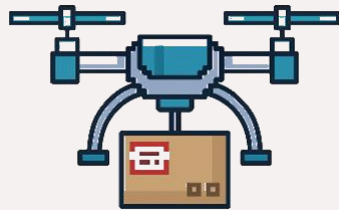


POST-DISASTER SUPPLY DELIVERY BY DRONES USING LINEAR PROGRAMMING

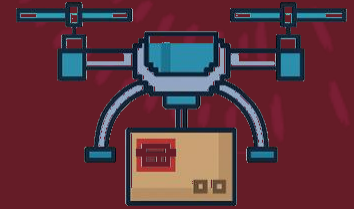


Γραμμική και Συνδυαστική Βελτιστοποίηση
[EE916]

Γέροντας Νικόλαος

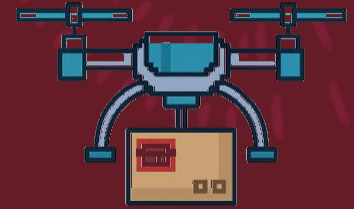


Εισαγωγή



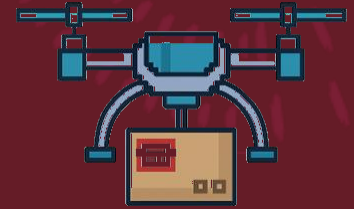
- Σε κρίσεις (όπως οι σεισμοί και οι πλημμύρες) η γρήγορη διανομή τροφίμων, φαρμάκων και νερού σε δύσβατα σημεία σώσει ζωές. Τα επανδρωμένα μέσα, ωστόσο, συχνά καθυστερούν λόγω πρόσβασης ή συντονισμού. Οι ημιαυτόνομοι δρόνοι (Μη Επανδρωμένα Αεροσκάφη) αποτελούν μία ασφαλή και ευέλικτη λύση στο πρόβλημα. Ωστόσο, το ερώτημα που πρέπει να απαντηθεί είναι:

Η ερώτηση...



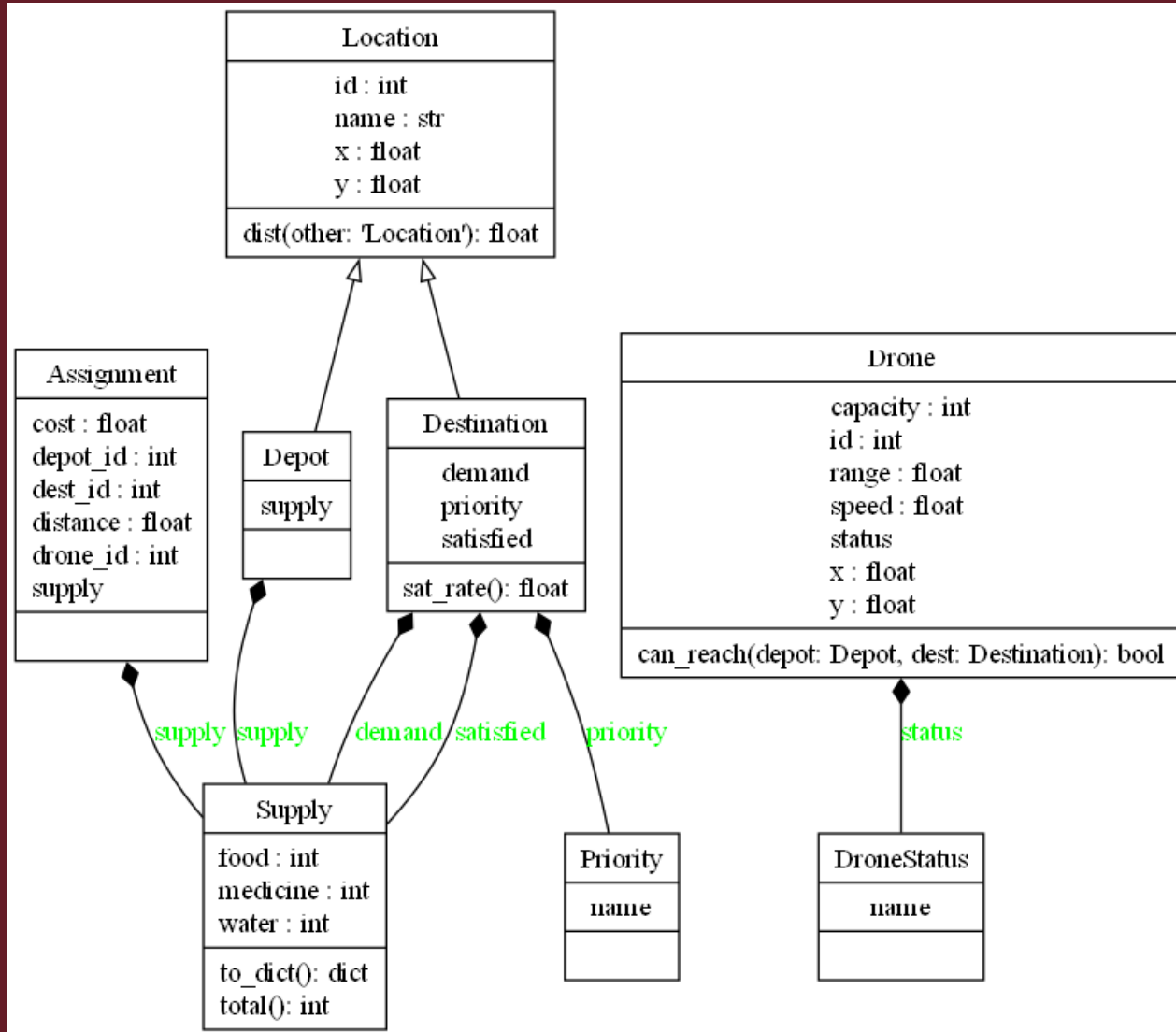
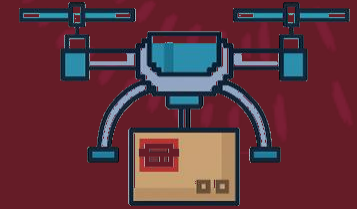
- Ποιος δρόμος πρέπει να σταλεί, από ποιο σημείο εφοδιασμού, προς ποιον προορισμό και με πόσο φορτίο, ώστε να ελαχιστοποιηθεί το συνολικό «κόστος ανθρωπιστικής παράδοσης»;

