η εμμονή με το συγκριμένο πρόβλημα δεν οφείλεται βέβαια στην απουσία εργαλείων και μεθόδων για την αντιμετώπισή του, αλλά σε μια σειρά από λόγους όπως:

- είναι ένα πρόβλημα γνωστό σε όλους, χωρίς να χρειάζονται αναλυτικές περιγραφές και ιδιαίτερες επεξηγήσεις, αφού όλα τα ανώτατα ιδρύματα διεθνώς έχουν παρόμοιες διαδικασίες.
- είναι απλό, χωρίς ιδιαιτερότητες, άρα και εύκολο στο να περιγραφτεί ή να μοντελοποιηθεί.

- οι λύσεις στοχεύουν κυρίως στην ανάπτυξη αποδοτικών αλγορίθμων που είναι εύκολο να δοκιμασθούν
- και κυρίως παρέχουν ένα εξαιρετικό περιβάλλον δοκιμής νέων τεχνικών επίλυσης προβλημάτων χρονοπρογραμματισμού.

Όπως είδαμε και στο πρώτο κεφάλαιο, τα προβλήματα χρονοπρογραμματισμού καλύπτουν όλο το φάσμα της ανθρώπινης δραστηριότητας και η επίλυσή τους αποτελεί μεγάλη πρόκληση. Όταν η επίλυση ενός προβλήματος συνδυάζεται και μ' ένα βέλτιστο αποτέλεσμα τότε έχουμε ένα πραγματικό άθλο. Δυστυχώς όμως, η λύση σε κάποια πρόβλημα δεν σημαίνει ότι τα προβλήματα λιγοστεύουν ή ότι η λύση τους έχει μόνιμο χαρακτήρα.

Πολλές φορές η επίλυση ενός προβλήματος είναι δυνατόν να δημιουργήσει άλλα προβλήματα. Για παράδειγμα η αύξηση της παραγωγής ενός αγαθού μπορεί να οδηγήσει σε νέο πρόβλημα, αυτό της μεταφοράς των αγαθών στην αγορά. Επίσης πρέπει να λάβουμε υπόψη ότι ζούμε σ' ένα περιβάλλον ιδιαίτερα ασταθές και περίπλοκο που οι τεχνολογικές εξελίξεις δείχνουν να δημιουργούν περισσότερες ανάγκες απ' αυτές που καλύπτουν. Απ' αυτό φαίνεται ότι η αναζήτηση τεχνικών επίλυσης προβλημάτων να είναι μια συνεχής και εξαιρετικά δύσκολη διαδικασία όπου τεχνικές και εργαλεία επίλυσης προβλημάτων βρίσκονται συνεχώς υπό δοκιμασία.

### **3.1 Η Οπτική του Προβλήματος**

Ένα πρόβλημα για να μπορεί να αντιμετωπισθεί κατάλληλα πρέπει να μελετηθεί, να αναλυθεί στις επιμέρους παραμέτρους του και αφού γίνει κατανοητό τότε θα μπορούσαμε να προχωρήσουμε στο σχεδιασμό μιας λύσης. Η λύση αυτή μετά από μια σειρά ελέγχων και δοκιμών θα μπορούσε να αποδειχθεί εάν είναι αποδεκτή ή όχι και αν είναι μια λύση που μας δίνει βέλτιστο αποτέλεσμα.

Για το πρόβλημα της δημιουργίας ενός εξεταστικού ημερολογίου για ανώτατο ίδρυμα πρέπει κατ' αρχάς να διακρίνουμε τους εμπλεκόμενους με αυτό, να μελετήσουμε το ρόλο που έχει ο καθένας στο πρόβλημα, πως αντιλαμβάνονται το πρόβλημα και πως μια πιθανή λύση θα

μπορούσε να χαρακτηριστεί αποδεκτή ή όχι, καλή ή λιγότερο καλή εξαιτίας των κριτηρίων ικανοποίησης που λογικά θα χρησιμοποιηθούν από τον κάθε εμπλεκόμενο.

Η δημιουργία του εξεταστικού ημερολογίου είναι ευθύνη συνήθως του διοικητικού προσωπικού (όταν υπάρχει) ενός ιδρύματος, λαμβάνει χώρα σε εύλογο χρονικό διάστημα πριν τις εξετάσεις και επαναλαμβάνεται δύο με τρεις φορές το χρόνο, ανάλογα με τον κανονισμό του κάθε ιδρύματος. Οι διοικητικοί υπάλληλοι για την κατάρτιση του εξεταστικού ημερολογίου λαμβάνουν υπόψη τους καταρχάς τη νομοθεσία και τους κανονισμούς ώστε να μην παραβιάζονται. Η διάρκεια των εξετάσεων θα πρέπει να είναι τέτοια ώστε να καλύπτονται πλήρως οι ακαδημαϊκές απαιτήσεις, επίσης να έχουν το χρόνο να ελέγξουν και να καταχωρήσουν τη βαθμολογία. Οι διοικητικοί υπάλληλοι ακόμα πρέπει να προσέξουν ώστε να μην παραβιάζεται το πρόγραμμα σπουδών, να βρεθούν οι κατάλληλες αίθουσες για την διαδικασία των εξετάσεων και να λαμβάνουν υπόψη τους τυχόν περιορισμούς εκ μέρους των εισηγητών και των επιτηρητών των εξετάσεων.

Οι καθηγητές που θα είναι εισηγητές επιθυμούν τα μαθήματα τα οποία εξετάζουν να μην συμπίπτουν χρονικά για να μπορούν να παρέχουν στους φοιτητές απρόσκοπτα τυχόν διευκρινίσεις και οδηγίες. Επίσης αιτήματα για εξαίρεσή τους από τις εξετάσεις για ορισμένες μέρες, όταν θα ήταν αναγκαίο λόγω ακαδημαϊκών ή ακόμα και προσωπικών λόγων, θα τους διευκόλυνε αρκετά.

Οι Φοιτητές θέλουν ανάμεσα στα εξεταζόμενα μαθήματα να υπάρχει εύλογο χρονικό διάστημα ώστε να προετοιμάζονται καλύτερα. Επίσης δεν θέλουν με τίποτα δύο ή περισσότερα μαθήματα που είναι πιθανόν να εξετάζονται να συμπίπτουν χρονικά ή να εξετάζονται την ίδια μέρα. Ενώ η γρήγορη ανάρτηση των βαθμών θα τους απάλλασσε από το άγχος και θα διευκόλυνε τις μελλοντικές τους επιλογές.

Οι Επιτηρητές δεν θα πρέπει βασικά να είναι και εισηγητές την ίδια χρονική στιγμή. Οι επιτηρήσεις να γίνονται σε μια αίθουσα εξέτασης και ο αριθμός των φοιτητών που εξετάζονται να είναι ανάλογος των επιτηρητών. Θα ήταν πολύ βολικό για αυτούς να είχαν πολλές επιτηρήσεις την ίδια μέρα. Η δίκαιη κατανομή των επιτηρήσεων σε πολλές περιπτώσεις συμβάλει στην απουσία εντάσεων και παραπόνων.

Θα μπορούσαν να αναφερθούν και άλλοι εμπλεκόμενοι με τις εξετάσεις όπως για παράδειγμα το προσωπικό καθαρισμού ή ασφάλειας αλλά για πρακτικούς λόγους δεν λαμβάνονται υπόψη στη παρούσα διατριβή.

## 3.2 Οι Περιορισμοί του Προβλήματος

Το πρόβλημα της κατάρτισης του ημερολογίου των εξετάσεων μπορεί να θεωρηθεί ως ένα πρόβλημα όπου ένας συγκεκριμένος αριθμός εξετάσεων πρέπει να προγραμματισθεί εντός ενός συγκεκριμένου χρονικού διαστήματος, το οποίο υποδιαιρείται σε μικρότερες χρονικές μονάδες τις χρονοθυρίδες, ενώ θα πρέπει να ικανοποιείται ένας αριθμός από περιορισμούς. Οι περιορισμοί βέβαια μπορεί να διαφέρουν από ίδρυμα σε ίδρυμα σε γενικές γραμμές όμως υπάρχουν και πολλές ομοιότητες.

Ανάλογα με το αν είναι υποχρεωτικό να εκπληρώνονται οι περιορισμοί κατατάσσονται σε δύο κατηγορίες, τους ανελαστικούς και τους ελαστικούς περιορισμούς. Οι ανελαστικοί (hard constraints) περιορισμοί πρέπει οπωσδήποτε να ικανοποιούνται και σε καμιά περίπτωση να μην παραβιάζονται για να μπορεί ένα πρόγραμμα εξετάσεων να γίνει αποδεκτό. Από την άλλη υπάρχει μια κατηγορία περιορισμών, οι ελαστικοί (soft constraints) περιορισμοί, η ικανοποίηση των οποίων δεν είναι υποχρεωτική αλλά επιθυμητή. Η ικανοποίηση ή όχι και σε πιο βαθμό των ελαστικών περιορισμών παίζει καθοριστικό ρόλο στην ποιότητα της λύσης.

### **Ανελαστικοί (hard constraints) Περιορισμοί:**

- Κανένας φοιτητής δεν μπορεί να εξετάζεται ταυτόχρονα σε δύο ή περισσότερα μαθήματα.
- Τα μαθήματα που εξετάζονται σε περισσότερες από μια πόλεις πρέπει να εξετάζονται την ίδια χρονική στιγμή.
- Η χωρητικότητα των αιθουσών μιας πόλης πρέπει να είναι επαρκής για το σύνολο των φοιτητών που εξετάζονται σ' αυτήν τη πόλη την ίδια χρονική στιγμή.

- Η εξεταστική περίοδος θα πρέπει να είναι ορισμένη σε συγκεκριμένες μέρες και για κάθε μέρα εξετάσεων να υπάρχουν καθορισμένες χρονικές περίοδοι (χρονοθυρίδες) όπου θα γίνονται εξετάσεις.

### **Ελαστικοί (soft constraints) Περιορισμοί:**

- Οι φοιτητές δεν θα πρέπει να εξετάζονται την ίδια μέρα σε περισσότερα από ένα μαθήματα.
- Είναι καλύτερα οι φοιτητές να μην εξετάζονται σε κάποιο μάθημα για διάστημα 2 ημερών τουλάχιστον μετά από μια εξέταση ώστε να μπορούν να προετοιμαστούν καλύτερα για το επόμενο μάθημα.
- Παρόλο που η νέα τεχνολογία επικοινωνιών θα μπορούσε να δώσει κάποιες λύσεις είναι προτιμότερο οι εισηγητές των εξεταζόμενων μαθημάτων να μην εξετάζουν ταυτόχρονα περισσότερα από ένα μαθήματα, για να μπορούν να υποστηρίξουν την εξεταστική διαδικασία καλύτερα.
- Να επιλέγεται για τις εξετάσεις πάντα μια αίθουσα στην οποία δεν θα περισσεύουν, στην καλύτερη περίπτωση ή θα περισσεύουν οι λιγότερες κενές θέσεις.
- Οι φοιτητές που συμμετέχουν στις εξετάσεις ενός μαθήματος είναι προτιμότερο να εξετάζονται όλοι σε μια αίθουσα εφόσον αυτό είναι εφικτό.

## **3.3 Μαθηματικό Μοντέλο των Περιορισμών**

Έστω ότι  $E$  είναι το σύνολο των εξεταστέων μαθημάτων,  $S$  το σύνολο των φοιτητών,  $EX$  το σύνολο των εξεταστών καθηγητών,  $C$  το σύνολο των πόλεων στις οποίες γίνονται εξετάσεις,  $R$  το σύνολο των αιθουσών γενικά σε όλες τις πόλεις και  $R_c$  το σύνολο των αιθουσών στην  $c$  πόλη,  $T$  το σύνολο των χρονοθυρίδων κατά τη διάρκεια των οποίων θα γίνονται οι εξετάσεις,  $L$  το σύνολο των μαθημάτων και τέλος  $D$  το σύνολο των εξεταστικών ημερών.

Καθορισμός και Επεξήγηση Παραμέτρων για τον Ορισμό των Περιορισμών:

$E_i^s$  Συμμετοχή του φοιτητή  $s$  στις εξετάσεις για το μάθημα  $i$ .

$S_i^E$  Πλήθος εξεταζόμενων φοιτητών στο μάθημα  $i$ .

$D_1$  Εξετάσεις την επόμενη μέρα μετά από εξετάσεις. Αν υπάρχει θα ισούται με το πλήθος των εξεταζόμενων μαθημάτων που υπάρχουν την επόμενη ημέρα, ενώ αν δεν υπάρχει με 0.

$D_2$  Εξετάσεις σε δύο μέρες μετά από μια εξεταστική ημέρα, αν υπάρχει θα ισούται με το πλήθος των εξεταζόμενων μαθημάτων που υπάρχουν σε δύο ημέρες, ενώ αν δεν υπάρχει με 0.

$E^l$  Εξέταση του μαθήματος  $l$ .

$E_i^{EX}$  Συμμετοχή του εισηγητή καθηγητή στις εξετάσεις για το μάθημα  $i$ .

$R_c^r$  Το άθροισμα των θέσεων από τις αίθουσες εξετάσεων για την πόλη  $c$ .

$R_{ic}^r$  Συμμετοχή της αίθουσας  $r$  στις εξετάσεις στην πόλη  $c$ . Ισούται με 1 αν συμμετέχει διαφορετικά με 0.

$R_{free}^r$  Κενές θέσεις στην αίθουσα  $r$ , αφού διατεθεί για διεξαγωγή εξετάσεων.

**Δεν μπορεί ένας φοιτητής να εξετάζεται ταυτόχρονα σε δύο ή περισσότερα μαθήματα:**

$$\forall t \in T \quad \sum_{i \in E} E_i^s t \leq 1$$

**Τα ίδια μαθήματα που εξετάζονται σε περισσότερες από μια πόλεις πρέπει να εξετάζονται την ίδια χρονική στιγμή:**

$$\forall t \in T, \forall c \in C \quad \sum_{l \in L} E^l c = t$$



Η χωρητικότητα των αιθουσών μιας πόλης πρέπει να είναι επαρκής για το σύνολο των φοιτητών που εξετάζονται σ' αυτήν τη πόλη την ίδια χρονική στιγμή σ' ένα μάθημα:

$$\forall t \in T, \forall c \in C \quad \sum_{i \in E} S_i^E t c \leq R_c^r$$

Οι φοιτητές δεν θα πρέπει να εξετάζονται την ίδια μέρα σε περισσότερα από ένα μαθήματα

$$\forall d \in D \quad \sum_{i \in E} E_i^s d \leq 1$$

Οι φοιτητές να μην εξετάζονται σε κάποιο μάθημα για διάστημα έως 2 ημερών τουλάχιστον μετά από μια εξέταση ώστε να μπορούν να προετοιμαστούν καλύτερα για το επόμενο μάθημα:

$$\sum_{i \in E} E_i^s D_1 = 0 \quad \text{και} \quad \sum_{i \in E} E_i^s D_2 = 0$$

Οι εισηγητές των εξεταζόμενων μαθημάτων να μην εξετάζουν ταυτόχρονα περισσότερα από ένα μαθήματα:

$$\forall t \in T \quad \sum_{i \in E} E_i^{EX} t \leq 1$$

Να επιλέγεται για τις εξετάσεις πάντα μια αίθουσα ανάμεσα στις διαθέσιμες, στην οποία θα περισσεύουν οι λιγότερες κενές θέσεις:

$$\forall r \in R, \forall i \in E \quad R^r - S_i^E = \min R_{free}^r$$

Οι φοιτητές που συμμετέχουν στις εξετάσεις ενός μαθήματος είναι προτιμότερο να εξετάζονται όλοι σε μια αίθουσα:

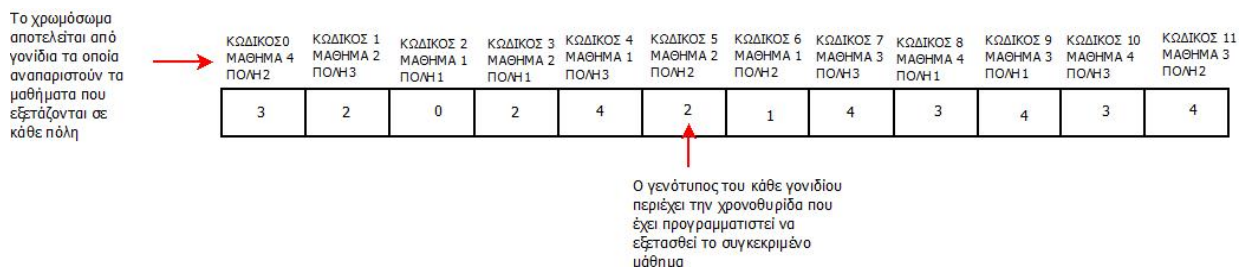
$$\forall i \in E, \forall c \in C \quad \sum_{r \in R} R_{ic}^r \leq 1$$

### 3.4 Κωδικοποίηση του Προβλήματος σε Μορφή Χρωμοσώματος

Το πιο σημαντικό μέρος ενός Γενετικού Αλγορίθμου είναι η κωδικοποίηση του προβλήματος σε μορφή χρωμοσωμάτων ώστε να μην χάνεται καμιά πληροφορία κατά την κωδικοποίηση των δεδομένων και να μπορεί να γίνει διάκριση μεταξύ ενός καλού χρωμοσώματος και ενός κακού. Επίσης θα πρέπει να γίνει με ένα μαθηματικό τρόπο ώστε η επεξεργασία να μπορεί να γίνει εύκολα από υπολογιστή.

Για τους παραπάνω λόγους επιλέχθηκε στο συγκεκριμένο πρόβλημα το χρωμόσωμα να αποτελείται από γονίδια, που το καθένα θα αναπαριστά ένα μάθημα. Όταν ένα μάθημα εξετάζεται σε διαφορετικές πόλεις τότε για κάθε πόλη και για το συγκεκριμένο μάθημα θα υπάρχει αντιπροσώπευση από ένα γονίδιο. Ο γενότυπος του κάθε γονιδίου θα περιέχει την χρονοθυρίδα, την ημέρα και την ώρα δηλαδή εξέτασης του συγκεκριμένου μαθήματος.

Η παρακάτω εικόνα αναπαριστά ένα χρωμόσωμα με δώδεκα θέσεις που κωδικοποιεί τέσσερα μαθήματα τα οποία εξετάζονται σε τρεις διαφορετικές πόλεις. Συνολικά έτσι αντιστοιχίζονται τα δώδεκα γονίδια του χρωμοσώματος σε δώδεκα μαθήματα.



Εικόνα 3.1 Χρωμόσωμα για τέσσερα μαθήματα που εξετάζονται σε τρεις πόλεις

Ο γενότυπος που είναι η τιμή που περιέχεται σε κάθε γονίδιο εκφράζει τις χρονοθυρίδες που έχουν ορισθεί και στις οποίες μπορούν να διεξαχθούν εξετάσεις. Όπως φαίνεται στην εικόνα παραπάνω τα μαθήματα 2, 3 και 4 ορίστηκαν για εξέταση την 2<sup>η</sup>, 4<sup>η</sup> και 3<sup>η</sup> χρονοθυρίδα αντίστοιχα σε όλες τις πόλεις στις οποίες θα γίνει εξέταση για τα μαθήματα αυτά. Ενώ το μάθημα 1 εξετάζεται στην πόλη 1 στην 0 χρονοθυρίδα, στην πόλη 3 στην 4<sup>η</sup> χρονοθυρίδα ενώ στην πόλη εξετάζεται στην 1<sup>η</sup> χρονοθυρίδα.

Το χρωμόσωμα διαμορφώνεται δυναμικά σε κάθε εκτέλεση του προγράμματος και συγκεκριμένα το μέγεθος του χρωμοσώματος καθορίζεται μετά την ανάγνωση του αρχείου που περιέχει τις δηλώσεις των μαθημάτων από τους φοιτητές. Κάθε φορά που διαβάζεται μια δήλωση ενός νέου μαθήματος αυτό παίρνει νέο κωδικό μαθήματος (με αύξουσα σειρά) και προστίθεται σε μια λίστα με όλα τα δηλωμένα μαθήματα. Επίσης στην πραγματικότητα αυτός ο κωδικός εκφράζει τη σειρά του μαθήματος πάνω στο χρωμόσωμα. Μετά την ανάγνωση του αρχείου με τις δηλώσεις των μαθημάτων και αφού καθορισθεί το τελικό πλήθος των μαθημάτων που έχουν δηλωθεί, καθορίζεται παραμετρικά το μέγεθος του χρωμοσώματος, όπως αυτό θα χρησιμοποιηθεί κατά τη εκτέλεση του γενετικού αλγορίθμου στη συνέχεια για να εντοπισθεί η καλύτερη λύση.

Με ανάλογο δυναμικό τρόπο καθορίζεται και το εύρος τιμών στο οποίο θα κυμαίνεται ο γενότυπος. Η τιμή δηλαδή που θα περιέχεται σε κάθε γονίδιο και που θα εκφράζει την χρονοθυρίδα που θα εξετάζεται το αντίστοιχο μάθημα. Το πρόγραμμα διαβάζει το αντίστοιχο αρχείο στο οποίο υπάρχουν αποθηκευμένες οι χρονοθυρίδες, οι ημέρες και οι ώρες που θα γίνονται οι εξετάσεις και μετά παραμετρικά καθορίζεται το εύρος τιμών των χρονοθυρίδων. Για παράδειγμα για την εικόνα 3.1 παραπάνω το εύρος των χρονοθυρίδων θα μπορούσε να είναι 0 έως 4.

### 3.5 Η Συνάρτηση Καταλληλότητας

Η αποδοχή ή όχι του προγράμματος των εξετάσεων που θα προκύψει καθώς και η ποιότητά του σε περίπτωση αποδοχής εξαρτάται απόλυτα από το είδος των περιορισμών καθώς και σε πιο βαθμό αυτοί παραβιάζονται. Έτσι δεν θα μπορούσε να γίνει αποδεκτό ένα πρόγραμμα εξετάσεων για παράδειγμα που ορίζει να γίνουν εξετάσεις κάποιων μαθημάτων χωρίς να μπορούν οι διαθέσιμες θέσεις στις αίθουσες εξετάσεων να καλύψουν τους φοιτητές που θα συμμετέχουν στις εξετάσεις.

Ανάλογα με το είδος των περιορισμών που παραβιάζονται θα ορίζεται και το ανάλογο κόστος το οποίο θα αθροίζεται. Έτσι η τελική λύση που θα προκύπτει κάθε φορά θα συνοδεύεται από ένα συνολικό κόστος που θα αντιπροσωπεύει τους περιορισμούς οι οποίοι παραβιάστηκαν αθροιστικά. Φυσικά οι ανελαστικοί περιορισμοί θα αντιστοιχούν σε πολύ υψηλό κόστος σε

σχέση με τους ανελαστικούς και έτσι λύσεις οι οποίες περιέχουν παραβιάσεις ανελαστικών περιορισμών εκ των πραγμάτων θα απορρίπτονται και δεν θα γίνονται δεκτές σε σχέση με άλλες λύσεις οι οποίες θα περιέχουν παραβιάσεις ελαστικών περιορισμών μόνο και άρα θα συνοδεύονται από μικρότερο κόστος.

Ο παρακάτω πίνακας περιέχει τους περιορισμούς που λαμβάνονται υπόψη στη δημιουργία του εξεταστικού προγράμματος, το κόστος παραβίασης για κάθε περιορισμό και το όνομα της μεταβλητής που χρησιμοποιείται στο λογισμικό που δημιουργεί το εξεταστικό πρόγραμμα:

A/A	Περιορισμός	Κόστος  Cost <sub>i</sub> ή C <sub>i</sub>	Μεταβλητή  Weight <sub>j</sub> ή W <sub>1</sub>
1	Δεν επιτρέπεται σε έναν φοιτητή να εξετάζεται ταυτόχρονα σε δύο ή περισσότερα μαθήματα	1000	stdSameHour
2	Τα ίδια μαθήματα που εξετάζονται σε περισσότερες από μια πόλεις πρέπει να εξετάζονται την ίδια χρονική στιγμή:	5000	lesSameHour
3	Η χωρητικότητα των αιθουσών μιας πόλης πρέπει να είναι επαρκής για το σύνολο των φοιτητών που εξετάζονται σ' αυτήν τη πόλη την ίδια χρονική στιγμή σ' ένα μάθημα	1000	wrongLessonRoom
4	Οι φοιτητές δεν θα πρέπει να εξετάζονται την ίδια μέρα σε περισσότερα από ένα μαθήματα	200	lesSameDay
5	Οι φοιτητές την επόμενη μέρα μετά από μια εξέταση να μην έχουν εξέταση σε άλλο μάθημα	100	lesNextDay

6	Οι φοιτητές δύο μέρες μετά από μια εξέταση να μην έχουν εξέταση σε άλλο μάθημα	50	lesNextTwoDays
7	Οι εισηγητές των εξεταζόμενων μαθημάτων να μην έχουν ταυτόχρονα περισσότερες από μια εξετάσεις	1000	ExmSameHour
8	Να επιλέγεται για τις εξετάσεις ενός μαθήματος πάντα μια αίθουσα, ανάμεσα στις διαθέσιμες, στην οποία θα περισσεύουν οι λιγότερες κενές θέσεις	100	multiRoomCapacity

Πίνακας 3.1 Κόστος παραβίασης περιορισμών

Η συνάρτηση καταλληλότητας θα παίρνει ως είσοδο μια συμβολοσειρά ενός χρωμοσώματος και θα επιστρέφει μια τιμή καταλληλότητας που είναι ανάλογη με το πόσο καλά το συγκεκριμένο χρωμόσωμα λύνει το πρόβλημα. Η τιμή καταλληλότητας επίσης θα παίζει ρόλο για το αν το άτομο που αντιπροσωπεύεται με το συγκεκριμένο χρωμόσωμα θα συνεχίσει την επιβίωσή του και αν θα επιλεγεί για αναπαραγωγή.

Η συνάρτηση καταλληλότητας που χρησιμοποιείται στο πρόγραμμα, της παρούσας εργασίας, για την αντιμετώπιση του προβλήματος και στην εξεύρεση λύσης που θα είναι αποδεκτή και όσο το δυνατό βέλτιστη δηλαδή θα παραβιάζει όσο το δυνατό λιγότερους περιορισμούς, είναι:

$$\text{Fitness} = \frac{1}{1 + F_{\text{cost}}}$$

**Συνάρτηση καταλληλότητας:**

και

**Συνάρτηση Κόστους:**  $F_{\text{cost}} = C_1W_1 + C_2W_2 + C_3W_3 + C_4W_4 + C_5W_5 + C_6W_6 + C_7W_7 + C_8W_8$

Κατά τη διάρκεια εκτέλεσης του προγράμματος και συγκεκριμένα στη διαδικασία αξιολόγησης του κάθε χρωμοσώματος υπολογίζεται η συνάρτηση καταλληλότητας. Η τιμή που θα έχει η συνάρτηση καταλληλότητας θα είναι μεγαλύτερη του μηδενός και μικρότερη – ίση του ένα, δηλαδή θα είναι  $0 < \text{Fitness} \leq 1$ . Χειρότερη λύση θα δίνει ένα χρωμόσωμα όσο η τιμή της συνάρτησης καταλληλότητας θα είναι κοντύτερα στο μηδέν. Ενώ όταν η συνάρτηση καταλληλότητας δώσει αποτέλεσμα ένα τότε το χρωμόσωμα θα δίνει βέλτιστη λύση.

Σύμφωνα με τον παραπάνω πίνακα 3.1, στις μεταβλητές αποθηκεύεται το πλήθος το παραβιάσεων των αντίστοιχων περιορισμών και για κάθε περιορισμό υπάρχει ένα κόστος το μέγεθος του οποίου καθορίζεται ανάλογα με τη σπουδαιότητα του περιορισμού που παραβιάζεται.

