



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**

**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

**«Ολοκληρωμένο πολυπρακτορικό σύστημα  
παραγγελιοληψίας μεταξύ πολλών σημείων αφετηρίας  
και προορισμού με περιορισμούς ως προς το χρόνο  
παράδοσης»**

Διπλωματική Εργασία

ΡΟΖΗΣ ΝΙΚΟΛΑΟΣ

ΧΩΡΑΦΑΪΔΗΣ ΔΑΜΙΑΝΟΣ

Επιβλέπων  
Σταμούλης Γεώργιος  
Καθηγητής

Επιβλέπων  
Θάνος Γεώργιος  
Μέλος Ε.ΔΙ.Π.

Μέλος επιτροπής  
Τσουκαλάς Ελευθέριος  
Καθηγητής

Βόλος 2019



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**

**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

**«Ολοκληρωμένο πολυπρακτορικό σύστημα  
παραγγελιοληψίας μεταξύ πολλών σημείων αφετηρίας  
και προορισμού με περιορισμούς ως προς το χρόνο  
παράδοσης»**

Διπλωματική Εργασία

ΡΟΖΗΣ ΝΙΚΟΛΑΟΣ

ΧΩΡΑΦΑΪΔΗΣ ΔΑΜΙΑΝΟΣ

Επιβλέπων  
Σταμούλης Γεώργιος  
Καθηγητής

Επιβλέπων  
Θάνος Γεώργιος  
Μέλος Ε.ΔΙ.Π.

Μέλος επιτροπής  
Τσουκαλάς Ελευθέριος  
Καθηγητής

Βόλος 2019



**UNIVERSITY OF THESSALY**

**SCHOOL OF ENGINEERING**

**DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**

**“A multi-agent order management system with time  
window constraints”**

Diploma Thesis

ROZIS NIKOLAOS

CHORAFIDIS DAMIANOS

Supervisor  
Georgios Stamoulis  
Professor

Supervisor  
George Thanos  
Laboratory Teaching  
Staff

Commissioner  
Lefteris Tsoukalas  
Professor

Volos 2019

## ΕΥΧΑΡΙΣΤΙΕΣ

Θα θέλαμε να εκφράσουμε την αμέριστη ευγνωμοσύνη μας σε όλους όσους συνεισέφεραν στην περάτωση αυτής της Διπλωματικής Εργασίας προσφέροντας την βοήθεια τους, αλλά και σε όλους τους διδάσκοντες του Τμήματος που καλλιέργησαν το γνωστικό υπόβαθρο για να προκύψει αυτό το αποτέλεσμα καθ' όλη τη διάρκεια της φοίτησης μας. Ιδιαίτερη μνεία αξίζει στον επιβλέποντα κ. Γεώργιο Θάνο, που από την πρώτη στιγμή στάθηκε αρωγός και ενίσχυσε την προσπάθειά μας για υλοποίηση του αντικειμένου της παρούσας εργασίας, δείχνοντας πραγματικό ενδιαφέρον καθοδηγώντας μας σε κάθε βήμα.

Ακόμα, θα θέλαμε να ευχαριστήσουμε τις οικογένειες μας και τους φίλους μας για την υποστήριξη τους όλα τα χρόνια των σπουδών μας.

**ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ  
ΔΙΚΑΙΩΜΑΤΩΝ**

«Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής».

Ο Δηλών

(Υπογραφή)

Ροζής Νικόλαος

5/7/2019

## ΠΕΡΙΛΗΨΗ

Στην παρούσα Διπλωματική Εργασία εξετάζουμε το πρόβλημα της δρομολόγησης και του προγραμματισμού ενός στόλου οχημάτων που επιδιώκουν να ικανοποιήσουν δυναμικές αιτήσεις παραλαβής και παράδοσης πακέτων σε αστική κλίμακα μεταξύ δύο -ανά αίτηση- σημείων. Αναπτύσσουμε ένα σύστημα για τη διευκόλυνση της συνεργασίας μεταξύ οργανισμών και φυσικών προσώπων που ασχολούνται κυρίως με τον κλάδο των αστικών εμπορευματικών μεταφορών (city logistics) και της εφοδιαστική αλυσίδας (supply chain), χωρίς αυτό να περιορίζει την δυνατότητα εφαρμογής του συστήματος σε άλλους τομείς. Οι υπηρεσίες γεωγραφικού εντοπισμού και οι έξυπνες κινητές συσκευές μας επιτρέπουν να διαχειριζόμαστε τις αιτήσεις και να λαμβάνουμε έγκαιρες αποφάσεις.

Κατανοώντας τις ανάγκες του κλάδου των αστικών εμπορευματικών μεταφορών υλοποιούμε μεθόδους που βελτιώνουν την αξιοποίηση του διαθέσιμου στόλου οχημάτων για την εξυπηρέτηση αιτήσεων παραγγελιών μεταξύ πολλών σημείων αφετηρίας και προορισμού με περιορισμούς ως προς τον χρόνο παράδοσης τους. Στην υλοποίηση μας λαμβάνουμε υπόψιν θεωρητικές προσεγγίσεις πάνω σε παρόμοια προβλήματα δρομολόγησης και αξιοποιούμε κατάλληλα τεχνολογικά εργαλεία που μας επιτρέπουν ένα χρήσιμο αποτέλεσμα.

## **ABSTRACT**

In this paper we are examining the problem of routing and scheduling a fleet of vehicles that seek to satisfy dynamic requests for packages pickup and delivery on an urban scale between two - per application - points. We are developing a system aiming at facilitating cooperation among entities and individuals who are primarily involved in the urban logistics and supply chain, nonetheless without limiting the applicability of the system to other sectors. Locating services and smart mobile devices allow us to handle requests and make timely decisions. By understanding the needs of the urban freight transport industry, we are implementing methods that improve the utilization of the available vehicle fleets so as to serve order requests among multiple start and destination (pickup and delivery) points imposing certain time constraints on delivery. In implementing this system, we take into consideration theoretical approaches over similar routing problems and we use appropriate technological tools that lead us to a useful result.

# ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

## Περιεχόμενα

<b>ΠΕΡΙΛΗΨΗ</b> .....	<b>vi</b>
<b>ABSTRACT</b> .....	<b>vii</b>
<b>ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ</b> .....	<b>viii</b>
<b>ΚΕΦΑΛΑΙΟ 1</b> .....	<b>1</b>
<b>ΕΙΣΑΓΩΓΗ</b> .....	<b>1</b>
1.1 Αντικείμενο της Διπλωματικής Εργασίας.....	<b>1</b>
1.2 Δομή της Διπλωματικής Εργασίας .....	<b>1</b>
<b>ΚΕΦΑΛΑΙΟ 2</b> .....	<b>3</b>
<b>ΘΕΩΡΗΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ</b> .....	<b>3</b>
2.1 Εισαγωγή.....	<b>3</b>
2.2 Travelling Salesman Problem (TSP) .....	<b>3</b>
2.3 Vehicle Routing Problem (VRP).....	<b>4</b>
2.4 Dial-a-Ride Problem (DARP) .....	<b>6</b>
2.5 Dynamic Vehicle Routing Problem (DVRP) .....	<b>6</b>
2.6 Vehicle Routing Pickup and Delivery Problem (VRPPD).....	<b>8</b>
2.7 Vehicle Routing Problem with Pickup and Delivery with Time Windows (VRPPDTW).....	<b>9</b>
2.8 Η προσέγγιση του συστήματος μας .....	<b>10</b>
<b>ΚΕΦΑΛΑΙΟ 3</b> .....	<b>11</b>
<b>ΤΕΧΝΟΛΟΓΙΕΣ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΕΡΓΑΛΕΙΑ</b> .....	<b>11</b>
3.1 Εισαγωγή.....	<b>11</b>
3.2 OR-Tools.....	<b>11</b>
3.3 OptaPlanner .....	<b>14</b>
3.4 Open Source Routing Machine (OSRM).....	<b>15</b>
3.5 Microsoft Bing Fleet Management.....	<b>16</b>
3.6 GraphHopper - Jsprit.....	<b>17</b>
3.7 Τεχνολογίες εφαρμογών πελάτη & διανομέα.....	<b>17</b>
3.8 Τεχνολογίες APIs (Application Programming Interfaces).....	<b>18</b>
<b>ΚΕΦΑΛΑΙΟ 4</b> .....	<b>20</b>



<b>ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ .....</b>	<b>20</b>
4.1 Εισαγωγή.....	20
4.2 Αρχιτεκτονική του συστήματος (Επισκόπηση) .....	20
4.3 Βάση Δεδομένων .....	21
4.4 Web App πελάτη .....	22
4.5 Mobile App διανομέα .....	26
4.6 Python εφαρμογή (server-side) .....	30
4.7 Παραδοχές .....	32
4.8 Προσομοίωση Λειτουργίας Συστήματος .....	32
<b>ΚΕΦΑΛΑΙΟ 5.....</b>	<b>34</b>
<b>ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΣΘΗΚΕΣ .....</b>	<b>34</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>36</b>
<b>ΠΑΡΑΡΤΗΜΑ Α.....</b>	<b>38</b>
<b>ΕΙΚΟΝΕΣ .....</b>	<b>38</b>

# ΚΕΦΑΛΑΙΟ 1

## ΕΙΣΑΓΩΓΗ

Στο παρόν κεφάλαιο πραγματοποιείται μια συνοπτική παρουσίαση του αντικειμένου της εργασίας, της δομής και του στόχου που εξυπηρετεί. Ακόμα, γίνεται σύντομη αναφορά στα κεφάλαια που ακολουθούν για πληροφόρηση του αναγνώστη.

### 1.1 Αντικείμενο της Διπλωματικής Εργασίας

Αντικείμενο της Διπλωματικής Εργασίας αποτελεί ο σχεδιασμός και η ανάπτυξη ενός ολοκληρωμένου πολυπρακτορικού συστήματος παραγγελιοληψίας μεταξύ πολλών σημείων αφετηρίας και προορισμού με περιορισμούς ως προς το χρόνο παράδοσης. Στόχος της εργασίας είναι η εμβάθυνση στο προς επίλυση πρόβλημα και ο συνδυασμός τεχνολογιών με μελέτη των σύγχρονων τάσεων στην ανάπτυξη ενός τέτοιου συστήματος.

Το θεωρητικό πρόβλημα που καλείται η εργασία να επιλύσει στην βασική του μορφή απαντάται στην βιβλιογραφία ως Vehicle Routing Problem (Πρόβλημα Δρομολόγησης Οχήματος) ως επέκταση του γνωστού Travel Salesman Problem (Πρόβλημα Πλανόδιου Πωλητή). Στο Vehicle Routing Problem δίνονται ως δεδομένα ένας στόλος οχημάτων, ένα σύνολο σταθμών ανεφοδιασμού και ένα σύνολο τοποθεσιών για αποστολή πακέτων.

Στο σύστημα που αναπτύξαμε και αναλύουμε στην παρούσα εργασία προσπαθούμε να μετατρέψουμε το πρόβλημα αυτό από στατικό σε δυναμικό και να προσεγγίσουμε τη λύση του ώστε να χρησιμοποιηθεί σε ανάγκες κατά τις οποίες δεν είναι γνωστά εξ' αρχής όλα τα δεδομένα, αλλά εμφανίζονται δυναμικά με την πάροδο του χρόνου μέσω αιτήσεων που γίνονται προς το σύστημα.

### 1.2 Δομή της Διπλωματικής Εργασίας

Στο Κεφάλαιο 2 γίνεται μια ανάλυση από την θεωρητική οπτική του προβλήματος που επιλύουμε επεξηγώντας όρους που έχουν να κάνουν με αυτό και τις εναλλακτικές εκφάνσεις με τις οποίες συναντάται έως σήμερα.

Στο Κεφάλαιο 3 παρουσιάζεται βήμα-βήμα η προγραμματιστική υλοποίηση των τριών μερών του συστήματος επιδεικνύοντας πώς μια παραγγελία καταχωρείται στο σύστημα, στη συνέχεια με ποιον τρόπο λαμβάνεται και τίθεται προς επεξεργασία και πώς τελικά καταλήγει ως εντολή εξυπηρέτησης δρομολογημένη στον διανομέα. Αναλύεται σύντομα η δομή της Βάσης Δεδομένων που χρησιμοποιείται, αναφέρονται πληροφορίες για την δοκιμή του συστήματος και τα συμπεράσματα που εξήχθησαν.

Στο Κεφάλαιο 4 γίνεται λόγος για παρόμοια λογισμικά που παρέχουν ολοκληρωμένη υπηρεσία διαχείρισης διανομών με παρεμφερή με τη δική μας προσέγγιση και ελεύθερα λογισμικά που μπορούν να ενσωματωθούν σε νέες υλοποιήσεις

Στο Κεφάλαιο 5 συνοψίζονται μερικά συμπεράσματα που εξήχθησαν κατά την διάρκεια και με την ολοκλήρωση την ανάπτυξης του συστήματος. Ακόμα, αναφέρονται μελλοντικές προοπτικές στη χρήση του με την υιοθέτηση τεχνολογικών εργαλείων.

