ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΠΡΟΒΛΗΜΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΜΕ ΠΕΡΙΟΡΙΣΜΕΝΗ ΧΩΡΗΤΙΚΟΤΗΤΑ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΑΣΙΑ

Μπούκοσης Δημήτριος

20/08/2017

Ευχαριστίες

Θέλω να ευχαριστήσω τον επιβλέποντα της διπλωματικής εργασίας μου, κ. Ιωάννη Μαρινάκη, για την πολύτιμη βοήθεια και καθοδήγησή του κατά τη διάρκεια της δουλειάς μου. Πάνω απ’όλα όμως είμαι ευγνώμων στους γονείς μου Ευάγγελο Μπούκοση και Ελένη Λουλάκη και τον αδερφό μου Μενέλαο Μπούκοση για την ολόψυχη αγάπη και υποστήριξή τους, ψυχολογική και οικονομική, μέσα σε αυτά τα χρόνια της κρίσης. Αφιερώνω αυτή την εργασία στην οικογένειά μου.

Δημήτριος Μπούκοσης

Περιεχόμενα

Εισαγωγή

Στη σημερινή εποχή, όπου ο παγκόσμιος ανταγωνισμός είναι έντονος, ο τομέας της διαχείρισης της εφοδιαστικής αλυσίδας θεωρείται ιδιαίτερα σημαντικός για τις επιχειρήσεις και τους διάφορους οργανισμούς. Η παρούσα διπλωματική εργασία πραγματεύεται ένα από τα σημαντικότερα προβλήματα της διαχείρισης της εφοδιαστικής αλυσίδας που αντιμετωπίζουν οι επιχειρήσεις, το πρόβλημα δρομολόγησης οχημάτων με περιορισμένη χωρητικότητα (Capacitated vehicle routing problem) και την απόπειρα επίλυσής του.

Το πρόβλημα δρομολόγησης οχημάτων (VRP) είναι ένα ενδιαφέρον και απαιτητικό πρόβλημα. Αφορά στη μεγάλης σημασίας ενοποίηση και συντονισμό δύο βασικών εννοιών της εφοδιαστικής αλυσίδας, της διαδικασίας διανομής των προϊόντων και του ελέγχου των αποθεμάτων. Πιο συγκεκριμένα, στα πλαίσια αυτής της εργασίας στο πρώτο κεφάλαιο θα γίνει αναφορά στην εφοδιαστική αλυσίδα γενικότερα, στη συνέχεια θα αναλυθεί πλήρως το πρόβλημα δρομολόγησης οχημάτων με περιορισμένη χωρητικότητα (CVRP) και θα σημειωθούν μέθοδοι και προσεγγίσεις επίλυσής του από τη διεθνή βιβλιογραφία. Τέλος, θα αναπτυχθεί αλγόριθμος σε matlab, με τον οποίο θα επεξεργαστούμε τα δεδομένα που διαθέτουμε, με σκοπό μία ικανοποιητική λύση στο πρόβλημα.

Κεφάλαιο 1: Εφοδιαστική Αλυσίδα

1.1 Διαχείρηση Εφοδιαστικής Αλυσίδας

Είναι αδιαμφισβήτητο γεγονός ότι τις τελευταίες δεκαετίες η ανάπτυξη της τεχνολογίας και των επιστημών είναι ραγδαία. Σε επίπεδο επιχειρήσεων κρίνεται πλέον ζωτικής σημασίας η χρήση νέων μέσων και διαδικασιών για τη βελτιστοποίηση της λειτουργίας τους και κατ'επέκταση τη μεγιστοποίηση του κέρδους ή ελαχιστοποίηση της ζημίας. Η αναζήτηση αυτή, νέων και πιο αποτελεσματικών στρατηγικών από τις εταιρείες, οφείλεται προφανώς στον έντονο ανταγωνισμό που επικρατεί εδώ και δεκαετίες σε εγχώριο, αλλά και παγκόσμιο επίπεδο.

Στη δεκαετία του 1980, οι εταιρείες ανακάλυψαν νέες κατασκευαστικές τεχνολογίες και στρατηγικές, οι οποίες τους επέτρεψαν να μειώσουν τα κόστη και να ανταγωνιστούν καλύτερα σε διαφορετικές αγορές. Μεγάλες ποσότητες πόρων διατέθηκαν για να εφαρμοστούν τέτοιες στρατηγικές. Τα τελευταία χρόνια, όμως, έχει γίνει πλέον ξεκάθαρο ότι οι εταιρείες έχουν μειώσει τα κατασκευαστικά κόστη τόσο όσο είναι πρακτικά εφικτό. Πολλές από αυτές τις εταιρείες ανακαλύπτουν ότι η αποτελεσματική διαχείριση της εφοδιαστικής αλυσίδας είναι το επόμενο βήμα που πρέπει να κάνουν για να αυξήσουν το κέρδος και το μερίδιο αγοράς τους.

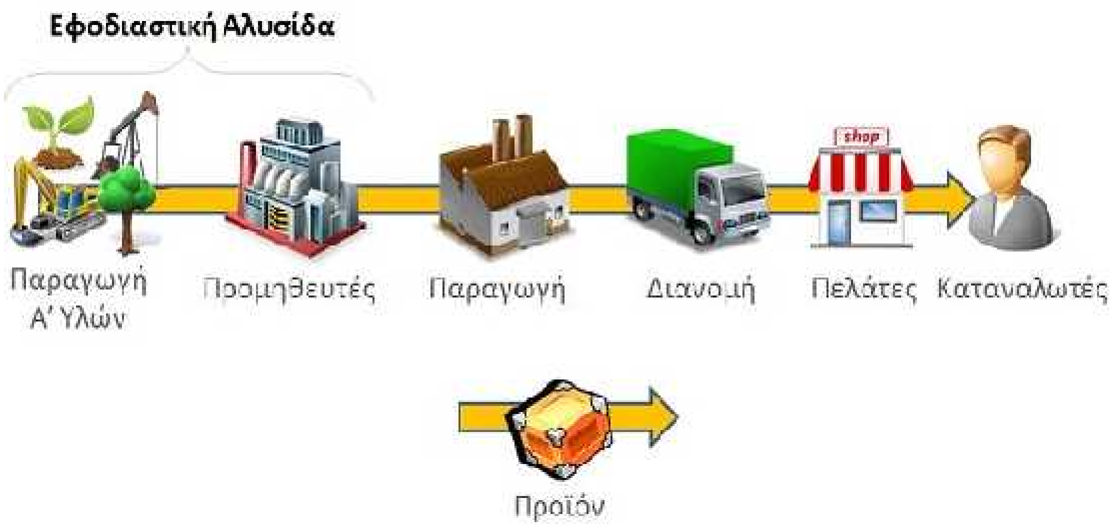
Με τον όρο **εφοδιαστική αλυσίδα** [1] εννοούμε όχι μόνο τη ροή υλικών από τον προμηθευτή πρώτων υλών ή τον κατασκευαστή μέχρι τον καταναλωτή, αλλά παράλληλα και τη ροή πληροφοριών μεταξύ των μελών της ίδιας αλυσίδας. Η διαχείρησή της γίνεται σε δύο επίπεδα:

- **Επίπεδο προγραμματισμού:** στο επίπεδο αυτό, αναλύονται τα δεδομένα προμηθειών, αναλώσεων παραγωγής, αποθεματοποίησης και πωλήσεων, γίνονται προβλέψεις και πλάνα πάνω στα οποία βασίζεται ο προγραμματισμός.
- **Επίπεδο εκτέλεσης:** στο στάδιο αυτό, εκτελείται το πλάνο που έχει καθορισθεί στο επίπεδο προγραμματισμού και παρακολουθείται η εξέλιξη του, βάσει των δεδομένων και πληροφοριών που συλλέγονται από όλο το εύρος της εφοδιαστικής αλυσίδας.

Η διαχείρηση της εφοδιαστικής αλυσίδας (Supply Chain Management) περιλαμβάνει το σχεδιασμό και τη διαχείρηση όλων των δραστηριοτήτων που εμπλέκονται στην προμήθευση, τη μετατροπή και τον έλεγχο της εφοδιαστικής αλυσίδας [2]. Περιλαμβάνει επίσης τις βασικές συνιστώσες του συντονισμού και της συνεργασίας με εταιρικά κανάλια, τα οποία μπορεί να είναι οι προμηθευτές, οι μεσάζοντες, οι τρίτοι πάροχοι υπηρεσιών και οι πελάτες. Επί της ουσίας, η διαχείρηση της εφοδιαστικής αλυσίδας ενσωματώνει την διαχείρηση της προσφοράς και της ζήτησης εντός και μεταξύ των εταιρειών. Πρωταρχική ευθύνη της είναι η σύνδεση των σημαντικών επιχειρηματικών λειτουργιών και των επιχειρηματικών διαδικασιών, σε ένα συνεκτικο και υψηλής απόδοσης επιχειρησιακό μοντέλο.

1.2 Εφοδιαστική και Εφοδιαστική Διαχείρηση

Παγκοσμίως χρησιμοποιείται ο όρος logistics, που είναι μία ελληνική λέξη και προέρχεται από τον όρο "λογιστική". Ο όρος χρησιμοποιήθηκε για πρώτη φορά κατά τους Βυζαντινούς χρόνους με την έννοια της μέριμνας για εξασφάλιση του στρατού με τρόφιμα, ρουχισμό, πολεμοφόδια και απαραίτητα υλικά. Είναι ξεκάθαρο ότι το αντικείμενο των logistics, όπως νοείται στην εποχή μας, είναι ευρύτατο.



Εικόνα 1: Logistics

Πρωτίστως, ο τομέας των logistics περιλαμβάνει τη διαχείριση της λήψης, αποθήκευσης, μετακίνησης και διανομής υλικού. Ειδικότερα, ο τομέας των μεταφορών κυριαρχείται σήμερα από τεχνολογίες που υποστηρίζουν τη λήψη αποφάσεων σε πραγματικό χρόνο. Η χρήση σύγχρονων υπολογιστικών εργαλείων επιτρέπει την ταχεία λήψη αποφάσεων που αφορούν ζητήματα εύρεσης εφαρμόσιμων δρομολογίων, βέλτιστων επιλογών, αξιόπιστης και ταχείας αξιολόγησης εναλλακτικών σχεδίων, καθώς και το γενικότερο σχεδιασμό υπηρεσιών σε δυναμικά επιχειρησιακά περιβάλλοντα.

Τα κόστη μεταφοράς κυμαίνονται μεταξύ του ενός τρίτου και των δύο τρίτων του συνολικού κόστους logistics [3]. Συνεπώς, είναι κομβικής σημασίας για κάθε εταιρεία η βελτίωση της απόδοσης, ώστε να επιτευχθεί η ορθή αξιοποίηση του εξοπλισμού διανομών και του προσωπικού. Οι διανομές αντιμετωπίζουν πολυσύνθετα προβλήματα, όπως ενδεικτικά :

- Υπολογισμός του βέλτιστου αριθμού σημείων εξυπηρέτησης και η τοποθέτηση αυτών, ώστε να ικανοποιηθούν οι ανάγκες των πελατών.
- Εύρεση του βέλτιστου αριθμού οχημάτων και των κατάλληλων διαδρομών.

Για παράδειγμα, η χρονική διάρκεια που απαιτείται για τη μεταφορά των αγαθών διαμορφώνει τα κόστη μεταφοράς, αφού καθορίζει το μέγεθος του απαιτούμενου στόλου οχημάτων, το απαραίτητο προσωπικό και τα κόστη κίνησης. Για να επιτευχθεί βέλτιστη εξυπηρέτηση των πελατών σε συνδυασμό με τα ελάχιστα δυνατά μεταφορικά κόστη είναι απαραίτητος ο εντοπισμός των διαδρομών εκείνων για τα οχήματα ώστε να μειωθεί ο χρόνος παράδοσης ή παραλαβής και η διανυόμενη απόσταση.

Γνωρίζουμε ότι για κάθε συγκεκριμένη εταιρεία υπάρχουν και διαφορετικές συνθήκες. Για παράδειγμα, μπορεί να υπάρχουν «Σταθερές παραδόσεις ή παραλαβές», δηλαδή να υπάρχει ζήτηση σταθερή εκ των προτέρων, εναλλακτικά μπορεί να έχουμε ζήτηση που δεν είναι γνωστή και οι παραγγελίες να δίνονται κατά τη διάρκεια επίσκεψης του οχήματος στον πελάτη. Επιπλέον, μπορεί να υπάρχουν απρόσμενα γεγονότα που αναγκάζουν μια εταιρεία να αλλάζει πλάνο διαδρομών. Άρα, απαιτούνται ρυθμίσεις σε πραγματικό χρόνο, όπως επανασχεδιασμός προγράμματος διαδρομών, για να προσαρμόζεται η δρομολόγηση στις νέες συνθήκες. Τέτοια γεγονότα μπορούν να περιλαμβάνουν συνθήκες κίνησης στους δρόμους, βλάβες οχημάτων, απρόβλεπτες απαιτήσεις πελατών και άλλα.

Από όλες τις παραμέτρους της δρομολόγησης, που θα εξεταστούν αναλυτικότερα και στα επόμενα κεφάλαια, γίνεται κατανοητό ότι δεν αρκεί ένα αποδοτικό αρχικό πλάνο για να έχουμε ένα επιτυχές αποτέλεσμα. Το αρχικό πλάνο δρομολόγησης θα πρέπει να είναι ευέλικτο, ώστε να παίρνουμε αποφάσεις σε πραγματικό χρόνο για την αντιμετώπιση απρόβλεπτων γεγονότων. Ενισχυτική στους ανωτέρω στόχους είναι και η αποτελεσματική χρήση της νέας τεχνολογίας.

1.3 Μεταφορά και Διανομές

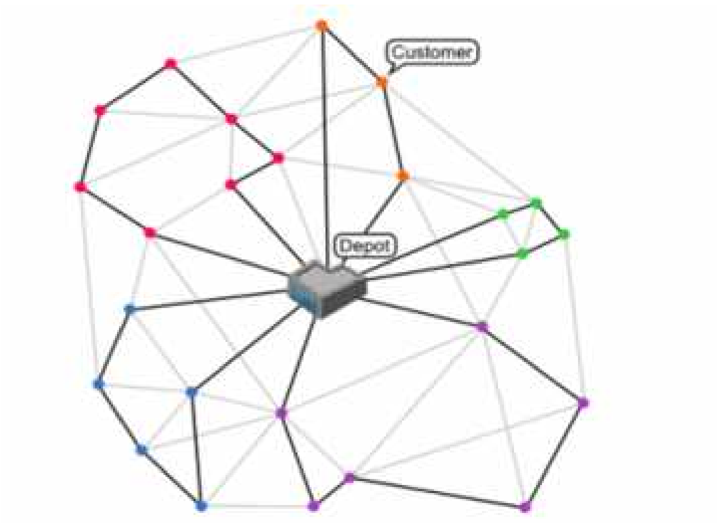
Η δραστηριότητα της μεταφοράς και διανομής αφορά τον τρόπο και τα μέσα για την μεταβίβαση υλικών, αγαθών και υπηρεσιών μέσω των φυσικών καναλιών. Οι μεταφορές και τα αποθέματα είναι κομβικής σημασίας για την εφοδιαστική αλυσίδα, καθώς αποτελούν τις πιο δαπανηρές δραστηριότητές της. Οι μεταφορές μόνο, συνήθως καταλαμβάνουν το μισό από το συνολικό κόστος της αλυσίδας, ενώ αποτελούν το σύνδεσμο μεταξύ της παραγωγής, της αποθήκευσης και της κατανάλωσης. Εξαιτίας της σημαντικότητάς τους, οι επιχειρήσεις δίνουν μεγάλη σημασία στα προβλήματα που σχετίζονται με μεταφορές, καθώς δεν πρέπει να υπάρχουν λάθη, για την ομαλή λειτουργία της αλυσίδας και τον περιορισμό του κόστους.

Οι μεταφορές εν γένει, χωρίζονται σε δύο μέρη:

1) Στις **εσωτερικές** μεταφορές, οι οποίες περιλαμβάνουν τόσο μεταφορά πρώτων υλών από τις πηγές προς τα εργοστάσια, όσο και μέρη των τελικών προϊόντων ανάμεσα σε διάφορα σημεία της εταιρείας ή από το εργοστάσιο στην αποθήκη και στα σημεία πωλήσεων (Inbound Logistics).
2) Στις **εξωτερικές** μεταφορές, που περιλαμβάνουν μεταφορά των τελικών προϊόντων από τις αποθήκες στους πελάτες, είτε άμεσα, είτε διαμέσου κέντρων διανομής (Outbound Logistics).

Προβλήματα στο σχεδιασμό των μεταφορών περιλαμβάνουν την επιλογή του στόλου μεταφοράς (μέγεθος στόλου, τύπος οχημάτων κλπ), τη δρομολόγηση οχημάτων (επιλογή βέλτιστων διαδρομών), το σχεδιασμό του δικτύου διανομής (δυναμικό σύστημα υπολογισμού των διαδρομών) και την επιλογή του προσωπικού που θα πραγματοποιεί τις διανομές.

Για την επίλυση των προβλημάτων που περιγράφηκαν παραπάνω, έχουν αναπτυχθεί διάφορες μεθοδολογίες και αλγόριθμοι βελτιστοποίησης. Παρακάτω θα δούμε τις κύριες κατηγορίες προβλημάτων δρομολόγησης.

