



## *Διπλωματική Εργασία*

Μεθευρετικός αλγόριθμος για το πρόβλημα δρομολόγησης  
οχημάτων με παράδοση και παραλαβή προϊόντων και  
χρονικούς περιορισμούς

Επιβλέπων Καθηγητής : Ιωάννης Μαρινάκης

Κοκκινάκη Μαρία

**ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ**  
**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ**

*Χανιά, Ιούλιος 2020*



### Ευχαριστίες

Με την περάτωση της παρούσας διπλωματικής εργασίας, έχω τη χαρά να ολοκληρώνω τις σπουδές μου στη σχολή “Μηχανικών Παραγωγής και Διοίκησης” του Πολυτεχνείου Κρήτης. Ως ελάχιστη δυνατή μνεία, σκοπός αυτής της παραγράφου είναι να ευχαριστήσω όλους όσους συνέβαλαν στην εκπόνησή της. Πιο συγκεκριμένα, θα ήθελα να ευχαριστήσω με ιδιαίτερη τιμή τον καθηγητή μου, κ. Μαρινάκη Ιωάννη για την πολύτιμη καθοδήγηση και εμπιστοσύνη που μου έδειξε. Στη συνέχεια, με μια ιδιαίτερα σημαντική αναφορά θα ήθελα να ευχαριστήσω τον διδάκτορα κ. Τσακιράκη Ελευθέριο, για την άψογη συνεργασία, τις κατάλληλες υποδείξεις στον κατάλληλο χρόνο, αλλά και στην αμέριστη υποστήριξη και υπομονή που έδειξε, συμβάλλοντας σημαντικά στην κατάρτιση της διπλωματικής μου εργασίας.

Τέλος θα ήθελα να ευχαριστήσω τους δικούς μου ανθρώπους , τους γονείς μου, τους συμφοιτητές και φίλους μου που με υπομονή και κουράγιο μου προσέφεραν την κατάλληλη ηθική και ψυχολογική συμπαράσταση για την ολοκλήρωση της παρούσας διπλωματικής εργασίας.



## Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1 <sup>ο</sup> .....	6
1.1.ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ .....	6
1.2.ΟΡΙΣΜΟΣ-ΕΝΝΟΙΑ LOGISTICS .....	7
1.3.ΑΝΤΙΚΕΙΜΕΝΙΚΟΣ ΣΤΟΧΟΣ ΜΙΑΣ ΕΦΟΔΙΑΣΤΙΚΗΣ ΑΛΥΣΙΔΑΣ .....	8
1.4.ΟΡΙΣΜΟΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΦΟΔΙΑΣΤΙΚΗΣ ΑΛΥΣΙΔΑΣ .....	8
1.5.ΛΕΙΤΟΥΡΓΙΕΣ ΤΩΝ LOGISTICS ΚΑΙ ΤΗΣ ΕΦΟΔΙΑΣΤΙΚΗΣ ΑΛΥΣΙΔΑΣ .....	9
ΚΕΦΑΛΑΙΟ 2 <sup>ο</sup> .....	11
2.1 ΤΟ ΠΡΟΒΛΗΜΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ (VEHICLE ROUTING PROBLEM-VRP) .....	11
2.2 ΤΟ ΠΡΟΒΛΗΜΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΜΕ ΤΑΥΤΟΧΡΟΝΗ ΠΑΡΑΔΟΣΗ ΚΑΙ ΠΑΡΑΛΑΒΗ ΚΑΙ ΧΡΟΝΙΚΟΥΣ ΠΕΡΙΟΡΙΣΜΟΥΣ (VRPPDTW) .....	13
2.2.1 ΤΟ ΠΡΟΒΛΗΜΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΜΕ ΠΑΡΑΛΑΒΗ ΚΑΙ ΠΑΡΑΔΟΣΗ ΚΑΙ ΧΡΟΝΙΚΟΥΣ ΠΕΡΙΟΡΙΣΜΟΥΣ (VRPPDTW) .....	13
ΚΕΦΑΛΑΙΟ 3 <sup>ο</sup> .....	18
3.1 ΑΠΛΟΙ ΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ.....	18
3.1.1 ΑΛΓΟΡΙΘΜΟΙ ΤΟΠΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ(LOCAL SEARCH ALGORITHMS).....	18
3.2 ΜΕΘΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ.....	21
3.2.1 ΔΙΑΔΙΚΑΣΙΑ ΑΠΛΗΣΤΗΣ ΤΥΧΟΠΟΙΗΜΕΝΗΣ ΠΡΟΣΑΡΜΟΣΤΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ(GRASP) .....	21
ΚΕΦΑΛΑΙΟ 4 <sup>ο</sup> .....	23
4.1 ΠΕΡΙΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ ΚΑΙ ΕΠΙΛΥΣΗΣ ΤΟΥ .....	23
ΚΕΦΑΛΑΙΟ 5 <sup>ο</sup> .....	29
5.1 ΑΠΟΤΕΛΕΣΜΑΤΑ.....	29
ΚΕΦΑΛΑΙΟ 6 <sup>ο</sup> .....	34
6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ .....	34
6.2 ΓΡΑΦΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΤΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ.....	38
Παράδειγμα lc106.....	39
Παράδειγμα lc107.....	40
Παράδειγμα lc109.....	41
Παράδειγμα lc201 .....	42
Παράδειγμα lc205 .....	43
Παράδειγμα lc206.....	44
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	45

## ΠΕΡΙΛΗΨΗ

Στη παρούσα διπλωματική εργασία επιλύεται το πρόβλημα δρομολόγησης οχημάτων με παράδοση και παραλαβή και χρονικούς περιορισμούς (Vehicle Routing Problem with Pickup, Delivery and Time Windows). Δεδομένου ενός αριθμού πελατών που πρέπει να εξυπηρετηθούν και ενός συγκεκριμένου πλήθους οχημάτων τα οποία είναι διαθέσιμα ζητείται η σχεδίαση διαδρομών καθώς και η ελαχιστοποίηση του κόστους των διαδρομών υπό συγκεκριμένους περιορισμούς. Τα οχήματα θα ξεκινάνε από την αποθήκη και μετά το πέρας της κάθε διαδρομής θα επιστρέφουν σε αυτή. Κύρια μέθοδος επίλυσης του προβλήματος αποτελεί η διαδικασία τυχαιοποιημένης άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης (Greedy Randomized Adaptive Search Procedure (GRASP)), η οποία χρησιμοποιείται για την εύρεση ενός συνόλου αρχικών λύσεων. Μια δεύτερη μέθοδος δημιουργίας αρχικών λύσεων, η οποία θα χρησιμοποιηθεί για τη σύγκριση των αποτελεσμάτων της με τη μέθοδο GRASP, είναι η μέθοδος του πλησιέστερου γείτονα. Για την βελτίωση των αρχικών λύσεων χρησιμοποιήθηκαν οι αλγόριθμοι τοπικής αναζήτησης (1-0 relocate, 1-1 exchange, 2-0 opt). Το πρόβλημα επιλύθηκε σε περιβάλλον MATLAB και εφαρμόστηκε σε παραδείγματα με γνωστό αριθμό διαδρομών και βέλτιστο κόστος διαδρομών. Τέλος παρουσιάζονται συγκριτικοί πίνακες των δύο μεθόδων με τα βέλτιστα αποτελέσματα καθώς και γραφικές αναπαραστάσεις των καλύτερων λύσεων.

## ΚΕΦΑΛΑΙΟ 1<sup>ο</sup>

### 1.1.ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

Ετυμολογικά ο όρος Εφοδιαστική/Logistics προέρχεται από τον ελληνικό όρο <<λόγος>> που σημαίνει λογική με την έννοια της εκλογίκευσης και έχει ως σκοπό την επίτευξη συγκεκριμένων στόχων. Μελετώντας την διεθνή βιβλιογραφία ο όρος εντοπίζεται από την αρχαιότητα έως και τις μέρες μας.

Σύμφωνα με κάποιους ιστορικούς ο όρος χρησιμοποιήθηκε αρχικά από τον Μέγα Αλέξανδρο ο οποίος επιδίωκε τη χρήση στρατηγικών και τεχνικών για τον εφοδιασμό και την τροφοδότηση της αυτοκρατορίας και των στρατευμάτων του. Αξιοσημείωτο ακόμα να αναφερθεί είναι ότι γενικότερα, η ανάπτυξη των πολιτισμών των αρχαίων Ελλήνων, Αιγυπτίων αλλά και της Ρωμαϊκής αυτοκρατορίας είχαν βασιστεί σε καινοτόμα για την εποχή τους συστήματα μεταφοράς (δίκτυα) που συνιστούν σημαντική συνιστώσα της Εφοδιαστικής/Logistics.

Επίσης, κατά τη διάρκεια του Β' Παγκοσμίου Πολέμου από τις ΗΠΑ έγινε χρήση <<Εφοδιαστικής>> με στόχο τη μαζική μεταφορά στρατιωτικών μονάδων αλλά και την κάλυψη βασικών αναγκών συνεχούς ανεφοδιασμού τροφίμων, πυρομαχικών και δευτερευόντων υλικών διαχείρισης της εφοδιαστικής αλυσίδας. Αυτό είχε ως αποτέλεσμα την ενασχόληση διάφορων μελετητών του αμερικανικού στρατού με τη σύσταση μοντέλων για τη βελτιστοποίηση της διαχείρισης της εφοδιαστικής αλυσίδας με στόχο τόσο τη μείωση του κόστους της όσο και την αποτελεσματικότητά της.

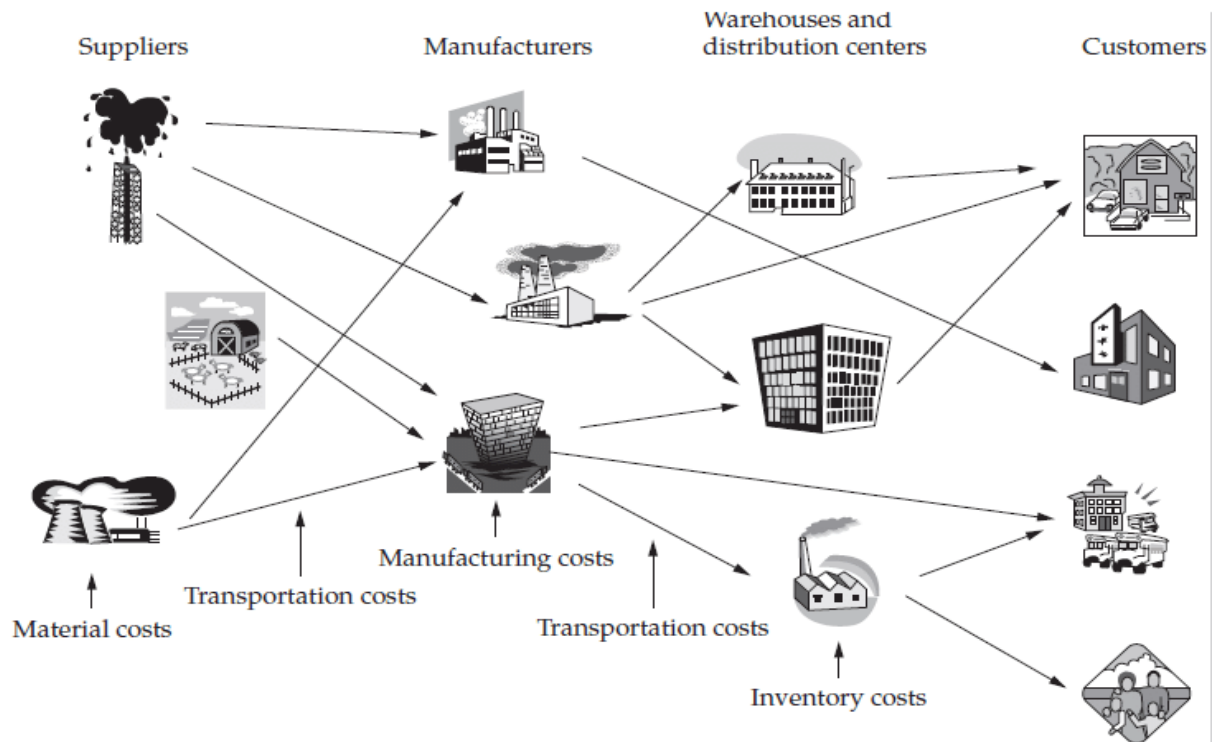
Αργότερα, τα μοντέλα Διαχείρισης Εφοδιαστικής Αλυσίδας – Logistics συνδέθηκαν άρρητα με το marketing αλλά και με τις άλλες επιχειρησιακές δραστηριότητες καθώς ήταν φανερό ότι θα συντελούσαν καθοριστικά στη βελτιστοποίηση της εξυπηρέτησης πελατών και στη μείωση του κόστους.

Ολοκληρώνοντας, στην εξέλιξη και τελική αναδιαμόρφωση των εν λόγω μοντέλων συνέβαλε και η ανάπτυξη της τεχνολογίας η οποία παρείχε πληθώρα δυνατοτήτων και προοπτικών.

## 1.2.ΟΡΙΣΜΟΣ-ΕΝΝΟΙΑ LOGISTICS

<<Μια εφοδιαστική αλυσίδα αποτελείται από όλα τα στάδια (μέλη) που σχετίζονται άμεσα ή έμμεσα με την ικανοποίηση της απαίτησης του πελάτη. Στην εφοδιαστική αλυσίδα περιλαμβάνονται εκτός από τον παραγωγό και τον προμηθευτή, οι μεταφορείς, οι αποθήκες, τα σημεία λιανικής πώλησης και οι ίδιοι οι πελάτες. >> (Βιδάλης,2017)

Η εφοδιαστική αλυσίδα ονομάζεται και δίκτυο εφοδιασμού ( logisticnetwork) ενώ συχνά αναπαρίσταται ως δίκτυο όπως παρακάτω :



Εικόνα 1 : logistic network(πηγή : [https://www.researchgate.net/figure/The-Logistics-Network-60\\_fig2\\_323626639](https://www.researchgate.net/figure/The-Logistics-Network-60_fig2_323626639))

Οι κόμβοι του δικτύου αναπαριστούν εγκαταστάσεις οι οποίες συνδέονται με τόξα που δηλώνουν μεταφορά φυσικών προϊόντων. Μεταξύ των εγκαταστάσεων του δικτύου ανταλλάσσονται προϊόντα (ακατέργαστα υλικά, προϊόντα υπό εξέλιξη, τελικά προϊόντα) , χρηματικές αξίες και πληροφορίες. Υπάρχουν δυο κατευθύνσεις ροών, η κατωφερής κατεύθυνση δηλαδή η κατεύθυνση που ξεκινάει από τους προμηθευτές με προορισμό την τελική κατανάλωση και η ανωφερής δηλαδή η κατεύθυνση από την κατανάλωση προς τους προμηθευτές. (Βιδάλης, 2017)



### 1.3.ΑΝΤΙΚΕΙΜΕΝΙΚΟΣ ΣΤΟΧΟΣ ΜΙΑΣ ΕΦΟΔΙΑΣΤΙΚΗΣ ΑΛΥΣΙΔΑΣ

Αντικειμενικός στόχος κάθε εφοδιαστικής αλυσίδας είναι η μεγιστοποίηση της καθαρής αξίας που δημιουργείται από ολόκληρο σύστημα. Η καθαρή αξία που δημιουργεί η εφοδιαστική αλυσίδα είναι η διαφορά της χρηματικής ροής από τον πελάτη και του συνολικού κόστους που δαπάνησε η εφοδιαστική αλυσίδα για την παραγωγή και διάθεσή του. Όλες οι ροές κοστίζουν. Επομένως, η κατάλληλη διαχείριση όλων αυτών των ροών αποτελεί βασικό παράγοντα για την επιτυχία της εφοδιαστικής αλυσίδας. (Βιδάλης, 2017)

### 1.4.ΟΡΙΣΜΟΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΦΟΔΙΑΣΤΙΚΗΣ ΑΛΥΣΙΔΑΣ

Η διαχείριση της εφοδιαστικής αλυσίδας είναι το σύνολο των μεθόδων που χρησιμοποιούνται με στόχο την αποδοτική ενοποίηση - σύνδεση μεταξύ των προμηθευτών, παραγωγών, αποθηκευτών και πωλητών έτσι ώστε τα παραγόμενα εμπορεύματα να διανέμονται στις σωστές ποσότητες, στα κατάλληλα σημεία αλλά και στον κατάλληλο χρόνο, με αποτέλεσμα να ελαχιστοποιούνται τα ολικά κόστη εντός του συστήματος, ενώ παράλληλα να επιτυγχάνονται τα επίπεδα εξυπηρέτησης. (Βιδάλης, 2017)

Σύμφωνα με τα παραπάνω, παρατηρείται η ιδιαίτερη σημασία που αποκτά η ροή των προϊόντων και των υπηρεσιών που προσφέρονται στην αγορά από το αρχικό σημείο προέλευσης πρώτων υλών και εργατικού δυναμικού μέχρι το σημείο προορισμού. Με σκοπό τη βελτιστοποίηση της συνολικής διαδρομής αρκεί να χωριστεί σε νοηματικά κοινά στάδια η ροή των διάφορων προϊόντων και οι υπηρεσίες πριν φτάσουν στα χέρια των τελικών καταναλωτών. Τα κοινά αυτά στάδια περικλείονται σε τρεις μεγαλύτερες κατηγορίες και περιγράφονται ως στάδια αποθήκευσης, μεταποίησης ή μεταφοράς. Έτσι, σε ένα οποιοδήποτε σύστημα Διαχείρισης Εφοδιαστικής Αλυσίδας – Logistics προκειμένου να αναλυθεί με στόχο τη βελτιστοποίηση του θα πρέπει πρώτα να αναγνωριστούν οι λειτουργίες που ανήκουν στα στάδια αυτά. (Βιδάλης, 2017)

## 1.5.ΛΕΙΤΟΥΡΓΙΕΣ ΤΩΝ LOGISTICS ΚΑΙ ΤΗΣ ΕΦΟΔΙΑΣΤΙΚΗΣ ΑΛΥΣΙΔΑΣ

### Οι βασικές λειτουργίες είναι:

- Αγορές – προμήθειες
- Διαχείριση αποθεμάτων
- Μεταφορά
- Αποθήκευση
- Διανομή
- Πληροφόρηση

Πριν την ανάλυση των παραπάνω λειτουργιών θα πρέπει η επιχείρηση να προσδιορίσει τη στρατηγική που θα ακολουθήσει. Αυτή θα διαμορφώσει την εικόνα της, την επιτυχία της ή την αποτυχία της.