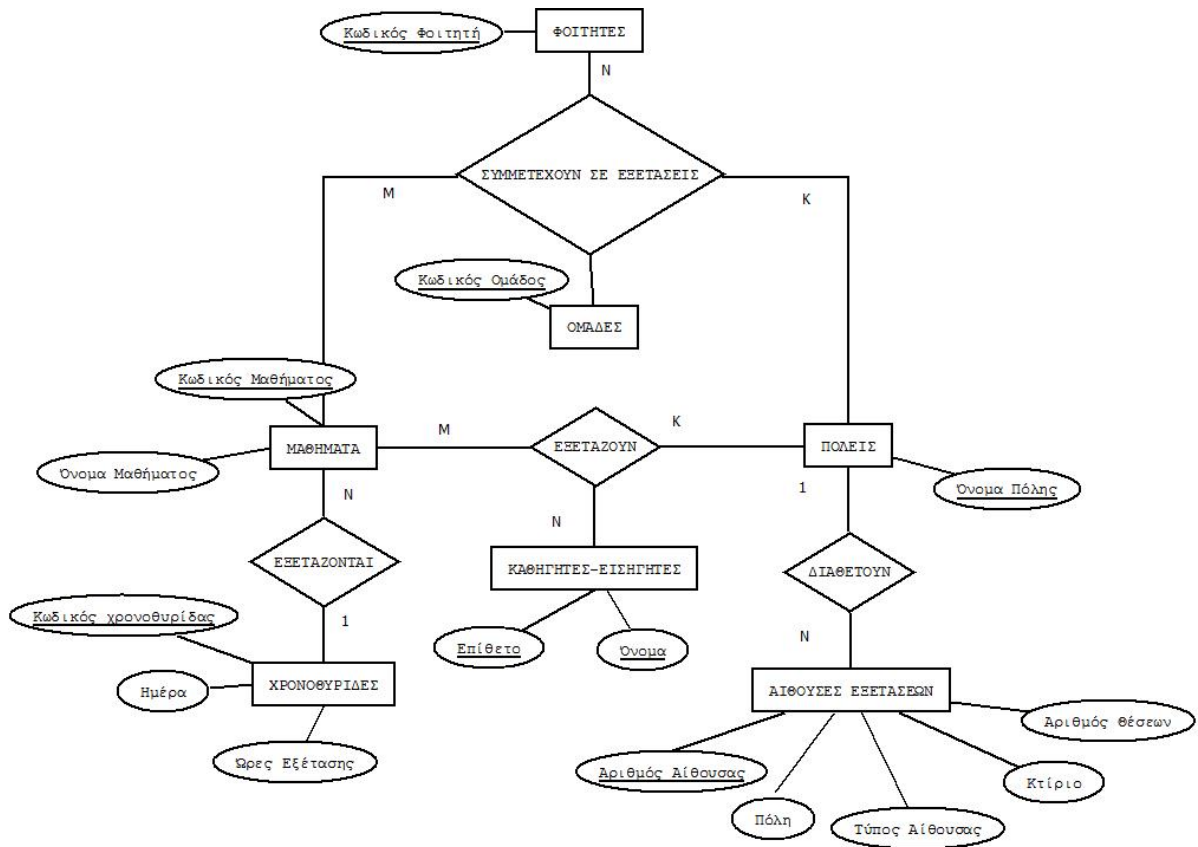
Το ECJ ToolKit που χρησιμοποιείται για την εκτέλεση του γενετικού αλγορίθμου χρησιμοποιεί την συνάρτηση καταλληλότητας στη μορφή:  $Fitness = \frac{1}{1 + F_{cost}}$  επειδή το πακέτο αυτό

επιδιώκει πάντα να μεγιστοποιεί τη συνάρτηση καταλληλότητας στο διάστημα από το μείον άπειρο έως το συν άπειρο. Έτσι για να μεγιστοποιηθεί η  $Fitness$  θα πρέπει η  $F_{cost}$  να ελαχιστοποιείται, δηλαδή να μηδενίζεται. Τελικά η  $Fitness$  θα έχει μέγιστη τιμή το 1 όταν η  $F_{cost}$  πάρει την τιμή 0.

## 3.6 Διάγραμμα Οντοτήτων Συσχετίσεων και τα Αρχεία Εισόδου

Έχοντας υπόψη το συνολικό πληροφοριακό σύστημα ενός ιδρύματος τριτοβάθμιας εκπαίδευσης, με έναν τρόπο αφαιρετικό, απομονώνονται τα αντικείμενα εκείνα που επηρεάζουν άμεσα την δημιουργία ενός εξεταστικού προγράμματος και χρησιμοποιούνται για την δημιουργία ενός διαγράμματος οντοτήτων συσχετίσεων. Ο σκοπός της δημιουργίας του διαγράμματος οντοτήτων συσχετίσεων είναι να καταγραφούν τα απαραίτητα δεδομένα που θα περιέχονται στα αρχεία εισόδου όπου το λογισμικό θα επεξεργάζεται για να μπορεί να δημιουργεί αυτόματα ένα εξεταστικό πρόγραμμα.

Η ακεραιότητα και η συνέπεια των δεδομένων εισόδου εννοείται ότι εξασφαλίζονται από το πληροφοριακό σύστημα του ιδρύματος από όπου θα μπορούσαν να προέρχονται τα δεδομένα αυτά και δεν είναι ευθύνη της παρούσας εργασίας για να εξασφαλίζει κάτι τέτοιο. Για παράδειγμα οι δηλώσεις των μαθημάτων από τους φοιτητές θεωρούνται ότι είναι έγκυρες, ότι έχουν ήδη ελεγχθεί και δεν περιέχουν λάθη.



Εικόνα 3.2: Διάγραμμα Οντοτήτων Συσχετίσεων

Το διάγραμμα Οντοτήτων – Συσχετίσεων όπως φαίνεται παραπάνω στην εικόνα 3.2 περιέχει τις βασικές οντότητες και συσχετίσεις που κρίνονται απαραίτητες και επηρεάζουν άμεσα τη διαδικασία δημιουργίας του εξεταστικού προγράμματος, παρέχοντας τα απαραίτητα δεδομένα εισόδου.

StudentI.essons.txt	Rooms.txt	Examiners.txt	Timeslots.txt
*Κωδικός Φοιτητή	*Πόλη	*Επίθετο	*Κωδικός Χρονοθυρίδας
*Όνομα Μαθήματος	°Κτίριο	*Όνομα	°Ημέρα
°Πόλη	*Αριθμός Αίθουσας	*Μάθημα	°Ώρες Εξέτασης
*Κωδικός Ομάδας	°Τύπος Αίθουσας	*Πόλη	
	°Αριθμός θέσεων		

Εικόνα 3.3 Τα αρχεία εισόδου

Τα δεδομένα εισόδου για το λογισμικό δημιουργίας προγράμματος εξετάσεων περιέχονται σε τέσσερα απλά αρχεία κειμένου (text) όπως φαίνονται παραπάνω στην εικόνα 3.3. Τα τέσσερα αυτά αρχεία περιέχουν δεδομένα που για το πληροφοριακό σύστημα ενός ιδρύματος τριτοβάθμιας εκπαίδευσης είναι πολύ εύκολο και απλό να δημιουργήσει.

Σύμφωνα με το διάγραμμα της εικόνας 3.2 από τη σχέση «ΣΥΜΜΕΤΕΧΟΥΝ ΣΕ ΕΞΕΤΑΣΕΙΣ» παράγεται το αρχείο **StudentLessons.txt** που περιέχει τέσσερα πεδία αλφαριθμητικών το «Κωδικός Φοιτητή», το πεδίο «Όνομα Μαθήματος», το πεδίο «Πόλη» και το πεδίο «Κωδικός Ομάδας». Τα πεδία αυτά έχουν ένα κενό χαρακτήρα ανάμεσά τους. Το αρχείο αυτό περιέχει τις δηλώσεις μαθημάτων των φοιτητών.

Από τη σχέση «ΕΞΕΤΑΖΟΥΝ» παράγεται το αρχείο **Examiners.txt** που περιέχει και αυτό τέσσερα πεδία αλφαριθμητικών το «Επίθετο», το «Όνομα», το «Μάθημα» και το «Πόλη» που διαχωρίζονται μεταξύ τους με ένα κενό χαρακτήρα. Το αρχείο αυτό περιέχει τους εισηγητές καθηγητές και τα μαθήματα που εξετάζουν.

Από την οντότητα «ΧΡΟΝΟΘΥΡΙΔΕΣ» παράγεται το αρχείο **Timeslots.txt** που περιέχει τρία πεδία αλφαριθμητικών το «Κωδικός Χρονοθυρίδας», το πεδίο «Ημέρα» και το πεδίο «Ωρες Εξέτασης» που διαχωρίζονται μεταξύ τους με ένα κενό χαρακτήρα. Το αρχείο αυτό περιέχει τις χρονοθυρίδες, τα χρονικά διαστήματα δηλαδή που μπορούν να γίνουν εξετάσεις.

Τέλος από την οντότητα «ΑΙΘΟΥΣΕΣ ΕΞΕΤΑΣΕΩΝ» παράγεται το αρχείο **Rooms.txt** που περιέχει πέντε πεδία αλφαριθμητικών το «Πόλη», «Κτίριο», «Αριθμός Αίθουσας», «Τύπος Αίθουσας» και «Αριθμός Θέσεων» που διαχωρίζονται μεταξύ τους με ένα κενό χαρακτήρα. Το αρχείο αυτό περιέχει τις αίθουσες στις οποίες μπορούν να γίνουν εξετάσεις.

## 3.7 Το UML Διάγραμμα της Εφαρμογής

Για την υλοποίηση της εφαρμογής έχουν δημιουργηθεί από την αρχή ή έχουν τροποποιηθεί σε μεγάλο βαθμό από παλαιότερη υλοποίηση[28], συνολικά 13 κλάσεις, όπως αυτές φαίνονται στο UML διάγραμμα της εφαρμογής, που παρουσιάζεται παρακάτω στην εικόνα 3.4.

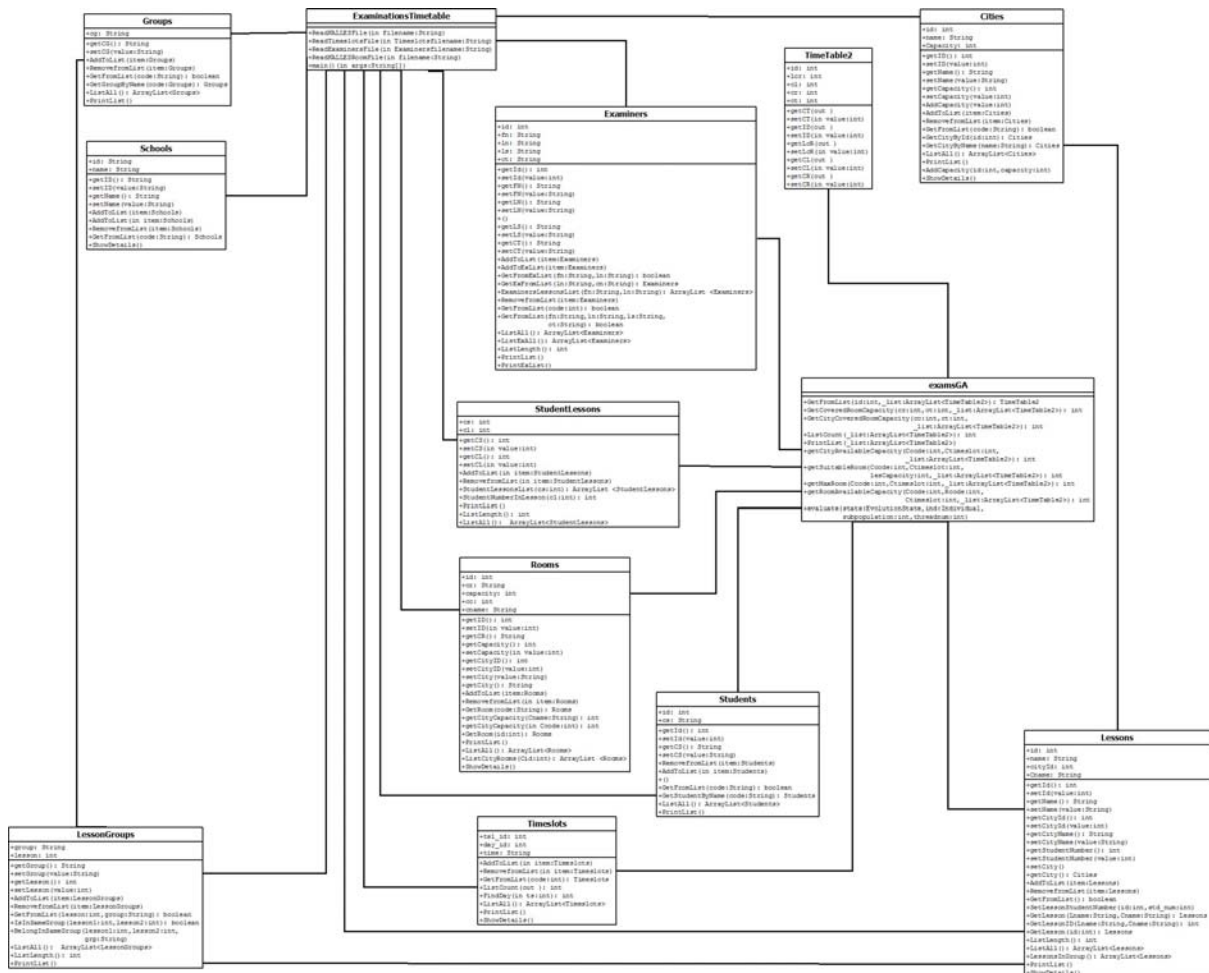
Αρχικά εκτελείται η κλάση `ExaminationsTimetable`, η οποία έχει ως κύριο καθήκον να διαβάσει τα αρχεία εισόδου και να δημιουργήσει τις αντίστοιχες λίστες με τα δεδομένα εισόδου. Πρώτα διαβάζει το αρχείο «`StudentLessons.txt`» (ενότητα 3.6, εικόνα 3.3) που περιέχει τις δηλώσεις μαθημάτων των φοιτητών. Παράλληλα δημιουργούνται και οι λίστες στη μνήμη του υπολογιστή, για τους φοιτητές (κλάση `Students`), για τις ομάδες μαθημάτων (κλάση `Groups`), για τις πόλεις



στις οποίες γίνονται εξετάσεις (κλάση Cities), (κλάση Lessons), και για τα μαθήματα που έχει δηλώσει ο κάθε φοιτητής (κλάση StudentLessons). Έπειτα διαβάζει το αρχείο «Rooms.txt» και δημιουργεί τη λίστα με τις αίθουσες εξετάσεων (κλάση Rooms). Στη συνέχεια διαβάζει το αρχείο «Timeslots.txt» δημιουργεί την αντίστοιχη λίστα με τις χρονοθυρίδες στις οποίες επιτρέπεται να πραγματοποιηθούν εξετάσεις (κλάση Timeslots). Τέλος διαβάζει το αρχείο «Examiners.txt» και δημιουργεί την λίστα με τους εξεταστές καθηγητές και μια άλλη λίστα με τα μαθήματα που εξετάζει ο κάθε εξεταστής (κλάση Examiners)

Μετά την αρχική προετοιμασία των δεδομένων εισόδου ξεκινά ο βρόγχος του γενετικού αλγορίθμου με τη βοήθεια του ECJ Toolkit, όπως αυτός αναπαριστάται στην εικόνα 3.5 της ενότητας 3.8. Για την εκτέλεση του γενετικού αλγορίθμου μεγάλη σημασία παίζουν τα αρχεία παραμέτρων (κατάληξη params), sum.params και ec.params. Σ' αυτά τα αρχεία παραμέτρων δηλώνονται στοιχεία όπως το πλήθος των γενεών για τις οποίες θα τρέξει το πρόγραμμα, το πλήθος του πληθυσμού, η μορφή διασταύρωσης, η επιλογή των ατόμων που θα συμμετέχουν στη διασταύρωση ή στην επόμενη γενιά, η πιθανότητα μετάλλαξης, πόσοι επεξεργαστές (ή πυρήνες) θα συμμετέχουν στη διαδικασία. Η διαδικασία της αξιολόγησης των πιθανών λύσεων, δηλαδή η συνάρτηση καταλληλότητας είναι το σπουδαιότερο μέρος τους γενετικού αλγορίθμου και εκτελείται από την κλάση examsGA.

Η κλάση examsGA περιέχει την διαδικασία με την οποία υπολογίζεται η τιμή καταλληλότητας, καλείται για κάθε μέλος του πληθυσμού και από το χρωμόσωμά του υπολογίζεται η τιμή καταλληλότητας. Εάν υπολογιστεί τιμή 1 ως τιμή καταλληλότητας για κάποιο χρωμόσωμα τότε αυτό σημαίνει ότι βρέθηκε βέλτιστη λύση, διαφορετικά συνεχίζεται ο αλγόριθμος έως να συμπληρωθεί ο μέγιστος αριθμός γενεών που μπορεί να τρέξει ο αλγόριθμος. Η κλάση examsGA επεξεργάζεται όλες τις λίστες με τα δεδομένα εισόδου προκειμένου να υπολογίσει την τιμή καταλληλότητας. Επίσης στη κλάση αυτή διαμορφώνεται το πρόγραμμα των εξετάσεων και στο τέλος εκτυπώνεται το πρόγραμμα εξετάσεων που έχει την καλύτερη τιμή καταλληλότητας.



Εικόνα 3.4 Το UML Διάγραμμα της Εφαρμογής

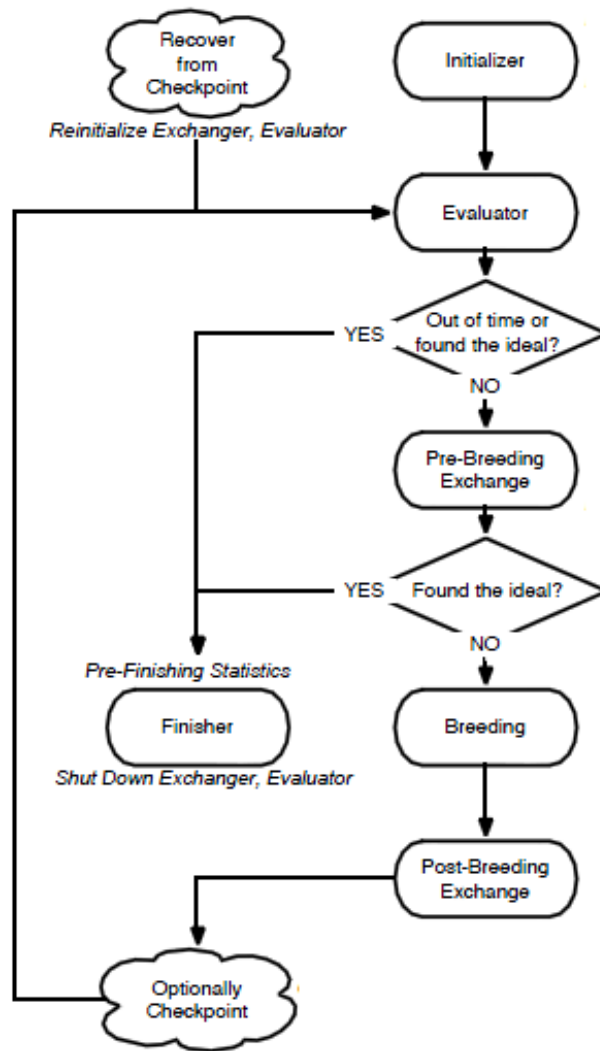
### 3.8 Το ECJ Toolkit

Το ECJ (Evolutionary Computation in Java)[17] είναι μια πλατφόρμα προγραμμάτων - ελεύθερου, ανοικτού κώδικα- γραμμένων σε java και έχει αναπτυχθεί από το Πανεπιστήμιο George Mason, πριν δέκα χρόνια περίπου. Έχει σχεδιαστεί για την ανάπτυξη και εκτέλεση μεγάλων σε όγκο και ιδιαίτερα απαιτητικών πειραμάτων με τη χρήση εξελικτικών αλγορίθμων.

Το ECJ είναι ένα γενικού σκοπού εργαλείο που επιτρέπει την εφαρμογή μεγάλου εύρους συνδυασμών, όπως είναι η επιλογή των ατόμων που θα συμμετέχουν στον εξελικτικό αλγόριθμο , η μέθοδος αναπαραγωγής, τις διαδικασίες αξιολόγησης και επιλογής, τον εξελικτικό αλγόριθμο που θα χρησιμοποιηθεί και τέλος αν ο αλγόριθμος θα εκτελεσθεί παράλληλα ή όχι.

Όπως φαίνεται στην εικόνα 3.5, όπου παρουσιάζεται ο βασικός βρόγχος εκτέλεσης ενός απλού γενετικού αλγορίθμου από το ECJ, βασικές μονάδες για την εκτέλεση του ECJ είναι:

- ο **Initializer** – μονάδα αρχικοποίησης -, όπου βασική εργασία της είναι η δημιουργία και προετοιμασία του πληθυσμού που θα συμμετάσχει στον αλγόριθμο.
- ο **Evaluator** – μονάδα αξιολόγησης -, όπου βασικό του καθήκον της μονάδας αυτής είναι η ανάθεση ποιοτικών χαρακτηριστικών στα μέλη ενός πληθυσμού, με τον υπολογισμό της συνάρτησης καταλληλότητας για το κάθε μέλος.
- ο **Breeder** – μονάδα αναπαραγωγής -, Η μονάδα αυτή δημιουργεί τη νέα γενιά του πληθυσμού από τη παλαιά γενιά, εφαρμόζοντας κάποιους κανόνες επιλογής ή και μετάλλαξης.
- ο **Exchanger** – μονάδα ανταλλαγής-, εξάγει μέλη του πληθυσμούς μιας διαδικασίας ECJ που εκτελείται σε μια άλλη διαδικασία ή μπορεί να εισάγει νέα μέλη στον πληθυσμό της από άλλες διαδικασίες ECJ. Επίσης καλείται για τη Στατιστική επεξεργασία διαφόρων στοιχείων από διάφορα σημεία όπως φαίνεται στην εικόνα 3.5.
- ο **Finalizer** – μονάδα τερματισμού -, Η μονάδα αυτή τερματίζει όλες τις διεργασίες-μονάδες που είναι ενεργές, κλείνει αρχεία και διαγράφει ενδιάμεσα στοιχεία.



Εικόνα 3.5 Διάγραμμα ροής του ECJ για την εκτέλεση απλών γενετικών αλγορίθμων

Το ECJ μπορεί σε κάποιο ενδιάμεσο στάδιο εκτέλεσης να αποθηκεύσει όλες τις παραμέτρους και στοιχεία που ισχύουν μέχρι τότε και να συνεχιστεί η εκτέλεση της διαδικασίας κάποια άλλη στιγμή, από το σημείο που σταμάτησε η διαδικασία.

## 3.9 Εφαρμογή του Γενετικού Αλγόριθμου σε απλά Παραδείγματα

Παρακάτω παρουσιάζονται απλά παραδείγματα στα οποία εφαρμόζεται ο γενετικός αλγόριθμος μέσω του λογισμικού που αναπτύχθηκε στα πλαίσια της παρούσας διατριβής. Σε κάθε εκτέλεση δοκιμάζονται μέρος ή το σύνολο των περιορισμών και παρουσιάζονται τα αποτελέσματα που δίνει το λογισμικό μαζί με το αντίστοιχο πρόγραμμα εξετάσεων όπως αυτό διαμορφώνεται από την επίδραση των περιορισμών. Στόχος αυτής της ενότητας είναι να ελεγχθεί η αποτελεσματικότητα του λογισμικού κάτω από την επίδραση των περιορισμών με τη χρήση κατάλληλων δοκιμαστικών δεδομένων.

### 3.9.1 Ορισμός του Προβλήματος

Στο πρόβλημα που μελετάται χρησιμοποιείται δείγμα με 200 φοιτητές οι οποίοι παρακολουθούν 10 διαφορετικά μαθήματα σε 4 πόλεις. 100 φοιτητές έχουν δηλώσει τα μαθήματα PLHS10, PLHS11, PLHS12, PLHS13 και PLHS14, ενώ οι υπόλοιποι 100 φοιτητές δηλώσανε τα μαθήματα PLHS21, PLHS22, PLHS23, PLHS24 και PLHS25. Το σύνολο των μαθημάτων λοιπόν που εξετάζονται είναι 40 (10 μαθήματα σε 4 πόλεις) και οι εξετάσεις πραγματοποιούνται σε 13 συνολικά αίθουσες εξετάσεων κατανεμημένες στις 4 πόλεις. Αρχικά οι χρονοθυρίδες είναι συνολικά 15 και επιμερίζονται σε 5 συνολικά ημέρες εξετάσεων. Τα δεδομένα εισόδου, εκτός από τις δηλώσεις των μαθημάτων των φοιτητών, που περιέχονται σε αντίστοιχα αρχεία εισόδου είναι:

Αίθουσες Εξετάσεων:

A/A	ΑΙΘΟΥΣΑ	ΠΟΛΗ	ΔΥΝΑΜΙΚΟΤΗΤΑ ΑΙΘΟΥΣΑΣ
0	B1100	ATHENS	50
1	B2200	ATHENS	40
2	B2201	ATHENS	20
3	B3210	ATHENS	100
4	UOM12	THESSALONIKI	30
5	UOM13	THESSALONIKI	25
6	UOM14	THESSALONIKI	20
7	AUTH1001	THESSALONIKI	60
8	AMTH32	PATRA	35
9	AMTH33	PATRA	15
10	PLH1	IOANNINA	25
11	PLH2	IOANNINA	20
12	PLH3	IOANNINA	35

Πίνακας 3.2 Αίθουσες Εξετάσεων

Εξεταζόμενα Μαθήματα:

A/A	ΜΑΘΗΜΑ	ΠΟΛΗ	ΔΗΛΩΣΕΙΣ ΦΟΙΤΗΤΩΝ
0	PLHS10	ATHENS	20
1	PLHS10	THESSALONIKI	55
2	PLHS10	PATRA	13
3	PLHS10	ΙΟΑΝΝΙΝΑ	12
4	PLHS11	ATHENS	20
5	PLHS11	THESSALONIKI	55
6	PLHS11	PATRA	13
7	PLHS11	ΙΟΑΝΝΙΝΑ	12
8	PLHS12	ATHENS	20
9	PLHS12	THESSALONIKI	55
10	PLHS12	PATRA	13
11	PLHS12	ΙΟΑΝΝΙΝΑ	12
12	PLHS13	ATHENS	20
13	PLHS13	THESSALONIKI	55
14	PLHS13	PATRA	13
15	PLHS13	ΙΟΑΝΝΙΝΑ	12
16	PLHS14	ATHENS	20
17	PLHS14	THESSALONIKI	55
18	PLHS14	PATRA	13
19	PLHS14	ΙΟΑΝΝΙΝΑ	12
20	PLHS21	ATHENS	35
21	PLHS21	THESSALONIKI	21
22	PLHS21	PATRA	24
23	PLHS21	ΙΟΑΝΝΙΝΑ	20
24	PLHS22	ATHENS	35
25	PLHS22	THESSALONIKI	21
26	PLHS22	PATRA	24
27	PLHS22	ΙΟΑΝΝΙΝΑ	20
28	PLHS23	ATHENS	35
29	PLHS23	THESSALONIKI	21
30	PLHS23	PATRA	24
31	PLHS23	ΙΟΑΝΝΙΝΑ	20
32	PLHS24	ATHENS	35
33	PLHS24	THESSALONIKI	21
34	PLHS24	PATRA	24
35	PLHS24	ΙΟΑΝΝΙΝΑ	20
36	PLHS25	ATHENS	35
37	PLHS25	THESSALONIKI	21
38	PLHS25	PATRA	24
39	PLHS25	ΙΟΑΝΝΙΝΑ	20

Πίνακας 3.3 Εξεταζόμενα Μαθήματα

Χρονοθυρίδες Εξετάσεων:

A/A ΧΡΟΝΟΘΥΡΙΔΑΣ	A/A ΗΜΕΡΑΣ	ΩΡΕΣ ΕΞΕΤΑΣΕΩΝ
0	1	09:00-12:00
1	1	13:00-16:00
2	1	17:00-20:00
3	2	09:00-12:00
4	2	13:00-16:00
5	2	17:00-20:00
6	3	09:00-12:00
7	3	13:00-16:00
8	3	17:00-20:00
9	4	09:00-12:00
10	4	13:00-16:00
11	4	17:00-20:00
12	5	09:00-12:00
13	5	13:00-16:00
14	5	17:00-20:00

Πίνακας 3.4 Χρονοθυρίδες Εξετάσεων

Εισηγητές Καθηγητές<sup>1</sup>:

A/A	ΟΝΟΜΑ	ΕΠΙΘΕΤΟ	ΜΑΘΗΜΑ	ΠΟΛΗ
0	Kostantinos	Papadopoulos	PLHS10	ATHENS
1	Nikolaos	Antoniu	PLHS11	ATHENS
2	Anna	Vagena	PLHS12	ATHENS
3	Iason	Petridis	PLHS13	ATHENS
4	Theodoros	Kalogiannis	PLHS14	ATHENS
5	Ioannis	Melitos	PLHS10	THESSALONIKI
6	Kostantinos	Papadopoulos	PLHS11	THESSALONIKI
7	Xrysoula	Dimopoulou	PLHS12	THESSALONIKI
8	Eirini	Antoniu	PLHS13	THESSALONIKI
9	Stelios	Orfanos	PLHS14	THESSALONIKI
10	Aikaterini	Stamatiou	PLHS10	PATRA
11	Petros	Ioannou	PLHS11	PATRA
12	Kostantinos	Kalimeris	PLHS12	PATRA
13	Nikolaos	Antoniu	PLHS13	PATRA
14	Anna	Georgiou	PLHS14	PATRA
15	Theodoros	Amanatidis	PLHS10	IOANNINA
16	Ioannis	Atsalos	PLHS11	IOANNINA
17	Georgios	Petrou	PLHS12	IOANNINA
18	Paulos	Keramidas	PLHS13	IOANNINA
19	Maria	Xristou	PLHS14	IOANNINA
20	Stelios	Melaxrinos	PLHS21	ATHENS
21	Konstantinos	krasas	PLHS22	ATHENS
22	Petros	Xristodoulou	PLHS23	ATHENS
23	Konstantinos	krasas	PLHS24	ATHENS
24	Nikolaos	Papadopoulos	PLHS25	ATHENS
25	Anna	Stamatopoulos	PLHS21	THESSALONIKI
26	Panagiotis	Petridis	PLHS22	THESSALONIKI
27	Theodoros	Stamatakis	PLHS23	THESSALONIKI
28	Ioannis	Arxontidis	PLHS24	THESSALONIKI
29	Georgios	Maronidis	PLHS25	THESSALONIKI
30	Xrysoula	Giraki	PLHS21	PATRA
31	Xristos	Aronis	PLHS22	PATRA
32	Georgios	Maronidis	PLHS23	PATRA
33	Aikaterini	Kokkinou	PLHS24	PATRA
34	Panagiotis	Petinakis	PLHS25	PATRA
35	Kostantinos	Eroglou	PLHS21	IOANNINA
36	Periklis	Zisis	PLHS22	IOANNINA
37	Georgia	Antoniu	PLHS23	IOANNINA
38	Georgia	Antoniu	PLHS24	IOANNINA
39	Antonia	Peraki	PLHS25	IOANNINA

Πίνακας 3.5 Εισηγητές Καθηγητές

Λαμβάνοντας υπόψη τα παραπάνω δεδομένα εισόδου και τον πίνακα 3.1 της ενότητας 3.5 που περιέχει το κόστος που συνεπάγεται από την παραβίαση του κάθε περιορισμού εκτελούμε το πρόγραμμα για την εξεύρεση ενός αποδεκτού εξεταστικού προγράμματος.

