

Εργασία 3η Τεχνητής Νοημοσύνης

Ονοματεπώνυμο: Απόστολος Καρβέλας

A.M.: 1115201800312

Πρόβλημα 1:

Το πρόβλημα ικανοποίησης περιορισμών, όπως και οι αλγόριθμοι Forward checking και MAC υπάρχουν στον έτοιμο κώδικα που παραδίδεται στην εκφώνηση.

Το αρχείο περιέχει το pdf με το read_me και τις θεωρητικές απαντήσεις, το main.py που είναι ο χρήσιμος κώδικας της csp μαζί με τις νέες υλοποιήσεις και 2 αρχεία search.py και utils.py.

Για την υλοποίηση της ευρετικής συνάρτησης dom/wdeg που λειτουργεί ως dynamic variable ordering για όλους τους αλγορίθμους χρειάζεται να αποθηκεύουμε τα βάρη των περιορισμών στο αντίστοιχο dictionary και να τα αυξάνουμε όταν κάποιος γείτονας καταλήγει σε αδιέξοδο λόγω των περιορισμών και τις ήδη αντιστοιχισμένες τιμές. Οπότε, για την εύρεση της επόμενης μεταβλητής σε κάποιον αλγόριθμο καλούμε την συνάρτηση αυτή ή οποία υπολογίζει το νούμερο dom ως το πλήθος των τιμών του πεδίου ορισμού της μεταβλητής και το διαιρεί με το wdeg που είναι το άθροισμα των βαρών των περιορισμών που παίρνει μέρος η μεταβλητή αυτή. Τέλος, βρίσκει το μικρότερο κλάσμα και επιστρέφει την μεταβλητή με το κλάσμα αυτό.

Το πρόβλημα ικανοποίησης περιορισμών δέχεται ως παραμέτρους 4 δεδομένα που τα διαβάζουμε από τα αρχεία στον συμπιεσμένο φάκελο rlfar.rar. Συγκεκριμένα, οι παράμετροι είναι:

- **Variables:** Οι μεταβλητές υλοποιούνται σε λίστα από νούμερα τα οποία διαβάζονται από τα αρχεία με πρόθεμα var παίρνοντας υπόψιν μόνο την πρώτη στήλη.
- **Domains:** Για την εύρεση των domains αρχικά διαβάζουμε τα αρχεία με πρόθεμα dom και δημιουργούμε dictionary με κλειδί τον αριθμό του domain και τιμές το σύνολο τιμών που του αντιστοιχεί. Στην συνέχεια, ενώ έχουμε διαβάσει τις μεταβλητές και το νούμερο του domain τους, δημιουργούμε το dictionary domains με κλειδί τις μεταβλητές και τιμές το σύνολο τιμών του domain της.
- **Neighbors:** Είναι dictionary με κλειδί μια μεταβλητή και τιμή τις μεταβλητές που παίρνουν μέρος σε κάποιο περιορισμό μαζί της. Τα κλειδιά είναι συμμετρικά, δηλαδή για κάθε μεταβλητή υπάρχει ως κλειδί. Για την εύρεση των δεδομένων διαβάζει τα αρχεία με πρόθεμα ctr και τοποθετεί σε ένα dictionary με κλειδί την πρώτη στήλη και τιμή την δεύτερη και ταυτόχρονα με κλειδί την δεύτερη στήλη έχοντας σαν τιμή την πρώτη στήλη.

- **Constraints:** Η constraint είναι συνάρτηση που έχει σαν παραμέτρους 2 μεταβλητές και 2 τιμές μεταβλητών και επιστρέφει True όταν οι μεταβλητές παίρνουν τις αντίστοιχες τιμές ικανοποιούν κάθε περιορισμό. Η υλοποίηση αυτή γίνεται μέσω ενός dictionary που έχει για πρώτο κλειδί την πρώτη από τις 2 μεταβλητές του περιορισμού και ως δεύτερο, την άλλη μεταβλητή. Στην συνέχεια, βάζουμε για τιμή μια λίστα η οποία το πρώτο στοιχείο της είναι ο χαρακτήρας ισότητας ή μεγαλύτερου, το δεύτερο είναι ο αριθμός k και σαν τελευταίο στοιχείο το βάρος της που αρχικοποιείται με 1.