## ΚΕΦΑΛΑΙΟ 2

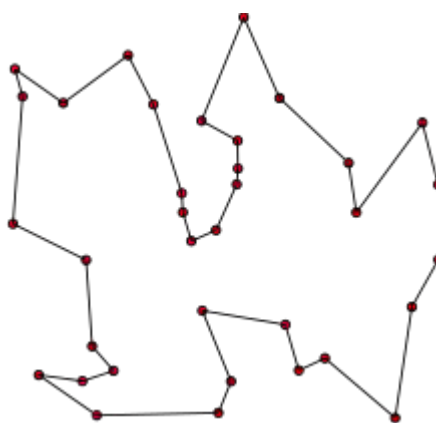
### ΘΕΩΡΗΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ

#### 2.1 Εισαγωγή

Στο κεφάλαιο αυτό εξηγούνται έννοιες που χρησιμοποιούνται στα προβλήματα δρομολόγησης, εντοπίζονται οι ιδιαιτερότητες του συστήματος μας και παρουσιάζεται η προτεινόμενη λύση. Οι έννοιες αυτές απαντούν σε γενικά προβλήματα δρομολόγησης οχημάτων και εξειδικεύουν ανάλογα με τις ανάγκες.

#### 2.2 Travelling Salesman Problem (TSP)

Το πρόβλημα του ταξιδιώτη πωλητή μπορεί να οριστεί ως εξής. Λαμβάνοντας ως δεδομένα έναν πλανόδιο πωλητή, ένα σύνολο πόλεων, τις αποστάσεις μεταξύ κάθε ζεύγους αυτών, στόχος είναι να βρεθεί η συντομότερη διαδρομή ώστε ο πωλητής να διέλθει από κάθε πόλη μία μόνο φορά και να επιστρέψει στην πόλη από την οποία ξεκίνησε. Πρόκειται για ένα NP-hard πρόβλημα στη συνδυαστική βελτιστοποίηση, που είναι σημαντικό στην επιχειρησιακή έρευνα και στη επιστήμη των υπολογιστών.



Εικόνα 1. Λύση ενός TSP: η γραμμή δείχνει το συντομότερο μονοπάτι που συνδέει κάθε κόκκινο σημείο-κόμβο [1]

Ως προς την θεωρία της υπολογιστικής πολυπλοκότητας, η έκδοση απόφασης του TSP ανήκει στην κλάση NP-complete προβλημάτων. Επομένως, είναι πιθανό ο χρόνος εκτέλεσης της χειρότερης περίπτωσης για οποιονδήποτε αλγόριθμο για το TSP να αυξάνεται εκθετικά με την αύξηση του αριθμού των πόλεων - κόμβων.

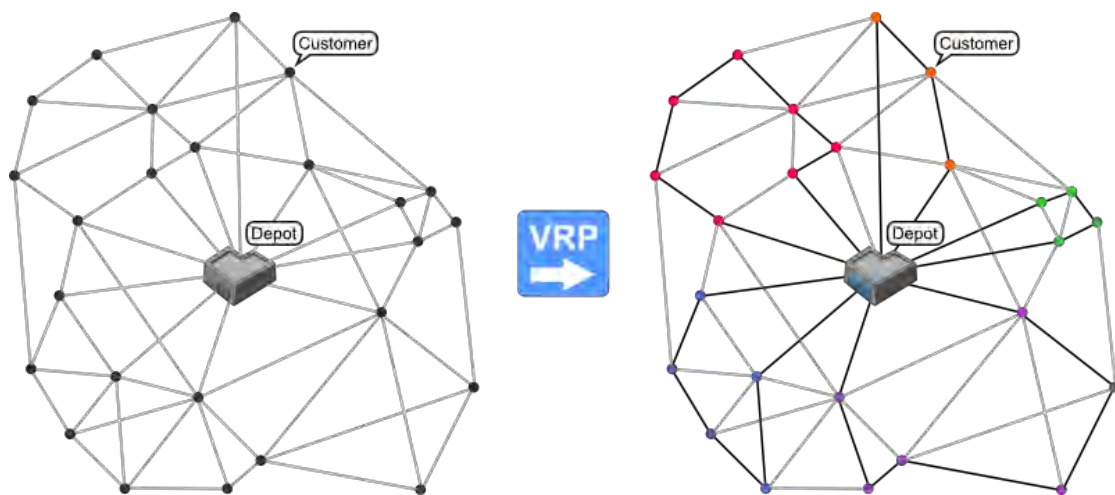
Για την ονομασία του προβλήματος δεν υπάρχει σαφής τεκμηρίωση για τον δημιουργό της. Το πρόβλημα διατυπώθηκε για πρώτη φορά το 1930 από τον ερευνητή Merrill Flood του Πανεπιστημίου Princeton και του ερευνητικού οργανισμού RAND Corporation και αποτελεί αδιαμφισβήτητα ένα από τα πιο έντονα μελετημένα προβλήματα βελτιστοποίησης. Χρησιμοποιείται ως σημείο αναφοράς για πολλές άλλες μεθόδους στην βελτιστοποίηση. Παρόλο που το πρόβλημα είναι υπολογιστικά δύσκολο, είναι γνωστός ένας μεγάλος αριθμός ευρετικών μεθόδων (heuristics) και αλγορίθμων, έτσι ώστε ορισμένες περιπτώσεις με δεκάδες χιλιάδες πόλεις να μπορούν να λυθούν πλήρως. Ακόμη και προβλήματα με εκατομμύρια πόλεις μπορούν να προσεγγιστούν σε ένα μικρό ποσοστό περί το 1% [2]. Οι μέθοδοι επίλυσης του TSP κατατάσσονται σε τρεις βασικές κατηγορίες: α) Τοπικής Αναζήτησης, β) Κατασκευής Δρομολογίων και γ) Μεταευρετικές (metaheuristics).

Το TSP έχει αρκετές εφαρμογές, όπως τα logistics, η κατασκευή μικροτσιπς κ.α. Ελαφρώς τροποποιημένο, εμφανίζεται ως μέρος ενός μεγαλύτερου προβλήματος σε πολλές επιστημονικές περιοχές, όπως για παράδειγμα η γενετική. Σε αυτές τις εφαρμογές, η έννοια της πόλης αντιπροσωπεύει, λόγου χάρη, πελάτες, σημεία συγκόλλησης ή τμήματα DNA και η έννοια της απόστασης αντιπροσωπεύει τον χρόνο ταξιδιού ή το κόστος ή ένα μέτρο ομοιότητας μεταξύ τμημάτων του DNA. Το TSP μπορεί να συναντηθεί ως μέρος μελέτης και στην αστρονομία, καθώς οι αστρονόμοι που παρατηρούν πολλές πηγές θα θέλουν να ελαχιστοποιήσουν το χρόνο που αφιερώνουν στη μετακίνηση του τηλεσκοπίου μεταξύ των πηγών. Σε πολλές εφαρμογές, μπορούν να επιβληθούν πρόσθετοι περιορισμοί, όπως περιορισμοί πόρων ή χρονικά παράθυρα.

### **2.3 Vehicle Routing Problem (VRP)**

Το Vehicle Routing Problem είναι ένα πολύ γνωστό συνδυαστικό πρόβλημα βελτιστοποίησης και ακέραιου προγραμματισμού. Λαμβάνει ως δεδομένα μία ή και περισσότερες αποθήκες, ένα σύνολο παραδόσεων που πρέπει να εξυπηρετηθούν και έναν στόλο διανομέων και στόχος του είναι η εύρεση του βέλτιστου συνόλου διαδρομών για την εξυπηρέτηση του δεδομένου συνόλου παραδόσεων. Γενικεύει το πρόβλημα του

ταξιδιώτη πωλητή (TSP). Πρώτη του εμφάνιση εντοπίζεται από τους George Dantzig και John Ramser το 1959 σε μια αναφορά [3], στην οποία συναντάται η πρώτη αλγοριθμική προσέγγιση με πεδίο εφαρμογής τις παραδόσεις βενζίνης. Συχνά, το πλαίσιο είναι η παράδοση αγαθών που βρίσκονται σε κεντρική αποθήκη σε πελάτες που έχουν παραγγείλει τέτοια αγαθά με στόχο του VRP να ελαχιστοποιήσει όσο το δυνατόν το συνολικό κόστος της διαδρομής. Το VRP έχει προκαλέσει ιδιαίτερο ενδιαφέρον από πολλούς ερευνητές κατά τις τελευταίες δεκαετίες, λόγω του ζωτικού ρόλου που διαδραματίζει στον σχεδιασμό των συστημάτων διανομής και της εφοδιαστικής αλυσίδας σε πολλούς τομείς, όπως συλλογή απορριμμάτων, παράδοση αλληλογραφίας, διαδικασία αποχιονισμού οδικού δικτύου κ.α.



Εικόνα 2. Λύση VRP με μία αποθήκη [4]

Οι Clarke και Wright, το 1964 [5] βελτίωσαν την προσέγγιση των Dantzig και Ramser χρησιμοποιώντας μια προσέγγιση βασισμένη στον συνδυασμό δύο πελατών στην ίδια διαδρομή. Η εν λόγω προσέγγιση ονομάζεται αλγόριθμος των Clarke και Wright ή αλγόριθμος εξοικονόμησης (the savings algorithm). Λαμβάνει ως δεδομένα τις προς εξυπηρέτηση παραγγελίες, τους προορισμούς, το διαθέσιμο στόλο διανομικών και τη μία ή περισσότερες αποθήκες από όπου παραλαμβάνονται τα προς παράδοση πακέτα. Η επίλυση του έχει σαν αποτέλεσμα τον καθορισμό ενός συνόλου διαδρομών για ένα υποσύνολο ή και το σύνολο των διαθέσιμων διανομικών. Οι διαδρομές αυτές ελαχιστοποιούν το κόστος διανομής. Το κόστος αυτό μπορεί να είναι οικονομικό, απόστασης ή άλλου είδους.

## **2.4 Dial-a-Ride Problem (DARP)**

Το πρόβλημα Dial-a-Ride (DARP) συνίσταται στο σχεδιασμό ενός συνόλου διαδρομών ελάχιστου κόστους για την ικανοποίηση ενός συνόλου αιτημάτων μεταφοράς. Κάθε αίτημα περιλαμβάνει τη μεταφορά ενός συνόλου χρηστών από ένα σύνολο σημείων αφετηρίας, που αντιστοιχούν σε σημεία παραλαβής σε ένα σύνολο προορισμών, που αντιστοιχούν σε σημεία παράδοσης. Οι χρήστες που σχετίζονται με ξεχωριστά αιτήματα μπορούν να μοιράζονται το ίδιο όχημα, εφόσον δεν υπάρχει υπέρβαση της χωρητικότητας του. Επιπλέον, ο μέγιστος χρόνος οδήγησης συνδέεται με κάθε αίτημα. Αντιστοιχεί στη μέγιστη διάρκεια της διαδρομής μεταξύ του σημείου παραλαβής και του σημείου παράδοσης. Στην συμβατική έκδοση, η μεταφορά παρέχεται από ένα στόλο από πανομοιότυπα οχήματα με βάση την ίδια αποθήκη. Ο στόχος είναι να σχεδιαστεί ένα σύνολο ελάχιστων οχημάτων ικανών να εξυπηρετήσουν όσο το δυνατόν περισσότερες αιτήσεις, υπακούοντας σε μια σειρά περιορισμών. Το πιο συνηθισμένο παράδειγμα προκύπτει στις υπηρεσίες μεταφοράς από πόρτα σε πόρτα για ηλικιωμένους ή άτομα με ειδικές ανάγκες. Ο Αθ. Λόης το 2010 προσέγγισε μία πτυχή του DARP, αναφέροντας την ως online Dial-a-Ride Problem που ασχολείται με τη συνεχή ροή των απαιτήσεων ταξιδιού κατά τη διάρκεια της λειτουργίας, κάτι που απαιτεί κατασκευή online αλγορίθμων, ικανών να προσαρμοστούν στη συνεχή ροή των εισερχόμενων πληροφοριών. Η μελέτη του γίνεται με χρήση έξι διαφορετικών dial-a-ride αλγορίθμων. [6]

## **2.5 Dynamic Vehicle Routing Problem (DVRP)**

Η πρώτη αναφορά σε ένα δυναμικό πρόβλημα δρομολόγησης οχήματος γίνεται από τους Wilson και Colvin [7]. Αυτοί μελέτησαν ένα Dial-a-Ride πρόβλημα ενός οχήματος, στο οποίο τα αιτήματα πελατών είναι διαδρομές από μια περιοχή σε έναν προορισμό που εμφανίζεται δυναμικά. Η προσέγγιση τους χρησιμοποιεί εισαγωγή ευρετικών (heuristics) που αποδίδουν καλά, δηλαδή με χαμηλό υπολογιστικό κόστος. Οι Bagchi και Nag πρότειναν μια εξειδικευμένη προσέγγιση του συστήματος για τη δυναμική δρομολόγηση των οχημάτων. [8] Ύστερα από αυτό, ο Har. Psaraftis εισήγαγε την λογική των άμεσων αιτημάτων [9], κατά τα οποία μια υπηρεσία εξυπηρέτησης αιτημάτων πελατών επιθυμεί

