

# Τεχνητή Νοημοσύνη

1115201800006\_project3.pdf

Αναγνωστόπουλος Αθανάσιος

AM: 1115201800006

## Πρόβλημα 1

Αρχικά βασίστηκα στο αρχείο `csp.py` από το Github (<https://github.com/aimacode/aima-python/blob/master/csp.py>). Επίσης χρειάστηκαν και τα αρχεία `search.py` και `utils.py`. τα οποία γίνονται `import` από το `csp.py`.

Όπως φαίνεται και στην αρχή της κλάσης CSP στα σχόλια χρειαζόμαστε μια λίστα από `variables` (`varList`) , ένα `dictionary` από `domains` (`domDict`), ένα ακόμη `dictionary` από `neighbors` (`neighDict`) και τέλος μια συνάρτηση `constraints` (`constraints`) για την σωστή λειτουργία της κλάσης.

### Δημιουργία `variables list`

Για την δημιουργία της `varList` διαβάζω τα αρχεία με πρόθεμα `var`, στα οποία η πρώτη γραμμή δείχνει το πόσες μεταβλητές υπάρχουν στο αρχείο και κάθε μια από τις επόμενες στήλες έχει 2 αριθμούς. Ο πρώτος είναι η μεταβλητή και ο δεύτερος το `domain` της μεταβλητής αυτής. Στην λίστα των μεταβλητών αποθηκεύουμε

μόνο τον πρώτο αριθμό κάθε γραμμής διαβάζοντας το αρχείο γραμμή-γραμμή και προσπερνώντας τον δεύτερο αριθμό (domain).

Σημείωση: Το κάθε domain το αποθηκεύω σε μία βοηθητική λίστα domainsList για την διευκόλυνση της δημιουργίας του domDict ώστε να μην χρειάζεται στην συνέχεια να ξαναδιαβάσω το αρχείο var.

### **Δημιουργία domains dictionary**

Για την δημιουργία του domDict διαβάζω τα αρχεία με πρόθεμα dom, στα οποία ο αριθμός στην πρώτη γραμμή δείχνει το πόσα domains περιέχει το αρχείο. Στις επόμενες γραμμές ο πρώτος αριθμός είναι ο αριθμός του domain, ο δεύτερος δείχνει το πλήθος των τιμών που έχει και μετά ακολουθούν οι τιμές. Για την δημιουργία του domDict ελέγχω για ποιο domain αναφέρεται κάθε μεταβλητή (με την βοήθεια του domainsList) και στην συνέχεια προσθέτω στο dictionary ένα νέο στοιχείο με key την μεταβλητή και τιμή όλες τις τιμές του συγκεκριμένου domain. Τέλος αποθηκεύω

### **Δημιουργία neighbors dictionary**

Για την δημιουργία του neighDict εφόσον έχω διαβάσει το αρχείο με πρόθεμα ctr και έχω το dictionary με τα constraints. Έτσι διατρέχοντας το dictionary με τα constraints αποθηκεύω σε ένα νέο dictionary κάθε μεταβλητή ως key και σαν τιμές όσες μεταβλητές συμμετέχουν μαζί με την μεταβλητή key σε κάποιον περιορισμό.

## **Δημιουργία constraints dictionary**

Για την δημιουργία του constraintsDict διαβάζω τα περιεχόμενα του αρχείου και τα αποθηκεύω στο βοηθητικό dictionary consDict όπου αποθηκεύω ως key ένα tuple με τις δύο μεταβλητές και ως τιμή ένα tuple με τον τελεστή '=' ή '>' και τον αριθμό. Στη συνέχεια δημιουργώ το σωστό dictionary για τα constraints όπου έχει δύο κλειδιά (import defaultdict) την μεταβλητη1 και την μεταβλητη2 και ένα tuple με 3 στοιχεία ως τιμή όπου το 1<sup>ο</sup> στοιχείο είναι ο τελεστής, το δεύτερο η τιμή και το 3<sup>ο</sup> το βάρος που θα χρειαστεί για τον υπολογισμό του dom/wdeg (τα βάρη αρχικοποιούνται με 1).