Πριν την εκτέλεση του προγράμματος για την εύρεση των πειραματικών τιμών δημιουργείται μια κλάση CSP με ορίσματα τα 4 δεδομένα που φτιάχνουμε διαβάζοντας τα αρχεία και εκτελεί την κύρια συνάρτηση `backtracking_search`. Η συνάρτηση αυτή, δέχεται ως ορίσματα το πρόβλημα ικανοποίησης περιορισμών, την ευρετική για το `variable ordering`, την ευρετική συνάρτηση για το `value ordering` και τέλος τον αλγόριθμο που θέλουμε να εκτελεσθεί. Στην συνέχεια εκτυπώνει τον χρόνο εκτέλεσης του αλγορίθμου, τον αριθμό των κόμβων που επισκέφτηκε, τις συγκρίσεις που έγιναν και αν είναι συνεπές ή μη (SAT/UNSAT).

Για τα αποτελέσματα εκτελέστηκε το πρόγραμμα για κάθε στιγμιότυπο του προβλήματος με την ευρετική `dom/wdeg` και την `unordered_domain_values` με κάθε αλγόριθμο `FC/MAC/MINCONFLICT/FC-CBJ`. Στην συνέχεια, για την εύρεση των πειραματικών τιμών έγιναν 1-5 εκτελέσεις και βρέθηκε ο μέσος όρος αυτών. Να σημειωθεί ότι, σε κάποια αρχεία που η εκτέλεση έπαιρνε πάνω από μια ώρα με τον αλγόριθμο `FC` θεωρήθηκε ως `UNSAT` επειδή δεν προλάβαινε να βγάλει αποτέλεσμα αλλά αν τελείωνε θα έβγαζε ίδιο αποτέλεσμα (SAT/UNSAT) με τον αλγόριθμο `MAC`. Στα αρχεία αυτά, οι στήλες `constraint` και `visited nodes` δείχνουν τα αντίστοιχα δεδομένα για τα πρώτα 5 λεπτά (300 δευτερόλεπτα) εκτέλεσης.

Τα αποτελέσματα αυτά είναι:

AVERAGE	FILE ID	METHOD	TIME (sec)	CONSTRAINT	VISITED NODES	SAT/UNSAT
	2-f24	FC	0.061	22374	236	SAT
	2-f24	MAC	0.375	221181	230	SAT
	2-f24	FC-CBJ	0.065	24110	238	SAT
	2-f24	MIN-CONFLICT	0.184	172738	250	UNSAT
	2-f25	FC	57.551	36481298	69400	UNSAT
	2-f25	MAC	199.134	147150966	48041	UNSAT
	2-f25	FC-CBJ	81.533	48955075	109118	UNSAT
	2-f25	MIN-CONFLICT	0.171	167882	250	UNSAT
	3-f10	FC	1.583	669160	1219	SAT
	3-f10	MAC	6.609	1953359	971	SAT
	3-f10	FC-CBJ	1.571	839733	1115	SAT
	3-f10	MIN-CONFLICT	0.4207	41211	450	UNSAT
	3-f11	FC	270.603	38962375	38903	UNSAT
	3-f11	MAC	100.241	53666687	8031	UNSAT
	3-f11	FC-CBJ	150.556	40702177	47299	UNSAT
	3-f11	MIN-CONFLICT	0.412	380786	450	UNSAT
	6-w2	FC	0.064	46258	61	UNSAT
	6-w2	MAC	0.412	426560	43	UNSAT
	6-w2	FC-CBJ	0.063	46395	61	UNSAT
	6-w2	MIN-CONFLICT	0.137	104144	250	UNSAT
	7-w1-f4	FC	1215.764	41877512	332659	SAT
	7-w1-f4	MAC	1.827	436587	451	SAT
	7-w1-f4	FC-CBJ	348.387	47491845	134782	SAT
	7-w1-f4	MIN-CONFLICT	0.158	96863	450	UNSAT

