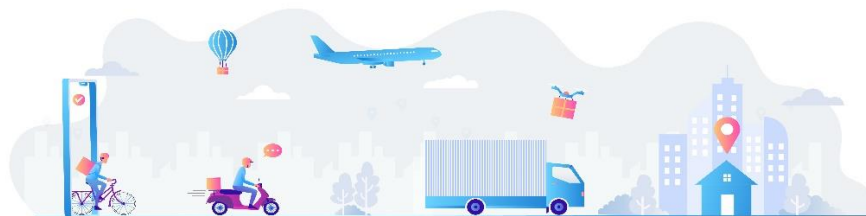




ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

ΑΛΓΟΡΙΘΜΟΣ ΜΕΤΑΒΛΗΤΗΣ ΓΕΙΤΟΝΙΑΣ ΑΝΑΖΗΤΗΣΗΣ ΓΙΑ ΤΟ ΠΡΟΒΛΗΜΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΓΙΑ ΤΑΥΤΟΧΡΟΝΕΣ ΔΙΑΝΟΜΕΣ ΚΑΙ ΠΑΡΑΛΑΒΕΣ ΠΡΟΪΟΝΤΩΝ.

Variable Neighborhood Search for the Vehicle Routing Problem with
Simultaneous Pickup and Delivery.



Νακόπουλος Νικόλαος

Επιβλέπων Καθηγητής: Δρ. Ιωάννης Μαρινάκης

Χανιά, Ιούνιος 2021

Ευχαριστίες

Αυτή η εργασία αποτελεί το τελευταίο μου εγχείρημα ως προπτυχιακός φοιτητής στο τμήμα Μηχανικών Παραγωγής και Διοίκησης του Πολυτεχνείου Κρήτης. Γι' αυτό το λόγο θα ήθελα να ευχαριστήσω όλους εκείνους που συνέβαλαν στη επίτευξη του στόχου μου. Αρχικά θα ήθελα να εκφράσω τις ιδιαίτερες ευχαριστίες μου στον Επιβλέποντα Καθηγητή μου κ. Ιωάννη Μαρινάκη για την καθοδήγηση και τη βοήθεια που μου προσέφερε, αλλά και για το ενδιαφέρον του στην πραγματοποίηση της εργασίας αυτής. Ακόμα, ένα τεράστιο ευχαριστώ στο φίλο και υποψήφιο διδάκτορα κ. Τσακιράκη Ελευθέριο, για τη συνεχή και ανιδιοτελή υποστήριξη του, καθώς και για την άριστη συνεργασία που υπήρξε μεταξύ μας όλο αυτό το διάστημα. Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για την ευκαιρία που μου έδωσαν να σπουδάσω στα πανέμορφα Χανιά καθώς και τους φίλους και κοντινούς μου ανθρώπους που μου προσέφεραν μοναδικές εμπειρίες, δημιουργώντας υπέροχες αναμνήσεις.

Πίνακας περιεχομένων

Ευχαριστίες	2
Πίνακας περιεχομένων	3
Περίληψη	6
Κεφάλαιο 1: Εισαγωγή	7
1.1 Εφοδιαστική Αλυσίδα (Supply Chain)	7
1.2 Διοίκηση Εφοδιαστικής Αλυσίδας (Supply Chain Management ή Logistics).....	8
1.3 Βασικές Λειτουργίες Εφοδιαστικής Αλυσίδας	8
Κεφάλαιο 2: Δρομολόγηση Οχημάτων.....	9
2.1 Το Πρόβλημα Δρομολόγησης Οχημάτων (Vehicle Routing Problem)	9
2.2 Πρόβλημα Δρομολόγησης Οχημάτων με Ταυτόχρονη Διανομή και Παραλαβή Προϊόντων κατά τη διάρκεια της διαδρομής (VRPSPD)	11
2.3 Άλλα προβλήματα δρομολόγησης οχημάτων	12
2.3.1 Δρομολόγηση οχημάτων με κέρδος (VRP with profits).....	12
2.3.2 Η ύπαρξη πολλαπλών αποθηκών (MDVR).	12
2.3.3 Πρόβλημα δρομολόγησης οχημάτων με δύο είδη πελατών κατά τη διάρκεια της διαδρομής (VRPB).	13
Κεφάλαιο 3: Ευρετικοί και Μεθευρετικοί Αλγόριθμοι.....	14
3.1 Απλοί Ευρετικοί Αλγόριθμοι	14
3.1.1 Αλγόριθμοι τοπικής αναζήτησης (local search algorithms).....	15
3.2 Μεθευρετικοί αλγόριθμοι	16
3.2.1 Αλγόριθμος μεταβλητής γειτονιάς αναζήτησης (Variable Neighborhood Search) 17	
3.2.2 Άλλοι αλγόριθμοι	18
Κεφάλαιο 4: Περιγραφή και Επίλυση του προβλήματος VRPSPD	19
4.1 Περιγραφή του προβλήματος.....	19
4.2 Παραγωγή αρχικών λύσεων μέσω του αλγορίθμου των εξοικονομήσεων των Clarke & Wright.....	20
4.3 Βελτιστοποίηση της αρχικής λύσης μέσω του αλγορίθμου Μεταβλητής Γειτονιάς Αναζήτησης (VNS)	21
4.3.1 Ο αλγόριθμος τοπικής αναζήτησης 1-0 relocate.....	22
4.3.2 Ο αλγόριθμος τοπικής αναζήτησης 1-1 exchange.....	24
4.3.3 Ο αλγόριθμος τοπικής αναζήτησης 2-opt.....	26
Κεφάλαιο 5: Αποτελέσματα.....	29
5.1 Προβλήματα	30

5.1.1 Πρόβλημα r101	30
5.1.2 Πρόβλημα r201	32
5.1.3 Πρόβλημα c201	34
5.1.4 Πρόβλημα R1_2_1	36
5.1.5 Πρόβλημα R2_2_1	38
5.1.6 Πρόβλημα C2_2_1.....	40
5.1.7 Πρόβλημα RC2_2_1	42
5.1.8 Πρόβλημα C1_4_1.....	44
Κεφάλαιο 6: Συγκρίσεις Αποτελεσμάτων και Συμπεράσματα	46
6.1 Αλγόριθμος των εξοικονομήσεων των Clarke & Wright.....	47
6.2 Αλγόριθμος μεταβλητής γειτονίας αναζήτησης (VNS)	49
6.3 Τελικά Συμπεράσματα	54
Βιβλιογραφία.....	55

Κατάλογος εικόνων

<i>Εικόνα 1.1:</i> Στάδια Εφοδιαστικής Αλυσίδας.....	7
<i>Εικόνα 1.2:</i> Λειτουργίες Εφοδιαστικής Αλυσίδας.....	9
<i>Εικόνα 2.1:</i> Πρόβλημα Δρομολόγησης Οχημάτων (VRP).....	11
<i>Εικόνα 2.2:</i> Σχέσεις μεταξύ διαφόρων VRP προβλημάτων.....	14
<i>Εικόνα 3.1:</i> Αλγόριθμος 1-0 Relocate.....	23
<i>Εικόνα 3.2:</i> Αλγόριθμος 1-1 exchange.....	25
<i>Εικόνα 3.3:</i> Αλγόριθμος 2-opt.....	27

Κατάλογος Γραφημάτων

<i>Γράφημα 1:</i> Αρχική λύση r101.....	30
<i>Γράφημα 2:</i> Τελική λύση r101.....	31
<i>Γράφημα 3:</i> Αρχική λύση r201.....	32
<i>Γράφημα 4:</i> Τελική λύση r201.....	33
<i>Γράφημα 5:</i> Αρχική λύση c201.....	34
<i>Γράφημα 6:</i> Τελική λύση c201.....	35
<i>Γράφημα 7:</i> Αρχική λύση R1_2_1.....	36
<i>Γράφημα 8:</i> Τελική λύση R1_2_1.....	37
<i>Γράφημα 9:</i> Αρχική λύση R2_2_1.....	38
<i>Γράφημα 10:</i> Τελική λύση R2_2_1.....	39
<i>Γράφημα 11:</i> Αρχική λύση C2_2_1.....	40
<i>Γράφημα 12:</i> Τελική λύση C2_2_1.....	41
<i>Γράφημα 13:</i> Αρχική λύση RC2_2_1.....	42
<i>Γράφημα 14:</i> Τελική λύση RC2_2_1.....	43
<i>Γράφημα 15:</i> Αρχική λύση C1_4_1.....	44
<i>Γράφημα 16:</i> Τελική λύση C1_4_1.....	45

Κατάλογος Πινάκων

<i>Πίνακας 5.1:</i> Χαρακτηριστικά Προβλημάτων.....	29
<i>Πίνακας 6.1:</i> Σύγκριση αρχικού Clarke & Wright με το βελτιωμένο Clarke & Wright.....	48
<i>Πίνακας 6.2:</i> Σύγκριση Αρχικών λύσεων με τη μέθοδο APA.....	49
<i>Πίνακας 6.3:</i> Σύγκριση αλγορίθμου VNS(1) με τη μέθοδο APA.....	51
<i>Πίνακας 6.4:</i> Σύγκριση αλγορίθμου VNS(1,2) με τη μέθοδο APA.....	52
<i>Πίνακας 6.5:</i> Σύγκριση Τελικών λύσεων (VNS) με τη μέθοδο APA.....	53

Περίληψη

Η συγκεκριμένη διπλωματική εργασία έχει ως αντικείμενο μελέτης το πρόβλημα Δρομολόγησης Οχημάτων με Ταυτόχρονη Διανομή και Παραλαβή Προϊόντων Κατά τη Διάρκεια της Διαδρομής (VRPSPD). Γενικά στα προβλήματα δρομολόγησής οχημάτων (VRP) ο κύριος στόχος είναι να ελαχιστοποιηθεί το κόστος (εξυπηρέτησης των πελατών ή μεταφοράς των προϊόντων ή διανομής ή παραλαβής των προϊόντων). Στο πρόβλημα της παρούσας εργασίας για το (VRPSPD), ένας πελάτης μπορεί να έχει την απαίτηση να του διανεμηθούν προϊόντα από την αποθήκη αλλά και να παραδώσει στο όχημα προϊόντα που θέλει να μεταφερθούν στην αποθήκη. Στη συνέχεια κατασκευάζεται αλγόριθμος για την επίλυση του παραπάνω προβλήματος. Αρχικά μέσω του αλγορίθμου των εξοικονομήσεων Clarke & Wright υπολογίζονται οι εξοικονομήσεις των πελατών και στη συνέχεια δημιουργούνται οι διαδρομές βάσει των καλύτερων εξοικονομήσεων. Οι διαδρομές που προκύπτουν με τη παραπάνω μέθοδο αποτελούν τις αρχικές μας λύσεις για το πρόβλημα. Στη συνέχεια αφού έχουν κατασκευαστεί οι διαδρομές-γειτονίες υλοποιείται ο αλγόριθμος Μεταβλητής Γειτονίας Αναζήτησης (VNS), στον οποίο εφαρμόζονται διαφορετικοί μέθοδοι τοπικής αναζήτησης ο 1-1 exchange, ο 1-0 relocate και ο 2-opt. Η εκμετάλλευση αυτού του γεγονότος μπορεί να οδηγήσουν σε διαφορετικά τοπικά βέλτιστα. Τέλος, υλοποιούμε τον αλγόριθμο σε 18 προβλήματα και συγκρίνουμε τα αποτελέσματά μας με τα βέλτιστα αυτών των παραδειγμάτων. Ο αλγόριθμος υλοποιείται στο προγραμματιστικό περιβάλλον MATLAB.

