ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ ΠΛΗΡΟΦΟΡΙΑΣ

ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΕΘΟΔΟΙ & ΕΡΓΑΛΕΙΑ ΤΝ (2ο εξάμηνο)

Εργασία: 1
Ακαδημαϊκό Έτος : 20224-2025
Εισηγητής: Ηλίας Σακελλαρίου

Άσκηση 1: Client/Server Allocation

Σε ένα κατανεμημένο σύστημα από edge servers πρέπει να αποφασίσετε πως θα κατανείμετε ένα σύνολο από υπηρεσίες οι οποίες απαιτούνται από τους πελάτες σας. Η υπηρεσία που απαιτεί να εκτελέσει κάθε πελάτης έχει απαιτήσεις σε δικτυακούς πόρους (ClientDemands), οι οποίες εκφράζονται ως ένας ακέραιος (μονάδα πόρου) για κάθε υπηρεσία, και ανάλογα σε ποιον server θα εγκατασταθεί έχει και ένα διαφορετικό κόστος δικτύωσης (NetworkingCost). Κάθε εργασία μπορεί να ανατεθεί για εκτέλεση σε ένα server. Οι servers μπορούν να εκτελέσουν πολλές εργασίες ταυτόχρονα και έχουν συγκεκριμένη χωρητικότητα (Capacity), δηλαδή δεν μπορούν να εκτελέσουν ταυτόχρονα ένα σύνολο εργασιών, όπου το άθροισμα των δικτυακών απαιτήσεων τους ξεπερνά την χωρητικότητα του server. Αν ένας server έχει εργασίες προς εκτέλεση τότε θεωρείται ενεργός, ενώ αν δεν έχει μη-ενεργός. Κάθε ενεργός server έχει ένα κόστος λειτουργίας (DeploymentCost).

Τα δεδομένα του προβλήματος είναι:

Ο αριθμός των servers:

```
Servers = 4;
```

Ο αριθμός των πελατών (1 υπηρεσία για κάθε πελάτη)

```
Clients = 6;
```

Η χωρητικότητα του κάθε server. Για παράδειγμα στο ακόλουθο η χωρητικότητα του server 1 είναι 100

```
Capacity = [100, 40, 60, 60];
```

Το κόστος λειτουργίας του κάθε server. Για παράδειγμα στο ακόλουθο το κόστος λειτουργίας του server 1 είναι 860:

```
DeploymentCost = [860, 350, 440, 580];
```

Οι απατήσεις των πελατών σε πόρους. Για παράδειγμα η υπηρεσία του πελάτη 1 έχει απαιτήσεις σε πόρους 12 μονάδες.

```
ClientDemands = [12, 17, 5, 13, 20, 25];
```

Το κόστος δικτύωσης ανά μονάδα πόρου υπολογίζεται ανάλογα με τον server στον οποίο εκτελείται η υπηρεσία. Για παράδειγμα η υπηρεσία πελάτη 1, έχει κόστος 27/μονάδα πόρου αν εκτελεστεί στον server 1, 66/μονάδα πόρου αν εκτελεστεί στον server 2, κοκ. Το κόστος δίνεται ως πίνακας δύο διαστάσεων με κάθε γραμμή να αφορά μια υπηρεσία και τις στήλες να είναι οι αντίστοιχοι servers:

```
NetworkingCost = [|27, 66, 44, 55
                  |53, 89, 68, 46
                  |17, 40, 18, 61
```

```
|20, 68, 44, 78
|42, 89, 65, 78
|57, 55, 49, 31|];
```

Το συνολικό κόστος προκύπτει ως το συνολικό **άθροισμα** του *κόστους λειτουργίας των ενεργών servers, δηλαδή εκείνων στους οποίους έχει ανατεθεί μια τουλάχιστον εργασία* και του συνολικού κόστους δικτύωσης.

Το *συνολικό κόστος δικτύωσης* είναι το άθροισμα του κόστους δικτύωσης για κάθε υπηρεσία. Το τελευταίο προκύπτει ως το γινόμενο των απαιτήσεων της υπηρεσίας επί το το κόστος δικτύωσης ανά μονάδα πόρου του server στον οποίο εγκαταστάθηκε. Για παράδειγμα αν η υπηρεσία 1 εγκατασταθεί στον server 2, το κόστος της θα είναι 12*66 = 792, ενώ αν εγκατασταθεί στον server 3 είναι 12*44 = 528.

Να υλοποιήσετε ένα πρόγραμμα σε MiniZinc το οποίο βρίσκει την λύση (στον πίνακα client_at) με το ελάχιστο συνολικό κόστος (μεταβλητή total_cost). Για κάθε υπηρεσία θα εμφανίζεται τους πόρους τους οποίους έχει δεσμεύσει σε κάθε server. Παράδειγμα εκτέλεσης:

```
client_at = [1, 3, 1, 1, 2, 3];
total_cost=6480;
```

Στην παραπάνω λύση, η υπηρεσία 1 θα εκτελεστεί στον server 1, η υπηρεσία 2 στον server 3, κοκ.

Σημείωση: Η λύση που φαίνεται ΔΕΝ είναι η βέλτιστη.

