



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

**Επίλυση του προβλήματος δρομολόγησης οχημάτων με πολλαπλές  
αποθήκες με χρήση μεθευρετικού αλγορίθμου περιορισμένης  
αναζήτησης**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Πασάλης Δημήτριος**

**Επιβλέπων: Μαρινάκης Ιωάννης**  
**Μέλη Επιτροπής: Σταυρουλάκης Γεώργιος**  
**Αντωνιάδης Αριστομένης**

**ΧΑΝΙΑ 2017**

# ΠΕΡΙΕΧΟΜΕΝΑ

## Περιεχόμενα

Περίληψη.....	4
Κεφάλαιο 1: .....	5
Εισαγωγή .....	5
1.1 Ορισμός Των Logistics .....	5
1.2 Βασικές Αρχές Των Logistics .....	5
1.3 Δραστηριότητες Των Logistics .....	6
1.4 Εφοδιαστική Αλυσίδα.....	8
1.5 Μεταφορά και Διανομές .....	9
Κεφάλαιο 2: .....	11
Είδη Προβλημάτων Δρομολόγησης .....	11
2.1 Πρόβλημα Δρομολόγησης Οχημάτων (VRP) .....	11
2.2 Το Πρόβλημα Δρομολόγησης Οχημάτων Με Πολλαπλές Αποθήκες (MULTIPLE DEPOTS VRP , MDVRP) .....	14
2.3.1 Πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα (VRPTW) .....	17
2.3.2 Το περιορισμένης χωρητικότητας πρόβλημα δρομολόγησης οχημάτων (CVRP).....	17
2.3.3 Πρόβλημα δρομολόγησης οχημάτων χωρίς την επιστροφή στην αποθήκη (OVRP ) .....	18
2.3.4 Στοχαστικά προβλήματα δρομολόγησης οχημάτων (SVRP) .....	18
2.3.5 Το Περιοδικό Πρόβλημα Δρομολόγησης Οχημάτων (PVRP) .....	19
2.3.6 Το δυναμικό πρόβλημα δρομολόγησης οχημάτων (DVRP) .....	19
2.3.7 Πρόβλημα Δρομολόγησης Οχημάτων με Δύο Είδη Πελατών (VRPB) .....	19
2.3.8 Πρόβλημα Δρομολόγησης Οχημάτων με Διανομή και Παραλαβή κατά τη Διάρκεια της Διαδρομής (VRPPD) 20	
Κεφάλαιο 3: .....	21
Αλγόριθμοι Βελτιστοποίησης Προβλημάτων.....	21
3.1 Εισαγωγή .....	21
3.2 Απλοί Ευρετικοί Αλγόριθμοι (Heuristics) .....	21
3.2.1 Αλγόριθμοι Απληστίας .....	22
3.2.2 Προσεγγιστικοί Αλγόριθμοι.....	23
3.2.3 Αλγόριθμοι Τοπικής Αναζήτησης .....	24
3.3 Μεθευρετικές Μέθοδοι .....	26
3.3.1 Αλγόριθμος Περιορισμένης Αναζήτησης ( Tabu Search) .....	28
Κεφάλαιο 4: .....	33
Περιγραφή και τρόπος επίλυσης του Προβλήματος .....	33

4.1 Συνοπτική περιγραφή προβλήματος .....	33
4.2 Μοντελοποίηση του προβλήματος .....	34
4.2 Εύρεση αρχικής εφικτής λύσης με τον αλγόριθμο πλησιέστερου γείτονα .....	39
4.3 Εύρεση της βέλτιστης λύσης με τη χρήση του αλγορίθμου περιορισμένης αναζήτησης .....	40
4.3.1 Ανταλλαγή μεταξύ κόμβων διαφορετικών διαδρομών .....	41
4.3.2 Ελαχιστοποίηση Αριθμού διαδρομών.....	42
4.3.3 Ανταλλαγή μεταξύ κόμβων τυχαίων διαδρομών.....	44
4.3.4 Βελτιστοποίηση κόμβων εντός της ίδιας διαδρομής.....	44
Κεφάλαιο 5: .....	46
Περιγραφή και ανάλυση των αποτελεσμάτων .....	46
5.1 Γενική περιγραφή .....	46
5.2 Αναλυτικά αποτελέσματα των προβλημάτων .....	47
Κεφάλαιο 6: .....	68
Τελικά συμπεράσματα .....	68
6.1 Σύγκριση αποτελεσμάτων.....	68
6.2 Ανάλυση αποτελεσμάτων και συμπεράσματα.....	68
Βιβλιογραφία.....	70

## Περίληψη

Η διανομή των προϊόντων από τις αποθήκες στους πελάτες, είναι ένα πρακτικό και απαιτητικό πρόβλημα της εφοδιαστικής αλυσίδας. Καλύτερες αποφάσεις κατά τα στάδια της δρομολόγησης μπορούν να οδηγήσουν σε μεγαλύτερα επίπεδα ικανοποίησης πελατών, καθώς περισσότεροι πελάτες μπορούν να εξυπηρετηθούν σε συντομότερο χρόνο και μείωση του κόστους διανομής.

Το πρόβλημα της διανομής γενικά μοντελοποιείται σαν το πρόβλημα της δρομολόγησης οχημάτων (Vehicle Routing Problem – VRP), στο οποίο γίνεται η υπόθεση ότι η εταιρεία εφοδιασμού έχει μια αποθήκη. Όμως σε περιπτώσεις όπου η εταιρεία εφοδιασμού έχει παραπάνω αποθήκες το VRP δεν είναι κατάλληλο. Για να αντιμετωπισθεί αυτός ο περιορισμός, η παρούσα εργασία επικεντρώνεται στην επίλυση του προβλήματος δρομολόγησης οχημάτων με πολλαπλές αποθήκες (Multi-Depot VRP – MDVRP). Το MDVRP αποτελεί ένα NP-δύσκολο πρόβλημα, που σημαίνει ότι δεν υπάρχει κάποιος αποτελεσματικός αλγόριθμος για την βέλτιστη επίλυσή του.

Κατά την εκπόνηση της παρούσας εργασίας αναπτύσσεται ένας μεθευρετικός αλγόριθμος περιορισμένης αναζήτησης, που προσπαθεί να βελτιώσει την αρχική λύση του ευρετικού αλγόριθμου του κοντινότερου γείτονα, για την επίλυση του προβλήματος. Η αποτελεσματικότητα του προτεινόμενου αλγόριθμου δοκιμάζεται μέσω υπολογιστικών μελετών σε προβλήματα διαφορετικών μεγεθών. Τέλος, τα αποτελέσματα που δίδει ο προτεινόμενος αλγόριθμος συγκρίνονται με τις καλύτερες λύσεις που υπάρχουν στη βιβλιογραφία και δίδονται προτάσεις για περαιτέρω βελτίωση του.

# Κεφάλαιο 1:

## Εισαγωγή

### 1.1 Ορισμός Των Logistics

Ο όρος logistics αποτελεί πολυσήμαντη και πολυσύνθετη έννοια, καλύπτοντας μια τεράστια μάζα διαδικασιών σχεδιασμού, υλοποίησης και ελέγχου στο επιχειρηματικό πεδίο. Τα βασικά στοιχεία που απαρτίζουν τα logistics είναι η διοίκηση και ο στρατηγικός σχεδιασμός της επιχείρησης, η βέλτιστη αξιοποίηση των πόρων της, η παραγωγή, η αποθήκευση και η διανομή των αγαθών, από την πρώτη ύλη μέχρι το έτοιμο προϊόν και από την παραγωγή στο ράφι. Θεωρητικά τα logistics εξυπηρετούν την κερδοφορία μιας επιχείρησης, εξασφαλίζοντας τη συνεχή διαθεσιμότητα των προϊόντων και των λοιπών πόρων της, επιτρέποντας παράλληλα την ομαλή ροή επιτέλεσης των διαδικασιών που αναφέρθηκαν παραπάνω.

Τα logistics αποσκοπούν στην παραγωγή προϊόντων με όσο το δυνατόν χαμηλότερο κόστος, στη διατήρηση των προϊόντων με τον καλύτερο δυνατό τρόπο, στην πλήρη αξιοποίηση των υλικών μέσων της επιχείρησης, στη μεταφορά των προϊόντων με το χαμηλότερο δυνατό κόστος και τις μικρότερες δυνατές καθυστερήσεις και τελικά στην επίτευξη κερδοφορίας και οικονομίας για την επιχείρηση.

### 1.2 Βασικές Αρχές Των Logistics

Τα logistics αποτελούν ένα σύνθετο σύστημα, το οποίο περιλαμβάνει επιμέρους διαδικασίες, που με τη σειρά του αποτελούνται από δραστηριότητες και στοιχεία. Οι βασικότεροι στόχοι του συστήματος κατά τη λειτουργία του, είναι οι εξής:

- Ελαχιστοποίηση του λειτουργικού κόστους
- Ελάχιστο δυνατό κόστος επενδύσεων
- Βελτιστοποίηση της ποιότητας προϊόντων και υπηρεσιών
- Υψηλό επίπεδο εξυπηρέτησης πελατών

Το κόστος των δραστηριοτήτων των logistics, αποτελεί ένα πολύ σημαντικό μέρος του συνολικού κόστους των προϊόντων, όπως αυτό έχει διαμορφωθεί όταν φτάνει στα χέρια των καταναλωτών.

Το κόστος των δραστηριοτήτων logistics, περιλαμβάνει:

- Δαπάνες μεταφορών
- Δαπάνες αποθήκευσης και συσκευασίας
- Κόστος διατήρησης αποθέματος
- Κόστος φυσικής διακίνησης υλικών και πληροφοριών
- Κόστος ελέγχου και απογραφής των αποθεμάτων
- Κόστος κτιρίων και μηχανημάτων και οι αποσβέσεις τους

Πρέπει να σημειωθεί πως είναι σημαντικό, στην προσπάθεια ελαχιστοποίησης του κόστους, να λαμβάνεται υπόψη το συνολικό κόστος και όχι το επιμέρους κόστος των διαφόρων δραστηριοτήτων, καθώς η μείωση του κόστους σε ένα τομέα μπορεί να επιφέρει την αύξηση του κόστους σε έναν άλλο ή ακόμα και να δημιουργήσει προβλήματα στην ποιότητα των προσφερόμενων υπηρεσιών, κυρίως λόγω μειωμένης αποδοτικότητας.

### 1.3 Δραστηριότητες Των Logistics

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

**Αγορές – Προμήθειες :** Ο όρος «αγορές» αναφέρεται στην απόκτηση προϊόντων, τα οποία μπορεί να ενσωματωθούν στο τελικό προϊόν, να είναι πρώτες ύλες, υλικά συσκευασίας, καύσιμα, κλπ, αλλά και υπηρεσιών. Οι ποσότητες αγοράς επηρεάζονται σε μεγάλο βαθμό από το επίπεδο εξυπηρέτησης των πελατών το οποίο έχει ορίσει η επιχείρηση. Παράλληλα, οι τιμές δεν αποτελούν πλέον το κύριο κριτήριο που καθορίζει τις αγορές, αντιθέτως απαιτείται η συνεργασία προμηθευτών, βιομηχανίας και αγοραστών. Η επιλογή των προμηθευτών των προϊόντων και των υπηρεσιών, αποτελεί ζήτημα νευραλγικής σημασίας για μια επιχείρηση, η οποία πρέπει να οργανώνει τόσο την επιλογή τους όσο και την αξιολόγησή τους, για την οποία χρησιμοποιούνται τα κριτήρια της συνέπειας στην τήρηση των

χρόνων μεταφορών και παράδοσης και της ταχύτητας που μπορεί να ανταποκριθεί ο προμηθευτής στη εποχικότητα ή σε μια αλλαγή της ζήτησης.

**Αποθήκευση:** Ο χώρος της αποθήκης αποτελεί ένα σημαντικό κομμάτι των logistics, η λειτουργία του οποίου οργανώνεται με τις πλέον εκσυγχρονισμένες μεθόδους. Η λειτουργία μιας αποθήκης απαιτεί τη σωστή οργάνωσή της, με τη χρήση εξοπλισμού ραφιών, ειδικών ραμπών, μέσων για τη μετακίνηση των προϊόντων, μηχανογραφικά συστήματα για την καλύτερη οργάνωση και εκμετάλλευση του χώρου και τέλος ρομποτικά συστήματα για εξειδικευμένες εργασίες. Μια αποθήκη είναι υπεύθυνη για τη διεκπεραίωση των εξής ενεργειών: την παραλαβή των προμηθευόμενων εμπορευμάτων, την τοποθέτησή τους στην αποθήκη, τη φύλαξη και διατήρησή τους, τη γρήγορη ανεύρεση και κυκλοφορία τους, την παρακολούθηση των εμπορευμάτων που παραμένουν στην αποθήκη ανά κωδικούς, παραγγελία, είδος, κλπ. και την προετοιμασία των εμπορευμάτων για τη διανομή τους.

**Αποθέματα:** Ο όρος «διαχείριση αποθεμάτων» αναφέρεται στο σωστό επίπεδο προϊόντων που μια επιχείρηση πρέπει να διαθέτει έτσι ώστε να μπορεί να ανταποκριθεί σε όλες τις υποχρεώσεις της. Δύο είναι τα ερωτήματα, που πρέπει να απαντηθούν σωστά, ώστε να επιτευχθεί το επιθυμητό αυτό επίπεδο: πόσες μονάδες από κάθε προϊόν που χρειάζεται, πρέπει να παραγγείλει η επιχείρηση και σε ποια χρονική στιγμή πρέπει να γίνουν οι παραγγελίες αυτές. Ο επιθυμητός στόχος είναι η εξασφάλιση του επιθυμητού επιπέδου εξυπηρέτησης πελατών με τον ελάχιστο δυνατό αριθμό αποθεμάτων, με συνέπεια τη μείωση του άμεσου και του έμμεσου κόστους τους.

**Μεταφορές:** οι μεταφορές αναφέρονται στην εύρεση του αποδοτικότερου τρόπου διακίνησης των προϊόντων από και προς την επιχείρηση. 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 ποιος είναι ο καταλληλότερος μεταφορέας. Ταυτόχρονα, ένα πολύ σημαντικό κριτήριο για τη σωστή επιλογή του τρόπου μεταφοράς είναι η χρονική διάρκεια, ώστε τα προϊόντα να παραδίδονται ακριβώς στην ώρα που επιθυμεί ο πελάτης, χωρίς καθυστερήσεις.

**Διανομές:** όπως και οι μεταφορές, οι διανομές αναφέρονται στη μεταφορά από τις αποθήκες ή τις εγκαταστάσεις μιας επιχείρησης προς τους πελάτες. Η διαφορά τους έγκειται στο γεγονός ότι οι μεταφορές αναφέρονται στη διακίνηση λίγων προϊόντων σε μεγάλες ποσότητες σε μικρό αριθμό

πελατών, ενώ οι διανομές αναφέρονται στη διακίνηση πολλών προϊόντων, σε μικρές ποσότητες και μεγάλο αριθμό πελατών.

Τέλος, πρέπει να αναφερθεί πως σημαντικό κομμάτι του συστήματος logistics αποτελεί και η 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.

## **1.4 Εφοδιαστική Αλυσίδα**

Με τον όρο Εφοδιαστική Αλυσίδα εννοούμε τη ροή υλικών και υπηρεσιών από τον προμηθευτή πρώτων υλών ή κατασκευαστή μέχρι τον καταναλωτή, και παράλληλα την ροή πληροφοριών μεταξύ των μελών της ίδιας αλυσίδας και σύμφωνα με τις ανάγκες των πελατών. Αποτελείται από όλα τα στάδια και όλους τους εμπλεκόμενους από τη στιγμή που προμηθεύονται οι πρώτες ύλες, μέχρι τη στιγμή που το τελικό προϊόν θα φτάσει στον πελάτη ή καταναλωτή. Δηλαδή, η εφοδιαστική αλυσίδα αποτελείται από τους κατασκευαστές, και προμηθευτές υλικών, τους χώρους αποθήκευσης, τα κέντρα διανομών, τους μεταφορείς, τους πωλητές, τους πελάτες, τις πρώτες ύλες, αλλά και έτοιμα προϊόντα που μπορεί να υπάρχουν ανάμεσα. Η εφοδιαστική αλυσίδα όμως δεν είναι στατική, αλλά δυναμική και μπορεί να περιλαμβάνει και ενδο-επιχειρησιακές δραστηριότητες ή και μία αλυσίδα επιχειρήσεων, τους προμηθευτές των προμηθευτών και εξωτερικούς συνεργάτες.

Βασική επιδίωξη της εφοδιαστικής αλυσίδας είναι η ικανοποίηση του πελάτη. Όλα τα στάδια και οι ροές προϊόντων υλικών ή πληροφοριών που περιλαμβάνει η εφοδιαστική αλυσίδα δημιουργούν δαπάνες οι οποίες προσαυξάνονται και στο τελικό προϊόν. Μοναδική πηγή εσόδων είναι ο πελάτης, οπότε είναι καθοριστικής σημασίας να μείνει ικανοποιημένος από το τελικό προϊόν που θα φτάσει στα χέρια του. Για αυτό και είναι απαραίτητη η βέλτιστη διαχείριση της εφοδιαστικής αλυσίδας για να παραλάβει το τελικό προϊόν ταχύτερα ο πελάτης, στην καλύτερη τιμή, παρέχοντας του την καλύτερη εξυπηρέτηση και με εμπεριεχόμενες τις νέες τεχνολογίες. Κύριος σκοπός είναι η ελαχιστοποίηση του συνολικού κόστους που προκύπτει εντός της αλυσίδας και σε όλα τα στάδια της, όπως τα έξοδα των μεταφορών, των διανομών κ.λπ. με αποτέλεσμα την τελική αύξηση των κερδών της επιχείρησης.



## 1.5 Μεταφορά και Διανομές

Η δραστηριότητα της μεταφοράς και διανομής αφορά τον τρόπο και τα μέσα για την μεταβίβαση υλικών και αγαθών ή υπηρεσιών μέσω των φυσικών καναλιών. Για να ολοκληρωθεί η μεταφορά πρέπει να γίνει επιλογή του μέσου, του τύπου οχήματος, ο καθορισμός των διαδρομών και των δρομολογήσεων, η επιλογή των μεταφορέων και αλλά. Τα έξοδα των μεταφορών αποτελούν συνήθως το μεγαλύτερο κομμάτι της πίτας των εξόδων της εφοδιαστικής αλυσίδας, άρα απορροφούν και μεγάλο κόστος από την επιχείρηση. Η εφοδιαστική διανομή ( Distribution logistics) είναι εκείνος ο τομέας της διαχείρισης της εφοδιαστικής αλυσίδας ο οποίος είναι υπεύθυνος για την παράδοση των προϊόντων στους πελάτες στον σωστό τόπο, στην σωστή χρονική στιγμή, σε σωστή κατάσταση και για το σωστό κόστος.

Οι μεταφορές χωρίζονται σε δύο μέρη:

1. Σε εσωτερικές, που περιλαμβάνουν τόσο μεταφορά πρώτων υλών από τις πηγές προς τα εργοστάσια όσο και μερών των τελικών προϊόντων ανάμεσα σε διάφορα της εταιρίας ή από το εργοστάσιο στην αποθήκη και στα σημεία πωλήσεων ( Inbound logistics)
2. Σε εξωτερικές μεταφορές που περιλαμβάνουν μεταφορά των τελικών προϊόντων από τις αποθήκες στους πελάτες άμεσα ή διαμέσου κέντρων διανομής (Outbound logistics).

