

## Τεχνητή Νοημοσύνη Εργασία 3

Όνομα: Μιχάλης Δεικτάκης

AM: 1115200800018

### Πρόβλημα 1 (exam timetabling):

#### 1. Σχετικά με την υλοποίηση:

Το πρόβλημα που μας δόθηκε έχω υλοποιήσει ως CSP με τον εξής τρόπο:

Αρχικά οι μεταβλητές είναι μια λίστα με τα Μαθήματα ενώ ως domain για την κάθε μεταβλητή έχω ένα dictionary με κλειδί την μεταβλητή και value μια λίστα από όλα τα διαθέσιμα spots με τη μορφή tuple (day, hour) τα οποία τελικά θα πρέπει να αντιστοιχηθούν. (Μπορούμε να έχουμε κενά spots προφανώς). Επίσης επειδή τα Μαθήματα τα διαβάζουμε από το csv προσθέτω και τα εργαστήρια ως μεταβλητές με όνομα Καθηγητή ίδιο και ίδιο εξάμηνο.

Τα constraints υλοποιούνται όπως αναφέρει και στο αρχείο csp ως μια συνάρτηση που επιστρέφει true/false για ένα ζευγάρι μεταβλητών με την αντίστοιχη τιμή τους. Επίσης για την υλοποίηση του αλγορίθμου dom/wdeg έχουμε ένα dictionary weights το οποίο έχει ως κλειδιά κάθε συνδυασμό (A,B) μεταβλητών που συμμετέχουν σε κάποιο constraint (εδώ όλες συμμετέχουν μεταξύ τους αφού δύο μεταβλητές δεν μπορούν να έχουν την ίδια θέση). Κάθε value αρχικοποιείται σε 1 και κάθε φορά που ένα από τα constraints προκαλεί domain wipeout (η μεταβλητή δεν έχει άλλες διαθέσιμες τιμές) αυξάνουμε κατά 1.

Έτσι χρησιμοποιώντας αυτό τον αλγόριθμο η μεταβλητή που θα επιλεγεί από τον αλγόριθμο backtracking θα έχει τον μικρότερο λόγο αριθμός\_διαθέσιμων\_μεταβλητών/weights\_από\_constraints\_που\_προκάλεσαν\_wipe , δηλαδή αυτή που έχει τις λιγότερες διαθέσιμες μεταβλητές και έχει δημιουργήσει τα περισσότερα wipeouts.

Για την εκτέλεση:

Καλείτε το πρόγραμμα με command line arguments 1. τον backtracking αλγόριθμο που θέλουμε να χρησιμοποιήσουμε και 2. την ευρετική συνάρτηση.

fc = forward\_checking , mac = mac , mc = min\_conflicts

mrv = mrv , dw = dom/wdeg (για τον min\_conflicts δεν χρειάζεται να δοθεί 2ο όρισμα)

π.χ. python3 exam\_schedule.py fc dw , python3 exam\_schedule.py mc

## 2. Αποτελέσματα:

Έτρεξα τους αλγόριθμους 10 φορές και θα παρουσιάσουμε αποτελέσματα χρόνου και αριθμό assignments ως μέσο όρο, ελάχιστο και μέγιστο.

	FC - MRV		FC - DW		MAC - MRV		MAC - DW		MN CONFLICTS	
	time (s)	assigns	time (s)	assigns	time (s)	assigns	time (s)	assigns	time (s)	assigns
1	0.0442	45	0.0385	48	0.4373	43	0.4187	43	0.4448	156
2	0.0471	45	0.0349	48	0.4612	43	0.4354	43	0.1662	79
3	0.0525	50	0.051	48	0.4286	43	0.4358	43	0.2459	106
4	0.0452	46	0.0398	48	0.4286	43	0.4112	43	0.2395	94
5	0.059	46	0.0416	48	0.4338	43	0.4186	43	0.2832	71
6	0.0506	49	0.0357	48	0.4276	43	0.4203	43	0.2253	86
7	0.0449	43	0.0425	48	0.4183	43	0.4135	43	0.1342	54
8	0.0469	51	0.036	48	0.4462	43	0.4071	43	0.2393	86
9	0.0544	49	0.0446	48	0.4294	43	0.4069	43	0.1282	64
10	0.05	51	0.0366	48	0.4103	43	0.4184	43	0.1155	63
AVER.	0.04948	47.5	0.04012	48	0.43213	43	0.41859	43	0.22221	85.9
MIN	0.0442	43	0.0349	48	0.4103	43	0.4069	43	0.1155	54
MAX	0.059	51	0.051	48	0.4612	43	0.4358	43	0.4448	156

Βλέπουμε λοιπόν ταχύτερος είναι ο FC με χρήση της dom/wdeg ενώ ο πιο αργός είναι ο MAC με χρήση της ευρετικής min.

