ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΑΓΡΟΝΟΜΩΝ ΚΑΙ ΤΟΠΟΓΡΑΦΩΝ ΜΗΧΑΝΙΚΩΝ

ΤΟΜΕΑΣ ΕΡΓΩΝ ΥΠΟΔΟΜΗΣ ΚΑΙ ΑΓΡΟΤΙΚΗΣ ΑΝΑΠΤΥΞΗΣ

ΕΡΓΑΣΤΗΡΙΟ ΣΥΓΚΟΙΝΩΝΙΑΚΗΣ ΤΕΧΝΙΚΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΒΕΛΤΙΣΤΟΣ ΣΧΕΔΙΑΣΜΟΣ ΔΙΚΤΥΟΥ ΔΙΑΔΡΟΜΩΝ ΓΙΑ ΑΥΤΟΝΟΜΑ ΟΧΗΜΑΤΑ

ΘΕΟΔΩΡΟΣ ΑΛΠΟΣ

Επιβλέπων:

Κωνσταντίνος Κεπαπτσόγλου, Επίκουρος Καθηγητής ΕΜΠ

Αθήνα, Οκτώβριος 2018

ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία έχεις ως αντικείμενο τον βέλτιστο σχεδιασμό δικτύου διαδρομών για αυτόνομα οχήματα ανταποκρινόμενο στη ζήτηση ενός συνόλου πελατών και με στόχο την αποτελεσματική εξυπηρέτησή τους. Το πρότυπο που δημιουργείται λαμβάνει τη μορφή ενός προβλήματος δρομολόγησης οχημάτων από έναν κόμβο με ενδιάμεσες στάσεις και επιστροφή στον αρχικό κόμβο με περιορισμούς στη χωρητικότητα και το μέγιστος εύρος των οχημάτων. Για την επίλυση του προβλήματος εφαρμόζεται ο αλγόριθμος βελτιστοποίησης με βάση τη λειτουργία των αποικιών των μυρμηγκιών μέσω του οποίου παράγεται ένα σύνολο διαδρομών που ικανοποιούν τη ζήτηση όλων των πελατών. Τα αποτελέσματα δείχνουν πως ο αλγόριθμος παράγει λογικά αποτελέσματα σε σχετικά μικρούς υπολογιστικούς χρόνους λαμβάνοντας υπόψη το πλήθος περιορισμών που χαρακτηρίζουν το πρόβλημα.

ABSTRACT

This study deals with the design of an optimal route network for autonomous vehicles responding to the demand of a set of customers. The problem is modeled as a one-to-many-to-one vehicle routing problem with vehicle capacity and range constraints. The Ant Colony Optimization Algorithm is applied to the problem in order to construct a set of routes that satisfies the demand of each customer. Results show that the algorithm is able to produce results in relatively short computational times considering the amount of constraints that describe the problem.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον Επίκουρο Καθηγητή της Σχολής Αγρονόμων Τοπογράφων Μηχανικών ΕΜΠ Κωνσταντίνο Κεπαπτσόγλου για τη δυνατότητα που μου προσέφερε να καταπιαστώ με ένα πραγματικά ενδιαφέρον θέμα και για την πολύτιμη καθοδήγησή του στο διάστημα που συνεργαστήκαμε.

Επίσης, θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στη διπλωματούχο Πολιτικό Μηχανικό ΕΜΠ Χριστίνα Ηλιοπούλου για την καθοριστική βοήθεια που μου προσέφερε σε όλη τη διάρκεια εκπόνησης της εργασίας καθώς και για τον πολύτιμο χρόνο που μου αφιέρωσε.

Τέλος, ευχαριστώ την οικογένεια μου και τους φίλους μου για την υποστήριξη που μου προσέφεραν σε όλα τα χρόνια της φοιτητικής μου πορείας.

<u>ΠΕΡΙΕΧΟΜΕΝΑ</u>

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

1.1 ΕΙΣΑΓΩΓΗ

Στις αστικές μετακινήσεις στην Ελλάδα αλλά και γενικότερα σε παγκόσμια κλίμακα έχουν κυριαρχήσει δύο τρόποι μετακινήσεων, τα ιδιωτικά αυτοκίνητα και η αστική συγκοινωνία. Τα δύο αυτά είδη δεν έχουν φέρει τα επιθυμητά αποτελέσματα και για αυτό κατά καιρούς έχουν γίνει προσπάθειες για νέα είδη μετακινήσεων, άλλες επιτυχημένες και άλλες όχι. Οι νέες προσπάθειες στρέφονται σε πιο ευέλικτους και φιλικότερους προς το περιβάλλον τρόπους. Μία από αυτές είναι τα αυτόνομα οχήματα τα οποία μπορούν να χρησιμοποιηθούν και ως ταξί. Τα οχήματα αυτά βασίζονται στην τεχνητή νοημοσύνη και σε ένα σύνολο αισθητήρων που τους επιτρέπουν να αντιλαμβάνονται το περιβάλλον γύρω τους και να αντιδρούν ανάλογα. Η μαζική έλευση αυτών των οχημάτων στην αγορά αναμένεται να αλλάξει τον τρόπο που μετακινούμαστε εντός και εκτός πόλεων.

Σχετικά με το είδος του καυσίμου που χρησιμοποιούν αυτά τα οχήματα, η γενική κατεύθυνση από τις κυβερνήσεις και τις εταιρίες κατασκευής οχημάτων είναι τα εναλλακτικά καύσιμα. Οχήματα εναλλακτικών καυσίμων είναι τα οχήματα που για την τροφοδότηση του κινητήρα δεν χρησιμοποιούν αποκλειστικά πετρέλαιο, όπως για παράδειγμα τα ηλεκτρικά και τα υβριδικά ηλεκτρικά. Επιπλέον, λαμβάνοντας υπόψη τα υψηλά επίπεδα ατμοσφαιρικής ρύπανσης στα μεγάλα αστικά κέντρα της Ευρώπης και όχι μόνο, σε συνδυασμό με τους μηδενικούς ρύπους που παράγονται από τα ηλεκτρικά οχήματα γίνονται εύκολα αντιληπτά τα πολλαπλά οφέλη που θα προέκυπταν αν ένα τέτοιο είδος οχημάτων κυριαρχούσε στις μετακινήσεις εντός των μεγάλων αστικών κέντρων. Αξίζει να αναφερθεί ότι στις Ηνωμένες Πολιτείες της Αμερικής, το 30% των εκπομπών αερίων του θερμοκηπίου καθώς και το 70% της χρήσης του πετρελαίου οφείλονται στα ιδιωτικά αυτοκίνητα (Bureau of Transportation Statistics, 2014). Τα τελευταία χρόνια έχουν γίνει πολλές προσπάθειες στην κατεύθυνση "καθαρότερων" καυσίμων και στην δημιουργία αποδοτικότερων κινητήρων ώστε να μειωθούν οι εκπομπές των επιβλαβών στο περιβάλλον αερίων. Οι δημόσιες συγκοινωνίες αν και έχουν την δυνατότητα να βελτιώσουν αυτά τα προβλήματα, δεν έχουν την απαιτούμενη ποιότητα υπηρεσιών και ευελιξία (Sinha, 2003). Επίσης, επειδή τα ιδιωτικά οχήματα παραμένουν σε ακινησία για μεγάλες περιόδους, μελέτες έχουν δείξει ότι ένα κοινόχρηστο αυτόνομο όχημα θα μπορούσε να αντικαταστήσει από 3 ως 11 ιδιωτικά οχήματα (Spieser et al. ,2014 ; Fagnant και Kockelman,2014). Επομένως, ένα ακόμα όφελος σε περίπτωση επικράτησης αυτού του είδους μετακίνησης θα ήταν μία αισθητή μείωση της κυκλοφοριακής συμφόρησης στους δρόμους.

Στο σχεδιασμό διαδρομών ως κοινή χρήση οχημάτων (vehicle sharing) ορίζουμε την περίπτωση που επιβάτες με ίδιες διαδρομές μετακινούνται ταυτόχρονα με το ίδιο όχημα αποκομίζοντας κάποια οφέλη όπως χαμηλότερο κόστος μετακίνησης. Ως κοινή χρήση διαδρομών (ride sharing) ορίζουμε την περίπτωση που επιβάτες με διαφορετικές διαδρομές (μπορεί να έχουν κοινή προέλευση, κοινό προορισμό ή και τίποτα από τα δύο κοινό) χρησιμοποιούν ταυτόχρονα το ίδιο όχημα για κάποιο μέρος της διαδρομής τους. Τα οφέλη από αυτών των δύο υπηρεσιών περιγράφονται αναλυτικότερα στη συνέχεια.

Για να αξιοποιηθούν πλήρως αυτές οι υπηρεσίες, είναι απαραίτητη η εύρεση αποδοτικών διαδρομών για τα οχήματα που θα τις εξυπηρετούν, η οποία αποτελεί ένα πρόβλημα που έχει μελετηθεί για αρκετές δεκαετίες. Όταν μία εταιρία έχει τη δυνατότητα να μειώσει το συνολικό μήκος των διαδρομών ή να αυξήσει τον αριθμό των οχημάτων, μπορεί να παρέχει καλύτερη υπηρεσία στους πελάτες και να λειτουργεί με έναν πιο αποδοτικό τρόπο. Ένα τυπικό πρόβλημα δρομολόγησης οχημάτων περιλαμβάνει τον ταυτόχρονο καθορισμό των διαδρομών για πολλά οχήματα από ένα κεντρικό αμαξοστάσιο σε ένα σύνολο πελατών και την επιστροφή των οχημάτων στο αμαξοστάσιο χωρίς να έχουν ξεπεραστεί οι περιορισμοί χωρητικότητας κάθε οχήματος. Αυτό το πρόβλημα έχει μεγάλη οικονομική σημασία τόσο για ιδιωτικές επιχειρήσεις όσο και για τον δημόσιο τομέα όπου πρέπει να καθοριστούν διαδρομές οχημάτων για συστήματα λεωφορείων, ταχυδρομικούς μεταφορείς και άλλες δημόσιες υπηρεσίες οχημάτων. Στόχος είναι η εύρεση της ελάχιστης απόστασης που διανύεται από το σύνολο των οχημάτων ώστε να εξυπηρετηθούν οι πελάτες.

1.2 ΑΝΤΙΚΕΙΜΕΝΟ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός αλγορίθμου" ο οποίος έχει ως στόχο τον βέλτιστο σχεδιασμό δικτύου διαδρομών για αυτόνομα οχήματα με στόχο την εξυπηρέτηση ενός συνόλου πελατών. Οι πελάτες έχουν την δυνατότητα να κάνουν αιτήματα για την εξυπηρέτησή τους μέσω κάποιας ιστοσελίδας ή κάποιας εφαρμογής επιλέγοντας τον κόμβο προέλευσης και τον κόμβο προορισμού τους από το σύνολο των σημείων του δικτύου. Αυτά τα αιτήματα μπορούν να γίνουν δεκτά μέχρι ένα συγκεκριμένο χρονικό όριο πριν την έναρξη της βάρδιας ενώ δεν γίνονται δεκτές νέες ζητήσεις εφόσον η βάρδια ξεκινήσει. Τα οχήματα, στην αρχή, είναι συγκεντρωμένα στο αμαξοστάσιο, ξεκινάνε από αυτό και στο τέλος, αφού έχουν εξυπηρετηθεί όλοι οι πελάτες, επιστρέφουν σε αυτό. Αυτό το είδος του προβλήματος αναφέρεται στη βιβλιογραφία ως "one-to-many-to-one vehicle routing problem". Ο αλγόριθμος εφαρμόζεται στο οδικό δίκτυο της Αθήνας, όπου θεωρούνται 31 σημεία εξυπηρέτησης σε όλη την πόλη. Οι συντεταγμένες τους και οι αποστάσεις μεταξύ τους βασίζονται στην υπηρεσία google maps. Σχετικά με τις αποστάσεις, κατά κύριο λόγο προτιμήθηκαν οι συντομότεροι δρόμοι και όπου ήταν δυνατόν αποφεύχθηκαν οι αυτοκινητόδρομοι εξαιτίας της χαμηλής ταχύτητας των οχημάτων σε σχέση με τα συμβατικά. Τα σημεία επιλέχθηκαν σε δρόμους και συμβολές δρόμων όπου συνήθως παρουσιάζεται έντονη κυκλοφοριακή συμφόρηση και έτσι αναμένεται και η ζήτηση να είναι μεγάλη. Τα σημεία αυτά λειτουργούν ως σταθμοί από όπου τα οχήματα μπορούν να παραλάβουν ή/και να αφήσουν επιβάτες. Η επιλογή της ζήτησης έγινε με τρόπο τέτοιο ώστε να φεύγει τουλάχιστον ένας πελάτης από κάθε κόμβο και ο προορισμός τους να είναι τυχαίος. Επίσης, θεωρείται κοινή χρήση οχημάτων αλλά και διαδρομών. Στη γενική του μορφή το πρόβλημα αναφέρεται στη διεθνή βιβλιογραφία ως "Vehicle Routing Problem" ("VRP") με παραλαβές και παραδόσεις (pick -up and delivery) αλλά έχει και στοιχεία του προβλήματος dial-a-ride ("DARP").

Το πρόβλημα δρομολόγησης οχημάτων έχει στενή σχέση με το πρόβλημα του περιοδεύοντος πωλητή (Travelling Salesman Problem). Τα προβλήματα αυτά είναι NP-Hard τα οποία θεωρούνται από τα πολυπλοκότερα προβλήματα βελτιστοποίησης καθώς δεν υπάρχει κάποιος πολυώνυμος αλγόριθμος που να μπορεί κάθε φορά να βρίσκει την απόλυτα βέλτιστη λύση (Bell και McMullen, 2004). Για τέτοια προβλήματα η χρήση ευρετικών και μεταευρετικών θεωρούνται λογικές προσεγγίσεις. Μία μεταευρετική είναι η βελτιστοποίηση με βάση την λειτουργία των αποικιών των μυρμηγκιών (ant colony optimization). Ως μέθοδος βελτιστοποίησης επιλέχθηκε η βελτιστοποίηση με βάση τη λειτουργία των αποικιών των μυρμηγκιών ("Ant Colony Optimization") η οποία θα αναλυθεί σε επόμενο κεφάλαιο. Με τη χρήση αυτής της μεθόδου, ο σκοπός του σχεδιασμού είναι να δημιουργηθεί ένα σύνολο διαδρομών τέτοιο ώστε κάθε όχημα να διανύει ακριβώς μία διαδρομή, κάθε επιβάτης να εξυπηρετείται από ακριβώς ένα όχημα και η συνολική απόσταση που διανύεται από το σύνολο των οχημάτων της καλύτερης λύσης να είναι η ελάχιστη. Επιπλέον, κάθε όχημα δεν επισκέπτεται το ίδιο σημείο περισσότερες από μία φορές και το σύνολο των επιβατών που μεταφέρει κάθε όχημα δεν επιτρέπεται να υπερβαίνει τη χωρητικότητά του.

Επιπρόσθετα, επειδή αναφερόμαστε σε ηλεκτρικά οχήματα πρέπει να λάβουμε υπόψη και το εύρος της απόστασης που μπορούν αυτά να διανύσουν πριν χρειαστούν επαναφόρτιση. Για τον λόγο αυτό συνήθως χρησιμοποιείται μία μέση τιμή εύρους απόστασης ή χρόνου που μπορεί να καλύψει κάθε ένα από αυτά τα ομοειδή οχήματα πριν χρειαστεί να επισκεφτεί έναν σταθμό φόρτισης που στη συγκεκριμένη περίπτωση θεωρήθηκε ότι βρίσκεται στο αμαξοστάσιο.

1.3 ΔΟΜΗ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Η συγκεκριμένη διπλωματική εργασία αποτελείται από πέντε κεφάλαια συμπεριλαμβανομένης της εισαγωγής, καθώς και από τη λίστα με τις βιβλιογραφικές πηγές και το παράρτημα.

Το κεφάλαιο 1 αποτελεί την εισαγωγή στην εργασία, εξηγώντας το αντικείμενό της αλλά και τον γενικότερο προβληματισμό που οδήγησε στην εκπόνησή της.

Στο κεφάλαιο 2 πραγματοποιείται μία εισαγωγή στα προβλήματα δρομολόγησης αυτόνομων οχημάτων και γίνεται μία εκτενής ανασκόπηση της σχετικής βιβλιογραφίας καθώς και της βιβλιογραφίας σχετικά με τη βελτιστοποίηση με βάση τη λειτουργία των αποικιών των μυρμηγκιών.

Στο κεφάλαιο 3 γίνεται η περιγραφή του μαθηματικού προτύπου, δηλαδή της αντικειμενικής συνάρτησης που αποτελεί τη συνάρτηση προς επίλυση καθώς και των περιορισμών που την περιγράφουν. Επίσης, αναφέρεται το περιβάλλον στο οποίο αναπτύχθηκε το μοντέλο, αναλύεται πλήρως η μέθοδος βελτιστοποίησης που ακολουθήθηκε και περιγράφεται αναλυτικά ο αλγόριθμος.

Στο κεφάλαιο 4 παρουσιάζονται τα αποτελέσματα που προέκυψαν από την εφαρμογή του αλγορίθμου, δηλαδή η βέλτιστη λύση και αυτή αναλύεται σχηματικά και αναλυτικά και, επίσης, πραγματοποιείται ανάλυση ευαισθησίας για διάφορες παραμέτρους του μοντέλου.

Στο κεφάλαιο 5 παρουσιάζονται τα συμπεράσματα που προέκυψαν από την πραγματοποίηση της εργασίας καθώς και προτάσεις για περαιτέρω έρευνα σχετικά με το πρόβλημα.

Στο κεφάλαιο της βιβλιογραφίας δίνεται η λίστα με τις βιβλιογραφικές πηγές που χρησίμευσαν ως πηγές για την παρούσα διπλωματική εργασία.

Στο παράρτημα παρατίθεται ο κώδικας που αναπτύχθηκε για την εφαρμογή της συγκεκριμένης βελτιστοποίησης ώστε να επιλυθεί το πρόβλημα.

ΚΕΦΑΛΑΙΟ 2: ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ

2.1 ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό πραγματοποιείται μία εισαγωγή στο πρόβλημα της δρομολόγησης αυτόνομων οχημάτων και αναφέρεται συνοπτικά η βιβλιογραφία των γενικότερων προβλημάτων στα οποία ανήκει το αντικείμενο της παρούσας εργασίας. Αυτά τα προβλήματα είναι τα προβλήματα δρομολόγησης με παραλαβές και παραδόσεις και ειδικότερα τα προβλήματα dial-a-ride. Επίσης, αναφέρονται οι τρόποι με τους οποίους επιλύθηκαν παρόμοια προβλήματα αλλά και η μέθοδος βελτιστοποίησης που χρησιμοποιήθηκε, δηλαδή η βελτιστοποίηση με βάση τη λειτουργία των αποικιών των μυρμηγκιών.

2.2. ΤΟ ΠΡΟΒΛΗΜΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΜΕ ΠΑΡΑΛΑΒΕΣ ΚΑΙ ΠΑΡΑΔΟΣΕΙΣ

2.2.1 Ορισμός

Το κλασσικό πρόβλημα δρομολόγησης οχημάτων (vehicle routing problem) περιλαμβάνει ένα σύνολο πελατών που πρέπει να εξυπηρετηθεί από έναν ομογενή στόλο οχημάτων που είναι σταθμευμένα σε ένα κεντρικό αμαξοστάσιο. Σκοπός του προβλήματος είναι η δημιουργία ενός συνόλου διαδρομών οι οποίες ξεκινούν και τερματίζουν στο αμαξοστάσιο έτσι ώστε να εξυπηρετούνται όλοι οι πελάτες. Παράλληλα θα πρέπει τα αιτήματα των πελατών που έχουν ανατεθεί στα οχήματα να μην υπερβαίνουν τη χωρητικότητα του οχήματος που εξυπηρετεί τη διαδρομή. Στόχος του προβλήματος είναι η ελαχιστοποίηση της συνολικής απόστασης που διανύεται από τα οχήματα. Το πρόβλημα δρομολόγησης με παραλαβές και παραδόσεις είναι μία παραλλαγή του αρχικού προβλήματος δρομολόγησης οχημάτων καθώς αυτά δεν απαιτείται μόνο να παραδίδουν προϊόντα σε πελάτες αλλά και να παραλαμβάνουν προϊόντα από αυτούς. Τα προβλήματα με παραλαβές και παραδόσεις ταξινομούνται σε τρεις κατηγορίες (Nagy και Salhi, 2005) :

- Πρώτα οι παραδόσεις και μετά οι παραλαβές: τα οχήματα δέχονται παραλαβές αφού έχουν ολοκληρώσει τις παραδόσεις.
- Μεικτές παραδόσεις και παραλαβές: τα οχήματα πραγματοποιούν παραλαβές και παραδόσεις με οποιαδήποτε ακολουθία κατά τη διάρκεια των διαδρομών.
- Ταυτόχρονες παραλαβές και παραδόσεις: τα οχήματα πραγματοποιούν παραλαβές και παραδόσεις ταυτόχρονα.

Στην συγκεκριμένη εργασία χρησιμοποιείται η τρίτη μέθοδος, δηλαδή τα οχήματα παραλαμβάνουν και αφήνουν επιβάτες ταυτόχρονα κατά τη διάρκεια των διαδρομών.

2.2.2 Βιβλιογραφική Ανασκόπηση

Τα προβλήματα δρομολόγησης είναι τα προβλήματα που περιλαμβάνουν τον σχεδιασμό διαδρομών ενός στόλου οχημάτων ώστε αυτά να προσφέρουν κάποιου είδους υπηρεσία σε επιβάτες οι οποίοι είναι κατανεμημένοι σε ένα δίκτυο (Laporte, 1992; Eksioglou, 2009). Υπάρχουν πολλές παραλλαγές στο είδος της απαιτούμενης υπηρεσίας και λογικοί περιορισμοί ενώ όλες οι παραλλαγές είναι NP-hard. Το δίκτυο περιέχει μία ζήτηση από τους μετακινούμενους, ο καθένας από τους οποίους επιθυμεί να τον παραλάβουν από μία τοποθεσία και να τον αφήσουν σε μία άλλη. Ο στόλος των οχημάτων χρησιμοποιείται για να προσφέρει μετακίνηση στους πελάτες.

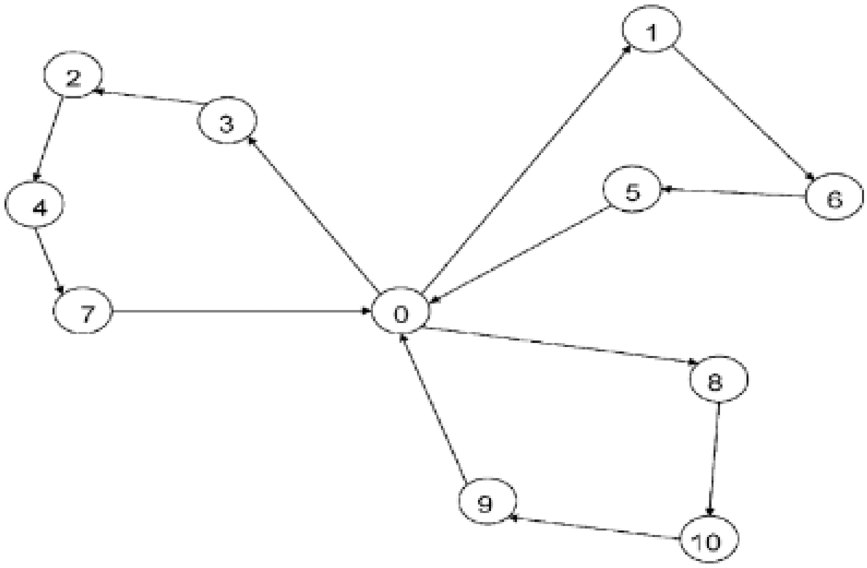
Το πρόβλημα δρομολόγησης εισήχθη από τον Min (1989) ως ένα πρόβλημα συλλογής και διανομής βιβλίων μεταξύ της Κεντρικής Βιβλιοθήκης και άλλων 22 βιβλιοθηκών στο Οχάιο χρησιμοποιώντας δύο οχήματα. Ο Min χρησιμοποίησε την προσέγγιση «πρώτα ομαδοποίηση και μετά δρομολόγηση» (cluster-first, route-second) και έλυσε το πρόβλημα του περιοδεύοντος πωλητή (Travelling Salesman Problem) με βελτιστοποίηση μικρότερων υποπροβλημάτων. Το 1992 ο Halse χρησιμοποιώντας την ίδια μέθοδο έλυσε το πρόβλημα αυτό ως προς περισσότερες μεταβλητές. Στην προσέγγισή του, οι κόμβοι διανέμονται εξαρχής στα οχήματα και στη συνέχεια το πρόβλημα λυνόταν με την χρήση ενός αλγορίθμου τοπικής αναζήτησης (3-opt). Ο Halse, επίσης, χρησιμοποίησε τη προσέγγιση Lagrange και τεχνικές δημιουργίας στηλών (column generation). Οι περιπτώσεις που μελέτησε αναφέρονταν σε ένα αμαξοστάσιο και σε περιπτώσεις με 22 ως 150 πελάτες. Οι Angelelli και Mansini (2002) εισήγαγαν στο πρόβλημα τα χρονικά παράθυρα (οι πελάτες έχουν ένα συγκεκριμένο χρονικό διάστημα μέσα στο οποίο μπορούν να δεχτούν επίσκεψη από ένα όχημα) και εφάρμοσαν προσέγγιση με διακλαδώσεις και αντίστοιχα βάρη (branch-and-price). Οι Gendreau et al. (1999) μελέτησαν το πρόβλημα του περιοδεύοντος πωλητή με παραδόσεις και παραλαβές. Αρχικά, έλυσαν το πρόβλημα αγνοώντας τους περιορισμούς και καθόρισαν τη σειρά των παραδόσεων και των παραλαβών.

Οι Casco, et al. (1988) πρότειναν μία προσέγγιση που βασιζόταν στο φορτίο. Το κόστος εισόδου για τους επιβάτες σχετιζόταν με το φορτίο που απέμενε να κατανεμηθεί κατά τη διάρκεια της διαδρομής του οχήματος. Οι Salhy και Nagy (1999) τροποποίησαν αυτή τη μέθοδο εισάγοντας τη λογική του συμπλέγματος. Στην διατύπωση του προβλήματος, οι κόμβοι αντιμετωπίζονταν ως «κομματιασμένοι» κόμβοι παραλαβής ή παράδοσης.

Αξιοσημείωτο, επίσης, είναι πως αυτοί ήταν που εισήγαγαν την ιδέα των πολλών αμαξοστασίων.

Ο Dethloff (2001) πρότεινε την χρήση ευρετικών που βασίζονται στην καταχώρηση (insertion-based heuristics) με τέσσερα διαφορετικά κριτήρια ως προς αυτή (διανυόμενη απόσταση, υπολειπόμενη χωρητικότητα, ακτινική επιβάρυνση και συνδυασμό των δύο προηγούμενων) και δημιούργησε 40 περιπτώσεις για να δοκιμάσει τον αλγόριθμό του. Συνέκρινε τα αποτελέσματα του με αυτά των Salgi και Nagy (1999) και κατέληξε σε μία βελτίωση του προβλήματος του Min. Οι Vural, et al. (2005) ανέφεραν βελτιώσεις στα αποτελέσματα των προβλημάτων του Dethloff εφαρμόζοντας έναν διπλό γενετικό αλγόριθμο. Σε αυτή τη προσέγγιση οι πρώτες διαδρομές δημιουργούνται και χωρίζονται σε υπο-διαδρομές, πραγματοποιείται μία τοπική αναζήτηση, μία διασταύρωση και στη συνέχεια εκτελούνται λειτουργίες μετάλλαξης. Οι Nagy και Salhi (2005) παρουσίασαν πολλές ευρετικές με τις οποίες πρώτοι βρήκαν λύση στο Πρόβλημα Δρομολόγησης Οχημάτων επιτρέποντας λύσεις που δεν ήταν εφικτές και στη συνέχεια τροποποιούσαν τη λύση ώστε να γίνει εφικτή για ένα η περισσότερα αμαξοστάσια. Αυτή η προσέγγιση παρουσιάζει βελτιώσεις σε σύγκριση με τα αποτελέσματα των Salhi και Nagy (1999). Οι Crispim και Brandao (2005) πρότειναν μία προσέγγιση υβριδικής αναζήτησης tabu εγγύτερου γείτονα. Εάν σε κάποιον σύνδεσμο εμφανίζονταν υπερφορτώσεις άλλαζαν τη σειρά των επιβατών στην πορεία μέχρι να καταστεί εφικτό το αποτέλεσμα . Για τη φάση της βελτίωσης χρησιμοποίησαν κινήσεις «εισαγωγής» και «ανταλλαγής» εφαρμόζοντας ποινή για τις υπερφορτώσεις. Άλλες σημαντικές έρευνες ήταν αυτές των Ropke και Pisinger (2006) οι οποίοι πρότειναν μία ευρετική βασισμένη σε μία αναζήτηση εγγύτερου γείτονα και αυτή των Wassan et al. (2008) οι οποίοι σχεδίασαν μία μεταευρετική αναζήτησης tabu.

