

Μάθημα: ΜΕΒΕΔΕ

Ομαδική Εργασία: 3-5 Άτομα

Βαθμολογία: 30% Τελικού Βαθμού

Διορία: 07/01/2019 - 15.00

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 100 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 και επισκέπτεται τους διάφορους πελάτες. Η διαδρομή τερματίζεται στον τελευταίο πελάτη που επισκέπτεται το όχημα.
5. Κάθε παραγγελία πρέπει να ικανοποιηθεί από μία και μόνο μία επίσκεψη κάποιου οχήματος. Επομένως, όταν ένα όχημα επισκέπτεται ένα πελάτη, μεταφέρει σε αυτόν το σύνολο των προϊόντων της παραγγελίας του.
6. Κάθε φορτηγό έχει μία συγκεκριμένη χωρητικότητα προϊόντων, επομένως τα μεταφερόμενα από το φορτηγό αγαθά δεν πρέπει να ξεπερνούν τη μέγιστη χωρητικότητα του φορτηγού
7. Υποθέστε πως τα οχήματα ταξιδεύουν με 35 km/hr και πως για κάθε πελάτη, ο χρόνος εκφόρτωσης των αγαθών είναι 15 λεπτά.
8. Στην αποθήκη βρίσκονται διαθέσιμα τα εξής φορτηγά:
  - a. 15 φορτηγά με μέγιστη χωρητικότητα 1500 kg.
  - b. 15 φορτηγά με μέγιστη χωρητικότητα 1200 kg.
9. Κάθε διαδρομή έχει ένα όριο συνολικής χρονικής διάρκειας 3.5 hr.

Στόχος που τίθεται είναι οι παραγόμενες διαδρομές να ελαχιστοποιούν τη συνολική απόσταση που διανύεται από τα φορτηγά.

Για να αντιμετωπίσετε το πρόβλημα, πρέπει να γράψετε ένα πρόγραμμα το οποίο:

1. Να παρέχει την υποδομή για την επίλυση του εξεταζόμενου προβλήματος και να δημιουργεί το input (2)
  - a. Κλάσεις (π.χ. Διαδρομή, Πελάτης (ή Παραγγελία), κτλ.)
  - b. Πίνακες (Κόστος μεταξύ πελατών, Χρονικές Αποστάσεις...)

Προφανώς δεν υπάρχει ένας μοναδικός (και σωστός τρόπος) για την αναπαράσταση του προβλήματος. Σκοπός που τίθεται είναι η πλήρης περιγραφή του μοντέλου, καθώς και η διευκόλυνση σας για την ανάπτυξη του αλγορίθμου βελτιστοποίησης στη συνέχεια.

2. Υλοποιείτε μία μέθοδο που θα δέχεται σαν όρισμα μία οποιαδήποτε λύση του προβλήματος και θα υπολογίζει την τιμή της αντικειμενικής συνάρτησης αυτής της λύσης. (1)
3. Κατασκευάστε έναν κατασκευαστικό αλγόριθμο, ο οποίος να δημιουργεί μία αρχική λύση. (2)
4. Σχεδιάστε ένα αλγόριθμο τοπικής έρευνας ο οποίος να χρησιμοποιεί την κίνηση μετακίνηση ενός πελάτη (relocation) (3)
  - a. Σε πόσες επαναλήψεις η μέθοδός σας παγιδεύεται σε τοπικό ελάχιστο;
  - b. Ποιο είναι η είναι το κόστος του τοπικού ελάχιστου και ποιο το σύνολο των διαδρομών;
5. Σχεδιάστε ένα VND αλγόριθμο ο οποίος θα χρησιμοποιεί τρεις τελεστές τοπικής έρευνας (2)

Οι κόμβοι του προβλήματος δημιουργούνται από τον παρακάτω κώδικα ο οποίος πρέπει να ενσωματωθεί στο πρόγραμμά σας. Οι συντεταγμένες δίνονται σε χιλιόμετρα. Ο πίνακας των αποστάσεων προκύπτει από τον υπολογισμό των Ευκλείδειων αποστάσεων μεταξύ των διάφορων κόμβων:

```
public void CreateAllNodesAndCustomerLists(int numberOfCustomers) {  
    //Create the list with the customers  
    customers = new ArrayList();  
    int birthday = 9021998; // if your bday is on 9 feb 1998  
    Random ran = new Random(birthday);  
    for (int i = 0 ; i < 100; i++)  
    {  
        Node cust = new Node();  
        cust.x = ran.nextInt(100);  
        cust.y = ran.nextInt(100);  
        cust.demand = 100*(1 + ran.nextInt(5));  
        cust.serviceTime = 0.25;  
        customers.add(cust);  
    }  
}
```

```
//Build the allNodes array and the corresponding distance matrix
allNodes = new ArrayList();
depot = new Node();
depot.x = 50;
depot.y = 50;
depot.demand = 0;
allNodes.add(depot);
for (int i = 0 ; i < customers.size(); i++)
{
    Node cust = customers.get(i);
    allNodes.add(cust);
}

for (int i = 0 ; i < allNodes.size(); i++)
{
    Node nd = allNodes.get(i);
    nd.ID = i;
}
}
```