<sup>1</sup> Τα ονόματα των καθηγητών είναι τυχαία και δεν έχουν καμιά σχέση με την πραγματικότητα.

### 3.9.2 1ο Σενάριο

Σε πρώτη φάση επιλέχθηκε να μην ισχύσουν οι περιορισμοί 5 και 6 του πίνακα 3.1 της ενότητας 3.5 δηλαδή να μην έχουν άλλη εξέταση οι φοιτητές την επόμενη μέρα των εξετάσεων ή μετά δύο μέρες λόγω ότι οι εξετάσεις διαρκούν 5 ημέρες και θα είναι αδύνατο να τηρηθούν αυτοί οι δύο περιορισμοί.

Κατά τη 366η γενεά βρέθηκε ιδανική λύση (βέλτιστη λύση) δίνοντας στη συνάρτηση καταλληλότητας τιμή 1, σύμφωνα με τους περιορισμούς που τέθηκαν σε ισχύ, οι μεταβλητές παραβίασης των περιορισμών (πίνακας 3.1, ενότητα 3.5) είχαν τιμή  $realfit= 0.0$  (συνολικό κόστος παραβίασης περιορισμών),  $lesSameHour= 0$ ,  $multiRoomCapacity= 0$ ,  $wrongLessonRoom= 0$ ,  $stdSameHour= 0$ ,  $ExmSameHour= 0$ ,  $lesSameDay= 0$ , ενώ το εξεταστικό πρόγραμμα που έδωσε η εκτέλεση είναι το παρακάτω:

EXAMINATION		TIMETABLING		SCHEDULE							
ts	l	day	time	Lesson	City	Room	StudentsInRoom	RoomCapacity	LessonCapacity	Examiner	
1	1	1	13:00-16:00	PLHS11	ATHENS	B2201	20	20	20	Nikolaos	Antoniou
1	1	1	13:00-16:00	PLHS11	THESSALONIKI	AUTH1001	55	60	55	Kostantinos	Papadopoulos
1	1	1	13:00-16:00	PLHS11	PATRA	AMH33	13	15	13	Petros	Ioannou
1	1	1	13:00-16:00	PLHS11	IOANNINA	PLH2	12	20	12	Ioannis	Atsalos
1	1	1	13:00-16:00	PLHS22	ATHENS	B2200	35	40	35	Konstantinos	krasas
1	1	1	13:00-16:00	PLHS22	THESSALONIKI	UOM13	21	25	21	Panagiotis	Petridis
1	1	1	13:00-16:00	PLHS22	PATRA	AMH32	24	35	24	Xristos	Aronis
1	1	1	13:00-16:00	PLHS22	IOANNINA	PLH1	20	25	20	Periklis	Zisis
5	2	17:00-20:00	PLHS12	ATHENS	B2201	20	20	20	20	Anna	Vagena
5	2	17:00-20:00	PLHS12	THESSALONIKI	AUTH1001	55	60	55	55	Xrysoula	Dimopoulou
5	2	17:00-20:00	PLHS12	PATRA	AMH33	13	15	13	13	Kostantinos	Kalimeris
5	2	17:00-20:00	PLHS12	IOANNINA	PLH2	12	20	12	12	Georgios	Petrou
5	2	17:00-20:00	PLHS23	ATHENS	B2200	35	40	35	35	Petros	Xristodoulou
5	2	17:00-20:00	PLHS23	THESSALONIKI	UOM13	21	25	21	21	Theodoros	Stamatakis
5	2	17:00-20:00	PLHS23	PATRA	AMH32	24	35	24	24	Georgios	Maronidis
5	2	17:00-20:00	PLHS23	IOANNINA	PLH1	20	25	20	20	Georgia	Antoniou
7	3	13:00-16:00	PLHS14	ATHENS	B2201	20	20	20	20	Theodoros	Kalogiannis
7	3	13:00-16:00	PLHS14	THESSALONIKI	AUTH1001	55	60	55	55	Stelios	Orfanos
7	3	13:00-16:00	PLHS14	PATRA	AMH33	13	15	13	13	Anna	Georgiou
7	3	13:00-16:00	PLHS14	IOANNINA	PLH2	12	20	12	12	Maria	Xristou
8	3	17:00-20:00	PLHS25	ATHENS	B2200	35	40	35	35	Nikolaos	Papadopoulos
8	3	17:00-20:00	PLHS25	THESSALONIKI	UOM13	21	25	21	21	Georgios	Maronidis
8	3	17:00-20:00	PLHS25	PATRA	AMH32	24	35	24	24	Panagiotis	Petinakis
8	3	17:00-20:00	PLHS25	IOANNINA	PLH2	20	20	20	20	Antonia	Peraki
9	4	09:00-12:00	PLHS13	ATHENS	B2201	20	20	20	20	Iason	Petridis
9	4	09:00-12:00	PLHS13	THESSALONIKI	AUTH1001	55	60	55	55	Eirini	Antoniou
9	4	09:00-12:00	PLHS13	PATRA	AMH33	13	15	13	13	Nikolaos	Antoniou
9	4	09:00-12:00	PLHS13	IOANNINA	PLH2	12	20	12	12	Paulos	Keramidas
10	4	13:00-16:00	PLHS24	ATHENS	B2200	35	40	35	35	Konstantinos	krasas
10	4	13:00-16:00	PLHS24	THESSALONIKI	UOM13	21	25	21	21	Ioannis	Arxontidis
10	4	13:00-16:00	PLHS24	PATRA	AMH32	24	35	24	24	Aikaterini	Kokkiniou
10	4	13:00-16:00	PLHS24	IOANNINA	PLH2	20	20	20	20	Georgia	Antoniou
14	5	17:00-20:00	PLHS10	ATHENS	B2201	20	20	20	20	Kostantinos	Papadopoulos
14	5	17:00-20:00	PLHS10	THESSALONIKI	AUTH1001	55	60	55	55	Ioannis	Melitos
14	5	17:00-20:00	PLHS10	PATRA	AMH33	13	15	13	13	Aikaterini	Stamatiou
14	5	17:00-20:00	PLHS10	IOANNINA	PLH2	12	20	12	12	Theodoros	Amanatidis
14	5	17:00-20:00	PLHS21	ATHENS	B2200	35	40	35	35	Stelios	Melaxrinos
14	5	17:00-20:00	PLHS21	THESSALONIKI	UOM13	21	25	21	21	Anna	Stamatopoulos
14	5	17:00-20:00	PLHS21	PATRA	AMH32	24	35	24	24	Xrysoula	Giraki
14	5	17:00-20:00	PLHS21	IOANNINA	PLH1	20	25	20	20	Kostantinos	Eroglou

### 3.9.3 2ο Σενάριο:

Στη συνέχεια επιλέχθηκε να ισχύσουν όλοι οι περιορισμοί του πίνακα 3.1 της ενότητας 3.5 ενώ οι εξετάσεις διαρκούν 5 ημέρες και για 1000 γενεές.

Στην 63η γενεά, βρέθηκε καλύτερη λύση χωρίς να είναι βέλτιστη με τιμές στις μεταβλητές  $realfit= 145000.0$  (συνολικό κόστος παραβίασης περιορισμών),  $lesSameHour= 0$ ,



multiRoomCapacity= 0, wrongLessonRoom= 0, stdSameHour= 0, ExmSameHour= 0, lesSameDay= 400, lesNextDay= 500, lesNextTwoDays= 300.

Όπως φαίνεται από τις τιμές των μεταβλητών αλλά και από το παρακάτω πρόγραμμα εξετάσεων που προέκυψε, το κόστος παραβίασης περιορισμών προέκυψε επειδή:

1. Σε 400 ( $400 \times 200 = 80000$  κόστος) περιπτώσεις εξετάζονται φοιτητές την ίδια μέρα σε 2 τουλάχιστον μαθήματα. Την 1<sup>η</sup> ημέρα εξετάζονται τα μαθήματα PLHS21 και PLHS24, από 100 ίδιους φοιτητές στο μάθημα σύνολο 200 περιπτώσεις παραβίασης του συγκεκριμένου περιορισμού συν την 4<sup>η</sup> ημέρα που εξετάζονται τα μαθήματα PLHS13 και PLHS10, από 100 ίδιοι φοιτητές στο κάθε μάθημα σύνολο άλλες 200 περιπτώσεις παραβίασης του συγκεκριμένου περιορισμού.
2. Σε 500 ( $500 \times 100 = 50000$  κόστος) περιπτώσεις γίνονται εξετάσεις ξανά την επόμενη μέρα. Την 2<sup>η</sup> ημέρα εξετάζεται το μάθημα PLHS23 ενώ την 1<sup>η</sup> μέρα εξεταζόταν τα μαθήματα PLHS21, και PLHS24, έτσι έχουμε 100 παραβιάσεις του αντίστοιχου περιορισμού. Την 4<sup>η</sup> ημέρα εξετάζονται τα μαθήματα PLHS13 και PLHS14 ενώ την 3<sup>η</sup> εξετάζεται το μάθημα PLHS14, άλλες 200 παραβιάσεις του συγκεκριμένου περιορισμού. Τέλος την 5<sup>η</sup> ημέρα εξετάζονται τα μαθήματα PLHS22 και PLHS11 ενώ την 4<sup>η</sup> ημέρα τα μαθήματα PLHS10 και PLHS25, δηλαδή άλλες 200 παραβιάσεις του περιορισμού.
3. 300 ( $300 \times 50 = 15000$  κόστος) φορές παραβιάστηκε ο περιορισμός ότι δεν θα πρέπει να υπάρχει εξεταστική 2 μέρες μετά από εξετάσεις. Αυτό συμβαίνει την 3<sup>η</sup> ημέρα όπου εξετάζεται το μάθημα PLHS14 ενώ την 1<sup>η</sup> ημέρα εξεταζόταν το μάθημα PLHS12 (100 παραβιάσεις), την 4<sup>η</sup> ημέρα επίσης εξετάζεται το μάθημα PLHS25 ενώ την 2<sup>η</sup> ημέρα εξεταζόταν το μάθημα PLHS23 (άλλες 100 παραβιάσεις του αντίστοιχου περιορισμού) και τέλος την 5<sup>η</sup> ημέρα εξετάζεται το μάθημα PLHS11 ενώ την 3<sup>η</sup> ημέρα εξεταζόταν το μάθημα PLHS14 (άλλες 100 παραβιάσεις).

Αθροίζοντας το κόστος των παραβιάσεων όπως προέκυψε παραπάνω υπολογίζεται το συνολικό κόστος 145000.

EXAMINATION		TIMETABLING		SCHEDULE		StudentsInRoom	RoomCapacity	LessonCapacity	Examiner	
ts\l	day	time	Lesson	City	Room					
0	1	09:00-12:00	PLHS12	ATHENS	B2201	20	20	20	Anna	Vagena
0	1	09:00-12:00	PLHS12	THESSALONIKI	AUTH1001	55	60	55	Xrysoula	Dimopoulou
0	1	09:00-12:00	PLHS12	PATRA	AMTH33	13	15	13	Kostantinos	Kalimeris
0	1	09:00-12:00	PLHS12	IOANNINA	PLH2	12	20	12	Georgios	Petrou
0	1	09:00-12:00	PLHS21	ATHENS	B2200	35	40	35	Stelios	Melaxrinos
0	1	09:00-12:00	PLHS21	THESSALONIKI	UOM13	21	25	21	Anna	Stamatopoulos
0	1	09:00-12:00	PLHS21	PATRA	AMTH32	24	35	24	Xrysoula	Giraki
0	1	09:00-12:00	PLHS21	IOANNINA	PLH1	20	25	20	Kostantinos	Eroglou
2	1	17:00-20:00	PLHS24	ATHENS	B2200	35	40	35	Konstantinos	Krasas
2	1	17:00-20:00	PLHS24	THESSALONIKI	UOM13	21	25	21	Ioannis	Arxontidis
2	1	17:00-20:00	PLHS24	PATRA	AMTH32	24	35	24	Aikaterini	Kokkinou
2	1	17:00-20:00	PLHS24	IOANNINA	PLH2	12	20	12	Georgia	Antoniou
5	2	17:00-20:00	PLHS23	ATHENS	B2200	35	40	35	Petros	Xristodoulou
5	2	17:00-20:00	PLHS23	THESSALONIKI	UOM13	21	25	21	Theodoros	Stamatakis
5	2	17:00-20:00	PLHS23	PATRA	AMTH32	24	35	24	Georgios	Maronidis
5	2	17:00-20:00	PLHS23	IOANNINA	PLH2	12	20	12	Georgia	Antoniou
7	3	13:00-16:00	PLHS14	ATHENS	B2201	20	20	20	Theodoros	Kalogiannis
7	3	13:00-16:00	PLHS14	THESSALONIKI	AUTH1001	55	60	55	Stelios	Orfanos
7	3	13:00-16:00	PLHS14	PATRA	AMTH33	13	15	13	Anna	Georgiou
7	3	13:00-16:00	PLHS14	IOANNINA	PLH2	12	20	12	Marla	Xristou
9	4	09:00-12:00	PLHS13	ATHENS	B2201	20	20	20	Iason	Petridis
9	4	09:00-12:00	PLHS13	THESSALONIKI	AUTH1001	55	60	55	Eirini	Antoniou
9	4	09:00-12:00	PLHS13	PATRA	AMTH33	13	15	13	Nikolaos	Antoniou
9	4	09:00-12:00	PLHS13	IOANNINA	PLH2	12	20	12	Paulos	Keramidas
11	4	17:00-20:00	PLHS10	ATHENS	B2201	20	20	20	Kostantinos	Papadopoulos
11	4	17:00-20:00	PLHS10	THESSALONIKI	AUTH1001	55	60	55	Ioannis	Melittos
11	4	17:00-20:00	PLHS10	PATRA	AMTH33	13	15	13	Aikaterini	Stamatiou
11	4	17:00-20:00	PLHS10	IOANNINA	PLH2	12	20	12	Theodoros	Amanatidis
11	4	17:00-20:00	PLHS25	ATHENS	B2200	35	40	35	Nikolaos	Papadopoulos
11	4	17:00-20:00	PLHS25	THESSALONIKI	UOM13	21	25	21	Georgios	Maronidis
11	4	17:00-20:00	PLHS25	PATRA	AMTH32	24	35	24	Panagiotis	Petinakis
11	4	17:00-20:00	PLHS25	IOANNINA	PLH1	20	25	20	Antonia	Peraki
13	5	13:00-16:00	PLHS11	ATHENS	B2201	20	20	20	Nikolaos	Antoniou
13	5	13:00-16:00	PLHS11	THESSALONIKI	AUTH1001	55	60	55	Kostantinos	Papadopoulos
13	5	13:00-16:00	PLHS11	PATRA	AMTH33	13	15	13	Petros	Ioannou
13	5	13:00-16:00	PLHS11	IOANNINA	PLH2	12	20	12	Ioannis	Atsalos
13	5	13:00-16:00	PLHS22	ATHENS	B2200	35	40	35	Konstantinos	Krasas
13	5	13:00-16:00	PLHS22	THESSALONIKI	UOM13	21	25	21	Panagiotis	Petridis
13	5	13:00-16:00	PLHS22	PATRA	AMTH32	24	35	24	Xristos	Aronis
13	5	13:00-16:00	PLHS22	IOANNINA	PLH1	20	25	20	Periklis	Zisis

### 3.9.4 3ο Σενάριο:

Όπως φάνηκε παραπάνω όταν οι εξετάσεις διαρκούν 5 ημέρες είναι αδύνατο να πετύχουμε βέλτιστη λύση. Για να βελτιωθεί λίγο η κατάσταση επιλέγεται οι εξετάσεις αρχικά να διαρκέσουν επιπλέον 5 ημέρες, συνολικά δηλαδή να έχουν διάρκεια 10 ημέρες και 30 χρονοθυρίδες. Παρόλο που δεν αρκούν ούτε οι 10 μέρες για να έχουμε βέλτιστη λύση, το αποτέλεσμα για 1000 γενεές είναι:

Στην 48<sup>η</sup> μόλις γενεά εντοπίστηκε τοπικό ελάχιστο με κόστος 30000 και τιμή στις μεταβλητές παραβίασης των περιορισμών: lesSameHour= 0, multiRoomCapacity= 0, wrongLessonRoom= 0, stdSameHour= 0, ExmSameHour= 0, lesSameDay= 0, lesNextDay= 200 και lesNextTwoDays= 200. Παρόλο που βελτιώθηκαν τα αποτελέσματα σε σχέση με το 2<sup>ο</sup> Σενάριο, δεν έχει εντοπισθεί βέλτιστη λύση διότι η διάρκεια των εξετάσεων των 10 ημερών δεν είναι αρκετή για βέλτιστη λύση. Για 5 μαθήματα ένας φοιτητής θα έπρεπε να έχει διάρκεια εξετάσεων 13 ημέρες τουλάχιστον για να ικανοποιούνται όλοι οι περιορισμοί.

EXAMINATION		TIMETABLING		SCHEDULE		StudentsInRoom	RoomCapacity	LessonCapacity	Examiner	
ts\l	day	time	Lesson	City	Room					
1	1	13:00-16:00	PLHS21	ATHENS	B2200	35	40	35	Stelios	Melaxrinos
1	1	13:00-16:00	PLHS21	THESSALONIKI	UOM13	21	25	21	Anna	Stamatopoulos
1	1	13:00-16:00	PLHS21	PATRA	AMTH32	24	35	24	Xrysoula	Giraki
1	1	13:00-16:00	PLHS21	IOANNINA	PLH2	12	20	12	Kostantinos	Eroglou
2	1	17:00-20:00	PLHS13	ATHENS	B2201	20	20	20	Iason	Petridis
2	1	17:00-20:00	PLHS13	THESSALONIKI	AUTH1001	55	60	55	Eirini	Antoniou
2	1	17:00-20:00	PLHS13	PATRA	AMTH33	13	15	13	Nikolaos	Antoniou
2	1	17:00-20:00	PLHS13	IOANNINA	PLH2	12	20	12	Paulos	Keramidas
5	2	17:00-20:00	PLHS24	ATHENS	B2200	35	40	35	Konstantinos	Krasas
5	2	17:00-20:00	PLHS24	THESSALONIKI	UOM13	21	25	21	Ioannis	Arxontidis
5	2	17:00-20:00	PLHS24	PATRA	AMTH32	24	35	24	Aikaterini	Kokkinou
5	2	17:00-20:00	PLHS24	IOANNINA	PLH2	12	20	12	Georgia	Antoniou
11	4	17:00-20:00	PLHS14	ATHENS	B2201	20	20	20	Theodoros	Kalogiannis
11	4	17:00-20:00	PLHS14	THESSALONIKI	AUTH1001	55	60	55	Stelios	Orfanos
11	4	17:00-20:00	PLHS14	PATRA	AMTH33	13	15	13	Anna	Georgiou
11	4	17:00-20:00	PLHS14	IOANNINA	PLH2	12	20	12	Marla	Xristou
14	5	17:00-20:00	PLHS22	ATHENS	B2200	35	40	35	Konstantinos	Krasas
14	5	17:00-20:00	PLHS22	THESSALONIKI	UOM13	21	25	21	Panagiotis	Petridis
14	5	17:00-20:00	PLHS22	PATRA	AMTH32	24	35	24	Xristos	Aronis
14	5	17:00-20:00	PLHS22	IOANNINA	PLH2	12	20	12	Periklis	Zisis
15	6	09:00-12:00	PLHS12	ATHENS	B2201	20	20	20	Anna	Vagena
15	6	09:00-12:00	PLHS12	THESSALONIKI	AUTH1001	55	60	55	Xrysoula	Dimopoulou
15	6	09:00-12:00	PLHS12	PATRA	AMTH33	13	15	13	Kostantinos	Kalimeris
15	6	09:00-12:00	PLHS12	IOANNINA	PLH2	12	20	12	Georgios	Petrou

20	7	17:00-20:00	PLHS11	ATHENS	B2201	20	20	Nikolaos	Antoniou
20	7	17:00-20:00	PLHS11	THESSALONIKI	AUTH1001	55	60	Kostantinos	Papadopoulos
20	7	17:00-20:00	PLHS11	PATRA	AMTH33	13	15	Petros	Ioannou
20	7	17:00-20:00	PLHS11	IOANNINA	PLH2	12	20	Ioannis	Atsalos
22	8	13:00-16:00	PLHS25	ATHENS	B2200	35	40	Nikolaos	Papadopoulos
22	8	13:00-16:00	PLHS25	THESSALONIKI	UOM13	21	25	Georgios	Maronidis
22	8	13:00-16:00	PLHS25	PATRA	AMTH32	24	35	Panagiotis	Petinakis
22	8	13:00-16:00	PLHS25	IOANNINA	PLH2	20	20	Antonia	Peraki
28	10	13:00-16:00	PLHS23	ATHENS	B2200	35	40	Petros	Xristodoulou
28	10	13:00-16:00	PLHS23	THESSALONIKI	UOM13	21	25	Theodoros	Stamatakis
28	10	13:00-16:00	PLHS23	PATRA	AMTH32	24	35	Georgios	Maronidis
28	10	13:00-16:00	PLHS23	IOANNINA	PLH2	20	20	Georgia	Antoniou
29	10	17:00-20:00	PLHS10	ATHENS	B2201	20	20	Kostantinos	Papadopoulos
29	10	17:00-20:00	PLHS10	THESSALONIKI	AUTH1001	55	60	Ioannis	Melitos
29	10	17:00-20:00	PLHS10	PATRA	AMTH33	13	15	Aikaterini	Stamatiou
29	10	17:00-20:00	PLHS10	IOANNINA	PLH2	12	20	Theodoros	Amanatidis

### 3.9.5 4ο Σενάριο:

Δοκιμάζοντας επιπλέον οι εξετάσεις να διαρκέσουν συνολικά 15 ημέρες, με τα ίδια υπόλοιπα δεδομένα. Το αποτέλεσμα για 1000 γενεές είναι:

Στην 51<sup>η</sup> γενεά εντοπίστηκε τοπικό ελάχιστο με κόστος 20000 και τιμές στις μεταβλητές παραβίασης των περιορισμών:  $lesSameHour=0$ ,  $multiRoomCapacity=0$ ,  $wrongLessonRoom=0$ ,  $stdSameHour=0$ ,  $ExmSameHour=0$ ,  $lesSameDay=0$ ,  $lesNextDay=100$  και  $lesNextTwoDays=200$ .

Σε σχέση με το 3ο Σενάριο σαφώς υπάρχει μια βελτίωση στη μείωση του κόστους που οφείλεται στην μείωση κατά 100 παραβιάσεις του περιορισμού που δεν επιτρέπει να γίνονται εξετάσεις δύο διαδοχικές ημέρες.

EXAMINATION		TIMETABLING		SCHEDULE					
ts1t	day	time	Lesson	City	Room	StudentsInRoom	RoomCapacity	LessonCapacity	Examiner
7	3	13:00-16:00	PLHS14	ATHENS	B2201	20	20	20	Theodoros
7	3	13:00-16:00	PLHS14	THESSALONIKI	AUTH1001	55	60	55	Stelios
7	3	13:00-16:00	PLHS14	PATRA	AMTH33	13	15	13	Anna
7	3	13:00-16:00	PLHS14	IOANNINA	PLH2	12	20	12	Maria
7	3	13:00-16:00	PLHS24	ATHENS	B2200	35	40	35	Kostantinos
7	3	13:00-16:00	PLHS24	THESSALONIKI	UOM13	21	25	21	Ioannis
7	3	13:00-16:00	PLHS24	PATRA	AMTH32	24	35	24	Aikaterini
7	3	13:00-16:00	PLHS24	IOANNINA	PLH1	20	25	20	Georgia
15	6	09:00-12:00	PLHS12	ATHENS	B2201	20	20	20	Anna
15	6	09:00-12:00	PLHS12	THESSALONIKI	AUTH1001	55	60	55	Xrysoula
15	6	09:00-12:00	PLHS12	PATRA	AMTH33	13	15	13	Kostantinos
15	6	09:00-12:00	PLHS12	IOANNINA	PLH2	12	20	12	Georgios
17	6	17:00-20:00	PLHS23	ATHENS	B2200	35	40	35	Petros
17	6	17:00-20:00	PLHS23	THESSALONIKI	UOM13	21	25	21	Theodoros
17	6	17:00-20:00	PLHS23	PATRA	AMTH32	24	35	24	Georgios
17	6	17:00-20:00	PLHS23	IOANNINA	PLH2	20	20	20	Georgia
21	8	09:00-12:00	PLHS11	ATHENS	B2201	20	20	20	Nikolaos
21	8	09:00-12:00	PLHS11	THESSALONIKI	AUTH1001	55	60	55	Kostantinos
21	8	09:00-12:00	PLHS11	PATRA	AMTH33	13	15	13	Petros
21	8	09:00-12:00	PLHS11	IOANNINA	PLH2	12	20	12	Ioannis
21	8	09:00-12:00	PLHS21	ATHENS	B2200	35	40	35	Stelios
21	8	09:00-12:00	PLHS21	THESSALONIKI	UOM13	21	25	21	Anna
21	8	09:00-12:00	PLHS21	PATRA	AMTH32	24	35	24	Xrysoula
21	8	09:00-12:00	PLHS21	IOANNINA	PLH1	20	25	20	Kostantinos
26	9	17:00-20:00	PLHS25	ATHENS	B2200	35	40	35	Nikolaos
26	9	17:00-20:00	PLHS25	THESSALONIKI	UOM13	21	25	21	Georgios
26	9	17:00-20:00	PLHS25	PATRA	AMTH32	24	35	24	Panagiotis
26	9	17:00-20:00	PLHS25	IOANNINA	PLH2	20	20	20	Antonia
34	12	13:00-16:00	PLHS13	ATHENS	B2201	20	20	20	Iason
34	12	13:00-16:00	PLHS13	THESSALONIKI	AUTH1001	55	60	55	Eirini
34	12	13:00-16:00	PLHS13	PATRA	AMTH33	13	15	13	Nikolaos
34	12	13:00-16:00	PLHS13	IOANNINA	PLH2	12	20	12	Paulos
42	15	09:00-12:00	PLHS22	ATHENS	B2200	35	40	35	Kostantinos
42	15	09:00-12:00	PLHS22	THESSALONIKI	UOM13	21	25	21	Panagiotis
42	15	09:00-12:00	PLHS22	PATRA	AMTH32	24	35	24	Xristos
42	15	09:00-12:00	PLHS22	IOANNINA	PLH2	20	20	20	Periklis
43	15	13:00-16:00	PLHS10	ATHENS	B2201	20	20	20	Kostantinos
43	15	13:00-16:00	PLHS10	THESSALONIKI	AUTH1001	55	60	55	Ioannis
43	15	13:00-16:00	PLHS10	PATRA	AMTH33	13	15	13	Aikaterini
43	15	13:00-16:00	PLHS10	IOANNINA	PLH2	12	20	12	Theodoros

Από τα παραπάνω σενάρια εκτέλεσης του προγράμματος που υλοποιεί τον γενετικό αλγόριθμο με τη χρήση δοκιμαστικών δεδομένων προκύπτουν τα εξής συμπεράσματα:

- Εντοπίζεται σχετικά γρήγορα βέλτιστη λύση εάν ισχύουν όλοι οι περιορισμοί όπως αυτοί τέθηκαν στον πίνακα 3.1 της ενότητας 3.5 εκτός των δύο ελαστικών περιορισμών που δεν επιτρέπουν νέα εξέταση μαθήματος, μια ή δυο μέρες μετά από άλλη εξέταση.
- Όσες επιπλέον ημέρες υπάρχουν σε μια εξεταστική περίοδο τόσο καλύτερα αποτελέσματα προκύπτουν και το εξεταστικό πρόγραμμα γίνεται καλύτερο.
- Το πρόγραμμα που υλοποιεί τον γενετικό αλγόριθμο παγιδεύεται σε τοπικό ελάχιστο και έτσι δεν υπάρχει πιθανότητα να εντοπισθεί το ολικό ελάχιστο. Αυτό οφείλεται στο υψηλότερο κόστος που υπάρχει για την παραβίαση των ανελαστικών περιορισμών σε σχέση με τους ελαστικούς περιορισμούς. Έτσι ικανοποιούνται πρώτα οι ανελαστικοί περιορισμοί και μετά γίνεται προσπάθεια να ικανοποιηθούν και οι ελαστικοί περιορισμοί. Αυτό όμως έχει ως συνέπεια να μην γίνεται καμιά αλλαγή στο πρόγραμμα αφού κάθε νέα αλλαγή που θα ικανοποιούσε έναν ελαστικό περιορισμό οδηγεί σε παραβίαση ενός άλλου ανελαστικού περιορισμού και κατά συνέπεια να αυξάνει το κόστος αντί να μειώνεται.
- Η μετάλλαξη (mutation) πολύ δύσκολα μπορεί να βοηθήσει στην περίπτωση που πρέπει να μετακινηθούν ταυτόχρονα 3 για παράδειγμα μαθήματα και να μετακινηθεί η εξέτασή τους σε άλλη χρονοθυρίδα αφού η πιθανότητα να συμβεί κάτι τέτοιο είναι ελάχιστη. Έτσι εάν η πιθανότητα να συμβεί μετάλλαξη σε κάποιο γονίδιο είναι 3% και να του αποδοθεί μια συγκεκριμένη χρονοθυρίδα, εάν οι χρονοθυρίδες είναι 15, δηλαδή συνολικά έχουμε πιθανότητα  $0,03 \cdot (1/15) = 0,2\%$ . Εάν η συγκεκριμένη μετάλλαξη πρέπει να συμβεί ταυτόχρονα σε τρία γονίδια τότε θα μπορούσε να συμβεί με πιθανότητα  $0,002 \cdot 0,002 \cdot 0,002 = 0,000000008$  ή  $0,00000008\%$  δηλαδή 8 πιθανότητες στο δισεκατομμύριο.

