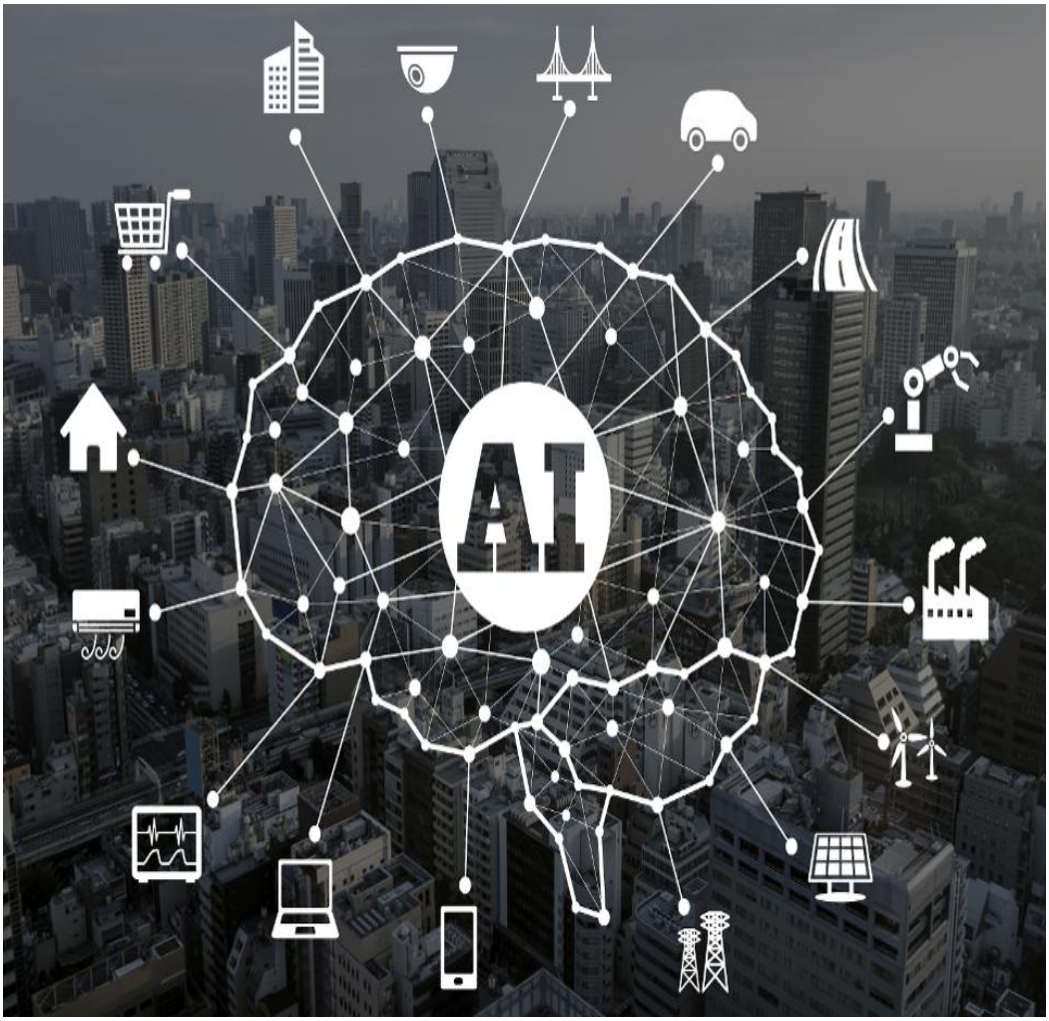

Τεχνητή Νοημοσύνη

PROJECT 1

Αντώνιος Κατωπόδης | 3140076
Αθανασία Μαρία Κουτσοπούλου | 3140092
Διονύσης Χασακής | 3140219

Αρχική Προσέγγιση

Αρχιτεκτονική και Μέθοδοι Τεχνητής Νοημοσύνης

Αρχίσαμε την διαδικασία σχεδιασμού του αλγορίθμου μέσω του A* αλγόριθμου. Η υλοποίηση ήταν σε Python. Αποφασίσαμε να αρχίσουμε με ένα κενό schedule, το οποίο το γεμίζαμε σε κάθε βήμα με ένα νέο μάθημα. Τα παιδιά πάντοτε θα περιείχαν ένα μάθημα παραπάνω από τον πατέρα τους.

