

Μεταπτυχιακό Πρόγραμμα Εφαρμοσμένης Πληροφορικής

Προχωρημένη Τεχνητή Νοημοσύνη

Εργασία 2

Προγραμματισμός με Περιορισμούς
Χειμερινό Εξάμηνο 2017-2018

Άσκηση 1: *Busy CPU*

Ένας server πρέπει να εκτελέσει μια σειρά από εργασίες:

- εργασία A, με cpu load 60, διάρκειας 2 ωρών
- εργασία B, με cpu load 30, διάρκειας 1 ωρών
- εργασία C, με cpu load 50, διάρκειας 2 ωρών
- εργασία D, με cpu load 40, διάρκειας 5 ωρών

Αν όλες οι εργασίες μπορούν να ξεκινήσουν την χρονική στιγμή 0, να γράψετε ένα πρόγραμμα σε MiniZinc το οποίο να βρίσκει τους χρόνους έναρξης κάθε εργασίας, ελαχιστοποιώντας το συνολικό χρόνο εκτέλεσης των εργασιών. Για παράδειγμα: Για παράδειγμα:

```
start = [2, 4, 0, 0];  
totalTime = 5;  
-----  
=====
```

Τα δεδομένα θα τα μοντελοποιήσετε εσείς στο αρχείο `cpu.mzn` που θα βρείτε στο `compus`.

Άσκηση 2 - Χρονοπρογραμματισμός Εργασιών Συνεργείου

Σε ένα συνεργείο υπάρχουν κάποια αυτοκίνητα προς επισκευή. Κάθε αυτοκίνητο απαιτεί το ακόλουθο σύνολο εργασιών:

- Ηλεκτρολογικές Εργασίες
- Μηχανολογικές Εργασίες
- Πλύσιμο (αυτόματο)

Για κάθε εργασία υπάρχει οι αντίστοιχοι τεχνικοί (Ηλεκτρολόγοι, Μηχανικοί), εκτός από το πλύσιμο που είναι αυτόματο. Ισχύουν τα ακόλουθα:

1. Κάθε **τεχνικός** μπορεί να εργαστεί σε **ένα μόνο αυτοκίνητο** σε κάθε χρονική στιγμή και σε κάθε αυτοκίνητο εργάζεται **μόνο ένας τεχνικός**.
2. Σε **κάθε** αυτοκίνητο μπορεί να γίνει **μόνο μια εργασία** κάθε φορά.
3. Υπάρχουν **3 ηλεκτρολόγοι** και **2 μηχανικοί**.
4. Οι μηχανολογικές και ηλεκτρολογικές εργασίες είναι ανεξάρτητες μεταξύ τους και μπορούν να εκτελεστούν με **οποιαδήποτε** σειρά.
5. Το πλύσιμο που πρέπει να γίνει **αφότου έχουν τελειώσει** οι άλλες δύο εργασίες.
6. Η αυτόματη μηχανή πλυσίματος μπορεί να εξυπηρετήσει **2 αυτοκίνητα κάθε φορά**.
7. Η εργάσιμη ημέρα έχει **8 ώρες**.

Τα αυτοκίνητα στο τρέχον πρόβλημα είναι έξι, και ο χρόνος που απαιτείται για να γίνουν οι αντίστοιχες εργασίες δίνεται από τους ακόλουθους πίνακες:

- **elecTime** ο χρόνος σε λεπτά που απαιτείται για ηλεκτρολογικές εργασίες, για το κάθε αυτοκίνητο,
- **mechTime**, ο χρόνος σε λεπτά που απαιτείται για μηχανολογικές εργασίες,
- **washTime**, ο χρόνος σε λεπτά που απαιτείται για πλύσιμο.

Η αρχική λίστα των αυτοκινήτων είναι η ακόλουθη:

```
array[CARS] of int: elecTime = [210,90,200,125,115,15];  
array[CARS] of int: mechTime = [130,150,120,80,120,135];  
array[CARS] of int: washTime = [20,10,10,10,20,20];
```

Να γράψετε ένα πρόγραμμα σε MiniZinc, το θα βρίσκει τους χρόνους έναρξης των ηλεκτρολογικών (startE), μηχανολογικών εργασιών (startM) και του πλυσίματος (startW) ώστε να ελαχιστοποιείται ο συνολικός χρόνος επισκευών (makespan). Για παράδειγμα:

```
makespan = 380;  
startE = [130, 150, 0, 0, 0, 115];  
startM = [0, 0, 250, 285, 130, 150];  
startW = [340, 240, 370, 365, 250, 285];  
-----  
=====
```

Τα δεδομένα βρίσκονται στο αρχείο carshop.mzn στο compus.