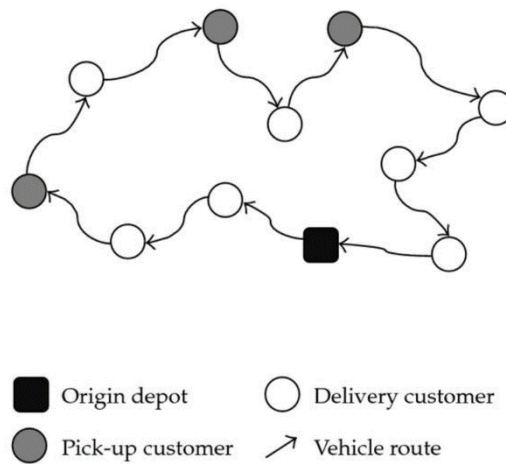
πάντα να εξυπηρετείται όσο το δυνατόν νωρίτερα, απαιτώντας άμεσο επανασχεδιασμό του τρέχοντος δρομολογίου του οχήματος. Μια ειδική έκδοση αυτού του προβλήματος είναι η έκδοση με τον παράγοντα της χωρητικότητας. Σε αυτήν την έκδοση του προβλήματος οι πελάτες πρέπει να καταταχθούν σε διαδρομές που να ικανοποιούν τους περιορισμούς της χωρητικότητας των οχημάτων. Το πρόβλημα αυτό αναφέρεται συνήθως ως Capacitated Vehicle Routing Problem (CVRP). Οι Bertsimas και van Ryzin [10] εξετάζουν την περίπτωση όπου μια συγκεκριμένη περιοχή εξυπηρετείται από έναν ομοιογενή στόλο  $m$  οχημάτων, που εκτελείται από  $m$  αποθήκες των οποίων οι θέσεις δεν χρειάζεται να είναι ξεχωριστές, όπου κάθε όχημα περιορίζεται στην εξυπηρέτηση ανώτατου ορίου πελατών πριν επιστρέψει στην αποθήκη. Οι Fleischmann et al. [11] πρότειναν ένα πλαίσιο που προσαρμόζεται σε online πληροφορίες κυκλοφορίας για μια συγκεκριμένη εφαρμογή του προβλήματος δυναμικής δρομολόγησης, που αποτελείται από τρεις στρατηγικές (προγραμματισμός με κανόνες εκχώρησης, προγραμματισμός με αλγόριθμο εκχώρησης, προγραμματισμός με αλγόριθμο εισαγωγής), οι οποίες καλύπτουν διάφορες εφαρμογές. Μια πληθώρα τεχνολογικών επιτευγμάτων έχει οδηγήσει στον πολλαπλασιασμό των εφαρμογών δρομολόγησης πραγματικού χρόνου. Με την εμφάνιση του Global Positioning System (GPS) το 1996, η ανάπτυξη και εξάπλωση της χρήσης smartphones, σε συνδυασμό με ακριβή Geographic Information Systems (GIS), έκανε τις εταιρείες ικανές να παρακολουθούν και να διαχειρίζονται τον στόλο οχημάτων τους σε πραγματικό χρόνο και με αποδοτικό κόστος. Από την παραδοσιακή δύο-βημάτων διαδικασία (σχεδιασμός-εκτέλεση), η δρομολόγηση οχημάτων μπορεί τώρα να γίνεται δυναμικά, δημιουργώντας νέες ευκαιρίες στην μείωση λειτουργικού κόστους, βελτίωση της εξυπηρέτησης των πελατών και μείωση των περιβαλλοντικών επιπτώσεων της χρήσης οχημάτων. Η προέλευση της δυναμικής ιδιότητας στην δρομολόγηση οχημάτων είναι η online άφιξη αιτημάτων πελατών κατά την διάρκεια της λειτουργίας. Ειδικότερα, τα αιτήματα μπορεί να είναι η ζήτηση για αγαθά ή υπηρεσίες. Σε αντίθεση με τα αντίστοιχα στατικά τους, τα δυναμικά προβλήματα δρομολόγησης περιλαμβάνουν νέα στοιχεία που αυξάνουν την πολυπλοκότητα των αποφάσεων και εισάγουν νέες προκλήσεις καθώς κρίνουν την αξία ενός δοσμένου δρομολογίου. Σε μερικές περιπτώσεις, όπως η παραλαβή του ταχυμεταφορέα, η μεταφορική εταιρεία ίσως αρνηθεί ένα αίτημα πελάτη. Σαν συνακόλουθο, μπορεί να απορρίψει ένα αίτημα είτε λόγω του ότι είναι απίθανο να το εξυπηρετήσει, είτε επειδή το κόστος εξυπηρέτησης του είναι πολύ υψηλό. Αυτή η



διαδικασία αποδοχής ή απόρριψης έχει χρησιμοποιηθεί σε πολλές προσεγγίσεις και αναφέρεται ως εγγύηση εξυπηρέτησης. Στην δυναμική δρομολόγηση, η ικανότητα ανακατεύθυνσης ενός κινούμενου οχήματος σε ένα νέο κοντινό αίτημα επιτρέπει πρόσθετη εξοικονόμηση. Παρόλα αυτά, προϋπόθεση είναι η γνώση της θέσης του οχήματος σε πραγματικό χρόνο και η ικανότητα άμεσης επικοινωνίας με τους οδηγούς ώστε να ανατεθούν στο δρομολόγιο τους νέοι κόμβοι. Αυτή η προσέγγιση έχει λάβει μειωμένου ενδιαφέροντος με κύριες συνεισφορές να έχουν ξεκινήσει από την Amelia Regan το 1995 [12], την μελέτη των ζητημάτων εκτροπής του Ichoua το 2000 [13] και εν συνεχεία των Branchini, Armentano και Lokketangen το 2009 [14]. Επίσης, στην δυναμική δρομολόγηση συχνά υπηρετείται διαφορετικός στόχος. Πιο συγκεκριμένα, ενόσω ο κοινός σκοπός στην στατική εκδοχή είναι η ελαχιστοποίηση του κόστους διαδρομής, στην δυναμική εκδοχή ίσως εισαχθούν νέες έννοιες όπως το επίπεδο εξυπηρέτησης, ο αριθμός εξυπηρετούμενων αιτημάτων, ή η μεγιστοποίηση των εσόδων. Σε αυτή την λογική προστίθεται και η έννοια του χρόνου απόκρισης, καθώς ένας πελάτης πιθανών επιθυμεί το αίτημα του να εξυπηρετηθεί όσο συντομότερα γίνεται, στην περίπτωση αυτή κύριος σκοπός ίσως είναι η ελαχιστοποίηση της καθυστέρησης μεταξύ της άφιξης ενός αιτήματος και της εξυπηρέτησης του.

## **2.6 Vehicle Routing Pickup and Delivery Problem (VRPPD)**

Στο πρόβλημα δρομολόγησης οχήματος με παραλαβή και παράδοση (VRPPD), ένας στόλος οχημάτων που έχει ως βάση ένα σύνολο τερματικών σταθμών πρέπει να ικανοποιεί ένα σύνολο αιτημάτων μεταφοράς. Κάθε αίτημα ορίζεται από ένα σημείο παραλαβής, ένα αντίστοιχο σημείο παράδοσης και μια ζήτηση που πρέπει να μεταφερθεί μεταξύ αυτών των θέσεων-κόμβων. Η λογική των παραπάνω προσεγγίσεων θεωρεί πως αφητηρία όλων των πακέτων είναι μια αποθήκη (depot). Συνεπώς, αντιμετωπίζουμε το πρόβλημα υιοθετώντας μια λογική ύπαρξης εξαρτήσεων μεταξύ αφητηρίας και προορισμού για εξυπηρέτηση μιας παραγγελίας που εισάγεται στο σύστημα με στόχο να εξυπηρετηθεί η παραλαβή της και η παράδοση της στις αντίστοιχες τοποθεσίες. Το VRP μπορεί να θεωρηθεί ως VRPPD στο οποίο τα οχήματα ξεκινούν από την αποθήκη με όλες τις παραλαβές για τη διαδρομή τους και κάθε επίσκεψη σε μια τοποθεσία της διαδρομής μπορεί να συνδεθεί με μια παράδοση.



Εικόνα 3. VRPPD ενός οχήματος [15]

## 2.7 Vehicle Routing Problem with Pickup and Delivery with Time Windows (VRPPDTW)

Η προηγούμενη προσέγγιση του προβλήματος παραλαβής και παράδοσης επεκτείνεται με τη χρήση χρονικών παραθύρων ως προς το περιθώριο παραλαβής από την αφετηρία και το περιθώριο παράδοσης στον προορισμό. Αναφέρεται, συχνά, και ως Pickup and Delivery Problem with Time Windows (PDPTW). Το εν λόγω πρόβλημα, λοιπόν, μοντελοποιεί την προσέγγιση κατά την οποία ένας στόλος οχημάτων πρέπει να εξυπηρετεί ένα πλήθος αιτημάτων μεταφοράς. Κάθε αίτημα καθορίζει μια θέση παραλαβής και μία θέση παράδοσης. Το όχημα πρέπει να δρομολογείται για την εξυπηρέτηση όλων των αιτημάτων, ικανοποιώντας τα χρονικά παράθυρα και τους περιορισμούς χωρητικότητας του οχήματος, ενώ ταυτόχρονα να βελτιστοποιεί μια συγκεκριμένη αντικειμενική συνάρτηση, όπως η συνολική απόσταση που πρόκειται να διανυθεί. Η χρήση των χρονικών παραθύρων σαν επέκταση του VRPPD διαθέτει και αυτή ποικίλες πρακτικές εφαρμογές, εν παραδείγματι του τομέα της εφοδιαστικής αλυσίδας ή όπως τη μεταφορά ατόμων με αναπηρία και ηλικιωμένων, τη μεταφορά αεροσκαφών και στρατευμάτων, καθώς και την παραλαβή και παράδοση για ολονύκτιους μεταφορείς ή λοιπές αστικές υπηρεσίες. Η εύρεση καλών λύσεων σε αυτά τα προβλήματα είναι σημαντική, διότι επιτρέπει στους υπεύθυνους σχεδιασμού να αξιοποιούν τον υπάρχοντα στόλο με τον αποδοτικότερο τρόπο από πλευράς κόστους, ώστε να ανταποκρίνονται στις απαιτήσεις των πελατών. Οι

προοπτικές σε αυτό το αναπτυσσόμενο πεδίο προσφέρθηκαν από τους Solomon και Desrosiers [16], Desrosiers et al. [17], και Savelsbergh και Sol [18].

## **2.8 Η προσέγγιση του συστήματος μας**

Για την υλοποίηση του συστήματος μας χρειάστηκε να κατασκευάσουμε μια αλγοριθμική αλληλουχία που εξασφαλίζει στο δικό μας σύστημα δυναμική υποστήριξη σε πραγματικό χρόνο ενώ επιλύει και ένα πρόβλημα δρομολόγησης με παραλαβές και παραδόσεις, με αυστηρές εξαρτήσεις μεταξύ αυτών, σεβόμενο χρονικά περιθώρια και δεδομένου κάθε στιγμή ενός ευέλικτου στόλου διανομένων. Πρόκειται επί της ουσίας για ένα συνδυασμό εννοιών και προβλημάτων όπως αναφέρθηκαν.

Καθορίσαμε την αντιμετώπιση της δρομολόγησης με τον ακόλουθο τρόπο. Αρχικά, έχουμε μια νέα καταχώρηση παραγγελίας προς εξυπηρέτηση στο σύστημα. Αμέσως, πραγματοποιείται εντοπισμός όλων των διαθέσιμων διανομένων εκείνη την χρονική στιγμή. Κατόπιν, υπολογίζονται και συγκρίνονται οι αποστάσεις των δύο κόμβων της νέας παραγγελίας (σημείο αφετηρίας και σημείο προορισμού) από όλους τους κόμβους που έχουν ενταχθεί έως τη δεδομένη χρονική στιγμή στο δρομολόγιο κάθε διαθέσιμου διανομέα, αλλά δεν έχουν ακόμα εξυπηρετηθεί. Ο υπολογισμός των αποστάσεων γίνεται με τη βοήθεια της φόρμουλας Haversine [19].

Η παραγγελία ανατίθεται στον διανομέα με την μικρότερη απόσταση κόμβων. Για τον διανομέα που επιλέχθηκε, επιλύεται εκ νέου πρόβλημα δρομολόγησης ως VRPPDTW μαζί με του δύο νέους κόμβους παραλαβής και παράδοσης.

## ΚΕΦΑΛΑΙΟ 3

### ΤΕΧΝΟΛΟΓΙΕΣ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΕΡΓΑΛΕΙΑ

#### 3.1 Εισαγωγή

Στο Κεφάλαιο αυτό γίνεται αναφορά σε έτοιμα εργαλεία τα οποία είναι διαθέσιμα στο διαδίκτυο και μπορούν να χρησιμοποιηθούν για την επίλυση του προβλήματος της δρομολόγησης κατά την ανάπτυξη ενός ανάλογου συστήματος με αυτό που υλοποιήθηκε. Δίνονται, επίσης, πληροφορίες για τεχνολογίες και εργαλεία που χρησιμοποιήθηκαν στην ανάπτυξη του συστήματος που παρουσιάζουμε.

#### 3.2 OR-Tools

Το OR-Tools [20] είναι μια σουίτα λογισμικού ανοιχτού κώδικα για βελτιστοποίηση, δημιουργημένη για την αντιμετώπιση προβλημάτων στον τομέα της δρομολόγησης οχημάτων, ρών, ακέραιου και γραμμικού προγραμματισμού καθώς και constraint programming (CP). Η σουίτα αυτή προσφέρεται από την Google. Στη σουίτα αυτή περιλαμβάνονται:

- Ένας CP solver
- Ένα απλό και ενοποιημένο interface με πληθώρα solvers γραμμικού και ακέραιου προγραμματισμού, όπως CBC, CLP, GLOP, GLPK, Gurobi, CPLEX και SCIP.
- Αλγορίθμους πλήρωσης σακιδίου και κάδου
- Αλγορίθμους για το Traveling Salesman Problem και Vehicle Routing Problem
- Αλγορίθμους γράφου (μικρότερο μονοπάτι, ροή ελαχίστου κόστους, μέγιστη ροή)

Η Google αναπτύσσει τα OR-Tools σε C++, αλλά παρέχει APIs ώστε να μπορούν να χρησιμοποιηθούν και από άλλες γλώσσες όπως Python, Java και C#.