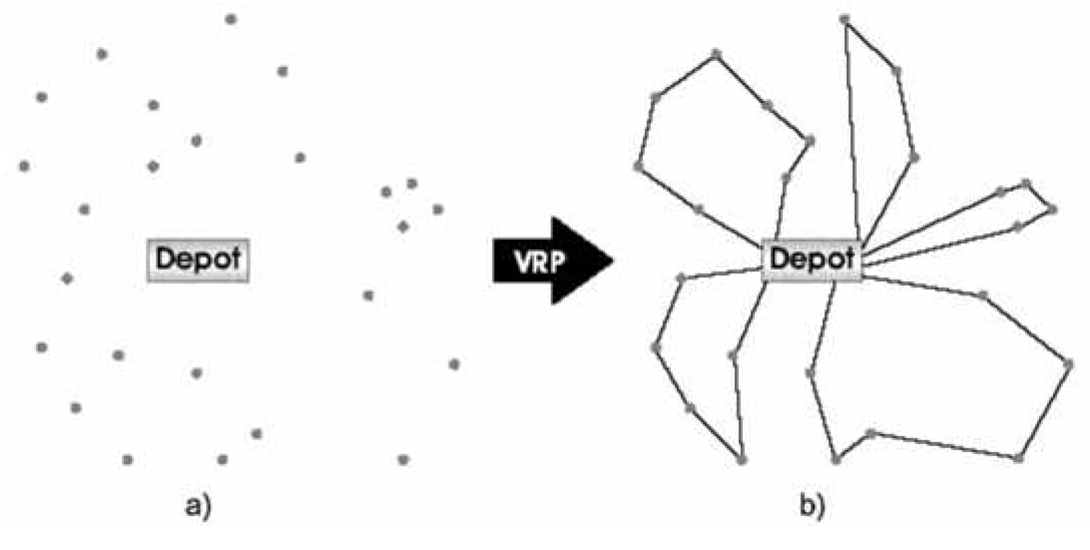


Σχήμα 1.1: Ένα δίκτυο μεταφορών

Κεφάλαιο 2: Δρομολόγηση Οχημάτων

Τα προβλήματα Δρομολόγησης Οχημάτων (Vehicle Routing Problem) είναι ένα πρόβλημα συνδυαστικής βελτιστοποίησης και ακέραιου προγραμματισμού που ρωτά «Ποιό είναι το βέλτιστο σύνολο των δρομολογιών, για ένα στόλο οχημάτων να διασχίσει, προκειμένου να παραδώσει προϊόντα σε ένα δεδομένο σύνολο πελατών;».

Η διανομή προϊόντων αφορά την εξυπηρέτηση, σε μια δεδομένη χρονική περίοδο ένα σύνολο από πελάτες από ένα σύνολο από οχήματα, που έχουν σαν αφετηρία τους μια συγκεκριμένη αποθήκη, χρησιμοποιούνται από ένα συγκεκριμένο αριθμό από οδηγούς, και πραγματοποιούν τις κινήσεις τους χρησιμοποιώντας ένα συγκεκριμένο οδικό δίκτυο. Γενικά, η σωστή επίλυση του προβλήματος δρομολόγησης οχημάτων έχει σαν αποτέλεσμα τον καθορισμό ενός συνόλου από διαδρομές, κάθε μια από αυτές ξεκινά και καταλήγει σε μια αποθήκη, ικανοποιώντας τις απαιτήσεις των πελατών, μη παραβιάζοντας κάποιον από τους περιορισμού και έχοντας ελαχιστοποιήσει το κόστος διανομής. Στο (Σχήμα 2.1) βλέπουμε πως από ένα σμήνος σημείων το πρόβλημα δρομολόγησης οχημάτων όρισε τις διαδρομές.



Σχήμα 2.1: Ένα απλό παράδειγμα VRP (Πηγή: neo.lcc.uma.es)

Για τον ορισμό του προς επίλυση προβλήματος ο ειδικός πρέπει να λάβει υπόψη τόσο τη φύση του προβλήματος διανομής, όσο και το μέγεθος της προς εξέταση εταιρίας. Ενδεικτικές πληροφορίες σχετικές με τις δραστηριότητες διανομής είναι οι ακόλουθες:

- Το μέγεθος του στόλου των οχημάτων που χρησιμοποιείται από την εταιρία.

- Ο αριθμός των οδηγών.
- Ο αριθμός των διαδρομών που πραγματοποιούνται καθημερινά και ο μέσος αριθμός στάσεων ανά διαδρομή.
- Οι διαδρομές εντός και εκτός πόλεως.
- Το συνολικό ετήσιο κόστος των δραστηριοτήτων διανομής.
- Το κόστος των πληρωμάτων.
- Οι μελλοντικές απαιτήσεις και προβλέψεις στον τομέα ενδεχόμενων βλαβών.
- Η τρέχουσα υπολογιστική δύναμη της εταιρίας για τη δυνατότητα υποστήριξης του δικτύου διανομής.
- Ο συνδυασμός δρομολογίων με άλλες δραστηριότητες.

Εκτός των πληροφοριών που προαναφέρθηκαν σχετικά με τα γενικά χαρακτηριστικά του συστήματος διανομής της εταιρίας, ο ειδικός πρέπει να λαμβάνει υπόψη του τα ακόλουθα χαρακτηριστικά των πελατών και των οχημάτων. Συγκεκριμένα, σχετικά με τα χαρακτηριστικά των πελατών:

- Το σημείο του γραφήματος, στο οποίο βρίσκεται ο πελάτης.
- Η ποσότητα των αγαθών (demand), που πρέπει να παραδοθούν είτε να συλλεχθούν από τον πελάτη.
- Τα χρονικά παράθυρα (time windows) κατά τη διάρκεια της ημέρας στις οποίες ο πελάτης μπορεί να εξυπηρετηθεί.
- Ο χρόνος που απαιτείται για την παράδοση ή τη συλλογή των προϊόντων από τον πελάτη (unloading or loading times), πιθανότατα εξαρτώμενος από το είδος του οχήματος.
- Το είδος του οχήματος που μπορεί να χρησιμοποιηθεί από το στόλο της εταιρίας για την εξυπηρέτηση κάποιου πελάτη.

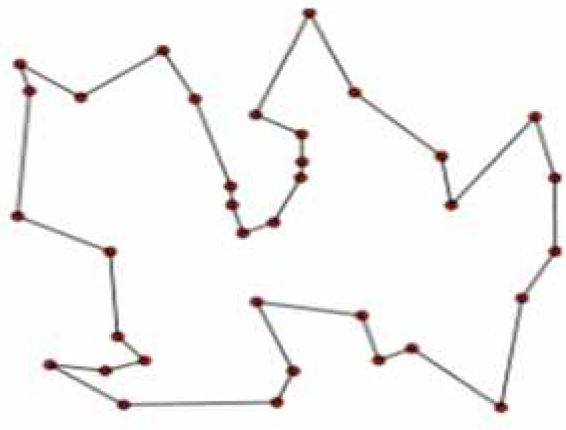
Μερικές φορές όμως, δεν είναι δυνατή η πλήρης ικανοποίηση των απαιτήσεων του κάθε πελάτη. Σε αυτές τις περιπτώσεις η ποσότητα των αγαθών που διανέμονται ή συλλέγονται μπορεί να μειωθεί ή ένα σύνολο πελατών να μην εξυπηρετηθεί. Για την αντιμετώπιση αυτών των καταστάσεων διάφορες προτεραιότητες που σχετίζονται με τη μερική ή την πλήρη έλλειψη εξυπηρέτησης μπορούν να ανατεθούν στους πελάτες.

Οι διαδρομές που εφαρμόζονται για να εξυπηρετήσουν κάποιους πελάτες ξεκινούν και καταλήγουν σε μια ή περισσότερες αποθήκες, οι οποίες αντιστοιχούν σε συγκεκριμένους κόμβους του δικτύου. Κάθε αποθήκη χαρακτηρίζεται από τον αριθμό και από το πλήθος των οχημάτων που βρίσκονται σε αυτή και από την συνολική ποσότητα προϊόντων που μπορούν να χειριστούν. Τυπικά χαρακτηριστικά των οχημάτων είναι τα ακόλουθα:

- Από ποια αποθήκη προέρχονται και αν υπαρχει πιθανότητα να τερματίσουν την διαδρομή τους σε άλλη αποθήκη από εκείνη που ξεκίνησαν.
- Η χωρητικότητα του οχήματος εκφρασμένη στο μέγιστο βάρος ή όγκο ή αριθμό παλετών που μπορεί να φορτωθεί στο όχημα.
- Αν τα οχήματα έχουν ένα τμήμα ή όχι και αν όχι πως θα φορτωθεί το κάθε τμήμα του οχήματος.
- Η πιθανή υποδιαίρεση των οχημάτων σε ομάδες, κάθε μια από τις οποίες χαρακτηρίζεται από τη χωρητικότητα και από το είδος των προϊόντων που μπορεί να μεταφέρει.
- Αν υπάρχουν διαθέσιμα μηχανήματα για την φόρτωση και την εκφόρτωση των οχημάτων.
- Το σύνολο των δρόμων που είναι προσπελάσιμοι από το όχημα.
- Το κόστος που συσχετίζεται με την λειτουργία του κάθε οχήματος.

2.1 Το πρόβλημα του Περιπλανόμενου Πωλητή (Travelling salesman problem)

Το πρόβλημα του πλανόδιου πωλητή (travelling salesman problem – TSP) αφορά την εύρεση της συντομότερης (σε χρόνο, απόσταση ή άλλο κόστος) διαδρομής για ένα όχημα (πωλητή) με αφετηρία κάποιο σημείο, π.χ. ένα κέντρο διανομής, και επιστροφή στο ίδιο σημείο αφού επισκεφθεί έναν σταθερό αριθμό πελατών ακριβώς μία φορά τον καθένα. Η αντίληψη που έχουμε γι'αυτήν την διαδρομή είναι επομένως αυτή του Σχήματος 2.2, ένας κύκλος που διέρχεται από τους κόμβους που αντιστοιχούν στο σημείο αφετηρίας και τους πελάτες ακριβώς μία φορά.

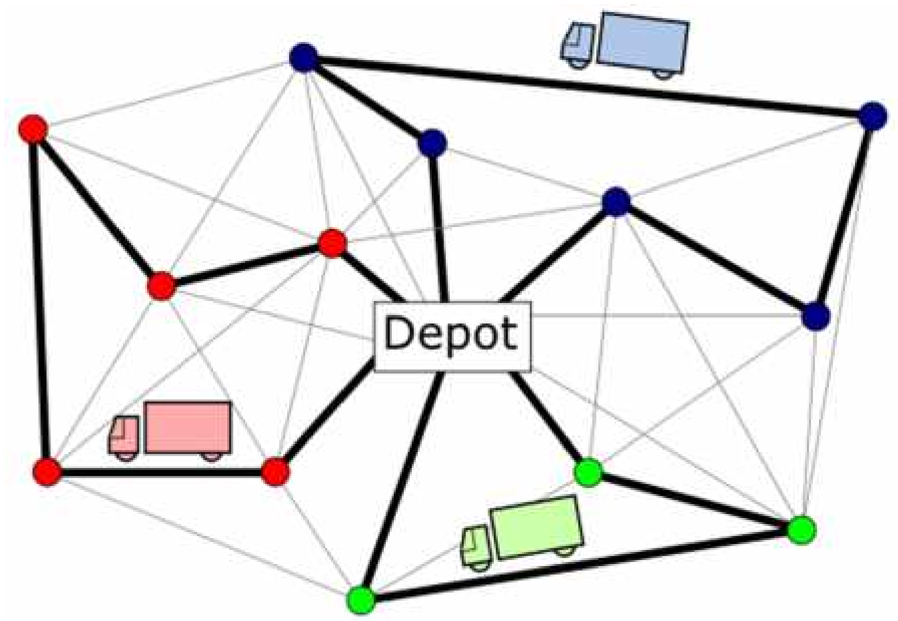


Σχήμα 2.2: Επίλυση του Πλανόδιου Πωλητή (Πηγή: en.wikipedia.org)

2.2 Το Περιορισμένης Χωρητικότητας πρόβλημα δρομολόγησης οχημάτων (Capacitated VRP)

Το περιορισμένης Χωρητικότητας πρόβλημα της δρομολόγησης οχημάτων (CVRP) είναι στην ουσία επέκταση του προβλήματος του πλανόδιου πωλητή στις περιπτώσεις που ένα μόνο όχημα δεν δύναται να επισκεφθεί όλους τους πελάτες. Σ'αυτήν την περίπτωση, αναζητούνται πολλαπλοί υπόκυκλοι (κυκλικές διαδρομές), όλοι με αρχή και τέλος το ίδο σημείο και όπου όλοι οι πελάτες καλύπτονται από το σύνολο των υπόκυκλων, χωρίς κάποιος πελάτης να καλύπτεται από περισσότερους του ενός υπόκυκλους.

Το CVRP είναι η πιο βασική εκδοχή του προβλήματος δρομολόγησης οχημάτων. Στο CVRP όλοι οι πελάτες δέχονται την παράδοση των προϊόντων, η ζήτησή τους είναι ντετερμινιστική και γνωστή εκ των προτέρων, ενώ δεν μπορεί να διασπαστεί. Τα οχήματα είναι ίδια, με το ίδιο επίπεδο χωρητικότητας και ξεκινάνε από την κεντρική αποθήκη και επιστρέφουν σε αυτή. Περιορισμό στο πρόβλημα αποτελεί το επίπεδο της ζήτησης σε κάθε όχημα. Στόχος είναι η ελαχιστοποίηση του κόστους για την εξυπηρέτηση των πελατών [4].



Σχήμα 2.3: Ένα παράδειγμα CVRP (Πηγή: www.cs.uni-dortmund.de)

Μπορούμε να μοντελοποιήσουμε το CVRP ως εξής:

Έχουμε n τοποθεσίες και i πελάτες, όπου i=1 είναι η κεντρική αποθήκη ανεφοδιασμού και i=2,3,4,...,n είναι οι πελάτες. Με q(i) συμβολίζεται η ζήτηση προϊόντων του κάθε πελάτη, ενώ με cij το κόστος της διαδρομής από τον πελάτη i στον πελάτη j. Τέλος διαθέτουμε K οχήματα με χωρητικότητα Q.

$$\min \sum_{i\in V}\sum_{j\in V} c_{ij} \sum_{K=1}^{K} x_{ijk} \tag{2.2.1}$$

Υ.π:

$$\sum_{©=1}^{\mathrm{K}} y_{ik} = 1, \ \vee i \in V \setminus \{0\}, \tag{2.2.2}$$

$$\sum_{k=1}^{K} y_{0k} = K \tag{2.2.3}$$

$$\sum_{j\in V} x_{ijk} = \sum_{j\in V} x_{jik} = y_{ik} \ , \quad \forall i \in V, k = 1,\ldots, K \tag{2.2.4}$$

$$\sum_{k=1}^{K} x_{0jk} = \sum_{k=1}^{K} x_{b0k}, \quad \forall j, b \in V \tag{2.2.5}$$

$$\sum_{i\in V} d_i y_{ik} \le C \ , \quad \forall k = 1,\ldots, K \tag{2.2.6}$$

$$\sum_{t\in S}\sum_{j\in S} c_{ij} x_{ijk} = D \ , \quad \begin{cases} S \subseteq V \\ k = 1,\ldots, K \end{cases} \tag{2.2.7}$$

$$\sum_{i\in S}\sum_{j\notin S} x_{ijk} \ge y_{hk} \ , \quad \begin{cases} \forall S \subseteq V \setminus \{0\}, h \in S \\ k = 1,\ldots, K \end{cases} \tag{2.2.8}$$

$$y_{ik} \in \{0,1\}, \forall i \in V, k = 1,\ldots, K \tag{2.2.9}$$

$$x_{ik} \in \{0,1\}, \forall i, j \in V, k = 1,\ldots, K \tag{2.2.10}$$

Όπου:

x_{ij} : Η μεταβλητή x_{ij} παίρνει την τιμή 0 αν το τόξο {i,j} δεν ανήκει στην βέλτιστη λύση και την τιμή 1 αν το τόξο ανήκει στ βέλτιστη διαδρομή.

X_{0j}: Η μεταβλητή αυτή παίρνει την τιμή 0 αν ντο τόξο {0,j} δεν ανήκει στη βέλτιστη διαδρομή, την τιμή 1 άν το τόξο ανήκει στη βέλτιστη διαδρομή και την τιμή 2 εφόσον το όχημα πραγματοποιεί διαδρομή για την εξυπηρέτηση ενός μόνο πελάτη.

c_{ij}: Το κόστος για να διασχίσει το όχημα το τόξο i,j.

V: Το σύνολο των κόμβων.

S: Ένα υποσύνολο κόμβων.

r(S): Ο ελάχιστος αριθμός οχημάτων που απαιτούνται για την εξυπηρέτηση του υποσυνόλου των κόμβων S.

K: Το σύνολο των οχημάτων.

d_i: Η ποσότητα των προϊόντων που απαιτεί ο κάθε πελάτης.

Η σχέση (2.2.1) είναι η προς ελαχιστοποίηση συνάρτηση, η οποία ανάλογα με το τι εκφράζει η μεταβλητή c_{ij} ελαχιστοποιεί τον χρόνο ή την απόσταση που απαιτούνται για την εξυπηρέτηση όλων των πελατών.

Όσον αφορά τους περιορισμούς, η σχέση (2.3.2) φροντίζει ώστε ο κάθε πελάτης να επισκέπτεται μόνο μία φορά. Η σχέση (2.3.3) εξασφαλίζει ότι και τα K διαθέσιμα οχήματα χρησιμοποιούνται για την επίλυση του προβλήματος. Η σχέση (2.3.4) υποχρεώνει το όχημα που πηγαίνει σε έναν πελάτη για να τον εξυπηρετήσει, να φεύγει από αυτόν. Η (2.3.5) εξασφαλίζει ότι κάθε όχημα k που φεύγει από την αποθήκη (κόμβος 0), προς τον πελάτη στον κόμβο j, θα επιστρέψει και πάλι στην αποθήκη i μετά τον τελευταίο πελάτη που θα εξυπηρετήσει, εάν αυτός είναι ο πελάτης b. Ο περιορισμός (2.3.6) είναι περιορισμός που αφορά την χωρητικότητα του οχήματος, ενώ ο (2.3.7) θέτει τα χιλιομετρικά ή χρονικά όρια στην απόσταση που μπορεί να διανύσει ένα όχημα κατά την διάρκει μια διαδρομής. Τέλος, η (2.3.8) εξασφαλίζει την συνέχεια της διαδρομής του οχήματος k.