Άσκηση 3 - Η Κατασκευαστική Εταιρεία

Μια κατασκευαστική εταιρεία πρέπει να κατασκευάσει ένα πολυόροφο κτίριο. Οι εργασίες που πρέπει να γίνουν σε κάθε όροφο είναι οι ακόλουθες:

- κατασκευή του ορόφου (μπετά),
- ηλεκτρολογική εγκατάσταση,
- υδραυλική εγκατάσταση, και οι
- λοιπές εργασίες.

Για κάθε όροφο η εταιρεία έχει υπολογίσει τις μέρες και τον αριθμό των εργαζομένων που απαιτούνται. Οι ανάγκες καταγράφονται στους ακόλουθους πίνακες.

Το σύνολο των ορόφων που θα κατασκευαστούν:

```
enum FLOOR = {a1,a2,a3,a4,a5,a6,a7};
```

Εργαζόμενοι που απαιτούνται για κάθε όροφο (buildWrks) και διάρκεια κατασκευής του ορόφου (buildDur). Προφανώς, ο όροφος a1 απαιτεί 12 εργαζόμενους για 12 ημέρες, ενώ ο a2 12 εργαζόμενους για 5 ημέρες:

```
buildWrks = [12,12,10,11,12,14,14];  
buildDur = [12,5,10,5,5,10,5];
```

Με παρόμοιο τρόπο ορίζονται οι κατασκευές ανά όροφο για τα ηλεκτρολογικά:

```
elecWrks = [10,5,7,3,3,2,1];  
elecDur = [10,30,20,25,40,15,25];
```

για τα υδραυλικά,

```
plumbingWrks = [4,3,2,5,3,4,3];  
plumbingDur = [8,11,18,14,14,13,9];
```

και για τις λοιπές εργασίες.

```
finishWrks = [3,4,4,6,5,4,4];  
finishDur = [10,25,23,10,9,3,13];
```

Ισχύουν οι ακόλουθοι περιορισμοί:

- Οι εργασίες κατασκευής και οι λοιπές εργασίες απαιτούν κτίστες (builders), ενώ οι ηλεκτρολογικές ηλεκτρολόγους (electricians) και οι υδραυλικές, υδραυλικούς (plumbers). Ένας κτίστης μπορεί να εργάζεται είτε στην κατασκευή του ορόφου (μπετά) είτε στις λοιπές εργασίες σε διαφορετικές χρονικές στιγμές.
- Πρέπει πρώτα να γίνει η κατασκευή του ορόφου (μπετά) πριν από κάθε άλλη εργασία στον όροφο.
- Οι όροφοι πρέπει να κατασκευαστούν με την σειρά (!), δηλ. δεν μπορεί να ξεκινήσει η κατασκευή (μπετά) του α2 (δεύτερος όροφος) πριν την ολοκλήρωση του α1 (πρώτος όροφος).
- Οι ηλεκτρολογικές και υδραυλικές εργασίες μπορούν να γίνουν με οποιαδήποτε σειρά, ακόμη και ταυτόχρονα σε ένα όροφο.
- Οι λοιπές εργασίες ξεκινούν μετά το πέρας των ηλεκτρολογικών και υδραυλικών εργασιών. Πρέπει δηλαδή να έχουν ολοκληρωθεί και οι υδραυλικές και οι ηλεκτρολογικές εργασίες, για να ξεκινήσουν οι λοιπές.
- Τα συνεργεία μπορούν να εργαστούν σε οποιοδήποτε όροφο ο οποίος έχει κατασκευαστεί (μπετά).
- Το έργο (όλοι οι όροφοι) πρέπει να ολοκληρωθεί μέσα σε 140 μέρες.