## **Δημιουργία συνάρτησης constraints**

Η συνάρτηση constraints παίρνει ως όρισμα δύο μεταβλητές A και B και δύο τιμές για τις μεταβλητές αυτές a και b. Η συνάρτηση ελέγχει αν επαληθεύεται ο περιορισμός για τις συγκεκριμένες μεταβλητές και τιμές που έχουν δοθεί και επιστρέφει True ή False ανάλογα με το αποτέλεσμα.

## **Dynamic Variable Ordering**

Με την χρήση της ευρετικής dom/wdeg δεν εισάγουμε τις μεταβλητές με την σειρά με αποτέλεσμα το πρόγραμμα να εκτελείται σε πολύ μικρότερο χρόνο. Στην συνάρτηση AC3 και forward\_checking (από το αρχείο CSP) αυξάνουμε τα βάρη όταν ένας γείτονας προκαλεί αδιέξοδο. Το dom υπολογίζεται ως το πλήθος των τιμών ενός συγκεκριμένου domain και το wdeg είναι το

άθροισμα από τα βάρη των περιορισμών. Τέλος η συνάρτηση επιστρέφει την μεταβλητή με το μικρότερο dom / wdeg.

## Αποτελέσματα

Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT	2 - f24	Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT
2-F24 - FC	0.178666114	22374	236	SAT	2-F24 - Min Conflicts	0.075275182	141412	240	UNSAT	
	0.148080825	22374	236	SAT		0.123681306	136806	240	UNSAT	
	0.1479386209	22374	236	SAT		Gia max_steps = 40	0.075340986	142258	240	UNSAT
	0.149932861	22374	236	SAT			0.075318	140182	240	UNSAT
	0.149183273	22374	236	SAT			0.075948238	140800	240	UNSAT
Mesos Oros	0.15475997				Mesos Oros	0.0851119024				
2-F24 - MAC	0.2295510768	165229	229	SAT	2-F24 - FC-CBJ	Mesos Oros	0.1506829	24110	238	
	0.231730699	165229	229	SAT						
	0.2317254543	165229	229	SAT						
	0.23599219	165229	229	SAT						
	0.231652736	165229	229	SAT						
Mesos Oros	0.232130431									
Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT	2 - f25	Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT
2-F25 - FC	8,717866611	36481298	69400	UNSAT	2-F25 - Min Conflicts	Mesos Oros	0.072213411	138658	240	UNSAT
2-F25 - MAC	155.0216219	103496885	53902	UNSAT	2-F24 - FC-CBJ	Mesos Oros	137,964627	48955075	109118	UNSAT
Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT	3 - f10	Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT
3-F10 - FC	4.151631	669160	1219	SAT	3-F10 - Min Conflicts	Mesos Oros	0.1815731	322910	440	UNSAT
3-F10 - MAC	3.4503192	1334713	898	SAT	3-F10 - FC-CBJ	Mesos Oros	4.146712	839733	1115	SAT
Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT	3 - f11	Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT
3-F11 - FC	166.10537	38962375	38903	UNSAT	3-F11 - Min Conflicts	Mesos Oros	0.1829395	322163	440	UNSAT
3-F11 - MAC	51.94022	26048481	8555	UNSAT	3-F11 - FC-CBJ	Mesos Oros	171,1817138	39702915	77806	UNSAT

Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT	6 - w2	Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT
6-w2 - FC						6-w2 - Min Conflicts				
Mesos Oros	0.086850881	46258	61	UNSAT		Mesos Oros Gia max_steps = 40	0.0580935478	89518	240	UNSAT
6-w2 - MAC						6-w2 - FC-CBJ				
Mesos Oros	0.10565590	93186	45	UNSAT		Mesos Oros	0.084309	46518	122	UNSAT
Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT	7 - w1 - f4	Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT
7-w1-f4- FC						7-w1-f4 - Min Conflicts				
Mesos Oros	783,6921	41877512	332659	SAT		Mesos Oros Gia max_steps = 40	0.06974720	82046	440	UNSAT
7-w1-f4 - MAC						7-w1-f4 - FC-CBJ				
Mesos Oros	1,0039646	278046	447	SAT		Mesos Oros Ta αποτελέσματα είναι για 5	14937845	247818		SAT
Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT	7 - w1 - f5	Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT
7-w1-f5- FC						7-w1-f5 - Min Conflicts				
Mesos Oros	Ta αποτελέσματα είναι για 1	26534096	254908	UNSAT		Mesos Oros Gia max_steps = 40	0.068414211	81451	440	UNSAT
7-w1-f5 - MAC						7-w1-f5 - FC-CBJ				
Mesos Oros	61,178055	47702013	12150	UNSAT		Mesos Oros Ta αποτελέσματα είναι για 5	13526837	256468		UNSAT
Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT	8 - f10	Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT
8-f10 - FC						8-f10 - Min Conflicts				
Mesos Oros	Ta αποτελέσματα είναι για 1	42162492	53650	SAT		Mesos Oros Gia max_steps = 40	0.27659	424114	720	UNSAT
8-f10 - MAC						8-f10 - FC-CBJ				
Mesos Oros	154,147	29561768	13133	SAT		Mesos Oros Ta αποτελέσματα είναι για 5	14539277	51299		SAT

Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT	8 - f11	Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT
8-f11 - FC						8-f11 - Min Conflicts				
Mesos Oros	Ta αποτελέσματα είναι για 1	33875439	48657	UNSAT		Mesos Oros	0.2662210	422879	720	UNSAT
						Gia max_steps = 40				
8-f11 - MAC						8-f11 - FC-CBJ				
Mesos Oros	26,524252	4712780	2041	UNSAT		Mesos Oros	Ta αποτελέσματα είναι για 5	18844255	50407	UNSAT
Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT	11	Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT
11 - FC						11 - Min Conflicts				
Mesos Oros	78,9137	2712764	7194	SAT		Mesos Oros	0.3085889	517132	720	UNSAT
						Gia max_steps = 40				
11 - MAC						11 - FC-CBJ				
Mesos Oros	48,283732	8951737	4374	SAT		Mesos Oros	74,020795	2813145	14388	SAT
Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT	14 - f27	Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT
14 - f27 - FC						14 - f27 - Min Conflicts				
Mesos Oros	Ta αποτελέσματα είναι για 1	6210223	52819	SAT		Mesos Oros	0.2758305	457611	956	UNSAT
						Gia max_steps = 40				
14 - f27 - MAC						14 - f27 - FC-CBJ				
Mesos Oros	47,40151619	1516443	3493	SAT		Mesos Oros	Ta αποτελέσματα είναι για 5	3251228	43880	UNSAT
Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT	14 - f28	Instance ID - Method	Time (sec)	Constraint Checks	Visited nodes	SAT/UNSAT
14 - f28 - FC						14 - f28 - Min Conflicts				
Mesos Oros	Ta αποτελέσματα είναι για 1	5563739	32293	UNSAT		Mesos Oros	0.28009486	452608	956	UNSAT
						Gia max_steps = 40				
14 - f28 - MAC						14 - f28 - FC-CBJ				
Mesos Oros	82,5518419	3960838	4245	UNSAT		Mesos Oros	Ta αποτελέσματα είναι για 5	3621266	31936	UNSAT

Σημείωση: Τα αποτελέσματα είναι χωρίς τον αλγόριθμο lcn για variable ordering αφού δεν έχει σημαντική διαφορά στους χρόνους.

Για τον αλγόριθμο FC παρατηρώ πως ελέγχει λιγότερα constraints ανά node από τον αλγόριθμο mac αν και επισκέπτεται περισσότερα nodes, αυτό συμβαίνει διότι ο mac απορρίπτει τις περιπτώσεις που οδηγούν σε αδιέξοδο νωρίτερα από τον αλγόριθμο forward\_checking.