Ως στρατηγική ορίζεται το σύνολο των αποφάσεων που θα σχεδιάσουν τις μακροχρόνιες ικανότητες, δεξιότητες και διαδικασίες, τεχνολογίες, πηγές και τις δραστηριότητες τακτικής με σκοπό τη δημιουργία και παράδοση αγαθών ή υπηρεσιών, όπως και τη δημιουργία αξίας για τον πελάτη. Πρόκειται για ένα χάρτη πορείας της επιχείρησης, ο οποίος δείχνει που θέλει να πάει και πώς σκοπεύει να φτάσει μέχρι εκεί, αλλά δε λέει πώς να οδηγήσει ή πώς να αντιμετωπίσει όλα τα μικροπροβλήματα που μπορεί να συναντήσει στο δρόμο. Αυτά καλύπτονται από το εκτελεστικό πρόγραμμα του σχεδίου της επιχείρησης. Η στρατηγική είναι εξίσου σημαντική τόσο βραχυπρόθεσμα όσο και μακροπρόθεσμα. Πρέπει να καθοδηγεί μέσα από τα βραχυπρόθεσμα μικροπροβλήματα προς τους στόχους της επιχείρησης με όσο το δυνατό μεγαλύτερη ακρίβεια. Ταυτόχρονα πρέπει να είναι τεκμηριωμένη με τέτοιο τρόπο ώστε να καθοδηγεί τα στελέχη πως να αντιδράσουν σε εναλλακτικά σενάρια.

Ένα σύστημα Logistics μπορεί να σχεδιαστεί από κάθε επιχείρηση με διαφορετικό τρόπο ανάλογα με τις ανάγκες και τη μορφή της. Παρόλα αυτά, οι λειτουργίες που πρέπει να διεκπεραιώνει το τμήμα Logistics είναι συγκεκριμένες και αναλύονται παρακάτω:

#### ➤ Αγορές – προμήθειες

Ο όρος αυτός αναφέρεται στην απόκτηση προϊόντων ή υπηρεσιών από τρίτους, από προμηθευτές, από πηγές εκτός της επιχείρησης τα οποία μπορεί να ενσωματωθούν στο τελικό προϊόν, πχ πρώτες ύλες ή υλικά συσκευασίας. Οι ποσότητες αγοράς επηρεάζονται σε μεγάλο βαθμό από το επίπεδο εξυπηρέτησης πελατών που ορίζει η επιχείρηση.

#### ➤ Διαχείριση αποθεμάτων

Η σημασία των αποθεμάτων για μια επιχείρηση είναι σημαντική γιατί συμβάλλει στην ομαλή και οικονομική λειτουργία της παραγωγικής διαδικασίας. Οι επιχειρήσεις καλούνται να διαθέσουν αποθέματα για να μπορούν να ανταποκρίνονται άμεσα εξυπηρετώντας τη ζήτηση. Η επένδυση που απαιτείται για τα αποθέματα επιβάλει προσοχή στον προγραμματισμό των αναγκών και στον έλεγχο αποθεμάτων. Ο στόχος της διαχείρισης αποθεμάτων είναι διττός, αφενός θα πρέπει να εξασφαλίζεται το βέλτιστο επίπεδο αποθεμάτων με βάση την δεδομένη εξυπηρέτηση της ζήτησης και αφετέρου δε θα πρέπει να

εξασφαλίζεται μέσω μια αποθεματικής πολιτικής η συνολική ελαχιστοποίηση του κόστους του συστήματος Logistics.

➤ Μεταφορά

Η εύρεση του βέλτιστου τρόπου μεταφοράς περιλαμβάνει την επιλογή των κατάλληλων μέσων μεταφοράς, αν τα μέσα αυτά θα είναι ιδιόκτητα ή όχι και στην περίπτωση που δεν είναι, ποιο είναι το καταλληλότερο μέσο μεταφοράς. Σημαντική είναι η χρονική διάρκεια ώστε τα προϊόντα να παραδοθούν ακριβώς στην ώρα που επιθυμεί ο πελάτης – επιχείρηση χωρίς καθυστέρηση.

➤ Αποθήκευση

Η ανάγκη της αποθήκευσης και της δημιουργίας των αποθηκευτικών χώρων έγκειται στην εξασφάλιση της συνεχούς ροής των προϊόντων από την παραγωγή ως την κατανάλωση. Ως αποθήκες ορίζονται χώροι μικροί ή μεγάλοι στους οποίους τοποθετούνται προϊόντα προς φύλαξη. Είναι μέρη ασφαλή τόσο από καιρικές συνθήκες όσο και από κλοπές αλλά και άλλες απώλειες. Τα προϊόντα φυλάγονται στις αποθήκες μέχρι να ζητηθούν για να χρησιμοποιηθούν στην παραγωγή στην κατανάλωση και στη μεταπώληση.

➤ Διανομή

Όπως η μεταφορά, έτσι και η διανομή αναφέρονται στη μεταφορά από τις αποθήκες ή τις εγκαταστάσεις μια επιχείρησης στους πελάτες. Η διαφορά τους έγκειται στο γεγονός ότι οι μεταφορές αναφέρονται στη διακίνηση λίγων αριθμών σε μεγάλες ποσότητες και σε μικρό αριθμό πελατών, ενώ οι διανομές αναφέρονται σε διακίνηση πολλών προϊόντων σε μικρές ποσότητες και σε μεγάλο αριθμό πελατών.

➤ Πληροφόρηση

Οι τεχνολογικές εξελίξεις επιτρέπουν τόσο την εύκολη επικοινωνία των απομακρυσμένων εγκαταστάσεων, όσο και τη δημιουργία βάσεων δεδομένων που επιτρέπουν την καταγραφή και τον έλεγχο των αποθεμάτων και των διανομών.

(Μπινιώρης,2004)

## ΚΕΦΑΛΑΙΟ 2<sup>ο</sup>

### 2.1 ΤΟ ΠΡΟΒΛΗΜΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ (VEHICLE ROUTING PROBLEM-VRP)

Το πρόβλημα δρομολόγησης οχημάτων (VRP) είναι ένα πρόβλημα συνδυαστικής βελτιστοποίησης και ακέραιου προγραμματισμού το οποίο ορίζει ένα βέλτιστο σύνολο διαδρομών για ένα στόλο οχημάτων για να παραδώσουν τα προϊόντα σε ένα δεδομένο σύνολο πελατών. Δεν υπάρχουν περιορισμοί στο πότε ή στο με ποια σειρά θα εξυπηρετηθούν αυτές οι οντότητες.

Επομένως, κύριος στόχος του προβλήματος αυτού είναι να κατασκευαστεί ένα χαμηλού κόστους, εφικτό σύνολο από διαδρομές, μία για κάθε όχημα. Μια διαδρομή είναι μια ακολουθία από τοποθεσίες τις οποίες ένα όχημα πρέπει να επισκεφτεί μαζί με μια ένδειξη της εξυπηρέτησης που παρέχει.

Η μοντελοποίηση του προβλήματος φαίνεται παρακάτω:

$$x_{ijk} = \begin{cases} 1, & \text{εάν το όχημα } k \text{ επισκέπτεται τον πελάτη } j \text{ αμέσως μετά τον πελάτη } i \\ 0, & \text{αλλιώς} \end{cases}$$

$$y_{jk} = \begin{cases} 1, & \text{εάν ο πελάτης } i \text{ επισκέπτεται από το όχημα } k \\ 0, & \text{αλλιώς} \end{cases}$$

$$\min \sum_{i,j} c_{ij} \sum_k x_{ijk} \quad (1)$$

υπό

$$\sum_k y_{ik} = \begin{cases} 1, & i = 2, \dots, n \\ m, & i = 1 \end{cases} \quad (2)$$

$$\sum_i q_i y_{ij} \leq Q_k \quad k = 1, \dots, m \quad (3)$$

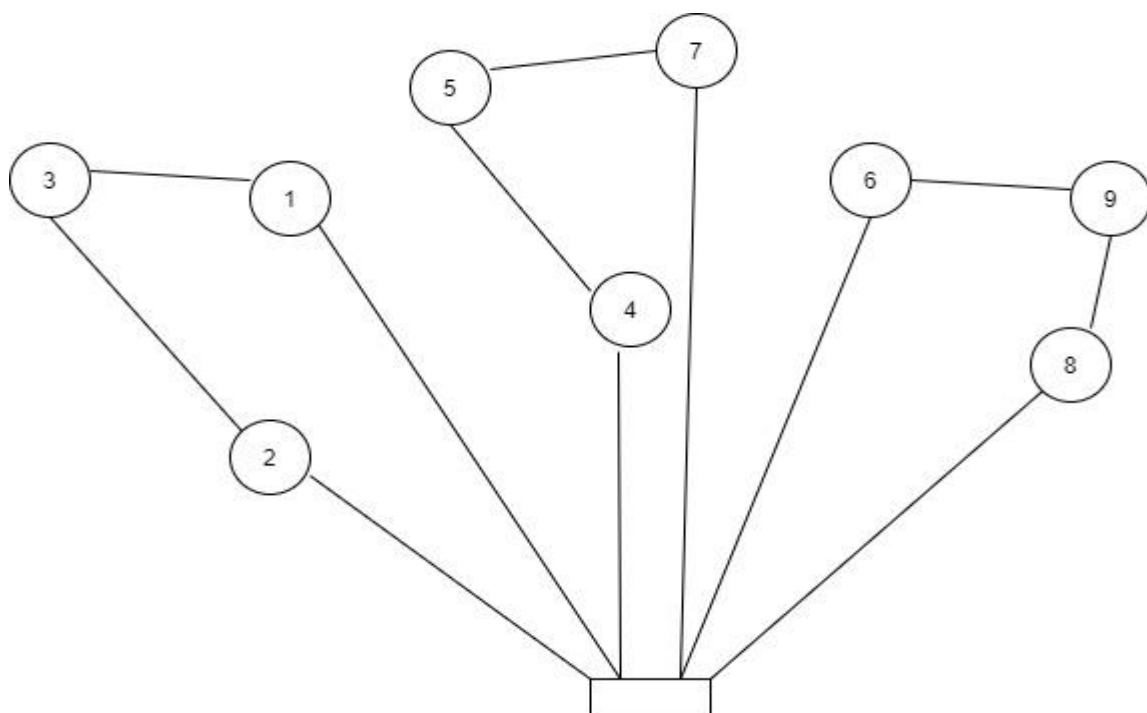
$$\sum_j x_{ijk} = \sum_j x_{jik} = y_{ik} \quad i = 1, \dots, n, \quad k = 1, \dots, m \quad (4)$$

$$\sum_{i,j \in S} x_{ijk} \leq |S| - 1 \quad \forall S \subseteq \{2, \dots, n\}, k = 1, \dots, m \quad (5)$$

$$y_{ik} \in \{0,1\}, i = 1, \dots, n, \quad k = 1, \dots, m \quad (6)$$

$$x_{ijk} \in \{0,1\}, i, j = 1, \dots, n, k = 1, \dots, m \quad (7)$$

Ο περιορισμός (2) δείχνει ότι κάθε πελάτης εκχωρείται σε ένα μόνο όχημα, εκτός από την αποθήκη που την επισκέπτονται όλα τα οχήματα. Ο περιορισμός (3) είναι ο περιορισμός της χωρητικότητας των οχημάτων. Ο περιορισμός (4) αφορά την αρχή διατήρησης ροής και διασφαλίζει την συνέχεια της διαδρομής του κάθε οχήματος. Ο περιορισμός (5) δείχνει ότι ένα όχημα που επισκέπτεται ένα πελάτη φεύγει από τον πελάτη. Οι περιορισμοί (6) και (7) είναι αυτοί που επιβάλλουν τη μη-αρνητικότητα και την δυαδικότητα των αντίστοιχων μεταβλητών. (Μαρινάκης, Μυγδαλάς, 2016)



**Σχήμα 1 : Παράδειγμα δρομολόγησης οχημάτων (VRP)**

Παρακάτω φαίνονται κάποιες από τις παραλλαγές του προβλήματος δρομολόγησης οχημάτων και είναι οι εξής:

- Το πρόβλημα του πλανόδιου πωλητή
- Το πρόβλημα δρομολόγησης οχημάτων με περιορισμό χωρητικότητας
- Το πρόβλημα δρομολόγησης οχημάτων με περιορισμό χωρητικότητας και χρόνους εξυπηρέτησης
- Το πρόβλημα δρομολόγησης οχημάτων με πολλαπλές επιστροφές στην αποθήκη
- Το ανοιχτό πρόβλημα δρομολόγησης οχημάτων
- Το ανοιχτό-κλειστό πρόβλημα δρομολόγησης οχημάτων
- Το πρόβλημα δρομολόγησης οχημάτων με χρονικούς περιορισμούς

- Το πρόβλημα δρομολόγησης οχημάτων με παραλαβή και παράδοση προϊόντων
- Το πρόβλημα δρομολόγησης οχημάτων με παραλαβή και παράδοση προϊόντων και χρονικούς περιορισμούς

(Μαρινάκης,Μυγδαλάς,2008)

## **2.2 ΤΟ ΠΡΟΒΛΗΜΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΜΕ ΤΑΥΤΟΧΡΟΝΗ ΠΑΡΑΔΟΣΗ ΚΑΙ ΠΑΡΑΛΑΒΗ ΚΑΙ ΧΡΟΝΙΚΟΥΣ ΠΕΡΙΟΡΙΣΜΟΥΣ (VRPPDTW)**

### **2.2.1 ΤΟ ΠΡΟΒΛΗΜΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΜΕ ΠΑΡΑΛΑΒΗ ΚΑΙ ΠΑΡΑΔΟΣΗ ΚΑΙ ΧΡΟΝΙΚΟΥΣ ΠΕΡΙΟΡΙΣΜΟΥΣ (VRPPDTW)**

Η παρούσα διπλωματική ασχολείται με την επίλυση του προβλήματος δρομολόγησης οχημάτων με παράδοση και παραλαβή προϊόντων και χρονικούς περιορισμούς.