Κεφάλαιο 1: Εισαγωγή

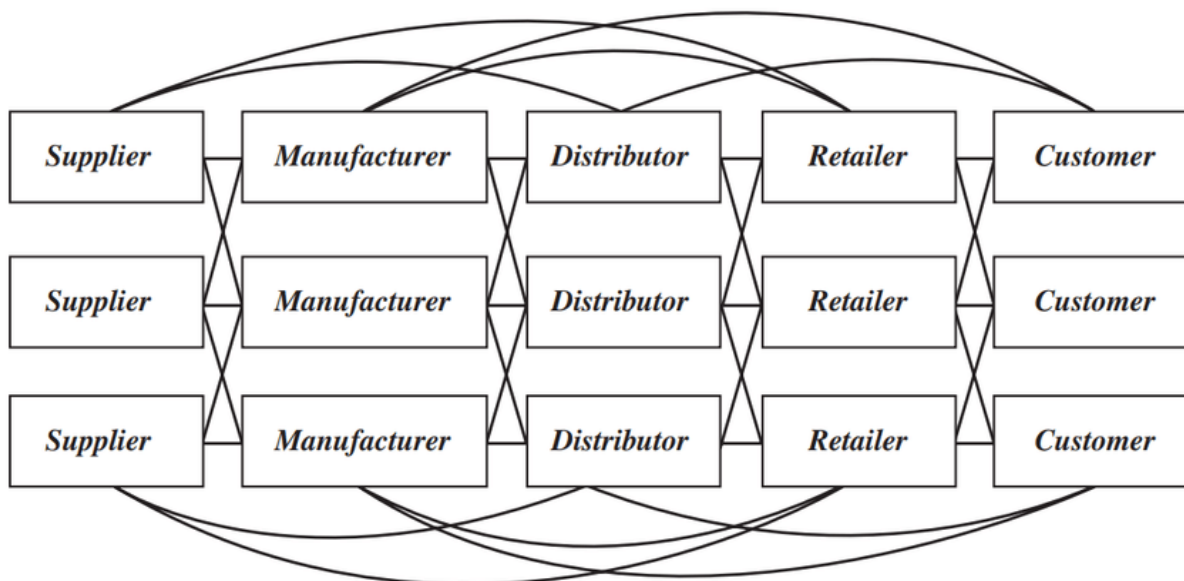
1.1 Εφοδιαστική Αλυσίδα (Supply Chain)

Με τον όρο Εφοδιαστική Αλυσίδα αναφερόμαστε στη διαδικασία μεταφοράς ενός προϊόντος ή μίας υπηρεσίας από το σημείο παραγωγής στο σημείο παράδοσης και έχει ως σκοπό τη μείωση του λειτουργικού κόστους της μεταφοράς των προϊόντων και την αύξηση της ικανοποίησης των πελατών. (2)

Συγκεκριμένα, η εφοδιαστική αλυσίδα αποτελείται από τα στάδια που εμπλέκονται, άμεσα ή έμμεσα, στη εκπλήρωση της απαίτησης ενός πελάτη. Η εφοδιαστική αλυσίδα δεν περιλαμβάνει μόνο κατασκευαστές και προμηθευτές αλλά και αποθήκες, μεταφορείς πωλητές, ακόμα και τους πελάτες. Μέσα σε κάθε οργανισμό όπως σε ένα κατασκευαστή, η εφοδιαστική αλυσίδα περιλαμβάνει όλες εκείνες τις λειτουργίες που εμπλέκονται στην παραλαβή και εκπλήρωση τη απαίτησης ενός πελάτη. Αυτές οι λειτουργίες περιλαμβάνουν, αλλά δεν περιορίζονται σε αυτές, την ανάπτυξη νέων προϊόντων, τη διαφήμιση, τις λειτουργίες, τη διανομή, τα οικονομικά και την εξυπηρέτηση πελατών. (Chopra S., 2007)

Μια τυπική εφοδιαστική αλυσίδα αποτελείται από τα παρακάτω στάδια:

1. Πελάτες
2. Εμπόρους Λιανικής
3. Εμπόρους Χονδρικής/Διανομείς
4. Κατασκευαστές
5. Προμηθευτές



Εικόνα 1.1: Στάδια Εφοδιαστικής Αλυσίδας. (Ιωάννης Μαρινάκης Μ. Μ., 2019)

1.2 Διοίκηση Εφοδιαστικής Αλυσίδας (Supply Chain Management ή Logistics)

Με τον όρο Logistics αναφερόμαστε στο τρόπο απόκτησης, διαχείριση και μεταφοράς πληροφοριών και υλικών μεταξύ των προμηθευτών και των πελατών. Με απλά λόγια η διοίκηση της εφοδιαστικής αλυσίδας εξασφαλίζει την ομαλή και επιτυχής λειτουργία της εφοδιαστικής αλυσίδας επικεντρώνοντας στη μεταφορά του σωστού ποσού πόρου, στην κατάλληλη τοποθεσία και χρονική στιγμή, σε άριστη κατάσταση και στο σωστό πελάτη.

το council of logistics management ορίζει τον όρο διαχείριση εφοδιαστικής (logistics management) ως την διαδικασία προγραμματισμού, υλοποίησης και ελέγχου για την επιτυχής ροή και αποθήκευση αγαθών και υπηρεσιών από την πηγή στο σημείο κατανάλωσης με σκοπό την εξυπηρέτηση των πελατών. Ο ορισμός αυτός περιέχει ροές τόσο υλικών όσο και υπηρεσιών και συνεπώς απευθύνεται τόσο σε διαδικασίες στον τομέα των δημόσιων υπηρεσιών και τραπεζών όσο και στον βιομηχανικό τομέα.

1.3 Βασικές Λειτουργίες Εφοδιαστικής Αλυσίδας

Βασικές Λειτουργίες της Εφοδιαστικής Αλυσίδας περιλαμβάνουν:

1. Προμήθειες
2. Παραγωγή
3. **Μεταφορές**
4. Αποθήκευση
5. **Διανομή**
6. Εξυπηρέτηση Πελατών

Η συγκεκριμένη εργασία ασχολείται αποκλειστικά με τη διανομή των προϊόντων.

Διανομή: Ο όρος διανομή αναφέρεται στο τρόπο μεταφοράς (μεταφορικά μέσα και διαδρομή μεταφοράς) που χρησιμοποιείται για να μετακινηθούν τα προϊόντα από το τόπο παραγωγής στις αποθήκες και από εκεί προς τους πελάτες. Τα δίκτυα διανομής αποτελούν τις αρτηρίες μιας επιχείρησης για την συνεχή και επιτυχημένη τροφοδοσία της αγοράς με καταναλωτικά προϊόντα και έχουν αποκτήσει μεγάλη βαρύτητα για την επιβίωση της επιχείρησης και τη μείωση των λειτουργικών της εξόδων. (2)



Εικόνα 1.2: Λειτουργίες Εφοδιαστικής Αλυσίδας. (1)

Κεφάλαιο 2: Δρομολόγηση Οχημάτων

2.1 Το Πρόβλημα Δρομολόγησης Οχημάτων (Vehicle Routing Problem)

Το Πρόβλημα της Δρομολόγησης Οχημάτων (VRP) παρουσιάστηκε από τους Dantzing & Ramser το 1959, ενώ σήμερα είναι ένα από τα πιο σημαντικά προβλήματα διανομής της εφοδιαστικής αλυσίδας. Το “πρόβλημα δρομολόγησης οχημάτων” έχει δοθεί σε ένα σύνολο προβλημάτων στα οποία πρέπει να ορίσουμε έναν αριθμό διαδρομών, σε ένα πλήθος οχημάτων τα οποία έχουν ως βάση μία ή περισσότερες αποθήκες και τα οποία πρέπει να εξυπηρετήσουν έναν αριθμό γεωγραφικά διασκορπισμένων πελατών και έχει ως στόχο τη μείωση του κόστους μεταφοράς των αγαθών από την αποθήκη στους πελάτες και το αντίστροφο.

Η διανομή προϊόντων αφορά την εξυπηρέτηση των πελατών από ένα πλήθος οχημάτων σε μια δεδομένη χρονική περίοδο, ξεκινώντας από την αποθήκη και οδηγώντας στο υπάρχων οδικό δίκτυο παραδίδονται τα προϊόντα στους τελικούς πελάτες. Στη συνέχεια για να λυθεί σωστά ένα πρόβλημα δρομολόγησης οχημάτων είναι ανάγκη να καθοριστεί ένα σύνολο από διαδρομές, όπου κάθε μια να ξεκινάει και να τερματίζει σε μια αποθήκη, ικανοποιώντας τις ανάγκες των πελατών, μη παραβιάζοντας κάποιον από τους περιορισμούς και έχοντας βελτιώσει το κόστος της διανομής. (Ιωάννης Μαρινάκης Μ. Μ., 2019)

Ένας ειδικός για την επίλυση ενός προβλήματος διανομής θα πρέπει να εστιάσει στη φύση του προβλήματος καθώς και στο μέγεθος της εταιρίας που εξετάζεται. Μερικές πληροφορίες που θα πρέπει να λάβει υπόψη κάποιος σχετικά με τις δραστηριότητες διανομής είναι:

- Το πλήθος των διαδρομών που υλοποιούνται καθημερινά.
- Ο αριθμός των οδηγών που διαθέτει η εταιρία.
- Το μέγεθος του στόλου των οχημάτων.
- Τα χιλιόμετρα που διανύονται εντός και εκτός αστικού κύκλου.
- Οι συνολικές δαπάνες για τις δραστηριότητες διανομής.
- Οι απαιτήσεις που χρειάζονται για πιθανές βλάβες στο μέλλον.
- Η υπολογιστική δύναμη της εταιρίας για τη δυνατότητα υποστήριξης του δικτύου διανομής.

Εκτός από τα γενικά χαρακτηριστικά του συστήματος διανομής μιας εταιρίας, πιο συγκεκριμένα υπάρχουν και χαρακτηριστικά για τους πελάτες και για τα οχήματα. Ένας ειδικός λοιπόν θα πρέπει να εξετάσει και αυτά για την επίλυση του προβλήματος. Όπως είναι:

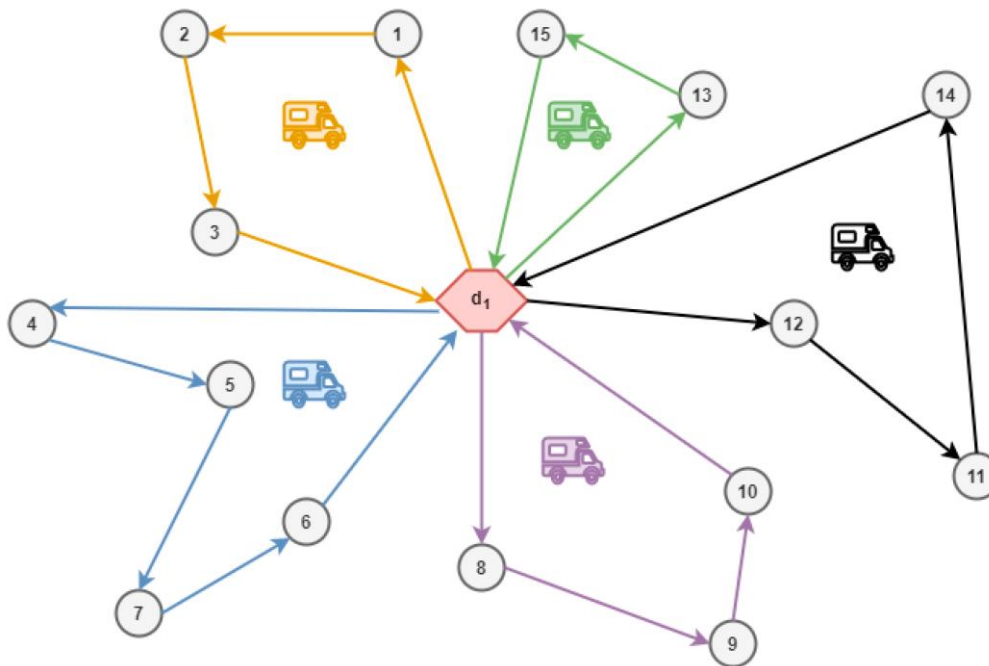
- Η ακριβής τοποθεσία που βρίσκεται ο πελάτης.
- Ο αριθμός και το είδος των προϊόντων που βρίσκονται προς παράδοση ή περισυλλογή από το πελάτη.
- Τα χρονικά παράθυρα, δηλαδή οι ώρες της ημέρας στις οποίες ένας πελάτης μπορεί να εξυπηρετηθεί.
- Το χρονικό διάστημα που χρειάζεται για τη φόρτωση και εκφόρτωση των αγαθών σε κάθε πελάτη.
- Τα είδη των οχημάτων που μπορούν να χρησιμοποιηθεί από την εταιρία για την αποτελεσματική εξυπηρέτηση των πελατών.

