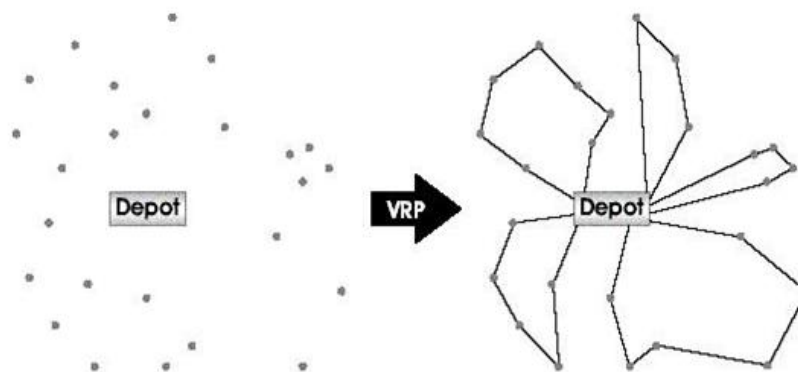




ΣΧΟΛΗ ΧΗΜΙΚΩΝ ΜΗΧΑΝΙΚΩΝ  
ΤΟΜΕΑΣ ΙΙ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ  
*ΕΠΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΔΡΟΜΟΛΟΓΗΣΗΣ ΣΤΟΛΟΥ ΟΧΗΜΑΤΩΝ ΜΕ*  
*ΕΦΑΡΜΟΓΕΣ ΣΤΙΣ ΜΕΤΡΗΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ ΚΟΙΝΩΦΕΛΩΝ*  
*ΟΡΓΑΝΙΣΜΩΝ*



ΚΟΡΟΓΙΑΝΝΟΥ ΛΟΥΚΙΑ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Χ.ΚΥΡΑΝΟΥΔΗΣ

*Αθήνα, Οκτώβριος 2011*

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέποντα την εργασία μου κ. Κυρανούδη Χρήστο για την πολύτιμη βοήθεια και καθοδήγησή του. Η βοήθεια αυτή συνέβαλε στη σωστή κατανόηση και μελέτη του θέματος. Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου για την κάθε είδους συμπαράσταση κατά την εκτέλεση της εργασίας αυτής αλλά και καθ' όλη τη διάρκεια των σπουδών μου. Τέλος ένα μεγάλο ευχαριστώ στον Λιάκο Κώστα για την πολύτιμη βοήθειά του στην εκπόνηση του θέματος.

## ΠΕΡΙΛΗΨΗ

Στην παρούσα διπλωματική εργασία εξετάστηκε η επίλυση του προβλήματος στόλου οχημάτων, με εφαρμογές στις μετρητικές διαδικασίες κοινωφελών οργανισμών. Προκειμένου να βρεθεί η βέλτιστη λύση, απαιτείται μεγάλο υπολογιστικό κόστος, χρονικά ασύμφορο. Για το λόγο αυτό χρησιμοποιούνται ευρεστικοί ή και μεταευρεστικοί αλγόριθμοι. Ο υπολογιστικός πυρήνας του αλγορίθμου που χρησιμοποιήθηκε είναι ένας ευρεστικός αλγόριθμος βελτιστοποίησης (*insertion*). Ουσιαστικό χαρακτηριστικό του αλγορίθμου αυτού είναι ότι βελτιστοποιεί τις διαδρομές συλλογής των απαιτούμενων μετρήσεων (*collection routes*). Ο κώδικας προγραμματίστηκε σύμφωνα με τη μεθοδολογία του αντικειμενοστραφούς προγραμματισμού (*OOP*) σε *Visual Basic 6.0*. Στο συγκεκριμένο πρόβλημα, έγινε εφαρμογή του κώδικα σε 4 διαφορετικά παραδείγματα, 2500 σημείων στο πλήθος το καθένα. Αρχικά, στο 1<sup>ο</sup> και 3<sup>ο</sup> παράδειγμα εφαρμόστηκε ο κώδικας για την περίπτωση χρήσης οχήματος με ταχύτητα 40 km/h. Ενώ στο 2<sup>ο</sup> και 4<sup>ο</sup> αντίστοιχα παράδειγμα εφαρμόστηκε ο κώδικας για την περίπτωση βαδίσματος με ταχύτητα 4km/h. Εξήχθησαν αποτελέσματα με τη δυνατότητα σύγκρισής τους και εξαγωγής περαιτέρω συμπερασμάτων.

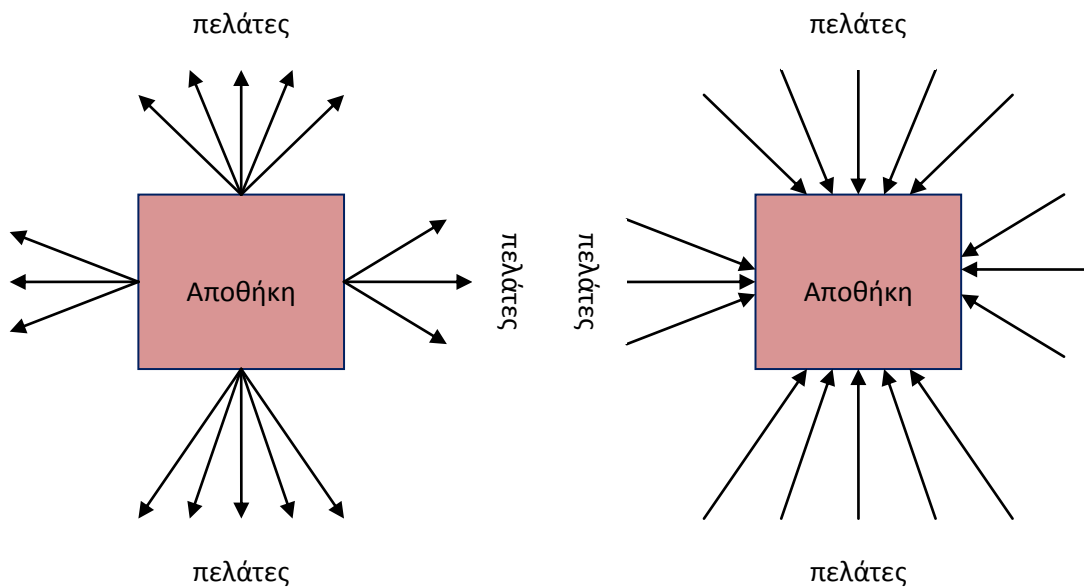
## Περιεχόμενα

ΟΡΙΣΜΟΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ .....	2
Η ΕΦΟΔΙΑΣΤΙΚΗ ΔΙΑΧΕΙΡΙΣΗ .....	3
Παράγοντες που επηρεάζουν τα Logistics .....	5
Η σημερινή εικόνα των Logistics στη χώρα μας.....	6
Οι παράγοντες που θα διαμορφώσουν το μέλλον των Logistics .....	7
ΔΡΟΜΟΛΟΓΗΣΗ.....	7
Δρομολόγηση Στόλου Οχημάτων.....	8
Συστήματα Διαχείρισης Στόλου Οχημάτων .....	10
Τηλεματική Διαχείριση Στόλου .....	11
Χαρακτηριστικά της Τηλεματικής .....	11
ΣΥΝΔΥΑΣΤΙΚΗ ΑΡΙΣΤΟΠΟΙΗΣΗ .....	13
Ένα Κλασικό Παράδειγμα: Traveling Salesman Problem.....	14
ΕΙΔΗ ΑΛΓΟΡΙΘΜΩΝ .....	14
Ακριβείς Αλγόριθμοι .....	14
Ευρεστικοί Αλγόριθμοι (Heuristics) .....	15
Αλγόριθμοι Τοπικής Έρευνας.....	15
Μεταευρεστικοί Αλγόριθμοι.....	16
Κατασκευαστικοί Ευρεστικοί .....	17
Κατασκευαστικές Μέθοδοι.....	17
Η ΓΛΩΣΣΑ VB – ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ (OOP).....	20
ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ.....	23
Βασική Πορεία Αλγορίθμου .....	23
Αποτελέσματα .....	24
Συμπεράσματα .....	50
Βιβλιογραφία .....	51
Παράρτημα.....	52

## ΟΡΙΣΜΟΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Το αντικείμενο της παρούσας εργασίας είναι η ανάπτυξη και υλοποίηση ενός ευρεστικού αλγορίθμου που ονομάζεται Ε.Σ.Δ. (Εύρεση Σύντομων Διαδρομών) για την επίλυση του προβλήματος της δρομολόγησης στόλου οχημάτων με εφαρμογές στις μετρητικές διαδικασίες κοινωφελών οργανισμών πχ. (ΔΕΗ, Φυσικό Αέριο, ΕΥΔΑΠ). Σκοπός της παρούσας εργασίας είναι η εφαρμογή σύγχρονων τεχνικών όπως οι ευρεστικοί αλγόριθμοι αριστοποίησης, ο αντικειμενοστραφής προγραμματισμός και τα γεωγραφικά συστήματα πληροφοριών στην αντιμετώπιση πραγματικών και επίκαιρων προβλημάτων εύρεσης βέλτιστων διαδρομών με εφαρμογή στις μετρητικές διαδικασίες κοινωφελών οργανισμών.

Η εφοδιαστική διαχείριση (logistics) ορίζεται ως η διαδικασία σχεδιασμού, υλοποίησης και ελέγχου της αποδοτικής και αποτελεσματικής κανονικής και αντίστροφης ροής και αποθήκευσης αγαθών, υπηρεσιών και σχετικών πληροφοριών από το σημείο προέλευσης μέχρι το σημείο κατανάλωσης με σκοπό την ικανοποίηση των απαιτήσεων των πελατών Στο συγκεκριμένο πρόβλημα Ε.Σ.Δ. η ροή χαρακτηρίζεται ως «αντίστροφη». Ο χαρακτηρισμός «αντίστροφη» δικαιολογείται από το γεγονός ότι η παρεχόμενη υπηρεσία στους πελάτες-δημότες δεν είναι η διανομή ενός προϊόντος όπως στις συνήθεις διανομές του ιδιωτικού τομέα αλλά η λήψη μετρήσεων από τους μετρητές που υπάρχουν στις οικίες των πελατών της αντίστοιχης κάθε φοράς κοινωφελούς εταιρείας.



Σχήμα 1

Φυσική διανομή αγαθών από μίας αποθήκη προς πλήθος πελατών

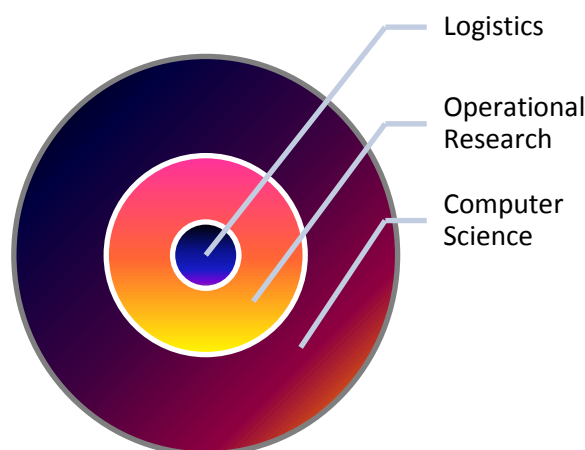
Συλλογή μετρήσεων κοινωφελών οργανισμών

## Η ΕΦΟΔΙΑΣΤΙΚΗ ΔΙΑΧΕΙΡΙΣΗ

Επιχειρησιακή Έρευνα (Operational Research) είναι ο επιστημονικός κλάδος που ασχολείται με την ανάπτυξη μαθηματικών μοντέλων, για την περιγραφή συστημάτων και διαδικασιών με κύριο σκοπό την αριστοποίησή τους και τη λήψη αποφάσεων. Ο όρος Επιχείρηση (Operations), έχει την έννοια της διαδικασίας, λειτουργίας και όχι της εταιρίας.

Η Εφοδιαστική Διαχείριση (*logistics*) είναι ένας τομέας της επιχειρησιακής έρευνας (*operations research*) που εμφανίζει εντυπωσιακή ανάπτυξη τα τελευταία χρόνια σε ετερόκλητους τομείς της οικονομίας όπως οι μεταφορές, οι τηλεπικοινωνίες, η βιομηχανία, η ηλεκτρονική, ο στρατός, η δημόσια διοίκηση κλπ.

Η θέση των *logistics* ως υποσύνολου της επιχειρησιακής έρευνας φαίνεται εποπτικά στο Σχήμα 2. Όπως προκύπτει από το σχήμα, η επιχειρησιακή έρευνα μπορεί να θεωρηθεί η ίδια ως υποσύνολο της επιστήμης των υπολογιστών, αν και είναι σαφής ο διεπιστημονικός της χαρακτήρας.



Σχήμα 2. Τα *logistics* ως υποσύνολο της επιχειρησιακής έρευνας και της επιστήμης των υπολογιστών.

Ο όρος «επιχειρησιακή έρευνα» ορίστηκε για πρώτη φορά κατά το Β΄ Παγκόσμιο Πόλεμο, όταν οι συμμαχικές δυνάμεις ώθησαν τη μελέτη πολλών προβλημάτων βελτιστοποίησης της επιχειρησιακής τους δραστηριότητας, όπως η βέλτιστη τοποθέτηση των *radar*, η διαχείριση πολεμικού υλικού, η διάταξη νηοπομπών, ο προγραμματισμός βομβαρδισμών κλπ. Οι στρατιωτικές εφαρμογές αποτέλεσαν την ώθηση ανάπτυξης του κλάδου και την επόμενη δεκαετία μετά τον πόλεμο με χαρακτηριστική τη συμβολή του ινστιτούτου *Rand Corporation* στις Η.Π.Α. Σημαντική ώθηση έδωσε επίσης το διαστημικό πρόγραμμα των Ηνωμένων Πολιτειών. Κατά τις δεκαετίες του 1960 και 1970 οι τεχνικές της επιχειρησιακής

έρευνας επεκτάθηκαν ραγδαία από τις στρατιωτικές σε πολιτικές εφαρμογές, όπως η διοίκηση επιχειρήσεων και μεγάλων κρατικών συστημάτων π.χ. ηλεκτρισμού και μεταφορών καθώς και ο προγραμματισμός του Κράτους. Τη δεκαετία του 1980 παρατηρήθηκε ευρύτατη εκλαΐκευση της επιχειρησιακής έρευνας χάρη στη διάδοση των μικρο-υπολογιστικών συστημάτων. Τα τελευταία χρόνια πραγματοποιείται μία γενίκευση της χρήσης της Επιχειρησιακής Έρευνας παράλληλα με τα Εφαρμοσμένα Μαθηματικά, την Αριθμητική Ανάλυση και την Τεχνητή Νοημοσύνη. Στη διάρκεια της δεκαετίας του 1990 νέα ώθηση στον κλασικό μαθηματικό προγραμματισμό (*mathematical programming*) έδωσαν οι ευρεστικοί αλγόριθμοι αριστοποίησης (*heuristics*) και κατόπιν οι μετα-ευρεστικοί αλγόριθμοι (*metaheuristics*), η εξέλιξη των οποίων αποτελεί ένα σημαντικό επίκεντρο της σημερινής έρευνας στην επιχειρησιακή έρευνα. Οι ενδιαφέροντες νέοι αλγόριθμοι βρίσκουν εφαρμογή σε προβλήματα συνδυαστικής αριστοποίησης μεγάλου μεγέθους όπως είναι κατά κανόνα τα σύγχρονα προβλήματα *logistics*, ειδικότερα στον τομέα των μεταφορών με χρήση G.I.S.

Κατά καιρούς έχουν δοθεί πολλοί ορισμοί του όρου «Logistics» ή αλλιώς «Εφοδιαστική», ένας εκ των οποίων είναι ο παρακάτω, όπως προσδιορίζεται από μία από τις μεγαλύτερες ομάδες ειδικών στα Logistics, το Council of Logistics Management (CLM) των Ηνωμένων Πολιτειών: η «εφοδιαστική διαχείριση» (*logistics*) ορίζεται, ως η διαδικασία σχεδιασμού, υλοποίησης και ελέγχου της αποδοτικής και αποτελεσματικής κανονικής και αντίστροφης ροής, αποθήκευσης αγαθών, υπηρεσιών και σχετικών πληροφοριών από το σημείο προέλευσης μέχρι το σημείο κατανάλωσης, με σκοπό την ικανοποίηση των απαιτήσεων των πελατών.

Αξίζει να σημειωθεί ότι γενικά παρατηρείται μία σύγχυση σχετικά με το περιεχόμενο και τη σύνδεση των εννοιών «Logistics» και «Supply Chain Management». Αυτό οφείλεται στην οπτική γωνία των εμπλεκομένων. Μερικοί, όπως ο Simchi-Levi, δε διακρίνουν καμία διαφορά μεταξύ των «Logistics» και «Supply Chain Management» και θεωρούν ότι ο όρος Supply Chain αποτελεί συνώνυμο του Logistics. Άλλοι, όπως το Council of Logistics Management (CLM) και οι Rushton, υποστηρίζουν ότι τα Logistics είναι τμήμα ή υποσύνολο των διαδικασιών σε μία εφοδιαστική αλυσίδα. Οι διαφορετικές αυτές απόψεις οφείλονται στο γεγονός ότι πολλές επιχειρήσεις (ακόμη και μεταξύ των διευθύνσεων στην ίδια επιχείρηση) δεν έχουν ίδιες αντιλήψεις σχετικά με την έννοια του όρου Supply Chain. Το πιθανότερο όμως είναι ότι η σωστή αντίληψη του όρου δεν είναι στατική, αλλά αλλάζει με το χρόνο, καθώς αλλάζουν οι συνθήκες και εντείνεται ο ανταγωνισμός.

Ο ορισμός των Logistics δε δηλώνει ότι τα Logistics προσδιορίζουν τις απαιτήσεις, ούτε ότι είναι μηχανικός σχεδιασμός, ούτε ότι είναι διοίκηση. Δηλώνει μόνο ότι όταν μία προκαθορισμένη διοίκηση, συγκεκριμένη τεχνική μεθοδολογία και συγκεκριμένες τεχνικές δραστηριότητες εμπλέκονται με ειδικές λειτουργίες υποστήριξης, τότε ο συνδυασμός των παραγόντων αυτών αποτελεί εφαρμογή των Logistics.

## Παράγοντες που επηρεάζουν τα Logistics

Η εφαρμογή των Logistics έχει διαφορετικούς τρόπους δράσης και διαφορετικά αποτελέσματα κατά περίπτωση, που εξαρτώνται από το περιβάλλον μέσα στο οποίο αναπτύσσονται και τους παράγοντες που το επηρεάζουν.

- Η οικονομική κατάσταση
- Η πολιτική κατάσταση
- Το κοινωνικό καθεστώς
- Το μορφωτικό επίπεδο
- Το ηθικό περιβάλλον
- Το τεχνολογικό περιβάλλον
- Το νομικό καθεστώς
- Το φυσικό περιβάλλον

Εκείνος όμως ο παράγοντας που έχει τη μεγαλύτερη βαρύτητα είναι ο ανθρώπινος παράγοντας και η ανθρώπινη λογική, που αν είναι αλόγιστη, μπορεί να έχει σαν αποτέλεσμα όχι μόνο την αποτυχία του οποιουδήποτε εγχειρήματος, αλλά και την καταστροφή του ίδιου του ανθρώπου και κατ' επέκταση της ίδιας της φύσης.

Σε ένα κλασικό σύστημα παραγωγής, η εφοδιαστική διαχείριση επικεντρώνεται στα στάδια της διαχείρισης των προμηθειών, του προγραμματισμού παραγωγής, της διοίκησης αποθεμάτων και του σχεδιασμού της διανομής των προϊόντων. Ενδεικτικές περιοχές εφαρμογών των Logistics περιλαμβάνουν τα: Business Logistics, Systems Logistics, Maritime Logistics, Logistics Υγείας, Logistics Στρατού, Περιβαλλοντικά Logistics, City Logistics, Crisis Logistics, Logistics Υπηρεσιών, Agro-Logistics και Reverse Logistics.

Τα Logistics βρίσκουν εφαρμογή σε δύο τομείς:

- Ο πρώτος τομέας είναι η επιχείρηση, η οποία πρέπει να οργανώσει την εισροή, την εσωτερική διακίνηση και την εκροή υλικών και προϊόντων κατά τέτοιο τρόπο, έτσι ώστε να εξασφαλίζει τη μέγιστη ικανοποίηση των πελατών της.
- Ο δεύτερος τομέας είναι η εφοδιαστική αλυσίδα, η οποία αποτελείται από όλες εκείνες τις επιχειρήσεις και οργανισμούς που είναι απαραίτητοι έτσι ώστε ένα προϊόν, από πρώτες ύλες, να καταλήξει στον τελικό πελάτη. Η αποτελεσματική οργάνωση και διοίκηση της ροής προϊόντων και πληροφοριών σε αυτήν την αλυσίδα αποτελεί επιτακτική ανάγκη σε μία παγκοσμιοποιημένη και ψηφιακή οικονομία, όπου ο ανταγωνισμός από ατομικός (επιχείρηση εναντίον επιχείρησης) γίνεται συλλογικός (εφοδιαστική αλυσίδα εναντίον εφοδιαστικής αλυσίδας).

Τα Logistics αφορούν σε:

Απαιτήσεις: Οι δραστηριότητες των Logistics εμπλέκονται με την ανάλυση, σύνθεση και καθορισμό των πόρων που απαιτούνται για να επιτύχουμε ένα σκοπό ή να



φέρουμε σε πέρας μία επιχείρηση κάτω από συγκεκριμένες συνθήκες. Ο συνολικός στόχος του να καθορίσουμε απαιτήσεις, είναι μία λειτουργία σχεδιασμού που εμπλέκει ταυτόχρονα και τη στρατηγική και τα Logistics. Ο καταμερισμός των κυρίων διαθέσιμων πόρων, αν είναι λιγότεροι από τους απαιτούμενους και η αξιολόγηση του αποτελέσματος των ελλείψεων για την επίτευξη των κυρίων στόχων, είναι κύριες ευθύνες της στρατηγικής και όχι λειτουργία των Logistics.

Σχεδιασμός: Αυτή η λειτουργία περιλαμβάνει όλο το πλάνο του σχεδιασμού, μέσα από λεπτομερή σχεδιασμό των προϊόντων, συστημάτων και υπηρεσιών, συμπεριλαμβανομένων της ανάπτυξης, δοκιμής και αξιολόγησης του σχεδιασμού. Το Logistics Engineering αφορά στο σχεδιασμό του εφοδιασμού και της συντήρησης, κάτω από το πρίσμα της αποτελεσματικότητας κόστους, σε αντίθεση με το σχεδιασμό της εύκολης παραγωγής ή χρήσης.

Εφοδιασμός: Αυτή η περιοχή εμπλέκει το φυσικό εφοδιασμό και διανομή όλων των διαθέσιμων πόρων, π.χ. προμήθειες, πρόσληψη και εκπαίδευση προσωπικού, υποστήριξη παραγωγής, συσκευασία, Διοίκηση Αποθεμάτων, διακίνηση και μεταφορές, διαδικασία παραγγελιών, αποθήκευση, αποσύρσεις, κλπ. Υπάρχουν λειτουργίες που δημιουργούν χρονική και χωροταξική χρησιμότητα, σε αντίθεση με τις λειτουργίες παραγωγής που χρησιμοποιούν χρησιμότητα τυποποίησης και τις λειτουργίες του marketing που δημιουργούν χρησιμότητα ιδιοκτησίας.

Συντήρηση: Η συντήρηση εκλαμβάνεται ευρέως ως η διατήρηση των εγκαταστάσεων, προϊόντων, ανθρώπινου δυναμικού, συστημάτων και υπηρεσιών των παραγωγών και χρηστών, συμπεριλαμβανομένης της προστασίας, διατήρησης και ανάκτησης όλων των διατιθέμενων πόρων.

Πόρους: Πρώτες ύλες (υλικά), εξοπλισμός, εγκαταστάσεις, προσωπικό, συμπεριλαμβανομένων των κεφαλαίων και πληροφοριών. Τα Logistics συχνά συνδέονται με τη διοίκηση των υλικών, όμως οι τεχνικές της διοίκησης των υλικών μπορούν επίσης να εφαρμοσθούν στη διοίκηση του ανθρώπινου δυναμικού, χρημάτων και πληροφοριών.

## **Η σημερινή εικόνα των Logistics στη χώρα μας**

Σήμερα η κατάσταση έχει διαμορφωθεί κατά τρόπο που να παρουσιάζεται η ακόλουθη εικόνα:

- Ο μεγάλες εταιρίες του ιδιωτικού τομέα έχουν αντιληφθεί την αναγκαιότητα των Logistics και κινούνται αποφασιστικά στην υλοποίηση οργανωτικών μεταβολών και στη δημιουργία, εκπόνηση ολοκληρωμένων διαδικασιών Logistics.
- Ο ευρύτερος δημόσιος τομέας αδυνατεί να συλλάβει την έννοια των Logistics και για το λόγο αυτό υπάρχουν μεγάλες δημόσιες επιχειρήσεις και οργανισμοί οι οποίοι λειτουργούν εντελώς αντιπαραγωγικά σε θέματα και λειτουργίες Logistics (π.χ. ΟΤΕ).

- Στις Ένοπλες Δυνάμεις -αν και ήταν πρωτοπόροι του κλάδου- υπάρχει μια ορατή στασιμότητα που προκαλεί σοβαρά προβλήματα στην επιχειρησιακή ετοιμότητά τους.
- Στις μικρομεσαίες ιδιωτικές επιχειρήσεις η εικόνα είναι απογοητευτική με ελάχιστες εξαιρέσεις.
- Στην ακαδημαϊκή κοινότητα τα Logistics αναφέρονται και διδάσκονται πρόχειρα και αποσπασματικά. Κάποιες προσπάθειες που άρχισαν με σωστούς ποιοτικούς οραματισμούς, εξελίχθηκαν σε κακέκτυπα διεθνών προτύπων.

## **Οι παράγοντες που θα διαμορφώσουν το μέλλον των Logistics**

Το μέλλον των Logistics θα εξαρτηθεί από:

- Τις υποδομές στην παιδεία σε όλα τα επίπεδα και στην ανάπτυξη και λειτουργία ποιοτικών προγραμμάτων, που θα καλύπτουν όλες τις εκφάνσεις των Logistics σε όσο το δυνατόν περισσότερα πανεπιστήμια, για να υπάρξει ο σχετικός ανταγωνισμός που οδηγεί στη βελτίωση της ποιότητας.
- Την ύπαρξη επιχειρηματιών με όραμα, που θα είναι σε θέση να αντιληφθούν τα πλεονεκτήματα των Logistics και των ολοκληρωμένων και όχι αποσπασματικών εφαρμογών.
- Τη λειτουργία ιδιωτικών πανεπιστημιακών οργανισμών που θα προσδώσουν άλλη δυναμική εκπαίδευσης στην πανεπιστημιακή κοινότητα.
- Την περαιτέρω δραστηριοποίηση των επιστημονικών εταιριών του κλάδου.
- Την αποδοχή της αναγκαιότητας εφαρμογής των Logistics σε κυβερνητικούς και λοιπούς οργανισμούς του ευρύτερου δημόσιου τομέα.
- Την αναβάθμιση των Logistics στις ελληνικές ένοπλες δυνάμεις
- Την ανάπτυξη πληροφοριακών συστημάτων Logistics από ελληνικές επιχειρήσεις που θα εξασκούν πίεση για εφαρμογή τους στην επιχειρηματική κοινότητα.

## **ΔΡΟΜΟΛΟΓΗΣΗ**

Δρομολόγηση, ονομάζεται η κατάστρωση και σχεδίαση της βέλτιστης διαδρομής σε ένα γράφο του οποίου οι ακμές χαρακτηρίζονται από κόστη. Ανάλογα με το πρόβλημα, κόστη μπορεί να είναι χρονικές καθυστερήσεις, ή χιλιομετρικές αποστάσεις. Σκοπός των προβλημάτων δρομολόγησης είναι να βρεθεί η βέλτιστη διαδρομή που οδηγεί από τον ένα κόμβο στον άλλο, μειώνοντας το κόστος στο ελάχιστο δυνατό. Συνεπώς, τα προβλήματα δρομολόγησης συνιστούν ουσιαστικά προβλήματα βελτιστοποίησης. Πράγματι, οι αλγόριθμοι και οι τεχνικές που χρησιμοποιούνται για την εύρεση της βέλτιστης διαδρομής, απαντώνται σε προβλήματα συνδυαστικής βελτιστοποίησης. Προκειμένου να γίνει κατανοητή η έννοια της δρομολόγησης, δίνεται ένα απτό παράδειγμα που σχετίζεται με το αντικείμενο που εξετάζουμε παρακάτω. Ένας γράφος λέγεται συνεκτικός, όταν για κάθε ζεύγος κορυφών του  $u, v$  υπάρχει ένα έγκυρο μονοπάτι που οδηγεί από την  $u$  στη

ν. Ένας συνεκτικός γράφος του οποίου οι ακμές φέρουν κόστη λέγεται δίκτυο. Ένα πρόβλημα που μπορεί να τεθεί σε ένα δίκτυο είναι το εξής: δεδομένου ενός αρχικού κόμβου να οδηγηθούμε σε έναν άλλον κόμβο, επιλέγοντας εκείνο το μονοπάτι που φέρει το ελάχιστο αθροιστικό κόστος επί των ακμών του γράφου, που οδηγούν διαδοχικά από τον αρχικό στον τελικό κόμβο προορισμού. Το πρόβλημα αυτό, της εύρεσης του βέλτιστου μονοπατιού, αποτελεί το γνωστό Πρόβλημα του Συντομότερου Μονοπατιού (Shortest Path Problem), σημείο αναφοράς για σχεδόν όλα τα προβλήματα δρομολόγησης.

