Τεχνητή Νοημοσύνη - Εργασία 1

Ξάνθος Ξανθόπουλος 3170125
Τζένη Μπολένα 3170117
Κωνσταντίνος Νικολούτσος 3170122

November 24, 2019

1 Περιγραφή προβλήματος

Αντικείμενο της πρώτης προγραμματιστικής άσκησης είναι η υλοποίηση ενός προγράμματος το οποίο για δεδομένα εισόδου, μια λίστα με μαθήματα και μία λίστα με καθηγητές, να παράγει το εβδομαδιαίο πρόγραμμα διδασκαλίας ενός γυμνασίου ικανοποιώντας όσο το δυνατόν καλύτερα έναν αριθμό περιορισμών.

2 Περιγραφή Προγράμματος

Για την επίλυση του συγκεκριμένου προβλήματος επιλέχθηκε μία παραλλαγή του αλγορίθμου Hill Climbing. Αρχικά δημιουργούμε ένα τυχαίο πρόγραμμα διδασκαλίας(state) το οποίο δεν ικανοποιεί εξ αρχής κανέναν από τους περιορισμούς του προβλήματος. Η μόνη κατάσταση που δεν επιτρέπεται είναι να έχουμε κάποιο μάθημα σε διαφορετική τάξη από τήν οποία ανήκει(πχ. Μαθηματικά Β' τάξης δεν μπορούν να υπάρχουν σε κάποιο τμήμα της Α' τάξης). Έπειτα ελέγχουμε το πρόγραμμα για κάθε έναν περιορισμό και εξάγουμε μια βαθμολογία. Κάθε φορά διαλέγουμε έναν περιορισμό και δημιουργούμε ένα γειτονικό state(που να διαφέρει κατά μία ανταλλαγή μαθημάτων ή κατα έναν καθηγητή σε ένα μάθημα). Αν το νέο state έχει καλύτερη βαθμολογία τότε αποδεχόμαστε το state αυτό και επαναλαμ΄βανουμε την διαδικασία έχοντας το ως δεδομένο. Σε αντίθετη περίπτωση το απορίπτουμε και διαλέγουμε εκ νέου έναν περιορισμό προς επίλυση.

2.1 Μορφή Δεδομένων Εισόδου

Για την αναπαράσταση των δεδομένων επιλέξαμε αρχεία της μορφής JSON για να είναι εύκολα αναγνώσιμα και επεξεργάσιμα από κάποιον άνθρωπο. Για κάθε

μάθημα έχουμε έναν μοναδικό κωδικό, το όνομα του μαθήματος, η τάξη στην οποία διδάσκεται καθώς και ο αριθμός των ωρών που πρέπει να διδαχθείστο κάθε τμήμα της τάξης. Για κάθε καθηγητή έχουμε έναν μοναδικό κωδικό, το όνομα του καθηγητή, τους κωδικούς των μαθημάτων που μπορεί να διδάξει, της μέγισες ώρες που μπορεί να κάνει μάθημα την ημέρα και τις μέγιστες ώρες που μπορεί να κάνει μάθημα την εβδομάδα.

Για τα δεδομένα μας έχουμε ορίσει συγκεκριμένουν περιορισμούς. Οι κωδικοί των μαθημάτων όσο και των καθηγητών θα πρέπει να ξεκινάνε από το μηδέν και μην παρουσιάζουν κενά στην αρίθμηση διοτί χρησημοποιούνται από το πρόγραμμα ως δείκτες. Επιπλέον οι κωδικοί των μαθημάτων που μπορεί να διδάξει ενας καθηγητής θα πρέπει να υπάρχουν. Τέλος οι μέγιστες ώρες που μπορεί να διδάξει κάθε καθηγητής θα πρέπει να είναι ένας θετικός μη μηδενικός αριθμός. Συγκερκιμένα ου ώρες που μπορεί να διδάξει εβδομαδιαία θα πρέπει να είναι λιγότερες ή ίσες με το γινόμενο των ωρών που μπορεί να διδάξει κάθε μέρα επί της μέρες διδασκαλίας. Αυτό ο τελαυταίος περιορισμός προκείπτει λογικά από το γεγοονός οτι δεν ύπαρχει τρόπος καποιος καθηγητής να δουλέψει περισσότερες ώρες από το γινόμενο αυτό.