Το πρόβλημα δρομολόγησης οχημάτων με παράδοση και παραλαβή προϊόντων και χρονικούς περιορισμούς (VRPPDTW) είναι η σύνθεση των παρακάτω δυο προβλημάτων :

- Του προβλήματος δρομολόγησης οχημάτων με διανομή και παραλαβή προϊόντων κατά τη διάρκεια της διαδρομής ( Vehicle routing problem with pickup and delivery)

Κάθε πελάτης σχετίζεται με δύο ποσότητες, τη ζήτηση  $d_i$  και τη παραλαβή  $p_i$  των προϊόντων που πρέπει να διανεμηθούν και να παραληφθούν αντίστοιχα. Για κάθε πελάτη καθορίζονται και δύο κόμβοι ο  $O_i$  και ο  $D_i$  που είναι οι κόμβοι από τους οποίους ξεκινάνε τα προϊόντα που πρέπει να διανεμηθούν στον πελάτη και καταλήγουν τα προϊόντα που συλλέγονται από τον πελάτη. Ένας πολύ βασικός περιορισμός είναι ότι σε κάθε πελάτη η διανομή των προϊόντων γίνεται πριν από τη παραλαβή, με σκοπό η συνολική φόρτωση ενός οχήματος όταν φτάσει σε ένα πελάτη  $i$  να είναι ίση με την αρχική φόρτωση του οχήματος, μείον το άθροισμα των προϊόντων που παρέδωσε στους πελάτες πριν τον  $i$ , συν το άθροισμα των προϊόντων που παρέλαβε από τους πελάτες πριν τον  $i$ .

Κύρια χαρακτηριστικά :

- ❖ Κάθε κύκλος περνάει από την αποθήκη.
  - ❖ Κάθε πελάτης επισκέπτεται από ένα μόνο κύκλο.
  - ❖ Κάθε όχημα αντιστοιχεί σε μια μόνο διαδρομή.
  - ❖ Η χωρητικότητα του οχήματος πρέπει να είναι μη αρνητική και να μην ξεπερνάει την χωρητικότητα του οχήματος.
  - ❖ Για κάθε πελάτη  $i$  ο κόμβος  $O_i$  αν είναι διαφορετικός από την αποθήκη πρέπει να εξυπηρετηθεί στην ίδια διαδρομή και πριν από τον πελάτη  $i$ .
  - ❖ Για κάθε πελάτη  $i$  ο κόμβος  $D_i$  αν είναι διαφορετικός από την αποθήκη πρέπει να εξυπηρετηθεί στην ίδια διαδρομή και μετά από τον πελάτη  $i$ .
- Του προβλήματος δρομολόγησης οχημάτων για την εξυπηρέτηση πελατών μέσα σε δεδομένα χρονικά περιθώρια ( Vehicle routing problem with time windows).

Κάθε πελάτης έχει ένα φορτίο που πρέπει να παραληφθεί και ο ίδιος ο πελάτης καθορίζει μια χρονική περίοδο που ονομάζεται χρονικό παράθυρο στην οποία η φόρτωση πρέπει να

πραγματοποιηθεί. Οι πελάτες εξυπηρετούνται από οχήματα με περιορισμένη χωρητικότητα έτσι ώστε η συνολική φόρτωση κάθε οχήματος δεν μπορεί να ξεπεράσει την χωρητικότητα των οχημάτων.

Για κάθε πελάτη υπάρχει ένας χρόνος εξυπηρέτησης και ένα χρονικό παράθυρο  $(\alpha, \beta)$  όπου ο αριθμός  $\alpha$  αναφέρεται στον νωρίτερο χρόνο άφιξης του φορτηγού στο πελάτη και ο αριθμός  $\beta$  στον αργότερο χρόνο άφιξης. Έτσι, με βάση το χρονικό παράθυρο διακρίνονται τρεις περιπτώσεις :

- (1) Το όχημα να φτάσει μέσα στο χρονικό παράθυρο το οποίο έχει ορίσει ο κάθε πελάτης και με το όχημα να παραμένει στη τοποθεσία έως ότου εξυπηρετηθεί ο πελάτης, μια περίπτωση η οποία προτιμάται των υπολοίπων .
- (2) Το όχημα να φτάσει νωρίτερα από τον νωρίτερο χρόνο άφιξης. Στην περίπτωση αυτή, το όχημα θα περιμένει στην τοποθεσία μέχρι να ανοίξει το χρονικό παράθυρο και να εξυπηρετηθεί ο πελάτης.
- (3) Το όχημα να φτάσει μετά τον αργότερο χρόνο άφιξης τον οποίο έχει ορίσει ο πελάτης, μια περίπτωση η οποία δεν προτιμάται καθόλου, εφόσον δε μπορεί να εξυπηρετηθεί ο πελάτης.

Κύρια χαρακτηριστικά :

- ❖ Κάθε κύκλος περνάει από την αποθήκη.
- ❖ Κάθε πελάτης επισκέπτεται μόνο ένα κύκλο.
- ❖ Το άθροισμα της ζήτησης των κόμβων που επισκέπτονται από ένα κύκλο δεν ξεπερνάει τη χωρητικότητα του οχήματος.
- ❖ Για κάθε πελάτη η εξυπηρέτηση πρέπει να ξεκινήσει και να ολοκληρωθεί μέσα στο χρονικό παράθυρο  $[\alpha_i, \beta_i]$ , ενώ το όχημα θα παραμείνει στο χώρο του πελάτη για χρόνο  $s_i$  μέχρι να ξεφορτώσει.

### **Μοντελοποίηση του προβλήματος :**

Υπάρχει ένας στόλος  $K$  οχημάτων όλα με την ίδια δυναμικότητα  $u$  και ένα σύνολο  $n$  πελάτες που πρέπει να εξυπηρετηθούν από ένα κέντρο το οποίο θα διαιρείται σε δύο κόμβους, 0 και  $n+1$ .

Γίνεται υπόθεση ότι για κάθε πελάτη  $i$  υπάρχει αίτημα για παράδοση  $d_i$  μονάδων και παραλαβή  $p_i$  μονάδων από το όχημα που τον επισκέπτεται και ότι η επίσκεψη του οχήματος θα πρέπει να πραγματοποιηθεί στο χρονικό διάστημα  $[a_i, b_i]$ . Επειδή κάθε πελάτης εξυπηρετείται από ένα και μόνο όχημα που τον επισκέπτεται μία και μόνη φορά, θα υποθέσουμε αναγκαστικά ότι  $0 \leq d_i, p_i \leq u$ ,  $\forall i$ . Επιπλέον, υποθέτοντας ότι η αναμονή δεν κοστίζει θεωρείται φυσιολογικό ένα όχημα να φθάνει νωρίτερα από τη χρονική στιγμή  $a_i$  και

να περιμένει έως την έναρξη του χρονικού διαστήματος εξυπηρέτησής του. Αναμενόμενο είναι η άφιξη του οχήματος μετά το χρόνο  $b_i$  να μη θεωρείται αποδεκτή.

$n$  : αριθμός πελατών

Για τη μοντελοποίηση του προβλήματος ορίζονται οι παρακάτω μεταβλητές :

$$x_{ij}^k = \begin{cases} 1, & \text{εάν το όχημα } k \text{ ταξιδεύει απευθείας από το } i \text{ στο } j \\ 0, & \text{αλλιώς } \forall i, \forall j, i \neq j, k = 1, \dots, K \end{cases}$$

$y_i^k$  = η ποσότητα από τις απομένουσες παραδόσεις που διεκπεραιώνονται από το όχημα  $k$  όταν αυτό εκκινεί από τον πελάτη  $i, \forall k, \forall i$

$z_i^k$  = η ποσότητα από τις απομένουσες παραλαβές που διεκπεραιώνονται από το όχημα  $k$  όταν αυτό εκκινεί από τον πελάτη  $i, \forall k, \forall i$

$t_i^k$  = η χρονική αρχή για την εξυπηρέτηση του πελάτη  $i$  από το όχημα  $k$ , όταν αυτό εκκινεί από τον πελάτη  $k, \forall i, \forall k$

Με αντικειμενική συνάρτηση

min

$$\sum_{k=1}^K \sum_{i=0}^{n+1} \sum_{j=0}^{n+1} c_{ij} x_{ij}^k, i \neq j \quad (1)$$

και περιορισμούς

υπό

$$\sum_{k=1}^K \sum_{j=0}^{n+1} x_{ij}^k = 1 \quad i = 1, \dots, n \quad (2)$$

$$\sum_{i=0}^{n+1} x_{il}^k - \sum_{j=0}^{n+1} x_{lj}^k = 0, l = 1, \dots, n \quad \forall k \quad (3)$$

$$\sum_{j=1}^n x_{0j}^k \leq 1, \forall k \quad (4)$$

$$\sum_{i=1}^n x_{i,n+1}^k - \sum_{j=1}^n x_{0j}^k = 0, \forall k \quad (5)$$

$$y_i^k + z_i^k \leq u, i = 1, \dots, n, \forall k \quad (6)$$

$$y_{n+1}^k = 0, \forall k \quad (7)$$



$$y_0^k = \sum_{i=1}^n \sum_{j=1}^n d_i x_{ij}^k, \forall k \quad (8)$$

$$z_{n+1}^k = \sum_{i=1}^n \sum_{j=1}^n p_i x_{ij}^k, \forall k \quad (9)$$

$$z_0^k = 0, \forall k \quad (10)$$

$$x_{ij}^k (z_i^k + p_j - z_j^k) = 0, \forall i, \forall j, i \neq j, \forall k \quad (11)$$

$$x_{ij}^k (y_i^k - d_j - y_j^k) = 0, \forall i, \forall j, i \neq j, \forall k \quad (12)$$

$$x_{ij}^k (t_i^k + t_{ij} - t_j^k) \leq 0, \forall i, \forall j, i \neq j, \forall k \quad (13)$$

$$a_i \leq t_i^k \leq b_i, \forall i, \forall k \quad (14)$$

$$y_i^k \geq 0, \forall i, \forall k \quad (15)$$

$$z_i^k \geq 0, \forall i, \forall k \quad (16)$$

$$x_{ij}^k \in \{0,1\}, \forall i, \forall j, i \neq j, \forall k \quad (17)$$

Η αντικειμενική συνάρτηση ελαχιστοποιεί το ολικό κόστος των διαδρομών.

Οι περιορισμοί (2) εξασφαλίζει την εξυπηρέτηση κάθε πελάτη απο ακριβώς ένα όχημα, ενώ οι περιορισμοί (3) εγγυώνται ότι το όχημα που εισέρχεται και εξέρχεται από ένα κόμβο είναι το ίδιο. Οι περιορισμοί (4) και (5) επιβάλλουν τη χρήση του κάθε οχήματος το πολύ μια φορά. Οι περιορισμοί (3),(4) και (5) αντιστοιχούν στην ισορροπία ροής και απαιτούν ότι κάθε όχημα εκκινεί από τον κόμβο αφετηρίας (κόμβος 1) το πολύ μια φορά, ότι εξέρχεται από ένα κόμβο μόνο εάν έχει εισέλθει σε αυτόν και ότι στο τέλος επιστρέφει στο σημείο εκκίνησης το πολύ μια φορά. Οι περιορισμοί (6) εγγυώνται ότι το φορτίο ενός οχήματος όταν αυτό εκκινεί από ένα πελάτη δεν υπερβαίνει τη δυναμικότητα του. Οι περιορισμοί (7) και (9) εγγυώνται ότι όταν ένα όχημα εκκινεί από το κέντρο έχει όλο το αναγκαίο φορτίο παραλαβής για τις παραδόσεις στους πελάτες που εξυπηρετεί και το φορτίο παραλαβής του είναι μηδενικό. Οι περιορισμοί (8) και (10) εγγυώνται ότι όταν ένα όχημα επιστρέφει στην

αφετηρία έχει επιτελέσει όλες τις διανομές και έχει παραλάβει όλες τις ποσότητες που του αντιστοιχούν. Οι μη γραμμικοί περιορισμοί (11),(12) επιβάλλουν τη μείωση του φορτίου παράδοσης ενός οχήματος  $k$  με  $d_j$  μονάδες όταν αυτό έχει κινηθεί απευθείας από το  $i$  στο  $j$  και αντίστοιχα την αύξηση του φορτίου παραλαβής με  $p_j$  μονάδες διότι η κίνηση αυτή επιφέρει την εξυπηρέτηση του πελάτη  $j$ . Οι μη γραμμικοί περιορισμοί (13) συνεπάγονται ότι εάν το όχημα  $k$  κινηθεί απευθείας από το  $i$  στο  $j$  τότε ο πελάτης  $j$  αρχίζει να εξυπηρετείται κάποια χρονική στιγμή που είναι ίση ή μεγαλύτερη της χρονικής στιγμής που άρχισε η εξυπηρέτηση του πελάτη  $i$  συν το χρόνο που χρειάστηκε το όχημα για το απευθείας ταξίδι από το  $i$  στο  $j$ . Οι περιορισμοί αυτοί επιτρέπουν το όχημα να παραμείνει σε αναμονή σε κάθε πελάτη μέχρι να έρθει η καθορισμένη από τον πελάτη στιγμή έναρξης του χρονικού διαστήματος εξυπηρέτησης. Οι περιορισμοί (14) επιβάλλουν την έναρξη εξυπηρέτησης να ανήκει χρονικά στα πλαίσια που έχει καθορίσει ο πελάτης. Τέλος οι περιορισμοί (15), (16),(17) είναι αυτοί που επιβάλλουν τη μη-αρνητικότητα και την δυαδικότητα των αντίστοιχων μεταβλητών. (Μαρινάκης,Μυγδαλάς,2008)

## ΚΕΦΑΛΑΙΟ 3<sup>ο</sup>

### 3.1 ΑΠΛΟΙ ΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Οι ευρετικοί αλγόριθμοι χρησιμοποιούνται για την επίλυση ενός προβλήματος συνδυαστικής βελτιστοποίησης. Συχνά, για την εξαγωγή των βέλτιστων αποτελεσμάτων γίνεται χρήση διαφορετικών τεχνικών. Μια λύση ενός ευρετικού αλγόριθμου γίνεται αποδεκτή αν ικανοποιεί τα κριτήρια όπως η ποιότητα της λύσης δηλαδή η απόκλισή της από τη βέλτιστη, η ευκολία απόκτησης μιας λύσης, η λογική πάνω στην οποία στηρίζονται οι κανόνες του ευρετικού αλγόριθμου που χρησιμοποιήθηκαν για να οδηγηθεί στη λύση. Για κάποιο πρόβλημα συνδυαστικής βελτιστοποίησης δεν υπάρχει μονάχα ένας ευρετικός αλγόριθμος που να δίνει τη βέλτιστη λύση αλλά έχουν αναπτυχθεί αρκετοί αλγόριθμοι οι οποίοι συγκρινόμενοι μεταξύ τους οδηγούν ολοένα και σε καλύτερες λύσεις.

Οι κατηγορίες των ευρετικών αλγορίθμων είναι οι εξής :

- Αλγόριθμοι απληστίας (greedy algorithms)
- Προσεγγιστικοί αλγόριθμοι (approximation algorithms)
- Αλγόριθμοι τοπικής αναζήτησης (local search algorithms)

*Η παρούσα διπλωματική επιλύεται με τη χρήση αλγορίθμων απληστίας και αλγορίθμων τοπικής αναζήτησης.*

#### 3.1.1 ΑΛΓΟΡΙΘΜΟΙ ΤΟΠΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ (LOCAL SEARCH ALGORITHMS)

Οι αλγόριθμοι τοπικής αναζήτησης προσπαθούν ξεκινώντας από μια αρχική εφικτή λύση να βελτιώσουν τη λύση με κάποια μέθοδο αναζήτησης στη γειτονιά της λύσης. Χρησιμοποιούν την τεχνική της επαναληπτικής βελτίωσης και λειτουργούν ως εξής:

- Επιλέγεται μια λύση από το χώρο και αποτιμάται. Αυτή τη λύση χαρακτηρίζεται τρέχουσα.
- Ετοιμάζεται ένας μετασχηματισμός στη τρέχουσα λύση για να εξαχθεί μια νέα λύση. Αξιολογείται η νέα λύση.
- Εάν η νέα λύση είναι καλύτερη από τη τρέχουσα τότε ανταλλάσσεται με την τρέχουσα αλλιώς απορρίπτεται.
- Τα παραπάνω βήματα επαναλαμβάνονται έως ότου κανένας μετασχηματισμός να μη μπορεί να βελτιώσει τις λύσεις.