### 3.9.6 5ο Σενάριο:

Δοκιμάζοντας ξανά το πρόγραμμα, οι εξετάσεις να διαρκέσουν συνολικά 15 ημέρες, όπου στις πρώτες 200 γενεές του γενετικού αλγορίθμου οι περιορισμοί 4, 5 και 6 (Οι φοιτητές δεν θα πρέπει να εξετάζονται την ίδια μέρα σε περισσότερα από ένα μαθήματα, Οι φοιτητές την επόμενη μέρα μετά από μια εξέταση να μην έχουν εξέταση σε άλλο μάθημα, Οι φοιτητές δύο

μέρες μετά από μια εξέταση να μην έχουν εξέταση σε άλλο μάθημα) πίνακα 3.1 της ενότητας 3.5 να έχουν **10πλάσια βαρύτητα**, με τα ίδια υπόλοιπα δεδομένα. Το αποτέλεσμα για 1000 γενεές, κατά τη δεύτερη εκτέλεση, είναι:

Κατά την 249<sup>η</sup> γενεά βρέθηκε ολικό ελάχιστο, με κόστος παραβίασης περιορισμών 0 και τιμή στη συνάρτηση καταλληλότητας ίση με 1, δεν παραβιάζεται δηλαδή κανείς περιορισμός.

EXAMINATION		TIMETABLING		SCHEDULE									
ts	day	time	Lesson	City	Room	Students	InRoom	RoomCapacity	LessonCapacity	Examiner			
0	1	09:00-12:00	PLHS22	ATHENS	B2200	35	40	40	35	Konstantinos			krasas
0	1	09:00-12:00	PLHS22	THESSALONIKI	UOM13	21	25	25	21	Panagiotis			Petridis
0	1	09:00-12:00	PLHS22	PATRA	AMTH32	24	35	35	24	Xristos			Aronis
0	1	09:00-12:00	PLHS22	IOANNINA	PLH2	20	20	20	20	Periklis			Zisis
2	1	17:00-20:00	PLHS11	ATHENS	B2201	20	20	20	20	Nikolaos			Antoniu
2	1	17:00-20:00	PLHS11	THESSALONIKI	AUTH1001	55	60	60	55	Kostantinos			Papadopoulos
2	1	17:00-20:00	PLHS11	PATRA	AMTH33	13	15	15	13	Petros			Ioannou
2	1	17:00-20:00	PLHS11	IOANNINA	PLH2	12	20	20	12	Ioannis			Atsalos
11	4	17:00-20:00	PLHS23	ATHENS	B2200	35	40	40	35	Petros			Xristodoulou
11	4	17:00-20:00	PLHS23	THESSALONIKI	UOM13	21	25	25	21	Theodoros			Stamatakis
11	4	17:00-20:00	PLHS23	PATRA	AMTH32	24	35	35	24	Georgios			Maronidis
11	4	17:00-20:00	PLHS23	IOANNINA	PLH2	20	20	20	20	Georgia			Antoniu
13	5	13:00-16:00	PLHS13	ATHENS	B2201	20	20	20	20	Iason			Petridis
13	5	13:00-16:00	PLHS13	THESSALONIKI	AUTH1001	55	60	60	55	Eirini			Antoniu
13	5	13:00-16:00	PLHS13	PATRA	AMTH33	13	15	15	13	Nikolaos			Antoniu
13	5	13:00-16:00	PLHS13	IOANNINA	PLH2	12	20	20	12	Paulos			Keramidas
19	7	13:00-16:00	PLHS24	ATHENS	B2200	35	40	40	35	Konstantinos			krasas
19	7	13:00-16:00	PLHS24	THESSALONIKI	UOM13	21	25	25	21	Ioannis			Arxontidis
19	7	13:00-16:00	PLHS24	PATRA	AMTH32	24	35	35	24	Aikaterini			Kokkinou
19	7	13:00-16:00	PLHS24	IOANNINA	PLH2	20	20	20	20	Georgia			Antoniu
25	9	13:00-16:00	PLHS12	ATHENS	B2201	20	20	20	20	Anna			Vagena
25	9	13:00-16:00	PLHS12	THESSALONIKI	AUTH1001	55	60	60	55	Xrysoula			Dimopoulou
25	9	13:00-16:00	PLHS12	PATRA	AMTH33	13	15	15	13	Konstantinos			Kalimeris
25	9	13:00-16:00	PLHS12	IOANNINA	PLH2	12	20	20	12	Georgios			Petrou
34	12	13:00-16:00	PLHS25	ATHENS	B2200	35	40	40	35	Nikolaos			Papadopoulos
34	12	13:00-16:00	PLHS25	THESSALONIKI	UOM13	21	25	25	21	Georgios			Maronidis
34	12	13:00-16:00	PLHS25	PATRA	AMTH32	24	35	35	24	Panagiotis			Petinakis
34	12	13:00-16:00	PLHS25	IOANNINA	PLH2	20	20	20	20	Antonia			Peraki
35	12	17:00-20:00	PLHS10	ATHENS	B2201	20	20	20	20	Kostantinos			Papadopoulos
35	12	17:00-20:00	PLHS10	THESSALONIKI	AUTH1001	55	60	60	55	Ioannis			Melitos
35	12	17:00-20:00	PLHS10	PATRA	AMTH33	13	15	15	13	Aikaterini			Stamatiou
35	12	17:00-20:00	PLHS10	IOANNINA	PLH2	12	20	20	12	Theodoros			Amanatidis
43	15	13:00-16:00	PLHS14	ATHENS	B2201	20	20	20	20	Theodoros			Kalogiannis
43	15	13:00-16:00	PLHS14	THESSALONIKI	AUTH1001	55	60	60	55	Stelios			Orfanos
43	15	13:00-16:00	PLHS14	PATRA	AMTH33	13	15	15	13	Anna			Georgiou
43	15	13:00-16:00	PLHS14	IOANNINA	PLH2	12	20	20	12	Maria			Xristou
43	15	13:00-16:00	PLHS21	ATHENS	B2200	35	40	40	35	Stelios			Melaxrinis
43	15	13:00-16:00	PLHS21	THESSALONIKI	UOM13	21	25	25	21	Anna			Stamatopoulos
43	15	13:00-16:00	PLHS21	PATRA	AMTH32	24	35	35	24	Xrysoula			Giraki
43	15	13:00-16:00	PLHS21	IOANNINA	PLH1	20	25	25	20	Kostantinos			Erogiou

### Συμπεράσματα:

- Σ' ένα πρόβλημα η ικανοποίηση κατά προτεραιότητα πρώτα των ανελαστικών περιορισμών και έπειτα των ελαστικών είναι δυνατόν αντί για βέλτιστη λύση (ολικό ελάχιστο) να μας παγιδεύσει σε τοπικό ελάχιστο και να έχουμε απλώς μια αποδεκτή λύση.
- Κατά τη διάρκεια εκτέλεσης του προγράμματος που υλοποιεί τον γενετικό αλγόριθμο μπορούμε να αλλάξουμε τη βαρύτητα των περιορισμών, έστω και για ένα συγκεκριμένο πλήθος γενεών, ώστε να βελτιωθεί το παραγόμενο αποτέλεσμα, ξεπερνώντας τα τοπικά ελάχιστα-μέγιστα.

## 3.10 Εφαρμογή του Γενετικού Αλγόριθμου σε Μεγάλα Προβλήματα με τη Χρήση Πραγματικών Δεδομένων

Έχοντας υπόψη τις ιδιαιτερότητες του προβλήματος κατάρτισης εξεταστικού προγράμματος, που αποτελεί κύριο θέμα της παρούσας εργασίας αλλά και την φύση του γενετικού αλγορίθμου είναι απαραίτητο να γίνουν εκτεταμένες δοκιμές με μεγάλο όγκο δεδομένων. Αυτό πρέπει να γίνει για να εξετασθούν οι διάφορες παράμετροι του γενετικού αλγορίθμου που παίζουν καθοριστικό ρόλο στην απόδοσή του, σε σχέση με την χρονική διάρκεια εκτέλεσής του από τη μια μεριά αλλά και στην αποτελεσματικότητά του από την άλλη.

Τα δεδομένα του προβλήματος[28] είναι 2916 φοιτητές, οι οποίοι έχουν κάνει 4716 δηλώσεις μαθημάτων, τα μαθήματα είναι 16 και γίνονται σε 8 πόλεις. Οι μοναδικοί συνδυασμοί μάθημα ανά πόλη είναι 75 συνολικά, οι εξεταστές καθηγητές είναι 71, οι αίθουσες εξετάσεων είναι 38 συνολικά και στις 8 πόλεις ενώ οι χρονοθυρίδες είναι 60 κατανεμημένες σε 20 εξεταστικές ημέρες.

Ο υπολογιστής στον οποίο πραγματοποιούνται οι εκτεταμένες δοκιμές είναι ένας φορητός υπολογιστής με επεξεργαστή i5 2410M - 2.30 GHz με 4 GB μνήμη και λειτουργικό σύστημα τα Windows 7 64 bit.

### 3.10.1 Παράλληλη Επεξεργασία σε Πολυπύρηνους Επεξεργαστές

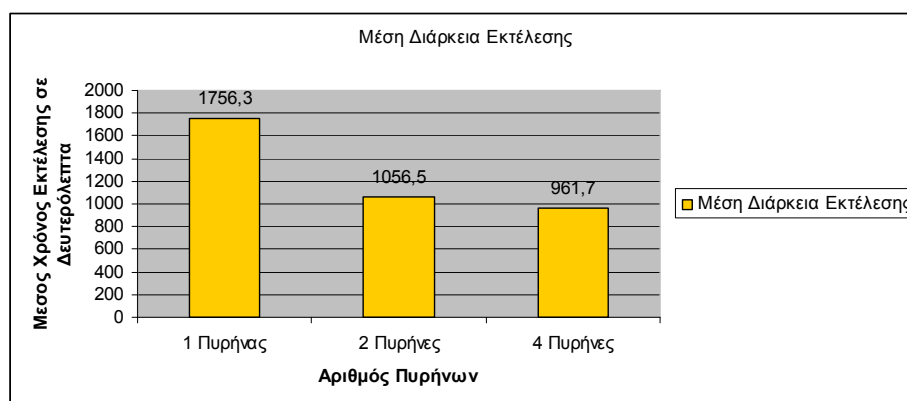
Ένα από τα δυνατά σημεία του ECJ είναι η δυνατότητα που δίνει σ' ένα πρόβλημα να μπορεί να εκτελεσθεί παράλληλα εκμεταλλευόμενο πλήρως την τεχνολογία των πολυπύρηνων επεξεργαστών.

Έτσι, θέλοντας να εκμεταλλευτούμε σε πρώιμο στάδιο τα οφέλη των πολυπύρηνων επεξεργαστών δοκιμάζουμε το πρόβλημα σε μια σειρά εκτελέσεων με τη χρήση<sup>2</sup> ενός, δύο και τεσσάρων πυρήνων. Τα αποτελέσματα παρουσιάζονται στα παρακάτω διαγράμματα.

---

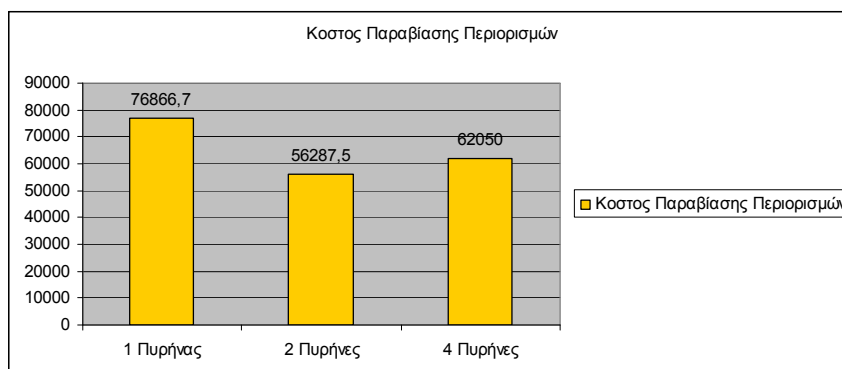
<sup>2</sup> Ενημερώνοντας κατάλληλα το αρχείο παραμέτρων ec.params και συγκεκριμένα τις παραμέτρους evalthreads και breedthreads

Εκτελώντας το λογισμικό, που δημιουργήθηκε στα πλαίσια αυτής της εργασίας, για την αυτόματη δημιουργία ενός εξεταστικού προγράμματος, που υλοποιεί τον γενετικό αλγόριθμο για 500 γενεές, υπολογίσθηκαν τα εξής: σε διαμόρφωση ενός πυρήνα ο μέσος χρόνος ήταν 1756,3 δευτερόλεπτα. Στη διαμόρφωση δύο πυρήνων ο μέσος χρόνος εκτέλεσης ήταν 1056,5 δευτερόλεπτα δηλαδή 40% περίπου καλύτερος από τον χρόνο του μονού πυρήνα. Στη διαμόρφωση τεσσάρων πυρήνων ο μέσος χρόνος που υπολογίσθηκε ήταν 961,7 δευτερόλεπτα δηλαδή περίπου 45,3% καλύτερος από την περίπτωση ενός πυρήνα και 9% περίπου καλύτερος από τον διπλό πυρήνα.



Εικόνα 3.6 Διάγραμμα Μέσου Χρόνου Εκτέλεσης 500 Γενεών

Άλλη κρίσιμη παράμετρος που εξετάστηκε ήταν η αποτελεσματικότητα δηλαδή πόσοι λιγότεροι περιορισμοί παραβιάζονται. Οι διαφορές που προέκυψαν ήταν σχετικά μικρές χωρίς να μπορεί να αποδειχθεί κάτι με ασφάλεια. Συγκεκριμένα την καλύτερη επίδοση είχε η διαμόρφωση των δύο πυρήνων με κόστος 56287,5 ακολούθησε η διαμόρφωση με τέσσερις πυρήνες με 62500 κόστος και τέλος η διαμόρφωση του ενός πυρήνα με 76866,7 κόστος.



Εικόνα 3.7 Διάγραμμα με το κόστος Παραβίασης Περιορισμών

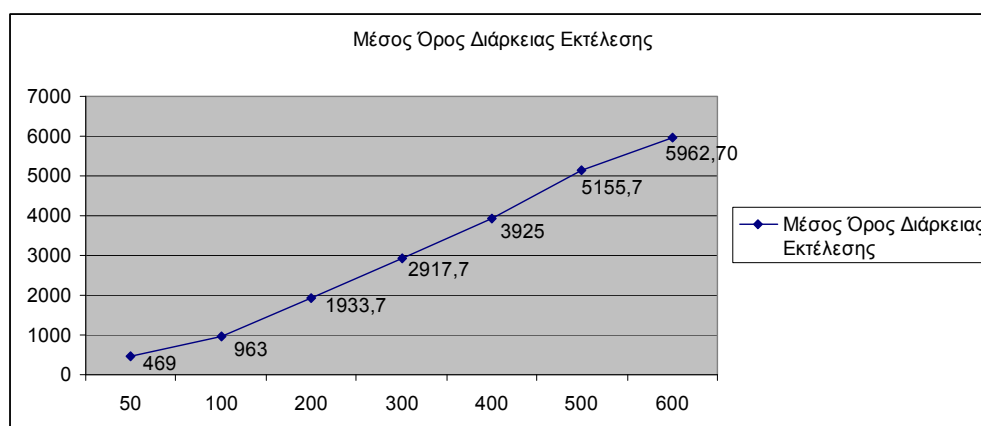
Λόγω της ταχύτερης εκτέλεσης του προγράμματος με τη διαμόρφωση των τεσσάρων πυρήνων, η διαμόρφωση αυτή θα παραμείνει και στις υπόλοιπες δοκιμές που θα εκτελεστούν.

### 3.10.2 Επιλογή Μεγέθους Πληθυσμού

Το μέγεθος του πληθυσμού παίζει σημαντικό ρόλο στην εξέλιξη του γενετικού αλγορίθμου επιδρώντας καθοριστικά από τη μια στην ελάττωση της ταχύτητας της διαδικασίας της εξέλιξης από γενιά σε γενιά του αλγορίθμου αλλά από την άλλη αυξάνεται η πιθανότητα να εντοπισθούν χρωμοσώματα με βελτιωμένη τιμή στη συνάρτηση καταλληλότητας, οδηγώντας γρηγορότερα σε βέλτιστη λύση.

Στην ενότητα αυτή θα εκτελεστούν μια σειρά δοκιμών, με σκοπό να εντοπισθεί το μέγεθος του πληθυσμού με την μικρότερη επιβάρυνση στη ταχύτητα εξέλιξης αλλά με τη μεγαλύτερη αποδοτικότητα. Θα δοκιμαστούν τα μεγέθη 50, 100, 200, 300, 400, 500 και 600 μελών για 500 γενεές.

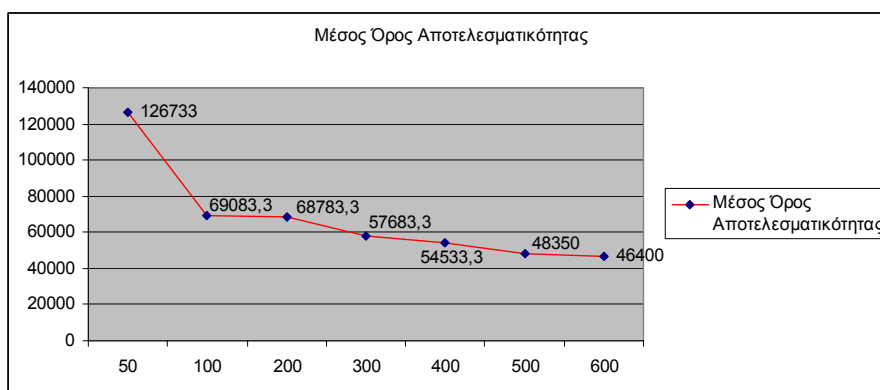
Όπως φαίνεται και στην εικόνα 3.8 παρακάτω, ο μέσος χρόνος διάρκειας εκτέλεσης του προγράμματος, για 500 γενεές, όπως ήταν αναμενόμενο αυξάνεται προοδευτικά. Πιο συγκεκριμένα όταν ο πληθυσμός ήταν 50 μέλη ο μέσος χρόνος ήταν 469 δευτερόλεπτα, για 100 μέλη ο μέσος χρόνος ήταν 963 δευτερόλεπτα, για 200 μέλη 1933,7 δευτερόλεπτα, για 300 μέλη ο μέσος χρόνος ήταν 2917,7 δευτερόλεπτα, για 400 μέλη 3925 δευτερόλεπτα, για 500 μέλη 5155,7 δευτερόλεπτα και για 600 μέλη ο μέσος χρόνος ήταν 5962,7 δευτερόλεπτα.



Εικόνα 3.8 Μέσος Όρος Διάρκειας Εκτέλεσης σε 500 Γενεές



Ο μέσος όρος της αποτελεσματικότητας όπως φαίνεται και στην εικόνα 3.9 είναι ανάλογος με το μέσο χρόνο εκτέλεσης και για πληθυσμούς με μικρό χρόνο εκτέλεσης είχαμε μεγάλο κόστος ενώ για μεγαλύτερους πληθυσμούς με μεγαλύτερο μέσο χρόνο εκτέλεσης αντιστοιχεί και μικρότερο κόστος. Τα αποτελέσματα αυτά θεωρούνται φυσιολογικά διότι οι μεγαλύτεροι πληθυσμοί παρόλο που απαιτείται για την επεξεργασία τους μεγαλύτερο χρονικό διάστημα, δίνουν καλύτερα αποτελέσματα με μικρότερο κόστος στις ίδιες γενεές σε σχέση με τους μικρότερους πληθυσμούς.



Εικόνα 3.9 Μέσος Όρος Αποτελεσματικότητας σε 500 Γενεές

Όπως φαίνεται από τις μετρήσεις αλλά και από τα διαγράμματα όπως αυτά παρουσιάζονται στις εικόνες 3.8 και 3.9 δεν μπορεί να υποστηριχθεί ότι υπάρχει ένα συγκεκριμένο μέγεθος πληθυσμού που δίνει ευδιάκριτα καλά αποτελέσματα και ξεχωρίζει αισθητά από τα υπόλοιπα μεγέθη πληθυσμού.

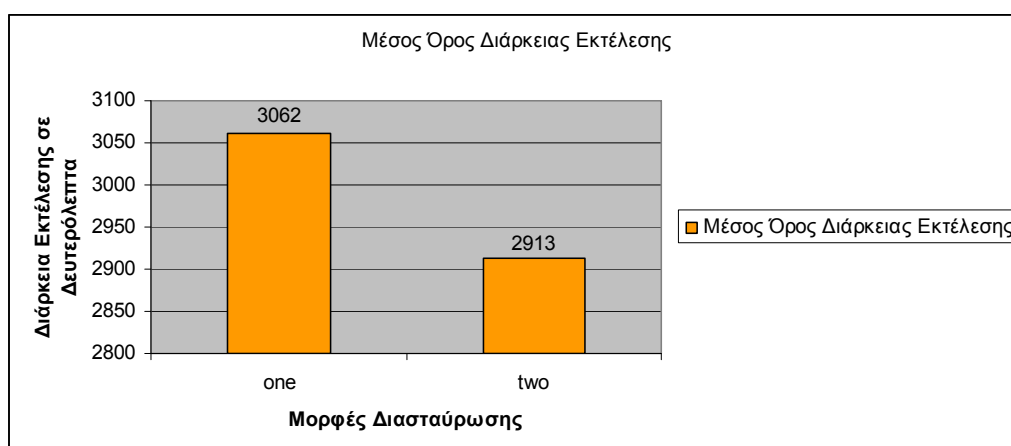
Στις επόμενες δοκιμές επιλέγεται να χρησιμοποιηθεί ο πληθυσμός με μέγεθος 300 απλώς και μόνο κάνοντας την υπόθεση ότι βρίσκεται κάπου στη μέση όσο αφορά την ταχύτητα αλλά και την απόδοση σε σχέση με τους υπόλοιπους πληθυσμούς που δοκιμάστηκαν.

### 3.10.3 Επιλογή Μορφής Διασταύρωσης

Κατά τη διαδικασία της διασταύρωσης γίνεται η ανταλλαγή του γενετικού υλικού ανάμεσα στα χρωμοσώματα που έχουν επιλεγεί να αναπαραχθούν. Ο τρόπος με τον οποίο γίνεται η ανταλλαγή του γενετικού υλικού δεν είναι καθορισμένος και διαφέρει από υλοποίηση σε υλοποίηση. Στην παρούσα ενότητα θα δοκιμασθούν οι δύο πιο παλιές και κοινές μορφές

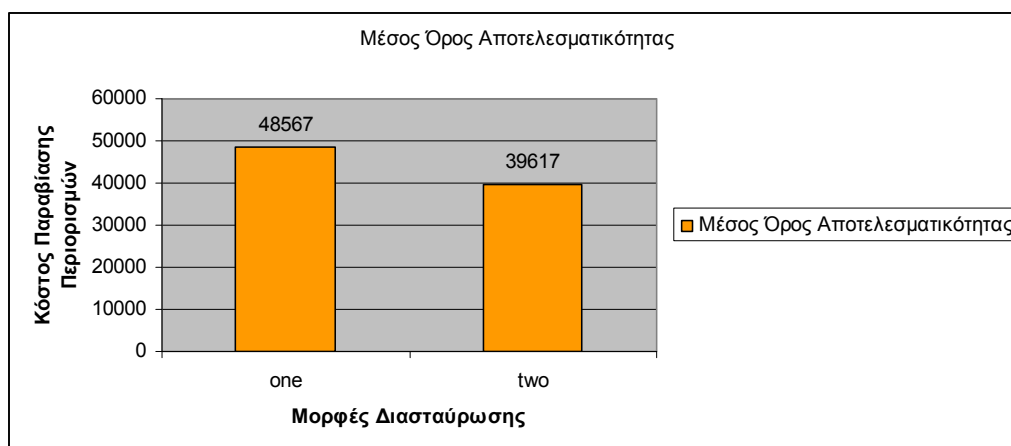
ανταλλαγής γενετικού υλικού με σκοπό να διακριβωθεί εάν επηρεάζουν την αποτελεσματικότητα ή την χρονική διάρκεια εκτέλεσης του προγράμματός μας σχετικά με τη δημιουργία του εξεταστικού ημερολογίου.

Για 500 γενεές, με πληθυσμό 300 ατόμων δοκιμάστηκαν οι μορφές διασταύρωσης ενός σημείου και δύο σημείων και προέκυψαν κάποια αξιολογικά αποτελέσματα. Ο μέσος όρος διάρκειας εκτέλεσης της διασταύρωσης δύο σημείων ήταν 2913 δευτερόλεπτα και ήταν 5% μικρότερος σε σχέση με την διασταύρωση ενός σημείου, που ήταν 3062 δευτερόλεπτα.



Εικόνα 3.10 Μέσος Όρος Διάρκειας Εκτέλεσης Διασταύρωσης Ενός και Δύο Σημείων σε 500 Γενεές

Η αποδοτικότητα της διασταύρωσης δύο σημείων με κόστος παραβίασης περιορισμών 39617 είναι 22,5% περίπου καλύτερη από τη διασταύρωση ενός σημείου που έχει κόστος 48567, όπως φαίνεται και στο διάγραμμα της εικόνας 3.10.



Εικόνα 3.11 Μέσος Όρος Αποτελεσματικότητας Διασταύρωσης Ενός και Δύο Σημείων σε 500 Γενεές

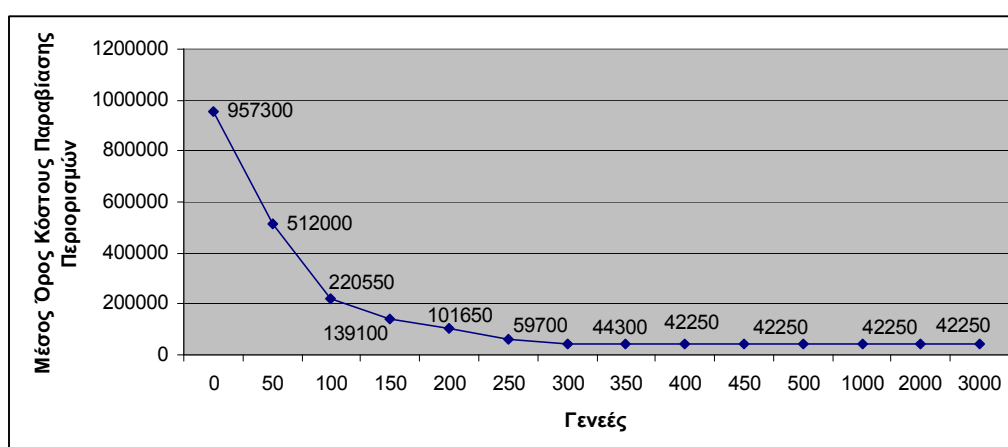
Από τις παραπάνω μετρήσεις αποδεικνύεται ότι για το πρόβλημα της δημιουργίας εξεταστικού προγράμματος με το οποίο ασχολείται η παρούσα εργασία η διασταύρωση δύο σημείων οδηγεί σε καλύτερα αποτελέσματα.

### 3.10.4 Υιοθέτηση των Προηγούμενων Συμπερασμάτων και Εκτέλεση του Αλγορίθμου για Περισσότερες Γενεές

Έχοντας ως οδηγό τα συμπεράσματα όπως αυτά διατυπώθηκαν από τα πειράματα που προηγήθηκαν, εκτελέστηκε σειρά δοκιμών για περισσότερες γενεές με σκοπό να διερευνηθεί εάν η μεγαλύτερη διάρκεια εκτέλεσης του αλγορίθμου βελτιώνει τα παραχθέντα αποτελέσματα και δίνει καλύτερες λύσεις στο πρόβλημα της κατάρτισης ενός εξεταστικού προγράμματος.

Οι ρυθμίσεις που χρησιμοποιήθηκαν είναι μέγεθος πληθυσμού 300 άτομα, μορφή διασταύρωσης δύο σημείων, χρήση και των 4 πυρήνων του υπολογιστικού συστήματος πιθανότητα μετάλλαξης γονιδίου 3% και για 3000 γενεές είχαμε τα εξής αποτελέσματα:

Με χρόνο εκτέλεσης 5 ώρες 1 λεπτό και 12 δευτερόλεπτα ή 18072 συνολικά δευτερόλεπτα δηλαδή η εκτέλεση διήρκησε για κάθε γενεά 6,02 δευτερόλεπτα και εντοπίστηκε αποδεκτή λύση όχι βέλτιστη με κόστος 42250. Όπως φαίνεται και στο διάγραμμα της παρακάτω εικόνας, η καλύτερη λύση βρέθηκε στο διάστημα από την 350<sup>η</sup> μέχρι την 400<sup>η</sup> γενεά και μετά για 2600 γενεές δεν υπήρξε καμιά βελτίωση.