Θα βρείτε τα απαραίτητα αρχεία στο eclass.

Άσκηση 2 - FireTrucks

Σε ένα σταθμό ανεφοδιασμού πυροσβεστικών οχημάτων, υπάρχουν 3 κρουνοί πλήρωσης νερού. Κάθε κρουνός έχει την δυνατότητα να πλήρωσης 50kg νερού το λεπτό. Τα διαθέσιμα πυροσβεστικά οχήματα θα πρέπει:

1. να γεμίζουν με νερό, και έπειτα, *κάποια χρονική στιγμή*
2. να ελεγχθούν από το μοναδικό κινητό συνεργείο που υπάρχει.

(α) Τα οχήματα έχουν διαφορετικές χωρητικότητες σε νερό (σε kg), και διαφορετικούς χρόνους εξυπηρέτησης από το συνεργείο. Οι σχετικές πληροφορίες δίνονται σε πίνακες στα αντίστοιχα αρχεία δεδομένων (MiniZinc data files), όπως:

```
no_firetrucks = 6;
```

ο αριθμός των οχημάτων

```
water_cap = [1200,900,1500,800,1300,600];
```

Η χωρητικότητα του κάθε οχήματος σε νερό

```
service_time = [15,10,20,10,5,20];
```

Ο χρόνος εξυπηρέτησης από το συνεργείο

Να γράψετε ένα πρόγραμμα σε MiniZinc το οποίο υπολογίζει

- **start_fill** πότε θα ξεκινήσει ο ανεφοδιασμός του κάθε οχήματος,
- **start_service** πότε θα ξεκινήσει ο έλεγχος από το συνεργείο σε κάθε όχημα,
- **total_service**, ο χρόνος που απασχολήθηκε το συνεργείο, (χωρίς τα όποια κενά στο πρόγραμμα, δηλαδή από την στιγμή που "μπήκε" σε αυτό το πρώτο πυροσβεστικό μέχρι την ολοκλήρωση του service του τελευταίου),

- **makespan** ο χρόνος ολοκλήρωσης ολόκληρου του προγράμματος, ο οποίος θα πρέπει να είναι ο ελάχιστος δυνατός.

Για παράδειγμα (λύση *δεν είναι η βέλτιστη*):

```
start_fill=[12, 0, 0, 18, 30, 0];
start_service=[70, 18, 30, 85, 95, 50];
total_service = 82;
makespan=100;
```

Όπου στο παράδειγμα το όχημα a, θα ξεκινήσει ανεφοδιασμό την χρονική στιγμή 12, και την χρονική στιγμή 70 τον έλεγχο από το συνεργείο, με χρόνο service 82, ενώ το συνολικό χρονοπρόγραμμα θα διαρκέσει συνολικά 100 λεπτά. Το παραπάνω παράδειγμα εκτέλεσης **ΔΕΝ είναι η βέλτιστη λύση**.

Θεωρείστε ότι το χρονοπρόγραμμα ξεκινά από την χρονική στιγμή 0 και πρέπει να ολοκληρωθεί σε 1000 λεπτά το πολύ.

Τα απαραίτητα αρχεία θα τα βρείτε στο eclass.

Σημείωσεις

- Τα απαραίτητα αρχεία για την υλοποίηση των τριών ασκήσεων θα τα βρείτε στο ECLASS (Έγγραφα-> Constraints->Assignments)**.**
  - Άσκηση 1: φάκελος ServerLocationProblem, αρχείο project serverAllocProject.mzp
  - Άσκηση2: φάκελος FireTrucks, αρχείο project FireTrucksProject.mzp

ΠΡΟΣΟΧΗ: ΝΑ ΑΚΟΛΟΥΘΗΣΕΤΕ ΑΥΣΤΗΡΑ ΤΑ ΟΝΟΜΑΤΑ ΤΩΝ ΜΕΤΑΒΛΗΤΩΝ ΠΟΥ ΔΙΝΟΝΤΑΙ ΠΑΡΑΠΑΝΩ (ΑΥΤΟΜΑΤΟΣ ΕΛΕΓΧΟΣ ΚΩΔΙΚΑ). ΜΗΝ ΑΛΛΑΖΕΤΕ ΤΑ ΟΝΟΜΑΤΑ ΤΩΝ ΜΕΤΑΒΛΗΤΩΝ.

ΜΗΝ ΑΛΛΑΖΕΤΕ ΤΗΝ ΓΡΑΜΜΗ **output**.

ΠΑΡΑΔΟΣΗ

Θα παραδώσετε εντός της ημερομηνίας που αναφέρεται στο ECLASS ΕΝΑ ΑΡΧΕΙΟ **zip** με τα ακόλουθα:

- Τα παραπάνω αρχεία **με όνομα και τη δομή καταλόγων** που έχουν στο ECLASS**, τα οποία θα περιέχουν τις λύσεις σας.**
- Ένα αρχείο **report.pdf** (***σε μορφή pdf***), στον αρχικό φάκελο του zip, 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 (1-2 όπου είναι εφικτό)
    - Bugs και προβλήματα που έχει ο κώδικάς σας.

Καλή επιτυχία