Η δρομολόγηση είναι μία έννοια που αφορά οποιοδήποτε δίκτυο, είτε αυτό πρόκειται για οδικό δίκτυο είτε για δίκτυο υπολογιστών. Αξίζει να αναφέρουμε ορισμένες βασικές εφαρμογές της δρομολόγησης πάνω σε γνωστές τεχνολογίες, όπως π.χ. τα Δίκτυα Υπολογιστών, το Παγκόσμιο Σύστημα Θεσιθεσίας (GPS), η Πλοήγηση (Navigation), και τέλος η Διαχείριση Στόλου Οχημάτων (Fleet Management) που αποτελεί το αντικείμενο της εργασίας μας.

## **Δρομολόγηση Στόλου Οχημάτων**

Τα προβλήματα δρομολόγησης στόλου οχημάτων VRP (Vehicle Routing Problem) αναφέρονται σε μία κατηγορία προβλημάτων προσδιορισμού διαδρομών κάθε οχήματος από ένα δοσμένο στόλο οχημάτων που ξεκινούν από μία ή περισσότερες αφετηρίες (depot) με σκοπό την επίσκεψη διαφόρων τοποθεσιών, γεωγραφικά διεσπαρμένων σε ένα γεωγραφικό δίκτυο, για λόγους διανομής. Ο στόχος των VRP είναι η ολοκλήρωση της επίσκεψης/εξυπηρέτησης των σημείων ζήτησης με το ελάχιστο μήκος (κόστος) του αθροίσματος των διαδρομών των οχημάτων που συμμετείχαν στη διαδικασία διανομής/συλλογής.

Στο συγκεκριμένο πρόβλημα, τα οχήματα ορισμένων επιχειρήσεων διακινούνται μεταξύ των πελατών της κάθε επιχείρησης, ξεκινώντας από ένα ή περισσότερους κεντρικούς μέσω ενός οδικού δικτύου. Στα οχήματα αυτά αντιστοιχεί ένα συγκεκριμένο πλήρωμα και μπορούν να εξυπηρετήσουν, ανάλογα με τη χωρητικότητά τους, μεγάλο αριθμό πελατών της επιχείρησης. Η διακίνηση γίνεται μέσω ενός πραγματικού οδικού δικτύου, στο οποίο η κυκλοφοριακή κίνηση μπορεί να επηρεάσει σημαντικά τις αποφάσεις που σχετίζονται με το πρόβλημα. Έτσι, η δρομολόγηση των οχημάτων των επιχειρήσεων αυτών είναι ένα σύνθετο πρόβλημα μαθηματικού προγραμματισμού, με αντικειμενική συνάρτηση το κόστος μεταφοράς και τους περιορισμούς εξυπηρέτησης που κάθε φορά ορίζονται από τους πελάτες, καθώς και τις ιδιαιτερότητες αυτών των περιορισμών. Σε όλες τις περιπτώσεις το ενδιαφέρον εστιάζεται στην αντιμετώπιση του προβλήματος μέσω μεταφοράς και διανομής, τόσο σε επίπεδο επιχειρησιακό όσο και σε επίπεδο στρατηγικού σχεδιασμού.

Οι αντικειμενικοί στόχοι του προβλήματος είναι η ελαχιστοποίηση του λειτουργικού κόστους που προκύπτει από τη δρομολόγηση των οχημάτων και του προγραμματισμού της εργασίας των πληρωμάτων τους. Η επίτευξη αυτών των στόχων έχει σαν αποτέλεσμα:

- Τον προσδιορισμό του ακριβούς αριθμού των εμπλεκόμενων οχημάτων και την καταγραφή των αντίστοιχων ποσοτήτων.
- Τον προσδιορισμό του ελάχιστου αριθμού των απαιτούμενων οχημάτων.
- Τον προσδιορισμό των βέλτιστων διαδρομών που θα ακολουθήσουν τα οχήματα δια μέσου του οδικού δικτύου.
- Τη βέλτιστη οργάνωση εργασίας των πληρωμάτων τους.

Να σημειωθεί πως σε κάθε περίπτωση οι στόχοι αυτοί θα πρέπει να ικανοποιούνται ταυτόχρονα. Κατά συνέπεια, το προσφερόμενο λογισμικό αφορά σε ένα σύστημα διαχείρισης και συντονισμού του στόλου και του προσωπικού που θα είναι υπεύθυνο για τη δρομολόγηση του στόλου των οχημάτων, με στόχο τη βέλτιστη δυνατή οργάνωση από πλευράς κόστους και ασφάλειας μεταφοράς και το οποίο:

- Χρησιμοποιεί στο υπολογιστικό περιβάλλον πραγματικά λειτουργικά δεδομένα που σχετίζονται με το κύκλωμα διανομής και δίνονται από τον πελάτη.
- Προτείνει εκείνα τα δρομολόγια που έχουν χαμηλότερο κόστος για την εταιρία, λαμβάνοντας υπόψη το υφιστάμενο οδικό δίκτυο.
- Βελτιώνει το επίπεδο εξυπηρέτησης των υπηρεσιών.
- Αξιοποιεί στο μέγιστο τις δυνατότητες και την αποδοτικότητα των αυτοκινήτων και του προσωπικού.
- Πραγματοποιεί τον καθημερινό προγραμματισμό της διανομής, ενώ υποστηρίζει και το στρατηγικό σχεδιασμό από πλευράς της επιχείρησης.
- Επιτυγχάνει τη μέγιστη δυνατή μείωση του κόστους διανομής.

Επιχειρήσεις που ωφελούνται με το παραπάνω σύστημα είναι όλες οι επιχειρήσεις που ασχολούνται με Logistics, οι εταιρίες διανομών και ταχυμεταφορών, οι μεταφορικές, καθώς επίσης και οι κοινωφελείς οργανισμοί, με εφαρμογή στις μετρητικές διαδικασίες, με τους οποίους και ασχολούμαστε στην παρούσα εργασία.

Στο παρελθόν τα συστήματα δρομολόγησης παρείχαν γενικές πληροφορίες. Ήταν χρήσιμα μόνο σε επίπεδο στρατηγικού σχεδιασμού και όχι στην πρακτική δραστηριοποίηση στο φορέα μεταφοράς. Στη σημερινή εποχή συνδέονται με συστήματα GIS (Geographical Information Systems) για τη δημιουργία αναλυτικών διαδρομών, υψηλής ακρίβειας και σε ελάχιστο χρόνο. Μέχρι τελευταία στην Ελλάδα, προσπάθειες για τη λειτουργία αυτών των εφαρμογών δεν έφεραν τα επιθυμητά αποτελέσματα, κατά ένα μεγάλο ποσοστό λόγω της απουσίας ενός GIS που θα πληρούσε τις απαραίτητες προϋποθέσεις.

Επενδύοντας σήμερα σε υψηλή τεχνολογία βέλτιστης δρομολόγησης, οι διαχειριστές του στόλου έχουν πλέον τη δυνατότητα να αξιοποιήσουν σε μέγιστο βαθμό τα οχήματα που διαθέτουν, καθώς και να ανεβάσουν τον πήχη στην εξυπηρέτηση των πελατών τους. Αποτελεσματικά, θα έχουν αυξημένη απόδοση δηλ. περισσότερα δρομολόγια με τον ίδιο αριθμό οχημάτων, πιο ισορροπημένα φορτία και άμεση επιστροφή των χρημάτων που επένδυσαν μέσω της οικονομίας αλλά και της αυξημένης παραγωγικότητας. Αξίζει να σημειωθεί, ότι η έννοια της βελτιστοποίησης

των δρομολογίων είναι υποκειμενική. Αυτό συμβαίνει καθώς κάθε διαχειριστής έχει διαφορετικές προτεραιότητες και στόχους που θέλει να ικανοποιήσει. Άλλοι ασχολούνται με τη μείωση του κόστους, ενώ άλλοι επικεντρώνονται στην καλύτερη εξυπηρέτηση των πελατών τους.

## **Συστήματα Διαχείρισης Στόλου Οχημάτων**

Τα συστήματα αυτά αποτελούν προεκτάσεις των συστημάτων βέλτιστης δρομολόγησης. Πρόκειται για μία κατηγορία ψηφιακών συστημάτων που λαμβάνουν αποφάσεις σχετικά με τις διαδρομές που πρέπει να ακολουθήσουν τα οχήματα ενός δοσμένου στόλου. Τα συστήματα διαχείρισης στόλου γνωστοποιούν με τη μέγιστη δυνατή ακρίβεια τις θέσεις των οχημάτων πάνω στο δίκτυο. Τα συστήματα αυτά αποτελούν ισχυρό εργαλείο, καθώς καταγράφουν και εποπτεύουν τη συνολική δραστηριότητα του στόλου καθ' όλη τη διάρκεια εκτέλεσης της δρομολόγησης. Τα συστήματα αυτά αποθηκεύουν ένα μεγάλο πλήθος πληροφοριών. Έτσι ο διαχειριστής συλλέγει μία πληθώρα αναφορών, που πριν ήταν αδύνατο:

- Λεπτομερείς αναφορές δρομολογίων.
- Αποδοτική χρήση οδηγών και φορτηγών στα δρομολόγια.
- Χρήση πόρων και κόστος αυτών.
- Σύνολο δρομολογίων με στοιχεία αποστάσεων, χρόνων, αριθμού πελατών, ποσοτήτων και κόστη τους.
- Επιστροφές και κόστη τους.
- Μετρήσεις παραγωγικότητας και αποδοτικότητας (συγκρίσεις πραγματικών σε σχέση με τα προϋπολογισμένα).
- Γραφικές απεικονίσεις των χρόνων οδήγησης, εξυπηρέτησης σημείων, διαλειμμάτων και κόστη τους.

Μέσα από τη χρήση των συστημάτων διαχείρισης στόλου (Fleet Management Systems), ο σημερινός επιχειρηματίας έχει τη δυνατότητα να βελτιώσει τη χρήση των οχημάτων του, να χρησιμοποιήσει πιο αποδοτικά το χρόνο ανθρώπων και οχημάτων και να διανύσει λιγότερα χιλιόμετρα με το ίδιο φορτίο. Όλα αυτά βοηθούν στη μείωση του λειτουργικού κόστους του, κάτι που μπορεί να μεταφραστεί σε αυξημένη κερδοφορία ή ανταγωνιστικότητα τιμών.

Τα πλεονεκτήματα που προκύπτουν από τη χρήση των συστημάτων διαχείρισης στόλου είναι πολλά. Αρχικά, σημαντική ποσότητα πληροφοριών που μία εταιρία αδυνατούσε να συγκεντρώσει, τώρα βρίσκεται στα χέρια της. Ταυτόχρονα, ένα σύστημα Fleet Management επιτελεί συν τοις άλλοις και το ρόλο ενός MIS (Management Information System). Η συγκέντρωση σημαντικών πληροφοριών έχει με τη σειρά της ως αποτέλεσμα να είναι εφικτή η παραγωγή αναφορών σχετικά με την πορεία της επιχείρησης, την αξιολόγηση του τρόπου λειτουργίας της, ενώ καταδεικνύει και τρόπους βελτίωσης και αποτελεσματικότερης αξιοποίησης της υπάρχουσας υποδομής. Οι πληροφορίες αυτές γίνονται περισσότερο σημαντικές σήμερα, αν αναλογιστεί κανείς το επίπεδο ανταγωνισμού στις σύγχρονες επιχειρήσεις. Σημαντικά είναι και τα οφέλη από την εξοικονόμηση πόρων που

αφορούν σε τηλεπικοινωνιακά κόστη. Η αξιοποίηση υπηρεσιών, όπως το SMS αλλά και όσες πηγάζουν από το GPRS, απαλλάσσει την επιχείρηση από υψηλούς τηλεπικοινωνιακούς λογαριασμούς, οι οποίοι προέρχονται από μεγάλο αριθμό κλήσεων προς κινητά και σταθερά τηλέφωνα των «εν κινήσει» εργαζομένων της.

## **Τηλεματική Διαχείριση Στόλου**

Για να αποτελέσουν τα συστήματα δρομολόγησης στόλου ολοκληρωμένες λύσεις για τον τομέα των logistics, θα πρέπει να συνδυάζονται με ένα σύστημα ασύρματης εποπτείας στόλου. Αυτή είναι η τεχνολογία του GPS (Global Positioning System). Αν τοποθετηθεί ένας δέκτης GPS σε ένα όχημα, μπορούμε να έχουμε απεικόνιση της ακριβούς θέσης του, με την προϋπόθεση ότι ο δέκτης λαμβάνει σήματα από 4 δορυφόρους ταυτόχρονα. Τα τελευταία χρόνια με την ανάπτυξη της τεχνολογίας έχει αυξηθεί θεαματικά η ακρίβεια του GPS στα επίπεδα των 3-5 μέτρων απόκλιση.

Τα στοιχεία της θέσης του οχήματος είναι αξιοποιήσιμα με τη βοήθεια μιας άλλης τεχνολογίας, αυτή της κινητής τηλεφωνίας εφοδιασμένης με τη δυνατότητα μετάδοσης data (GPRS). Τα δεδομένα της θέσης μεταφέρονται μέσα από GPRS πρωτόκολλα στον πελάτη ή σε κεντρικούς σταθμούς τηλεματικής, προσφέροντας έτσι σημαντικές δυνατότητες εποπτείας και ελέγχου του στόλου στους δρομολογητές.

Σήμερα, υπάρχει πλήθος επιλογών σε δέκτες GPS, που συνδυάζονται με μια κάρτα κινητής GPRS. Η τιμή διαφέρει ανάλογα με τις δυνατότητες της συσκευής αλλά και τις υπηρεσίες που παρέχονται. Το κόστος μειώνεται με την ανάπτυξη της τεχνολογίας, δεν παύει όμως να είναι σημαντικό όταν μιλάμε για στόλους 20,40 ή 100 οχημάτων.

## **Χαρακτηριστικά της Τηλεματικής:**

### **Θέση και κατάσταση των οχημάτων σε πραγματικό χρόνο**

- Υπάρχει άμεση εικόνα των οχημάτων ανά πάσα στιγμή.
- Παρακολούθηση των οχημάτων σε δυναμικούς χάρτες, όπου η θέση και οι λοιπές πληροφορίες ανανεώνονται αυτόματα (GPS tracker).
- Ενημέρωση για την ταχύτητα και την κατεύθυνση κίνησής τους, την κατάσταση λειτουργίας του κινητήρα και το χρόνο στάσης, αν το όχημα είναι σταματημένο.
- Προβολή της θέσης των πελατών στο χάρτη με τη λειτουργία αυτόματης φόρτωσης και εντοπισμός του πλησιέστερου πελάτη σε κάθε όχημα.
- Ομαδοποίηση των οχημάτων σε γεωγραφικούς τομείς ή σε ομάδες ανάλογα με το έργο που εκτελούν.
- Οι δυνατότητες διαχείρισης είναι απεριόριστες.

### **Πλήρες ιστορικό διαδρομών**

- Όλες οι πληροφορίες κίνησης των οχημάτων αποθηκεύονται και είναι διαθέσιμες για διαχείριση ανά πάσα στιγμή.

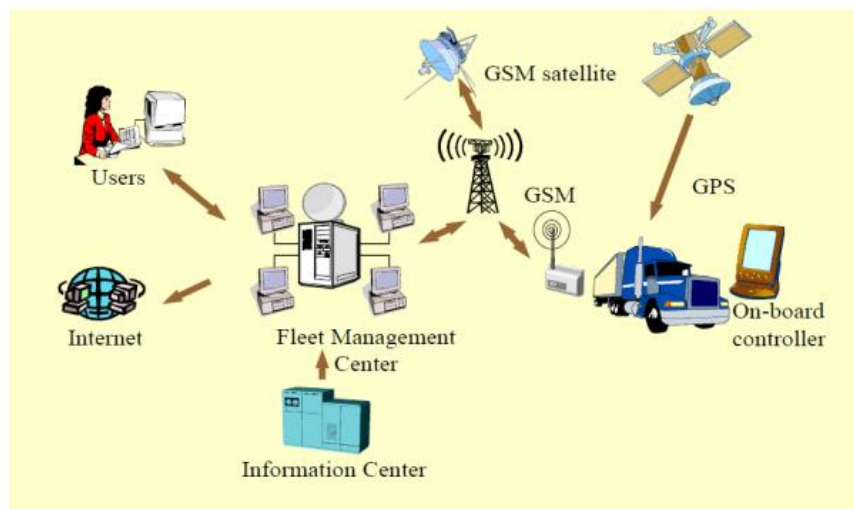
- Αναζήτηση δρομολογίων περασμένων ημερομηνιών, όπου για κάθε όχημα υπάρχει πλήρη ανάλυση κινήσεων, στάσεων και επισκέψεων σε πελάτες.
- Βεβαίωση ότι τηρήθηκε η προγραμματισμένη διαδρομή και ότι δεν υπήρχαν άσκοπες μετακινήσεις ή στάσεις.
- Υπάρχει ο πλήρης έλεγχος των οχημάτων στην οθόνη του υπολογιστή.

### Μέτρηση κατανάλωσης καυσίμου

Η επιχείρησή πολύ πιθανόν να δαπανά υψηλά ποσά σε καύσιμα, τα οποία δεν είναι σε θέση να ελέγξει πλήρως. Υπάρχει η δυνατότητα για πλήρη έλεγχο της κατανάλωσης καυσίμου του κάθε οχήματος, ώστε να υπάρχει η βεβαιότητα ότι δεν υπάρχουν αποκλίσεις από τις αναμενόμενες τιμές. Δεν χρειάζεται ειδική συσκευή, ούτε τρύπημα του δοχείου καυσίμων για τοποθέτηση επιπλέον αισθητήρα. Η πληροφορία της στάθμης καυσίμου παρέχεται από τον εργοστασιακό αισθητήρα που διαθέτουν όλα τα οχήματα. Τα δεδομένα επεξεργάζονται με ειδικές μαθηματικές μεθόδους, ώστε να επιτευχθεί η μέγιστη ακρίβεια.

### Αναφορές κίνησης

- Υπολογισμός σε ελάχιστο χρόνο των χιλιομέτρων που έχει διανύσει το κάθε όχημα ανά μήνα ή τρίμηνο, τη μέση κατανάλωση καυσίμου και τη μέγιστη ταχύτητα κίνησης.
- Αυτόματες ειδοποιήσεις.
- Υπάρχει η δυνατότητα ειδοποίησης με SMS ή e-mail, όταν ξεπεραστεί κάποιο από τα όρια που έχουν τεθεί.
- Υπάρχει η δυνατότητα να φαίνεται άμεσα ποιο όχημα ξεπερνά το όριο ταχύτητας, ποιο κινείται εκτός της επιτρεπτής περιοχής (geofencing), ή ποιο υπερβαίνει τον ανώτατο χρόνο στάσης.
- Επίσης, η καρτέλα οχήματος ειδοποιεί έγκαιρα για τα service, τις αλλαγές λαδιών ή ελαστικών, ώστε να γίνεται εύκολη η διαχείριση ακόμα και πολύ μεγάλων στόλων.



Σχήμα 3: Τηλεματική Διαχείριση Στόλου

## ΣΥΝΔΥΑΣΤΙΚΗ ΑΡΙΣΤΟΠΟΙΗΣΗ

Η συνδυαστική αριστοποίηση (*combinatorial optimization*) είναι ένα ενδιαφέρον πεδίο των εφαρμοσμένων μαθηματικών που συνδυάζει τεχνικές από τη συνδυαστική, το γραμμικό προγραμματισμό και τη θεωρία των αλγορίθμων για να λύσει προβλήματα αριστοποίησης σε διακριτές δομές. Ένα πρόβλημα συνδυαστικής αριστοποίησης συνήθως περιλαμβάνει διακριτές δομές, όπως ένα δίκτυο ή μία οικογένεια συνόλων μαζί με ένα σύνολο αριθμών, οι οποίοι μπορεί να αντιπροσωπεύουν π.χ. κόστη ή χωρητικότητες. Συχνά επομένως αναφερόμαστε στη συνδυαστική αριστοποίηση και σαν διακριτή αριστοποίηση (*discrete optimization*). Μία μορφοποίηση ενός τυπικού προβλήματος συνδυαστικής/διακριτής αριστοποίησης έχει ως εξής :

Με δεδομένα:

1. ένα διακριτό σύνολο λύσεων  $X$
2. και μία αντικειμενική συνάρτηση  $f(x): x \in X \rightarrow \mathbb{R}$

Ζητάμε να βρούμε

τη λύση  $x \in X: f(x) \leq f(y), \forall y \in X$  (για ελαχιστοποίηση)

ή τη λύση  $x \in X: f(x) \geq f(y), \forall y \in X$  (για μεγιστοποίηση)

Η επίλυση των προβλημάτων διακριτής αριστοποίησης από ακριβείς αλγορίθμους, είναι πρακτικά αδύνατη. Οι ακριβείς αλγόριθμοι εξετάζουν όλο το χώρο λύσεων του προβλήματος και επιλέγουν το παγκόσμιο βέλτιστο. Ένας ακριβής αλγόριθμος είναι βέβαιο ότι θα καταλήξει σε παγκόσμιο βέλτιστο, αλλά η εξέταση του συνόλου των λύσεων δημιουργεί σημαντικές απαιτήσεις σε υπολογιστικό χρόνο και σε μνήμη υπολογιστή.

Μία πιο μεθοδολογική προσέγγιση στα προβλήματα αριστοποίησης που είναι πολύ χρήσιμη στη συνδυαστική αριστοποίηση, περιγράφεται στη βιβλιογραφία. Σύμφωνα με αυτήν, ένα πρόβλημα αριστοποίησης αναφέρεται σε κάποιο σύνολο υποδειγμάτων, που εκφράζονται με κάποια κατάλληλη καλά καθορισμένη σύνταξη. Κάθε υπόδειγμα συνδέεται με ένα σύνολο λύσεων, έτσι ώστε κάθε λύση να αποκτά κάποια συγκεκριμένη τιμή, δοθέντος του υποδείματος. Η επίλυση του προβλήματος αριστοποίησης αναφέρεται στην απόδοση για κάθε υπόδειγμα της καλύτερης (βέλτιστης) λύσης, η οποία θα αναφέρεται στη μεγαλύτερη ή στη μικρότερη τιμή όλων των λύσεων που προκύπτουν από την περιγραφή του προβλήματος αριστοποίησης. Συγκεκριμένα, ένα πρόβλημα αριστοποίησης  $Q$  είναι μία τετράδα  $(I_Q, S_Q, f_Q, \text{opt}_Q)$ , όπου  $I_Q$  είναι το σύνολο των δυνατών υποδειγμάτων,  $S_Q$  μία συνάρτηση τέτοια ώστε για κάθε υπόδειγμα  $x \in I_Q$ , η  $S_Q(x)$  είναι ένα σύνολο λύσεων του  $Q$  για το  $x$ ,  $f_Q$  είναι η αντικειμενική συνάρτηση έτσι ώστε για κάθε ζεύγος  $x \in I_Q$  και  $y \in S_Q(x)$ , η ποσότητα  $f_Q(x,y)$  να είναι ένας πραγματικός αριθμός και ο δείκτης  $\text{opt}_Q \in \{\min, \max\}$  καθορίζει το πρόβλημα σαν πρόβλημα ελαχιστοποίησης ή μεγιστοποίησης.



## Ένα Κλασικό Παράδειγμα: Traveling Salesman Problem

Στη γενική μορφή του προβλήματος περιοδεύοντα πωλητή (*traveling salesman problem*) δίνεται ένα πεπερασμένο σύνολο σημείων  $V$  και ένα κόστος  $c_{uv}$  μετακίνησης μεταξύ κάθε ζεύγους  $u, v \in V$ . Το πρόβλημα απαιτεί να βρεθεί η διαδρομή ελάχιστους κόστους. Το πρόβλημα περιοδεύοντα πωλητή μπορεί να μοντελοποιηθεί σαν ένα πρόβλημα γράφων θεωρώντας έναν πλήρη γράφο  $G = (V, E)$  και αναθέτοντας σε κάθε ακμή  $uv \in E$  το κόστος  $c_{uv}$ . Μία διαδρομή είναι ένας πλήρης γύρος του  $G$  που περνά από κάθε κόμβο. Συχνά οι διαδρομές ονομάζονται και Χαμιλτονιανοί βρόχοι (*circuits*). Το TSP είναι από τα γνωστότερα προβλήματα συνδυαστικής αριστοποίησης. Αντίθετα με τις περιπτώσεις της αντιστοίχισης (*matching*) ή των ροών σε δίκτυα, δεν είναι γνωστός κανένας πολυωνυμικός αλγόριθμος για τη λύση του TSP γενικά. Πράγματι, το πρόβλημα περιοδεύοντα πωλητή ανήκει στην κλάση των NP-κοπιωδών (*NP-hard*) προβλημάτων. Πολλοί άνθρωποι μάλιστα πιστεύουν ότι δεν υπάρχει καμία πολυωνυμική μέθοδος επίλυσης, καθώς ένας τέτοιος αλγόριθμος θα υπονοούσε ότι θα μπορούσαμε να λύνουμε κυριολεκτικά κάθε πρόβλημα συνδυαστικής αριστοποίησης σε πολυωνυμικό χρόνο.

Ωστόσο, προβλήματα TSP προκύπτουν στην πράξη και σχετικά μεγάλα μεγέθη τους μπορούν σήμερα να λυθούν αποτελεσματικά ως την άριστη λύση. Στη βιβλιογραφία αναφέρονται αρκετοί ευρεστικοί αλγόριθμοι αριστοποίησης (*heuristics*) για το πρόβλημα TSP και επιδεικνύονται σε ένα σετ δεδομένων 1173 σημείων από τη βιβλιοθήκη TSPLIB. Οι κυριότεροι αλγόριθμοι κατασκευής διαδρομής (*tour construction*) είναι οι αλγόριθμοι κοντινότερου γείτονα (*nearest neighbor*), οι μέθοδοι μακρινότερης, κοντινότερης και φθηνότερης εισαγωγής (*farthest, nearest and cheapest insertion*) με προτεινόμενη την μακρινότερη εισαγωγή και το *Christofides' Heuristic* που χρησιμοποιεί στοιχεία από τη θεωρία των δένδρων. Οι κυριότεροι αλγόριθμοι βελτίωσης διαδρομής (*tour improvement*) είναι οι αλγόριθμοι *2-opt* και *3-opt*, γνωστοί και ως *2-exchange* ή *swaps* και *3-exchange* αντίστοιχα, η μέθοδος *Lin-Kernighan*, που είναι στην ουσία μία μέθοδος *k-opt* με  $\delta$ -μονοπάτια ( *$\delta$ -paths*) και η αλυσιδωτή μέθοδος *Lin-Kernighan*, που είναι μία τροποποίηση της *Lin-Kernighan* που χρησιμοποιεί *4-interchange* κτυπήματα (*kicks*) για να απεγκλωβιστεί από τοπικά βέλτιστες διαδρομές. Πρέπει να σημειωθεί ότι τα θεωρητικά όρια για τις μεθόδους κατασκευής είναι μικρής πρακτικής χρησιμότητας - αν εξαιρεθεί το *christofides' heuristic* - και επομένως αν θέλουμε να δώσουμε ένα όριο χειρότερης περίπτωσης στην απόδοση των υπόλοιπων αλγορίθμων κατασκευής, αυτό είναι τόσο γενικό που δεν έχει πρακτικό ενδιαφέρον ο υπολογισμός του.

## ΕΙΔΗ ΑΛΓΟΡΙΘΜΩΝ

### Ακριβείς Αλγόριθμοι

Προκειμένου να παραχθεί η καλύτερη δυνατή λύση (ο παγκόσμιος άριστος), είναι απαραίτητο να χρησιμοποιηθεί αλγόριθμος, ο οποίος θα βρει όλες τις δυνατές λύσεις

του προβλήματος, ώστε να εντοπιστεί αυτή που θα αριστοποιήσει την αντικειμενική συνάρτηση. Αν και οι αλγόριθμοι αυτοί είναι πολύ χρήσιμοι σε εύκολα προβλήματα, όταν τίθεται θέμα εφαρμογής σε προβλήματα αυξημένης πολυπλοκότητας, όπως το παρόν, είναι ασύμφοροι. Το υπολογιστικό κόστος, ο χρόνος δηλαδή που απαιτείται για να βρεθεί η βέλτιστη λύση, είναι τεράστιο, ενώ η απαιτούμενη μνήμη υπερβολική. Για τους παραπάνω λόγους, οι ακριβείς αλγόριθμοι δε χρησιμοποιούνται για τη λύση των VRPTW αυτούσιοι, αλλά μόνο σε συνδυασμό με μεθόδους οι οποίες μειώνουν τον αριθμό των δυνατών τιμών κάθε μεταβλητής.

