Τεχνητή νοημοσύνη

Η εργασία μας αφορά την ανάπτυξη ενός γενετικού αλγορίθμου, ο οποίος χρησιμοποιείται για την εύρεση του ιδανικού προγράμματος ενός γυμνασίου. Λαμβάνει ως είσοδο 2 json αρχεία τα οποία περιέχουν τα δεδομένα των μαθημάτων, πόσες ώρες πρέπει να εμφανίζονται, και των καθηγητών, πόσες ώρες έχουν περιθώριο να διδάξουν και για ποια μαθήματα.

Τρόπος χρήσης

Μέσα στο **chromosome.h** μπορούν να οριστούν ελεύθερα οι μέγιστες ώρες, μέρες και τα τμήματα που θα έχει το κάθε έτος, τα οποία είναι κοινά μεταξύ τους. Στην αρχή του main.cpp ορίζονται το **POPULATION_SIZE**, ο επιτρεπόμενος αριθμός ***MAX_GEN*** που θα τρέξει ο αλγόριθμος. Ορίζονται επιπλέον τα paths των δεδομένων εισόδου στις μεταβλητές **LESSON_DATA_PATH** και **TEACHER_DATA_PATH**. Μόλις τελειώσει ο αλγόριθμος, είτε επειδή βρήκε το τέλειο πρόγραμμα, είτε επειδή ξεπεράστηκε το **MAX_GEN**, θα δημιουργηθεί ένα *json* file που θα περιέχει το πρόγραμμα του σχολείου.

Δυνατότητες και αρχιτεκτονική

Ο αλγόριθμος μας υπολογίζει το **score** (ή αλλιώς *fitness*) με την χρήση 8 συναρτήσεων, όπου η κάθε μία ελέγχει έναν διαφορετικό περιορισμό:

1. Κάθε μάθημα εμφανίζεται τον απαιτούμενο αριθμό φορών σε κάθε έτος, σε κάθε τμήμα
calculateSatisfyLessonHoursScore(chrom, lessons);

2. Κάθε καθηγητής εργάζεται σύμφωνα με των διαθέσιμων καθημερινών και εβδομαδιαίων ωρών
calculateDailyWeeklyLimitScore(chrom, teachers);

3. Κάθε καθηγητής βρίσκεται σε ένα τμήμα σε κάθε στιγμή στο χρόνο
calculateTeacherConflictScore(chrom);

4. Δεν υπάρχουν κενά ανάμεσα σε μαθήματα μιας σχολικής μέρας
calculateNoFreePeriodsScore(chrom);

5. Κάθε καθηγητής εργάζεται μέχρι 2 συνεχόμενες ώρες
calculateConsecutiveHoursScore(chrom);

6. Οι ώρες ενός τμήματος πρέπει να είναι ομοιόμορφα διατεταγμένες μέσα στην βδομάδα
calculateAverageUniformityScore(chrom);

7. Τα μαθήματα μιας εβδομάδας πρέπει να είναι ομοιόμορφα διατεταγμένα μέσα στην βδομάδα
calculateAllLessonHourSpreadScore(chrom, lessons);

8. Οι καθηγητές εργάζονται παρόμοιες ώρες μέσα στην βδομάδα (30% ελαστικότητα)

teachSimilarHoursPerWeek(chrom, teachers);

Κάθε μία από αυτές τις μεθόδους αξιοποιεί ένα κλάσμα, το οποίο δηλώνει πόσο κοντά βρίσκεται σε μία αποδεκτή λύση. Το καθένα από αυτά πολλαπλασιάζεται με το 1000, για να έχουμε ένα αποδεκτό εύρος από ακεραίους.

Κάθε *χρωμόσωμα* αποτελείται από **nClassesPerGrade * nGrades * nDaysPerWeek * nHoursPerDay** κελιά, όπου κάθε κελί αντιστοιχεί σε μία μέρα για ένα τμήμα, και περιέχει ένα pair μαθήματος και καθηγητή. Από έναν τυχαιοποιημένο αριθμό, επιλέγεται ένα τυχαίο **id** *μάθημα*, και από το επιλεγμένο *μάθημα*, έναν **id** από τους διαθέσιμους *καθηγητές* τους.