Προβλήματα στον σχεδιασμό των μεταφορών περιλαμβάνουν την επιλογή του στόλου μεταφοράς (μέγεθος του στόλου και χρήση διαφορετικών τύπων οχημάτων), τη δρομολόγηση οχημάτων ( επιλογή των βέλτιστων διαδρομών, λαμβάνοντας υπόψη τη δομή του δικτύου, τις αποστάσεις και τη χωρητικότητα των διαδρομών), το σχεδιασμό του δικτύου διανομής ( βελτίωση δρομολογίων, του χρονοπρογραμματισμού των δρομολογίων και επιλογή διαφόρων ενδιάμεσων αποθηκών) και την επιλογή του προσωπικού που θα πραγματοποιήσει τις διανομές ( καθορισμός απαιτήσεων προσωπικού). Μερικά από τα λειτουργικά προβλήματα που περιλαμβάνονται στον όρο μεταφορά είναι του ελέγχου της διαδικασίας αποστολής των χρονοπρογραμματισμός των πληρωμάτων πραγματοποιήσουν τις μεταφορές.

Ένα σύστημα μεταφοράς μπορεί να αναπαρασταθεί με την μορφή δικτύου κόμβων και τόξων, όπου οι κόμβοι τυπικά αντιπροσωπεύουν πόλεις, αεροδρόμια, στάσεις και αποθήκες και τα τόξα αντιπροσωπεύουν τους συνδέσμους ή τις διαδρομές μεταξύ των κόμβων. Οι κόμβοι και τα τόξα μπορεί να έχουν περιορισμούς χωρητικότητας. Διαφορετικοί τρόποι είναι συνήθως διαθέσιμοι για την μεταφορά των φορτίων μεταξύ δύο σημείων. Υπάρχουν πέντε βασικοί τρόποι μεταφοράς, οι σιδηροδρομικοί, οδικοί, θαλάσσιοι, αεροπορικοί και αγωγοί μεταφορών.

## Κεφάλαιο 2:

### Είδη Προβλημάτων Δρομολόγησης

#### 2.1 Πρόβλημα Δρομολόγησης Οχημάτων (VRP)

Το Πρόβλημα Δρομολόγησης Οχημάτων (Vehicle Routing Problem) είναι ένα από τα πιο μελετημένα υπολογιστικά προβλήματα βελτιστοποίησης, το πρόβλημα εξετάζει την βέλτιστη δρομολόγηση οχημάτων κατά την παράδοση ή/και παραλαβή προϊόντων στους/από τους πελάτες, σε δεδομένη χρονική περίοδο, στα πλαίσια επιχειρησιακών δραστηριοτήτων.

Η θεωρητική έρευνα και πρακτική εφαρμογή ξεκίνησε το 1959 από τους Dantzig και Ramser με το πρόβλημα «αποστολής φορτηγών», χρησιμοποιώντας μέθοδο βασισμένη σε Γραμμικό προγραμματισμό και υπολογισμούς με το χέρι κατέληξαν σε μία σχεδόν βέλτιστη λύση.

Η σωστή επίλυση ενός προβλήματος δρομολόγησης οχημάτων έχει σαν αποτέλεσμα τον καθορισμό ενός συνόλου από διαδρομές, με ελάχιστο ολικό κόστος, με σκοπό την εξυπηρέτηση ενός πλήθους πελατών γεωγραφικά διασκορπισμένων, χωρίς να παραβιαστεί κάποιος περιορισμός. Τα οχήματα ξεκινάνε από μία ή περισσότερες αποθήκες και έχουν την δυνατότητα να επιστρέψουν σε αυτές. Υπάρχουν πολλές μεταβλητές, οι οποίες δημιουργούν παραλλαγές στο αρχικό πρόβλημα.

Χαρακτηριστικά των πελατών:

- Τη γεωγραφική θέση του πελάτη, άρα το σημείο του γραφήματος στο οποίο βρίσκεται (road graph).
- Την ποσότητα των αγαθών (demand) τα οποία πρέπει είτε να παραδοθούν, είτε να συλλεχθούν από τον πελάτη.
- Τα χρονικά διαστήματα (time windows) κατά τη διάρκεια της ημέρας, μέσα στα οποία ο πελάτης μπορεί να εξυπηρετηθεί.

- Χρόνος που απαιτείται για την εξυπηρέτηση κάθε πελάτη ( service time).Η παράμετρος μπορεί να αφορά χρόνο φορτοεκφόρτωσης (unloading or loading time).

Χαρακτηριστικά οχημάτων:

- Από ποια αποθήκη προέρχονται και αν υπάρχει η δυνατότητα να τερματίσουν σε άλλη αποθήκη από αυτή που ξεκίνησαν ή να μην επιστρέψουν κάπου.
- Η χωρητικότητα του οχήματος.
- Αν τα οχήματα είναι χωρισμένα σε τμήματα και πώς αυτά θα φορτωθούν.
- Αν τα οχήματα διαφέρουν μεταξύ τους, ανάλογα με την χωρητικότητα και το είδος των αγαθών που μεταφέρουν.
- Το σύνολο των δρόμων που είναι προσπελάσιμοι από το όχημα.
- Το κόστος που συσχετίζεται με την λειτουργία του κάθε οχήματος.

Μερικά χαρακτηριστικά των οδηγών, κάποια από τα οποία μπορεί να είναι επιβεβλημένα για την καλή υγεία τους και πρέπει να τηρηθούν.

- Οκτάωρο ύπνου την ημέρα
- Όχι παραπάνω από 10 ώρες συνεχής οδήγησης
- Όχι παραπάνω από 6 μέρες την εβδομάδα
- Όχι παραπάνω από 15 ώρες την ημέρα οδήγησης

Ανάλογα με το πλήθος και το είδος των περιορισμών, τροποποιείται και το πρόβλημα δρομολόγησης, οι πιο βασικοί που συναντάμε στην βιβλιογραφία είναι:

1. Η συνολική ποσότητα που μεταφέρει το όχημα δεν πρέπει να ξεπερνάει τη χωρητικότητα του.
2. Υπάρχουν πελάτες που ζητούν παράδοση ή παραλαβή ή και τα δύο
3. Τα χρονικά παράθυρα εξυπηρέτησης που θέλουν οι πελάτες
4. Οι οδηγοί μπορεί να δουλεύουν κάποια συγκεκριμένη χρονική περίοδο.

5. Τα οχήματα μπορεί να μεταφέρουν παραπάνω από ένα προϊόντα.
6. Οι πελάτες να πρέπει να εξυπηρετηθούν με συγκεκριμένη σειρά.

Από τα παραπάνω χαρακτηριστικά προκύπτουν και συγκεκριμένοι περιορισμοί που πρέπει να ληφθούν υπ' όψιν κατά την υλοποίηση και την αναζήτηση της βέλτιστης λύσης, ώστε η τελική λύση να είναι εφικτή και πραγματοποιήσιμη. Οι στόχοι ορίζονται από την εκάστοτε επιχείρηση ανάλογα με το τι θέλει να επιτύχει, τις απαιτήσεις των πελατών, σε ποιο αγοραστικό κοινό θέλει να απευθυνθεί και τις προτεραιότητες που η ίδια βάζει. Οι στόχοι της επίλυσης προβλημάτων δρομολόγησης οχημάτων είναι οι εξής:

1. Η ελαχιστοποίηση του συνολικού κόστος μεταφοράς των προϊόντων. Το συνολικό κόστος εξαρτάται από τη συνολική απόσταση ή τον συνολικό χρόνο που απαιτείται και από το πάγιο κόστος ανά όχημα ή/και οδηγό.
2. Η ελαχιστοποίηση του αριθμού των οχημάτων .
3. Η ισορροπία μεταξύ των διαδρομών σε σχέση με τις απαιτούμενες ώρες που χρειάζονται.
4. Η ελαχιστοποίηση των ποινών που πηγάζουν από την μερική εξυπηρέτηση πελατών.

Τα προβλήματα δρομολόγησης οχημάτων ανήκουν στην κατηγορία των NP Hard προβλημάτων της συνδυαστικής βελτιστοποίησης της επιχειρησιακής έρευνας. Ανάλογα με τους περιορισμούς που περιγράφηκαν παραπάνω, υπάρχουν διάφορες παραλλαγές και θα γίνει μία αναφορά στα επόμενα κεφάλαια. Στην παρούσα διπλωματική εργασία θα μας απασχολήσει το πρόβλημα δρομολόγησης οχημάτων με πολλαπλές αποθήκες. Για την επίλυση των VRP προβλημάτων έχουν αναπτυχθεί διάφοροι αλγόριθμοι που έχουν σκοπό να βελτιστοποιήσουν το τελικό αποτέλεσμα και θα παρουσιαστούν σε επόμενο κεφάλαιο.

## **2.2 Το Πρόβλημα Δρομολόγησης Οχημάτων Με Πολλαπλές Αποθήκες (MULTIPLE DEPOTS VRP , MDVRP)**

Στο πρόβλημα δρομολόγησης οχημάτων με πολλαπλές αποθήκες διακρίνουμε δύο κύριες κατηγορίες, αναλόγως με την κατανομή των οχημάτων και της ζήτησης.

Στην πρώτη κατηγορία, στόλος των οχημάτων, αλλά και οι γεωγραφικά διασκορπισμένοι πελάτες είναι κατανεμημένοι σε περισσότερες της μιας αποθήκες. Έτσι κάθε όχημα εξυπηρετεί την ζήτηση, που έχει ανατεθεί στην εκάστοτε αποθήκη, με την οποία είναι συνδεδεμένο. Κάθε όχημα λοιπόν καταλήγει στην αποθήκη από την οποία έχει ξεκινήσει. Το πρόβλημα δρομολόγησης οχημάτων με πολλαπλές αποθήκες μπορεί υπό αυτές τις προϋποθέσεις να θεωρηθεί ως πολλαπλά προβλήματα δρομολόγησης οχημάτων γενικής μορφής.

Στη δεύτερη κατηγορία, οι γεωγραφικά διασκορπισμένοι πελάτες είναι κατανεμημένοι σε περισσότερες της μιας αποθήκες, ενώ ο στόλος των οχημάτων είναι κοινός για όλες τις αποθήκες. Τα οχήματα λοιπόν δεν καταλήγουν αναγκαστικά στην αποθήκη από την οποία ξεκίνησαν. Οι πελάτες εξακολουθούν να εξυπηρετούνται αποκλειστικά από συγκεκριμένες αποθήκες, με τη διαφορά όμως ότι ο στόλος των οχημάτων που απασχολεί κάθε αποθήκη δεν είναι σταθερός, καθ' όλες τις περιόδους εξυπηρέτησης, αλλά μεταβάλλεται προκειμένου να καλύψει την ζήτηση των πελατών της.

Σκοπός του προβλήματος είναι να εξυπηρετηθούν όλοι οι πελάτες ενώ υπάρχουν πάνω από μια αποθήκες, ελαχιστοποιώντας παράλληλα τον αριθμό των χρησιμοποιούμενων οχημάτων και τα συνολικά χιλιόμετρα που διανύονται.

## Μοντέλο MDVRP

### Δείκτες

$i$ : Δείκτης Πελάτη

$j$ : Δείκτης Πελάτη

$k$ : Δείκτης Οχήματος

$d$ : Δείκτης Αποθήκης

### Σταθερές

$C_{ij}$ : Κόστος μεταφοράς από τον πελάτη  $i$  στον πελάτη  $j$

$C'_{di}$ : Κόστος μεταφοράς από την αποθήκη  $d$  στον πελάτη  $i$

$d_i$ : Ζήτηση του πελάτη  $j$

$N$ : Σύνολο πελατών

$B$ : Υποσύνολο πελατών

$Q_k$ : Χωρητικότητα οχήματος  $k$

$V$ : Σύνολο Οχημάτων

$D$ : Σύνολο αποθηκών

$G$ : Σύνολο πελατών και Αποθηκών

$M$ : Άπειρο (Μεγάλος Θετικός Αριθμός)

### Μεταβλητές

$X_{ijk}$ : 1 εάν το όχημα  $k$  πηγαίνει από τον πελάτη  $i$  στον πελάτη  $j$ , αλλιώς 0

$Y_{dj k}$ : 1 εάν το όχημα  $k$  πηγαίνει από την αποθήκη  $d$  στον πελάτη  $j$ , αλλιώς 0

$Z_{id k}$ : 1 εάν το όχημα  $k$  πηγαίνει από τον πελάτη  $i$  στην αποθήκη  $d$ , αλλιώς 0

### Αντικειμενική Συνάρτηση

$$\text{Min} \sum_{i=1}^N \sum_{j=1, j \neq i}^N \sum_{k=1}^V X_{ijk} C_{ij} + \sum_{d=1}^D \sum_{i=1}^N \sum_{k=1}^V Y_{id k} C'_{di} + \sum_{i=1}^N \sum_{d=1}^D \sum_{k=1}^V Z_{id k} C'_{di}$$

### Υπό περιορισμούς

$$\sum_{d=1}^D \sum_{k=1}^V Y_{djk} + \sum_{\substack{j=1 \\ j \neq i}}^N \sum_{k=1}^V X_{ijk} = \mathbf{1} \quad \forall i$$

$$\sum_{d=1}^D \sum_{k=1}^V Z_{djk} + \sum_{\substack{j=1 \\ j \neq i}}^N \sum_{k=1}^V X_{ijk} = \mathbf{1} \quad \forall i$$

$$\sum_{d=1}^D \sum_{i=1}^N Y_{djk} - \sum_{d=1}^D \sum_{k=1}^V Z_{djk} = \mathbf{0} \quad \forall k$$

$$\sum_{d=1}^D \sum_{i=1}^N Y_{djk} d_i + \sum_{\substack{j=1 \\ j \neq i}}^N \sum_{i=1}^N X_{ijk} d_j \leq \mathbf{0} \quad \forall k$$

$$\sum_{d=1}^D Y_{djk} - \sum_{\substack{j=1 \\ j \neq i}}^N X_{ijk} + \sum_{\substack{j=1 \\ j \neq i}}^N X_{jik} - \sum_{d=1}^D Z_{idk} = \mathbf{0} \quad \forall k, i$$

$$\sum_{\substack{j=1 \\ j \neq i}}^N \sum_{i=1}^N X_{ijk} d_j \leq \left( \sum_{d=1}^D \sum_{i=1}^N Y_{djk} \right) * \mathbf{M} \quad \forall k$$



### **2.3.1 Πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα (VRPTW)**

Στο συγκεκριμένο πρόβλημα οι πελάτες-κόμβοι έχουν περισσότερα χαρακτηριστικά και η λύση θα πρέπει να ικανοποιήσει περισσότερους περιορισμούς. Σε αυτή την κατηγορία προβλημάτων έχουμε ένα σύνολο πελατών γεωγραφικά διασκορπισμένων σε συγκεκριμένη περιοχή. Ο κάθε ένας από αυτούς έχει συγκεκριμένη ποσότητα φορτίου που πρέπει να παραλάβει και η ιδιαιτερότητα του προβλήματος είναι ότι ο πελάτης πρέπει να εξυπηρετηθεί μέσα σε μια χρονική περίοδο, το χρονικό παράθυρο. Πριν και μετά το χρονικό παράθυρο, ο πελάτης δεν μπορεί να εξυπηρετηθεί. Τα οχήματα περιορισμένης χωρητικότητας βρίσκονται την στιγμή  $t_c=0$  στην αποθήκη. Έτσι το πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα ( Vehicle Routing Problem with Time Windows / VRPTW) προκύπτει σαν επέκταση του CVRP με ίδια αντικειμενική συνάρτηση ελαχιστοποίησης του κόστους, αλλά με επιπλέον περιορισμό την εξυπηρέτηση των πελατών μέσα σε συγκεκριμένα χρονικά παράθυρα ( time windows).

### **2.3.2 Το περιορισμένης χωρητικότητας πρόβλημα δρομολόγησης οχημάτων (CVRP)**

Στο πρόβλημα δρομολόγησης οχημάτων με περιορισμένη χωρητικότητα έχουμε μία αποθήκη, ένα στόλο οχημάτων με το κάθε όχημα να έχει συγκεκριμένη χωρητικότητα. Ένας αριθμός πελατών πρέπει να εξυπηρετηθεί με τον κάθε πελάτη να έχει μία συγκεκριμένη ζήτηση. Εδώ υπάρχει ο περιορισμός χωρητικότητας, με την συνολική ζήτηση των πελατών που θα εξυπηρετηθούν από ένα όχημα, να απαγορεύεται να υπερβεί τη συνολική χωρητικότητα του οχήματος. Η εξυπηρέτηση για τον κάθε πελάτη διαρκεί μία συγκεκριμένη χρονική διάρκεια, και το φορτηγό μετά από κάθε εξυπηρέτηση θα πρέπει να έχει ένα χρονικό περιθώριο ικανό για να μπορέσει να επιστρέψει στην αποθήκη. Μετά την εξυπηρέτηση όλων των πελατών που ανήκουν στον κάθε κύκλο, το όχημα επιστρέφει στην αποθήκη. Εδώ στόχος

είναι η ελαχιστοποίηση της ευκλείδειας απόστασης που θα διανυθεί συνολικά. Τα χαρακτηριστικά είναι ότι ο κάθε κύκλος περνάει από την αποθήκη, κάθε πελάτης θα δεχθεί επίσκεψη από έναν μόνο κύκλο και το άθροισμα της ζήτησης των κόμβων που επισκέπτονται από έναν κύκλο δεν πρέπει να ξεπερνάει τη συνολική χωρητικότητα του οχήματος.

### **2.3.3 Πρόβλημα δρομολόγησης οχημάτων χωρίς την επιστροφή στην αποθήκη (OVRP )**

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 φορτηγό 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.

### **2.3.4 Στοχαστικά προβλήματα δρομολόγησης οχημάτων (SVRP)**

Σε αυτά τα προβλήματα ένα όχημα πεπερασμένης χωρητικότητας φεύγει από την αποθήκη και πρέπει να παραδώσει ένα μέρος του φορτίου του σε ένα συγκεκριμένο αριθμό πελατών, των οποίων 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 εξυπηρετήσει όλους τους πελάτες διανύοντας την όσο το δυνατόν μικρότερη απόσταση. Η ζήτηση είναι στοχαστική μεταβλητή, με την κατανομή της όμως, να είναι γνωστή.

### **2.3.5 Το Περιοδικό Πρόβλημα Δρομολόγησης Οχημάτων (PVRP)**

Το περιοδικό πρόβλημα δρομολόγησης οχημάτων ( Periodic Vehicle Routing Problem/ PVRP) είναι ισοδύναμο με το MDVRP αν αντικαταστήσουμε της αποθήκες με ημέρες (έστω ημέρες, η μετρική μονάδα του χρόνου). Στο περιοδικό VRP οι πελάτες εξυπηρετούνται μία ή περισσότερες φορές μέσα στο χρονικό ορίζοντα, με διαδρομές που πραγματοποιούνται σε κάθε μία χρονική περίοδο.

### **2.3.6 Το δυναμικό πρόβλημα δρομολόγησης οχημάτων (DVRP)**

Στα δυναμικά προβλήματα δρομολόγησης οχημάτων (Dynamic Vehicle Routing Problem ) κάποια από τα δεδομένα του προβλήματος δεν είναι γνωστά εκ των προτέρων. Καινούργιες πληροφορίες γνωστοποιούνται και μορφοποιούν το πρόβλημα καθώς οι διαδρομές βρίσκονται σε εξέλιξη και στις περισσότερες περιπτώσεις απαιτείται άμεση ή γρήγορη απόκριση. Οι νέες αυτές πληροφορίες συχνά αφορούν καινούργιους πελάτες που πρέπει να εισαχθούν στην διαδρομή ή άλλους υπάρχοντες που τελικά ακυρώνονται. Ακόμα μεταβαλλόμενες παράμετροι μπορεί να είναι ο χρόνος ταξιδιού ή η ζήτηση του κάθε πελάτη.