### διαδικασία *local\_search*

#### **begin**

t μια αρχική λύση του προβλήματος

**do while** βρίσκετε μια βελτιωμένη λύση (*improve(t)*)

*t=improve(t)*

**return t**

#### **end**

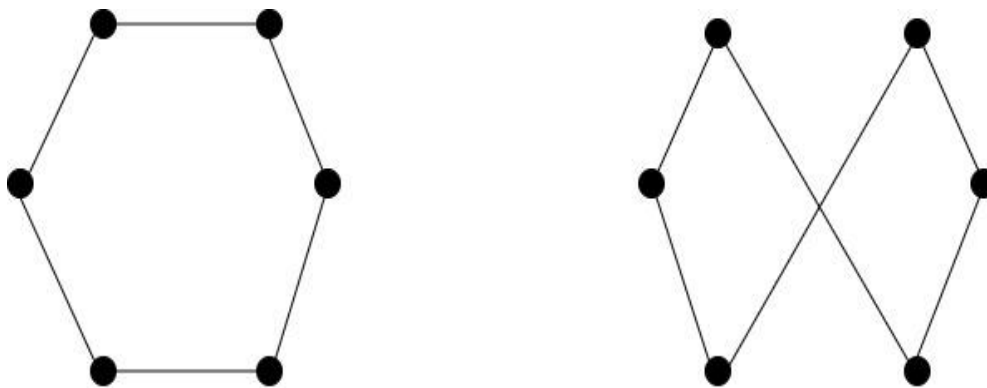
(Μαρινάκης,Μυγδαλάς,2008)

Οι σημαντικότερες μέθοδοι οι οποίες θα αναλυθούν στη συνέχεια είναι ταυτόχρονα και αυτές που θα χρησιμοποιηθούν στη διπλωματική :

- 2- opt
- 1-1 exchange
- 1-0 relocate

### **2-opt**

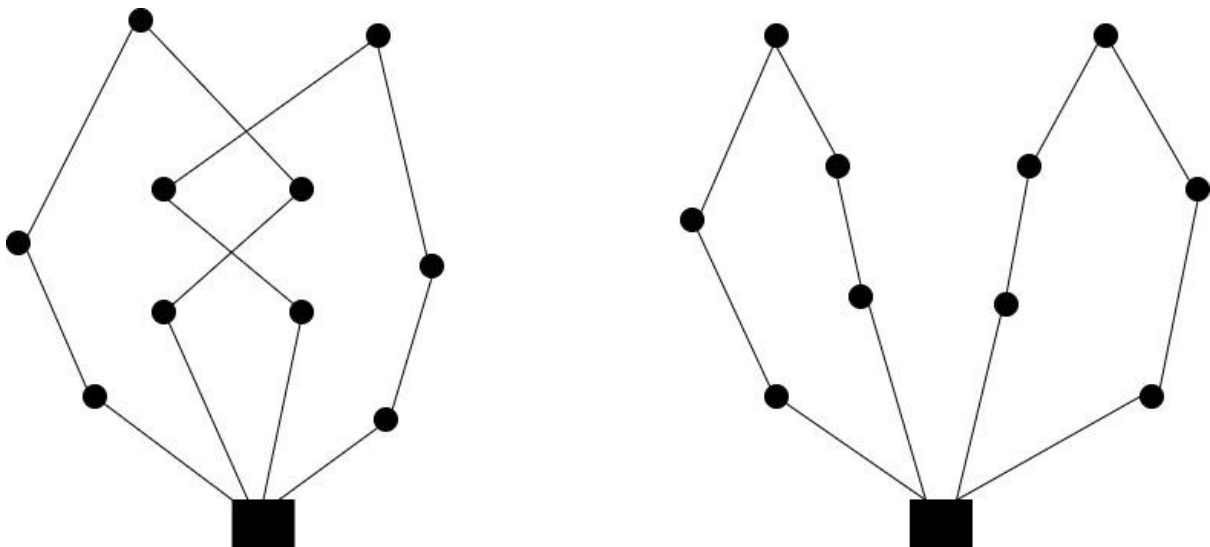
Η μέθοδος αυτή αναφέρεται στη διαγραφή 2 ακμών και στην επανασύνδεσή δυο μονοπατιών με διαφορετικό τρόπο έτσι ώστε να καθοριστεί μια καινούρια διαδρομή. Προϋπόθεση για να διατηρηθεί η νέα λύση είναι να έχει μικρότερο κόστος από την αρχική.



Σχήμα 2 : Αλγόριθμος 2-opt

### 1-1 exchange

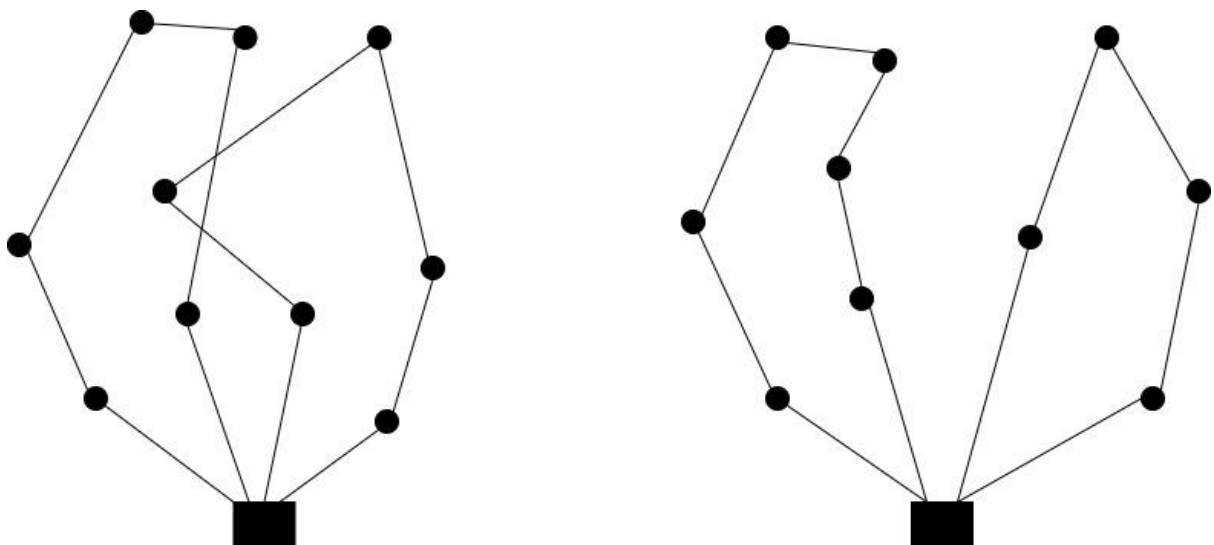
Με την ίδια λογική δηλαδή τη μείωση του κόστους χρησιμοποιείται και η μέθοδος 1-1 exchange. Στην περίπτωση αυτή υπάρχει ταυτόχρονη ανταλλαγή 2 πελατών που βρίσκονται όμως σε διαφορετικές διαδρομές.



Σχήμα 3 : Αλγόριθμος 1-1 exchange

### 1-0 relocate

Η μέθοδος 1-0 επανατοποθέτηση (1-0 relocate) αναφέρεται στην διαγραφή ενός πελάτη από μια διαδρομή και στην επανατοποθέτησή του σε μια άλλη διαδρομή με απώτερο στόχο τη βελτίωση του κόστους.



Σχήμα 4 : Αλγόριθμος 1-0 relocate

## 3.2 ΜΕΘΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Οι μεθευρετικοί αλγόριθμοι είναι μέθοδοι επίλυσης που συνδυάζουν διαδικασίες τοπικής αναζήτησης και υψηλότερου επιπέδου στρατηγικές για να δημιουργήσουν μια διαδικασία που είναι ικανή να ξεφύγει από κάποιο τοπικό ελάχιστο. Για την επίλυση του προβλήματος δρομολόγησης με παράδοση και παραλαβή και χρονικούς περιορισμούς χρησιμοποιείται η Διαδικασία Άπληστης Τυχαιοποιημένης Προσαρμοστικής Αναζήτησης (Greedy Randomized Adaptive Search Procedure).

Τα χαρακτηριστικά των μεθευρετικών είναι τα εξής:

- Μοντελοποιούν ένα φαινόμενο που υπάρχει στη φύση.
- Μπορούν να μεταφερθούν εύκολα σε παράλληλη μορφή.
- Είναι προσαρμοστικοί αλγόριθμοι.

### 3.2.1 ΔΙΑΔΙΚΑΣΙΑ ΑΠΛΗΣΤΗΣ ΤΥΧΟΠΟΙΗΜΕΝΗΣ ΠΡΟΣΑΡΜΟΣΤΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ (GRASP)

Η διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης είναι μια επαναληπτική διαδικασία για την εύρεση προσεγγιστικών λύσεων σε προβλήματα συνδυαστικής βελτιστοποίησης. Αυτή η τυχαιοποιημένη τεχνική παρέχει μια εφικτή λύση σε κάθε επανάληψη. Οι επαναλήψεις του GRASP σταματάνε όταν κάποιο κριτήριο σταματήματος ικανοποιείται. Το τελικό αποτέλεσμα είναι απλά η καλύτερη λύση που βρέθηκε από όλες τις επαναλήψεις. Κάθε φάση αποτελείται από δύο άλλες επιμέρους, μια φάση κατασκευής μιας αρχικής λύσης και μια διαδικασία τοπικής αναζήτησης για βελτιστοποίηση αυτής της λύσης. Στη φάση κατασκευής μια τυχαιοποιημένη συνάρτηση απληστίας χρησιμοποιείται για να κατασκευαστεί μια αρχική λύση. Στη συνέχεια, η αρχική λύση θα βελτιωθεί με τη βοήθεια της τοπικής αναζήτησης.

Η φάση της κατασκευής αρχικής λύσης αποτελείται από την πρόσθεση ενός στοιχείου στην μη ολοκληρωμένη λύση σε κάθε επανάληψη. Η στρατηγική επιλογής του επόμενου στοιχείου βασίζεται στην τυχαία επιλογή από μια λίστα υποψηφίων που ονομάζεται λίστα περιορισμού των υποψηφίων για εισαγωγή στη λύση, στην οποία κάθε στοιχείο κατατάσσεται βάσει μιας συνάρτησης απληστίας. Η κατασκευή της λίστας υποψηφίων είναι ίσως το πιο σημαντικό κομμάτι της μεθόδου αφού από αυτό ελέγχεται η διασπορά των λύσεων που θα παραχθούν. Η τυχαία επιλογή του στοιχείου σημαίνει ότι δεν είναι ανάγκη να επιλεγεί το πρώτο στοιχείο στη λίστα.

Η λύση που δημιουργείται στη φάση κατασκευής δεν εγγυάται ότι είναι τοπικό ελάχιστο και για αυτό το λόγο μια φάση τοπικής αναζήτησης εφαρμόζεται για να παράξει μια λύση που να περιέχει τοπικό ελάχιστο.

Μια γενικευμένη μορφή του αλγορίθμου παρουσιάζεται στη συνέχεια.

### **GRASP**

$c(s^*) = \infty$

**for** για όσο το κριτήριο σταματήματος του GRASP δεν ικανοποιείται

**do**

Κατασκευή μιας αρχικής λύσης  $s$

Εφαρμογή τοπικής αναζήτησης στην  $s$

**if**  $c(s) < c(s^*)$  **then**

$s^* = s$

**endif**

**endfor**

**return**  $s^*$

(Μαρινάκης,Μυγδαλάς,2008)

## ΚΕΦΑΛΑΙΟ 4<sup>ο</sup>

Στο κεφάλαιο αυτό θα γίνει μια αναλυτική περιγραφή του κώδικα που χρησιμοποιήθηκε για την επίτευξη των αποτελεσμάτων. Ο κώδικας χωρίζεται σε δύο στάδια. Το πρώτο στάδιο είναι η εύρεση αρχικών λύσεων με τη βοήθεια δυο μεθόδων όπως θα περιγραφούν. Το δεύτερο στάδιο αφορά τη βελτίωση των αρχικών λύσεων με μεθόδους τοπικής αναζήτησης μέσω των οποίων θα εξαχθούν και τα τελικά αποτελέσματα.

### 4.1 ΠΕΡΙΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ ΚΑΙ ΕΠΙΛΥΣΗΣ ΤΟΥ

Τα δεδομένα για κάθε παράδειγμα όπως παρουσιάζονται και στους παρακάτω πίνακες απαρτίζονται από πληθώρα πληροφοριών όπως είναι ο αριθμός των πελατών που πρέπει να εξυπηρετηθούν ορίζοντας ως κόμβο 1 την αποθήκη από όπου και ξεκινάει το όχημα, ο αριθμός των οχημάτων που είναι διαθέσιμος, η χωρητικότητα του κάθε οχήματος καθώς και τις συντεταγμένες  $x$  και  $y$  στις οποίες βρίσκεται ο κάθε πελάτης. Επιπλέον γνωστή θεωρείται η ζήτηση του κάθε προϊόντος η οποία μπορεί να είναι θετική ή αρνητική. Αυτό καθορίζεται από το αν πρέπει ένα προϊόν να παραληφθεί ή να παραδοθεί. Επομένως στην περίπτωση που ένα προϊόν θα παραληφθεί από τον εκάστοτε πελάτη τότε η ζήτηση θα είναι θετική ενώ όταν παραδοθεί θα είναι αρνητική. Ακόμα ως δεδομένο υπάρχει το λεγόμενο παράθυρο του κάθε πελάτη δηλαδή ο νωρίτερος χρόνος, η έναρξη του παραθύρου, και ο αργότερος χρόνος, η λήξη του παραθύρου. Πιο συγκεκριμένα ο νωρίτερος χρόνος αναφέρεται στη χρονική στιγμή από την οποία αρχίζει η εξυπηρέτηση του πελάτη χωρίς να υπάρχει αναμονή και ο αργότερος χρόνος αναφέρεται στη χρονική στιγμή μέχρι την οποία θα μπορεί να εξυπηρετηθεί κάποιος πελάτης. Από τη στιγμή που το όχημα φτάσει στο πελάτη υπάρχει συγκεκριμένος χρόνος ώστε να εξυπηρετηθεί και δίδεται ως χρόνος εξυπηρέτησης. Ως τελευταίο δεδομένο δίδεται η στήλη “παραλαβής” η οποία δείχνει τον πελάτη από όπου θα παραληφθεί ένα προϊόν και η στήλη “παραδόσης” δηλαδή τον πελάτη στον οποίο θα παραδοθεί αντίστοιχα το συγκεκριμένο προϊόν.

Στον παρακάτω πίνακα απεικονίζεται ενδεικτικά ο πίνακας με τα δεδομένα έτσι όπως περιγράφηκαν.

number of vehicles				capacity		Speed(not used)		
K				Q		S		
Task no.	x	y	Demand	Earliest time	Latest time	Service time	pickup	delivery
1	$X_1$	$Y_1$	$Q_1$	$E_1$	$L_1$	$S_1$	$P_1$	$D_1$
2	$X_2$	$Y_2$	$Q_2$	$E_2$	$L_2$	$S_2$	$P_2$	$D_2$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Πίνακας 1 : Αναπαράσταση δεδομένων ( Li & Lim)



Για την μοντελοποίηση και επίλυση του προβλήματος χρησιμοποιήθηκε το περιβάλλον του προγράμματος MATLAB.

Αφού εισαχθούν στη MATLAB τα δεδομένα δημιουργούνται ορισμένοι πίνακες που θα βοηθήσουν στη μοντελοποίηση του προβλήματος. Οι τετραγωνικοί αυτοί πίνακες δείχνουν τις τιμές των αποστάσεων μεταξύ των πελατών καθώς και το χρόνο που διανύει το όχημα ώστε να μεταφερθεί από ένα πελάτη σε ένα άλλο και είναι πρακτικά ίδιοι εφόσον υπολογίζονται από τη σχέση  $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

- d: απόσταση των δύο σημείων
- (x1,y1): συντεταγμένες του 1<sup>ου</sup> σημείου
- (x2,y2): συντεταγμένες του 2<sup>ου</sup> σημείου

Επιπλέον δημιουργούνται ορισμένοι μηδενικοί πίνακες όπου με την ολοκλήρωση μια διαδρομής ,δηλαδή με την επιστροφή του οχήματος στην αποθήκη , θα αποθηκεύονται σε αυτούς η διαδρομή και το συνολικό της κόστος αντίστοιχα.

Γίνεται υπόθεση ότι το όχημα ξεκινάει από την αποθήκη άδαιο και ότι από τα ζευγάρια πελατών που δημιουργούνται από τα δεδομένα θα εξυπηρετήσει πρώτα αυτό με τη θετική ζήτηση.