Για την διασταύρωση, επιλέγονται 2 τυχαία *χρωμοσώματα*, όσο πιο υψηλό **score** έχουν, τόσο πιο πιθανό είναι να επιλεχθούν. Χωρίζονται τυχαία σε 2 κομμάτια και δημιουργούνται 2 καινούργια *χρωμοσώματα*, όπου το καθένα έχει ένα κομμάτι από κάθε γονιό.

Για την μετάλλαξη, το κάθε *γονίδιο* ενός έχει μία πιθανότητα ίση με το **MUTATION_PROB**, ώστε να επιλεχθεί ένα τυχαίο *μάθημα*, και έναν τυχαίο από τους διαθέσιμους *καθηγητές* για το επιλεγμένο *μάθημα*.

Γίνεται χρήση 64 bit random number generation για την μεγιστοποίηση της τυχαιοποίησης των δεδομένων και της λειτουργίας.

Μέθοδοι

Πειραματιστήκαμε με διάφορες μεθόδους για την ρύθμιση του ρυθμού μετάλλαξης και της συνάρτησης scoring.

Mutation Rate

- Σταθερός ρυθμός
- Μείωση με προκαθορισμένα βήματα
- Γραμμική μείωση
- Σταθερός ρυθμος, με εφαρμογή μόνο σε χρωμοσώματα με fitness κάτω από τον μέσο όρο
- Mutation με βάση το fitness, αύξηση mutation όταν δεν βελτιώνεται ο μέσος όρος του fitness για αρκετά generations

Scoring

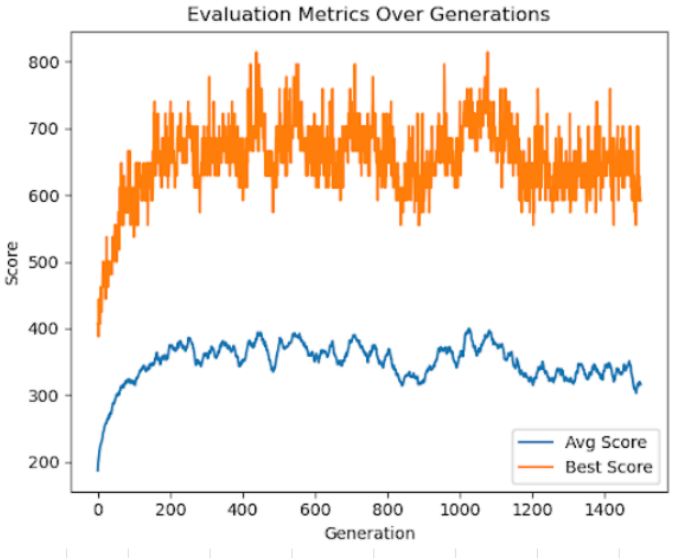
- Γραμμικές
- Εκθετικές, όπως 2, 3, 4, 6