Γενικά βλέπουμε πως ο dom/wdeg μειώνει ελάχιστα τον χρόνο εκτέλεσης του προγράμματος για το συγκεκριμένο csr παρόλο που εκτελεί περισσότερες πράξεις για την επιλογή της μεταβλητής (μεγαλύτερη πολυπλοκότητα). Αυτό συμβαίνει διότι τελικά ο αλγόριθμος επιλέγει πρώτα πιο “δύσκολη” μεταβλητή σε σχέση με τον min αλλά η διαφορά είναι πολύ μικρή σε διότι τα wipouts που εμφανίζονται είναι λίγα (6 συγκεκριμένα).

Επίσης είναι λογικό να βλέπουμε τον MAC να έχει 10πλάσιο χρόνο από τον FC αφού ελέγχει για μια μεταβλητή όλες τις περιπτώσεις ασυνέπειας. Βέβαια όπως βλέπουμε και από τα αποτελέσματα ο αριθμός από assignments είναι ακριβώς 43 για τον mac (όσες και οι μεταβλητές) ενώ για τον FC είναι κατά μέσο όρο 47.5 που σημαίνει πως ανέθεσε τιμή σε μια μεταβλητή και παρακάτω βρήκε ασυνέπεια.

Τέλος για τον min\_conflicts βλέπουμε την τυχαιότητα σε θέμα χρόνου όσο και αναθέσεων αφού χρησιμοποιεί την rand για επιλογή της μεταβλητής για την οποία θα αναθέσει τιμή. Παρ’όλα αυτά βρίσκει σε αρκετά καλό χρόνο λύση (ο μέσος όρος είναι ανάμεσα στους δύο προηγούμενους) ενώ θα δούμε μεγαλύτερη χρησιμότητα στο παρακάτω ερώτημα.

## 3. Ελάχιστος χρόνος εξεταστικής:

Αρχικά κάτω από 15 ημέρες δεν επαρκούν οι θέσεις για να βρεθεί λύση.

Για 15 ημέρες σύμφωνα με τους περιορισμούς που μας έχουν δοθεί κοιτώντας συγκεκριμένα το 7ο εξάμηνο έχουμε 16 μαθήματα οπότε αναγκαστικά χρειαζόμαστε τουλάχιστον 16 μέρες. Έχοντας αυτό ως λογική και κάνοντας δοκιμές για επίλυση για 16-20 μέρες με τους παραπάνω αλγόριθμους βλέπουμε ότι μπορεί να επιλυθεί και για 16 μέρες.

Σχετικά με τους αλγόριθμους σε αυτή την περίπτωση εδώ ο min conflicts μας βρίσκει πάντα μια γρήγορη λύση με μέσο όρο περίπου 0.6 s.

Όσον αφορά τον FC με χρήση του min με 20 μέρες βρίσκει γρήγορη λύση ενώ για λιγότερες τρέχει αρκετή ώρα 5 λεπτά+ ενώ με χρήση του dom/wdeg βρίσκει λύση γρήγορα για 17+

μέρες επίσης παρατηρώ πως βρίσκει ασχέτως των ημερών πάντα την ίδια λύση (ίδιες αναθέσεις).

Όσον αφορά τον mac για λιγότερες από 20 μέρες κάνει ομοίως πολύ ώρα το οποίο είναι και λογικό αφού είναι πιο αργός από τον FC

## Πρόβλημα 2:

### 1.

Αρχικά ως μεταβλητές για το πρόβλημα σκέφτηκα να ορίσω τον χρόνο που έλειπε ο κάθε ύποπτος από από την αίθουσα:

$X1 = \text{time\_gone\_Giannis}$

$X2 = \text{time\_gone\_Maria}$

$X3 = \text{time\_gone\_Olga}$

Τα πεδία τιμών για την κάθε μεταβλητή θα είναι χρόνος σε λεπτά από 10 λεπτά ελάχιστο πιθανό έως 150 λεπτά σύμφωνα με το μέγιστο χρόνο που θα χρειαζόταν να λείπει κάποιος για την κλοπή δηλαδή

$\text{domain}[X1] = [10, 11, 12, \dots, 149, 150]$

$\text{domain}[X2] = [10, 11, 12, \dots, 149, 150]$

$\text{domain}[X2] = [10, 11, 12, \dots, 149, 150]$

Όσον αφορά τους περιορισμούς έχουμε:

$X1 \leq 90$

$X2 \leq 60$

$X3 \leq 30$

$85 \leq X1 \leq 150 \text{ or } 85 \leq X2 \leq 150 \text{ or } 85 \leq X3 \leq 150$  (τουλάχιστον μια από τις μεταβλητές έχει τιμή μεγαλύτερη από 85 και μικρότερη από 150)

### 2.

Αναθέτοντας τιμές στις μεταβλητές σύμφωνα με τους περιορισμούς μας βρίσκουμε ότι μόνο θέτοντας τιμή από 85 έως 90 στην μεταβλητή  $X1$  μπορούμε να βρούμε λύση στο πρόβλημα μας που να είναι συνεπής και επίσης καμία άλλη μεταβλητή δεν μπορεί να πάρει τιμή από 85 έως 150. Οπότε ο ένοχος είναι ο Γιάννης και αυτόν συνέλαβε ο αστυνόμος.

### 3.

Εδώ θα μπορούσαμε να χρησιμοποιήσουμε τον αλγόριθμο Forward Checking σε συνδυασμό με τον ευρετικό μηχανισμό MRV.

Αρχικά επειδή το πλήθος διαθέσιμων τιμών είναι ίσο και οι μεταβλητές εμπλέκονται στον ίδιο αριθμό περιορισμών θα πάρουμε μια μεταβλητή στην τύχη.

Αν επιλέξουμε την μεταβλητή  $X1$  θα διαγραφούν όλες οι τιμές έως την 85 διότι θα κοιτάξει για τις επόμενες μεταβλητές και θα προκαλέσει wipe out (δεν υπάρχει τιμή για τις άλλες μεταβλητές που να ικανοποιεί όλους τους περιορισμούς) επιλέγοντας λοιπόν πάλι με τον min εφόσον η  $X1$  έχει τις λιγότερες διαθέσιμες τιμές θα επιλεγεί πάλι έως ότου πάρει την τιμή 85 οπότε και θα βρούμε λύση.

Αν επιλεγεί τυχαία μια μεταβλητή από τις  $X_2$   $X_3$  στην τιμή 10 θα κοιτάξει και θα δει πως υπάρχει διαθέσιμη τιμή για τις άλλες 2 μεταβλητές και ομοίως θα καταλήξει σε λύση με την  $X_1$  να παίρνει την τιμή 85 και οι άλλες 2 μεταβλητές την τιμή 10.

### Πρόβλημα 3:

#### 1.

Αρχικά έχουμε  $n$  εργασίες και η κάθε εργασία θα έχει  $m$  ενέργειες  
Επίσης υπάρχουν  $m$  μηχανές που μπορούν να εκτελούν μια ενέργεια κάθε χρονική στιγμή

Κάθε εργασία έχει starting time  $S \geq 0$  και deadline  $D > 0$

Κάθε ενέργεια  $j$  χρειάζεται χρόνο  $d_j$  για να ολοκληρωθεί

Θα ορίσουμε μια ενέργεια της εργασίας  $i$  ως  $T_{ij}$ .

Αρχικά θα ορίσουμε τις μεταβλητές μας.

Αυτές θα είναι μια οι ενέργειες  $T_{ij}$ .

Το πεδίο τιμών κάθε μεταβλητής  $T_{ij}$  θα είναι δuple (tuple) του τύπου  $(S_{ij}, M_j)$  όπου:

$S_{ij}$  = starting\_time\_of\_  $T_{ij}$  (η χρονική στιγμή που θα ξεκινήσει θα είναι από 0 έως  $D - d_j$ )

$M_j$  = machine (η μηχανή που θα εκτελέσει την ενέργεια με πλήθος από 1 έως  $m$ )

πχ  $(0, 1)$ ,  $(1, 1)$ ... $(t, j)$ , ...,  $(D - d, m)$  κάποιες τιμές για κάθε ενέργεια

προσοχή πως εδώ το  $D$  είναι χρονική στιγμή και όχι διάρκεια, και επίσης τον περιορισμό του deadline τον έχουμε υλοποιήσει μέσα στο πεδίο τιμών εφόσον είναι γνωστός από πριν, δηλαδή καμία μεταβλητή δεν μπορεί να πάρει τιμή τέτοια ώστε  $S_{ij} + d_j > D$  (αυτό μας ενδιαφέρει πιο πολύ για την τελευταία).