Το κάθε State του αλγορίθμου μας ήταν ένα schedule για όλα τμήματα του Γυμνάσιου. Το schedule ήταν έναν δυσδιάστατος πίνακας με 5 γραμμές και 7 στήλες. Οι γραμμές αντιπροσωπεύαν τις μέρες και οι στήλες τις ώρες. Το κάθε κελί του πίνακα περιείχε με τη σειρά του ένα πίνακα μεγέθους ίσο με των αριθμών των τάξεων. Ο πίνακας αυτός περιείχε ένα teacher και ένα lesson, και ανάλογα την θέση του πίνακα αφορούσε διαφορετικό τμήμα. Δηλαδή, το πρώτο κελί του κάθε array στο κάθε κελί του 2d array αφορούσε το A1 κτλ.

Δημιουργούσαμε, λοιπόν, τα παιδία με τη σειρά των lessons. Σε κάθε βήμα μια ώρα μαθήματος τοποθετούταν σε κάθε πιθανή μέρα και ώρα για κάθε πιθανό τμήμα, επίσης κατά την εισαγωγή γινόταν ο κατάλληλος έλεγχος προκειμένου να τοποθετήσουμε αντίστοιχα και κάποιον δάσκαλο, που διδάσκει το μάθημα αυτό, αλλά δεν έχει ήδη τοποθετηθεί σε ίδιο timeslot (ώρα και μέρα).
Το state μας κρατούσε και μια μεταβλητή depth. Υποθέσαμε ότι τα hard constraint ήτανε να μην έχουν τα τμήματα κενό και να οι καθηγητές να μην διδάσκουν παραπάνω από 2 ώρες συνεχόμενα. Τα υπόλοιπά constraint ήταν soft. Πιο συγκεκριμένα αν κάποιο hard constraint παραβιάζονταν έστω και μία φορά θα επέστρεφε score 100 ενώ αντιθέτως κάθε παραβίαση soft constraint θα επέστρεφε score 1. Βάλαμε ως συνθήκη τερματισμού να επιλέγουμε:

1) *Στο schedule να έχουν εισχωρήσει όλα τα μαθήματα*
2) *Να το schedule να έχει score <100(αν παραβιάζεται έστω ένα hard constraint θα επιστρέψει 100)*

Ο αλγόριθμος ήταν σωστός και έκανε τα σωστά βήματα και όντως θα παρήγαγε ένα σωστό πρόγραμμα. Όμως ο χρόνος που χρειάζονταν για να μας δώσει ένα πρόγραμμα ο αλγόριθμός ήταν σχεδόν εκθετικός. Εξάλλου τα πρώτα παιδιά που θα παραχθούν θα είναι

315 = (days)*sizeof(hours)*sizeof(grades) με αρκετά παρόμοιο score και όσο κατεβαίνουμε θα έχουμε n-d(d = depth), με αποτέλεσμα να μη μπορεί στην αρχή να επιλέξει τόσο εύκολα ανάμεσα σε states.

Για αυτό το λόγο τελικά αποφασίσαμε να στραφούμε σε άλλη λύση. Έτσι δοκιμάσαμε μια εντελώς διαφορετική προσέγγισή…

Τελική Προσέγγιση

Αρχιτεκτονική και Μέθοδοι Τεχνητής Νοημοσύνης

Η υλοποίηση μας έχει ως βάση τον A* με αρκετές παραλλαγές. Το πρόγραμμά μας πλέον έχει ως κεντρική μονάδα τις τάξεις και τα τμήματά αυτών. Δηλαδή το state, είναι ένα array που περιέχει τα προγράμματα των τμημάτων κάθε τάξης. Το κάθε τμήμα έχει το δικό της πρόγραμμα και όλα τα προγράμματα των τμημάτων μαζί ως ομάδα, μας δίνουν ένα πλήρες schedule για το σχολείο. Το προγράμματα κάθε τμήματος είναι πίνακες 5 X 7 και τα κελία τους είναι ζεύγη που δείχνουν ποιος τι μάθημα διδάσκεται και από ποιον καθηγητή τη δεδομένη ώρα και μέρα.

Αντί να αρχίσουμε με κενό πίνακα, αρχίζουμε με έναν γεμάτο πίνακα στον οποίο έχουμε βάλει όλα τα μαθήματα. Τα μαθήματα εισάγονται τυχαία στον πίνακα.