Εικόνα 3.12 Διάγραμμα λύσεων με αντίστοιχο κόστος για 3000 γενεές

Από τα παραπάνω αποτελέσματα γίνεται φανερό ότι ο αλγόριθμος παγιδεύτηκε σε τοπικό ελάχιστο. Η λύση για να ξεπεραστεί αυτό είναι η υποβολή κατ' επανάληψη του προγράμματος για εκτέλεση με σκοπό να ξεπεραστεί τυχαία το τοπικό ελάχιστο ή η εκτέλεση του προγράμματος για πολύ μεγάλο αριθμό γενεών με σκοπό η μετάλλαξη να συμβεί στα κατάλληλα γονίδια και να οδηγηθούμε σε καλύτερη λύση.

Η τελευταία πρόταση για βελτίωση των αποτελεσμάτων θα δοκιμασθεί στο 5<sup>ο</sup> κεφάλαιο της παρούσας δοκιμής όπου το πρόγραμμα θα προσαρμοσθεί κατάλληλα και θα δοκιμασθεί εκτεταμένα στις υποδομές του HellasGrid. Του οργανισμού δηλαδή που παρέχει υπηρεσίες υπολογιστικού πλέγματος στην χώρα μας.

# Κεφάλαιο 4

## Πλέγμα Υπολογιστικών Συστημάτων–Grid Computing

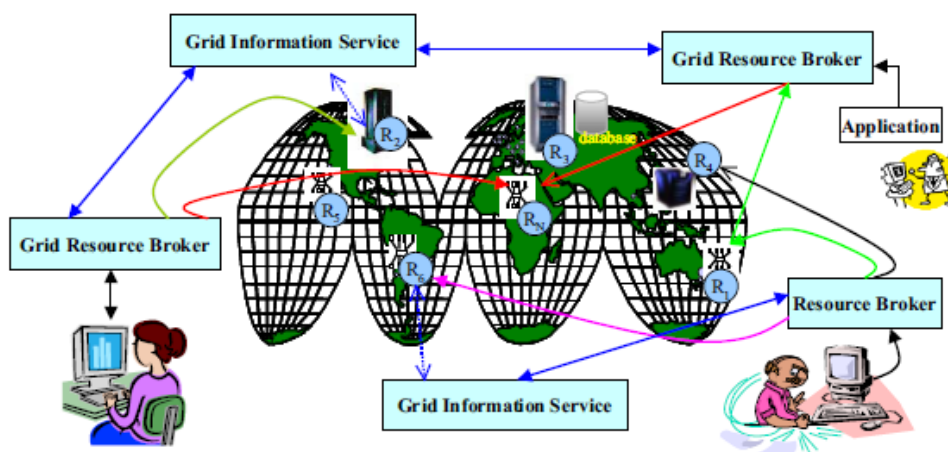
Η μεγάλη τεχνολογική εξέλιξη των υπολογιστικών συστημάτων και η παράλληλη αλματώδης ανάπτυξη των δικτύων επικοινωνιών σε συνδυασμό με την δημιουργία κατάλληλου λογισμικού είχαν σαν αποτέλεσμα την πραγματοποίηση ενός από τα πιο παλιά οράματα της επιστημονικής κοινότητας, αυτό της συνένωσης και της κοινής χρήσης των πόρων που είναι απαραίτητοι για την έρευνα και την επιστημονική αναζήτηση.

Είναι πλέον δυνατόν πόροι όπως υποδομές δικτύωσης, υπολογιστικά συστήματα, εξειδικευμένο λογισμικό, αποθηκευτικά μέσα, αρχεία δεδομένων, επιστημονικές μελέτες και επιστημονικά όργανα παρατηρήσεων από όλο τον κόσμο να συνδεθούν μεταξύ τους και να διαμοιράζονται στον επιστημονικό κόσμο ή σε όποιον έχει ανάγκη την πρόσβαση σε αυτούς τους πόρους, με τρόπο αποτελεσματικό και προπάντων οικονομικό.

## 4.1 Εισαγωγή – Τι είναι το Grid

Ο όρος «Grid Computing» - Υπολογιστικό πλέγμα - για πρώτη φορά δημοσιοποιήθηκε στην επιστημονική κοινότητα το 1998 από τους Foster & Kesselman με τη δουλειά τους «The Grid: blueprint for a new computing infrastructure»[11] και αναφερόταν σε κατακεντρωμένα υπολογιστικά συστήματα τα οποία λειτουργούσαν μέσα στα πλαίσια πολλών υπό διαχείριση τομέων.

Το Grid είναι ένα κατακεντρωμένο δίκτυο υπολογιστών που μπορεί να λειτουργήσει ως μια ενιαία πληροφοριακή και υπολογιστική οντότητα. Έχει διακριτές διαφορετικές μορφές, όπως τα υπολογιστικά Grids που έχουν υψηλές επιδόσεις στην εκτέλεση υπολογισμών, τα Grids παροχής υπηρεσιών πρόσβασης που επιτρέπουν την συνεργασία επιστημόνων οι οποίοι βρίσκονται διασκορπισμένοι σε μακρινές τοποθεσίες και σε Grids δεδομένων που χρησιμοποιούνται για την αποθήκευση και διαχείριση δεδομένων[20].



Εικόνα 4.1 Το παγκόσμιο περιβάλλον υπολογιστικών συστημάτων πλέγματος

Ο στόχος των υπολογιστών Grid είναι πρωταρχικά να παρέχουν απρόσκοπτες και υψηλής ποιότητας υπηρεσίες στις διάφορες επιστημονικές κοινότητες. Παρέχοντας αυξημένη στιγμιαία αλλά και συνολική υπολογιστική δυνατότητα στα επιστημονικά έργα και μάλιστα μέσα σε ένα ασφαλές περιβάλλον. Επίσης δίνεται η δυνατότητα διακίνησης, διαχείρισης και επεξεργασίας μεγάλου όγκου πληροφοριών. Όλα αυτά συμβάλουν στη ποιοτική αλλά και την ποσοτική αύξηση των επιστημονικών επιτευγμάτων σε ένα ευρύ φάσμα επιστημών που κυμαίνεται από την βιοπληροφορική και την προσομοίωση κλιματικών συνθηκών έως την νανοτεχνολογία και την παραγωγή νέων υλικών

## 4.2 Το πρόγραμμα EGEE – Enabling Grids for E-science και ο οργανισμός European Grid Infrastructure

Το καταναεμημένο υπολογιστικό πλέγμα σχεδιάσθηκε αρχικά το 1999 με σκοπό να αναλύσει τα πειραματικά δεδομένα που παράγονταν από τον Μεγάλο Επιταχυντή Αδρονίων στο CERN που βρίσκεται στα Γάλλο-Ελβετικά σύνορα. Το Ευρωπαϊκό πρόγραμμα DataGrid ξεκίνησε το 2001 και ηγήθηκε της έρευνας για την ανάπτυξη τεχνολογιών Grid. Τότε ιδρύθηκε η οργανωτική δομή, αναλύθηκαν οι απαιτήσεις του προγράμματος και αναπτύχθηκε το middleware (το ενδιάμεσο λογισμικό που συνδέει το υλικό των πόρων) και παρείχε εκπαίδευση στους χρήστες. Το 2004 αντικαταστάθηκε από πρόγραμμα EGEE που ανέλαβε την περαιτέρω ανάπτυξη του δικτύου.

Το Ευρωπαϊκό πρόγραμμα EGEE[09], είχε ως κύριο σκοπό του την παροχή σε επιστήμονες, του ακαδημαϊκού ή του βιομηχανικού χώρου, της δυνατότητας να έχουν πρόσβαση σε υπολογιστικούς και αποθηκευτικούς πόρους μεγάλης κλίμακας χωρίς τους περιορισμούς της γεωγραφικής τοποθεσίας. Επίσης το EGEE επεδίωκε την εξοικείωση όσο το δυνατόν περισσότερων απλών χρηστών με τις τεχνολογίες του υπολογιστικού πλέγματος.

Το πρόγραμμα EGEE μέχρι τη στιγμή που ήταν ενεργό σύνδεε περίπου 300 υπολογιστικά κέντρα σε παγκόσμια κλίμακα, παρέχοντας υπηρεσίες με 40000 επεξεργαστές και αρκετά Petabytes αποθηκευτικής δυνατότητας. Σε καθημερινή βάση η υποδομή αυτή εξυπηρετούσε αρκετές χιλιάδες χρηστών από 200 εικονικούς οργανισμούς. Το πρόγραμμα EGEE δεν είναι πλέον ενεργό και όλη η καταναεμημένη υπολογιστική υποδομή που κτίσθηκε από τα προγράμματα DataGrid (2002-2004) και EGEE-I, -II και -III (2004-2010) υποστηρίζεται πλέον από τον οργανισμό European Grid Infrastructure – EGI.

Ο οργανισμός EGI[10] από το 2010 παρέχει υπηρεσίες και συμβάλει στη διατήρηση μιας πανευρωπαϊκής υποδομής για την υποστήριξη των ευρωπαϊκών ερευνητικών κοινοτήτων και των διεθνών συνεργατών τους.

## 4.3 Παράγοντες που ευνόησαν την ανάπτυξη του Υπολογιστικού Πλέγματος

Κατά τη διάρκεια της δεκαετίας του 1990 μια σειρά από σημαντικές τεχνολογικές εξελίξεις είχαν ως αποτέλεσμα την αλλαγή του περιβάλλοντος των πληροφοριακών συστημάτων και μέσα σ' αυτό το περιβάλλον οι υπολογιστές πλέγματος ευνοήθηκαν και εξελίχθηκαν.

Οι εξελίξεις αυτές είναι[13]:

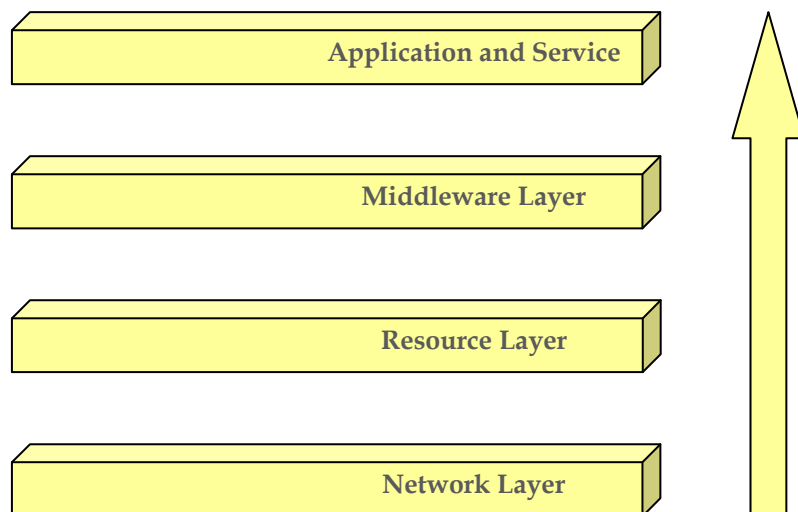
- Το **World Wide Web**, που είναι αποτέλεσμα της δουλειά του Tim Berners Lee, ενός ερευνητή του CERN το 1990. Ο παγκόσμιος ιστός πλέον παίζει καθοριστικό ρόλο σε όλες τις μορφές της ανθρώπινης επικοινωνίας.
- Το λειτουργικό σύστημα **Linux**, που δημιουργήθηκε το 1991 από τον Linus Torvalds από πανεπιστήμιο του Ελσίνκι, είναι το ελεύθερο λειτουργικό σύστημα και χρησιμοποιείται από πολλές υπολογιστικές μηχανές στο πλέγμα.
- Οι **Συστάδες Υπολογιστών (pc clusters)** ήταν αποτέλεσμα της δουλειάς των Donald Becker και Thomas Sterling στη NASA το 1994 και είχε ως αποτέλεσμα φτηνοί υπολογιστές με κάρτες Ethernet να μπορούν να συνδεθούν μεταξύ τους με υψηλές ταχύτητες, παρέχοντας υψηλές αποδόσεις που μόνο μεγάλοι και ακριβοι υπολογιστές είχαν.
- Η γλώσσα προγραμματισμού **java** αναπτύχθηκε το 1995 από μηχανικούς της τότε εταιρείας Sun Microsystems και είναι τελείως ανεξάρτητη από το υπολογιστικό σύστημα στο οποίο εκτελείται.



## 4.4 Η δομή του Grid

Τα βασικά επίπεδα της υποδομής του υπολογιστικού πλέγματος από κάτω προς τα πάνω είναι:

- Το **Επίπεδο Δικτύου (Network Layer)** περιλαμβάνει τις φυσικές διασυνδέσεις και τον εξοπλισμό δικτύωσης όπως είναι η υποδομή καλωδίωσης, οι δρομολογητές, οι γέφυρες, τα Switches κτλ
- Το **Επίπεδο των Πόρων (Resource Layer)** περιλαμβάνει τις υπολογιστικές μηχανές, τους αποθηκευτικούς χώρους, τα επιστημονικά όργανα καθώς και τις υλοποιήσεις τους όπως είναι οι υπερυπολιστές, οι συστάδες υπολογιστών κτλ.
- Το **Επίπεδο του Ενδιάμεσου Λογισμικού (Middleware Layer)** περιλαμβάνει τις υπηρεσίες διαμοιρασμού, πρόσβασης και χρονοπρογραμματισμού των πόρων, υπηρεσίες ασφάλειας, υπηρεσίες μέτρησης του χρόνου πρόσβασης και τήρησης στατιστικών στοιχείων.
- Το **Επίπεδο των Εφαρμογών και των Υπηρεσιών (Application and Serviceware Layer)** περιλαμβάνει τις επιστημονικές εφαρμογές που εκτελούνται στο υπολογιστικό πλέγμα και μπορεί να είναι εφαρμογές προσομοίωσης, μοντέλα πρόβλεψης καιρικών φαινομένων, περίπλοκοι αλγόριθμοι κτλ.



Εικόνα 4.2 Η δομή του Grid

## 4.5 Βασικές αρχές του Υπολογιστικού Πλέγματος

Για την εύρυθμη λειτουργία του υπολογιστικού πλέγματος ισχύουν 5 βασικές αρχές:

**Διαμοιρασμός των Πόρων:** Ο Διαμοιρασμός των πόρων στο υπολογιστικό πλέγμα αποτελεί κεντρική ιδέα και σκοπός είναι όλοι οι χρήστες του πλέγματος να απολαμβάνουν αυτό το πλεονέκτημα και κατά συνέπεια να αυξάνει η αποτελεσματικότητά τους. Ο διαμοιρασμός των πόρων μπορεί να περιλαμβάνει υπολογιστική ισχύ, μηχανικό εξοπλισμό, προγράμματα και αρχεία δεδομένων όπως και ειδικευμένο επιστημονικό εξοπλισμό, όπως τηλεσκόπια μικροσκόπια κτλ.

**Ασφαλής Πρόσβαση:** Η ασφάλεια σε διαμοιραζόμενους πόρους είναι από τις πιο δύσκολες υποθέσεις στο πλέγμα και αφορά τις περιοχές:

1. **Πολιτική Πρόσβασης** δηλαδή ποιος πόρος μοιράζεται, από ποιόν και πότε.
2. **Ταυτοποίηση** πως δηλαδή γίνεται η αναγνώριση των χρηστών και των πόρων.
3. **Εξουσιοδότηση** τι επιτρέπεται να γίνει βάση των κανόνων που ισχύουν.

**Αποτελεσματική χρήση των πόρων:** Το πλέγμα δίνει τη δυνατότητα στους χρήστες να υποβάλουν εργασίες και αυτές να μοιράζονται και να εκτελούνται παράλληλα σε πολλούς υπολογιστές με αποτέλεσμα να εκτελούνται οι εργασίες πολύ γρηγορότερα. Υπάρχουν όμως δυσκολίες που πρέπει να αντιμετωπισθούν, όπως είναι ο μεγάλος όγκος των προς εκτέλεση εργασιών με αποτέλεσμα να δημιουργούνται ουρές αναμονής.

**Εκμηδένιση των Αποστάσεων:** Η χρήση των δημοσίων δικτύων επιτρέπει την πρόσβαση στις υποδομές του υπολογιστικού πλέγματος από μακριά και έτσι δεν υφίσταται πλέον το πρόβλημα της απόστασης.

**Καθιέρωση Κοινών Ανοικτών Προτύπων:** Στο υπολογιστικό πλέγμα υπάρχουν συνδεδεμένοι πολλοί διαφορετικοί πόροι οι οποίοι πρέπει να εργάζονται μαζί χωρίς προβλήματα. Αυτό μπορεί να συμβεί όταν ισχύουν κοινά πρότυπα τα οποία ισχύουν για όλους και για επιτευχθεί η εξέλιξη ομαλά και απροβλημάτιστα θα πρέπει τα πρότυπα να είναι ανοικτά και να έχουν πρόσβαση όσοι ασχολούνται με τη βελτίωσή τους και την εξέλιξή τους.

## 4.6 Ο Οργανισμός HellasGrid

Το HellasGrid[15] είναι η μεγαλύτερη υποδομή υπολογιστικού πλέγματος στην ευρύτερη περιοχή της Νοτιοανατολικής Ευρώπης και ένα από τα πιο γρήγορα αναπτυσσόμενα υπολογιστικά πλέγματα σε πανευρωπαϊκό επίπεδο.

Το βασικό έργο του HellasGrid είναι η δυνατότητα που παρέχει στους χρήστες του για σύνδεσή τους με υπολογιστές υψηλών επιδόσεων και επίσης παροχή υπηρεσιών υψηλής υπολογιστικής απόδοσης στην Ελληνική ακαδημαϊκή και ερευνητική κοινότητα. Οι πόροι από την υποδομή του πλέγματος χρησιμοποιούνται από Έλληνες ερευνητές αλλά και ερευνητές από άλλες χώρες οι οποίοι εμπλέκονται σε διάφορα ευρωπαϊκά προγράμματα. Τα τελευταία χρόνια ένας ολοένα αυξανόμενος αριθμός επιστημόνων που προσέρχονται από διάφορους επιστημονικούς χώρους χρησιμοποιούν τις υποδομές του HellasGrid για να καλύψουν τις ανάγκες τους για υπολογιστική ισχύ και αποθηκευτικούς χώρους.

Η πρόσβαση στο HellasGrid είναι ελεύθερη για την ακαδημαϊκή και ερευνητική κοινότητα αφού αρχικά ακολουθηθεί μια σχετικά απλή διαδικασία εγγραφής που περιγράφεται αναλυτικά στον σύνδεσμο: [https://access.hellasgrid.gr/...](https://access.hellasgrid.gr/)



Εικόνα 4.3 Η υποδομή του HellasGrid

Η σημερινή υποδομή HellasGrid αποτελείται από 12 υπολογιστικούς κόμβους, υπολογιστικών και αποθηκευτικών πόρων και αναλυτικά παρουσιάζονται στον πίνακα 4.1

HellasGrid Infrastructure Overview		
	CPU Cores	Storage (TBs)
HG-01-GRNET	64	20
HG-02-IASA	120	4
HG-03-AUTH	120	6
HG-04-CTI-CEID	120	4
HG-05-FORTH	120	4
HG-06-EKT	628	12
GR-01-AUTH	222	82
GR-04-FORTH-ICS	10	-
GR-06-IASA	20	2
GR-07-UOI-HEPLAB	28	20
GR-09-UOA	10	2
GR-10-UOI	120	2
<b>Total</b>	<b>1582</b>	<b>158</b>

Πίνακας 4.1 Πίνακας με τους υπολογιστικούς κόμβους στο HellasGrid

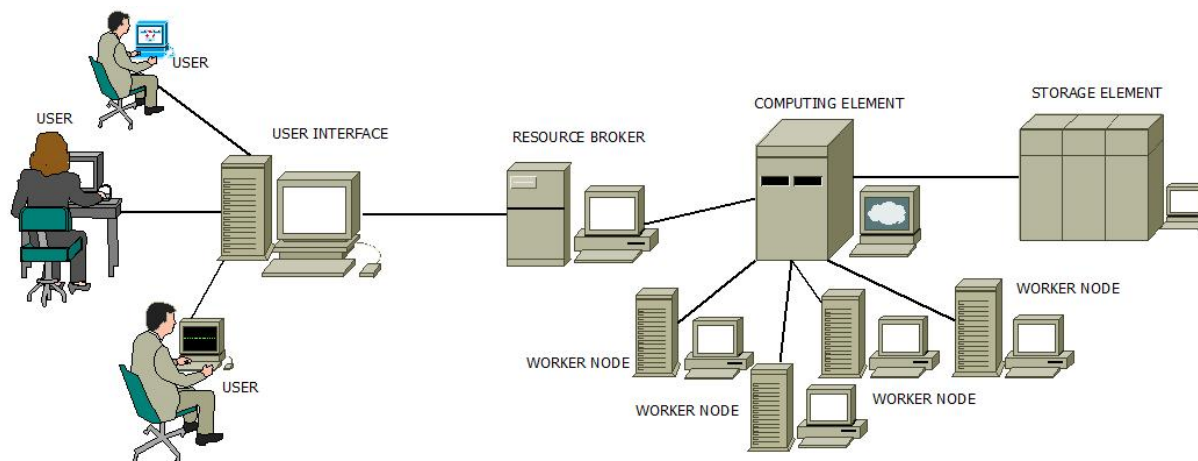
Η Αρχή Πιστοποίησης HellasGrid εκδίδει ψηφιακά πιστοποιητικά για τους νέους χρήστες ή τους χρήστες που είχαν και παλαιότερα εκδώσει ψηφιακά πιστοποιητικά. Ψηφιακά πιστοποιητικά επίσης εκδίδονται και για σύνδεση διακομιστών. Δύο ομάδες υποστήριξης παρέχουν υπηρεσίες υποστήριξης σχετικά με τις διαδικασίες σύνδεσης στις υποδομές του HellasGrid, για τις συνήθεις εργασίες που εκτελούνται αλλά και για την φόρτωση και εκτέλεση εφαρμογών πάνω στο υπολογιστικό πλέγμα.

## 4.7 Εικονικοί Οργανισμοί - Virtual Organizations (VO)

Τα δικαιώματα πρόσβασης στην υποδομή του Grid για ένα χρήστη καθορίζονται από το Εικονικό Οργανισμό - Virtual Organization (VO) στον οποίο ανήκει. Τα Virtual Organizations είναι εικονικές ομάδες χρηστών, οι οποίοι έχουν κοινό τομέα έρευνας ή απλά ανήκουν σε μια ευρύτερη γεωγραφική περιοχή. Χρήστες που ανήκουν σε διαφορετικά VOs έχουν πρόσβαση σε διαφορετικούς πόρους της υποδομής. Όταν κάποιος χρήστης αποκτά πρόσβαση στο HellasGrid, τότε ο εικονικός οργανισμός που θα πρέπει να εγγραφεί είναι το SEE VO (South Eastern European Virtual Organisation), στο οποίο δικαίωμα εγγραφής έχουν οι ερευνητές της Νότιο Ανατολικής Ευρώπης.

## 4.8 Υποβολή εργασιών στο Grid

Για να μπορεί ένας χρήστης να έχει πρόσβαση στις υπηρεσίες που είναι διαθέσιμες στις δομές του HellasGrid θα πρέπει αρχικά να κάνει ένα λογαριασμό ακολουθώντας τη προβλεπόμενη διαδικασία. Στη συνέχεια θα πρέπει να συνδεθεί στο (πλησιέστερο – βάση απόστασης συνήθως) User Interface που του έχει υποδειχθεί και για να μπορεί να εκτελεί εργασίες θα πρέπει να εγκαταστήσει το πιστοποιητικό ασφαλείας.



Εικόνα 4.2 Υποβολή εργασιών στο Grid

Η παραπάνω εικόνα 4.2 παρουσιάζει σε σχηματική μορφή πως θα μπορούσε να είναι ένας κόμβος στο Grid και μια ενδεικτική αναπαράσταση των επιμέρους μερών του. Κατά τη διαδικασία υποβολής εργασιών από τους χρήστες το κάθε στοιχείο του κόμβου όπως παρουσιάζεται παρακάτω παίζει έναν ιδιαίτερο ρόλο.

Το **User Interface** επιτελεί σημαντικές λειτουργίες όπως να επιτρέπει ένα χρήστη να έχει πρόσβαση στο Grid, να υποβάλλονται εργασίες από το χρήστη προς εκτέλεση, να ελέγχονται ποιοι πόροι είναι κατάλληλοι για την εργασία που υποβλήθηκε, να μπορεί να ακυρωθεί μια εργασία η οποία υποβλήθηκε νωρίτερα, να εμφανίζει την κατάσταση μιας εργασίας που έχει υποβληθεί προς εκτέλεση, να γίνεται ανάκτηση των δεδομένων εξόδου και επίσης να γίνεται ανάκτηση της εξόδου από έναν πρόγραμμα.

Το **Resource Broker** δέχεται τις εντολές των χρηστών για εκτέλεση εργασιών και αναζητά τους κατάλληλους πόρους για την εκτέλεση των εργασιών

Το **Computing Element** διαχειρίζεται μια ομάδα ομοειδών κόμβων που ο καθένας ονομάζεται Worker Node, εκτελεί τις εργασίες που έχουν σταλεί και παρακολουθεί την πορεία εκτέλεσης των εργασιών.

Το **Storage Element** παρέχει την δυνατότητα πρόσβασης και χρήσης του αποθηκευτικού χώρου.

# Κεφάλαιο 5

## Εκτέλεση του Λογισμικού σε Περιβάλλον Υπολογιστικού Πλέγματος και Παράλληλης Επεξεργασίας

Είναι γενικά παραδεκτό ότι οι γενετικοί αλγόριθμοι λόγω των μεγάλων απαιτήσεων σε επεξεργαστική ισχύ και μνήμη απαιτούν παράλληλο περιβάλλον εκτέλεσης. Στο 3<sup>ο</sup> κεφάλαιο φάνηκε ότι η εκτέλεση του προγράμματος σε υπολογιστή με τη χρήση δύο και κυρίως τεσσάρων πυρήνων μείωσε τον χρόνο εκτέλεσης σχεδόν στο μισό.

Σ' αυτό το κεφάλαιο θα γίνουν εκτεταμένες δοκιμές σε μεγάλους υπολογιστές που διαθέτουν αρκετούς επεξεργαστές και μεγάλη μνήμη. Οι υπολογιστές αυτοί αποτελούν είτε μέρος της υποδομής του HellasGrid είτε άλλων διασυνδεδεμένων παράλληλων υπολογιστικών συστημάτων στο Grid και που έχουν πρόσβαση τα μέλη του εικονικού οργανισμού SEE (South Eastern Europe).

## 5.1 Παράλληλη Επεξεργασία

Το ECJ ToolKit[17] που χρησιμοποιήθηκε για την αναπαράσταση του γενετικού αλγορίθμου στη παρούσα διατριβή γενικά υποστηρίζει διάφορες μεθόδους παράλληλης επεξεργασίας, οι οποίες μάλιστα θα μπορούσαν να συνδυαστούν και μεταξύ τους. Οι μέθοδοι παράλληλης επεξεργασίας είναι:

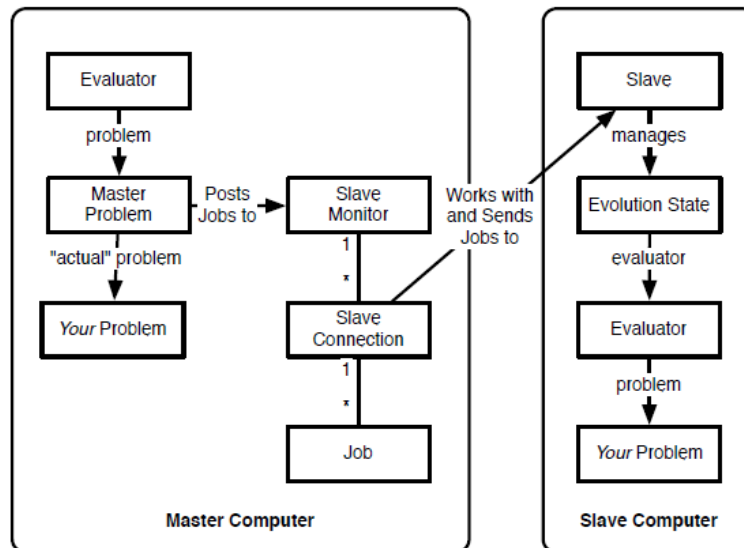
1. **Μέθοδος Κατανεμημένης Αξιολόγησης (Distributed Evaluation).** Σύμφωνα με αυτή τη μέθοδο σε ένα περιβάλλον κατανεμημένης επεξεργασίας στέλνεται ένας αριθμός ατόμων (χρωμοσωμάτων) σε απομακρυσμένους υπολογιστές με σκοπό να αξιολογηθούν. Συνήθως αυτό γίνεται κατά τη διαδικασία εξέλιξης των γενεών αλλά μπορεί να γίνει και με ένα τρόπο ασύγχρονο σύμφωνα με τον οποίο τα άτομα στέλνονται για αξιολόγηση μ' ένα σταθερό τρόπο.
2. **Μέθοδος των Νησιών (Islands Method).** Όπου πολλαπλές παράλληλες εξελικτικές διαδικασίες εκτελούνται και περιστασιακά ανταλλάσσουν μεταξύ τους στοιχεία που αφορούν τα άτομα που επεξεργάζονται.
3. **Πολυνηματική Εκτέλεση των Διαδικασιών της Αναπαραγωγής και της Αξιολόγησης.** Μ' αυτή τη μέθοδο μπορούν να εκτελεστούν παράλληλα οι δύο κυριότερες διαδικασίες του γενετικού αλγορίθμου, της αναπαραγωγής και της αξιολόγησης.