2.2 Δημιουργία Αρχικού State

Για να δημιουργήσουμε το αρχικό μας state αφού διαβάσουμε τα αρχεία εισόδου δημιουργούμε ζεύγη μαθήμάτων με καθηγητές που μπορούν να τα διδάξουν. Το πλήθος των ζεύγων για κάθε μάθημα είναι ίσο με το πλήθος τω ωρών που πρέπει να διδαχθεί το μάθημα σε κάθε τμήμα επί τον αριθμό των τμημάτων. Έχοντας υποθέσει ότι το μέγιστο των ωρών που μπορεί να γίνει μάθημα καθημερινά είναι εφτά ώρες και κάθε ταξη ότι έχει τρία τμήματα, έχουμε έναν μέγιστο αριθμό από ζεύγη που ανέρχεται σε 105. Σε περίπτωση που δέν έχουμε τόσα ζεύγει αλλά έχουμε λιγότερα τότε συμπληρώνουμε με κενά ζεύγη που αντιστοιχούν σε κενές ώρες στο πρόγραμμα διδασκαλίας. Τέλος γεμίζουμε της ώρες κάθε τμήματος με τυχαία ζέυγη από αυτά που δημιουργήσαμε. Η ίδια διαδικασία επαναλμβάνεται για κάθε τάξη.

2.3 Υλοποίηση Περιορισμών

Κάθε περιορισμός που έπρεπε να επιλύει το πρόβλημα μοντελοποιήθηκε στην δικιά του κλάση επεκτείνωντας ο κάθε ένας όμως μία αφηρημένη κλάση που περιγράφει της απαραίτητες μεθόδους που πρέπει να έχει ένας περιορισμός. Εκτός από τις μεθόδους κάθε περιορισμός έχει μία λίστα με τα κελιά του state που τον παραβιάζουν καθλως και ένα βάρος που υποδικνύει την συμαντικότητα

του.

2.3.1 Η μέθοδος evaluate

Ο κάθε περιορισμός δεχόμενος σαν είσοδο το τρέχων state καθώς και γενικές πληροφορίες όπως την λίστα των μαθημάτων, την λίστα των καθηγητών και πληροφορίες που παράγωνται απο αυτές, εντοπίζει τα σημεία στα οποία το state τον παραβιάζει και τα αποθηκεύει σε μία λίστα. Κάθε μεμονομένη παραβίαση του περιορισμού αυτού μέσα στο state συνεισφέρει στο τελικό σκορ του.

2.3.2 Η μέθοδος resolve

Κάθε περιορισμός πρέπει να είναι σε θέση να αλλάξει το state με τρόπο τέτοιο ώστε να προσπαθεί να επιλύσει παραβιάσεις του ευατού του δημιουργώντας όσο των δυνατόν λιγότερες παραβιάσεις σε άλλους περιορισμούς. Σε κάθε εκτέλεση της μεθόδου αυτής επιλέγετε τυχαία ένα σημείο που προκαλεί πρόβλημα από την λίστα με τα προβληματικά σημεία, αλλάζει το state και αποθηκεύει όσες πληροφορίες χρειάζεται ώστε να αναιρέσει ότι αλλαγή μόλις έκανε.

2.3.3 Η μέθοδος undo

Η μέθοδος αυτή χρησημοποιώντας την πληροφορίες που αποθήκευσε η resolve επαναφέρει το state στην πρότερη του κατάσταση.

2.4 Αλγόριθμος Hill Climbing

Ο αλγόριθμος που χρησημοποιήθηκε είναι μια παραλλαγή του Hill Climbing. Αρχικά δημιουργούμε το αρχικό state καθώς και κάθε έναν περιορισμό και τους προσθέτουμε σε μια λίστα απο τον σημαντικότερο προς τον πιο ασύμαντο. Έπειτα έως ότου η βαθμολογία του state φτάσει κάτω από ένα όριο η ξεπεράσουμε έναν μέγιστο αριθμό βημάτων υπολογίζουμε την βαθμολογία του state σαν το άθροισμα του σκορ κάθε περιορισμού, διαλέγουμε τον πρώτο περιορισμό στην λίστα που παραβιάζεται και προσπαθούμε σταδιακά να τον επιλύσουμε. Οι αυστηροί περιορισμοί επιλύωνται με συγκεκριμένη σειρά η οποία διασφαλίζει μια αρκετά γρήγορη ταχύτητα σύγκλισης στην λύση καθώς με αυτή την σειρά ελαχιστοποιούνται η νέες παραβιάσεις ενός περιορισμού από την επίλυση ενός άλλου. Οι ασθενείς περιορισμοί ωστόσο σε κάθε βήμα ανακατεύονται ώστε να μειώνεται η πιθανότητα να κολλήσει ο αλγόριθμος σε κάποιο τοπικό μέγιστο κάποιου περιρισμού με υψηλότερη σημαντικότητα. Αν το νέο state έχει χαμηλότερη βαθμολογία από το τρέχων τότε γίνεται αυτό τρέχων και η διαδικασία

επαναλαμβάνεται.