### **2.3.7 Πρόβλημα Δρομολόγησης Οχημάτων με Δύο Είδη Πελατών (VRPB)**

Ακόμα ένα πρόβλημα που αποτελεί επέκταση του κλασσικού CVRP είναι το πρόβλημα δρομολόγησης οχημάτων με δύο είδη πελατών κατά την διάρκεια της διαδρομής ( Vehicle Routing Problem with Backhauls / VRPB). Το πρώτο είδος είναι οι πελάτες που απαιτούν την διανομή κάποιας ποσότητας προϊόντων (linehauls customers) και το δεύτερο απαιτεί μια ποσότητα του προϊόντος να περισυλλεχθεί από αυτόν (backhauls customers). Εισάγουμε επιπλέον τους περιορισμούς ότι : σε κάθε διαδρομή οι πελάτες του δεύτερου τύπου επισκέπτονται έπειτα από τους πελάτες του πρώτου τύπου και ότι οι διαδρομές που περιλαμβάνουν πελάτες μόνο του δεύτερου τύπου δεν επιτρέπονται.

### **2.3.8 Πρόβλημα Δρομολόγησης Οχημάτων με Διανομή και Παραλαβή κατά τη Διάρκεια της Διαδρομής (VRPPD)**

Όταν ο κάθε πελάτης για να εξυπηρετηθεί απαιτεί να του διανεμηθούν προϊόντα από το όχημα αλλά και το όχημα να παραλάβει προϊόντα από αυτόν, προκύπτει το πρόβλημα δρομολόγησης οχημάτων με διανομή και παραλαβή κατά τη διάρκεια της διαδρομής (Vehicle Routing Problem with pick-up and delivery /VRPPD). Στη βασική μορφή του προβλήματος, σε κάθε πελάτη αντιστοιχούνται δύο ποσότητες, η ποσότητα που θα διανεμηθεί  $d$ , και αυτή που θα παραληφθεί  $p$ . . Ακόμα για κάθε πελάτη καθορίζονται δύο κόμβοι: ο  $O$  και ο  $D$  που είναι οι κόμβοι από τους οποίους ξεκινάνε τα προϊόντα που πρέπει να διανεμηθούν στον πελάτη και αυτοί που καταλήγουν τα προϊόντα που συλλέγονται από τον πελάτη

## **Κεφάλαιο 3:**

### **Αλγόριθμοι Βελτιστοποίησης Προβλημάτων**

#### **3.1 Εισαγωγή**

Στο κεφάλαιο αυτό θα ασχοληθούμε με διάφορες ευρετικές και μεθευρετικές μεθόδους οι οποίες χρησιμοποιούνται για την επίλυση προβλημάτων δρομολόγησης οχημάτων (VRP).

#### **3.2 Απλοί Ευρετικοί Αλγόριθμοι (Heuristics)**

Οι ευρετικοί αλγόριθμοι χρησιμοποιούνται συνήθως όταν δεν υπάρχει καμία γνωστή μέθοδος για να βρούμε τη βέλτιστη λύση, υπό τους συγκεκριμένους περιορισμούς. Οι δύο στόχοι είναι η διαπίστωση αλγορίθμων με αποδεκτούς (καλούς) χρόνους τρεξίματος αλλά και την εύρεση καλής ποιότητας λύσεων. Οι ευρετικοί αλγόριθμοι παρουσιάζουν περιορισμένη σχετικά εξερεύνηση του χώρου όπου αναζητούνται οι λύσεις και τα αποτελέσματα που προκύπτουν να είναι αρκετά καλά και χωρίς μεγάλο χρόνο υπολογισμού.

Για κάθε πρόβλημα βελτιστοποίησης δεν υπάρχει μόνο ένας ευρετικός αλγόριθμος που δίνει τη βέλτιστη λύση, ακόμα ενδέχεται στο ίδιο πρόβλημα αλλά για ορισμένες τιμές των παραμέτρων να παρέχει καλύτερες λύσεις κάποιος ευρετικός αλγόριθμος ενώ για άλλες τιμές άλλος ευρετικός αλγόριθμος.

Υπάρχει ένας αρκετά μεγάλος αριθμός ευρετικών αλγορίθμων, που ο καθένας έχει κάποια ιδιαίτερα χαρακτηριστικά. Σύμφωνα με τα οποία κατατάσσονται σε 3 κατηγορίες:

- Αλγόριθμοι Απληστίας (Greedy Algorithms)
- Προσεγγιστικοί Αλγόριθμοι (Approximation Algorithms)
- Αλγόριθμοι Τοπικής Αναζήτησης (Local Search Algorithms)

### 3.2.1 Αλγόριθμοι Απληστίας

Οι αλγόριθμοι απληστίας προσπαθούν να οδηγήσουν σε μια εφικτή λύση του προβλήματος, αλλά πολλές φορές χρειάζονται πάρα πολύ μεγάλο χρόνο γιατί είναι μυωπικοί αλγόριθμοι, δηλαδή βλέπουν μόνο μπροστά.

Τα περισσότερα προβλήματα δρομολόγησης οχημάτων έχουν  $n$  κόμβους, ένας αλγόριθμος απληστίας λειτουργεί σταδιακά, χρησιμοποιώντας ένα από τα δεδομένα (κόμβους) κάθε φορά, με στόχο να το εισάγει σε ένα υποσύνολο που ικανοποιεί τους περιορισμούς δημιουργώντας έτσι μια εφικτή λύση. Η λύση κατασκευάζεται τμηματικά, βρίσκοντας δηλαδή την καλύτερη επιλογή από το σημείο που βρισκόμαστε μια δεδομένη στιγμή και δεν ελέγχει άλλες πιθανές επιλογές κόμβου που πιθανώς να οδηγούσαν σε καλύτερο αποτέλεσμα.

Ένα παράδειγμα αλγόριθμου απληστίας είναι ο αλγόριθμος του πλησιέστερου γείτονα ο οποίος θα χρησιμοποιηθεί για την εύρεση μιας αρχικής λύσης, σε όλες τις παραλλαγές του προβλήματος που εξετάστηκαν.

#### Αλγόριθμος Του Πλησιέστερου Γείτονα

Σε αυτή την ευρετική διαδικασία, ο πωλητής ξεκινά από κάποιο αρχικό κόμβο και μετά επισκέπτεται τον κόμβο που είναι πλησιέστερος στον αρχικό. Από εκεί επισκέπτεται τον πλησιέστερο κόμβο που δεν έχει ακόμη επισκεφτεί, μέχρις ότου όλοι οι κόμβοι περιληφθούν στην διαδρομή του, και τότε επιστρέφει στον αρχικό κόμβο. Τα βασικά βήματα του αλγόριθμου είναι:

**Βήμα 1:** Ξεκίνα με οποιοδήποτε αρχικό κόμβο (αποθήκη) σαν ξεκίνημα της διαδρομής

**Βήμα 2:** Βρίσκουμε τον πλησιέστερο κόμβο που δεν συμπεριλαμβάνεται στη διαδρομή και τον προσθέτουμε στο μονοπάτι.

**Βήμα 3:** Επαναλαμβάνουμε το Βήμα 2 μέχρις ότου όλοι οι κόμβοι ανήκουν στην διαδρομή.

### 3.2.2 Προσεγγιστικοί Αλγόριθμοι

Οι προσεγγιστικοί αλγόριθμοι λειτουργούν σαν τους αλγόριθμους απληστίας, μόνο που για να λύσουν το πρόβλημα χρησιμοποιούν επιπλέον πληροφορία. Ο αλγόριθμος του Christofides ή των Clark & Wright είναι ένα παράδειγμα.

Τα είδη της επιπλέον πληροφορίας μπορούν να ταξινομηθούν στις παρακάτω προσεγγίσεις:

1. Ομαδοποίηση πρώτα - Δρομολόγηση έπειτα
2. Δρομολόγηση πρώτα - Ομαδοποίηση έπειτα
3. Εξοικονομήσεις / Καταχώρηση
4. Βελτίωση ή Ανταλλαγή
5. Προσέγγιση μαθηματικού προγραμματισμού

6. Αλληλεπιδρών βελτιστοποίηση

7. Ακριβής διαδικασία

### **Αλγόριθμος Του Christofides**

Οι αλγόριθμοι απληστίας δημιουργούσαν τη λύση εκ του μηδενός χωρίς να χρησιμοποιούν προηγούμενη γνώση πάνω στο πρόβλημα. Αυτός ο αλγόριθμος ξεκινάει από ένα ελάχιστο τανύον δέντρο και αφού το μετατρέψει σε κύκλο του Euler στην συνέχεια το μετατρέπει σε ένα κύκλο του Hamilton.

### **Αλγόριθμος Των Εξοικονομήσεων Των Clark & Wright**

Θεωρείται ο πρώτος κατασκευαστικός αλγόριθμος που προτάθηκε για την επίλυση του προβλήματος δρομολόγησης οχημάτων. Με τη μέθοδο αυτή υπολογίζονται οι εξοικονομήσεις όλων των πελατών και στη συνέχεια δημιουργούνται οι διαδρομές με βάση τις καλύτερες εξοικονομήσεις.

### **3.2.3 Αλγόριθμοι Τοπικής Αναζήτησης**

Η τοπική αναζήτηση (local search algorithm) βασίζεται στην αρχαιότερη μέθοδο βελτιστοποίησης, στην μέθοδο δοκιμής και σφάλματος. Η μέθοδος είναι απλή και έχει αποδειχθεί πολύ επιτυχημένη στην πράξη σε ένα πολύ μεγάλο αριθμό προβλημάτων συνδυαστικής βελτιστοποίησης.



Οι αλγόριθμοι τοπικής αναζήτησης ξεκινάνε από μία αρχική τυχαία λύση και συνεχίζουν τη διαδικασία αναζήτησης εκτελώντας διαδοχικές παραλλαγές στην αρχική λύση. Μία κίνηση εφαρμόζεται στην ήδη υπάρχουσα λύση για να παραχθεί μία καινούρια από την γειτονιά της υπάρχουσας λύσης. Η απόφαση για το αν θα επιτραπεί κάποια κίνηση εξαρτάται από την ποιότητα των παραγόμενων λύσεων. Σε περίπτωση θετικής απόφασης τότε η υπάρχουσα λύση θα αντικατασταθεί από κάποια γειτονική η οποία θα χρησιμοποιηθεί ως αφετηρία για τις επακόλουθες δοκιμές, διαφορετικά η αναζήτηση συνεχίζεται με την υπάρχουσα λύση.

Οι βασικές μορφές τοπικής αναζήτησης είναι 3:

- λ-opt: ο πιο γνωστός από αυτούς είναι ο 2opt. Αυτή η μέθοδος αποτελείται γενικά από τη διαγραφή 2 ακμών και την επανασύνδεση δύο μονοπατιών με διαφορετικό τρόπο για να καθορίσουμε μια καινούρια διαδρομή.
- (λ-n)-exchange: ο πιο γνωστός από αυτούς είναι ο 1-1 ανταλλαγή(exchange) στο συγκεκριμένο αλγόριθμο 2 κόμβοι αλλάζουν θέση μεταξύ τους. Σημαντική είναι και η υποκατηγορία λ-0 ανταλλαγή γνωστή ως λ-0 επανατοποθέτηση(relocate). Χαρακτηριστικό παράδειγμα είναι 1-0 επανατοποθέτηση, όπου στη συγκεκριμένη μέθοδο ένας κόμβος επανατοποθετείται από το σημείο που βρίσκεται σε κάποιο άλλο σημείο.
- Κυκλικές κινήσεις (Cyclic moves): Αυτές οι σύνθετες κινήσεις εκτελούν μια σειρά κινήσεων οι οποίες ξεχωριστά δεν βελτιώνουν αναγκαστικά τη λύση, αλλά στο σύνολό τους μειώνουν τη συνολική απόσταση ταξιδιού. Το πιο γνωστό παράδειγμα είναι το NEC (Node Ejection Chain), η οποία θα μπορούσε να παρομοιαστεί με μια  $(1,1,1,\dots,1)$ -exchange.

Στα Προβλήματα Δρομολόγησης Οχημάτων με Πολλαπλές Αποθήκες (MD VRP) μπορούμε να εφαρμόσουμε τις παραπάνω μορφές με κάποιες μικρές διαφοροποιήσεις:

- Intra-Route: Σε αυτή την περίπτωση εφαρμόζουμε μόνο τις δυο πρώτες μορφές μέσα στην ίδια διαδρομή. Ο περιορισμός της ποσότητας των διαδρομών που αξιολογούνται ταυτόχρονα, μειώνει την ποσότητα των κόμβων που έχουν ληφθεί υπόψη και δεν επηρεάζει τους περιορισμούς φορτιού, μειώνοντας την υπολογιστική δυσκολία.
- Inter-Route: Εδώ μπορούν να εφαρμοστούν και οι 3 μορφές αλλά μόνο σε ξεχωριστές διαδρομές.

Το πιο συνηθισμένο είναι να εφαρμόζονται και οι δυο παραπάνω επιλογές, η μια έπειτα της άλλης.

### 3.3 Μεθευρετικές Μέθοδοι

Οι μεθευρετικοί αλγόριθμοι με τη χρήση της τοπικής αναζήτησης και υψηλότερου επιπέδου στρατηγικών επιτυγχάνει την αποφυγή τοπικών ελάχιστων για την εύρεση της βέλτιστης λύσης σε σύνθετα προβλήματα. Οι αλγόριθμοι αυτού του τύπου χρησιμοποιούνται κυρίως για την επίλυση πολύπλοκων προβλημάτων που τα δεδομένα είναι λίγα, η βέλτιστη λύση δεν είναι ορατή και η πλήρης καταγραφή των λύσεων είναι αδύνατη, λόγω του μεγάλου εύρους των λύσεων. Όταν όμως δοθεί μια οποιαδήποτε λύση μπορούμε να τη δοκιμάσουμε και να κρίνουμε πόσο ικανοποιητική είναι. Κοινό στοιχείο σχεδόν όλων των μεθευρετικών αλγορίθμων

είναι η επεξεργασία τυχαίων αναζητήσεων και γενικότερα παραδοσιακών αλγορίθμων σαν υπό διαδικασίες τους.

Επιπλέον, τα διάφορα είδη των μεθευρετικών αλγορίθμων εφαρμόζονται και αποτελούν αντικείμενο μελέτης σε πολλά διαφορετικά επιστημονικά πεδία . Η κατηγοριοποίησή τους πραγματοποιείται ανάλογα με τις λύσεις τις οποίες χρησιμοποιούν. Δηλαδή, υπάρχουν αλγόριθμοι που βασίζονται σε μία λύση και κάνουν αναζήτηση στη γειτονιά αυτής της λύσης και οι αλγόριθμοι που έχουν έναν πληθυσμό από λύσεις οι οποίες προσπαθούν να κάνουν αναζήτηση σε όλο τον χώρο των λύσεων.

Μια τρίτη κατηγορία είναι οι υβριδικές μορφές των παραπάνω κατηγοριών (hybrid algorithms) που χρησιμοποιούν μία λύση και κάνουν εξερεύνηση στην γειτονιά αναζήτησης γύρω από την λύση αλλά έχουν πάρα πολύ ισχυρές δυνατότητες εκμετάλλευσης ή εντατικοποίησης της περιοχής, γύρω από την λύση.

Για την επίλυση των V.R.P μπορούν να χρησιμοποιηθούν διάφοροι μεθευρετικοί αλγόριθμοι, οι βασικότεροι από τους οποίους είναι:

1. Αλγόριθμος Περιορισμένης Αναζήτησης (Tabu Search)
2. Αλγόριθμος Βελτιστοποίησης Αποικίας Μυρμηγκιών (Ant Colony Optimization)
3. Προσομοιωμένη Ανόπτηση (Simulated Annealing)
4. Αλγόριθμος Διασκορπισμένης Αναζήτησης (Scatter Search)
5. Γενετικοί Αλγόριθμοι (Genetic Algorithms)
6. Νευρωνικά Δίκτυα (Neural Networks)

### 3.3.1 Αλγόριθμος Περιορισμένης Αναζήτησης ( Tabu Search)

Ο αλγόριθμος της περιορισμένης αναζήτησης είναι μία μέθοδος επίλυσης που συνδυάζει τη διαδικασία τοπικής αναζήτησης και στρατηγική υψηλότερου επιπέδου για την αποφυγή παγίδευσης σε τοπικό ελάχιστο, δηλαδή ανήκει στην κατηγορία των μεθευρετικών αλγορίθμων. Η μέθοδος αρχικά παρουσιάστηκε από τον Fred W. Glover το 1986.

Ο αλγόριθμος περιορισμένης αναζήτησης χρησιμοποιεί ένα ευρετικό αλγόριθμο για να μετακινηθεί από την μία λύση στην άλλη. Υπάρχει όμως ο κίνδυνος να εγκλωβιστεί σε κάποιο τοπικό ελάχιστο και να μην είναι δυνατή η εξερεύνηση άλλων περιοχών για τη βελτίωση της λύσης καθώς είναι πιθανό το ενδεχόμενο να πραγματοποιούνται οι ίδιες εναλλαγές. Το παραπάνω πρόβλημα αποφεύγεται με τη χρήση μνήμης για τις προηγούμενες πραγματοποιηθέντες κινήσεις ώστε να ξεφύγει από το τοπικό ελάχιστο. Για να αποφευχθεί η επανάληψη των προηγούμενων λύσεων, οι κινήσεις που πραγματοποιούνται καταγράφονται σε μία λίστα, η οποία ονομάζεται λίστα περιορισμένων κινήσεων (tabu list). Οι κινήσεις που θα εισαχθούν στην λίστα απαγορεύεται να επαναληφθούν και έτσι αποφεύγεται η επανάληψη των ίδιων λύσεων. Οι συγκεκριμένες κινήσεις παραμένουν στη λίστα για ένα συγκεκριμένο αριθμό επαναλήψεων που το ορίζει ο χρήστης ή προσαρμόζεται δυναμικά ανάλογα με τις απαιτήσεις του προβλήματος. Η ανανέωση της λίστας απαγορευμένων κινήσεων γίνεται δυναμικά και λειτουργεί σύμφωνα με το σύστημα FIFO (First In- First Out). Η διαδικασία αυτή ονομάζεται μνήμη μικρής περιόδου (short term memory). Μία άλλη στρατηγική που μπορεί να προστεθεί, είναι η εισαγωγή περιορισμών που αφορούν τη συχνότητα. Σύμφωνα με αυτή, μετράτε ο αριθμός των φορών που μια συγκεκριμένη κίνηση εμφανίζεται κατά τη διάρκεια της αναζήτησης και με βάση τον περιορισμό συχνότητας εμφάνισης μίας κίνησης, γίνεται προσπάθεια να εμποδιστεί η επανάληψη αυτής της κίνησης. Αυτή η διαδικασία ονομάζεται μνήμη μακράς διάρκειας (long term memory), και βοηθάει να εξερευνηθούν κάποιοι αναξιόποητοι χώροι. Η στρατηγική αυτή, ονομάζεται στρατηγική διάχυσης (diversification). Υπάρχουν όμως και άλλες στρατηγικές. Μία από αυτές έχει στόχο, να παραμείνουν

οι μεταβλητές που εμφανίζονται πολύ συχνά, εντός της τελικής λύσης. Η λογική αυτής της στρατηγικής είναι ότι η βέλτιστη λύση θα βρίσκεται κοντά στο υποσχόμενο σημείο, και μπορεί να βρεθεί αν γίνουν κάποιες μικρό αλλαγές με αναζήτηση στη γειτονιά των μεταβλητών που κρατάμε. Η στρατηγική αυτή ονομάζεται στρατηγική εντατικοποίησης (intensification strategies) και η μνήμη ονομάζεται μνήμη μεσαίας περιόδου (medium term memory). Μερικές φορές είναι ευνοϊκότερο να αγνοήσουμε τους περιορισμούς. Αυτό μπορεί να γίνει, όταν μία κίνηση που θα πραγματοποιηθεί να επιφέρει καλύτερο αποτέλεσμα στη συνάρτηση κόστους. Σε αυτή την περίπτωση θα αγνοήσουμε τον περιορισμό και θα πραγματοποιήσουμε την κίνηση αφού θα μας επιφέρει καλύτερο αποτέλεσμα. Η αγνόηση αυτή των περιορισμών ονομάζεται κριτήριο απενεργοποίησης των περιορισμών (aspiration criterion) και λέμε ότι ενεργοποιήθηκε το κριτήριο φιλοδοξίας. Στην περίπτωση που το αποτέλεσμα όλων των κινήσεων επιφέρει μία κατώτερη λύση, τότε θα επιλέξουμε την πρώτη μη απαγορευμένη κίνηση, όπως περιγράφηκε και παραπάνω.