AVERAGE	FILE ID	METHOD	TIME (sec)	CONSTRAINT	VISITED NODES	SAT/UNSAT
	7-w1-f5	FC	>1h	8441534(300secs)	77312(300secs)	UNSAT
	7-w1-f5	MAC	139.688	100317809	12933	UNSAT
	7-w1-f5	FC-CBJ	>1h	13522837(300secs)	256468	UNSAT
	7-w1-f5	MIN-CONFLICT	0.165	96688	450	UNSAT
	8-f10	FC	>1h	8603073(300secs)	15539(300secs)	UNSAT
	8-f10	MAC	349.254	98708538	15812	SAT
	8-f10	FC-CBJ	>1h	14584537(300secs)	491257	UNSAT
	8-f10	MIN-CONFLICT	0.636	505838	730	UNSAT
	8-f11	FC	>1h	11490650(300secs)	15534(300secs)	UNSAT
	8-f11	MAC	46.811	12025848	2020	UNSAT
	8-f11	FC-CBJ	>1h	17356268(300secs)	51320	UNSAT
	8-f11	MIN-CONFLICT	0.689	500704	730	UNSAT
	11	FC	10.866	2712764	7194	SAT
	11	MAC	85.688	20017881	4577	SAT
	11	FC-CBJ	10.502	3428483	6972	SAT
	11	MIN-CONFLICT	0.778	611970	730	UNSAT
	14-f27	FC	>1h	2353311(300secs)	14839(300secs)	UNSAT
	14-f27	MAC	158.977	4369369	8172	SAT
	14-f27	FC-CBJ	>1h			UNSAT
	14-f27	MIN-CONFLICT	0.751	552656	966	UNSAT
	14-f28	FC	>1h	2737238(300secs)	10333(300secs)	UNSAT
	14-f28	MAC	176.13	8701280	5900	UNSAT
	14-f28	FC-CBJ	>1h			UNSAT
	14-f28	MIN-CONFLICT	0.778	546810	966	UNSAT

Οι στήλες με επικεφαλίδα FILE ID και METHOD είναι τα αρχεία με τα στιγμιότυπα του προβλήματος και τον αλγόριθμο που χρησιμοποιήσαμε αντίστοιχα.

Για την ακριβή εξέταση των αλγορίθμων χρησιμοποιήθηκαν 4 κριτήρια σύγκρισης:

- **TIME(sec):** Ο χρόνος εκτέλεσης του αλγορίθμου σε δευτερόλεπτα.
- **CONSTRAINT:** Οι συγκρίσεις που θα κάνει ο αλγόριθμος για 2 μεταβλητές.
- **VISITED NODES:** Οι κόμβοι που επισκέφτηκε ο αλγόριθμος.
- **SAT/UNSAT:** Είναι SAT όταν βρεθεί λύση, αλλιώς UNSAT.

Κατά την εξέταση των αποτελεσμάτων παρατηρούμε κάποια ενδιαφέρον μοτίβα. Αρχικά, ο FC είναι πιο απλοϊκός αλγόριθμος από τους άλλους οπότε χρειάζεται λιγότερες συγκρίσεις για την εντόπιση και επίσκεψη ενός κόμβου. Το γεγονός αυτό είναι θετικό για πιο εύκολα στιγμιότυπα αφού έχει λιγότερους συνολικούς κόμβους για να εξετάσει και κάθε κόμβος παίρνει σχετικά λίγο χρόνο, όμως για πιο πολύπλοκα αρχεία θα έχει τεράστιο χρόνο εκτέλεσης αφού θα έχει και πολύ περισσότερους κόμβους. Ο MAC ως πιο περίπλοκος από την FC, χρειάζεται περισσότερες συγκρίσεις για την επίσκεψη ενός κόμβου αλλά βρίσκει την λύση εξετάζοντας σχετικά λίγους κόμβους. Αυτό δουλεύει υπέρ σε μεγάλα-πολύπλοκα αρχεία αφού θα εξετάσει πολύ λιγότερους κόμβους από τον FC αλλά σε απλοϊκά αρχεία αργεί περισσότερο από τους άλλους για τον λόγο ότι χρειάζεται πολλές συγκρίσεις. Η FC-CBJ έχει σαν κύριο αλγόριθμο την FC απλά με την προσθήκη του backjump. Αυτό σημαίνει ότι όπως και ο MAC θα κάνει περισσότερες συγκρίσεις για την επίσκεψη ενός κόμβου αλλά στο τέλος θα επισκέπτεται λιγότερους.