Η διαφορά με τον τυπικό Hill Climbing είναι ότι τον επόμενο state δεν διαλέγεται από όλα τα γειτονικά state αλλά παρέχεται από την μέθοδο resolve του κάθε περιορισμού η οποία κατσκευάζει ένα state από ένα υποσύνολο του χώρου των λύσεων που θεωρούμε ότι έχει περισσότερες πιθανότητες να περιέχει μια λύση. Με αυτόν τον τρόπο βελτιώνεται η απόδοση του αλγορίθμου και οδειγεί γρηγορότερα σε λύση.

3 Περιορισμοί

Παρακάτω περιγράφεται αναλυτικά για κάθε περιορισμό ο υπολογισμός του σκορ του καθώς και η διαδικασία επίλυσης του. Οι πρώτοι τέσσερις περιορισμοί που παρουσιάζονται είναι αυστηροί και πρέπει να επιλύωνται πάντα εφόσον τα δεδομένα είναι σωστά.

3.1 Ο περιορισμός LessonMaxHours

Ο περιορισμός αυτός ελέγχει το αν διδάσκεται σε κάθε τμήμα ο σωστός αριθμός ωρών για το κάθε μάθημα. Αποτελεί τον σημαντικότερο περιορισμό και πρώτο προς επίλυση επειδή κανένας αλλός περιορισμός κατα την επίλυση του μετέπειτα δεν μπορεί να τον παραβιάσει ξανά. Για τον υπολογισμό του σκορ διαβάζει όλο το state μετράει τον αριθμό των ωρών κάθε μαθήματος σε κάθε τμήμα. Το σκόρ τέλος υπολογίζεται σαν το γινόμενο του βάρους του περιορισμού επί το άθροισμα των απολύτων τιμών των αποκλείσεων των ωρών διδασκαλίας κάθε μαθήματος σε κάθε τμήμα από τον απαιτούμενο αριθμό.

Για την επίλυση του διαλέγουμε από την λίστα με τα προβληματικά σημεία ένα μάθημα το οποίο εμφανίζεται παραπάνω φορές απο ώσες πρέπει. Έπειτα το αντιμετθέτουμε με ένα διαφορετικό μάθημα που διδάσκεται και αυτό παραπάνω ώρες από της απαιτούμενες αλλα σε διαφορετικό τμήμα της ίδιας τάξης ή αν δεν υπάρχει κάποιο μάθημα που να ανήκει σε αυτή την κατηγορία τότε πραγματοποιούμε την αντιμετάθεση με ένα κενό.

3.2 Ο περιορισμός ConcurrentLessons

Ο περιορισμός αυτός ελέγχει αν υπάρχει καθηγητής που να κάνει μάθημα σε δύο η περισσότερα διαφορετικά τμήματα μάθημα την ίδια ώρα και μέρα. Αποτελεί τον δεύτερο σημαντικότερο περιορισμό επειδή λόγο του οτί ο κάθε καθηγητής επιλέγεται τυχαία για το αν θα κάνει μάθημα η επίλυση αυτού λύνει ταυτόχρονα

και αλλούς περιορισμούς. Για τον υπολογισμό του σκορ διαβάζει όλο το state μετράει τον αριθμό των ωρών κάθε καθηγητή κάθε ώρα και κάθε μέρα. Το σκόρ τέλος υπολογίζεται σαν το γινόμενο του βάρους του περιορισμού επί το άθροισμα των ωρών διδασκαλίας κάθε καθηγητη που υπερβαίνει την μονάδα.

Για την επίλυση του διαλέγουμε από την λίστα με τα προβληματικά σημεία μία τυχαία ωρα και μερα και τμήμα όπου παρουσιάζεται πρόβλημα. Έπειτα βρήσκουμε το μάθημα που διδάσκετα εκείνη την ώρα και του καθηγητές που μπορούν να το κάνουν. Αν υπάρχει μόνο ένας καθηγητής για αυτό το μάθημα το αντιμεταθέτω με ενα αλλο μαθημα μια τυχαία ώρα και μέρα αλλά στο ίδιο τμήμα. Σε αντίθετη περίπτωση διαλέγω έναν τυχαίο καθηγητή υποψήφιο ως προς αλλαγή και κοιτάω αν κανει εκείνη την ώρα μάθημα σε κάποιο τμήμα. Αν δεν κάνει ανταλλάσω του καθηγητές αλλιώς αντιμεταθέτω και σε αυτή την περίπτωση το μάθημα σε τυχαία ώρα και μέρα.