Ακόμα παρατηρώ πως για προγράμματα μεγαλύτερου μεγέθους ο mac είναι πιο γρήγορος από τον αλγόριθμο FC. Για το γεγονός αυτό ευθύνεται το συγκριτικά υψηλότερο κόστος των nodes σε σχέση με αυτό των περιορισμών.

Για τον αλγόριθμο min\_conflicts παρατηρώ πως σε κάθε περίπτωση το πρόβλημα είναι UNSAT ακόμα και για αρχεία για τα οποία υπάρχει λύση. Επίσης δουλεύει μόνο σε μικρά προβλήματα γιατί βρίσκει μία καλή τοπική λύση που οδηγεί σε αδιέξοδα (άπληστος αλγόριθμος).

Σημείωση: Το έτρεξα με max\_steps = 40 γιατί για max\_steps = 100.000 ο αλγόριθμος έπαιρνε πολύ χρόνο χωρίς να βγάζει αποτελέσματα.

Για τον αλγόριθμο FC - CBJ παρατηρώ πως είναι πιο πολύπλοκος και γρηγορότερος από τον αλγόριθμο FC, αλλά όχι με σημαντική διαφορά χρόνου. Ακόμα επισκεπτόμαστε λιγότερα nodes από τον FC αν και ελέγχει περισσότερα constraints.

Τέλος ο αλγόριθμος μας είναι σημαντικά πιο γρήγορος από τον αλγόριθμο FC – CBJ ειδικά όσο μεγαλύτερο είναι το πρόβλημα.

## Πρόβλημα 2

Σχετικά με το πρόβλημα τοποθέτησης των επίπλων:

### Περιορισμοί (constraints):

- A. Τα έπιπλα δεν πρέπει να εφάπτονται.
- B. Τα έπιπλα δεν πρέπει να πατάνε το ένα πάνω στο άλλο.
- Γ. Το γραφείο θα πρέπει να βρίσκεται κοντά στην πηγή φωτός (μπαλκονόπορτα).
- Δ. Θα πρέπει να υπάρχει χώρος ώστε να ανοίγει η πόρτα μετά την τοποθέτηση των επίπλων.
- Ε. Τα έπιπλα θα πρέπει να βρίσκονται μέσα στις διαστάσεις του δωματίου.