Ένας πελάτης επιλέγεται εάν δεν έχει εξυπηρετηθεί ακόμα και αν ικανοποιεί ταυτόχρονα κάποιους περιορισμούς. Επομένως στη περίπτωση που επιλεγεί πρέπει ο διαθέσιμος χώρος του οχήματος να είναι θετικός και μικρότερος της συνολικής χωρητικότητας του και επιπλέον να ικανοποιούνται οι χρονικοί περιορισμοί. Όσον αφορά τους χρονικούς περιορισμούς προτιμούνται οι πελάτες στους οποίο το όχημα θα φτάσει εντός των χρονικών παραθύρων τους , αμέσως μετά οι πελάτες στους οποίους το όχημα φτάνει πριν τον νωρίτερο χρόνο εξυπηρέτησης τους ενώ απορρίπτονται εκείνοι στους οποίους το όχημα θα φτάσει μετά τον αργότερο χρόνο τους. Έτσι δημιουργείται μια λίστα από πελάτες, ένα διάνυσμα, που ικανοποιούν τους παραπάνω περιορισμούς και από την οποία θα επιλεγεί αυτός στον οποίο θα μεταφερθεί στη συνέχεια το όχημα.

Η επιλογή του επόμενου πελάτη στη παρούσα διπλωματική γίνεται με δύο τρόπους:

- 1) Με βάση τη πρώτη μέθοδο επιλέγεται εκείνος ο πελάτης, ο οποίος είναι ο κοντινότερος με βάση την απόσταση του από τον προηγούμενο . Η διαδικασία επαναλαμβάνεται μέχρι να εξυπηρετηθεί και ο τελευταίος πελάτης.
- 2) Ως δεύτερη μέθοδο χρησιμοποιείται η διαδικασία της άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης. Στη περίπτωση αυτή δημιουργείται μια λίστα στην οποία εισάγονται οι τέσσερις κοντινότεροι πελάτες οι οποίοι θα έχουν ίση πιθανότητα να επιλεγούν. Η επιλογή κάθε φορά είναι τυχαία. Η διαδικασία επαναλαμβάνεται έως ότου αδειάσει η λίστα με τους διαθέσιμους πελάτες.

Επομένως από τη παραπάνω διαδικασία των αρχικών λύσεων δημιουργείται ένα σύνολο διαδρομών και ένα συνολικό κόστος των διαδρομών αυτών. Χρησιμοποιώντας τις τοπικές αναζητήσεις βασικός στόχος είναι η μείωση του κόστους αυτού βελτιώνοντας τις διαδρομές ή ακόμα και μειώνοντας τις.

- Ως πρώτη τοπική αναζήτηση χρησιμοποιείται η μέθοδος 1-0 relocate. Σύμφωνα με τη μέθοδο αυτή, όπως έχει ήδη αναφερθεί, επιλέγεται ένα ζευγάρι κόμβων, δηλαδή ο πελάτης με τη θετική ζήτηση και ο αντίστοιχος με την αρνητική ζήτηση, και επανατοποθετούνται σε οποιαδήποτε άλλη διαδρομή με στόχο τη μείωση του κόστους. Έτσι ξεκινώντας η διαδικασία ξεκινάει να επιλέγει κόμβους από τη τελευταία διαδρομή των αρχικών λύσεων που θεωρούνται περισσότερο προβληματικές από τις υπόλοιπες, δηλαδή είναι η διαδρομή με τους λιγότερους κόμβους και με μεγάλο κόστος διαδρομής. Στη συνέχεια ξεκινώντας από τη πρώτη διαδρομή η μέθοδος θα εξετάσει αν ο κόμβος με θετική ζήτηση μπορεί να μπει σε οποιαδήποτε θέση μειώνοντας το κόστος και ικανοποιώντας τους χρονικούς περιορισμούς και τη χωρητικότητα του οχήματος. Εάν υπάρξει εφικτή θέση τότε αποθηκεύεται η τροποποιημένη αυτή διαδρομή και η διαδικασία συνεχίζει με τη τοποθέτηση του κόμβου με την αρνητική ζήτηση. Για τον κόμβο αυτό η μέθοδος θα εξετάσει όλες τις πιθανές θέσεις δεξιά από τη θέση στην οποία τοποθετήθηκε ο κόμβος με θετική ζήτηση και στη περίπτωση που βρεθεί θέση θα γίνει η επανατοποθέτηση των δύο κόμβων από τη τελευταία διαδρομή στη πρώτη. Εάν όμως δεν υπάρξει εφικτή θέση είτε για τον θετικό κόμβο είτε για τον αρνητικό κόμβο στη πρώτη διαδρομή η αναζήτηση θα συνεχιστεί στη δεύτερη διαδρομή έως ότου εξετάσουμε όλες τις διαδρομές. Η διαδικασία επαναλαμβάνεται μέχρι να εξεταστούν όλοι οι κόμβοι σε όλες τις πιθανές θέσεις.

Αφού πραγματοποιηθούν αλλαγές με τη μέθοδο 1-0 relocate θα υπάρξει ένα σύνολο διαδρομών τις περισσότερες φορές αρκετά μικρότερο από το σύνολο των αρχικών λύσεων και επιπλέον αρκετά βελτιωμένο ως προς τη μείωση του συνολικού κόστους. Για να επιτευχθεί όμως ένα καλύτερο αποτέλεσμα χρησιμοποιούνται συνδυαστικά ακόμα δυο μέθοδοι, η μέθοδος 1-1 exchange και η μέθοδος 2-opt. Στα παραδείγματα με χωρητικότητα οχήματος 200 και 700 ο αριθμός των επαναλήψεων για τον οποίο θα υλοποιηθούν οι μέθοδοι αυτοί είναι 1000 ενώ ο αριθμός των επαναλήψεων για τα παραδείγματα με χωρητικότητα οχήματος 1000 είναι 3000.

- Στη μέθοδο 1-1 exchange γίνεται ανταλλαγή κόμβων μεταξύ 2 διαδρομών. Σύμφωνα με τη μέθοδο αυτή, επιλέγονται με τυχαίο τρόπο 2 διαδρομές από το σύνολο διαδρομών. Για κάθε διαδρομή επιλέγεται τυχαία ένας κόμβος με θετική ζήτηση και υπολογίζεται η θέση στην άλλη διαδρομή στην οποία μπορεί να τοποθετηθεί με μικρότερο κόστος. Εφόσον έχει βρεθεί η θέση για το θετικό κόμβο εξετάζονται οι πιθανές θέσεις οι οποίες βρίσκονται δεξιά του θετικού κόμβου στις οποίες μπορεί να τοποθετηθεί ο αρνητικός κόμβος χωρίς να παραβιάζονται οι περιορισμοί. Μόλις ολοκληρωθεί η διαδικασία και οι νέες διαδρομές δίνουν μικρότερο κόστος από το κόστος πριν τις αλλαγές τότε οι αλλαγές αυτές γίνονται αποδεκτές και ενημερώνεται εκ νέου το σύνολο των διαδρομών.

### Βήμα 1<sup>ο</sup>

Επιλέγονται τυχαία οι δυο παρακάτω διαδρομές. Στη συνέχεια από το πλήθος των κόμβων με θετική ζήτηση επιλέγεται τυχαία ένας κόμβος από κάθε διαδρομή. Έστω ότι από τη πρώτη διαδρομή, με κόστος  $c_1$ , επιλέγεται ο κόμβος 41 με ζευγάρι του τον 107 (κόμβος με αρνητική ζήτηση) και από τη δεύτερη, με κόστος  $c_2$ , διαδρομή ο 91 με ζευγάρι του τον 85.

1	44	43	42	<b>41</b>	45	46	47	49	<b>107</b>	52	53	50	1
1	<b>91</b>	88	87	84	83	<b>85</b>	86	89	90	92	51	48	1

### Βήμα 2<sup>ο</sup>

Από τις διαδρομές αυτές αφαιρούνται οι επιλεγμένοι κόμβοι έτσι ώστε να αποφασιστεί σε ποιά θέση στη δεύτερη διαδρομή εάν μπει ο 41 θα δώσει μικρότερο κόστος και το ίδιο για τη πρώτη διαδρομή εάν μπει ο 91.

1	44	43	42	45	46	47	49	52	53	50	1
1	88	87	84	83	86	89	90	92	51	48	1

### Βήμα 3<sup>ο</sup>

Έστω ότι δημιουργούνται οι παρακάτω δύο διαδρομές.

1	44	<b>91</b>	43	42	45	46	47	49	52	53	50	1
1	88	87	84	83	<b>41</b>	86	89	90	92	51	48	1

### Βήμα 4<sup>ο</sup>

Εάν οι διαδρομές είναι εφικτές ως προς τους περιορισμούς τότε η μέθοδος ψάχνει θέση για τα ζευγάρια των θετικών κόμβων δηλαδή για τους κόμβους 107 και 85. Δύο διαδρομές που μπορούν να προκύψουν είναι οι εξής :

1	44	<b>91</b>	43	42	45	46	47	49	52	53	<b>85</b>	50	1
1	88	87	84	83	<b>41</b>	86	89	<b>107</b>	90	92	51	48	1

### Βήμα 5<sup>ο</sup>

Εάν και οι δυο παραπάνω διαδρομές είναι εφικτές και ισχύει ότι  $c_1' + c_2' < c_1 + c_2$  τότε γίνεται αποδεκτή η αλλαγή και ενημερώνεται τόσο ο πίνακας των διαδρομών όσο και ο πίνακας των κοστών.

- Στη μέθοδο 2-opt πραγματοποιείται μετατόπιση κόμβων σε μια διαδρομή. Επομένως γίνεται τυχαία επιλογή μιας διαδρομής και στη συνέχεια τυχαία επιλογή ενός κόμβου. Εάν ο κόμβος αυτός έχει θετική ζήτηση μπορεί να τοποθετηθεί σε οποιαδήποτε θέση αριστερά του ζευγαριού του, δηλαδή του κόμβου με αρνητική ζήτηση, ενώ εάν ο κόμβος που επιλέγεται έχει αρνητική ζήτηση μπορεί να τοποθετηθεί σε θέσεις δεξιά του ζευγαριού του, δηλαδή του κόμβου με θετική ζήτηση. Δεκτή γίνεται η αλλαγή στη περίπτωση που ικανοποιεί τους περιορισμούς και δίνει βελτιωμένο κόστος. Αναλυτικά φαίνονται παρακάτω τα βήματα της μεθόδου.

### Βήμα 1<sup>ο</sup>

Επιλέγεται τυχαία η παρακάτω διαδρομή. Το κόστος της διαδρομής είναι c.

1	44	43	42	41	45	47	46	49	52	51	102	53	50	48	1
---	----	----	----	----	----	----	----	----	----	----	-----	----	----	----	---

### Βήμα 2<sup>ο</sup>

Επιλέγεται τυχαία ο κόμβος 42.

1<sup>η</sup> περίπτωση : Έστω ότι ο κόμβος 42 έχει θετική ζήτηση και το ζευγάρι του, δηλαδή ο κόμβος με αρνητική ζήτηση είναι ο 46.

1	44	43	<b>42</b>	41	45	47	<b>46</b>	49	52	51	102	53	50	48	1
---	----	----	-----------	----	----	----	-----------	----	----	----	-----	----	----	----	---

Ελέγχονται λοιπόν οι θέσεις αριστερά του 46 και όποια θέση δώσει εφικτή λύση ως προς τους περιορισμούς και  $c' < c$  γίνεται δεκτή. Επομένως μια λύση θα μπορούσε να είναι η εξής :

1	<b>42</b>	43	44	41	45	47	<b>46</b>	49	52	51	102	53	50	48	1
---	-----------	----	----	----	----	----	-----------	----	----	----	-----	----	----	----	---

2<sup>η</sup> περίπτωση : Έστω ότι ο κόμβος 42 έχει αρνητική ζήτηση και το ζευγάρι του, δηλαδή ο κόμβος με θετική ζήτηση είναι ο 44.

1	<b>44</b>	43	<b>42</b>	41	45	47	46	49	52	51	102	53	50	48	1
---	-----------	----	-----------	----	----	----	----	----	----	----	-----	----	----	----	---

Ελέγχονται λοιπόν οι θέσεις δεξιά του 44 και όποια θέση δώσει εφικτή λύση ως προς τους περιορισμούς και  $c' < c$  γίνεται δεκτή. Επομένως μια λύση θα μπορούσε να είναι η εξής :

1	<b>44</b>	43	41	45	47	46	49	52	51	<b>42</b>	102	53	50	48	1
---	-----------	----	----	----	----	----	----	----	----	-----------	-----	----	----	----	---

### **Βήμα 3<sup>ο</sup>**

Ενημερώνεται το σύνολο των διαδρομών αντικαθιστώντας τη παλιά διαδρομή με τη καινούρια καθώς και το πίνακα με τα κόστη αντικαθιστώντας το  $c$  με το  $c'$ .

## ΚΕΦΑΛΑΙΟ 5<sup>ο</sup>

### 5.1 ΑΠΟΤΕΛΕΣΜΑΤΑ

Τα δεδομένα που χρησιμοποιήθηκαν κατηγοριοποιούνται ως προς τη χωρητικότητα των οχημάτων που είναι διαθέσιμα. Επομένως στη παρούσα διπλωματική υπάρχουν για τα οχήματα οι χωρητικότητες 200,700 και 1000. Παρακάτω παρουσιάζονται τα αποτελέσματα σε πίνακες όπου οι δυο πρώτες στήλες αναφέρονται στις βέλτιστες διαδρομές και στα βέλτιστα κόστη όπως δόθηκαν από τη βιβλιογραφία. Στη συνέχεια φαίνονται αναλυτικά τα αποτελέσματα των δύο μεθόδων. Πρώτα παρουσιάζεται η μέθοδος του κοντινότερου γείτονα και έπειτα η διαδικασία άπληστης τυχαιοποιημένης αναζήτησης. Η όγδοη στήλη αφορά το καλύτερο κόστος των παραδειγμάτων ανεξαρτήτως μεθόδου και η τελευταία στήλη αφορά τις βέλτιστες διαδρομές οι οποίες εξήχθησαν από τον αλγόριθμο. Τέλος στους παρακάτω πίνακες παρουσιάζονται οι αποκλίσεις των αποτελεσμάτων των δύο μεθόδων και των καλύτερων λύσεων από τα βέλτιστα.