## Ευρεστικοί Αλγόριθμοι (Heuristics)

Οι ευρεστικοί αλγόριθμοι είναι οι σημαντικότεροι προσεγγιστικοί αλγόριθμοι στη συνδυαστική αριστοποίηση. Πρόκειται για στρατηγικές που έχουν ως βασικό στόχο τη μείωση του πλήθους των εξεταζομένων λύσεων για την εύρεση του παγκόσμιου βέλτιστου, με προφανή εξοικονόμηση χρόνου και μνήμης. Το βασικό μειονέκτημα των ευρεστικών αλγορίθμων είναι ότι καταλήγουν στην καλύτερη λύση από αυτές που εξετάζονται, δηλαδή σε ένα τοπικό βέλτιστο, το οποίο βεβαίως είναι πιθανό να είναι και παγκόσμιο βέλτιστο, αν και αυτό δεν είναι δυνατόν να αποδειχτεί. Το τεράστιό τους πλεονέκτημα όμως έγκειται στην ικανότητά τους να χειριστούν προβλήματα με τεράστια υποδείγματα και αστρονομικό αριθμό πιθανών λύσεων. Το μειονέκτημα του εγκλωβισμού σε τοπικά ακρότατα επιλύεται με την υιοθέτηση των μετα-ευρεστικών αλγορίθμων.

Χαρακτηριστικό των *heuristics* - όπως και των *metaheuristics* - είναι η ενιαία λογική των αλγορίθμων αριστοποίησης, ανεξάρτητα από το ειδικό πρόβλημα αριστοποίησης στο οποίο κάθε αλγόριθμος εφαρμόζεται. Κυριότερος αντιπρόσωπος των *heuristics* με γενική εφαρμογή σε πλήθος προβλημάτων θεωρείται η τοπική έρευνα (*local search*). Ως μία διαδικασία επαναλαμβανόμενης τοπικής έρευνας του χώρου λύσεων από πολλαπλά σημεία εκκίνησης με προσαρμογή της έρευνας στα αποτελέσματά της μπορεί να θεωρηθεί η μέθοδος GRASP (*Greedy Randomized Adaptive Search Procedure*). Η μέθοδος GRASP κινείται στο μεταίχμιο μεταξύ *heuristics* και *metaheuristics*.

## Αλγόριθμοι Τοπικής Έρευνας

Οι τεχνικές τοπικής έρευνας αποτελούν μια ευρύτατη κλάση υπολογιστικών αλγορίθμων ευρεστικής φύσης που επιλύουν προβλήματα διακριτής αριστοποίησης. Ο αριθμός των εφικτών λύσεων αν και πολύ μεγάλος είναι πεπερασμένος. Η τεχνική που χρησιμοποιείται είναι η έρευνα σε βάθος στο σύνολο των εφικτών λύσεων  $S$ , ώστε να βρεθούν τοπικά ακρότατα της αντικειμενικής συνάρτησης. Στο τέλος κάθε επανάληψης δημιουργείται μια καινούρια λύση, η οποία προκύπτει από την προηγούμενη, και ανήκει στο γενικό σύνολο λύσεων  $S$ . Με άλλα λόγια οι νέες αυτές λύσεις ανήκουν στη *γειτονιά* των προηγούμενων λύσεων. Αυτό σημαίνει πως οι παραγόμενες λύσεις είναι συγγενικές με τις αρχικές και άρα μεταφέρουν τη μυωπική συμπεριφορά τους. Για να τερματιστεί η επαναληπτική διαδικασία ορίζονται κριτήρια

αποδοχής τελικής λύσης. Ένα σύνθημα φαινόμενο των αλγορίθμων τοπικής έρευνας είναι ο σύντομος τερματισμός τους λόγω της παγίδευσης τους σε περιοχές τοπικών ακρότατων.

## Μεταερευνητικοί Αλγόριθμοι

Οι ευρεστικοί αλγόριθμοι, αν και μειώνουν σημαντικά τον υπολογιστικό χρόνο επίλυσης, παρουσιάζουν δύο βασικά μειονεκτήματα: πρώτον, δεν εξάγουν ικανοποιητικά αποτελέσματα, εγκλωβίζονται σε τοπικά ελάχιστα και τερματίζονται, επομένως οι λύσεις τους είναι σπανίως υψηλής ποιότητας. Δεύτερον, δεν έχουν γενική εφαρμογή σε προβλήματα διακριτής βελτιστοποίησης, επειδή κατασκευάζονται με βάση τους περιορισμούς κάθε προβλήματος.

Σε αυτά τα αδιέξοδα, η λύση έρχεται από έναν νέο τύπο αλγορίθμων, ο οποίος επιχειρεί να συνδυάσει βασικές αρχές ευρεστικών μεθόδων (κατασκευαστικών και βελτιωτικών) με ένα υψηλό επίπεδο πλαισίου λειτουργίας, με σκοπό την εξερεύνηση του χώρου των λύσεων. Ουσιαστικά πρόκειται για στρατηγικές που επιλέγονται με τέτοιο τρόπο, ώστε να επιτυγχάνεται μια δυναμική ισορροπία ανάμεσα στην εκμετάλλευση της συσσωρευμένης εμπειρίας της έρευνας και της εξερεύνησης του χώρου λύσεων. Ο χώρος λύσεων που επιλέγεται έχει διπλή σημασία: περιέχει το δυνατόν υψηλότερης ποιότητας λύσεις, ενώ ταυτόχρονα αποκλείει περιοχές οι οποίες δεν έχουν ενδιαφέρον είτε ως αδύνατες, είτε ως «κακές» λύσεις. Η έρευνα που διεξάγεται από ένα μεταερευνητικό αλγόριθμο πρέπει να είναι αρκετά ευφυής ώστε να εντατικοποιείται αφενός στις περιοχές με υψηλής ποιότητας λύσεις (*μηχανισμός εντατικοποίησης*) και αφετέρου να μετακινείται σε ανεξερεύνητες περιοχές του χώρου των λύσεων όταν κρίνεται απαραίτητο (*μηχανισμός διαφοροποίησης*). Η φιλοσοφία των μεταερευνητικών τεχνικών, μπορούμε να πούμε πως ακολουθεί τα εξής βήματα:

1. Δημιούργησε μία αρχική λύση X
2. Αποτίμησε το κόστος της X
3. Δημιούργησε μία νέα λύση Y στη γειτονιά της X
4. Αποτίμησε το κόστος της Y
5. Αν Κόστος (Y) < Κόστος (X) θέσε X=Y και πήγαινε στο βήμα 3
6. Τέλος

Οι κυριότεροι μεταερευνητικοί αλγόριθμοι είναι οι εξής:

- Μέθοδοι τοπικής βελτιστοποίησης
- Γενετικοί Αλγόριθμοι (Genetic Algorithms)
- Κοινωνία μυρμηγκιών (Ant colonies)
- Σμήνος σωματιδίων (Particle Swarm Optimization)
- Προσομοιωμένη απόπτωση (Simulated Annealing)
- Ντετερμινιστική απόπτωση (Deterministic Annealing)
- Απαγορευμένης Αναζήτησης (Tabu search)
- Υβρίδια των παραπάνω μεθόδων

## **Κατασκευαστικοί Ευρεστικοί**

Οι κατασκευαστικοί αλγόριθμοι δημιουργούν λύσεις από το μηδέν. Επιλέγουν σημεία-πελάτες, μέχρις ότου δημιουργηθεί μία εφικτή λύση. Η διαδρομή που κατασκευάζεται κάθε φορά προκύπτει από την ελαχιστοποίηση κριτηρίων. Οι διαδοχικές μέθοδοι κατασκευάζουν μία διαδρομή ανά φορά, από σημείο σε σημείο για κάθε φορτηγό, ενώ οι παράλληλες μέθοδοι πολλές διαδρομές ταυτόχρονα. Η παραγωγή λύσης υψηλής ποιότητας κατά τη διάρκεια εφαρμογής μιας κατασκευαστικής μεθόδου, είναι μεγάλης σημασίας, καθώς ακόμα κι αν χρησιμοποιείται βελτιωτική μέθοδος, η μυωπική συμπεριφορά της λύσης θα υπάρχει και στη γειτονιά της αρχικής λύσης.

## **Κατασκευαστικές Μέθοδοι**

### **Πλεονεκτικός**

Η αρχή της μεθόδου είναι πολύ απλή. Κάθε πελάτης προστίθεται στο τέλος της διαδρομής, πριν δηλαδή επιστρέψει το φορτηγό στην αφετηρία. Το κριτήριο δρομολόγησης κάθε πελάτη είναι η ελαχιστοποίηση της απόστασης (ή/και του χρονικού ορίζοντα) μεταξύ του τελευταία δρομολογημένου και του υπό εξέταση πελάτη. Τα κριτήρια αποδοχής κάθε υπό εξέταση πελάτη είναι να μην παραβιάζονται τα χρονικά παράθυρα του πελάτη και της αφετηρίας και να υπάρχει αρκετό διαθέσιμο φορτίο στο φορτηγό.

Κάθε διαδρομή τερματίζεται όταν δεν μπορεί να εισαχθεί άλλος πελάτης. Τα αποτελέσματα αυτού του αλγορίθμου είναι φτωχής ποιότητας. Λόγω της απλότητάς του, οι λύσεις παράγονται ταχύτατα.

### **Route-first cluster-second**

Η συνεισφορά του Solomon στα προβλήματα VRPTW είναι οπωσδήποτε μεγάλη. Εκτός από τα 56 προβλήματα που δημιούργησε και χρησιμοποιούνται για την αξιολόγηση και σύγκριση αλγορίθμων, δημιουργεί και τις πρώτες σοβαρές μεθόδους προσέγγισης της λύσης. Η συγκεκριμένη «route-first cluster-second» αποτελείται από δύο στάδια: αρχικά δημιουργείται μία μεγάλη διαδρομή, η οποία ενώνει όλους τους πελάτες. Στη συνέχεια διαμερίζεται σε άλλες μικρότερες. Για την κατασκευή της μεγάλης – αρχικής διαδρομής, χρησιμοποιούνται αλγόριθμοι περιοδεύοντος πωλητή (TSP). Επειδή τα αποτελέσματα από έναν τέτοιο αλγόριθμο διαφέρουν ανάλογα με τη μέθοδο κατασκευής της αρχικής διαδρομής, είναι αδύνατον να συγκριθεί με τους άλλους αλγορίθμους.

### **Savings**

Άλλη μία μέθοδος προτείνεται από τον Solomon ως επέκταση του γνωστού “Savings”. Αρχικά αυτός ο αλγόριθμος σχεδιάστηκε για την επίλυση προβλημάτων VRP και μάλλον είναι ο γνωστότερος κατασκευαστικός αλγόριθμος. Η διαδικασία μοιάζει αντίστροφη της route-first cluster-second. Ξεκινάει με μία λύση κατά την

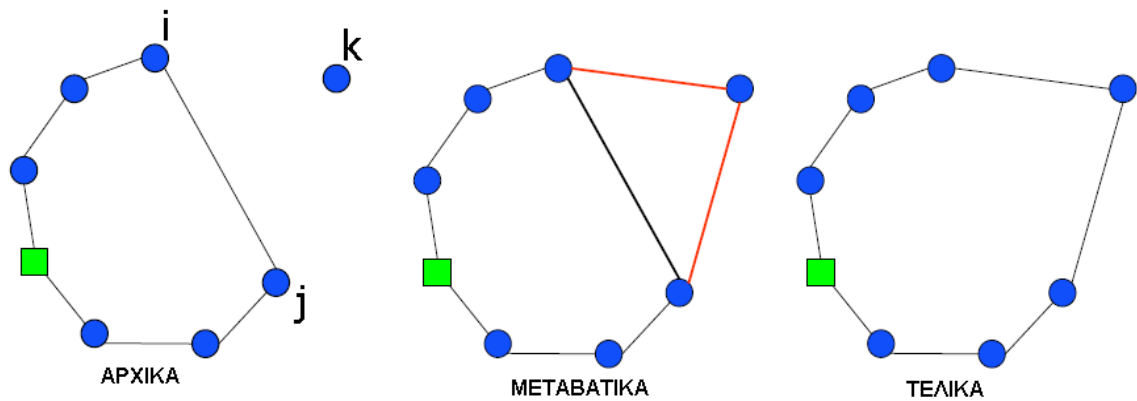
οποία κάθε πελάτης εξυπηρετείται από ακριβώς ένα φορτηγό. Συνδυάζοντας στη συνέχεια δύο διαδρομές, μικραίνει το κόστος εξυπηρέτησης  $S_{ij}=d_{i0}+d_{0j}-d_{ij}$  (όπου με  $S_{ij}$  συμβολίζεται το κέρδος από τον συγκερασμό δύο διαδρομών  $d_{i0}$  και  $d_{0j}$ ) ενώ με  $d_{ij}$  συμβολίζεται η μεταξύ των δύο πελατών απόσταση. Καθώς η διαδικασία επαναλαμβάνεται, ακυρώνονται κάποια δρομολόγια, ενώ πυκνώνουν κάποια άλλα, μέχρι να μην είναι δυνατή η συγχώνευση δύο διαδρομών. Ενώ αρχικά το μόνο κριτήριο που υπήρχε στα VRP ήταν η ελαχιστοποίηση του  $S_{ij}$  με την ένωση δύο σημείων, ο Solomon έρχεται να προσθέσει τον έλεγχο της εφικτότητας της λύσης. Αυτό συνεχίζεται μέχρι να εξαντληθεί το φορτίο του κάθε φορτηγού προκειμένου να βρεθεί ο κοντινότερος χρονικά ή/και γεωμετρικά πελάτης. Ο Solomon, θέλοντας να εξασφαλίσει μία ισορροπία ανάμεσα στο χρονικό και γεωγραφικό κριτήριο, εισάγει ένα άνω όριο χρόνου αναμονής σε κάθε διαδρομή. Η συγκεκριμένη μέθοδος μπορεί εύκολα να αναπτυχθεί είτε διαδοχικά είτε παράλληλα σε όλες τις διαδρομές.

### **Insertion**

Οι σημαντικότεροι αλγόριθμοι του Solomon είναι οι insertion και ονομάζονται I1, I2 και I3. Ο πλέον αποτελεσματικός και πιο δημοφιλής είναι ο I1. Πολλές φορές χρησιμεύει ως βάση σύγκρισης με άλλους αλγόριθμους. Η βασική αρχή τους είναι κοινή. Κάθε διαδρομή αρχίζει με έναν πελάτη - κέντρο και οι υπόλοιποι μη δρομολογημένοι πελάτες εισάγονται σε αυτήν όσο αυτό είναι εφικτό.

Σε περίπτωση που υπάρχουν μη δρομολογημένοι πελάτες όταν τελειώσει μία διαδρομή, αρχίζει μια νέα.

Ο αρχικός πελάτης επιλέγεται είτε ως ο πιο απομακρυσμένος γεωγραφικά από την αποθήκη, είτε ως ο πελάτης του οποίου το χρονικό του παράθυρο ανοίγει νωρίτερα. Η εισαγωγή του νέου αδρομολόγητου πελάτη γίνεται ανάμεσα σε δύο δρομολογημένους. Σε αυτές τις μεθόδους υπάρχουν μεταβλητές που καθορίζουν πόσο σημαντική είναι για τη λύση η απόσταση του νέου πελάτη από την αποθήκη, σε αντίθεση με την προστιθέμενη διανυθείσα απόσταση και τον προστιθέμενο απαιτούμενο χρόνο. Ο I2 φαίνεται να στοχεύει στην ελαχιστοποίηση της ολικής διανυθείσας απόστασης και του ολικού χρόνου, ενώ ο I3 στοχεύει στην εξυπηρέτηση των πελατών που επείγονται.



Σχήμα 4. Αλγόριθμος Insertion

Σε μεταγενέστερες μελέτες διορθώνεται ένα κριτήριο του Solomon το οποίο υποτιμούσε το χρόνο που προστίθεται κατά την εισαγωγή ενός νέου πελάτη μεταξύ της αποθήκης και του πρώτου πελάτη. Με τα νέα κριτήρια, σε σχετικά μικρά δρομολόγια, μπορεί να επιτευχθεί βελτίωση έως και 50%.

Νωρίτερα, έχει πραγματοποιηθεί από τους Potvin και Rousseau μια διαφορετική χρήση του Π1, η παράλληλη. Αρχικά χρησιμοποιείται η αρχική μορφή του Π1, η διαδοχική, προκειμένου να οριστεί ο αριθμός των χρησιμοποιούμενων φορτηγών ανά πρόβλημα και για να γίνει η επιλογή του αρχικού πελάτη. Στη συνέχεια, χρησιμοποιούνται αυτά τα δεδομένα για να γίνει η ανάθεση των υπολοίπων πελατών. Το κριτήριο επιλογής του επόμενου πελάτη σχετίζεται με το κατά πόσο συμφέρει να δρομολογηθεί ένας πελάτης σε μια διαδρομή από ότι στις άλλες. Περαιτέρω μελέτες έδειξαν πως η χρήση πολλών επεξεργαστών παράλληλα, μειώνει γραμμικά τον υπολογιστικό χρόνο στα παράλληλα μέρη του αλγορίθμου.

Πρόσφατα ο Ιοαννου χρησιμοποιώντας τη γενική μορφή του Π1, όπως προτάθηκε από τον Solomon, ρύθμισε τα κριτήρια έτσι ώστε να ελαχιστοποιείται η επίδραση της εισχώρησης νέων πελατών στους πελάτες που βρίσκονται στην υπό κατασκευή διαδρομή αλλά και στους αδρομολόγητους και στον υπό εξέταση πελάτη.

### Sweep

Πρόκειται για συνδυασμό δύο αλγορίθμων, του αλγορίθμου σάρωσης (sweep heuristic) και του Π1. Η βάση της μεθόδου είναι αντίστοιχη με τη «Route-first cluster-second» με ανάποδη όμως σειρά. Αρχικά ομαδοποιούνται οι πελάτες κάνοντας χρήση του sweep heuristic, ανατίθεται δηλαδή σε κάθε φορτηγό ένας αριθμός πελατών. Η ομαδοποίηση γίνεται χωρίζοντας τους πελάτες σε περιοχές ανάλογα με τη γωνία που σχηματίζουν από το κέντρο βάρους του προβλήματος. Στη συνέχεια χρησιμοποιείται ο Π1 για τη δρομολόγησή τους.

## Η ΓΛΩΣΣΑ VB - ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ (OOP)

Η *Visual Basic* - ή σε συντομογραφία *VB* - είναι μία γλώσσα που πολλοί προγραμματιστές, συμπεριλαμβανομένης της *Microsoft*, χρησιμοποιούν για να γράψουν νέες εφαρμογές λογισμικού. Το 1994 υπήρχαν πάνω από ένα εκατομμύριο εφαρμογές *Visual Basic* σε εμπορική χρήση παγκοσμίως. Αν κοιτάξει κανείς προσεκτικά σε κάθε σύγχρονη βάση δεδομένων για παραθυρικό περιβάλλον, σε κάθε λογιστικό φύλλο ή πακέτο επεξεργασίας κειμένου από τους κορυφαίους κατασκευαστές λογισμικού, θα ανακαλύψει ότι η μάκρο γλώσσα του λογισμικού είναι είτε μια ποικιλία *Visual Basic* είτε σχεδόν πανομοιότυπη με αυτή. Για αυτόν το λόγο καθένας που επιθυμεί να γίνει ένας «ειδικός» στα *Windows* πρέπει να κατέχει αυτήν την γλώσσα και όλοι οι χρήστες των *Windows*, εκτός από τους πολύ περιστασιακούς, θα έπρεπε να έχουν μια ιδέα για αυτήν.

Ο παραδοσιακός προγραμματισμός είναι βασικά γραμμικός και βασίζεται στη ροή της εκτέλεσης των εντολών. Οι λειτουργίες εκτελούνται για έναν ορισμένο αριθμό επαναλήψεων ή μέχρι να φτάσουν σημεία απόφασης που είναι γραμμένα στο πρόγραμμα και η διακοπή μιας εκτελούμενης διεργασίας είναι είτε δύσκολη είτε αδύνατη. Τα προγράμματα συνήθως σχεδιάζονται από πάνω προς τα κάτω, ίσως ακολουθώντας τη μέθοδο του Δομημένου Προγραμματισμού του *Jackson* (*Jackson Structured Programming method – JSP*), αναλύοντας πολύπλοκες λειτουργίες σε διαδοχικά απλούστερες. Μερικές φορές μια προσέγγιση με υποδείγματα (*modular approach*) επιχειρείται, δημιουργώντας ένα πρόγραμμα από ένα σύνολο μερικών λίγο έως πολύ ανεξάρτητων συναρτήσεων και διαδικασιών. Θεωρητικά αυτή η προσέγγιση καθιστά ικανή την επαναχρησιμοποίηση των υποδειγμάτων (*modules*) σε άλλα προγράμματα. Πρακτικά όμως υπάρχουν πολύ λίγες ρουτίνες που μπορούν να επαναχρησιμοποιηθούν χωρίς σημαντική επεξεργασία.

Αντίθετα, η επαναχρησιμοποίηση του λογισμικού (*software reuse*) είναι γνωστό ότι διευκολύνεται αν ακολουθηθεί η μεθοδολογία του αντικειμενοστραφούς προγραμματισμού (*object oriented programming*). Η *Visual Basic* είναι αντικειμενοστραφής (*object oriented*), δηλαδή εξελίσσεται γύρω από έτοιμα - ή και *customized* - αντικείμενα (*objects*) τα οποία έχουν τις δικές τους ιδιότητες (*properties*). Επίσης η *Visual Basic* κινείται μέσω συγκεκριμένων γεγονότων (*event driven*), δηλαδή όλες οι δραστηριότητες σε ένα πρόγραμμα «πυροδοτούνται» από ένα γεγονός (*event*), π.χ. το κλικ του ποντικιού σε ένα κουμπί.

Οι γλώσσες με προσανατολισμό στο αντικείμενο (*object oriented languages*), δίνουν πολύ μεγάλη έμφαση σε παρόμοιες λογικές. Κεντρικό σημείο του αντικειμενοστραφούς προγραμματισμού είναι η χρήση των λεγόμενων δεικτών (*pointers*). Είναι γνωστό ότι οι τύποι δεδομένων π.χ. στην *Pascal* διακρίνονται σε απλούς, σε δομημένους και σε δείκτες (*pointers*). Απλοί τύποι είναι ο ακέραιος (*integer*) και ο πραγματικός (*real*) αριθμός, ο χαρακτήρας (*char*) και ο λογικός (*boolean*) τύπος. Δομημένοι τύποι είναι ο πίνακας (*array*), η συμβολοσειρά (*string*), η

εγγραφή (*record*), το αρχείο (*file*) και το σύνολο (*set*) που αποτελούνται από απλούς ή από άλλους δομημένους τύπους. Ο ορισμός των δεικτών (*pointers*) προϋποθέτει την έννοια του κόμβου μιας δομής δεδομένων (*node*) και της υποδιαίρεσης του κόμβου που είναι γνωστή ως πεδίο (*field*). Πράγματι, κάθε μορφή δομής δεδομένων αποτελείται από ένα σύνολο κόμβων (*nodes*). Τα δεδομένα ενός κόμβου μπορεί να αντιστοιχούν σε μία ή περισσότερες μεταβλητές που αποθηκεύονται σε μια ή περισσότερες λέξεις (*words*) της κύριας μνήμης του υπολογιστή. Κάθε λέξη αποτελείται από 8 έως 64 δυαδικά ψηφία (*bits*) ή με άλλα λόγια από 1 έως 8 χαρακτήρες (*bytes*). Μια υποδιαίρεση του κόμβου λέγεται πεδίο (*field*). Τα πεδία ενός κόμβου μπορούν να καταταγούν σε δύο κατηγορίες: τα δεδομένα (*data*) και το δεσμό (*link*) ή δείκτη (*pointer*).

Ο δείκτης (*pointer*) δείχνει πάντοτε τη διεύθυνση ενός κόμβου. Η διεύθυνση μπορεί να είναι απόλυτη (*absolute address*) είτε σχετική (*relative address*) ως προς μια διεύθυνση βάσης (*base address*). Στον κόμβο μιας δομής δεδομένων μπορεί να υπάρχουν κανένα, ένα ή περισσότερα πεδία των κατηγοριών δεδομένα και δείκτες.

Με βάση τους πιο πάνω ορισμούς μπορούν να προδιαγραφούν τα είδη των δομών δεδομένων. Η πιο απλή μορφή δομής δεδομένων είναι ο πίνακας (*array*). Οι κόμβοι του πίνακα, που ονομάζονται στοιχεία, περιέχουν μόνο δεδομένα. Στις γραμμικές λίστες (*linear lists*) η διαδοχή των κόμβων μπορεί να επιτευχθεί είτε με την τοποθέτηση σε συνεχόμενες θέσεις μνήμης είτε με τη χρήση δεικτών. Η δομή του δένδρου (*tree*) είναι η πιο ενδιαφέρουσα μορφή μη γραμμικών λιστών. Τέλος, η δομή του γράφου ή γραφήματος (*graph*) αποτελεί μια γενίκευση τόσο των γραμμικών λιστών όσο και της δομής του δένδρου. Οι κόμβοι των δένδρων και των γράφων περιέχουν συνήθως δεδομένα και δύο ή περισσότερους δείκτες.

Στα παραπάνω θεωρήσαμε γνωστή την έννοια της **δομής δεδομένων** (*data structure*). Γενικά, δομή δεδομένων είναι ένα σύνολο αποθηκευμένων δεδομένων που υφίστανται επεξεργασία από ένα σύνολο λειτουργιών που καλούνται από το υπόλοιπο πρόγραμμα.

Επίσης κεντρικής σημασίας στην πληροφορική είναι η έννοια του **αλγόριθμου** (*algorithm*). Ο αλγόριθμος είναι ένα πεπερασμένο σύνολο εντολών αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που αν ακολουθηθούν επιτυγχάνεται ένα επιθυμητό αποτέλεσμα. Κάθε αλγόριθμος πρέπει να ικανοποιεί τα πιο κάτω κριτήρια:

- Είσοδος (*input*): Καμία, μία ή περισσότερες ποσότητες να δίνονται ως είσοδοι στον αλγόριθμο
- Έξοδος (*output*): Ο αλγόριθμος να δημιουργεί τουλάχιστον μια ποσότητα ως αποτέλεσμα
- Καθορισμός (*definiteness*): Κάθε εντολή να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της



- Περατότητα (*finiteness*): Ο αλγόριθμος να τελειώνει μετά από πεπερασμένα βήματα εκτέλεσης των εντολών του. Μία διαδικασία που δεν τελειώνει μετά από ένα πεπερασμένο αριθμό βημάτων λέγεται απλώς υπολογιστική διαδικασία (*computational procedure*)
- Αποτελεσματικότητα (*effectiveness*): Κάθε μεμονωμένη εντολή του αλγορίθμου να είναι απλή. Αυτό σημαίνει ότι μια εντολή δεν αρκεί να έχει ορισθεί, αλλά πρέπει να είναι και εκτελέσιμη.

Πρέπει να τονισθεί ότι υπάρχει μεγάλη εξάρτηση μεταξύ της δομής δεδομένων και του αλγορίθμου που επεξεργάζεται τη δομή. Το πρόγραμμα μάλιστα πρέπει να θεωρεί τη δομή δεδομένων και τον αλγόριθμο ως μια αδιάσπαστη ενότητα. Η παρατήρηση αυτή δικαιολογεί την εξίσωση που διατυπώθηκε από τον *Wirth* το 1976:

Αλγόριθμοι + Δομές Δεδομένων = Προγράμματα

Παρακάτω γίνεται κάποια αναφορά ορισμών στην ονοματολογία της γλώσσας *Visual Basic 6.0* κυρίως σχετικά με τις λειτουργίες αντικειμενοστραφούς προγραμματισμού (*object oriented programming*).