Στο σχήμα 2.1 φαίνεται ένα παράδειγμα επίλυσης προβλήματος δρομολόγησης one-to-many-to-one, δηλαδή όπου τα οχήματα έχουν ως αφετηρία ένα αμαξοστάσιο, εξυπηρετούν ζητήσεις σε πολλές κόμβους και στο τέλος επιστρέφουν σε αυτό.



Σχήμα 2.1: Παράδειγμα επίλυσης προβλήματος δρομολόγησης one-to-many-to-one

2.2.3 Προβλήματα dial-a-ride

2.2.3.1 Εισαγωγή

Τα προβλήματα dial-a-ride (DARP) σχετίζονται με τη δρομολόγηση οχημάτων και τον προγραμματισμό τους για ένα συγκεκριμένο αριθμό χρηστών οι οποίοι κάνουν αιτήματα για παραλαβές και παραδόσεις ανάμεσα σε προελεύσεις και προορισμούς. Στόχος τους είναι η δημιουργία ενός συνόλου διαδρομών ελαχίστου κόστους ικανές να εξυπηρετήσουν όσο το δυνατόν περισσότερους πελάτες υπακούοντας σε ένα σύνολο περιορισμών. Η βιβλιογραφία σχετικά με αυτά τα προβλήματα αναφέρεται σε αυτό το κεφάλαιο.

2.2.3.2. Βιβλιογραφία

Τα προβλήματα dial a ride έχουν μελετηθεί αρκετά στο παρελθόν. Επιπλέον, επειδή τα προβλήματα dial a ride είναι NP-hard, ο σχεδιασμός αποτελεσματικών αλγόριθμων έχει λάβει μεγάλη προσοχή. Η επιλογή διαδρομών για μεγάλο αριθμό οχημάτων και τα αντίστοιχα αποτελέσματά της στην κυκλοφοριακή συμφόρηση δεν έχουν μελετηθεί στο παρελθόν.

Τα DARP μπορούν να ταξινομηθούν με πολλούς τρόπους. Αρχικά, χωρίζονται σε στατικές και δυναμικές παραλλαγές. Στις στατικές παραλλαγές (Berbeglia, 2007) υποθέτουμε ότι όλη η ζήτηση είναι γνωστή εκ των προτέρων, ενώ στις δυναμικές (Cordeau, 2007) ότι μπορεί να προκύψει καινούρια ζήτηση αφού έχουν ξεκινήσει κάποιες διαδρομές οχημάτων. Η δυναμική ζήτηση αυξάνει την πολυπλοκότητα του προβλήματος επειδή οι αλγόριθμοι πρέπει να λαμβάνουν υπόψη την αβεβαιότητα της ζήτησης ενώ οι διαδρομές δεν μπορούν να προγραμματιστούν από πριν. Επειδή τα DARP είναι NP-hard, χρειάζονται αρκετό χρόνο ώστε να επιλυθούν ακριβώς. Για αυτό προηγούμενες εργασίες χρησιμοποιούν γρήγορες ευρετικές για να λύσουν το DARP σε πραγματικό χρόνο (Calvo και Colorni, 2007 ; Ichua, 2007) συμπεριλαμβάνοντας συστήματα συντεταγμένων μεγάλης κλίμακας (Melachrinoudis, 2007).

Τα προβλήματα δρομολόγησης οχημάτων περιλαμβάνουν μεταβλητές που αφορούν τα οχήματα και τους επιβάτες. Αυτές που σχετίζονται με τα οχήματα σε περιπτώσεις που υπάρχει κοινή χρήση οχημάτων από διαφορετικούς επιβάτες εκφράζονται συνήθως με την χωρητικότητά τους (Agatz, 2012) αν και έχουν μελετηθεί και άλλοι παράγοντες στο παρελθόν (Parragh, 2012) . Η κοινή χρήση οχημάτων έχει μελετηθεί αρκετά και μπορεί να μειώσει τον αριθμό των απαιτούμενων διαδρομών και του απαιτούμενου στόλου (Fagnant Kockelman, 2015 ; Levin,2017). Όμως, η κοινή χρήση μπορεί να επεκτείνει την εφικτή περιοχή και τις αναθέσεις οχημάτων και έτσι αυξάνεται και η πολυπλοκότητα του προβλήματος (Agatz, 2012). Επιπλέον, οι πελάτες έχουν επιθυμητούς χρόνους παράδοσης και παραλαβής. Αυτοί οι πρόσθετοι περιορισμοί έχουν μελετηθεί σε προηγούμενες έρευνες και αλγορίθμους (πχ Jaw et al., 1986.). Φυσικά, οι συγκεκριμένοι αυτοί χρόνοι είναι δυνατό να κάνουν το σύστημα ανέφικτο. Επομένως συνήθως αυτοί οι χρόνοι είναι υποθετικοί. Περιορισμοί στο μέγεθος του στόλου ή μεγάλοι χρόνοι ταξιδιών που προκαλούνται από την κυκλοφοριακή συμφόρηση μπορούν να εμποδίσουν τα οχήματα να παραλάβουν τους επιβάτες ή να φτάσουν σε αυτούς στους επιθυμητούς χρόνους.

Τα προβλήματα DARP συνήθως διατυπώνονται με ακέραιο προγραμματισμό ανιχνεύοντας τις αλυσιδωτές διαδρομές κάθε οχήματος του στόλου ξεχωριστά. Ως τέτοια, οι αλγόριθμοι επίλυσης περιλαμβάνουν ακριβείς μεθόδους για την επίλυση ακέραιων προγραμμάτων όπως οι αλγόριθμοι branch-and-bound και branch-and-cut (Cordeau,2006 ; Ropke, 2009). Επειδή αυτά τα προβλήματα είναι NP-hard και συχνά χρειάζεται να επιλυθούν είτε γρήγορα είτε σε πραγματικό χρόνο, προηγούμενες έρευνες έχουν μελετήσει τη χρήση ευρετικών και μεταευρετικών (πχ. Levin και Boyles, 2016). Επίσης, η υπόθεση συνεχούς ροής είναι ένα σύνηθες φαινόμενο σε τέτοιου είδους προβλήματα (Chiu, 2011) αν και όχι τόσο σε αυτά με μικρό μέγεθος στόλου.

Τα προβλήματα δρομολόγησης οχημάτων συνήθως υποθέτουν ότι οι χρόνοι διαδρομών μεταξύ των κόμβων είναι προκαθορισμένοι. Επομένως, η βιβλιογραφία σε αυτό το πρόβλημα επικεντρώνεται στην αποφυγή των συνεπειών της κυκλοφοριακής συμφόρησης όπως οι αυξημένοι χρόνοι (Ahn και Shin, 1991; Kok, 2012), μειωμένη αξιοπιστία των υπηρεσιών(Fu, 2012 ; Xiang, 2008) και μεγαλύτερες εκπομπές αερίων (Lin, 2014 και Figliozzi, 2010). Η υπόθεση των προκαθορισμένων χρόνων εμφανίζεται στις περισσότερες πρακτικές εφαρμογές των προβλημάτων dial-a-ride επειδή ο αριθμός του στόλου είναι σχετικά μικρός σε σχέση με τα υπόλοιπα οχήματα του δικτύου.

2.3 ΚΟΙΝΟΧΡΗΣΤΑ ΑΥΤΟΝΟΜΑ ΟΧΗΜΑΤΑ

2.3.1 Εισαγωγή

Τα κοινόχρηστα αυτόνομα οχήματα (Κ.Α.Ο.) είναι αυτόνομα οχήματα τα οποία έχουν τη δυνατότητα να μεταφέρουν διαφορετικούς επιβάτες κατά τη διάρκειας μίας διαδρομής λόγω σύμπτωσης αυτής ή μέρους της. Παρόλο που το πρόβλημα δρομολόγησης κοινόχρηστων αυτόνομων οχημάτων (SAV routing problem) είναι ένας τύπος DARP, διαφέρει σημαντικά από προηγούμενες μελέτες του DARP εξαιτίας του μεγάλου αριθμού οχημάτων που περιέχει. Σε αυτό το κεφάλαιο αναφέρεται η βιβλιογραφία σχετικά με αυτά τα οχήματα.

2.3.2 Βιβλιογραφική Ανασκόπηση

Τα ιδιωτικά οχήματα ξοδεύουν πολλές ώρες κατά τη διάρκεια της ημέρας μένοντας στάσιμα ενώ οι ιδιοκτήτες τους βρίσκονται στον χώρο εργασίας ή απασχολούνται σε άλλες δραστηριότητες. Προηγούμενες μελέτες πρότειναν ότι τα αυτόνομα οχήματα (Α.Ο.) θα μπορούσαν να ταξιδεύουν άδεια ώστε να αποφεύγουν τις χρεώσεις των χώρων στάθμευσης (Levin και Boyles, 2015) ή/και να προσφέρουν υπηρεσίες σε διάφορα νοικοκυριά (de Almeida Correia και van Arem, 2016). Αφού τα Α.Ο. θα μπορούσαν να ταξιδεύουν εντός του δικτύου χωρίς να μεταφέρουν επιβάτες, προηγούμενες μελέτες πρότειναν ένα σύστημα Κ.Α.Ο. Πιο συγκεκριμένα, διαφορετικοί επιβάτες θα μπορούσαν να μοιράζονται το ίδιο όχημα οργανώνοντας τις διαδρομές τους σε διαφορετικές χρονικές στιγμές (Fagnant και Kockelman, 2015). Προτάθηκε, επίσης, τα ιδιωτικά οχήματα να παρέχουν μία dial-a-ride υπηρεσία σε πολλούς επιβάτες του ίδιου νοικοκυριού και αναπτύχθηκε μία διαμόρφωση ισορροπίας χρηστών (“user equilibrium formulation”) (De Almeida Correia και van Arem, 2016). Όμως, τα Κ.Α.Ο. δεν περιορίζουν τον ιδιοκτήτη τους σχετικά με τις διαδρομές. Οι ιδιοκτήτες ιδιωτικών αυτόνομων οχημάτων θα μπορούσαν να τα παρέχουν για δημόσια χρήση αντί ενός αντίτιμου και τα Κ.Α.Ο. θα μπορούσαν να είναι υπό την ιδιοκτησία εταιριών ταξί. Έτσι, τα Κ.Α.Ο. θα εξελίσσονταν σε μία υπηρεσία φτηνότερων ταξί η οποία θα ήταν και πιο ευέλικτη στη ζήτηση. Χωρίς την ανάγκη ενός οδηγού, το κόστος υπηρεσίας θα μπορούσε να προσεγγίσει αυτό του ιδιωτικού οχήματος. Προηγούμενες έρευνες ασχολήθηκαν με την αντικατάσταση Ι.Χ. με Κ.Α.Ο. (Fagnant και Kockelman, 2014) και εκτίμησαν ότι ένα Κ.Α.Ο. θα μπορούσε να αντικαταστήσει μέχρι και έντεκα Ι.Χ. Ο Chen (2016) προέβλεψε ότι μεταξύ 14% και 39% των μετακινούμενων θα επέλεγε τα Κ.Α.Ο.. Βασισμένος σε μία έρευνα προτιμήσεων ο Krueger (2016) εκτίμησε την χρήση Κ.Α.Ο. σε διαφορετικές πληθυσμιακές ομάδες και την εξάρτησή της από τους χρόνους ταξιδιού και αναμονής.

Οι Fagnant και Kockelman (2014) παρατήρησαν ένα ποσοστό αντικατάστασης ενός Κ.Α.Ο. αντί για έντεκα Ι.Χ. σε ένα δίκτυο πλέγματος. Ο Fagnant (2015) μελέτησε το δίκτυο και τη ζήτηση σε αυτό του Austin, Texas και βρήκε ένα ποσοστό αντικατάστασης ενός Κ.Α.Ο. αντί για 9,3 Ι.Χ. Ενώ ο Chen (2016) ανέφερε ένα ποσοστό αντικατάστασης μόλις του ενός Κ.Α.Ο. αντί για 3,7 Ι.Χ., είχε χρησιμοποιήσει ηλεκτρικά Κ.Α.Ο. τα οποία έχουν παρόμοιο κόστος ανά μίλι με τα Ι.Χ.. Ο Burns (2013) έδειξε ότι ένας πολύ μικρότερος στόλος Κ.Α.Ο. θα μπορούσε να προσφέρει υπηρεσίες σε μετακινούμενους τόσο εντός πόλης όσο και στα προάστιά της. Ο Spieser (2014) υπολόγισε ότι ένα Κ.Α.Ο. θα αντικαθιστούσε τρία Ι.Χ. στη Σιγκαπούρη. Φυσικά, ενώ ο αριθμός των οχημάτων στον δρόμο θα μειωνόταν, ο αριθμός των ταξιδιών ανά όχημα θα αυξανόταν αντίστοιχα ώστε να εξυπηρετηθούν όλοι οι μετακινούμενοι.

Τα Κ.Α.Ο. μετέφεραν μόνο έναν μετακινούμενο τη φορά στις προηγούμενες μελέτες, αλλά οι Fagnant και Kockelman (2015) και Levin (2017) μελέτησαν τις επιρροές της κοινής χρήσης αυτών των οχημάτων. Ο Zhu (2016) βρήκε ότι η κοινή χρήση θα μπορούσε να μειώσει τα μίλια που διανύονται στο 34% σε σχέση με τα συμβατικά οχήματα. Παρά το γεγονός ότι η κοινή χρήση θα μπορούσε να μειώσει το κόστος ανά μετακινούμενο και ανά μίλι ταξιδιού (Fagnant και Kockelman, 2015, Levin, 2017), αυξάνει και τον χρόνο διαδρομής και θα μπορούσε ενδεχομένως να μειώσει και την άνεση των επιβατών. Και οι δύο παράγοντες είναι πιθανό να μειώνουν την προθυμία μετακινούμενων να επιλέξουν Κ.Α.Ο. αντί για ιδιωτικά οχήματα. Ακόμη, μετακινούμενοι που χρησιμοποιούν τα μέσα μαζικής μεταφοράς μπορεί να προτιμούν αυτά τα οχήματα σε αντίθεση με άλλες επιλογές για μετακίνηση από τον τερματικό σταθμό ενός μέσου δημόσιας συγκοινωνίας στον τελικό τους προορισμό (Yap, 2016).

Παρά το ενδιαφέρον που έχει εμφανιστεί για τα κοινόχρηστα αυτόνομα οχήματα, η δρομολόγησή τους δεν έχει λάβει την αντίστοιχη προσοχή. Προηγούμενες μελέτες χρησιμοποίησαν ευρετικές με προσομοιώσεις βασισμένες σε πράκτορες (agent-based) για τον σχεδιασμό των διαδρομών των Κ.Α.Ο., οι οποίες ήταν επαρκώς αποδοτικές για να προβάλλουν τα πιθανά οφέλη. Όμως, οι περισσότερες έρευνες δεν έχουν συμπεριλάβει τα οφέλη των Κ.Α.Ο. στην κυκλοφοριακή συμφόρηση. Στόλοι Κ.Α.Ο. αρκετά μεγάλοι ώστε να αντικαταστήσουν μεγάλους αριθμούς Ι.Χ. (Fagnant και Kockelman, 2014; Fagnant, 2015; Fagnant και Kockelman, 2015) θα είχαν αντίστοιχα μεγάλα πλεονεκτήματα στη συμφόρηση στους δρόμους. Επομένως, η δρομολόγηση των Κ.Α.Ο. γίνεται αρκετά σημαντική καθώς περισσότερες βέλτιστες λύσεις μειώνουν τον χρόνο ταξιδιών καθώς και την συμφόρηση που δημιουργείται από αυτά. Μία απλή ευρετική θα μπορούσε να δημιουργήσει μία διαδικασία ανατροφοδότησης στην οποία τα Κ.Α.Ο. δημιουργούν σημαντικές καθυστερήσεις το ένα στο άλλο, πιθανώς δημιουργώντας ακραίες καταστάσεις κυκλοφοριακής συμφόρησης. Αυτά τα αποτελέσματα δεν έχουν παρατηρηθεί σε πολλές προηγούμενες εργασίες καθώς οι χρόνοι διαδρομής θεωρήθηκαν προκαθορισμένοι. Όμως, ο Levin (2017) ενσωμάτωσε μία συμπεριφορά Κ.Α.Ο. βασισμένη σε πράκτορες μέσα σε ένα δυναμικό μοντέλο φορτίων, και βρέθηκε ότι αυτά θα μπορούσαν να δημιουργήσουν συμφόρηση χειρότερη από αυτή των Ι.Χ. εξαιτίας της επανατοποθέτησής τους στο αμαξοστάσιο όταν αυτά είναι άδεια.

Βέλτιστες λύσεις στο πρόβλημα σχεδιασμού διαδρομών αυτόνομων οχημάτων θα μπορούσαν επομένως να οδηγήσουν σε μειωμένους χρόνους ταξιδιών, κόστους σχεδιασμού και χρόνων αναμονής. Για παράδειγμα, ο Burns (2013) βρήκε ότι ένας μειωμένος αριθμός στόλου θα μπορούσε να βελτιώσει τους χρόνους αναμονής και την χρησιμοποίηση των οχημάτων στο Μανχάταν. Ο Alonso-Mora (2017) δημιούργησε ένα μαθηματικό μοντέλο για κοινή χρήση οχημάτων σε πραγματικό χρόνο καθώς και μία ευρετική ανάθεσης και δοκιμάστηκε σε έναν μεγάλο αριθμό ταξί στην Νέα Υόρκη. Ο Pavone (2012) διερεύνησε την εξισορρόπηση φορτίου και τις επανατοποθετήσεις όταν τα οχήματα είναι άδεια για την κινητικότητα σε συστήματα ζήτησης χρησιμοποιώντας μία προσομοίωση στη Matlab μαζί με μία δοκιμή ρομποτικής μικρής κλίμακας.

Το πρόβλημα δρομολόγησης για κοινόχρηστα αυτόνομα οχήματα είναι ενδιαφέρον ότι μπορεί να έχει είτε στατική είτε δυναμική ζήτηση ανάλογα με την ώρα της ημέρας που μας ενδιαφέρει. Οι επιβάτες που χρησιμοποιούν αυτό το μέσο μεταφοράς αντί για ιδιωτικά αυτοκίνητα μπορούν να έχουν προγραμματίσει τις διαδρομές που πρόκειται να κάνουν ακόμα και αρκετό καιρό νωρίτερα. Για παράδειγμα, επιβάτες που κάνουν τη διαδρομή κατοικία - τόπος εργασίας έχουν ένα καθορισμένο πρόγραμμα ενώ άλλα είδη διαδρομών (όπως για αναψυχή) τείνουν να δημιουργούν δυναμική ζήτηση. Είναι γεγονός πως η στατική ζήτηση (είναι γνωστή πριν την αρχή της βάρδιας) έχει μεγάλη εφαρμογή στις ώρες αιχμής.

2.4 ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΜΕ ΒΑΣΗ ΤΗ ΛΕΙΤΟΥΡΓΙΑ ΤΩΝ ΑΠΟΙΚΙΩΝ ΤΩΝ ΜΥΡΜΗΓΚΙΩΝ

2.4.1 Εισαγωγή

Η νοημοσύνη σμήνους (swarm intelligence) βασίζεται σε αλγορίθμους που μιμούνται τη φυσική συμπεριφορά ενός ζώου ή κάποιου φυτικού συστήματος. Η συμπεριφορά με στόχο την αναζήτηση τροφής, αναπαραγωγή και αποικιοποίηση διάφορων ειδών χρησιμοποιείται για να λύσει πολλά πραγματικά προβλήματα της μηχανικής. Το σύστημα βελτιστοποίησης με βάση τη λειτουργία των αποικιών των μυρμηγκιών (Ant Colony Optimization, “ACO”) είναι μία νέα μεταευρετική προσέγγιση η οποία είναι εμπνευσμένη από τη συμπεριφορά αληθινών αποικιών μυρμηγκιών κατά την αναζήτηση τροφής.

2.4.2 Βιβλιογραφική ανασκόπηση

Το σύστημα των αποικιών των μυρμηγκιών αρχικά εισήχθη από τους Dorigo και Gambardella το 1997(Dorigo and Gambardella, 1997; Dorigo et al, 1996). Αλγόριθμοι βασισμένοι στα μυρμήγκια αναφέρονται στην βιβλιογραφία σε προσπάθειες επίλυσης μεγάλου αριθμού μηχανικών και συμβατικών μαθηματικών προβλημάτων και ιδιαίτερα για να λύσουν συνδυαστικά προβλήματα βελτιστοποίησης.

Ο Bullnheimer (1999) προτείνει ένα σύστημα μυρμηγκιών βασισμένο στην κατάταξη ως μία παραλλαγή του κλασσικού αλγορίθμου. Εν τω μεταξύ, οι Stutzle και Hoos (2000) συνεισέφεραν στη βελτίωση του συστήματος περιορίζοντας τη συγκέντρωση φερομόνης. Το προτεινόμενο σύστημα μυρμηγκιών εξασφαλίζει τη διαδικασία ενημέρωσης της φερομόνης για την εύρεση καλύτερων λύσεων με βάση το ιστορικό αναζήτησης. Μία ευρύτερη ανάλυση της συγκεκριμένης μεταευρετικής και της εφαρμογής της παρέχεται από τους Dorigo και Stutzle (2004) και τους Dorigo και Blum (2005). Οι Solnon και Bridge (2006) πρότειναν έναν γενικό αλγόριθμο ACO για την επίλυση προβλημάτων επιλογής υποσυνόλου όπως προβλήματα μέγιστης κλίσης (maximum clique problems), προβλήματα σακιδίου (knapsack problems) και προβλήματα ικανοποίησης περιορισμών. Παρά το γεγονός ότι τα προβλήματα διαφέρουν, η στρατηγική αναζήτησης που υιοθετήθηκε βασίζεται σε μία ισχυρή εκμετάλλευση του ιστορικού αναζήτησης που οδηγεί στην διαδικασία αναζήτησης μυρμηγκιών.

Σχεδιασμοί αλγορίθμων ACO και ποικιλία εφαρμογών αναφέρονται από τους Mullen (2009), Pedemonte (2011), Mohan και Baskaran (2012) και Neto και Filbo (2013). Πρόσφατη αναζήτηση σε συνδυαστικά προβλήματα βελτιστοποίησης και σε προβλήματα δικτύων δείχνουν ότι αλγόριθμος βελτιστοποίησης με βάση τη λειτουργία των αποικιών των μυρμηγκιών έχει εφαρμοστεί με επιτυχία στο πρόβλημα του περιοδεύοντος πωλητή, στο πρόβλημα τετραγωνικής ανάθεσης (quadratic assignment problem) και στο job-shop scheduling problem (Yu, 2009). Το βασικό μοντέλο διαδρομών, το πρόβλημα του περιοδεύοντος πωλητή έχει μελετηθεί εκτενώς από την ερευνητική κοινότητα. Μία πληθώρα αλγορίθμων μυρμηγκιών έχουν αναφερθεί ότι σχετίζονται με αυτό και τις παραλλαγές του (Zhao,2010; Elloumia,2014; Saenphon,2014; Ariyasingha και Fernando, 2015). Η διαδικασία «αυτό-μάθησης» και η λήψη αποφάσεων βασισμένη στις πιθανότητες από τη συλλογική νοημοσύνη των μυρμηγκιών καθιστούν τον αλγόριθμο πολύ ελκυστικό για την επίλυση προβλημάτων που είναι βασισμένα σε δίκτυα όπως τα προβλήματα δρομολόγησης.

2.5 ΔΡΟΜΟΛΟΓΗΣΗ ΗΛΕΚΤΡΙΚΩΝ ΟΧΗΜΑΤΩΝ

2.5.1 Εισαγωγή

Σε αυτό το κεφάλαιο παρουσιάζεται η βιβλιογραφία σχετικά με τον σχεδιασμό διαδρομών στα ηλεκτρικά οχήματα. Η χρήση ηλεκτρικών οχημάτων απαιτεί την ενσωμάτωση περιορισμών απόστασης που σχετίζονται με τη φόρτιση της μπαταρίας. Οι περιορισμοί στην απόσταση μπορούν να περιλαμβάνουν χρονικούς περιορισμούς ή περιορισμούς απόστασης υποθέτοντας μια μέση ταχύτητα για τα οχήματα. Εξαιτίας της πληθώρας βενζινάδικων και της κυριαρχίας των οχημάτων που χρησιμοποιούν βενζίνη, δεν έχει δοθεί ιδιαίτερη έμφαση στους περιορισμούς απόστασης.

2.5.2 Βιβλιογραφική Ανασκόπηση

Το πρόβλημα δρομολόγησης ηλεκτρικών οχημάτων είναι μία επέκταση του κλασσικού προβλήματος σχεδιασμού διαδρομών (VRP). Όταν χρησιμοποιούνται χρονικά διαστήματα (time windows), η εξυπηρέτηση ενός πελάτη πρέπει να γίνει εντός ενός συγκεκριμένου χρονικού πλαισίου , περίπτωση που βρίσκει μεγάλη εφαρμογή στα προβλήματα του πραγματικού κόσμου (Braysy και Gendreau, 2005).

Ένα άλλο σχετικό πρόβλημα είναι η επέκταση του προβλήματος δρομολόγησης με πολλά αμαξοστάσια (Crevier, 2007). Σε αυτό, τα οχήματα είναι κατανεμημένα σε αμαξοστάσια που βρίσκονται διασκορπισμένα στον χώρο και το κάθε όχημα πρέπει να επιστρέψει στο αμαξοστάσιο από το οποίο ξεκίνησε την διαδρομή του. Αυτό το πρόβλημα βρήκε εφαρμογή στην εξυπηρέτηση παραγγελιών λαχανικών στο Μόντρεαλ. Το μοντέλο αυτό θεωρεί ενδιάμεσα αμαξοστάσια από τα οποία τα οχήματα προμηθεύονται με αγαθά κατά την διάρκεια μίας διαδρομής.

Στη συνέχεια ο Tarantilis (2008) μετονόμασε το πρόβλημα σε "Πρόβλημα δρομολόγησης με ενδιάμεσες εγκαταστάσεις ανατροφοδότησης". Προτείνει μία υβριδική ευρετική τοπικής αναζήτησης και ακολουθεί μία διαδικασία τριών βημάτων. Πρώτα, δημιουργείται μία αρχική λύση μέσω μίας ευρετικής που λειτουργεί με στόχο τον περιορισμό του κόστους. Δεύτερον, εφαρμόζεται ένας αλγόριθμος που χρησιμοποιεί το Travelling Salesman σε τοπική φάση. Τρίτον, η λύση βελτιώνεται ακόμα περισσότερο μέσω μίας καθοδηγούμενης τοπικής αναζήτησης.

Σχετικά περιορισμένος αριθμός εργασιών έχει δημοσιευτεί σε προβλήματα βελτιστοποίησης σχετικά με εναλλακτικά καύσιμα. Τα περισσότερα άρθρα ασχολούνται με την χωροθέτηση σταθμών επαναφόρτισης στο πλαίσιο της κατασκευής υποδομών για συμπιεσμένο φυσικό αέριο (Boostani, 2010) ή ηλεκτρισμό (Qiu, 2011). Για να μοντελοποιηθεί η ζήτηση καυσίμου για διαδρομές μικρής απόστασης σε αστικές περιοχές, οι επιβάτες συνήθως αντιστοιχίζονται σε κόμβους και δημιουργείται μία διατύπωση βασισμένη σε αυτούς τους κόμβους. Λαμβάνοντας υπόψη τις διαδρομές μεγάλων αποστάσεων, χρησιμοποιείται η ροή μεταξύ προορισμού και προέλευσης ως ένας τρόπος μέτρησης της ζήτησης (Wang και Lin, 2009; Wang, 2010).

Άλλες εργασίες στοχεύουν στην εύρεση της συντομότερης διαδρομής σε σχέση με την ενέργεια έχοντας γνωστή την προέλευση και την τοποθεσία το οποίο μπορεί να χρησιμοποιηθεί, για παράδειγμα, σε συστήματα πλοήγησης. Με δεδομένη τη χωρητικότητα της μπαταρίας, ο σκοπός είναι η μεγιστοποίηση του επιπέδου της ενέργειας στον προορισμό όπου θετικοί σύνδεσμοι αναπαριστούν κατανάλωση ενέργειας και αρνητικοί αναπαριστούν ανάρρωση (Artmeier, 2010). Οι Wang και Shen (2007) προτείνουν ένα πρόβλημα προγραμματισμού για ηλεκτρικά λεωφορεία το οποίο ονομάζεται «Πρόβλημα δρομολόγησης με περιορισμούς χρόνου διαδρομής και καυσίμου» (Vehicle Scheduling Problem with Route and Fueling Time Constraints). Αναθέτουν ένα χρονοδιάγραμμα με διαδρομές οι οποίες είναι γνωστές εκ των προτέρων σε λεωφορεία με στόχο την ελαχιστοποίηση του συνολικού χρόνου που αυτά παραμένουν αδρανή. Το εύρος διαδρομών περιορίζεται από τη φόρτιση των οχημάτων έτσι ώστε κάθε όχημα να πρέπει να επαναφορτιστεί πριν από αρκετές διαδρομές.