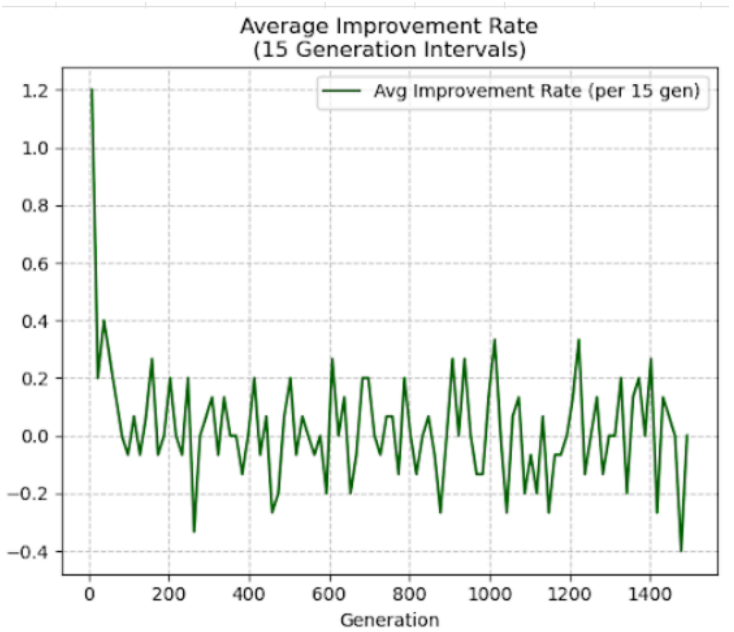
Πειράματα

A) Πειράματα με μοναδικό fitness function: satisfy Lesson Hours

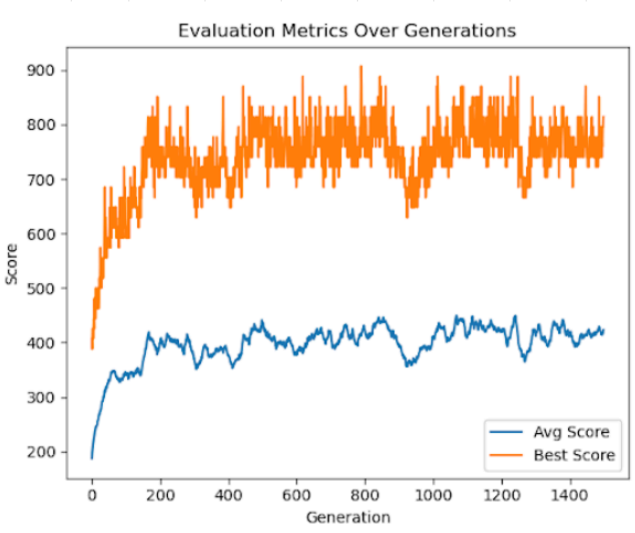
** Στο simpleLessons.json και simpleTeachers.json dataset*

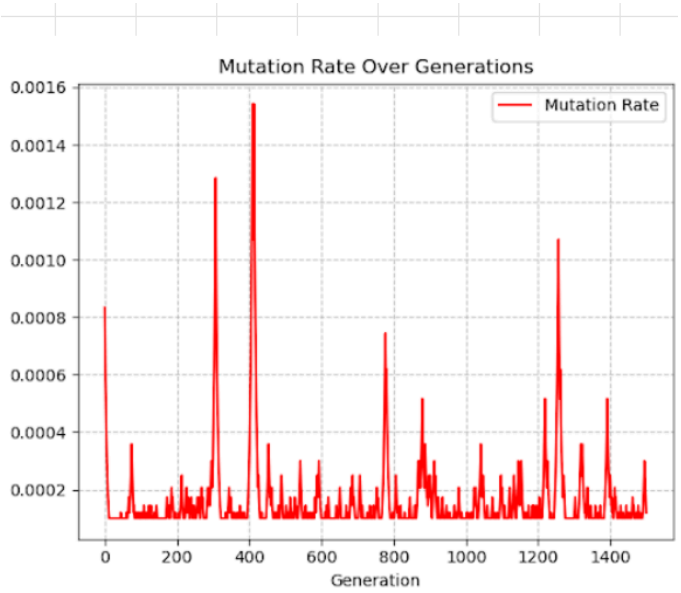
A1. Σταθερό Mutation Rate
Best Fitness Achieved: 81.4%
Population: 2.000
Generations: 1.500
Σταθερό Mutation Rate: 0.001
Seed: 2358873219