Η περιορισμένη αναζήτηση δεν συγκλίνει με φυσικό τρόπο και το κριτήριο σταματήματος το καθορίζουμε εμείς. Αυτό μπορεί να είναι είτε ένας συγκεκριμένος, μεγάλος αριθμός επαναλήψεων, είτε όταν η διαδικασία επαναληφθεί αρκετές φορές χωρίς να βελτιώνεται η λύση.

Ο ψευδοκώδικας του αλγορίθμου είναι ο ακόλουθος.

```
sBest ← s0
tabuList ← []
tabuList.push(s0)
while (not stoppingCondition())
    sNeighborhood ← getNeighbors(bestCandidate)
    bestCandidate ← sNeighborhood.firstElement
    for (sCandidate in sNeighborhood)
        if ( (not tabuList.contains(sCandidate)) and (fitness(sCandidate) >
fitness(bestCandidate)) )
            bestCandidate ← sCandidate
```

```

        end
    end
    if (fitness(bestCandidate) > fitness(sBest))
        sBest ← bestCandidate
    end
    tabuList.push(bestCandidate)
    if (tabuList.size > maxTabuSize)
        tabuList.removeFirst()
    end
end
return sBest

```

Πιο αναλυτικά, οι στρατηγικές αναζήτησης που ακολουθεί ο αλγόριθμος είναι η εντατικοποίηση (intensification) της αναζήτησης γύρω από ένα τοπικό βέλτιστο και η διάχυση (diversification) της έρευνας σε νέες περιοχές του εφικτού συνόλου. Οπότε θα μπορούσε κάποιος να υποθέσει, ότι ο αλγόριθμος είναι μια τεχνική καθοδήγησης (guiding technique) της τοπικής αναζήτησης βασισμένη σε προσαρμοστική μνήμη, δίνοντας στην περιορισμένη αναζήτηση της ιδιότητες μιας μεθευρετικής μεθόδου. Είναι μέθοδος προσδιοριστικού (deterministic) χαρακτήρα που προσπαθεί να επιτύχει ότι θεωρητικά υπόσχεται η προσομοιωμένη απόσβεση (simulated annealing) που είναι μέθοδος τυχαιοποίησης (στοχαστικού χαρακτήρα). Έτσι τα βασικά σημεία της μεθόδου μπορούν να συνοψιστούν στα εξής τέσσερα σημεία:

- **Μνήμη “H”** βασισμένη στην ιστορία της τοπικής αναζήτησης σε αντίθεση προς την τοπική αναζήτηση που δεν διαθέτει μνήμη.
- **Περιορισμοί Tabu** που βασιζόμενοι στην μνήμη απαγορεύουν συγκεκριμένες κινήσεις (κυρίως στοιχειώδεις πράξεις σε λύσεις για μετάβαση σε άλλες).
- **Κριτήρια φιλοδοξίας (aspiration)** που επιτρέπουν υπέρβαση των περιορισμών της μεθόδου π.χ. εάν η χρήση κάποιων απαγορευμένων

κινήσεων παρήγαγαν λύση καλύτερη από όλες που έχουν παραχθεί μέχρι εκείνη τη στιγμή.

- **Κριτήρια διάχυσης (diversification)** που επιτρέπουν την επιβολή ιδιοτήτων λύσεων που ιστορικά παρήγαγαν καλά αποτελέσματα και την διάχυση της αναζήτησης σε νέες περιοχές εφικτών λύσεων.

Ο ψευδοκώδικας του αλγορίθμου με τα παραπάνω κριτήρια είναι ο ακόλουθος.

Αλγόριθμος περιορισμένης αναζήτησης

Κατασκευή μίας αρχικής λύσης  $s_0$

$s^* = s_0$  ! Αρχικοποίηση της βέλτιστης λύσης

$f(s^*) = f(s_0)$

Κύρια Φάση

**Do while**

Υπολογισμός μιας γειτονικής λύσης  $s'$

**if**  $f(s') < f(s^*)$  **then**

$s^* = s'$

$f^* = f(s')$

**endif**

Αποθήκευσε την τελευταία κίνηση στη λίστα περιορισμένων κινήσεων και ταυτόχρονα αν έχει συμπληρωθεί το μέγεθος της λίστας διέγραψε την παλαιότερη εγγραφή

Κάλεσε κάθε  $k_1$  επαναλήψεις τη στρατηγική εντατικοποίησης

**if**  $f(\text{Sintensification}) < f(s^*)$  **then**

$s^* = \text{Sintensification}$

$f^* = f(\text{Sintensification})$

**endif**

Κάλεσε κάθε  $k_2$  επαναλήψεις τη στρατηγική διαφοροποίησης

**if**  $f(\text{Sctiversification}) < f(s^*)$  **then**

```
s* =Sctiversification  
f*=f(Sctiversification)
```

```
endif
```

```
Enddo
```

Επέστρεψε τη βέλτιστη λύση  $s^*$ .



## Κεφάλαιο 4:

### Περιγραφή και τρόπος επίλυσης του Προβλήματος

#### 4.1 Συνοπτική περιγραφή προβλήματος

Στην παρούσα διπλωματική εργασία ασχολούμαστε με την επίλυση του προβλήματος δρομολόγησης οχημάτων περιορισμένης χωρητικότητας με πολλαπλές αποθήκες (Multi-Depot Vehicle Routing Problem/ MDVRP).

Σκοπός του προβλήματος, όπως αναφέρθηκε στα προηγούμενα κεφάλαια, είναι ο καθορισμός των βέλτιστων διαδρομών στόλου οχημάτων που ξεκινάνε από τις αποθήκες και εξυπηρετούν όλους τους πελάτες, ώστε να ελαχιστοποιηθεί το συνολικό κόστος. Τα οχήματα του στόλου είναι ομογενή, έχουν δηλαδή ίδια πεπερασμένη χωρητικότητα. Κάθε πελάτης έχει πεπερασμένη ζήτηση που πρέπει να καλυφθεί. Θεωρούμε ότι το κόστος της διαδρομής προκύπτει αθροιστικά από τις επιμέρους αποστάσεις μεταξύ των πελατών που ανήκουν στην διαδρομή.

Οι περιορισμοί που διέπουν τα προβλήματα είναι οι εξής:

1. Κάθε πελάτης πρέπει να εξυπηρετείται μια μόνο φορά, από ένα και μόνο ένα φορτηγό.
2. Το σύνολο της ζήτησης που πρέπει να καλυφθεί σε κάθε διαδρομή, δεν θα πρέπει να ξεπερνάει την χωρητικότητα του οχήματος.
3. Όλα τα οχήματα πρέπει να ξεκινάνε και να τερματίζουν στην ίδια αποθήκη.

Τα παραπάνω προβλήματα θα επιλυθούν με την εφαρμογή της περιορισμένης αναζήτησης (Tabu Search / TS). Η μέθοδος χωρίζεται σε δύο φάσεις: την εύρεση μίας αρχικής εφικτής λύσης με τον αλγόριθμο του πλησιέστερου γείτονα (Nearest Neighbor Algorithm) και έπειτα βελτίωση της με εφαρμογή τοπικής αναζήτησης (Local Search).

## 4.2 Μοντελοποίηση του προβλήματος

Το πρόβλημα που καλούμαστε να επιλύσουμε, είναι το πρόβλημα δρομολόγησης οχημάτων, με χρονικά παράθυρα για τους πελάτες. Στο συγκεκριμένο πρόβλημα διαθέτουμε μία αποθήκη, ένα συγκεκριμένο αριθμό πελατών με συγκεκριμένη ζήτηση, ένα στόλο οχημάτων ίδιας και πεπερασμένης χωρητικότητας το καθένα. Στόχος είναι η εξυπηρέτηση όλων των πελατών, ικανοποιώντας τους περιορισμούς και ταυτόχρονα η αναζήτηση της βέλτιστης λύσης, ώστε τα φορτηγά να διανύσουν την όσο το δυνατόν μικρότερη ευκλείδεια απόσταση. Σε πρώτη φάση, θα βρεθεί μία αρχική υπό βέλτιστη εφικτή λύση που ικανοποιεί όλους τους περιορισμούς και στη συνέχεια έχοντας ως δεδομένη αυτή την αρχική λύση, καλούμαστε να τη βελτιστοποιήσουμε, με τη χρήση του μεθευρετικού αλγορίθμου περιορισμένης αναζήτησης (Tabu Search Algorithm). Οι αλγόριθμοι επιλύθηκαν στο προγραμματιστικό περιβάλλον της Matlab έχοντας γνωστά δεδομένα εισόδου.

Βάση της παραπάνω περιγραφής τα δεδομένα που χρειαζόμαστε για την επίλυση των προβλημάτων είναι το πλήθος των πελατών, η ζήτηση του κάθε ένα, η γεωγραφική τους θέση καθώς και η χωρητικότητα των οχημάτων. Τα δεδομένα αυτά θα τα παραλάβουμε από ένα αρχείο κειμένου .txt και θα τα εισάγουμε στο προγραμματιστικό περιβάλλον Matlab. Κάθε γραμμή του αρχείου αντιστοιχεί σε μία παράμετρο του προβλήματος.

Αναλυτικό παράδειγμα παρουσιάζεται παρακάτω.

p01 - Notepad

Στοιχεία Προβλήματος												
Στοιχεία Οχημάτων												
1	37	52	0	7	1	4	1	2	4	8		
2	49	49	0	30	1	4	1	2	4	8		
3	52	64	0	16	1	4	1	2	4	8		
4	20	26	0	9	1	4	1	2	4	8		
5	40	30	0	21	1	4	1	2	4	8		
6	21	47	0	15	1	4	1	2	4	8		
7	17	63	0	19	1	4	1	2	4	8		
8	31	62	0	23	1	4	1	2	4	8		
9	52	33	0	11	1	4	1	2	4	8		
10	51	21	0	5	1	4	1	2	4	8		
11	42	41	0	19	1	4	1	2	4	8		
12	31	32	0	29	1	4	1	2	4	8		
13	5	25	0	23	1	4	1	2	4	8		
14	12	42	0	21	1	4	1	2	4	8		
15	36	16	0	10	1	4	1	2	4	8		
16	52	41	0	15	1	4	1	2	4	8		
17	27	23	0	3	1	4	1	2	4	8		
18	17	33	0	41	1	4	1	2	4	8		
19	13	13	0	9	1	4	1	2	4	8		
20	57	58	0	28	1	4	1	2	4	8		
21	62	42	0	8	1	4	1	2	4	8		
22	42	57	0	8	1	4	1	2	4	8		
23	16	57	0	16	1	4	1	2	4	8		
24	8	52	0	10	1	4	1	2	4	8		
25	7	38	0	28	1	4	1	2	4	8		
26	27	68	0	7	1	4	1	2	4	8		
27	30	48	0	15	1	4	1	2	4	8		
28	43	67	0	14	1	4	1	2	4	8		
29	58	48	0	6	1	4	1	2	4	8		
30	58	27	0	19	1	4	1	2	4	8		
31	37	69	0	11	1	4	1	2	4	8		
32	38	46	0	12	1	4	1	2	4	8		
33	46	10	0	23	1	4	1	2	4	8		
34	61	33	0	26	1	4	1	2	4	8		
35	62	63	0	17	1	4	1	2	4	8		
36	63	69	0	6	1	4	1	2	4	8		
37	32	22	0	9	1	4	1	2	4	8		
38	45	35	0	15	1	4	1	2	4	8		
39	59	15	0	14	1	4	1	2	4	8		
40	5	6	0	7	1	4	1	2	4	8		
41	10	17	0	27	1	4	1	2	4	8		
42	21	10	0	13	1	4	1	2	4	8		
43	5	64	0	11	1	4	1	2	4	8		
44	30	15	0	16	1	4	1	2	4	8		
45	39	10	0	10	1	4	1	2	4	8		
46	32	39	0	5	1	4	1	2	4	8		
47	25	32	0	25	1	4	1	2	4	8		
48	25	55	0	17	1	4	1	2	4	8		
49	48	28	0	18	1	4	1	2	4	8		
50	56	37	0	10	1	4	1	2	4	8		
51	20	20	0	0	0	0						
52	30	40	0	0	0	0						
53	50	30	0	0	0	0						
54	60	50	0	0	0	0						

Στοιχεία Πελατών

Στοιχεία Αποθηκών

**Στοιχεία Προβλήματος p01:**

- type m n t
- 2 4 50 4

Όπου **type=2**, **m**(αριθμός οχημάτων)=**4**, **n**(αριθμός πελατών)=**50**, **t**(αποθήκες)=**4**

**type: κατηγορία προβλήματος**

- 0 (VRP)
- 1 (PVRP)
- 2 (MDVRP)
- 3 (SDVRP)
- 4 (VRPTW)
- 5 (PVRPTW)
- 6 (MDVRPTW)
- 7 (SDVRPTW)

**Στοιχεία Οχημάτων p01:**

- D Q
- 0 80
- 0 80
- 0 80
- 0 80

Όπου **D**(μέγιστη διάρκεια μιας διαδρομής)=**0** και **Q**(μέγιστο φορτίο ενός οχήματος)=**80**

**Στοιχεία Πελατών p01:**

- **i/ x/ γ/ d/q/f/a/list/e/l**
- 1 37 52 0 7 1 4 1 2 4 8
- 2 49 49 0 30 1 4 1 2 4 8
- 3 52 64 0 16 1 4 1 2 4 8
- ....
- 48 25 55 0 17 1 4 1 2 4 8
- 49 48 28 0 18 1 4 1 2 4 8
- 50 56 37 0 10 1 4 1 2 4 8

**Στοιχεία Αποθηκών p01:**

- **i/x/y/d/q/f/a**
- 51 20 20 0 0 0 0
- 52 30 40 0 0 0 0
- 53 50 30 0 0 0 0
- 54 60 50 0 0 0 0

Οι γραμμές περιέχουν για κάθε πελάτη/αποθήκη τις ακόλουθες πληροφορίες:

**i x y d q f a list e l**

Όπου

i=αριθμός πελατών

**x=χ**συντεταγμένη

$y = \gamma$  συντεταγμένη

**d=διάρκεια υπηρεσιών**

**q=απαίτηση**

**f=συχνότητα επίσκεψης**

**a**=αριθμός πιθανών συνδυασμών επίσκεψης

**list**=κατάλογος όλων των πιθανών συνδυασμών επίσκεψης

**e=αρχή χρόνου (νωρίτερος χρόνος για την έναρξη υπηρεσίας)**

Ι=τέλος του χρόνου (πιο πρόσφατος χρόνος για το τέλος της υπηρεσίας)

Διαβάζουμε και αποθηκεύουμε τα δεδομένα σε αντίστοιχες μεταβλητές.

Numv ο αριθμός οχημάτων

Numc ο αριθμός πελατών

Numd	ο αριθμός αποθηκών
Loadv	διάνυσμα με τη χωρητικότητα κάθε οχήματος
Cords	πίνακας (Numc,2) με τις συντεταγμένες κάθε πελάτη
Demand	διάνυσμα με τη ζήτηση κάθε πελάτη
dcords	πίνακας(Numd,2) με τις συντεταγμένες κάθε αποθήκης

Στη συνέχεια δημιουργήσαμε πίνακα που περιέχει τις αποστάσεις μεταξύ όλων των πελατών. Ο πίνακας συμβολίζεται ως  $dist(i,j)$ , όπου  $i$  ο προηγούμενος πελάτης και  $j$  ο επόμενος. Η απόσταση που υπολογίζεται είναι ευκλείδεια και ο πίνακας είναι συμμετρικός, που σημαίνει ότι η απόσταση που χρειαζόμαστε για να πάμε από έναν πελάτη  $i$  στον επόμενο  $j$ , ισούται με την αντίστροφη από τον  $j$  στον  $i$ . Σημαντικό είναι να μην ληφθεί υπόψη η απόσταση του πελάτη από τον εαυτό του και τα σημεία στον πίνακα  $dist(i,j)$ , η διαγώνιος, απειρίζονται. Ακόμα δημιουργούμε ένα πίνακα  $distdepot(i,j)$  που περιέχει τις αποστάσεις μεταξύ όλων των αποθηκών και των πελατών. Οι πίνακες γεμίζουν με τις παρακάτω εξισώσεις.

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

$$distdepot(i,j) = \sqrt{(cords(i,1) - dcords(j,1))^2 + (cords(i,2) - dcords(j,2))^2}$$

Τώρα έχουμε όλους πίνακες και όλες τις μεταβλητές βάσει των δεδομένων που μας δόθηκαν. Στη συνέχεια του προβλήματος θα χρειαστούν και άλλοι πίνακες και μεταβλητές που θα περιγραφούν στη συνέχεια.

## 4.2 Εύρεση αρχικής εφικτής λύσης με τον αλγόριθμο πλησιέστερου γείτονα

Έχοντας αποθηκεύσει τα δεδομένα του προβλήματος, το επόμενο βήμα είναι να ξεκινήσουμε την επίλυση του. Στην πρώτη φάση της επίλυσης, χρησιμοποιούμε και προσαρμόζουμε αναλόγως τον αλγόριθμο του πλησιέστερου γείτονα, για την εύρεση μίας αρχικής εφικτής λύσης. Το πρώτο βήμα στον αλγόριθμο, είναι η αρχικοποίηση των μεταβλητών που θα χρειαστούμε για την επίλυση του προβλήματος. Αρχικά, δηλώνονται οι μεταβλητές που δηλώνουν την αρχική κατάσταση που επικρατεί πριν ξεκινήσει η διαδικασία επίλυσης του προβλήματος. Τη χρονική στιγμή 0 γνωρίζουμε ότι τα φορτηγά βρίσκονται στην αποθήκη, ετοιμάζεται να ξεκινήσει το πρώτο όχημα κάθε αποθήκης, δεν έχει εξυπηρετηθεί κανένας πελάτης και τα φορτηγά που θα εκτελέσουν τα πρώτα δρομολόγια έχουν πλήρη διαθέσιμη χωρητικότητα. Επομένως ορίζουμε τις μεταβλητές.

Diadromes      διάνυσμα με όλα τα δρομολόγια

cargoload      διάνυσμα με το συνολικό φορτίο του κάθε δρομολογίου

Flag            βοηθητικό δυαδικό διάνυσμα ελέγχου πληρότητας του κάθε δρομολογίου

w              αριθμός δρομολογίων

x              αριθμός μεγέθους κάθε δρομολογίου

Στη αρχή ο αλγόριθμος δημιουργεί ένα πρώτο σετ διαδρομών (τόσες όσες και ο αριθμός των αποθηκών), εισάγει στο κάθε δρομολόγιο τον κοντινότερο πελάτη και απειρίζει την ζήτηση του αντίστοιχου πελάτη έτσι ώστε να μην είναι δυνατών ένας πελάτης να εξυπηρετηθεί πάνω από μια φορά. Στη συνέχεια με τη βοήθεια της while, γίνεται μια επαναληπτική διαδικασία που βρίσκει τον κοντινότερο πελάτη στον τελευταίο κάθε δρομολογίου και τον προσθέτει στην διαδρομή, μέχρις ότου να μην μπορεί κανένα όχημα να ικανοποιήσει κανέναν από τούς εναπομείναντες πελάτες. Τότε τα δρομολόγια κλειδώνουν, τα οχήματα επιστρέφουν στις αρχικές τους αποθήκες και δημιουργούνται νέες διαδρομές (τόσες όσες και ο αριθμός των αποθηκών). Η παραπάνω διαδικασία επαναλαμβάνεται μέχρι να μην υπάρχουν πελάτες που δεν εξυπηρετούνται.