Παρότι γνωρίζουμε αρκετά για τα χαρακτηριστικά αυτών των προβλημάτων πολλές φορές δεν πετυχαίνετε η πλήρης ικανοποίηση των πελατών. Για να αποφευχθεί όμως αυτό Σε αυτές τις περιπτώσεις η ποσότητα των προϊόντων που διανέμονται ή συλλέγονται μπορεί να μειωθεί ή μερικοί πελάτες να μην μπορέσουν να εξυπηρετηθούν. Για την αντιμετώπιση αυτών των καταστάσεων διάφορες προτεραιότητες που σχετίζονται με τη μερική ή την πλήρη έλλειψη εξυπηρέτησης μπορούν να ανατεθούν στους πελάτες. (Ιωάννης Μαρινάκης Μ. Μ., 2019)

Οι διαδρομές για την εξυπηρέτηση των πελατών ξεκινούν και τελειώνουν σε μια ή περισσότερες αποθήκες. Κάθε αποθήκη χαρακτηρίζεται 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:

- Η αποθήκη προέλευσης του οχήματος και αν η αποθήκη που θα καταλήξει είναι διαφορετική.
- Η μέγιστη χωρητικότητα του οχήματος που μπορεί να φορτωθεί.
- Αν το όχημα χωρίζεται σε επιμέρους τμήματα.
- Αν το όχημα μπορεί να διαιρεθεί σε ομάδες ανάλογα με το είδος του προϊόντος ή αν ο χώρος διατίθεται για τα προϊόντα για παραλαβή ή προς παράδοση.
- Αν το όχημα διαθέτει συρόμενη πόρτα ή υπάρχουν μηχανήματα για να βοηθήσουν στη φόρτωση και στην εκφόρτωση του οχήματος.

- Οι δρόμοι στους οποίους μπορεί να περάσει το συγκεκριμένο όχημα.
- Τα συνολικά έξοδα του οχήματος για τη μεταφορά των αγαθών.



Εικόνα 2.1: Πρόβλημα Δρομολόγησης Οχημάτων (VRP). (3)

2.2 Πρόβλημα Δρομολόγησης Οχημάτων με Ταυτόχρονη Διανομή και Παραλαβή Προϊόντων κατά τη διάρκεια της διαδρομής (VRPSPD)

Σε αυτό το πρόβλημα μπορεί να παραληφθούν προϊόντα από κάποιον πελάτη ή ακόμα και να θέλει να του διανεμηθούν προϊόντα από το συγκεκριμένο όχημα. Για κάθε πελάτη λοιπόν, έχουμε δύο μεγέθη, τα προϊόντα που πρέπει να διανεμηθούν στους πελάτες (d_i) και τα αγαθά που πρόκειται να παραληφθούν από τους πελάτες (p_i) με προορισμό την αποθήκη. Όμως τις περισσότερες φορές αυτό που μας ενδιαφέρει είναι η ποσότητα που δείχνει την διαφορά ανάμεσα στις δύο ποσότητες $q_i = d_i - p_i$. Επίσης για κάθε πελάτη ορίζονται και δύο κόμβοι, ο κόμβος από τον οποίο αρχίζουν τα οχήματα με τα προϊόντα που επρόκειτο να διανεμηθούν στους πελάτες O_i και ο κόμβος που καταλήγουν τα προϊόντα που συλλέγονται από τους πελάτες D_i . Πολλές φορές αυτοί οι δύο κόμβοι ταυτίζονται και χαρακτηρίζεται ως η κεντρική αποθήκη. Ο κύριος περιορισμός που υπάρχει σε αυτό το πρόβλημα είναι ότι σε κάθε πελάτη η διανομή των προϊόντων γίνεται πριν από την παραλαβή, έτσι η συνολική φόρτωση ενός οχήματος όταν φύγει από έναν πελάτη, ισούται με το φορτίο του οχημάτων πριν φτάσει στο πελάτη μείον τα προϊόντα που θα παραδοθούν συν τα προϊόντα που θα παραληφθούν από τον πελάτη στο όχημα. (Ιωάννης Μαρινάκης Μ. Μ., 2019)

Τα κύρια χαρακτηριστικά του προβλήματος είναι:

- Κάθε διαδρομή ξεκινάει και καταλήγει σε αποθήκη.
- Κάθε όχημα αντιστοιχεί μόνο σε μία διαδρομή.
- Κάθε πελάτης επισκέπτεται από μόνο ένα όχημα.
- Η συνολική ποσότητα που μεταφέρει ένα όχημα πρέπει να μην ξεπερνάει τη μέγιστη χωρητικότητα του οχήματος και να είναι θετική.
- Για κάθε πελάτη i ο κόμβος O_i , αν διαφέρει από την αποθήκη πρέπει να εξυπηρετηθεί πριν από τον πελάτη i και στην ίδια διαδρομή.
- Για κάθε πελάτη i ο κόμβος D_i , αν διαφέρει από την αποθήκη πρέπει να εξυπηρετηθεί μετά από τον πελάτη i και στην ίδια διαδρομή.

2.3 Άλλα προβλήματα δρομολόγησης οχημάτων

2.3.1 Δρομολόγηση οχημάτων με κέρδος (VRP with profits)

Στο συγκεκριμένο πρόβλημα δρομολόγησης οχημάτων με κέρδος ή Vehicle Routing Problem with profits έχουμε ένα σύνολο από κόμβους που αποτελούν τους πελάτες μιας επιχείρησης, οι οποίοι έχουν συγκεκριμένη ζήτηση και αποστάσεις μεταξύ τους. Στόχος του προβλήματος είναι να εξυπηρετηθούν πελάτες οι οποίοι θα μας οδηγήσουν σε μέγιστο κέρδος χωρίς να παραβιάζουμε κάποιο από τους περιορισμούς του συνολικού φορτίου και συνολικού χρόνου εξυπηρέτησης.

Ο στόχος του προβλήματος είναι η κατασκευή διαδρομών χαμηλού κόστους μία διαδρομή για κάθε όχημα χωρίς να παραβιαστεί κάποιος από τους περιορισμούς του συνολικού φορτίου των οχημάτων και του συνολικού χρόνου εξυπηρέτησης (χρόνος από την αποθήκη και πίσω σε αυτή), έτσι ώστε να ωφεληθεί η επιχείρηση από το βελτιωμένο κόστος παράδοσης λόγω της συνολικής απόστασης που έχει καλυφθεί, του χρόνου παράδοσης των προϊόντων και το κέρδος από κάθε πελάτη. Κάθε διαδρομή αποτελεί ένα σύνολο (ακολουθία) τοποθεσιών (κόμβων) που πρέπει να επισκεφθούν τα οχήματα παράδοσης για την κάλυψη της ζήτησης. Όταν τα οχήματα επισκεφθούν όλους τους κόμβους της διαδρομής τότε γυρνάνε πίσω στην αποθήκη.

2.3.2 Η ύπαρξη πολλαπλών αποθηκών (MDVR).

Το συγκεκριμένο πρόβλημα αποτελεί παραλλαγή του προβλήματος δρομολόγησης οχημάτων, συγκεκριμένα χρησιμοποιούνται παραπάνω από μια αποθήκες:

Οι δύο τρόποι επίλυσης αυτού του προβλήματος είναι:

1. κάθε αποθήκη έχει τον δικό της αριθμό οχημάτων και πελατών.
2. κάθε όχημα ξεκινάει από μια αποθήκη, αλλά μπορεί να επισκεφθεί και άλλες αποθήκες είτε για να παραδώσει ή να παραλάβει προϊόντα και να συνεχίσει την εξυπηρέτηση των πελατών, είτε να τερματίσει σε άλλη αποθήκη.

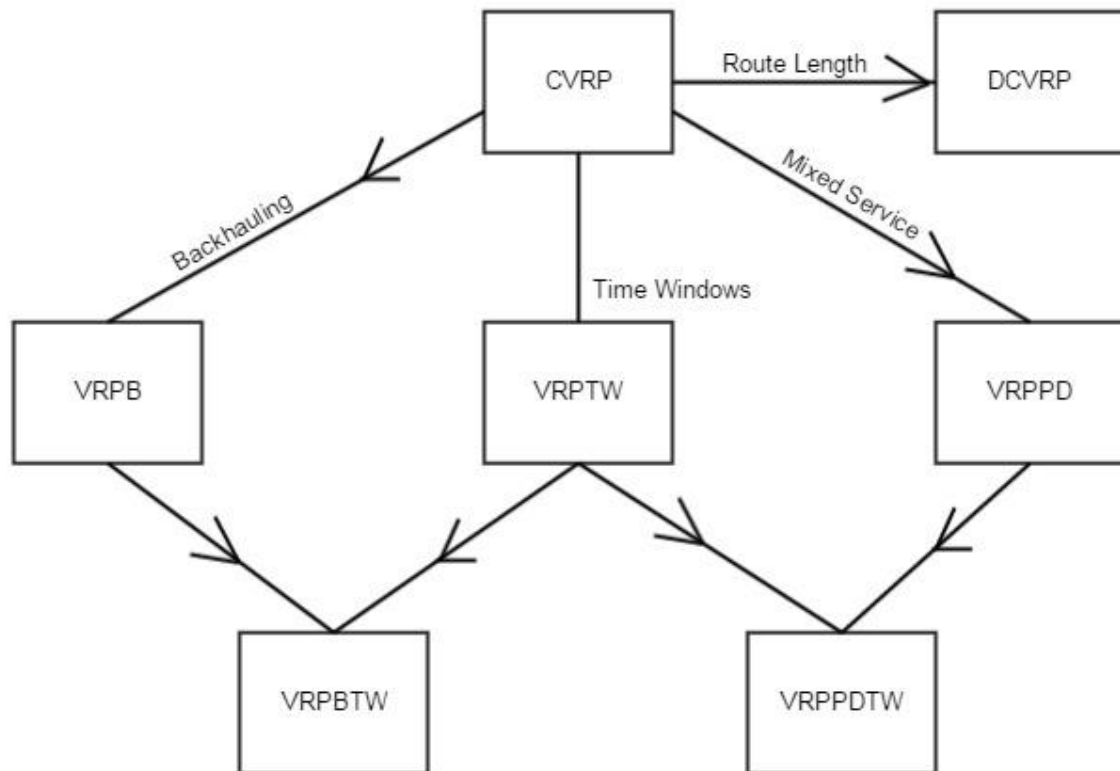
Αρχικά το πρόβλημα μπορεί να λυθεί σε 2 φάσεις, την ομαδοποίηση των πελατών με τις αποθήκες, δηλαδή ο κάθε πελάτης να ανήκει σε μια μόνο αποθήκη και τη δημιουργία των δρομολογίων ανάλογα με τα οχήματα που διαθέτει η κάθε αποθήκη. Η δυσκολία αυτού του προβλήματος είναι στην ανάθεση των πελατών σε κάθε αποθήκη και γι' αυτό το λόγο χρειάζεται ιδιαίτερη προσοχή σε αυτό το σημείο.

2.3.3 Πρόβλημα δρομολόγησης οχημάτων με δύο είδη πελατών κατά τη διάρκεια της διαδρομής (VRPB).

Αυτό το πρόβλημα αποτελεί ακόμα μια επέκταση του προβλήματος δρομολόγησης οχημάτων όπου οι πελάτες χωρίζονται σε δύο ομάδες. Στη πρώτη ομάδα ανήκουν οι πελάτες που πρέπει να τους διανεμηθούν προϊόντα (linehauls customers) και στη δεύτερη ομάδα οι πελάτες που θέλουν να παραδώσουν προϊόντα. Επίσης τα δύο είδη πελατών μπορούν να αναμειχθούν μέσα σε μία διαδρομή αλλά σε καμία περίπτωση ένας πελάτης δεν θέλει ταυτόχρονα παραλαβή και παράδοση προϊόντων. Τέλος οι πελάτες που τους διανέμονται προϊόντα εξυπηρετούνται πρώτα και έπειτα οι πελάτες της δεύτερης ομάδας.