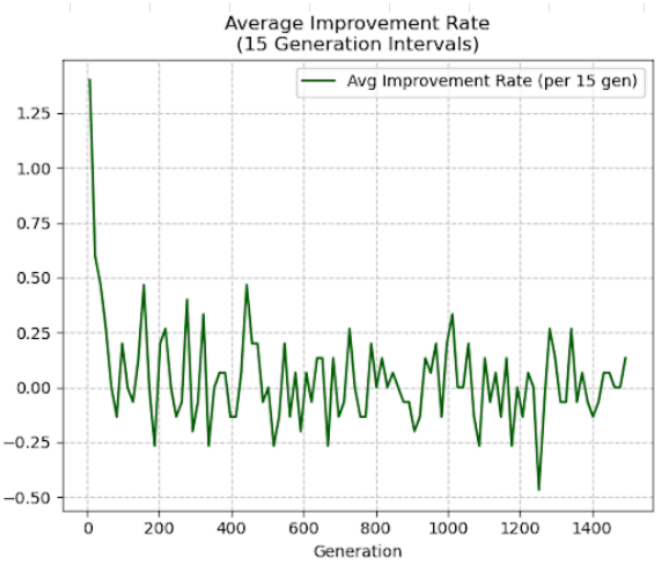




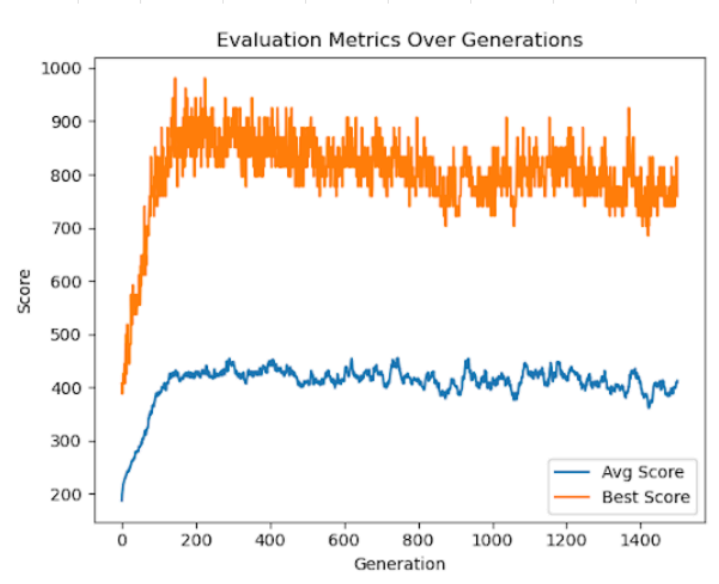
A2 Μεταβαλλόμενο Mutation Rate με βάση την βελτίωση του Fitness
Best Fitness Achieved: 90.7%
Population: 2.000
Generations: 1.500
Αρχικό Mutation Rate: 0.001
Mutation Rate Range: [0.001, 0.1]
Adjustment Factor: 1.2
Improvement Threshold: 0.2
Generation Threshold: 10
Seed: 2358873219