Περιορισμοί:

α)  $S_{ij} + d_j \leq S_{ij+1}$  (Εδώ κωδικοποιούμε την πληροφορία πως κάθε ενέργεια  $T_{ij}$  μιας συγκεκριμένης εργασίας  $i$  θα πρέπει να τελειώσει πριν ξεκινήσει η επόμενη  $T_{ij+1}$ )

β) if  $M_j == M_l$ :

$S_{kl} + d_l \leq S_{ij}$  or  $S_{kl} \geq S_{ij} + d_j$

(Εδώ κωδικοποιούμε την πληροφορία πως μια μηχανή μπορεί να εκτελεί μια συγκεκριμένη ενέργεια κάθε χρονική στιγμή)

Δηλαδή αν η μηχανή μιας μεταβλητής  $T_{ij}$  είναι η ίδια με της  $T_{kl}$  τότε είτε η  $T_{kl}$  θα τελειώσει πριν ξεκινήσει η  $T_{ij}$ , είτε θα ξεκινήσει αφού τελειώσει η  $T_{ij}$ .

#### 2.

Έχουμε 3 ενέργειες με 12 εργασίες σύνολο και 4 μηχανές

Θέτουμε για το παράδειγμά μας  $d_i = 1$  για όλες τις ενέργειες,  $S_i = 0$  και  $D = 10$

Οπότε έχουμε τις μεταβλητές με τις αναθέσεις :

$T_{11} = (0, M_1)$

$T_{12} = (1, M_1)$

$T_{13} = (2, M_1)$

$T_{14} = (3, M_1)$

$T_{21} = (0, M_2)$

$T_{22} = (1, M_2)$

$T_{23} = (2, M_2)$

$T_{24} = (3, M_2)$

$T_{31} = (0, M_3)$

$T_{32} = (0, M_4)$

$T_{33} = (1, M_4)$

$T_{34} = (2, M_4)$

Σε αυτή την περίπτωση η ανάθεση είναι συνεπής κι εφόσον είναι και πλήρης κάθε μεταβλητή έχει μια τιμή είναι και μια λύση του προβλήματος.

$T_{11} = (0, M_1)$

$T_{12} = (2, M_1)$  // μη συνεπής ανάθεση τιμών στις μεταβλητές  $T_{12}$  ,  $T_{13}$

$T_{13} = (1, M_1)$  // παραβιάζουν τον περιορισμό α)

$T_{14} = (3, M_1)$

$T_{21} = (0, M_2)$

$T_{22} = (1, M_2)$

$T_{23} = (2, M_2)$

$T_{24} = (3, M_2)$

$T_{31} = (0, M_3)$

$T_{32} = (0, M_4)$

$T_{33} = (1, M_4)$

$T_{34} = (2, M_4)$

**3.** Θα επέλεγα τον mac γι αυτή την περίπτωση πιστεύω πως είναι αρκετα σημαντικό να κρατάμε τη συνέπεια ακμής ώστε να διαγραφούν όσο το δυνατότερο περισσότερες τιμές οι οποίες οδηγούν σε ασυνέπεια και σύμφωνα με τη μοντελοποίηση είναι αρκετες. πχ για μια μεταβλητή  $T_{i1}$  οι τιμές  $(d_1+d_2, M_j) \dots (D-d_1, M_j)$  είναι πολύ πιθανό να οδηγήσουν σε ασυνέπεια.

#### Πρόβλημα 4:

- $(A \wedge B \wedge C \Rightarrow D) \Leftrightarrow (A \Rightarrow (B \Rightarrow (C \Rightarrow D)))$

Θα προχωρήσουμε σε ανάλυση σε CNF πρώτα εσωτερικά στις δύο προτάσεις

$$\begin{array}{ll}
(A \wedge B \wedge C \Rightarrow D) & (A \Rightarrow (B \Rightarrow (C \Rightarrow D))) \\
\neg(A \wedge B \wedge C) \vee D & \neg A \vee (B \Rightarrow (C \Rightarrow D)) \\
(\neg(A \wedge B) \vee (\neg C)) \vee D & \neg A \vee (\neg B \vee (C \Rightarrow D)) \\
(\neg A \vee \neg B \vee \neg C) \vee D & \neg A \vee (\neg B \vee (\neg C \vee D)) \\
\neg A \vee \neg B \vee \neg C \vee D & \Leftrightarrow \neg A \vee \neg B \vee \neg C \vee D
\end{array}$$