Το σύστημα της παρούσας διπλωματικής εργασίας, χρησιμοποιεί για την δρομολόγηση των διανομμένων τον επιλύτη (solver) που παρέχεται από τα OR-Tools. Για την χρήση του solver στην επεξεργασία που πραγματοποιείται από το σύστημα χρειάζεται απόφαση του διαχειριστή για μια σειρά παραμέτρων που θα χρησιμοποιηθούν για την επίλυση της δρομολόγησης.

### Όρια αναζήτησης και τερματισμός της επίλυσης

Αρχικά, ο διαχειριστής πρέπει να αποφασίσει για τα όρια αναζήτησης που τερματίζουν την επίλυση όταν προσεγγιστεί ένα συγκεκριμένο όριο, όπως το μέγιστο χρονικό διάστημα ή το απαιτούμενο πλήθος λύσεων που βρέθηκαν. Ο επιλύτης παρέχει τις ακόλουθες παραμέτρους για καθορισμό των ορίων.

Name	Type	Default	Description
<code>solution_limit</code>	int64	kint64max	Limit to the number of solutions generated during the search.
<code>time_limit.seconds</code>	int64	kint64max	Limit in seconds to the time spent in the search.
<code>lns_time_limit.seconds</code>	int64	100	Limit in seconds to the time spent in the completion search for each local search neighbor.

Εικόνα 4. Google OR-Tools όρια του solver [21]

### Στρατηγική πρώτης λύσης

Επόμενη απόφαση που ο διαχειριστής του συστήματος χρειάζεται να λάβει είναι η στρατηγική πρώτης λύσης μέσα από μια σειρά διαθέσιμων επιλογών που δίνονται από τον solver ως παράμετροι. Οι επιλογές φαίνονται ακολούθως.

Option	Description
AUTOMATIC	Lets the solver detect which strategy to use according to the model being solved.
PATH_CHEAPEST_ARC	Starting from a route "start" node, connect it to the node which produces the cheapest route segment, then extend the route by iterating on the last node added to the route.
PATH_MOST_CONSTRAINED_ARC	Similar to <b>PATH_CHEAPEST_ARC</b> , but arcs are evaluated with a comparison-based selector which will favor the most constrained arc first. To assign a selector to the routing model, use the method <code>ArcIsMoreConstrainedThanArc()</code> .
EVALUATOR_STRATEGY	Similar to <b>PATH_CHEAPEST_ARC</b> , except that arc costs are evaluated using the function passed to <code>SetFirstSolutionEvaluator()</code> .
SAVINGS	Savings algorithm (Clarke & Wright). Reference: Clarke, G. & Wright, J.W.: "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", Operations Research, Vol. 12, 1964, pp. 568-581.
SWEEP	Sweep algorithm (Wren & Holliday). Reference: Anthony Wren & Alan Holliday: Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Points Operational Research Quarterly (1970-1977), Vol. 23, No. 3 (Sep., 1972), pp. 333-344.
CHRISTOFIDES	Christofides algorithm (actually a variant of the Christofides algorithm using a maximal matching instead of a maximum matching, which does not guarantee the 3/2 factor of the approximation on a metric travelling salesman). Works on generic vehicle routing models by extending a route until no nodes can be inserted on it. Reference: Nicos Christofides, Worst-case analysis of a new heuristic for the travelling salesman problem, Report 388, Graduate School of Industrial Administration, CMU, 1976.
ALL_UNPERFORMED	Make all nodes inactive. Only finds a solution if nodes are optional (are element of a disjunction constraint with a finite penalty cost).
BEST_INSERTION	Iteratively build a solution by inserting the cheapest node at its cheapest position; the cost of insertion is based on the global cost function of the routing model. As of 2/2012, only works on models with optional nodes (with finite penalty costs).
PARALLEL_CHEAPEST_INSERTION	Iteratively build a solution by inserting the cheapest node at its cheapest position; the cost of insertion is based on the arc cost function. Is faster than <b>BEST_INSERTION</b> .
LOCAL_CHEAPEST_INSERTION	Iteratively build a solution by inserting each node at its cheapest position; the cost of insertion is based on the arc cost function. Differs from <b>PARALLEL_CHEAPEST_INSERTION</b> by the node selected for insertion; here nodes are considered in their order of creation. Is faster than <b>PARALLEL_CHEAPEST_INSERTION</b> .
GLOBAL_CHEAPEST_ARC	Iteratively connect two nodes which produce the cheapest route segment.
LOCAL_CHEAPEST_ARC	Select the first node with an unbound successor and connect it to the node which produces the cheapest route segment.
FIRST_UNBOUND_MIN_VALUE	Select the first node with an unbound successor and connect it to the first available node. This is equivalent to the <b>CHOOSE_FIRST_UNBOUND</b> strategy combined with <b>ASSIGN_MIN_VALUE</b> .

Εικόνα 5. Google OR-Tools επιλογές στρατηγικής πρώτης λύσης [21]

Ο επιλύτης δρομολόγησης δεν επιστρέφει πάντοτε τη βέλτιστη λύση σε ένα TSP, επειδή τα προβλήματα δρομολόγησης είναι υπολογιστικά απαιτητικά. Προκειμένου να βρεθεί μια καλύτερη λύση, μπορεί να χρησιμοποιηθεί μια πιο προηγμένη στρατηγική αναζήτησης, που ονομάζεται καθοδηγούμενη τοπική αναζήτηση, η οποία δίνει τη δυνατότητα στον επιλύτη να ξεφύγει από ένα τοπικό ελάχιστο - μια λύση που είναι μικρότερη από όλες τις κοντινές διαδρομές, αλλά που δεν είναι το ολικό ελάχιστο. Αφού

απομακρυνθεί από το τοπικό ελάχιστο, συνεχίζει την αναζήτηση. Οι στρατηγικές τοπικής αναζήτησης που διατίθενται εντοπίζονται ως εξής.

Option	Description
AUTOMATIC	Lets the solver select the metaheuristic.
GREEDY_DESCENT	Accepts improving (cost-reducing) local search neighbors until a local minimum is reached.
GUIDED_LOCAL_SEARCH	Uses guided local search to escape local minima (cf. <a href="http://en.wikipedia.org/wiki/Guided_Local_Search">http://en.wikipedia.org/wiki/Guided_Local_Search</a> ); this is generally the most efficient metaheuristic for vehicle routing.
SIMULATED_ANNEALING	Uses simulated annealing to escape local minima (cf. <a href="http://en.wikipedia.org/wiki/Simulated_annealing">http://en.wikipedia.org/wiki/Simulated_annealing</a> ).
TABU_SEARCH	Uses tabu search to escape local minima (cf. <a href="http://en.wikipedia.org/wiki/Tabu_search">http://en.wikipedia.org/wiki/Tabu_search</a> ).
OBJECTIVE_TABU_SEARCH	Uses tabu search on the objective value of solution to escape local minima

Εικόνα 6. Google OR-Tools επιλογές στρατηγικής τοπικής αναζήτησης [21]

Στην περίπτωση της χρήσης του συστήματος στη διανομή γρήγορου φαγητού σε μια πόλη σαν τον Βόλο περιορίζουμε τον επιλύτη μεταξύ 1000 και 2000 miliseconds λόγω των περιορισμών που θέτουμε εξ αρχής με αποτέλεσμα να έχουμε λίγους κόμβους της τάξης των δεκάδων για την επίλυση. Ο περιορισμός αυτός απαιτεί αναπροσαρμογή σε περιπτώσεις χρήσης του συστήματος με εμφανή κλιμάκωση παραμέτρων, όπως το πλήθος των διανομέων ή η έκταση της γεωγραφικής περιοχής.

Επίσης, δεν χρειάζεται, για τον ίδιο λόγο, να κάνουμε τοπική αναζήτηση. Ως προς την στρατηγική πρώτης λύσης χρησιμοποιούμε την “PATH\_CHEAPEST\_ARC”, η οποία λειτουργεί ως εξής: ξεκινώντας από έναν κόμβο «εκκίνησης», τον συνδέει στον κόμβο που παράγει το φθηνότερο τμήμα διαδρομής και στη συνέχεια επεκτείνει τη διαδρομή μεταβαίνοντας στον τελευταίο κόμβο που προστέθηκε στη διαδρομή.

### 3.3 OptaPlanner

Το OptaPlanner [22] είναι λογισμικό ανοικτού κώδικα το οποίο επιλύει προβλήματα προγραμματισμού εργασιών. Είναι γραμμένο σε Java και παρέχει ένα Java API για την κλήση των αλγορίθμων που υλοποιούνται από αυτό.

Επιλύει περιπτώσεις προβλημάτων όπως οι ακόλουθες:

- Χρονικός προγραμματισμός εργαζομένων: νοσηλευτές, επισκευαστές
- Προγραμματισμός ατζέντας: προγραμματισμός συναντήσεων, ραντεβού, εργασιών συντήρησης, διαφημίσεις
- Εκπαιδευτικό ωρολόγιο πρόγραμμα: προγραμματισμός μαθημάτων ή εξετάσεων, παρουσιάσεις συνεδρίων
- Ροή οχημάτων: σχεδιασμός διαδρομών οχημάτων (φορτηγά, τρένα, σκάφη, αεροπλάνα) για τη μεταφορά εμπορευμάτων ή / και επιβατών μέσω πολλαπλών προορισμών χρησιμοποιώντας γνωστά εργαλεία χαρτογράφησης
- Συσκευασία δεξαμενών: γέμισμα εμπορευματοκιβωτίων, φορτηγών, πλοίων και αποθηκών αποθήκευσης με αντικείμενα, αλλά και πληροφορίες για τη συσκευασία σε πόρους ηλεκτρονικών υπολογιστών, όπως στο cloud computing
- Προγραμματισμός εργοταξίου: σχεδιασμός γραμμών συναρμολόγησης αυτοκινήτων, προγραμματισμός ουράς μηχανών, προγραμματισμός εργασιών
- Κοπή αποθέματος: ελαχιστοποίηση των αποβλήτων κατά την κοπή χαρτιού, χάλυβα, χαλί
- Αθλητικός προγραμματισμός: σχεδιασμός παιχνιδιών και προγραμμάτων κατάρτισης για ποδοσφαιρικά πρωταθλήματα, πρωταθλήματα μπίτζμπολ
- Χρηματοοικονομική βελτιστοποίηση: βελτιστοποίηση επενδυτικού χαρτοφυλακίου, διάδοση κινδύνου

### **3.4 Open Source Routing Machine (OSRM)**

Το Open Source Routing Machine [23] είναι ένας μηχανισμός δρομολόγησης υψηλής απόδοσης για συντομότερα μονοπάτια του οδικού δικτύου υλοποιημένος σε C++. Συνδυάζει τους εξελιγμένους αλγορίθμους δρομολόγησης με τα ανοικτά και δωρεάν δεδομένα οδικού δικτύου του έργου OpenStreetMap. Ο συντομότερος υπολογισμός διαδρομής σε δίκτυο ηπειρωτικών διαστάσεων μπορεί να διαρκέσει μερικά δευτερόλεπτα, αν γίνει χωρίς την αποκαλούμενη τεχνική επιτάχυνσης. Το OSRM χρησιμοποιεί μια εφαρμογή των ιεραρχιών συστολής (contraction hierarchies) και είναι σε



θέση να υπολογίσει και να εξάγει μια συντομότερη διαδρομή μεταξύ οποιασδήποτε προέλευσης και προορισμού μέσα σε λίγα χιλιοστά του δευτερολέπτου, οπότε ο καθαρός υπολογισμός της διαδρομής απαιτεί πολύ λιγότερο χρόνο. Οι περισσότερες προσπάθειες δαπανώνται για τη σήμανση της διαδρομής και τη μετάδοση της γεωμετρίας μέσω του δικτύου. Δεδομένου ότι έχει σχεδιαστεί να είναι συμβατό με το OpenStreetMap, τα αρχεία δεδομένων OSM μπορούν να εισαχθούν εύκολα.

### **3.5 Microsoft Bing Fleet Management**

Πρόκειται για μία νέα σουίτα υπηρεσιών που παρέχεται εντός των Χαρτών Bing από την Microsoft με σκοπό την βελτιστοποίηση των λειτουργιών ενός στόλου οχημάτων μαζί με μείωση του κόστους καθώς και την βελτίωση CRM που περιλαμβάνουν διαχείριση στόλου και υπηρεσίες logistics [24].

Παρέχονται τα εξής:

- Distance Matrix API: Υπολογίζει τους χρόνους ταξιδιού και την απόσταση για σενάρια «πολλά προς πολλά», με προαιρετικό ιστόγραμμα για την πρόβλεψη της κυκλοφορίας.
- Isochrone API: Υπολογίζει την περιοχή που μπορεί να διανυθεί μέσα σε καθορισμένη απόσταση ή σε καθορισμένο χρόνο
- Truck Routing API: Προσδιορίζει τις διαδρομές ταξιδιού λαμβάνοντας μέριμνα για τις ιδιότητες ενός φορτηγού ή επαγγελματικού οχήματος (π.χ. περιορισμοί μεγέθους, ταχύτητας, νομικές κ.α.
- Snap to Road API: Κατά την πλοήγηση μέσω του GPS, μερικές φορές το ίχνος ή οι συντεταγμένες δεν ταιριάζουν πάντα με το δρόμο ή το μονοπάτι που διανύθηκε. Χρησιμοποιώντας το ίχνος GPS των οχημάτων, καθώς και τα χαρακτηριστικά οδού επιστροφής, όπως το όριο ταχύτητας, οδηγεί σε μονοπάτια που δείχνουν πιο λογικά βάσει δεδομένων.