2.3 Το Πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα (Time Windows VRP)

Το πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα μπορεί να καθοριστεί ως ακολούθως: έχουμε ένα σύνολο από πελάτες που είναι διεσπαρμένοι σε μία γεωγραφική περιοχή και πρέπει να εξυπηρετηθούν από ένα πλήθος οχημάτων που είναι αρχικά τοποθετημένα σε μία δεδομένη απόθηκη. Κάθε πελάτης έχει ένα φορτίο που πρέπει να περισυλλέγει και ο πελάτης καθορίζει μια χρονική περίοδο (time window) 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.

Ο σκοπός είναι να βρεθεί ένα σύνολο από διαδρομές για τα οχήματα, όπου κάθε διαδρομή ξεκινάει και τελειώνει στην αποθήκη, εξυπηρετεί ένα υπερσύνολο από τους πελάτες χωρίς να παραβιάζεται η χωρητικότητα και οι περιορισμοί για τα χρονικά παράθυρα, καθώ ελαχιστοποιείται το συνολικό μήκος των διαδρομών.

Στην ουσία, δηλαδή, το πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα είναι μία επέκταση του προβλήματος περιορισμένης χωρητικότητας στην οποία οι περιορισμοί χωρητικότητας ισχύουν με τον ίδιο τρόπο που ίσχυαν και προηγούμενα αλλά επιπλέον ο κάθε πελάτης πρέπει να εξυπηρετηθεί μέσα σε μία χρονική περίοδο [α_i, b_i], η οποία ονομάζεται χρονικό παράθυρο. Επιπλέον, δίνονται δεδομένα για τον χρόνο ταξιδιού t_{ij}, για κάθε τόξο (i,j) και φυσικά ο χρόνος εξυπηρέτησης για τον κάθε πελάτη. Η εξυπηρέτηση για κάθε πελάτη πρέπει να ξεκινήσε μέσα στο χρονικό παράθυρο που μπορεί ο πελάτης να εξυπηρετηθεί, και το όχημα πρέπει να παραμείνει στην τοποθεσία που βρίσκεται ο πελάτης για χρόνο s_i.

Επιπλέον, αν κάποιο όχημα φτάσει σε κάποιον πελάτη νωρίτερα από τον προκαθορισμένο χρόνο, στις περισσότερες περιπτώσεις το όχημα επιτρέπεται να παραμείνει στην τοποθεσία του πελάτη μέχρι να ξεκινήσει το χρόνικό παράθυρο.

Στις περισσότερες φορές ο πίνακας κόστους και ταξιδιού συμπίπτουν, και τα χρονικά παράθυρα καθορίζονται με βάση το γεγονός ότι όλα τα οχήματα φεύγουν από την αποθήκη τη χρονική στιγμή 0. Επιπλέον, τα χρονικά παράθυρα απαιτούν ένα πλήρη προσανατολισμό της κάθε διαδρομής ακόμα και αν οι αρχικοί πίνακες είναι συμμετρικοί. Άρα στις περισσότερες φορές το πρόβλημα προτυποποιείται σαν μη συμμετρικό.

Υπάρχουν δύο ειδών χρονικά παράθυρα, τα χαλαρά κατά τα οποία αν ένα όχημα φτάσει σε κάποιο πελάτη κάποια χρονική στιγμή εκτός του χρονικού παραθύρου μπορεί να ξεκινήσει την εξυπηρέτηση του εκείνη τη στιγμή, εν αντιθέση με τα σκληρά χρονικά παράθυρα που η εξυπηρέτηση του πελάτη πρέπει να γίνει αυστηρά εντός του χρονικού παραθύρου. Σε αυτές τις περιπτώσεις αν ένα όχημα φτάσει νωρίτερα στον πελάτη, θα περιμένει για να αρχίσει την εξυπηρέτηση.

2.4 Ύπαρξη Πολλαπλών Αποθηκών (Multidepot VRP)

Μια πολύ ενδιαφέρουσα παραλλαγή του προβλήματος δρομολόγησης οχημάτων είναι η παραλλαγή που χρησιμοποιεί παραπάνω από μία αποθήκες. Υπάρχουν δύο τρόποι για την επίλυση αυτού του προβλήματος:

1. Η κάθε μία από τις αποθήκες να έχει το δικό της αριθμό οχημάτων και τους δικούς της πελάτες να εξυπηρετήσει. Σε τέτοιες περιπτώσεις η εταιρία έχει να αντιμετωπίσει στην ουσία έναν αριθμό από απλά VRPs.
2. Στην δεύτερη περίπτωση ένα όχημα ξεκινάει από μια αποθήκη και, είτε τερματίζει σε μια άλλη, είτε ενδιάμεσα σταματάει σε κάποια άλλη αποθήκη για να φορτώσει, για παράδειγμα, επιπλέον προϊόντα και στη συνέχεια τερματίζει την διαδρομή.

Το συγκεκριμένο πρόβλημα μπορεί να θεωρηθεί σαν πρόβλημα ομαδοποίησης αφού ο στόχος είναι να βρεθούν οι διαδρομές των οχημάτων που ανήκουν σε κάθε μία αποθήκη. Το πρόβλημα μπορεί να επιλυθεί σαν πρόβλημα δύο φάσεων, στην πρώτη φάση έχουμε ανάθεση των πελατών στις αποθήκες και στη δεύτερη φάση δημιουργούνται τα δρομολόγια για κάθε μία αποθήκη και για κάθε όχημα.

2.5.Πρόβλημα Δρομολόγησης Οχημάτων με Διανομή και παραλαβή προϊόντων (Pick-up and Delivery VRP)

Το πρόβλημα περιλαμβάνει διανομές και παραλαβές από τους πελάτες κατά την διάρκεια μιας διαδρομής. Σε αυτό το πρόβλημα ο πελάτης μπορεί να θέλει να του διανεμηθούν προϊόντα από το όχημα που θα περάσει, αλλά και το όχημα να παραλάβει από αυτόν κάποια προϊόντα. Στη βασική μορφή του προβλήματος, κάθε πελάτης συσχετίζεται με δύο ποσότητες d_i και p_i που αντιπροσωπεύουν των την ζήτηση των προϊόντων που πρέπει να διανεμηθούν και να παραληφθούν από τον πελάτη i, αντίστοιχα. Πολλές φορές μόνο μία ποσότητα $d_i = d_i – p_i$ χρησιμοποιείται για τον κάθε πελάτη, που δείχνει την διαφορά ανάμεσα στις δύο ποσότητες (πολλές φορές αυτό μπορεί να οδηγήσει σε αρνητικό αριθμό). 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. Πολλές φορές αυτοί οι δύο κόμβοι αναφέρονται στον ίδιο κόμβο όπου είναι η κεντρική αποθήκη όπως την περιγράψαμε στο πρόβλημα δρομολόγησης οχημάτων περιορισμένης χωρητικότητας.

Ένας πολύ βασικό περιορισμός είναι ότι σε κάθε πελάτη η διανομή των προϊόντων γίνεται πριν από την παραλαβή, έτσι η συνολική φόρτωση ενός οχήματος όταν φτάσει στον πελάτη i είναι ίση με την αρχική φόρτωση του οχήματων 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.

Τα κύρια χαρακτηριστικά του προβλήματος είναι τα ακόλουθα:

- Κάθε κύκλος περνάει από την αποθήκη.
- Κάθε πελάτης επισκέπτεται από ένα μόνο κύκλο.
- Κάθε όχημα αντιστοιχεί μόνο σε μία διαδρομή.
- Η συνολική ποσότητα που μεταφέρει ένα όχημα πρέπει να είναι μη αρνητική και να μην ξεπερνάει τη 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 πριν από τον πελάτη i.

2.6 Άλλα προβλήματα Δρομολόγησης Οχημάτων

- **Πρόβλημα δρομολόγησης Οχημάτων χωρίς την επιστροφη στην αποθήκη (Open VRP)**

Σε όλα τα προηγούμενα προβλήματα μετά την εκτέλεση των δρομολογίων το όχημα έπρεπε να επιστρέψει στην αποθήκη εντός ενός χρονικού περιθωρίου. Σε αυτή την κατηγορία προβλημάτων, το όχημα δεν επιστρέφει στην αποθήκη, αλλά συνεχίζει να εκτελεί δρομολόγια μέχρι να αδειάσει το φορτίο του. Αυτή η κατηγορία προβλημάτων συναντάται στην περίπτωση που ο στόλος των φορτηγών δεν ανήκουν στην ιδιοκτησία της επιχείρησης, αλλά συνεργάζεται με μεσάζοντες για την εκτέλεση των δρομολογίων. Έτσι, αντί το φορτηγό να ξοδεύει χρόνο για να γυρίσει στην αποθήκη, τον αξιοποιεί για να εξυπηρετήσει περισσότερους πελάτες. Στόχος είναι η μείωση της συνολικής απόστασης που θα διανυθεί, εξυπηρετώντας όλους τους πελάτες.

- **Στοχαστικά προβλήματα δρομολόγησης οχημάτων (Stochastic or Probabilistic Vehicle routing problems)**

Σε αυτά τα προβλήματα ένα όχημα πεπερασμένης χωρητικότητας φεύγει από την αποθήκη και πρέπει να παραδώσει ένα μέρος του φορτίου σε ένα συγκεκριμένο αριθμό πελατών, των οποίως η ζήτηση γίνεται γνωστή τη χρονική στιγμή που θα φτάσει στον πελάτη. Το όχημα εκτελεί μία διαδρομή και μπορεί να ανεφοδιαστεί όταν χρειάζεται, με τα σημεία στα οποία πραγματοποιούνται οι επιστροφές να είναι στοχαστικά. Το όχημα πρέπει να εξυπηρετήσει όλους τους πελάτες διανύοντας την όσο το δυνατόν μικρότερη απόσταση. Η ζήτηση είναι στοχαστική μεταβλητή, με την κατανομή της όμως να είναι γνωστή.

- **Προβλήματα δρομολόγησης με δυναμική ζήτηση (Dynamic VRPs)**

Η διαφορά αυτής της κατηγορίας προβλημάτων έγκειται στο γεγονός ότι στα προηγούμενα ήταν γνωστή η γεωγραφική θέση όλων των πελατών από την αρχή, πριν ξεκινήσει η εκτέλεση κάποιου δρομοογίου με τα δεδομένα να παραμένουν ως έχουν γνωστα. Σε αντίθεση, στα προβλήματα δυναμική ζήτησης μπορεί να προκύψει κάποιος καινούριος πελάτης κατά τη διάρκεια εκτέλεσης ενός δρομολογίου. Αυτή η κατηγορία προβλημάτων συναντάται κυρίως σε εργασίες που σχετίζοντα με την παροχή υπηρεσιών, σε δουλειές που ο οδηγός λαμβάνει εντολές σε πραγματικό χρόνο, όπως αυτή του ταξιτζή ή του οδηγού ασθενοφόρου [5].

Κεφάλαιο 3: Αλγόριθμοι βελτιστοποίησης προβλημάτων εφοδιαστικής αλυσίδας

3.1 Απλοί Ευρετικοί αλγόριθμοι (Heuristics)

Στις μέρες μας, όσο αυξάνει το μέγεθος των προβλημάτων που προσπαθούμε να επιλύσουμε και η πολυπλοκότητά του, η επίλυση ενός προβλήματος βελτιστοποίησης και ιδιαίτερα συνδυαστικής βελτιστοποίησης γίνεται ολοένα και δυσκολότερη, συγκεκριμένα η αναζήτηση της ολικά βέλτιστης λύσης σε λογικό χρόνο έχει γίνει πρακτικά αδύνατη. Για να επιλυθούν προβλήματα αυτής της μορφής πολύ συχνά καταφεύγουμε σε διαφορετικές τεχνικές που μας οδηγούν σε μία σχεδόν βέλτιστη, συνάμα ικανοποιητική λύση. Μία λύση ενός ευρετικού αλγορίθμου γίνεται αποδεκτή όταν κανοποιεί τα κριτήρια της ποιότητας της λύσης και της ευκολίας απόκτησής της.

Για όλα τα προβλήματα βελτιστοποίησης δεν υπάρχει μονάχα ένας αλγόριθμος που δίνει τη βέλτιστη λύση, αλλά έχουν αναπτυχθεί αρκετοί αλγόριθμοι, οι οποίοι συγκρινόμενοι μεταξύ του, οδηγούν ολοένα σε καλύτερες λύσεις. Σε ότι αφορά την ποιότητα των λύσεω,ν, σε μερικά προβλήματα είναι αδύνατο να βρεθεί η βέλτιστη λύση για κάποιο πρόβλημα σε ικανοποιητικό χρόνο.

Οι αλγόριθμοι χωρίζονται σε διάφορες κατηγορίες:

- Αλγόριθμοι απληστίας (greedy algorithms).

  Οι αλγόριθμοι απληστίας προσπαθούν να καταλήξουν σε μία εφικτή λύση στο πρόβλημα, αλλά πολλές φορές χρειάζονται πολύ μεγάλο χρόνο εφαρμογής, γιατι βλέπουν μόνο μπροστά στο πρόβλημα.

- Προσεγγιστικοί αλγόριθμοι (approximation algorithms).

  Οι προσεγγιστικοί αλγόριθμοι προσπαθούν να επιλύσουν το πρόβλημα χρησιμοποιώντας επιπλέον πληροφορία.

- Αλγόριθμοι τοπικής αναζήτησης (local search algorithms).

  Οι αλγόριθμοι τοπικής αναζήτησης προοσπαθούν να καταλήξουν στη λύση ξεκινώντας από μία αρχική εφικτή λύση και τη βελτιώνουν με κάποια μέθοδο αναζήτησης καλύτερης λύσης στη *''γειτονιά''* της λύσης.

Οι δύο πρώτες κατηγορίες μπορούμε να πούμε ότι χρησιμοποιούνται για την παραγωγή της αρχικής λύσης, ενώ η τρίτη κατηγορία για την βελτίωση της ήδη υπάρχουσας λύσης.

3.1.1 Κλασσικοί Ευρετικοί Αλγόριθμοι (Classical Heuristics)

Από την πληθώρα ευρετικών αλγορίθμων που έχουν εφευρεθεί για την επίλυση του προβλήματος δρομολόγησης οχημάτων (VRP), περιγράφουμε τους κλασσικούς ευρετικούς αλγορίθμους για τέτοια προβλήματα όπως τους ταξινόμησαν οι Laporte & Semet (2002) [5] στα πλαίσια των παρακάτων ενοτήτων: κατασκευαστικοί, μέθοδος δύο φάσεων και μέθοδος βελτίωσης διαδρομής.

1) Κατασκευαστικοί Ευρετικοί αλγόριθμοι.

Η μέθοδος κατασκευής της διαδρομής ήταν η πρώτη που εφαρμόστηκε σε προβλήματα δρομολόγησης οχημάτων περιορισμένης χωρητικότητας (CVRP). Οι αλγόριθμοι αυτοί ξεκινούν από ένα άδειο διάνυσμα λύσης και επαναληπτικά δημιουργούν διαδρομές εισάγοντας έναν πελάτη σε κάθε επανάληψη, μέχρι να εξυπηρετηθούν όλοι οι πελάτες. Η διαδικασία έχει δύο εκδοχές, την παράλληλη και την ακολουθητική, ανάλογα με τον αριθμό των επιλέξιμων διαδρομών για την εισαγωγή ενός πελάτη.

2) Ευρετικοί αλγόριθμοι δύο φάσεων.

Η μέθοδος δύο φάσεων βασίζεται στην αποσύνθεση της διαδικασίας της λύσης ενός προβλήματος δρομολόγησης οχημάτων σε δύο χωριστά υποπροβλήματα.

a) Ομαδοποίηση (clustering) πελατών σε υποσύνολα και αντιστοίχηση μία διαδρομής σε κάθε ένα από αυτά.
b) Δρομολόγηση (routing) των οχημάτων, δηλαδή καθορισμός της σειράς που θα επισκεφθούν τους πελάτες.

3) Ευρετικοί αλγόριθμοι βελτίωσης της διαδρομής.

Εδώ αναφερόμαστε στις μεθόδους τοπικής αναζήτησης που χρησιμοποιούνται στην βελτίωση λύσεων που έχουν προκύψει από άλλους ευρετικούς αλγορίθμους.

3.1.2 Αλγόριθμοι απληστίας (greedy algorithms)

Ένας άπληστος αλγόριθμος είναι ένας αλγόριθμος που ακολουθεί την ευρετική επίλυση προβλημάτων και προσπαθεί να κατασκευάσει τη τοπική βέλτιστη επιλογή σε κάθε στάδιο με την ελπίδα να βρεθεί μια ολική βέλτιστη. Σε πολλά προβλήματα, μια άπληστη στρατηγική σε γενικές γραμμές δεν παράγει μια βέλτιστη λύση, αλλά παρ 'όλα αυτά μια άπληστη προσέγγιση μπορεί να δώσει τοπικά βέλτιστες λύσεις που προσεγγίζουν μια παγκόσμια βέλτιστη λύση σε ένα εύλογο χρονικό διάστημα.

Επειδή όμως οι αλγόριθμοι αυτοί χαρακτηρίζονται ως μυωπικοί, δηλαδή βλέπουν μονο μπροστά και η λύση κατασκευάζεται τμηματική, βρίσκοντας την καλύτερη λύση από το σημείο που βρισκόμαστε μια δεδομένη στιγμή και δεν ελέγχονται άλλες λύσεις που πιθανά να δίνουν καλύτερο αποτέλεσμα, οι αλγόριθμοι αυτοί δίνουν βέλτιστη λύση μόνο υπό προϋποθέσεις. Στις περισσότερες περιπτώσεις δίνουν ικανοποιητικές αρχικές εφικτές λύσεις, οι οποίες βελτιώνονται με άλλους αλγόριθμους.