Για το πρόβλημα VRPB ισχύουν:

- Τα οχήματα ακολουθούν μια διαδρομή και καταλήγουν στην αποθήκη από όπου ξεκίνησαν.
- Δεν πρέπει να ξεπεραστεί η χωρητικότητα του οχήματος και για τις δύο ομάδες.
- Κάθε πελάτης επισκέπτεται μία φορά.
- Η μέγιστη απόσταση που μπορεί να διανύσει ένα όχημα δεν πρέπει να ξεπερνιέται.
- Οι πελάτες της πρώτης ομάδας εξυπηρετούνται πρώτοι.
- Δεν επιτρέπονται διαδρομές που εξυπηρετούν μόνο πελάτες από τη δεύτερη ομάδα.
- Πρέπει να μειωθεί η συνολική απόσταση που θα διανύσουν τα οχήματα.



Εικόνα 2.2: Σχέσεις μεταξύ διαφόρων VRP προβλημάτων. (4)

Κεφάλαιο 3: Ευρετικοί και Μεθευρετικοί Αλγόριθμοι

3.1 Απλοί Ευρετικοί Αλγόριθμοι

Για τη βελτιστοποίηση της λύσης ενός προβλήματος ή την εύρεση της βέλτιστης λύσης μπορεί να χρησιμοποιηθούν διάφοροι αλγόριθμοι δεν υπάρχει αλγόριθμος που να δουλεύει εξίσου καλά σε όλα τα προβλήματα. Γι' αυτό το λόγο έχουν αναπτυχθεί εκατοντάδες αλγόριθμοι και παραλλαγές του για την επίλυση των προβλημάτων δρομολόγησης οχημάτων.

Αρχικά οι πρώτοι αλγόριθμοι που συναντάμε είναι οι ευρετικοί αλγόριθμοι καθώς στις περισσότερες φορές είναι αυτοί που κατασκευάζουν την αρχική μας λύση ή την βελτιώνουν. Οι ευρετικοί αλγόριθμοι χωρίζονται σε:

- Αλγόριθμοι απληστίας (greedy algorithms).
Σκοπός τους είναι η κατασκευή μιας αρχικής λύσης αλλά στις περισσότερες περιπτώσεις χρειάζονται αρκετό χρόνο για την υλοποίηση τους και δεν συνιστώνται συχνά.
- Προσεγγιστικοί αλγόριθμοι (approximation algorithms).

Σκοπός τους είναι η λύση του προβλήματος χρησιμοποιώντας κάποιων επιπλέον πληροφοριών.

- Αλγόριθμοι τοπικής αναζήτησης (local search algorithms).
Σκοπός τους είναι η βελτιστοποίηση μιας ήδη υπάρχουσας λύσης, οπότε για να εφαρμοστεί ένας αλγόριθμος τοπικής αναζήτησης πρέπει να υπάρχει μια εφικτή αρχική λύση.

3.1.1 Αλγόριθμοι τοπικής αναζήτησης (local search algorithms).

Ένας αλγόριθμος τοπικής αναζήτησης ξεκινάει από μια αρχική εφικτή λύση και προσπαθεί να βελτιώσει τη λύση με μια μέθοδο αναζήτησης στη γειτονιά της λύσης. Χρησιμοποιείται η τεχνική της επαναληπτικής βελτίωσης και λειτουργεί ως εξής:

1. Αρχικά επιλέγεται μια λύση από το χώρο και αποτελεί τη τρέχουσα λύση.
2. Ετοιμάζεται ένας μετασχηματισμός στη τρέχουσα λύση για να εξαχθεί μια νέα λύση.
3. Αξιολογείται η νέα λύση.
4. Εάν η νέα λύση είναι καλύτερη από τη τρέχουσα τότε ανταλλάσσεται με την τρέχουσα αλλιώς απορρίπτεται.
5. Τα παραπάνω βήματα επαναλαμβάνονται έως ότου κανένας μετασχηματισμός να μη μπορεί να βελτιώσει τις λύσεις.

διαδικασία *local_search*

begin

t μια αρχική λύση του προβλήματος

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

t=improve(t)

return *t*

end

(Ιωάννης Μαρινάκης Α. Μ., 2008)

Οι δημοφιλέστεροι αλγόριθμοι τοπικής αναζήτησης και αυτοί που θα χρησιμοποιηθούν στη συγκεκριμένη διπλωματική εργασία είναι οι εξής:

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

Θα αναλυθούν περαιτέρω στο επόμενο κεφάλαιο.

3.2 Μεθευρετικοί αλγόριθμοι

Όταν γίνεται η εφαρμογή μια μεθόδου τοπικής αναζήτησης είναι πολύ εύκολο να κολλήσουμε σε κάποιο τοπικό ελάχιστο, ειδικά αν η αρχική μας λύση είναι κοντά σε ένα τοπικό ελάχιστο. Η εφαρμογή της τοπικής αναζήτησης θα μας οδηγήσει σε αυτό το τοπικό ελάχιστο. Σε αυτή τη περίπτωση είναι πιθανό να παγιδευτεί ο αλγόριθμος μας σε αυτή τη λύση και να μη μπορέσει να βελτιώσει περαιτέρω τη λύση μας. Για την επίλυση αυτού του προβλήματος έχουν προταθεί διάφοροι αλγόριθμοι που συμβάλλουν ώστε ο αλγόριθμος μας να ξεφύγει από αυτό το τοπικό ελάχιστο. Οι αλγόριθμοι αυτοί ονομάζονται μεθευρετικοί αλγόριθμοι.

Οι μεθευρετικοί αλγόριθμοι χωρίζονται σε 4 κύριες κατηγορίες αναλόγως με το τρόπο που ξεπερνάνε το τοπικό ελάχιστο.

1. Επαναληπτικές διαδικασίες που αρχίζουν από διαφορετικές αρχικές λύσεις.
 - ❖ Η μέθοδος της διαδικασίας άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης.
 - ❖ Αλγόριθμοι πολυεναρκτήριας τοπικής αναζήτησης.
 - ❖ Αλγόριθμοι της επαναληπτικής τοπικής αναζήτησης.
2. Αλγόριθμοι που δέχονται γειτονικές κινήσεις που δεν βελτιώνουν τη λύση. Με αυτόν τον τρόπο μπορεί στις επόμενες κινήσεις να ξεπεράσουμε το τρέχων τοπικό ελάχιστο και να οδηγηθούμε σε επόμενο τοπικό ελάχιστο με καλύτερη λύση από το προηγούμενο.
 - ❖ Προσομοιωμένη ανόπτηση.
 - ❖ Περιορισμένη αναζήτηση.
3. Αλγόριθμοι που αλλάζουν τη γειτονιά αναζήτησης. Όταν βρεθούν σε τοπικό ελάχιστο αλλάζουν τον αλγόριθμο που χρησιμοποιούν.
 - ❖ αλγόριθμος μεταβλητής γειτονιάς αναζήτησης.
 - ❖ αλγόριθμος επέκτασης της γειτονιάς αναζήτησης.
4. Αλγόριθμοι που αλλάζουν την αντικειμενική συνάρτηση ή τους περιορισμούς του προβλήματος.

- ❖ αλγόριθμος καθοδηγούμενης τοπικής αναζήτησης.

3.2.1 Αλγόριθμος μεταβλητής γειτονιάς αναζήτησης (Variable Neighborhood Search)

Ο Αλγόριθμος μεταβλητής γειτονιάς αναζήτησης προτάθηκε από τους Hansen και Mladenovic. Σκοπός του είναι η αναζήτηση ενός συνόλου γειτονιών ώστε να βρεθεί μια βελτιωμένη λύση. Η αναζήτηση μπορεί να γίνει με τυχαίο τρόπο αλλά και με πιο συστηματικό ώστε να ξεφύγει ο αλγόριθμος από ένα τοπικό ελάχιστο. Ο αλγόριθμος αυτός επωφελείται από το γεγονός ότι οι διαφορετικές τοπικές αναζητήσεις μπορούν να οδηγήσουν σε διαφορετικά τοπικά ελάχιστα.

Ο αλγόριθμος μεταβλητής γειτονιάς αναζήτησης είναι ένας στοχαστικός αλγόριθμος, όπου ένα σύνολο γειτονιών επιλέγονται (N_l , όπου $l = 1, \dots, lmax$). Στη πορεία κάθε επανάληψη του αλγορίθμου ακολουθεί τα παρακάτω βήματα:

- ❖ Ανακίνηση.
- ❖ Τοπική αναζήτηση.
- ❖ Κίνηση.

Πιο συγκεκριμένα σε κάθε επανάληψη μία αρχική λύση s' δημιουργείται από την τρέχουσα γειτονιά αναζήτησης. Επιπλέον υλοποιείται μια τοπική αναζήτηση εφαρμόζεται στη προηγούμενη λύση s' με σκοπό να παραχθεί η λύση s'' . Σε επόμενο βήμα ελέγχεται η αντικειμενική συνάρτηση και αν το κόστος αυτής της λύσης είναι βελτιωμένο σε σχέση με τη προηγούμενη τότε η καινούρια λύση s'' αντικαθιστά την αρχική μας λύση s' . Τέλος η ίδια διαδικασία αρχίζει από την αρχή με τη χρήση της γειτονιάς N_1 και τη καινούρια λύση s'' που βρέθηκε προηγουμένως. Αν δεν βρεθεί μία βελτιωμένη λύση, ο αλγόριθμος συνεχίζει στην επόμενη γειτονιά αναζήτησης.

Ο ψευδοκώδικας του αλγορίθμου της μεταβλητής γειτονιάς αναζήτησης παρουσιάζεται παρακάτω.

Αλγόριθμος Μεταβλητής Γειτονιάς Αναζήτησης

Αρχικοποίηση

Επέλεξε ένα σύνολο γειτονιών (N_l , όπου $l = 1, \dots, lmax$)

Επέλεξε μια αρχική λύση s_0

$l = 1$

repeat

 Δημιούργησε μία λύση s' στη γειτονιά του N_l

$s'' = LS(s)$, Εφάρμοσε μία διαδικασία τοπικής αναζήτησης στο s'

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

$s = s''$

$l = 1$


```

    else
         $l = l + 1$ 
    endif
until  $l \leq l_{max}$ 
Επέστρεψε τη βέλτιστη λύση.
(Ιωάννης Μαρινάκης Μ. Μ., 2019)

```

3.2.2 Άλλοι αλγόριθμοι

Περιορισμένη Αναζήτηση (Tabu Search)

Στον αλγόριθμο της περιορισμένης αναζήτησης γίνεται χρήση ενός αλγορίθμου τοπικής αναζήτησης για τη μετακίνηση μεταξύ των λύσεων, όμως και εδώ μπορεί να παγιδευτούμε σε τοπικό ελάχιστο. Για να ξεφύγει λοιπόν ο αλγόριθμος, χρησιμοποιεί τη μνήμη του από τις προηγούμενες κινήσεις του. Οι κινήσεις που έχουν ελεγχθεί αποθηκεύονται στη (tabu list) με σκοπό να μην ξανά επισκεφθούν. Σε κάποιες περιπτώσεις επιτρέπονται κινήσεις μέσα από αυτή τη λίστα αν όμως επιφέρουν μια βελτιωμένη λύση από την προηγούμενη, γνωστό ως κριτήριο απενεργοποίησης των περιορισμών (aspiration criterion).

Αν υπάρχουν κάποιες μεταβλητές που βρίσκονται μέσα στη λύση συχνά, οι μεταβλητές αυτές εξακολουθούν να παραμένουν μέσα στη λύση καθώς η βέλτιστη λύση θα βρίσκεται κάπου εκεί κοντά στη γειτονιά τους. Αυτή η διαδικασία ονομάζεται στρατηγική εντατικοποίησης της λύσης (intensification strategies) και αυτή είναι η μνήμη μεσαίας περιόδου (medium strategies).