Στο βαθμό που γνωρίζουμε, υπάρχουν τρεις δημοσιεύσεις που πραγματεύονται ρητά τα ιδιαίτερα χαρακτηριστικά των εναλλακτικών καυσίμων και τα προσαρμόζουν στο πρόβλημα δρομολόγησης οχημάτων. Ο Goncalves (2011) εξετάζει ένα πρόβλημα δρομολόγησης με παραδόσεις και παραλαβές με έναν συνδυαστικό στόλο που αποτελείται από ηλεκτρικά οχήματα και οχήματα που χρησιμοποιούν κινητήρες εσωτερικής καύσης. Στόχος του είναι να ελαχιστοποιήσει τα συνολικά κόστη τα οποία αποτελούνται από προκαθορισμένα κόστη σχετικά με το όχημα και μεταβλητά κόστη. Εξετάζει, επίσης, περιορισμούς σχετικά με τον χρόνο και την χωρητικότητα και υποθέτει έναν χρόνο επαναφόρτισης των ηλεκτρικών οχημάτων, ο οποίος υπολογίζεται από τη συνολική διανυόμενη απόσταση και το εύρος από τη χρήση μίας μπαταρίας. Όμως, δεν ενσωματώνει την πραγματική τοποθεσία των σταθμών επαναφόρτισης στο μοντέλο του. Επομένως, προτείνει ένα συνδυαστικό στόλο οχημάτων με έναν επιπρόσθετο παράγοντα που εξαρτάται από την απόσταση.

Οι Erdogan και Miller-Hooks (2012) είναι οι πρώτοι που συνδυάζουν ένα πρόβλημα δρομολόγησης με την πιθανότητα της επαναφόρτισης ενός οχήματος σε έναν σταθμό κατά τη διάρκεια της διαδρομής. Η εργασία αφορά κυρίως στόλους οχημάτων που λειτουργούν σε μία ευρεία γεωγραφική περιοχή και χρησιμοποιούν βιοκαύσιμο, υγρό φυσικό αέριο ή φυσικό αέριο. Για αυτά τα καύσιμα υπάρχει ένα περιορισμένο σύνολο υποδομών ανατροφοδότησης ενώ οι χρόνοι ανεφοδιασμού θεωρούνται προκαθορισμένοι. Το πρόβλημα θεωρεί μία μέγιστη διάρκεια διαδρομής και περιορισμό στη διαθεσιμότητα καυσίμου. Το καύσιμο καταναλώνεται με ένα συγκεκριμένο ποσοστό ανά διανυόμενη απόσταση και μπορεί να αναπληρωθεί σε οποιαδήποτε στιγμή.

ΚΕΦΑΛΑΙΟ 3: ΜΕΘΟΔΟΛΟΓΙΑ

3.1 ΕΙΣΑΓΩΓΗ

Το συγκεκριμένο πρόβλημα διατυπώνεται ως μία επέκταση του προβλήματος δρομολόγησης οχημάτων (Vehicle Routing Problem). Σε αυτή την ενότητα περιγράφεται το μαθηματικό πρότυπο που εκφράζει το πρόβλημα, η αντικειμενική συνάρτηση και οι περιορισμοί που της επιβάλλονται. Επίσης, περιγράφεται η βελτιστοποίηση αποικιών μυρμηγκιών η οποία χρησιμοποιήθηκε για την εύρεση της βέλτιστης λύσης του προβλήματος.

3.2 ΜΑΘΗΜΑΤΙΚΟ ΠΡΟΤΥΠΟ

Ο στόχος του αλγορίθμου είναι η ελαχιστοποίηση του λειτουργικού κόστους το οποίο εκφράζεται με τη συνολική απόσταση που διανύουν όλα τα οχήματα μίας πλήρους λύσης. Τα οχήματα ταξιδεύουν με μία σταθερή ταχύτητα, οι προελεύσεις και οι προορισμοί των πελατών είναι γνωστοί εκ των προτέρων και τα οχήματα ταξιδεύουν ανάμεσα σε δύο σημεία χρησιμοποιώντας τη διαδρομή με τη μικρότερη απόσταση αποφεύγοντας τους αυτοκινητόδρομους όπου αυτό είναι δυνατό. Η αντικειμενική συνάρτηση του μαθηματικού προτύπου εκφράζει τη συνολική απόσταση που διανύουν τα οχήματα.

3.2.1 Αντικειμενική Συνάρτηση

Το δίκτυο μπορεί να προσδιοριστεί ως ένα γράφημα G=(N,A) όπου N= {1,…,n}, ενώ ο κόμβος 0 αντιπροσωπεύει την αρχή της διαδρομής. Επίσης, τα οχήματα έχουν μία μέγιστη χωρητικότητα Q(=4 στη συγκεκριμένη περίπτωση) και εξυπηρετούν όσο το δυνατόν περισσότερους πελάτες υπακούοντας σε αυτόν τον περιορισμό. Κάθε όχημα εξυπηρετεί μία διαδρομή i-j απευθείας είτε όταν οι κόμβοι i και j αντιπροσωπεύουν την προέλευση και τον προορισμό αντίστοιχα κάποιου συνόλου πελατών είτε όταν ο κόμβος i είναι ο προορισμός κάποιων πελατών και ο κόμβος j η προέλευση του επόμενου συνόλου πελατών που θα εξυπηρετηθούν.

Έστω:

- M: το σύνολο των οχημάτων
- n: ο αριθμός των ζητήσεων που πρέπει να εξυπηρετηθούν.
- Q: η μέγιστη χωρητικότητα κάθε οχήματος
- c_{ij} : το κόστος πραγματοποίησης της διαδρομής ij, δηλαδή στη συγκεκριμένη περίπτωση η απόσταση μεταξύ των κόμβων ij βασισμένη στο google maps. Υποθέτουμε ότι το μητρώο των αποστάσεων είναι συμμετρικό, δηλαδή ισχύει $c_{ij}=c_{ji}$.
- Q_i^m: το φορτίο του οχήματος m όταν φεύγει από τον κόμβο i
- q_i^m: ο αριθμός πελατών στον κόμβο i που θα εξυπηρετηθούν από το όχημα m
- K ={0,1,…k…} το σύνολο των επιβατών μιας και κάθε ζήτηση i μπορεί να περιέχει περισσότερους από έναν επιβάτες.
- L: το μέγιστο μήκος διαδρομής που μπορεί να καλύψει ένα όχημα και

Η αντικειμενική συνάρτηση είναι η εξής: $minZ = \sum_{m\in M}\sum_{i\in N}\sum_{j\in N} c_{ij} * x_{ij}^m$ (3.1) ,
υπό τους περιορισμούς:

- $x_{ij}^m = \begin{cases} 1 \ , & \text{αν το όχημα } m \text{ κάνει απευθείας την διαδρομή } ij \\ 0 \ , & \text{αλλιώς} \end{cases}$ (3.2)
- $\sum_j x_{ij}^m = 1, \ \forall \ i\in N, \ m\in M$ (3.3)
- $\sum_j x_{ij}^m = \sum_i x_{ij}^m, \ \forall$ i∈N, j∈N, m∈M (3.4)
- $Q_j^m \geq (Q_i^m + q_j^m)\, x_{ij}^m$, $\forall$ i∈N, $J\in N, \ m\in M$ (3.5)

- max {0,q_i^m} ≤Q_i^m ≤ min {Q,Q + q_i^m}, ∀ i∈N, $m∈M$ (3.6)

- $\sum_{i∈N} \sum_{j∈N} c_{ij} * x_{ij}^m$ ≤ L, ∀ $m∈M$ (3.7)

- $\sum_{j∈N} x_{0j}^m$ =1, ∀ m∈M (3.8)

- $\sum_{i∈N} x_{i0}^m$=1, ∀ m∈M (3.9)

- x_{ij}^m∈S, ∀ i∈N, j∈N, m∈M (3.10)

- S=$\{x_{ij}^m : \sum_{i∈B} \sum_{j∈B} x_{ij}^m \leq |B| -1, \ B⊆=N, |B|≥2\}$ (3.11)

3.2.2 Περιορισμοί

Η εξίσωση (3.2) ορίζει τις τιμές της μεταβλητής x_{ij}^m. Οι περιορισμοί (3.3) και (3.4) διασφαλίζουν ότι κάθε αίτημα εξυπηρετείται ακριβώς μία φορά και ότι οι κόμβοι προέλευσης και προορισμού δέχονται επίσκεψη από το ίδιο όχημα. Οι περιορισμοί χωρητικότητας επιβάλλονται με τις ανισώσεις (3.5) και (3.6). Η ανίσωση (3.7) διασφαλίζει ότι κάθε όχημα δεν θα υπερβεί τα χιλιόμετρα που του επιτρέπει η μπαταρία του οχήματος να διανύσει .Τα οχήματα έχουν σταθερή ταχύτητα και χωρητικότητα. Η ζήτηση είναι γνωστή στην αρχή της βάρδιας και οι προορισμοί και προελεύσεις της είναι μεταξύ όλων των κόμβων εκτός του αμαξοστασίου .Οι στάσεις των οχημάτων είναι προκαθορισμένες και δεν αλλάζουν. Οι περιορισμοί 3.8 και 3.9 εξασφαλίζουν ότι κάθε όχημα ξεκινά και τερματίζει στον κόμβο 0 (αμαξοστάσιο). Οι περιορισμοί 3.10 και 3.11 αποκλείουν το σχηματισμό υπό-διαδρομών (sub-tour_elimination).

3.3 ΠΕΡΙΒΑΛΛΟΝ ΥΛΟΠΟΙΗΣΗΣ ΜΟΝΤΕΛΟΥ

Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την ανάπτυξη του κώδικα ήταν η Python η οποία μαζί με τη C++ είναι οι δύο γλώσσες που έχουν κυριαρχήσει στην επίλυση προβλημάτων δρομολόγησης. Η Python αποτελεί μία γλώσσα υψηλού επιπέδου προγραμματισμού, δηλαδή μία γλώσσα προγραμματισμού η οποία με αριθμητικά και γλωσσικά στοιχεία συνηθισμένα σε επιστήμονες και τεχνικούς, επιτρέπει εύκολα τη δημιουργία προγραμμάτων και εφαρμογών χωρίς την απαίτηση της γνώσης της εσωτερικής δομής και λειτουργίας του υπολογιστή. Χαρακτηριστικά της είναι η ευκολία χρήσης και το συντακτικό της που επιτρέπει στους προγραμματιστές να εκφράσουν έννοιες σε λιγότερες γραμμές κώδικα σε σχέση με τις περισσότερες άλλες γλώσσες. Επίσης, διαθέτει πολλές βιβλιοθήκες οι οποίες καθιστούν την εκτέλεση κάποιων εργασιών ευκολότερη.

Στην αρχή, η Python ήταν γλώσσα σεναρίων και χρησιμοποιούταν στο λειτουργικό σύστημα Amoeba. Το 2000 κυκλοφόρησε η Python 2.0, το 2008 ακολούθησε η έκδοση 3.0 της οποίας πολλά από τα καινούρια χαρακτηριστικά μεταφέρθηκαν στις εκδόσεις 2.6 και 2.7 οι οποίες είναι προς τα πίσω συμβατές. Ο συγκεκριμένος κώδικας αναπτύχθηκε σε Python 2.7 και σε λειτουργικό σύστημα Ubuntu 16.04.

Για την εύρεση των αποστάσεων των σημείων με βάση τους πραγματικούς δρόμους χρησιμοποιήθηκε η υπηρεσία Google Maps, η οποία είναι μία υπηρεσία χαρτογράφησης στο διαδίκτυο. Για βοηθητικούς λόγους, όπως για παράδειγμα για την παραγωγή κάποιων γραφημάτων χρησιμοποιήθηκε και το Microsoft Excel.

3.4 ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΜΕ ΒΑΣΗ ΤΗ ΛΕΙΤΟΥΡΓΙΑ ΤΩΝ ΑΠΟΙΚΙΩΝ ΤΩΝ ΜΥΡΜΗΓΚΙΩΝ ΓΙΑ ΤΗ ΔΡΟΜΟΛΟΓΗΣΗ ΟΧΗΜΑΤΩΝ

3.4.1 Εισαγωγή

Η βελτιστοποίηση με βάση τη λειτουργία των αποικιών των μυρμηγκιών είναι μέρος του ευρύτερου πεδίου της νοημοσύνης σμήνους με την οποία οι επιστήμονες μελετούν τη συμπεριφορά μοτίβων μελισσών, τερμιτών, μυρμηγκιών και άλλων κοινωνικών εντόμων με σκοπό να προσομοιώσουν διαδικασίες.

Η ικανότητα σμηνών εντόμων να επιβιώνουν στην φύση και να επιλύουν περίπλοκα προβλήματα σχετικά με την επιβίωσή τους ώθησε τους επιστήμονες να αναπτύξουν υπολογιστικούς αλγορίθμους με στόχο την επίλυση περίπλοκων προβλημάτων. Αλγόριθμοι τεχνητής νοημοσύνης όπως ο αλγόριθμος βελτιστοποίησης με βάση τη λειτουργία των αποικιών των μυρμηγκιών εφαρμόζονται σε σύνθετα υπολογιστικά προβλήματα βελτιστοποίησης και χρησιμοποιούνται για να δημιουργήσουν μεθόδους αυτό-οργάνωσης σε τέτοια προβλήματα.

Ο συγκεκριμένος αλγόριθμος είναι μία μεταευρετική τεχνική η οποία χρησιμοποιεί τεχνητά μυρμήγκια για να βρει λύσεις σε συνδυαστικά υπολογιστικά προβλήματα. Βασίζεται στη συμπεριφορά αληθινών μυρμηγκιών και διαθέτει ενισχυμένες ικανότητες όπως η απομνημόνευση προηγούμενων δράσεων και η γνώση των αποστάσεων προς άλλες τοποθεσίες. Στη φύση, ένα μυρμήγκι από μόνο του δεν είναι ικανό να επικοινωνήσει και να βρει αποτελεσματικά τροφή, αλλά σαν σύνολο τα μυρμήγκια έχουν την ικανότητα να επιλύουν περίπλοκα προβλήματα και να βρίσκουν τροφή την οποία και συλλέγουν για την αποικία τους. Τα μυρμήγκια επικοινωνούν μεταξύ τους μέσω μίας χημικής ένωσης η οποία ονομάζεται φερομόνη. Καθώς ένα μυρμήγκι ταξιδεύει, αποθέτει συνεχώς μία ποσότητα φερομόνης την οποία μπορούν να ακολουθήσουν τα υπόλοιπα μυρμήγκια. Κάθε μυρμήγκι μετακινείται με έναν σχεδόν τυχαίο τρόπο αλλά όταν βρει κάποια ποσότητα φερομόνης πρέπει να αποφασίσει αν θα την ακολουθήσει. Αν ακολουθήσει το μονοπάτι, η φερομόνη αυτού του μυρμηγκιού ενισχύει την ήδη υπάρχουσα ποσότητα φερομόνης και η αυξημένη αυτή ποσότητα ενισχύει την πιθανότητα το επόμενο μυρμήγκι να ακολουθήσει αυτό το μονοπάτι. Επομένως, όσο περισσότερα μυρμήγκια χρησιμοποιούν ένα μονοπάτι τόσο πιο ελκυστικό γίνεται αυτό για τα υπόλοιπα. Αυτό επηρεάζει άμεσα την πιθανότητα επιλογής για το επόμενο που θα φύγει από την φωλιά. Με το πέρασμα του χρόνου, καθώς περισσότερα μυρμήγκια είναι σε θέση να διανύσουν τη συντομότερη διαδρομή, η φερομόνη συσσωρεύεται γρηγορότερα στις συντομότερες διαδρομές και στις πιο μεγάλες γίνεται πιο ανίσχυρη. Επίσης, η εξάτμιση της φερομόνης καθιστά λιγότερο ελκυστικές τις διαδρομές που είναι πιο δύσκολο να ανιχνευθούν και επομένως μειώνει τη χρήση τους. Από την άλλη, η συνεχής τυχαία επιλογή διαδρομών από ξεχωριστά μυρμήγκια βοηθά την αποικία να ανακαλύψει εναλλακτικές διαδρομές και διασφαλίζει την επιτυχή πλοήγηση γύρω από εμπόδια που μπορεί να διακόπτουν μία διαδρομή. Η επιλογή μονοπατιών από τα μυρμήγκια είναι μία ψευδοτυχαία αναλογική διαδικασία και είναι ένα στοιχείο “κλειδί” για την προσομοίωση του αλγορίθμου αυτής της βελτιστοποίησης. Η χρήση των αποικιών των μυρμηγκιών εφαρμόστηκε αρχικά στο πρόβλημα του περιοδεύοντος πωλητή και στο πρόβλημα τετραγωνικής αντιστοίχισης (Quadratic Assignment Problem) και από τότε έχει εφαρμοστεί και σε άλλα προβλήματα όπως τα προβλήματα χωροταξικού σχεδιασμού και τα προβλήματα δρομολόγησης οχημάτων. .

3.4.2 Αρχική λύση

Στην αρχή της διαδικασίας, κατασκευάζεται μία αρχική λύση με τη χρήση της ευρετικής εγγύτερου γείτονα προσθέτοντας επαναληπτικά κόμβους σε μία διαδρομή οχήματος. Σε αυτή τη λύση χρησιμοποιείται ένα όχημα θεωρητικά άπειρου εύρους και άπειρης χωρητικότητας ώστε να επισκεφτεί από μία φορά κάθε κόμβο του δικτύου με στόχο την ενίσχυση της φερομόνης των συντομότερων διαδρομών. Το όχημα ξεκινάει από το αμαξοστάσιο και επισκέπτεται τον κοντινότερο πελάτη με βάση την απόσταση και συνεχίζει έως ότου επισκεφτεί όλους τους κόμβους και επιστρέψει εν τέλει στο αμαξοστάσιο.

3.4.3 Κατασκευή διαδρομών

Κατά τη βελτιστοποίηση με βάση τη λειτουργία των αποικιών των μυρμηγκιών, ένα μυρμήγκι αντιπροσωπεύει ένα όχημα και η διαδρομή του κατασκευάζεται σταδιακά διαλέγοντας κόμβους μέχρι να εξυπηρετηθούν όλοι οι επιβάτες.

Βήμα 1: Επιλογή κόμβου

Αρχικά, κάθε μυρμήγκι ξεκινάει από την αφετηρία. Το μυρμήγκι διαλέγει τον επόμενο κόμβο που θα επισκεφτεί από μία λίστα εφικτών τοποθεσιών και η διαθέσιμη χωρητικότητα του οχήματος ενημερώνεται. Στην παρούσα εργασία, προτού επιλεχθεί ένας επιβάτης, ελέγχεται αν το όχημα μπορεί να τον εξυπηρετήσει πηγαίνοντάς τον στον προορισμό του και έπειτα να επιστρέψει στο αμαξοστάσιο. Εάν έχει αυτή τη δυνατότητα, στη διαδρομή προστίθεται ο κόμβος στον οποίο βρίσκεται ο πελάτης και ως επόμενος κόμβος ο προορισμός του. Για να επιλέξει τον επόμενο πελάτη χρησιμοποιεί την εξίσωση πιθανότητας: $\mathrm{Pij} = \frac{(t_{ij})\left(n_{ij}^{\beta}\right)}{\sum_{u \notin M_k}(t_{iu})\left(n_{iu}^{\beta}\right)}$, (3.8) , όπου t_{iu} είναι η ποσότητα φερομόνης για τη διαδρομή από το i και όλες τις πιθανές τοποθεσίες u. Η τιμή n_{ij} ονομάζεται εγγύτητα και ορίζεται ως το αντίστροφο της απόστασης ανάμεσα σε δύο τοποθεσίες και η παράμετρος β καθορίζει τη σημασία της απόστασης σε σχέση με τη ποσότητα φερομόνης στον αλγόριθμο επιλογής (β>0). Οι τοποθεσίες που ένα μυρμήγκι έχει ήδη επισκεφτεί αποθηκεύεται στη μνήμη των μυρμηγκιών M_K τα οποία δεν λαμβάνονται υπόψη στην επιλογή. Η τιμή q είναι μία τυχαία τιμή εντός του ορίου [0,1] και η τιμή q_0 είναι μία παράμετρος. Όταν πραγματοποιείται κάθε επιλογή απόφασης το μυρμήγκι διαλέγει τον κόμβο με την μεγαλύτερη τιμή από την εξίσωση (3.8) εκτός της περίπτωσης όπου το q είναι μεγαλύτερο από το q_0. Σε αυτή την περίπτωση το μυρμήγκι διαλέγει τυχαία έναν πελάτη να είναι ο επόμενος εάν μπορεί να εξυπηρετήσει τη ζήτησή του και να γυρίσει στο αμαξοστάσιο χωρίς να έχει εφαρμοστεί ο περιορισμός καυσίμου. Η μεθοδολογία κατασκευάζει μία πλήρη διαδρομή για το πρώτο μυρμήγκι πριν το δεύτερο ξεκινήσει την δική του διαδρομή.

Βήμα 2: Κατασκευή διαδρομών για κάθε όχημα

Η προηγούμενη διαδικασία συνεχίζεται μέχρι ένας προκαθορισμένος αριθμός μυρμηγκιών m κατασκευάσουν το καθένα από μία πλήρη και εφικτή διαδρομή. Με τη συγκεκριμένη μέθοδο βελτιστοποίησης κάθε μυρμήγκι κατασκευάζει μία πλήρη διαδρομή με την οποία εξυπηρετούνται όλοι οι πελάτες. Σε αυτή την εργασία επειδή όπως ανέφερα λαμβάνεται υπόψη ο παράγοντας του καυσίμου, κάθε όχημα μπορεί να εξυπηρετεί πελάτες έως ότου καλύψει ένα συγκεκριμένο μήκος διαδρομής. Όταν συμβεί αυτό, το όχημα επιστρέφει στο αμαξοστάσιο και ξεκινάει το επόμενο.

Βήμα 3: Ολοκλήρωση κατασκευής διαδρομών

Η διαδικασία ολοκληρώνεται όταν εξυπηρετηθούν όλοι οι πελάτες ή όταν δεν υπάρχουν επιπλέον διαθέσιμα οχήματα. Επειδή η παρούσα εργασία λαμβάνει δεδομένη ζήτηση και αριθμό οχημάτων, παρατηρήθηκε ότι ολοκληρώνεται αφού εξυπηρετηθεί όλη η ζήτηση.

Βήμα 4: Εκκίνηση επαναλήψεων

Όταν δημιουργηθεί ένα πλήρες σύνολο διαδρομών που βελτιστοποιεί την αντικειμενική συνάρτηση, το πείραμα εκτελείται από την αρχή για τα αρχικά δεδομένα και για προκαθορισμένο αριθμό επαναλήψεων. Το μόνο δεδομένο που δεν λαμβάνει την αρχική του τιμή, είναι οι τιμές τις φερομόνης κάθε ζεύγους, καθώς η γνώση των προηγούμενων τιμών της λειτουργεί προς όφελος της βελτιστοποίησης.

Βήμα 5: Τερματισμός διαδικασίας

Η διαδικασία τερματίζει όταν εκτελεστούν όλα τα πειράματα για τον αριθμό επαναλήψεων που έχει δώσει ο χρήστης.

3.4.4 Ενημέρωση της έντασης του ίχνους της φερομόνης

Με στόχο τη βελτίωση των μελλοντικών επιλογών, τα ίχνη της φερομόνης στα μονοπάτια πρέπει να ενημερώνονται ώστε να αντικατοπτρίζουν την απόδοση των μυρμηγκιών και την ποιότητα των λύσεων που έχουν βρεθεί. Αυτή η ενημέρωση είναι το στοιχείο “κλειδί” στην προσαρμοστική διαδικασία μάθησης της βελτιστοποίησης και βοηθάει στη διασφάλιση της βελτίωσης των λύσεων. Αυτή η ενημέρωση περιλαμβάνει τοπικές ενημερώσεις των μονοπατιών μετά από μεμονωμένες λύσεις που παράγονται αλλά και γενικές ενημερώσεις του καλύτερου συνόλου διαδρομών όταν ένας προκαθορισμένος αριθμός m λύσεων έχει επιτευχθεί.

3.4.4.1 Τοπική ενημέρωση

Αρχικά, η τοπική ενημέρωση διεξάγεται μειώνοντας τη ποσότητα φερομόνης σε όλους τους κόμβους που έχουν δεχτεί επίσκεψη με στόχο την προσομοίωση της φυσικής εξάτμισης της φερομόνης και για να διασφαλίσει ότι κανένας σύνδεσμος δεν θα συγκεντρώσει όλη τη φερομόνη. Αυτό πραγματοποιείται με την παρακάτω εξίσωσης τοπικής ενημέρωσης της φερομόνης:

$t_{ij}=(1-a)t_{ij} + a*t_0$ (3.9)

Όπου α είναι η παράμετρος που ελέγχει την ταχύτητα εξάτμισης και t_0 είναι η αρχική τιμή της έντασης της φερομόνης που έχει αποδοθεί σε όλους τους συνδέσμους του γραφήματος G. Στη συγκεκριμένη εργασία η ποσότητα αυτή αρχικοποιείται για κάθε σύνδεσμο ως το αντίστροφο της απόστασης μεταξύ των κόμβων του.

3.4.4.2 Γενική ενημέρωση

Αφού ένας προκαθορισμένος αριθμός μυρμηγκιών m δημιουργήσει μία πλήρη λύση, πραγματοποιείται η γενική ενημέρωση των ιχνών φερομόνης. Αυξάνονται, δηλαδή, οι τιμές της φερομόνης σε όλους τους συνδέσμους που συμπεριλαμβάνονται στην μικρότερη διαδρομή ενός από τα συνολικά m μυρμήγκια, δηλαδή από ένα όχημα από όσα χρησιμοποιήθηκαν και το οποίο πραγματοποίησε τη συντομότερη διαδρομή. Η γενική ενημέρωση πραγματοποιείται σύμφωνα με την παρακάτω σχέση:

$t_{ij}-(1-a)*t_{ij} +a/L$ (3.10),

όπου L η συντομότερη διαδρομή.

Αυτή η ενημέρωση ενισχύει την χρήση μικρότερων διαδρομών και αυξάνει την πιθανότητα στις μελλοντικές διαδρομές να συμπεριλαμβάνονται σύνδεσμοι που υπάρχουν και στις καλύτερες λύσεις. Η διαδικασία αυτή επαναλαμβάνεται για έναν προκαθορισμένο αριθμό επαναλήψεων και η καλύτερη λύση από όλες τις επαναλήψεις παρουσιάζεται ως αποτέλεσμα του μοντέλου και θα έπρεπε να αναπαριστά μία καλή προσέγγιση της βέλτιστης λύσης του προβλήματος.

Σχετικά με την τιμή του L, κάποιες έρευνες όπως αυτή των Bell και McMullen (2004) την λαμβάνουν όπως αναφέρθηκε προηγουμένως, δηλαδή ως το μήκος της συντομότερης διαδρομής από το σύνολο των διαδρομών της λύσης. Άλλες όπως αυτή των Gambardella, et al. (1999), θεωρούν ότι το L ισούται με το συνολικό μήκος όλων των διαδρομών που προκύπτουν σε κάθε λύση. Μετά από πειράματα και με τους δύο τρόπους, η πρώτη παρατηρήθηκε ότι έδινε λίγο καλύτερες τιμές της αντικειμενικής συνάρτησης και έτσι προτιμήθηκε έναντι της δεύτερης.

3.4.5 Κοινή χρήση οχημάτων στον αλγόριθμο.

Με στόχο την μείωση της συνολικής διανυόμενης απόστασης από τα οχήματα, αφού ολοκληρωθεί η επαναληπτική διαδικασία, ελέγχεται εάν υπάρχουν διαδρομές οι οποίες εξυπηρετούν αποκλειστικά ένα ζεύγος ζήτησης και εάν αυτά τα ζεύγη μπορούν να εξυπηρετηθούν από προηγούμενα οχήματα. Σε περίπτωση που αυτές τα ζεύγη προέλευσης-προορισμού μπορούν να εξυπηρετηθούν από προηγούμενες διαδρομές ελέγχοντας τα φορτία των οχημάτων στις αντίστοιχες στιγμές, οι περιττές διαδρομές διαγράφονται από το αποτέλεσμα και οι αντίστοιχοι πελάτες εξυπηρετούνται από τα προηγούμενα οχήματα.