3.2 Αλγόριθμοι Τοπικής Αναζήτησης (Local Search)

Η τοπική αναζήτηση βασίζεται στην αρχαιότερη μέθοδο βελτιστοποίησης, στη μέθοδο δοκιμής και σφάλματος [6]. Η τοπική αναζήτηση έχει αποδειχθεί πολύ επιτυχημένη στην πράξη σε ένα πολύ μεγάλο αριθμό προβλημάτων.

Ένας τυπικός αλλά πολύ γενικός αλγόριθμος τοπικής αναζήτησης ξεκινάει από μία αρχική εφικτή λύση t και χρησιμοποιεί μία υπορουτίνα 2_opt(i) για να ψάξει για μια καλύτερη λύση στη γειτονιά της αρχικής λύσης t. Όσο βρίσκεται μια βελτιωμένη λύση, εφαρμόζεται και επαναλαμβάνεται η διαδικασία αναζήτησης για βελτίωση της λύσης. Ότα φτάσουμε σε τοπικό ελάχιστο σταματάμε.

Διαδικασία local_search

Begin

t μια αρχική λύση του προβλήματος

do while βρίσκετε μια βελτιωμένη λύση (2_opt (t))

t = 2_opt (t)

return t

end

Ένα πολύ σημαντικό χαρακτηριστικό της τοπικής αναζήτησης είναι το γεγονός ότι μπορεί να εκτελείται από πολλά διαφορετικό αρχικά σημεία και να επιλέγεται το καλύτερο σαν βέλτιστη λύση. Σε αυτές τις περιπτώσεις μια άλλη απόφαση που πρέπει να παρθεί είναι πόσα θα είναι τα αρχικά σημεία που θα πρέπει να επιλέξουμε. Το επόμενο πρόβλημα που πρέπει να επιλυθεί για να είμαστε σε θέση να πούμε ότι η τοπική αναζήτηση που εφαρμόζουμε οδηγεί σε ικανοποιητικά αποτελέσματα είναι το ότι θα πρέπει να γίνει σωστή επιλογή της γειτονιάς που θα πραγματοποιηθεί η αναζήτηση.

Τέλος, το σημαντικότερο ίσως στοιχείο για την επιτυχία της διαδικασίας είναι η επιλογή της μεθόδου που θα χρησιμοποιηθεί για την αναζήτηση. Στην βιβλιογραφία του προβλήματος υπάρχουν αρκετές μέθοδοι που έχουν εφαρμοστεί για την επίλυση του. Οι σημαντικότερες μέθοδοι τοπικής αναζήτησης είναι η μέθοδος 2-opt, η μέθοδος 3-opt, η περιορισμένη αναζήτηση και ο αλγόριθμος των Lin-Kernigham.

3.3 Μεθευρετικοί Αλγόριθμοι (Metaheuristics)

3.3.1 Εισαγωγή στους Μεθευρετικούς αλγόριθμους

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 [1]. Τα τελευταία χρόνια οι περισσότεροι αλγόριθμοι που έχουν αναπτυχθεί για την επίλυση των προβλημάτων συνδυαστικής βελτιστοποίησης ανήκουν σε αυτή την κατηγορία. Οι γενετικοί αλγόριθμοι εξετάζουν σε κάθε βήμα ένα πληθυσμό από λύσεις. Κάθε πληθυσμός παράγεται από τον προηγούμενο του, συνδυάζοντας τα καλύτερα στοιχεία του και διαγράφοντας τα χειρότερα.

Οι μεθευρετικοί αλγόριθμοι συνήθως χρησιμοποιούν πιο παραδοσιακούς ευρετικούς αλγορίθμους σαν υποδιαδικασίες τους. Πολλές φορές μπορούν να επιτραπούν σε κάποιο μεθευρετικό αλγόριθμο, και βήματα που οδηγούν σε μη εφικτή ενδιάμεση λύση σε κάποιο βήμα του αλγορίθμου. Ο λόγος που επιτρέπεται αυτό είναι για να αποφευχθεί κάποιο τοπικό ελάχιστο και η ολική λύση που θα εξαχθεί από τον αλγόριθμο να είναι καλύτερη. Ένα επιπλέον χαρακτηριστικό αυτών των αλγορίθμων είναι ότι προσωμοιάζουν μια διαδικασία που συνήθως εφαρμόζεται στη φύση. Τα στοιχεία που χρησιμοποιούν αυτοί οι αλγόριθμοι και μεταφορικά τα παρατηρούμε και στη φύση είναι τα εξής:

- Χρησιμοποιούν έναν αριθμό από επαναληπτικές δοκιμές.
- Περιλαμβάνουν ένα ή περισσότερους πράκτορες (νευρώνες, μόρια κ.λπ.).
- Λειτουργούν (στην περίπτωση των πολυ-πρακτόρων) βάση ενός μηχανισμού συνεργασίας και ανταγωνισμού.
- Περιλαμβάνουν διαδικασίες αυτο-τροποποιήσεων των ευρετικών παραμέτρων ή ακόμα και της αναπαράστασης του προβλήματος.

Τα χαρακτηριστικά των μεθευρετικών είναι τα εξής:

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

3.3.2 Η μέθοδος της Άπληστης Τυχαιοποιημένης Προσαρμοστικής Αναζήτησης (GRASP)

Η διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης (Greedy Randomized Adaptive Search Procedure) είναι μια επαναληπτική διαδικασία για την εύρεση προσεγγιστικών λύσεων σε προβλήματα συνδυαστικής βελτιστοποίησης [8]. Αυτη 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 επανάληψη αποτελείται από δύο φάσεις, μία φάση κατασκευής μια αρχικής λύσης (construction phase) και μία διαδικασία τοπικής αναζήτησης (local search phase) για βελτιστοποίηση της λύσης. Στην φάση κατασκευής, μια τυχαιοποιημένη συνάρτηση απληστίας χρησιμοποιείται για να κατασκευαστεί η αρχική λύση Αυτή η λύση στη συνέχεια βελτιώνεται με μεθόδους local search.

<u>Φάση Κατασκευής</u>

Η φάση κατασκευής επιτυγχάνεται μέσω της μεθόδου τυχαιοποιημένης απληστίας (randomized greed). Η μέθοδος μπορεί να περιγραφεί ως επαναληπτική 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 (Restricted Candidate List) για εισαγωγή στη λύση. Η λίστα περιορισμού δημιουργείται, εισάγοντας έναν περιορισμένο αριθμό στοιχείων, τα οποία κατατάσσονται βάσει μια συνάρτησης απληστίας. Η τυχαία επιλογή του στοιχείου σημαίνει ότι δεν είναι ανάγκη να επιλεγεί το πρώτο στοιχείο στη λίστ. Η κατασκευή της λίστας περιορισμού των υποψηφίων είναι το πιο σημαντικό κομμάτι της μεθόδου πιθανότατα, αφού από αυτό ελέγχεται η διασπορά των λύσεων που παράγονται. Αν για παράδειγμα η λίστα ειναι πολύ μικρή τότε η λύσει που θα παράγονται θα είναι περίπου όμοιες μεταξύ τους, ενώ αν είναι μεγάλη θα είναι ουσιαστικά τελείως τυχαιοποιημένη λύση με τεράστιες διακυμάνσεις στην τιμή.

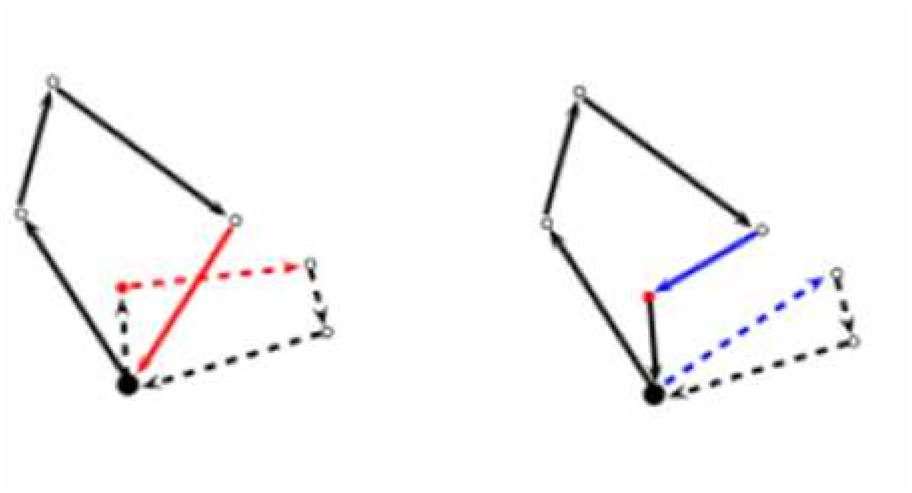
Εφαρμόζοντας τη μέθοδο σε προβλήματα δρομολόγησης οχημάτων, θεωρούμε τη λίστα περιορισμού υποψηφίων, ένα υποσύνολο πελατών, οι οποίοι έχουν ταξινομηθεί με βάση το κόστος που έχει συμφωνηθεί να βελτιστοποιηθεί και από αυτούς διαλέγουμε τυχαία έναν. Έπειτα, σύμφωνα με την μέθοδο της τυχαιοποιημένης απληστίας, εξετάζουμε αν ο πελάτης που επιλέχθηκε τυχαία από τη λίστα περιορισμού, ικανοποιεί τους περιορισμούς του αντίστοιχου προβλήματος που επιλύουμε και μπορεί να αποτελέσει μέρος της διαδρομής. Η διαδικασία αυτή συνεχίζεται μέχρι κάποιος πελάτης να μην ικανοποιεί τους περιορισμού, οπότε το φορτηγό επιστρέφει στην αποθήκη. Η μέθοδος επαναλαμβάνεται μέχρι να εξυπηρετηθούν όλοι οι πελάτες και να σχηματιστούν διαδρομές, που αποτελούν εφικτή λύση, συνήθως όμως δεν αντιστοιχεί σε τοπικό (ή ολικό) βέλτιστο, γι'αυτό και ακολουθεί η φάση της τοπικής αναζήτησης.

Φάση Τοπικής Αναζήτησης

Η τοπική αναζήτηση εφαρμόζεται ώστε από μία αρχική εφικτή λύση, να οδηγηθούμε σε καλύτερη που θα περιέχει τοπικό ελάχιστο. Για να εφαρμοστεί η φάση τοπικής αναζήτησης, καθορίζεται μια συνάρτηση που να κάνει αναζήτηση στη γειτονιά της αρχικής λύσης μέχρι να φτάσει σε τοπικό ελάχιστο.

Σύμφωνα με τον Waters [9] υπάρχουν τέσσερις διαδικασίες ανταλλαγής που μπορούν να εφαρμοστούν μεταξύ δύο ή περισσότερων διαδρομών σε προβλήματα δρομολόγησης οχημάτων:

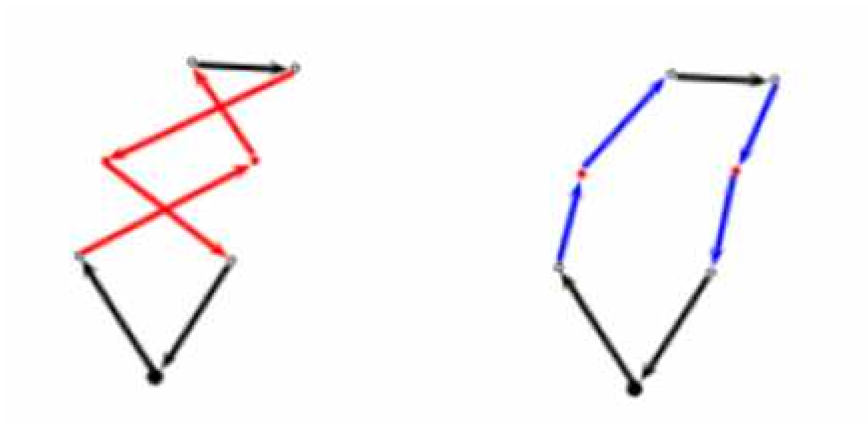
1) **1-0 Relocate (1-0 Επανατοποθέτηση)**
   Μία απλή διαγραφή ενός πελάτη από μία διαδρομή και επανατοποθέτηση του σε μία άλλη διαδρομή με καλύτερο κόστος.



Σχήμα 3.1: Μέθοδος 1-0 Relocate

2) **1-1 Exchange (1-1 Ανταλλαγή)**

   Μία ταυτόχρονη ανταλλαγή δύο πελατών που βρίσκονται σε διαφορετικές διαδρομές.



Σχήμα 3.2: Μέθοδος 1-1 Exchange

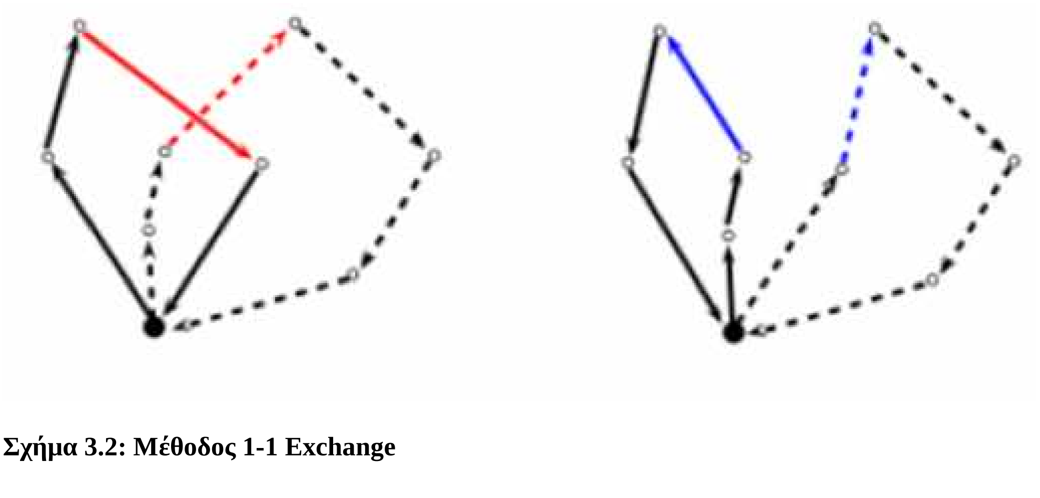
3) **<u>1-2 Exchange (1-2 Ανταλλαγή)</u>**
   Μία συνδυασμένη ανταλλαγή όπου ένας πελάτης μιας διαδρομής ανταλλάσσεται με δυο πελάτες που βρίσκονται σε διαφορετική διαδρομή.

4) **<u>2-2 Exchange (2-2 Ανταλλαγή)</u>**
   Μία συνδυασμένη ανταλλαγή όπου δύο πελάτες μιας διαδρομής ανταλλάσσονται με δυο πελάτες που βρίσκονται σε διαφορετική διαδρομή.

Άλλες αποτελεσματικές μέθοδοι τοπικής αναζήτησης:

<u>Μέθοδος 2-opt</u>

Η μέθοδος αυτή σε γενικές γραμμές διαγράφει 2 ακμές και επανασυνδέει τα δύο μονοπάτια με διαφορετικό τρόπο, έτσι ώστε να καθοριστεί μια καινούρια διαδρομή, η οποία θα αποφέρει καλύτερο κόστος [10].



Σχήμα 3.2: Μέθοδος 1-1 Exchange

Η διαδικασία της μεθόδου είναι η ακόλουθη:

Βήμα 1. Έστω T η τρέχουσα διαδρομή.

Βήμα 2. Για κάθε κόμβο i = 1, …, n:

Εξετάξουμε όλες τις πιθανές 2-opt κινήσεις που μπορεί να γίνουν από την i και την επόμενη της μέσα στη διαδρομή. Αν με αυτό τον τρόπο μπορούμε να μειώσουμε το κόστος της διαδρομής, τότε επιλέγουμε την καλύτερη 2-opt κίνηση και εφαρμόζουμε τις αλλαγές στη διαδρομή T.

Βήμα 3. Αν δε βρεθεί καλύτερη λύση, σταματάμε.

Ένα πολύ σημαντικό σημείο του αλγορίθμου είναι η επιλογή των τόξων που θα διαγραφούν από τη λύση και των τόξων που θα εισχωρηθούν σε αυτή. Υπάρχουν αρκετές προσεγγίσεις που έχουν ληφθεί υπ’όψιν στην βιβλιογραφία. Μια προσέγγιση είναι να διαγράψουμε τα δύο χειρότερα τόξα και να τα αντικαταστήσουμε με δύο άλλα τόξα. Στο συγκεκριμένο αλγόριθμο όταν θα διαγραφούν δύο τόξα, τότε υπάρχει μόνο ένας τρόπος για να επανασυνδεθούν, αλλιώς δεν διατηρείται ο κύκλος και δημιουργούνται δύο διαφορετικοί κύκλοι. Ο δεύτερος τρόπος είναι να διαγραφεί το χειρότερο και να δοκιμάσουμε διάφορες παραλλαγές με τα υπόλοιπα τόξα για να δούμε πότε βελτιώνεται η λύση. Τέλος, ο τρίτος τρόπος είναι να διαγράφονται δύο τόξα κάθε φορά τυχαία. Και οι τρεις τρόποι έχουν τα πλεονεκτήματα και τα μειονεκτήματά τους. Για παράδειγμα ο τρίτος τρόπος μπορεί να είναι πιο γρήγορος από τους άλλους γιατί δεν χρειάζεται να υπολογίζει σε κάθε επανάληψη τα χειρότερα τόξα, όμως ταυτόχρονα ο ίδιος ακριβώς λόγος μπορεί να κάνει τον αλγόριθμο πιο αργό αν δεν μπορεί να βρεθεί βελτιωμένη λύση.

<u>Μέθοδος 3-opt</u>

Αντίστοιχη με την προηγούμενη μέθοδο, μόνο που μπορούμε να διαχωρίσουμε τον αλγόριθμο σε τρία μέρη αντί για δύο και να συνδυάσουμε τα μονοπάτια με τον καλύτερο δυνατό τρόπο.