Τέλος έχουμε και τη στρατηγική της διάχυσης (diversification), αυτή η στρατηγική έχει ως στόχο την αναζήτηση των ανεξερευνήτων περιοχών. (Ιωάννης Μαρινάκης Μ. Μ., 2019)

Διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης (GRASP)

Η διαδικασία αυτή είναι μια επαναληπτική διαδικασία και τελειώνει όταν κάποιο κριτήριο τερματισμού ικανοποιηθεί. Επίσης, έχει ως στόχο την εύρεση προσεγγιστικών λύσεων στα προβλήματα συνδυαστικής βελτιστοποίησης και σε κάθε επανάληψη δίνεται και μια εφικτή λύση. Η τελική μας λύση είναι η βέλτιστη εφικτή λύση που έχει βρεθεί μέχρι τη στιγμή του τερματισμού του αλγορίθμου.

Κάθε επανάληψη χωρίζεται σε δύο φάσεις:

- ❖ **Φάση κατασκευής μιας αρχικής λύσης (construction phase)**

Μια τυχαιοποιημένη συνάρτηση απληστίας κατασκευάζει μια αρχική λύση.

- ❖ **Διαδικασία τοπικής αναζήτησης**

Με την εφαρμογή τη τοπικής αναζήτησης η λύση βελτιώνεται.

Κεφάλαιο 4: Περιγραφή και Επίλυση του προβλήματος VRPSPD

4.1 Περιγραφή του προβλήματος

Όπως σε κάθε πρόβλημα υπάρχουν δεδομένα, ζητούμενα και περιορισμοί το ίδιο συμβαίνει και στα προβλήματα δρομολόγησης οχημάτων. Στο συγκεκριμένο πρόβλημα λοιπόν έχουμε:

Δεδομένα:

- ❖ Πλήθος των πελατών.
- ❖ Συντεταγμένες των πελατών.
- ❖ Ζήτηση των πελατών.
- ❖ Παραλαβή των πελατών .

Περιορισμοί:

- ❖ Χωρητικότητα των φορτηγών.
- ❖ Ένας πελάτης επισκέπτεται μόνο μια φορά.
- ❖ Ένα φορτηγό ξεκινάει από την αποθήκη και καταλήγει στην αποθήκη.

Δεν υπάρχει περιορισμός για το στόλο των φορτηγών.

Ζητούμενα:

- ❖ Εξυπηρέτηση όλων των πελατών
- ❖ Ελαχιστοποίηση του συνολικού κόστους των διαδρομών

Στη συνέχεια ορίζουμε με τον αριθμό 1 την αποθήκη και με τους υπόλοιπους θετικούς αριθμούς τους πελάτες μας. Επίσης το κόστος της διαδρομής που διανύουμε για να πάμε από το κόμβο i στο κόμβο j , θεωρούμε ότι είναι ίσο με την ευκλείδεια απόσταση αυτών των δύο κόμβων.

Παράδειγμα:

Έστω το διάνυσμα

1	2	3	1
---	---	---	---

Σημαίνει ότι από την αποθήκη πήγαμε στο πελάτη 2, από το πελάτη 2 πήγαμε στο πελάτη 3 και από το πελάτη 3 γυρίσαμε στην αποθήκη. $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$

Το κόστος ώστε να πάμε από την αποθήκη στο πελάτη 2 είναι ίσο με:

$$COST(1,2) = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}$$

Από την επεξεργασία, τον υπολογισμό όλων των παραπάνω και μέσω του αλγορίθμου Clarke & Wright κατασκευάζεται μια πολύ καλή αρχική λύση, που στη συνέχεια βελτιώνεται περαιτέρω με την υλοποίηση του αλγορίθμου της Μεταβλητής Γειτονιάς Αναζήτησης (VNS).

4.2 Παραγωγή αρχικών λύσεων μέσω του αλγορίθμου των εξοικονομήσεων των Clarke & Wright

Αρχικά αφού είναι γνωστές οι συντεταγμένες της αποθήκης και των πελατών υπολογίζονται εύκολα οι αποστάσεις μεταξύ τους και αποθηκεύονται στο πίνακα distance. Επίσης δημιουργούνται οι αρχικές διαδρομές που αποτελούνται από ένα μόνο κόμβο (πελάτη) και υπολογίζονται τα κόστη και οι συνολικές διανομές και παραλαβές κάθε διαδρομής στους πίνακες cost, delivery και pickup αντίστοιχα.

Στη συνέχεια ελέγχεται η εφικτότητα των ενώσεων των διαδρομών και για τις ενώσεις που είναι εφικτές υπολογίζονται οι εξοικονομήσεις.

Για να είναι μια λύση εφικτή θα πρέπει η χωρητικότητα του οχήματος να μην υπερβαίνει τη μέγιστη χωρητικότητα Q_{\max} του εκάστοτε οχήματος. Γι' αυτό το λόγο υπολογίζουμε 3 αθροίσματα:

1. Της **συνολική ζήτησης** των πελατών που έχουν ανατεθεί σε ένα όχημα (ds), όπου το άθροισμα αυτό δεν θα πρέπει να ξεπερνά τη μέγιστη χωρητικότητα του οχήματος Q_{\max} .
2. Της **συνολική παραλαβής** των πελατών που έχουν ανατεθεί σε ένα όχημα (ps), όπου το άθροισμα αυτό δεν θα πρέπει να ξεπερνά τη μέγιστη χωρητικότητα του οχήματος Q_{\max} .
3. Ένα που θα πρέπει σε κάθε κίνηση το άθροισμα των **εναπομενουσών προϊόντων** που θα πρέπει να διανεμηθούν στους πελάτες και το άθροισμα των **συλλεχθέντων προϊόντων** (qs) από τους πελάτες να μην ξεπερνάει τη συνολική χωρητικότητα του οχήματος Q_{\max} .

Για να γίνει ο υπολογισμός της εξοικονόμησης 2 διαδρομών θα χρειαστούν τα αρχικά κόστη των δύο διαδρομών και ο υπολογισμός του κόστους της συγχωνευμένης διαδρομής που προκύπτει από την ένωση αυτών των δύο διαδρομών.

Για παράδειγμα, έστω οι διαδρομές:

1: $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$ με κόστος $cost_1 = 10$

2: $1 \rightarrow 4 \rightarrow 5 \rightarrow 1$ με κόστος $cost_2 = 10$

Και η ένωση τους

12: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 1$ με κόστος $cost_{12} = 15$

Τότε η εξοικονόμηση από την ένωση των διαδρομών 1 και 2 υπολογίζεται από το τύπο $eks_{12} = cost_1 + cost_2 - cost_{12} = 5$

Δηλαδή, συμφέρει να γίνει η ένωση αυτών των δύο διαδρομών.

Επιπλέον, μετά τον υπολογισμό των εξοικονομήσεων για κάθε ζεύγος διαδρομών στο πίνακα eks, επιλέγεται η μεγαλύτερη εξοικονόμηση 2 διαδρομών και γίνεται η ένωση τους, οπότε οι 2 προηγούμενες διαδρομές αντικαθίστανται από μια νέα διαδρομή, την ένωση τους. Η διαδικασία αυτή συνεχίζεται μέχρις ότου να μην υπάρχει κάποια άλλη εφικτή ένωση με θετική εξοικονόμηση.

4.3 Βελτιστοποίηση της αρχικής λύσης μέσω του αλγορίθμου Μεταβλητής Γειτονίας Αναζήτησης (VNS)

Στο σημείο αυτό έχουμε μια πολύ καλή αρχική λύση με τον αλγόριθμο των εξοικονομήσεων, στη συνέχεια λοιπόν θα εφαρμόσουμε τον αλγόριθμο μεταβλητής γειτονιάς αναζήτησης για την βελτιστοποίηση και την εύρεση καλύτερης λύσης.

Εφαρμόζουμε τον αλγόριθμο VNS καθώς στόχος μας είναι να ξεφύγουμε από τα τοπικά ελάχιστα που έχουν δημιουργηθεί από την εφαρμογή του αλγορίθμου των εξοικονομήσεων. Αυτό είναι εφικτό καθώς ο αλγόριθμος VNS αναζητεί ένα πλήθος γειτονιών για την εύρεση καλύτερης λύσης μέσω των διάφορων αλγορίθμων τοπικής αναζήτησης.

Στο συγκεκριμένο πρόβλημα για τη σύνθεση του VNS, αρχικά δοκιμάσαμε τους αλγορίθμους 1-1 exchange και 1-0 relocate. Όμως τα αποτελέσματα δεν ήταν τα επιθυμητά καθώς η βελτίωση της λύσης ήταν ασήμαντη. Το πρόβλημα λοιπόν ήταν ότι αρχικά εφαρμοζόταν ο 1-1 exchange μας παρήγαγε μια καλύτερη λύση και στη συνέχεια έτρεχε ο 1-0 relocate, αλλά η εφαρμογή του 1-1 exchange μόνο του μας είχε οδηγήσει σε λίγο καλύτερα τοπικά ελάχιστα και συνεπώς 1-0 relocate δεν είχε πολλά περιθώρια βελτίωσης της λύσης.

Στη συνέχεια, δοκιμάσαμε να συγχωνέψουμε τους δύο αλγορίθμους σε έναν, όπου στην αρχή εφαρμόζεται ο 1-1 exchange και έπειτα ο 1-0 relocate. Η διαφορά όμως από την προηγούμενη εφαρμογή τους είναι ότι τώρα αφότου ο 1-1 exchange μας επιφέρει καλύτερη λύση, κρατάμε αυτή σαν την αρχική μας και εφαρμόζουμε τον 1-0 relocate στην νέα μας αρχική λύση. Αυτό είχε ως αποτέλεσμα τη σημαντική βελτίωση της λύσης αλλά και πάλι υπήρχαν περιθώρια επιπλέον βελτίωσης.

Επιπλέον, για τη βελτιστοποίηση της προηγούμενης λύσης δοκιμάσαμε την προσθήκη και την εφαρμογή του αλγορίθμου τοπικής αναζήτησης 2-opt στο πρόβλημα μας. Έτσι ο VNS αποτελείται από τους αλγορίθμους τοπικής αναζήτησης 1-1 exchange, 1-0 relocate και 2-opt όπου μετά από την εφαρμογή μιας τοπικής αναζήτησης, αν έχουμε καλύτερη λύση από την αρχική λύση, ανανεώνουμε την αρχική λύση με αυτή της καλύτερης λύσης και συνεχίζουμε στην εφαρμογή των επόμενων αλγορίθμων. Επίσης, παρατηρήσαμε ότι και η σειρά με την οποία εφαρμόζονται οι αλγόριθμοι στο πρόβλημα μας επηρεάζει αισθητά τη λύση. Μετά από

δοκιμές για την εύρεση της καλύτερης δυνατής σειράς εφαρμογής των αλγορίθμων, καταλήξαμε ότι τα βέλτιστα αποτελέσματα κάποιες φορές έρχονται από την εφαρμογή της σειράς 1: (1-0 relocate) - (1-1 exchange) - (2-opt) και κάποιες άλλες από τη σειρά 2: (1-0 relocate) - (2-opt) - (1-1 exchange).

Συνεπώς, για την εύρεση της βελτιωμένης λύσης δημιουργήθηκαν 2 VNS από την σειρά 1 και τη σειρά 2, Όπου η τελική μας λύση είναι η βέλτιστη λύση που προκύπτει από την εφαρμογή αυτών των δύο VNS.

Τέλος, συγκρίναμε τα αποτελέσματα μας με τις βέλτιστες λύσεις που προκύπτουν από τη μέθοδο APA.