#### **4.3 Εύρεση της βέλτιστης λύσης με τη χρήση του αλγορίθμου περιορισμένης αναζήτησης**

Η λύση που βρέθηκε προηγουμένως είναι εφικτή, αλλά δεν είναι βέλτιστη. Για να βελτιστοποιηθεί το συνολικό κόστος και για να μειωθεί ο αριθμός των διαδρομών γίνεται χρήση του αλγορίθμου περιορισμένης αναζήτησης (Tabu Search). Η μέθοδος αυτή παρότι είναι αρκετά περίπλοκη και δύσκολο να εφαρμοστεί εξ' ολοκλήρου, αποδίδει, δίνοντας αρκετά καλά αποτελέσματα. Η μέθοδος στηρίζεται σε εφαρμογές της τοπικής αναζήτησης, όπως είναι οι ανταλλαγές κόμβων μεταξύ διαφορετικών διαδρομών, οι εναλλαγές κόμβων ίδιας διαδρομής Στοιχεία Αποθηκών μία διαδρομή σε μία άλλη. Το κύριο χαρακτηριστικό της περιορισμένης αναζήτησης είναι ότι, χρησιμοποιεί μνήμη για την αποφυγή επανάληψης ίδιων κινήσεων. Για να εφαρμοστεί η μέθοδος απαιτείται η ύπαρξη μίας αρχικής εφικτής λύσης, την οποία καλούμαστε να βελτιώσουμε με διάφορες στρατηγικές.



Σε πρώτο βήμα ορίζουμε τις μεταβλητές από τα υπάρχοντα δεδομένα, που θα χρειαστούν στη συνέχεια. Επιπλέον θα χρειαστεί μια ουρά (FIFO) `tabu_list`, που θα αρχικοποιηθεί με μηδενικά και θα είναι διαστάσεων  $(cap, 2)$ . Αυτός είναι ο πίνακας απαγορευμένων κινήσεων και όταν πραγματοποιείται κάποια αλλαγή, τα στοιχεία που συμμετέχουν στην αλλαγή θα εισέρχονται σε αυτόν. Έτσι θα αποκλειστεί η περίπτωση να ξανά πραγματοποιηθεί αυτή η εναλλαγή για έναν αριθμό `cap` επαναλήψεων. Έτσι θα αποκλειστεί το ενδεχόμενο να οδηγηθούμε σε κύκλους γύρω από μία λύση.

Στον αλγόριθμο χρησιμοποιήθηκαν τέσσερις στρατηγικές εμπνευσμένες από την τοπική αναζήτηση, για τη βελτίωση του αποτελέσματος. Η πρώτη είναι η στρατηγική ανταλλαγής κόμβων μεταξύ διαφορετικών διαδρομών, η δεύτερη είναι η στρατηγική επανατοποθέτησης με σκοπό τη μείωση των οχημάτων, η τρίτη είναι παρόμοια με τη πρώτη και η τέταρτη είναι η στρατηγική εναλλαγής κόμβων μέσα στην ίδια διαδρομή χωρίς `Tabu`. Στη συνέχεια θα αναλυθεί ο τρόπος λειτουργίας της κάθε μίας και τι αποτέλεσμα μας αποφέρει.

#### **4.3.1 Ανταλλαγή μεταξύ κόμβων διαφορετικών διαδρομών**

Αυτό το τμήμα του αλγορίθμου θα εξετάσει αν μπορεί να βελτιωθεί το αποτέλεσμα και να μειωθεί το συνολικό κόστος. Στο αλγοριθμικό κομμάτι αυτό, θα πραγματοποιηθεί η ανταλλαγή κόμβων μεταξύ διαφορετικών διαδρομών. Η ανταλλαγή κόμβων μεταξύ διαφορετικών διαδρομών είναι μία από τις διαδικασίες της τοπικής αναζήτησης, 1-1 ανταλλαγή (1-1 exchange). Στη διαδικασία αυτή, επιλέγονται δύο κόμβοι, διαφορετικών διαδρομών που ικανοποιούν τους περιορισμούς, και γίνεται ταυτόχρονη ανταλλαγή.

Η βασική ιδέα του τμήματος αυτού είναι ότι αν αλλάξει η σειρά εξυπηρέτησης ανάμεσα σε δύο πελάτες, χωρίς την παραβίαση των περιορισμών, υπάρχει η πιθανότητα να μειωθεί η συνολική απόσταση. Για αυτό σε κάθε επανάληψη θα

επιχειρηθούν αρκετές αλλαγές, για να αναζητηθεί ο συνδυασμός εκείνος που βρίσκει το βέλτιστο αποτέλεσμα, χωρίς υπάρξει πρόβλημα καταπάτησης των περιορισμών.

Η πρώτη διαδικασία που εκτελείται, είναι η αναζήτηση των κόμβων που θα γίνει η ανταλλαγή. Συνεπώς, το πρώτο βήμα είναι η εύρεση γειτονιάς λύσεων. Αρχικά επιλέγεται ένας τυχαίος κόμβος. Από αυτόν τον κόμβο θα αναζητήσουμε και θα εξετάσουμε, size (μεταβλητή που ορίζει πόσες ανταλλαγές θα εξεταστούν) ανταλλαγές που δύναται να πραγματοποιηθούν. Δοκιμάζουμε δηλαδή την ανταλλαγή με κόμβους που ανήκουν σε διαφορετικές διαδρομές. Η ανταλλαγή όμως, δεν μπορεί να πραγματοποιηθεί με οποιονδήποτε κόμβο και σε αυτό το σημείο ελέγχεται αν είναι εφικτή η ανταλλαγή με τον εκάστοτε κόμβο.

Στη συνέχεια θα πρέπει να βρεθεί ποια ανταλλαγή θα πραγματοποιηθεί τελικά. Αρχικά ορίζουμε ως πιθανή ανταλλαγή αυτή που έχει το μικρότερο κόστος. Ακόμα δεν ξέρουμε όμως αν είναι επιτρεπτή και εδώ θα γίνει ο έλεγχος της `tabu_list`. Αν η κίνηση υπάρχει ήδη στον πίνακα `tabu_list`, τότε δεν μπορεί να πραγματοποιηθεί. Επομένως στη συνέχεια θα επιλεγθεί το δεύτερο μικρότερο κόστος, το τρίτο κ.ο.κ. μέχρι να βρεθεί η ανταλλαγή εκείνη που δεν βρίσκεται στην απαγορευμένη λίστα. Υπάρχει όμως μία εξαίρεση, σε περίπτωση που το νέο αποτέλεσμα είναι καλύτερο από το βέλτιστο κόστος που έχουμε. Σε αυτή την περίπτωση ενεργοποιείται το κριτήριο φιλοδοξίας, το οποίο χρησιμοποιείται όταν η κίνηση είναι απαγορευμένη αλλά μας δίνει καλύτερο αποτέλεσμα, οπότε η κίνηση θα πραγματοποιηθεί.

#### **4.3.2 Ελαχιστοποίηση Αριθμού διαδρομών**

Αυτό το τμήμα του αλγορίθμου βελτιστοποίησης εξετάζει αν μπορούν να μειωθεί ο συνολικός αριθμός των διαδρομών, συνεπώς και ο αριθμός των χρησιμοποιημένων οχημάτων. Το συγκεκριμένο τμήμα έχει διαφορετική λειτουργία και λογική, συγκριτικά με τα άλλα. Εδώ στόχος είναι η μεταφορά των κόμβων σε διαφορετικές διαδρομές για την εξάλειψη της διαδρομής ή εάν δεν μπορούν να μετακινηθούν όλα τα στοιχεία έτσι ώστε να μειωθούν τα δρομολόγια να γίνουν αλλαγές στα μεγέθη των δρομολογίων κάτι που δεν ήταν εφικτό με την προηγούμενη στρατηγική. Το τμήμα

του αλγορίθμου, χρησιμοποιεί τη λογική του αλγορίθμου της τοπικής αναζήτησης. Πιο συγκεκριμένα, εκτελεί διαδοχικά 1-0 επανατοποθετήσεις (1-0 relocate), δηλαδή διαγράφει ένα πελάτη από μία διαδρομή και τον τοποθετεί σε μία άλλη.

Το πρώτο βήμα στο πρόβλημα μας είναι η εύρεση της υποψήφιας προς διαγραφή, διαδρομή. Αντιλαμβανόμαστε πως δεν γίνεται να επιλεγθεί η διαδρομή που εξυπηρετεί τους περισσότερους πελάτες, γιατί θα είναι δύσκολη η μετακίνηση όλων των πελατών και πιθανών αδύνατη. Επιλέγεται το φορτηγό που εξυπηρετεί τον μικρότερο αριθμό πελατών. Εφόσον είναι γνωστή η υποψήφια για διαγραφή διαδρομή, εξετάζονται όλοι οι πελάτες που εμπεριέχονται σε αυτήν με τυχαία σειρά, αν μπορούν να μετακινηθούν σε άλλη διαδρομή χωρίς την παραβίαση του περιορισμού του συνολικού φορτίου. Ο πελάτης δεν μπορεί να μετακινηθεί σε μία διαδρομή με διαθέσιμη υπολειπόμενη χωρητικότητα μικρότερη από τη ζήτηση του.

Οι μετακινήσεις των πελατών γίνονται εντός μίας επαναληπτικής διαδικασίας. Σε κάθε μετακίνηση που εκτελείται, ανανεώνεται ο πίνακας δρομολογίων, όπως και η `tabu_list` και το διάνυσμα φορτίων των διαδρομών. Η επαναληπτική διαδικασία θα τελειώσει όταν όλοι οι κόμβοι της μικρότερης διαδρομής έχουν μετακινηθεί ή δεν υπάρχει άλλη εφικτή μετακίνηση εντός της διαδρομής. Αν έχουν μετακινηθεί όλοι οι πελάτες στις υπόλοιπες διαδρομές, τότε:

- Μειώνονται τα οχήματα
- Μειώνονται οι διαδρομές από τον πίνακα διαδρομών
- Υπολογίζεται το συνολικό κόστος
- Ξανά εφαρμόζεται η διαδικασία στο μικρότερο δρομολόγιο μιας άλλης αποθήκης

Σε αυτό το σημείο χρειάζεται να αναφέρουμε ότι το κόστος που θα προκύψει μπορεί να είναι μεγαλύτερο από το προηγούμενο βέλτιστο κόστος. Όμως, επειδή ο στόχος αυτής της στρατηγικής είναι η μείωση της συνολικής απόστασης και των διαδρομών, το αποδεχόμαστε. Το κόστος μπορεί εύκολα να αλλάξει και να βελτιωθεί με

κατάλληλες ανταλλαγές. Η μείωση των οχημάτων όμως είναι δύσκολο να επιτευχθεί και δεν πρέπει να την απορρίπτουμε.

Σε περίπτωση που δεν είναι δυνατή η μείωση των δρομολογίων, εφαρμόζονται οι αλλαγές μόνο εάν οδήγησαν σε βελτίωση του συνολικού κόστους.

#### **4.3.3 Ανταλλαγή μεταξύ κόμβων τυχαίων διαδρομών**

Αυτό το τμήμα του αλγορίθμου θα λειτουργήσει μόνο αν υπήρξαν αλλαγές στα δρομολόγια έχοντας εφαρμόσει την παραπάνω στρατηγική και θα εξετάσει αν μπορεί να βελτιωθεί το αποτέλεσμα και να μειωθεί το συνολικό κόστος. Στο αλγοριθμικό κομμάτι αυτό, θα πραγματοποιηθεί η ανταλλαγή κόμβων μεταξύ τυχαίων διαδρομών. Η ανταλλαγή κόμβων μεταξύ διαδρομών είναι μία από τις διαδικασίες της τοπικής αναζήτησης, 1-1 ανταλλαγή (1-1 exchange). Στη διαδικασία αυτή, επιλέγονται δύο κόμβοι, που ικανοποιούν τους περιορισμούς, και γίνεται ταυτόχρονη ανταλλαγή. Η διαδικασία είναι παρόμοια με την πρώτη και στόχος είναι η βελτίωση των αλλαγμένων δρομολογίων λόγω των επανατοποθετήσεων που έγιναν στο προηγούμενο βήμα.

#### **4.3.4 Βελτιστοποίηση κόμβων εντός της ίδιας διαδρομής**

Προκειμένου να βελτιώσουμε περισσότερο την λύση μας, θα χρησιμοποιήσουμε τον αλγόριθμο τοπικής αναζήτησης 2-opt. Η διαδικασία εφαρμόζεται σε κάθε στήλη του πίνακα λύσης, άρα σε κάθε διαδρομή. Σε κάθε στήλη υπολογίζουμε από τον πίνακα με τις αποστάσεις, τον ακριβότερο πελάτη, αυτόν δηλαδή που εμφανίζει την μεγαλύτερη απόσταση από τον προηγούμενο του και διαλέγουμε και έναν ακόμα τυχαίο (ο οποίος δεν είναι η αποθήκη). Κόβουμε το διάνυσμα της γραμμής σε αυτά τα δύο σημεία, κρατάμε το τμήμα που δημιουργείται, το αντιστρέφουμε και το επανασυνδέουμε στο σημείο που είναι κενό το διάνυσμα.

Η νέα διαδρομή που παράγεται είναι σίγουρα κομμάτι εφικτής λύσης, αφού οι πελάτες παραμένουν ίδιοι, άρα και η συνολική τους ζήτηση. Υπολογίζουμε το κόστος της νέας διαδρομής και αν είναι καλύτερο από το κόστος της αρχικής, αντικαθιστούμε το νέο διάνυσμα στην λύση. Εάν το κόστος δεν βελτιώνεται παραμένουμε στην αρχική λύση.

Η διαδικασία αυτή επαναλαμβάνεται για κάποιο αριθμό επαναλήψεων, που ορίζουμε εμείς. Οι περισσότερες επαναλήψεις αυξάνουν τις πιθανότητες για καλύτερο αποτέλεσμα. Τα αποτελέσματα της κάθε επανάληψης του 2-opt αλγόριθμου πάνω στην αρχική εφικτή λύση, αποθηκεύονται σε έναν πίνακα, υπολογίζεται το συνολικό κόστος της κάθε λύσης, συγκρίνονται μεταξύ τους και κρατάμε την λύση με το ελάχιστο κόστος.

## Κεφάλαιο 5:

### Περιγραφή και ανάλυση των αποτελεσμάτων

#### 5.1 Γενική περιγραφή

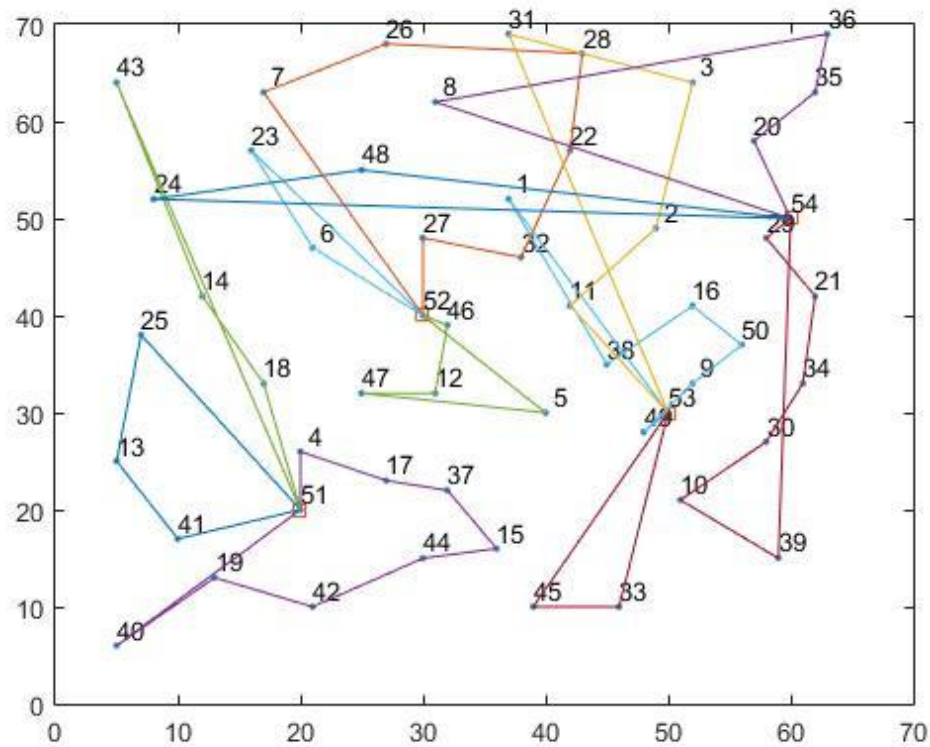
Ο προτεινόμενος αλγόριθμος υλοποιείται στο προγραμματιστικό περιβάλλον της MATLAB και δοκιμάζεται σε αρκετά προβλήματα δρομολόγησης οχημάτων, τα οποία είναι διαθέσιμα στο διαδίκτυο, με αριθμούς πελατών από 50 έως 360 και αριθμούς αποθηκών από 2 έως 9. Επιπλέον, πραγματοποιείται σύγκριση των αποτελεσμάτων με τα καλύτερα γνωστά αποτελέσματα στην βιβλιογραφία. Τα προβλήματα που επιλύονται έχουν προταθεί από τον Cordeau.

Το κάθε πρόβλημα επιλύθηκε 10 φορές και κρατήθηκε το καλύτερο αποτέλεσμα. Κάθε βελτιωτικό τμήμα είχε ως κριτήριο τερματισμού της 50.000 επαναλήψεις. Ακόμα το όριο της `tabu_list` έπαιρνε δυο τιμές ανάλογα το μέγεθος του προβλήματος, 10 θέσεις για μικρού μεγέθους και 16 για τα μεγάλα προβλήματα. Τέλος τα σετ των πιθανών επιλογών που δημιουργούνται πριν την κάθε επιλογή του `local_search` ανέρχονται στα 25.