### 3.6 GraphHopper - Jsprit

Το GraphHopper είναι μια μηχανή δρομολόγησης της JAVA παρέχοντας API κατευθύνσεων διαδρομών [25]. Μαζί με αυτό έχει αναπτυχθεί και το εργαλείο Jsprit. Το Jsprit είναι open-source εργαλείο της JAVA ειδικά για επίλυση Traveling Salesman Problems και Vehicle Routing Problems [26]. Βασίζεται σε μια ενιαία μετα-ευρετική λύση για:

- Capacitated VRP
- Multiple Depot VRP
- VRP with Time Windows
- VRP with Backhauls
- VRP with Pickups and Deliveries
- VRP with Heterogeneous Fleet
- Time-dependent VRP
- Traveling Salesman Problem
- Dial-a-Ride Problem
- Various combination of these types

### 3.7 Τεχνολογίες εφαρμογών πελάτη & διανομέα

Η λειτουργία του συστήματός επιτελείται -όσον αφορά τους χρήστες- από δύο εφαρμογές, μια Web Progressive εφαρμογή (WPA) και μία Hybrid Mobile εφαρμογή, αναπτυγμένες αμφότερες σε Ionic Framework version 3, αναφορά στο οποίο γίνεται στη συνέχεια. Κύρια λειτουργία της πρώτης εφαρμογής είναι η εισαγωγή νέων παραγγελιών στο σύστημα από ταυτοποιημένους χρήστες προς δρομολόγηση και εξυπηρέτηση τους, ενώ κύρια λειτουργία της δεύτερης εφαρμογής είναι να ενημερώνει τους ταυτοποιημένους διανομείς για τα δρομολόγια που πρέπει να εκτελέσουν για να εξυπηρετήσουν τις καταχωρημένες στο σύστημα παραγγελίες.

#### 3.7.1 Ionic Framework

Το Ionic Framework [27] είναι ένα open-source εργαλείο UI για τη δημιουργία υψηλής ποιότητας εφαρμογών για κινητά και επιτραπέζιους υπολογιστές που χρησιμοποιούν

τεχνολογίες ιστού (HTML, CSS και JavaScript). Το Ionic Framework επικεντρώνεται στην αλληλεπίδραση του χρήστη με την εφαρμογή (έλεγχοι, αλληλεπιδράσεις, χειρονομίες, κινούμενα σχέδια). Είναι εύκολο να ενσωματώνει άλλες βιβλιοθήκες ή frameworks, όπως η Angular, η οποία μπορεί να χρησιμοποιηθεί αυτόνομα χωρίς frontend framework χρησιμοποιώντας ένα απλό script include.

### 3.7.2 Leaflet

Το Leaflet [28] είναι μια open-source βιβλιοθήκη της Javascript για mobile-friendly διαδραστικούς χάρτες. Λειτουργεί αποτελεσματικά σε όλες τις μεγάλες πλατφόρμες επιτραπέζιων και φορητών υπολογιστών, μπορεί να επεκταθεί με πολλά πρόσθετα, να διαθέτει ένα όμορφο, εύκολο στη χρήση και καλά τεκμηριωμένο API και έναν απλό, ευανάγνωστο πηγαίο κώδικα.

### 3.7.3 OpenStreetMap

Το OpenStreetMap (OSM) [29] είναι μια υπηρεσία χαρτών με ελεύθερη άδεια η οποία αναπτύσσεται από μια κοινότητα εθελοντών που συνεισφέρουν και διατηρούν δεδομένα σχετικά με δρόμους, μονοπάτια, σημεία εστίασης, εμπορικά καταστήματα, στάσεις μέσων μαζικής μεταφοράς, και πολλά περισσότερα, σε όλον τον κόσμο. Η δημιουργία και ανάπτυξη του OSM προκλήθηκε από περιορισμούς στη χρήση ή τη διαθεσιμότητα πληροφοριών χάρτη σε μεγάλο μέρος του κόσμου και την εμφάνιση φθηνών φορητών δορυφορικών συσκευών πλοήγησης. Το OSM θεωρείται σημαντικό δείγμα εθελοντικών γεωγραφικών πληροφοριών.

## 3.8 Τεχνολογίες APIs (Application Programming Interfaces)

### 3.8.1 Endpoints

Πρόκειται για έναν builder APIs που εγκαθίσταται μέσω του PIP διαχειριστή πακέτων στην Python. [30] Το Endpoints μεταφράζει τα requests σε python modules χωρίς παραμετροποίηση.

### 3.8.2 PyMySQL

Η PyMySQL [31] είναι βιβλιοθήκη που παρέχεται για την Python, μέσω της οποίας επιτυγχάνεται επικοινωνία με την Βάση Δεδομένων και δίνεται η δυνατότητα εκτέλεσης SQL queries μέσα από την Python εφαρμογή.

### 3.8.3 PHP Scripts

Αρχεία σε γλώσσα PHP που βρίσκονται στον server του συστήματος τα οποία αναπτύχθηκαν και χρησιμοποιούνται για την επικοινωνία Εφαρμογών – Βάσης Δεδομένων. Μετά από HTTP POST requests που στέλνουμε στις διευθύνσεις που βρίσκονται scripts πραγματοποιούν σύνδεση με την Βάση Δεδομένων, εκτελούν καθορισμένα SQL queries λαμβάνοντας παραμέτρους από το επίπεδο των εφαρμογών και στη συνέχεια επιστρέφουν στο επίπεδο εφαρμογών, σε μορφή JSON, την απόκριση του εκάστοτε query.

## ΚΕΦΑΛΑΙΟ 4

### ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ

#### 4.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται επισκόπηση της αρχιτεκτονικής του συστήματος, της δομής του και παρουσιάζονται η Βάση Δεδομένων και τα μέρη που συνδυάζει. Πιο συγκεκριμένα, το σύστημα αποτελείται από τρία ξεχωριστά μέρη στο προγραμματιστικό σκέλος. Ο πυρήνας του συστήματος είναι το back-end server-side μαζί με μια MySQL βάση δεδομένων. Με αυτό επικοινωνούν μια Web εφαρμογή εισαγωγής παραγγελιών και μια Mobile εφαρμογή που αντιστοιχεί σε κάθε διανομέα, αναπτυγμένες σε Ionic Framework.

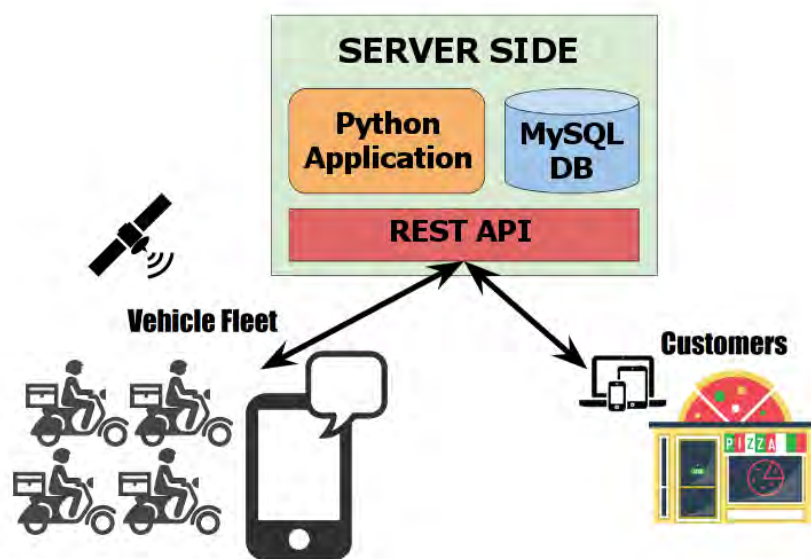
#### 4.2 Αρχιτεκτονική του συστήματος (Επισκόπηση)

Προκειμένου να γίνει αντιληπτό πώς δουλεύει το σύστημα που παρουσιάζουμε είναι αναγκαίο να γίνει μια επισκόπηση της συστήματος ως προς την αρχιτεκτονική του. Αρχικά, αποτελείται από έναν server στον οποίο βρίσκεται η Python εφαρμογή στην οποία αναφερθήκαμε. Εκεί βρίσκεται και η MySQL Βάση Δεδομένων που χρησιμοποιούμε. Για την επικοινωνία των μερών και την λειτουργία του συστήματος έχουν υλοποιηθεί 4 είδη Web Services.

- Επικοινωνία Web Application (Restaurant side) - Python Application
- Επικοινωνία Python Application – MySQL DB
- Επικοινωνία Web Application (Restaurant side) – MySQL DB
- Επικοινωνία Mobile Applications (Delivery-agent side) – MySQL DB

Για το πρώτο έχει γίνει χρήση του Endpoints module της Python, για το δεύτερο έχει γίνει χρήση του PyMySQL module της Python και για τα υπόλοιπα δύο χρησιμοποιούνται PHP scripts. Τα συγκεκριμένα εργαλεία αναφέρθηκαν στην Ενότητα [3.8](#).

Η ακόλουθη εικόνα μπορεί να δώσει μια επισκόπηση της αρχιτεκτονικής του συστήματος.



Εικόνα 7. Αρχιτεκτονική του συστήματος

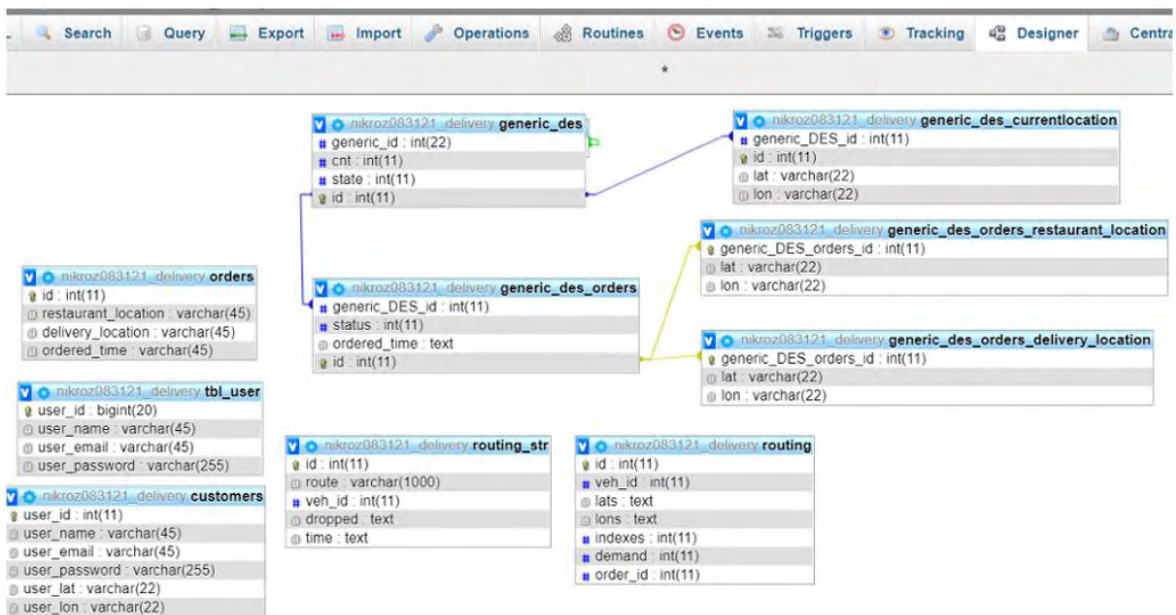
#### 4.3 Βάση Δεδομένων

Για την λειτουργία του συστήματος που αναπτύσσουμε, όπως γίνεται αντιληπτό, απαιτείται η αποθήκευση, η ανάκτηση και η τροποποίηση δεδομένων. Για τον λόγο αυτό δημιουργήσαμε με Σχεσιακή Βάση Δεδομένων σε MySQL. Η Βάση Δεδομένων αυτή φιλοξενείται στον server του συστήματος. Ας εμβαθύνουμε λίγο στα περιεχόμενα της, στους πίνακες και τις συσχετίσεις.

Η Βάση Δεδομένων αποτελείται από τους εξής 10 πίνακες:

- Πίνακας καταχωρημένων πελατών
- Πίνακας καταχωρημένων διανομέων
- Πίνακας τρεχουσών πληροφοριών διανομέων
- Πίνακας τοποθεσίας διανομέων
- Πίνακας πληροφορίας νέων παραγγελιών
- Πίνακας σημείων αφετηρίας παραγγελιών
- Πίνακας σημείων προορισμού παραγγελιών
- Πίνακας ιστορικού παραγγελιών
- Πίνακας τρέχοντος δρομολογίου κάθε διανομέα
- Πίνακας ιστορικού δρομολογίων κάθε διανομέα

Για καλύτερη κατανόηση, το διάγραμμα - σχήμα της Βάσης Δεδομένων απεικονίζεται, με τις στήλες κάθε πίνακα, στην εικόνα που ακολουθεί με επισήμανση των συσχετίσεων μεταξύ στοιχείων (στηλών) των πινάκων.