Ο ψευδοκώδικας για τη μέθοδο GRASP, δίνεται παρακάτω:

For όλες τις επαναλήψεις

  For n βήματα

    Εφαρμογή Τυχαιοποιημένης Συνάρτησης Απληστίας (για δημιουργία νέας λύσης).

    Αποθήκευση λύσης και αντίστοιχου κόστους

  End

  For (Όλες τις λύσεις)

```
        Εφαρμογή τοπικής αναζήτησης
        Υπολογισμός της αντικειμενικής συνάρτησης της λύσης
        If κόστος νέας λύσης < κόστος παλαιότερης
            Αντικατάσταση παλαιάς λύσης και του αντίστοιχου κόστους
        End
    End
End
```

Κεφάλαιο 4: Περιγραφή του Προβλήματος

4.1 Γενικά

Στην παρούσα διπλωματική εργασία ασχολούμαστε με την επίλυση του προβλήματος της δρομολόγησης οχημάτων με περιορισμένη χωρητικότητα (Capacitated VRP ή CVRP για συντομία).

Σκοπός του προβλήματος, όπως αναφέρθηκε και σε προηγούμενα κεφάλαια, είναι ο καθορισμός των βέλτιστων διαδρομών στόλου οχημάτων που ξεκινάνε από μία κεντρική αποθήκη και εξυπηρετούν όλους τους πελάτες, έτσι ώστε να ελαχιστοποιηθεί το συνολικό κόστος. Τα οχήματα του στόλου είναι ομογενή, έχουν δηλαδή ίδια πεπερασμένη χωρητικότητα, η οποία οριοθετείται από το εκάστοτε πρόβλημα, ενώ όλα ξεκινούν από την ίδια αποθήκη. Κάθε πελάτης έχει πεπερασμένη ζήτηση που πρέπει να καλυφθεί. Θεωρούμε ότι το κόστος μία διαδρομής προκύπτει αθροιστικά από τις επιμέρους αποστάσεις μεταξύ των πελατών που ανήκουν στη διαδρομή.

Οι περιορισμοί που διέπουν το πρόβλημα είναι οι εξής:

1) Κάθε πελάτης πρέπει να εξυπηρετείται μόνο μία φορά και από ένα μόνο φορτηγό.
2) Το σύνολο των διανομών και παραλαβών που πρέπει να γίνουν σε κάθε διαδρομή, δεν πρέπει να ξεπερνάει την χωρητικότητα του οχήματος.
3) Η συνολική απόσταση που διανύει το όχημα της διαδρομής δεν θα πρέπει να ξεπερνάει ένα μέγιστο μήκος διαδρομής, το οποίο είναι προκαθορισμένο, αυτό συνυπολογίζουμε κάθε φορά και τον χρόνο εξυπηρέτησης κάθε πελάτη. Στο συγκεκριμένο πρόβλημα δεν υπάρχει μέγιστο μήκος διαδρομής.
4) Όλα τα οχήματα πρέπει να ξεκινάνε και να τερματίζουν στην ίδια αποθήκη.
5) Η συνολική ποσότητα που μεταφέρει ένα όχημα προς παράδοση, πρέπει να είναι μη αρνητική και να μην ξεπερνάει την χωρητικότητα του οχήματος.

Άρα, όταν κάποια συνθήκη περιορισμού δεν ικανοποιείται ή έχουν εξυπηρετηθεί όλοι οι πελάτες, τα οχήματα είναι αναγκασμένα να επιστρέψουν στην αποθήκη από την οποία ξεκίνησαν.

Το παραπάνω πρόβλημα θα επιλυθεί με την εφαρμογή της μεθόδου GRASP που αναφέραμε στο προηγούμενο κεφάλαιο. Η αρχική λύση βρέθηκε με τον αλγόριθμο του πλησιέστερου γείτονα, ενώ μετά την αρχική λύση εφαρμόστηκε ο αλγόριθμος n-opt για την βελτίωση της λύσης.

4.2 Αλγόριθμος Πλησιέστερου Γείτονα (Nearest Neighbour)

<u>1η Εκδοχή</u>

Ο αλγόριθμος πλησιέστερου γείτονα είναι ένας από τους πρώτους αλγορίθμους που χρησιμοποιήθηκαν για την επίλυση του προβλήματος του πλανόδιου πωλητή. Σε αυτό, ο πωλητής ξεκινάει τυχαία σε μία από τις πόλεις και επαναληπτικά επισκέπτεται την κοντινότερη σε αυτόν πόλη που δεν έχει επισκεφθεί. Παράγει γρήγορα μία αρχική λύση, η οποία βέβαια σπάνια είναι η βέλτιστη. Ο αλγόριθμος αυτός είναι πολύ εύκολος στην εφαρμογή και εκτελείται γρήγορα, αλλά πολλές φορές ''χάνει'' διαδρομές, οι οποίες είναι προφανώς οικονομικότερες ακόμα και με το ανθρώπινο μάτι, εξαιτίας της άπληστης φύσης του.

<u>2η Εκδοχή</u>

Η διαφορά της δεύτερης εκδοχής του αλγορίθμου πλησιέστερου γείτονα με την πρώτη είναι ότι σε αυτή την περίπτωση ο αλγόριθμος δεν σταματάει όταν ο περιορισμός χωρητικότητας πάψει να ισχύει.

Για παράδειγμα, αν βρισκόμαστε στον πελάτη i και ο κοντινότερος σε αυτόν j δεν μπορεί να επισκεφθεί, γιατί ο περιορισμός χωρητικότητας δεν ικανοποιείται, στην πρώτη περίπτωση, το φορτηγό θα γύριζε στην αποθήκη. Εδώ όμως, το φορτήγο ελέγχει τον αμέσως κοντινότερο του i, μετά τον j όπου αν και αυτός δεν ''χωράει'' στη διαδρομή ελέγχει τον επόμενο κ.ο.κ. Με αυτό τον τρόπο μπορεί να εξασφαλιστεί μία διαφορετική διαδρομή από αυτή που θα μας έδινε η 1η εκδοχή του αλγορίθμου, με συνήθως καλύτερες λύσεις, λόγω του ότι μπορούμε να ξεκολλήσουμε από έναν ''δύσκολο'' πελάτη και να μην σπαταλήσουμε ένα φορτηγό επιπλέον γι'αυτόν.

4.3 Μοντελοποίηση του προβλήματος

Σύμφωνα λοιπόν με την παραπάνω περιγραφή, τα δεδομένα που χρειάζονται για την επίλυση του προβλήματος είναι το πλήθος των πελατών, η ζήτηση και η παραλαβή από και προς τον καθένα, ο χρόνος εξυπηρέτησης και η γεωγραφική του θέση. Χρειαζόμαστε επίσης τη χωρητικότητα των οχημάτων καθώ και την τιμή του μέγιστου μήκους διαδρομής. Τα δεδομένα αυτά παραλαμβάνονται από αρχεία κειμένου .txt και εισάγωνται στο πρόγραμμα Matlab. Κάθε γραμμή του αρχείου αντιστοιχεί και σε μία παράμετρο του προβλήματος.

Στο συγκεκριμένο πρόβλημα έχουν χρησιμοποιηθεί τα δεδομένα τα οποία είχαν προτείνει Augerat (1995) και έχουν ευρέως χρησιμοποιηθεί για το πρόβλημα του CVRP. Τα δεδομένα που θα χρησιμοποιήσουμε πάρθηκαν από τον ιστότοπο http://www.vrp-rep.org στον οποίο το database δώθηκε από τον Hubert Lobit. Ο κάθε πελάτης έχει 2 συντεταγμένες, που αποτελούν τις καρτεσιανές συντεταγμένες του στο χάρτη, κάθε πελάτης έχει τη ζήτησή του, ενώ στο πάνω μέρος δίνεται αναλυτικά το πλήθος των πελατών, η χωρητικότητα του φορτηγού, σε κάποια παραδείγματα το άνω χρονικό παράθυρο, το οποίο είναι 9.999, για να μη λαμβάνεται υπόψη.

Στην εικόνα παρακάτω φαίνεται η μορφή που έχουν τα δεδομένα για το πρόβλημα.

```
31  --> Πελάτης
100  --> Χωρητικότητα
9999  --> Μέγιστο Μήκος
NODE_COORD_SECTION
1          17          76
2          24          6
3          96          29
4          14          19
5          14          32
6          0           34
7          16          22
8          20          26
9          22          28
10         17          23
DEMAND_SECTION
1          0
2          25
3          3
4          13
5          17
6          16
7          9
8          22
9          10
10         16
```

Εικόνα 4.1: Παράδειγμα 10 πελατών

Στο συγκεκριμένο παράδειγμα δεν υπάρχει χρόνος εξυπηρέτησης. Τα δεδομένα περάστηκαν στο περιβάλλον Matlab μέσω του εργαλείου Import, που παρέχεται στη Matlab πηγαίνοντας home → Import data → open *.txt και οι μεταβλητές που χρειάστηκε να αποθηκεύσουμε ήταν:

Q: Η χωρητικότητα κάθε φορτηγού,

d: Διάνυσμα για τη ζήτηση κάθε πελάτη,

coordinates: Πίνακας (N,2) με τις συντεταγμένες X,Y,

Στη συνέχεια δημιουργήθηκε ο πίνακας που περιέχει τις αποστάσεις μεταξύ όλων των πελατών, με την βοήθεια του πίνακα coordinates. Ο πίνακας ονομάζεται D(i,j), όπου i είναι ο πελάτης στον οποίο βρισκόμαστε σε κάθε βήμα και j ο πελάτης τον οποίο επισκεπτόμαστε. Το κόστος που έχει η διαδρομή για να πάμε από έναν πελάτη i στον j είναι η ευκλείδια απόσταση μεταξύ τους. Ο πίνακας είναι συμμετρικός, δηλαδή το κόστος που χρειαζόμαστε για να πάμε από τον πελάτη i στον j είναι ίδιο με το κόστος από τον j στον i. Σημαντική σημείωση για τον πίνακα D(i,j) είναι ότι τα σημεία της διαγωνίου του πίνακα απειρίζονται αφού δεν θέλουμε να λαμβάνονται υπ'όψιν οι αποστάσεις από έναν πελάτη στον εαυτό του.

Γεμίζουμε πάνω από τη διαγώνιο με την παρακάτω εξίσωση και στη συνέχεια τον κάνουμε συμμετρικό και απειρίζουμε τη διαγώνιο.

$$D(i,j) = \sqrt{(coordinates(i,1) - coordinates(j,1))^2 + (coordinates(i,2) - coordinates(j,2))^2}$$

Για λόγους ευκολίας θεωρούμε ότι όλες οι μονάδες των μεταβλητών του προβλήματος είναι ίδιες (service time, costs κ.λπ.) και εκφράζονται όλες σε μονάδες απόστασης, ενώ το μέγεθος του στόλου των οχημάτων που μπορεί να χρησιμοποιηθεί είναι άπειρο.

4.4 Εύρεση Αρχικής Λύσης

Μετά λοιπον την εισαγωγή όλων των δεδομένων του προβλήματος, με τη μορφή που διευκολύνουν την επίλυση, συνέχεια έχει η αρχικοποίηση των μεταβλητών και η εύρεση της αρχικής λύσης. Θα δηλώσουμε δηλαδή τις βασικές μεταβλητές και την κατάστασή τους πριν ξεκινήσουμε την επίλυση του προβλήματος. Έχουμε:

EfiktosGiaEisagwgh: Διάνυσμα, το οποίο εμπεριέχει τους πελάτες που ελέγχουμε για εφικτότητα.

Aneksypiretitoipel: Είναι ένα διάνυσμα, το οποίο έχει μέσα τους πελάτες που εξυπηρετούμε.

Routes: Το διάνυσμα που κρατάμε τις διαδρομές.

Kalyteros_pel_gia_eisagwgh:

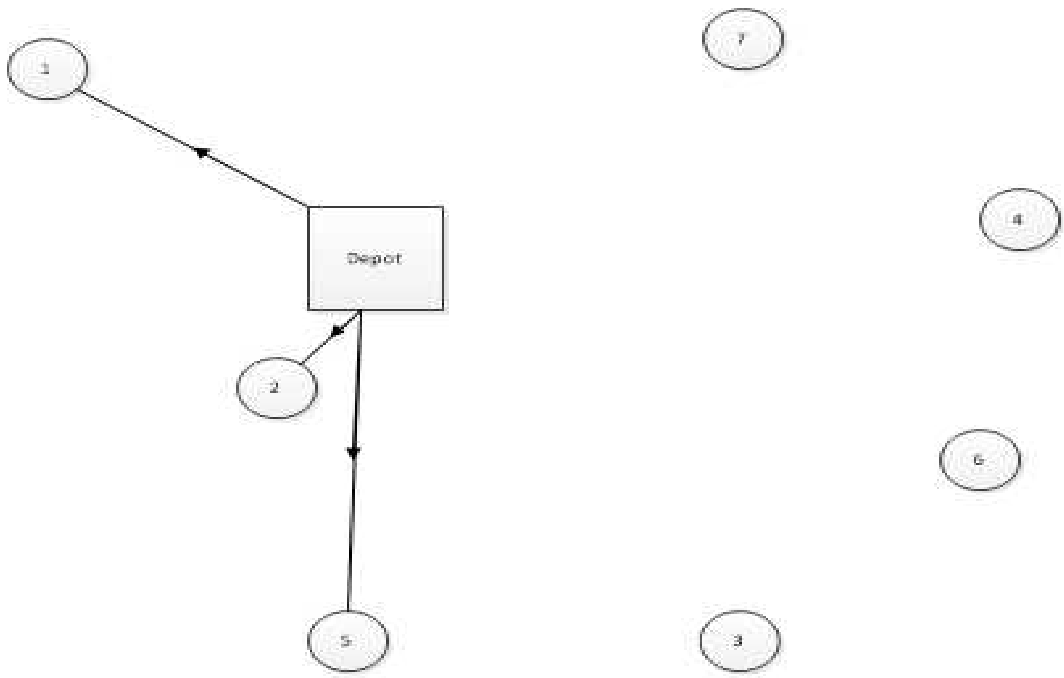
Είναι ο πελάτης που θα εξυπηρετήσουμε αμέσως μετά.

TotalDistance: Η απόσταση που διανύουμε στο σύνολο της διαδρομής.

4.4.1 Κατασκευή μιας αρχικής λύσης

Στο παρόν υποκεφάλαιο θα λύσουμε ένα μικρής διάστασης πρόβλημα με τον αλγόριθμο που επιλύσαμε το πρόβλημα δρομολόγησης οχημάτων με περιορισμένη χωρητικότητα. Το παρόν πρόβλημα περιέχει 7 πελάτες με τον καθένα να απέχει μία συγκεκριμένη απόσταση από την αποθήκη και μεταξύ τους.

Ο αλγόριθμος δουλεύει βηματικά, από τον πίνακα με τα κόστη βρίσκουμε κάθε φορά από τον πελάτη i όλους τους εφικτούς πελάτες. Στη συνέχεια βρίσκουμε τους κοντινότερους και τους προσθέτουμε σε ένα διάνυσμα, μαζί με τα index τους, τις θέσεις που βρίσκονται αυτοί οι πελάτες. Βρίσκουμε τους 3 κοντινότερους πελάτες από εκεί που βρισκόμαστε και τυχαία επιλέγουμε έναν, σύμφωνα με τη θεωρία του grasp, για να μπορέσουμε να ξεφύγουμε από τοπικό ελάχιστο και να γλιτώσουμε "κακές" διαδρομές. Υπολογίζουμε τα κόστη, εάν είναι εφικτός ο κάθε πελάτης που βάζουμε στη διαδρομή και εάν όλα τηρούνται καταλήγουμε σε μία διαδρομή. Ο αλγόριθμος τελειώνει όταν έχουν εξυπηρετηθεί όλοι οι πελάτες, δηλαδή όταν το length του διανύσματος Aneksypiretitoipel είναι μηδέν.



Σχήμα 4.2: Παράδειγμα με 7 πελάτες

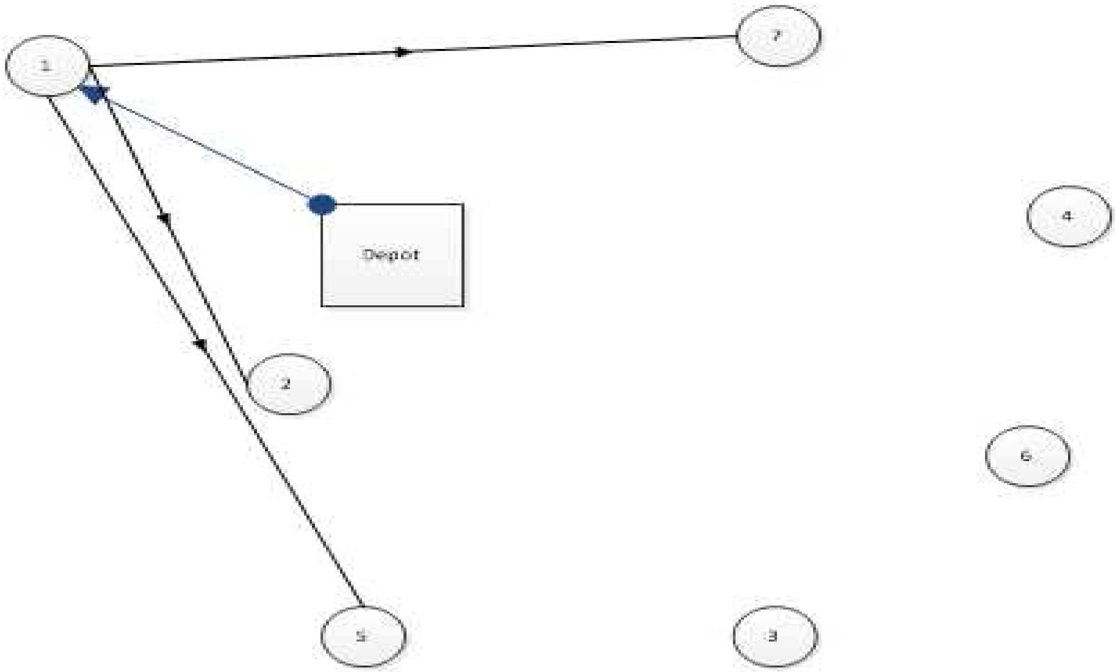
Βήμα 1^{o}: Από τους 3 κοντινότερους, όπως βλέπουμε παραπάνω, διαλέγουμε τυχαία έναν, π.χ τον 1.