Παραδείγματα	Βέλτιστες λύσεις		Λύσεις με κοντινότερο		Λύσεις με grasp		Καλύτερη λύση ως προς κόστος	Καλύτερη λύση ως προς τις διαδρομές
	Διαδρομές	Κόστος	Διαδρομές	Κόστος	Διαδρομές	Κόστος		
<b>lr101</b>	19	1650	<b>22</b>	<b>1814</b>	25	2135	1814	22
<b>lr102</b>	17	1487	<b>18</b>	<b>1636</b>	22	1982,8	1636	18
<b>lr103</b>	13	1292	18	1780,5	<b>17</b>	<b>1605,4</b>	1605,4	17
<b>lr104</b>	9	1013	<b>14</b>	<b>1307,6</b>	<b>14</b>	1454,9	1307,6	14
<b>lr105</b>	14	1377	<b>16</b>	<b>1557,4</b>	19	1896,9	1557,4	16
<b>lr106</b>	12	1252	<b>15</b>	<b>1542,8</b>	17	1789,4	1542,8	15
<b>lr107</b>	10	1111	<b>14</b>	<b>1456,6</b>	<b>14</b>	1573,8	1456,6	14
<b>lr108</b>	9	968	<b>12</b>	<b>1236,5</b>	13	1378,1	1236,5	12
<b>lr109</b>	11	1208	<b>14</b>	<b>1466,6</b>	15	1603,6	1466,6	14
<b>lr110</b>	10	1159	<b>13</b>	<b>1375,1</b>	15	1391,4	1375,1	13
<b>lr111</b>	10	1108	<b>14</b>	<b>1409,2</b>	<b>14</b>	1523,6	1409,2	14
<b>lr112</b>	9	1003	<b>13</b>	<b>1323,3</b>	<b>13</b>	1325,7	1323,3	13
<b>lr201</b>	4	1253	7	1627,8	<b>6</b>	<b>1418,7</b>	1418,7	6
<b>lr202</b>	3	1197	6	1738,8	<b>5</b>	<b>1607,5</b>	1607,5	5
<b>lr203</b>	3	949	<b>4</b>	<b>1344,1</b>	5	1611,5	1344,1	4
<b>lr204</b>	2	849	<b>4</b>	<b>1343,3</b>	5	1430,6	1343,3	4
<b>lr205</b>	3	1054	<b>5</b>	<b>1614,3</b>	<b>5</b>	1763,8	1614,3	5
<b>lr206</b>	3	931	<b>5</b>	<b>1435,5</b>	<b>5</b>	1708,7	1435,5	5
<b>lr207</b>	2	903	<b>4</b>	1487,5	<b>4</b>	<b>1395,2</b>	1395,2	4
<b>lr208</b>	2	734	<b>4</b>	<b>1164,5</b>	5	1524,5	1164,5	4
<b>lr209</b>	3	930	<b>5</b>	<b>1448,6</b>	6	1490,3	1448,6	5
<b>lr210</b>	3	964	<b>5</b>	<b>1575,5</b>	<b>5</b>	1831,5	1575,5	5
<b>lr211</b>	2	911	<b>4</b>	<b>1119,3</b>	<b>4</b>	1478,4	1119,3	4
<i>Απόκλιση από βέλτιστα</i>				<b>36%</b>		<b>53,4%</b>	<b>33,8%</b>	

Πίνακας 2 : Αποτελέσματα παραδειγμάτων lr1\_ χωρητικότητας 200 και lr\_2 χωρητικότητας 1000

Λύσεις από κοντινότερο	<b>19</b>
Λύσεις από GRASP	<b>4</b>

Πίνακας 3

	Βέλτιστες λύσεις		Λύσεις με κοντινότερο		Λύσεις με grasp		Καλύτερη λύση ως προς κόστος	Καλύτερη λύση ως προς τις διαδρομές
Παραδείγματα	Διαδρομές	Κόστος	Διαδρομές	Κόστος	Διαδρομές	Κόστος		
lc101	10	828	10	828,9	13	1454,6	828,9	10
lc102	10	828	11	1106	11	1099,2	1099,2	11
lc103	9	1035	13	1259	12	1278,4	1259	12
lc104	9	860	10	1159,5	11	1479,1	1159,5	10
lc105	10	828	11	910	15	1535,1	910	11
lc106	10	828	10	856	14	1632,5	856	10
lc107	10	828	10	852,9	12	1163,7	852,9	10
lc108	10	826	10	860	11	901	860	10
lc109	9	1000	10	1013,4	11	979,9	979,9	10
lc201	3	591	3	591,5	5	799,7	591,5	3
lc202	3	591	5	784,9	5	994,9	784,9	5
lc203	3	591	4	1138	5	1155,8	1138	4
lc204	3	590	4	1135,8	5	1294,6	1135,8	4
lc205	3	588	4	661,3	5	1255,4	661,3	4
lc206	3	588	4	697,4	5	1197,5	697,4	4
lc207	3	588	4	720,4	4	988,7	720,4	4
lc208	3	588	4	646,4	5	1220,9	646,4	4
Απόκλιση από βέλτιστα				23,13%		67,39%	22,89%	

Πίνακας 4 : Αποτελέσματα παραδειγμάτων lc1\_ χωρητικότητας 200 και lc\_2 χωρητικότητας 700

Λύσεις από κοντινότερο	15
Λύσεις από GRASP	2

Πίνακας 5



Παραδείγματα	Βέλτιστες λύσεις		Λύσεις με κοντινότερο		Λύσεις με grasp		Καλύτερη λύση ως προς κόστος	Καλύτερη λύση ως προς τις διαδρομές
	Διαδρομές	Κόστος	Διαδρομές	Κόστος	Διαδρομές	Κόστος		
<b>lrc101</b>	14	1708	<b>18</b>	<b>2008</b>	20	2239,8	2008	18
<b>lrc102</b>	12	1558	<b>15</b>	<b>1809,9</b>	17	2126,4	1809,9	15
<b>lrc103</b>	11	1258	<b>15</b>	1826,2	<b>15</b>	<b>1739,7</b>	1739,7	15
<b>lrc104</b>	10	1128	<b>13</b>	<b>1500</b>	14	1518,6	1500	13
<b>lrc105</b>	13	1637	<b>19</b>	<b>2043,9</b>	<b>19</b>	2168	2043,9	19
<b>lrc106</b>	11	1424	<b>15</b>	1857,1	<b>15</b>	<b>1845,5</b>	1845,5	15
<b>lrc107</b>	11	1230	<b>13</b>	1612,9	14	<b>1530,1</b>	1530,1	13
<b>lrc108</b>	10	1147	<b>13</b>	1515,5	14	<b>1441,7</b>	1441,7	13
<b>lrc201</b>	4	1406	<b>6</b>	2152	6	<b>1901,8</b>	1901,8	6
<b>lrc202</b>	3	1374	<b>6</b>	1961,1	6	<b>1736,2</b>	1736,2	6
<b>lrc203</b>	3	1089	<b>6</b>	<b>1533,3</b>	<b>6</b>	1739,4	1533,3	6
<b>lrc204</b>	3	818	<b>5</b>	<b>1375,2</b>	6	1509,8	1375,2	5
<b>lrc205</b>	4	1302	6	1852,3	<b>5</b>	<b>1638,8</b>	1638,8	5
<b>lrc206</b>	3	1159	<b>5</b>	1849,3	<b>5</b>	<b>1629,5</b>	1629,5	5
<b>lrc207</b>	3	1062	7	<b>1691,1</b>	<b>6</b>	1752,1	1691,1	6
<b>lrc208</b>	3	852	<b>5</b>	<b>1479,2</b>	<b>5</b>	1836,5	1479,2	5
<i>Απόκλιση από βέλτιστα</i>				<b>41,8%</b>		<b>44%</b>	<b>36,2%</b>	

Πίνακας 6 : Αποτελέσματα παραδειγμάτων lrc1\_ χωρητικότητας 200 και lrc\_2 χωρητικότητας 1000

Λύσεις από κοντινότερο	8
Λύσεις από GRASP	8

Πίνακας 7

Συνοψίζοντας παρατηρείται ότι και στα τρία είδη παραδειγμάτων Ic, Irc, Ir μικρότερο ποσοστό απόκλισης εμφανίζει η μέθοδος του πλησιέστερου γείτονα. Με τη μέθοδο αυτή φαίνεται ότι ο κώδικας δούλεψε καλύτερα για τα παραδείγματα Ic αφού το ποσοστό απόκλισης είναι 23,13% έναντι των άλλων δυο ποσοστών 36% και 41,8%. Η μέθοδος GRASP φαίνεται ότι δίνει μεγαλύτερες αποκλίσεις και για τα τρία είδη παραδειγμάτων.

Βάσει αυτών των δεδομένων είναι προφανές ότι οι καλύτερες λύσεις που δίνονται δημιουργούνται ως επί το πλείστον από τη μέθοδο του κοντινότερου γείτονα. Αυτό όμως δε σημαίνει ότι κάποιες από τις λύσεις που προήλθαν από τη μέθοδο τη διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης (GRASP) δεν επιλέχθηκαν ως καλύτερες λύσεις.

Αυτό φαίνεται στους βοηθητικούς πίνακες 3, 5 και 7 οι οποίοι δείχνουν πόσες λύσεις επιλέχθηκαν από κάθε μέθοδο. Πιο αναλυτικά για τα παραδείγματα Ir από τις συνολικά 23 καλύτερες λύσεις οι 19 ήταν με τη μέθοδο του κοντινότερου και οι 4 με τη διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης (GRASP). Για τα παραδείγματα Irc από τις συνολικά 16 καλύτερες λύσεις υπήρξαν 8 από κάθε μέθοδο. Ενώ για τα παραδείγματα Ic, οι 15 λύσεις ήταν από τη μέθοδο του πλησιέστερου και οι 2 από τη μέθοδο GRASP από τις συνολικά 17 λύσεις.

## ΚΕΦΑΛΑΙΟ 6<sup>ο</sup>

### 6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Στη παρούσα διπλωματική ζητήθηκε το πρόβλημα δρομολόγησης με παραλαβή και παράδοση και χρονικούς περιορισμούς να επιλυθεί με τη διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης (grasp). Όπως έχει ήδη αναφερθεί η διαδικασία αυτή δημιουργεί μια λίστα από πελάτες, οι οποίοι ικανοποιούν τους περιορισμούς και δίνοντάς τους ίση πιθανότητα επιλογής, επιλέγει τυχαία τον επόμενο πελάτη που θα επισκεφτεί το όχημα. Ως αρχική προσέγγιση του προβλήματος μετά τη μοντελοποίησή του ήταν ότι για την εύρεση των αρχικών λύσεων η μέθοδος έδινε πληθώρα αποτελεσμάτων λόγω τυχειότητας. Παρατηρήθηκε ότι όσες φορές και να έτρεχε το πρόγραμμα τόσο το κόστος των διαδρομών όσο και το πλήθος των οχημάτων που θα χρειάζονταν δεν πλησίαζαν τα βέλτιστα. Υπήρχαν περιπτώσεις στις οποίες το πλήθος των διαδρομών που εμφάνιζε ο κώδικας ήταν μεγαλύτερο από το πλήθος των οχημάτων που υπήρχαν στην αποθήκη. Το μειονέκτημα αυτό οδήγησε στη προσπάθεια εύρεσης άλλων μεθόδων ή ακόμα και παραλλαγής της μεθόδου.

Η πρώτη προσπάθεια βελτίωσης των λύσεων ήταν η επεξεργασία της λίστας των υποψηφίων. Η τυχειότητα φάνηκε να μη δουλεύει ικανοποιητικά όταν δινόταν στη λίστα μεγάλος αριθμός υποψηφίων. Αυτό γινόταν διότι κάθε υποψήφιος είχε την ίδια πιθανότητα να επιλεγεί. Ως αποτέλεσμα αυτού εμφανίζονταν λύσεις με πολύ μικρές διαδρομές με αρκετά μεγάλο κόστος. Η μείωση της λίστας των υποψηφίων πελατών φάνηκε να δουλεύει καθώς το κόστος βελτιωνόταν. Δοκιμάστηκαν αρκετά μήκη στη λίστα. Τα καλύτερα αποτελέσματα πάρθηκαν όταν η λίστα είχε μήκος 4 , δηλαδή αποτελούνταν από τους 4 κοντινότερους πελάτες, καθένας από τους οποίους είχε 25% πιθανότητα να επιλεγεί.

Ως συνέχεια στην αναζήτηση βελτίωσης του κώδικα δοκιμάστηκε μια παραλλαγή του grasp. Αυτή η μέθοδος ήταν ο grasp με κοντινότερα παράθυρα, μέθοδος η οποία λειτουργεί με τη λογική που δουλεύει και ο grasp με βάση το κοντινότερο γείτονα απλά η λίστα δημιουργείται από κόμβους σύμφωνα με το ποιον θα επισκεφτούμε συντομότερα δηλαδή με βάση το χρονικό παράθυρο. Αυτή η μέθοδος απορρίφθηκε διότι πελάτες που είχαν πολύ μικρά όρια εξυπηρέτησης εξυπηρετούνταν στις τελευταίες διαδρομές με αποτέλεσμα τη δημιουργία πολλών προβληματικών διαδρομών ενώ ταυτόχρονα διάλεγε πελάτες οι οποίοι είχαν πολύ μεγάλο χρονικό παράθυρο και θα μπορούσαν να εξυπηρετηθούν σε κάποια μεταγενέστερη χρονική στιγμή.

Παράλληλα δημιουργήθηκε η ιδέα επίλυσης του προβλήματος με τη χρήση άλλης μεθόδου. Αυτή η μέθοδος ήταν του πλησιέστερου γείτονα. Δηλαδή αντί για να δημιουργούνται λίστα τυχαίας επιλογής επόμενου πελάτη θα επιλεγόταν ο αμέσως κοντινότερος. Η μέθοδος έδωσε αρκετά βελτιωμένες αρχικές λύσεις όπως ήταν αναμενόμενο ως προς το κόστος αλλά και ως προς τον αριθμό των διαδρομών. Παρόλο που δεν ήταν η κύρια μέθοδος επίλυσης, η ανάγκη για επίτευξη καλύτερων αποτελεσμάτων και αξιολόγησης της μεθόδου grasp οδήγησε στη διατήρηση της μεθόδου.

Παρόλα αυτά η διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης έδινε αρκετά μεγάλο αριθμό διαδρομών και το συνολικό κόστος τους απείχε πολύ από το βέλτιστο. Ακόμα η μέθοδος του πλησιέστερου γείτονα παρόλο που ήταν πιο κοντά στο κόστος από ότι η μέθοδος grasp εμφάνιζε και αυτή απόκλιση ως προς τον αριθμό των οχημάτων. Και στις δυο περιπτώσεις εμφανιζόταν ένα σημαντικό ζήτημα και αυτό ήταν οι προβληματικές διαδρομές. Αυτές ήταν οι διαδρομές που αποτελούνταν από 2 έως 6 πελάτες. Η ανάγκη

απαλοιφής των διαδρομών αυτών και βελτίωσης των υπόλοιπων οδήγησε στη προσθήκη τριών τοπικών αναζητήσεων. Οι τοπικές αυτές αναζητήσεις ήταν οι 1-0 relocate , 2 opt και 1-1 exchange. Η λειτουργία των τοπικών αυτών αναζητήσεων έχει αναλυθεί εκτενέστερα στα παραπάνω κεφάλαια.

Όπως όμως υπήρχαν παραλλαγές στις μεθόδους με τις οποίες εξήχθησαν οι αρχικές λύσεις έτσι υπήρξαν παραλλαγές και στις τοπικές αναζητήσεις έτσι ώστε η λειτουργία τους να είναι όσο το δυνατόν πιο αποτελεσματική. Πιο συγκεκριμένα, αρχικά η μέθοδος 1-0 relocate δούλευε με βάση τη τυχαιότητα δηλαδή έκανε τυχαία απαλοιφή κόμβων από μια διαδρομή και τις επανατοποθετούσε τυχαία σε κάποια άλλη. Παρατηρήθηκε βελτίωση στις λύσεις όμως με αυτό το τρόπο λειτουργίας δεν εξαλείφθηκε το πρόβλημα των προβληματικών λύσεων. Στη συνέχεια δοκιμάστηκε η διαδικασία να δουλύνει πιο στοχευμένα δηλαδή οι κόμβοι να τοποθετούνται σε μια διαδρομή αυτόματα στη θέση που θα ικανοποιούσε το χρονικό τους παράθυρο. Η δοκιμή αυτή απέτυχε διότι ενώ τοποθετούνταν οι επιλεγμένοι κόμβοι χωρίς να παραβιάζουν τους περιορισμούς υπήρχαν πελάτες που στη συνέχεια της διαδρομής έβγαιναν εκτός χρονικών ορίων και δεν ήταν εφικτό να εξυπηρετηθούν. Έτσι καταλήξαμε στο βηματικό έλεγχο των διαδρομών. Πιο αναλυτικά παίρνοντας κόμβους από τις προβληματικές διαδρομές δοκιμάσαμε να τους τοποθετήσουμε σε οποιαδήποτε θέση στη διαδρομή τοποθετώντας πρώτα το κόμβο με θετική ζήτηση και στη συνέχεια γινόταν έλεγχος για τον κόμβο με αρνητική ζήτηση.

Όπως για την αναζήτηση 1-0 relocate υπήρξαν κάποιες παραλλαγές έτσι και για την τοπική αναζήτηση 2-opt δοκιμάστηκαν αρκετοί τρόποι λειτουργίας. Αρχικά αφού επιλεγθεί ένα ζευγάρι κόμβων δοκιμάστηκε η τυχαία τοποθέτηση τους μέσα στη διαδρομή μια στρατηγική που απορρίφθηκε αμέσως εφόσον μπορούσε να δώσει μια διαδρομή όπου να έπρεπε να εξυπηρετηθεί πρώτα ο κόμβος με αρνητική ζήτηση, δηλαδή ο κόμβος παραλαβής, και μετά ο κόμβος με θετική ζήτηση, ο κόμβος παράδοσης. Η επόμενη δοκιμή ήταν η τυχαία πάλι επιλογή ενός ζευγαριού σε μια διαδρομή και στη συνέχεια γινόταν τυχαία αναδιάταξη στους κόμβους που βρίσκονταν ανάμεσα στο επιλεγμένο ζευγάρι. Η δοκιμή αυτή εμφάνιζε συνεχή παραβίαση της χωρητικότητας αφού στις περισσότερες περιπτώσεις είχαμε τιμή μικρότερη του μηδενός. Τέλος αυτή η περίπτωση που δούλεψε και βελτίωσε και τις λύσεις ήταν αυτή όπου επιλεγόταν τυχαία ένας κόμβος και αναλόγως αν είχε θετική ή αρνητική ζήτηση ελέγχουμε όλες τις θέσεις αριστερά του ζευγαριού του ή δεξιά αντίστοιχα και τον τοποθετούσαμε.