Επιπλέον σε αυτή τη προσέγγιση hard constrain είναι:

1) Να μην υπάρχει ο καθηγητής την ίδια ώρα σε δύο διαφορετικά τμήματα, π.χ. δεν μπορεί να διδάσκει και στο A1 και στο B5 ταυτόχρονα.
2) Κανένας καθηγητής να μην έχει παραπάνω ώρες διδασκαλία την εβδομάδα από τις ώρες που ορίζονται στο αρχείο.
3) Κανένας καθηγητής να μην έχει παραπάνω ημερήσιες ώρες διδασκαλίας από αυτές που ορίζονται στο αρχείο.
4) Κανένα μάθημα να μην διδάσκεται από καθηγητή που δεν το διδάσκει.

Κατά την αξιολόγηση ενός schedule, οι hard constraint είναι υπεύθυνοι να γυρίσουν την απαραίτητη πληροφορία που σχετίζεται με την θέση όπου παραβιάστηκαν, προκειμένου να αντιμετωπιστούν οι παραβιάσεις.

Από την άλλη τα soft constraints είναι:

1) Να μην υπάρχουν κενά στα προγράμματα των τμημάτων.
2) Να μην διδάσκει κανένας καθηγητής παραπάνω από 2 ώρες συνεχόμενα.

3) Ο ημερήσιος αριθμός ωρών διδασκαλίας κάθε τμήματος να είναι κατά το δυνατόν ομοιόμορφος όλες τις ημέρες.
4) Οι ώρες διδασκαλίας κάθε μαθήματος σε ένα τμήμα να είναι κατά το δυνατόν ομοιόμορφα κατανεμημένες σε όλες τις ημέρες της εβδομάδας.
5) Ο αριθμός ωρών διδασκαλίας ανά εβδομάδα να είναι κατά το δυνατόν ομοιόμορφος για όλους τους καθηγητές.

Τα παιδία σε κάθε βήμα δημιουργούνται πλέον με swaps. Κάθε παιδί είναι πιστό αντίγραφο του πατέρα του, με μοναδική διαφορά δύο κελία του πίνακα έχουν πλέον γίνει swap. Κάθε φορά, έχουμε μια λίστα με τα κελιά όπου παραβιάζουν κάποιον hard constraint. Έστω τα κελία αυτά ότι λέγονται προβληματικά. Κάνουμε ένα swap μεταξύ ενός προβληματικού και κάθε ένα από τα υπόλοιπα στο συγκεκριμένο πρόγραμμα του τμήματος, όπου η κάθε αλλαγή παράγει και ένα νέο παιδί. Κάθε παραγόμενο παιδί το εισάγουμε στο open set του προβλήματός μας.

Η επιλογή του νέου πατέρα από το open set γίνεται επιλέγοντας αυτόν με το μικρότερο hard score, το οποίο hard score δίνετε από τα hard constraint αποκλειστικά. Αν δύο ή παραπάνω αντικείμενα του open set έχουν το ίδιο hard score, τότε θα επιλέξουμε οποιοδήποτε από αυτά τα παιδιά έχει το μικρότερο soft score. Το soft score το αποφασίζουν τα soft constraint.

Η κεντρική ιδέα είναι να βρούμε ένα αποδεκτό πρόγραμμα, που να ικανοποίει τα hard constraints και όσο τον δυνατόν γίνεται τα soft constraints. Ωστόσο, ένα πρόγραμμα που ικανοποιεί όλους τους hard constraints, αποτελεί αποδεκτή τελική λύση. Το μονοπάτι που θα πάρουμε για να φτάσουμε σε αυτό το πρόγραμμα θα επηρεάζεται από τα hard constraints κυρίως, αλλά και από τα soft constraints.

Τέλος, έχουμε υλοποιήσει και μία μέθοδο Unit Testing η οποία τρέχει δέκα φορές τον αλγόριθμο και κοιτάει αν με τις ίδιες εισόδους καταφέρνει να τερματίσει.

Δυνατότητές Προγράμματος

Το πρόγραμμα παρέχει την δυνατότητα με δεδομένα τα σωστά αρχεία teacher και lessons να επιστρέφει ένα υπαρκτό πρόγραμμα. Το πρόγραμμα γράφει στο ./data/schedule.txt το schedule κάθε τμήματος και επιστρέφει στο command line το χρόνο και τα hard και soft scores που πήρε το ολοκληρωμένο state του προγράμματος.

