

Μοντελοποίηση προβλήματος:

- Στη `main` διαβάζω τα στοιχεία από το csv αρχείο και τα αποθηκεύω σε 5 διαφορετικές παράλληλες λίστες.
- Για την μοντελοποίηση των slot έχω ακολουθήσει την εξής μορφή. Είναι ένα tuple 2 στοιχείων. Το 1ο στοιχείο είναι ένας int που δηλώνει τη μέρα που θα δοθεί το μάθημα και το 2ο είναι ένα string που δηλώνει το slot. (πχ) (1,'9-12') σημαίνει: 1η μέρα, από 9 έως 12.
- Variables: Οι μεταβλητές παίρνουν επακριβώς τους τίτλους των μαθημάτων, όπως ακριβώς είναι γραμμένοι στο csv.
- Domains: Όλα τα μαθήματα παίρνουν ως πιθανές τιμές όλα τα slots εκτός από τα μαθήματα που έχουν εργαστήριο, που δε μπορούν να μπουν στο διάστημα '3-6' μιας και δε θα υπάρχει χρόνος για να εξεταστεί τόσο η θεωρία όσο και το εργαστήριο.
- Neighbors: Ως neighbors σε κάθε μάθημα βάζουμε όλα τα υπόλοιπα μαθήματα μιας και υπάρχει ο περιορισμός ότι σε ένα συγκεκριμένο slot (μία συγκεκριμένη μέρα σε μια συγκεκριμένη ώρα) γίνεται να αναθέσουμε μόνο ένα μάθημα.
- Constraints: Στη συνάρτηση για τους περιορισμούς ελέγχω τα παρακάτω. Αν κάποιο από αυτά ισχύει τότε γυρνάω False.
 - Ίδια μέρα ή ίδια ώρα ή ίδιο μάθημα.
 - Μαθήματα ίδιου έτους την ίδια μέρα.
 - Ύπαρξη κενού, μικρότερου των 2 ημερών για δύσκολα μαθήματα.
 - Μαθήματα του ίδιου καθηγητή την ίδια μέρα.
 - Πρόβλημα με εργαστήριο.

dom/wdeg:

Για την υλοποίηση αυτή της ευρετικής συνάρτησης, έχω προσθέσει και αφαιρέσει κώδικα από το `csp.py` επομένως θα πρέπει να τρέξετε τα δικά μου αρχεία.

Ο τρόπος την υλοποίησα μοιάζει με τον `min` που είναι εξίσου ευρετική. Αυτό που γίνεται είναι ότι για κάθε γείτονα που δεν είναι στο assignment της κάθε μεταβλητής, βρίσκω το μέγεθος του domain, προσθέτω όλα τα βάρη για την συγκεκριμένη μεταβλητή και επιστρέφω την αναλογία του μήκους των domains (δηλ το πλήθος των πιθανών τιμών) δια το άθροισμα των βαρών (αν είναι διάφορο του 1).

ΣΗΜΕΙΩΣΕΙΣ:

- Για την υλοποίηση του `dom/wdeg` έχω γράψει στο αρχείο `csp.py`, το οποίο έχει υποστεί μια μικρή επεξεργασία (έχει τροποποιηθεί και η `__init__` του `csp`). Γι'αυτό το λόγο, ένα μεγάλο κομμάτι του `csp.py` που δε μας ενδιέφερε έχει αφαιρεθεί εντελώς. Για την σωστή λειτουργία και εξέταση του προγράμματός μου θα πρέπει να χρησιμοποιήσετε το δικό μου `csp.py`
- Το version της python που χρησιμοποίησα είναι python 3.8.10, οπότε για την εκτέλεση αρκεί να γράψετε: **`python3.8 timetabling.py`**
- Στο αρχείο `timetabling.py`, στο σημείο που ξεκινάει η 'main' υπάρχουν σε σχόλια διάφορες εντολές που αφορούν την εκτέλεση του προγράμματος και την τεχνική που ενδιαφέρεστε να εξετάσετε. Αφαιρέστε τα σχόλια που θέλετε και τρέξτε το πρόγραμμα.

