Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

**Τεχνητή Νοημοσύνη**
Εργασία 3

Ονοματεπώνυμο: Μητρόπουλος Γεώργιος — Αριθμός Μητρώου: 1115202000128

---

# 1 Πρόβλημα 1:(The radio link frequency assignment problem - RLFA)

Σχεδιαστικές επιλογές κώδικα:
Ο κώδικας που παρέχεται εκτελείται με την εντολή:
python rlfa.py $\{file\}\{argorithm\}$,
όπου $\{file\}$=το αναγκωριστικό κάθε *.txt* αρχείου π.χ. $2-f25$ και
όπου $\{algorithm\}$=1,2,3,4

1 για $FC-BT$
2 για $MAC-BT$
3 για $FC-CBJ$
4 για *Min_conflicts*

Ο κώδικας που έχω υλοποιήσει είναι στο αρχείο *rlfa.py*. Χρησιμοποιήθηκαν τα αρχεία *csp.py*, *utils.py* και *search.py* από το *Github* του βιβλίου, τα οποία δεν τα έχω τροποποίησει. Συγκεκριμένα υλοποίησα μια κλάση *rlfa* που κληρονομεί από την *CSP* του *csp.py*. Η κλάση αυτή ορίζει τις μεταβλητές του προβλήματος (*read_variables*, τα πεδία τιμών (*make_domains*), τους γείτονες από κάθε μεταβλητή (*make_neighbors*) και χρησιμοποιεί μια συνάρτηση που ελέγχει για κάθε ανάθεση αν ικανοποιούνται οι περιορισμοί του προβλήματος ($f$).Αναλυτικότερες λεπτομέρειες υπάρχουν σε μορφή σχολίων εντος του κώδικα.Επίσης έχουν οριστεί κάποιες επιπλέον δομές που αποθηκεύουν πληροφορίες όπως *dictionary* για τα βάρη των περιορισμών και άλλα.
Η συνάρτηση *forwardchecking*, έχει παρθεί όπως είναι από το *csp.py* και έχουν προστεθεί μερικές επιπλέον γραμμές για την αύξηση των βαρών των *constraints* και την ανανέωση των *conflictssets* που αφορούν την υλοποιήση του $fc-cbj$ αλγορίθμου.
Παρομοίως για την συνάρτηση *mac*, όπου δίνεται σαν όρισμα ο αλγόριθμος $AC3$ στον όποιο έχουν προστεθεί η αύξηση των βαρών των *constraints* στην περίπτωση του $wipe-out$.
Η ευρετική συνάρτηση *dom_wdeg* ψάχνει για κάθε μεταβλητή που δεν της έχει ανατεθεί τιμή και βρίσκει για κάθε *unassigned* γείτονα της, τον τύπο $len(dom(var)/wdeg(var)$ όπως περιγράφεται στις διαφάνειες. Τέλος επιστρέφει την μεταβλητή που αντιστοιχεί στον μικτρότερο αριθμό του προαναφερόμενου τύπου.
Η συνάρτηση *backjumping_search* χρησιμοποιείται για την υλοποιήση του $fc-cbj$ αλγορίθμου, και ουσιαστικά είναι μία επέκταση της συνάρτησης *backtracking_search* του *csp.py.*, με την διαφορά ότι διαχειρίζεται τα *conflictssets* όταν υπάρχει αδιέξοδο για μία μεταβλητή ώστε να δουλέψει σωστά ο αλγόρθμος σύμφωνα

με τις διαφάνειες. Τέλος, βρίσκεται η $main$ όπου ανάλογα τα ορίσματα που δόθηκαν στην γραμμή εντολών εκτελεί τον αντίστοιχο αλγόρθμο και εκτυπώνει το αποτέλεσμα (αν υπάρχει) μάζι με τον χρόνο εκτέλεσης και τα συνολικά $assigns$ των μεταβλητών.

Ακολουθεί πίνακας με ενδεικτικές εκτελέσεις των αλγορίθμων.