3.4.6 Μοντελοποίηση αλγορίθμου

Αρχικοποίηση των τιμών της φερομόνης και της ποσότητας n.
Εύρεση διαδρομής για ένα υποθετικό όχημα άπειρου καυσίμου με χρήση της αναζήτησης του εγγύτερου γείτονα.
Γενική ενημέρωση της φερομόνης στα ζεύγη που προέκυψαν σε αυτή τη διαδρομή.
Για συγκεκριμένο αριθμό επαναλήψεων:
    Για διαθέσιμο στόλο οχημάτων:
        Εάν έχει εξυπηρετηθεί όλη η ζήτηση:
            Τερματισμός επανάληψης
        Αλλιώς
            Εύρεση υποψήφιων κόμβων
            Επιλογή επόμενου κόμβου
            Τοπική ενημέρωση ίχνους φερομόνης
            Επιλογή προορισμού της ζήτησης από τον προηγούμενο κόμβο.
            Τοπική ενημέρωση ίχνους φερομόνης
            Ενώ χιλιόμετρα ≤ μέγιστη απόσταση:
                Εάν υπάρχουν υποψήφιοι κόμβοι και δεν έχει εξυπηρετηθεί όλη η ζήτηση:
                    Εύρεση υποψήφιων κόμβων
                    Επιλογή επόμενου κόμβου
                    Τοπική ενημέρωση ίχνους φερομόνης
                    Επιλογή προορισμού της ζήτησης από τον προηγούμενο κόμβο

Τοπική ενημέρωση ίχνους φερομόνης

Αλλιώς

Εκκίνηση νέου οχήματος

Εάν υπάρχει διαδρομή με συνολικά 4 κόμβους και η ζήτησή που εξυπηρέτησε μπορεί να εξυπηρετηθεί από προηγούμενη διαδρομή:

Διαγραφή αυτής της διαδρομής και εισαγωγή των πελατών που εξυπηρέτησε σε προηγούμενη διαδρομή.

Εύρεση μικρότερης υποδιαδρομής από το σύνολο των διαδρομών της λύσης:

Γενική ενημέρωση του ίχνους της φερομόνης στη συγκεκριμένη υποδιαδρομή

Εάν αυτή η διαδρομή είναι καλύτερη από κάθε άλλη μέχρι στιγμής:

Καλύτερη διαδρομή γίνεται η συγκεκριμένη διαδρομή

3.5 ΠΕΡΙΓΡΑΦΗ ΑΛΓΟΡΙΘΜΟΥ

3.5.1 Αναπαράσταση της λύσης

Τα σημεία που επιλέχθηκαν ως κόμβοι του συγκεκριμένου δικτύου έχουν αντιστοιχιστεί σε αριθμούς από το 0 ως το 30. Πιο συγκεκριμένα, το 0 συμβολίζει το αμαξοστάσιο από όπου φεύγουν τα οχήματα και πρέπει να επιστρέψουν σε αυτό πριν τη λήξη της βάρδιας και δεν υπάρχει ζήτηση από και προς αυτό το σημείο. Επομένως, κάθε διαδρομή μίας λύσης πρέπει να ξεκινάει και να τελειώνει στο 0. Έτσι, έστω ότι έχουμε τις ζητήσεις 1-19, 2-6, 3-16, 4-16, 5-7 , μία λύση θα μπορούσε να εκφραστεί ως εξής:

[0,3,4,16,0] Διαδρομή οχήματος Α

[0,1,19,2,6,0] Διαδρομή οχήματος Β

[0,5,7,0] Διαδρομή οχήματος Γ

3.5.2 Επιλογή Ζήτησης και Πληθυσμού πελατών και οχημάτων

Για την εκτέλεση του αλγορίθμου στην αρχική του μορφή έχουμε θεωρήσει ότι ζήτηση υπάρχει ακριβώς μία φορά από κάθε κόμβο του δικτύου με τον προορισμό της κάθε ζήτησης να επιλέγεται τυχαία κάθε φορά. Επίσης, έχουμε ορίσει ότι κάθε ζήτηση αντιστοιχεί σε έναν έως τέσσερις πελάτες. Ακόμη, δεν λαμβάνονται υπόψη χρονικά παράθυρα (time windows) μέσα στα οποία οι πελάτες απαιτούν να εξυπηρετηθούν. Ωστόσο, η περιορισμένη εμβέλεια των οχημάτων αλλά και ο περιορισμός της διάρκειας διαδρομών δεν επιτρέπουν μεγάλες διαδρομές, ενώ θεωρείται ότι οι επιβάτες είναι εργαζόμενοι που μετακινούνται σε συγκεκριμένες περιόδους της ημέρας.

Σχετικά με τον αριθμό των οχημάτων, θεωρούμε πως ο διαθέσιμος στόλος είναι ίσος με 30. Σε περιπτώσεις που ο αλγόριθμος χρειαστεί να εξυπηρετήσει μεγαλύτερη ζήτηση, ο αριθμός των απαιτούμενων οχημάτων θα είναι μεγαλύτερος. Τα οχήματα όπως αναφέραμε είναι αυτόνομα αυτοκίνητα και είναι πανομοιότυπα μεταξύ τους. Το μέγιστο εύρος τους θεωρήθηκε 75 χιλιόμετρα έως ότου χρειαστεί να επιστρέψουν στο αμαξοστάσιο για επαναφόρτιση. Ωστόσο, καθώς η διάρκεια της διαδρομής επηρεάζει το επίπεδο εξυπηρέτησης επιβατών, θεωρήθηκε μέγιστη απόσταση για κάθε διαδρομή ίση με 29 χιλιόμετρα. Η χωρητικότητά τους ορίστηκε στους 4 επιβάτες και, επίσης, θεωρήθηκε σταθερή ταχύτητα 50 χιλιόμετρα την ώρα. Επίσης, επειδή ο χρησιμοποιούμενος στόλος είναι πολύ μικρός σε σχέση με τα υπόλοιπα οχήματα του δικτύου δεν λήφθηκε υπόψη τυχόν κυκλοφοριακή συμφόρηση που μπορεί να προκληθεί μεταξύ τους. Τέλος, τα οχήματα ξεκινούν ταυτόχρονα από το αμαξοστάσιο.

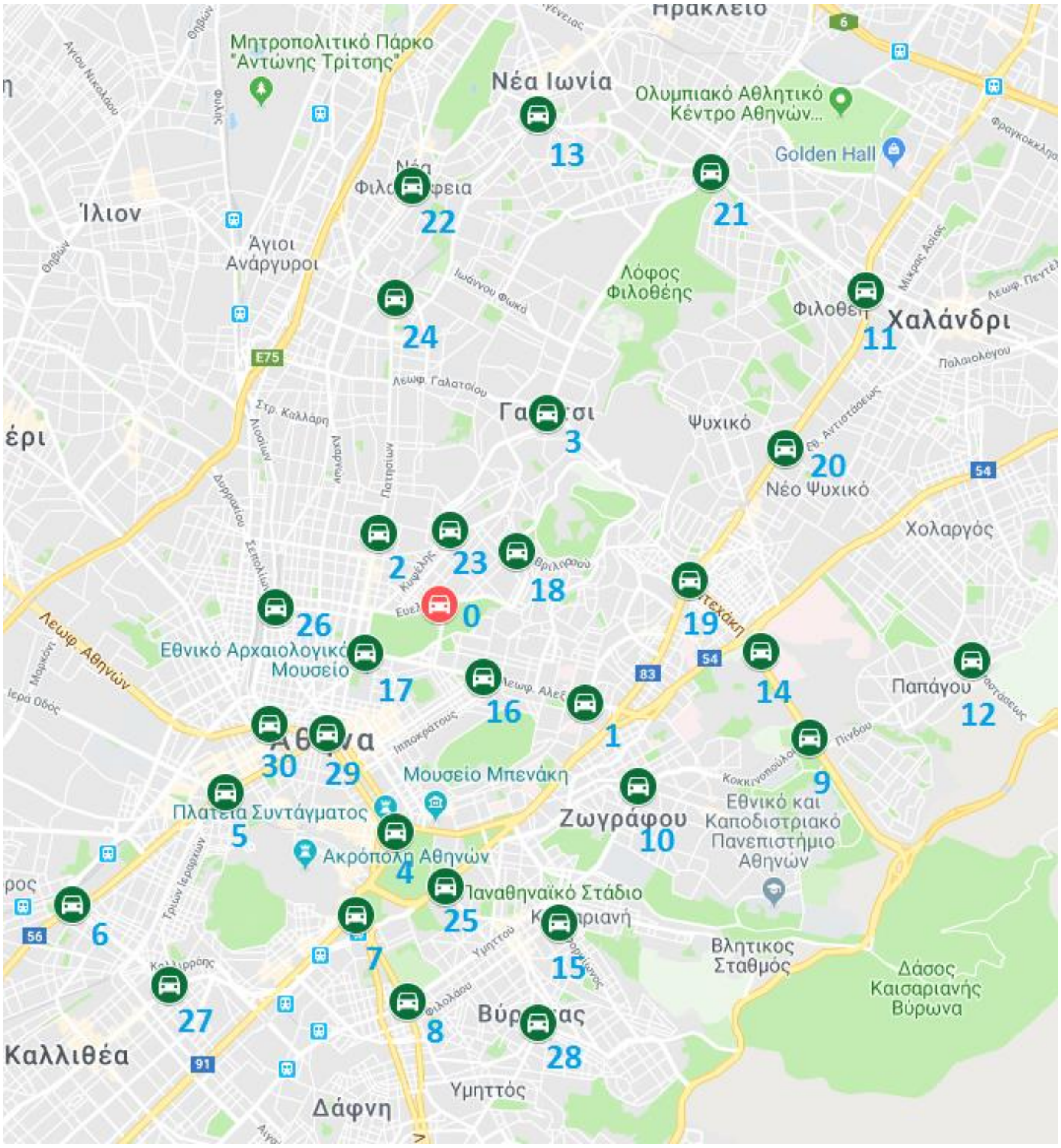
ΚΕΦΑΛΑΙΟ 4: ΑΠΟΤΕΛΕΣΜΑΤΑ

4.1 ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό περιγράφονται τα αποτελέσματα από την εφαρμογή της βελτιστοποίησης με βάση τη λειτουργία των αποικιών των μυρμηγκιών και παρουσιάζονται αναλυτικά και σχηματικά οι βέλτιστες διαδρομές που προέκυψαν από αυτή. Επίσης, πραγματοποιείται μία ανάλυση ευαισθησίας ως προς κάποιες παραμέτρους του προβλήματος.

4.2 ΔΕΔΟΜΕΝΑ ΠΕΡΙΟΧΗΣ ΜΕΛΕΤΗΣ

Ο αλγόριθμος επιλέχθηκε να εφαρμοστεί στο κέντρο και βορειοδυτικά της Αθήνας σε περιοχές όπου παρουσιάζεται έντονη κυκλοφοριακή συμφόρηση και έτσι αναμένεται η ζήτηση να είναι μεγάλη. Στο σχήμα 4.1 φαίνονται τα σημεία κατανεμημένα στον χώρο, όπου με πράσινο συμβολίζονται οι στάσεις και με κόκκινο το αμαξοστάσιο:



Σχήμα 4.1: Περιοχή Μελέτης

Στον πίνακα 4.1 φαίνονται οι συντεταγμένες των σημείων που χρησιμοποιήθηκαν καθώς και η αντιστοίχισή τους με τους αριθμούς που συμβολίζουν το κάθε ένα από αυτά και στον πίνακες 4.2., 4.3 και 4.4 οι αποστάσεις μεταξύ τους βασισμένες στο google maps.

Πίνακας 4.1: Συντεταγμένες σημείων

Name	Latitude	Longitude	ID
DEPOT EVELPIDON	37.995921	23.740837	0
Ampelokipoi	37.9871	23.7572	1
Patision	38.0023	23.7339	2
Galatsiou-Veikou	38.013	23.753	3
Sintagma	37.97564	23.73574	4
Peiraios-Iera Odos	37.97907	23.71653	5
Konstantinoupoleos-Peiraios	37.96905	23.69921	6
Amalias-Sugrou	37.9682	23.7313	7
Ilioupoleos-Imitou	37.9604	23.7371	8
Goudi	37.98394	23.782697	9
Papagou-Oulof Palme	37.97964	23.76334	10
Kapodistriou Xalandri	38.0239	23.789	11
Antistaseos- Papagou	37.990872	23.801034	12
Peukakia Nea Ionia	38.039633	23.751983	13
Katexaki	37.99164	23.777108	14
Kaisariani	37.967535	23.754323	15
Trikoupi – Alexandras	37.989459	23.745686	16
Pedion toy Areos	37.991602	23.732325	17
Nea Kipseli	38.000611	23.749553	18
Kifisias-Katexaki	37.998092	23.769137	19
Psixiko	38.009866	23.779918	20
Veikou- Antistaseos	38.03447	23.771519	21
Filadelfeia	38.033262	23.737615	22
Kipseli	38.002482	23.742027	23
Ano Patisia	38.023246	23.73582	24
Panathinaiko Stadio	37.970722	23.741492	25
Liosion	37.995611	23.72229	26
Thiseos	37.96196	23.710261	27
Vironas	37.958561	23.751901	28
Omonoia	37.98444	23.72812	29
Metaksourgeio	37.98522	23.721555	30

Πίνακας 4.2: Αποστάσεις μεταξύ των σημείων (1)

	0	1	2	3	4	5	6	7	8	9	10
0	0	2600	1500	3400	3500	3500	5600	4100	5900	6400	3800
1	2600	0	4000	6100	3300	5200	7800	4200	4200	3700	1200
2	1500	4000	0	3200	3400	3500	5500	4100	5900	7900	4800
3	3400	6100	3200	0	5800	5900	7900	6500	8300	5800	6600
4	3500	3300	3400	5800	0	2500	5100	1200	3000	5300	3600
5	3500	5200	3500	5900	2500	0	2300	3200	5000	7800	5400
6	5600	7800	5500	7900	5100	2300	0	5700	4400	11800	7500
7	4100	4200	4100	6500	1200	3200	5700	0	2000	6700	4800
8	5900	4200	5900	8300	3000	5000	4400	2000	0	6500	4600
9	6400	3700	7900	5800	5300	7800	11800	6700	6500	0	1900
10	3800	1200	4800	6600	3600	5400	7500	4800	4600	1900	0
11	7300	6000	8000	5300	11200	10400	12600	10000	10800	7200	8500
12	7700	5300	9200	6800	10200	9800	11800	9100	8900	3400	6700
13	7500	9600	5300	5500	8700	12600	16300	13300	10600	10500	11500
14	5000	3300	6600	4400	5700	7700	9500	6900	6700	2100	4200
15	5300	3000	5300	7600	2200	4700	5900	3500	3100	3700	1800
16	1700	1600	2500	4100	2300	3100	5600	4200	4300	3800	3400
17	1300	3000	1300	4300	2200	2500	4900	4000	4200	5300	5100
18	1000	3500	2400	2300	3900	4600	7100	5800	5900	5100	7200
19	3300	1900	5400	2700	8300	6000	8500	5900	5700	3100	4400
20	5000	3500	6800	3700	6800	7600	10100	7500	7200	4700	6000
21	5400	7500	6100	3400	9500	9800	12200	11800	11600	10500	9400
22	5400	8300	3800	4300	7200	9700	12100	14400	9100	9500	10000
23	1000	3700	1400	1800	4100	4300	6800	5900	6000	6100	5400
24	3700	6900	2700	2900	6400	9700	12100	8200	8300	8100	8800
25	4300	3500	4300	6700	1300	3700	4500	1900	1700	4700	2700
26	2600	4200	2500	3900	3100	2500	5000	4900	5100	6500	6000
27	6800	8800	6800	9200	4700	3300	1400	2800	3600	8500	6800
28	6200	4100	6100	11900	3100	5600	5700	3400	2200	5000	3000
29	2300	3900	2200	4600	1300	1500	3900	3100	3200	5700	4100
30	2700	5000	2600	5000	2000	1200	3700	3900	4000	6500	4900

Πίνακας 4.3: Αποστάσεις μεταξύ των σημείων (2)

11	12	13	14	15	16	17	18	19	20
7300	7700	7500	5000	5300	1700	1300	1000	3300	5000
6000	5300	9600	3300	3000	1600	3000	3500	1900	3500
8000	9200	5300	6600	5300	2500	1300	2400	5400	6800
5300	6800	5500	4400	7600	4100	4300	2300	2700	3700
11200	10200	8700	5700	2200	2300	2200	3900	8300	6800
10400	9800	12600	7700	4700	3100	2500	4600	6000	7600
12600	11800	16300	9500	5900	5600	4900	7100	8500	10100
10000	9100	13300	6900	3500	4200	4000	5800	5900	7500
10800	8900	10600	6700	3100	4300	4200	5900	5700	7200
7200	3400	10500	2100	3700	3800	5300	5100	3100	4700
8500	6700	11500	4200	1800	3400	5100	7200	4400	6000
0	6400	5100	5100	9100	6300	7700	5400	3400	3200
6400	0	10000	2900	8800	6100	7500	5700	4200	3500
5100	10000	0	9000	11800	10200	6100	7200	7300	7100
5100	2900	9000	0	4600	5900	4600	3000	1000	2600
9100	8800	11800	4600	0	3700	5100	9900	7900	5700
6300	6100	10200	3900	3700	0	1400	2100	3500	4500
7700	7500	6100	4600	5100	1400	0	2100	4400	6100
5400	5700	7200	3000	9900	2100	2100	0	2100	3700
3400	4200	7300	1000	7900	3500	4400	2100	0	1600
3200	3500	7100	2600	5700	4500	6100	3700	1600	0
3600	9400	2000	8500	10100	6500	6400	5300	6000	4300
6900	11100	1800	7400	13400	6300	5100	6100	6500	7600
6500	6800	5000	4100	5800	2100	1800	1100	3100	6700
7900	8700	2800	6000	12900	5400	3800	4700	5000	8200
10100	8900	9100	5100	1500	3500	3200	5700	4700	9700
8900	8500	7300	5800	6300	2600	1400	3900	5800	9200
11100	10500	14400	8000	5300	6900	5700	8400	7700	11400
12600	9100	12600	9300	1500	5300	5100	7500	5300	12900
8600	8200	7000	5500	3800	2300	1100	3600	5500	8900
9300	8600	9400	6900	5000	2800	1600	4100	6000	12600

Πίνακας 4.4: Αποστάσεις μεταξύ των σημείων (3)

21	22	23	24	25	26	27	28	29	30
5400	5400	1000	3700	4300	2600	6800	6200	2300	2700
7500	8300	3700	6900	3500	4200	8800	4100	3900	5000
6100	3800	1400	2700	4300	2500	6800	6100	2200	2600
3400	4300	1800	2900	6700	4900	9200	11900	4600	5000
9500	7200	4100	6400	1300	3100	4700	3100	1300	2000
9800	9700	4300	9700	3700	2500	3300	5600	1500	1200
12200	12100	6800	12100	4500	5000	1400	5700	3900	3700
11800	14400	5900	8200	1900	4900	2800	3400	3100	3900
11600	9100	6000	8300	1700	5100	3600	2200	3200	4000
10500	9500	6100	8100	4700	6500	8500	5000	5700	6500
9400	10000	5400	8800	2700	6000	6800	3000	4100	4900
3600	6900	6500	7900	10100	8900	11100	12600	8600	9300
9400	11100	6800	8700	8900	8500	10500	9100	8200	8600
2000	1800	5000	2800	9100	7300	14400	12600	7000	9400
8500	7400	4100	6000	5100	5800	8000	9300	5500	6900
10100	13400	15800	12900	1500	6300	5300	1500	3800	5000
6500	6300	2100	5400	3500	2600	6900	5300	2300	2800
6400	5100	1800	3800	3200	1400	5700	5100	1100	1600
5300	6100	1100	4700	5700	3900	8400	7500	3600	4100
6000	6500	3100	5000	4700	5800	7700	5300	5500	6000
4300	7600	6700	8200	9700	9200	11400	12900	8900	12600
0	3800	4800	4400	12100	7900	13100	10600	7500	8100
3800	0	4300	1500	8100	6600	13200	13300	6000	8100
4800	4300	0	2900	5100	3300	7500	6800	2900	3400
4400	1500	2900	0	6600	4400	12100	8500	4600	5000
12100	8100	5100	6600	0	4500	3800	1800	2700	3500
7900	6600	3300	4400	4500	0	5900	5500	1700	1400
13100	13200	7500	12100	3800	5900	0	4800	4400	4100
10600	13300	6800	8500	1800	5500	4800	0	4600	6000
7500	6000	2900	4600	2700	1700	4400	4600	0	800
8100	8100	3400	5000	3500	1400	4100	6000	800	0

Η ζήτηση φαίνεται στον πίνακα 4.5. Συγκεκριμένα, φαίνεται ο κόμβος προέλευσης και ο κόμβος προορισμού μαζί με τον αριθμό των πελατών που αναμένουν την αντίστοιχη μετακίνηση:

Πίνακας 4.5: Ζήτηση

Από κόμβο	Σε κόμβο	Αριθμός Πελατών
1	14	2
2	12	4
3	5	1
4	26	4
5	1	4
6	16	1
7	27	2
8	3	2
9	17	1
10	19	2
11	2	4
12	22	1
13	14	3
14	22	1
15	5	3
16	8	4
17	4	1
18	16	1
19	28	1
20	18	3
21	24	1
22	12	1
23	4	2
24	9	4
25	21	2
26	22	2
27	13	3
28	3	1
29	9	3
30	7	1

4.3 ΒΑΘΜΟΝΟΜΗΣΗ ΑΛΓΟΡΙΘΜΟΥ

Η διαδικασία της βαθμονόμησης του αλγορίθμου σχετίζεται με την εύρεση του βέλτιστου συνδυασμού των παραμέτρων που λαμβάνει ως δεδομένα για την εκτέλεση της διαδικασίας της βελτιστοποίησης. Αυτοί οι παράμετροι είναι οι εξής:

- Παράμετρος εξάτμισης φερομόνης α : Αυτή η μεταβλητή χρησιμοποιείται κάθε φορά που ενημερώνεται το ίχνος της φερομόνης ενός ζεύγους κόμβων. Ο σκοπός της είναι η προσομοίωση της φυσικής διαδικασίας της εξάτμισης. Σύμφωνα με τη βιβλιογραφία, οι τιμές που μπορεί να λάβει είναι $0 \leq α \leq 1$. Οι Gilmour και Dras (2005) αναφέρουν ότι πειραματικά έχει βρεθεί η τιμή 0.1 ως η καλύτερη για την εκτέλεση της βελτιστοποίησης. Επειδή, όμως, η βέλτιστη τιμή αυτή της παραμέτρου μπορεί να διαφέρει για κάθε εργασία σε συνδυασμό με τους υπόλοιπους παράγοντες, θα εξεταστούν πειραματικά κάποιες τιμές της παραμέτρου σε συνδυασμό με την δεύτερο παράμετρο του προβλήματος, ώστε να χρησιμοποιηθούν αυτές που σε συνδυασμό δίνουν το καλύτερο αποτέλεσμα. Οι τρεις τιμές της μεταβλητής α που θα εξεταστούν είναι οι 0.1, 0.5 και 0.8.
- Παράμετρος σημασίας της απόστασης έναντι της φερομόνης β: Όπως αναφέρθηκε και πριν, η πιθανότητα κατανομής P_{ij}, ίσως η σημαντικότερη μεταβλητή της μεθόδου, υπολογίζεται μέσω του τύπου $P_{ij}= (τ_{ij})(n_{ij})^{β}/\sum_{u \notin M_k} (\tau_{ij})(n_{ij}^{\beta})$. Οι τιμές της, δηλαδή, εξαρτώνται από την τιμή που εκφράζει την ένταση του ίχνους της φερομόνης στον συγκεκριμένο σύνδεσμο, την εγγύτητα n_{ij} η οποία ορίζεται ως το αντίστροφο της απόσταση για ένα ζεύγος κόμβων αλλά και από την τιμή της παραμέτρου β η οποία καθορίζει τη συσχέτιση μεταξύ των ποσοτήτων n_{ij} και $τ_{ij}$. Οι τιμές της παραμέτρου β που θα εξεταστούν είναι οι τιμές 0.5, 1 και 2.

Ας σημειωθεί, εδώ, ότι για να γίνει η σύγκριση των αποτελεσμάτων κάθε συνδυασμού μεταβλητών, χρησιμοποιήθηκαν τα δεδομένα που αναφέρονται στο κεφάλαιο «4.2 Δεδομένα περιοχής μελέτης», η σύγκριση γίνεται με βάση την καλύτερη τιμής της αντικειμενικής συνάρτησης και πραγματοποιήθηκαν 30000 επαναλήψεις για κάθε συνδυασμό.

Πίνακας 4.6: Τιμές της αντικειμενικής συνάρτησης για διαφορετικούς συνδυασμούς των παραμέτρων α και β

α	β	L (m)
0.1	0.5	288000
	1	286300
	2	286300
0.5	0.5	285300
	1	286500
	2	285300
0.8	0.5	286800
	1	285300
	2	286500

Από τον πίνακα 4.6 παρατηρούμε ότι η καλύτερη τιμή της αντικειμενικής συνάρτησης και συγκεκριμένα η L=285300m εμφανίζεται στις εξής περιπτώσεις:

- α=0.5 και β=0.5
- α=0.5 και β=2
- α=0.8 και β=1.

Από τους τρεις αυτούς συνδυασμούς έπρεπε να επιλεγεί ένας και αποφασίστηκε να επιλεγεί ο α=0.5 και β=0.5, ο οποίος δίνει μεγαλύτερη έμφαση στις τιμές της φερομόνης σε σχέση με την απόσταση.

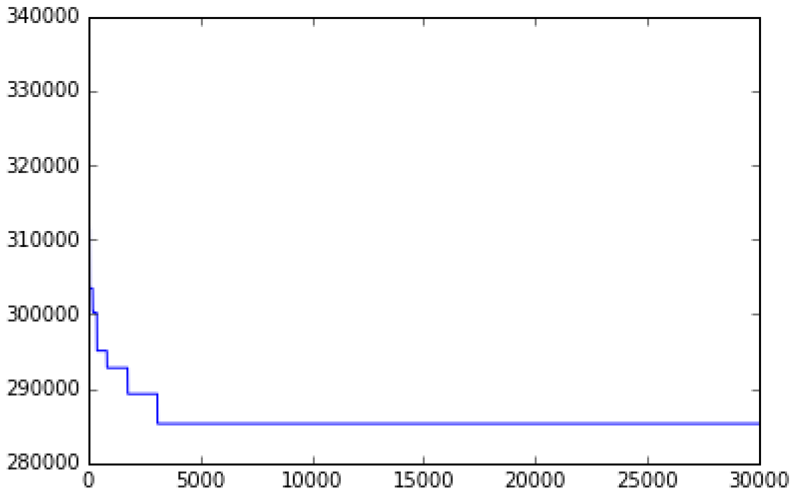
Έστω Z το συνολικό μήκος της αρχικής διαδρομής που προκύπτει από την αναζήτηση εγγύτερου γείτονα. Με βάση αυτό το μήκος και πριν την εκκίνηση του αλγορίθμου, οι φερομόνες των ζευγών που ανήκουν σε αυτή τη διαδρομή ενημερώνονται μέσω της γενικής ενημέρωσης της φερομόνης χρησιμοποιώντας ως είσοδο στη συνάρτηση το Z/100, το οποίο πειραματικά βρέθηκε ότι αποτελεί μία ικανοποιητική τιμή για την αύξηση των τιμών των ιχνών των ζευγών που ανήκουν σε αυτή τη διαδρομή και για το συγκεκριμένο μέγεθος δικτύου.

4.4 ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΛΓΟΡΙΘΜΟΥ

Με τη χρήση των τιμών α=0.5 και β=0.5 και με τα δεδομένα που έχουν αναφερθεί και προηγουμένως, ο αλγόριθμος εκτελέστηκε σε υπολογιστή με επεξεργαστή 3.0 GHz και μνήμη RAM 5 GB και η εκτέλεσή του διήρκησε 5 λεπτά και 51 δευτερόλεπτα για 30000 πειράματα με τα αποτελέσματα που φαίνονται στον πίνακα 4.7.

