ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών
ΙΔΡΥΘΕΝ ΤΟ 1837

Τεχνητή Νοημοσύνη Ι

Βασιλική Χριστοφιλοπούλου
1115202000216

Δεκέμβριος 2024

Χειμερινό Εξάμηνο 2024-2025

Project 3

Table of Contents

1 Πρόβλημα 1: Exam timetabling

1.1 Περιγραφή ζητουμένου

Η αναπαράσταση του CSP περιλαμβάνει:

- **Μεταβλητές** Τα μαθήματα.
- **Τιμές** Οι διαθέσιμες ημέρες και ώρες.
- **Περιορισμοί**
  - Διαφορετικές ημέρες για τα μαθήματα του ίδιου έτους.
  - Ακολουθία θεωρίας και εργαστηρίου.
  - Απόσταση τουλάχιστον 2 ημερών για τα δύσκολα μαθήματα.
  - Διαφορετικές ημέρες για μαθήματα του ίδιου καθηγητή.

Για να υλοποιηθούν οι ζητούμενοι αλγόριθμοι απο το αποθετηριο του github χρησιμοποιήθηκαν τα αρχεία: csp.py, search.py, utils.py

Για την υλοποίηση του προβλήματος δημιουργήθηκε μια νέα κλάση Exam_Timetabling, η οποία στοχεύει να κατανείμει τα μαθήματα σε διάφορες ημέρες και χρονικές ζώνες, με βάση διάφορους περιορισμούς.
Αρχικά, διαβάζεται το αρχείο h3-data.csv και για κάθε μάθημα αποθηκεύονται σε λίστες (semester, variables, professors, difficulty) τα δεδομένα προκειμένου να αξιοποιηθούν στην συνέχεια. Αν ένα μάθημα έχει εργαστήριο (collumn 5 - true), τότε δημιουργείται ένα νεό μάθημα με όνομα (μάθημα_Lab) για το οποίο έπισης αποθηκεύονται οι τιμές εξάμηνο, καθηγητής και δυσκολία. (Κάθε εργαστηριο θεωρείται ευκολο, προκειμένου να μην υπάρξει concflict με τον περιορισμό ότι χρειάζεται απόσταση τουλάχιστον 2 ημερών για δύσκολα μαθήματα). Στην συνέχεια, δημιουργούνται λεξικά (lesson_semester, lesson_professors, lesson_difficulty) για εύκολη αναζήτηση του εξαμήνου, του καθηγητή και της δυσκολίας για κάθε μάθημα. Έπειτα για κάθε μάθημα είναι αναγκαίο να βρεθεί το πεδίο τιμών που μπορεί να πάρει. Τα ζευγάρια που μπορεί να τοποθετηθουν θεωρία και εργαστήριο είναι "9-12" - "12-3" και "12-3" - "3-6". Συνεπως. μπορούμε να αντιληθούμε ότι:

- ένα εργαστηριο δεν μπορεί να είναι στην βάρδια "9-12"
- μια θεωρία που έχει εργαστήριο δεν μπορεί να είναι στην βάρδια "3-6"
- και ενα μάθημα χωρίς εργαστήριο μπορεί να είναι σε όλες τις βάρδιες

Με άξονα τα παραπάνω δημιουργούμε το domain του κάθε μαθήματος αλλά και τους γείτονές του.

Δημιουργήθηκε επίσης μια συνάρτηση constraints, η οποία υλοποιεί τους περιορισμούς που περιγράφονται απο την άσκηση.

Στο τέλος, η κύρια συνάρτηση if __name__ == '__main__': δημιουργεί ένα αντικείμενο του τύπου Exam_Timetabling, το οποίο φορτώνει τα δεδομένα από το αρχείο CSV και καλεί τη συνάρτηση solve_timetabling για να λύσει το πρόβλημα του χρονοδιαγράμματος για 21 ημέρες, και εκτυπώνει τα σταστιστικά των αλγοριθμων.

1.2 dom/wdeg

Χρησιμοποιήθηκαν απο το csp.py οι αλγόριθμοι fc, mac και min conflicts σε συνδυασμό με τις ευρετικές mrv και dom/wdeg που υλοποιήθηκε απο μένα (Αρχείο csp.py γραμμές 373 - 393).