Όμως, έχετε καθυστερήσει την έναρξη του έργου, και μπορείτε να το ξεκινήσετε μετά την 50 ημέρα (\geq). Αυτό που σας ενδιαφέρει είναι ποιος είναι ο ελάχιστος αριθμός εργαζομένων (builders, electricians, plumbers) για να ολοκληρωθεί το έργο μέσα στις ημερομηνίες.

Να υλοποιήσετε το πρόβλημα σε MiniZinc και να εμφανίζεται τους χρόνους έναρξης των εργασιών κατασκευής (startB), ηλεκτρολογικών (startE), υδραυλικών (startP), και λοιπών (startF) εργασιών, καθώς και το αριθμό των εργαζομένων που απαιτούνται. Προφανώς το άθροισμα των τελευταίων πρέπει να είναι ελάχιστο. Για παράδειγμα:

```
startB = [50, 62, 67, 77, 82, 87, 97];
startE = [62, 67, 97, 82, 87, 107, 102];
startP = [62, 70, 96, 82, 96, 119, 110];
startF = [72, 102, 117, 107, 127, 136, 127];
builders = 14;
electricians = 15;
plumbers = 5;
-----
=====
```

Τα δεδομένα βρίσκονται στο αρχείο building.mzn στο compus.

Άσκηση 4 – Παραγωγή Ενέργειας

Είστε υπεύθυνοι για την παραγωγή ενέργειας σε μια νησιωτική περιοχή. Υπάρχουν 5 γεννήτριες, οι οποίες έχουν διαφορετική ικανότητα παραγωγής ενέργειας και μπορούν να λειτουργήσουν σε τέσσερις διαφορετικές θέσεις λειτουργίας. Έτσι στη θέση (mode):

- L025, η γεννήτρια παράγει ακριβώς το 25% της ενέργειας της,
- L050, η γεννήτρια παράγει ακριβώς το 50% της ενέργειας της,
- L100, η γεννήτρια παράγει ακριβώς το 100% της ενέργειας της,

- SHDN η γεννήτρια είναι κλειστή για να "κρυώσει".

Η παραγωγή της κάθε γεννήτριας στο 25% (1/4 της συνολικής απόδοσης) δίνεται από ένα μονοδιάστατο πίνακα:

```
array[GEN] of int : power25 = [20,25,15,30,40];
```

Σύμφωνα με το παραπάνω η πρώτη γεννήτρια παράγει 20 μονάδες ενέργειας στο mode L025, $20 \cdot 2 = 40$ μονάδες ενέργειας στο mode L050, $20 \cdot 4 = 80$ μονάδες ενέργειας στο mode L100 και προφανώς 0 στο mode SHDN.

Υπάρχει ένα πλήθος περιορισμών στη λειτουργία τους:

- Οι γεννήτριες πρέπει να "ζεσταθούν" δουλεύοντας μια μέρα σε mode L025 μετά από mode SHDN ή κατά την αρχική τους εκκίνηση.
- Οι γεννήτριες μπορούν να μείνουν στη θέση SHDN για απεριόριστο αριθμό ημερών.
- Μετά το "ζέσταμα" (mode L025) μιας μέρας, μια γεννήτρια πρέπει να μπει σε mode L050 ή L100.
- Η παραγωγή μιας γεννήτριας δεν μπορεί να πέσει σε χαμηλότερο ποσοστό παραγωγής, δηλαδή για παράδειγμα δεν μπορεί να πάει από L050 σε L025, αλλά ούτε και από L100 σε L050 ή L025.
- Όταν μια γεννήτρια παράξει 9 φορές το πόσο της ενέργειας που παράγει στο 25% (power25), θα πρέπει να κλείσει (SHDN). Για λόγους τεχνικούς δεν μπορεί να κλείσει ούτε αργότερα, ούτε νωρίτερα.