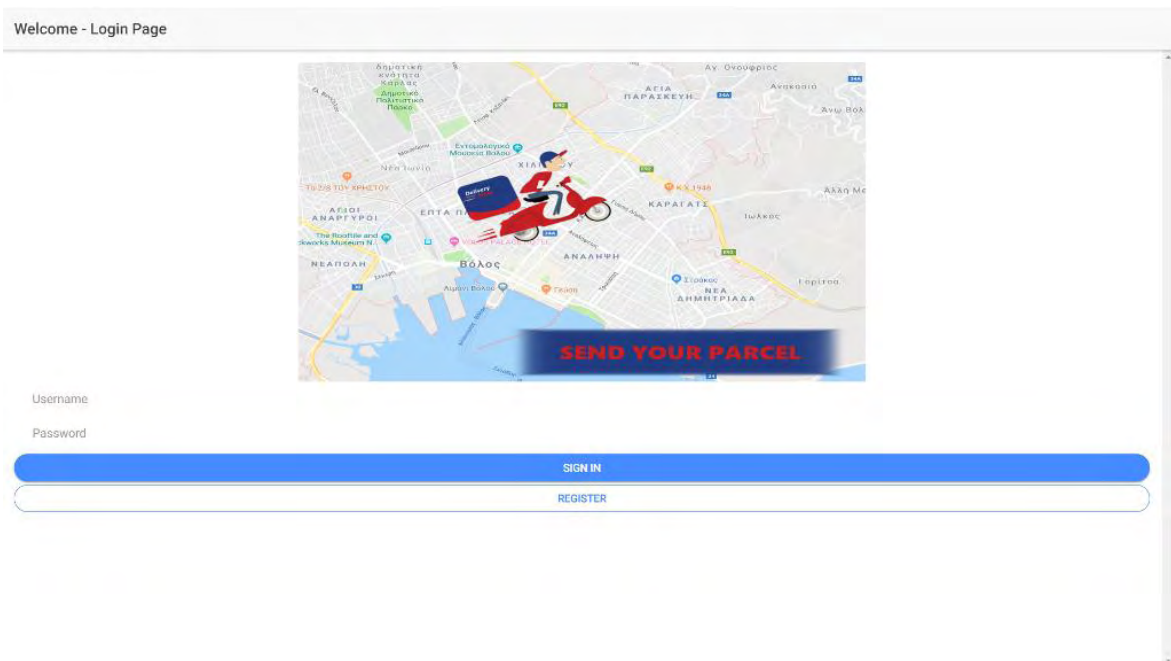


Εικόνα 8. Database schema

#### 4.4 Web App πελάτη

Για την καταχώρηση μιας νέας παραγγελίας στο σύστημα πρέπει ο αιτών εξυπηρέτηση να έχει πρόσβαση στην Web Progressive εφαρμογή μέσω ενός browser με πρόσβαση στο διαδίκτυο. Χρησιμοποιώντας το username και το password που αντιστοιχούν στον

λογαριασμό που έχει δημιουργηθεί σε συνεννόηση με τον διαχειριστή του συστήματος. Η εν λόγω σελίδα φαίνεται όπως στην ακόλουθη εικόνα.



Εικόνα 9. Αρχική σελίδα πελάτη (interface εστιατορίου)

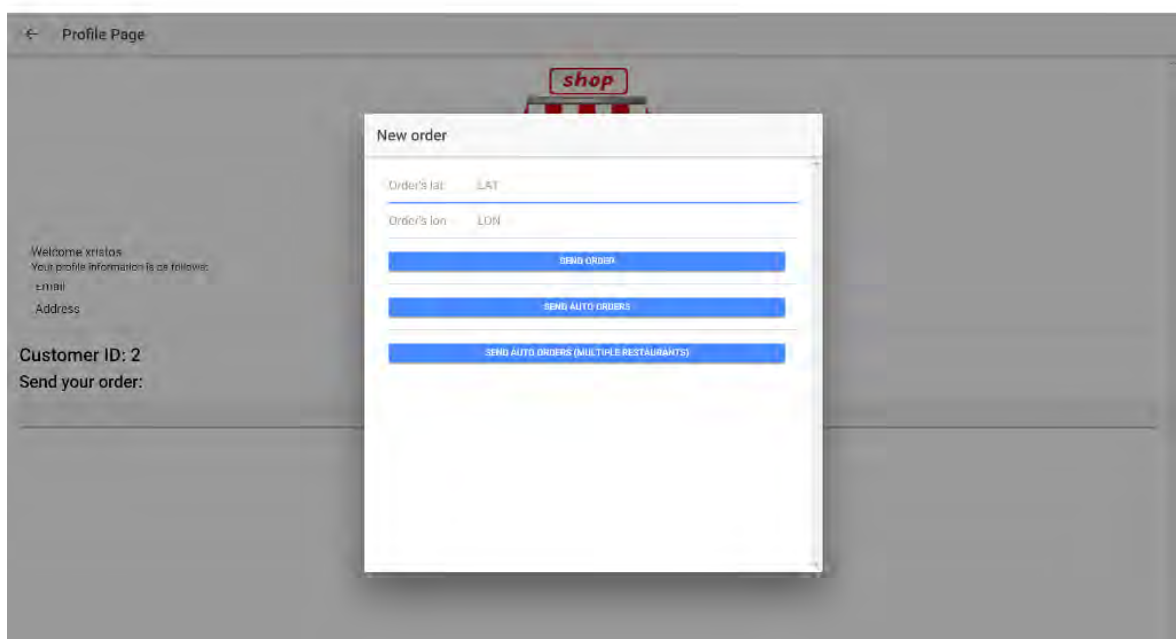
Πραγματοποιώντας Login με τα στοιχεία του ο χρήστης της εφαρμογής αποκτά πρόσβαση στις λειτουργίες της εφαρμογής. Κατόπιν, θα εμφανιστεί η Profile Page που αναγράφει τα στοιχεία του και τις πληροφορίες της διεύθυνσης του όπως φαίνεται στην παρακάτω εικόνα.





Εικόνα 10. Profile page εστιατορίου

Ο χρήστης της εφαρμογής όταν έχει ένα πακέτο που πρέπει να στείλει σε κάποιο προορισμό, πατά το κουμπί “NEW ORDER” με το οποίο εμφανίζεται ένα αναδυόμενο παράθυρο εντός της εφαρμογής με τα πεδία συντεταγμένων του προορισμού όπου επιθυμεί να αποστείλει το πακέτο που αντιστοιχεί στην συγκεκριμένη παραγγελία. Η επόμενη Εικόνα δείχνει αυτό που περιγράφεται.



Εικόνα 11. Σελίδα καταχώρηση νέας παραγγελίας (interface)

Η εφαρμογή αυτή είναι ανεπτυγμένη σε Ionic Framework για χρήση μέσω browser. Η βασική λειτουργικότητα των σελίδων στο συγκεκριμένο Framework προέρχεται από αρχεία γλώσσας Typescript και η διεπαφή χρήστη από HTML και CSS αρχεία σε συνδυασμό με χρήση Angular.

Από την εισαγωγή της παραγγελίας στο σύστημα μέσω της συγκεκριμένης φόρμας λαμβάνει χώρα ένα HTTP POST request στον server στο port που «ακούει» το back-end του συστήματος. Μέσω αυτού η απαιτούμενη πληροφορία στέλνεται σε μορφή json όπως αυτό που ακολουθεί.

```
{"orders":  
[{"restaurant_location": 39.358612, 22.949314,  
  "ordered_time":20:24:59,  
  "id" :783,  
  "delivery_location": 39.358047, 22.954802}  
]}
```

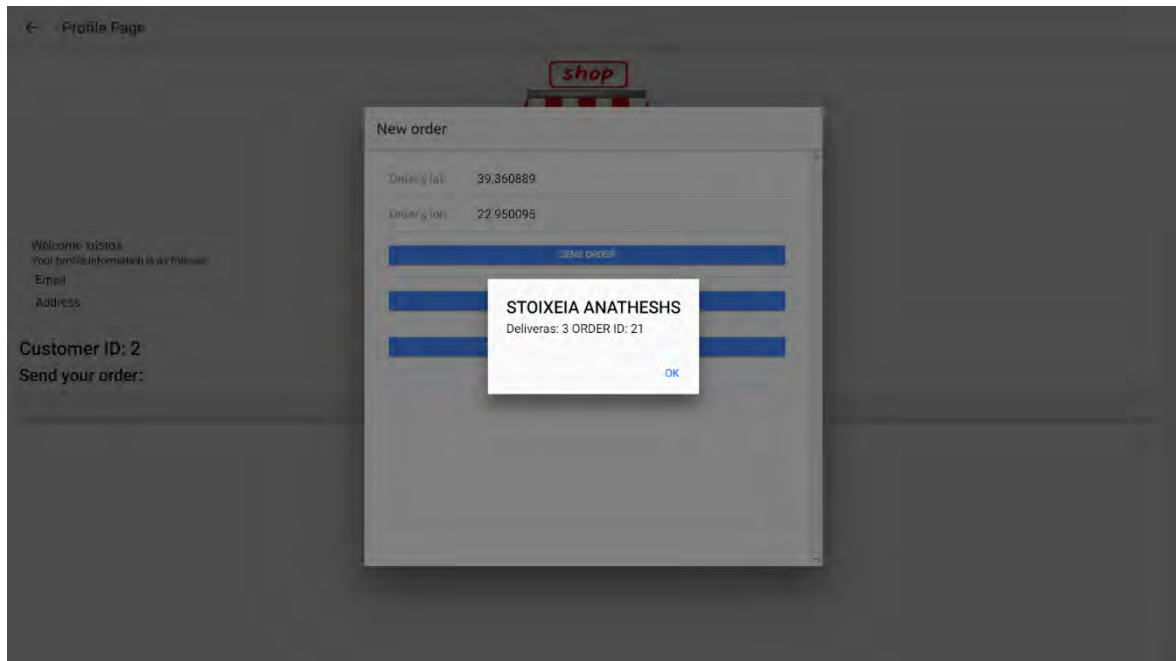
Εικόνα 12. JSON - αποστολή αιτήματος παραγγελίας

Όπως φαίνεται, στέλνονται τέσσερις παράμετροι:

- Τοποθεσία Παραλαβής (καταστήματος)
- Ώρα καταχώρησης παραγγελίας
- ID παραγγελίας
- Τοποθεσία Παράδοσης

Στο σημείο αυτό οι παράμετροι παραλαμβάνονται από την back-end Python εφαρμογή που βρίσκεται στον server και επιστρέφεται απόκριση μορφής JSON όταν η παραγγελία ανατεθεί σε κάποιον διανομέα. Πρόκειται για χρήση του πρώτου κατά σειρά Web Service όπως εμφανίζεται στην Ενότητα [4.2](#). Η απόκριση περιλαμβάνει, στην ορθή λειτουργία,

μήνυμα επιτυχίας και το αναγνωριστικό του συγκεκριμένου διανομέα και εμφανίζεται στον πελάτη όπως στην εικόνα που ακολουθεί.



Εικόνα 13. Μήνυμα ανάθεσης παραγγελίας στον διανομέα με το αναγνωριστικό 3

#### 4.5 Mobile App διανομέα

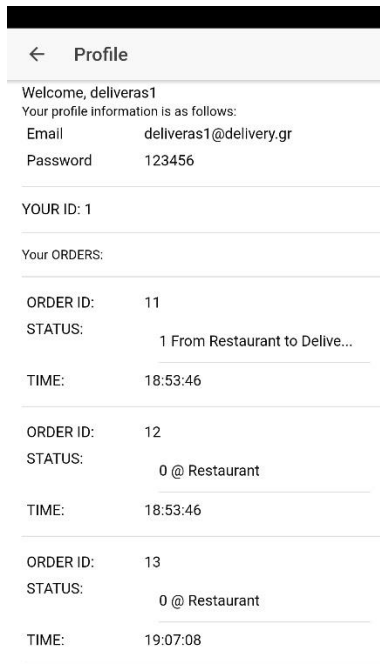
Η εφαρμογή που χρησιμοποιεί ο διανομέας σε smartphone είναι ο διάυλος επικοινωνίας μέσω του οποίου ενημερώνεται για την δρομολόγηση του, ενημερώνει τη Βάση Δεδομένων για τις πράξεις του και κάνει ορατή την τοποθεσία του. Για την ορθή λειτουργία της προϋπόθεση είναι η συνεχής σύνδεση στο Διαδίκτυο και η ενεργοποιημένη λειτουργία του GPS της συσκευής.

Πιο αναλυτικά, κάθε διανομέας του στόλου έχει σε smartphone την mobile εφαρμογή και την χρησιμοποιεί με τον τρόπο θα εξηγήσουμε. Αρχικά, ο χρήστης για να εισαχθεί στο σύστημα και να εξυπηρετεί παραγγελίες πρέπει να συνδεθεί στην εφαρμογή με τα στοιχεία username και password που έχουν τεθεί κατά την καταχώρηση του στο σύστημα υπό την εποπτεία του διαχειριστή.