Βήμα 2^{o}: Ελέγχουμε τους περιορισμούς, οι οποίοι πρέπει να ισχύουν ταυτόχρονα. Οι περιορισμοί είναι:

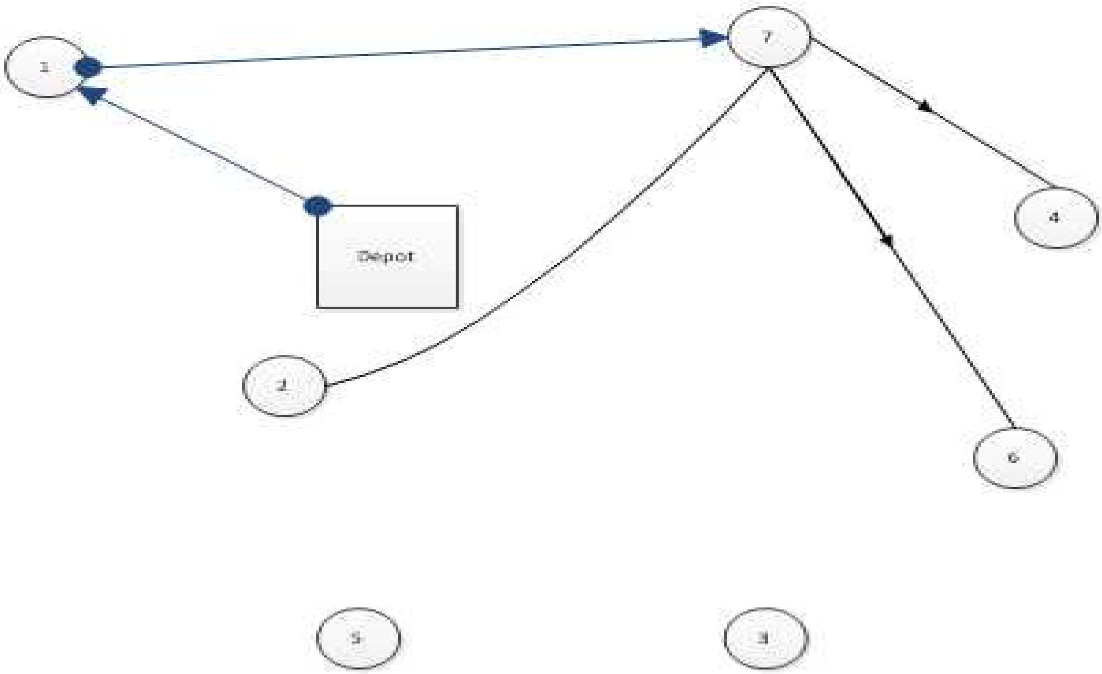
1. `(Load(h,e)+Demand(Route(h,c)))>Capacity`

όπου μας εξασφαλίζουν ότι δεν θα υπάρξει κάποιο μη εφικτό πρόβλημα με τη χωρητικότητα των φορτηγών και ότι το φορτηγό δεν θα ξεπεράσει το μέγιστο επιτρεπόμενο μήκος που μπορεί να διανύσει.

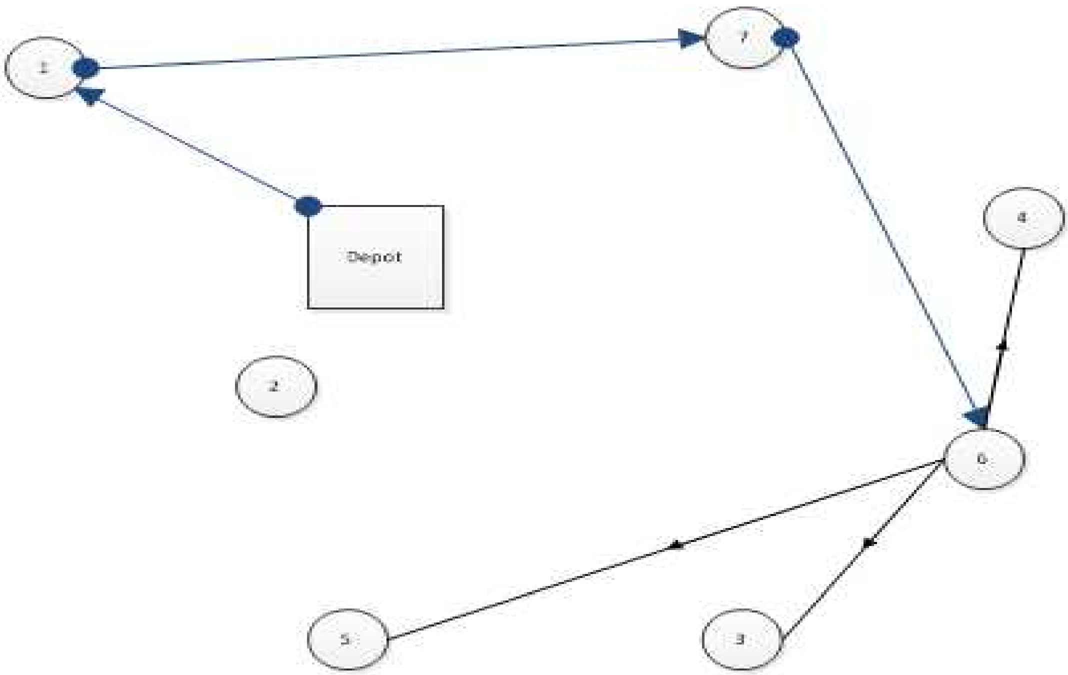
Εάν όλοι οι περιορισμοί ικανοποιούνται, ο πελάτης εισέρχεται στη διαδρομή. Άρα θα έχουμε το εξής:



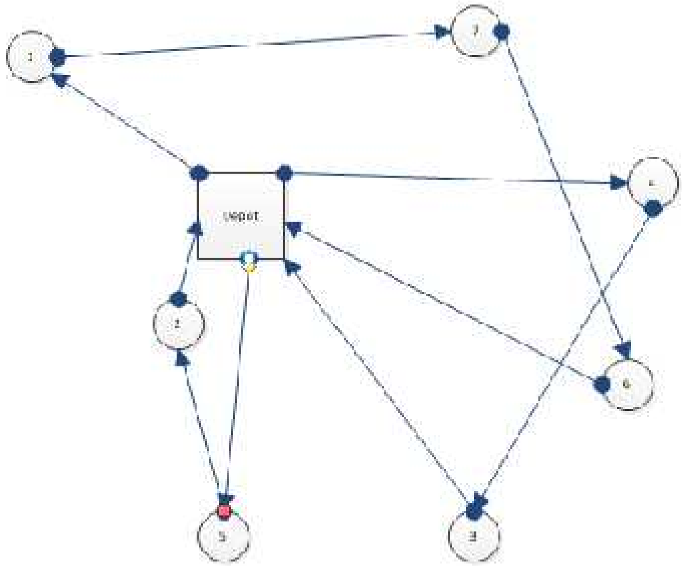
Από τον 1 δηλαδή θα εξετάσουμε τους 3 κοντινότερούς του, τους **[2 5 7].** Τυχαία επιλέγεται ο 7. Ελέγχοντας τους περιορισμού, βλέπουμε ότι τους ικανοποιεί άρα μπορεί να εισαχθεί στην διαδρομή.



Παρατηρούμε λοιπόν τώρα ότι η διαδρομή έχει γίνει **[0 1 7]** και εξετάζονται για εισαγωγή ένας εκ των **[2 4 6].** Ο αλγόριθμος επιλέγει τυχαία τον 4 π.χ., αλλά βλέπουμε ότι δεν ικανοποιούνται οι περιορισμοί. Αντί να επιστρέψει στην αποθήκη, ο αλγόριθμος θα εξετάσει, τυχαία πάλι, έναν εκ των 2 ή 6 και θα δει ότι ο 6 τους ικανοποιεί τους περιορισμούς, άρα θα πάει εκεί.



Τώρα λοιπόν οι επόμενοι προς εξέταση πελάτες είναι οι **[4 3 5]**. Εξετάζει και τους 3 και οι περιορισμοί δεν ικανοποιούνται, άρα γυρνάμε στην αποθήκη, οι μεταβλητές μας γυρίζουν στις αρχικές τιμές τους και η διαδρομή ολοκληρώνεται ως [0 1 7 6 0], όπου μηδέν η αποθήκη. Συνεχίζοντας τον αλγόριθμο μπορούμε να δούμε την τελική λύση παρακάτω.



Ο αλγόριθμος δίνει 3 διαδρομές: **[0 1 7 6 0], [0 4 3 0], [0 5 2 0]**. Πρέπει να αναφερθεί ότι το τελικό κόστος της διαδρομής θα είναι το άθροισμα του κόστους κάθε μετακίνησης προς και από τους πελάτες.

4.5 Μέθοδος Τοπικής Αναζήτησης για Βελτίωση Λύσης

4.5.1 Αλγόριθμος 2-opt

Από τις μεθόδους τοπικής αναζήτησης που αναφέραμε σε προηγούμενο κεφάλαιο, θα χρησιμοποιηθεί η μέθοδος n-opt. Η διαδικασία εφαρμόζεται σε κάθε γραμμή του πίνακα λύσης, άρα σε κάθε διαδρομή.

Δημιουργούμε ένα function noptimal.m, το οποίο θα κάνει τη διαδικασία του αλγορίθμου n-opt. Η συνάρτηση αυτή δέχεται ως εισόδους την υπάρχουσα διαδρομή και τον πίνακα με τα κόστη, εκτελεί τον αλγόριθμο και επιστρέφει τη νέα διαδρομή με το βελτιωμένο κόστος.

Ο αλγόριθμος δουλεύει ως εξής:

Ελέγχουμε όλα τα ζευγάρια ένα προς ένα, πάντα εκτός της αποθήκης και κάνουμε τις αντιμεταθέσεις. Υπολογίζουμε το νέο κόστος και σε περίπτωση που αυτό είναι καλύτερο από το αρχικό, κρατάμε τη λύση. Ο αλγόριθμος συνεχίζεται με τη νέα λύση και προχωράει σε επόμενες αντιμεταθέσεις που μπορεί να είναι πιθανές. Όταν εξετάσει το σύνολο των σημείων και βρει τις αλλαγές στη διαδρομή που βελτιώνουν τη λύση, τότε προχωράει στην επόμενη διαδρομή.

Ας εξηγήσουμε τον αλγόριθμο με ένα απλό παράδειγμα 7 πελατών:

Route(i) με totalDistance 137.5

1	3	4	2	6	5	7	8	1

Αυτή είναι η διαδρομή την οποία μας έδωσε η αρχική λύση. Εμείς θα ξεκινήσουμε τον αλγόριθμο από τον πρώτο πελάτη, δηλαδή τον 3, και θα δοκιμάσουμε διάφορες αλλαγές μέχρι να βρεθεί καλύτερη λύση. Αρχικά θα εξετάσουμε την παρακάτω αλλαγή.

1	3	4	2	6	5	7	8	1

Δηλαδή η νέα διαδρομή να γίνει ως εξής:

1	6	2	4	3	5	7	8	1

Opt_cost: 138.7

Μετά από την αλλαγή των πελατών στη διαδρομή υπολογίζουμε το συνολικό κόστος όλου του διανύσματος Route. Αυτό το κόστος το ονομάζουμε opt_cost και μας δίνει το βελτιωμένο κόστος μετά την εφαρμογή του αλγορίθμου. Εάν το νέο κόστος είναι μικρότερο από το totalDistance που είχαμε βγάλει στην αρχική λύση τότε κρατάμε τη νέα λύση και επαναλαμβάνουμε τον αλγόριθμο σε όλο το μήκος του διανύσματος. Εάν η λύση δεν είναι καλύτερη από την αρχική προχωράμε στον επόμενο πελάτη και ξαναεφαρμόζουμε τον αλγόριθμο.

Στο παράδειγμά μας δεν λειτούργησε με την πρώτη ο αλγόριθμος άρα πρέπει να κρατήσουμε το αρχικό διάνυσμα ως έχει και να προχωρήσουμε στον επόμενο πελάτη, δηλαδή να ελέγξουμε την παρακάτω αλλαγή:

1	3	4	2	6	5	7	8	1

Κάνοντας την αλλαγή αυτή ξαναυπολογίζεται το opt_cost και βλέπουμε ότι βγαίνει 136.1 μονάδες κόστους, δηλαδή μικρότερο από το realcost που βγάλαμε νωρίτερα.

Προφανώς ο περιορισμός χωρητικότητας δεν χρειάζεται να ελεγχθεί αφού κάνουμε αλλαγές εντός της διαδρομής και γνωρίζουμε ότι η διαδρομή είναι εφικτή.

1	3	5	6	2	4	7	8	1

Στη συνέχεια ο αλγόριθμος συνεχίζει στον επόμενο πελάτη, δηλαδή στον 6 πλέον, και ελέγχει έαν μία τέτοια αλλαγή στο διάνυσμα [6 2 4 7], θα μπορούσε να δίνει καλύτερο αποτέλεσμα. Κάνοντας ξανά την αλλαγή παρατηρούμε ότι δίνει καλύτερα αποτελέσματα όντως, άρα η νέα λύση θα είναι:

1	3	5	7	4	2	6	8	1

Με opt_cost = 135.4 μονάδες κόστους.

Στο συγκεκριμένο παράδειγμα έχουμε ακόμα μία επανάληψη, να ελεγχθεί η παρακάτω αλλαγή στο διάνυσμα για καλυτέρευση της λύσης.

1	3	5	7	8	6	2	4	1

Κάνοντάς τη βλέπουμε ότι δεν επιφέρεται καμία βελτίωση της λύσης, με σκοπό η τελική μας βελτιωμένη διαδρομή να είναι από αυτήν:

1	**3**	**4**	**2**	**6**	**5**	**7**	**8**	**1**

Με κόστος 137.5 μονάδες κόστους.

Σε αυτήν:

1	**3**	**5**	**7**	**4**	**2**	**6**	**8**	**1**

Με κόστος 135.4 μονάδες κόστους.

Βλέπουμε λοιπόν ότι ο αλγόριθμος 2-opt βελτίωσε τη λύση μας, όμως όπως είναι λογικό δεν έχει βρεθεί η βέλτιστη λύση για το πρόβλημα.

Να σημειωθεί ότι μπορούν να γίνουν και άλλες πιθανές αλλαγές, με τον ίδιο τρόπο σε αυτή τη διαδρομή, αλλά λόγω ευκολίας δεν τις παρουσιάζουμε.

4.6 Ψευδοκώδικας και Κατασκευή Λύσης

Ολόκληρος ο αλγόριθμος που δημιουργήθηκε από την φάση κατασκευής, μέχρι και την βελτιστοποίηση της λύσης με την βοήθεία του αλγορίθμου n-opt, επαναλαμβάνεται πολλές φορές, γιατί λόγω της τυχαιότητας που διαθέτει ο κώδικας, παράγει διαφορετικά αποτελέσματα από λύση σε λύση. Στη συγκεκριμένη εργασία, για κάθε σετ δεδομένων ο αλγόριθμος "έτρεξε" περίπου 50 φορές, για να βρεθεί καλή λύση και να διαφύγουμε από τοπικό ελάχιστο. Έτσι, κάθε βέλτιστη λύση ανά επανάληψη, αποθηκευόταν και μετά το πέρας των επαναλήψεων συγκρίνονταν τα συνολικά κόστη. Η βέλτιστη λύση, δηλαδή η μικρότερη-πιό οικονομική ήταν αυτή που κρατάγαμε.