4.3.1 Ο αλγόριθμος τοπικής αναζήτησης 1-0 relocate

Με αυτή τη μέθοδο μας δίνεται η ευκαιρία να πάρουμε έναν κόμβο από τη θέση του και να τον μεταφέρουμε ανάμεσα σε δύο άλλους. Η μέθοδος 1-0 Relocate βασίζεται στην απλή ιδέα της διαγραφής ενός πελάτη από μια διαδρομή και την επανατοποθέτησή του σε μια άλλη διαδρομή με λιγότερο κόστος. Παρακάτω παρουσιάζεται η διαδικασία:

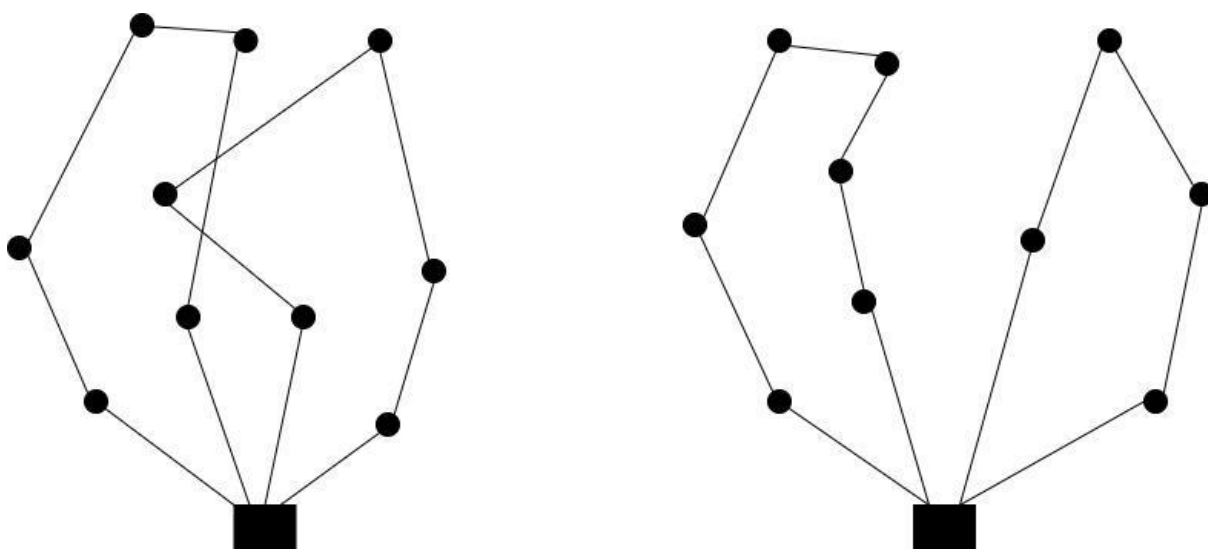
Βήμα 1: Υπολογίζεται η απόσταση που διανύει το όχημα που εξυπηρετεί το κόμβο που θέλουμε να μεταφέρουμε σε άλλη διαδρομή καθώς και την απόσταση της διαδρομής που πρόκειται να αποσταλεί αυτός ο κόμβος.

Βήμα 2: Υπολογίζονται οι νέες αποστάσεις αφού τοποθετήθηκε ο κόμβος στο νέο μονοπάτι και τον αφαιρέσαμε από την παλιά διαδρομή.

Βήμα 3: Γίνεται έλεγχός στις νέες αποστάσεις όπου αν είναι μικρότερες από τις προηγούμενες, τοποθετούμε τον κόμβο στην διαδρομή του νέου οχήματος.

Βήμα 4: Η διαδικασία επαναλαμβάνεται μέχρι να μην υπάρχει κάποια επιπλέον αλλαγή.

Βήμα 5: Τερματίζεται ο αλγόριθμος (1-0 Relocate) και επιστρέφονται οι νέες διαδρομές.



Εικόνα 4.1: Αλγόριθμος 1-0 Relocate (Ιωάννης Μαρινάκης Μ. Μ., 2019)

Για παράδειγμα, έστω οι διαδρομές:

1: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 1$

2: $1 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 1$

Για να γίνει μια κίνηση 1-0 relocate θα πρέπει ο κόμβος που θα φύγει από μία διαδρομή και η επανένωση του σε μια άλλη να έχει ως αποτέλεσμα την μείωση του συνολικού κόστους και των δύο διαδρομών.

Εξετάζουμε τυχαία λουπόν αν η μετακίνηση του κόμβου 3 από την πρώτη διαδρομή στη δεύτερη και συγκεκριμένα ανάμεσα στο κόμβο 6 και 7 θα επιφέρει μικρότερο συνολικό κόστος.

Αρχικά για τη διαδρομή 1 υπολογίζω τα κόστη σύνδεσης του κόμβου που θέλω να διώξω με το προηγούμενο κόμβο, το κόστος από το κόμβο 2 στο 3 και το κόστος σύνδεσης με τον επόμενο κόμβο το κόστος από το κόμβο 3 στο 4. Δηλαδή:

$$Costarx_1 = cost_{23} + cost_{34}$$

Για τη διαδρομή 2 υπολογίζω το κόστος της ένωσης των δύο κόμβων όπου η προσθήκη του νέου κόμβου θα γίνει ανάμεσα τους και συγκεκριμένα το κόστος από το κόμβο 6 στο 7. Δηλαδή:

$$Costarx_2 = cost_{67}$$

Άρα το άθροισμα αυτών των δύο μας δίνει το αρχικό κόστος.

$$Cost_{arx} = Cost_{arx_1} + Cost_{arx_2}$$

Στη συνέχεια αν είναι εφικτή η αλλαγή, οι δύο διαδρομές θα γίνουν:

1: $1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 1$

2: $1 \rightarrow 6 \rightarrow 3 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 1$

Οπότε για τη διαδρομή 1 το νέο κόστος υπολογίζεται από την ένωση του προηγούμενου και του επόμενου κόμβου από αυτόν που μετακινήσαμε, συγκεκριμένα το κόστος από το κόμβο 2 στο κόμβο 4. Δηλαδή:

$$Cost_{tel_1} = cost_{24}$$

Για τη διαδρομή 2 το νέο κόστος υπολογίζεται από την ένωση των κόμβων που υπήρχαν πριν με τον κόμβο που μεταφέρθηκε, συγκεκριμένα το κόστος από το κόμβο 6 στο κόμβο 3 και το κόστος από το κόμβο 3 στο κόμβο 7. Δηλαδή:

$$Cost_{tel_2} = cost_{63} + cost_{37}$$

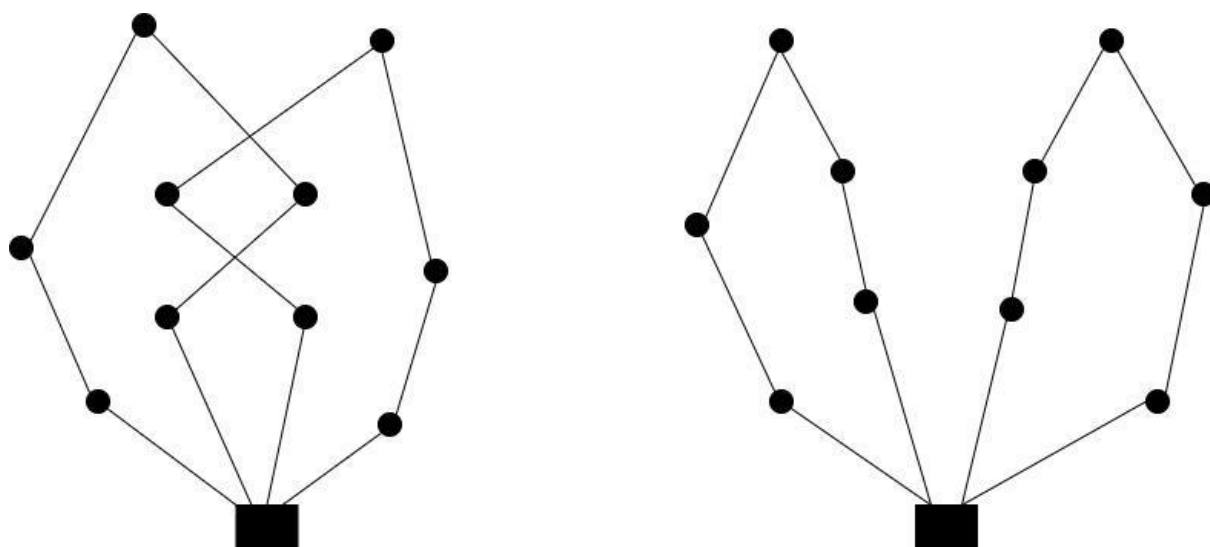
Άρα το άθροισμα αυτών των δύο μας δίνει το τελικό κόστος.

$$Cost_{tel} = Cost_{tel_1} + Cost_{tel_2}$$

Σε επόμενο στάδιο ελέγχω αυτά τα δύο κόστη και αν $Cost_{tel} < Cost_{arx}$ τότε η αλλαγή αυτή αφού επιφέρει μικρότερο συνολικό κόστος θα γίνει. Αν όμως $Cost_{tel} > Cost_{arx}$ τότε η αλλαγή δεν θα γίνει και θα συνεχιστεί ο έλεγχος για επόμενα ζευγάρια κόμβων, η διαδικασία αυτή γίνεται μέχρις ότου να μην υπάρχουν άλλες μετακινήσεις που να μειώνουν το συνολικό κόστος των διαδρομών.

4.3.2 Ο αλγόριθμος τοπικής αναζήτησης 1-1 exchange

Με τη μέθοδο 1-1 exchange μπορούμε να ανταλλάξουμε ένα κόμβο από τη θέση του με ένα άλλο κόμβο είτε της ίδιας διαδρομής είτε κάποιας άλλης με σκοπό τη μείωση του συνολικού κόστους.



Εικόνα 4.2: Αλγόριθμος 1-1 exchange (Ιωάννης Μαρινάκης Μ. Μ., 2019)

Για παράδειγμα, έστω οι διαδρομές:

1: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 1$

2: $1 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 1$

Για να γίνει μια κίνηση 1-1 exchange μεταξύ 2 διαδρομών θα πρέπει οι 2 κόμβοι που ανταλλάσσονται να έχουν ως αποτέλεσμα την μείωση του συνολικού κόστους των δύο διαδρομών.

Εξετάζουμε τυχαία λοιπόν αν η ανταλλαγή του κόμβου 3 από την πρώτη διαδρομή με το κόμβο 8 της δεύτερης διαδρομής θα επιφέρει μικρότερο συνολικό κόστος.

Αρχικά για τη διαδρομή 1 υπολογίζω τα κόστη σύνδεσης του κόμβου που θέλω να ανταλλάξω με το προηγούμενο κόμβο, το κόστος από το κόμβο 2 στο 3 και το κόστος σύνδεσης με τον επόμενο κόμβο, το κόστος από το κόμβο 3 στο 4. Δηλαδή:

$$Costarx_1 = cost_{23} + cost_{34}$$

Για τη διαδρομή 2 υπολογίζω πάλι τα κόστη σύνδεσης του κόμβου που θέλω να ανταλλάξω με το προηγούμενο κόμβο, το κόστος από το κόμβο 7 στο 8 και το κόστος σύνδεσης με τον επόμενο κόμβο, το κόστος από το κόμβο 8 στο 9. Δηλαδή:

$$Costarx_2 = cost_{78} + cost_{89}$$

Άρα το άθροισμα αυτών των δύο μας δίνει το αρχικό κόστος.

$$Cost_{arx} = Costarx_1 + Costarx_2$$

Στη συνέχεια αν είναι εφικτή η αλλαγή, οι δύο διαδρομές θα γίνουν:

1: $1 \rightarrow 2 \rightarrow \mathbf{8} \rightarrow 4 \rightarrow 5 \rightarrow 1$

2: $1 \rightarrow 6 \rightarrow 7 \rightarrow \mathbf{3} \rightarrow 9 \rightarrow 1$

Οπότε για τη διαδρομή 1 το νέο κόστος υπολογίζεται όπως πριν απλά αντί για τον κόμβο 3 στη θέση του έχω το κόμβο 8. Δηλαδή:

$$Cost_{tel_1} = cost_{28} + cost_{84}$$

Για τη διαδρομή 2 το νέο κόστος υπολογίζεται όπως πριν απλά αντί για τον κόμβο 8 στη θέση του έχω το κόμβο 3. Δηλαδή:

$$Cost_{tel_2} = cost_{73} + cost_{39}$$

Άρα το άθροισμα αυτών των δύο μας δίνει το τελικό κόστος.

$$Cost_{tel} = Cost_{tel_1} + Cost_{tel_2}$$

Σε επόμενο στάδιο ελέγχω αυτά τα δύο κόστη και αν $Cost_{tel} < Cost_{arx}$ τότε η αλλαγή αυτή αφού επιφέρει μικρότερο συνολικό κόστος θα γίνει. Αν όμως $Cost_{tel} > Cost_{arx}$ τότε η αλλαγή δεν θα γίνει και θα συνεχιστεί ο έλεγχος για επόμενα ζευγάρια κόμβων, η διαδικασία αυτή γίνεται μέχρις ότου να μην υπάρχουν άλλες ανταλλαγές που να μειώνουν το συνολικό κόστος των διαδρομών.