| | fc-bt | time(sec) | assigns | mac-bt | time(sec) | assigns | fc-cbj | time(sec) | assigns | min_conflicts | time(sec) | assigns |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2-f24 | | 0.087 | 463 | | 0.12 | 302 | | 0.078 | 400 | | 128.7 | 100.200 |
| 2-f25 | | 27.5 | 135.709 | | 40.6 | 28.661 | | 7 | 30.524 | | 454.9 | 100.680 |
| 3-f10 | | 14.1 | 122.229 | | 0.83 | 766 | | 7.4 | 54.318 | | 279.6 | 100.400 |
| 3-f11 | | 234.6 | 1.814.304 | | 16.3 | 8.955 | | 22.6 | 154.669 | | 500 | xxx |
| 6-w2 | | 0.03 | 253 | | 0.0 | 42 | | 0.03 | 253 | | 500 | 100.200 |
| 7-w1-f4 | | 2.2 | 31.130 | | 0.2 | 479 | | 1 | 13.292 | | 500 | xxx |
| 7-w1-f5 | | 500 | 4.917.287 | | 12.9 | 8.999 | | 125.4 | 1.192.780 | | 500 | xxx |
| 8-f10 | | 500 | 2.601.836 | | 29.7 | 16.661 | | 500 | 2.631.862 | | 500 | xxx |
| 8-f11 | | 33.8 | 204.922 | | 25.5 | 17.749 | | 34 | 155.768 | | 500 | xxx |
| 11 | | 3.8 | 6.228 | | 4.3 | 2.954 | | 2.6 | 4.559 | | 500 | xxx |
| 14-f27 | | 500 | 1.400.959 | | 9.7 | 17.827 | | 325.3 | 1.029.978 | | 500 | xxx |
| 14-f28 | | 21.1 | 45.789 | | 32.1 | 42.461 | | 9.5 | 22.420 | | 500 | xxx |

| |
|---|
| Result |
| None |
| Timeout |

Όλοι οι αλγόριθμοι που βρίσκονται στον παραπάνω πίνακα εκτελέστηκαν με την βοήθεια της ευρετικής συνάρτησης $dom/wdeg$.
Παρατηρούμε λοιπόν πως μεταξύ των $fc-bt, fc-cbj$, ο πιο αποδοτικός και γρήγορος αλγόριθμος είναι ο $fc-cbj$ με 1 $timeout$ ενώ ο $fc-bt$ 3, κατι που ήταν αναμενόμενο αφού αυτός ο αλγόριθμος οταν έρθει σε αδιέξοδο ¨πηδάει' στην μεταβλητή που βρίσκεται πιο βαθιά στο σύνολο συγκρούσεων της μεταβλητής που βρέθηκε σε αδιέξοδο και αυτό εχει ως αποτέλεσμα την αποφυγή άχρηστων αναθέσεων τιμών σε μεταβλητές και την εκ νέου εξερεύνηση μεγάλων τμημάτων του χώρου αναζήτησης.
Αυτό φαίνεται και στον πινάκα καθώς εκτός απο την πιο γρήγορη εκτέλεση του φαίνεται οτι πραγματοποιεί λιγότερες αναθέσεις τιμών συγκριτικά με τον άλλο αλγόριθμο κάνοντας ετσι την εκτέλεση πολυ ταχυτερη.Ειδικά σε περιπτώσεις όπου ο αλγόριθμος $fc-bt$ φαίνεται αρκετά αργός και με παρα πολλές αναθέσεις, ο $fc-cbj$ είναι με διαφορά πιο αποδοτικός.
Συγκρίνοντας και τους 3 αλγορίθμους παρατηρούμε πως ο $mac-bt$ είναι με διαφορά ο πιο αποδοτικός καθώς ο αλγόριθμος $fc$ όπως γνωρίζουμε ελέγχει για κάθε ανάθεση τα πεδία τιμών απο τις γειτονικές μεταβλητές και κλαδεύει τις τιμές που δεν ειναι συνεπείς με την συγκεκριμένη ανάθεση ενω ο $mac$ εντοπίζει εγκαίρως τις ακμές του γράφου αναζήτησης που δεν ειναι συνεπείς, δηλαδή τις αναθέσεις που δεν προσφέρουν κάτι ως προς την λύση του προβλήματος.
Όσον αφορά τον αλγόρθμο $min\_conflicts$, βλέπουμε πως σε όλες τις περιπτώσεις η εκτέλεση διήρκησε πάνω απο 500 δευτερόλεπτα ($timeout$) και αυτό οφείλεται στο γεγονός οτι οι λύσεις των περισσότερων στιγμιοτύπων δεν είναι πυκνά κατανενημένες στον χώρο καταστάσεων κάθε στιγμιοτύπου.