Πρέπει να δημιουργηθεί το πρόγραμμα λειτουργίας των γεννητριών για τις επόμενες 15 μέρες. Στο διάστημα αυτό, οι γεννήτριες θα πρέπει να παράγουν *κάθε μέρα πλην της πρώτης*, ενέργεια μεγαλύτερη ή ίση από 150 μονάδες και μικρότερη ή ίση από 200 μονάδες. Να γράψετε ένα πρόγραμμα σε MiniZinc το οποίο να βρίσκει αυτό το πρόγραμμα. Το MiniZinc πρόγραμμά σας θα εμφανίζει το πρόγραμμα των γεννητριών, και την συνολική ημερήσια παραγωγή, σε δύο μορφές: (α) σε αναγνώσιμη μορφή (β) σε μορφή όπου εμφανίζονται οι τιμές των αντίστοιχων πινάκων (Data in Tables). Για παράδειγμα:

```

1    2    3    4    5    6    7    8    9    10   11   12   13   14   15
L025 L100 L100 SHDN SHDN L025 L050 L050 L100 SHDN SHDN L025 L100 L100 SHDN
L025 L050 L050 L050 L050 SHDN SHDN SHDN L025 L050 L050 L100 SHDN L025 L050
L025 L050 L050 L050 L050 SHDN L025 L050 L050 L050 L050 SHDN L025 L100 L100
SHDN SHDN SHDN SHDN L025 L050 L050 L100 SHDN SHDN SHDN SHDN SHDN SHDN L025
SHDN SHDN L025 L050 L050 L050 L050 SHDN L025 L050 L050 L050 L050 SHDN L025
- Power Production -----
60  160  200  160  190  160  195  190  175  160  160  200  175  165  180

- Data in Tables -----
schedule=[L025, L100, L100, SHDN, SHDN, L025, L050, L050, L100,
SHDN, SHDN, L025, L100, L100, SHDN, L025, L050, L050, L050, L050,
SHDN, SHDN, SHDN, L025, L050, L050, L100, SHDN, L025, L050, L025,
L050, L050, L050, L050, SHDN, L025, L050, L050, L050, L050, SHDN,
L025, L100, L100, SHDN, SHDN, SHDN, SHDN, L025, L050, L050, L100,
SHDN, SHDN, SHDN, SHDN, SHDN, SHDN, L025, SHDN, SHDN, L025, L050,
L050, L050, L050, SHDN, L025, L050, L050, L050, L050, SHDN, L025];

production=[60, 160, 200, 160, 190, 160, 195, 190, 175, 160, 160,
200, 175, 165, 180];
-----
```

Τα δεδομένα βρίσκονται στο αρχείο generators.mzn στο compus.

Σημειώσεις

- Τα απαραίτητα αρχεία για την υλοποίηση των τριών ασκήσεων θα τα βρείτε στο compus (Εγγραφα->ConstraintProgramming->CourseWork->CW2.
 - Άσκηση 1: αρχείο cpu.mzn
 - Άσκηση 2: αρχείο carshop.mzn
 - Άσκηση 3: αρχείο building.mzn
 - Άσκηση 4: αρχείο generators.mzn
- Ο κώδικάς σας θα πρέπει να είναι γενικός, δηλαδή να μπορεί να εκτελεστεί και με άλλες παραμέτρους, παρόμοιες με αυτές που δίνονται σε κάθε άσκηση.

ΠΑΡΑΔΟΣΗ

Θα παραδώσετε εντός της ημερομηνίας που αναφέρεται στο compus τα ακόλουθα:

- Τα παραπάνω αρχεία με όνομα και τη δομή καταλόγων που έχουν στο compus **σε ένα αρχείο code.zip**.
- Ένα αρχείο **report.pdf** (σε μορφή pdf) το οποίο θα περιέχει:
 - Στην πρώτη σελίδα το Όνομά σας, τον Αριθμό μητρώου σας και το email σας.
 - Για κάθε μια από τις ασκήσεις:
 - τον **κώδικα** σας (ασχέτως αν βρίσκεται και στο αντίστοιχο αρχείο) και σχολιασμό σχετικά με αυτόν.
 - Παραδείγματα εκτέλεσης
 - Bugs και προβλήματα που έχει ο κώδικάς σας.

ΠΡΟΣΟΧΗ: ΝΑ ΑΚΟΛΟΥΘΗΣΕΤΕ ΑΥΣΤΗΡΑ ΤΑ ΟΝΟΜΑΤΑ ΤΩΝ ΜΕΤΑΒΛΗΤΩΝ ΠΟΥ ΔΙΝΕΤΑΙ ΠΑΡΑΠΑΝΩ (ΑΥΤΟΜΑΤΟΣ ΕΛΕΓΧΟΣ ΚΩΔΙΚΑ)

Καλή επιτυχία (και *have fun with MiniZinc!*)