Ακολουθεί ο ψευδοκώδικας του αλγορίθμου που κατασκευάστηκε:

ΑΡΧΗ

Εισαγωγή των δεδομένων του προβλήματος

Αρχικοποίηση μεταβλητών και πινάκων

ΌΣΟ υπάρχουν Ανεξυπηρέτητοι Πελάτες

Συνάρτηση EfiktoiPelates.m για να βρεθεί το διάνυσμα των εφικτών πελατών

Εφαρμογή του αλγορίθμου grasp

Δημιουργία της λίστας περιορισμών

Τυχαία επιλογή πελάτη από τη λίστα περιορισμών

Ο πελάτης τοποθετείται στη διαδρομή, αφού ο πελάτης είναι εφικτός

Ενημερώνονται οι μεταβλητές

ΑΝ Δεν υπάρχουν ανεξυπηρέτητοι πελάτες ή δεν υπάρχουν πελάτες που είναι εφικτοί για εισαγωγή

Επαναφέρουμε τις μεταβλητές

Ολοκληρώνουμε τη διαδρομή

ΤΕΛΟΣ_ΑΝ

Υπολογισμός Κόστους λύσης

ΤΕΛΟΣ_ΟΣΟ

```
Εφαρμογή αλγορίθμου k-opt.
ΓΙΑ όλο το διάνυσμα της διαδρομή
    Flip Πελατών που είναι στη διαδρομή
    Υπολογισμός του βελτιωμένου κόστους
    ΑΝ νέο κόστο < παλιού κόστους
        Η αλλαγή της διαδρομής γίνεται
    ΑΛΛΙΩΣ
        Οι πελάτες επιστρέφουν στην αρχική τους θέση
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΓΙΑ
Εμφάνιση αποτελεσμάτων
ΤΕΛΟΣ
```

Κεφάλαιο 5: Παρουσίαση Αποτελεσμάτων

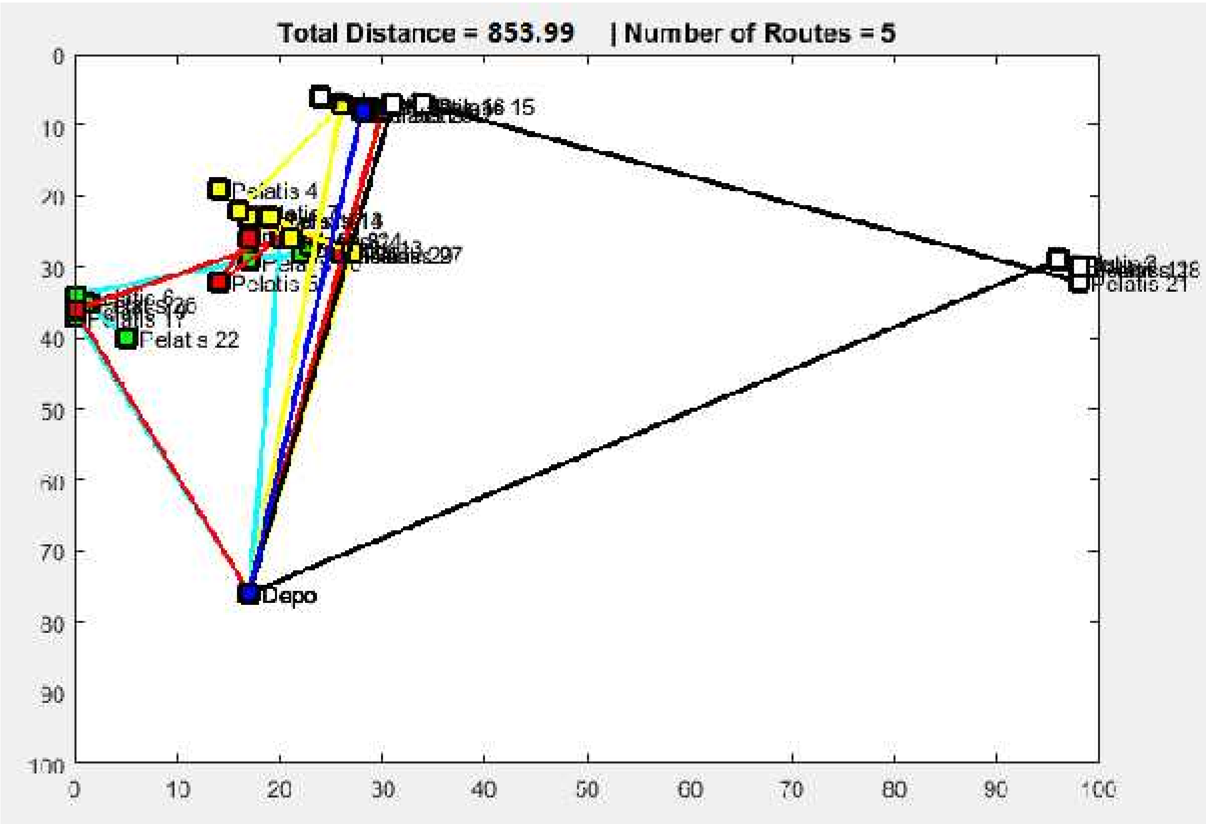
Με σκοπό την επίλυση του προβλήματος Capacitated VRP τρέξαμε μία σειρά από προτυποποιημένα προβλήματα αναφοράς (benchmark problems). Όπως αναφέραμε και στην αρχή χρησιμοποιήσαμε τα προβλήματα που προτάθηκαν από τον Augerat (1995) και έχουν ευρέως χρησιμοποιηθεί για το πρόβλημα του CVRP.

Πρόβλημα	Πελάτες	Χωρητικότητα	Βέλτιστη Λύση
B-n31-k5	31	100	672
B-n34-k5	34	100	788
B-n35-k5	35	100	955
B-n38-k6	38	100	805
B-n41-k6	41	100	829
B-n45-k6	45	100	678
B-n50-k8	45	100	1313

Τα παραπάνω προβλήματα χρησιμοποιήθηκαν για την επίλυση του CVRP. Οι παρουσιάσεις των λύσεων θα γίνουν παρακάτω.

5.1 Λύσεις Προβλήματος B-n31-k5

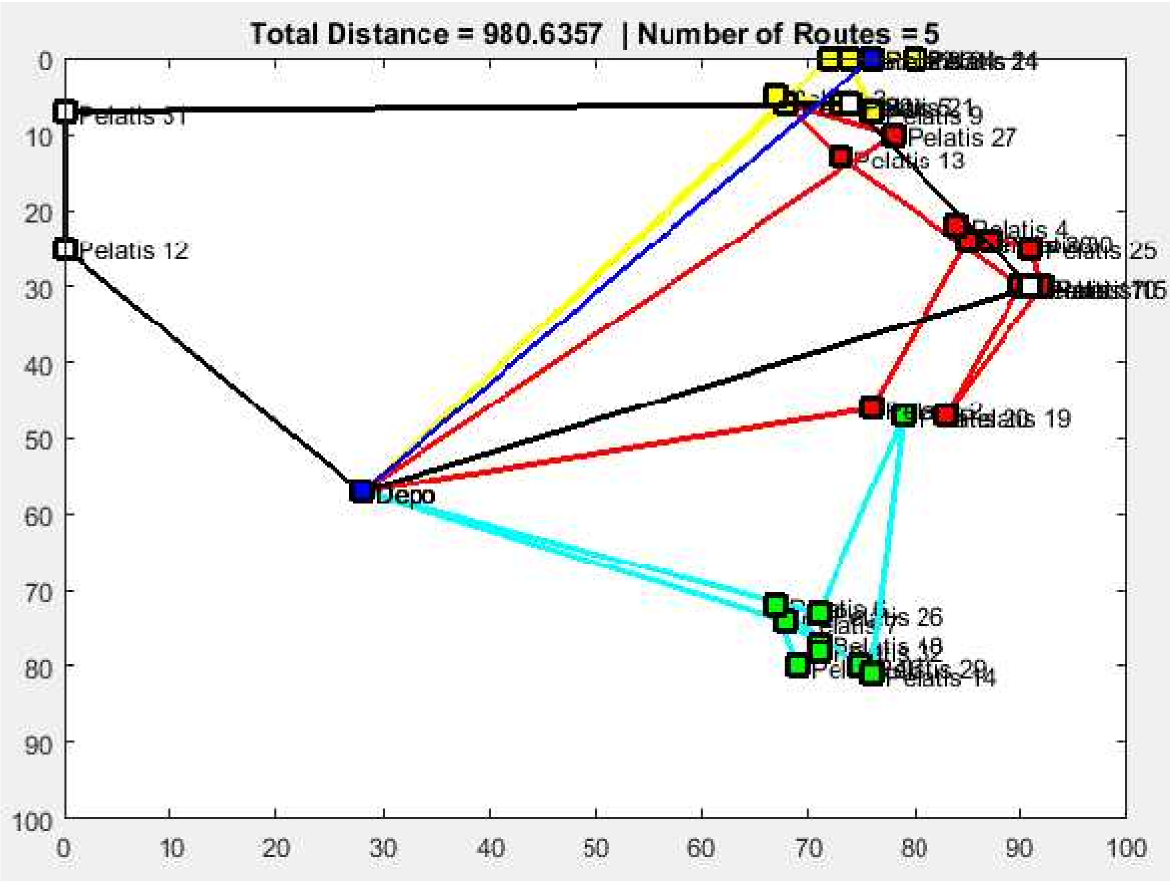
Οι διαδρομές είναι:	**Με κόστη διαδρομών**
[1 17 26 22 6 30 9 13 31 1]	132,47
[1 19 8 5 23 14 29 12 1]	178,85
[1 27 18 24 10 4 7 20 1]	159,28
[1 16 2 15 21 28 11 3 1]	249,63
[1 25 1]	133,76



Εικόνα 5.1: Γράφημα λύσης

5.2 Λύσεις Προβλήματος B-n34-k5

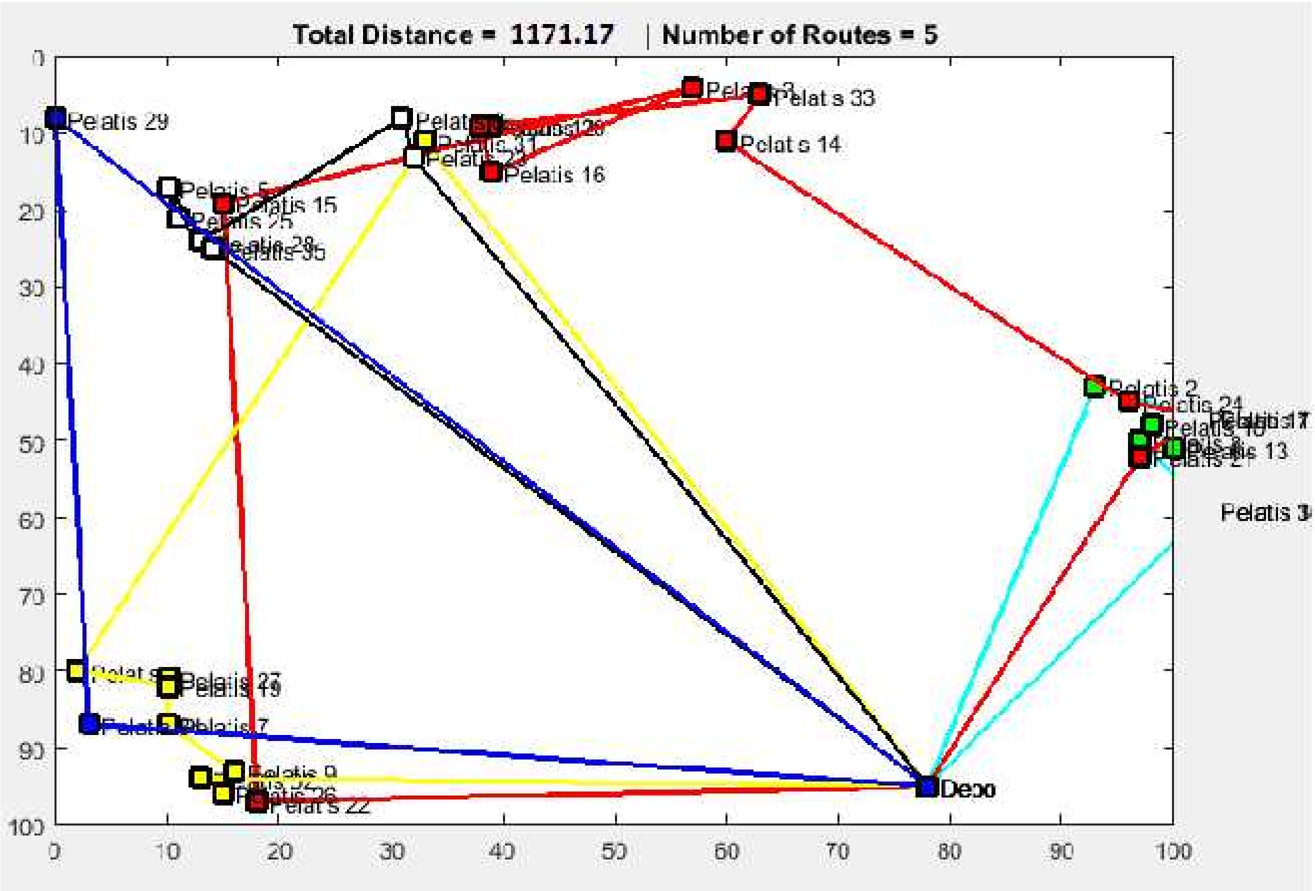
Οι διαδρομές είναι:	**Με κόστη διαδρομών**
[1 7 18 32 16 6 29 14 20 26 1]	179,72
[1 2 28 30 4 25 15 19 17 13 22 27 1]	240,99
[1 8 5 3 9 33 11 24 23 1]	178,40
[1 10 21 31 12 1]	232,48
[1 34 1]	149,03



Εικόνα 5.2: Γράφημα λύσης

5.3 Λύσεις Προβλήματος B-n35-k5

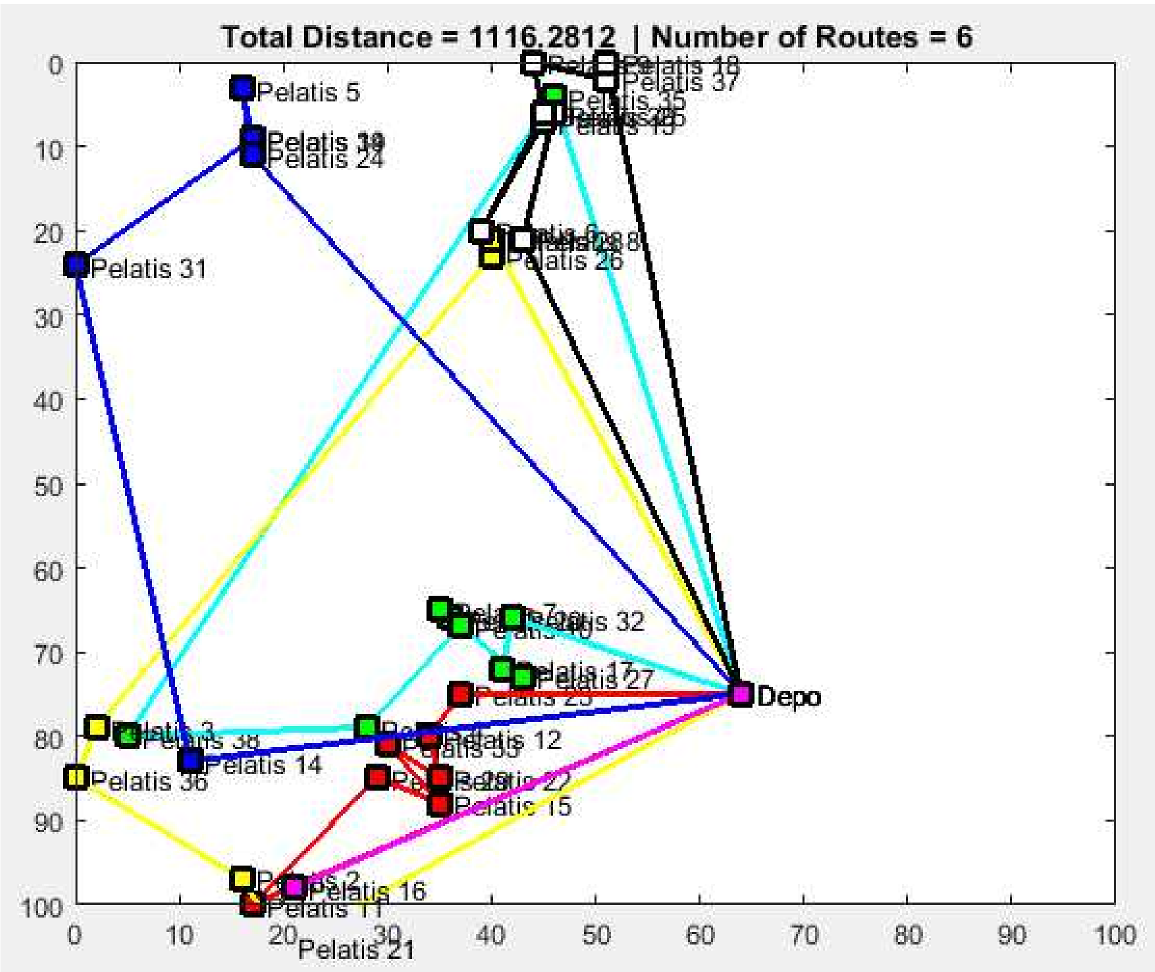
Οι διαδρομές είναι:	**Με κόστη διαδρομών**
[1 18 30 8 10 13 17 2 1]	108,92
[1 21 11 24 14 33 20 12 16 3 15 22 1]	330,52
[1 32 26 9 7 27 19 4 31 1]	245,66
[1 23 6 28 25 5 35 1]	214,72
[1 34 29 1]	271,32