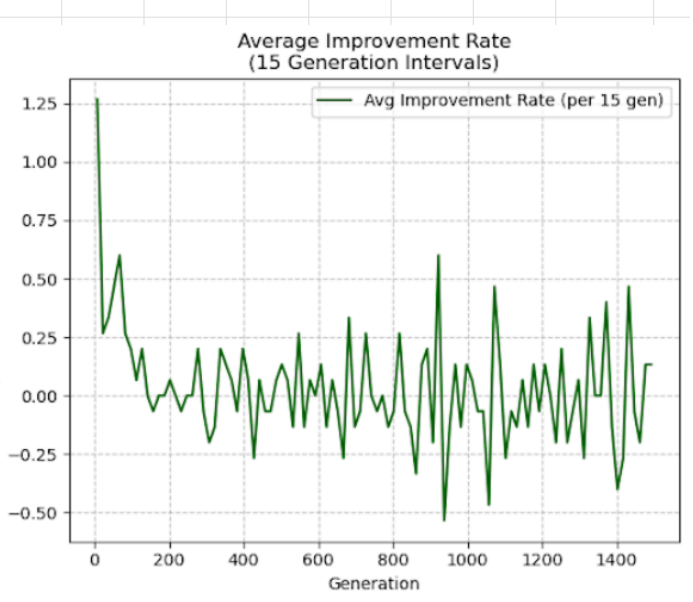




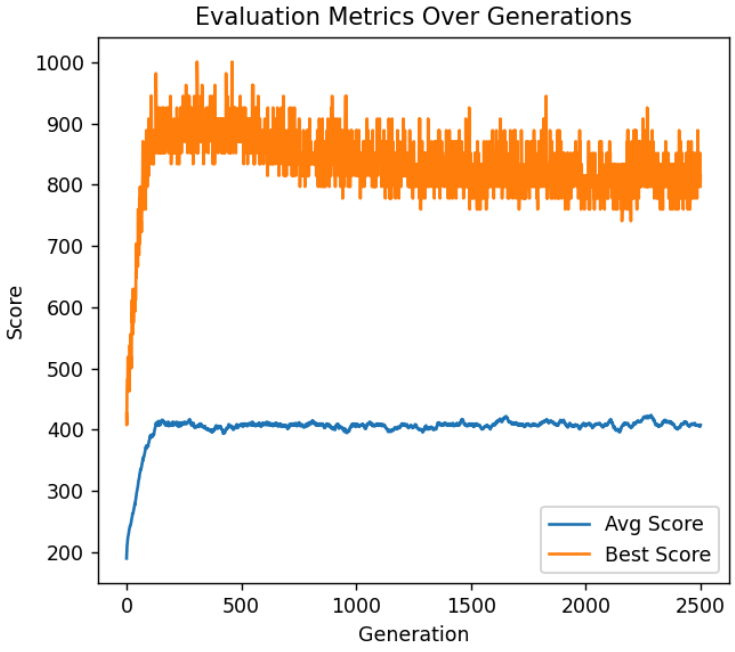


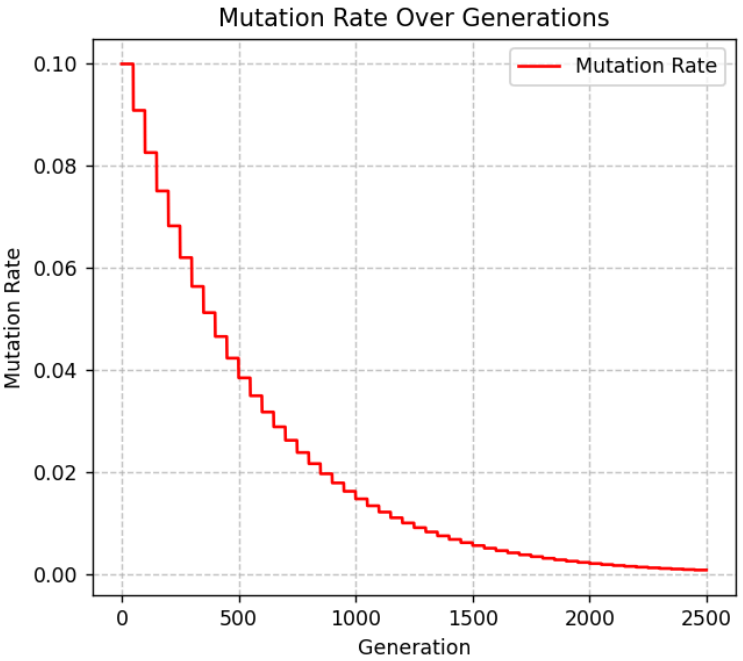
A3. Σταθερό Mutation Rate, εφαρμοσμένο μόνο σε χρωμοσώματα με fitness κάτω από τον μέσο όρο.
Best Fitness Achieved: 98.1%
Population: 2.000
Generations: 1.500
Σταθερό Mutation Rate: 0.1
Seed: 2358873219