4.3.3 Ο αλγόριθμος τοπικής αναζήτησης 2-opt

Μια κίνηση 2-opt διαγράφει δύο άκρα, χωρίζοντας τη διαδρομή σε δύο μέρη, και στη συνέχεια επανασυνδέει τα μονοπάτια με άλλον δυνατό τρόπο, ώστε να έχουμε μείωση του κόστους. Αυτό ισοδυναμεί με αντιστροφή της σειράς των πελατών μεταξύ των δύο άκρων.

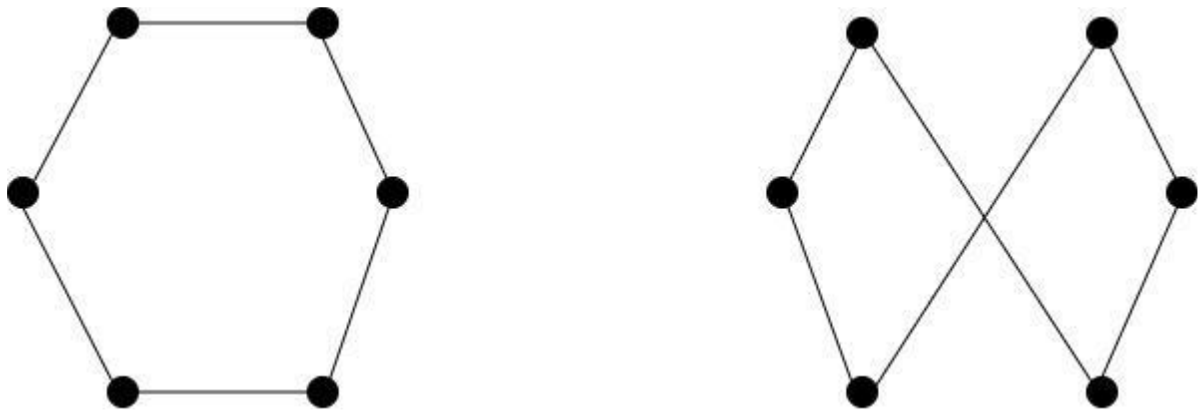
Η διαδικασία της μεθόδου 2-opt είναι η ακόλουθη:

Βήμα 1: Έστω D η τρέχουσα διαδρομή.

Βήμα 2: Για κάθε κόμβο $i=1, \dots, n$: Ελέγχονται όλες οι πιθανές αλλαγές που μπορεί να εφαρμοστούν από τον i και την επόμενη της μέσα στην διαδρομή. Αν με αυτό τον τρόπο

μπορεί να ελαχιστοποιηθεί το κόστος της διαδρομής, Επιλέγεται η καλύτερη 2-opt αλλαγή και γίνονται οι αλλαγές.

Βήμα 3: Αν δεν βρεθεί επιπλέον βελτίωση στις διαδρομές, τότε ο αλγόριθμος τερματίζεται.



Εικόνα 4.3: Αλγόριθμος 2-opt (Ιωάννης Μαρινάκης Μ. Μ., 2019)

Για παράδειγμα, έστω η διαδρομή:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 1$$

Για να γίνει μια κίνηση 2-opt θα πρέπει η διαγραφή 2 τόξων και η επανασύνδεση των κόμβων αυτών με άλλο τρόπο στην ίδια διαδρομή να μειώνει το τελικό κόστος της διαδρομής.

Εξετάζουμε τυχαία αν η διαγραφή των τόξων 1-2 και 5-6 θα επιφέρει μικρότερο κόστος. Οπότε η διαδρομή έχει χωριστεί σε 3 τμήματα:

$$1 \dots 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \dots 6 \rightarrow 1$$

Αρχικά υπολογίζω τα κόστη των δύο τόξων που διέγραψα του 1-2 και του 5-6. Δηλαδή

$$Cost_{arx} = cost_{12} + cost_{56}$$

Για τη δημιουργία της νέας διαδρομής κρατάω σταθερά το αρχικό τμήμα (1) και το τελικό τμήμα (6-1), αλλάζω τη σειρά στο μεσαίο τμήμα (2-3-4-5). Δηλαδή:

$$1 \dots 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \dots 6 \rightarrow 1$$

Και τα επανασυνδέω, άρα η νέα διαδρομή θα γίνει:

$$1 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 6 \rightarrow 1$$

Στη συνέχεια υπολογίζω τα κόστη των δύο τόξων που επανασυνδέθηκαν του 1-5 και του 2-6. Δηλαδή:

$$Cost_{tel} = cost_{15} + cost_{26}$$

Τέλος, ελέγχω αυτά τα δύο κόστη και αν $Cost_{tel} < Cost_{arx}$ τότε η αλλαγή αυτή αφού επιφέρει μικρότερο συνολικό κόστος θα γίνει. Αν όμως $Cost_{tel} > Cost_{arx}$ τότε η αλλαγή δεν θα γίνει και θα συνεχιστεί ο έλεγχος για επόμενα ζευγάρια κόμβων, η διαδικασία αυτή γίνεται μέχρις ότου να μην υπάρχουν άλλες διαγραφές τόξων που να μειώνουν το συνολικό κόστος της διαδρομής.

Κεφάλαιο 5: Αποτελέσματα

Στην συγκεκριμένη διπλωματική εργασία εξετάστηκαν 18 προβλήματα δρομολόγησης οχημάτων. Για τα προβλήματα αυτά είχαμε ως δεδομένα το πλήθος των κόμβων (**n**) τη χωρητικότητα του οχήματος (**Q**), τη συνολική ζήτηση των πελατών (**D**), τα προϊόντα προς παραλαβή (**P**) και τις συντεταγμένες των κόμβων. Ο αριθμός των κόμβων κυμαίνεται από 100 έως 400, η χωρητικότητα από 200 έως 1.000, η ζήτηση και τα προϊόντα προς παραλαβή είναι μεταξύ των 1.458 έως 12.470 προϊόντων. Στο παρακάτω πίνακα φαίνονται αναλυτικά τα δεδομένα για κάθε πρόβλημα.

Πρόβλημα	n	Q	D	P
<i>r101</i>	100	200	1458	2339
<i>r201</i>	100	1000	1458	2262
<i>c101</i>	100	200	1810	3070
<i>c201</i>	100	700	1810	2910
<i>rc101</i>	100	200	1724	1912
<i>rc201</i>	100	1000	1724	2076
<i>R1_2_1</i>	200	200	3513	4406
<i>R2_2_1</i>	200	1000	3513	4406
<i>C1_2_1</i>	200	200	3530	5370
<i>C2_2_1</i>	200	700	3770	6010
<i>RC1_2_1</i>	200	200	3558	4473
<i>RC2_2_1</i>	200	1000	3558	4299
<i>R1_4_1</i>	400	200	7109	10433
<i>R2_4_1</i>	400	1000	7109	9571
<i>C1_4_1</i>	400	200	7190	12470
<i>C2_4_1</i>	400	700	7560	10050
<i>RC1_4_1</i>	400	200	7127	10065
<i>RC2_4_1</i>	400	1000	7127	10100

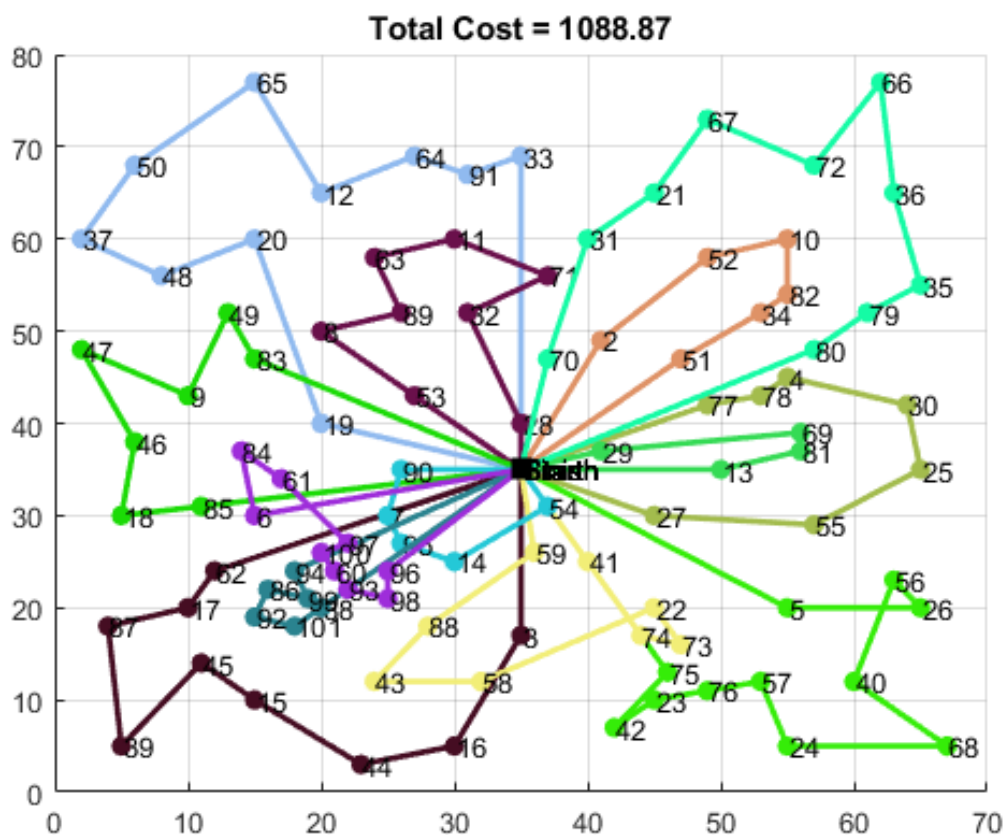
Πίνακας 5.1: Χαρακτηριστικά Προβλημάτων

5.1 Προβλήματα

5.1.1 Πρόβλημα r101

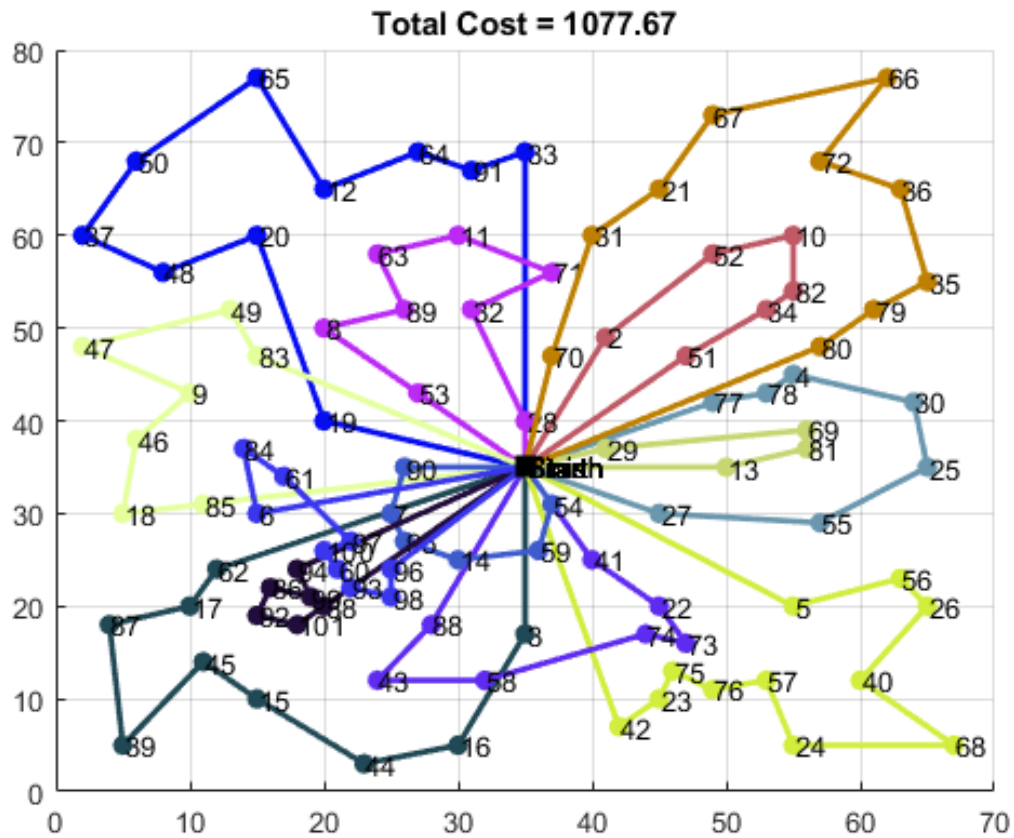
Για το πρόβλημα r101 έχουμε 100 κόμβους με μέγιστο φορτίο του οχήματος 200 τεμάχια και συνολική ζήτηση 1458 προϊόντα και 2339 προϊόντα προς παραλαβή. Οι αρχικές διαδρομές που προκύπτουν μέσω του αλγορίθμου εξοικονομήσεων Clarke & Wright απεικονίζονται στο παρακάτω γράφημα, έχουμε **13 διαδρομές** με συνολικό κόστος **1088,77 μονάδες**.