Πίνακας 4.7: Αποτελέσματα αλγορίθμου

Όχημα	Διαδρομή	Συνολικό κόστος διαδρομών (m)
1	[0,18,16,17,4,25,21,0]	285300
2	[0,23,4,7,27,6,16,1,14,20,18,0]	
3	[0,2,12,14,22,0]	
4	[0,5,1,10,19,11,2,0]	
5	[0,16,8,28,3,0]	
6	[0,22,12,9,17,0]	
7	[0,29,9,19,28,15,5,0]	
8	[0,26,22,24,9,0]	
9	[0,3,5,30,7,4,26,0]	
10	[0,21,24,13,14,0]	
11	[0,27,13,0]	



Σχήμα 4.2: Εξέλιξη της αναζήτησης

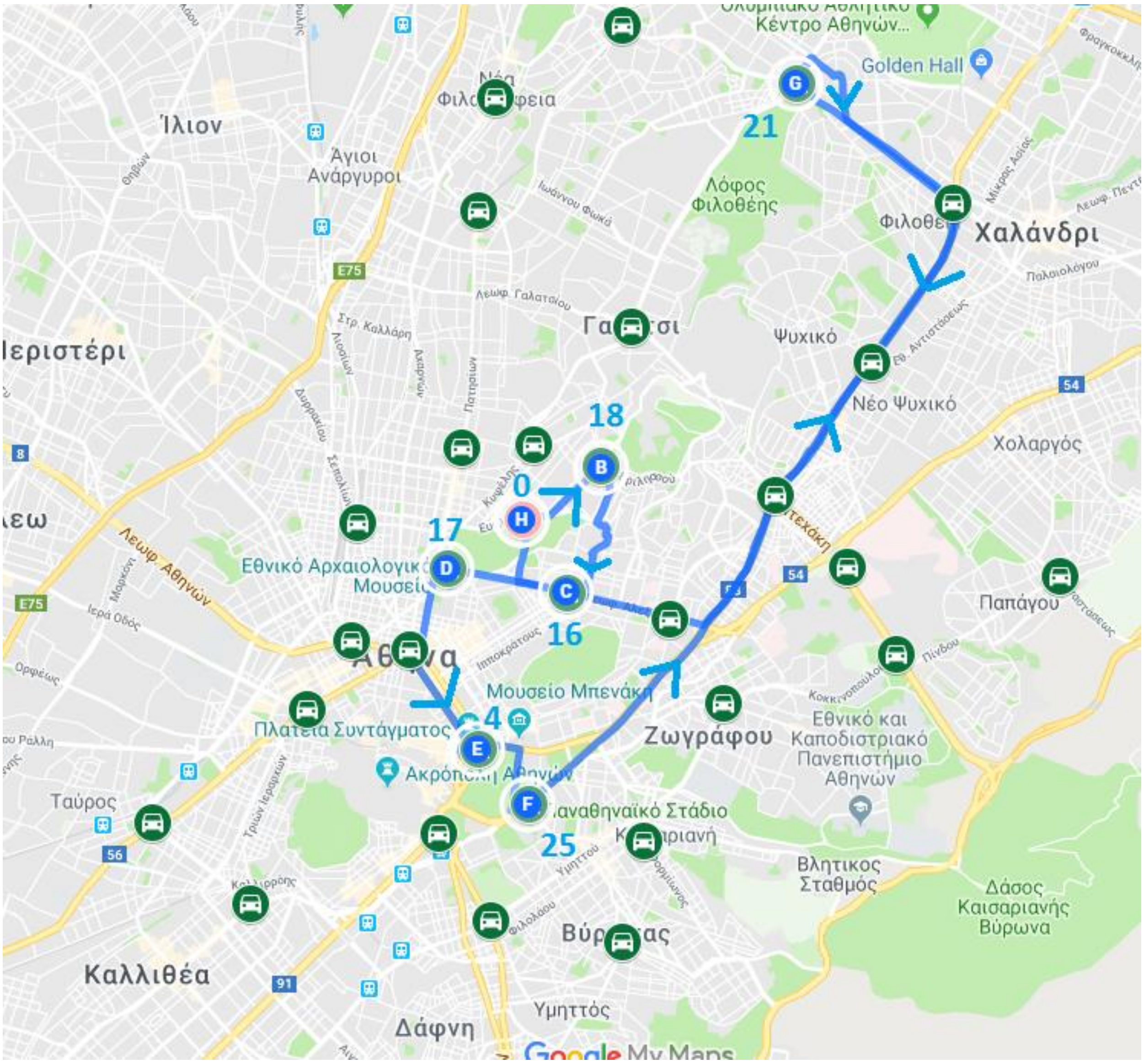
Στον πίνακα 4.7 παρουσιάζονται τα αποτελέσματα της καλύτερης λύσης. Συγκεκριμένα, φαίνεται η διαδρομή κάθε οχήματος που χρησιμοποιήθηκε καθώς και η τιμή που λαμβάνει η αντικειμενική συνάρτηση στη βέλτιστη λύση. Επίσης, για να δοθεί στον χρήστη μία αίσθηση του χρόνου των διαδρομών, τυπώνονται και οι διάρκειες της μεγαλύτερης και της μικρότερης διαδρομής. Για τη βέλτιστη λύση, προέκυψαν οι χρόνοι 35.96 και 26.76 λεπτά αντίστοιχα, οι οποίοι θεωρούνται αποδεκτοί με βάση την κατεύθυνση της εργασίας προς γρήγορη εξυπηρέτηση των πελατών.

Επίσης, αναλύοντας τα αποτελέσματα διαπιστώνουμε ότι γίνεται δύο φορές χρήση του ride sharing (κοινή χρήση διαδρομής) και συγκεκριμένα πραγματοποιείται στα οχήματα 3 και 5 για τις ζητήσεις 12-22 και 8-3 αντίστοιχα.

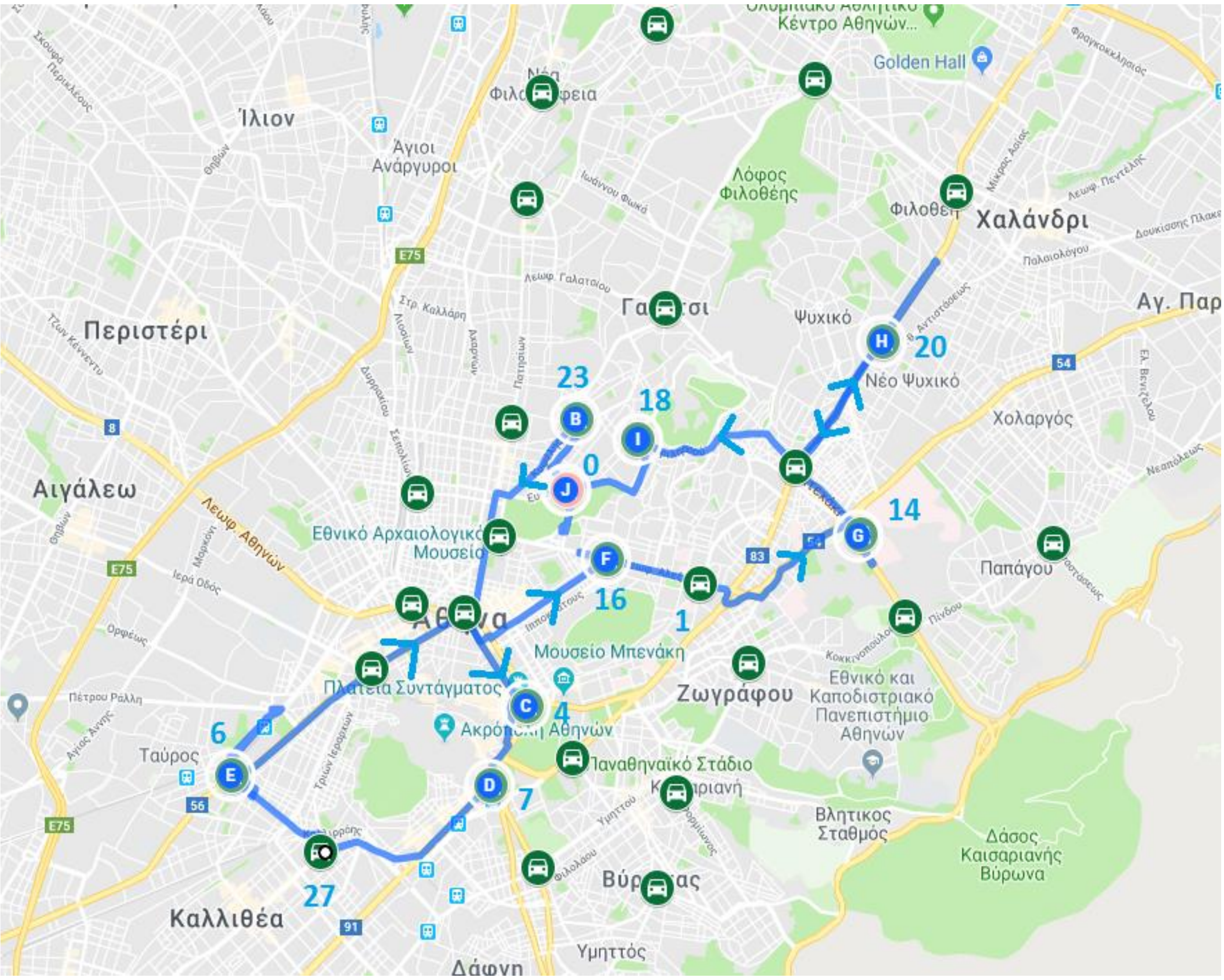
Στο σχήμα 4.2 φαίνονται οι καλύτερες τιμές τις αντικειμενικής συνάρτησης που προκύπτουν κάθε φορά με την πάροδο των επαναλήψεων. Όπως φαίνεται, οι βέλτιστες τιμές βρίσκονται σχετικά γρήγορα (πριν τις 5000 επαναλήψεις) και μετά από αυτές δεν βρίσκεται κάποια καλύτερη.

4.4.1 Γραφική ανάλυση των αποτελεσμάτων

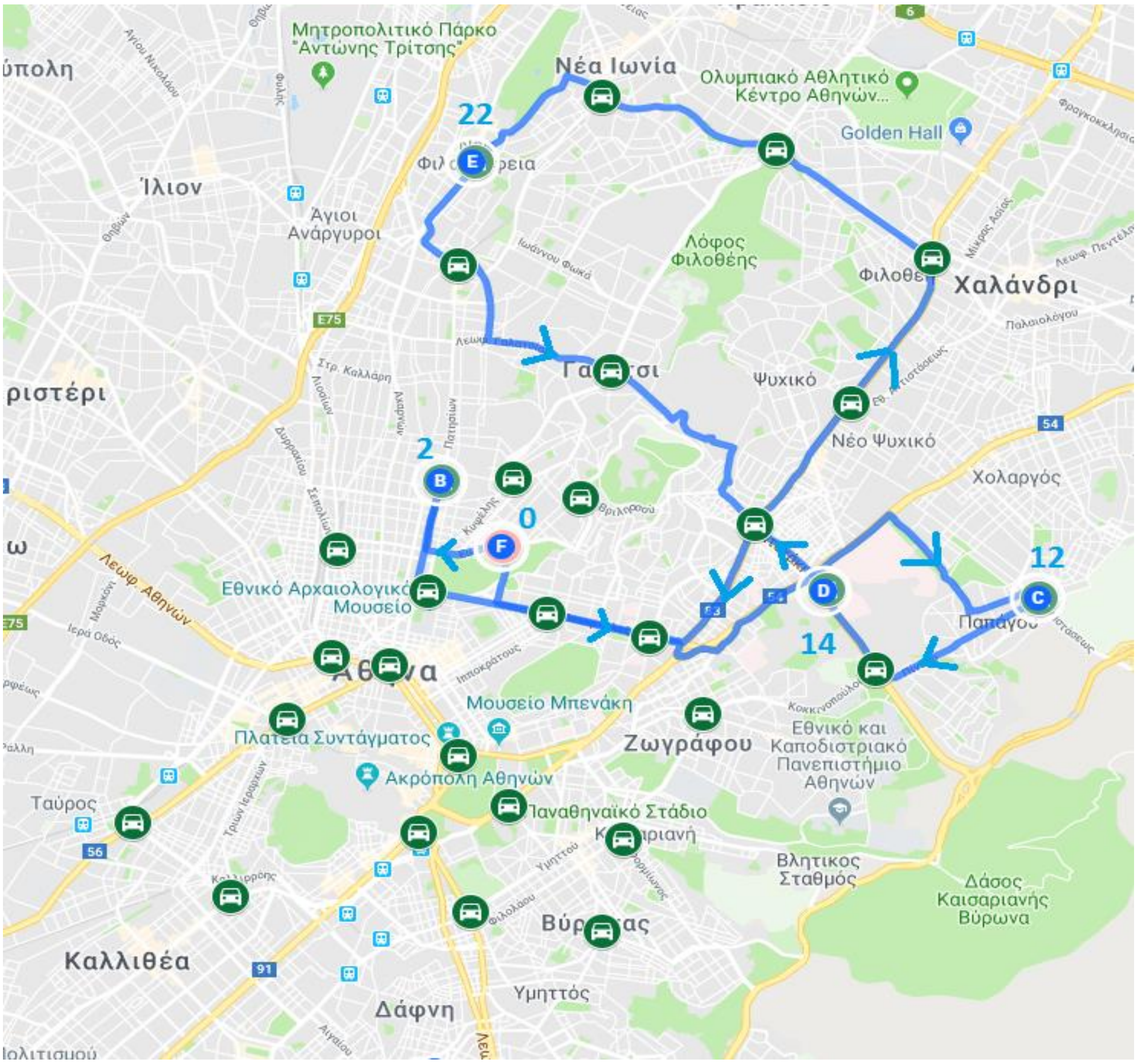
Οι διαδρομές των οχημάτων σχηματικά φαίνονται στα παρακάτω σχήματα:



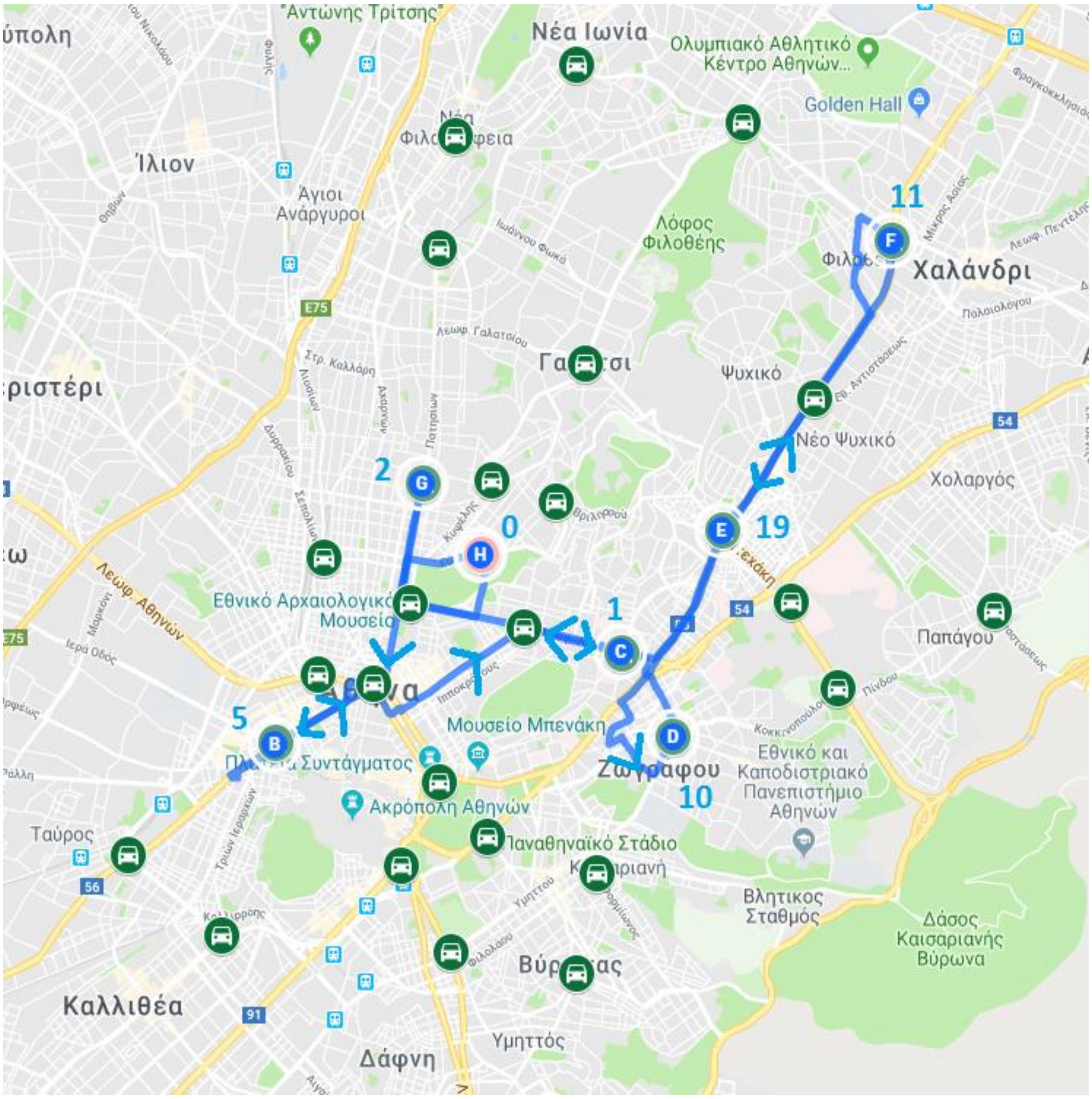
Σχήμα 4.3: Διαδρομή οχήματος 1



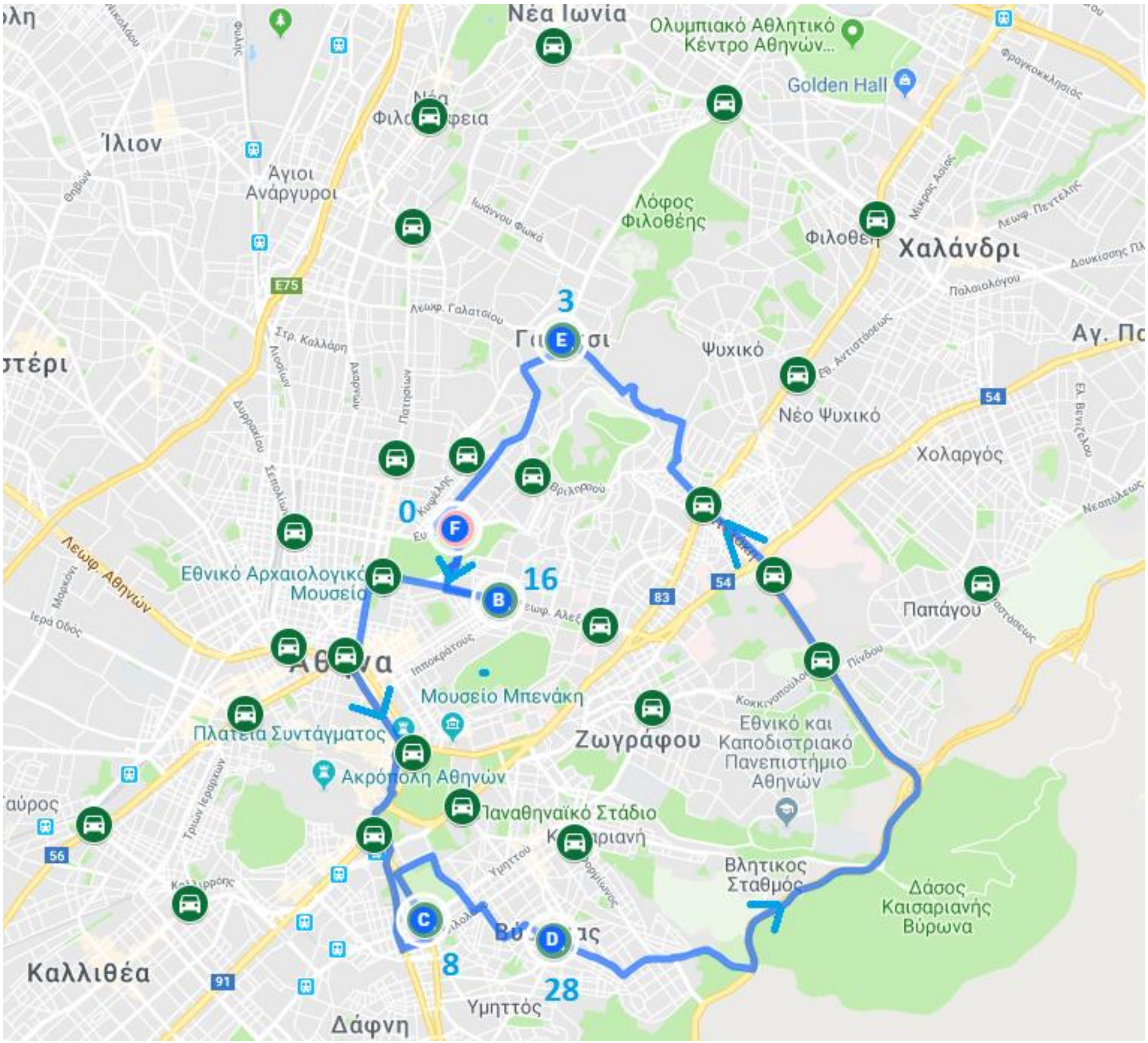
Σχήμα 4.4: Διαδρομή οχήματος 2



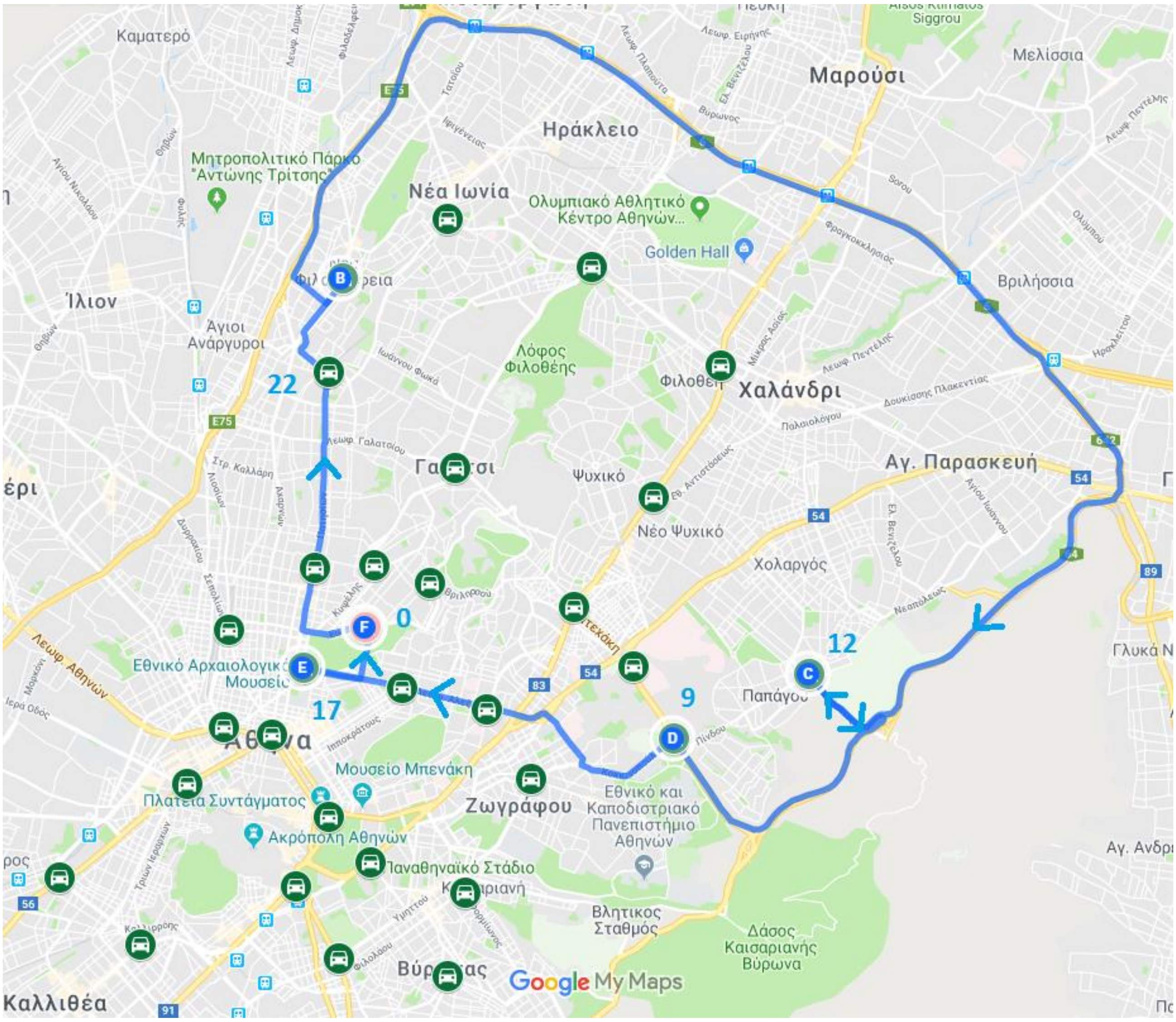
Σχήμα 4.5: Διαδρομή οχήματος 3



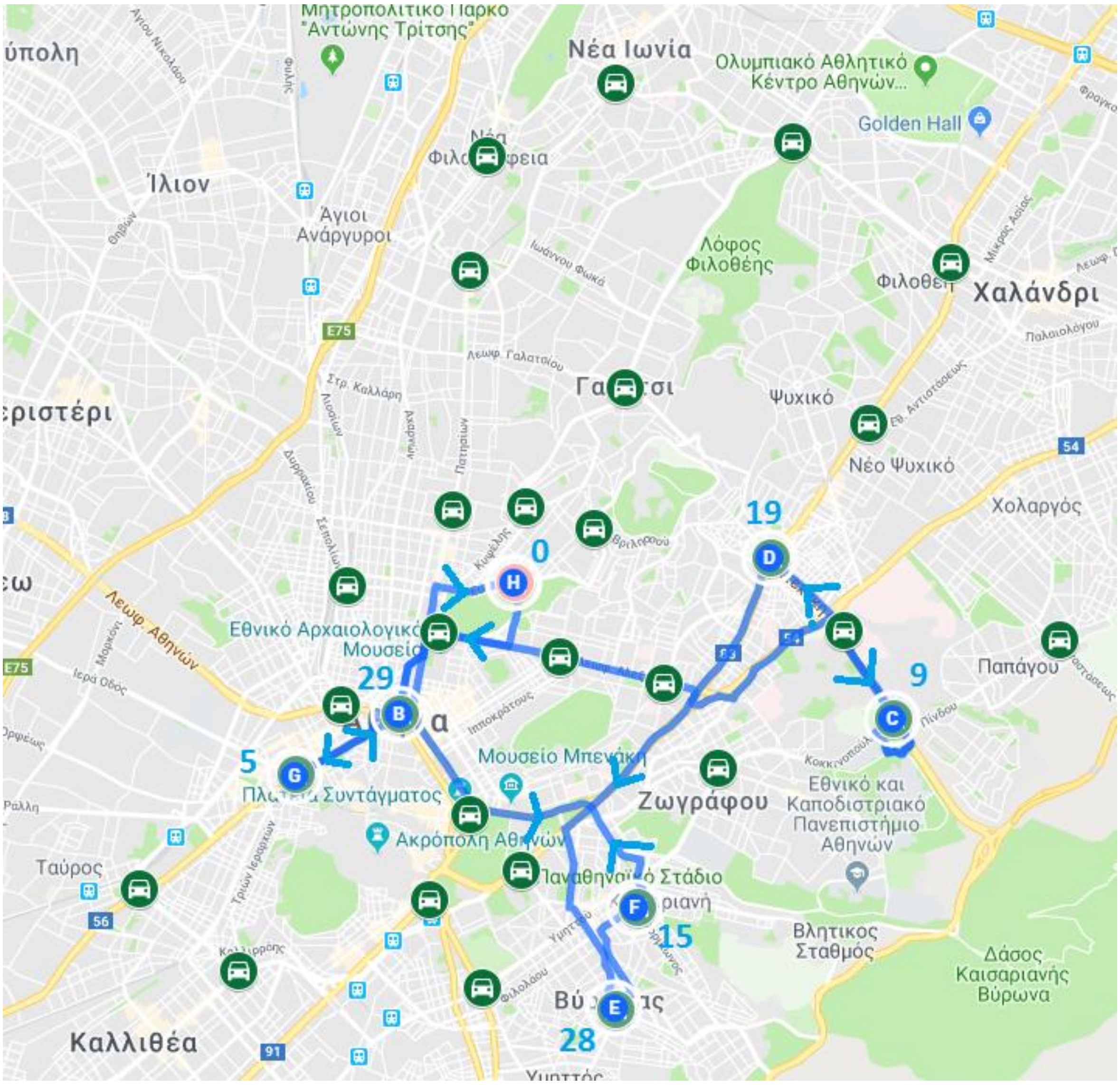
Σχήμα 4.6: Διαδρομή οχήματος 4



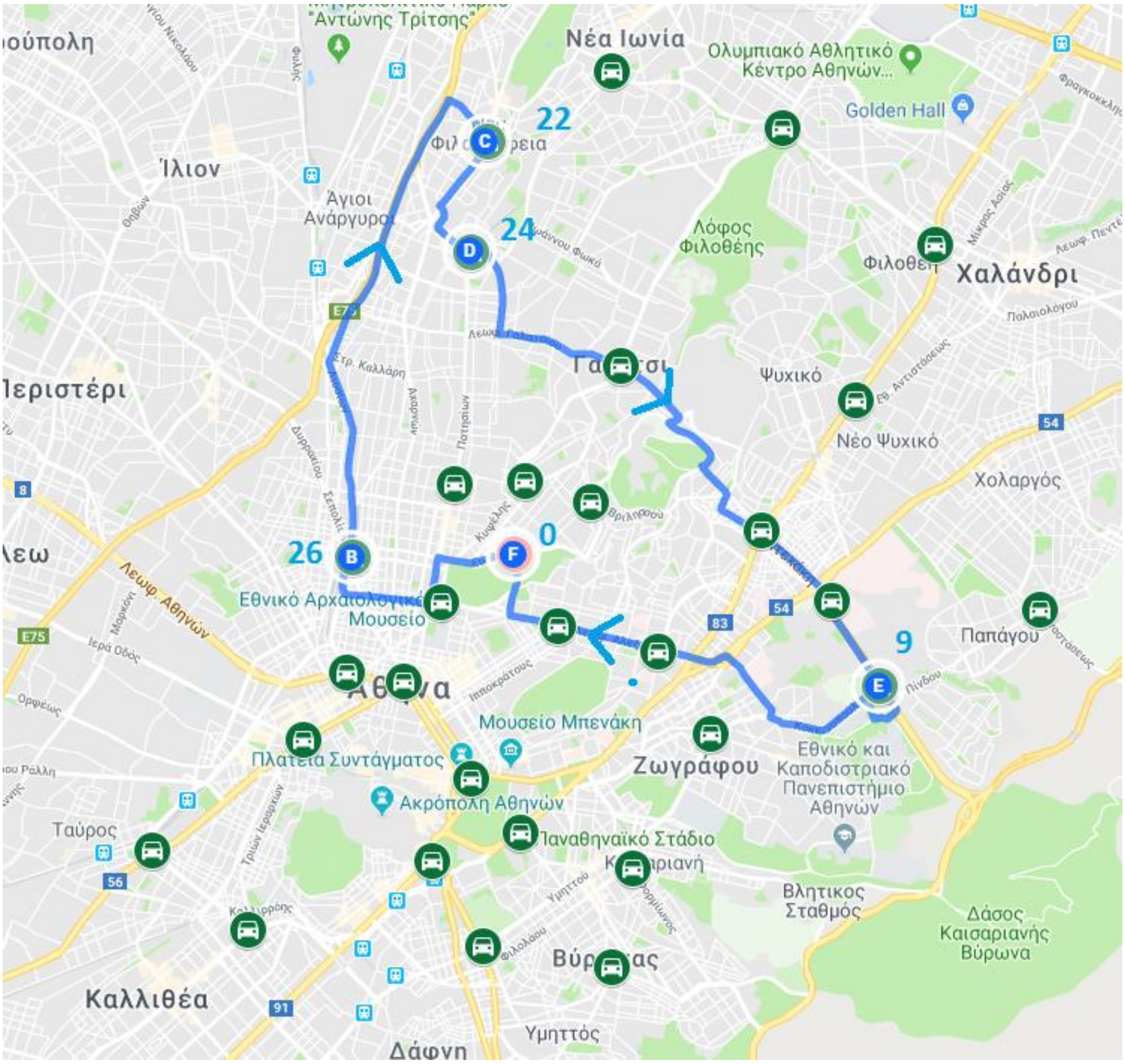
Σχήμα 4.7: Διαδρομή οχήματος 5



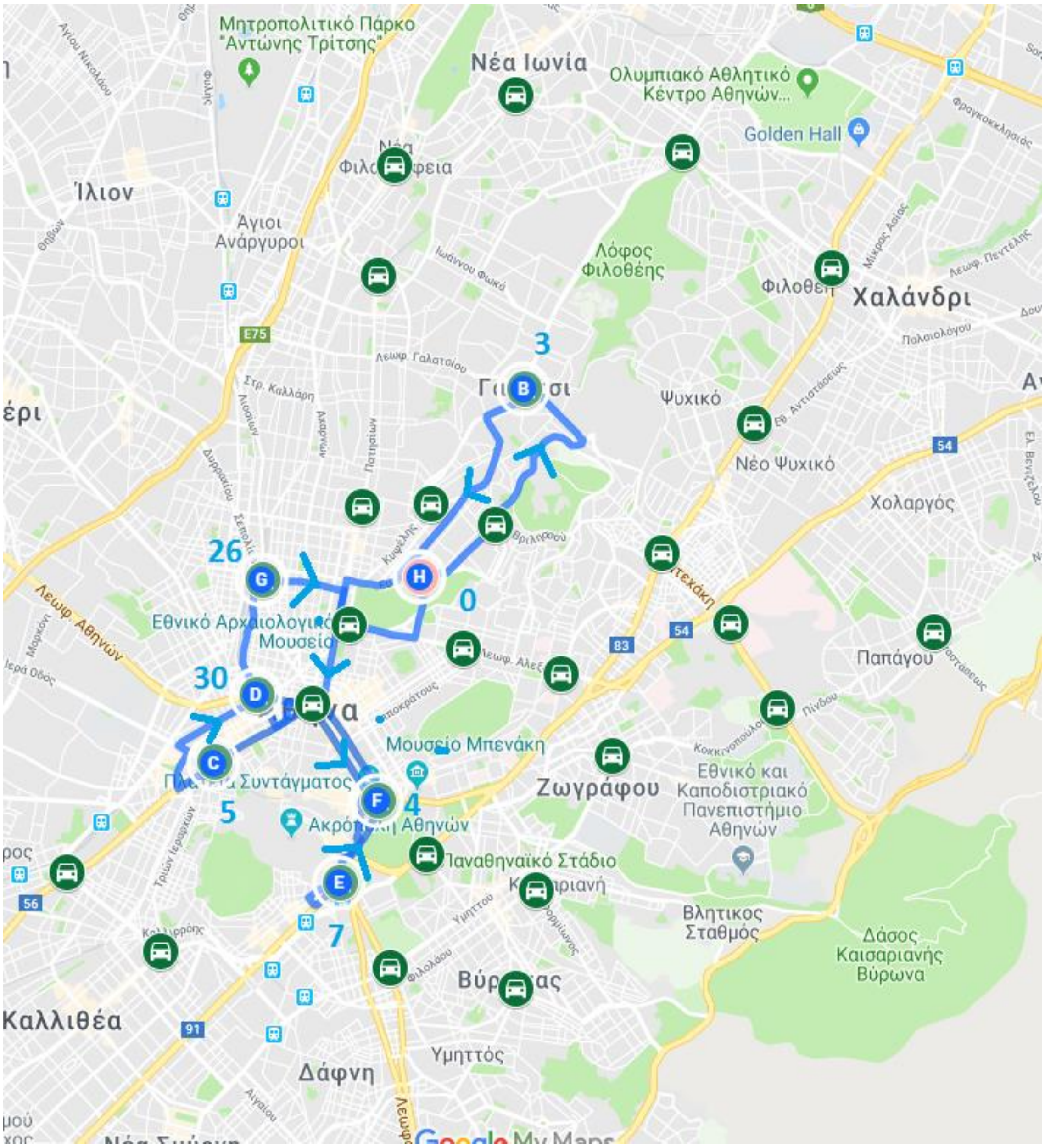
Σχήμα 4.8: Διαδρομή οχήματος 6



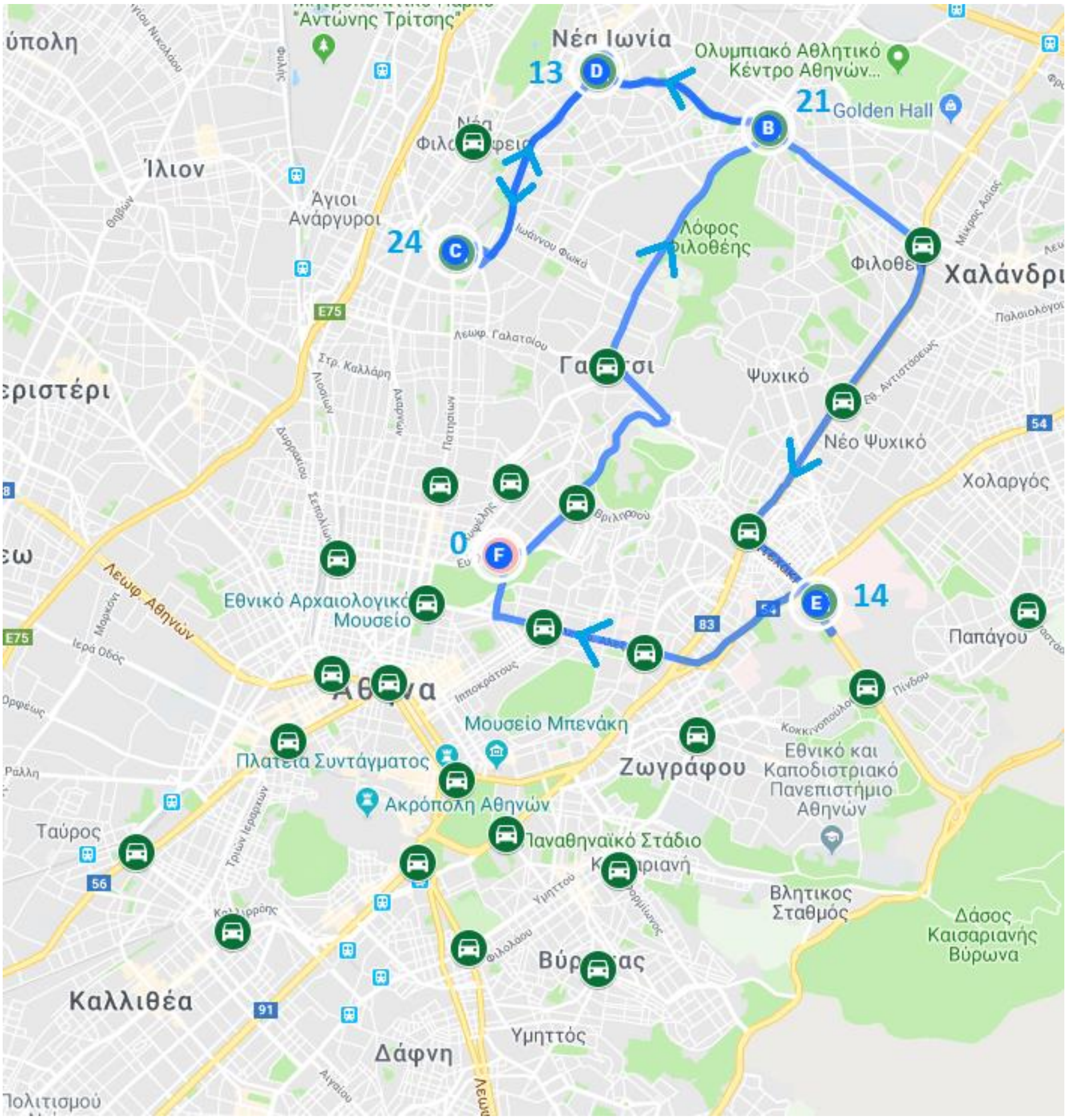
Σχήμα 4.9: Διαδρομή οχήματος 7



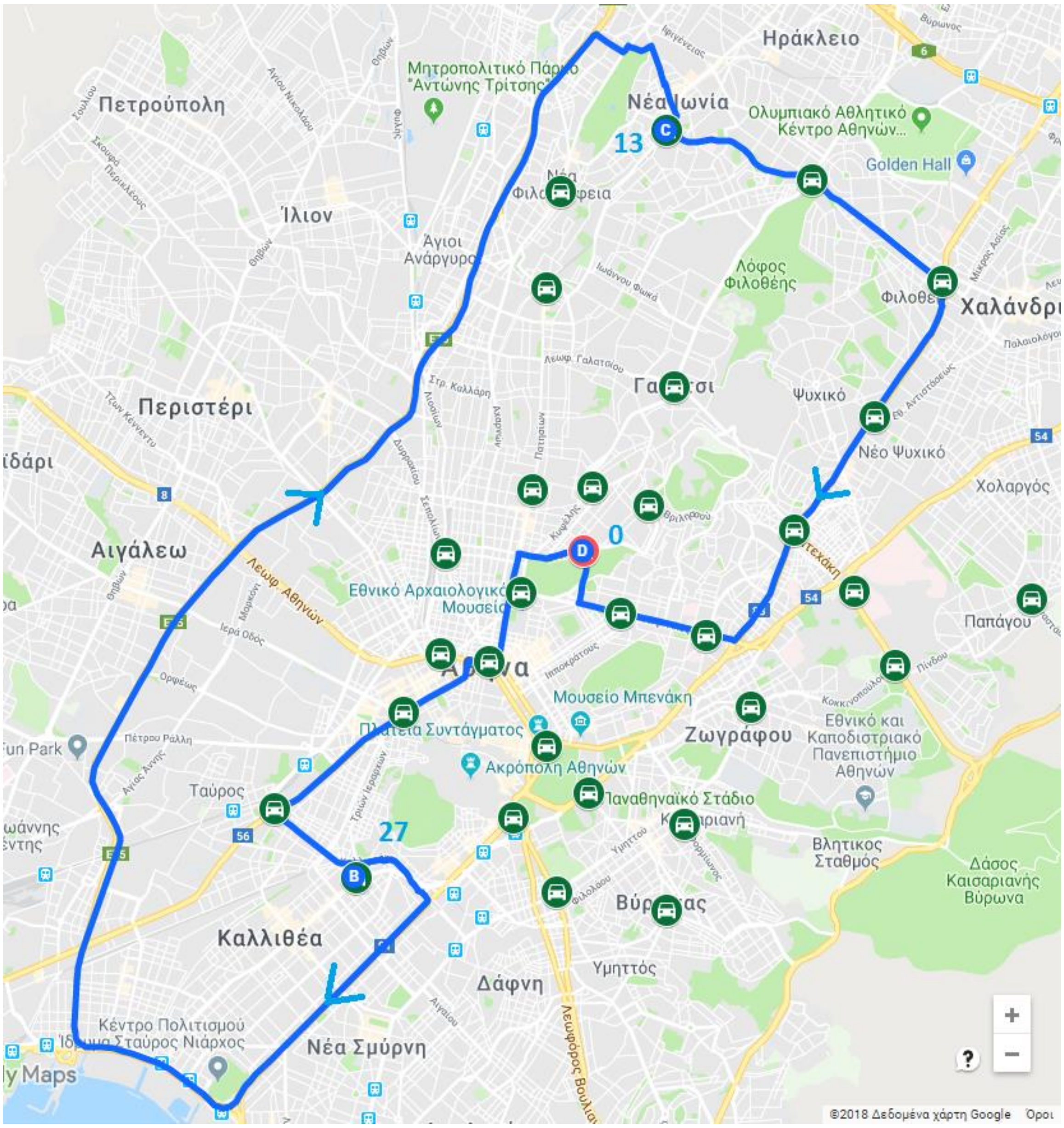
Σχήμα 4.10: Διαδρομή οχήματος 8



Σχήμα 4.11: Διαδρομή οχήματος 9



Σχήμα 4.12: Διαδρομή οχήματος 10



Σχήμα 4.13: Διαδρομή οχήματος 11

4.5 ΑΝΑΛΥΣΗ ΕΥΑΙΣΘΗΣΙΑΣ

Εφόσον βρέθηκε η λύση με την οποία το πρόβλημα επιλύεται με βέλτιστο τρόπο, πραγματοποιείται ανάλυση ευαισθησίας για αυτή τη λύση. Οι παράμετροι με τις οποίες θα εφαρμοστεί η ανάλυση είναι το μέγιστο εύρος διαδρομής και η χωρητικότητα..

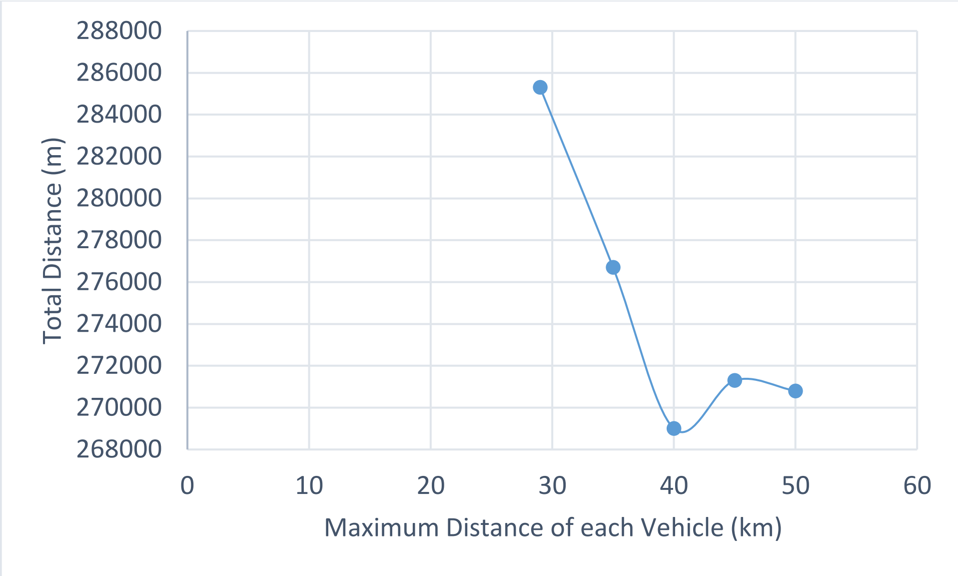
4.5.1 Μέγιστο εύρος διαδρομής

Όπως είναι λογικό, η μεταβολή του μέγιστου εύρους κάθε διαδρομής επηρεάζει τις τιμές της αντικειμενικής συνάρτησης, καθώς μεγαλύτερο εύρος συνεπάγεται μεγαλύτερη ευελιξία για το οχήματα άρα και λιγότερα χιλιόμετρα ώστε να πηγαίνουν και να ξεκινάνε και πάλι από το αμαξοστάσιο. Επομένως, κατά κύριο λόγο αύξηση του μέγιστου εύρους διαδρομής σημαίνει μείωση της συνολικής απόστασης που διανύει το σύνολο των οχημάτων. Όπως, όμως, φαίνεται και στο σχήμα 4.14 όταν το εύρος αυξηθεί πάρα πολύ, οι τιμές της αντικειμενικής συνάρτησης αρχίζουν να συγκλίνουν. Αυτό συμβαίνει γιατί τα οχήματα σπαταλούν αρκετό χρόνο στον δρόμο καθώς μετακινούνται από σημείο σε σημείο μέσα στο δίκτυο.

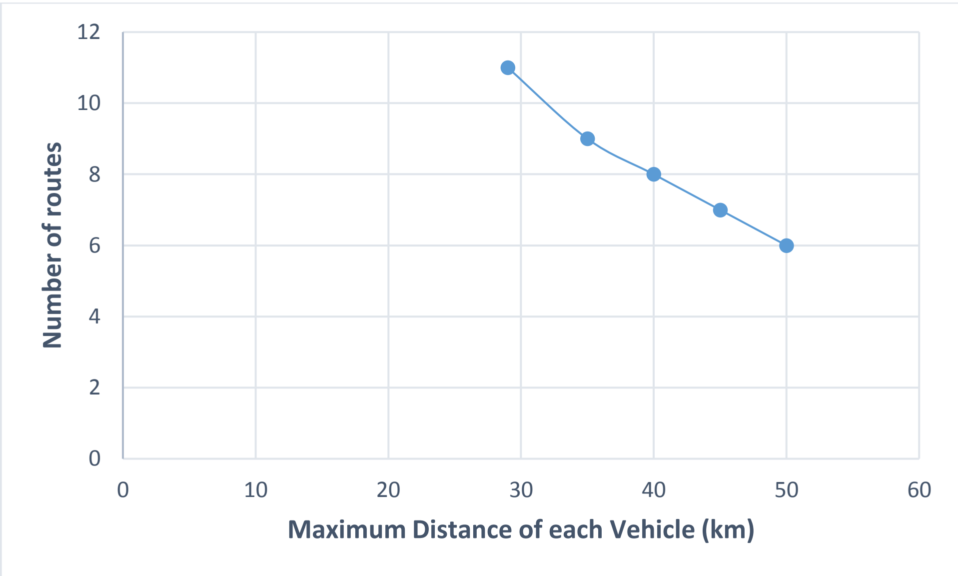
Σε ότι αφορά τον αριθμό των διαδρομών, αύξηση του μέγιστου εύρους διαδρομής συνεπάγεται, όπως είναι λογικό, τη μείωση του αριθμού τους καθώς επιτρέπεται στα οχήματα να εξυπηρετήσουν μεγαλύτερο αριθμό πελατών προτού αυτά χρειαστεί να επιστρέψουν στο αμαξοστάσιο. Τα αποτελέσματα φαίνονται στο σχήμα 4.15.

Τα προηγούμενα αποτελέσματα είναι λογικά λαμβάνοντας υπόψη ότι δεν συμπεριλαμβάνεται κάποιος χρονικός περιορισμός. Για αυτόν τον λόγο ελέγχθηκαν οι διάρκειες των μικρότερων και μεγαλύτερων διαδρομών. Όπως είναι λογικό, όσο αυξάνεται το μέγιστος εύρος κάθε διαδρομής, αυξάνονται και οι χρόνοι. Τα αποτελέσματα φαίνονται στο σχήμα 4.16.

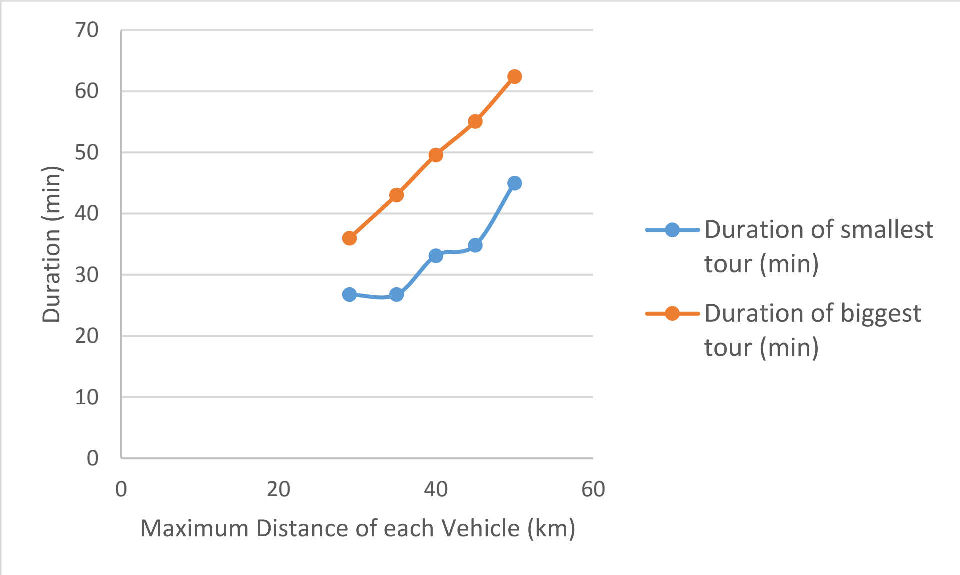
Όλες οι δοκιμές έγιναν για εύρος οχημάτων 29,35,40,45 και 50 χιλιόμετρα.



Σχήμα 4.14: Μεταβολή της αντικειμενικής συνάρτησης σε σχέση με το εύρος των οχημάτων



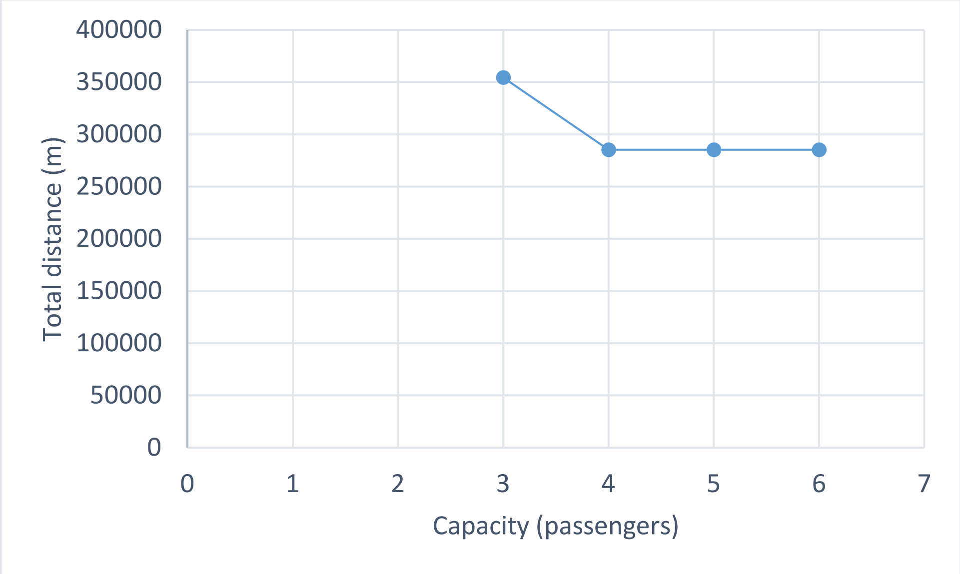
Σχήμα 4.15: Μεταβολή του αριθμού των διαδρομών σε σχέση με το μέγιστο εύρος



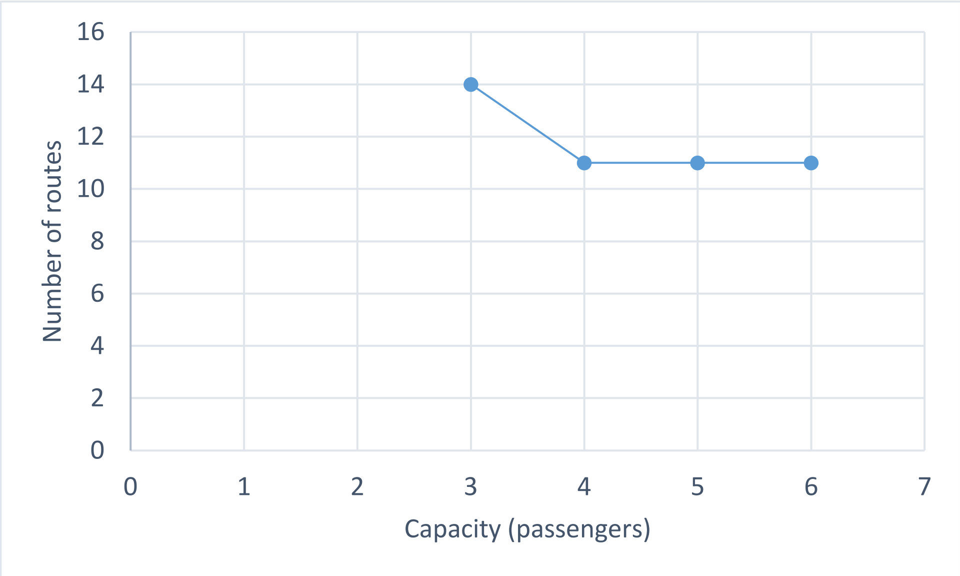
Σχήμα 4.16: Μεταβολή της διάρκειας της μικρότερης και της μεγαλύτερης υπο-διαδρομής σε σχέση με το μέγιστο εύρος των οχημάτων

4.5.2 Χωρητικότητα οχημάτων

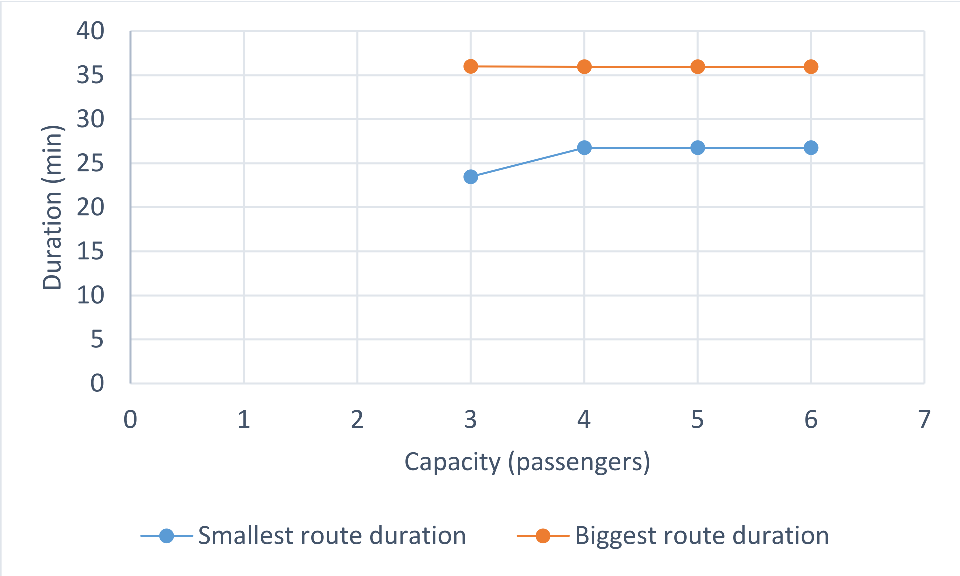
Όπως είναι αναμενόμενο, η αύξηση της χωρητικότητας κάθε οχήματος μειώνει την τιμή της αντικειμενικής συνάρτησης αλλά και τον αριθμό των απαιτούμενων διαδρομών. Αυτό συμβαίνει γιατί κάθε όχημα, με αύξηση της χωρητικότητας, έχει τη δυνατότητα να εξυπηρετήσει μεγαλύτερο αριθμό πελατών. Σχετικά με τους χρόνους διαδρομών, αναμένεται αύξηση του μέσου όρου τους. Στο σχήμα φαίνεται ότι οι χρόνοι των μικρότερων διαδρομών αυξάνονται, ενώ οι χρόνοι των μέγιστων διαδρομών μειώνονται ελάχιστα σε σημείο που θα μπορούσαμε να πούμε πως μένουν στάσιμοι. Το γεγονός ότι από χωρητικότητα 4 και πάνω τα αποτελέσματα παραμένουν ίδια οφείλεται στην επιβολή μικρού εύρους διαδρομής για κάθε όχημα και σε περιορισμούς που έχουν αναφερθεί σχετικά με τη ζήτηση και με τον αλγόριθμο.