Επιπλέον προστέθηκαν στην init του csp οι παρακάτω γραμμέςπου αρχικοποοιούν μετρητές που χρησιμοποιούνται στις μετρικές αλλά και τα βάρη για τον dom/wdeg.

```
self.backtracks = 0       # Initialize the backtrack counter
self.nodes_visited = 0    # Initialize the nodes visited counter

    self.weights = dict()   # for dom_wdeg
    for var1 in self.variables:
        for var2 in self.variables:
            self.weights[(var1, var2)] = 1
```

Για την υλοποίηση του dom/wdeg βασίστηκα στο pdf που υπάρχει στην εργασία και επιπλέον η συνάρτηση βασίζεται σε δύο κριτήρια για την επιλογή της καλύτερης μεταβλητής:

- **Degree Heuristic:** Αντί να επιλέγουμε μια τυχαία μεταβλητή, προτιμούμε τις μεταβλητές που συμμετέχουν σε περιορισμούς με άλλες μη ανατεθειμένες μεταβλητές (μεγαλύτερη "βαρύτητα" ή "degree").
- **Domain Heuristic:** Επιλέγουμε τις μεταβλητές με το μικρότερο μέγεθος τομέα ή με τη μικρότερη πιθανότητα σύγκρουσης για να περιορίσουμε τον αριθμό των επόμενων επιλογών.

1.3 Σύγκριση Αλγορίθμων CSP

Για την αξιολόγηση των αλγορίθμων που υλοποιήθηκαν, επιλέχθηκαν οι εξής μετρικές:

- **Χρόνος Εκτέλεσης (Time Taken)**
  Ο χρόνος που χρειάζεται ο αλγόριθμος για να βρει μια λύση.
- **Συνολικοί Αναθέσεις (Total nassigns)**
  Αντιπροσωπεύει τον αριθμό των αναθέσεων μεταβλητών κατά τη διάρκεια της εκτέλεσης.
- **Έλεγχοι Περιορισμών (Constraints Checked)**
  Ο αριθμός των ελέγχων περιορισμών που έγιναν για να επαληθευτεί η εγκυρότητα της λύσης.
- **Αναδρομή (Backtracks)**
  Η αναδρομή αναφέρεται στις περιπτώσεις που ο αλγόριθμος πρέπει να "επιστρέψει πίσω" και να δοκιμάσει άλλες επιλογές για μια μεταβλητή, λόγω αποτυχίας να ικανοποιήσει τους περιορισμούς.
- **Επισκέψεις Κόμβων (Nodes Visited)**
  Ο αριθμός των κόμβων που επισκέπτεται ο αλγόριθμος κατά την αναζήτηση λύσης.

Σε μια απο τις φορές που έτρεξα το πρόγραμμα για 21 μέρες εξεταστικής και 3 slots πήρα τα εξής αποτελέσματα:

Algorithm	Time (s)	Total nassigns	Constraints Checked	Backtracks	Nodes Visits
FC + MRV	0.3946s	43	189560	43	44
FC + DOM/WDEG	0.0239s	86	195881	43	44
MAC + MRV	0.0290s	129	7224	43	44
MAC + DOM/WDEG	0.0464s	172	7224	43	44
Min Conflicts	4.0174s	236	46039	43	44

Table 1: 21 days and 3 slots

Απο τον ανωτέρω πίνακα μπορούμε να αντιληθούμε ότι:

1. **Ταχύτητα**
   Ο αλγόριθμος FC + DOM/WDEG είναι ο ταχύτερος με χρόνο 0.0239 δευτερόλεπτα, ενώ ο αλγόριθμος Min Conflicts χρειάζεται πολύ περισσότερο χρόνο (7.7799 δευτερόλεπτα). Οι αλγόριθμοι που συνδυάζουν Forward Checking (FC) και Degree/Weighted Degree (DOM/WDEG) είναι σαφώς πιο γρήγοροι, κάτι που δείχνει την αποτελεσματικότητα των τεχνικών προεπεξεργασίας.
2. **Συνολικοί Αναθέσεις (Total nassigns)**
   Ο αλγόριθμος FC + MRV έχει τον χαμηλότερο αριθμό αναθέσεων (43), ενώ οι άλλοι αλγόριθμοι, εκτός από Min Conflicts (638), έχουν επίσης πιο χαμηλούς αριθμούς αναθέσεων. Ο αριθμός αυτός είναι σημαντικός, καθώς υποδηλώνει την αποδοτικότητα στην κατανομή των μεταβλητών.
3. **Έλεγχοι Περιορισμών (Constraints Checked)**
   Οι αλγόριθμοι με FC + MRV και FC + DOM/WDEG πραγματοποίησαν πολλούς περισσότερους ελέγχους περιορισμών (πάνω από 180,000), ενώ ο αλγόριθμος με MAC χρειάστηκε πολύ λιγότερους ελέγχους (7224).