Εικόνα 5.3: Γράφημα λύσης

5.4 Λύσεις Προβλήματος B-n38-k6

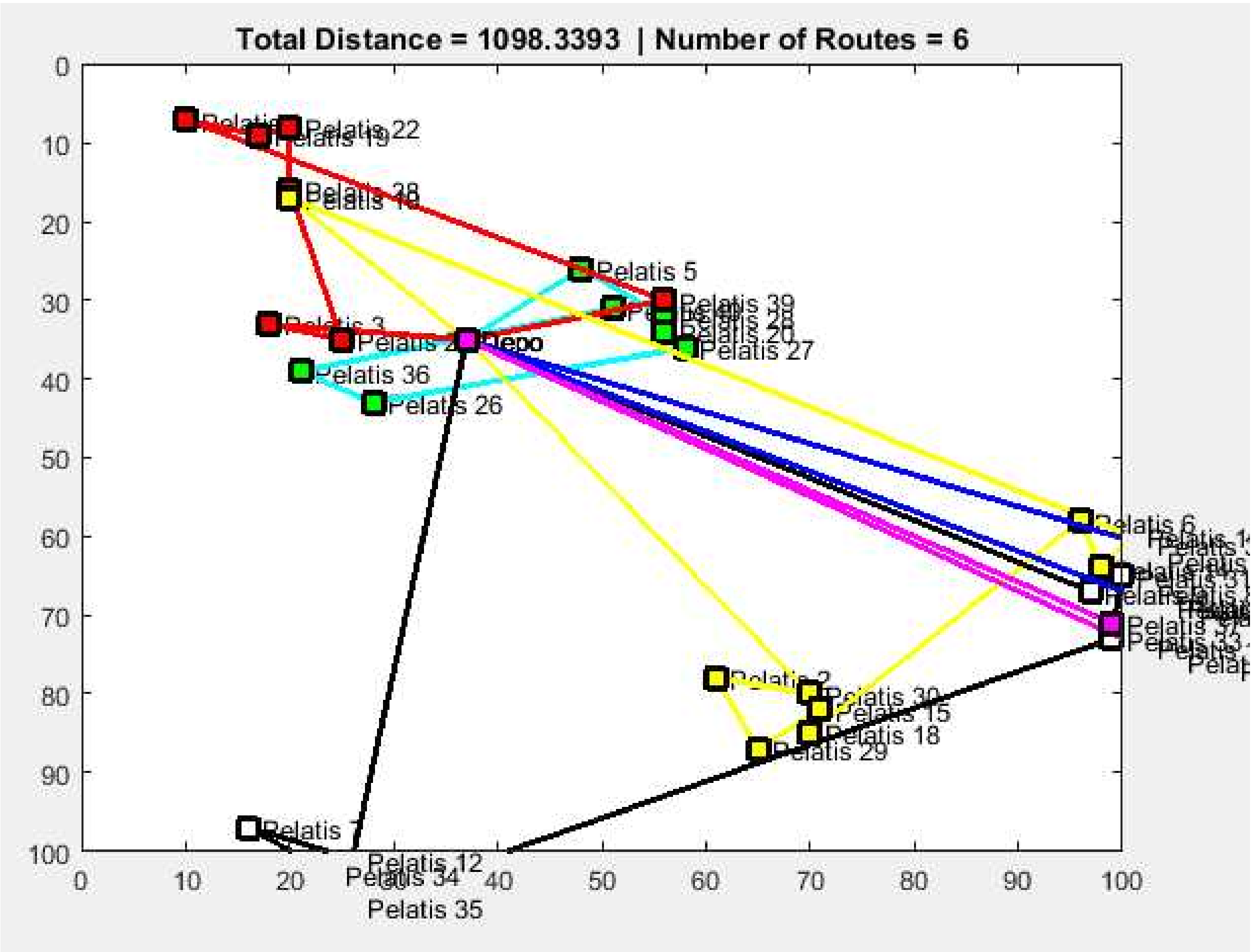
Οι διαδρομές είναι:	Με κόστη διαδρομών
[1 32 17 27 20 7 10 4 38 35 1]	243.85
[1 23 12 22 33 15 29 11 1]	132,08
[1 21 2 36 3 26 28 1]	217.29
[1 8 25 13 6 30 9 37 18 1]	195.68
[1 14 31 19 34 5 24 1]	229.83
[1 16 1]	97.52



Εικόνα 5.4: Γράφημα λύσης

5.5 Λύσεις Προβλήματος B-n41-k6

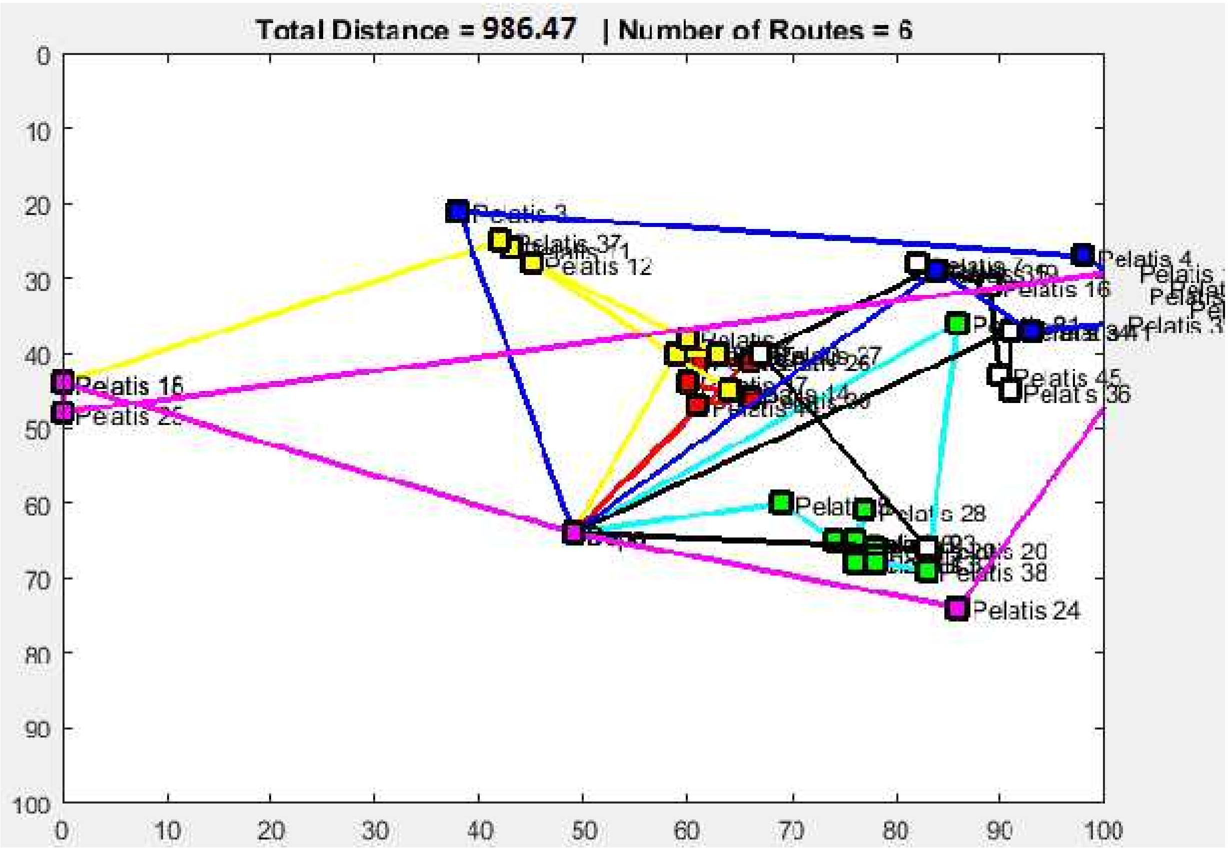
Οι διαδρομές είναι:	**Με κόστη διαδρομών**
[1 5 28 20 27 26 36 40 1]	113,51
[1 3 25 38 22 19 4 39 1]	135.55
[1 30 2 29 15 18 6 14 16 10 1]	251.11
[1 12 7 34 35 33 31 9 1]	252.35
[1 32 8 17 23 21 41 1]	170.02
[1 24 13 11 37 1]	175.77



Εικόνα 5.5: Γράφημα λύσης

5.6 Λύσεις Προβλήματος B-n45-k6

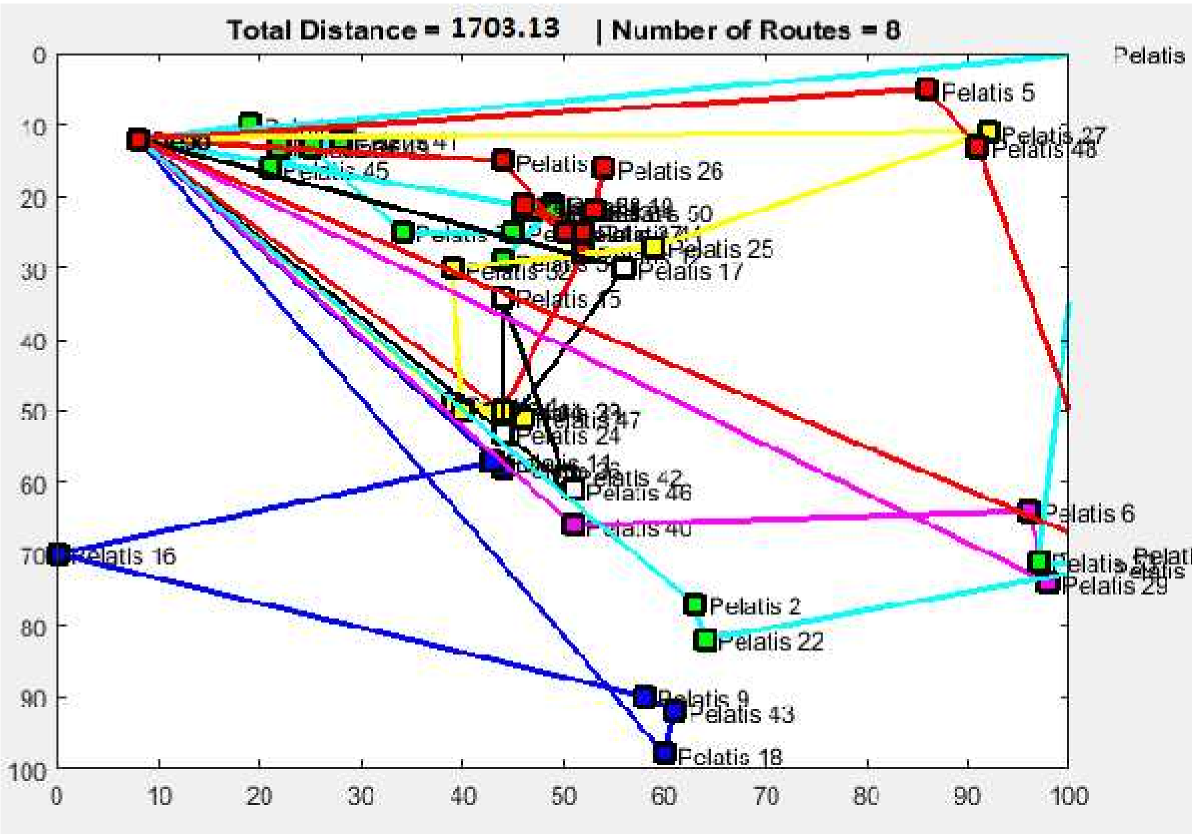
Οι διαδρομές είναι:	**Με κόστη διαδρομών**
[1 5 40 23 9 28 13 29 33 38 8 21 1]	120.12
[1 44 30 17 31 26 1]	68.99
[1 6 43 2 12 14 11 37 18 1]	201.88
[1 19 41 39 22 4 3 1]	180.82
[1 24 42 32 10 25 15 1]	252.39



Εικόνα 5.6: Γράφημα λύσης

5.7 Λύσεις Προβλήματος B-n50-k8

Οι διαδρομές είναι:	Με κόστη διαδρομών
[1 3 35 45 41 49 7 21 28 13 10 31 34 1]	122.45
[1 8 37 38 44 50 26 12 23 1]	157.82
[1 4 39 47 14 32 25 27 1]	212.62
[1 17 24 15 42 46 1]	189.34
[1 36 11 16 9 43 18 1]	246.29
[1 40 6 29 1]	233.56
[1 2 22 30 33 19 1]	281.6
[1 5 48 20 1]	259.46



Εικόνα 5.7: Γράφημα λύσης

Κεφάλαιο 6: Σύγκριση Αποτελεσμάτων και Συμπεράσματα

Στο προηγούμενο κεφάλαιο παρουσιάσαμε τα καλύτερα αποτελέσματα του αλγορίθμου για τα διάφορα προβλήματα αναφοράς. Ακολουθεί η σύγκρισή τους και για τα δύο προβλήματα που εξετάστηκαν, με τις αντίστοιχες παγκόσμιες βέλτιστες τιμές και υπολογίζεται η ποσοστιαία απόκλιση της δικής μας λύσης.

Πρόβλημα	Πελάτες	Αποτέλεσμα Αλγορίθμου Grasp	Παγκοσμίως βέλτιστη τιμή	Απόκλιση από τη βέλτιστη
B-n31-k5	31	853,99	672	27,1%
B-n34-k5	34	980,63	788	24,4%
B-n35-k5	35	1171,17	955	22,6%
B-n38-k6	38	1116,28	805	38,7%
B-n41-k6	41	1098,34	829	32,5%
B-n45-k6	45	986,47	678	45,5%
B-n50-k8	50	1703,13	1313	29,7%

Βλέπουμε ότι οι λύσεις που παράγονται από τον αλγόριθμο δεν είναι θετικές. Είναι αρκετά «μακριά» από τη βέλτιστη τιμή, έτσι ώστε να θεωρούνται έστω ικανοποιητικές.

Αυτό που παρατηρήθηκε είναι ότι αρχικά, ο αλγόριθμος του πλησιέστερου γείτονα, που χρησιμοποιήθηκε για την εύρεση της αρχικής λύσης, δεν παρήγαγε αποτελέσματα που θα μπορούσαν να μας δώσουν καλές λύσεις, γιατί είναι αλγόριθμος ιδιαίτερα άπληστος, με συνέπεια να «χάνει» διαδρομές καλύτερες σε συνολικό κόστος, μόνο και μόνο επειδή επιλέγει να επισκεφθεί αρχικά κοντινότερο πελάτη. Τα δεδομένα, πιθανότατα να είναι έτσι δημιουργημένα, ώστε να επιβάλλεται η απόδραση από τα τοπικά ελάχιστα.

Προσπαθήσαμε να λύσουμε το παραπάνω πρόβλημα με τον αλγόριθμο τυχαιοποιημένης προσαρμοστικότητας (grasp), διαλέγοντας τυχαία έναν από τους 3 κοντινότερους πελάτες από κάθε κόμβο που επισκεπτόμασταν, αλλά δυστυχώς οι λύσεις είχαν εξαρχής αρκετά μεγάλη απόκλιση για να πραγματοποιηθεί η απόδραση από τις άπληστες διαδρομές.

Τέλος, ο αλγόριθμος k-opt, βελτίωσε τις λύσεις, αλλά όχι σε σημείο να θεωρούνται ικανοποιητικές, ενώ παρατηρήθηκε πως σε αρκετά προβλήματα υπήρχαν κάποια σημεία καμπής, δηλαδή πελάτες που είχαν αρκετά ψηλή ζήτηση για να εξυπηρετηθούν σε μία μεγάλη διαδρομή, έτσι κατέληγαν σε μία αρκετά μικρή (από θέμα πελατών) διαδρομή, αλλά ταυτόχρονα πολύ ακριβή. Θα μπορούσαμε να βελτιώσουμε αυτό το φαινόμενο με κάποιους αλγόριθμους τοπικής αναζήτησης όπως ο 1-0 relocate, ο οποίος θα εξέταζε αν χωράει ένας πελάτης από αυτούς να εισέλθει σε μία διαδρομή, με σκοπό τη μείωση του κόστους.

Από τον παραπάνω πίνακα για την σύγκριση των βέλτιστων τιμών του αλγορίθμου, μπορούμε να εξάγουμε τα εξής συμπεράσματα:

- Η κατά μέσο όρο απόκλιση για το πρόβλημα δρομολόγησης οχημάτων με περιορισμένη χωρητικότητα είναι 31,5%.
- Η τυπική απόκλιση είναι 7,57%.

Παρατηρούμε ότι ο αλγόριθμος λειτούργησε λίγο καλύτερα για τα μικρής κλιμακας προβλήματα σε σχέση με τα μεγάλης κλίμακας, αφού σε αυτά παρουσίασε αρκετά μεγαλύτερη απόκλιση.

Κατά τη διάρκεια εφαρμογής του αλγορίθμου για τα διάφορα προβλήματα έγιναν οι παρακάτω παρατηρήσεις:

- Ο αλγόριθμος G.R.A.S.P λειτουργεί καλύτερα όταν αυξάνουμε τις επαναλήψεις του, δηλαδή όταν δημιουργούμε πολλές τυχαίες αρχικές λύσεις, με τη μέθοδο της τυχαιοποιημένη απληστίας.
- Η διατήρηση του αριθμού, μέσω του οποίου τυχαία επιλέγουμε τον επόμενο πελάτη, πρέπει να έχει συγκεκριμένη τιμή και δεν πρέπει να είναι ούτε πολύ μικρός, ούτε πολύ μεγάλος.
- Η αύξηση και η μείωση στο μέγεθος της λίστας περιορισμού, οδηγούσε σε χειρότερη λύση.
- Τέλος, η παρούσα διπλωματική χρειάζεται περαιτέρω βελτιστοποίηση, για να είναι ικανοποιητικές οι λύσεις.

Μελλοντική ενέργεια για βελτίωση της ποιότητας των λύσεων που παράγει ο αλγόριθμος που δημιουργήθηκε, αποτελεί η χρήση άλλων αλγορίθμων τοπικής αναζήτησης (π.χ. του 1-0 relocate ή του 2-2 exchange) ή και η χρήση μεθευρετικών ή εξελικτικών αλγορίθμων.

Βιβλιογραφία

[1]: Μαρινάκης Ι., Μ. Α. (2008). *Σχεδιασμός και Βελτιστοποίηση της Εφοδιαστικής Αλυσίδας.* Θεσσαλονίκη: Σοφία.

[2]: *http://cscmp.org/sites/default/files/user_uploads/resources/downloads/glossary-2013.pdf*. (2013). Retrieved 2016

[3]: Γ., Μουρκούσης. (2008). *Μεθοδολογίες Ανάπτυξης Πληροφοριακών Συστημάτων Υποστήριξης Λήψης.* Patras.

[4]: Toth, P., & Vigo, D. (2002). *The Vehicle Routing Problem.* Society for Industrial and Applied Mathematics.

[5]: Laporte, G., Louveaux, F., & Van Hamme, L. (2002). An integer-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research 50*, 415-423.

[6]: Papadimitrioy , C., & Steiglitz, K. (1982). *Combinatorial Otimization - Aglorithms and Complexity.* New Jersey: Prentice - Hall.

[7]: Resende M.G.C, T. Festa. (1995). "Greedy Randomized Adaptive Search Procedure". *Journal of Global Optimization vol 6*, 109-133.

[8]: Festa P., Resende. M. (2002). *Essays and surveys in metaheuristics.* Springer US.

[9]: C.D.J, Waters. (1987). A solution procdure for the vehicle routing scheduling problem basen on iterative route improvement. *Journal of Operational Research Society*, 833-839.

[10]: S., Lin. (1965). Computer Solutions of the Traveling Salesman Problem. *Technical Journal, 44*, 2245-2269.