Μάθημα: Προχωρημένα Θέματα Διοικητικής Επιστήμης και Τεχνολογίας

Ομαδική Εργασία: 2-4 Άτομα

Βαθμολογία: 50% Τελικού Βαθμού (Competition)

Διορία: Ιούνιος 2024 (Ακριβής ημερομηνία θα ανακοινωθεί στη συνέχεια)

Δήλωση Ομάδων:

https://docs.google.com/spreadsheets/d/1E8A9JilHDVUONhX5EdhXqGeNbqQeD0azoCEdIkQdumw/edit?usp=sharing

(2 sheets: Full-Time & Part-Time)

Διορία Δήλωσης Ομάδων: 20/05/2023

Οδηγίες Υποβολής: Υποβολή από τον Αρχηγό κάθε ομάδας στο mzaueb@gmail.com.

Στα πλαίσια της άσκησης θα επιλύσουμε ένα πρόβλημα δρομολόγησης οχημάτων.

Το επιχειρησιακό σενάριο έχει ως εξής:

1. Η κεντρική αποθήκη μίας επιχείρησης λαμβάνει ένα πλήθος παραγγελιών από 200 πελάτες.
2. Στην κεντρική αποθήκη είναι βασισμένα φορτηγά αυτοκίνητα.
3. Σκοπός που τίθεται είναι η κατασκευή μίας διαδρομής για κάθε ένα από τα οχήματα με σκοπό την εξυπηρέτηση των πελατών
4. Κάθε διαδρομή ξεκινά από την κεντρική αποθήκη, επισκέπτεται τους διάφορους πελάτες και ολοκληρώνεται με επιστροφή στην κεντρική αποθήκη.
5. Κάθε πελάτης είναι συνδεδεμένος με δύο ποσότητες:
    - Delivery: Ποσότητα προϊόντος που πρέπει να μεταφερθεί από την αποθήκη στον πελάτη
    - Pick-up: Ποσότητα προϊόντος που πρέπει να μεταφερθεί από τον πελάτη στην κεντρική αποθήκη
6. Κάθε φορτηγό έχει ένα ανώτατο επιτρεπόμενο βάρος μεταφερόμενου φορτίου (χωρητικότητα) ίσο με 12 tn.

Στόχος που τίθεται είναι οι παραγόμενες διαδρομές να ελαχιστοποιούν τα συνολικά χιλιόμετρα που διανύονται.

Το κοινό δίκτυο που θα θεωρήσουν όλες οι ομάδες παράγεται από τον παρακάτω κώδικά. Οι συντεταγμένες δίνονται σε km και η ζήτηση σε τόνους. Οι αποστάσεις μεταξύ των κόμβων υπολογίζονται ως ευκλείδειες αποστάσεις.

```
import random
class Node:
    def __init__(self, id, delivery, pick_up, xx, yy):
        self.ID = id
        self.delivery = delivery
        self.pick_up = pick_up
        self.x = xx
        self.y = yy
```

```
random.seed(1)
capacity = 12
all_nodes = []
depot = Node(0, 0, 0, 50, 50)
all_nodes.append(depot)
customers = 200
for i in range(customers):
    id = i + 1
    delivery = round(random.normalvariate(2, 0.5), 1)
    pick_up = round(random.normalvariate(2, 0.5), 1)
    xx = random.randint(0, 100)
    yy = random.randint(0, 100)
    cust = Node(id, delivery, pick_up, xx, yy)
    all_nodes.append(cust)
```

Σημείωση: Στα αρχεία του moodle, έχει προστεθεί κώδικας για το VRP. Προτείνεται η χρήση και τροποποίηση του ώστε να αντιμετωπίζει το παραπάνω επιχειρησιακό σενάριο.

Βαθμολόγηση και οδηγίες:

Η ομάδα με τη χειρότερη τιμή αντικειμενικής συνάρτησης η οποία θα σχηματίσει μία λύση που ικανοποιεί όλους τους περιορισμούς λαμβάνει 7.5.

Η ομάδα με την καλύτερη τιμή αντικειμενικής συνάρτησης η οποία θα σχηματίσει μία λύση που ικανοποιεί όλους τους περιορισμούς λαμβάνει 10.

Οι υπόλοιπες ομάδες που θα δημιουργήσουν μία εφικτή διαδρομή θα βαθμολογηθούν γραμμικά βάσει της τιμής της αντικειμενικής συνάρτησης.

Οι ομάδες με λύσεις που δεν ικανοποιούν όλους τους περιορισμούς θα βαθμολογηθούν βάσει του κώδικα τους.

Οι λύσεις πρέπει να βρίσκονται σε ένα αρχείο με τη μορφή του αρχείου example_solution.txt.

Τέλος για έλεγχο της ποιότητας της λύσης σας, παρέχεται ένα script με το όνομα sol_checker.py με το οποίο μπορείτε να ελέγξετε πως η λύση σας ικανοποιεί όλους τους κανόνες του επιχειρησιακού σεναρίου και να πιστοποιήσετε πως η τιμή της αντικειμενικής συνάρτησης είναι αύτη που υπολογίσατε.

Για να τρέξει ο έλεγχος βάζετε το αρχείο της λύσης σας στο ίδιο folder με το sol_checker.py και κατόπιν εκτελείτε το sol_checker.py.