## 5.2 Αναλυτικά αποτελέσματα των προβλημάτων

### 5.2.1 Αποτελέσματα προβλήματος p01

Αρχική λύση από αλγόριθμο πλησιέστερου γείτονα

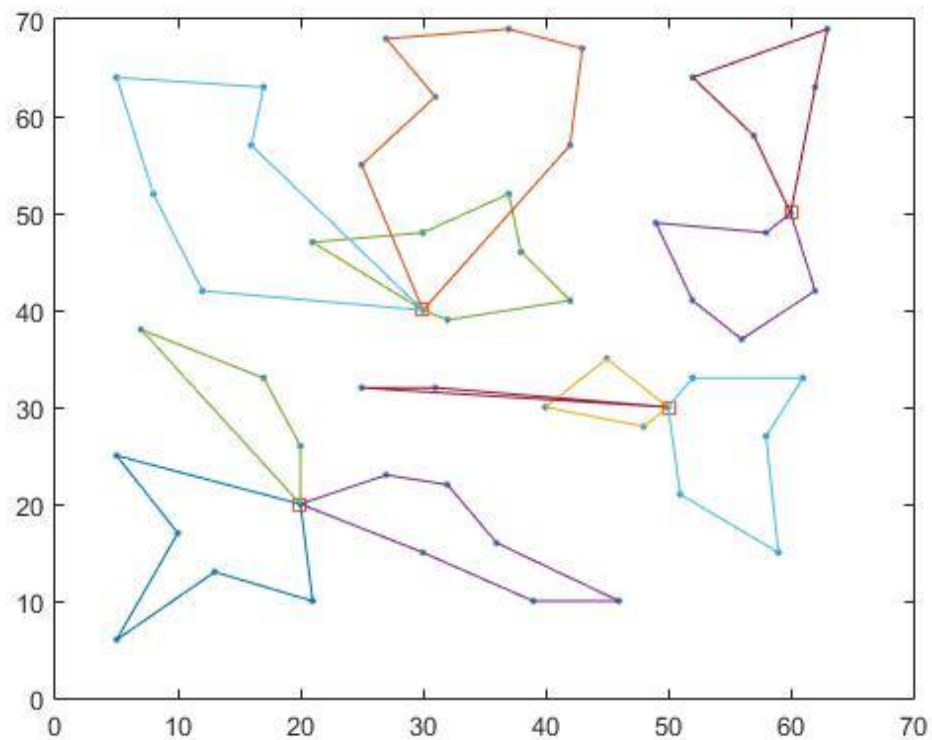


Συνολικό κόστος/απόσταση : 902,5678

Δρομολόγια

51	52	53	54	51	52	53	54	51	52	53	54
4	46	49	29	41	27	11	20	18	6	33	48
17	12	9	21	13	32	2	35	14	23	45	24
37	47	50	34	25	22	3	36	43	52	53	54
15	5	16	30	51	28	31	8	51	0	0	0
44	52	38	10	0	26	53	54	0	0	0	0
42	0	1	39	0	7	0	0	0	0	0	0
19	0	53	54	0	52	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0	0	0

## Τελικά αποτελέσματα αλγορίθμου



**Συνολικό κόστος/απόσταση : 603,5542**

### Δρομολόγια

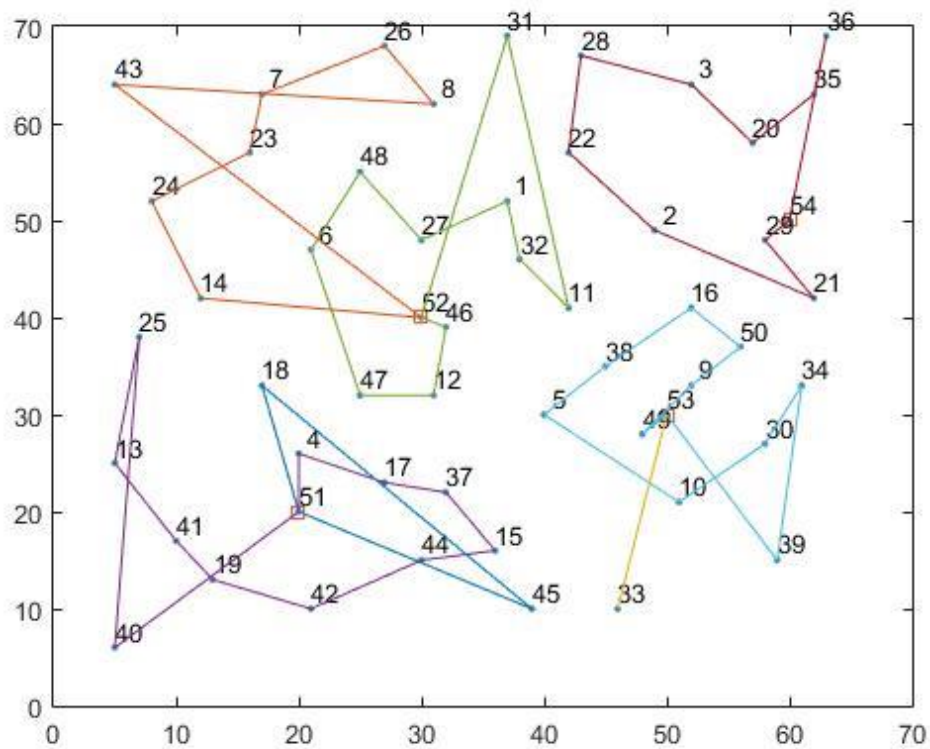
51	52	53	54	51	52	53	54	51	52	53
44	6	9	35	42	48	49	21	25	23	12
45	27	34	36	19	8	5	50	18	7	47
33	1	30	3	40	26	38	16	4	43	53
15	32	39	20	41	31	53	2	51	24	0
37	11	10	54	13	28	0	29	0	14	0
17	46	53	0	51	22	0	54	0	52	0
51	52	0	0	0	52	0	0	0	0	0

### Φορτία

**71    73    75    67    79    80    54    69    78    77    54**  
**5.2.2 Αποτελέσματα προβλήματος p02**

Αρχική λύση από αλγόριθμο πλησιέστερου γείτονα



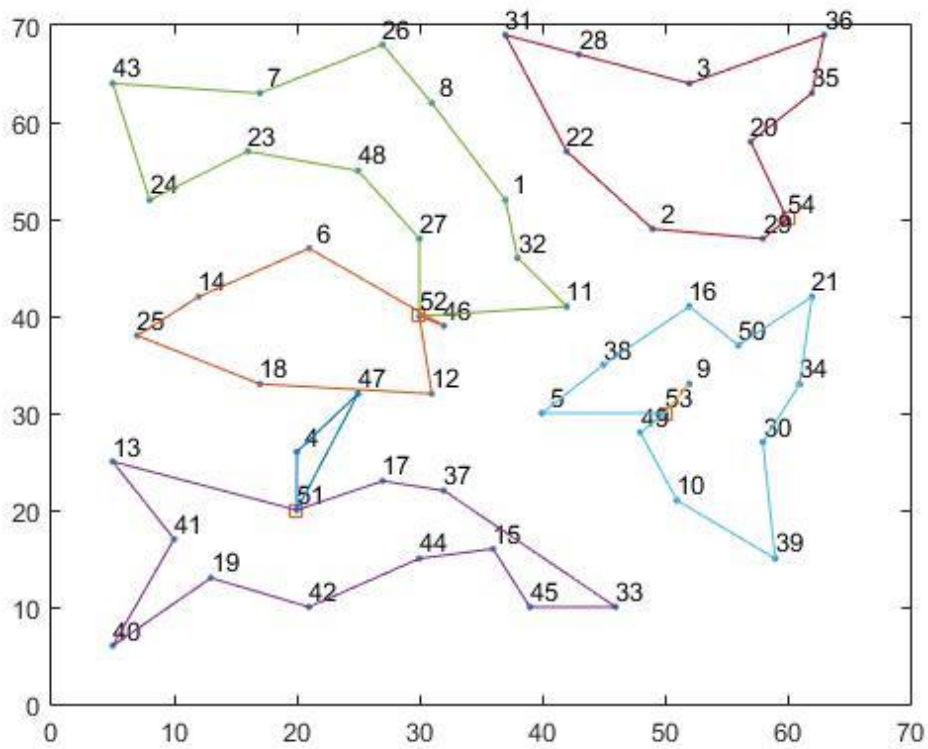


**Συνολικό κόστος/απόσταση : 686,9223**

### Δρομολόγια

51	52	53	54	51	52	53	54
4	46	49	29	18	14	33	54
17	12	9	21	45	24	53	0
37	47	50	2	51	23	0	0
15	6	16	22	0	7	0	0
44	48	38	28	0	26	0	0
42	27	5	3	0	8	0	0
19	1	10	20	0	43	0	0
41	32	30	35	0	52	0	0
13	11	34	36	0	0	0	0
25	31	39	54	0	0	0	0
40	52	53	0	0	0	0	0
51	0	0	0	0	0	0	0

Τελικά αποτελέσματα αλγορίθμου



**Συνολικό κόστος/απόσταση : 504,9399**

#### Δρομολόγια

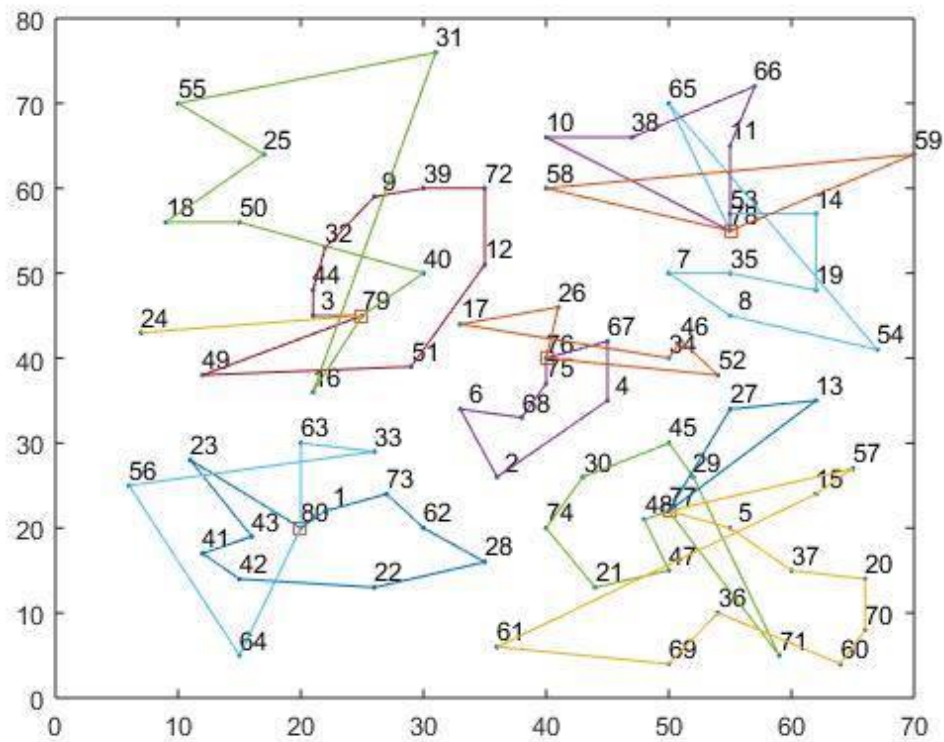
51	52	53	54	51	52	53	54
17	11	5	20	4	46	9	54
37	32	38	35	47	6	53	0
33	1	16	36	51	14	0	0
45	8	50	3	0	25	0	0
15	26	21	28	0	18	0	0
44	7	34	31	0	12	0	0
42	43	30	22	0	52	0	0
19	24	39	2	0	0	0	0
40	23	10	29	0	0	0	0
41	48	49	54	0	0	0	0
13	27	53	0	0	0	0	0
51	52	0	0	0	0	0	0

#### Φορτία

150	156	151	136	34	139	11	0
-----	-----	-----	-----	----	-----	----	---

#### 5.2.3 Αποτελέσματα προβλήματος p03

Αρχική λύση από αλγόριθμο πλησιέστερου γείτονα

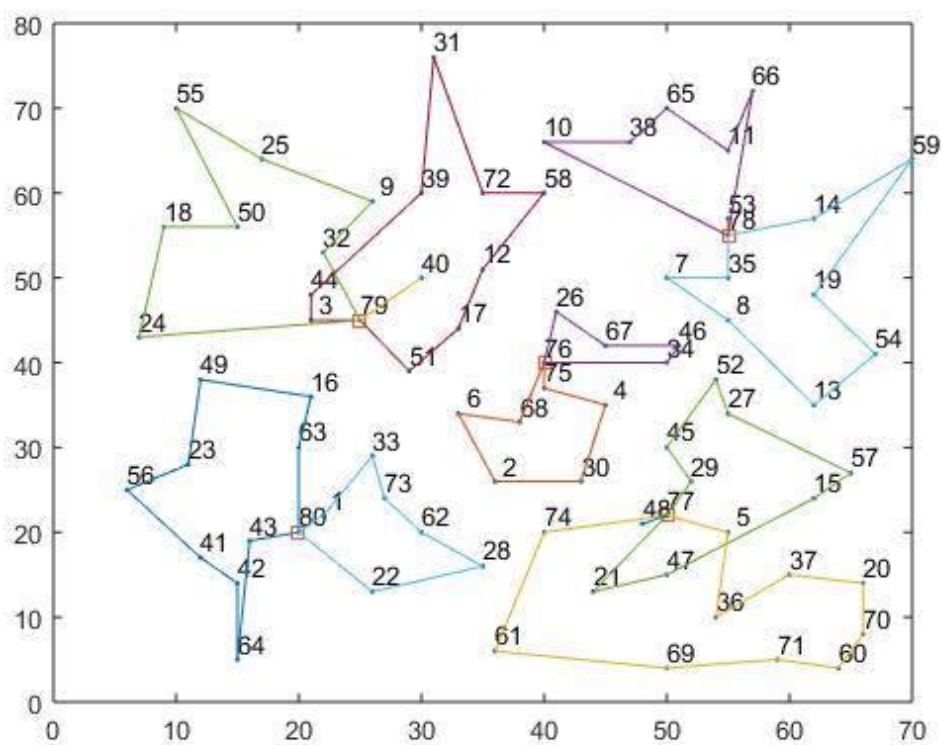


**Συνολικό κόστος/απόσταση : 938,93**

### Δρομολόγια

76	77	78	79	80	76	77	78	79	80	76	77	78	79
75	48	53	3	1	26	5	11	40	63	76	27	58	24
68	47	14	44	73	17	37	66	50	33	0	13	59	79
6	21	19	32	62	34	20	38	18	56	0	77	78	0
2	74	35	9	28	46	70	10	25	64	0	0	0	0
4	30	7	39	22	52	60	78	55	80	0	0	0	0
67	45	8	72	42	76	36	0	31	0	0	0	0	0
76	29	54	12	41	0	69	0	16	0	0	0	0	0
0	71	65	51	43	0	61	0	79	0	0	0	0	0
0	77	78	49	23	0	15	0	0	0	0	0	0	0
0	0	0	79	80	0	57	0	0	0	0	0	0	0
0	0	0	0	0	0	77	0	0	0	0	0	0	0

Τελικά αποτελέσματα αλγορίθμου



**Συνολικό κόστος/απόσταση : 711,93**

#### **Δρομολόγια**

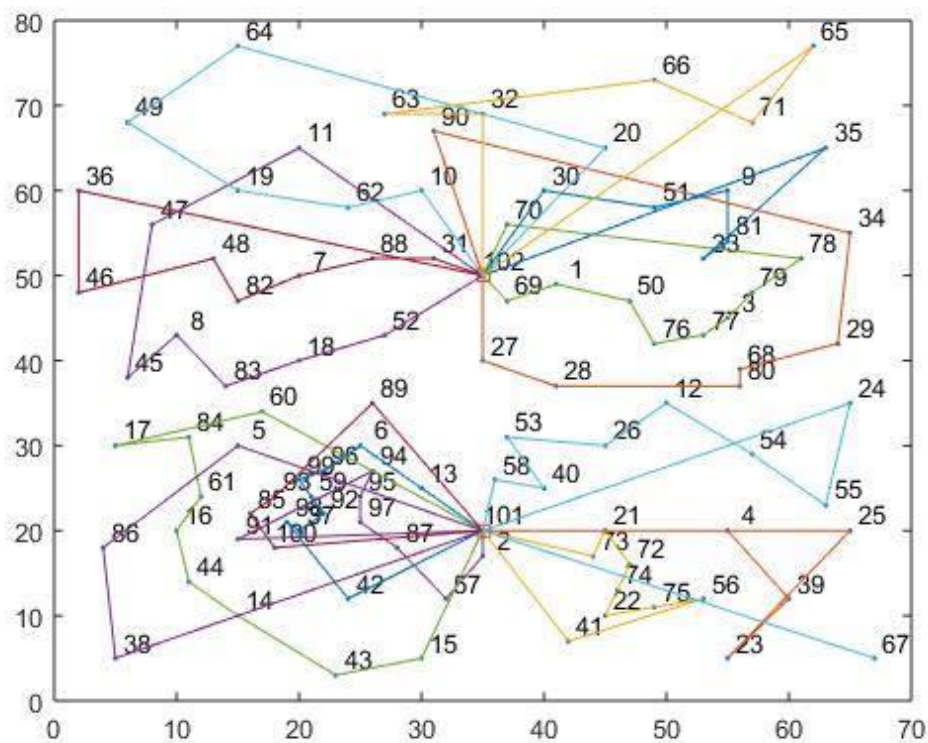
76	77	78	79	80	76	77	78	79	80	76	77	78	79
34	21	14	51	43	68	74	66	32	1	76	48	53	40
46	47	59	17	64	6	61	11	9	33	0	77	78	79
67	15	19	12	42	2	69	65	25	73	0	0	0	0
26	57	54	58	41	30	71	38	55	62	0	0	0	0
76	27	13	72	56	4	60	10	50	28	0	0	0	0
0	52	8	31	23	75	70	78	18	22	0	0	0	0
0	45	7	39	49	76	20	0	24	80	0	0	0	0
0	29	35	44	16	0	37	0	79	0	0	0	0	0
0	77	78	3	63	0	36	0	0	0	0	0	0	0
0	0	0	79	80	0	5	0	0	0	0	0	0	0
0	0	0	0	0	0	77	0	0	0	0	0	0	0

#### **Φορτία**

94	139	139	139	139	127	129	133	140	110	0	20	22	33
----	-----	-----	-----	-----	-----	-----	-----	-----	-----	---	----	----	----