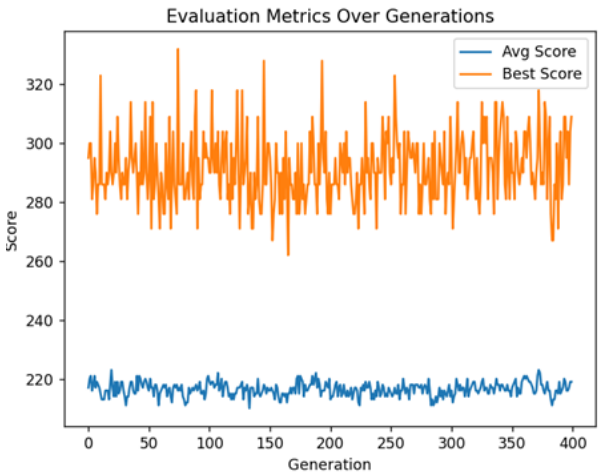
A4. Μεταβλητό mutation rate ανά 50 γενιές
Best Fitness Achieved: 100%
Population: 20.000
Generations: 2.500
Αρχικό Mutation Rate: 0.2
Mutation Decrease: 20%
Seed: 1426592087



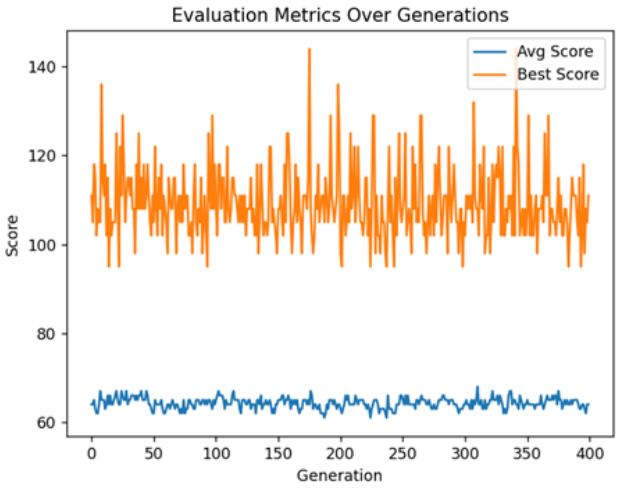


A5. Σύγκριση σταθερού mutation με γραμμικό, τετραγωνισμένο και κυβισμένο score
** Στο lessons.json και teachers.json dataset*
Population: 200
Generations:400
Mutation Rate: 0.2
Seed: 1
Όσο πιο υψηλή η δύναμη, τόσο πιο ισχυρή η αύξηση, όμως πέρα από την ^5 δύναμη, παρατηρούμε μικρές αυξήσεις ως προς το υπολογιστικό κόστος.