### 5.1.1 Η Μέθοδος της Κατανεμημένης Αξιολόγησης (Distributed Evaluation)

Η κατανεμημένη αξιολόγηση συνδέει μια διαδικασία «Αφέντης»-«Master», διαδικασία που έχει τον κεντρικό έλεγχο δηλαδή, με  $N$  το πλήθος διαδικασίες «Σκλάβους»-«Slaves». Η διαδικασία αφέντης ελέγχει την όλη επαναληπτική εκτέλεση του γενετικού αλγορίθμου και όταν ένα άτομο πρέπει να αξιολογηθεί αυτό γίνεται από κάποια διαδικασία σκλάβου.

Η κατανεμημένη αξιολόγηση είναι χρήσιμη όταν ο κερδισμένος χρόνος από την παράλληλη αξιολόγηση είναι περισσότερος από τον χρόνο που χάνεται από τη μεταφορά των ατόμων-χρωμοσωμάτων από την διεργασία-master στις διεργασίες-slaves για αξιολόγηση και την αντίστροφη μεταφορά των αποτελεσμάτων από τις διεργασίες-slaves στη διεργασία-master.





Εικόνα 5.1 Κατανεμημένη Αξιολόγηση

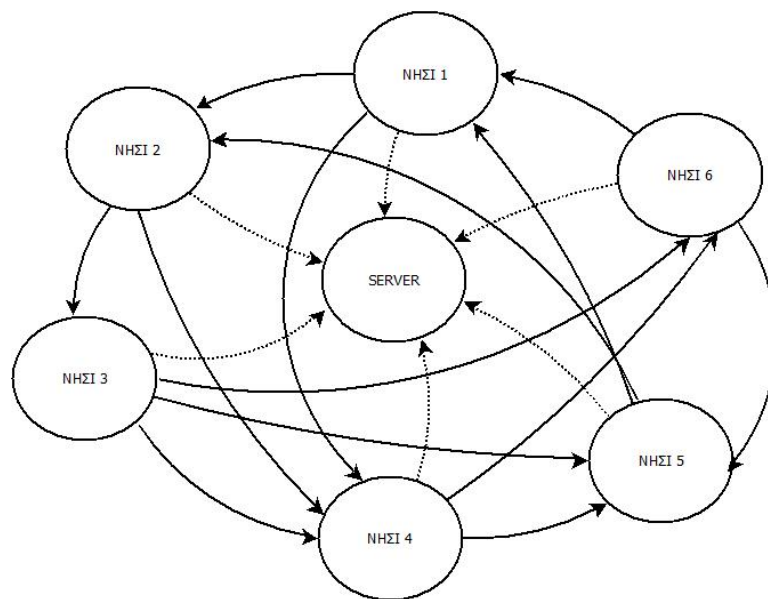
Κατά τη διάρκεια εκτέλεσης του γενετικού αλγορίθμου με τη μέθοδο της κατανεμημένης αξιολόγησης, αρχικά οι υπολογιστές-slaves που είναι διαθέσιμοι μέσω της ip-διεύθυνσης του υπολογιστή-master, συνδέονται μαζί του και ακούν μια συγκεκριμένη πόρτα που έχει ορισθεί από πριν για αυτό το σκοπό. Ο υπολογιστής-master έχει τη λίστα με τις απευθείας συνδέσεις των υπολογιστών-slaves που ονομάζεται «Slave Monitor». Όταν ο υπολογιστής-master έχει άτομα-χρωμοσώματα έτοιμα προς αξιολόγηση τα στέλνει σε κάποιο υπολογιστή-slave που είναι διαθέσιμος μέσω του «Slave Monitor». Ο υπολογιστής-slave αφού ολοκληρώσει την αξιολόγηση στέλνει πίσω το αποτέλεσμα ή αν έχει ορισθεί, μπορεί να εκτελέσει την εξελικτική διαδικασία, να γίνει αναπαραγωγή και να σταλούν στον υπολογιστή-master τα νέα άτομα-χρωμοσώματα που προέκυψαν.

### 5.1.2 Η Μέθοδος των Νησιών (Islands Method)

Στη μέθοδο των νησιών, ξεχωριστές διαδικασίες (οι οποίες αποκαλούνται νησιά), οι οποίες είναι διασυνδεδεμένες στα πλαίσια ενός δικτύου, εκτελούν τον γενετικό αλγόριθμο ενώ μπορούν να ανταλλάξουν μεταξύ τους κατάλληλα μηνύματα που περιέχουν άτομα-χρωμοσώματα ευνοώντας έτσι την εξέλιξη του αλγορίθμου. Μια ειδική μονάδα που ονομάζεται εναλλάκτης (Exchanger) αναλαμβάνει τη μεταφορά των ατόμων-χρωμοσωμάτων από νησί σε νησί πριν και αμέσως μετά την αναπαραγωγή.

Τα περισσότερα σημαντικά θέματα στη μέθοδο των νησιών έχουν να κάνουν με την ιδιαίτερη τοπολογία των νησιών που θα επιλεγεί. Τέτοια θέματα θα μπορούσε να είναι: ποιο νησί θα στέλνει άτομα-χρωμοσώματα και ποιο νησί θα είναι παραλήπτης, πόσα άτομα-χρωμοσώματα θα στέλνονται τη φορά, πόσο συχνά θα στέλνονται, θα στέλνονται σύγχρονα ή ασύγχρονα; Το ECJ αντιμετωπίζει τα θέματα της τοπολογίας των νησιών αλλά και τις συνδέσεις στο δίκτυο, μέσω μιας ειδικής διαδικασίας που ονομάζεται server της νησιωτικής μεθόδου

Ο κάθε κόμβος νησί συνδέεται με τον server και καταχωρεί το όνομά του και την socket port στην οποία θα ακούει. Ο server, όταν καταχωρηθούν όλοι οι κόμβοι νησιά, αποφασίζει ποια νησιά θα επικοινωνούν μεταξύ τους και πως και όταν γίνει αυτό ξεκινά να εκτελείται ο γενετικός αλγόριθμος. Αν έχει επιλεγεί η σύγχρονη επικοινωνία μεταξύ των νησιών ο server δίνει εντολή κάθε φορά να ξεκινά η επόμενη γενεά στην εξέλιξη του γενετικού αλγορίθμου εφόσον έχει τελειώσει η εκτέλεση για την προηγούμενη γενεά σε όλους τους κόμβους νησιά. Όταν σε ένα νησί βρεθεί η βέλτιστη λύση τότε αυτό ανακοινώνεται σε όλα τα υπόλοιπα νησιά και τερματίζουν την εκτέλεσή τους.



Εικόνα 5.2 Παράδειγμα Μοντέλου Νησιού

Η παραπάνω εικόνα 5.2 παρουσιάζει ένα παράδειγμα του μοντέλου νησιού, όπου υπάρχουν 6 νησιά και ένας server. Όλα τα νησιά έχουν συνδεθεί με τον server και έχουν δηλώσει όνομα και socket port στην οποία θα ακούν τα υπόλοιπα νησιά και τον server. Ο server καθορίζει ποια νησιά θα επικοινωνούν μεταξύ τους.

### 5.1.3 Πολυνηματική Εκτέλεση των Διαδικασιών της Αναπαραγωγής και της Αξιολόγησης.

Το ECJ υποστηρίζει πολυνηματική επεξεργασία στα δύο σημαντικότερα στάδια του γενετικού αλγορίθμου, της αναπαραγωγής και της αξιολόγησης. Ανάλογα με το υπολογιστικό σύστημα στο οποίο γίνεται η εκτέλεση του αλγορίθμου μπορεί να ορισθεί το πλήθος των νημάτων για τις διαδικασίες της αναπαραγωγής και της αξιολόγησης ενημερώνοντας τις κατάλληλες παραμέτρους `-breedthreads` και `evalthreads` - στο αρχείο παραμέτρων `ec.params`. Για παράδειγμα αν θέλαμε να έχουμε να εκτελέσουμε τις δύο αυτές διαδικασίες παράλληλα με δύο νήματα την καθεμιά θα ορίζαμε:

`breedthreads = 2` και `evalthreads = 2`

Το πλήθος των νημάτων που μπορεί να χρησιμοποιηθεί εξαρτάται από το πλήθος των πυρήνων ή των επεξεργαστών που διαθέτει το υπολογιστικό σύστημα στο οποίο θα εκτελεσθεί ο αλγόριθμος. Εάν θέλουμε να έχουμε ακριβώς τόσα νήματα όσα και οι επεξεργαστές τότε στις μεταβλητές `breedthreads` και `evalthreads` δίνουμε την τιμή `auto`.



Εικόνα 5.3 Η παράλληλη επεξεργασία της αξιολόγησης και της αναπαραγωγής από δύο πυρήνες σε σχέση με τη σειριακή επεξεργασία του ενός πυρήνα

Επειδή ο γενετικός αλγόριθμος είναι μια στοχαστική διαδικασία, δηλαδή σε κάποια στάδια εξέλιξης του αλγορίθμου εμπεριέχεται η τύχη, είναι πολύ σημαντικό όταν χρησιμοποιείται το ECJ για παράλληλη επεξεργασία να ληφθεί ιδιαίτερη μέριμνα για την παραγωγή τυχαίων αριθμών.

Το ECJ χρησιμοποιεί μια ιδιαίτερη ρουτίνα για την παραγωγή τυχαίων αριθμών. Η ρουτίνα αυτή χρησιμοποιεί αριθμούς-σπόρους, ακεραίους των 32-bit, με βάση αυτούς τους σπόρους δημιουργούνται από τη γεννήτρια σειρές τυχαίων αριθμών.

Αν θέλουμε να έχουμε πραγματικά παράλληλη επεξεργασία πρέπει να χρησιμοποιήσουμε για κάθε πυρήνα – επεξεργαστή έναν διαφορετικό αριθμό σπόρο. Αν δώσουμε τον ίδιο ο κάθε πυρήνας – επεξεργαστής θα παράγει τους ίδιους αριθμούς και έτσι θα έχουμε μια σειριακή επεξεργασία η οποία θα εκτελείται επαναληπτικά. Στο αρχείο παραμέτρων `ec.params` μπορεί να δηλωθεί ο αριθμός σπόρος να παίρνεται αυτόματα από το ρολόι του συστήματος και να είναι κάθε φορά διαφορετικός. Αυτό γίνεται αν στη παράμετρο `seed` δώσουμε τιμή `time` δηλαδή `seed.0 = time`. Αν δεν δοθεί αριθμός σπόρος το ECJ δίνει αυτόματα για κάθε πυρήνα – επεξεργαστή τιμές από το ρολόι.

## 5.2 Επιλογή Παράλληλης Μεθόδου

Για την υλοποίηση και την μετατροπή του προγράμματος, που δημιουργεί αυτόματα εξεταστικό πρόγραμμα, ώστε να μπορεί να εκτελείται σε περιβάλλον παράλληλης επεξεργασίας, από τις τρεις μεθόδους υλοποίησης όπως αυτές παρουσιάσθηκαν στην ενότητα 5.1, που θα μπορούσαν να επιλεγούν, αποφασίσθηκε το πρόγραμμα να υλοποιηθεί με την Τρίτη μέθοδο. Επιλέχθηκε δηλαδή η μέθοδος της **Πολυνηματικής Εκτέλεσης των Διαδικασιών της Αναπαραγωγής και της Αξιολόγησης**. Η επιλογή αυτή βασίστηκε στους παρακάτω λόγους:

- Είναι η πιο απλή μέθοδος που εύκολα μπορεί να υλοποιηθεί και το κυριότερο μπορεί να δοκιμασθεί και να εκτελεσθεί σε υπολογιστές που διαθέτουν πολλούς επεξεργαστές ή πυρήνες. Έτσι το πρόγραμμα γίνεται διαθέσιμο σε υπολογιστικά συστήματα του τύπου προσωπικού υπολογιστή έως σε συστήματα που ανήκουν στις υποδομές του Υπολογιστικού Πλέγματος.
- Είναι γρήγορη μέθοδος και προσφέρει καλά αποτελέσματα με μεγάλη οικονομία χρόνου, αφού οι διαδικασίες της αξιολόγησης και της αναπαραγωγής εκτελούνται παράλληλα.

- Το πρόγραμμα που δημιουργήθηκε θα μπορούσε σε πραγματικό χρόνο να δημιουργεί αυτόματα ένα πρόγραμμα εξετάσεων καθώς δεν εξαρτάται μόνο από τις υποδομές του υπολογιστικού πλέγματος.
- Υπάρχει η δυνατότητα της εκμετάλλευσης των τεχνολογικών εξελίξεων χωρίς καμιά αλλαγή στο πρόγραμμα αφού μελλοντικά είναι πιθανόν να εμφανισθούν προσωπικοί υπολογιστές ή μεγαλύτεροι υπολογιστές με πολλούς επεξεργαστές ή με επεξεργαστές που θα διαθέτουν ακόμη περισσότερους πυρήνες.

### 5.3 Παράλληλη Εκτέλεση του Προγράμματος σε Συγκεκριμένο Server (ui.afroditi.hellasgrid.gr)

Αφού πραγματοποιήθηκε η σύνδεση στις υποδομές του HellasGrid και συγκεκριμένα στο User Interface με όνομα ui.afroditi.hellasgrid.gr το οποίο βρίσκεται στο Επιστημονικό Κέντρο Υπολογιστών (Scientific Computing Center) του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης.

Hostname	ui.afroditi.hellasgrid.gr
Operating System (version)	Scientific Linux (5.4)
Architecture	x86_64
Middleware (version)	gLite (3.2)
CPU Model	Intel(R) Xeon(TM) CPU Core 2 Duo E5405
CPU Cores	8 (Click <a href="#">here</a> for an overview)
Clock frequency [GHz]	2.0
RAM (GB)	16

Πίνακας 5.1 Τεχνικά χαρακτηριστικά του Server που έχει το ρόλο του User Interface στο ui.afroditi.hellasgrid.gr

Αποφασίσθηκε να γίνουν μια σειρά από εκτεταμένες δοκιμές στο υπολογιστικό σύστημα, τα τεχνικά χαρακτηριστικά του οποίου παρουσιάζονται στον πίνακα 5.1, ώστε να συγκριθούν τα αποτελέσματα που προέκυψαν από τις δοκιμές όπως παρουσιάστηκαν στο 3<sup>ο</sup> κεφάλαιο της παρούσας διατριβής κατά πρώτο λόγο και κατά δεύτερο να αποφασισθεί η πιθανότητα

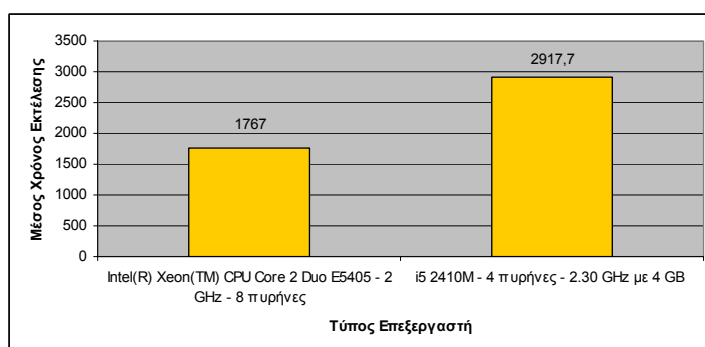
μετάλλαξης που θα δίνει τις καλύτερες λύσεις ώστε να χρησιμοποιηθεί αργότερα στις δοκιμές που θα γίνουν στο υπολογιστικό πλέγμα.

Τα δεδομένα του προβλήματος που θα χρησιμοποιηθούν στις δοκιμές που θα ακολουθήσουν είναι τα ίδια ακριβώς με αυτά που χρησιμοποιήθηκαν στο 3<sup>ο</sup> κεφάλαιο για να έχουμε ένα λόγο αναφοράς. Αναλυτικά είναι 2916 φοιτητές, οι οποίοι έχουν κάνει 4716 δηλώσεις μαθημάτων, τα μαθήματα είναι 16 και γίνονται σε 8 πόλεις. Οι συνδυασμοί εξεταζόμενο μάθημα ανά πόλη είναι 75 συνολικά, οι εξεταστές καθηγητές είναι 71, οι διαθέσιμες αίθουσες εξετάσεων είναι 38 συνολικά και στις 8 πόλεις ενώ οι χρονοθυρίδες για τις εξετάσεις είναι 60 κατανομημένες σε 20 ημέρες.

### 5.3.1 Σύγκριση Αποτελεσμάτων

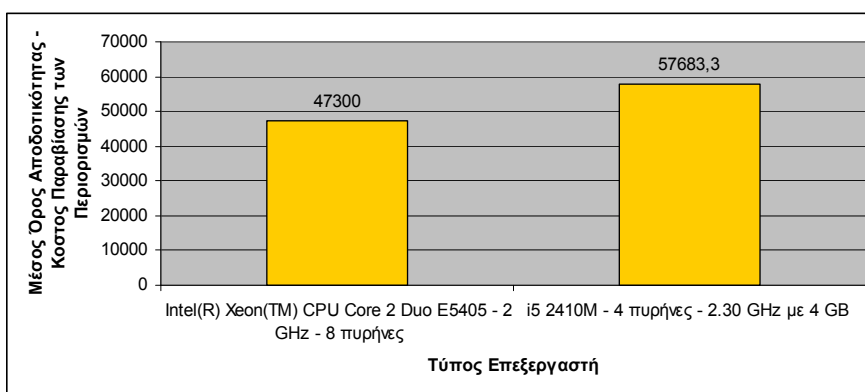
Εκτελώντας το λογισμικό, για την αυτόματη δημιουργία ενός εξεταστικού προγράμματος, που υλοποιεί τον γενετικό αλγόριθμο για 500 γενεές και πληθυσμό 300 ατόμων στον server με όνομα ui.afroditi.hellasgrid.gr και επεξεργαστή Intel(R) Xeon(TM) CPU Core 2 Duo E5405 με ταχύτητα 2 GHz και 8 πυρήνες και συνολική μνήμη 16 GB, ο μέσος χρόνος εκτέλεσης ήταν 29 λεπτά και 27 δευτερόλεπτα ή 1767 συνολικά δευτερόλεπτα. Ενώ σε φορητό υπολογιστή με επεξεργαστή i5 2410M με 4 πυρήνες, ταχύτητα 2.30 GHz με 4 GB μνήμη το ίδιο πρόγραμμα έκανε 48 λεπτά και 37,7 δευτερόλεπτα ή 2917,7 συνολικά δευτερόλεπτα. Δηλαδή στην πρώτη περίπτωση το πρόγραμμα έτρεξε 39,5% πιο γρήγορα.

Τα αποτελέσματα παρουσιάζονται στο γράφημα της εικόνας 5.4 παρακάτω.



Εικόνα 5.4 Μέσος Χρόνος Εκτέλεσης στους δύο υπολογιστές

Όσον αφορά την μέση αποδοτικότητα δηλαδή το μέσο κόστος παραβίασης των περιορισμών όπως αυτοί έχουν διαμορφωθεί στον καθορισμό του προβλήματος, για τον server, όπως φαίνεται στην παρακάτω εικόνα, το μέσο κόστος ήταν 47300 ενώ για τον φορητό υπολογιστή το μέσο κόστος ήταν 57683,3. Δηλαδή στον server το μέσο κόστος παραβίασης των περιορισμών ήταν 18% μικρότερο. Πιθανόν αυτή η διαφορά να οφείλεται στο γεγονός ότι όταν χρησιμοποιούνται περισσότεροι επεξεργαστές ή πυρήνες, παράγονται περισσότερες σειρές τυχαίων αριθμών για τα χρωμοσώματα που χρησιμοποιούνται με αποτέλεσμα να αυξάνουν οι πιθανότητες για τον εντοπισμό καλύτερης τελικής λύσης.



Εικόνα 5.5 Μέσος Όρος Αποδοτικότητας στους δύο επεξεργαστές

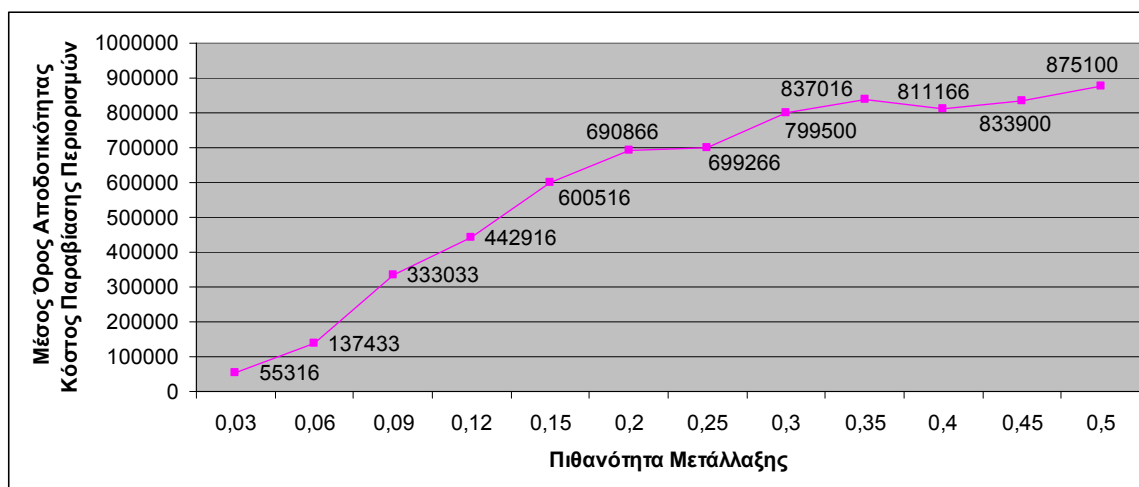
Στην ενότητα 3.10.1 επίσης είχε βρεθεί ότι με τη χρήση δύο ή τεσσάρων πυρήνων είχαμε καλύτερη μέση αποδοτικότητα σε σχέση με τον ένα πυρήνα.

### 5.3.2 Επιλογή Πιθανότητας Μετάλλαξης

Στον server με όνομα `ui.afroditi.hellasgrid.gr` έγιναν εκτεταμένες δοκιμές για 3000 γενεές δοκιμάζοντας πιθανότητες που μπορούν να χρησιμοποιηθούν στο στάδιο της μετάλλαξης. Στο στάδιο της μετάλλαξης για κάθε γονίδιο υπάρχει μια πιθανότητα να αλλάξει τιμή και να πάρει μια άλλη τυχαία. Η μετάλλαξη συμβάλει κυρίως στο να ξεπερνιούνται τοπικά ελάχιστα-μέγιστα και να εντοπίζονται καλύτερες λύσεις.

Τα αποτελέσματα από τις δοκιμές παρουσιάζονται παρακάτω στο διάγραμμα της εικόνας 5.6. Οι πιθανότητες που δοκιμάστηκαν ήταν από 3% έως 50% και μελετήθηκε η επίπτωση που θα

είχαν στην αποδοτικότητα του προγράμματος δηλαδή το μέσο κόστος παραβίασης των περιορισμών.



Εικόνα 5.6 Διάγραμμα με τον μέσο όρο αποδοτικότητας σε σχέση με την πιθανότητα μετάλλαξης

Από τα αποτελέσματα όπως αυτά παρουσιάζονται αναλυτικά στο παραπάνω διάγραμμα την καλύτερη αποδοτικότητα την έχουμε για την πιθανότητα μετάλλαξης 3% με μέσο κόστος παραβίασης των περιορισμών 55316 και είναι η χαμηλότερη τιμή των δοκιμών. Μεγαλύτερες πιθανότητες σύμφωνα με τις δοκιμές που έγιναν έδωσαν χειρότερη αποδοτικότητα με το κόστος να αυξάνει σταδιακά με χειρότερη τιμή αυτή με μέσο κόστος 875100 να αντιστοιχεί στην πιθανότητα 50% που ήταν και η μεγαλύτερη των δοκιμών.

Για την ερμηνεία των παραπάνω αποτελεσμάτων θα πρέπει να λάβουμε υπόψη την μέθοδο επιλογής που εφαρμόζεται στον γενετικό αλγόριθμο που εκτελέσαμε. Στο αρχείο παραμέτρων `sum.params` υπάρχουν οι δηλώσεις `select.tournament.size = 5` και `breed.elite.0 = 3`. Αυτό σημαίνει ότι για την αναπαραγωγή τα υποψήφια μέλη χωρίζονται αρχικά με τυχαίο τρόπο σε ομάδες των 5 μελών και από κάθε ομάδα επιλέγεται το μέλος με την καλύτερη τιμή καταλληλότητας. Επίσης συμπληρωματικά επιλέγονται από όλο το πληθυσμό 3 μέλη με την καλύτερη τιμή καταλληλότητας. Έτσι στα υποψήφια μέλη για αναπαραγωγή θα περιέχονται θεωρητικά τα καλύτερα μέλη της προηγούμενης γενιάς.

Μετά την αναπαραγωγή, θεωρητικά προκύπτουν ακόμα καλύτεροι απόγονοι και κατά τη διαδικασία της μετάλλαξης, αλλάζουν ορισμένα γονίδια σύμφωνα με τη πιθανότητα μετάλλαξης που έχει οριστεί και ο σκοπός είναι να προκύψουν καλύτεροι απόγονοι τυχαία. Εξαιτίας όμως της φύσης των γονιδίων που χρησιμοποιούνται στο παρόν πρόβλημα, δηλαδή μαθήματα που



πρέπει να εξετάζονται την ίδια ώρα σε πολλές πόλεις, κάθε τυχαία αλλαγή στα γονίδια αντί να βελτιώνει το αποτέλεσμα, εκφυλίζει τελικά τα μέλη του πληθυσμού δίνοντας χειρότερες λύσεις. Έτσι όσο αυξάνεται η πιθανότητα μετάλλαξης τόσο χειρότερα αποτελέσματα παίρνουμε.

### 5.3.3 Εκτέλεση του Προγράμματος για Μεγάλο Πλήθος Γενεών

Για να έχουμε καλύτερη εικόνα σχετικά με τη συμπεριφορά του λογισμικού που δημιουργήθηκε στα πλαίσια αυτής της διατριβής όπως και για τον έλεγχο της ακρίβειας των παραγόμενων αποτελεσμάτων οδηγηθήκαμε στην απόφαση να εκτελέσουμε το λογισμικό για μεγάλο σχετικά πλήθος γενεών.

Στον ίδιο server με όνομα `ui.afroditi.hellasgrid.gr` εκτελέστηκε το πρόγραμμα για 20000 γενεές (κατά τη διάρκεια αργιών). Η διάρκεια εκτέλεσης ήταν 23 ώρες 10 λεπτά και 57 δευτερόλεπτα. Για την εκτέλεση κάθε γενιάς απαιτήθηκαν 4,17 δευτερόλεπτα ή για την εκτέλεση 1000 γενεών απαιτήθηκαν 69,54 λεπτά.

Όσον αφορά την αποτελεσματικότητα, αυτή διαμορφώθηκε στις 57100 μονάδες κόστους και προέκυψε στην 6884η γενιά. Οι μεταβλητές που αντιπροσωπεύουν τους περιορισμούς είχαν τιμή: `lesSameHour = 0`, `multiRoomCapacity = 32`, `wrongLessonRoom = 0`, `stdSameHour = 35`, `ExmSameHour = 0`, `lesSameDay = 0`, `lesNextDay = 150`, `lesNextTwoDays = 78`. Ενδιαφέρον παρουσιάζει η αμέσως προηγούμενη καλύτερη τιμή η οποία προέκυψε στην 326η γενιά με κόστος 57800 και τιμές στις μεταβλητές: `lesSameHour= 0`, `multiRoomCapacity= 32`, `wrongLessonRoom= 0`, `stdSameHour= 35`, `ExmSameHour= 0`, `lesSameDay= 0`, `lesNextDay= 151` και `lesNextTwoDays= 90`.

Από τα παραπάνω αποτελέσματα εύκολα μπορούμε να βγάλουμε το συμπέρασμα ότι η εκτέλεση για μεγάλο πλήθος γενεών δεν βελτιώνει ουσιαστικά τα παραγόμενα αποτελέσματα τουλάχιστον για τα δεδομένα που χρησιμοποιήθηκαν στην παρούσα εργασία. Το κόστος 57100 που υπολογίστηκε με πιθανότητα μετάλλαξης 3%, είναι κοντά στη μέση τιμή που υπολογίστηκε και παρουσιάζεται στο διάγραμμα της εικόνας 5.6.