Σχήμα 4.17: Μεταβολή της συνολικής απόστασης σε σχέση με τη χωρητικότητα



Σχήμα 4.18: Μεταβολή του αριθμού των διαδρομών σε σχέση με τη χωρητικότητα



Σχήμα 4.19: Μεταβολή των χρόνων της μικρότερης και μεγαλύτερης διαδρομής σε σχέση με τη χωρητικότητα.

4.6 ΣΥΜΠΕΡΑΣΜΑΤΑ

4.6.1 Παράμετροι του αλγορίθμου

Τα αποτελέσματα που προέκυψαν από την εφαρμογή του αλγορίθμου της βελτιστοποίησης με βάση τη λειτουργία των αποικιών των μυρμηγκιών στο συγκεκριμένο πρόβλημα έδειξαν πως η επιλογή του συντελεστή εξάτμισης της φερομόνης και του συντελεστή σημασίας της απόστασης σε σχέση με τη φερομόνη παίζει σημαντικό ρόλο καθώς επηρεάζει τα αποτελέσματα. Από τους διαφορετικούς συνδυασμούς που δοκιμάστηκαν προέκυψαν διαφορετικές τιμές της συνολικής απόστασης που αν και αυτές οι διαφορές ήταν μικρές, πρέπει να ληφθούν υπόψη στην επίλυση ενός προβλήματος βελτιστοποίησης. Έτσι, στο συγκεκριμένο πρόβλημα πιο αποδοτικός θεωρήθηκε ο συντελεστής 0.5 για την εξάτμιση της φερομόνης σε συνδυασμό με την τιμή 0.5 για τον συντελεστή της σημασίας της απόστασης σε σχέση με τη φερομόνη.

4.6.2 Ανάλυση ευαισθησίας

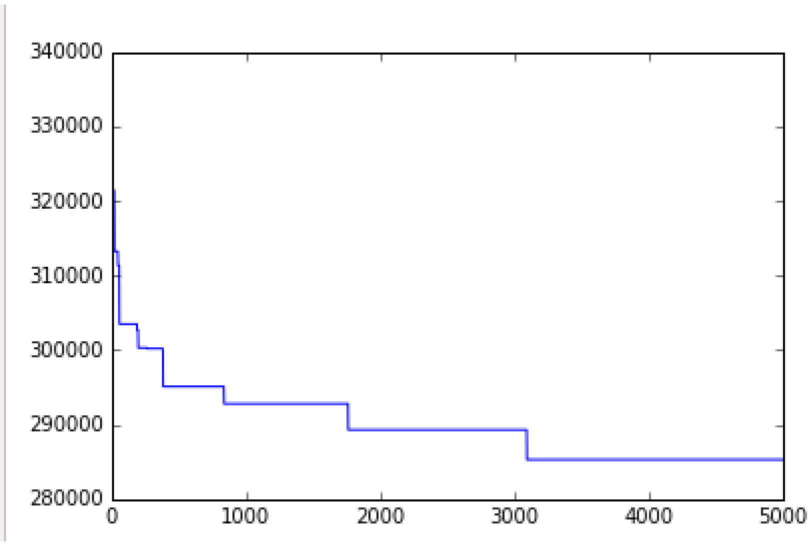
Η ανάλυση ευαισθησίας που πραγματοποιήθηκε με βάση το καλύτερο αποτέλεσμα του αλγορίθμου έδειξε πως η αντικειμενική συνάρτηση είναι ευαίσθητη σε μεταβολές του μέγιστου εύρους διαδρομής κάθε οχήματος και της χωρητικότητας.

Αρχικά, η μεταβολή του μέγιστου εύρους διαδρομής κάθε οχήματος έχει ως αποτέλεσμα μείωση της συνολικής απόστασης που διανύεται από το σύνολο του στόλου. Αυτή η μεταβολή οφείλεται στο γεγονός ότι ένα σύνολο οχημάτων με μεγαλύτερο εύρος διανύει συνολικά μικρότερη απόσταση μετακινούμενο από και προς το αμαξοστάσιο. Μείωση, επίσης, παρατηρείται και στον αριθμό των διαδρομών καθώς κάθε όχημα μπορεί να εξυπηρετήσει πλέον μεγαλύτερο αριθμό πελατών. Αυτά συμβαίνουν γιατί δεν λαμβάνεται υπόψη κάποιος συγκεκριμένος χρονικός περιορισμός. Μελετώντας τους χρόνους των μικρότερων και μεγαλύτερων διαδρομών παρατηρήθηκε ότι, όντως, αυτοί αυξάνονται γεγονός που αποτελεί αρνητικό παράγοντα για το συγκεκριμένο πρόβλημα καθώς μας ενδιαφέρει η όσο το δυνατόν ταχύτερη εξυπηρέτηση των πελατών.

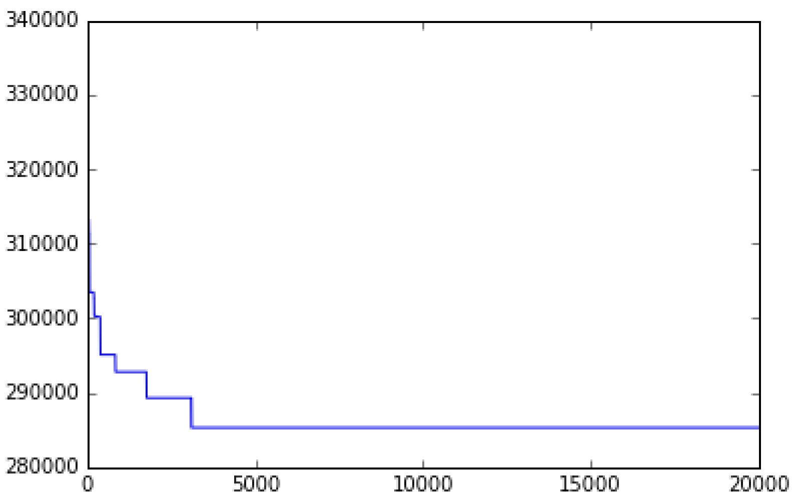
Αναφορικά με την ανάλυση σχετικά με τη μεταβολή της συνολικής απόστασης σε σχέση με την χωρητικότητα κάθε οχήματος, τα αποτελέσματα έδειξαν πως αύξηση της χωρητικότητας οδηγεί σε μείωση της συνολικής απόστασης. Μείωση, επίσης, σημειώνεται και στον αριθμό των διαδρομών καθώς κάθε όχημα έχει τη δυνατότητα να εξυπηρετήσει μεγαλύτερο αριθμό πελατών εξαιτίας της αυξημένης χωρητικότητας. Σχετικά με τους χρόνους διαδρομών, δεν παρατηρείται αξιοσημείωτη μεταβολή στον μέγιστο χρόνο διαδρομής, ενώ με αυξημένη χωρητικότητα παρατηρείται αύξηση στον χρόνο της μικρότερης διαδρομής. Τα ταυτόσημα αποτελέσματα για χωρητικότητα 4,5 και 6 σχετίζονται με το γεγονός ότι ο αλγόριθμος εκτελείται για σχετικά μικρό εύρος διαδρομών με στόχο την γρήγορη εξυπηρέτηση των πελατών αλλά και με κάποιους περιορισμούς στη ζήτηση και τη δομή του αλγορίθμου οι οποίοι αποτρέπουν την υπέρβαση της χωρητικότητας.

4.6.3 Σύγκλιση του αλγόριθμου

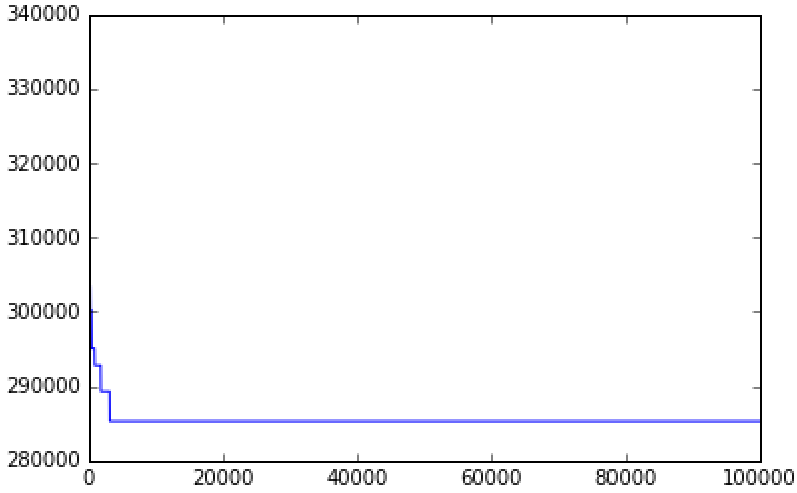
Οι μικρές σχετικά διαφορές που παρουσιάζουν τα αποτελέσματα της εφαρμογής του αλγόριθμου βελτιστοποίησης με βάση τη λειτουργία των αποικιών των μυρμηγκιών για τις διαφορετικές παραμέτρους του, δείχνουν πως αποφεύγεται ο εγκλωβισμός σε κάποιο τοπικό βέλτιστο, ή αλλιώς ότι ο αλγόριθμος συγκλίνει. Παρακάτω, παρατίθενται κάποια παραδείγματα για διαφορετικό αριθμό επαναλήψεων που δείχνουν ότι ο αλγόριθμος συγκλίνει στην ίδια τιμή της αντικειμενικής συνάρτησης (προφανώς για τα ίδια δεδομένα και παραμέτρους).



Σχήμα 5.1: Σύγκλιση αλγορίθμου για 5000 επαναλήψεις



Σχήμα 5.2: Σύγκλιση αλγορίθμου για 20000 επαναλήψεις



Σχήμα 5.3: Σύγκλιση αλγορίθμου για 100000 επαναλήψεις

ΚΕΦΑΛΑΙΟ 5: ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΕΡΕΥΝΑ

5.1 ΣΥΝΟΨΗ ΜΕΛΕΤΗΣ

Αντικείμενο της παρούσας διπλωματικής εργασίας αποτέλεσε ο βέλτιστος σχεδιασμός δικτύου διαδρομών αυτόνομων οχημάτων, με εφαρμογή του μοντέλου σε 31 σημεία στο κέντρο της Αθήνας με στόχο την εξυπηρέτηση της ζήτησης ενός συνόλου πελατών.

Αρχικά πραγματοποιήθηκε μία εκτενής περιγραφή των ήδη υπαρχουσών εργασιών που ασχολούνται με το πρόβλημα δρομολόγησης οχημάτων και της βιβλιογραφίας των γενικότερων προβλημάτων στο οποίο ανήκει και το συγκεκριμένο, δηλαδή η βιβλιογραφία σχετικά με τα “vehicle routing problem”, “pick up and delivery problem” και “dial-a-ride problem”. Ακόμη, περιγράφηκαν συνοπτικά εργασίες σχετικές με τη μέθοδο βελτιστοποίησης που χρησιμοποιήθηκε για την επίλυση του προβλήματος, δηλαδή τη «βελτιστοποίηση με βάση τη λειτουργία των αποικιών των μυρμηγκιών».

Στο επόμενο κεφάλαιο διατυπώθηκε μαθηματικά το πρόβλημα και επεξηγήθηκε η αντικειμενική συνάρτηση που οι περιορισμοί που τέθηκαν. Στόχος της συνάρτησης είναι η ελαχιστοποίηση της συνολικής απόστασης που διανύει το σύνολο των οχημάτων που χρησιμοποιούνται σε μία πλήρη λύση με την οποία εξυπηρετούνται όλοι οι πελάτες. Επίσης, αναλύθηκε πλήρως η μεθοδολογία της συγκεκριμένης βελτιστοποίησης με την θεωρία και τις εξισώσεις που την εκφράζουν.

Στο τέταρτο κεφάλαιο εφαρμόστηκε ο αλγόριθμος που κατασκευάστηκε για την επίλυση του προβλήματος ο οποίος κατασκευάστηκε με δεδομένα από την περιοχή της Αθήνας και τυχαία ζήτηση. Αφού παρουσιάστηκαν τα αποτελέσματα που προέκυψαν από την εφαρμογή του αλγορίθμου για διαφορετικούς συνδυασμούς παραμέτρων του, παρουσιάστηκε αναλυτικά και σχηματικά το βέλτιστο δίκτυο διαδρομών. Επίσης, πραγματοποιήθηκε ανάλυση ευαισθησίας ως προς το μέγιστο εύρος διαδρομής και την χωρητικότητα κάθε οχήματος.

Τέλος, στο συγκεκριμένο κεφάλαιο παρατίθενται τα συμπεράσματα που προέκυψαν από την εκπόνηση της διπλωματικής εργασίας και κάποιες προτάσεις για περαιτέρω διερεύνηση του προβλήματος.

5.2 ΣΥΜΠΕΡΑΣΜΑΤΑ

Αν και το αντικείμενο της εργασίας, δηλαδή το πρόβλημα της δρομολόγησης οχημάτων έχει αρχίσει να απασχολεί την επιστημονική κοινότητα από την δεκαετία του 1960, η βελτιστοποίηση με βάση τη λειτουργία των αποικιών των μυρμηγκιών δεν έχει εφαρμοστεί αρκετά σε αυτό. Η μεγάλη διαφορά της συγκεκριμένης εργασίας από τις περισσότερες, είναι ότι λαμβάνει υπόψη τη ζήτηση ενός συνόλου πελατών η οποίοι έχουν διαφορετικούς προορισμούς. Ακόμη, λαμβάνει υπόψη και το οικονομικό συμφέρον του φορέα της δημιουργίας μιας τέτοιας πρωτοβουλίας, καθώς ελαχιστοποιεί το κόστος το οποίο εκφράζεται με την έννοια της συνολική απόστασης ενώ λαμβάνει υπόψη και λειτουργικούς περιορισμούς σχετικούς με τα οχήματα.

Ο αλγόριθμος εφαρμόστηκε σε ένα σύνολο σημείων εντός της Αθήνας των οποίων η επιλογή έγινε έτσι ώστε να βρίσκονται σε περιοχές που αναμένεται να υπάρχει μεγάλη ζήτηση και σε ρεαλιστικές συνθήκες. Σε όλες τις λύσεις, εξυπηρετούνται όλοι οι πελάτες με σχετικά μικρό αριθμό οχημάτων αν ληφθεί υπόψη και ο αυστηρός περιορισμός του εύρους διαδρομής που επιβάλλεται σε κάθε ένα από αυτά. Έτσι, τα αποτελέσματα του αλγορίθμου είναι αποδεκτά και καλύπτουν τις ανάγκες τόσο των μετακινούμενων όσο και των φορέων που μπορεί να εμπλέκονται στην παροχή τέτοιων υπηρεσιών.

5.3 ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΕΡΕΥΝΑ

Όπως έχει αναφερθεί το πρόβλημα δρομολόγησης ανήκει στην κατηγορία των NP-hard προβλημάτων, επομένως δεν είναι γνωστή η ακριβής λύση. Έτσι, και στη συγκεκριμένη μελέτη, είναι δυνατόν να εφαρμοστούν κάποιες παραλλαγές στις οποίες μπορεί να βασιστεί μελλοντική έρευνα ώστε να βελτιώσει και να επεκτείνει το πρόβλημα.

5.3.1 Παραλλαγές στην αντικειμενική συνάρτηση και τους περιορισμούς

Η αντικειμενική συνάρτηση του προβλήματος λήφθηκε ίση με το άθροισμα των μηκών των διαδρομών που διανύονται από όλα τα οχήματα σε μία πλήρη λύση. Είναι, όμως, εφικτό να εισαχθούν περισσότερες μεταβλητές στην αντικειμενική συνάρτηση όπως για παράδειγμα διαφορετικοί τύποι οχημάτων, δηλαδή οχήματα με διαφορετικές χωρητικότητες ή διαφορετικά μέγιστα εύρη διαδρομών.

Σχετικά με τον χρονικό περιορισμό στον αλγόριθμο, αυτός δίνεται μέσω του μέγιστου εύρους διαδρομής κάθε οχήματος καθώς αυτά τα δύο σχετίζονται λαμβάνοντας υπόψη ότι θεωρούμε σταθερή ταχύτητα και δεν λαμβάνεται υπόψη τυχόν κυκλοφοριακή συμφόρηση που μπορεί να προκύψει εντός του δικτύου. Θα μπορούσε, ο χρονικός περιορισμός να εισαχθεί ως ευέλικτος περιορισμός (soft constraint) μέσω της χρήσης συνάρτησης ποινής και πιθανώς να γίνονται αποδεκτές και λύσεις που ελαφρώς ξεπερνούν το όριο.

Στην παρούσα εργασία θεωρήθηκε ένας συγκεκριμένος αριθμός οχημάτων που μπορούν να χρησιμοποιηθούν, τα οποία επαρκούν για να εξυπηρετηθούν όλοι οι πελάτες. Μία μελλοντική εργασία θα μπορούσε να θέτει έναν περιορισμό μέγιστου αριθμού οχημάτων χωρίς να είναι απαραίτητο ότι θα εξυπηρετηθεί ολόκληρη η ζήτηση. Έτσι, θα μπορούσε να εισαχθεί στην αντικειμενική συνάρτηση και η ανικανοποίητη ζήτηση.

5.3.2 Παραλλαγές στον αλγόριθμο επίλυσης

Στην επίλυση του προβλήματος της δρομολόγησης οχημάτων έχουν χρησιμοποιηθεί πολλές μέθοδοι, κυρίως ακέραιου προγραμματισμού, ευρετικές και μεταευρετικές. Τα τελευταία χρόνια οι υβριδικές μέθοδοι που συνδυάζουν διαφορετικές μεθόδους βελτιστοποίησης, πχ μεταευρετικό αλγόριθμο και ευρετική μέθοδο θεωρούνται πολλά υποσχόμενες για την επίλυση σύνθετων προβλημάτων.

Υπάρχουν πολλές μετατροπές που μπορούν να πραγματοποιηθούν στον αλγόριθμο που κατασκευάστηκε σε αυτή την εργασία ώστε να γίνει ακόμα πιο αποτελεσματικός κυρίως όσον αφορά την ευελιξία του σχετικά με διαφορετικά αιτήματα . Επίσης, είναι πιθανό η αρχική λύση που προκύπτει με τη χρήση της αναζήτησης εγγύτερου γείτονα να επηρεάζει το τελικό αποτέλεσμα και η χρήση μιας άλλης μεθόδου να δίνει καλύτερα αποτελέσματα.

5.3.3 Προτάσεις για επέκταση του προβλήματος

Το πρόβλημα επιλύθηκε για 1 αμαξοστάσιο και 30 στάσεις με βάση τις οποίες παράγεται η ζήτηση. Ο τρόπος που δομήθηκε το πρόβλημα επιτρέπει να γίνουν αλλαγές κυρίως ως προς το μέγεθος του δικτύου αλλά με τις κατάλληλες μετατροπές και για μεγαλύτερο αριθμό πελατών.

Ακόμη, όπως είναι δομημένος ο αλγόριθμος, η κοινή χρήση διαδρομής (ride sharing) ελέγχεται στο τέλος κάθε διαδρομής αν μπορεί να γίνει για διαδρομές που αποτελούνται από 4 σημεία. Σε περίπτωση χρήσης μεγαλύτερου εύρους οχήματος, αυτός ο έλεγχος θα έπρεπε να αλλάξει καθώς μεγαλώνουν και οι διαδρομές. Το σχετικά μικρό εύρος που ορίστηκε για κάθε όχημα ήταν απαραίτητο καθώς στόχος της εργασίας είναι η γρήγορη εξυπηρέτηση των πελατών λαμβάνοντας υπόψη και τη χαμηλή ταχύτητα των αυτόνομων οχημάτων σε σύγκριση με τα συμβατικά.

Ενδιαφέρον, επίσης, θα είχε αν στο πρόβλημα εισάγονταν «χρονικά παράθυρα» (time windows) μέσα στα οποία κάθε πελάτης απαιτεί να εξυπηρετηθεί. Προφανώς, η πολυπλοκότητα του προβλήματος θα αυξανόταν σημαντικά, ειδικά σε μεγάλα δίκτυα.

Τέλος, μία ερευνητική κατεύθυνση αφορά στην περίπτωση που νέα ζήτηση παραγόταν κατά τη διάρκεια εκτέλεσης της διαδικασίας, γεγονός που επίσης θα αύξανε κατά πολύ την πολυπλοκότητα της διαδικασίας με κίνδυνο να καταστεί ανέφικτη. Εάν αυτό συνέβαινε θα αναφερόμασταν σε δυναμική κοινή χρήση διαδρομής (dynamic ride sharing) αντί για στατική.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Agatz, N. & Erera, A. & Salvensbergh, M. & Wang, X. (2012). Optimization for dynamic ride-sharing: A review. European Journal of Operational Research 223 (2012) 295-303.
2. Ahn, B.-H. & Shin, J.Y. (1991). Vehicle-routing with time windows and time-varying congestion. J.Oper.Res.Soc.42 (5), 393-400.
3. Alonso-Mora, J. & Samaranayake, S. & Wallar, A & Frazzoli, E. & Rus, D. (2017). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proc. Natl. Acad. Sci., 462-467.
4. Angelelli, E. & Mansini, R. (2002). The vehicle routing problem with time windows and simultaneous pick up and delivery. In Quantitative approaches to distribution logistics and supply chain management series. Lecture Notes in Economics and Mathematical Systems (pp.249-267).
5. Ariyasingha, I.D.I.D. & Fernando, T.G.I. (2015). Performance analysis of the multi-objective ant colony optimization algorithms for the traveling salesman problem. Swarm Evolutionary Comput. 23, 11-26.
6. Artmerier, A. & Haselmayr, J. & Leucker, M. & Sachenbacher. The shortest path problem revisited: Optimal routing for electric vehicles. In KI 2010: Advances in Artificial Intelligence, LNCS, Volume 6359, pages 309-316, Springer, 2010.
7. Bell, J. E. & McMullen, P. R. (2004). Ant colony optimization techniques for the vehicle routing problem. Advance Engineering Informatics 18 (2004) 41-48.
8. Berbeglia, G. & Cordeau, J.F. & Gribkovskaia, I. & Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey. Sociedad de Estadistica e Investigacion Operativa 2007.
9. Boostani, A.& Ghodsi, R. & Miab, A.K. Optimal location of compressed natural gas (CNG) refueling station using the arc demand coverage model. In Proceedings of the 2010 Fourth Asia International Conference on Mathematical/ Analytical Modelling and Computer Simulation, AMS'10, pages 193-198. IEEE Computer Society, 2010.
10. Braysy, O. & Gendreau M. (2005a) Vehicle routing problem with time windows, part i: Route construction and local search algorithms. Transportation Science 39 (1): 104-118.
11. Braysy, O. & Gendreau M. (2005b) Vehicle routing problem with time windows, part ii: Metaheuristics. Transportation Science 39 (1): 119-139.
12. Bullnheimer, B. & Hartl, R.F. & Strauss, C. Applying the Ant System to the Vehicle Routing Problem, in Meta-heuristics: Advances and Trends in Local Search for

Optimization, S.Voss. S. Martello, I.H. Osman and C. Roucairol (eds.), Kluwer Academic Publishers, Boston, 1999, 285-296.

13. Bureau of Transportation Statistics (2014). Pocket Guide to Transportation. Department of Transportation, Bureau of Transportation Statistics, Washington, U.S.
14. Burns, L.D. & Jordan, W.C. & Scarborough, B.A. (2013). Transforming Personal Mobility. The Earth Institute.
15. Calvo, R.W. & Colorni, A. (2007). An effective and fast heuristic for the dial-a-ride problem. 4OR 5 (1), 61-73.
16. Casco, D.O. & Golden B.L. & Wasil, E.A. (1988). Vehicle routing with backhauls: Models, algorithms and case studies. In B.L. Golden & A.A. Assad (Eds.), Vehicle routin: Methods and studies (pp.127-147). Amsterdam: Elsevier
17. Catay, B. (2010) A new saving-based ant algorithm for the Vehicle Routing Problem with Simultaneous Pickup and Delivery. Expert Systems with Applications 37 (2010) 6809-6817
18. Chen, T.D. & Kockelman, K.M. & Hanna, J.P. (2016). Operations of a shared, autonomous, electric vehicle fleet: implications of vehicle & charging infrastructure decisions. Transport. Res. Part A: Policy Pract. 94, 243-254.
19. Chiu, T.-C. & Bottom, J. & Mahut, M. & Paz, A. & Balakrishna, R. & Waller, T. & Hicks, J. (2011). Dynamic Traffic Assignment: A Primer. Transportation Research E-Circular (E-C153).
20. Cordeau, J.F. & Laporte, G. (2007). The dial-a-ride problem: models and algorithms. Ann Oper Res (2007) 153: 29-46.
21. Cordeau, J-F. (2006). A branch-and-cut algorithm for the dial-a-ride problem. Oper. Res. 54(3), 573-586.
22. Crispim, J. & Brandao, J. (2005). Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls. Journal of the Operational Research Society, 56, 1296-1302.
23. De Almeida Correia, G.H. & van Arem, B. (2016). Solving the user optimum privately owned automated vehicles assignment problem (uo-poavap): a model to explore the impacts of self-driving vehicles on urban mobility. Transport. Res. Part B. Methodol. 87, 65-88.
24. Dethloff, J. (2001). Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up. OR Spektrum, 23, 79-96.
25. Dial, R. (1995). Autonomous dial-a-ride transit Introductory overview. Transpn.Res.-C, Vol.3, No.5, pp 261-275, 1995.

26. Dorigo, M & Maniezzo, V & Colorni, A. The Ant System: Optimization by a Colony of Cooperating Agents, IEEE Transactions on Systems, Man, and Cybernetics- Part B 26, 1996, 29-41.
27. Dorigo, M. & Blum, C. Ant colony optimization theory: A survey. Theoretical Computer Science 344 (2005) 243 – 278.
28. Dorigo, M. & Gambardella, LM. Ant colonies for the travelling salesman problem. BioSystem 1997;43(1):73-81.
29. Dorigo, M. & Stutzle, T. Ant colony optimization. Bradford Book; 2004.
30. Eksioglou, B., Vural, A.V., Reisman, A., (2009). The vehicle routing problem: a taxonomic review. Comput. Indust. Eng. 57 (3), 1472-1483.
31. Elloumia, W. & Abed, H.E. & Abraham, A. & Alimi, A.M. (2014). A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP. Appl.Soft.Comput. 25, 234-241.
32. Erdogan, S. & Miller-Hooks, E. A green vehicle routing problem. Transportation Research Part E: Logistics and Transportation Review, 48(1): 100-114, 2012.
33. Fagnant, D. & Kockelman K. (2014). The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. Transportation Research Part C, Vol 40 (2014): 1-13.
34. Fagnant, D. & Kockelman K. (2015). Operations of a shared autonomous vehicle fleet for the Austin, Texas market. Transportation Research Record No.2536: 98-106, 2015.
35. Fagnant, D. J. (2015). Dynamic ride-sharing and optimal fleet sizing for a system of shared autonomous vehicles. Proceeding of the 94th Annual Meeting of the Transportation Research Board in Washington DC, January 2015.
36. Figliozzi, M. (2010). Vehicle routing problem for emissions minimization. Transport. Res. Rec.: J. Transport. Res. Board (2197), 1-7.
37. Fu, L. (2002). Scheduling dial-a-ride paratransit under time-varying, stochastic congestion. Transport. Res. Part B: Methodol. 36 (6), 485-506.
38. Furuhata, M. & Dessouky, M. & Ordonez, F. & Brunet, M.E. & Wang, X. & Koenig, S. (2013). Transportation Research Part B 57 (2013) 28-46.
39. Gajpal, Y. & Abad, P. (2009): An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. Computers & Operations Research 36 (2009) 3215-3223.
40. Gambardella, L. M. & Taillard, E. & Agazzi, G. (1999). MACS – VRPTW: A multiple ant colony system for vehicle routing problems with time windows. Technical Report IDSIA, IDSIA-06-99, Lugano, Switzerland, 1999.