Τρόπος Χρήσης Προγράμματος

Αρχικά πρέπει να ορίσουμε τα 2 .txt αρχεία με τα δεδομένα του προγράμματος στο ./data/ directory. Τα 2 αυτά αρχεία είναι το teachers.txt όπου στην πρώτη γραμμή έχουμε τη συμβολοσειρά «Id Name MaxHoursPerDay MaxHoursPerWeek» και σε κάθε νέα γραμμή εισάγουμε τα αντίστοιχα properties για έναν καθηγητή, χωρισμένο το κάθε ένα με κενό. Σε περίπτωση που το όνομα είναι πάνω από μια λέξη, θα πρέπει να συνδεθεί με έναν ειδικό χαρακτήρα πχ. «Νεοελληνική_Γλώσσα». Επιπλέον, το δεύτερο αρχείο που θα πρέπει να προστεθεί είναι το lessons.txt όπου, όπως και στο προηγούμενο αρχείο, η πρώτη γραμμή θα είναι η συμβολοσειρά «Id Name HourPerClass Teachers» και σε κάθε νέα γραμμή πρέπει να εισαχθούν οι πληροφορίες για ένα μάθημα, διαχωρισμένα με κενά. Το HourPerClass θα πρέπει να έχει 3 τιμές διαχωρισμένες με κόμμα όπου η κάθε μία θα αντιπροσωπεύει τις ώρες που θα πρέπει να διδαχθεί το μάθημα στην x τάξη. Π.χ. Αν βάλουμε 3,2,2 αυτό σημαίνει πως στην Α τάξη θα πρέπει να διδαχθεί 3 ώρες, στην Β 2 ώρες κ.ο.κ. Με τον ίδιο τρόπο ανατίθενται και οι καθηγητές που θα διδάξουν το μάθημα αυτό βάζοντας τα id τους διαχωρισμένα πάλι με κόμμα. Έπειτα τρέχουμε τη main method του προγράμματος.

Παραδείγματα Χρήσης Προγράμματος

Με δεδομένα τα ήδη υπάρχοντα αρχεία lessons.txt, teachers.txt στο ./data/ directory του .zip αρχείου, μετά το τρέξιμο του αλγορίθμου έχουμε το ακόλουθο αποτέλεσμα και το νέο αρχείο schedule.txt στο ίδιο directory με τα άλλα αρχεία εισόδων.

Ένα παράδειγμα του αρχείου εξόδου είναι το εξής, όπου αναφέρεται το πρόγραμμα για το τμήμα A1:

```
A1:
   Day: 0 | Ιστορία , Αργυρός | Νεοελληνική_Γλώσσα , Κατωπόδης | Αγγλικά , Πανταζής | Μαθηματικά , Μέντη | Μαθηματικά , Μέντη | Γεωγραφία , Φινέ | Ιστορία , Αργυρός
   Day: 1 | Αγγλικά , Πανταζής | Αγγλικά , Πανταζής | Χημεία , Ιαλινά | Χημεία , Ιαλινά | Νεοελληνική_Γλώσσα , Κατωπόδης | Νεοελληνική_Γλώσσα , Κατωπόδης | Γυμναστική , Παπαθανασίου
   Day: 2 | Φυσική , Παππάς | Νεοελληνική_Γλώσσα , Κατωπόδης | Νεοελληνική_Γλώσσα , Κατωπόδης | Γυμναστική , Παπαθανασίου | Γυμναστική , Παπαθανασίου | Φυσική , Παππάς | Μαθηματικά , Μέντη
   Day: 3 | Χημεία , Ιαλινά | Νεοελληνική_Γλώσσα , Κατωπόδης | Ιστορία , Αργυρός | Γυμναστική , Παπαθανασίου | Νεοελληνική_Γλώσσα , Κατωπόδης | Μαθηματικά , Μέντη | Αγγλικά , Πανταζής
   Day: 4 | Φυσική , Παππάς | Ιστορία , Αργυρός | Νεοελληνική_Γλώσσα , Κατωπόδης | Φυσική , Παππάς | Ιστορία , Αργυρός | Μαθηματικά , Μέντη | Γεωγραφία , Φινέ
```

Το πρόγραμμά μας παράγει ένα ικανοποιητικό πρόγραμμα για τους hard constraint με score 146.