Παρακάτω παρουσιάζεται το αποδεκτό πρόγραμμα των εξετάσεων όπως διαμορφώθηκε μετά από την εκτέλεση 2000 γενεών:

EXAMINATION TIMETABLING SCHEDULE									
tslt	day	time	Lesson	City	Room	StudentsInRoom	RoomCapacity	LessonCapacity	Examiner
0	1	09:00-12:00	AAA37	LARISSA	L1L1801	14	18	14	Eystratios Kalimeris
0	1	09:00-12:00	AAA37	THESSALONIKI	AUTH2607	38	45	38	Kostantinos Antoniou
0	1	09:00-12:00	AAA37	PEIRAIAS	P1PE702	12	80	12	Nikolaos Georgiou
0	1	09:00-12:00	AAA37	ATHINA	AMF11	51	110	51	Anna Nikolaou
3	2	09:00-12:00	AAA22	XANTHI	X1X102	16	16	16	Periklis Zisis
3	2	09:00-12:00	AAA22	LARISSA	L1L2804	12	12	12	Georgia Antoniou
3	2	09:00-12:00	AAA22	THESSALONIKI	AUTH2608	70	100	70	Iason Iasonidis
3	2	09:00-12:00	AAA22	PEIRAIAS	P1PE702	40	80	40	Antonia Peraki
3	2	09:00-12:00	AAA22	PAT	PA1PAM13	26	100	26	Ioannis Fytrakis
3	2	09:00-12:00	AAA22	IRAKLEIO	HB1H401	21	50	21	Mixalis Kostantinou
3	2	09:00-12:00	AAA22	ATHINA	AMF21	138	200	138	Xrysoula Diavati
9	4	09:00-12:00	AAA24	LARISSA	L1L2804	12	12	12	Eirini Moyraki
9	4	09:00-12:00	AAA24	THESSALONIKI	AUTH2608	58	100	58	Stelios Iakovou
9	4	09:00-12:00	AAA24	THESSALONIKI	AUTH2608	30	100	30	Petroula Xristou
9	4	09:00-12:00	AAA24	PEIRAIAS	P1PE702	22	80	22	Aikaterini Petrividou
9	4	09:00-12:00	AAA36	PEIRAIAS	P1PE702	22	80	22	Ioannis Melaxrinos
9	4	09:00-12:00	AAA24	IRAKLEIO	HB1H401	14	50	14	Mixalis Papadopoulos
9	4	09:00-12:00	AAA36	IRAKLEIO	HB1H401	11	50	11	Georgios krasas
9	4	09:00-12:00	AAA24	ATHINA	AMF11	110	110	110	Kostantinos Antoniou
9	4	09:00-12:00	AAA36	ATHINA	B4304	36	50	36	Stamatia Xristodoulou
12	5	09:00-12:00	AAA10	XANTHI	X1X103	36	36	52	Xrysoula Dimopoulou
12	5	09:00-12:00	AAA10	XANTHI	X1X102	16	16	52	Xrysoula Dimopoulou
12	5	09:00-12:00	AAA10	LARISSA	L1L3805	35	35	64	Eirini Antoniou
12	5	09:00-12:00	AAA10	LARISSA	L1L4806	20	20	64	Eirini Antoniou
12	5	09:00-12:00	AAA10	LARISSA	L1L2804	9	12	64	Eirini Antoniou
12	5	09:00-12:00	AAA10	IOANNINA	I21P12	12	12	36	Stelios Orfanos
12	5	09:00-12:00	AAA10	IOANNINA	HB1P11	9	9	36	Stelios Orfanos
12	5	09:00-12:00	AAA10	IOANNINA	I11P10	9	9	36	Stelios Orfanos
12	5	09:00-12:00	AAA10	IOANNINA	I11P11	6	9	36	Stelios Orfanos
12	5	09:00-12:00	AAA10	THESSALONIKI	AUTH2608	100	100	232	Aikaterini Stamatiou
12	5	09:00-12:00	AAA10	THESSALONIKI	AUTH2606	45	45	232	Aikaterini Stamatiou
12	5	09:00-12:00	AAA10	THESSALONIKI	AUTH2607	45	45	232	Aikaterini Stamatiou
12	5	09:00-12:00	AAA10	THESSALONIKI	AUTH1603	28	28	232	Aikaterini Stamatiou
12	5	09:00-12:00	AAA10	THESSALONIKI	AUTH1602	14	20	232	Aikaterini Stamatiou
12	5	09:00-12:00	AAA10	PEIRAIAS	P1PE701	80	80	158	Petros Ioannou
12	5	09:00-12:00	AAA10	PEIRAIAS	P1PE702	78	80	158	Petros Ioannou
12	5	09:00-12:00	AAA10	PAT	PA1PAM13	100	100	104	Kostantinos Kalimeris
12	5	09:00-12:00	AAA10	PAT	PA1P21	4	15	104	Kostantinos Kalimeris
12	5	09:00-12:00	AAA10	IRAKLEIO	HB1H402	60	60	108	Nikolaos Antoniou
12	5	09:00-12:00	AAA10	IRAKLEIO	HB1H401	48	50	108	Nikolaos Antoniou
12	5	09:00-12:00	AAA10	ATHINA	AMF21	200	200	530	Anna Georgiou
12	5	09:00-12:00	AAA10	ATHINA	AMF11	110	110	530	Anna Georgiou
12	5	09:00-12:00	AAA10	ATHINA	B4116	50	50	530	Anna Georgiou
12	5	09:00-12:00	AAA10	ATHINA	B4303	50	50	530	Anna Georgiou
12	5	09:00-12:00	AAA10	ATHINA	B4304	50	50	530	Anna Georgiou
12	5	09:00-12:00	AAA10	ATHINA	B1108	20	20	530	Anna Georgiou
12	5	09:00-12:00	AAA10	ATHINA	B1109	20	20	530	Anna Georgiou
12	5	09:00-12:00	AAA10	ATHINA	B3112	19	19	530	Anna Georgiou
12	5	09:00-12:00	AAA10	ATHINA	B2111	11	16	530	Anna Georgiou
19	7	13:00-16:00	AAA40	PAT	PA1P21	8	15	8	Nikolaos Stamatopoulos
19	7	13:00-16:00	AAA35	ATHINA	B4304	33	50	33	Xristos Petridis
22	8	13:00-16:00	AAA12	XANTHI	X1X103	36	36	42	Kostantinos Papadopoulos
22	8	13:00-16:00	AAA12	XANTHI	X1X102	6	16	42	Kostantinos Papadopoulos
22	8	13:00-16:00	AAA12	LARISSA	L1L3805	35	35	38	Nikolaos Antoniou
22	8	13:00-16:00	AAA12	LARISSA	L1L2804	3	12	38	Nikolaos Antoniou
22	8	13:00-16:00	AAA12	IOANNINA	I21P12	12	12	23	Anna Vagena
22	8	13:00-16:00	AAA12	IOANNINA	HB1P11	9	9	23	Anna Vagena
22	8	13:00-16:00	AAA12	IOANNINA	I11P11	2	9	23	Anna Vagena
22	8	13:00-16:00	AAA12	THESSALONIKI	AUTH2608	100	100	167	Iason Petridis
22	8	13:00-16:00	AAA12	THESSALONIKI	AUTH2606	45	45	167	Iason Petridis
22	8	13:00-16:00	AAA12	THESSALONIKI	AUTH1605	22	25	167	Iason Petridis
22	8	13:00-16:00	AAA12	PEIRAIAS	P1PE701	80	80	85	Theodoros Kalogiannis
22	8	13:00-16:00	AAA12	PEIRAIAS	P1PE702	5	80	85	Theodoros Kalogiannis
22	8	13:00-16:00	AAA12	PAT	PA1PAM13	52	100	52	Ioannis Melitos
22	8	13:00-16:00	AAA12	ATHINA	AMF21	200	200	372	Georgios Asteriou
22	8	13:00-16:00	AAA12	ATHINA	AMF11	110	110	372	Georgios Asteriou
22	8	13:00-16:00	AAA12	ATHINA	B4116	50	50	372	Georgios Asteriou
22	8	13:00-16:00	AAA12	ATHINA	B2111	12	16	372	Georgios Asteriou
22	8	13:00-16:00	AAA44	ATHINA	B3114	9	17	9	Xristos Petridis
29	10	17:00-20:00	AAA30	LARISSA	L1L2804	12	12	12	Athanasios Asteriou
29	10	17:00-20:00	AAA30	THESSALONIKI	AUTH2608	47	100	47	Georgios Dimopoulos
29	10	17:00-20:00	AAA30	PEIRAIAS	P1PE702	23	80	23	Athanasios Antoniou
29	10	17:00-20:00	AAA30	PAT	PA1PAM13	22	100	22	Xanthippi Orfanou
29	10	17:00-20:00	AAA30	IRAKLEIO	HB1H401	11	50	11	Stayros Stamatiou
29	10	17:00-20:00	AAA30	ATHINA	AMF11	103	110	103	Aikaterini Ioannou
33	12	09:00-12:00	AAA23	LARISSA	L1L1801	16	18	16	Antonios Vagenas
33	12	09:00-12:00	AAA23	THESSALONIKI	AUTH2607	34	45	34	Anna Petrividou
33	12	09:00-12:00	AAA23	PEIRAIAS	P1PE702	25	80	25	Iason Kalogiannis
33	12	09:00-12:00	AAA23	ATHINA	AMF11	55	110	55	Dimitrios Melitos
41	14	17:00-20:00	AAA11	XANTHI	X1X103	17	36	17	Theodoros Amanatidis
41	14	17:00-20:00	AAA11	LARISSA	L1L3805	24	35	24	Ioannis Atsalos
41	14	17:00-20:00	AAA11	THESSALONIKI	AUTH2608	95	100	95	Georgios Petrou
41	14	17:00-20:00	AAA11	PEIRAIAS	P1PE702	57	80	57	Paulos Keramidas
41	14	17:00-20:00	AAA11	PAT	PA1PAM13	37	100	37	Maria Xristou
41	14	17:00-20:00	AAA11	IRAKLEIO	HB1H401	24	50	24	Stelios Melaxrinos
41	14	17:00-20:00	AAA11	ATHINA	AMF21	200	200	232	Konstantinos krasas
41	14	17:00-20:00	AAA11	ATHINA	B4304	32	50	232	Konstantinos krasas
42	15	09:00-12:00	AAA31	THESSALONIKI	AUTH2608	55	100	55	Ioannis Amanatidis
42	15	09:00-12:00	AAA31	PEIRAIAS	P1PE702	21	80	21	Konstantinos Atsalos
42	15	09:00-12:00	AAA31	IRAKLEIO	HB1H401	14	50	14	Panagiotis Petrou
42	15	09:00-12:00	AAA31	ATHINA	AMF11	73	110	73	Konstantinos Keramidas
47	16	17:00-20:00	AAA21	XANTHI	X1X102	15	16	15	Georgios Maronidis
47	16	17:00-20:00	AAA21	LARISSA	L1L1801	17	18	17	Xrysoula Griraki
47	16	17:00-20:00	AAA21	THESSALONIKI	AUTH2608	94	100	94	Xristos Aronis
47	16	17:00-20:00	AAA21	PEIRAIAS	P1PE702	44	80	44	Stelios Petrakis
47	16	17:00-20:00	AAA21	PAT	PA1PAM13	31	100	31	Aikaterini Kokkinou
47	16	17:00-20:00	AAA21	IRAKLEIO	HB1H401	26	50	26	Panagiotis Petinakis
47	16	17:00-20:00	AAA21	ATHINA	AMF21	160	200	160	Kostantinos Eroglou
49	17	13:00-16:00	AAA20	XANTHI	X1X103	17	36	17	Petros Xristodoulou
49	17	13:00-16:00	AAA20	LARISSA	L1L1801	15	18	15	Kostantinos Nikolaou
49	17	13:00-16:00	AAA20	THESSALONIKI	AUTH2608	95	100	95	Nikolaos Papadopoulos
49	17	13:00-16:00	AAA20	PEIRAIAS	P1PE702	47	80	47	Anna Stamatopoulos
49	17	13:00-16:00	AAA20	PAT	PA1PAM13	34	100	34	Panagiotis Petridis
49	17	13:00-16:00	AAA20	IRAKLEIO	HB1H401	14	50	14	Theodoros Stamatakis
49	17	13:00-16:00	AAA20	ATHINA	AMF21	169	200	169	Ioannis Arxontidis
59	20	17:00-20:00	AAA42	PEIRAIAS	P1PE702	20	80	20	Athanasios Nikolaou
59	20	17:00-20:00	AAA42	ATHINA	AMF11	55	110	55	Kostantinos Papadopoulos

## 5.4 Εκτέλεση του Προγράμματος στο Grid

Η υποδομή του οργανισμού HellasGrid, δίνει τη δυνατότητα στους χρήστες της χώρας μας να εκτελέσουν προγράμματα που απαιτούν μεγάλη υπολογιστική ισχύ. Παρόλο που η διαδικασία υποβολής προγραμμάτων προς εκτέλεση προς το Grid είναι μια απλή διαδικασία, δεν ενδείκνυται για εφαρμογές που απαιτούν άμεση απόκριση διότι τα προγράμματα στο πλέγμα εκτελούνται εφόσον υπάρχουν διαθέσιμοι πόροι και συνήθως... δεν υπάρχουν. Έτσι για να ξεκινήσει την εκτέλεσή του ένα πρόγραμμα μπορεί να απαιτηθεί χρόνος αναμονής που μπορεί να φτάσει και τις μερικές ημέρες.

Για τις ανάγκες τις παρούσας διατριβής εκτελέσθηκαν δύο σενάρια ώστε να συγκριθούν τα παραγόμενα αποτελέσματα, με τα αποτελέσματα που είχαν προκύψει από προηγούμενες δοκιμές.

**1<sup>ο</sup> Σενάριο**, για 10000 γενεές, με πληθυσμό 300 ατόμων και πιθανότητα μετάλλαξης 3%, είχαμε το εξής αποτέλεσμα:

Το πρόγραμμα εκτελέσθηκε στον κόμβο με hostname:

**sardalya24.ulakbim.gov.tr**

ο οποίος με 8 επεξεργαστές εκτέλεσε το πρόγραμμα σε 11 ώρες 42 λεπτά και 35 δευτερόλεπτα, ή σε 42155 δευτερόλεπτα δηλαδή για κάθε γενιά απαιτήθηκαν 4,21 δευτερόλεπτα. Όσον αφορά την αποτελεσματικότητα το κόστος παραβίασης των περιορισμών ήταν 47900 μονάδες και επιτευχθεί κατά την επεξεργασία της 8189ης γενιάς. Οι μεταβλητές που απεικονίζουν την παραβίαση των περιορισμών είχαν τιμή: lesSameHour= 0, multiRoomCapacity= 32, wrongLessonRoom= 4, stdSameHour= 25, ExmSameHour= 0, lesSameDay= 20, lesNextDay= 32, lesNextTwoDays= 170.

Ενδιαφέρον προκαλεί η μεταβλητή wrongLessonRoom που έχει τιμή 4. Αυτό σημαίνει ότι σε μαθήματα δεν υπήρχε αρκετός χώρος στις αίθουσες εξετάσεων ώστε να πραγματοποιηθούν οι εξετάσεις και έτσι το πρόγραμμα στη παρούσα φάση δεν μπορεί να είναι αποδεκτό. Αν παρατηρήσουμε το πρόγραμμα που φαίνεται παρακάτω, πρόκειται για το μάθημα AAA24. Το μάθημα αυτό εξετάζεται στη Λάρισα την 2η ημέρα εξετάσεων από 09:00 έως 12:00, αλλά δεν μπορεί να εξετασθεί στις πόλεις Θεσσαλονίκη, Πειραιά, Ηράκλειο και Αθήνα επειδή δεν υπάρχει

χώρος διαθέσιμος στις αίθουσες εξετάσεων. Εκείνη την ώρα όπως φαίνεται στο πρόγραμμα εξετάζεται το μάθημα AAA10 και στις πόλεις που αναφέρθηκαν προηγουμένως δεν υπάρχει χώρος στις αίθουσες εξετάσεων για το μάθημα AAA24.

Το πρόγραμμα των εξετάσεων όπως διαμορφώθηκε μετά από την εκτέλεση του προγράμματος.

EXAMINATION TIMETABLING SCHEDULE										
tslt	day	time	Lesson	City	Room	StudentsInRoom	RoomCapacity	LessonCapacity	Examiner	
3	2	09:00-12:00	AAA10	XANTHI	XIX103	36	36	52	Xrysoula	Dimopoulou
3	2	09:00-12:00	AAA10	XANTHI	XIX102	16	16	52	Xrysoula	Dimopoulou
3	2	09:00-12:00	AAA10	LARISSA	L1L3805	35	35	64	Eirini	Antonou
3	2	09:00-12:00	AAA10	LARISSA	L1L4806	20	20	64	Eirini	Antonou
3	2	09:00-12:00	AAA10	LARISSA	L1L2804	9	12	64	Eirini	Antonou
3	2	09:00-12:00	AAA24	LARISSA	L1L1801	12	18	12	Eirini	Moyraki
3	2	09:00-12:00	AAA10	IOANNINA	I2IP12	12	12	36	Stelios	Orfanos
3	2	09:00-12:00	AAA10	IOANNINA	HB1IP1	9	9	36	Stelios	Orfanos
3	2	09:00-12:00	AAA10	IOANNINA	I1IP10	9	9	36	Stelios	Orfanos
3	2	09:00-12:00	AAA10	IOANNINA	I1IP11	6	9	36	Stelios	Orfanos
3	2	09:00-12:00	AAA10	THESSALONIKI	AUTH2608	100	100	232	Aikaterini	Stamatou
3	2	09:00-12:00	AAA10	THESSALONIKI	AUTH2606	45	45	232	Aikaterini	Stamatou
3	2	09:00-12:00	AAA10	THESSALONIKI	AUTH2607	45	45	232	Aikaterini	Stamatou
3	2	09:00-12:00	AAA10	THESSALONIKI	AUTH1603	28	28	232	Aikaterini	Stamatou
3	2	09:00-12:00	AAA10	THESSALONIKI	AUTH1602	14	20	232	Aikaterini	Stamatou
3	2	09:00-12:00	AAA10	PEIRAIAS	P1PE701	80	80	158	Petros	Ioannou
3	2	09:00-12:00	AAA10	PEIRAIAS	P1PE702	78	80	158	Petros	Ioannou
3	2	09:00-12:00	AAA10	PAT	PALPAM13	100	100	104	Kostantinos	Kalimeris
3	2	09:00-12:00	AAA10	PAT	PALP21	4	15	104	Kostantinos	Kalimeris
3	2	09:00-12:00	AAA10	IRAKLEIO	HB1H402	60	60	108	Nikolaos	Antonou
3	2	09:00-12:00	AAA10	IRAKLEIO	HB1H401	48	50	108	Nikolaos	Antonou
3	2	09:00-12:00	AAA10	ATHINA	AMP21	200	200	530	Anna	Georgiou
3	2	09:00-12:00	AAA10	ATHINA	AMP11	110	110	530	Anna	Georgiou
3	2	09:00-12:00	AAA10	ATHINA	B4116	50	50	530	Anna	Georgiou
3	2	09:00-12:00	AAA10	ATHINA	B4303	50	50	530	Anna	Georgiou
3	2	09:00-12:00	AAA10	ATHINA	B4304	50	50	530	Anna	Georgiou
3	2	09:00-12:00	AAA10	ATHINA	B1108	20	20	530	Anna	Georgiou
3	2	09:00-12:00	AAA10	ATHINA	B1109	20	20	530	Anna	Georgiou
3	2	09:00-12:00	AAA10	ATHINA	B3112	19	19	530	Anna	Georgiou
3	2	09:00-12:00	AAA10	ATHINA	B2111	11	16	530	Anna	Georgiou
4	2	09:00-16:00	AAA23	LARISSA	L1L1801	16	18	16	Antonios	Vagenas
4	2	13:00-16:00	AAA23	THESSALONIKI	AUTH2607	34	45	34	Anna	Petridou
4	2	13:00-16:00	AAA23	PEIRAIAS	P1PE702	25	80	25	Iason	Kalogiannis
4	2	13:00-16:00	AAA23	ATHINA	AMP11	55	110	55	Dimitrios	Melitos
6	3	09:00-12:00	AAA20	XANTHI	XIX103	17	36	17	Petros	Xristodoulou
6	3	09:00-12:00	AAA20	LARISSA	L1L1801	15	18	15	Kostantinos	Nikolaou
6	3	09:00-12:00	AAA20	THESSALONIKI	AUTH2608	95	100	95	Nikolaos	Papadopoulos
6	3	09:00-12:00	AAA20	PEIRAIAS	P1PE702	47	80	47	Anna	Stamatopoulou
6	3	09:00-12:00	AAA20	PAT	PALPAM13	34	100	34	Panagiotis	Petridis
6	3	09:00-12:00	AAA20	IRAKLEIO	HB1H401	14	50	14	Theodoros	Stamatakis
6	3	09:00-12:00	AAA20	ATHINA	AMP21	169	200	169	Ioannis	Arxontidis
17	6	17:00-20:00	AAA22	XANTHI	XIX102	16	16	16	Periklis	Zisis
17	6	17:00-20:00	AAA22	LARISSA	L1L2804	12	12	12	Georgia	Antonou
17	6	17:00-20:00	AAA22	THESSALONIKI	AUTH2608	70	100	70	Iason	Iasonidis
17	6	17:00-20:00	AAA36	THESSALONIKI	AUTH2608	30	100	30	Petroula	Xristou
17	6	17:00-20:00	AAA22	PEIRAIAS	P1PE702	40	80	40	Antonia	Peraki
17	6	17:00-20:00	AAA36	PEIRAIAS	P1PE702	22	80	22	Ioannis	Melaxrinou
17	6	17:00-20:00	AAA22	PAT	PALPAM13	26	100	26	Ioannis	Melaxrinou
17	6	17:00-20:00	AAA22	IRAKLEIO	HB1H401	21	50	21	Mixalis	Kostantinou
17	6	17:00-20:00	AAA36	IRAKLEIO	HB1H401	11	50	11	Georgios	Krasas
17	6	17:00-20:00	AAA22	ATHINA	AMP21	138	200	138	Xrysoula	Diavati
17	6	17:00-20:00	AAA36	ATHINA	B4304	36	50	36	Stamatia	Xristodoulou
23	8	17:00-20:00	AAA21	XANTHI	XIX102	15	16	15	Georgios	Maronidis
23	8	17:00-20:00	AAA21	LARISSA	L1L1801	17	18	17	Xrysoula	Giraki
23	8	17:00-20:00	AAA21	THESSALONIKI	AUTH2608	94	100	94	Xristos	Aronis
23	8	17:00-20:00	AAA21	PEIRAIAS	P1PE702	44	80	44	Stelios	Petrakis
23	8	17:00-20:00	AAA21	PAT	PALPAM13	31	100	31	Aikaterini	Kokkinou
23	8	17:00-20:00	AAA21	IRAKLEIO	HB1H401	26	50	26	Panagiotis	Petinakis
23	8	17:00-20:00	AAA21	ATHINA	AMP21	160	200	160	Kostantinos	Eroglou
27	10	09:00-12:00	AAA11	XANTHI	XIX103	17	36	17	Theodoros	Amanatidis
27	10	09:00-12:00	AAA11	LARISSA	L1L3805	24	35	24	Ioannis	Atsalos
27	10	09:00-12:00	AAA37	LARISSA	L1L1801	14	18	14	Eystratios	Kalimeris
27	10	09:00-12:00	AAA11	THESSALONIKI	AUTH2608	95	100	95	Georgios	Petrou
27	10	09:00-12:00	AAA37	THESSALONIKI	AUTH2607	38	45	38	Kostantinos	Antonou
27	10	09:00-12:00	AAA11	PEIRAIAS	P1PE702	57	80	57	Paulos	Keramidas
27	10	09:00-12:00	AAA37	PEIRAIAS	P1PE702	22	80	22	Nikolaos	Georgiou
27	10	09:00-12:00	AAA11	PAT	PALPAM13	37	100	37	Maria	Xristou
27	10	09:00-12:00	AAA11	IRAKLEIO	HB1H401	24	50	24	Stelios	Melaxrinou
27	10	09:00-12:00	AAA11	ATHINA	AMP21	200	200	232	Konstantinos	Krasas
27	10	09:00-12:00	AAA11	ATHINA	B4304	32	50	232	Konstantinos	Krasas
27	10	09:00-12:00	AAA37	ATHINA	AMP11	51	110	51	Anna	Nikolaou
33	12	09:00-12:00	AAA31	THESSALONIKI	AUTH2608	55	100	55	Ioannis	Amanatidis
33	12	09:00-12:00	AAA31	PEIRAIAS	P1PE702	21	80	21	Konstantinos	Atsalos
33	12	09:00-12:00	AAA31	IRAKLEIO	HB1H401	14	50	14	Panagiotis	Petrou
33	12	09:00-12:00	AAA31	ATHINA	AMP11	73	110	73	Konstantinos	Keramidas
44	15	17:00-20:00	AAA30	LARISSA	L1L2804	12	12	12	Athanasios	Asteriou
44	15	17:00-20:00	AAA30	THESSALONIKI	AUTH2608	47	100	47	Georgios	Dimopoulos
44	15	17:00-20:00	AAA30	PEIRAIAS	P1PE702	23	80	23	Athanasios	Antonou
44	15	17:00-20:00	AAA30	PAT	PALPAM13	22	100	22	Xanthippi	Orfanou
44	15	17:00-20:00	AAA30	IRAKLEIO	HB1H401	11	50	11	Stayros	Stamatou
44	15	17:00-20:00	AAA30	ATHINA	AMP11	103	110	103	Aikaterini	Ioannou
46	16	13:00-16:00	AAA40	PAT	PALP21	8	15	8	Nikolaos	Stamatopoulos
47	16	17:00-20:00	AAA35	ATHINA	B4304	33	50	33	Xristos	Petridis
49	17	13:00-16:00	AAA12	XANTHI	XIX103	36	36	42	Kostantinos	Papadopoulos
49	17	13:00-16:00	AAA12	XANTHI	XIX102	6	16	42	Kostantinos	Papadopoulos
49	17	13:00-16:00	AAA12	LARISSA	L1L3805	35	35	38	Nikolaos	Antonou
49	17	13:00-16:00	AAA12	LARISSA	L1L2804	3	12	38	Nikolaos	Antonou
49	17	13:00-16:00	AAA12	IOANNINA	I2IP12	12	12	23	Anna	Vagena
49	17	13:00-16:00	AAA12	IOANNINA	HB1IP1	9	9	23	Anna	Vagena
49	17	13:00-16:00	AAA12	IOANNINA	I1IP11	2	9	23	Anna	Vagena
49	17	13:00-16:00	AAA12	THESSALONIKI	AUTH2608	100	100	167	Iason	Petridis
49	17	13:00-16:00	AAA12	THESSALONIKI	AUTH2606	45	45	167	Iason	Petridis
49	17	13:00-16:00	AAA12	THESSALONIKI	AUTH1605	22	25	167	Iason	Petridis
49	17	13:00-16:00	AAA12	PEIRAIAS	P1PE701	80	80	85	Theodoros	Kalogiannis
49	17	13:00-16:00	AAA12	PEIRAIAS	P1PE702	5	80	85	Theodoros	Kalogiannis
49	17	13:00-16:00	AAA12	PAT	PALPAM13	52	100	52	Ioannis	Melitos
49	17	13:00-16:00	AAA12	ATHINA	AMP21	200	200	372	Georgios	Asteriou
49	17	13:00-16:00	AAA12	ATHINA	AMP11	110	110	372	Georgios	Asteriou
49	17	13:00-16:00	AAA12	ATHINA	B4116	50	50	372	Georgios	Asteriou
49	17	13:00-16:00	AAA12	ATHINA	B2111	12	16	372	Georgios	Asteriou
55	19	13:00-16:00	AAA42	PEIRAIAS	P1PE702	20	80	20	Athanasios	Nikolaou
55	19	13:00-16:00	AAA42	ATHINA	AMP11	55	110	55	Kostantinos	Papadopoulos
56	19	17:00-20:00	AAA44	ATHINA	B2111	9	16	9	Xristos	Petridis

**2ο Σενάριο**, για 10000 γενεές, με πληθυσμό 300 ατόμων και πιθανότητα μετάλλαξης 25%, είχαμε το εξής αποτέλεσμα:

Το πρόγραμμα εκτελέστηκε στον κόμβο με hostname:

### **wn08.hpcg**

ο οποίος με 16 επεξεργαστές έτρεξε το πρόγραμμα σε 6 ώρες 7 λεπτά και 14 δευτερόλεπτα ή 22034 δευτερόλεπτα δηλαδή για την κάθε γενιά απαιτήθηκε 2,2 δευτερόλεπτα. Η αποτελεσματικότητα στις 703500 μονάδες κόστους στην 9692η γενιά και οι μεταβλητές των περιορισμών είχαν τιμές: lesSameHour= 129, multiRoomCapacity= 32, wrongLessonRoom= 0, stdSameHour= 28, ExmSameHour= 0, lesSameDay= 50, lesNextDay= 69 και lesNextTwoDays= 208.

Όσον αφορά τα δύο πρώτα σενάρια, ο δεύτερος κόμβος με τους 16 επεξεργαστές έτρεξε το πρόγραμμα 91% περίπου γρηγορότερα από τον πρώτο με τους 8 επεξεργαστές. Επίσης η αποτελεσματικότητα του πρώτου ήταν 47900 για πιθανότητα μετάλλαξης 3% ενώ του δεύτερου 703500 για πιθανότητα μετάλλαξης 25%, παρακολουθούν την καμπύλη αποδοτικότητας όπως παρουσιάζεται στο διάγραμμα της εικόνας 5.6.