Στη περίπτωση της τοπικής αναζήτησης 1-1 exchange αρχικά η επιλογή ζευγαριών και η ανταλλαγή τους από δυο διαφορετικές διαδρομές γινόταν τυχαία. Αυτό μείωνε αρκετά τη πιθανότητα να πραγματοποιηθεί κάποια αλλαγή για αυτό η αναζήτηση τροποποιήθηκε ώστε να γίνεται πιο στοχευμένα και να βελτιωθεί η πιθανότητα να υπάρξει εφικτή αλλαγή. Έτσι έχοντας επιλέξει τα ζευγάρια κόμβων γίνεται αναζήτηση στις διαδρομές έτσι ώστε να τοποθετηθούν σε θέσεις όπου θα δημιουργούνται συνδέσεις με το μικρότερο δυνατό κόστος.

Στη συνέχεια, σύμφωνα με τις παραπάνω δυο μεθόδους και με τις παραλλαγές των τοπικών αναζητήσεων στις οποίες καταλήξαμε θα περιγραφούν πιο αναλυτικά πως δούλεψαν ξεχωριστά τα παραδείγματα για τις διαφορετικές κατηγορίες χωρητικότητας.

### Μέθοδος GRASP :

- ❖ Στα παραδείγματα χωρητικότητας 200 οι διαδρομές που παίρνουμε από τις αρχικές λύσεις είναι σχεδόν οι διπλάσιες από αυτές στις οποίες καταλήγουμε. Αρχικά με την τοπική αναζήτηση 1-0 relocate εστιάζουμε στο να εξαλείψουμε τις προβληματικές διαδρομές και οι κόμβοι τους να τοποθετηθούν σε διαδρομές με μεγαλύτερο μήκος και μικρότερο κόστος. Μόλις επιτευχθεί αυτό απομένει ένα σύνολο διαδρομών του

οποίου το πλήθος είναι προφανώς μικρότερο από το αρχικό. Για περαιτέρω βελτίωση των λύσεων αυτό ως προς το κόστος χρησιμοποιούνται οι τοπικές αναζητήσεις 2-opt και 1-1 exchange. Το σχετικά μικρό μήκος των διαδρομών, εννοώντας μήκος διαδρομών έως 16 κόμβους, δεν δίνει πολλαπλές λύσεις για εσωτερική αναδιάταξη των διαδρομών και για το λόγο αυτό το πλήθος των επαναλήψεων που έτρεξαν οι τοπικές αναζητήσεις ήταν μικρότερο συγκριτικά με τα παραδείγματα χωρητικότητας 700 και 1000. Ο αριθμός των επαναλήψεων αυτών ήταν 1000. Η διάρκεια που τρέχει το πρόγραμμα είναι αρκετά μεγάλη λόγω του μεγάλου πλήθους αρχικών διαδρομών.

- ❖ Στα παραδείγματα χωρητικότητας 700 οι προβληματικές που δημιουργούνται από τις αρχικές λύσεις είναι συγκριτικά με τα παραδείγματα χωρητικότητας 200 λιγότερες. Παρόλα αυτά η αφαίρεση των μικρών σε μήκος διαδρομών θεωρείται αναγκαία αφού το κόστος τους ήταν πολύ μεγάλο. Το μήκος των διαδρομών που απομένουν είναι μεγάλο και καθιστά την αναδιαμόρφωση των διαδρομών εφικτή δίνοντας πολλές επιλογές για να μειώσουμε τα κόστη των επιμέρους διαδρομών. Αυτό επιτυγχάνεται με τις τοπικές αναζητήσεις 2-opt και 1-1 exchange.
- ❖ Στα παραδείγματα χωρητικότητας 1000 αντιμετωπίσαμε παρόμοιες δυσκολίες όπως και στα παραδείγματα με χωρητικότητα οχήματος 700. Η αφαίρεση των προβληματικών διαδρομών έγινε μεταφέροντας τους κόμβους σε διαδρομές μεγαλύτερου μήκους όπου υπήρχαν αρκετές πιθανές θέσεις για να τους τοποθετήσουμε. Το μεγάλο μήκος διαδρομών καθιστά δυνατό ώστε οι τοπικές αναζητήσεις 2-opt και 1-1 exchange να δουλέψουν ικανοποιητικά και σε μεγάλο αριθμό επαναλήψεων.

#### Μέθοδος του πλησίεστερου γείτονα :

- ❖ Στα παραδείγματα χωρητικότητας 200 τα οποία κατηγοριοποιούνται στα 1c , 1r και 1rc ο κώδικας δούλεψε διαφορετικά. Αυτό το συμπέρασμα εξήχθη διότι στα παραδείγματα 1c ο κώδικας έδωσε από τις αρχικές λύσεις αρκετά ικανοποιητικές λύσεις πολύ κοντά στις βέλτιστες και ως προς το συνολικό κόστος αλλά και ως προς το πλήθος των διαδρομών. Για το λόγο αυτό οι βελτιώσεις που πήραμε από τις τοπικές αναζητήσεις ήταν ελάχιστες. Κυρίως δούλεψαν οι τοπικές αναζητήσεις 2-opt και 1-1 exchange αφού δεν παρατηρήθηκε εμφάνιση προβληματικών διαδρομών. Επίσης ο αριθμός των επαναλήψεων που χρειάστηκαν ήταν αρκετά μικρός. Η διαδικασία ολοκληρώθηκε σε μικρό χρονικό διάστημα. Τα προβλήματα εμφανίστηκαν πιο έντονα στα παραδείγματα 1r και 1rc τα οποία εμφάνισαν αρχικές λύσεις περισσότερες από τις βέλτιστες. Καθοριστικό ρόλο στη μείωση των διαδρομών έπαιξε η τοπική αναζήτηση 1-0 relocate η οποία δουλεύοντας στοχευμένα κατάφερε να μειώσει το πλήθος τους. Η διαδικασία για να καταλήξουμε στις τελικές λύσεις ήταν χρονοβόρα.
- ❖ Στα παραδείγματα χωρητικότητας 700 ο κώδικας από τις αρχικές κιάλας λύσεις μας έδωσε λύσεις πολύ κοντά στις βέλτιστες έτσι όπως δίνονται στη βιβλιογραφία ως προς το κόστος αλλά και ως προς τον αριθμό των οχημάτων που χρειάζονται. Η χρήση των τοπικών αναζητήσεων ήταν μηδαμινή ενώ ο χρόνος στον οποίο καταλήξαμε στις αρχικές λύσεις ήταν πολύ γρήγορος.

- ❖ Ακόμα και στα παραδείγματα χωρητικότητας 1000 ο κώδικας έδωσε ελάχιστες προβληματικές διαδρομές από τις αρχικές λύσεις. Έντονη επίδραση στη διαμόρφωση του τελικού αποτελέσματος είχαν οι τοπικές αναζητήσεις 2-opt και 1-exchange οι οποίες έτρεξαν για 3000 επαναλήψεις λόγω των πολλών εναλλακτικών που δίνει το μεγάλο μήκος διαδρομών των παραδειγμάτων.

Συμπερασματικά ως πιο χρονοβόρα τοπική αναζήτηση θεωρείται η αναζήτηση 1-0 relocate. Ο κώδικας στα παραδείγματα όπου μπορούσε να δουλέψει η τοπική αυτή αναζήτηση καθυστέρουσε πολύ. Αυτό οφείλεται στο ότι έπρεπε να ελεγχθούν όλες οι πιθανές θέσεις στις οποίες μπορούσαν να τοποθετηθούν κόμβοι και στη συνέχεια να γίνει έλεγχος ώστε να τηρούνται οι περιορισμοί. Οι άλλες δύο τοπικές αναζητήσεις ανεξάρτητα από τον αριθμό των επαναλήψεων που έτρεχε ο κώδικας έβγαζαν πολύ γρήγορα τις αλλαγές.

Συνολικά, ως πιο έγκυρη μέθοδο θεωρείται η μέθοδος του πλησιέστερου γείτονα και αυτό γιατί όχι μόνο δίνει μικρότερο συνολικό κόστος αλλά και παρέχει τη δυνατότητα να χρησιμοποιηθούν λιγότερα οχήματα. Τα παραδείγματα στα οποία δούλεψε καλύτερα ο κώδικας είναι τα παραδείγματα lc201-lc208 χωρητικότητας 700 και τα παραδείγματα lc101-lc109 χωρητικότητας 200.

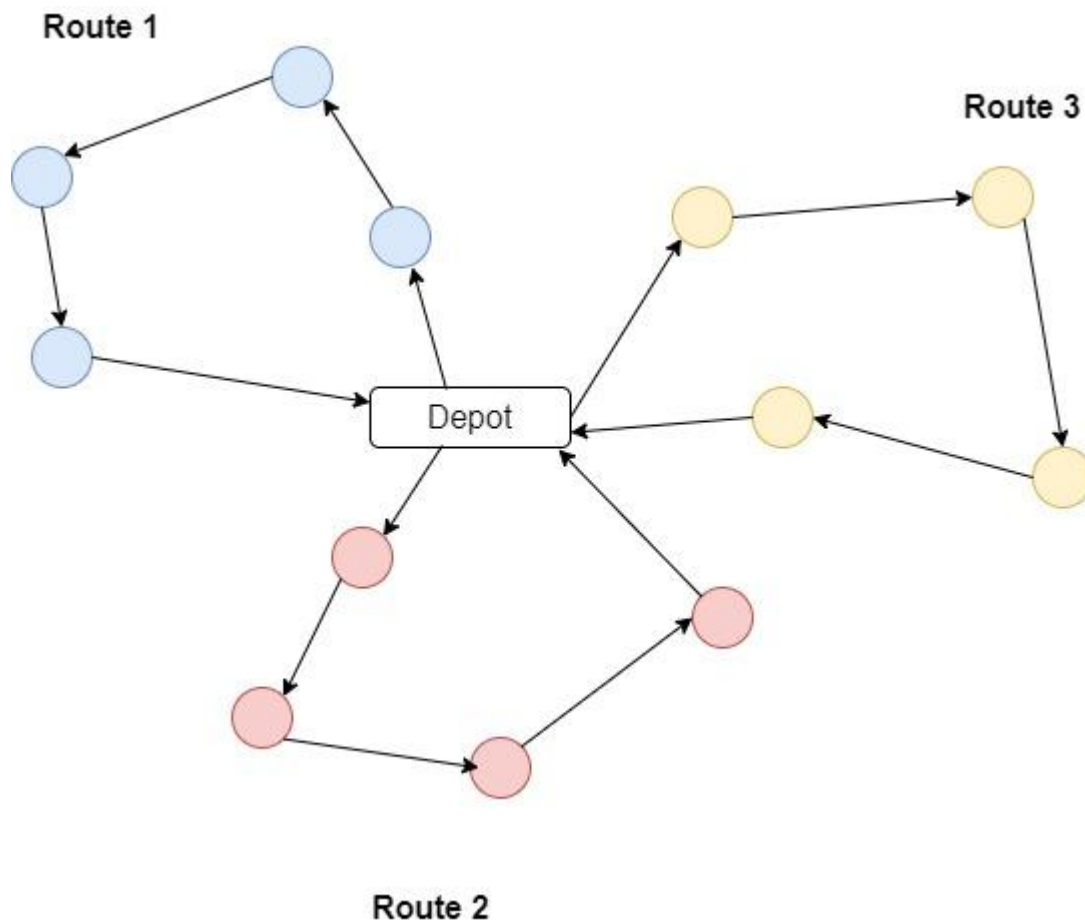
Παρόλο που χρειάστηκε να τρέξουμε τη μέθοδο του grasp αρκετές φορές ώστε να ελέγξουμε ότι θα εξάγουμε το καλύτερο δυνατό αποτέλεσμα, υπήρξαν φορές όπου πήραμε καλύτερο τελικό κόστος από ότι με τη μέθοδο του πλησιέστερου γείτονα και αυτό φαίνεται στους πίνακες αποτελεσμάτων.

Τα προβλήματα που δημιουργήθηκαν καθ' όλη τη διάρκεια εκπόνησης της εργασίας ήταν αρκετά αλλά μεθοδικά και βηματικά όπως περιγράφηκαν και παραπάνω αντιμετωπίστηκαν.

## 6.2 ΓΡΑΦΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΤΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Στη παράγραφο αυτή απεικονίζονται γραφικά τα αποτελέσματα των καλύτερων παραδειγμάτων και φαίνονται αναλυτικά οι διαδρομές οι οποίες εξήχθησαν από τον αλγόριθμο.

Το παρακάτω γράφημα αναπαριστά μια γενική μορφή διαδρομών οι οποίες ξεκινούν από την αποθήκη και καταλήγουν σε αυτή.



Σχήμα 5 : Γραφική αναπαράσταση διαδρομών



## Παράδειγμα lc106

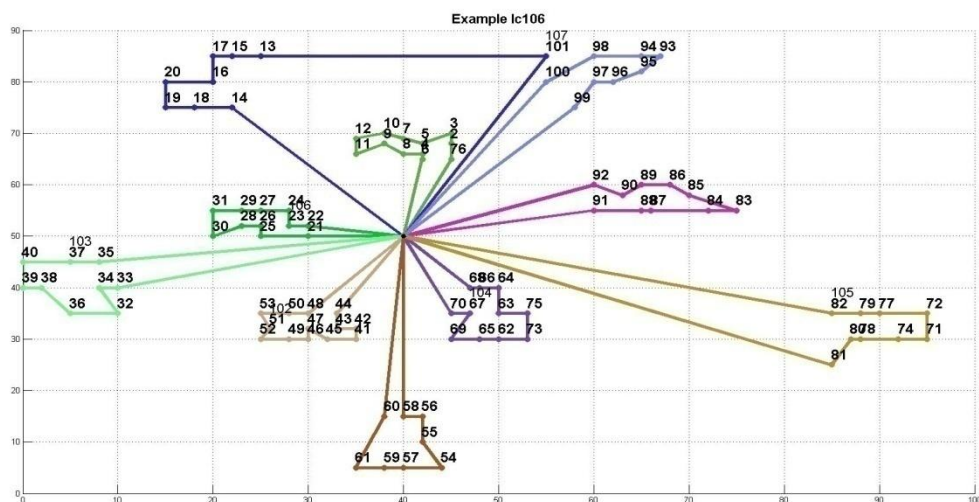
Τα χαρακτηριστικά του παραδείγματος είναι τα εξής:

Χωρητικότητα οχήματος : 200

Αριθμός πελατών : 106

Πλήθος διαδρομών : 10

Η λύση βρέθηκε με τη μέθοδο του πλησιέστερου γείτονα.



Σχήμα 6 : Γραφική αναπαράσταση παραδείγματος lc106

<b>Route 1</b>	1	21	25	26	28	30	31	29	27	24	23	106	22	1	0	0	<b>50,8</b>
<b>Route 2</b>	1	68	66	64	63	75	73	62	65	69	67	104	70	1	0	0	<b>59,4</b>
<b>Route 3</b>	1	6	4	8	9	11	12	10	7	5	3	2	76	1	0	0	<b>59,6</b>
<b>Route 4</b>	1	44	43	42	41	45	47	46	49	52	51	102	53	50	48	1	<b>64,8</b>
<b>Route 5</b>	1	91	88	87	84	83	85	86	89	90	92	1	0	0	0	0	<b>76</b>
<b>Route 6</b>	1	14	18	19	20	16	17	15	13	101	107	1	0	0	0	0	<b>125,8</b>
<b>Route 7</b>	1	99	97	96	95	93	94	98	100	1	0	0	0	0	0	0	<b>93</b>
<b>Route 8</b>	1	33	34	32	36	38	39	40	37	103	35	1	0	0	0	0	<b>97,2</b>
<b>Route 9</b>	1	58	56	55	54	57	59	61	60	1	0	0	0	0	0	0	<b>101,8</b>
<b>Route 10</b>	1	82	105	79	77	72	71	74	78	80	81	1	0	0	0	0	<b>127,2</b>
<b>Συνολικό Κόστος:856</b>																	

Πίνακας 5 : Διαδρομές παραδείγματος lc106



## Παράδειγμα lc107

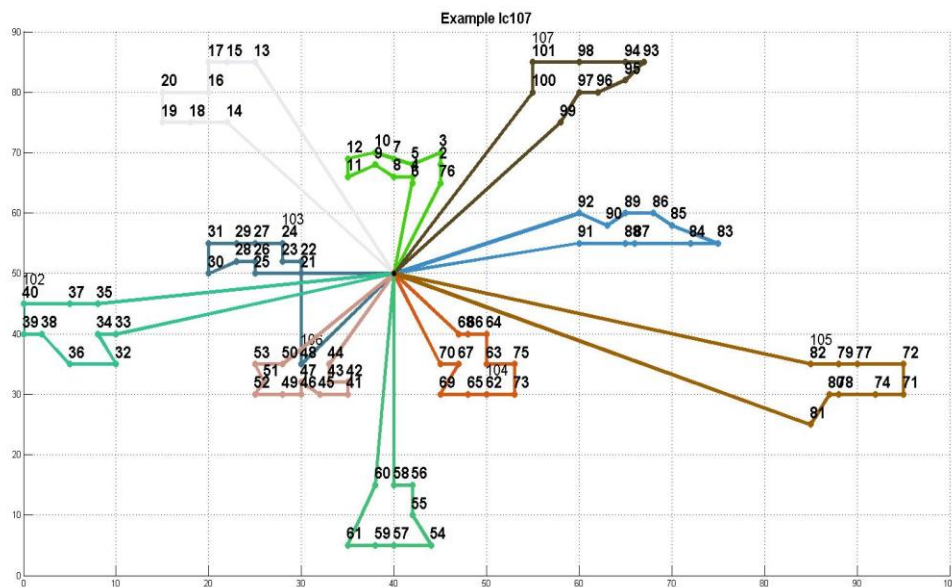
Τα χαρακτηριστικά του παραδείγματος είναι τα εξής:

Χωρητικότητα οχήματος : 200

Αριθμός πελατών : 106

Πλήθος διαδρομών : 10

Η λύση βρέθηκε με τη μέθοδο του πλησιέστερου γείτονα.