Τώρα για τον αλγόριθμο min conflict παρατηρούμε διαφορετικό μοτίβο. Για αρχή, ο αλγόριθμος αυτός θυμίζει άπληστο οπότε θεωρητικά δουλεύει γρήγορα θα μικρότερα αρχεία. Στο πρόγραμμα όμως, έχοντας max_steps να ισούται με 50 δεν προλαβαίνει να βρει κάποια λύση και επιστρέφει UNSAT. Το γεγονός αυτό κάνει την min conflict έναν πολύ γρήγορο αλλά απλοϊκό αλγόριθμο οπότε ισχύουν οι παρατηρήσεις που κάναμε και για τον FC με συγκρίσεις τους αλγορίθμους MAC/FC-CBJ

Πρόβλημα 2:

Το πρόβλημα ικανοποίησης περιορισμών ορίζεται από:

- **Πρόβλημα:** Το πρόβλημα ορίζεται ως την τοποθέτηση των επίπλων στον δωμάτιο.
- **Μεταβλητές:** Τα έπιπλα και οι διαστάσεις του καθενός.
- **Σύνολο τιμών:** Οι διαστάσεις του δωματίου
- **Περιορισμοί:**
 1. Τα έπιπλα δεν εφάπτονται μεταξύ τους.
 2. Τα έπιπλα δεν πρέπει να πατάνε το ένα πάνω στο άλλο.
 3. Το γραφείο πρέπει να βρίσκεται δίπλα σε κάποιο φως.
 4. Τα έπιπλα δεν μπορούν να βρίσκονται κοντά στην πόρτα.
 5. Κάθε έπιπλο πρέπει να βρίσκεται μέσα στα όρια του δωματίου.

Το πρόβλημα είναι συνεπές αν θεωρήσουμε ότι η μπαλκονόπορτα εκπέμπει τουλάχιστον 160εκ πλάτος και 80εκ μήκος φωτός (διαστάσεις του γραφείου).

Μια από τις λύσεις είναι η ακόλουθη (σωστή αναλογία διάσταση-δωμάτιο):



Πρόβλημα 3:

1.

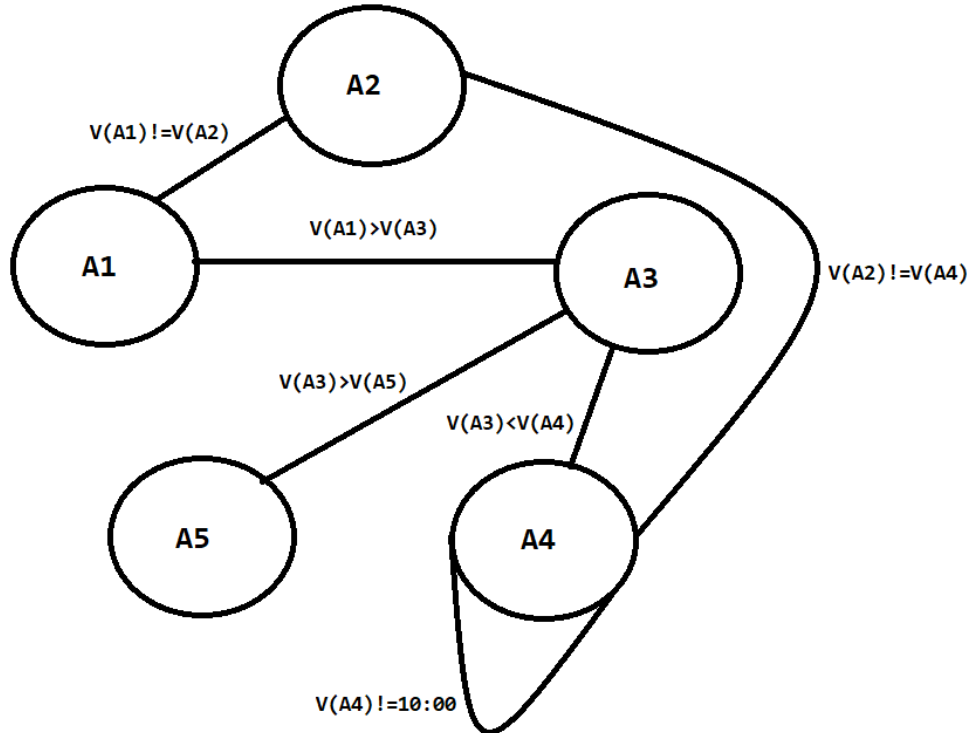
Το πρόβλημα ικανοποίησης περιορισμών ορίζεται από:

- **Πρόβλημα:** Χρονοπρογραμματισμός ενεργειών .
- **Μεταβλητές:** Ενέργειες A1,...,A5(και χρόνος εκτέλεσης).
- **Σύνολο τιμών:** Ώρες που μπορούν να αρχίσουν (9:00, 10:00 ή 11:00)
- **Περιορισμοί:**
 1. Η A1 πρέπει να αρχίσει μετά την A3.
 2. Η A3 πρέπει να αρχίσει πριν την A4.
 3. Η A3 πρέπει να αρχίσει μετά την A5.
 4. Η A2 δεν μπορεί να εκτελείται την ίδια ώρα με την A1.
 5. Η A2 δεν μπορεί να εκτελείται την ίδια ώρα με την A4.

6. Η A4 δεν μπορεί να αρχίσει στις 10:00.

2.

Γράφος περιορισμών για το πρόβλημα:



3.

Μεταβλητές:

$A1 = \{9.00, 10.00, 11.00\}$,

$A2 = \{9.00, 10.00, 11.00\}$,

$A3 = \{9.00, 10.00, 11.00\}$,

$A4 = \{9.00, 10.00, 11.00\}$,

$A5 = \{9.00, 10.00, 11.00\}$

Arcs:

$A1 > A3$, $A3 < A1$, $A3 < A4$, $A4 > A3$, $A3 > A5$, $A5 < A3$, $A2 \neq A1$, $A1 \neq A2$, $A2 \neq A4$, $A4 \neq A2$, $A4 \neq 10$

Agenda:

~~$A1 < A3$~~ , βγάζουμε το 9 από το σύνολο τιμών της $A1$

~~$A3 < A1$~~ , βγάζουμε το 11 από το σύνολο τιμών της $A3$

~~$A3 < A4$~~ ,

~~$A4 > A3$~~ , βγάζουμε το 9 από το σύνολο τιμών της $A4$

~~$A3 > A5$~~ , βγάζουμε το 9 από το σύνολο τιμών της $A3$

~~$A5 < A3$~~ , βγάζουμε το 10,11 από το σύνολο τιμών της $A5$

~~$A2 \neq A1$~~ ,

~~$A1 \neq A2$~~ ,

~~$A2 \neq A4$~~ ,

~~$A4 \neq A2$~~ ,

~~$A4 \neq 10$~~ βγάζουμε το 10 από το σύνολο τιμών της $A4$

~~$A1 > A3$~~ , βγάζουμε το 10 από το σύνολο τιμών της $A1$

~~$A3 < A4$~~ ,

~~$A4 > A3$~~ ,

~~$A3 > A5$~~ ,

~~$A2 \neq A4$~~ , βγάζουμε το 11 από το σύνολο τιμών της $A2$

~~$A3 < A1$~~ ,

~~$A2 \neq A1$~~ ,

~~$A1 \neq A2$~~ ,

~~$A4 \neq A2$~~ ,

Οπότε τελικό αποτέλεσμα είναι:

$A1 = \{11.00\}$,

$A2 = \{9.00, 10.00\}$,

$A3 = \{10.00\}$,

$A4 = \{11.00\}$,

$A5 = \{9.00\}$.