Για 20 μέρες και 3 slots:

Algorithm	Time (s)	Total nassigns	Constraints Checked	Backtracks	Nodes Visits
FC + MRV	0.3594	43	178791	43	44
FC + DOM/WDEG	0.0239	86	185112	43	44
MAC + MRV	0.0320	129	7224	43	44
MAC + DOM/WDEG	0.0335	172	7224	43	44
Min Conflicts	0.7953	215	16314	43	44

Table 2: 20 days and 3 slots

Συνεπώς, οι αλγόριθμοι με FC + DOM/WDEG και MAC εμφανίζονται πιο γρήγοροι σε σύγκριση με τους άλλους, λόγω του pruning που κάνουν, ωστόσο η επιλογή του κατάλληλου αλγορίθμου εξαρτάται από το αν προτιμάται η ταχύτητα ή αποδοτικότητα σε πιο περιορισμένα πλαίσια χρόνου και πόρων.

1.4 Ελάχιστη διάρκεια εξεταστικής

Για να βρούμε τον ελάχιστο χρόνο διάρκειας της εξεταστικής για το πρόβλημα του προγράμματος εξετάσεων (exam timetabling), έχουμε

- 38 μαθήματα συνολικά
- 7 δύσκολα μαθήματα
- 5 εργαστήρια

Απο τον περιορισμό ότι πρέπει να υπάρχουν τουλάχιστον 2 ημέρες απόσταση μεταξυ των δύσκολων μαθημάτων, έχουμε ότι κατ ελάχιστο η εξεταστική πρέπει να έχει 2 * 7 = 14 ημέρες.
Απο τον περιορισμό ότι μαθήματα του ίδιου εξαμήνου δεν μπορούν να εξετάζονται την ίδια ημέρα, έχουμε ότι:

- 1ο εξάμηνο : 5 μαθήματα
- 3ο εξάμηνο : 5 μαθήματα
- 5ο εξάμηνο : 12 μαθήματα
- 7ο εξάμηνο : 16 μαθήματα

Άρα με την υπόθεση ότι κάθε μέρα εξετάζεται ένα μάθημα απο το 7ο εξάμηνο που έχει και τα περισσότερα έχουμε ότι κατ ελάχιστο η εξεταστική πρέπει να έχει 16 ημέρες.

Οι υπόλοιποι περιορισμοί προσθέτουν επιπλέον απαιτήσεις, αλλά δεν επηρεάζουν άμεσα την ελάχιστη διάρκεια της εξεταστικής, δεδομένου ότι ο αριθμός των μαθημάτων που πρέπει να εξεταστούν είναι το κύριο κριτήριο.

Άρα, η ελάχιστη διάρκεια της εξεταστικής είναι τουλάχιστον 16 ημέρες, καθώς αυτός είναι ο μεγαλύτερος περιορισμός.

2 Πρόβλημα 2: Μοντελοποίηση με προβλήματα ικανοποίησης περιορισμών

Το πρόβλημα ικανοποίησης περιορισμών μοντελοποιείται ως εξής:

- **Μεταβλητές**
  Κρεβάτι, γραφείο, καρέκλα γραφείου, καναπές
- **Πεδίο ορισμού**
  Για κάθε μια απο τις μεταβλητές έχουμε:
  - Πλάτος $\in [0, 300\text{cm}]$
  - Μήκος $\in [0, 400\text{cm}]$
  - Ύψος $\in [0, 230\text{cm}]$
  - Βάθος $\in [0, 60\text{cm}]$
- **Περιορισμοί τοποθέτησης**
  1. Τα έπιπλα δεν πρέπει να εφάπτονται ή να αλληλεπικαλύπτονται
  2. Το κρεβάτι πρέπει να ακουμπά σε έναν τοίχο.
  3. Το γραφείο πρέπει να είναι:
     - Σε τοίχο.
     - Απέναντι από την μπαλκονόπορτα (πηγή φωτός).
- **Μαθηματική μοντελοποίηση**
  Για κάθε έπιπλο i, ορίζουμε:
  - x_i: Η συντεταγμένη της κάτω-αριστερής γωνίας του επίπλου στον άξονα x
  - y_i: Η συντεταγμένη της κάτω-αριστερής γωνίας του επίπλου στον άξονα y
  - w_i, d_i: Το πλάτος και το βάθος του επίπλου αντίστοιχα.
- **Περιορισμοί**

- **Τοποθέτηση εντός του δωματίου:**

$$0 \le x_i \le 300 - w_i, \quad 0 \le y_i \le 400 - d_i$$

- **Αποφυγή επικάλυψης:** Για κάθε δύο έπιπλα i, j:

$$(x_i + w_i \le x_j) \vee (x_j + w_j \le x_i) \vee (y_i + d_i \le y_j) \vee (y_j + d_j \le y_i)$$

- **Κρεβάτι σε τοίχο:**

$$(x_{\text{bed}} = 0) \vee (x_{\text{bed}} + w_{\text{bed}} = 300) \vee (y_{\text{bed}} = 0) \vee (y_{\text{bed}} + d_{\text{bed}} = 400)$$

- **Γραφείο απέναντι από την πηγή φωτός** (μπαλκονόπορτα στη θέση $x \in [200, 300]$, $y = 400$):

$$x_{\text{desk}} \in [0, 300 - w_{\text{desk}}], \quad y_{\text{desk}} = 0$$

• **Λύση**
Ακολουθώντας τους περιορισμούς, και ξεκινώντας απο την τοποθέτηση του γραφείου προκύπτει η εξής πιθανή διαρρύθμιση: (Θεωρώντας ότι η αρίθμηση του πλάτους είναι απο αριστερά προς τα δεξιά και του μήκους απο κάτω προς τα πάνω)

- **Γραφείο**
Εφόσον πρέπει να είναι απέναντι απο την μπαλκονόπορτα, το γραφείο έχει

$$x_{\text{desk}} \in [140, 300], \quad y_{\text{desk}} \in [0, 80]$$

- **Καρέκλα γραφείου**
Μπορεί να τοποθετηθεί κοντά στο γραφείο χωρίς να επηρεάζει τη λειτουργικότητα και αφήνοντας 20cm απόσταση απο το γραφείο

$$x_{\text{chair}} \in [200, 241], \quad y_{\text{chair}} = [100, 144]$$

- **κρεβάτι**
Το κρεβάτι μπορεί να τοποθετηθεί

$$x_{\text{bed}} \in [0, 200], \quad y_{\text{bed}} = [300, 400]$$

- **Καναπές**
Μπορεί να τοποθετηθεί σε ελεύθερο χώρο χωρίς να επικαλύπτεται με άλλα έπιπλα,

$$x_{\text{sofa}} \in [0, 221], \quad y_{\text{sofa}} = [150, 253]$$

Ακολουθεί μια οπτικοποίηση της λύσης:

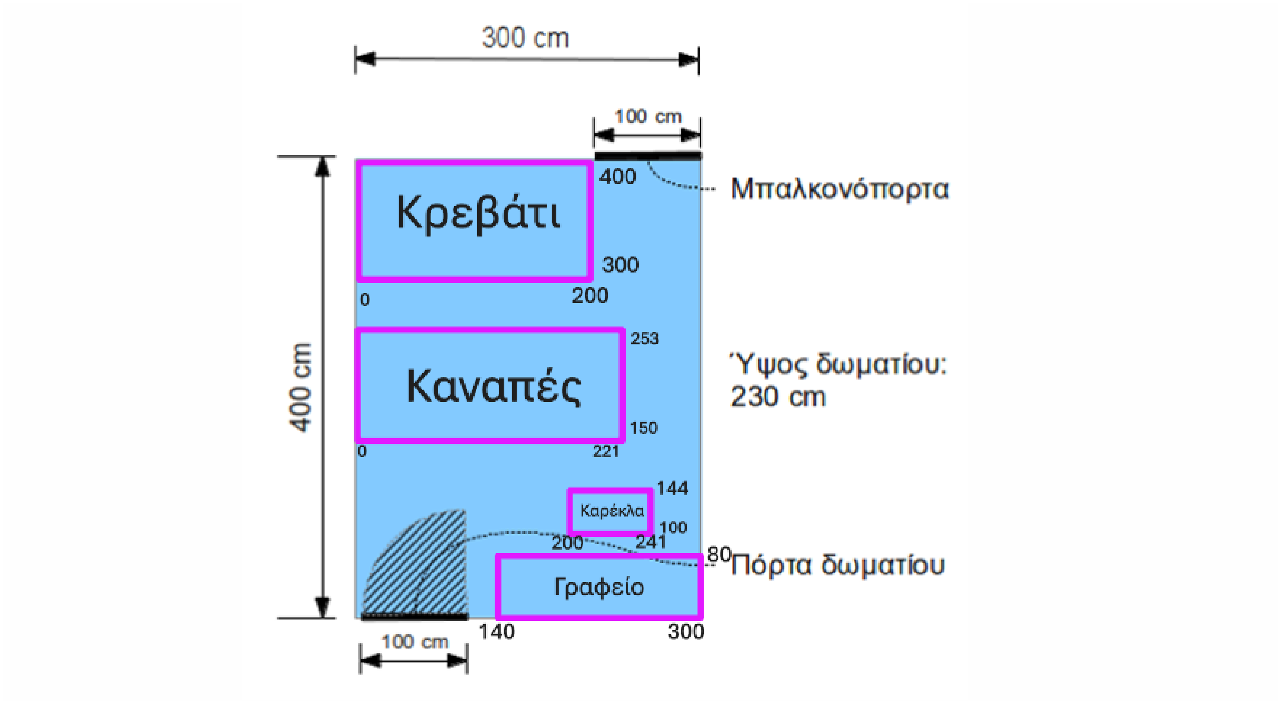

Figure 1: Πιθανή λύση

Το πρόβλημα έχει λύση, και η παραπάνω διάταξη ικανοποιεί τους περιορισμούς αισθητικής και χώρου.

3 Πρόβλημα 3: Μοντελοποίηση με προβλήματα ικανοποίησης περιορισμών

Το πρόβλημα ικανοποίησης περιορισμών μοντελοποιείται ως εξής:

3.1 Μοντελοποίηση

- **Μεταβλητές**
  A1, A2, A3, A4, A5
- **Πεδίο ορισμού**
  09:00, 10:00, 11:00
- **Περιορισμοί**
  - **Η A1 πρέπει να αρχίσει μετά την A3:**
$$X_{A1} > X_{A3}$$
  - **Η A3 πρέπει να αρχίσει πριν την A4 και μετά την A5:**
$$X_{A5} < X_{A3} < X_{A4}$$
  - **Η A2 δεν μπορεί να εκτελείται την ίδια ώρα με την A1 ή την A4**
$$X_{A2} \neq X_{A1} \quad \text{και} \quad X_{A2} \neq X_{A4}$$
  - **Η A4 δεν μπορεί να αρχίσει στις 10:00:**
$$X_{A4} \neq 10:00$$

3.2 Γράφος προβλήματος

Ακολουθεί ο γράφος του προβλήματος:

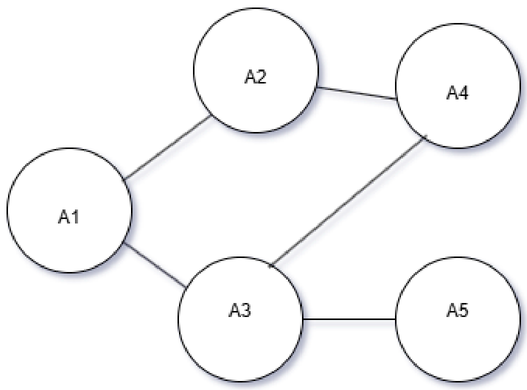

Figure 2: Γράφος προβλήματος

3.3 Αλγόριθμο συνέπειας τόξου AC-3

Θα εφαρμόσουμε τον αλγόριθμο συνέπειας τόξου AC-3 στο πρόβλημα μέχρι αυτό να γίνει συνεπές (arc consistent). Αρχικά κάθε μεταβλητη μπορεί να πάρει κάθε τιμή του πεδίου ορισμού:
A1=$[9:00, 10:00, 11:00]$
A2=$[9:00, 10:00, 11:00]$
A3=$[9:00, 10:00, 11:00]$
A4=$[9:00, 10:00, 11:00]$
A5=$[9:00, 10:00, 11:00]$

Με την σειρά ικανοποιούμε τους περιορισμούς και κάθε φορά μεταβάλλουμε τις τιμές που μπορούν να πάρουν οι μεταβλητές .

1. Αρχικά, η A1 πρέπει να αρχίσει μετά την A3 οπότε διαγράφουμε την τιμή 9:00 απο τις τιμές του A1.

$$A1 = \{10:00, 11:00\}$$
$$A2 = \{9:00, 10:00, 11:00\}$$
$$A3 = \{9:00, 10:00, 11:00\}$$
$$A4 = \{9:00, 10:00, 11:00\}$$
$$A5 = \{9:00, 10:00, 11:00\}$$

2. Το A3 θα πρέπει να έχει τιμές μικρότερες απο το A1 οπότε διαγράφουμε την τιμή 11:00

$$A1 = \{10:00, 11:00\}$$
$$A2 = \{9:00, 10:00, 11:00\}$$
$$A3 = \{9:00, 10:00\}$$
$$A4 = \{9:00, 10:00, 11:00\}$$
$$A5 = \{9:00, 10:00, 11:00\}$$

3. Το A3 θα πρέπει να έχει μικρότερες τιμές απο το A4 όπου και έχει οπότε παραμένει ίδιο ως προς τις τιμές .

$$A1 = \{10:00, 11:00\}$$
$$A2 = \{9:00, 10:00, 11:00\}$$
$$A3 = \{9:00, 10:00\}$$
$$A4 = \{9:00, 10:00, 11:00\}$$
$$A5 = \{9:00, 10:00, 11:00\}$$

4. Το A4 θα πρέπει να έχει τιμές μεγαλύτερες απο το A3 οπότε διαγράφουμε απο το A3 την τιμή 9:00.

$$A1 = \{10:00, 11:00\}$$
$$A2 = \{9:00, 10:00, 11:00\}$$
$$A3 = \{9:00, 10:00\}$$
$$A4 = \{10:00, 11:00\}$$
$$A5 = \{9:00, 10:00, 11:00\}$$

5. Το Α3 θα πρέπει να έχει τιμές μεγαλύτερες απο το Α5 άρα θα διαγραφεί η τιμή 9:00 απο το Α3.

$$A1 = \{10:00, 11:00\}$$
$$A2 = \{9:00, 10:00, 11:00\}$$
$$A3 = \{10:00\}$$
$$A4 = \{9:00, 10:00, 11:00\}$$
$$A5 = \{9:00, 10:00, 11:00\}$$

6. Το Α5 θα πρέπει να έχει τιμές μικρότερες απο το Α3 οπότε θα διαγραφούν οι τιμές 10:00 και 11:00 απο το Α5.

$$A1 = \{10:00, 11:00\}$$
$$A2 = \{9:00, 10:00, 11:00\}$$
$$A3 = \{10:00\}$$
$$A4 = \{9:00, 10:00, 11:00\}$$
$$A5 = \{9:00\}$$

7. Το Α4 δεν θα πρέπει να έχει την τιμή 10:00.

$$A1 = \{10:00, 11:00\}$$
$$A2 = \{9:00, 10:00, 11:00\}$$
$$A3 = \{10:00\}$$
$$A4 = \{9:00, 11:00\}$$
$$A5 = \{9:00\}$$

8. Το Α3 θα πρέπει να έχει τιμές μικρότερες απο το Α1 οπότε διαγράφουμε την τιμή 10:00

$$A1 = \{11:00\}$$
$$A2 = \{9:00, 10:00, 11:00\}$$
$$A3 = \{10:00\}$$
$$A4 = \{9:00, 11:00\}$$
$$A5 = \{9:00\}$$

9. Το Α4 θα πρέπει να έχει τιμές μεγαλύτερες απο το Α3 οπότε διαγράφουμε απο το Α4 την τιμή 9:00.

