

**Μέθοδοι Βελτιστοποίησης στη διοικητική Επιστήμη**

**Απλή Ομαδική Εργασία ΜΕ.ΒΕ.ΔΕ 2020-2021**

|  |  |
| --- | --- |
| **Αριθμός μητρώου** | **Ονοματεπώνυμο** |
| 8180080 | Ειρήνη Μαρία Νικολακοπούλου |
| 8180138 | Ναταλία Τσαμτσούρη |
| 8180022 | Αικατερίνη Διβινή |
| 8180056 | Άγγελος Κωνσταντόπουλος |

**Αθήνα, Ιανουάριος2021**

Πίνακας περιεχομένων

[1ο Ερώτημα 3](#_Toc62065607)

[2ο Ερώτημα 3](#_Toc62065608)

[3ο Ερώτημα 3](#_Toc62065609)

[4ο Ερώτημα 4](#_Toc62065610)

[4.α. Υποερώτημα 4](#_Toc62065611)

[4.β. Υποερώτημα 4](#_Toc62065612)

[5ο Ερώτημα 4](#_Toc62065613)

# 1ο Ερώτημα

* **Αρχείο:***Vrp\_model.py*
* **Εντολές για τη δημιουργία του μοντέλου στην main:**

*m1 = Model()*

*m1.BuildModel()*

# 2ο Ερώτημα

* **Αρχείο:** *Solver.py*
* **Μέθοδος:** *CalculateDistance()*
* **Εντολές για τον υπολογισμό της αντικειμενικής συνάρτησης (ως όρισμα βάλαμε την αρχική λύση του προβλήματος) και για την εκτύπωση της τιμής αυτής:**

*dist = s1.CalculateDistance(s1.sol)*

*print(“Total Distance: “, dist)*

# 3οΕρώτημα

* **Αρχείο:** *Solver.py*
* **Μέθοδος:** *solve()*

Κατασκευάσαμε την αρχική λύση μέσω NearestNeighbor.

* **Εντολές για την κατασκευή και την εκτύπωση της αρχικής λύσης στην Main:**

*s1.solve(m1)*

*sol = s1.solve()*

# 4ο Ερώτημα

* **Αρχείο:** *Solver.py*
* **Μέθοδος:** *LocalSearch()*
* **Εντολές για την εκτέλεση τοπικής έρευνας με relocation, εκτύπωση των επαναλήψεων και της νέας λύσης στη Main:**

*s1.LocalSearch()*

*s1.Reportsolution(s1.sol)*

## 4.α. Υποερώτημα

* Η μέθοδος μας παγιδεύεται σε τοπικό ελάχιστο μετά από 38 επαναλήψεις (μεταβλητή count).

## 4.β. Υποερώτημα

* Η τιμή της αντικειμενικής συνάρτησης μετά την εκτέλεση της τοπικής έρευνας είναι 1363.3980934209508, ενώ ο αριθμός των διαδρομών είναι ίσος με 24.
* **Εντολή για εκτύπωση συνόλου διαδρομών στην Main:**

*print(“Total routes (after local search) “, len(s1.sol.routes))*

Προκειμένου να εφαρμόσουμε τη μέθοδο relocation, προσθέσαμε στo matrix έναν ακόμα κόμβο, από τον οποίο όλοι οι πελάτες απείχαν απόσταση ίση με το 0. Τον κόμβο αυτόν τον προσθέσαμε και στο τέλος κάθε διαδρομής που είχε δημιουργηθεί. Κατά την ολοκλήρωση των επαναλήψεων και της παραγωγής της νέας λύσης, αφαιρούμε τον νέο κόμβο που προσθέσαμε αρχικά από τις νέες διαδρομές. Την ίδια λογική ακολουθήσαμε και στο επόμενο ερώτημα.

# 5ο Ερώτημα

* **Αρχείο:** *Solver.py*
* **Μέθοδος:** *VND()*
* **Εντολές για την εκτέλεση του VNDκαι εκτύπωση της λύσης στην Μain:**

*s1.VND()*

*s1.reportSolution(s1.sol)*