41. Gendreau, M. & Laporte, G. & Vigo, D. (1999). Heuristics for the travelling salesman problem with pick up and delivery. Computers and Operations Research, 26, 699-714.
42. Gilmour, S. & Dras, M. (2005). Understanding the Pheromone System within Ant Colony Optimization. Australasian Joint Conference on Artificial Intelligence, 2005.
43. Goncalves, F. & Cardoso, S.R. & Relvas, S. & Barbosa-Povoa, A.P.F.D. Optimization of a distribution network using electric vehices. A VRP problem. Technical report, CEG-IST, UTL, Lisboa, 2011.
44. Halse, K. (1992). Modeling and solving complex vehicle routing problems. Unpublished PhD thesis, Department of Mathematical Modeling, Technical University of Denmark.
45. Hiermann, G. & Puchinger, J. & Hartl, R. F. (2014). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. Department of Business Administration, University of Vienna, Austria.
46. Ichoua, S. & Gendreau, M. & Potvin, J-Y. (2007). Planned route optimization for real-time vehicle routing. In: Dynamic Fleet Management. Springer, pp, 1-18.
47. Jabir, E. & Vinay Panicker, V. & Sridharan, R. (2017). Design and development of a hybrid ant colony – variable neighbourhood search algorithm for a multi – depot green vehicle routing problem. Transportation Research Part D 56 (2017) 422-457
48. Jaw, J-J & Odoni, A.R. & Psaraftis, H.N. & Wilson, N.H. (1986). A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transport. Res. Part B: Methodol.20(3),243-257.
49. Jiaqi Ma & Xiaopeng Li & Fang Zhou & Wei Hao (2017). Designing optimal autonomous vehicle sharing and reservation systems: A linear programming approach. Transportation Research Part C 84 (2017) 124-141.
50. Kok, A.L. & Hans, E. & Schtutten, J. (2012). Vehicle routing under time-dependent travel times: the impact of congestion avoidance. Comput.Oper.Res. 39 (5), 910-918.
51. Krueger, R. & Rashidi, T.H. & Rose, J.M. (2016). Preferences for shared autonomous vehicles. Transport. Res. Part C: Emerg. Technol. 69, 343-355.
52. Laporte, G., (1992). The vehicle routing problem: an overview of exact and approximate algorithms. Eur. J. Oper. Res. 59 (3), 345-358
53. Levin, M. W. (2017). Congestion – aware system optimal route choice for shared autonomous vehicles. Transportation Research Part C 82 (2017) 229-247.
54. Levin, M.W. & Boyles, S.D. (2015). Intersection auctions and reservation-based control in dynamic traffic assignment. Transport. Res. Rec.: Transport. Res. Board (2497), 35-44.

55. Levin, M.W. & Boyles, S.D., (2016). A cell transmission model for dynamic lane reversal with autonomous vehicles. Transport. Res. Part C: Emerg. Technol. 68. 126-143.
56. Lin, C. & Choy, K.L. & Ho, G.T. & Chung, S. & Lam, H. (2014). Survey of green vehicle routing problem: past and future trends. Expert Syst. Appl. 41 (4), 1118-1138.
57. Melachrinoudis, E. & Ilhan, A.B. & Min, H. (2007). A dial-a-ride problem for client transportation in a health-care organization. Comput.Oper.Res.34(3), 742-759.
58. Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. Transportation Research A., 23, 377-386.
59. Mohan, B.C. & Baskaran, R. (2012). A survey: Ant colony optimization based recent research and implementation on several engineering domain. Expert. Syst. Appl. 39, 4618-4627.
60. Mullen, R.J. & Monekosso, D. & Barman, S. & Remagnino, P. (2009). A review of ant algorithms. Expert. Syst. Appl. 36, 9608-9617.
61. Nagy, G. & Salhi, S., (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. European Journal of Operational Research, 34, 336-344.
62. Neto, R.F.T. & Filbo, M.G. (2013). Literature review regarding ant colony optimization applied to scheduling problems: Guidelines for implementation and directions for future research. Eng. Appl. Artif. Intell. 26, 150-161.
63. Parragh, S.N. & Cordeau, J-F. & Doerner, K.F. & Hartl, R.F. (2012). Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. OR Spectrum 34(3), 593-633.
64. Pavone, M. & Smith, S.I. & Frazzoli, E. & Rus, D. (2012). Robotic load balancing for mobility-on-demand systems. Int.J.Robot. Res. 31 (7), 839-854.
65. Pedemonte, M. & Nesmachnow, S. & Canela, H. (2011). A survey on parallel ant colony optimization. Appl. Soft. Comput. 11,5181-5197.
66. Qiu, Y. & Liu, H. & Wang, D. & Liu, X. Intelligent strategy on coordinated charging of PHEV with TOU price. In Power and Energy Engineering Conference APPEEC 2011 AsiaPacific, pages 1-5, 2011.
67. Ropke, S. & Cordeau, J. F. (2009). Branch-and-Cut-and-Price for the Pickup and Delivery Problem with Time Windows. Transportation Science, 43(3), 267 -286.
68. Ropke, S. & Pisinger, D. (2006). A unified heuristic for vehicle routing problems with backhauls. European Journal of Operational Research, 171, 750-775.

69. Saenphon, T. & Phimoltares, S. & Lursinsap, C. (2014). Combining new fast opposite gradient search with ant colony optimization for solving travelling salesman problem. Eng. Appl. Artif. Intell. 35, 324-334.
70. Salhi, S. & Nagy, G. (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. Journal of the Operational Research Society, 50, 1034-1042.
71. Sinha, K.C., (2003). Sustainability and urban public transportation. J. Transp. Eng. 129 (4), 331-341.
72. Solnon, C. & Bridge, D. (2006). An ant colony optimization meta-heuristic for subset selection problems. System Engineering using Particle Swarm Optimization. Nova Science publisher, pp.7-29.
73. Spieser, K. & Treleaven, K. & Zhang, R. & Frazzoli, E & Morton, D. & Pavone, M. (2014). Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: a case study in singapore. In: Road Vehicle Automation. Springer, pp. 229-245.
74. Stuetzle, T. & HOOS, HH. Max-min ant system. Future Generation Computer Systems 2000; 16: 889-914.
75. Sundar, K. & Venkatachalam, S. & Rathinam, S. (2017). An Exact Algorithm for a Fuel-Constrained Autonomous Vehicle Path Planning Problem.
76. Tarantilis CD, Zachariadis EE, Kiranoudis CT (2008). A hybrid guided local search for the vehicle-routing problem with intermediate replenishment facilities. INFORMS Journal on Computing 20(1): 154-168.
77. Vural, A. V. & Catay, B. & Eksioglu, B. (2005). A dual GA approach to capacitated vehicle routing problem with simultaneous pick-up and deliveries. In Proceedings of the IIE research conference, Atlanta, GA (CD-ROM).
78. Wang, H. & Shen, J. Heuristic approaches for solving transit vehicle scheduling problem with route and fueling time constraints. Applied Mathematics and Computation, 190 (2): 1237-1249, 2007.
79. Wang, Y.-W. & Lin, C.-C. (2009). Locating road-vehicle refueling stations. Transportation Research Part E: Logistics and Transportation Review, 45 (5): 821-829, 200.9.
80. Wang, Y.-W. & Wang, C.-R. Locating passenger vehicle refueling stations. Transportation Research Part E: Logistics and Transportation Reviews, 46 (5): 791-901, 2010.

81. Wassan, N. A. & Nagy, G. (2014). Vehicle routing problem with deliveries and pickups: Modelling issues and meta-heuristics solution approaches. International Journal of Transportation Vol.2, No.1 (2014), pp.95-110.
82. Wassan, N.A. & Wassan, A.H. & Nagy, G. (2008). A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries. Journal of Combinatorial Optimization, 15, 368-386.
83. Xiang, Z. & Chu, C. & Chen, H. (2008). The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments. Eur.J.Oper.Res. 185 (2), 534-551.
84. Yap, M.D. & Correia, G. & van Arem, B. (2016). Preferences of travelers for using automated vehicles as last mile public transport of multimodal train trips. Transport. Res. Part A: Policy Pract. 94, 1-16.
85. Yeqian Lin & Wenquan Li & Feng Qiu & He Xu (2012). Reasearch on Optimization of Vehicle Routing Problem for Ride-sharing Taxi. 8th International Conference on Traffic and Transportation Studies Changsha, China, August 1-3, 2012. Procedia- Social and Behavioral Sciences 32 (2012) 494-502.
86. Yousefikhoshbakht, M. & Khorram, E. (2012).Solving the vehicle routing problem by a hybrid meta-heuristic algorithm . Yousefikhoshbakht and Khorram Journal of Industrial Engineering International 2012, 8:11.
87. Yu, B. & Yang, Z.Z. & Yao, B. (2009). An improved ant colony optimization for vehicle routing problem. Eur. J. Oper. Res. 196 (1), 171-176.
88. Zaho, N. & Wu, Z. & Zhao, Y. & Quan, T. (2010). Ant colony optimization algorithm with mutation mechanism and its applications.
89. Zhu, M. & Liu, X.-Y. & Tang, F. & Qiu, M. & Shen, W. & Shu, W. & Wu, M.-Y. (2016). Public vehicles for future urban transportation. IEEE Trans. Intell. Transport. Syst. 17 (12), 3344-3353.

ΠΑΡΑΡΤΗΜΑ

ΚΩΔΙΚΑΣ PYTHON

```
from __future__ import division
import numpy as np
import math
import random
import copy

maxdist=29000
seed=9
random.seed(seed)

cust1=[]
ind=0
while ind<30: # 30 epivates
    b=(ind+1)
    c=random.randint(1,30)
    if b!=c:
             cust1.append([b,c])
             ind=ind+1

dist=[[0,2600,1500,3400,3500,3500,5600,4100,5900,6400,3800,7300,7700,7500,50
00,5300,1700,1300,1000,3300,5000,5400,5400,1000,3700,4300,2600,6800,6200,230
0,2700],
[2600,0,4000,6100,3300,5200,7800,4200,4200,3700,1200,6000,5300,9600,3300,300
0,1600,3000,3500,1900,3500,7500,8300,3700,6900,3500,4200,8800,4100,3900,5000
],
[1500,4000,0,3200,3400,3500,5500,4100,5900,7900,4800,8000,9200,5300,6600,530
0,2500,1300,2400,5400,6800,6100,3800,1400,2700,4300,2500,6800,6100,2200,2600
],
[3400,6100,3200,0,5800,5900,7900,6500,8300,5800,6600,5300,6800,5500,4400,760
0,4100,4300,2300,2700,3700,3400,4300,1800,2900,6700,4900,9200,11900,4600,500
0],
[3500,3300,3400,5800,0,2500,5100,1200,3000,5300,3600,11200,10200,8700,5700,2
200,2300,2200,3900,8300,6800,9500,7200,4100,6400,1300,3100,4700,3100,1300,20
00],
[3500,5200,3500,5900,2500,0,2300,3200,5000,7800,5400,10400,9800,12600,7700,4
700,3100,2500,4600,6000,7600,9800,9700,4300,9700,3700,2500,3300,5600,1500,12
00],
[5600,7800,5500,7900,5100,2300,0,5700,4400,11800,7500,12600,11800,16300,9500
,5900,5600,4900,7100,8500,10100,12200,12100,6800,12100,4500,5000,1400,5700,3
900,3700],
[4100,4200,4100,6500,1200,3200,5700,0,2000,6700,4800,10000,9100,13300,6900,3
500,4200,4000,5800,5900,7500,11800,14400,5900,8200,1900,4900,2800,3400,3100,
3900],
[5900,4200,5900,8300,3000,5000,4400,2000,0,6500,4600,10800,8900,10600,6700,3
100,4300,4200,5900,5700,7200,11600,9100,6000,8300,1700,5100,3600,2200,3200,4
000],
[6400,3700,7900,5800,5300,7800,11800,6700,6500,0,1900,7200,3400,10500,2100,3
700,3800,5300,5100,3100,4700,10500,9500,6100,8100,4700,6500,8500,5000,5700,6
500],
[3800,1200,4800,6600,3600,5400,7500,4800,4600,1900,0,8500,6700,11500,4200,18
00,3400,5100,7200,4400,6000,9400,10000,5400,8800,2700,6000,6800,3000,4100,49
00],
[7300,6000,8000,5300,11200,10400,12600,10000,10800,7200,8500,0,6400,5100,510
0,9100,6300,7700,5400,3400,3200,3600,6900,6500,7900,10100,8900,11100,12600,8
600,9300],
```

```
[7700,5300,9200,6800,10200,9800,11800,9100,8900,3400,6700,6400,0,10000,2900,
8800,6100,7500,5700,4200,3500,9400,11100,6800,8700,8900,8500,10500,9100,8200
,8600],
[7500,9600,5300,5500,8700,12600,16300,13300,10600,10500,11500,5100,10000,0,9
000,11800,10200,6100,7200,7300,7100,2000,1800,5000,2800,9100,7300,14400,1260
0,7000,9400],
[5000,3300,6600,4400,5700,7700,9500,6900,6700,2100,4200,5100,2900,9000,0,460
0,5900,4600,3000,1000,2600,8500,7400,4100,6000,5100,5800,8000,9300,5500,6900
],
[5300,3000,5300,7600,2200,4700,5900,3500,3100,3700,1800,9100,8800,11800,4600
,0,3700,5100,9900,7900,5700,10100,13400,15800,12900,1500,6300,5300,1500,3800
,5000],
[1700,1600,2500,4100,2300,3100,5600,4200,4300,3800,3400,6300,6100,10200,3900
,3700,0,1400,2100,3500,4500,6500,6300,2100,5400,3500,2600,6900,5300,2300,280
0],
[1300,3000,1300,4300,2200,2500,4900,4000,4200,5300,5100,7700,7500,6100,4600,
5100,1400,0,2100,4400,6100,6400,5100,1800,3800,3200,1400,5700,5100,1100,1600
],
[1000,3500,2400,2300,3900,4600,7100,5800,5900,5100,7200,5400,5700,7200,3000,
9900,2100,2100,0,2100,3700,5300,6100,1100,4700,5700,3900,8400,7500,3600,4100
],
[3300,1900,5400,2700,8300,6000,8500,5900,5700,3100,4400,3400,4200,7300,1000,
7900,3500,4400,2100,0,1600,6000,6500,3100,5000,4700,5800,7700,5300,5500,6000
],
[5000,3500,6800,3700,6800,7600,10100,7500,7200,4700,6000,3200,3500,7100,2600
,5700,4500,6100,3700,1600,0,4300,7600,6700,8200,9700,9200,11400,12900,8900,1
2600],
[5400,7500,6100,3400,9500,9800,12200,11800,11600,10500,9400,3600,9400,2000,8
500,10100,6500,6400,5300,6000,4300,0,3800,4800,4400,12100,7900,13100,10600,7
500,8100],
[5400,8300,3800,4300,7200,9700,12100,14400,9100,9500,10000,6900,11100,1800,7
400,13400,6300,5100,6100,6500,7600,3800,0,4300,1500,8100,6600,13200,13300,60
00,8100],
[1000,3700,1400,1800,4100,4300,6800,5900,6000,6100,5400,6500,6800,5000,4100,
5800,2100,1800,1100,3100,6700,4800,4300,0,2900,5100,3300,7500,6800,2900,3400
],
[3700,6900,2700,2900,6400,9700,12100,8200,8300,8100,8800,7900,8700,2800,6000
,12900,5400,3800,4700,5000,8200,4400,1500,2900,0,6600,4400,12100,8500,4600,5
000],
[4300,3500,4300,6700,1300,3700,4500,1900,1700,4700,2700,10100,8900,9100,5100
,1500,3500,3200,5700,4700,9700,12100,8100,5100,6600,0,4500,3800,1800,2700,35
00],
[2600,4200,2500,3900,3100,2500,5000,4900,5100,6500,6000,8900,8500,7300,5800,
6300,2600,1400,3900,5800,9200,7900,6600,3300,4400,4500,0,5900,5500,1700,1400
],
[6800,8800,6800,9200,4700,3300,1400,2800,3600,8500,6800,11100,10500,14400,80
00,5300,6900,5700,8400,7700,11400,13100,13200,7500,12100,3800,5900,0,4800,44
00,4100],
[6200,4100,6100,11900,3100,5600,5700,3400,2200,5000,3000,12600,9100,12600,93
00,1500,5300,5100,7500,5300,12900,10600,13300,6800,8500,1800,5500,4800,0,460
0,6000],
[2300,3900,2200,4600,1300,1500,3900,3100,3200,5700,4100,8600,8200,7000,5500,
3800,2300,1100,3600,5500,8900,7500,6000,2900,4600,2700,1700,4400,4600,0,800]
,
[2700,5000,2600,5000,2000,1200,3700,3900,4000,6500,4900,9300,8600,9400,6900,
5000,2800,1600,4100,6000,12600,8100,8100,3400,5000,3500,1400,4100,6000,800,0
]]


all_costs = []
nodes=range(len(dist))
vehicles=range(len(dist)) # vehicles=number of nodes
```

```
#nearest neighbour solution
def nnt(graph,startNode):

  # graph: 2D array with numberOfNodes rows and columns and the weight of the
edges as values.
  # startNode: the node where the tour begins and ends.
  # returns a list containing the nearest-neighbour-tour.

    tour = [startNode]
    remNodes = range(len(graph)) # remaining nodes
    remNodes.remove(startNode)
    curNode = startNode # current node
    sum1=0#elegxos gia to kausimo
    for n in range(len(graph)-1):
        dist = [ (graph[curNode][i],i) for i in remNodes] # list with length
from curNode to i in the form (length, i)
        sum1=sum1+min(dist)[0]
        remNodes.remove(min(dist)[1]) # remove node that is nearest to curNode
        tour.append(min(dist)[1]) # append node that is nearest to curNode
        curNode = min(dist)[1] # set node that is nearest to curNode as curNode
        sum1=sum1+min(dist)[0]
    tour.append(startNode) # append the starting node to end of tour (so we
have a cycle)
    return tour




#gets the lengths of each tour
def gtl(graph,tour):

    length = 0
    for i in range(len(tour)-1):
        length += graph[tour[i]][tour[i+1]]
    return length

nntour=nnt(dist,0) # nearest neighbour solution
Z=gtl(dist,nntour)# length of nearest neighbour solution


L=10000000000000000  # initiliaze min length


#initiliaze pheromones (1/dist)
t=[[0 for xx in range(len(dist))] for yy in range(len(dist))]
for i in range(len(dist)):
    for j in range(len(dist)):
        if dist[i][j]!=0:
            t[i][j]=1/dist[i][j]
        else:
            t[i][j]=0

t_init = [[x for x in y] for y in t] #keep initial values of pheromones needed
for updates

#closeness (1/d)
n=[[0 for xx in range(len(dist))] for yy in range(len(dist))]
for i in range(len(dist)):
    for j in range(len(dist)):
            if dist[i][j]!=0:
```

```
                n[i][j]= 1/dist[i][j]
            else:
                n[i][j]=0

b=0.5 #weighs the relative importance of the heuristic value
#calculate n^b
n1=[[0 for xx in range(len(dist))] for yy in range(len(dist))]
for i in range(len(dist)):
    for j in range(len(dist)):
        n1[i][j]=math.pow(n[i][j],b)
Ponom=np.multiply(t,n1)

#initialize Propabilities
P=[[0.0 for xx in range(len(dist))] for yy in range(len(dist))]


#update Propabilities
def Prop(graph,i,candidate_nodes,Pr,t,n1):

    prcopy = Pr.copy()
    pp=0

    for j in candidate_nodes:
        pp+=(t[i][j]*n1[i][j])

    for k in candidate_nodes:
        prcopy[i][k]=(t[i][k]*n1[i][k])/pp
    return prcopy

r=0.5# value for pheromone evaporation
q0=0.9

demcopy=np.zeros((31,31))
for i in range(len(cust1)):
    k=cust1[i][0]
    j=cust1[i][1]
    demcopy[k][j]=random.randint(1,4)

#local update of pheromones
def localupdate(pheromones,lastnode,currentnode,Tinitial,r):
    nval=(1-r)*pheromones[lastnode][currentnode]+r*t_init[lastnode][current-
node]
    pheromones[lastnode][currentnode]=nval
    return pheromones


#global update of pheromones
def globalupdate(pheromones,lastnode,currentnode,length_of_shortest_tour):
    nval=(1-r)*pheromones[lastnode][currentnode]+r*(1/length_of_short-
est_tour)
    pheromones[lastnode][currentnode]=nval
    return pheromones

P=np.array(P)

load=np.array([[0 for _ in range(len(nodes))] for _ in range(len(vehicles))])#
This keeps track of passengers IN each vehicle AFTER leaving node i

def choosenextnode(current,available_nodes,Propab,load,dem):
    index=-1 #index for finding max
    k=-1#hypothetical next node
    available_nodes=np.array(available_nodes)
```

```
    if  len(available_nodes)>0 : # if reachable is not empty
        q=random.random()
        if q>=0.9:
            k=random.choice(available_nodes)

        else:
            for l in (available_nodes):
                if Propab[current][l]>index:
                    index=Propab[current][l]
                    k=l
        return k
    else: # if no any other node can be visited
        return 0

#finds the nodes tha can be visited
def         remaining_nodes(current,dist,nodes,Propab,km,cap,load,dem,custom-
ers,veh,tour):# IF FUEL IS ENOUGH AND VEH IS NOT FULL NODE IS REACHABLE
    a=[]

    if current==0:
              for i in range(1,30):
                     if     dist[current][i]+dist[i][0]+km<=maxdist     and
sum(dem[i][:])>0:
                            a.append(i)          # if there is at least one
cust departing from i


    else:
              for i in range(len(dist)):
                  if i!=current:
                      if  dist[current][i]+dist[i][cust[i-1][1]]+dist[cust[i-
1][1]][0]+km<=maxdist and (sum(dem[i][:])>0 ) and load[current][veh]<cap and
cust[i-1][1] not in tour:# i 3i sinthiki deixnei oti xwraei na parei toulax-
iston 1 akoma
                         a.append(i)
    a=[x for x in a if x not in tour]
    return a

#each time finds next node and updates load, demand and km spent
def                           findload(load,current,cap,dist,km,tour,veh,ex-
istingnode,reach,dem,P,cust,demcopy,demand_node):#current  here  is  o:  o  has
been added to the route, but we must decide load

    #existingnode is the previous, current is the one we are at now but need
to find the pickup load and node after is the next one, the destination of
our pickup
    #Load for each vehicle is the load when it departs from the specific node
    #f i lista me tous komvous proorismou pou prepei na afisw pelates

    nodeafter=choosenextnode(current,reach,P,load,dem)
    loadcopy=copy.deepcopy(load)

    if       loadcopy[existingnode][veh]+dem[current][nodeafter]-demcopy[ex-
istingnode][current]<=cap:
        loadcopy[current][veh]=loadcopy[existingnode][veh]+dem[cur-
rent][nodeafter] -demcopy[existingnode][current] #load for all vehicles?
        tobedelivered=  dem[current][nodeafter]
        dem[current][nodeafter]=0
        km=km+dist[current][nodeafter]
        tour.append(nodeafter)
```

```
    else:
        dem[current][nodeafter]=dem[current][nodeafter]-cap
        tobedelivered=  cap-load[existingnode][veh]
        loadcopy[current][veh]=cap
        km=km+dist[current][nodeafter]
        tour.append(nodeafter)


    if dem[current][nodeafter]<0:
        dem[current][nodeafter]=0
    return tour,dem,km,loadcopy

#resets load,demand and customers
def reset(vehicles):
    load=np.array([[0 for _ in range(len(vehicles))] for _ in range(len(vehi-
cles))])
    cust=cust1
    dem=demcopy.copy() # copy of dem so as not to change
    return cust,load,dem,demcopy


# global update of the first solution (nearest neighbour solution)
listzeugi1=[]
for f in range(len(nntour)-1):
    listzeugi1.append([nntour[f],nntour[f+1]])
for i11 in range(len(nodes)):
    for j11 in range(len(nodes)):
        if [i11,j11]  in listzeugi1:
            t=globalupdate(t_init,i11,j11,Z/100)

#Finds possible destinations of current node
def destinations(dests,cust,dem,tour):
    dests=[]
    for kk in range(len(cust)):
        if dem[tour[-1]][kk]>0:
            if kk not in dests:
                dests.append(kk)
    return dests

J=Z #initiliaze best total distance
mincost=1000000000

costval=[]# store cost for each iteration
for lll in range(5000):
    costoftour=[]
    cost = 0
    cust,load,dem,demcopy=reset(vehicles)
    tours=[]
    for ii in vehicles:
        dests=[]
        cap=4#max capacity
        load[0][ii]=0
        tour=[0]
        km=0
        existingnode=0
        reachable=remaining_nodes(ex-
istingnode,dist,nodes,P,km,cap,load,dem,cust,ii,tour)# first time is empty

        P=Prop(dist,0,reachable,P,t,n1)
        nextnode=choosenextnode(existingnode,reachable,P,load,dem)
        t=localupdate(t,0,nextnode,t_init,r)
        tour.append(nextnode)
```

```
        if nextnode==0:
            break
        km=dist[0][nextnode]
        lastnode=nextnode
        dests=destinations(dests,cust,dem,tour)
        while dests:
            for dest in dests:
                if dest!=tour[-1]:
                    P=Prop(dist,tour[-1],dests,P,t,n1)

                    tour,dem,km,load=findload(load,last-
node,cap,dist,km,tour,ii,0,dests,dem,P,cust,demcopy,1)
                    t=localupdate(t,tour[-2],tour[-1],t_init,r)

                    dests.remove(tour[-1])
        while km<=maxdist:

            lastnode=tour[-1]

            reachable=remaining_nodes(tour[-
1],dist,nodes,P,km,cap,load,dem,cust,ii,tour)

            P=Prop(dist,lastnode,reachable,P,t,n1)
            tour,dem,km,load=findload(load,lastnode,4,dist,km,tour,ii,tour[-
2],reachable,dem,P,cust,demcopy,0)
            t=localupdate(t,tour[-2],tour[-1],t_init,r)
            dests=destinations(dests,cust,dem,tour)

            if not reachable or np.all(dem==0) : #if there is no available
node or all demand is satisfied

                km=km+dist[tour[-1]][0]
                tour.append(0)
                break
            else:
                P=Prop(dist,tour[-1],dests,P,t,n1)

                tour,dem,km,load=findload(load,tour[-
1],cap,dist,km,tour,ii,tour[-2],dests,dem,P,cust,demcopy,1)
                t=localupdate(t,tour[-2],tour[-1],t_init,r)
                nextnode=lastnode
        if tour[-1]==tour[-2]:
            del tour[-1]
        tours.append(tour)

#Ride Sharing
    for pair in cust:
            node=pair[0]
            nodenext=pair[1]
            for route in tours:
               if node in route and nodenext in route:
                   if route.index(node)<route.index(nodenext):
                       for route2 in tours:
                           if route2!=route:
                               if node in route2 and nodenext in route2:
                                   if  route2.index(node)<route2.index(noden-
ext) and len(route2)==4:
                                       if   load[node][tours.index(route)]==0
and route.index(nodenext)-route.index(node)<=2:  #has dropped
                                           if           load[route[route.in-
dex(node)+1]][tours.index(route)]+demcopy[node][nodenext]<=cap:  ###222222  #
if does not interfere with next
```

```
                                                     load[node][tours.in-
dex(route)]=demcopy[node][nodenext]
                                                     load[route[route.in-
dex(node)+1]][tours.index(route)] +=demcopy[node][nodenext]
                                                     load[node][tours.in-
dex(route2)] = 0
                                                     tours.remove(route2)
    #GLobal update of each solution
    bestmikos=gtl(dist,tours[0])
    for tour in tours:
        listzeugi=[]
        for f in range(len(tour)-1):
            listzeugi.append([tour[f],tour[f+1]])
            mikostour=gtl(dist,tour)
            if mikostour < bestmikos:
                bestmikos = mikostour
                besttour = listzeugi

    for i2 in range(len(nodes)):
        for j2 in range(len(nodes)):
            if [i2,j2] in listzeugi:
                t=globalupdate(t,i2,j2,bestmikos)
    tours2=[x for y in tours for x in y]
    costoftour.append(gtl(dist,tours2))
    cost=sum(costoftour)
    all_costs.append(cost)
    if cost<mincost:
        mincost=cost
        J=mincost
        besttours=tours
        bestload=load
        print "Minimum cost found", mincost

    costval.append(mincost)

for i in range(len(besttours)):
    print "Tour of Vehicle No" + str(i+1) + " with route: ",  besttours[i]


bigest_tour=(gtl(dist,besttours[0])/50000)*60+(len(besttours[0])-
2)/5#minutes
smallest_tour=(gtl(dist,besttours[0])/50000)*60+(len(besttours[0])-
2)/5#minutes
for i in (besttours):
    a=(gtl(dist,i)/50000)*60+(len(i)-2)/5
    if a<smallest_tour:
        smallest_tour=a
    if a>bigest_tour:
        bigest_tour=a

print "Duration of the smallest tour is", smallest_tour,"minutes", "and dura-
tion of the biggest tour is", bigest_tour, "minutes"
import matplotlib.pyplot as plt
plt.plot(costval)
```