... η απάντηση



- Η απάντηση στο ερώτημα δίνεται με γραμμικό προγραμματισμό!
- Συγκεκριμένα, αναπτύχθηκε ένα πρόγραμμα σε Python που υπολογίζει τη βέλτιστη ανάθεση αποστολών από αποθήκες σε προορισμούς, με χρήση γραμμικού προγραμματισμού, λαμβάνοντας υπόψη τις ανάγκες, τη διαθεσιμότητα προμηθειών και τις δυνατότητες των δρόμων.

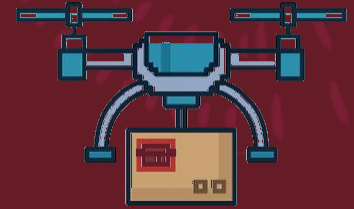
Μοντελοποίηση



- UML διάγραμμα του αρχείου `models.py`

- Συνοψίζει τη δομή των βασικών κλάσεων που υλοποιούν το πρόβλημα.

Μοντελοποίηση - Περιορισμοί



- Για κάθε προορισμό $j \in J$ και τύπο προμήθειας $s \in S$:

$$\sum_{(d,i,j) \in R: \mathbf{dest}=j} x_{dij s} + \mathbf{unmet}_{js} = D_{js}$$

- Η παραδοθείσα ποσότητα συν τη μη καλυμμένη ζήτηση πρέπει να ισούται με τη συνολική ζήτηση σε κάθε σημείο ανάγκης.

- Για κάθε δρόνο $d \in D$:

$$\sum_{(d,i,j) \in R} \sum_{s \in S} x_{dij s} \leq C_d$$

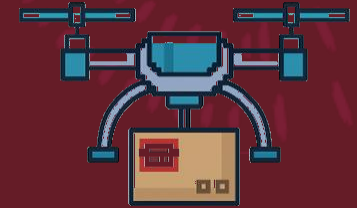
- Το συνολικό φορτίο που μεταφέρει κάθε δρόνος σε όλες τις ανατεθειμένες διαδρομές δεν μπορεί να υπερβεί τη χωρητικότητά του.

- Για κάθε depot $i \in I$ και τύπο προμήθειας $s \in S$:

$$\sum_{(d,i,j) \in R: \mathbf{depot}=i} x_{dij s} \leq S_{is}$$

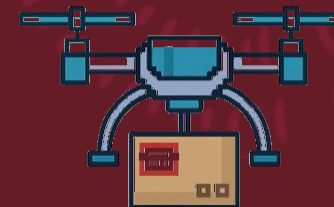
- Η συνολική ποσότητα κάθε τύπου προμήθειας που διανέμεται από ένα σημείο εφοδιασμού δεν μπορεί να υπερβεί το διαθέσιμο απόθεμά του.