## 2 Πρόβλημα 2: (Μοντελοποίηση με προβλήματα ικανοποίησης περιορισμών)

1. Ορισμός προβλήματος ικανοποίησης περιορισμών.
α) Μεταβλητές:
$X_{\alpha\delta}$ ο χρόνος από την αίθουσα προς τα δωμάτια,$X_{\delta\alpha}$ το αντίθετο,$X_{\xi}$ ο χρόνος ξεκούρασης και $X_{\alpha\chi\rho}$ τον

χρόνο για να πάει κανείς προς το χρηματοκιβώτιο. Αντίστοιχα $X_{\chi\rho\alpha}$ τον χρόνο επιστροφής προς την αίθουσα. Τέλος ορίζουμε $X_{\pi}$ τον χρόνο της παραβίασης του χρηματοκιβωτίου. Να σημειωθεί ότι κάθε μεταβλητή είναι ξεχωριστή για κάθε ύποπτο. Δηλαδή έχουμε π.χ. $X_{\alpha\delta 1}$ για τον κ. Γιάννη, $X_{\alpha\delta 2}$ για την κ. Μαρία κ.τ.λ.
β) Ένα πεδίο πιθανών τιμών το οποίο αντιστοιχεί στα λεπτά της ώρας για την κάθε μεταβλητή.
γ) Ένα σύνολο περιορισμών το οποίο ορίζει τα λεπτά που μπορούν να ανατεθούν σε κάθε μεταβλητή. Συγκεκριμένα: $5 \leq X_{\alpha\delta}$,$X_{\delta\alpha} \leq 10$ και $20 \leq X_{\chi\rho\alpha}$,$X_{\alpha\chi\rho} \leq 30$ και $45 \leq X_{\pi} \leq 90$. Επίσης ορίζεται ο χρόνος ολοκλήρωσης ομιλίας ως 9:30 για τον κύριο Γιάννη, 10:00 για την κυρία Μαρία και 10:30 για την κυρία Όλγα καθώς και τον χρόνο απονομής του βραβείου στις 11:00.

2. Ο αστυνόμος Σιεσπής συνέλαβε τον κύριο Γιάννη.
Εξήγηση: Αρχικά έχουμε ότι ο ελάχιστος χρόνος που χρειάζεται κανείς για να κλέψει το έπαθλο είναι 85 λεπτά. Στην συνάρτηση $X_{\alpha\chi\rho}$+ $X_{\pi}$+ $X_{\chi\rho\alpha}$, αν θέσουμε τις ελάχιστες τιμές στις μεταβλητές θα δούμε ότι έχουμε 85. Επίσης δεν υφίσταται το ενδεχόμενο κάποιος και να επέστρεψε σπίτι του και να έκλεψε το έπαθλο. Συνεπώς κάποιος είπε ψέματα.
Οπότε στην κυρία Όλγα, έχουμε ότι: αν θεωρήσουμε ότι είναι ένοχη χρειάζεται τουλάχιστον 85 λεπτά από τις 10:30 για να κλέψει το έπαθλο. Από την στιγμή που ήταν όμως στην απονομή δεν είχε τον απαραίτητο χρόνο. Άρα πράγματι γύρισε σπίτι της, οπότε θέτουμε τις κατάλληλες τιμές στις μεταβλητές $X_{\alpha\delta 3}$, $X_{\xi 3}$ και $X_{\delta\alpha 3}$ έχοντας ότι $X_{\alpha\delta 3} + X_{\xi 3} + X_{\delta\alpha 3} = 30$, μιας και αυτόν τον χρόνο είχε ελεύθερο στην διάθεση της η κ. Όλγα μέχρι την απονομή. Οι υπόλοιπες μεταβλητές δεν παίρνουν κάποια τιμή.
Η κυρία Μαρία επίσης δεν είναι ένοχη επειδή τελείωσε την ομιλία της στις 10:00 και μέχρι την απονομή δεν είχε 85 λεπτά για να κλέψει το έπαθλο. Οπότε θέτουμε όπως και πριν τις τιμές που χρειάζονται οι μεταβλητες $X_{\alpha\delta 2}$ , $X_{\xi 2}$ και $X_{\delta\alpha 2}$ έτσι ώστε $X_{\alpha\delta 2} + X_{\xi 2} + X_{\delta\alpha 2} = 60$ όπου αυτός ήταν ο ελεύθερος χρόνος της κ. Μαρίας.
Φτάνουμε λοιπόν στον κύριο Γιάννη ο οποίος τελείωσε την ομιλία του στις 9:30. Βλέπουμε ότι αν προσθέσουμε 85 λεπτά φτάνουμε στις 10:55. Εφόσον δεν υπάρχει άλλος ύποπτος βρίσκουμε ότι ο κύριος Γιάννης έκλεψε το έπαθλο. Οπότε θέτουμε τις απαραίτητες τιμές στις μεταβλητές $X_{\alpha\delta 1}$ , $X_{\xi 1}$ και $X_{\delta\alpha 1}$ έτσι ώστε $X_{\alpha\delta 1} + X_{\xi 1} + X_{\delta\alpha 1} = 90$ μιας και αυτός είναι ο χρόνος που έχει ο κύριος Γιάννης από την στιγμή που τελείωσε η ομιλία του μέχρι να παραβιάσει το χρηματοκιβώτιο και να επιστρέψει στην απονομή. Οι υπόλοιπες μεταβλητές δεν παίρνουν κάποια τιμή.