### Μεταβλητές (variables):

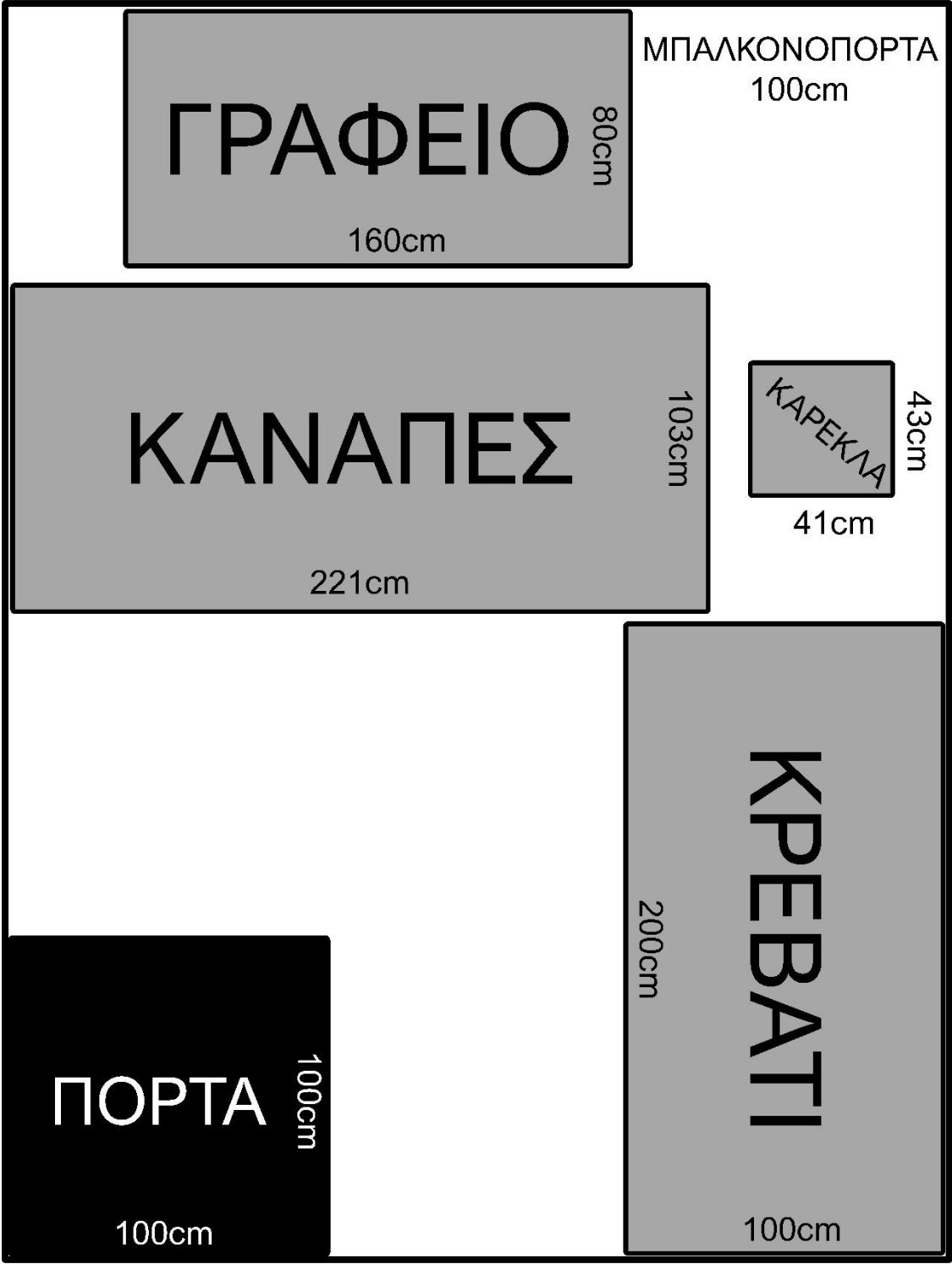
Οι διαστάσεις κάθε επίπλου {γραφείο, καναπές, κρεβάτι, καρέκλα}.

**Domains:**

Όλες οι θέσεις του δωματίου που μπορούν να τοποθετηθούν έπιπλα.

Επίσης υποθέτω πως σε αυτή την θέση η μπαλκονόπορτα φωτίζει το γραφείο.

Στο σχήμα που ακολουθεί είναι το scale.





## Πρόβλημα 3

### 1.

Σχετικά με το πρόβλημα χρονοπρογραμματισμού:

#### Περιορισμοί (constraints):

- A. Η A1 πρέπει να αρχίσει μετά την A3.
- B. Η A3 πρέπει να αρχίσει πριν την A4.
- Γ. Η A3 πρέπει να αρχίσει μετά την A5.
- Δ. Η A2 δεν μπορεί να εκτελείται την ίδια ώρα με την A1.
- Ε. Η A2 δεν μπορεί να εκτελείται την ίδια ώρα με την A4.
- Ζ. Η A4 δεν μπορεί να αρχίσει στις 10:00.