#### **5.2.4 Αποτελέσματα προβλήματος p04**

Αρχική λύση από αλγόριθμο πλησιέστερου γείτονα

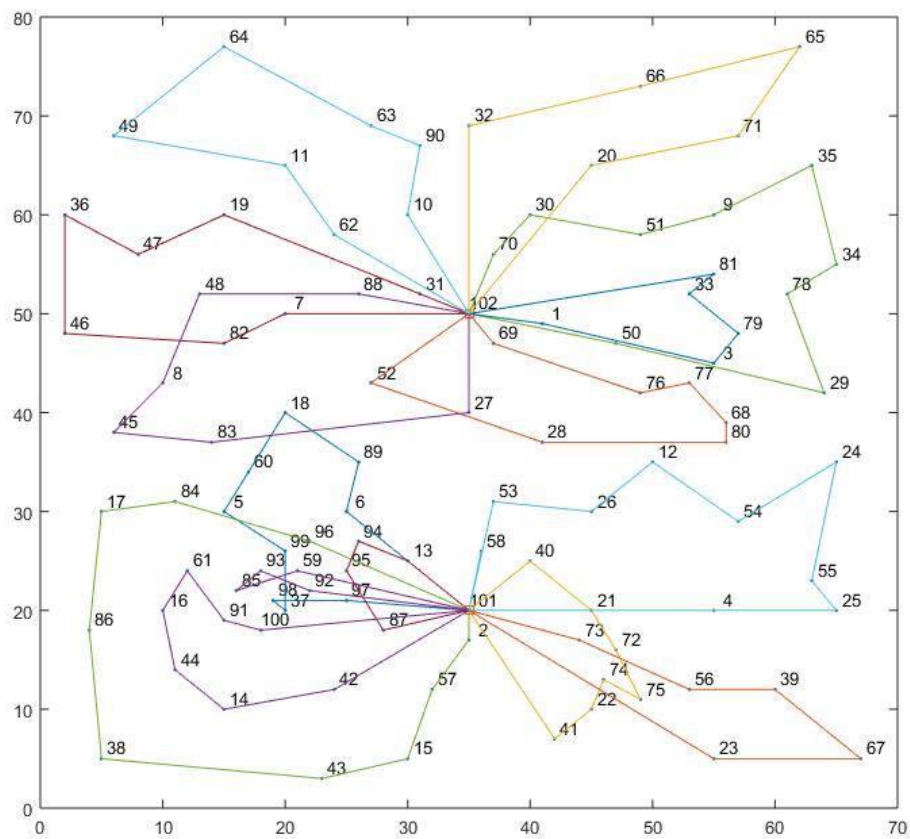


Συνολικό κόστος/απόσταση : 1309,7

### Δρομολόγια

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

Τελικά αποτελέσματα αλγορίθμου



**Συνολικό κόστος/απόσταση : 1076,7**

#### Δρομολόγια

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

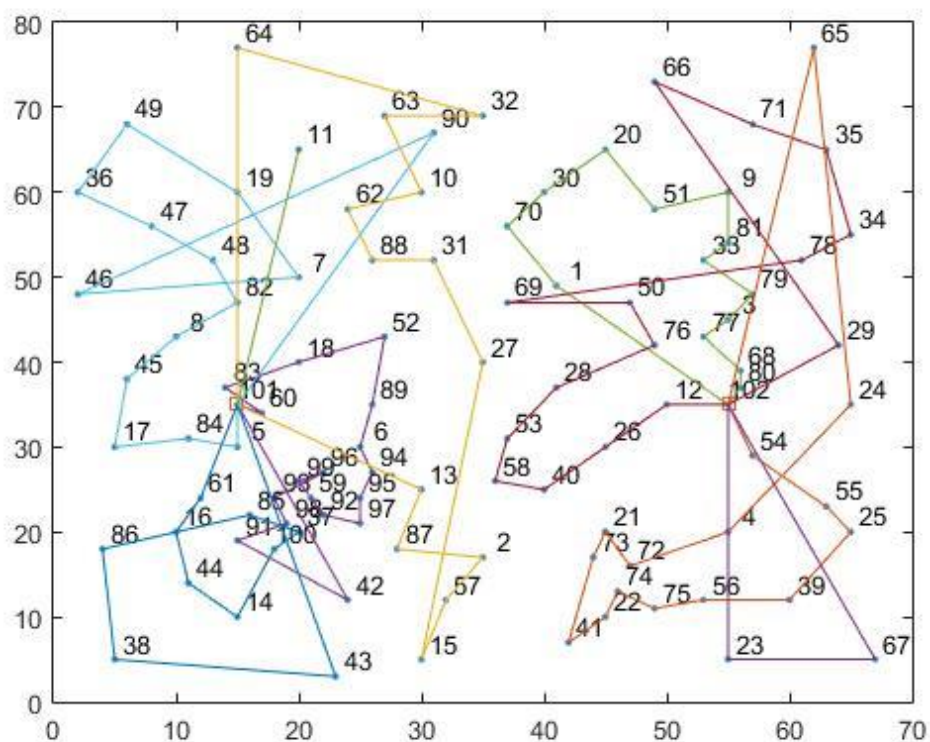
#### Φορτία

93 99 98 98 98 100 94 97 100 99 96 83 100 92 93 0 18

#### 5.2.5 Αποτελέσματα προβλήματος p05



Αρχική λύση από αλγόριθμο πλησιέστερου γείτονα

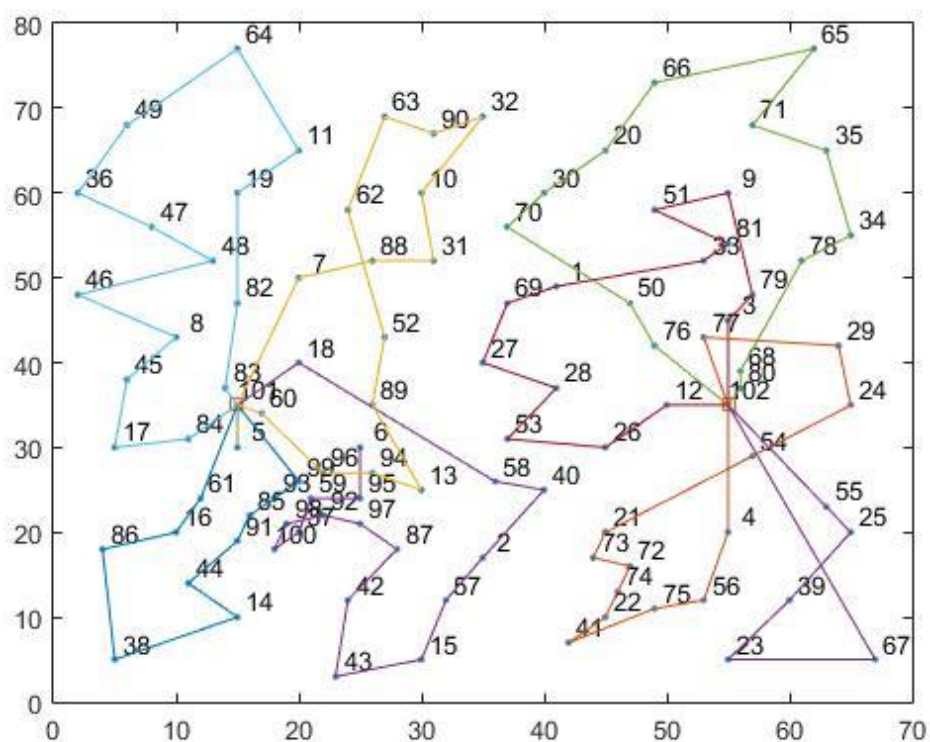


Συνολικό κόστος/απόσταση : 1151,3

#### Δρομολόγια

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

Τελικά αποτελέσματα αλγορίθμου



**Συνολικό κόστος/απόσταση : 841,4980**

#### Δρομολόγια

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

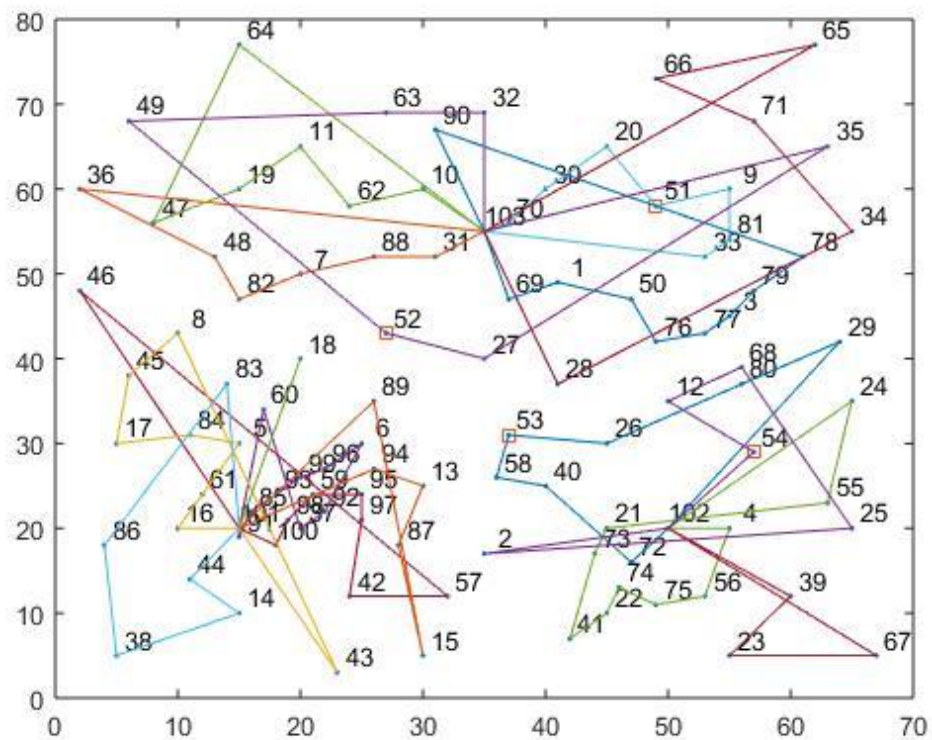
#### Φορτία

196	188	198	197	194	163	200	93	26	0
-----	-----	-----	-----	-----	-----	-----	----	----	---

#### 5.2.6 Αποτελέσματα προβλήματος p06



Αρχική λύση από αλγόριθμο πλησιέστερου γείτονα

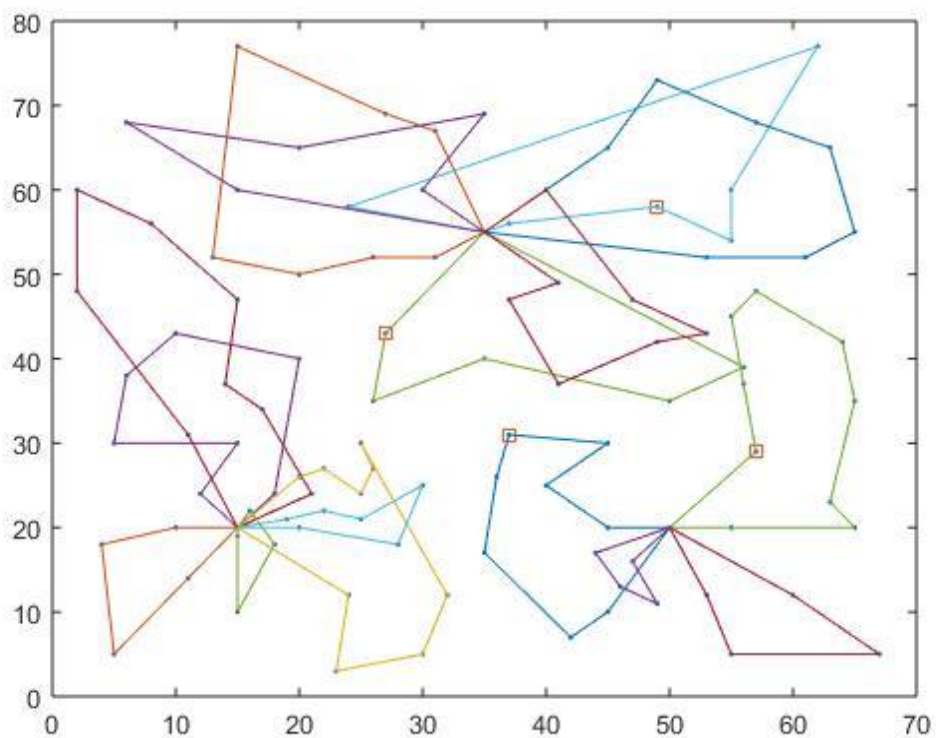


Συνολικό κόστος/απόσταση : 1378,8

Δρομολόγια

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

Τελικά αποτελέσματα αλγορίθμου



**Συνολικό κόστος/απόσταση : 1054,2**

#### Δρομολόγια

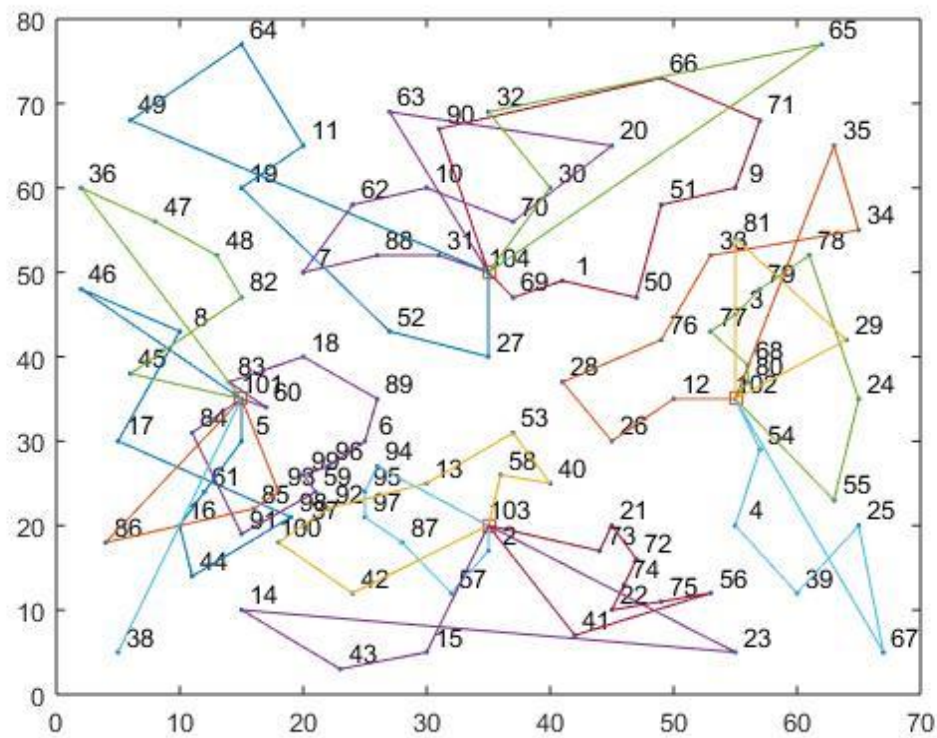
101	102	103	101	102	103	101	102	103	101	102	103	101	102	103	101	102	103
93	4	70	59	21	31	42	73	52	37	39	33	16	102	19	91	102	1
18	25	51	60	40	88	43	74	89	87	67	78	86	0	49	14	0	69
8	55	81	83	26	7	15	75	27	13	23	34	38	0	11	100	0	28
45	24	9	82	53	48	57	72	12	97	56	35	44	0	32	85	0	76
17	29	65	47	58	64	6	102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

#### Φορτία

100	99	96	98	99	99	97	60	95	81	91	85	88	0	98	79	0	93
-----	----	----	----	----	----	----	----	----	----	----	----	----	---	----	----	---	----

#### 5.2.7 Αποτελέσματα προβλήματος p07

Αρχική λύση από αλγόριθμο πλησιέστερου γείτονα

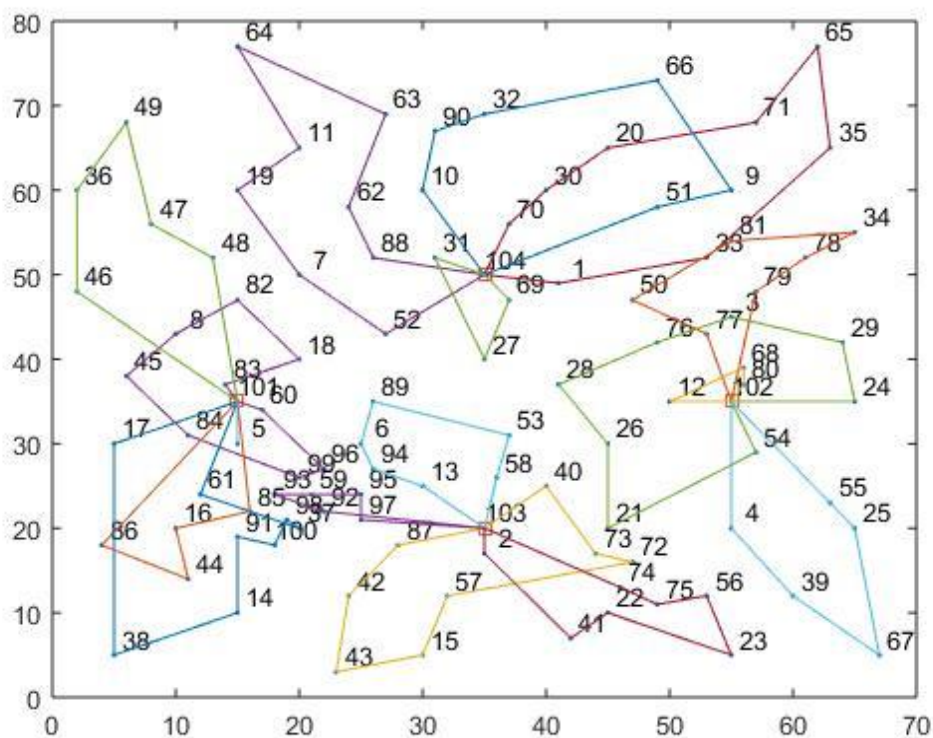


**Συνολικό κόστος/απόσταση : 1246,8**

#### Δρομολόγια

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

Τελικά αποτελέσματα αλγορίθμου



Συνολικό κόστος/απόσταση : 1006,5

#### Δρομολόγια

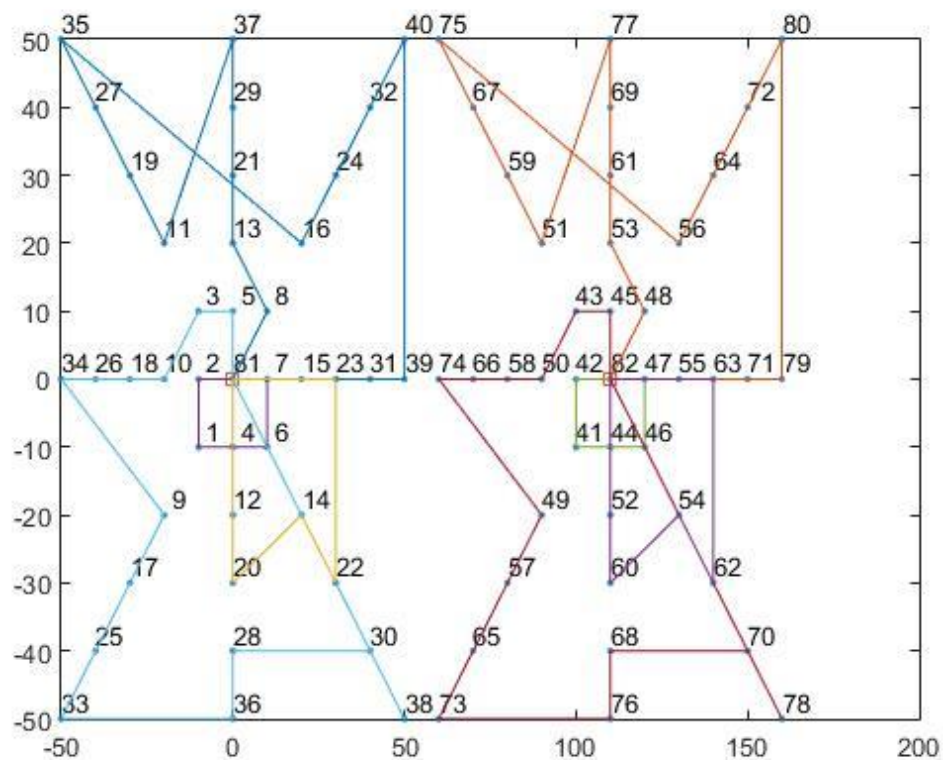
101	102	103	104	101	102	103	104	101	102	103	104	101	102	103	104	101
60	54	13	70	17	77	87	52	46	4	74	51	86	12	97	31	5
96	21	94	30	38	50	42	7	36	39	75	9	44	68	95	27	101
99	26	6	20	14	81	43	19	49	67	56	66	16	80	59	69	0
84	28	89	71	91	34	15	11	47	25	23	32	85	102	93	104	0
45	76	53	65	100	78	57	64	48	55	22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

#### Φορτία

94 100 100 99 87 93 96 90 99 83 91 93 113 61 84 49 26

#### 5.2.8 Αποτελέσματα προβλήματος p12

Αρχική λύση από αλγόριθμο πλησιέστερου γείτονα

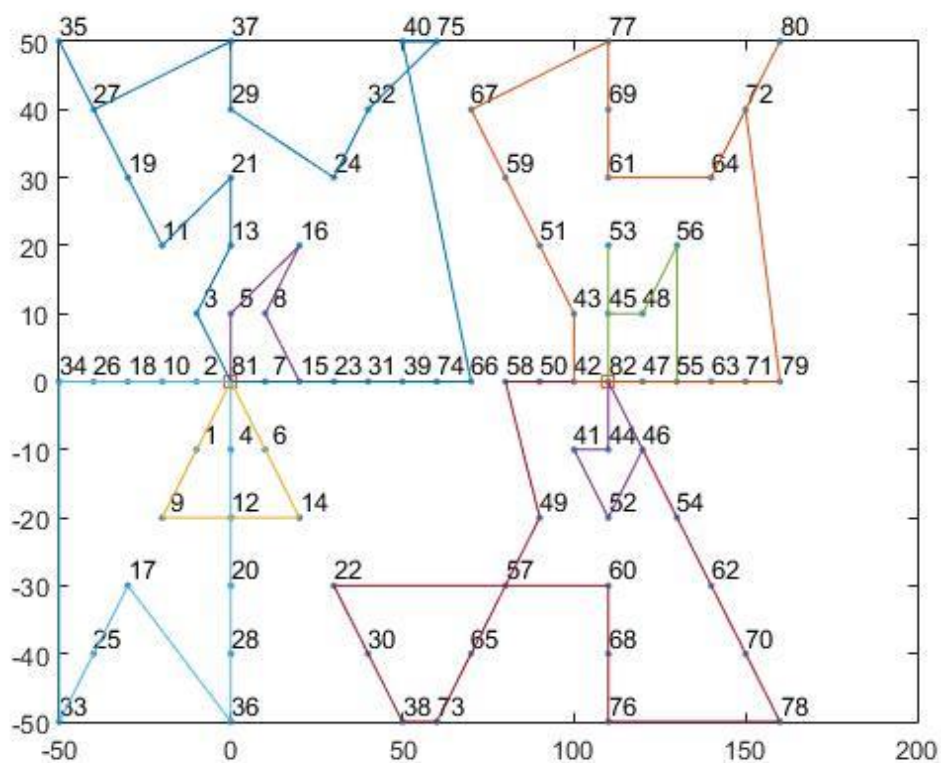


Συνολικό κόστος/απόσταση : 1738,7

Δρομολόγια

81	82	81	82	81	82	81	82
2	42	5	45	8	48	12	52
1	41	3	43	13	53	20	60
4	44	10	50	21	61	14	54
6	46	18	58	29	69	22	62
7	47	26	66	37	77	23	63
81	82	34	74	11	51	15	55
0	0	9	49	19	59	81	82
0	0	17	57	27	67	0	0
0	0	25	65	35	75	0	0
0	0	33	73	16	56	0	0
0	0	36	76	24	64	0	0
0	0	28	68	32	72	0	0
0	0	30	70	40	80	0	0
0	0	38	78	39	79	0	0
0	0	81	82	31	71	0	0
0	0	0	0	81	82	0	0