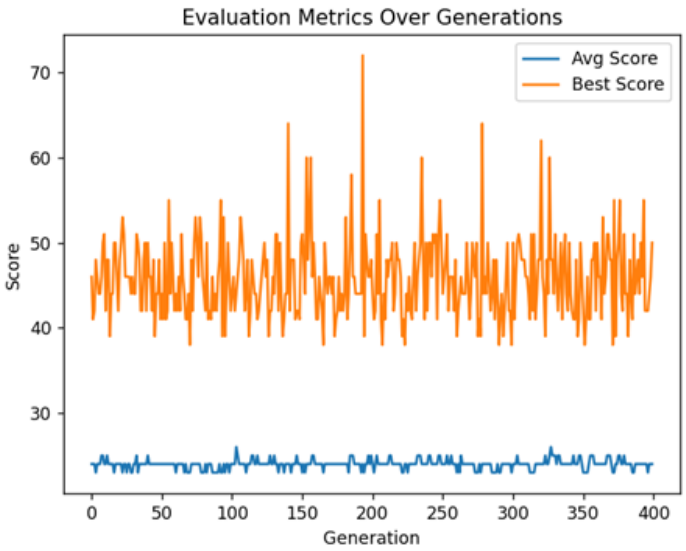
40%

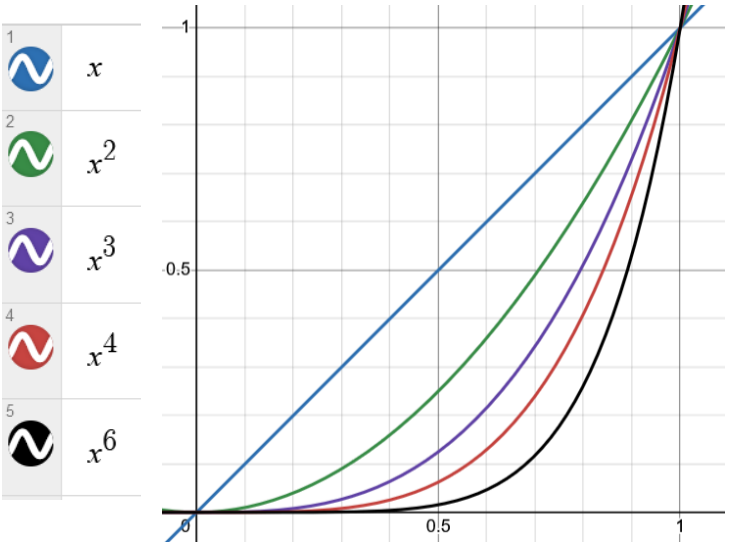


42%



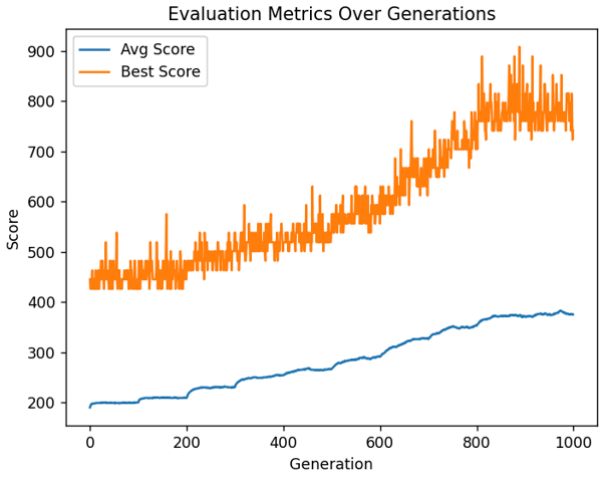
45%





* Επειδή τα αποτελέσματα των scoring functions κυμαίνονται εντός του [0,1], η αύξηση της δύναμης στην οποία υψώνουμε τιμωρεί πιο έντονα τις μικρές τιμές.

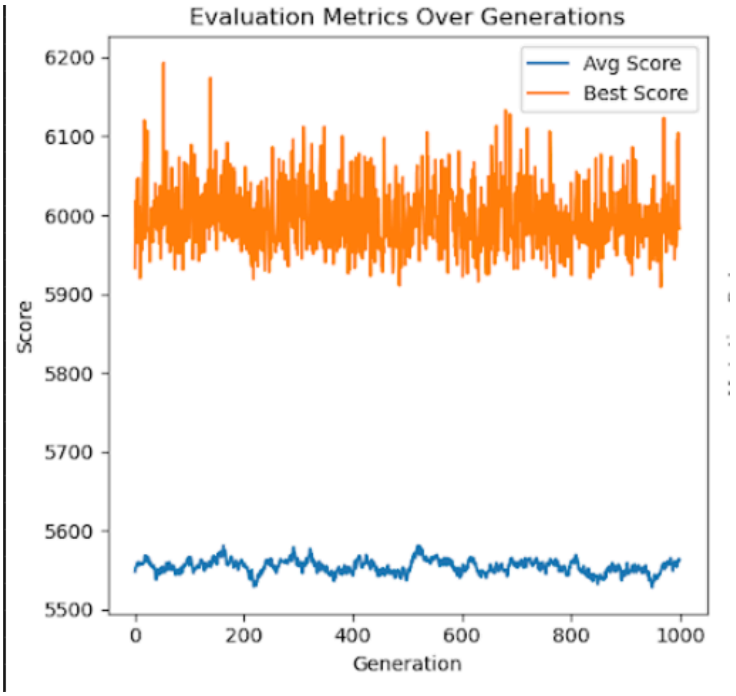
Α6. Μεταβαλλόμενο mutation rate με step
** Στο lessons.json και teachers.json dataset*
Population: 25000
Generations: 1000
Αρχικό Mutation Rate: 0.1
Seed: 413143562