Σχήμα 7: Γραφική αναπαράσταση παραδείγματος lc107

<b>Route 1</b>	1	21	25	26	28	30	31	29	27	24	103	23	22	48	106	1	<b>75,6</b>
<b>Route 2</b>	1	68	66	64	63	75	73	62	104	65	69	67	70	1	0	0	<b>59,4</b>
<b>Route 3</b>	1	6	4	8	9	11	12	10	7	5	3	2	76	1	0	0	<b>59,6</b>
<b>Route 4</b>	1	44	43	42	41	45	47	46	49	52	51	53	50	1	0	0	<b>63,9</b>
<b>Route 5</b>	1	91	88	87	84	83	85	86	89	90	92	1	0	0	0	0	<b>76</b>
<b>Route 6</b>	1	14	18	19	20	16	17	15	13	1	0	0	0	0	0	0	<b>95,8</b>
<b>Route 7</b>	1	99	97	96	95	93	94	98	101	107	100	1	0	0	0	0	<b>95,9</b>
<b>Route 8</b>	1	33	34	32	36	38	39	40	102	37	35	1	0	0	0	0	<b>97,2</b>
<b>Route 9</b>	1	58	56	55	54	57	59	61	60	1	0	0	0	0	0	0	<b>101,8</b>
<b>Route 10</b>	1	82	105	79	77	72	71	74	78	80	81	1	0	0	0	0	<b>127,2</b>
Συνολικό Κόστος: 852,9																	

Πίνακας 8 : Διαδρομές παραδείγματος lc107

### Παράδειγμα lc109

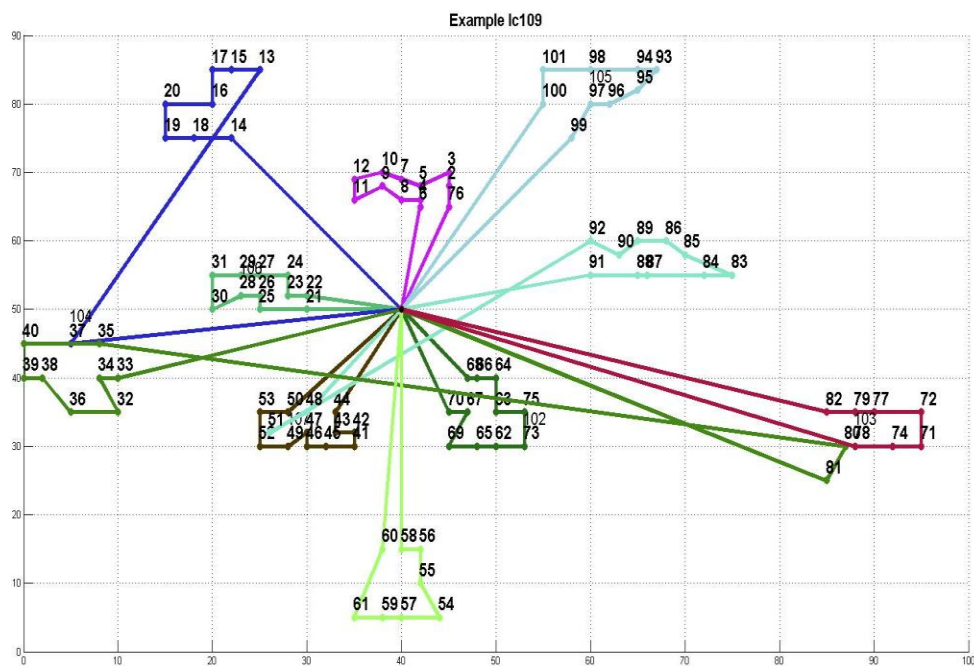
Τα χαρακτηριστικά του παραδείγματος είναι τα εξής:

Χωρητικότητα οχήματος : 200

Αριθμός πελατών : 106

Πλήθος διαδρομών : 10

Η λύση βρέθηκε με τη μέθοδο του πλησιέστερου γείτονα.



Σχήμα 8 : Γραφική αναπαράσταση παραδείγματος lc109

<b>Route 1</b>	1	21	25	26	28	106	30	31	29	27	24	23	22	1	<b>68,5</b>
<b>Route 2</b>	1	68	66	64	63	75	73	102	62	65	69	67	70	1	<b>81,4</b>
<b>Route 3</b>	1	6	4	8	9	11	12	10	7	5	3	2	76	1	<b>59,6</b>
<b>Route 4</b>	1	44	43	42	41	45	46	47	49	107	52	53	50	1	<b>65,5</b>
<b>Route 5</b>	1	91	88	87	84	83	85	86	89	90	92	51	48	1	<b>120,7</b>
<b>Route 6</b>	1	14	18	19	20	16	17	15	13	37	104	1	0	0	<b>137,8</b>
<b>Route 7</b>	1	99	97	105	96	95	93	94	98	101	100	1	0	0	<b>95,9</b>
<b>Route 8</b>	1	33	34	32	36	38	39	40	35	80	81	1	0	0	<b>202,1</b>
<b>Route 9</b>	1	58	56	55	54	57	59	61	60	1	0	0	0	0	<b>101,8</b>
<b>Route 10</b>	1	82	79	77	72	71	74	78	103	1	0	0	0	0	<b>121,4</b>
<b>Συνολικό Κόστος: 1055,2</b>															

Πίνακας 9 : Διαδρομές παραδείγματος lc109

## Παράδειγμα lc201

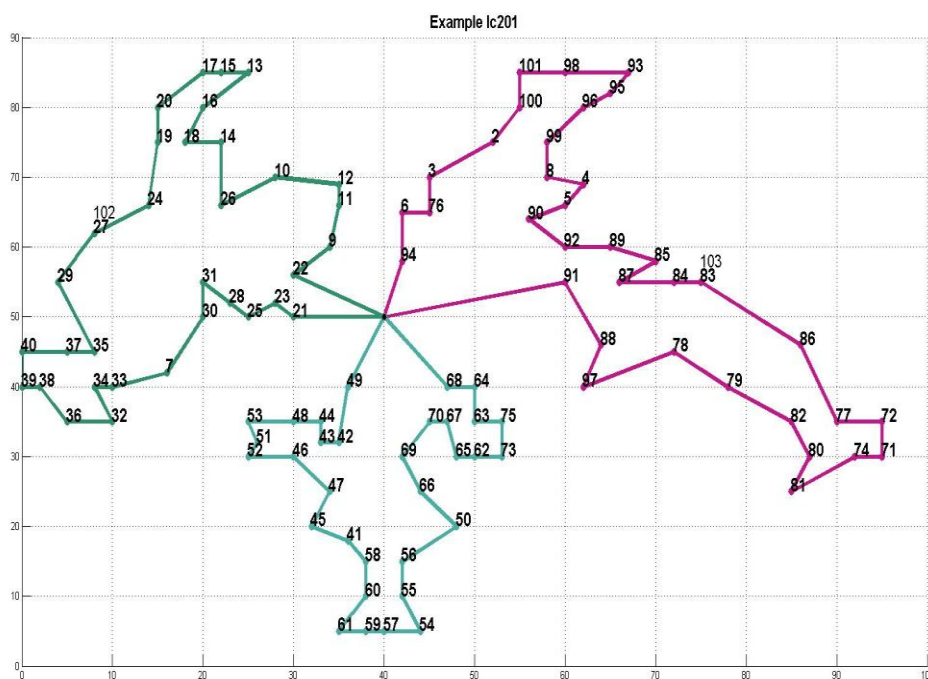
Τα χαρακτηριστικά του παραδείγματος είναι τα εξής:

Χωρητικότητα οχήματος : 700

Αριθμός πελατών : 102

Πλήθος διαδρομών : 3

Η λύση βρέθηκε με τη μέθοδο του πλησιέστερου γείτονα.



Σχήμα 9 : Γραφική αναπαράσταση παραδείγματος lc201

Route 1	1	94	6	76	3	2	100	101	98	93	95	96	99	8	4	5	90	92	89	85	87	238,2
	84	83	103	86	77	72	71	74	81	80	82	79	78	97	88	91	1					
Route 2	1	21	23	25	28	31	30	7	33	34	32	36	38	39	40	37	35	29	27	102	24	195,3
	19	20	17	15	13	16	18	14	26	10	12	11	9	22	1							
Route 3	1	68	64	63	75	73	62	65	67	70	69	66	50	56	55	54	57	59	61	60	58	158
	41	45	47	46	52	51	53	48	44	43	42	49	1									
Συνολικό Κόστος: 591,5																						

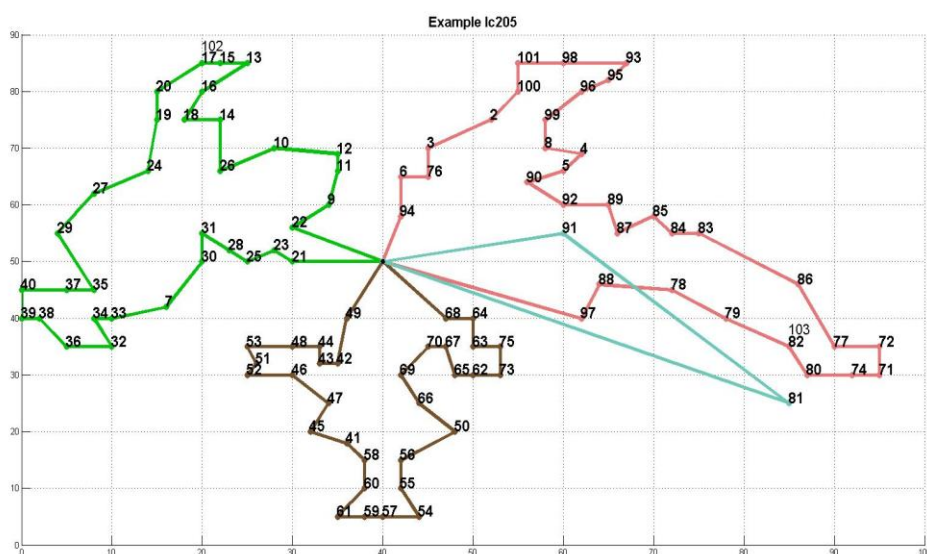
Πίνακας 10 : Διαδρομές παραδείγματος lc201

Τα χαρακτηριστικά του παραδείγματος είναι τα εξής:

Αριθμός πελατών : 102

Πλήθος διαδρομών : 4

Η λύση βρέθηκε με τη μέθοδο του πλησιέστερου γείτονα.



**Σχήμα 10 : Γραφική αναπαράσταση παραδείγματος Ic205**

[illegible]

### Πίνακας 11 : Διαδρομές παραδείγματος lc205

## Παράδειγμα lc206

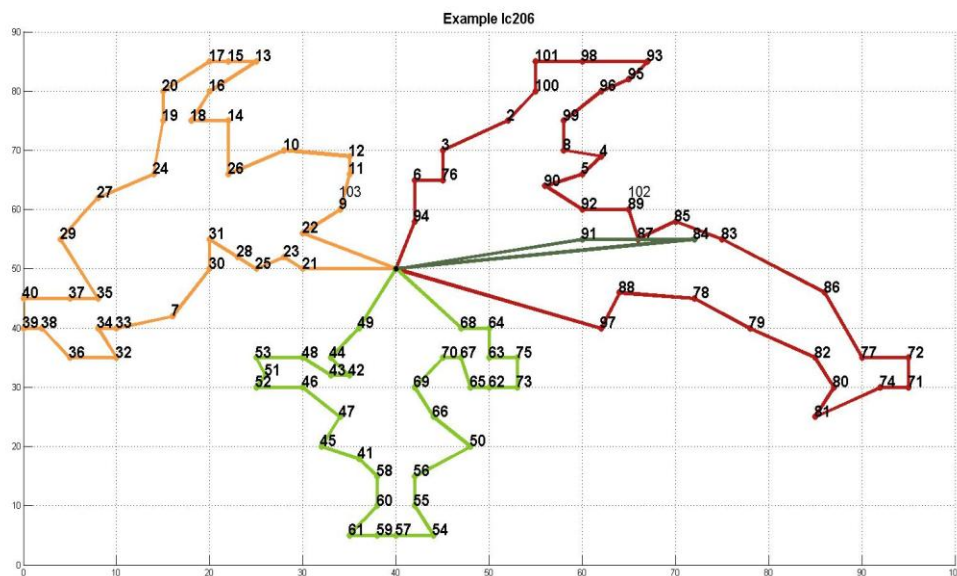
Τα χαρακτηριστικά του παραδείγματος είναι τα εξής:

Χωρητικότητα οχήματος : 700

Αριθμός πελατών : 102

Πλήθος διαδρομών : 4

Η λύση βρέθηκε με τη μέθοδο του πλησιέστερου γείτονα.



Σχήμα 11 : Γραφική αναπαράσταση παραδείγματος lc206

Route 1	1	94	6	76	3	2	100	101	98	93	95	96	99	8	4	5	90	92	89	102	87	226
	85	83	86	77	72	71	74	81	80	82	79	78	88	97	1							
Route 2	1	21	23	25	28	31	30	7	33	34	32	36	38	39	40	37	35	29	27	24	19	195,3
	20	17	15	13	16	18	14	26	10	12	11	9	103	22	1							
Route 3	1	68	64	63	75	73	62	65	67	70	69	66	50	56	55	54	57	59	61	60	58	157,6
	41	45	47	46	52	51	53	48	43	42	44	49	1									
Route 4	1	84	91	1																		65
Συνολικό Κόστος : 644																						

Πίνακας 12 : Διαδρομές παραδείγματος lc206

## ΒΙΒΛΙΟΓΡΑΦΙΑ

Ιωάννης Μαρινάκης & Αθανάσιος Μυγδαλάς (2008). Σχεδιασμός και Βελτιστοποίηση της Εφοδιαστικής Αλυσίδας, Θεσσαλονίκη : Εκδόσεις "σοφία" αε.

Ιωάννης Μαρινάκης & Αθανάσιος Μυγδαλάς (2016). Συνδιαστική Βελτιστοποίηση, Αθήνα : ΕΚΔΟΣΕΙΣ ΝΕΩΝ ΤΕΧΝΟΛΟΓΙΩΝ.

Μιχάλης Βιδάλης (2017), Εφοδιαστική (Logistics) Μια ποσοτική προσέγγιση, Αθήνα : Εκδόσεις Κλειδάριθμος.

Δρ. Μπινιώρης Σπύρος (2004), Εισαγωγή στη διαχείριση της εφοδιαστικής αλυσίδας, Cyprus : BROKEN HILL PUBLISHERS LTD

<http://cgi.di.uoa.gr/~ys02/dialekseis2013/local-search2spp.pdf>

[https://en.wikipedia.org/wiki/Vehicle\\_routing\\_problem](https://en.wikipedia.org/wiki/Vehicle_routing_problem)

<https://www.sintef.no/projectweb/top/pdptw/li-lim-benchmark/>