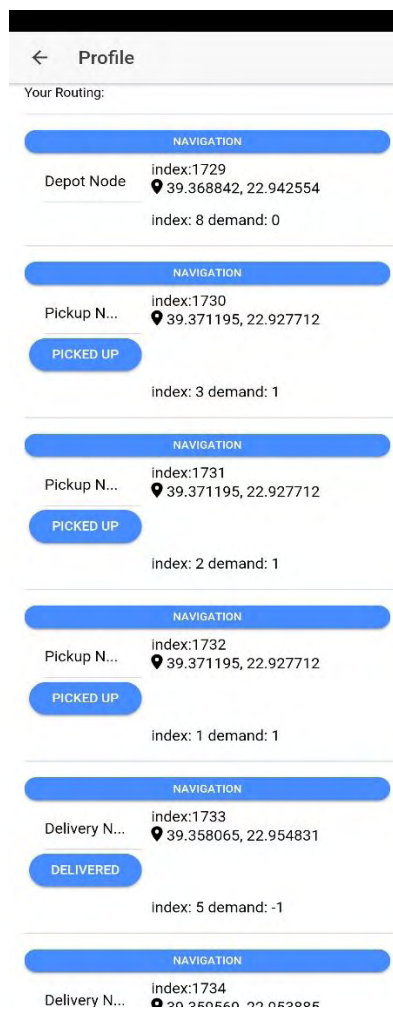


Εικόνα 14. Αρχική σελίδα mobile app διανομέα

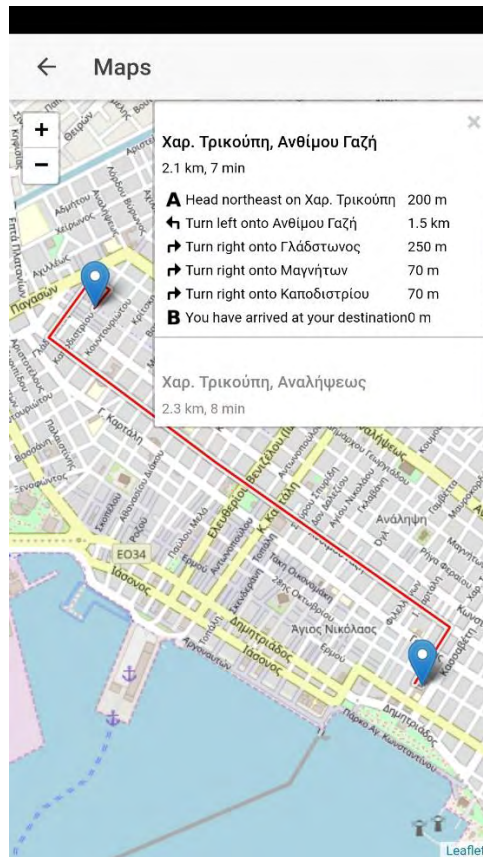
Με την εισαγωγή στοιχείων και την σύνδεση ο χρήστης - διανομέας οδηγείται στην κύρια σελίδα της εφαρμογής, όπου του εμφανίζονται το ID αναγνωριστικό του, όπως ορίζεται από το σύστημα, και η λίστα των ανατεθειμένων σε αυτόν παραγγελιών. Ταυτόχρονα, του εμφανίζεται το δρομολόγιο εξυπηρέτησης των παραγγελιών αυτών όπως έχει προκύψει από την back-end επεξεργασία. Κάθε κόμβος – τοποθεσία του δρομολογίου απεικονίζεται μαζί με πληροφορία ως προς το είδος του κόμβου (κόμβος παραλαβής, κόμβος παράδοσης, σταθμός οχημάτων). Κάτω από κάθε κόμβο υπάρχει ένα κουμπί, στην περίπτωση κόμβου παραλαβής εμφανίζεται με την ταμπέλα “PICKED” και το πάτημα του καθιστά την παραγγελία παραληφθείσα, στην περίπτωση κόμβου παράδοσης εμφανίζεται με την ταμπέλα “DELIVERED” και το πάτημα του καθιστά την παραγγελία παραδοθείσα. Εκτός αυτών, σε κάθε κόμβο υπάρχει το κουμπί “NAVIGATION” που πλοηγεί τον χρήστη από την τρέχουσα τοποθεσία του στον αντίστοιχο κόμβο μέσω της γρηγορότερης διαδρομής.



Εικόνα 15. Κύρια σελίδα mobile app διανομέα



Εικόνα 16. . Κύρια σελίδα mobile app διανομέα (routing)



Εικόνα 17. Πλοήγηση διανομέα σε κόμβο του δρομολογίου

Η εφαρμογή αυτή είναι ανεπτυγμένη σε Ionic Framework ως υβριδική με σκοπό την συμβατότητα της με smartphones που τρέχουν Android, iOS ή Windows λειτουργικό σύστημα. Στην συγκεκριμένη απεικόνιση έχει χτιστεί για χρήση σε Android συσκευές. Όπως και στην πρώτη εφαρμογή έτσι και στην παρούσα, η βασική λειτουργικότητα των σελίδων προέρχεται από αρχεία γλώσσας Typescript και η διεπαφή χρήστη από HTML και CSS αρχεία σε συνδυασμό με χρήση Angular.

Αρχικά, όσον αφορά σύνδεση του χρήστη, ο ίδιος εισάγει τα στοιχεία του στα κατάλληλα πεδία εισαγωγής και πατά το κουμπί SIGN IN. Πατώντας το κουμπί λαμβάνει χώρα ένα http post request σε ένα PHP script που βρίσκεται στον server. Το script αυτό επικοινωνεί με τη βάση εκτελώντας SQL ερώτημα για να ελεγχθεί αν υπάρχει χρήστης με αυτά τα στοιχεία στη βάση και λαμβάνει απάντηση ως προς αυτό την οποία επιστρέφει σε μορφή JSON στην εφαρμογή ως απόκριση στο post request. Η εφαρμογή ανάλογα την απόκριση που θα λάβει οδηγεί τον χρήστη στην επόμενη σελίδα, ή του εμφανίζει μήνυμα σφάλματος. Σε περίπτωση που είναι γίνει η επικοινωνία αυτή και είναι σωστά τα στοιχεία,

ο χρήστης οδηγείται στην κύρια σελίδα. Κατά την φόρτωση της σελίδας αυτής, με τον ίδιο τρόπο, εκτελείται post request σε PHP script που εκτελεί SQL ερώτημα και επιστρέφει πληροφορίες, όπως το email και το ID αναγνωριστικό του χρήστη-διανομέα. Μόλις λάβει η εφαρμογή το αναγνωριστικό εκτελείται νέο http post request σε PHP script, το οποίο εκτελεί SQL ερώτημα για τις ανατεθειμένες παραγγελίες στον διανομέα με αναγνωριστικό ID ίδιο με το ID που έλαβε προηγουμένως. Εν συνεχεία, εκτελείται νέο http post request σε άλλο PHP script, το οποίο εκτελεί ερώτημα για να λάβει το δρομολόγιο του χρήστη με το συγκεκριμένο ID. Όλα αυτά λαμβάνουν χώρα κατά την φόρτωση της σελίδας και μόλις ολοκληρωθούν εμφανίζεται η σελίδα με τις παραπάνω πληροφορίες στον χρήστη.

Όταν ο χρήστης επιθυμεί να λάβει πληροφορίες πλοήγησης για κάποιον κόμβο του δρομολογίου του πατά το κουμπί “NAVIGATION”. Από εκεί, οδηγείται σε μια σελίδα με χάρτη και πληροφορίες πλοήγησης από το σημείο που βρίσκεται (όπως λαμβάνεται από το GPS), με χρήση των βιβλιοθηκών Leafletjs, Leaflet Routing Machine και το εργαλείο παροχής υπηρεσιών χαρτογράφησης OpenStreetMap.

Την στιγμή που ο διανομέας παραλάβει μια παραγγελία πατώντας το αντίστοιχο κουμπί “PICKED” εκτελεί νέο http post request σε PHP script, το οποίο εκτελεί SQL ερώτημα ανανέωσης της κατάστασης της συγκεκριμένης παραγγελίας στην βάση δεδομένων καθιστώντας της παραληφθείσα. Με τον ίδιο τρόπο πατώντας το κουμπί “DELIVERD” την καθιστά παραδοθείσα.

#### **4.6 Python εφαρμογή (server-side)**

Το back-end μέρος του συστήματος είναι ο πυρήνας της προσέγγισης που προτείνουμε ως λύση στο ζήτημα της διαχείρισης αιτήσεων εξυπηρέτησης παραγγελιών. Το κομμάτι αυτό αναπτύχθηκε σε γλώσσα προγραμματισμού Python. Μέσω αυτού γίνεται η επεξεργασία των αιτημάτων πελατών και διαμορφώνεται η δρομολόγηση των διανομών. Μέσα σε αυτό υλοποιούμε Application Programming Interfaces (APIs) για την λήψη των αιτήσεων που αντιστοιχούν σε νέες παραγγελίες προς εξυπηρέτηση και λαμβάνει χώρα επικοινωνία

με την MySQL Βάση Δεδομένων. Πολύ σημαντική είναι η χρήση μέρους των εργαλείων OR-TOOLS μέσω βιβλιοθήκης της Python για επίλυση του προβλήματος δρομολόγησης.

Αρχικά, η Python εφαρμογή λαμβάνει ένα αίτημα μέσω του API που δημιουργήθηκε μέσω Endpoints. Από αυτό το σημείο και έπειτα, γίνεται επεξεργασία του αιτήματος βάσει της αλγοριθμικής αλληλουχίας που αναφέρθηκε στην Ενότητα 2.8.

Πιο αναλυτικά, το σύστημα έχει λάβει τις συντεταγμένες του καταστήματος που επιδιώκει να στείλει -ως αφετηριακό σημείο- ένα πακέτο, την ώρα κατά την οποία καταχωρείται το αίτημα, καθώς και τις συντεταγμένες της τοποθεσίας προορισμού αυτού του πακέτου.

### 3.6.1 Ο χειρισμός ενός νέου αιτήματος

Η άφιξη ενός νέου αιτήματος στην Python εφαρμογή ακολουθείται από την άντληση των διαθέσιμων διανομών από την Βάση Δεδομένων. Ένας διανομέας είναι διαθέσιμος όταν βρίσκεται ενεργός, εν ώρα εργασίας, και δεν έχει ξεπεράσει το όριο των ταυτόχρονων παραγγελιών που μπορεί να εξυπηρετεί, το όριο αυτό ρυθμίζεται βάσει των αναγκών, του είδους των μεταφερόμενων πακέτων, των χρονικών περιθωρίων και του μεγέθους των αποστάσεων από τον διαχειριστή του συστήματος. Μόλις το σύστημα αντλήσει του διαθέσιμους διανομείς, αντλεί και όλους τους κόμβους οι οποίοι έχουν ανατεθεί σε αυτούς μέχρι εκείνη την χρονική στιγμή. Πάραυτα, πραγματοποιούνται συγκρίσεις μεταξύ των δύο κόμβων της νέας παραγγελίας και όλων των μέχρι τότε ανατεθειμένων σε δρομολόγιο κόμβων, όσον αφορά όλους τους διαθέσιμους διανομείς. Η νέα παραγγελία ανατίθενται στον διανομέα, στον οποίου τους ανατεθειμένους κόμβους βρίσκεται κοντύτερα -βάσει της Ευκλείδειας Απόστασης- κάποιος εκ των δύο νέων κόμβων. Σε περίπτωση που κανείς διανομέας δεν έχει ανατεθειμένους κόμβους σε αναμονή να διέλθει, τότε λαμβάνεται υπόψιν η θέση του διανομέα ως κριτήριο ανάθεσης και επιλέγεται εκείνος που βρίσκεται πιο κοντά στην αφετηρία της νέας παραγγελίας.

Αν κατά την εκτέλεση της επεξεργασίας, ο διανομέας αγγίζει το ανώτατο όριο παραγγελιών που έχει οριστεί από τον διαχειριστή του συστήματος, τότε η κατάσταση του σημειώνεται ως «μη διαθέσιμος». Στο τέλος της διαδικασίας το νέο δρομολόγιο του διανομέα προστίθεται στη Βάση Δεδομένων. Ο πελάτης - αποστολέας του αιτήματος



ενημερώνεται για την ανάθεση της παραγγελίας σε διανομέα, καθώς και την ταυτότητα αυτού, με σχετική απόκριση στην εφαρμογή του. Παράλληλα, ενημερώνεται και η mobile εφαρμογή του διανομέα.

#### **4.7 Παραδοχές**

Στο σημείο αυτό είναι απαραίτητο να αναφερθούν κάποιες παραδοχές που κάνουμε και τις χρησιμοποιούμε στην επίλυση του προβλήματος. Οι παραδοχές αυτές έχουν να κάνουν με την περίπτωση χρήσης του συστήματος μας στον κλάδο της διανομής γρήγορου φαγητού σε μια πόλη με έκταση σαν του Βόλου. Αρχικά, σαν εμπειρική παραδοχή λαμβάνουμε την μέση ταχύτητα ενός διανομέα ίση με 30χλμ/ώρα. Παράλληλα, θεωρούμε πως ο μέσος χρόνος εξυπηρέτησης του διανομέα εκτός οχήματος ισούται με 180 δευτερόλεπτα. Πρόκειται για τον χρόνο που χρειάζεται για να παραλάβει ένα πακέτο από τον πελάτη έως ότου ξεκινήσει την εκτέλεση της διαδρομής ή γενικότερα τον χρόνο από την άφιξη σε έναν κόμβο έως ότου αναχωρήσει από αυτόν.

#### **4.8 Προσομοίωση Λειτουργίας Συστήματος**

Μετά την ανάπτυξη του συστήματος ήρθαμε αντιμέτωποι με το ζήτημα της δοκιμής του, η εκτέλεση της οποίας θα μας οδηγούσε σε εξαγωγή συμπερασμάτων. Για τον σκοπό αυτό εκτελέσαμε σενάρια προσομοίωσης. Δημιουργήσαμε συναρτήσεις, οι οποίες αφενός έστελναν περιοδικά εικονικές παραγγελίες στο σύστημα (από την πλευρά του πελάτη π.χ εστιατορίου), αφετέρου παρουσίαζαν εικονική εξυπηρέτηση από πλευράς διανομέων ως προς την παράδοση και παραλαβή των εικονικών παραγγελιών που καταχωρούνταν.