**3ο Σενάριο**, για 3000 γενεές, με πληθυσμό 300 ατόμων και πιθανότητα μετάλλαξης 3%, είχαμε το εξής αποτέλεσμα:

Το πρόγραμμα εκτελέστηκε στον κόμβο με hostname:

### **oar15.grid.metu.edu.tr**

Ο οποίος με 4 επεξεργαστές έτρεξε το πρόγραμμα σε 11 ώρες 20 λεπτά και 55 δευτερόλεπτα ή 40855 δευτερόλεπτα δηλαδή 13,6 δευτερόλεπτα τη γενιά. Η αποτελεσματικότητα είχε κόστος 45850 μονάδες που επετεύχθη στην 252η γενιά. Οι τιμές για τις μεταβλητές των περιορισμών έχουν ως εξής: lesSameHour= 0, multiRoomCapacity= 32, wrongLessonRoom= 0, stdSameHour= 25, ExmSameHour= 0, lesSameDay= 44, lesNextDay= 42 και lesNextTwoDays= 93. Το εξεταστικό πρόγραμμα που διαμορφώθηκε αυτόματα είναι το παρακάτω:

#### EXAMINATION TIMETABLING SCHEDULE

tslt	day	time	Lesson Capacity	Lesson Examiner	City	Room	StudentsInRoom	RoomCapacity	
4	2	13:00-16:00	AAA36	THESSALONIKI	AUTH2607	30	45	Petroula	Xristou
4	2	13:00-16:00	AAA36	PEIRAIAS	PIPE702	22	80	Ioannis	Melaxrinos

4	2	13:00-16:00	AAA36	IRAKLEIO	HB1H401	11	50	11	Georgios	Krasas
4	2	13:00-16:00	AAA36	ATHINA	B4304	36	50	36	Stamatis	Xristodoulou
5	2	17:00-20:00	AAA24	LARISSA	L1L2804	12	12	12	Eirini	Moyraki
5	2	17:00-20:00	AAA24	THESSALONIKI	AUTH2608	58	100	58	Stelios	Iakovou
5	2	17:00-20:00	AAA24	PEIRAIAS	P1PE702	22	80	22	Aikaterini	Petridou
5	2	17:00-20:00	AAA24	IRAKLEIO	HB1H401	14	50	14	Mixalis	Papadopoulos
5	2	17:00-20:00	AAA24	ATHINA	AMF11	110	110	110	Kostantinos	Antoniu
8	3	17:00-20:00	AAA11	XANTHI	X1X103	17	36	17	Theodoros	Amanatidis
8	3	17:00-20:00	AAA11	LARISSA	L1L3805	24	35	24	Ioannis	Atsalos
8	3	17:00-20:00	AAA11	THESSALONIKI	AUTH2608	95	100	95	Georgios	Petrou
8	3	17:00-20:00	AAA11	PEIRAIAS	P1PE702	57	80	57	Paulos	Keramidas
8	3	17:00-20:00	AAA11	PAT	PA1PAM13	37	100	37	Maria	Xristou
8	3	17:00-20:00	AAA11	IRAKLEIO	HB1H401	24	50	24	Stelios	Melaxrinos
8	3	17:00-20:00	AAA11	ATHINA	AMF21	200	200	232	Konstantinos	Krasas
8	3	17:00-20:00	AAA11	ATHINA	B4304	32	50	32	Konstantinos	Krasas
10	4	13:00-16:00	AAA22	XANTHI	X1X102	16	16	16	Petrikis	Zisis
10	4	13:00-16:00	AAA22	LARISSA	L1L2804	12	12	12	Georgia	Antoniu
10	4	13:00-16:00	AAA22	THESSALONIKI	AUTH2608	70	100	70	Iason	Iasonidis
10	4	13:00-16:00	AAA22	PEIRAIAS	P1PE702	40	80	40	Antonia	Petaki
10	4	13:00-16:00	AAA22	PEIRAIAS	P1PE702	20	80	20	Athanasios	Nikolaou
10	4	13:00-16:00	AAA22	PAT	PA1PAM13	26	100	26	Ioannis	Fytrakis
10	4	13:00-16:00	AAA22	IRAKLEIO	HB1H401	21	50	21	Mixalis	Kostantinou
10	4	13:00-16:00	AAA22	ATHINA	AMF21	138	200	138	Xrysoula	Diavati
10	4	13:00-16:00	AAA22	ATHINA	AMF21	55	200	55	Kostantinos	Papadopoulos
15	6	09:00-12:00	AAA12	XANTHI	X1X103	36	36	42	Kostantinos	Papadopoulos
15	6	09:00-12:00	AAA12	XANTHI	X1X102	6	16	42	Kostantinos	Papadopoulos
15	6	09:00-12:00	AAA12	LARISSA	L1L3805	35	35	38	Nikolaos	Antoniu
15	6	09:00-12:00	AAA12	LARISSA	L1L2804	3	12	38	Nikolaos	Antoniu
15	6	09:00-12:00	AAA12	IOANNINA	I21P12	12	12	23	Anna	Vagena
15	6	09:00-12:00	AAA12	IOANNINA	HB1IP1	9	9	23	Anna	Vagena
15	6	09:00-12:00	AAA12	IOANNINA	I1IP11	2	9	23	Anna	Vagena
15	6	09:00-12:00	AAA12	THESSALONIKI	AUTH2608	100	100	167	Iason	Petridis
15	6	09:00-12:00	AAA12	THESSALONIKI	AUTH2606	45	45	167	Iason	Petridis
15	6	09:00-12:00	AAA12	THESSALONIKI	AUTH1605	22	25	167	Iason	Petridis
15	6	09:00-12:00	AAA12	PEIRAIAS	P1PE701	80	80	85	Theodoros	Kalogiannis
15	6	09:00-12:00	AAA12	PEIRAIAS	P1PE702	5	80	85	Theodoros	Kalogiannis
15	6	09:00-12:00	AAA12	PAT	PA1PAM13	52	100	52	Ioannis	Melitos
15	6	09:00-12:00	AAA12	ATHINA	AMF21	200	200	372	Georgios	Asteriou
15	6	09:00-12:00	AAA12	ATHINA	AMF11	110	110	372	Georgios	Asteriou
15	6	09:00-12:00	AAA12	ATHINA	B4116	50	50	372	Georgios	Asteriou
15	6	09:00-12:00	AAA12	ATHINA	B2111	12	16	372	Georgios	Asteriou
16	6	13:00-16:00	AAA44	ATHINA	B2111	9	16	9	Xristos	Petridis
18	7	09:00-12:00	AAA35	ATHINA	B4304	33	50	33	Xristos	Petridis
26	9	17:00-20:00	AAA40	PAT	PA1P21	8	15	8	Nikolaos	Stamatopoulos
27	10	09:00-12:00	AAA31	THESSALONIKI	AUTH2608	55	100	55	Ioannis	Amanatidis
27	10	09:00-12:00	AAA31	PEIRAIAS	P1PE702	21	80	21	Konstantinos	Atsalos
27	10	09:00-12:00	AAA31	IRAKLEIO	HB1H401	14	50	14	Panagiotis	Petrou
27	10	09:00-12:00	AAA31	ATHINA	AMF11	73	110	73	Konstantinos	Keramidas
29	10	17:00-20:00	AAA21	XANTHI	X1X102	15	16	15	Georgios	Maronidis
29	10	17:00-20:00	AAA21	LARISSA	L1L1801	17	17	17	Xrysoula	Giraki
29	10	17:00-20:00	AAA21	THESSALONIKI	AUTH2608	94	100	94	Xristos	Aronis
29	10	17:00-20:00	AAA21	PEIRAIAS	P1PE702	44	80	44	Stelios	Petrakis
29	10	17:00-20:00	AAA21	PAT	PA1PAM13	31	100	31	Aikaterini	Kokkinou
29	10	17:00-20:00	AAA21	IRAKLEIO	HB1H401	26	50	26	Panagiotis	Petinakis
29	10	17:00-20:00	AAA21	ATHINA	AMF21	160	200	160	Kostantinos	Eroglou
40	14	13:00-16:00	AAA20	XANTHI	X1X103	17	36	17	Petros	Xristodoulou
40	14	13:00-16:00	AAA20	LARISSA	L1L1801	15	18	15	Kostantinos	Nikolaou
40	14	13:00-16:00	AAA20	THESSALONIKI	AUTH2608	95	100	95	Nikolaos	Papadopoulos
40	14	13:00-16:00	AAA20	PEIRAIAS	P1PE702	47	80	47	Anna	Stamatopoulos
40	14	13:00-16:00	AAA20	PAT	PA1PAM13	34	100	34	Panagiotis	Petridis
40	14	13:00-16:00	AAA20	IRAKLEIO	HB1H401	14	50	14	Theodoros	Stamatakis
40	14	13:00-16:00	AAA20	ATHINA	AMF21	169	200	169	Ioannis	Arxontidis
43	15	13:00-16:00	AAA10	XANTHI	X1X103	36	36	52	Xrysoula	Dimopoulou
43	15	13:00-16:00	AAA10	XANTHI	X1X102	16	16	52	Xrysoula	Dimopoulou
43	15	13:00-16:00	AAA10	LARISSA	L1L3805	35	35	64	Eirini	Antoniu
43	15	13:00-16:00	AAA10	LARISSA	L1L4806	20	20	64	Eirini	Antoniu
43	15	13:00-16:00	AAA10	LARISSA	L1L2804	9	12	64	Eirini	Antoniu
43	15	13:00-16:00	AAA10	IOANNINA	I21P12	12	12	36	Stelios	Orfanos
43	15	13:00-16:00	AAA10	IOANNINA	HB1IP1	9	9	36	Stelios	Orfanos
43	15	13:00-16:00	AAA10	IOANNINA	I1IP11	9	9	36	Stelios	Orfanos
43	15	13:00-16:00	AAA10	IOANNINA	I1IP11	6	9	36	Stelios	Orfanos
43	15	13:00-16:00	AAA10	THESSALONIKI	AUTH2608	100	100	232	Aikaterini	Stamatiou
43	15	13:00-16:00	AAA10	THESSALONIKI	AUTH2606	45	45	232	Aikaterini	Stamatiou
43	15	13:00-16:00	AAA10	THESSALONIKI	AUTH2607	45	45	232	Aikaterini	Stamatiou
43	15	13:00-16:00	AAA10	THESSALONIKI	AUTH1603	28	28	232	Aikaterini	Stamatiou
43	15	13:00-16:00	AAA10	THESSALONIKI	AUTH1602	14	20	232	Aikaterini	Stamatiou
43	15	13:00-16:00	AAA10	PEIRAIAS	P1PE701	80	80	158	Petros	Ioannou
43	15	13:00-16:00	AAA10	PEIRAIAS	P1PE702	78	80	158	Petros	Ioannou
43	15	13:00-16:00	AAA10	PAT	PA1PAM13	100	100	104	Kostantinos	Kalimeris
43	15	13:00-16:00	AAA10	PAT	PA1P21	4	15	104	Kostantinos	Kalimeris
43	15	13:00-16:00	AAA10	IRAKLEIO	HB1H402	60	60	108	Nikolaos	Antoniu
43	15	13:00-16:00	AAA10	IRAKLEIO	HB1H401	48	50	108	Nikolaos	Antoniu
43	15	13:00-16:00	AAA10	ATHINA	AMF21	200	200	530	Anna	Georgiou
43	15	13:00-16:00	AAA10	ATHINA	AMF11	110	110	530	Anna	Georgiou
43	15	13:00-16:00	AAA10	ATHINA	B4116	50	50	530	Anna	Georgiou
43	15	13:00-16:00	AAA10	ATHINA	B4303	50	50	530	Anna	Georgiou
43	15	13:00-16:00	AAA10	ATHINA	B4304	50	50	530	Anna	Georgiou
43	15	13:00-16:00	AAA10	ATHINA	B1108	20	20	530	Anna	Georgiou
43	15	13:00-16:00	AAA10	ATHINA	B1109	20	20	530	Anna	Georgiou
43	15	13:00-16:00	AAA10	ATHINA	B3112	19	19	530	Anna	Georgiou
43	15	13:00-16:00	AAA10	ATHINA	B2111	11	16	530	Anna	Georgiou
44	15	17:00-20:00	AAA37	LARISSA	L1L1801	14	18	14	Eystratios	Kalimeris
44	15	17:00-20:00	AAA37	THESSALONIKI	AUTH2607	38	45	38	Kostantinos	Antoniu
44	15	17:00-20:00	AAA37	PEIRAIAS	P1PE702	12	80	12	Nikolaos	Georgiou
44	15	17:00-20:00	AAA37	ATHINA	AMF11	51	110	51	Anna	Nikolaou
46	16	13:00-16:00	AAA30	LARISSA	L1L2804	12	12	12	Athanasios	Asteriou
46	16	13:00-16:00	AAA30	THESSALONIKI	AUTH2608	47	100	47	Georgios	Dimopoulos
46	16	13:00-16:00	AAA30	PEIRAIAS	P1PE702	23	80	23	Athanasios	Antoniu
46	16	13:00-16:00	AAA30	PAT	PA1PAM13	22	100	22	Xanthippi	Orfanou
46	16	13:00-16:00	AAA30	IRAKLEIO	HB1H401	11	50	11	Stavros	Stamatiou
46	16	13:00-16:00	AAA30	ATHINA	AMF11	103	110	103	Aikaterini	Ioannou
54	19	09:00-12:00	AAA23	LARISSA	L1L1801	16	18	16	Antonios	Vagenas
54	19	09:00-12:00	AAA23	THESSALONIKI	AUTH2607	34	45	34	Anna	Petridou
54	19	09:00-12:00	AAA23	PEIRAIAS	P1PE702	25	80	25	Iason	Kalogiannis
54	19	09:00-12:00	AAA23	ATHINA	AMF11	55	110	55	Dimitrios	Melitos

Συγκρίνοντας τα αποτελέσματα του 3<sup>ου</sup> σεναρίου με αυτά της ενότητας 3.10.4 που αφορούσαν έναν φορητό υπολογιστή παρατηρούμε ότι στον φορητό υπολογιστή το πρόγραμμα έτρεξε πολύ πιο γρήγορα (6,02 έναντι 13,6 δευτερολέπτων για κάθε γενιά ) ενώ η αποτελεσματικότητα (κόστος 42250 έναντι 45850) είναι ανάλογη.

# Επίλογος

Το πρόβλημα της κατάρτισης ενός εξεταστικού ημερολογίου είναι κύρια ένα πρόβλημα χρονοπρογραμματισμού και έχει αρκετές ομοιότητες με ανάλογα προβλήματα που παρουσιάστηκαν στην αρχή της εργασίας. Οι τεχνικές επίλυσης προβλημάτων χρονοπρογραμματισμού πάλι, που οι πιο ενδεικτικές παρουσιάστηκαν στην αρχή της διατριβής, αλλάζουν παρακολουθώντας τις εξελίξεις κυρίως στον τεχνολογικό τομέα προσπαθώντας να παραμείνουν αποτελεσματικές διαχρονικά και να μην ξεπεραστούν.

Για την λύση του προβλήματος έγινε ιδιαίτερη μελέτη, καταγράφηκαν και ταξινομήθηκαν οι περιορισμοί σε ανελαστικούς και ελαστικούς με σκοπό να τύχουν ιδιαίτερης αντιμετώπισης από το λογισμικό που ετοιμάστηκε για να προσφέρει λύση στο πρόβλημα.

Η εφαρμογή του γενετικού αλγορίθμου για την επίλυση του προβλήματος κατάρτισης εξεταστικού ημερολογίου έγινε με σκοπό την εκμετάλλευση των πλεονεκτημάτων που έχει ο συγκεκριμένος αλγόριθμος, όπως είναι η διερεύνηση μεγάλου όγκου πιθανών και απίθανων λύσεων, η εύκολη χρήση του και η αξιοπιστία του.

Από την άλλη επειδή ο γενετικός αλγόριθμος απαιτεί μεγάλη υπολογιστική ισχύ, ιδιαίτερη μέριμνα δόθηκε ώστε το πρόγραμμα που δημιουργήθηκε και υλοποιεί τον γενετικό αλγόριθμο να είναι όσο το δυνατόν πιο γρήγορο. Αυτό έγινε με τη χρήση ενός απλού αλλά κατάλληλου για το πρόβλημα χρωμοσώματος αλλά και με την χρήση κατάλληλων προγραμματιστικών τεχνικών που μειώνουν τον χρόνο εκτέλεσης ιδιαίτερα στη διαδικασία αξιολόγησης των πιθανών λύσεων του προβλήματος.

Έγιναν πολλές δοκιμές στο πρόγραμμα με σκοπό να μελετηθεί η συμπεριφορά του ιδίως η ταχύτητα εκτέλεσής του αλλά και η αποτελεσματικότητά του. Το πρόγραμμα δοκιμάστηκε σε περιβάλλον προσωπικού υπολογιστή, σε περιβάλλον Linux-Unix Server και στις υποδομές του Υπολογιστικού Πλέγματος μέσω του οργανισμού HellasGrid. Οι δοκιμές αφορούσαν παραμέτρους του γενετικού αλγορίθμου και πτυχές του ίδιου του προβλήματος ενώ καταγράφηκαν χρήσιμα συμπεράσματα.

Θέματα που μελλοντικά θα μπορούσαν να διερευνηθούν σε σχέση με τη παρούσα διατριβή είναι:

- Η χρήση ενός πιο απλού χρωμοσώματος, όπου τα μαθήματα θα εμφανίζονται μια φορά στα γονίδια ανεξάρτητα σε ποια πόλη εξετάζονται. Αυτό είναι πολύ εύκολο να υλοποιηθεί και θα μείωνε παράλληλα και τον χρόνο εκτέλεσης του προγράμματος.
- Δημιουργία ενός γραφικού περιβάλλοντος εργασίας, όπου ο χρήστης θα μπορούσε εύκολα να αλλάζει τις παραμέτρους που χρησιμοποιούνται από το πρόγραμμα και έχουν σχέση με το αλγόριθμο, για παράδειγμα τον πληθυσμό που χρησιμοποιείται, το πλήθος των γενεών, την πιθανότητα μετάλλαξης κ.τ.λ. Επίσης θετική επίδραση στην αποτελεσματικότητα του προγράμματος θα είχε εάν μπορούσαν κάποιες από τις παραμέτρους να αλλάζουν κατά τη διάρκεια εκτέλεσης του προγράμματος, για παράδειγμα η πιθανότητα μετάλλαξης για τις πρώτες 1000 γενεές να ήταν 3% και μετά ανά χίλιες γενεές να αυξάνει κατά ένα ποσοστό.
- Η υλοποίηση του γενετικού αλγορίθμου χρησιμοποιώντας τη μέθοδο των νησιών, όπως αυτή περιγράφεται στην ενότητα 5.1.2, σε περιβάλλον παράλληλης επεξεργασίας ή στις υποδομές του Υπολογιστικού Πλέγματος του HellasGrid.
- Η ενσωμάτωση κι άλλων περιορισμών στο πρόβλημα, όπως χρήση επιτηρητών, η εξέταση κάποιων μαθημάτων σε ειδικές αίθουσες ή εργαστήρια, η εξάρτηση κάποιων μαθημάτων από κάποια άλλα, η εξέταση ορισμένων μαθημάτων σε συγκεκριμένες εξεταστικές ημέρες. Θα ήταν επίσης πολύ ωφέλιμο εάν οι περιορισμοί μπορούσαν να ενεργοποιηθούν ή να απενεργοποιηθούν εύκολα πριν από την εκτέλεση του προγράμματος.





## Βιβλιογραφία

- [01] Roman Barták, “Constraint Programming: In Pursuit of the Holy Grail”, Charles University, Faculty of Mathematics and Physics, Department of Theoretical Computer Science
- [02] T Birbas, S Daskalaki and E Housos “Timetabling for Greek High Schools”, Journal of the Operational Research Society (1997) 48, p1191-1200, Jstor.
- [03] Alberto Caprara · Michele Monaci · Paolo Toth, “Models and algorithms for a Staff Scheduling Problem”, Math. Program., Ser. B 98: p 445–476, Springer-Verlag 2003
- [04] Alberto Caprara, Michele Monaci, Paolo Toth, Pier Luigi Guida, “A Lagrangian heuristic Algorithm for a Real-World Train Timetabling Problem”, Discrete Applied Mathematics 154 (2006) p 738 – 753, Elsevier
- [05] M Carey and S Carville, “Testing schedule performance and Reliability for Train Stations”, journal of the Operational Research Society (2000) Si, p 666-682, Jstor
- [06] Malachy Carey and David Lockwood, “A Model, Algorithms and Strategy for Train Pathing”, Journal of the Operational Research Society( 1995)46, p 988-1005
- [07] P. D. D. Dominic Z S. Kaliyamoorthy M. Saravana Kumar, “Efficient Dispatching Rules for Dynamic Job Shop Scheduling”, p 70-75, Springer-Verlag 2004
- [08] A.T. Ernst, H. Jiang, M. Krishnamoorthy, D. Sier, “Staff Scheduling and Rostering: A Review of Applications, Methods and Models”, European Journal of Operational Research 153 (2004) p 3-27, Elsevier.
- [09] European Grid Infrastructure, the Enabling Grids for E-Science in Europe, from Wikipedia [http://en.wikipedia.org/wiki/European\\_Grid\\_Infrastructure](http://en.wikipedia.org/wiki/European_Grid_Infrastructure)
- [10] European Grid Infrastructure, <http://www.egi.eu>

- [11] F. Gagliardi, B. Jones, F. Grey, M.-E. Bigin AND M. Heikkurinen, "Building an infrastructure for scientific Grid computing: status and goals of the EGEE project", *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, Vol. 363, No. 1833, Scientific Grid Computing (Aug. 15, 2005), pp. 1729-1742
- [12] gLite - Lightweight Middleware for Grid Computing, <http://glite.cern.ch/>
- [13] GridCafé, the place to explore grid computing in a simple and stimulating way, <http://www.gridcafe.org/>
- [14] L. Hatzikonstantis and C. B. Besant, "Job-Shop Scheduling Using Certain Heuristic Search Algorithms", p 251-261, *Advanced Manufacturing Technology*, 1992
- [15] HellasGrid CA, A member of the International Grid Trust Federation (**IGTF**) <http://www.hellasgrid.gr/>
- [16] E.L. Lawler, "A Dynamic Programming Algorithm for Pre-Emptive Scheduling of a Single Machine to Minimize the Number of Late Jobs", *Annals of Operations Research* 26(1990), p125-133
- [17] Sean Luke, "The ECJ Owner's Manual", Department of Computer Science, George Mason University, 2013
- [18] Kimmo Nurmi, Dries Goossens, Thomas Bartsch, Flavia Bonomo, Dirk Briskorn, Guillermo Duran, Jari Kyngäs, Javier Marenco, Celso C. Ribeiro, Frits Spieksma, Sebastián Urrutia and Rodrigo Wolf-Yadlin – "A Framework for Scheduling Professional Sports Leagues",
- [19] K. Papoutsis, C Valouxis and E. Housos "A Column Generation Approach for the Timetabling Problem of Greek High Schools", *Journal of the Operational Research Society* (2003) 54, p230-238, Jstor.
- [20] Kingsley Purdam , Mark Elliot , Stephen Pickles & Duncan Smith, "GRID Computing and Disclosure Control", p161-176, 2007
- [21] Celso C. Ribeiro, "Sports Scheduling: Problems and Applications", p 1-27, Department of Computer Science, Universidade Federal Fluminense, Brazil

- [22] Stuart Russell and Peter Norvig , “Artificial Intelligence: A Modern Approach” Third edition, Prentice Hall Series in Artificial Intelligence, 2010, 4th chapter
- [23] Michael S. Waterman, Thomas H. Byers, - “A Dynamic Programming Algorithm to Find All Solutions in a Neighbourhood of the Optimum” p. 179-188
- [24] Thomas Weise, Global Optimization Algorithms – Theory and Application –2009, <http://www.it-weise.de/>
- [25] Χ. Γκόγκος, Διδακτορική Διατριβή, Πανεπιστημίου Πατρών, “Αλγόριθμοι Συνδυαστικής Βελτιστοποίησης με Έμφαση σε Μεταερευνητικές Τεχνικές”, 2009
- [26] Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας, Η. Σακελλαρίου, “Τεχνητή Νοημοσύνη”, Β΄ Έκδοση, 2005, διαφάνειες βιβλίου: [http://aibook.csd.auth.gr/slides\\_toc.html](http://aibook.csd.auth.gr/slides_toc.html)
- [27] Σπυρίδων Λυκοθανάσης, “Γενετικοί Αλγόριθμοι και Εφαρμογές”, Ελληνικό Ανοικτό Πανεπιστήμιο, Πάτρα 2001
- [28] Γεώργιος Παπαϊωάννου, “Ανάπτυξη Εξεταστικού Ημερολογίου”, Μεταπτυχιακή Διατριβή στα Πληροφορικά Συστήματα, Ανοικτό Πανεπιστήμιο Κύπρου, 2012
- [29] Ιωάννης Τσούλος, “Γενετικοί Αλγόριθμοι”, <http://users.cs.uoi.gr/~itsoulos/genetic.php>

# Παράρτημα Α

## Επισυναπτόμενο Λογισμικό

### A.1 Επισυναπτόμενος Κώδικας του Λογισμικού

Η παρούσα Μεταπτυχιακή Εργασία συνοδεύεται από το λογισμικό που δημιουργήθηκε για την αυτόματη δημιουργία του προγράμματος εξετάσεων για ένα ίδρυμα τριτοβάθμιας εκπαίδευσης. Το λογισμικό περιέχεται στο αρχείο **program.zip** μαζί με τις αναλυτικές οδηγίες για την εκτέλεσή του οι οποίες περιέχονται σε αρχείο με όνομα README.txt.

Οι οδηγίες παρουσιάζονται και παρακάτω.

### A.2 Οδηγίες για την Εκτέλεση του Λογισμικού

#### Για Εκτέλεση σε Περιβάλλον Προσωπικού Υπολογιστή:

Αφού αποσυμπιεστεί το αρχείο program.zip, εμφανίζονται δύο φάκελοι. Ο φάκελος esj περιέχει τους υποφακέλους Exams και es. Ο φάκελος Exams περιέχει τον κώδικα που υλοποιεί το πρόβλημα της κατάρτισης εξεταστικού ημερολογίου. Ο φάκελος Exams\files περιέχει τα αρχεία δεδομένων που είναι απαραίτητα για την εκτέλεση του προγράμματος. Ο φάκελος es περιέχει κώδικα που υλοποιεί τον γενετικό αλγόριθμο.

Ο φάκελος files περιέχει διάφορα δοκιμαστικά αρχεία εισόδου, που μπορούν να χρησιμοποιηθούν όλα, αρκεί να συμφωνούν τα περιεχόμενα των αρχείων και γίνουν προηγουμένως οι κατάλληλες αλλαγές στον πηγαίο κώδικα στο αρχείο

«ExaminationsTimetable.java». Αν γίνουν αλλαγές στον πηγαίο κώδικα θα χρειασθεί ξανά μεταγλώττιση.

Τα αρχεία με τα δεδομένα εισόδου είναι 4 τον αριθμό και έχουν γενικό όνομα: «StudentLessons», «Examiners», «Timeslots» και «Rooms», όλα με κατάληξη txt. Όταν επιλέγονται άλλα δοκιμαστικά αρχεία θα πρέπει να συμφωνούν τα στοιχεία που περιέχουν, για παράδειγμα το αρχείο με τους εξεταστές να έχει έναν εξεταστή για κάθε μάθημα που εξετάζεται σε μια πόλη. Όταν υπάρχουν λάθη ή ελλείψεις στα δεδομένα εισόδου ΠΟΛΥ ΠΙΘΑΝΟΝ να μην δουλέψει σωστά το πρόγραμμα.

Για να εκτελέσουμε το λογισμικό στο pc, σε περιβάλλον windows, αντιγράφουμε το φάκελο ecj στο σκληρό δίσκο και για να είναι πιο εύκολο μετά όταν εκτελούμε το πρόγραμμα, κατευθίαν στο C:.

Είναι πολύ πιθανό να χρειασθεί να εκτελεσθεί η παρακάτω εντολή, αφού ανοίξουμε την Γραμμή Εντολών, εκτελούμε την εντολή:

```
set CLASSPATH=c:\ecj
```

Μεταφερόμαστε στον κατάλογο c:\ecj και εκτελούμε το πρόγραμμα με την εντολή:

```
java Exams.ExaminationsTimetable -file Exams\sum.params
```

Σε περιβάλλον linux, αφού αντιγράψουμε τον κατάλογο ecj στη ρίζα (για ευκολία και μόνο) εκτελούμε την εντολή: `set CLASSPATH=/ecj` και μετά για την εκτέλεση: `java Exams.ExaminationsTimetable -file Exams/sum.params`

Εάν χρησιμοποιούμε το Cygwin Terminal πάνω σε windows αντιγράφουμε τον κατάλογο ecj όπως παραπάνω αλλά δίνουμε την εντολή: `set CLASSPATH=c:/ecj` και μετά `java Exams.ExaminationsTimetable -file Exams/sum.params`

Το αρχείο sum.params περιέχει διάφορες παραμέτρους απαραίτητες για την εκτέλεση του προγράμματος. Αν θέλουμε να αλλάξουμε το πλήθος των γενεών που θέλουμε να εκτελέσουμε στο πρόγραμμα, ανοίγουμε το αρχείο sum.params μ' έναν επεξεργαστή κειμένου και δίνουμε στην παράμετρο:

generations, το πλήθος των γενεών που θέλουμε έστω π.χ. 100 όπως παρακάτω:

```
generations = 100
```

### **Για Εκτέλεση σε Περιβάλλον Grid:**

Ο φάκελος grid περιέχει το λογισμικό και τα αρχεία δεδομένων ώστε να εκτελεσθεί το πρόγραμμα σε υπολογιστικό περιβάλλον Grid. Τα περιεχόμενα του καταλόγου είναι αναλυτικά τα εξής:

Αρχεία με τα δεδομένα εισόδου:

ExaminersGP.txt

RoomsGP.txt

StudentLessonsGP.txt

TimeslotsGP.txt

Αρχεία με κατάληξη .params είναι αρχεία παραμέτρων.

Το αρχείο με αποτελέσματα από την τελευταία εκτέλεση, out.stat.

Το αρχείο grid.jdl, είναι το jdl αρχείο, χρησιμοποιείται για την υποβολή του προγράμματος στο grid. Το αρχείο script.sh είναι το script που θα εκτελεσθεί στον απομακρυσμένο server.

Τέλος το αρχείο gridExams.jar είναι το jar αρχείο που περιέχει όλες τις κλάσεις που είναι απαραίτητες για την υλοποίηση του προγράμματος και την παραγωγή εξεταστικού προγράμματος.

Για οδηγίες σχετικά με το πώς υποβάλλεται προς εκτέλεση[12] μια εργασία σε περιβάλλον Grid παρέχονται στο σύνδεσμο:

<http://goc.grid.auth.gr/wiki/bin/view/Groups/ALL/SubmitYourJobsToTheGrid>