#### Μεταβλητές (variables):

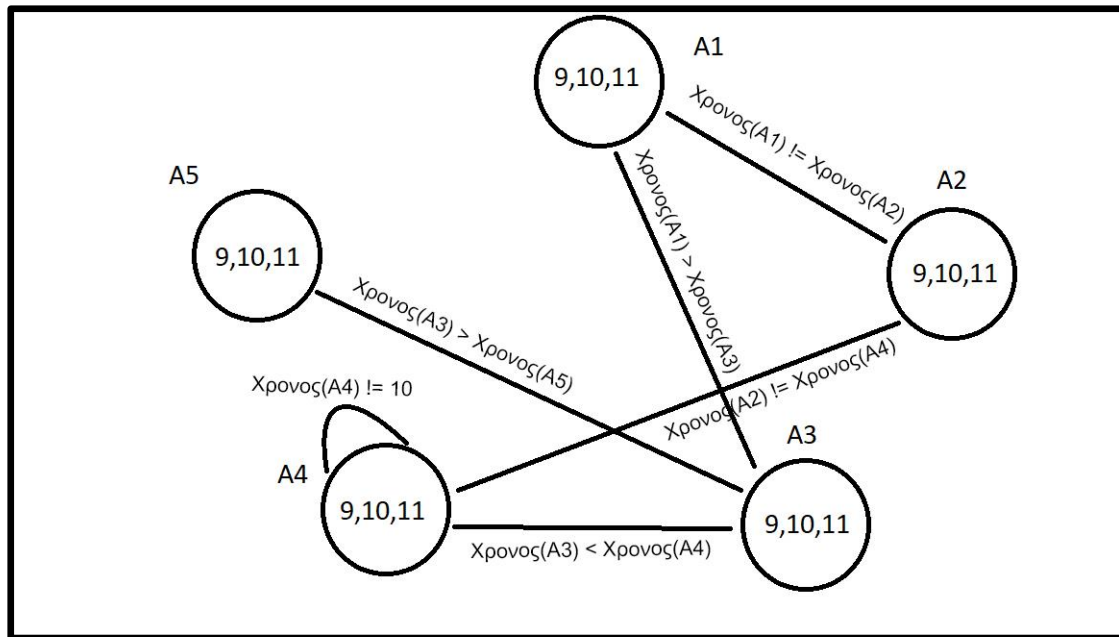
Κάθε μια από τις ενέργειες {A1, A2, A3, A4, A5}.

#### Domains:

Είναι ο χρόνος που μπορεί να αρχίσει κάθε ενέργεια, δηλαδή {9:00, 10:00, 11:00}.

### 2.

Γράφος περιορισμών:



3.

(επόμενη σελίδα)

	Agents	Arcs
$A1 = 1, 0, 11$	$A1 \rightarrow A3$	$A1 > A3$
$A2 = 9, 10, 0$	<del><math>A3 \leftarrow A1</math></del>	$A3 < A1$
$A3 = 0, 10, 0$	$A3 \leftarrow A4$	$A3 < A4$
$A4 = 0, 10, 11$	$A3 \rightarrow A5$	$A4 > A3$
$A5 = 9, 10, 0$	<del><math>A4 \rightarrow A3</math></del>	$A3 > A5$
	<del><math>A3 \rightarrow A5</math></del>	$A5 < A3$
	<del><math>A5 \leftarrow A3</math></del>	$A2 \neq A1$
	<del><math>A1 \neq A1</math></del>	<del><math>A1 \neq A4</math></del>
	<del><math>A1 \neq A2</math></del>	$A1 \neq A2$
	<del><math>A2 \neq A4</math></del>	$A2 \neq A4$
	<del><math>A4 \neq A2</math></del>	$A4 \neq A2$
	<del><math>A4 \neq 10</math></del>	$A4 \neq 10$
	<del><math>A1 \rightarrow A3</math></del>	
	<del><math>A3 \leftarrow A4</math></del>	
	<del><math>A4 \rightarrow A3</math></del>	
	<del><math>A3 \rightarrow A5</math></del>	
	<del><math>A1 \neq A4</math></del>	
	<del><math>A3 &lt; A1</math></del>	
	<del><math>A2 \neq A1</math></del>	