Τελικά αποτελέσματα αλγορίθμου



Συνολικό κόστος/απόσταση : 1559,6

Δρομολόγια

81	82	81	82	81	82	81	82
7	53	2	50	3	42	1	46
15	45	10	58	13	43	9	52
8	48	18	49	21	51	12	41
16	56	26	57	11	59	14	44
5	55	34	65	19	67	6	82
81	47	33	73	35	77	81	0
0	82	25	38	27	69	0	0
0	0	17	30	37	61	0	0
0	0	36	22	29	64	0	0
0	0	28	60	24	80	0	0
0	0	20	68	32	72	0	0
0	0	4	76	75	79	0	0
0	0	81	78	40	71	0	0
0	0	0	70	66	63	0	0
0	0	0	62	74	82	0	0
0	0	0	54	39	0	0	0
0	0	0	82	31	0	0	0
0	0	0	0	23	0	0	0
0	0	0	0	81	0	0	0

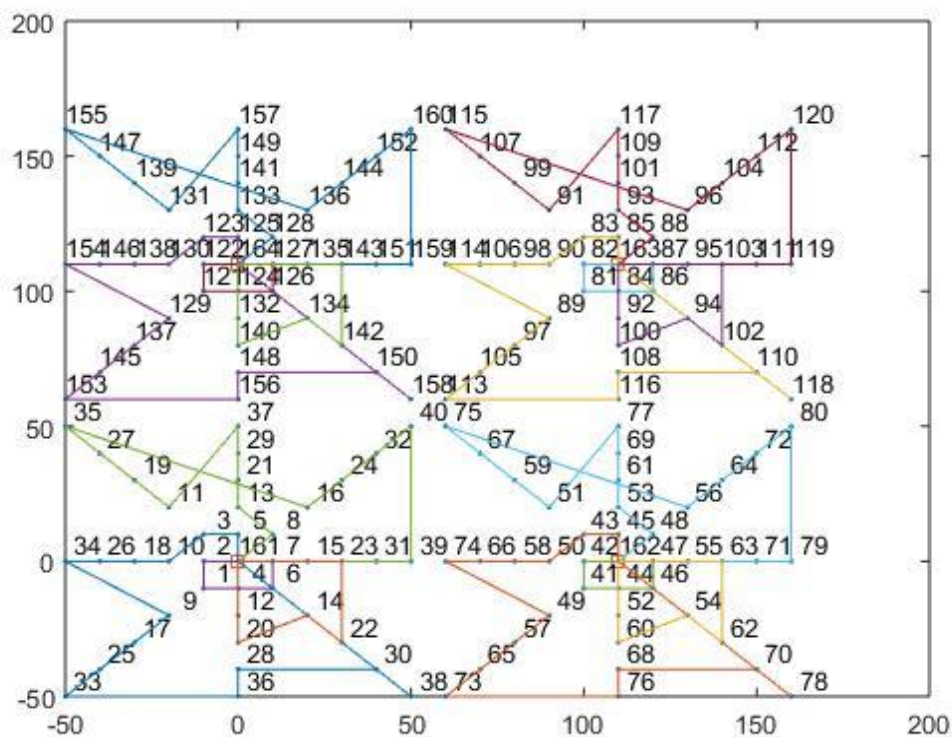
Φορτία

52 60 53 56 60 59 48 44

## 5.2.9 Αποτελέσματα προβλήματος p15



Αρχική λύση από αλγόριθμο πλησιέστερου γείτονα



**Συνολικό κόστος/απόσταση : 3473,6**

**Δρομολόγια**

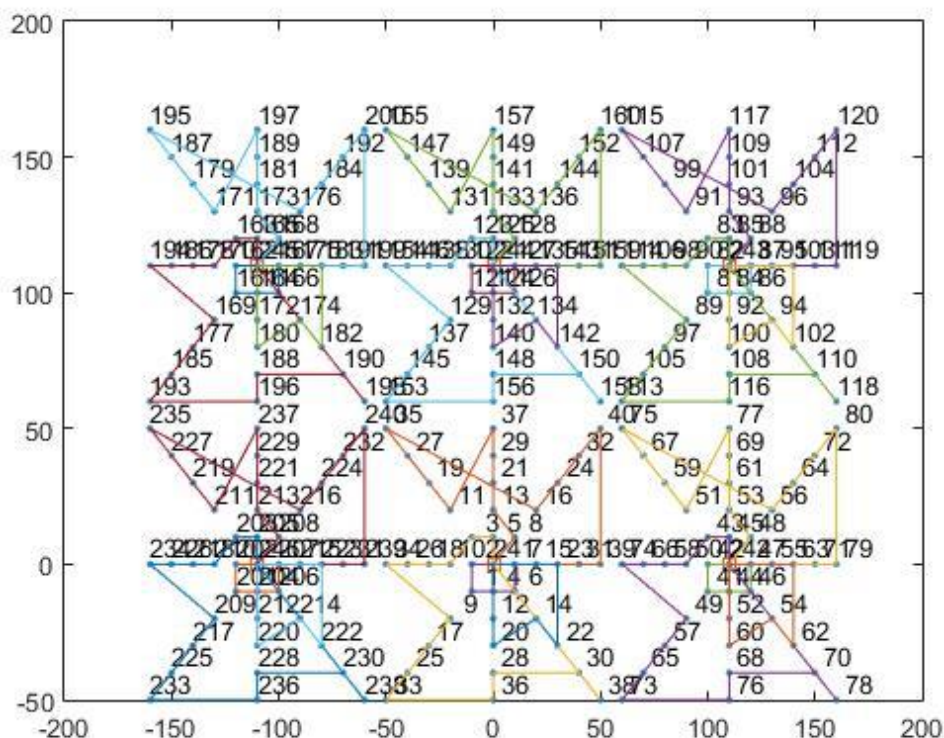
161	162	163	164	161	162	163	164	161	162	163	164	161	162	163	164
2	42	82	122	5	45	85	125	8	48	88	128	12	52	92	132
1	41	81	121	3	43	83	123	13	53	93	133	20	60	100	140
4	44	84	124	10	50	90	130	21	61	101	141	14	54	94	134
6	46	86	126	18	58	98	138	29	69	109	149	22	62	102	142
7	47	87	127	26	66	106	146	37	77	117	157	23	63	103	143
161	162	163	164	34	74	114	154	11	51	91	131	15	55	95	135
0	0	0	0	9	49	89	129	19	59	99	139	161	162	163	164
0	0	0	0	17	57	97	137	27	67	107	147	0	0	0	0
0	0	0	0	25	65	105	145	35	75	115	155	0	0	0	0
0	0	0	0	33	73	113	153	16	56	96	136	0	0	0	0
0	0	0	0	36	76	116	156	24	64	104	144	0	0	0	0
0	0	0	0	28	68	108	148	32	72	112	152	0	0	0	0
0	0	0	0	30	70	110	150	40	80	120	160	0	0	0	0
0	0	0	0	38	78	118	158	39	79	119	159	0	0	0	0
0	0	0	0	161	162	163	164	31	71	111	151	0	0	0	0
0	0	0	0	0	0	0	0	161	162	163	164	0	0	0	0

Τελικά αποτελέσματα αλγορίθμου





Αρχική λύση από αλγόριθμο πλησιέστερου γείτονα



**Συνολικό κόστος/απόσταση : 5216,6**

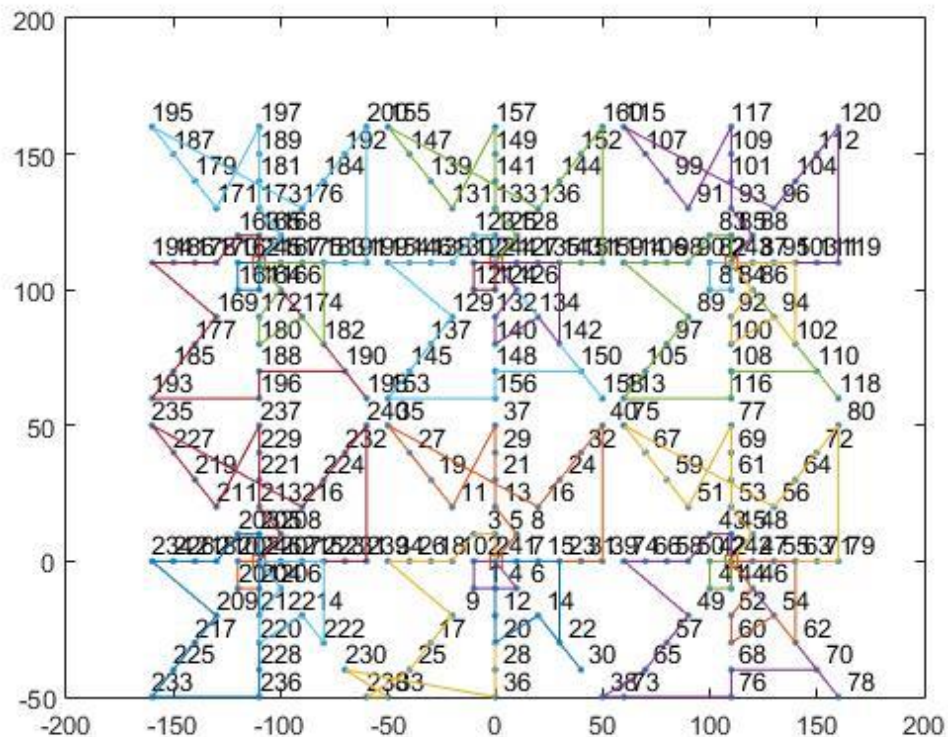
**Δρομολόγια**

241	242	243	244	245	246	241	242	243	244	245	246
5	45	85	125	165	205	8	48	88	128	168	208
3	43	83	123	163	203	13	53	93	133	173	213
10	50	90	130	170	210	21	61	101	141	181	221
18	58	98	138	178	218	29	69	109	149	189	229
26	66	106	146	186	226	37	77	117	157	197	237
34	74	114	154	194	234	11	51	91	131	171	211
9	49	89	129	169	209	19	59	99	139	179	219
17	57	97	137	177	217	27	67	107	147	187	227
25	65	105	145	185	225	35	75	115	155	195	235
33	73	113	153	193	233	16	56	96	136	176	216
36	76	116	156	196	236	24	64	104	144	184	224
28	68	108	148	188	228	32	72	112	152	192	232
30	70	110	150	190	230	40	80	120	160	200	240
38	78	118	158	198	238	39	79	119	159	199	239
241	242	243	244	245	246	31	71	111	151	191	231
0	0	0	0	0	0	241	242	243	244	245	246

241	242	243	244	245	246	241	242	243	244	245	246
2	42	82	122	162	202	12	52	92	132	172	212

1	41	81	121	161	201	20	60	100	140	180	220
4	44	84	124	164	204	14	54	94	134	174	214
6	46	86	126	166	206	22	62	102	142	182	222
7	47	87	127	167	207	23	63	103	143	183	223
241	242	243	244	245	246	15	55	95	135	175	215
0	0	0	0	0	0	241	242	243	244	245	246

Τελικά αποτελέσματα αλγορίθμου



Συνολικό κόστος/απόσταση : 5089,6  
Δρομολόγια

241	242	243	244	245	246	241	242	243	244	245	246
12	46	86	126	166	206	2	42	82	122	162	202
20	52	92	132	172	212	1	41	81	121	161	201
14	60	100	140	180	220	4	44	84	124	164	204
30	54	94	134	174	214	6	242	243	244	245	246
22	62	102	142	182	222	241	0	0	0	0	0
23	63	103	143	183	223	0	0	0	0	0	0
15	55	95	135	175	215	0	0	0	0	0	0
7	47	87	127	167	207	0	0	0	0	0	0
241	242	243	244	245	246	0	0	0	0	0	0

241	242	243	244	245	246	241	242	243	244	245	246
5	45	85	125	165	205	8	48	88	128	168	208

3	43	83	123	163	203	13	53	93	133	173	213
10	50	90	130	170	210	21	61	101	141	181	221
18	58	98	138	178	218	29	69	109	149	189	229
26	66	106	146	186	226	37	77	117	157	197	237
34	74	114	154	194	234	11	51	91	131	171	211
9	49	89	129	169	209	19	59	99	139	179	219
17	57	97	137	177	217	27	67	107	147	187	227
25	65	105	145	185	225	35	75	115	155	195	235
238	38	113	153	193	233	16	56	96	136	176	216
33	73	116	156	196	236	24	64	104	144	184	224
230	76	108	148	188	228	32	72	112	152	192	232
36	68	110	150	190	246	40	80	120	160	200	240
28	70	118	158	198	0	39	79	119	159	199	239
241	78	243	244	245	0	31	71	111	151	191	231
0	242	0	0	0	0	241	242	243	244	245	246

### Φορτία

50	60	60	60	60	60	48	36	36	36	36	36
60	60	60	60	60	57	60	60	60	60	60	60

## Κεφάλαιο 6:

### Τελικά συμπεράσματα

#### 6.1 Σύγκριση αποτελεσμάτων

Στο προηγούμενο κεφάλαιο παρουσιάστηκαν τα αποτελέσματα των αλγορίθμων πλησιέστερου γείτονα και περιορισμένης αναζήτησης. Σε αυτό το κεφάλαιο θα γίνει αξιολόγηση αυτών των αποτελεσμάτων. Αρχικά θα παρουσιαστούν πίνακες, που θα περιλαμβάνουν συγκριτικά αποτελέσματα ανάμεσα στην αρχική λύση και την βέλτιστη της και ανάμεσα στην βέλτιστη λύση και την παγκοσμίως βέλτιστη. Η σύγκριση των αποτελεσμάτων αναγράφεται σε % απόκλιση από τη βέλτιστη λύση, σε % βελτίωση από την αρχική λύση. Αναλυτικά παρουσιάζονται στο παρακάτω πίνακα.

Πρόβλημα	Κόστος NNH	Κόστος Τελικό	Βελτίωση	Καλύτερη γνωστή λύση	Απόκλιση
p01	902,56	603,55	33%	576,87	4%
p02	686,92	504,93	26%	473,53	6%
p03	938,93	711,93	24%	641,19	10%
p04	1309,7	1076,7	18%	1001,59	7%
p05	1151,3	841,49	27%	750,03	11%
p06	1378,8	1054,2	24%	876,5	17%
p07	1246,8	1006,5	19%	885,8	12%
p12	1738,7	1559,6	10%	1318,95	15%
p15	3473,6	3423,6	1%	2505,42	27%
p18	5216,6	5089,6	2%	3702,85	27%

#### 6.2 Ανάλυση αποτελεσμάτων και συμπεράσματα

Από τον παραπάνω πίνακα συγκριτικών αποτελεσμάτων οδηγούμαστε σε κάποιες στατιστικές παρατηρήσεις, βάση των οποίων μπορεί να εκτιμηθεί η ποιότητα των αποτελεσμάτων και εν συνέχεια των αλγορίθμων. Οι κυριότερες στατιστικές παρατηρήσεις είναι:

- Ο μέσος όρος απόκλισης από τη παγκοσμίως βέλτιστη λύση είναι

14%. Το εύρος απόκλισης είναι σε κάποια 4% το ελάχιστο και 27% το μέγιστο.

- Ο μέσος όρος βελτίωσης των αρχικών εφικτών λύσεων του αλγορίθμου πλησιέστερου γείτονα ανέρχεται στο 19%. Το εύρος κυμαίνεται ανάμεσα στο 1% βελτίωσης της αρχικής λύσης, και 33% το μέγιστο ποσοστό βελτίωσης της αρχικής λύσης.

Συνολικά, μπορούμε να οδηγηθούμε στο συμπέρασμα ότι ο αλγόριθμος βελτιστοποίησης με τη χρήση της περιορισμένης αναζήτησης είναι αρκετά αποτελεσματικός. Αρχικά η απόκλιση από το παγκοσμίως βέλτιστο είναι σχετικά μικρή. Σε αρκετά προβλήματα έχουμε απόκλιση κοντά στο 10%. Τις μεγαλύτερες αποκλίσεις στα αποτελέσματα τις παρατηρούμε στα προβλήματα που εκτελούνται πολλές διαδρομές.

Ο αλγόριθμος Tabu Search ήταν ικανοποιητικός και συγκριτικά με τη βελτίωση των αρχικών εφικτών λύσεων. Βελτίωσε τη λύση σχεδόν σε όλα τα προβλήματα κατά αρκετά ικανοποιητικό ποσοστό. Καθόλου αποτελεσματικός φάνηκε στα δυο τελευταία προβλήματα, όπου παρατηρείτε μια συμμετρία στα δρομολόγια της κάθε αποθήκης, πιθανώς να υπήρχε βελτίωση με χρήση ξεχωριστών `local_search` όπου με μια αλλαγή θα επηρεάζονταν συμμετρικά όλο τα δρομολόγια.

## Βιβλιογραφία

1. Αθανάσιος Μυγδαλάς, Ιωάννης Μαρινάκης, Αθανασία Μαυρομάτη. Σημειώσεις μαθ Διαχείρισης Εφοδιαστικής αλυσίδας.
2. Ιωάννης Μαρινάκης - Μυγδαλάς Αθανάσιος: "Σχεδιασμός Και Βελτιστοποίηση Της Εφοδιαστικής Αλυσίδας"
3. A tabu search metaheuristic algorithm for MDVRP, De Freitas
4. A tabu search metaheuristic algorithm for MDVRP, Elsevier Science
5. Fred Glover (1986). "Future Paths for Integer Programming and Links to Artificial Intelligence". *Computers and Operations Research*. **13** (5): 533–549. [doi:10.1016/0305-0548\(86\)90048-1](https://doi.org/10.1016/0305-0548(86)90048-1).
6. Fred Glover (1989). "Tabu Search – Part 1". *ORSA Journal on Computing*. **1** (2): 190–206. [doi:10.1287/ijoc.1.3.190](https://doi.org/10.1287/ijoc.1.3.190).
7. Fred Glover (1990). "Tabu Search – Part 2". *ORSA Journal on Computing*. **2** (1): 4–32. [doi:10.1287/ijoc.2.1.4](https://doi.org/10.1287/ijoc.2.1.4).
8. [http://www.ise.ncsu.edu/fangroup/ie789.dir/IE789F\\_tabu.pdf](http://www.ise.ncsu.edu/fangroup/ie789.dir/IE789F_tabu.pdf)
9. F. Glover; M. Laguna (1997). *Tabu Search*. Kluwer Academic Publishers.
10. Fred Glover (1990). "Tabu Search: A Tutorial". *Interfaces*.
11. M. Malek; M. Huruswamy; H. Owens; M. Pandya (1989). *Serial and parallel search techniques for the traveling salesman problem*. *Annals of OR: Linkages with Artificial Intelligence*.
12. F. Glover, M. Laguna & R. Marti (2000). *Fundamentals of Scatter Search and Path Relinking*. *Control and Cybernetics*. **29**. pp. 653–684.
13. M. Laguna & R. Marti (2003). *Scatter Search: Methodology and Implementations in C*. Kluwer Academic Publishers, Boston.
14. D. Gamboa, C. Rego & F. Glover (2005). "Data Structures and Ejection Chains for Solving Large Scale Traveling Salesman Problems". *European Journal of Operational Research*. **160** (1): 154–171. [doi:10.1016/j.ejor.2004.04.023](https://doi.org/10.1016/j.ejor.2004.04.023).