Τα αντικείμενα (*objects*) είναι προγραμματιζόμενα στοιχεία του περιβάλλοντος ενός προγράμματος *Visual Basic*. Για παράδειγμα αντικείμενα είναι τα εργαλεία ή χειριστήρια (*controls*) που μπορούμε να χρησιμοποιήσουμε για να προσθέσουμε σε ένα πρόγραμμα γραφικά, ετικέτες, κουμπιά, πλαίσια καταλόγων, ράβδους κύλισης, μενού και γεωμετρικά σχήματα. Επίσης η ίδια η φόρμα διασύνδεσης χρήστη (*userform*) είναι ένα αντικείμενο της *Visual Basic*. Τα χειριστήρια και η φόρμα είναι περιπτώσεις αντικειμένων που είναι «ορατά» στο χρήστη του προγράμματος κάθε φορά που το εκτελεί. Υπάρχουν ωστόσο και αντικείμενα με τα οποία μπορούμε να εκτελούμε ειδικές «παρασκηνιακές» εργασίες σε κάποιο πρόγραμμα της *Visual Basic*. Αυτά τα πανίσχυρα αντικείμενα εκτελούν πολύ χρήσιμες εργασίες, αλλά είναι αόρατα στο χρήστη όταν εκτελείται το πρόγραμμα. Σε αυτά περιλαμβάνονται αντικείμενα που χειρίζονται πληροφορίες βάσεων δεδομένων, αλληλεπιδρούν με εφαρμογές *Windows* και παρακολουθούν το χρόνο στα προγράμματα.

Οι λειτουργικές μονάδες τάξεων (*class modules*) ή σε ελεύθερη μετάφραση «υποδείγματα κλάσεων αντικειμένων» χρησιμοποιούνται για τη δημιουργία νέων *customized* αντικειμένων από τον προγραμματιστή. Αποτελούν το καλούπι που παράγει αντικείμενα κάποιου επιθυμητού είδους.

Οι συλλογές αντικειμένων (*collections*) είναι ομάδες αντικειμένων που σχετίζονται μεταξύ τους και συγκεκριμένα συνήθως ομάδες των δεικτών (*pointers*) των αντικειμένων.

Οι δυναμικοί πίνακες (*dynamic arrays*) είναι πίνακες οι διαστάσεις των οποίων ορίζονται κατά τον χρόνο εκτέλεσης είτε από τον ίδιο τον χρήστη είτε όταν η λογική

του προγράμματος προσδιορίζει το μέγεθός του με βάση τις συγκεκριμένες συνθήκες. Η διαστασιολόγηση ενός δυναμικού πίνακα απαιτεί να γίνει κάποια κράτηση μνήμης στο στάδιο του σχεδιασμού συνήθως μέσω μιας *variant* μεταβλητής .

Οι συναρτήσεις ή διαδικασίες *Function (Functions)* είναι ομάδες εντολών που τοποθετούνται σε βασικές λειτουργικές μονάδες (*modules*) ή λειτουργικές μονάδες τάξεων στον *OOP (class modules)* μεταξύ των εντολών *Function* και *End Function*. Οι εντολές στο εσωτερικό της συνάρτησης είναι αυτές που παράγουν χρήσιμο έργο. Μια συνάρτηση εκτελεί μια εργασία, συνήθως κάποιους υπολογισμούς, και επιστρέφει μια τιμή μέσω του ονόματός της.

Οι υπορουτίνες ή διαδικασίες *Sub (Subs)* είναι παρόμοιες με τις διαδικασίες *Function* με μόνη διαφορά ότι οι τιμές που επιστρέφουν οι υπορουτίνες δεν αντιστοιχούν με τα ονόματά τους. Η κλήση μιας υπορουτίνας επιτρέπει την επεξεργασία και επιστροφή πολλών μεταβλητών με τη χρήση μόνο μιας γραμμής κώδικα. Οι περισσότερες συναρτήσεις επιστρέφουν μόνο μια τιμή, ενώ οι υπορουτίνες μπορούν να επιστρέφουν πολλές.

Τέλος, αξίζει να διευκρινήσουμε τις έννοιες των δημόσιων (*Public*) και ιδιωτικών (*Private*) συλλογών, συναρτήσεων, υπορουτινών κ.λ.π. Για παράδειγμα μια συλλογή (*collection*) ορισμένη σε μια λειτουργική μονάδα τάξεων (*class module*) αν είναι ιδιωτική (*Private*) θα μπορεί να είναι προσβάσιμη μόνο μέσα στη συγκεκριμένη μονάδα τάξεων. Αντίθετα, αν οριστεί ως δημόσια (*Public*) θα είναι προσβάσιμη από κάθε λειτουργική μονάδα τάξεων (*class module*) του προγράμματος.

## **ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ**

Σκοπός του πειραματικού μέρους είναι η δημιουργία ενός αλγορίθμου Insertion ο οποίος βελτιστοποιεί τη μετρητική διαδικασία των κοινωφελών οργανισμών. Η γλώσσα που χρησιμοποιήθηκε για τον κώδικα ήταν η VB (αντικειμενοστραφής προγραμματισμός) η οποία αναλύθηκε παραπάνω.

### **Βασική Πορεία Αλγορίθμου**

Αρχικά, ορίζονται οι μεταβλητές και στη συνέχεια με την εντολή Read καλείται να διαβάσει τα δεδομένα που είναι οι συντεταγμένες των πελατών που πρέπει να εξυπηρετηθούν από το αρχείο excel που είναι καταχωρημένες. Ορίζει τα δεδομένα της αποθήκης που την ονομάζει office. Ύστερα ορίζει τα δεδομένα των πελατών και των οχημάτων ενώ δηλώνει πως το πρώτο και το τελευταίο σημείο κάθε φορτηγού είναι η αποθήκη office. Εν συνεχεία με την υπορουτίνα Insertion (Sub Insertion), υπολογίζει την βέλτιστη διαδρομή. Δημιουργεί τη συλλογή με τους πελάτες προς εξυπηρέτηση και τη συλλογή με τους πελάτες που δεν θα εξυπηρετηθούν. Για κάθε σημείο που θα επισκεφτεί το όχημα υπολογίζει τις αποστάσεις του πελάτη από το σημείο αυτό και το επόμενο και στη συνέχεια υπολογίζει την πρόσθετη απόσταση που θα διανυθεί. Αν ο χρόνος που θα χρειαστεί το όχημα για την εξυπηρέτηση των

πελατών είναι μικρότερος από τον μέγιστο διαθέσιμο τότε εξετάζει αν η απόσταση είναι μικρότερη της ελάχιστης. Αν η απόσταση είναι όντως μικρότερη ορίζει αυτόν τον πελάτη σαν το αμέσως επόμενο σημείο που θα επισκεφτεί. Αν όμως δεν μπορεί να επισκευτεί άλλον πελάτη τον βάζει στη συλλογή Missing. Αυτό πραγματοποιείται μέχρις ότου εξυπηρετηθούν όλοι οι πελάτες, δηλαδή μέχρι ότου δεν θα έχω πελάτες στη συλλογή Missing. Τέλος γίνεται ο υπολογισμός του κόστους (χρόνου) που βαρύνει το κάθε όχημα και προσθέτει για κάθε ένα από αυτά το κόστος και το φορτίο στη λύση και παρουσιάζει τα αποτελέσματα.

Ο αλγόριθμος αυτός εφαρμόστηκε σε τέσσερα παραδείγματα των οποίων τα αποτελέσματα θα αναλυθούν παρακάτω. Στο πρώτο παράδειγμα, για 2500 σημεία (πελάτες) θέσαμε ως χρόνο εξυπηρέτησης (service time) 1 min και ότι ο κάθε υπάλληλος (όχημα) εργάζεται 8 ώρες τη μέρα. Στη συνέχεια ορίσαμε την ταχύτητα του οχήματος, κατά προτίμηση μηχανάκι, ίση με 40 km/h. Στο δεύτερο παράδειγμα για τα ίδια σημεία υποθέσαμε πως ο υπάλληλος μεταφέρεται στην περιοχή όπου βρίσκονται οι πελάτες με Μ.Μ.Μ (π.χ λεωφορείο) και έτσι ορίσαμε ως νέα αποθήκη τη στάση του λεωφορείου. Στη συνέχεια ο υπάλληλος εξυπηρετεί τους πελάτες με βάδισμα και μάλιστα με 4 km/h. Ομοίως πορευτήκαμε και στα άλλα δύο παραδείγματα για άλλα 2500 σημεία το καθένα.

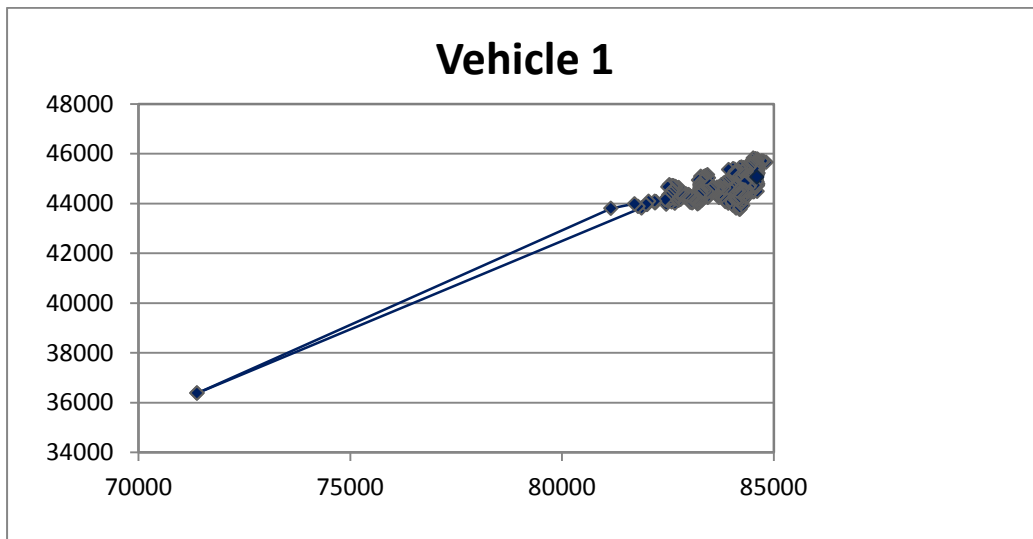
## Αποτελέσματα

- 1<sup>ο</sup> Παράδειγμα

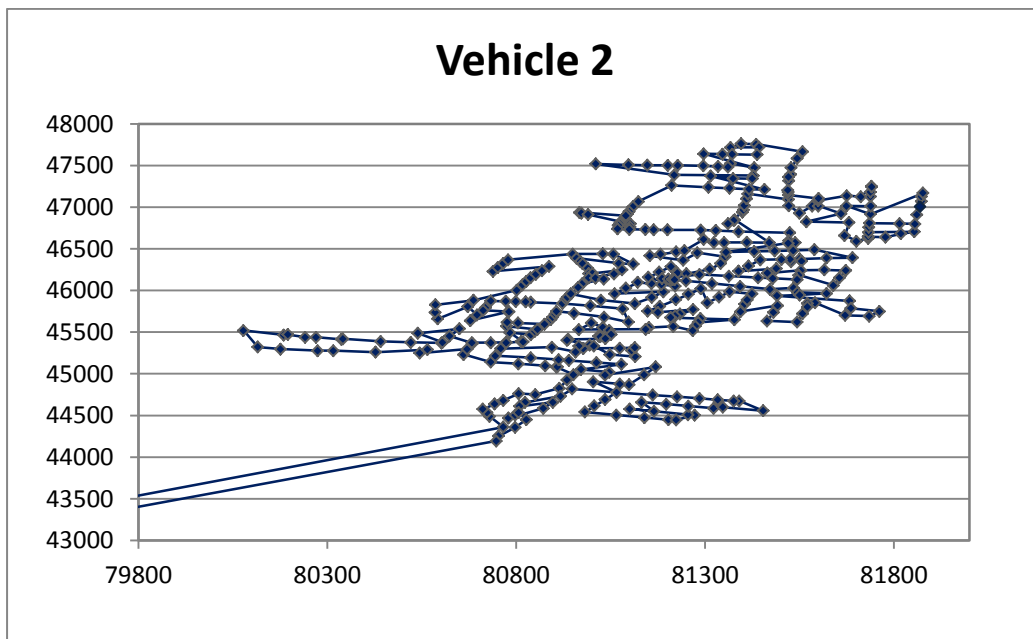
ΟΧΗΜΑΤΑ	ΤΑΧΥΤΗΤΑ(km/h)	ΦΟΡΤΙΟ	ΚΟΣΤΟΣ(min)
1	40	404	479,8685
2	40	399	479,9895
3	40	400	479,5405
4	40	384	479,3611
5	40	391	479,0485
6	40	382	478,9885
7	40	140	226,2065

Συνεπώς, εξυπηρετούνται όλοι οι πελάτες (2500) με τη χρήση 7 οχημάτων και συνολικό κόστος (χρόνος) ίσο με 3102 min. Παρακάτω, παρατίθενται τα διαγράμματα για τη διαδρομή που ακολούθησε το κάθε όχημα καθώς και ένα συγκεντρωτικό διάγραμμα που δείχνει τις διαδρομές όλων των οχημάτων.

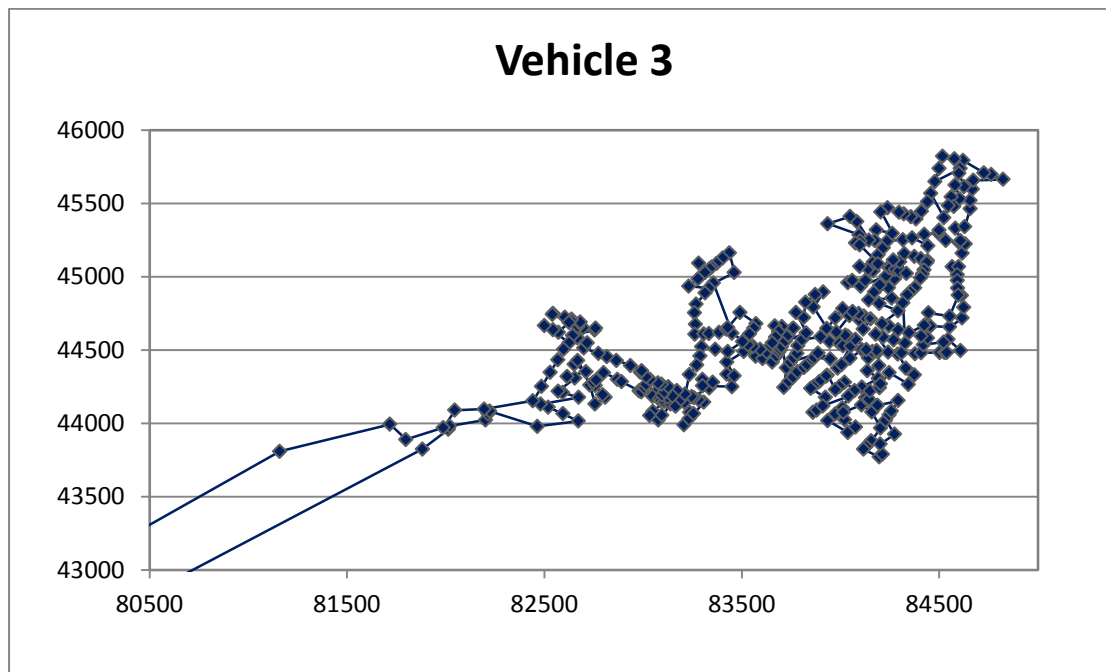
Για το όχημα 1:



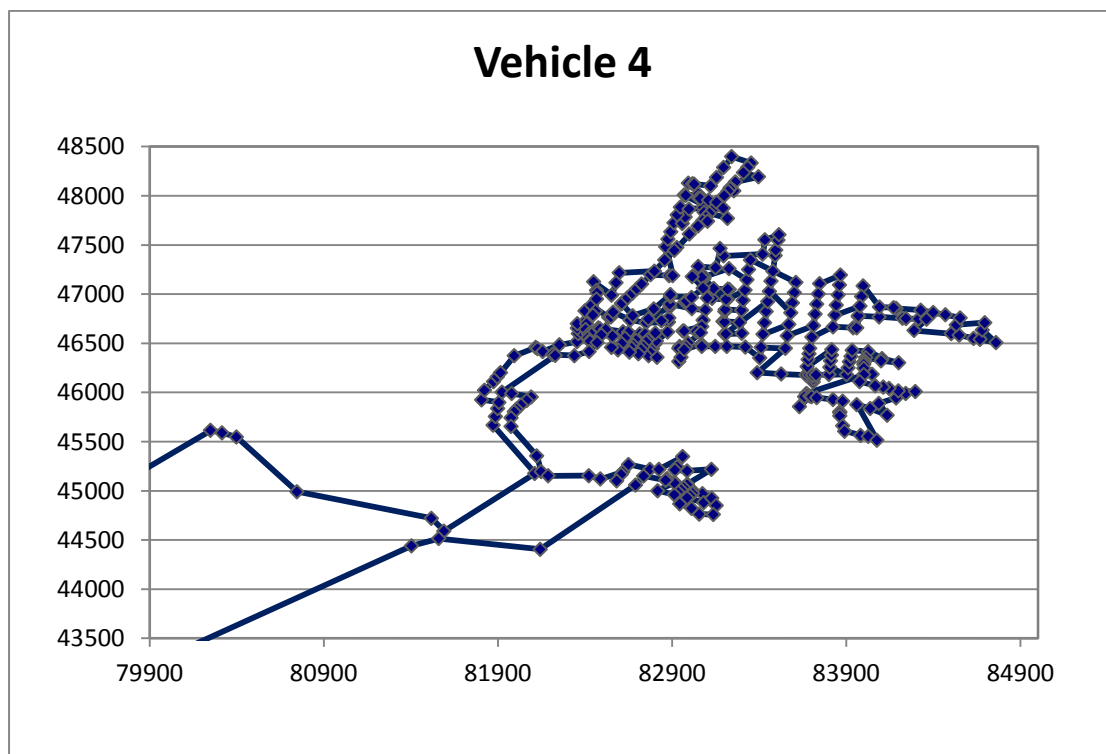
Για το όχημα 2:



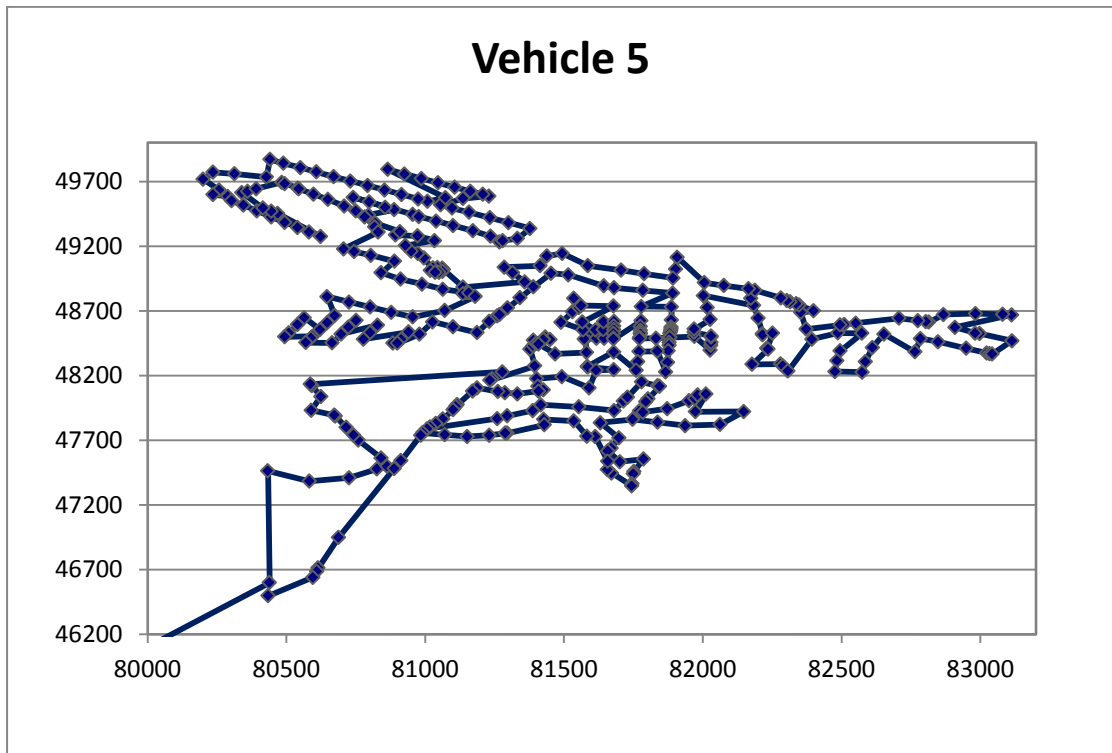
Για το όχημα 3:



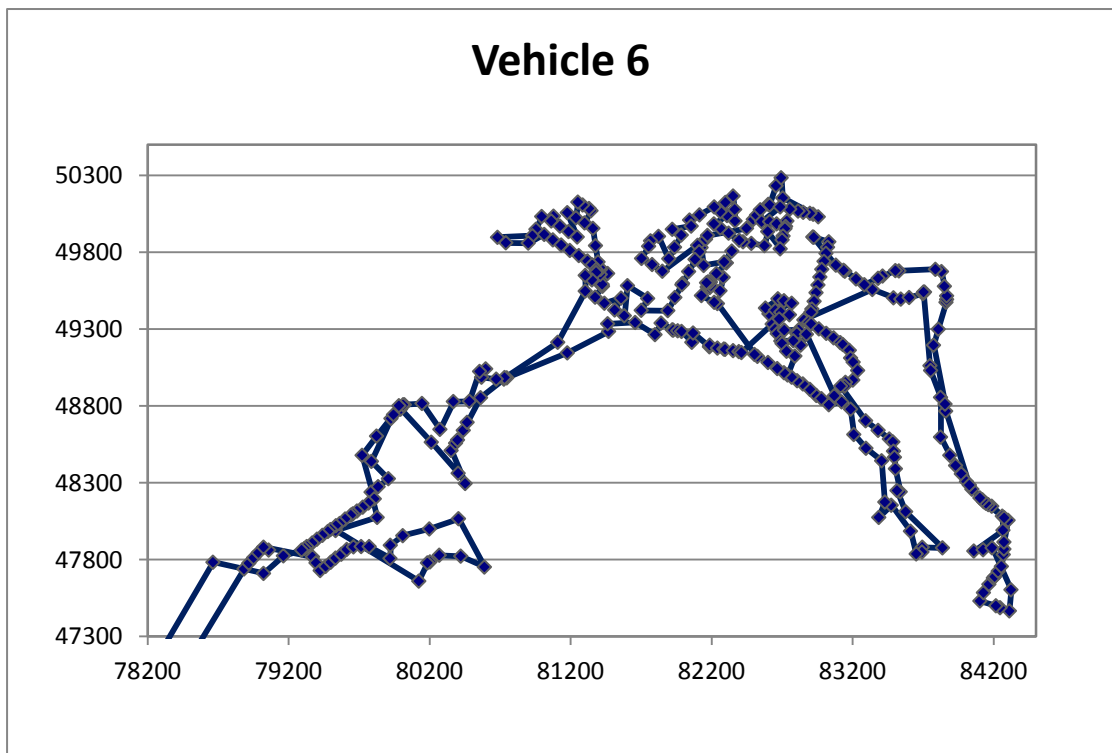
Για το όχημα 4:



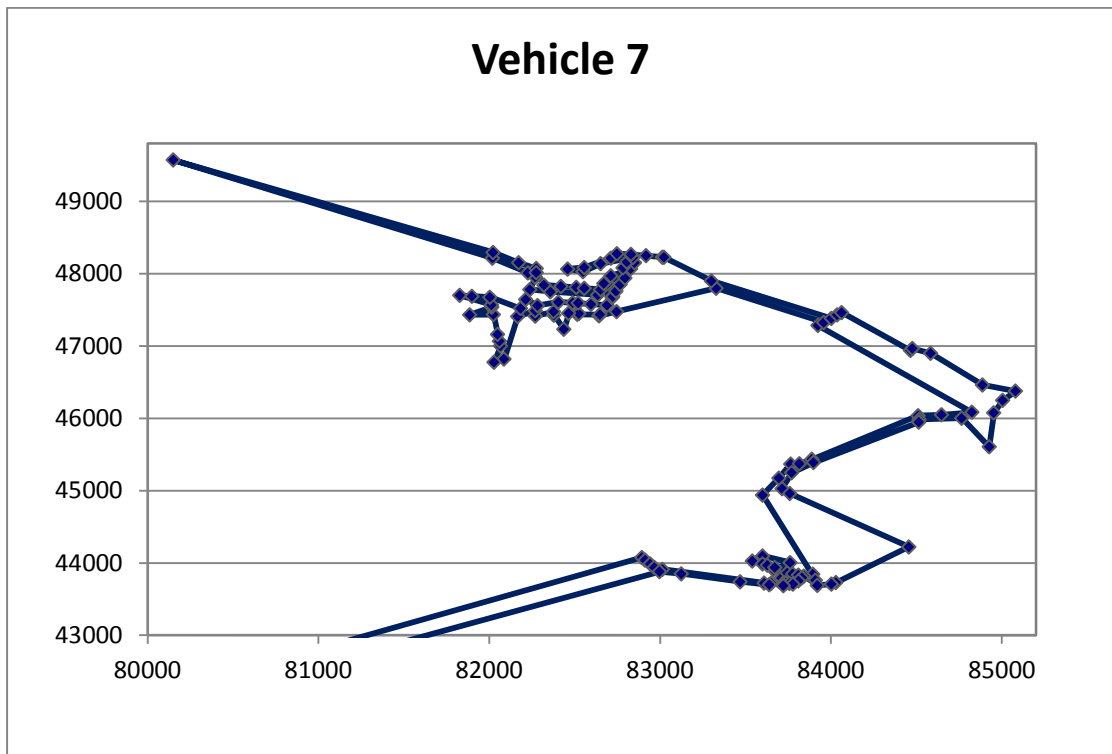
Για το όχημα 5:



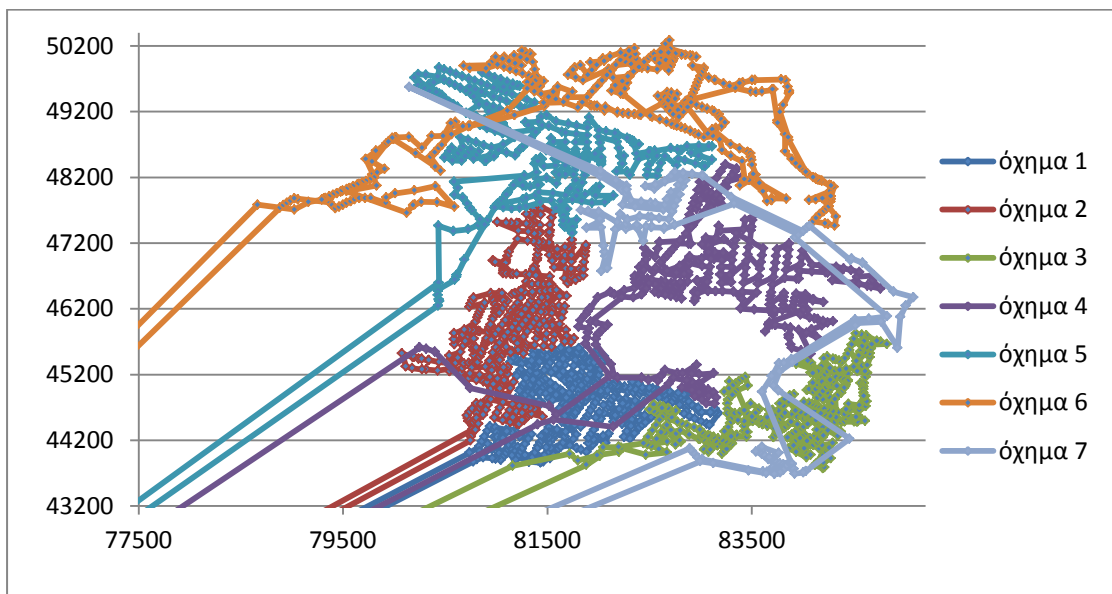
Για το όχημα 6:



Για το όχημα 7:



Συγκεντρωτικό Διάγραμμα:

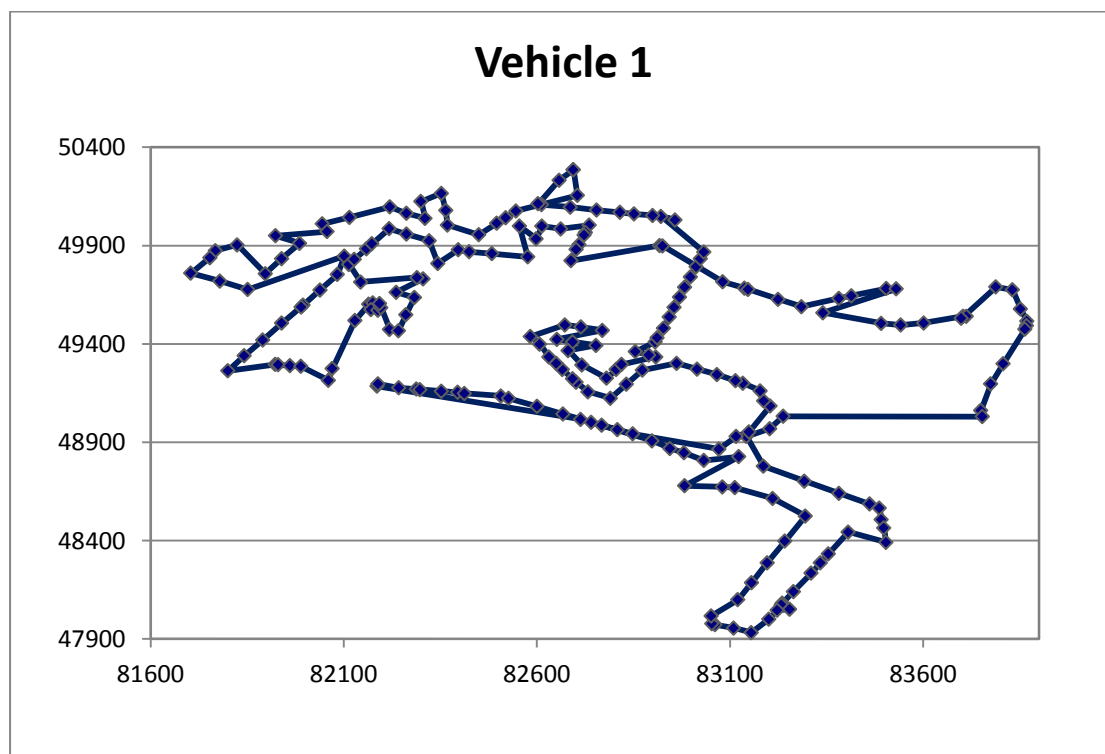


- 2<sup>ο</sup> Παράδειγμα

ΟΧΗΜΑΤΑ	ΤΑΧΥΤΗΤΑ(km/h)	ΦΟΡΤΙΟ	ΚΟΣΤΟΣ(min)
1	4	479,77	219
2	4	479	203
3	4	478,82	212
4	4	478,35	185
5	4	479,52	164
6	4	479,19	177
7	4	479,82	156
8	4	479,71	183
9	4	478,29	168
10	4	479,56	166
11	4	478,34	160
12	4	478,73	140
13	4	478,07	158
14	4	477,99	142
15	4	377,29	67

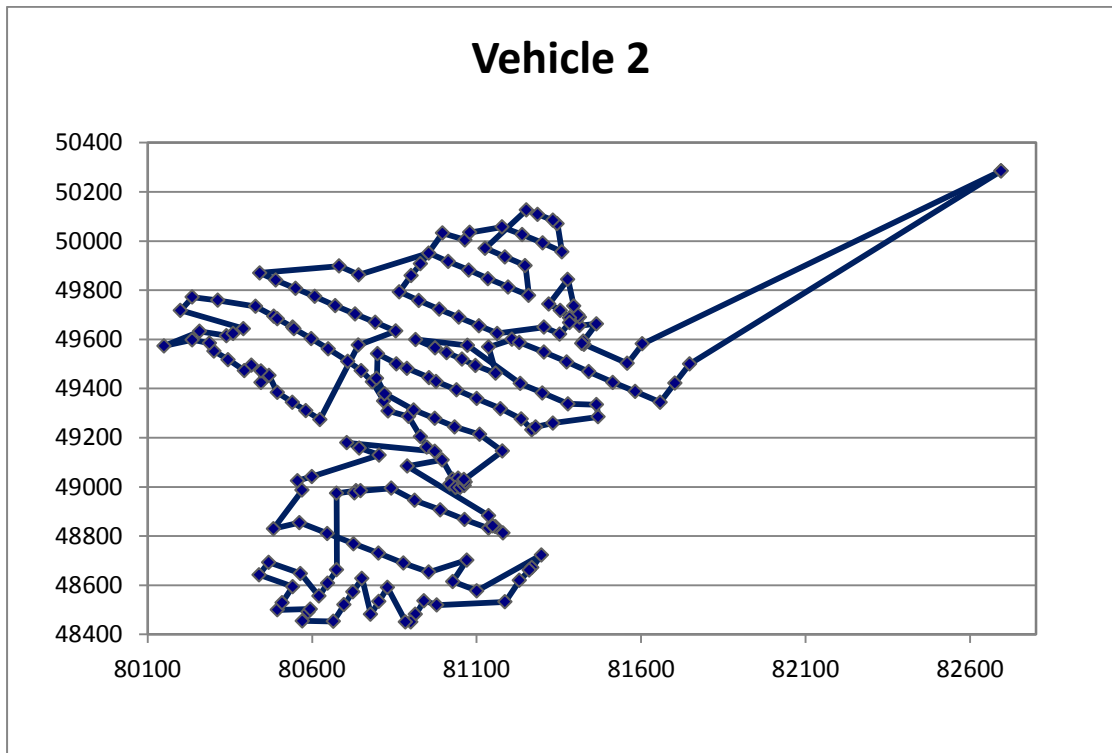
Συνεπώς, εξυπηρετούνται όλοι οι πελάτες (2500) με τη χρήση 15 οχημάτων και συνολικό κόστος (χρόνος) περίπου ίσο με 7083 min. Παρακάτω, παρατίθενται τα διαγράμματα για τη διαδρομή που ακολούθησε το κάθε όχημα καθώς και ένα συγκεντρωτικό διάγραμμα που δείχνει τις διαδρομές όλων των οχημάτων.

Για το όχημα1:

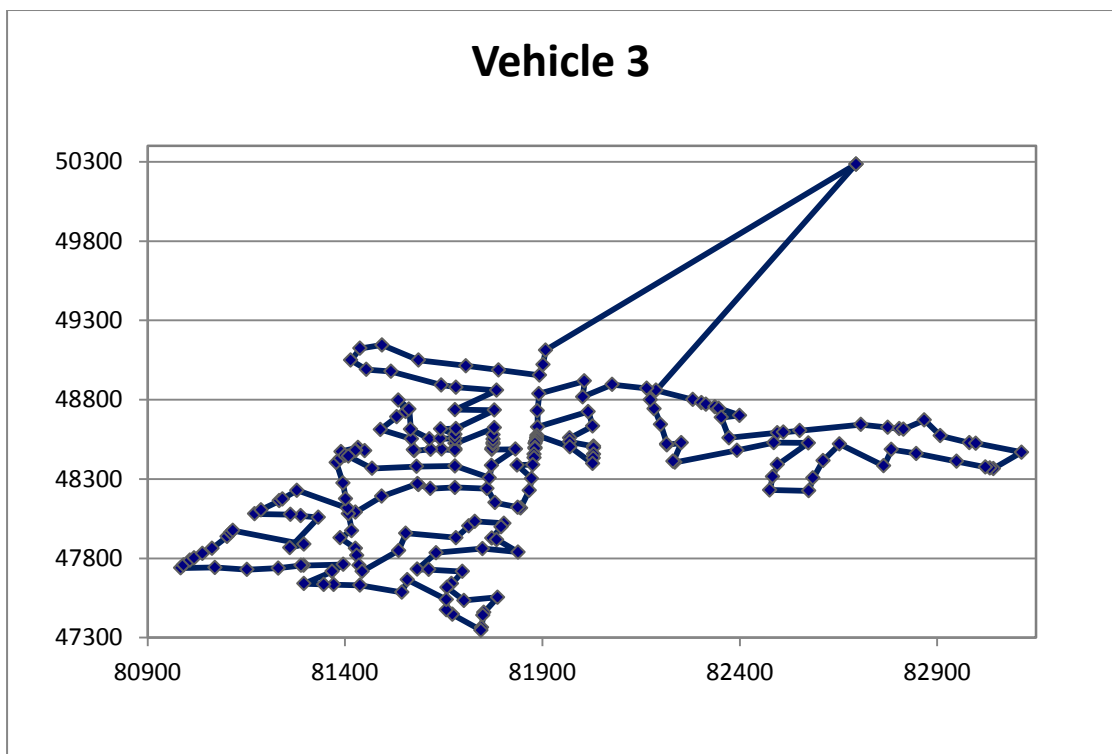




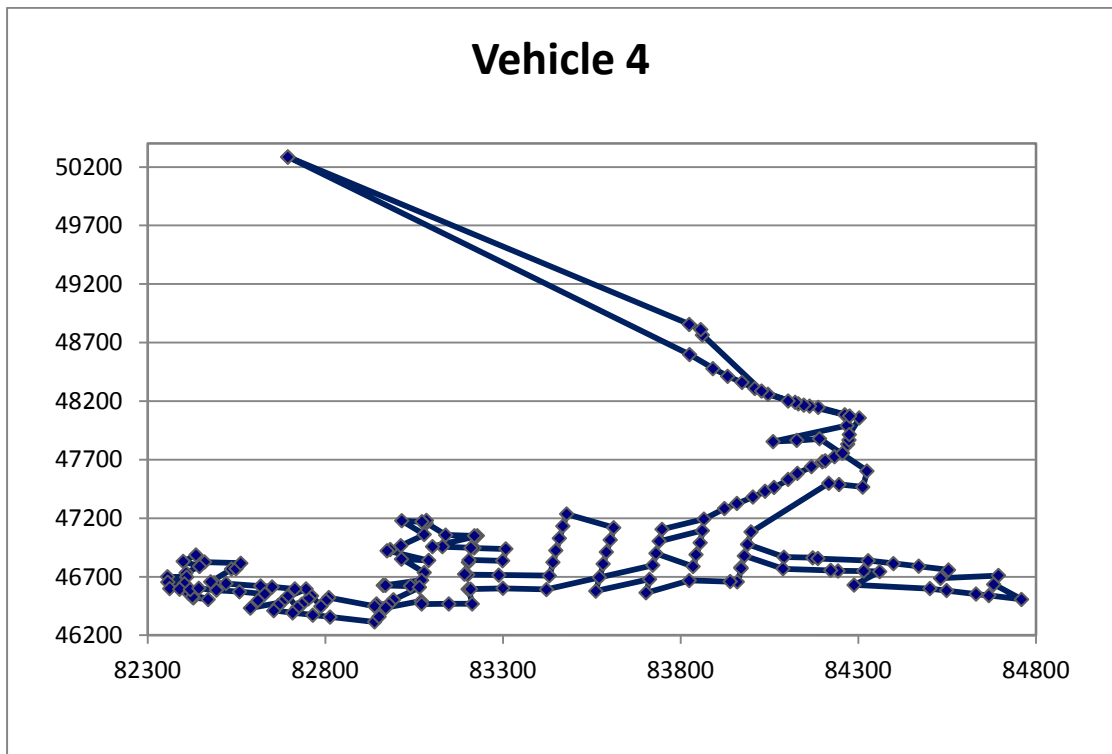
Για το όχημα 2:



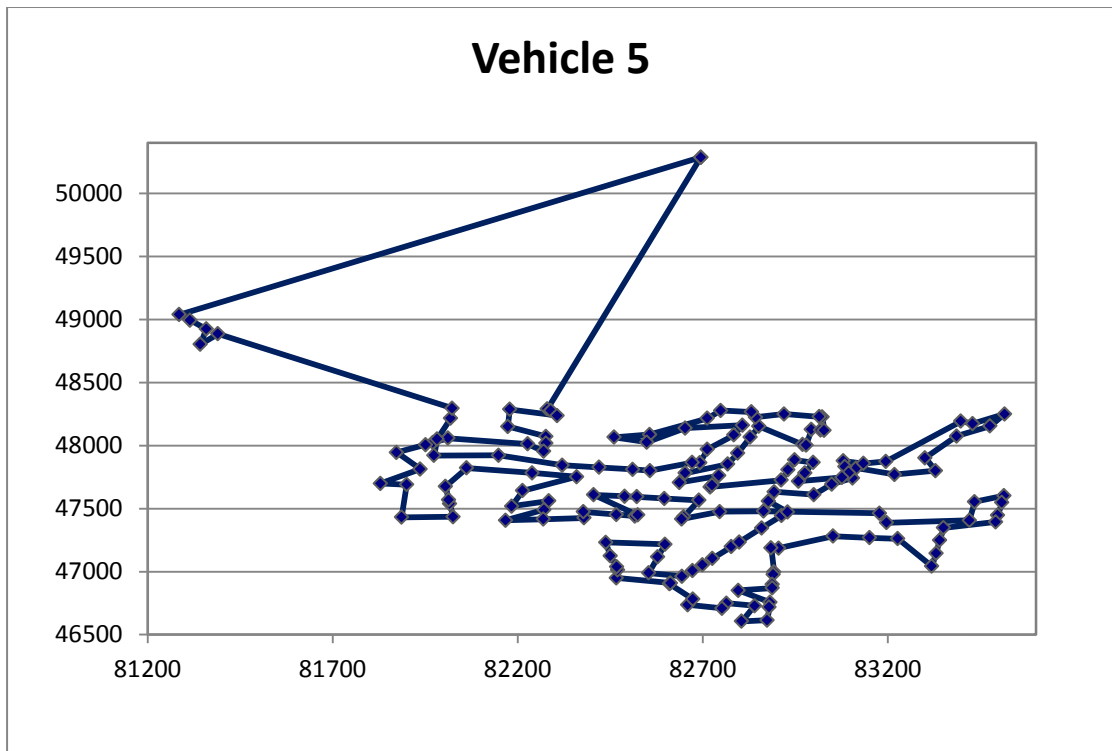
Για το όχημα 3:



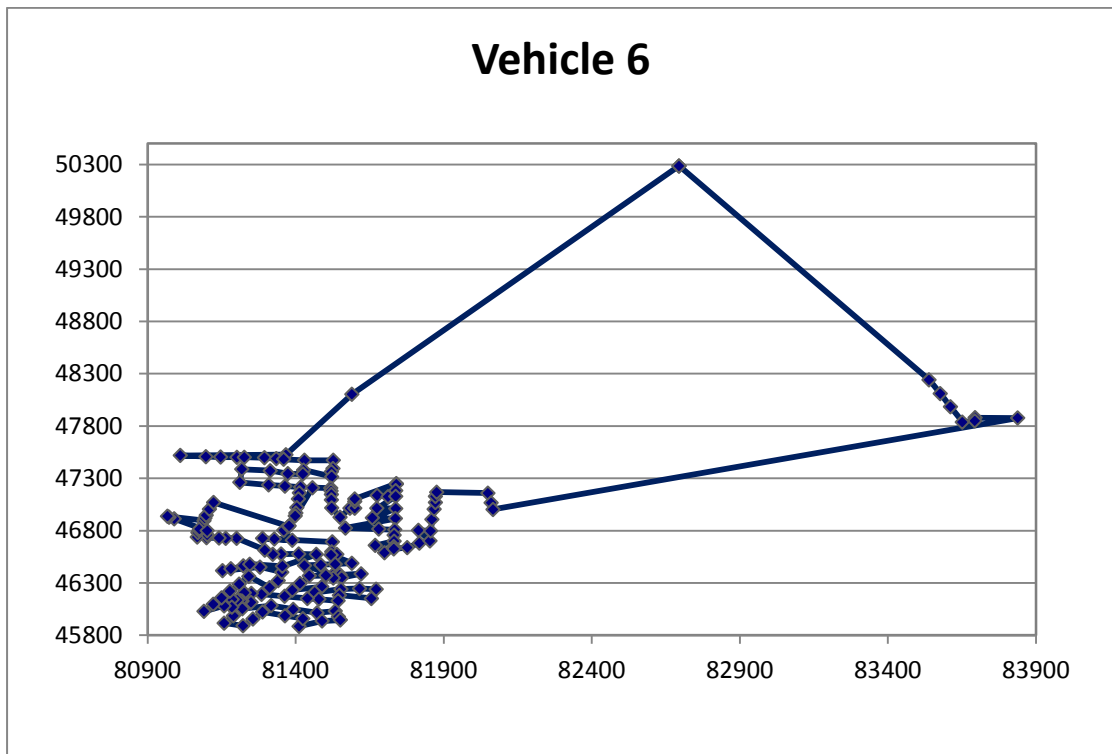
Για το όχημα 4:



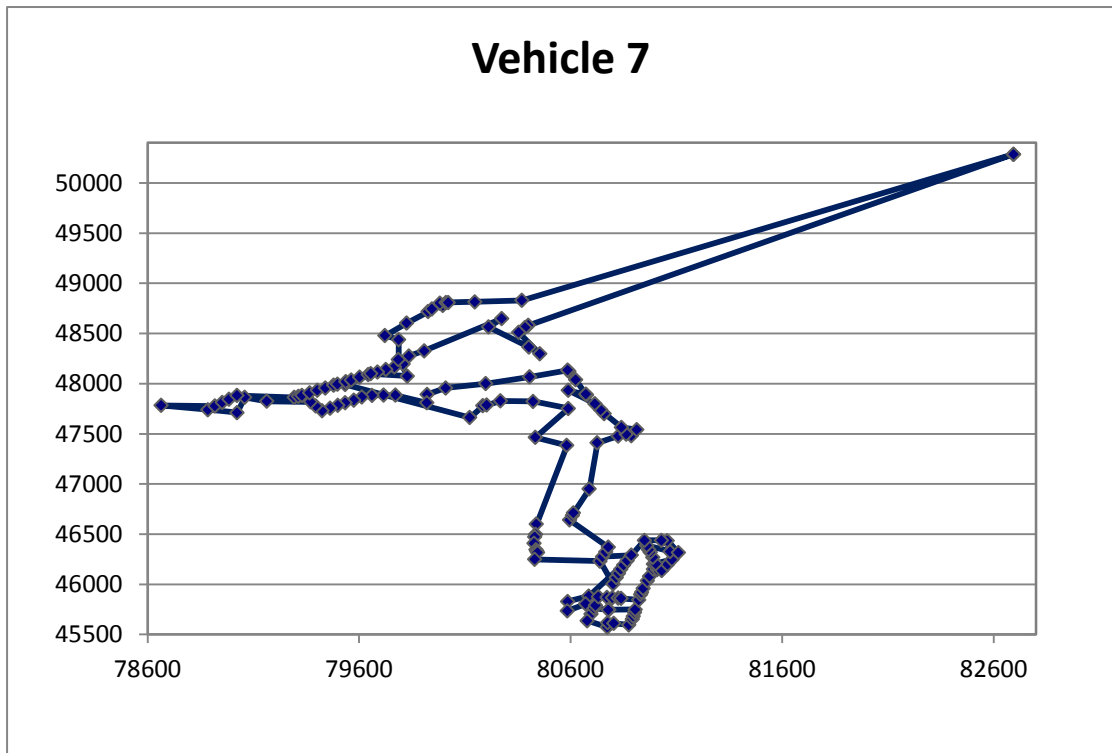
Για το όχημα 5:



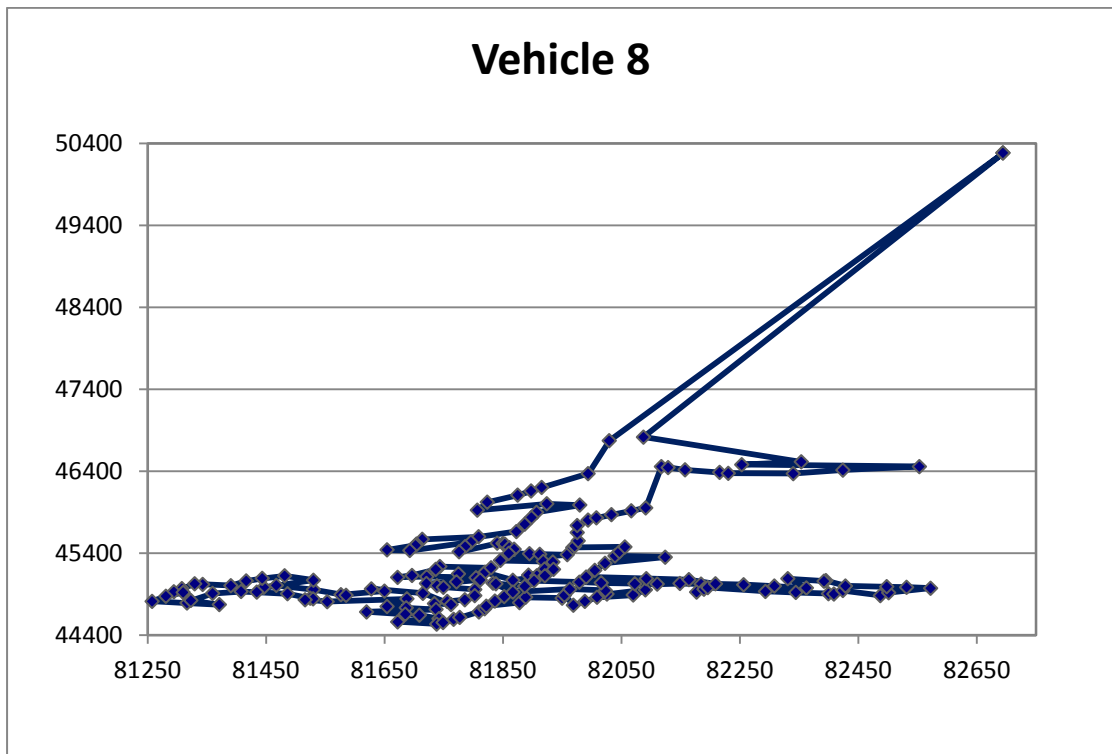
Για το όχημα 6:



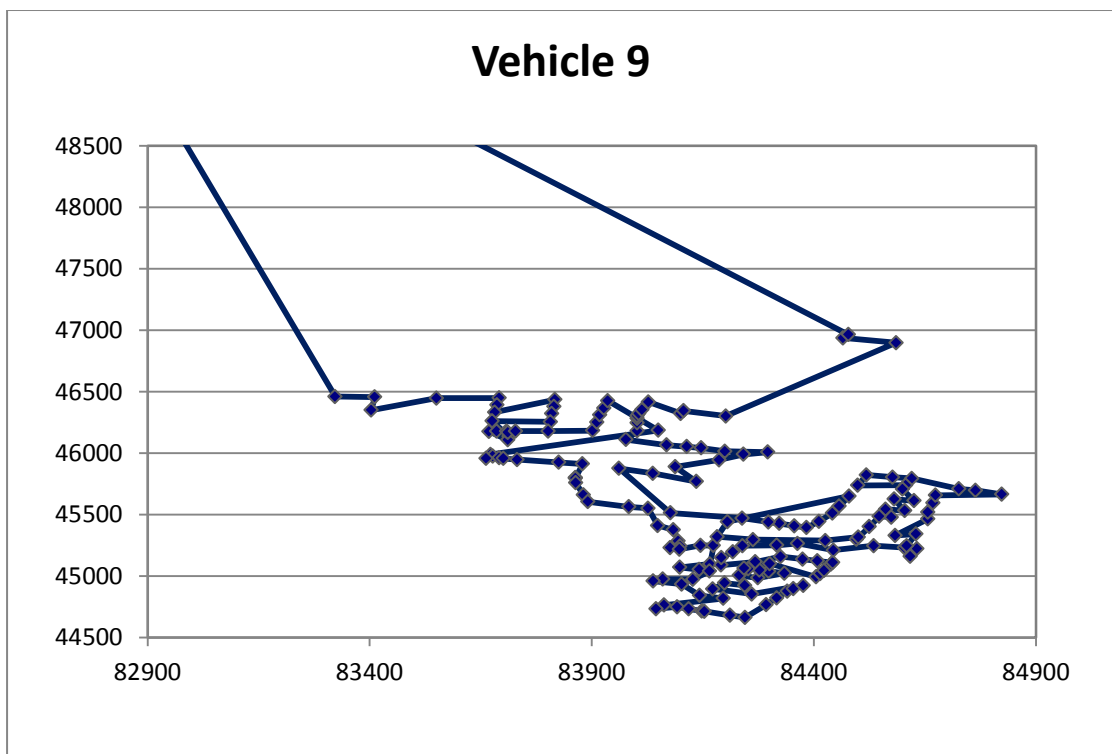
Για το όχημα 7:



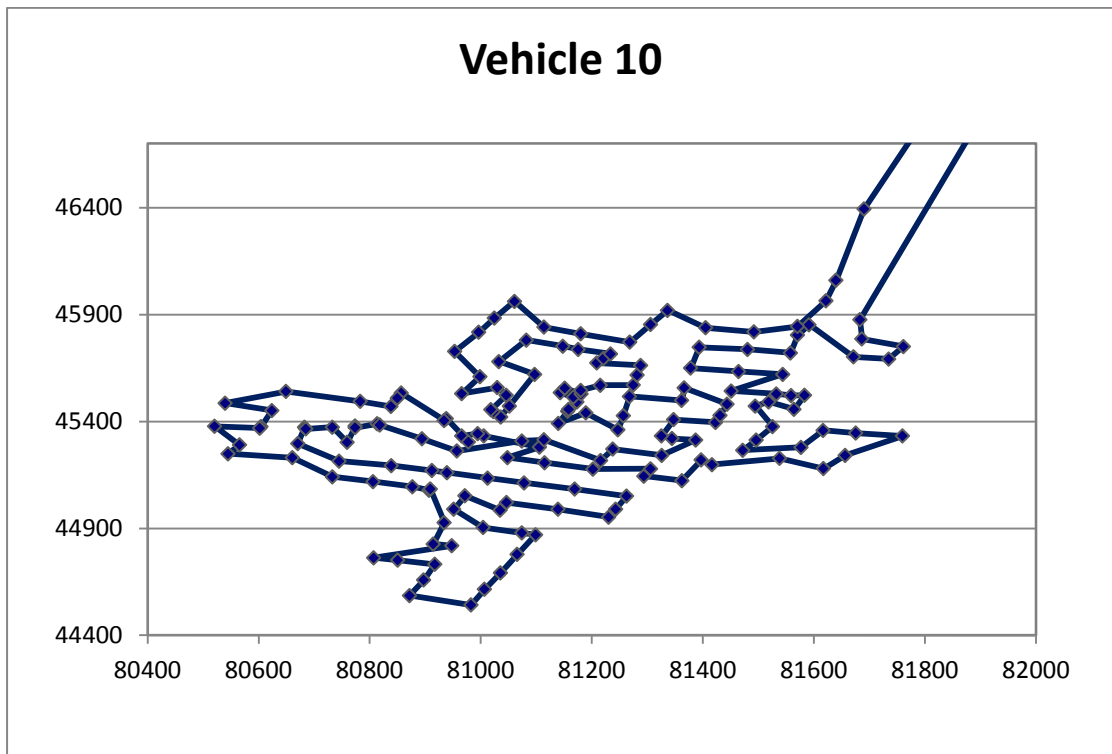
Για το όχημα 8:



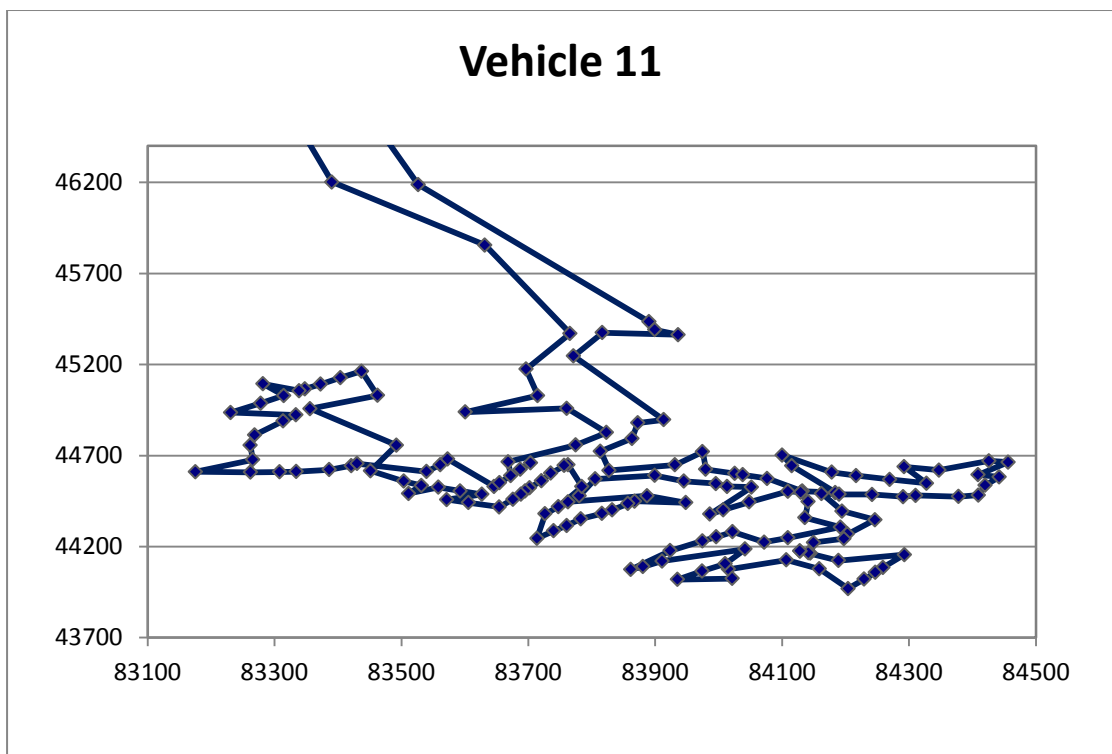
Για το όχημα 9:



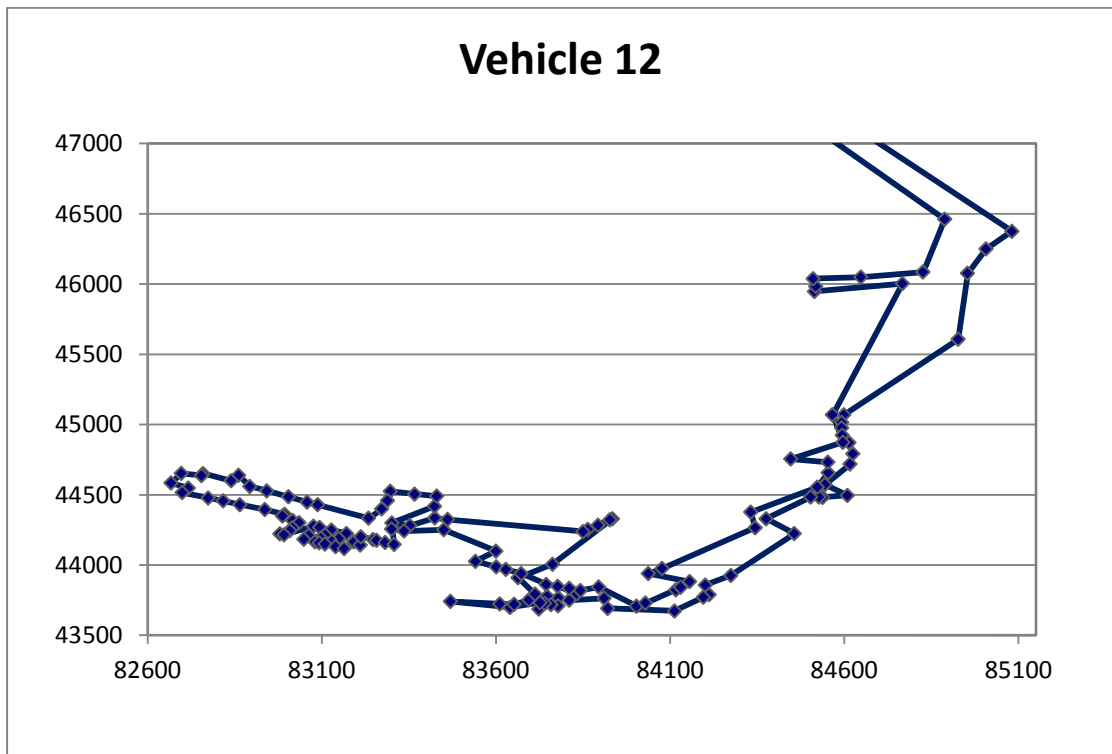
Για το όχημα 10:



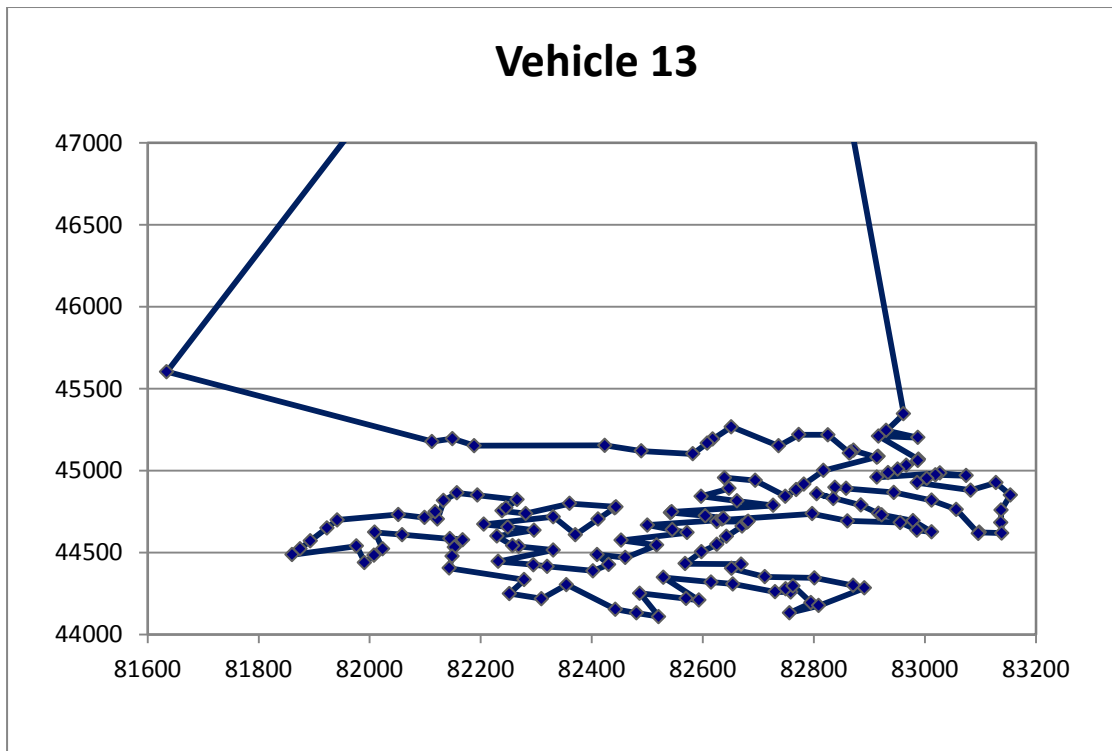
Για το όχημα 11:



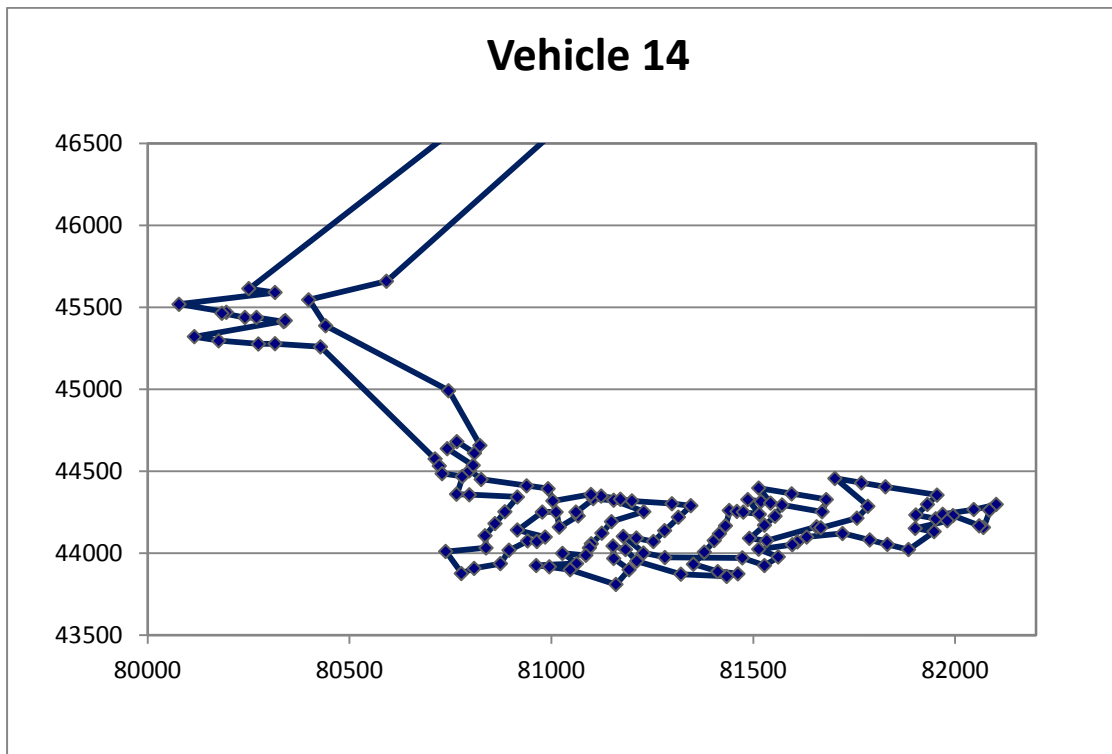
Για το όχημα 12:



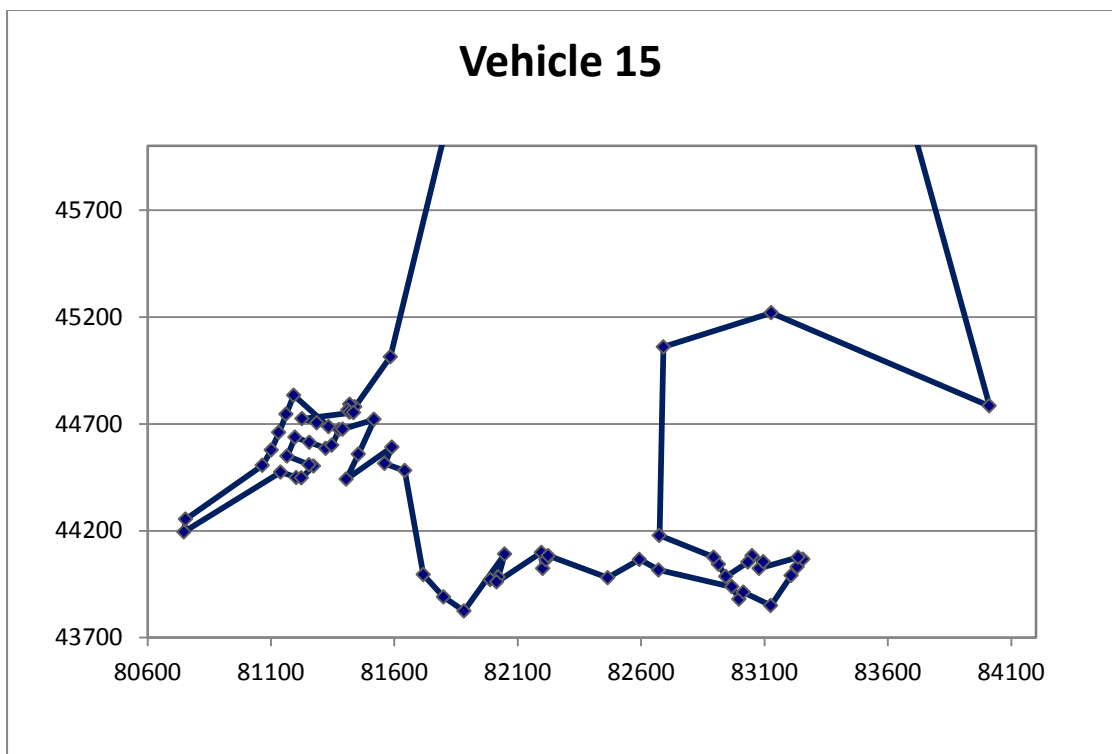
Για το όχημα 13:



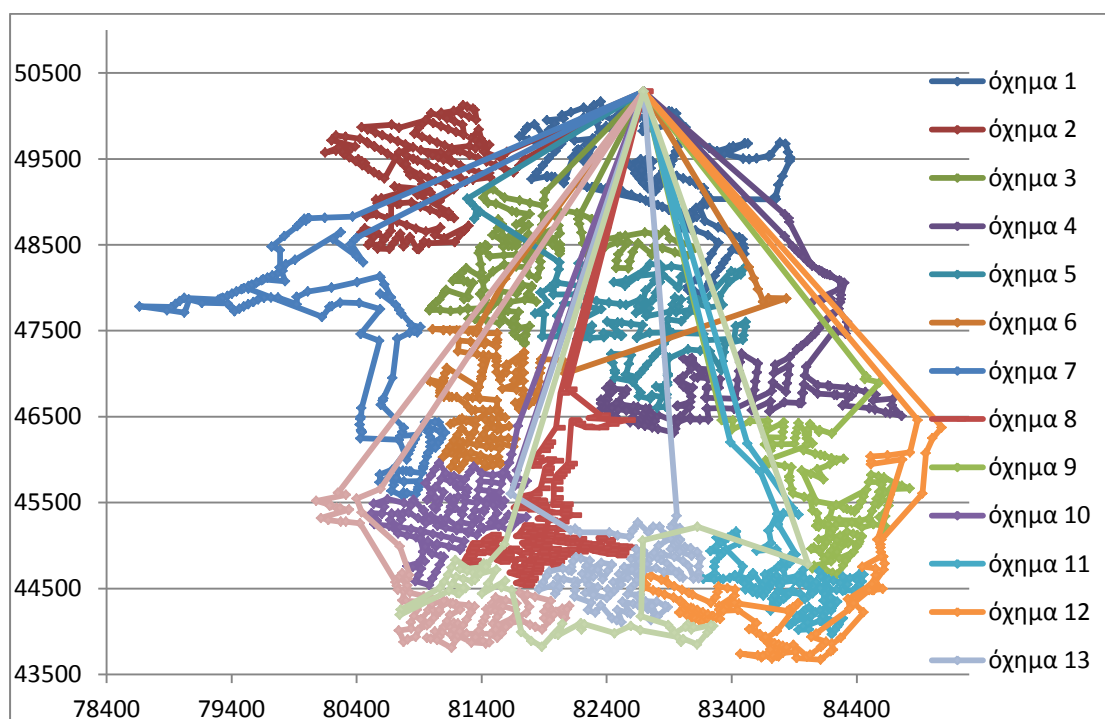
Για το όχημα 14:



Για το όχημα 15:



### Συγκεντρωτικό Διάγραμμα:



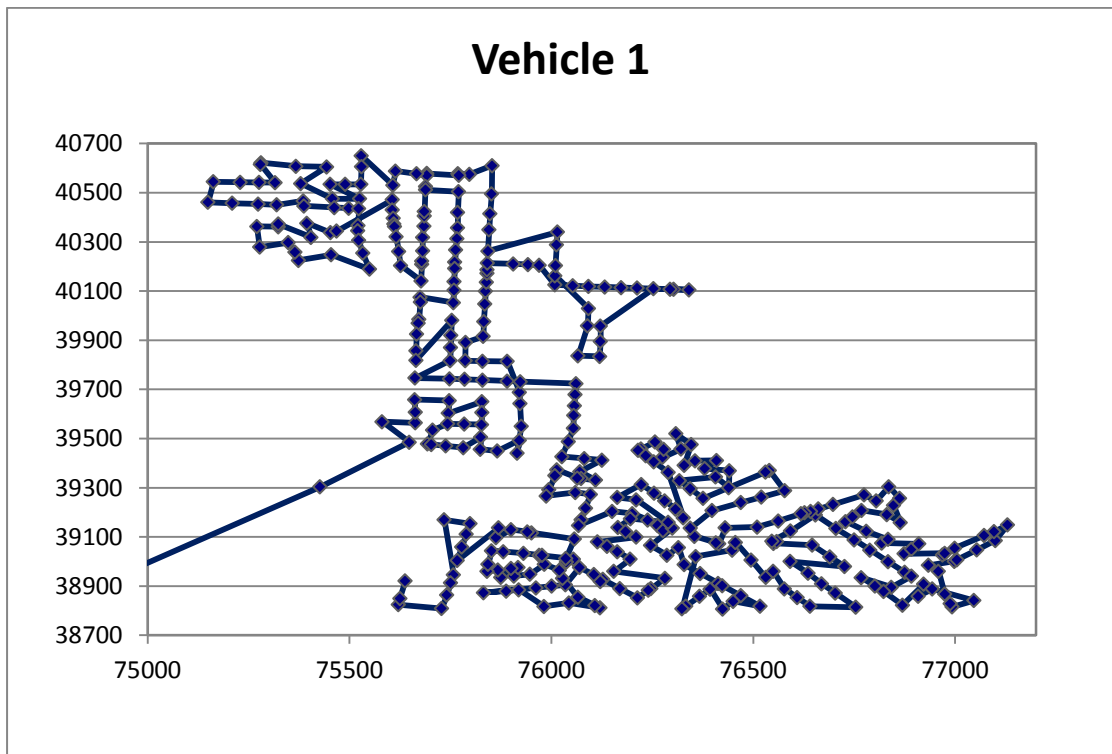
- 3<sup>ο</sup> Παράδειγμα

ΟΧΗΜΑΤΑ	ΤΑΧΥΤΗΤΑ(km/h)	ΦΟΡΤΙΟ	ΚΟΣΤΟΣ(min)
1	40	428	479,912
2	40	427	479,725
3	40	422	479,2715
4	40	409	479,0275
5	40	415	479,1535
6	40	395	478,4793
7	40	4	33,196

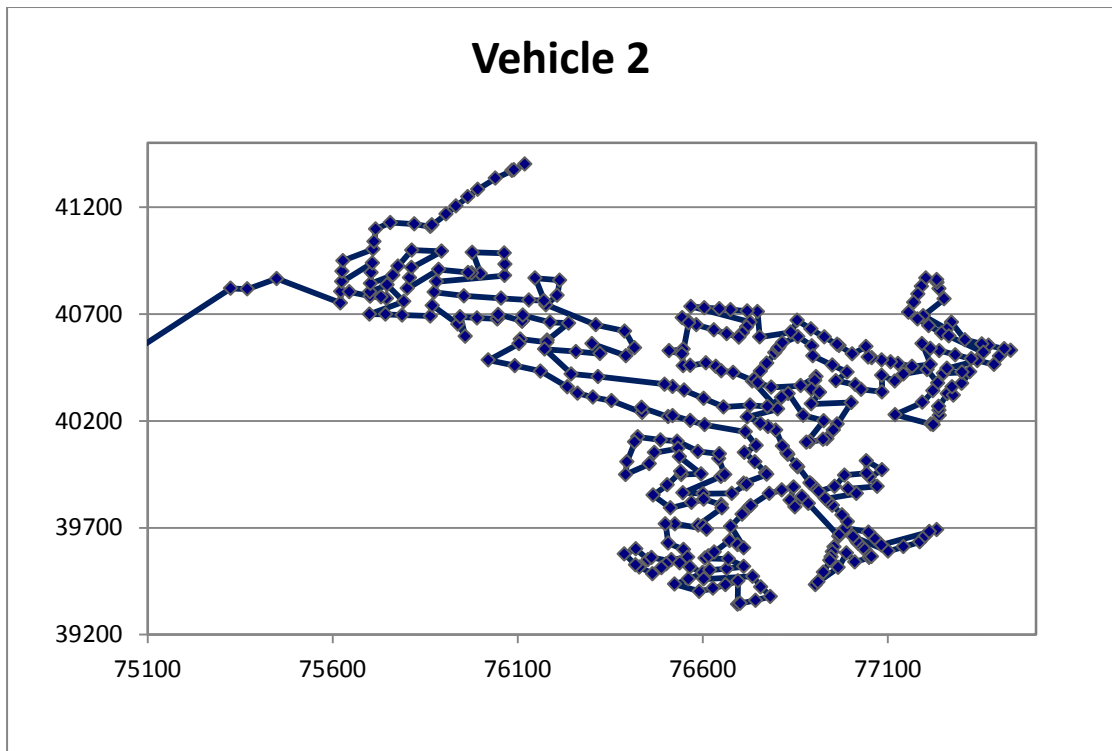
Συνεπώς, εξυπηρετούνται όλοι οι πελάτες (2500) με τη χρήση 7 οχημάτων και συνολικό κόστος (χρόνος) περίπου ίσο με 2906 min. Παρακάτω, παρατίθενται τα διαγράμματα για τη διαδρομή που ακολούθησε το κάθε όχημα καθώς και ένα συγκεντρωτικό διάγραμμα που δείχνει τις διαδρομές όλων των οχημάτων.



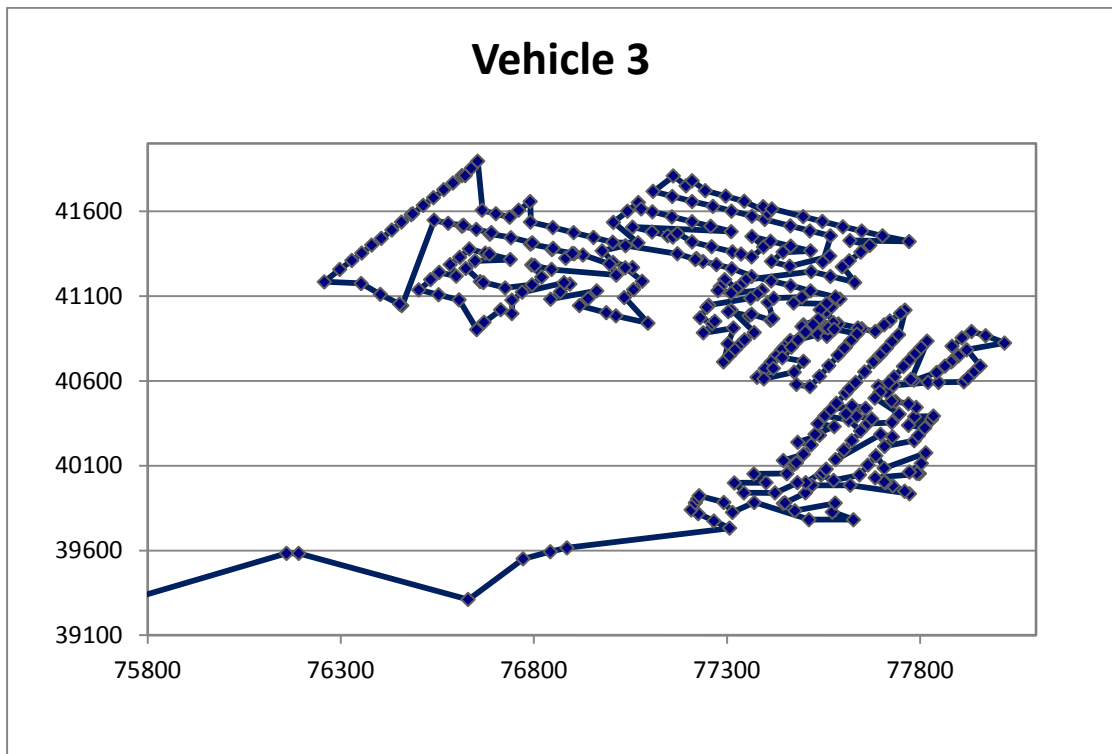
Για το όχημα 1:



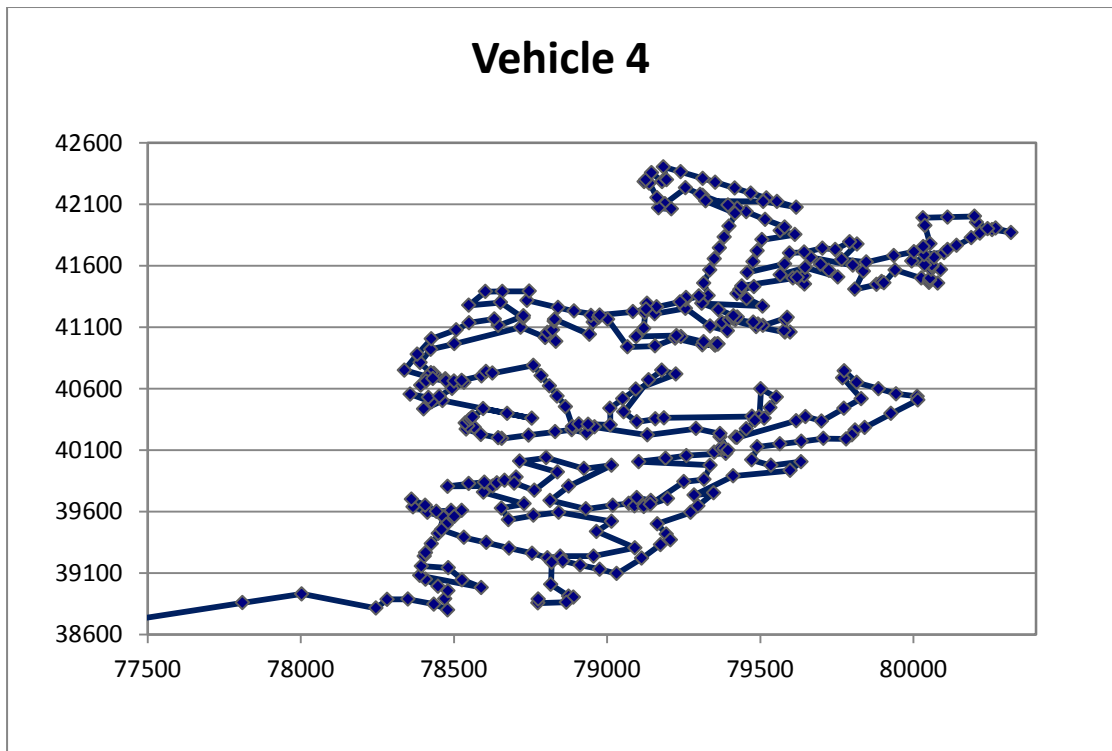
Για το όχημα 2:



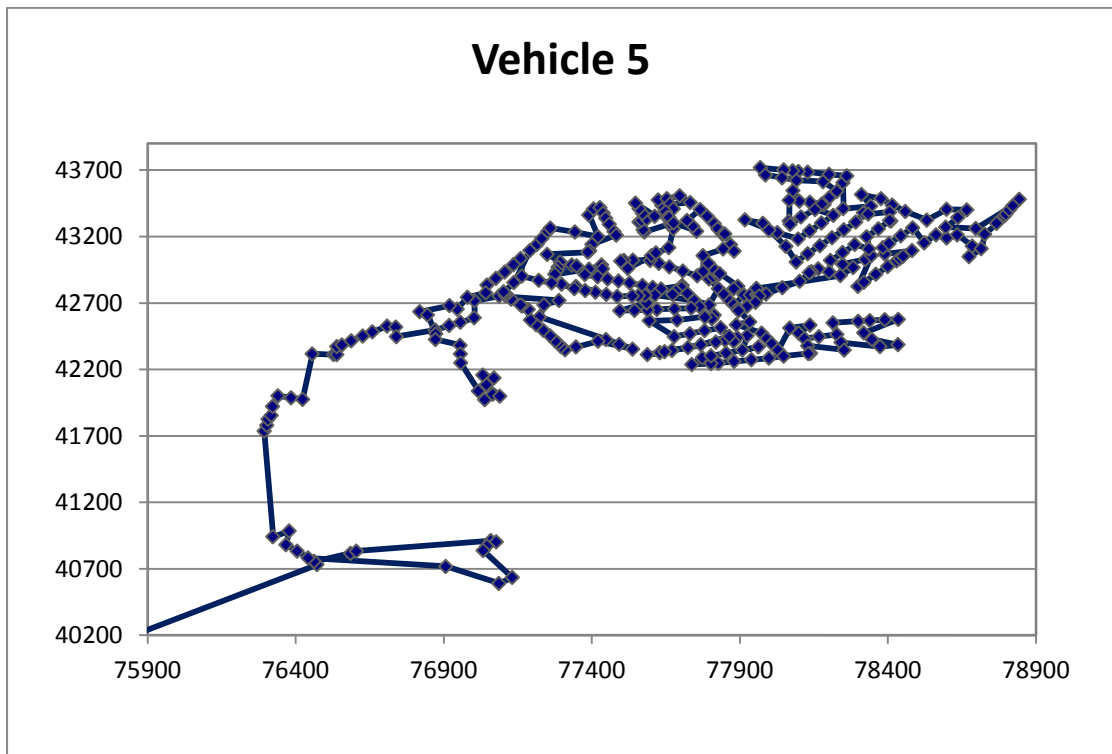
Για το όχημα 3:



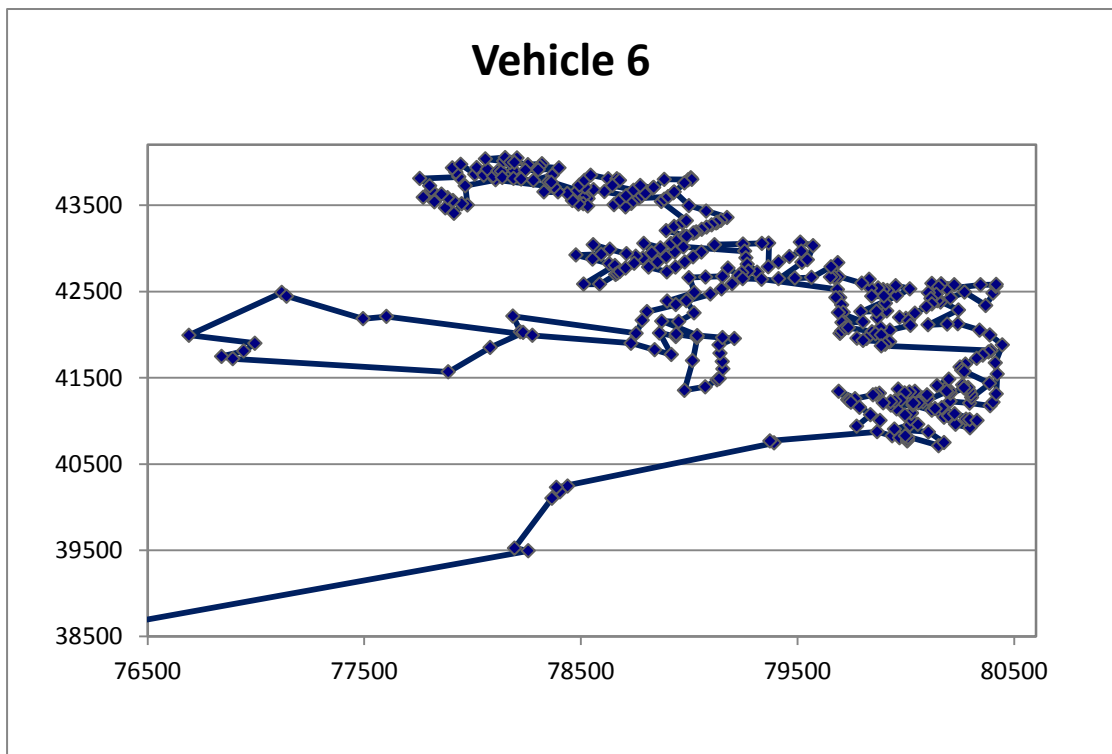
Για το όχημα 4:



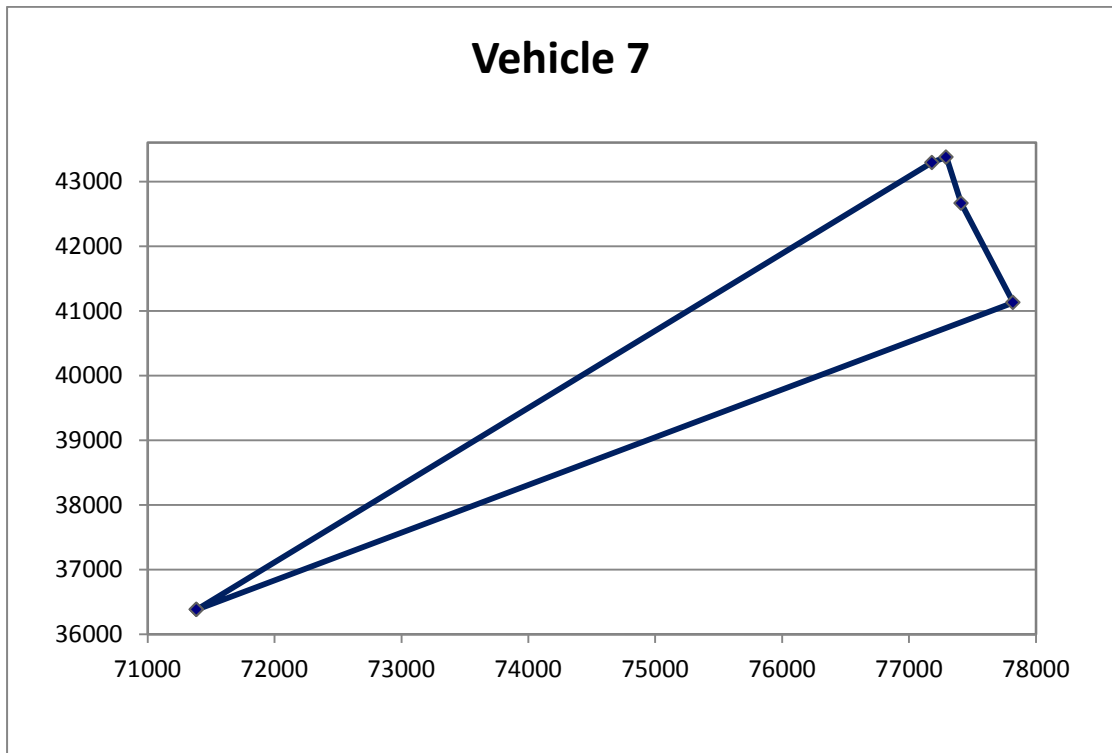
Για το όχημα 5:



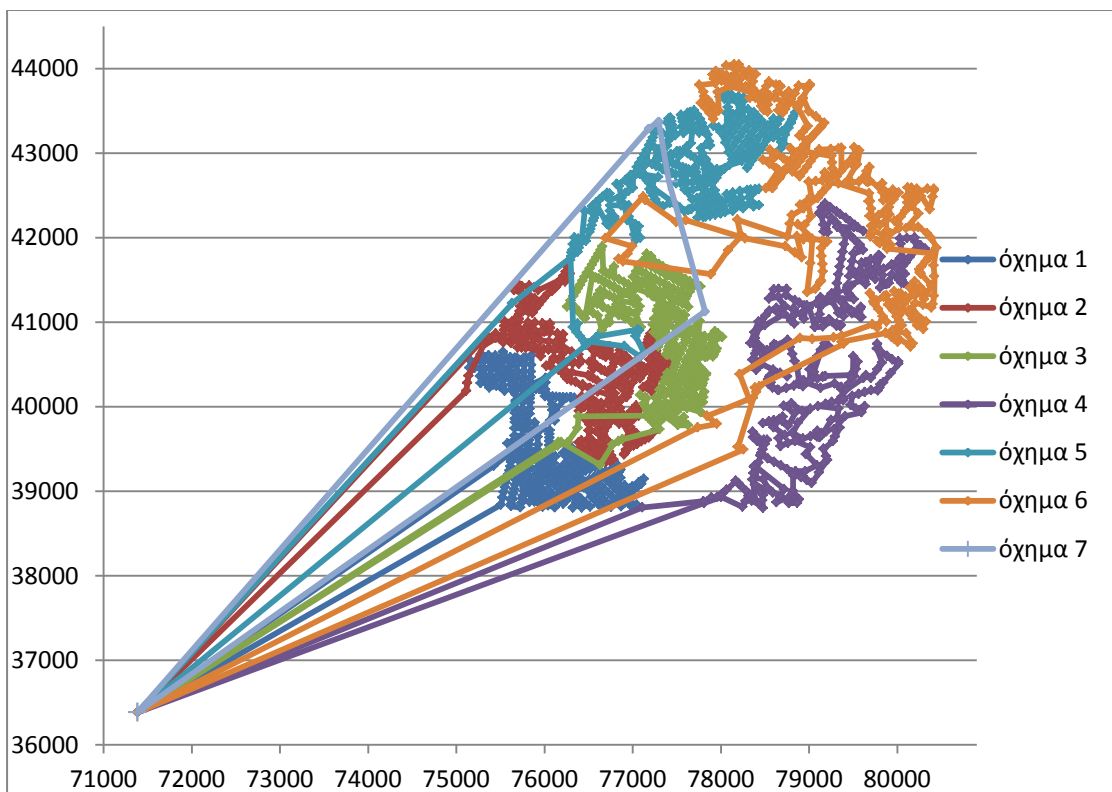
Για το όχημα 6:



Για το όχημα 7:



Συγκεντρωτικό διάγραμμα:

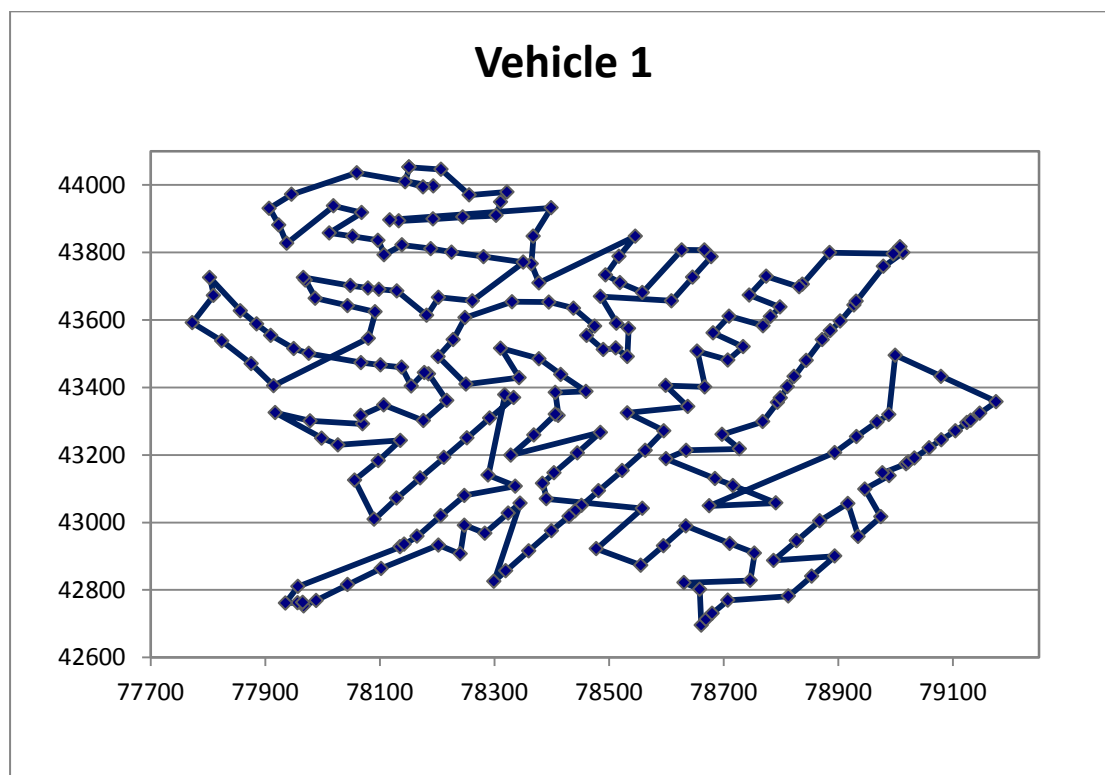


- 4<sup>ο</sup> Παράδειγμα

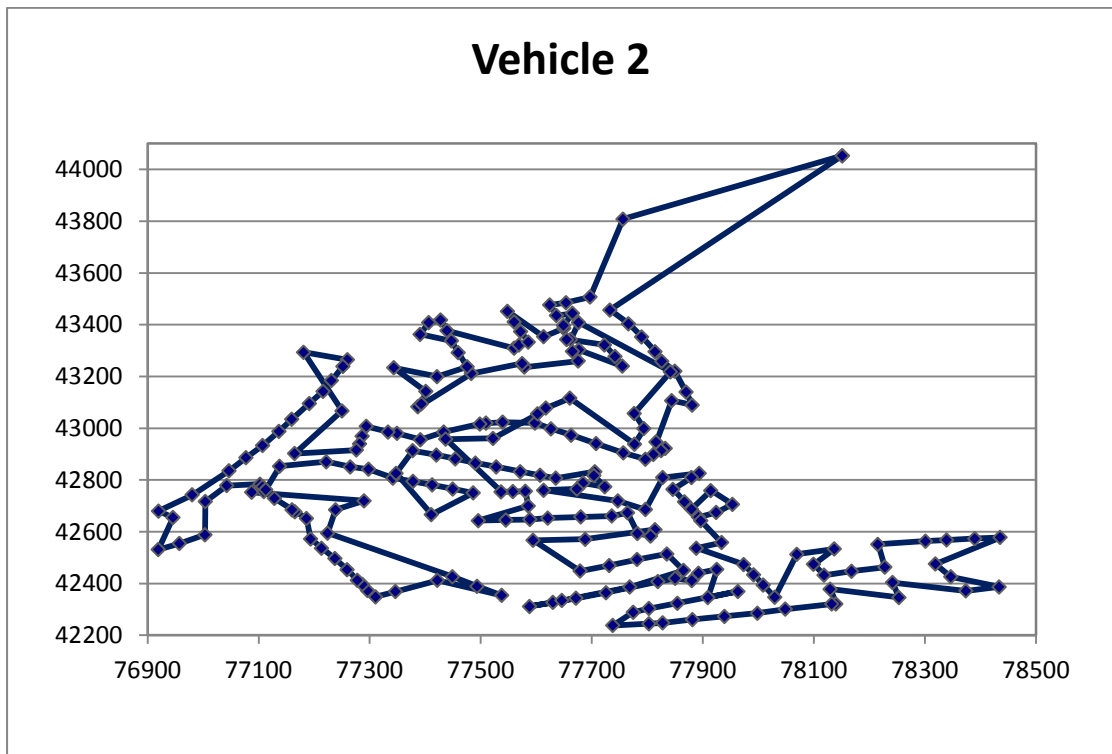
ΟΧΗΜΑΤΑ	ΤΑΧΥΤΗΤΑ(km/h)	ΦΟΡΤΙΟ	ΚΟΣΤΟΣ(min)
1	4	479,43	237
2	4	478,09	232
3	4	479,095	220
4	4	479,455	211
5	4	479,14	184
6	4	479,665	202
7	4	479,33	209
8	4	478,66	160
9	4	478,725	287
10	4	478,715	161
11	4	478,795	184
12	4	479,6	182
13	4	448,215	111

Συνεπώς, εξυπηρετούνται όλοι οι πελάτες (2500) με τη χρήση 13 οχημάτων και συνολικό κόστος (χρόνος) περίπου ίσο με 6191 min. Παρακάτω, παρατίθενται τα διαγράμματα για τη διαδρομή που ακολούθησε το κάθε όχημα καθώς και ένα συγκεντρωτικό διάγραμμα που δείχνει τις διαδρομές όλων των οχημάτων.

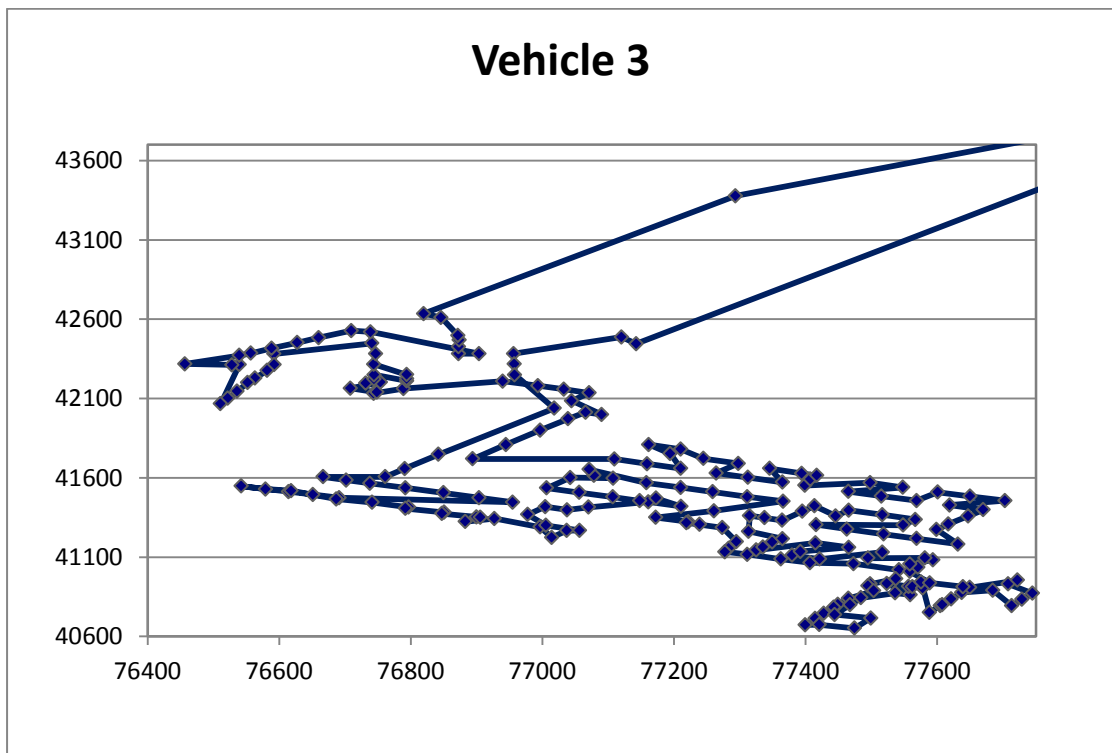
Για το όχημα 1:



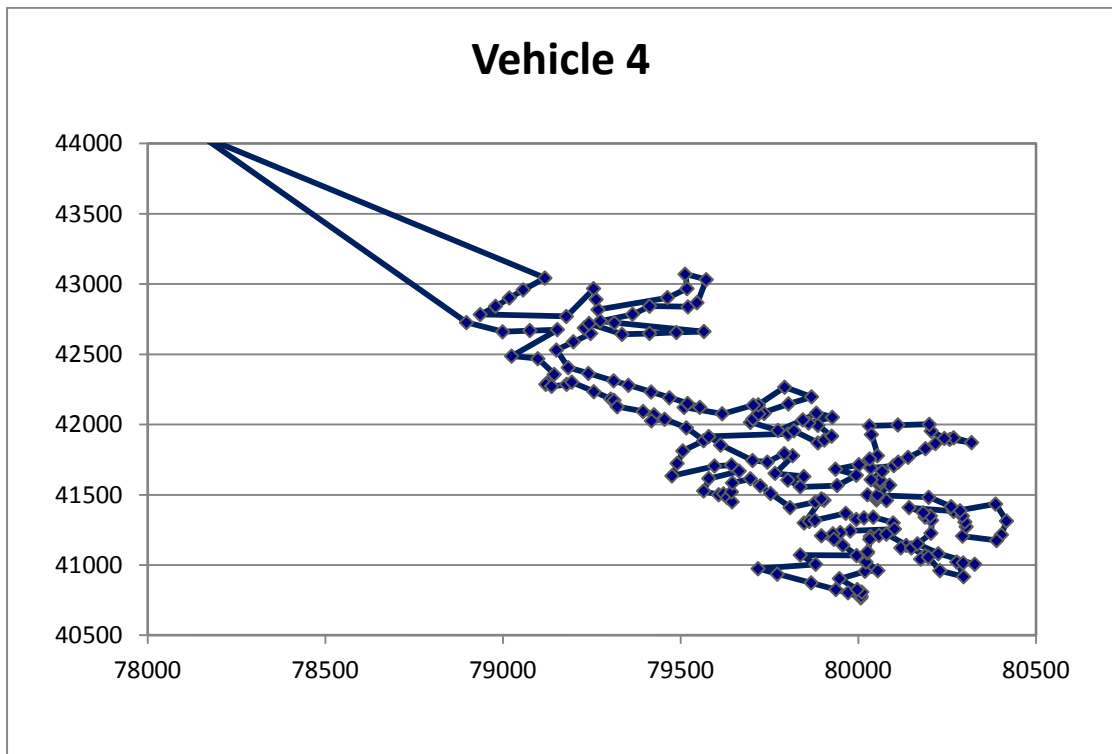
Για το όχημα 2:



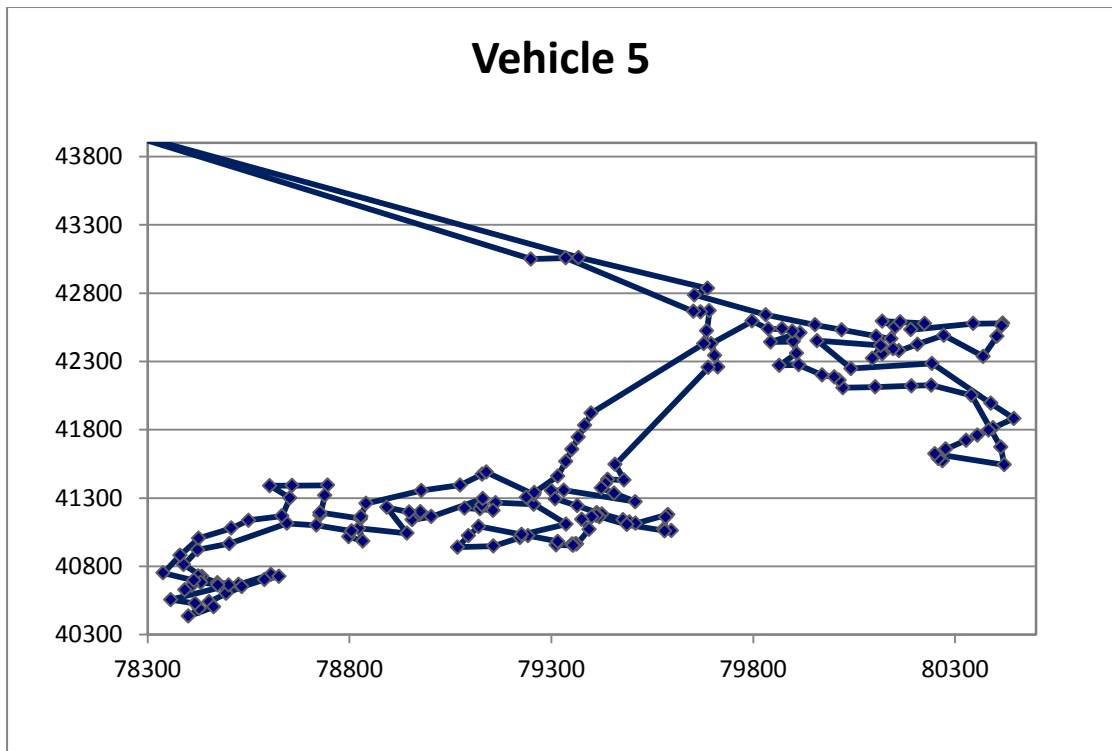
Για το όχημα 3:



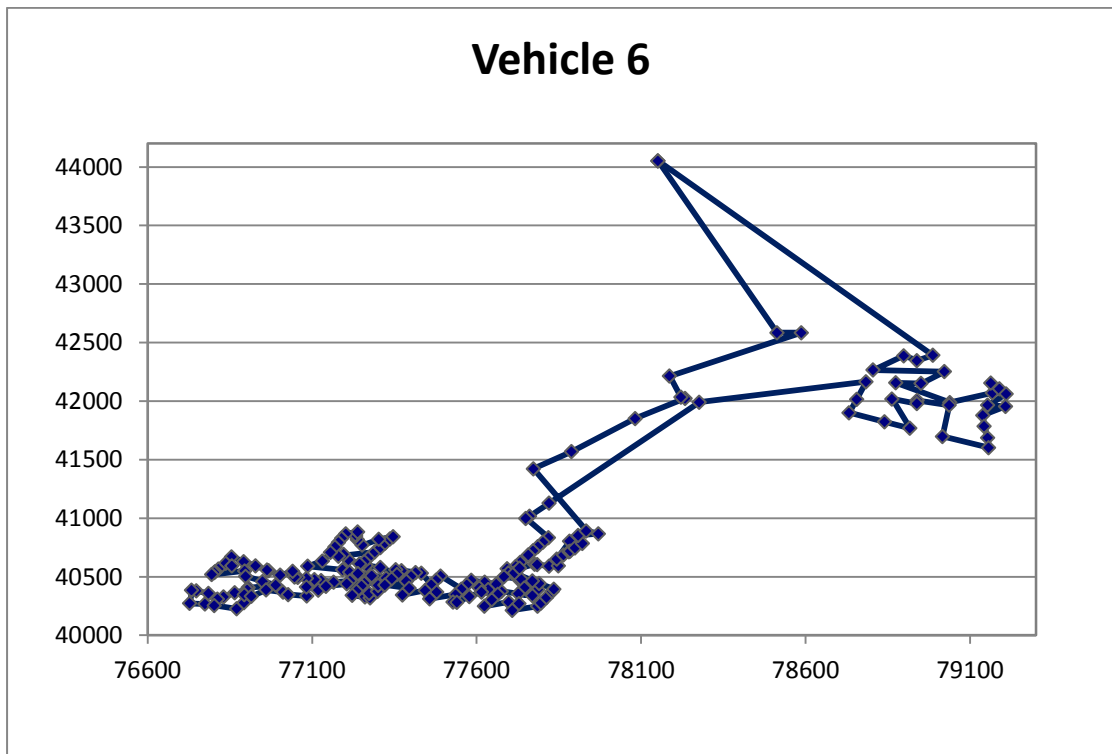
Για το όχημα 4:



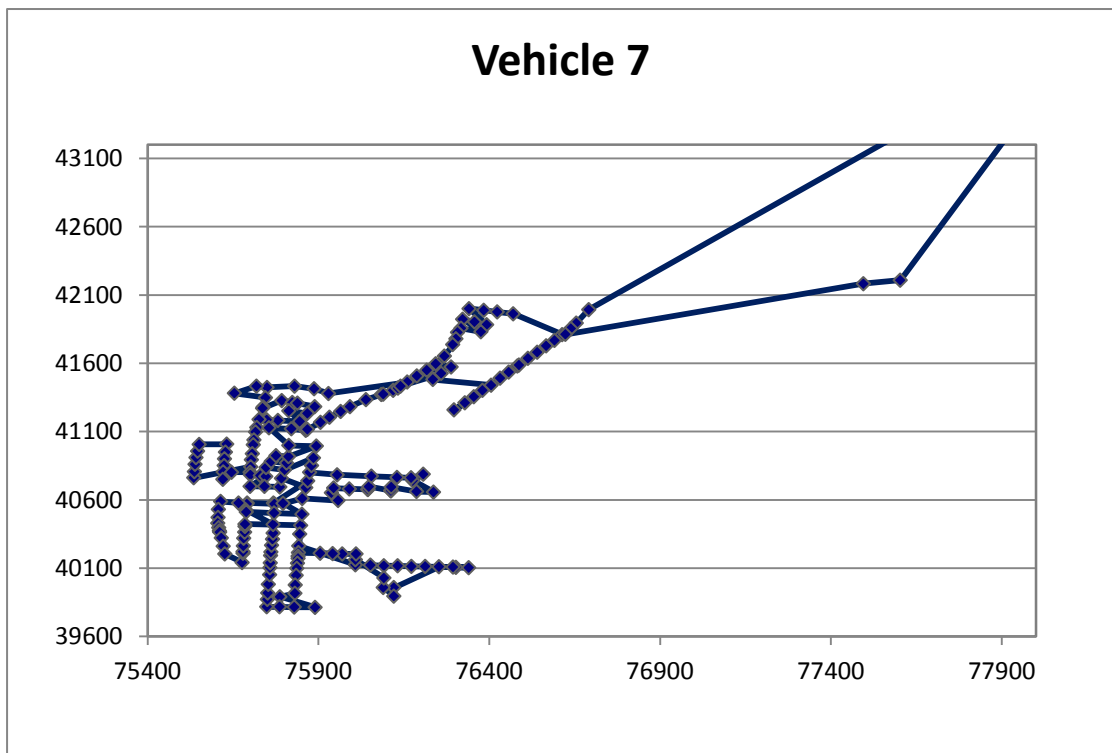
Για το όχημα 5:



Για το όχημα 6:

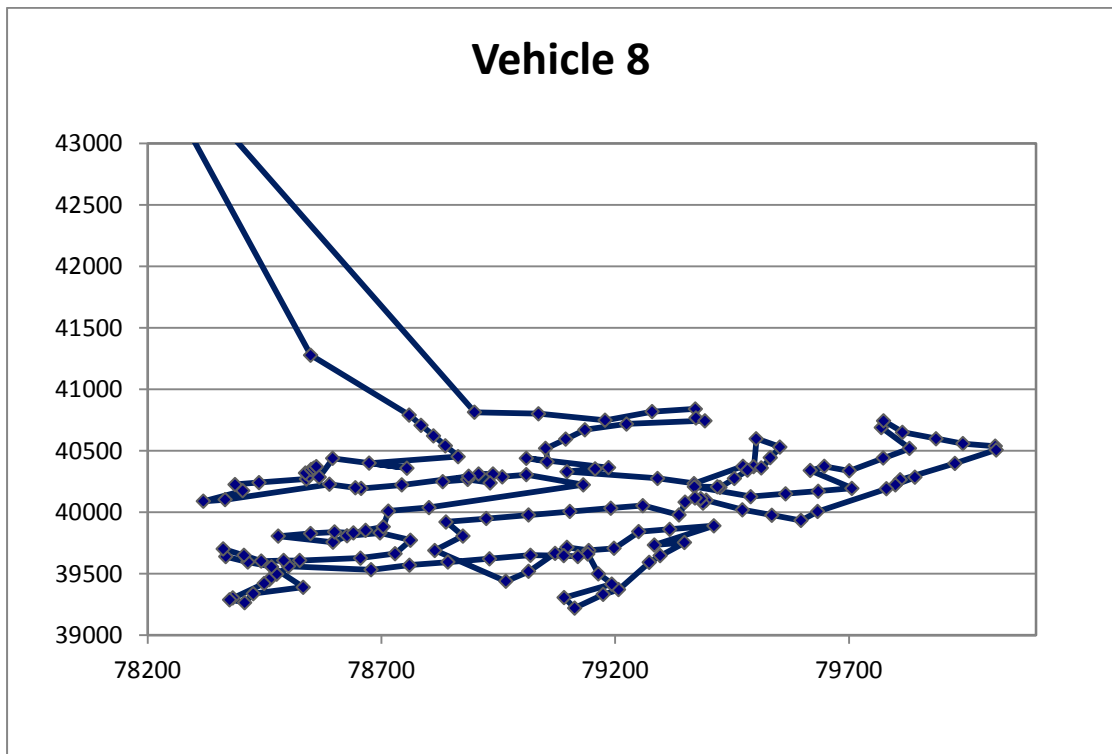


Για το όχημα 7:

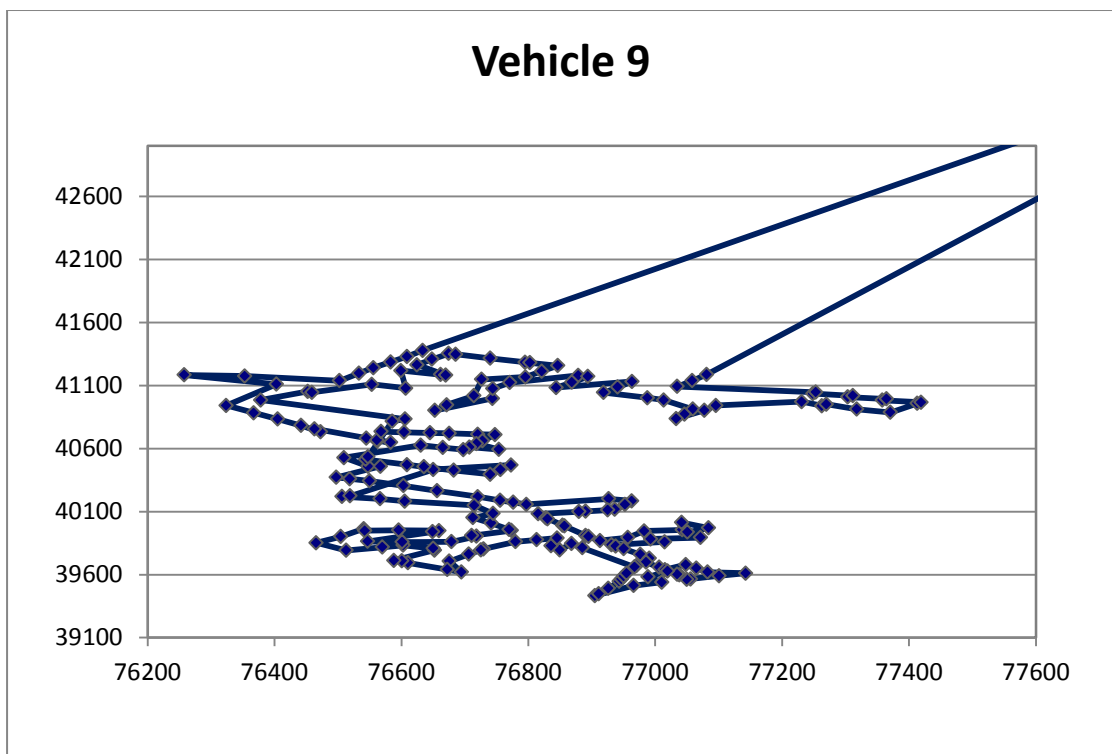




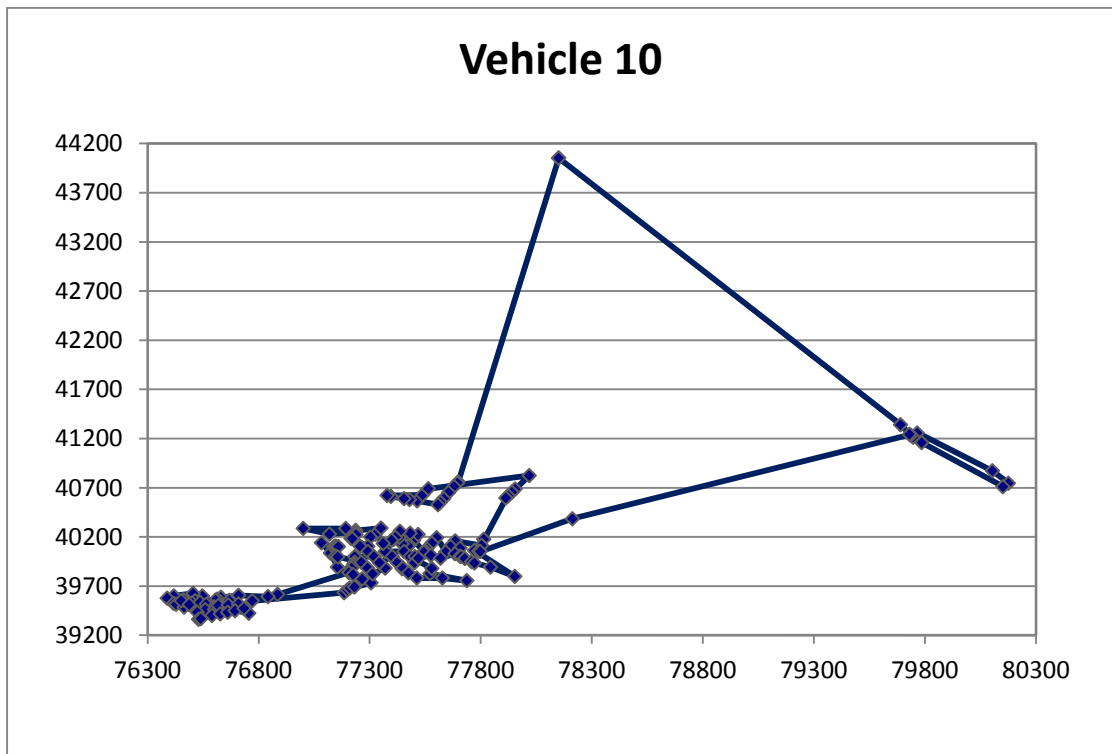
Για το όχημα 8:



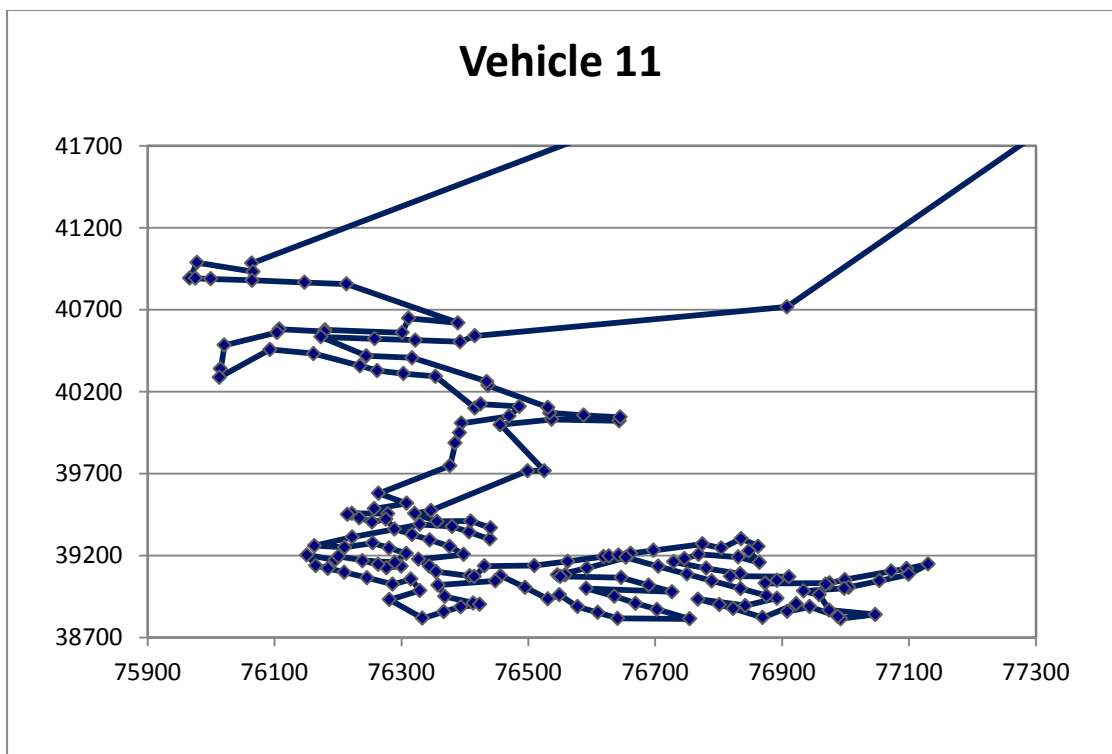
Για το όχημα 9:



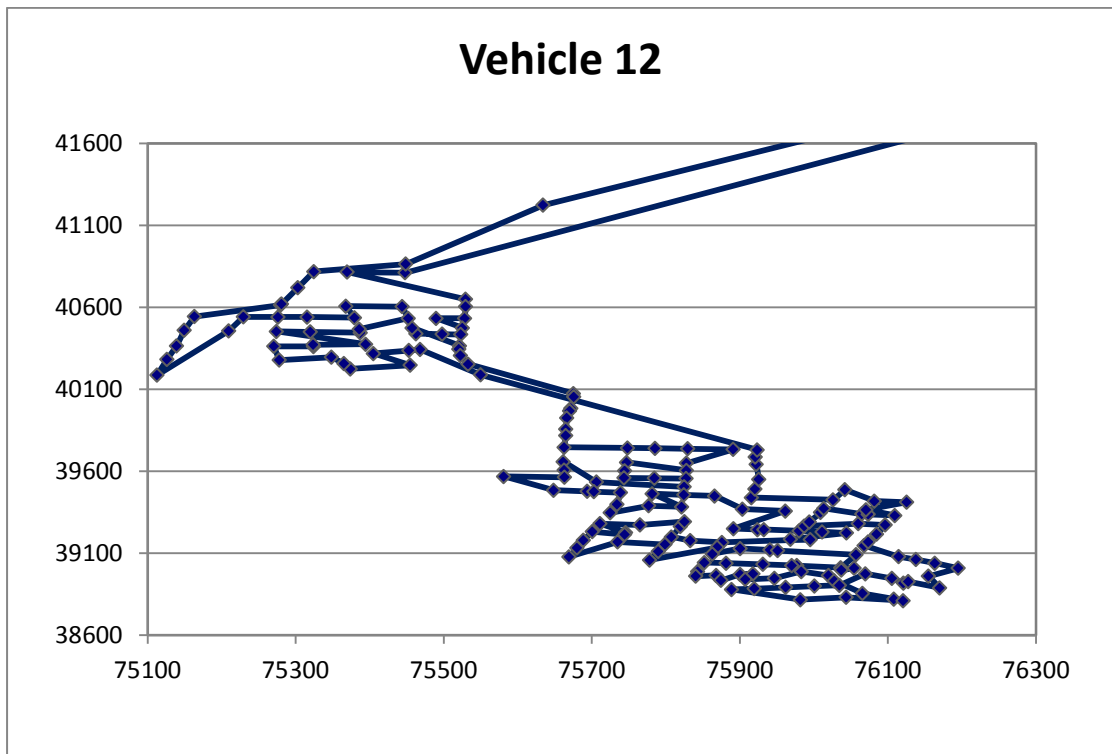
Για το όχημα 10:



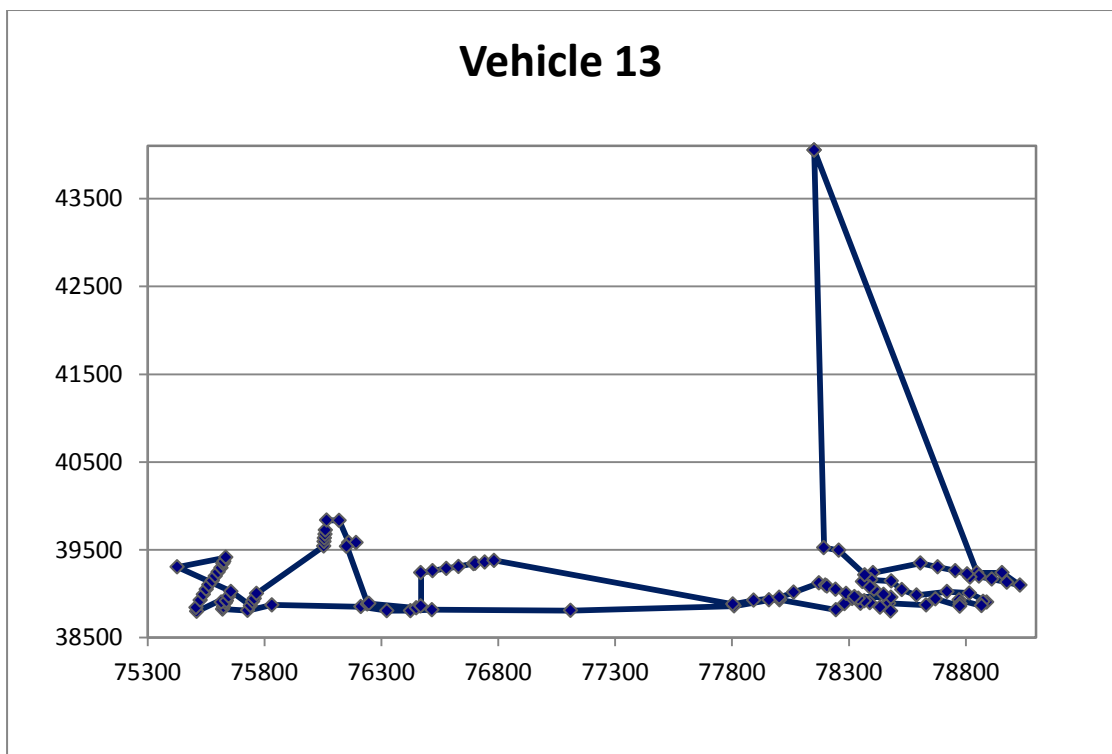
Για το όχημα 11:



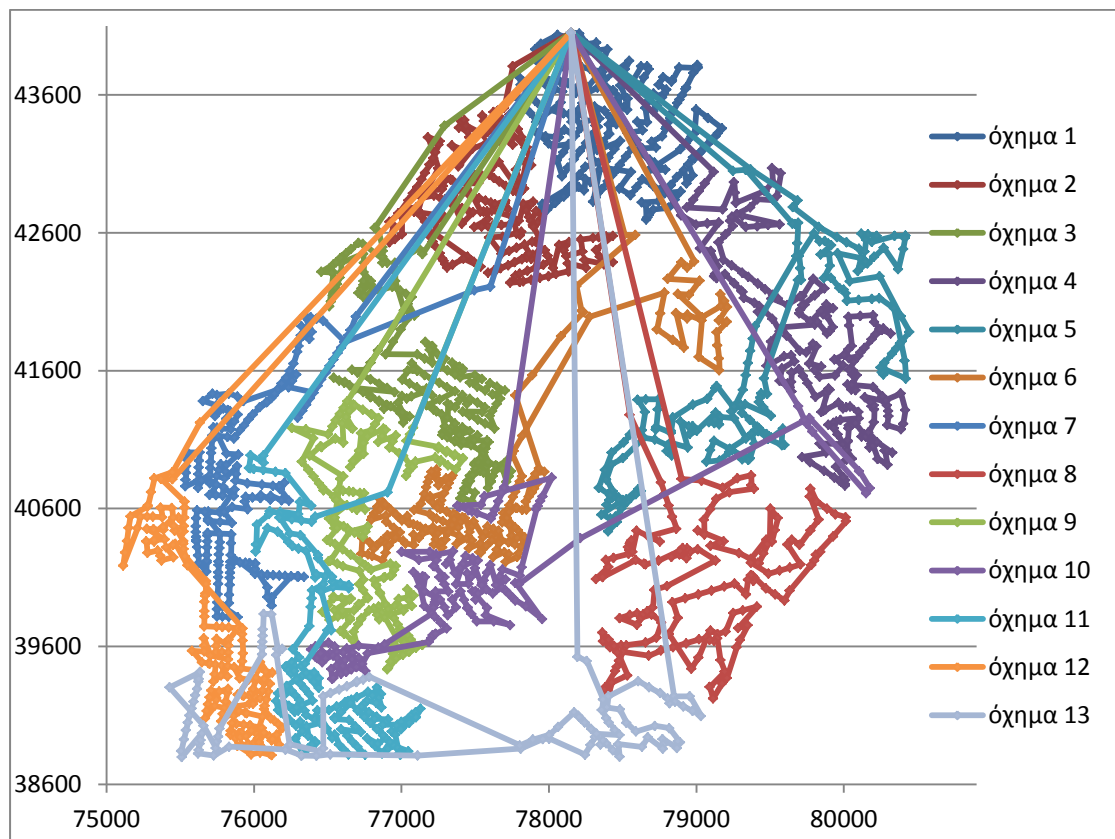
Για το όχημα 12:



Για το όχημα 13:



**Συγκεντρωτικό διάγραμμα:**



## ΣΥΜΠΕΡΑΣΜΑΤΑ

Συμπερασματικά λοιπόν θα λέγαμε, πως τα αποτελέσματα που προέκυψαν από την εφαρμογή του κώδικα στα προηγούμενα παραδείγματα είναι αρκετά ικανοποιητικά. Αξίζει όμως να σημειωθεί πως όσο μειώνεται η ταχύτητα με την οποία κινείται το όχημα-υπάλληλος τόσο αυξάνεται ο αριθμός των απαιτούμενων οχημάτων για να εξυπηρετηθούν όλοι οι πελάτες. Επίσης στο 1<sup>ο</sup> και 3<sup>ο</sup> παράδειγμα οι συντεταγμένες της αποθήκης (office) είναι πολύ απομακρυσμένες από την περιοχή στην οποία βρίσκονται οι συντεταγμένες των πελατών. Η απόσταση αυτή μπορεί να καλυφθεί μόνο με τη χρήση κάποιου οχήματος, στη συγκεκριμένη περίπτωση με μηχανάκι. Για το λόγο αυτό θεωρήσαμε στο 2<sup>ο</sup> και 4<sup>ο</sup> παράδειγμα αντίστοιχα ότι ο υπάλληλος μεταφέρεται με Μ.Μ.Μ (π.χ με λεωφορείο) στην περιοχή που κατοικούν οι πελάτες, θεωρώντας όμως πλέον ως αποθήκη (office) τη στάση του λεωφορείου που θα κατέβει πρώτα, και στη συνέχεια περπατώντας θα εξυπηρετήσει τους πελάτες. Το συγκεκριμένο πρόβλημα βελτιστοποίησης διαδρομών για τη συλλογή των μετρήσεων είναι πολύ σημαντικό στις μέρες μας καθώς υπενθυμίζουμε ότι πρόκειται για μια μέθοδο γραφικής επίλυσης των διαδρομών, χρονοβόρα και προσεγγιστική που περιλαμβάνει διαδικασίες δοκιμής-σφάλματος όπως και εμπειρικές σταθερές και αυθαίρετες παραδοχές. Τέλος, η πρωτοτυπία του εγχειρήματός μας συνοπτικά έγκειται στην εφαρμογή των *heuristics* στην διαδικασία συλλογής των μετρήσεων και την αντικειμενοστραφή υλοποίηση. Και οι δύο παράγοντες επιτρέπουν την επίλυση πολύ μεγάλων προβλημάτων συλλογής μετρήσεων σε ρεαλιστικό χρόνο.

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

1. Κυρανούδης Χ. Εφοδιαστική διαχείριση. Εισαγωγή, βασικές έννοιες, προοπτικές. Σημειώσεις. Αθήνα, Εθνικό Μετσόβιο Πολυτεχνείο (2001).
2. Resende M. GRASP: Greedy Randomized Adaptive Search Procedures. A metaheuristic for combinatorial optimization. Algorithms&Optimization Research Dept. AT&T Labs Research, New Jersey (1999).
3. Hillier F., Lieberman G. Εισαγωγή στην επιχειρησιακή έρευνα. Τόμος Α. Τεύχος Α. Εκδόσεις Παπαζήση (1985).
4. Κάπρος Π. Μοντέλα μαθηματικού προγραμματισμού. Σημειώσεις παραδόσεων Εθνικό Μετσόβιο Πολυτεχνείο (1991).
5. Kotler P. Μάρκετινγκ μανάτζμεντ. Ανάλυση, σχεδιασμός, υλοποίηση και έλεγχος. Τόμος Β. 7<sup>η</sup> Έκδοση, Εκδόσεις Interbooks (1991).
6. Wendell S. Physical Distribution: Key to Improved Volume and Profits. Journal of Marketing. January (1969).
7. Cook W., Cunningham W., Pulleyblank W., Schrijver A. Combinatorial Optimization. Wiley Interscience Series in Discrete Mathematics and Optimization, USA (1998).
8. Fink A., Voss S., Woodruff D. An Adoption Path for Intelligent Heuristic Search Componentware – Concepts and Applications. Virtual Environments for Advanced Modeling, August 2-4, Monterey, California, USA (2000).
9. McBride P. Programming in Visual Basic. Guernsey Press Company Ltd, Great Britain (1994).
10. Clark J., Webb C., Webb J. Microsoft Visual Basic 6.0 Step by Step. Microsoft Press, U.S.A. (1998).
11. Calvo R., Cordone R. A heuristic approach to the overnight security service problem. Computers & Operations Research 30 (2003).
12. Μανωλόπουλος Ι. Δομές δεδομένων. Τόμος Α. Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Εκδόσεις Art of Text (1992).
13. Resende M. A GRASP for the Maximum Covering Problem. AT&T Labs Research, New Jersey (1997).

## Παράρτημα

Εδώ παρατίθεται ο κώδικας που χρησιμοποιήθηκε για την εύρεση των βέλτιστων διαδρομών.

```
Option Explicit
```

```
Public clients As Collection
```

```
Public result As solution
```

```
Public depot As client
```

```
Public WorkHours As Double
```

```
Private Sub Class_Initialize()
```

```
    Set clients = New Collection
```

```
    Set result = New solution
```

```
    WorkHours = 8
```

```
End Sub
```

```
' Διαβάζει τα δεδομένα από το φύλλο 1
```

```
Sub read()
```

```
    Dim nn As Long, i As Long
```

```
    ' Δεδομένα έδρας
```

```
    Set depot = New client
```

```
    depot.demand = Worksheets(1).Cells(2, 1).Value
```

```
    depot.x = Worksheets(1).Cells(2, 2).Value
```

```
    depot.y = Worksheets(1).Cells(2, 3).Value
```

```
    depot.name = "Office"
```

```
    depot.aa = 0
```

```
depot.bb = 0
```

```
depot.st = 0
```

```
' Δεδομένα πελατών
```

```
nn = Worksheets(1).Cells(5, 2).Value
```

```
For i = 1 To nn
```

```
    Dim cc As client
```

```
    Set cc = New client
```

```
    cc.demand = Worksheets(1).Cells(i + 6, 1).Value
```

```
    cc.x = Worksheets(1).Cells(i + 6, 2).Value
```

```
    cc.y = Worksheets(1).Cells(i + 6, 3).Value
```

```
    cc.name = Worksheets(1).Cells(i + 6, 4).Value
```

```
    cc.st = Worksheets(1).Cells(i + 6, 5).Value
```

```
    cc.aa = 5
```

```
    cc.bb = 10
```

```
    clients.Add cc
```

```
Next i
```

```
' Δεδομένα οχημάτων
```

```
nn = Worksheets(1).Cells(5, 8).Value
```

```
For i = 1 To nn
```

```
    Dim vv As vehicle
```

```
    Set vv = New vehicle
```

```
    vv.name = Worksheets(1).Cells(i + 6, 7).Value
```

```
    vv.speed = Worksheets(1).Cells(i + 6, 8).Value
```

```
    vv.cost = 0
```

```
    vv.load = 0
```

```
' Το πρώτο και το τελευταίο σημείο κάθε φορτηγού είναι η αποθήκη
```

```
vv.clients.Add depot
```



```
vv.clients.Add depot
result.vehicles.Add vv
Next i
End Sub
```

'Υπολογίζει την βέλτιστη διαδρομή

```
Sub insertion()
```

```
Dim i As Long, j As Long, k As Long, i1 As Long, i2 As Long, i3 As Long
```

```
Dim d1 As Long, d2 As Long, d12 As Long, d3 As Long, dd As Long
```

```
' Set result = New solution
```

' Δημιουργεί τη συλλογή με τους πελάτες προς εξυπηρέτηση

```
Dim chelp As Collection
```

```
Set chelp = New Collection
```

' Δημιουργεί τη συλλογή με τους πελάτες που δεν θα εξυπηρετηθούν

```
Dim missing As Collection
```

```
Set missing = New Collection
```

```
For i = 1 To clients.Count
```

```
Dim cl As client
```

```
Set cl = clients.Item(i)
```

```
chelp.Add cl
```

```
Next i
```

```
While chelp.Count > 0
```

'Εμφανίζει στο κελί E1 του τρίτου φύλλου το πλήθος των πελατών που απομένει να προστεθούν στη λύση

```
Worksheets(3).Cells(1, 5) = chelp.Count
```

```
i1 = -1
```

```
i2 = -1
```

i3 = -1

dd = 1000000

' Για κάθε πελάτη που απομένει να εξυπηρετηθεί

For i = 1 To chelp.Count

Dim c1 As client

Set c1 = chelp.Item(i)

' Για κάθε όχημα

For j = 1 To result.vehicles.Count

Dim v1 As vehicle

Set v1 = result.vehicles.Item(j)

If v1.load + c1.demand <= depot.demand Then

' Για κάθε σημείο που θα επισκευτεί το όχημα

For k = 1 To v1.clients.Count - 1

Dim c2, c3 As client

Set c2 = v1.clients.Item(k)

Set c3 = v1.clients.Item(k + 1)

'Υπολογίζει τις αποστάσεις του πελάτη από το σημείο αυτό και το επόμενο

d1 = Sqr((c2.x - c1.x) ^ 2 + (c2.y - c1.y) ^ 2)

d2 = Sqr((c1.x - c3.x) ^ 2 + (c1.y - c3.y) ^ 2)

d12 = Sqr((c2.x - c3.x) ^ 2 + (c2.y - c3.y) ^ 2)

' Υπολογίζει την πρόσθετη απόσταση που θα διανυθεί

d3 = d1 + d2 - d12

Dim ClientTime As Double, VehSpeed As Double

VehSpeed = v1.speed \* 1000 / 60

ClientTime = d3 / VehSpeed + c1.st

'cost = 0 χρόνος που θα χρειαστεί το όχημα για την εξυπηρέτηση των πελατών

'Αν ο χρόνος που απαιτείται είναι μικρότερος από τον μέγιστο διαθέσιμο

```

' τότε εξέτασε αν η απόσταση είναι μικρότερη της ελάχιστης
If v1.cost + ClientTime <= WorkHours * 60 Then
' Αν η απόσταση d3 είναι μικρότερη της ελάχιστης ορίζει
' αυτόν τον πελάτη σαν το επόμενο σημείο που θα επισκευτεί

    If d3 < dd Then

        dd = d3

        i1 = i

        i2 = j

        i3 = k

    End If

End If

Next k

End If

Next j

Next i

' Αν δεν μπορεί να επισκευτεί άλλον πελάτη τους βάζει στη συλλογή missing
If i1 = -1 Then

    For i = chelp.Count To 1 Step -1

        Dim c4 As client

        Set c4 = chelp.Item(i)

        chelp.Remove (i)

        missing.Add c4

    Next i

' Διαφορετικά αφαιρεί τον πελάτη από τους πελάτες προς εξυπηρέτηση
' και τον βάζει στους πελάτες που θα επισκευτεί το φορτηγό

Else

    Dim ss As client

```

```

Set ss = chelp.Item(i1)

chelp.Remove (i1)

Dim ss1 As vehicle

Set ss1 = result.vehicles.Item(i2)

'Υπολογισμός του κόστους (χρόνου) που βαρύνει το όχημα ss1
'λαμβάνοντας υπόψη το νέο πελάτη που ss που εξυπηρετεί

ss1.cost = ss1.cost + dd * 60 / ss1.speed / 1000 + ss.st

ss1.load = ss1.load + ss.demand

ss1.clients.Add ss, , i3

End If

Wend

Dim MyRow As Long, sRange As String, myRange As Range

MyRow = 2

Worksheets(3).UsedRange.Clear

' Για κάθε φορτηγό προσθέτει το κόστος και το φορτίο στη λύση
' και παρουσιάζει τα αποτελέσματα

For i = 1 To result.vehicles.Count

Dim vv2 As vehicle

Set vv2 = result.vehicles.Item(i)

result.cost = result.cost + vv2.cost

result.load = result.load + vv2.load

Worksheets(3).Cells(MyRow - 1, 1) = "VehNum"

Worksheets(3).Cells(MyRow - 1, 2) = "VehCost"

Worksheets(3).Cells(MyRow - 1, 3) = "VehLoad"

Worksheets(3).Cells(MyRow, 1) = i

Worksheets(3).Cells(MyRow, 2) = vv2.cost

Worksheets(3).Cells(MyRow, 3) = vv2.load

```

```
MyRow = MyRow + 1
```

```
Worksheets(3).Cells(MyRow, 1) = "Client_x"
```

```
Worksheets(3).Cells(MyRow, 2) = "Client_y"
```

```
Worksheets(3).Cells(MyRow, 2) = "Name"
```

```
MyRow = MyRow + 1
```

' Εμφανίζει τις συντεταγμένες και τα ονόματα των πελατών του φορτηγού i, με τη σειρά που θα τους επισκευτεί

```
For Each cl In vv2.clients
```

```
    Worksheets(3).Cells(MyRow, 1) = cl.x
```

```
    Worksheets(3).Cells(MyRow, 2) = cl.y
```

```
    Worksheets(3).Cells(MyRow, 3) = cl.name
```

```
    MyRow = MyRow + 1
```

```
Next cl
```

```
MyRow = MyRow + 1
```

```
Next i
```

```
End Sub
```