ΕΡΩΤΗΜΑ 2:

	Time Elapsed	nassigns	Constraints Counter
MRV + MAC	1.8303978443145752	38	828613
MRV + FC	0.07281255722045898	38	51410
DOM/WDEG + MAC	1.794358253479004	38	833679
DOM/WDEG + FC	0.06195354461669922	38	52456
MIN CONFLICTS	0.06490612030029297	38	44141

Όπως βλέπουμε από τα πάνω αποτελέσματα, ο MAC δεν είναι ιδιαίτερα αποδοτικός τόσο στο χρονικό κομμάτι όσο και στα constraint checks. Από την άλλη ο FC φαίνεται να είναι πιο αποδοτικός από το MAC, ανεξάρτητα τον ευρετικό αλγόριθμο που χρησιμοποιείται. Για τους 2 παραπάνω αλγορίθμους, ο ευρετικός dom/wdeg φαίνεται να γλιτώνει λίγο χρόνο κατά την εκτέλεση αλλά γίνονται περισσότερα constraint checks, σε σχέση με τον MRV. Τέλος βλέπουμε ότι από τους πιο αποδοτικούς αλγορίθμους σε κάθε περίπτωση είναι ο MIN CONFLICTS ο οποίος και είναι αρκετά γρήγορος σε θέμα χρόνο αλλά και κάνει τα λιγότερα constraint checks από τα υπόλοιπα.

ΕΡΩΤΗΜΑ 3:

<u>9-12</u>	<u>12-3</u>	<u>3-6</u>
Αλγοριθμική Επιχειρησιακή Έρευνα	<u>Γραμμική Άλγεβρα</u>	Σχεδίαση Ψηφιακών Συστημάτων -VHDL
Προηγμένα Θέματα Αλγορίθμων	Διακριτά Μαθηματικά	Λειτουργικά Συστήματα
Ειδικά Θέματα Θεωρητικής Πληροφορικής: Αλγόριθμοι Δομικής Βιοπληροφορικής	Εισαγωγή στην Πληροφορική και στις Τηλεπικοινωνίες	<u>Αρχιτεκτονική Υπολογιστών II</u>
Αλγόριθμοι Βιοπληροφορικής	Λογική Σχεδίαση	Τηλεπικοινωνιακά Δίκτυα
Προηγμένοι Επιστημονικοί Υπολογισμοί	Εισαγωγή στον Προγραμματισμό	-
Υπολογιστική Πολυπλοκότητα	<u>Ανάλυση II</u>	Ψηφιακή Επεξεργασία Σήματος
Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα	Πιθανότητες και στοιχεία Στατιστικής	Αρχές Γλωσσών Προγραμματισμού
Ανάπτυξη Λογισμικού για Πληροφοριακά Συστήματα	Αντικειμενοστραφής Προγραμματισμός	-
Επικοινωνία Ανθρώπου Μηχανής	<u>Σήματα και Συστήματα</u>	
ΤΠΕ στη Μάθηση (Πληροφορική κ Εκπαίδευση)	Εργαστήριο Κυκλωμάτων και Συστημάτων	
<u>Τεχνητή Νοημοσύνη II (ΤΟ ΔΕΥΤΕΡΟ ΚΑΛΥΤΕΡΟ!!!)</u>	-	
Δομή και Θεσμοί της Ευρωπαϊκής Ένωσης	Γραφικά I	
Σχεδίαση VLSI Κυκλωμάτων	<u>Αριθμητική Ανάλυση</u>	
Ειδικά Θέματα Επικοινωνιών και Επεξεργασίας Σήματος: Ενισχυτική μηχανική μάθηση και στοχαστικά παιγνία	Εφαρμοσμένα Μαθηματικά	
Ανάπτυξη Λογισμικού για Συστήματα Δικτύων και Τηλεπικοινωνιών	<u>Υλοποίηση Συστημάτων Βάσεων Δεδομένων</u>	-
Συστήματα Πληροφορικής και e-Προσβασιμότητα για μαθητές με αναπηρία	Τεχνητή Νοημοσύνη (ΤΟ ΚΑΛΥΤΕΡΟ!!!!)	-

Οι ελάχιστες μέρες ενός προγράμματος είναι 16 και ο λόγος είναι ότι στο 7ο έτος έχουμε 16 μαθήματα. Από τη στιγμή λοιπόν που υπάρχει ο περιορισμός “Τα μαθήματα του ίδιου έτους θα πρέπει να εξετάζονται σε διαφορετικές ημέρες” τότε θα έχουμε πρόγραμμα με ΤΟΥΛΑΧΙΣΤΟΝ 16 μέρες. Από το παραπάνω πρόγραμμα όμως φαίνεται ότι ο συγκεκριμένος περιορισμός είναι και ο πιο σοβαρός καθώς γίνεται να δημιουργήσουμε πρόγραμμα με ακριβώς τις ελάχιστες ημέρες, δηλαδή 16.