3. Μία μέθοδος διάδοσης περιορισμών είναι η χρήση του αλγορίθμου *ForwardChecking*. Η λογική είναι ότι στα σημεία όπου θέτουμε τιμές στις μεταβλητές από τις συναρτήσεις $X_{\alpha\delta} + X_{\xi} + X_{\delta\alpha}$ και $X_{\alpha\chi\rho} + X_{\pi} + X_{\chi\rho\alpha}$ , οταν θέσουμε μία τιμή ελέγχουμε τις υπόλοιπες μεταβλητές όπου δεν έχουν πάρει τιμές και διαγράφουμε τις τιμές που είναι ασυνεπείς με την προηγούμενη ανάθεση. Π.χ: Για την κ. Όλγα έχουμε την συνάρτηση $X_{\alpha\delta 3} + X_{\xi 3} + X_{\delta\alpha 3} = 30$. Αν θέσουμε στην $X_{\alpha\delta 3}$ την τιμη 10 και στην $X_{\xi 3}$ την τιμη 15, τότε για την μεταβλητή $X_{\delta\alpha 3}$ μας μένει μόνο η τιμή 5 επειδή οποιαδήποτε μεγαλύτερη τιμή θα βγάλει άθροισμα πάνω από 30. Οπότε σε αυτή την περίπτωση διαγράφουμε τις τιμές πάνω από 5 όταν θα χρειαστεί να θέσουμε τιμή στην μεταβλητή $X_{\delta\alpha 3}$.

## 3 Πρόβλημα 3: (Μοντελοποίηση με προβλήματα ικανοποίησης περιορισμών)

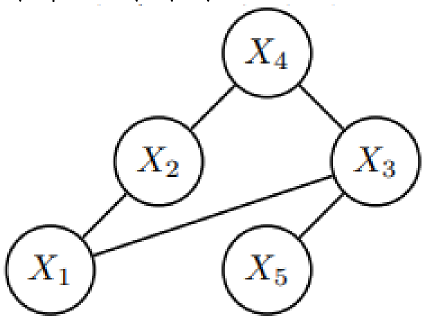
1.
Μεταβλητές: $V = \{Xi\}$, $i$=1,2,3,4,5

Πεδία μεταβλητών: $D = \{D4 = \{9, 11\}, Di = \{9, 10, 11\}\}, i = 1, 2, 3, 4, 5$

Περιορισμοί: $C = \{X1 > X3$,
$X5 < X3 < X4$,

$X2 \neq X1,$
$X2 \neq X4\}$

2.Γράφος Περιορισμών:




3. Αλγόριθμος συνέπειας τόξου $AC - 3$

- Αρχικά όλες οι ακμές ειναι ασυνεπείς.

- X1=9 $\implies D2 = \{10, 11\}, D3 = \{\} \implies$ , άρα η ακμή (X1,X3) μη συνεπής.

- X1=10 $\implies D2 = \{9, 11\}, D3 = \{9\} \implies$ πρέπει να εξεταστούν οι ακμές (X4,X2),(X4,X3),(X5,X3)
  (X4,X2):συνεπής
  (X4,X3): μη συνεπής $\implies D4 = \{11\} \implies$ πρέπει να εξεταστούν οι ακμές (X3,X4),(X2,X4)
  (X5,X3):μη συνεπής $\implies D5 = \{\}$

- X1=11 $\implies D2 = \{9, 10\}, D3 = \{9, 10\} \implies$ πρέπει να εξεταστούν οι ακμές (X4,X2),(X4,X3),(X5,X3)
  (X4,X2):συνεπής
  (X4,X3): μη συνεπής $\implies D4 = \{11\} \implies$ πρέπει να εξεταστούν οι ακμές (X3,X4),(X2,X4)
  (X5,X3): μη συνεπής $\implies D5 = \{9\} \implies$ πρέπει να εξεταστούν οι ακμές (X3,X5), (X4,X3)
  (X3,X4):συνεπής
  (X2,X4):συνεπής
  (X3,X5):μη συνεπής $\implies D3 = \{10\} \implies$ πρέπει να εξεταστούν οι ακμές (X5,X3),(X4,X3)
  (X4,X3):συνεπής
  (X5,X3):συνεπής

- X2=9 $\implies D1 = \{11\}, D4 = \{11\}$

Οπότε καταλήγουμε ότι:
X1=11,X2=9,X3=10,X4=11,X5=9