Η δοκιμή αυτή, λοιπόν, πραγματοποιήθηκε βάσει των παραδοχών που αναφέρθηκαν παραπάνω υιοθετώντας το σενάριο της χρήσης του συστήματος μας στην διανομή γρήγορου φαγητού μεταξύ εστιατορίων και πελατών τους στην πόλη του Βόλου. Για την δοκιμή χρησιμοποιήθηκαν έξι εικονικά εστιατόρια σε συγκεκριμένες τοποθεσίες μέσα στην πόλη, καθώς και καταχωρήθηκαν οχτώ εικονικοί διανομείς στο σύστημα. Επίσης, ως προορισμοί των παραγγελιών χρησιμοποιήθηκαν τυχαία παραγόμενες τοποθεσίες σε ακτίνα 2 χιλιομέτρων από ένα ορισθέν ως κέντρο. Κατά την προσομοίωση

χρησιμοποιώντας ποινές δόθηκε η δυνατότητα στο σύστημα κατά την διαδικασία δρομολόγησης ενός διανομέα να απορρίπτει την εξυπηρέτηση κάποιας/ων παραγγελίας/ων αν δεν ήταν δυνατόν να εξυπηρετηθούν κατά τους χρονικούς περιορισμούς που ετέθησαν στις εκτελέσεις. Με τον τρόπο αυτό καταλήξαμε σε συμπεράσματα ως προς τα όρια που πρέπει να χρησιμοποιεί το σύστημα στο σενάριο αυτό. Πιο συγκεκριμένα, το άνω όριο των 5 ταυτόχρονων παραγγελιών για κάθε διανομέα εξασφαλίζει σίγουρη επιτυχία της λειτουργίας. Ως προς τους χρονικούς περιορισμούς βάσει των δυνατοτήτων εξυπηρέτησης αλλά και των αναγκών των εστιατορίων του Βόλου καταλήξαμε στην χρήση 20 λεπτών της ώρας χρονικού παραθύρου για την παραλαβή ενός πακέτου και 40 λεπτών της ώρας για την παράδοση του, μετρώντας από την ώρα καταχώρησης του αιτήματος εξυπηρέτησης.

Αυτά τα συμπεράσματα έχουν να κάνουν αποκλειστικά με το σενάριο που επιλέξαμε.

## ΚΕΦΑΛΑΙΟ 5

### ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΣΘΗΚΕΣ

Η υλοποίηση ενός ολοκληρωμένου πολυπρακτορικού συστήματος με τις λειτουργίες που αναλύσαμε απαιτεί πλήρη κατανόηση του προβλήματος δρομολόγησης οχημάτων και όλων των επεκτάσεων του. Η ανάπτυξη του χρειάζεται ευελιξία και ορθή χρήση των τεχνολογιών για να μην υπάρχουν καθυστερήσεις ή σημεία που υποκρύπτονται λάθη.

Θεωρούμε πως η υλοποίηση μας είναι ολοκληρωμένη ως προς τους αρχικούς στόχους που θέσαμε για την λειτουργία του συστήματος και τη χρηστικότητα του στον τομέα των city logistics με έμφαση στην εστίαση, και ιδιαίτερα στις ανάγκες της αγοράς του γρήγορου φαγητού. Ωστόσο, στην υπάρχουσα υλοποίηση θα μπορούσε κανείς να ανακαλύψει προοπτικές, οι οποίες με κατάλληλες προσθήκες θα ήταν ικανές να βελτιώσουν περαιτέρω την λειτουργία οργανισμών ή επιχειρήσεων της αγοράς.

Πιο συγκεκριμένα, βελτιώσεις σε επίπεδο εφαρμογών όπως:

- Υποστήριξη πολλών χρηστών με ρόλους σε διαφορετικά επίπεδα πρόσβασης
- Ανάπτυξη εφαρμογής συνολικής διαχείρισης της υπηρεσίας του συστήματος με λειτουργία παρακολούθηση του στόλου οχημάτων
- Λειτουργίας υποστήριξης ταχυδρομικών διευθύνσεων με λειτουργία αυτόματης συμπλήρωσης για τον πελάτη κατά την αποστολή παραγγελιών

Σε αλγοριθμικό επίπεδο, μια χρήσιμη βελτίωση θα ήταν η προσθήκη κριτηρίου για το μέγεθος φορτίου που ανατίθεται στους διανομείς. Ειδικότερα, προτείνεται και επισημαίνεται για μελλοντική ανάπτυξη, κατά την διαδικασία ανάθεσης, να ελέγχεται το υφιστάμενο φορτίο του κάθε διανομέα εκείνη τη στιγμή και, εντούτοις, να συνδυάζεται η κοντινότερη απόσταση με το υφιστάμενο φορτίο με σκοπό να ανατίθεται η παραγγελία στον κοντινότερο διανομέα με το μικρότερο φορτίο. Με το τρόπο αυτό επιτυγχάνεται ο στόχος μια στοιχειώδους εξισορρόπησης φορτίου μεταξύ των διανομέων. Μια ακόμη

ωφέλιμη προσθήκη για οργανισμούς, επιχειρήσεις και εργαζομένους που κάνουν χρήση του συστήματος είναι ο υπολογισμός, βάσει ρυθμού αιτημάτων, του πλήθους του απαιτούμενου αριθμού διανομέων για την επιτυχή εξυπηρέτηση όλου του όγκου των αιτημάτων. Επιπλέον, μεγαλύτερη προσαρμοστικότητα στη λειτουργία του συστήματος θα προσέδιδε ο υπολογισμός μέσης ταχύτητας για κάθε διανομέα ώστε κατά τη δρομολόγηση να εξατομικεύεται η παράμετρος της ταχύτητας από την οποία εξαρτάται το δρομολόγιο που προκύπτει.

## BIBΛΙΟΓΡΑΦΙΑ

- [1] «Wikipedia» [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Travelling\\_salesman\\_problem](https://en.wikipedia.org/wiki/Travelling_salesman_problem).
- [2] [Ηλεκτρονικό]. Available: <http://www.math.uwaterloo.ca/tsp/world/>.
- [3] G. B. Dantzig και J. H. Ramser, «The Truck Dispatching Problem» *Management Science*, 10 1959.
- [4] [Ηλεκτρονικό]. Available: <http://neo.lcc.uma.es/vrp/vehicle-routing-problem/>.
- [5] G. a. W. J. Clarke, «Scheduling of Vehicle Routing Problem from a Central Depot to a Number of Delivery Points» *Operations Research*, pp. 568-581, 1964.
- [6] A. N. Λότης, «On the online dial a ride problem» Βόλος, 2010.
- [7] N. a. C. N. Wilson, Computer control of the Rochester Dial-A-Ride System, Cambridge: Massachusetts Institute of Technology, Center for Transportation Studies, 1977.
- [8] B. B. a. B. Nag, «Dynamic Vehicle Scheduling: An Expert Systems Approach» *International Journal of Physical Distribution and Logistics Management*, pp. 10-18, 1991.
- [9] H. N. Psaraftis, «A dynamic-programming solution to the single vehicle many-to-many immediate request dial-a-ride problem» *Transportation Science*, pp. 130-154, 5 1980.
- [10] G. v. R. Dimitris J. Bertsimas, «Stochastic and Dynamic Vehicle Routing in the Euclidean Plane with Multiple Capacitated Vehicles» *Operations Research*, pp. 60-76, Jan. - Feb. 1993.
- [11] S. G. E. S. B. Fleischmann, « Dynamic Vehicle Routing on Online Traffic Information» *Tans Science*, pp. 420-433, 2004.
- [12] H. S. M. P. J. Amelia C. Regan, «Improving efficiency of commercial vehicle operations using real-time information: potential uses and assignment strategies» *Transportation Research Record*, pp. 188-198, 1 July 1995.
- [13] S. G. M. P. J. Ichoua, «Diversion issues in realtime vehicle dispatching» *Transportation Science*, p. 426-438, 2000.
- [14] V. A. A. L. R. M. Branchini, «Adaptive granular local search heuristic for a dynamic vehicle routing problem» *Computers & Operations Research*, pp. 2955-2968.
- [15] I. A. a. A. B. Goran Martinovic, «Single-Commodity Vehicle Routing Problem with Pickup and Delivery Service» *Mathematical Problems in Engineering*, pp. 1024-123X, January 2008.
- [16] M. M. S. a. J. DESROSIERS, «Time Window Constrained Routing and Scheduling Problems» *Transportation Science*, pp. 1-13, February 1988.
- [17] Y. D. M. M. F. S. Jacques Desrosiers, «Time Constrained Routing and Scheduling» σε *Handbooks in Operations Research and Management Science, Vol 8, Network Routing*, Elsevier Science B.V., 1995, pp. 35-139.
- [18] M. W. P. S. a. M. SOL, «The General Pickup and Delivery Problem» *Transportation Science*, pp. 17-29, February 1995.

- [19] «Wikipedia: Haversine formula» [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Haversine\\_formula](https://en.wikipedia.org/wiki/Haversine_formula).
- [20] «Google OR-Tools» Google, [Ηλεκτρονικό]. Available: <https://developers.google.com/optimization/introduction/overview>.
- [21] «Google OR-Tools / Routing Options» [Ηλεκτρονικό]. Available: [https://developers.google.com/optimization/routing/routing\\_options](https://developers.google.com/optimization/routing/routing_options).
- [22] «OptaPlanner» [Ηλεκτρονικό]. Available: <https://www.optaplanner.org/>.
- [23] «Project-osrm.org» [Ηλεκτρονικό]. Available: <http://project-osrm.org/>.
- [24] «Microsoft Fleet Management» [Ηλεκτρονικό]. Available: <https://www.microsoft.com/en-us/maps/fleet-management>.
- [25] «graphhopper.com» [Ηλεκτρονικό]. Available: <https://www.graphhopper.com/>.
- [26] «Jsprit» [Ηλεκτρονικό]. Available: <https://jsprit.github.io/>.
- [27] «<https://ionicframework.com/>» Ionic, [Ηλεκτρονικό]. Available: <https://ionicframework.com/>.
- [28] «leafletjs.com» [Ηλεκτρονικό]. Available: <https://leafletjs.com/>.
- [29] «openstreetmap.org» OpenStreetMap, [Ηλεκτρονικό]. Available: <https://www.openstreetmap.org>.
- [30] «Github: Endpoints - API builder» [Ηλεκτρονικό]. Available: <https://github.com/Jaymon/endpoints>.
- [31] «PyMySQL library documentation» [Ηλεκτρονικό]. Available: <https://pymysql.readthedocs.io/en/latest/>.
- [32] P. Toth και D. Vigo, The Vehicle Routing Problem. Monographs on Discrete Mathematics and Applications., Philadelphia: Society for Industrial and Applied Mathematics, 2002.
- [33] «jungleworks.com» Jungleworks, [Ηλεκτρονικό]. Available: <https://jungleworks.com/tookan/>.
- [34] «routific.com» Routific, [Ηλεκτρονικό]. Available: <https://routific.com/>.
- [35] «opendoorlogistics.com» OPENDOORLOGISTICS, [Ηλεκτρονικό]. Available: <https://www.opendoorlogistics.com/software/odl-live/>.
- [36] M. T. Goodrich και R. Tamassia, «18.1.2 The Christofides Approximation Algorithm» σε *Algorithm Design and Applications*, Wiley, 2015, p. 513–514.
- [37] N. Christofides, «Worst-case analysis of a new heuristic for the travelling salesman problem» Graduate School of Industrial Administration, CMU, 1976.
- [38] «pypi.org» [Ηλεκτρονικό]. Available: <https://pypi.org/project/endpoints/>.

## ΠΑΡΑΡΤΗΜΑ Α

### ΕΙΚΟΝΕΣ

Εικόνα 1. Λύση ενός TSP: η γραμμή δείχνει το συντομότερο μονοπάτι που συνδέει κάθε κόκκινο σημείο-κόμβο [1].....	3
Εικόνα 2. Λύση VRP με μία αποθήκη [4].....	5
Εικόνα 3. VRPPD ενός οχήματος [15].....	9
Εικόνα 4. Google OR-Tools όρια του solver [21].....	12
Εικόνα 5. Google OR-Tools επιλογές στρατηγικής πρώτης λύσης [21].....	13
Εικόνα 6. Google OR-Tools επιλογές στρατηγικής τοπικής αναζήτησης [21].....	14
Εικόνα 7. Αρχιτεκτονική του συστήματος.....	21
Εικόνα 8. Database schema.....	22
Εικόνα 9. Αρχική σελίδα πελάτη (interface εστιατορίου).....	23
Εικόνα 10. Profile page εστιατορίου.....	24
Εικόνα 11. Σελίδα καταχώρηση νέας παραγγελίας (interface).....	24
Εικόνα 12. JSON - αποστολή αιτήματος παραγγελίας.....	25
Εικόνα 13. Μήνυμα ανάθεσης παραγγελίας στον διανομέα με το αναγνωριστικό 3.....	26
Εικόνα 14. Αρχική σελίδα mobile app διανομέα.....	27
Εικόνα 15. Κύρια σελίδα mobile app διανομέα.....	28
Εικόνα 16. . Κύρια σελίδα mobile app διανομέα (routing).....	28
Εικόνα 17. Πλοήγηση διανομέα σε κόμβο του δρομολογίου.....	29