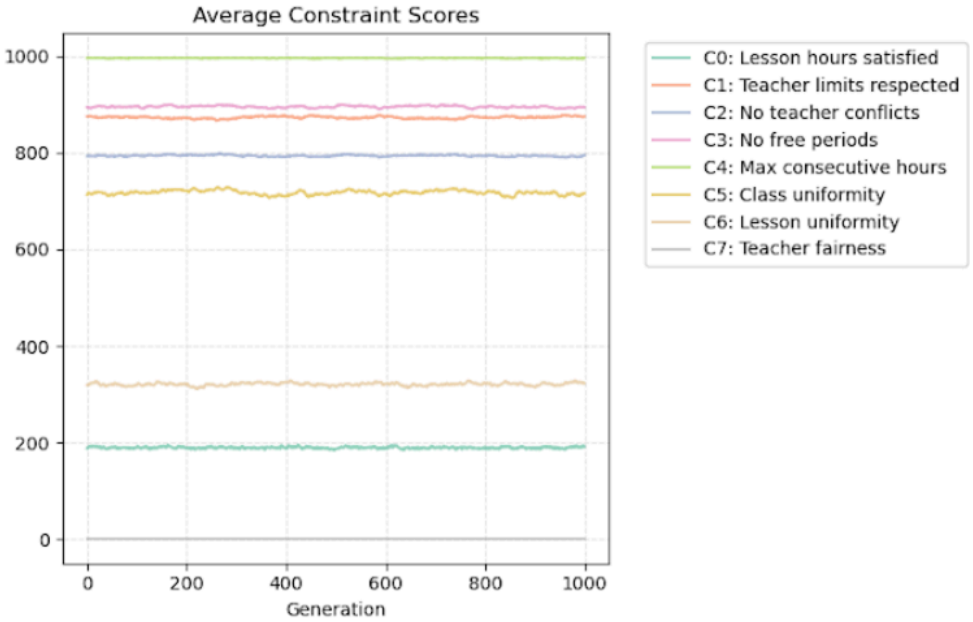



```
case 100:
    MUTATION_PROB = 0.05L;
    break;
case 200:
    MUTATION_PROB = 0.02L;
    break;
case 300:
    MUTATION_PROB = 0.01L;
    break;
case 400:
    MUTATION_PROB = 0.007L;
    break;
case 500:
    MUTATION_PROB = 0.004L;
    break;
case 600:
    MUTATION_PROB = 0.002L;
    break;
case 700:
    MUTATION_PROB = 0.001L;
    break;
case 800:
    MUTATION_PROB = 0.0005L;
    break;
```

Β. Πειράματα με όλα τα fitness functions
** Στο simpleTeachers.json και simpleLessons.json dataset*

Β1. Σταθερό mutation
Best Fitness Achieved: 77.5%
Population: 2.000
Generations: 1.000
Mutation Rate: 0.05





B2. Μεταβαλλόμενο mutation ανά 50 γενιές με γραμμικό score
Best Fitness Achieved: 70.4%
Population: 20.000
Generations: 5.000
Αρχικό Mutation Rate: 0.1
Minimum Mutation Rate: 0.0001
ADJUSTMENT_FACTOR: 1.2