Μεταβλητές και Περιορισμοί



UML Οντότητα – Κλάση/Γνώρισμα/Μέθοδος	Μεταβλητή ή Παράμετρος MILP	Ερμηνεία
Drone.capacity	C_d	Μέγιστο φορτίο δρόνου d - Περιορισμός
Drone.range + can_reach()	Φίλτρο διαδρομών	Έγκυρες αποστολές – Έμμεσος περιορισμός
Depot.supply	$S_{i,s}$	Διαθέσιμη ποσότητα τύπου s στο σημείο εφοδιασμού i - Περιορισμός
Destination.demand	$D_{j,s}$	Απαίτηση τύπου s στο σημείο ανάγκης j - Περιορισμός
y_{dij}	Δυαδική	Ανάθεση διαδρομής/αποστολής σε έναν δρόνο
$x_{dij,s}$	Συνεχής	Μονάδες τύπου s που μεταφέρονται
$unmet_{js}$	Slack μεταβλητές	Ανάγκη που δεν καλύπτεται

Αντικειμενική συνάρτηση



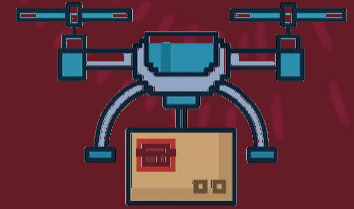
$$Z = \min \left\{ \sum_{d,i,j} [\text{dist}(i,j) \cdot \mathbf{w}_{\text{priority}}(j) \cdot y_{d,ij}] + \sum_{j,s} [\text{UNMET_PENALTY} \cdot \mathbf{w}_{\text{priority}}(j) \cdot \text{unmet}_{j,s}] \right\}$$

- Το **γνώρισμα priority** των σημείων ανάγκης επηρεάζει το κόστος στην αντικειμενική συνάρτηση, με βάρη:

priority_w = {Priority.HIGH: 3, Priority.MEDIUM: 2, Priority.LOW: 1}

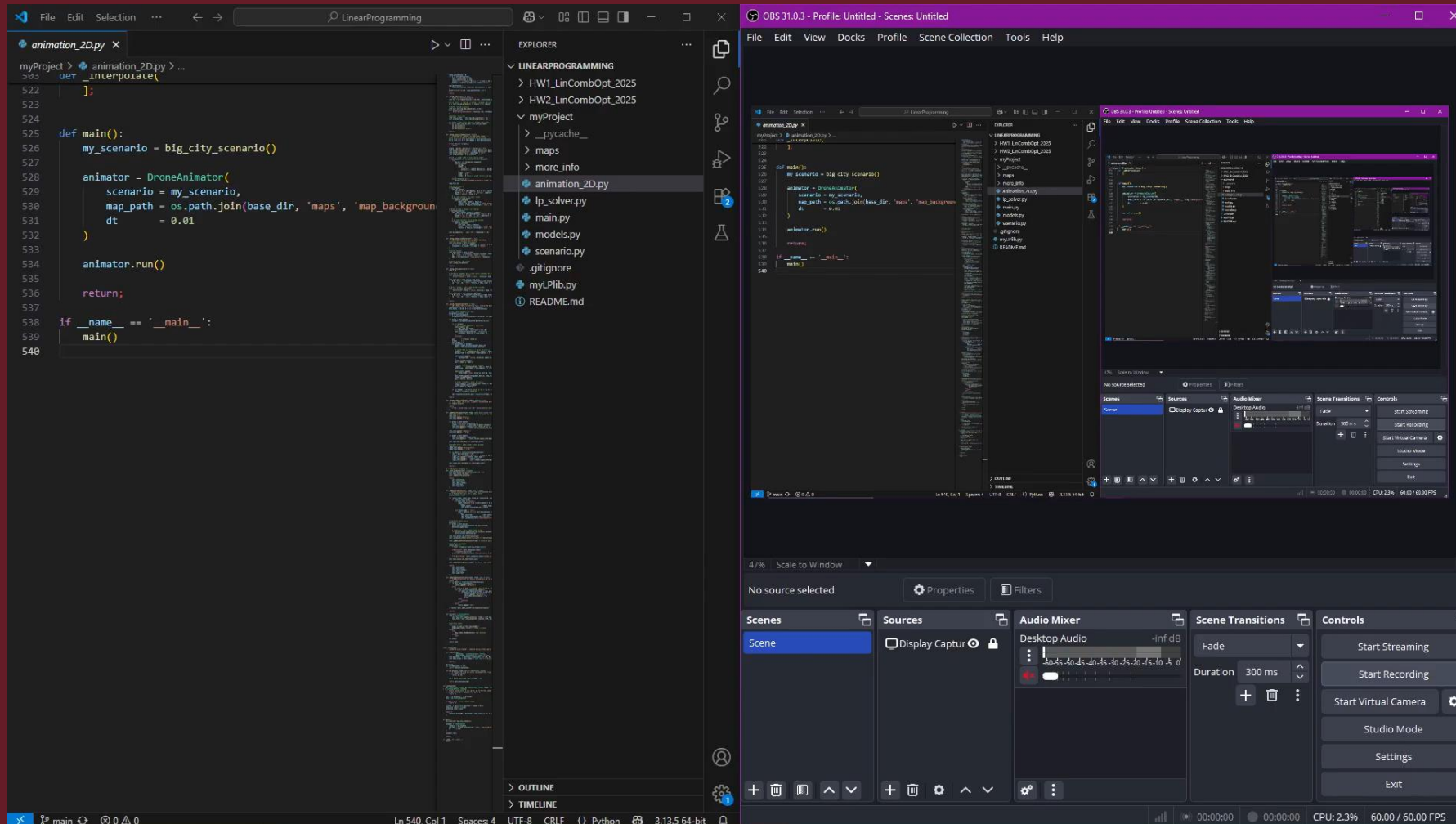
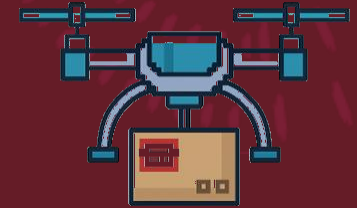
ώστε να προτιμώνται κρίσιμες αποστολές!