3.3 Οι περιορισμοί TeacherMaxHoursWeekly και TeacherMaxHoursDaily

Οι περιορισμοί αυτοί λειτουργούν αρκετά παρόμοια ωστόσο η επιλυση τους με τη σωστή σειρά εξασφαλίζει ταχύτερη σύγκλιση κι απλοποιεί την επίλυση του δεύτερου. Και οι δύο για να υπολογίσουν το σκόρ του διαβάζουν όλο το state και μετράνε πόσες ώρες κανει μάθημα ο κάθε καθηγητής κάθε εβδομάδα και κάθε μέρα αντίστοιχα. Το σκόρ υπολογίζεται σαν το γινόμενο του βάρους του κάθε περιορισμού επί το άθροισμα των ωρών διδασκαλίας κάθε καθηγητη που υπερβαίνει τις μέγιστες ώρες του.

Για να επιλύσουμε τον εβδομαδιαίο περιορισμό διαλέγουμε έναν τυχαίο καθηγητή που υπερβαίνει το όριο και τον αντικαθιστούμε με κάποιον άλλο τυχαία που να μπορεί να διδάξει το μάθημα του πρώτου μία τυχαία μέρα και ώρα σε ένα τυχαίο τμήμα. Για την επίλυση τώρα του ημερήσιου περιορισμού εφόσον έχει λυθεί ο ενδομαδιαίος αρκεί απλά να αντιμεταθεσουμε μία ωρα που κάνει μάθημα ένας καθηγητής από μία μέρα που έχει θέμα σε μια άλλη τυχαία μέρα.

3.4 Οι περιορισμοί UniformLessons, UniformHours και UniformTeacherHours

Οι τρείς αυτοί ασθενείς περιορισμοί είναι αρκετά όμοιοι. Ο κάθε ένας τους υπολογίζει το κατα πόσο είναι ομοιόμορφα διαμειρασμένα τα μαθήματα, οι ώρες διδασκαλίας και οι ώρες που κάνουν μάθημα οι καθηγητές αντίστοιχα και υπολογίζουν το σκορ ανάλογα με τον πόσο απέχει κάθε ένα στοιχείο από τον

μέσο όρο. Παρόμοιος είναι και ο τρόπος επίλυσης. Κάθε φορά επιλέγετε ένα προβληματικό στοιχεία και αντιμετατίθεται σε μια τυχαία θέση. Για κάθε περιορισμό βεβαια ισχύουν διαφορετικοί περιορισμοί για το που μπορεί να βρισκεται αυτη η τυχαία θέση.

3.5 Ο περιορισμός EmptyHours

Ο περιορισμός αυτός είναι αρκετα απλός. Το σκόρ του ειναι το ιόμενο του βάρους επί το πόσο απέχουν τα κενά από την τελευταία μη κενή θέση και για να επιλυθεί απλά αντιμεταθέτουμε το κενο με την τελευταία ώρα που υπάρχει μάθημα.

3.6 Ο περιορισμός ConsucutiveHours

Τέλος ο περιορισμός αυτός ελέγχει το αν υπάρχουν καθηγητές που κάνουν τρείς η παράνω ώρες παραπάνω μάθημα. Για να επιλυθεί απλα διαλέγουμε μια από τις ώρες αυτές τυχαία και την αντιμεταθέτουμε σε μια τυχαια ώρα και μέρα.

4 Τρόπος Χρήσης Προγράμματος

Για να περάσουμε τα δεδομένα στο πρόγραμμα αρκεί να περάσουμε την διαδρομή των αρχείων σαν παραμέτρους κατά την εκτέλεση του προγράμματος καθώς επίσης το threshold και το πληθος των επαναλήψεων για τον αλγόριθμο. Για τα δεδομένα εισόδου θα πρέπει να ισχύει ότι αναφέρθηκε παραπάνω. Αν τα δεδομένα είναι σωστά δηλαδή υπάρχει επαρκής αριθμός καθηγητών το παραγόμενο πρόγραμμα πληρεί τους αυστηρους τουλάχιστον περιορισμούς. Για threshold ίσο με μηδέν το πρόγραμμα παράγει μία λύση που αποτελεί τοπικό μέγιστο μιάς περιοχής του χώρου λύσεων.

Σε περίπτωση που κάποιος θέλει να προσθέσει περιορισμούς αρκεί να δημιουργήσει μια νέα κλάση που να επεκτείνει την αφηρημένη κλάση Constraint και να την προσθέσει στην λίστα με τους περιορισμούς στον αλγόριθμο Hill Climbing.