$$A1 = \{11:00\}$$
$$A2 = \{9:00, 10:00, 11:00\}$$
$$A3 = \{10:00\}$$
$$A4 = \{9:00, 11:00\}$$
$$A5 = \{9:00\}$$

10. Το Α2 δεν θα πρέπει να εκτελεστεί την ίδια ώρα με το Α1 και το Α2 δεν θα πρέπει να εκτελεστεί την ίδια ώρα με το Α4. Άρα διαγράφουμε απο το Α2 την τιμή 11:00 και την τιμή 9:00 απο το Α4

$$A1 = \{11:00\}$$
$$A2 = \{9:00, 10:00\}$$
$$A3 = \{10:00\}$$
$$A4 = \{11:00\}$$
$$A5 = \{9:00\}$$

Συνεπώς, μετα την ολοκλήρωση του αλγορίθμου οι μεταβλητές θα έχουν τις ακόλουθες τιμές:

$$A1 = \{11:00\}$$
$$A2 = \{9:00, 10:00\}$$
$$A3 = \{10:00\}$$
$$A4 = \{11:00\}$$
$$A5 = \{9:00\}$$

4 Πρόβλημα 4: (Χρονικοί περιορισμοί)

4.1 Ερώτημα 1

Απο το κείμενο που δίνεται απο την άσκηση έχουμε, έστω:

$$P_1 = \text{Η Μαρία πάει στην δουλειά}$$
$$P_2 = \text{Η Ελένη πάει στην δουλειά}$$

Η P_1 συσχετίζεται με την X_1 που είναι η ώρα που η Μαρία φεύγει απο το σπίτι της και X_2 που είναι η ώρα που η Μαρία φτάνει στην δουλειά της. Όμοια η P_2 συσχετίζεται με την X_3 που είναι η ώρα που η Ελένη φεύγει απο το σπίτι της και X_4 που είναι η ώρα που η Ελένη φτάνει στην δουλειά της

Με βάση τα δεδομένα που έχουμε, μπορούμε να θέσουμε τους εξής περιορισμούς:

1. Η διάρκεια του ταξιδιού της Μαρίας είναι μεταξύ 30 και 40 λεπτά
   $30 \leq X_2 - X_1 \leq 40$
2. Η διάρκεια του ταξιδιού της Ελένης είναι μεταξύ 5 και 15 λεπτά
   $5 \leq X_4 - X_3 \leq 15$
3. Μαρία φτάνει στο γραφείο μεταξύ 8:30 και 8:50
   $8:30 \leq X_2 \leq 8:50$
4. Η Ελένη φτάνει 15 λεπτά μετά τη Μαρία
   $X_4 = X_2 + 15$

Ψάχνουμε να βρούμε πότε φυγε η Ελένη από το σπίτι της; Αυτό απαιτεί την επίλυση του προβλήματος, δηλαδή να βρούμε το X_3, χρησιμοποιώντας τους περιορισμούς που δίνονται.

Ένα πρόβλημα είναι συνεπές αν υπάρχουν τιμές για τις μεταβλητές που ικανοποιούν όλους τους περιορισμούς. Αν οι περιορισμοί οδηγούν σε αντιφάσεις ή απροσδιόριστες λύσεις, το πρόβλημα δεν είναι συνεπές.

Εξετάζουμε τους περιορισμούς:

- Η Μαρία φεύγει από το σπίτι της μεταξύ 8:00 και 8:10.
- Ο χρόνος που χρειάζεται για το ταξίδι της Μαρία είναι μεταξύ 30 και 40 λεπτών.

  $\implies$ Άρα, η Μαρία φτάνει στο γραφείο της μεταξύ 8:30 και 8:50.

- Η Ελένη φτάνει 15 λεπτά μετά τη Μαρία.