Δημιουργία των σεναρίων

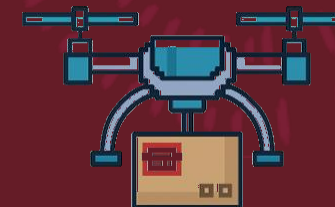


- Με βάση τις δοκιμές και την οπτικοποίηση μέσω animation, το σενάριο `big_city_scenario()` αποδείχθηκε το πιο κατανοητό και ενδιαφέρον. Το σενάριο περιλαμβάνει 3 δρόνους, 2 σημεία εφοδιασμού και 6 σημεία ανάγκης με διαφορετικές απαιτήσεις και προτεραιότητες.

Animation εκτέλεση



Αποτέλεσμα στο τερματικό



-> Λύση σεναρίου:

Δρόνος 0:

-> Stadium	Απόσταση: 37.0	Φορτίο: 5F 0W 0M
-> Hospital #1	Απόσταση: 36.2	Φορτίο: 50F 40W 25M
Συνολική απόσταση: 73.3		

Δρόνος 1:

-> Stadium	Απόσταση: 37.0	Φορτίο: 0F 30W 10M
-> School	Απόσταση: 44.4	Φορτίο: 30F 20W 10M
Συνολική απόσταση: 81.5		

Δρόνος 2:

-> Clinic	Απόσταση: 47.2	Φορτίο: 25F 15W 15M
-> Hospital #2	Απόσταση: 38.4	Φορτίο: 45F 30W 20M
Συνολική απόσταση: 85.7		

Χρόνος επίλυσης: 0.938s

Ποσοστά κάλυψης προμηθειών ανά προορισμό:

! Stadium	: 56.2%
✓ School	: 100.0%
✓ Clinic	: 100.0%
✓ Hospital #1	: 100.0%
✗ Resident	: 0.0%
✓ Hospital #2	: 95.0%

Μέση κάλυψη: 75.2%

- Η λύση MILP επέλεξε να μη στείλει κανέναν δρόνο στον κόμβο Resident, με αποτέλεσμα ποσοστό κάλυψης 0%. Αυτό δεν αποτελεί «σφάλμα» του αλγορίθμου, αλλά ορθολογική συνέπεια της αντικειμενικής συνάρτησης!

Ευχαριστώ πολύ για την προσοχή σας!