Μετά σε ολόκληρη την πρόταση (επειδή έχει μεγάλο μέγεθος)

$$\begin{aligned}
& (\neg A \vee \neg B \vee \neg C \vee D) \Rightarrow (\neg A \vee \neg B \vee \neg C \vee D) \wedge (\neg A \vee \neg B \vee \neg C \vee D) \Rightarrow (\neg A \vee \neg B \vee \neg C \vee D) \equiv \\
& \neg(\neg A \vee \neg B \vee \neg C \vee D) \vee (\neg A \vee \neg B \vee \neg C \vee D) \wedge \neg(\neg A \vee \neg B \vee \neg C \vee D) \vee (\neg A \vee \neg B \vee \neg C \vee D) \equiv \\
& (\neg \neg A \wedge \neg \neg B \wedge \neg \neg C \wedge \neg D) \vee (\neg A \vee \neg B \vee \neg C \vee D) \wedge (\neg \neg A \wedge \neg \neg B \wedge \neg \neg C \wedge \neg D) \vee (\neg A \vee \neg B \vee \neg C \vee D) \equiv \\
& (A \wedge B \wedge C \wedge D) \vee (\neg A \vee \neg B \vee \neg C \vee D) \wedge (A \wedge B \wedge C \wedge D) \vee (\neg A \vee \neg B \vee \neg C \vee D) \equiv \\
& (A \vee \neg A) \wedge (B \vee \neg B) \wedge (C \vee \neg C) \wedge (D \vee D) \wedge (A \vee \neg A) \wedge (B \vee \neg B) \wedge (C \vee \neg C) \wedge (D \vee D) \equiv \\
& (D \vee D) \wedge (D \vee D)
\end{aligned}$$

και καταλήγουμε σε ταυτολογία.

Άρα 1) είναι έγκυρη 2) είναι ικανοποιήσιμη 3) λάθος 4) έχει τουλάχιστον ένα μοντέλο 5) είναι ταυτολογία 6) Δεν μπορεί να μετατραπεί σε horn

$$\bullet A \wedge (A \Rightarrow B) \wedge (A \Rightarrow \neg B)$$

$$\begin{aligned}
& A \wedge (A \Rightarrow B) \wedge (A \Rightarrow \neg B) \equiv \\
& A \wedge (\neg A \vee B) \wedge (\neg A \vee \neg B) \equiv \\
& A \wedge (\neg A \vee (B \wedge \neg B)) \equiv \\
& A \wedge \neg A
\end{aligned}$$

Άρα 1) δεν είναι έγκυρη 2) δεν είναι ικανοποιήσιμη 3) είναι μη ικανοποιήσιμη 4) δεν έχει κανένα μοντέλο 5) δεν είναι ταυτολογία 6) δεν είναι/μπορεί να γίνει Horn

$$\bullet (A \vee B) \wedge (\neg A \vee C) \wedge \neg B \wedge \neg C$$

$$\begin{aligned}
& (A \vee B) \wedge (\neg A \vee C) \wedge \neg B \wedge \neg C \equiv \\
& \neg B \wedge (A \vee B) \wedge \neg C \wedge (\neg A \vee C) \equiv \\
& ((A \wedge \neg B) \vee (B \wedge \neg B)) \wedge ((\neg A \wedge \neg C) \vee (C \wedge \neg C)) \equiv \\
& (A \wedge \neg B) \wedge (\neg A \wedge \neg C) \equiv \\
& \neg(\neg A \vee B) \wedge \neg(A \vee C) \equiv \\
& \neg((\neg A \vee B) \vee (A \vee C)) \equiv \\
& \neg(\neg A \vee A \vee B \vee C) \equiv \\
& \neg(\neg A \vee A) \equiv \\
& A \wedge \neg A
\end{aligned}$$

Άρα 1)δεν είναι έγκυρη 2)δεν είναι ικανοποιήσιμη 3)είναι μη ικανοποιήσιμη 4)δεν έχει κανένα μοντέλο 5) δεν είναι ταυτολογία 6)δεν είναι σε μορφή Horn

$$\bullet (A \vee B) \wedge (\neg A \vee C) \wedge (B \vee C)$$

A	B	C	$A \vee B$	$\neg A \vee C$	$B \vee C$	X
TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE
TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE
FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE

1)δεν είναι έγκυρη 2) είναι ικανοποιήσιμη 3) δεν είναι μη ικανοποιήσιμη 4) έχει τουλάχιστον ένα μοντέλο(έχει 4), 5)δεν είναι ταυτολογία

## Πρόβλημα 6:

Σύμφωνα με τις διαφάνειες

- Είναι καλά ορισμένη (atomic sentenc)
- Είναι καλά ορισμένη (πρόταση σε παρένθεση)
- Δεν είναι καλά ορισμένη ( το  $\equiv$  δεν είναι binary )
- Δεν είναι καλά ορισμένη (ομοίως δεν είναι binary σύμβολο)
- Δεν είναι καλά ορισμένη (Στις σταθερές δεν έχουμε το 1 θεωρώ πως δεν είναι ίδιο με το true, επίσης δεν είναι Σύμβολο σύμφωνα με τις διαφάνειες).