  $\implies$ Επομένως, φτάνει στο γραφείο μεταξύ 8:45 και 9:05.

- Το ταξίδι της Ελένης διαρκεί μεταξύ 5 και 15 λεπτών

  $\implies$ άρα η Ελένη ξεκινά το ταξίδι της μεταξύ 8:30 και 8:50.

Αυτοί οι περιορισμοί δεν προκαλούν σύγκρουση και επομένως το πρόβλημα είναι συνεπές.

Δύο λύσεις για το πρόβλημα είναι:

1. **Λύση 1**
   Η Μαρία ξεκινά στις 8:00, και το ταξίδι της διαρκεί 30 λεπτά άρα φτάνει στις 8:30.
   Η Ελένη ξεκινά στις 8:30, και το ταξίδι της διαρκεί 15 λεπτά άρα φτάνει στις 8:45.

2. **Λύση 2**
   Η Μαρία ξεκινά στις 8:10, και το ταξίδι της διαρκεί 40 λεπτά άρα φτάνει στις 8:50.
   Η Ελένη ξεκινά στις 8:35, και το ταξίδι της διαρκεί 10 λεπτά άρα φτάνει στις 8:45.

4.2 Ερώτημα 2

παρακάτω δίνεται ο γράφος για το δεδομένο πρόβλημα.

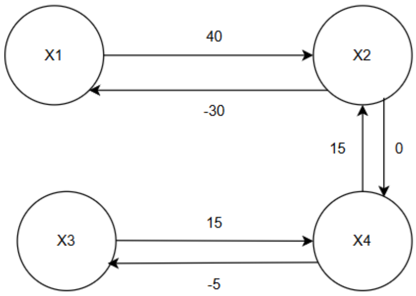

Figure 3: Γράφος προβλήματος

4.3 Ερωτήματα 3,4

Για την υλοποίηση του αλγορίθμου σε python αξιοποιήθηκε τόσο η μεθοδολογία που δίνεται στο άρθρο σελίδα 12 όσο και κώδικας από το geeksforgeeks και από εδώ. Επιπλέον, εκτυπώνεται τόσο ο πίνακας με τις αποστάσεις όσο και ένας γράφος για το ερώτημα 4.

4.4 Ερώτημα 5

Από την θεωρία ξέρουμε ότι ένα CSP είναι k-consistent εάν για κάθε ανάθεση τιμών σε k-1 μεταβλητές $X_1 = v_1, ..., X_k = v_{k-1}$ που ικανοποιεί όλους τους περιορισμούς που εμπλέκουν τις μεταβλητές $X_1, ..., X_{k-1}$ μπορεί να επεκταθεί σε μια ανάθεση.

Ένα χρονικό πρόβλημα όπως αυτό της άσκησης έχει μεταβλητές οι οποίες συνδέονται μεταξύ τους μέσω χρονικών περιορισμών, που προσδιορίζουν τη σχέση μεταξύ των γεγονότων, όπως οι ώρες που συμβαίνουν ή οι διάρκειες των δραστηριοτήτων. Οι περιορισμοί αυτοί δημιουργούν μια δυναμική μεταξύ των μεταβλητών, και κάθε ανάθεση σε μια μεταβλητή επηρεάζει τις υπόλοιπες μεταβλητές.

Όταν επεκτείνουμε τις αναθέσεις σε περισσότερες από μία μεταβλητές (δηλαδή σε k consistency), πρέπει να ελέγχουμε αν οι αναθέσεις σε k-1 μεταβλητές (που ικανοποιούν όλους τους περιορισμούς μεταξύ τους) μπορούν να επεκταθούν σε μια ανάθεση που ικανοποιεί όλους τους περιορισμούς και για την k-στη μεταβλητή.

Για το δεδομένο πρόβλημα της Μαρία και της Ελένης, η λύση μας είναι συνεπής, καθώς δεν υπάρχει καμία σύγκρουση στους περιορισμούς και μπορούμε να επιτύχουμε μια πλήρη ανάθεση στις μεταβλητές. Η διάδοση περιορισμών θα ενισχύσει τη συνέπεια, επιβεβαιώνοντας ότι οι χρονικές σχέσεις είναι αποδεκτές σε κάθε στάδιο του προβλήματος.