Παρακάτω παρουσιάζονται αναλυτικά οι αρχικές διαδρομές.



Γράφημα 1: Αρχική λύση r101

Οι βελτιωμένες διαδρομές και η τελική λύση του προβλήματος μέσω του αλγορίθμου της Μεταβλητής Γειτονιάς Αναζήτησης (VNS), απεικονίζεται στο παρακάτω γράφημα με συνολικό κόστος **1077,67 μονάδες**.

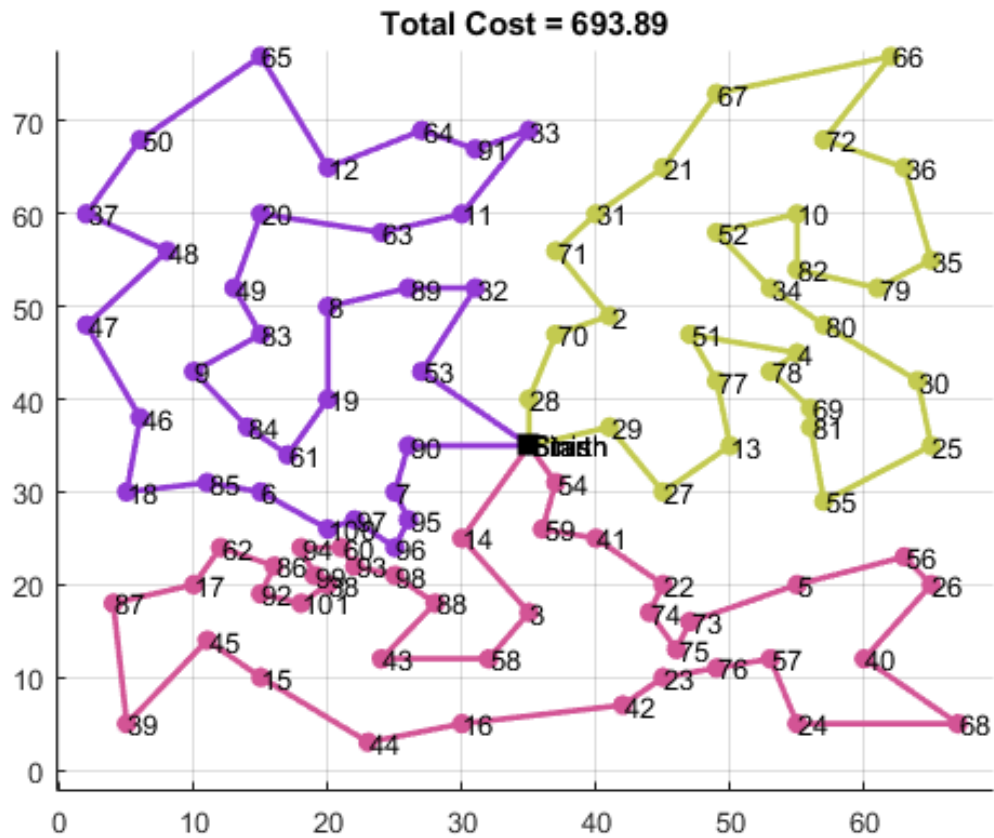


Γράφημα 2: Τελική λύση r101

Οι βασικότερες αλλαγές που παρατηρούνται εδώ είναι η 1-1 exchange ανταλλαγή του κόμβου 26 με τον 56, η 2-οpt αλλαγή των κόμβων 22-73-74 και η 2-0 relocate μετακίνηση του κόμβου 59 σε άλλη διαδρομή.

5.1.2 Πρόβλημα r201

Για το πρόβλημα r201 έχουμε 100 κόμβους με μέγιστο φορτίο του οχήματος 1000 τεμάχια και συνολική ζήτηση 1458 προϊόντα και 2262 προϊόντα προς παραλαβή. Οι αρχικές διαδρομές που προκύπτουν μέσω του αλγορίθμου εξοικονομήσεων Clarke & Wright απεικονίζονται στο παρακάτω γράφημα, έχουμε **3 διαδρομές** με συνολικό κόστος **706,01 μονάδες**.

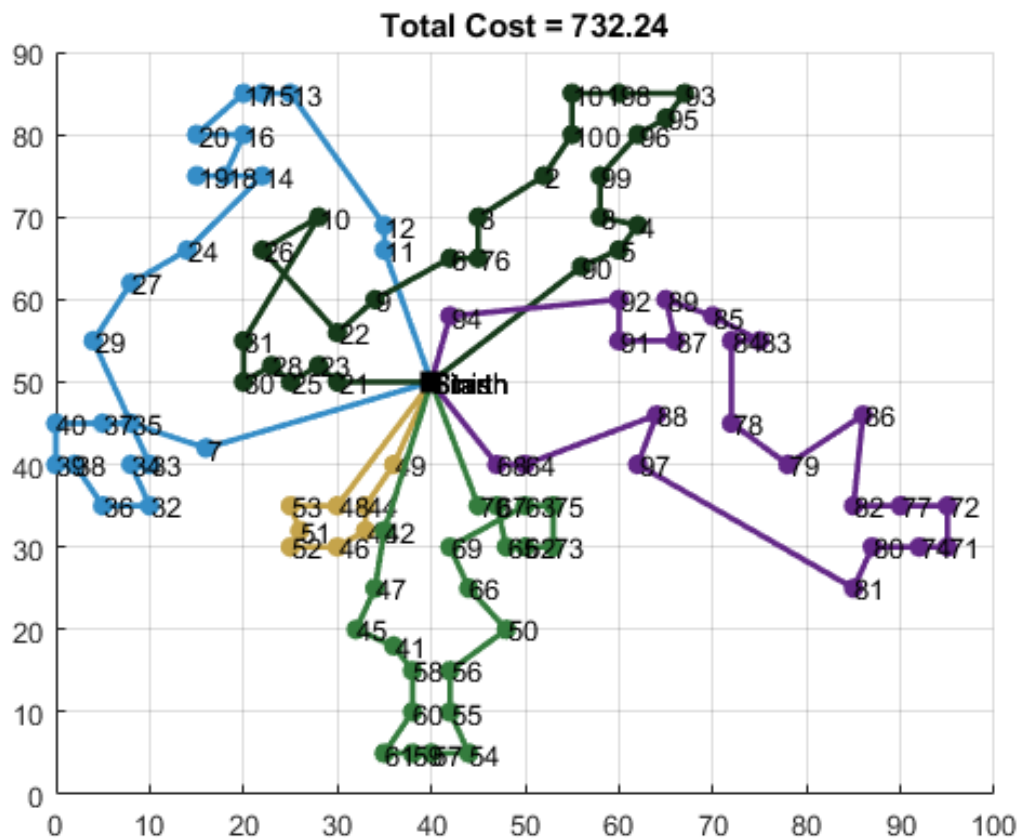


Γράφημα 4: Τελική λύση r201

Οι βασικότερες αλλαγές που παρατηρούνται εδώ είναι η 1-1 exchange ανταλλαγή του κόμβου 26 με τον 56, η 2-ορτ αλλαγή των κόμβων 22-74-75-73 και δεν παρατηρήθηκε κάποια 2-0 relocate μετακίνηση.

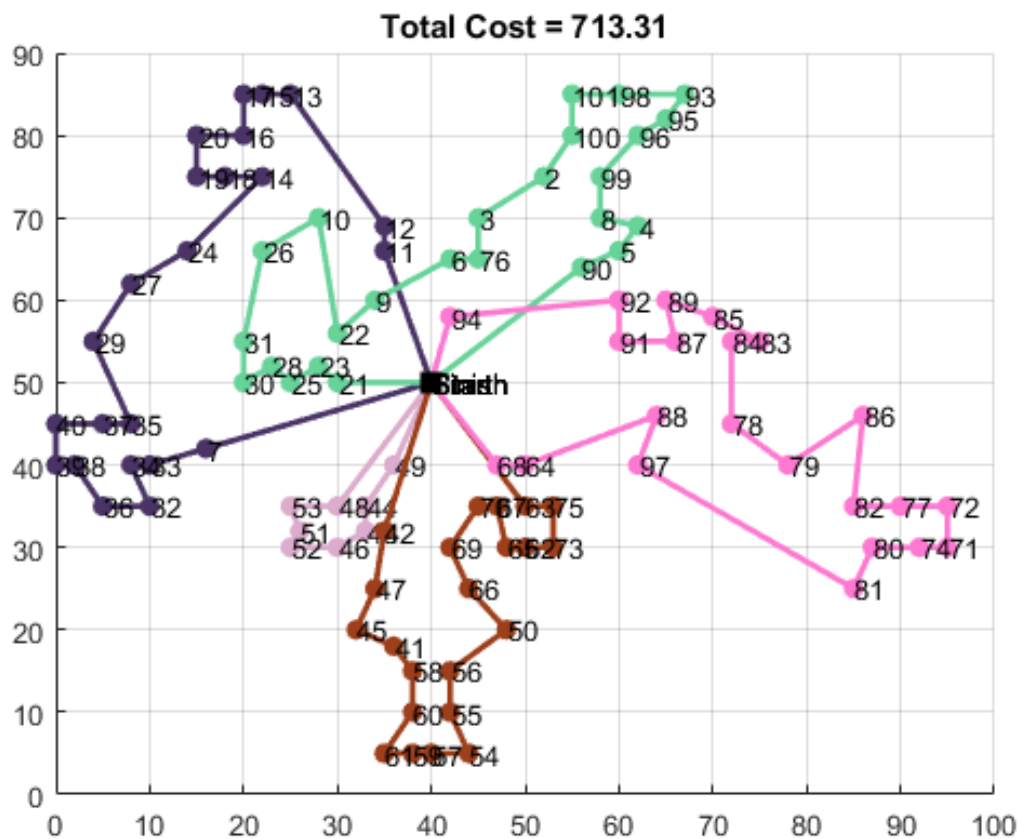
5.1.3 Πρόβλημα c201

Για το πρόβλημα c201 έχουμε 100 κόμβους με μέγιστο φορτίο του οχήματος 700 τεμάχια και συνολική ζήτηση 1810 προϊόντα και 2910 προϊόντα προς παραλαβή. οι αρχικές διαδρομές που προκύπτουν μέσω του αλγορίθμου εξοικονομήσεων Clarke & Wright απεικονίζονται στο παρακάτω γράφημα, έχουμε **5 διαδρομές** με συνολικό κόστος **732,24 μονάδες**.



Γράφημα 5: Αρχική λύση c201

Οι βελτιωμένες διαδρομές και η τελική λύση του προβλήματος μέσω του αλγορίθμου της Μεταβλητής Γειτονιάς Αναζήτησης (VNS), απεικονίζεται στο παρακάτω πίνακα με συνολικό κόστος **713,31 μονάδες**.

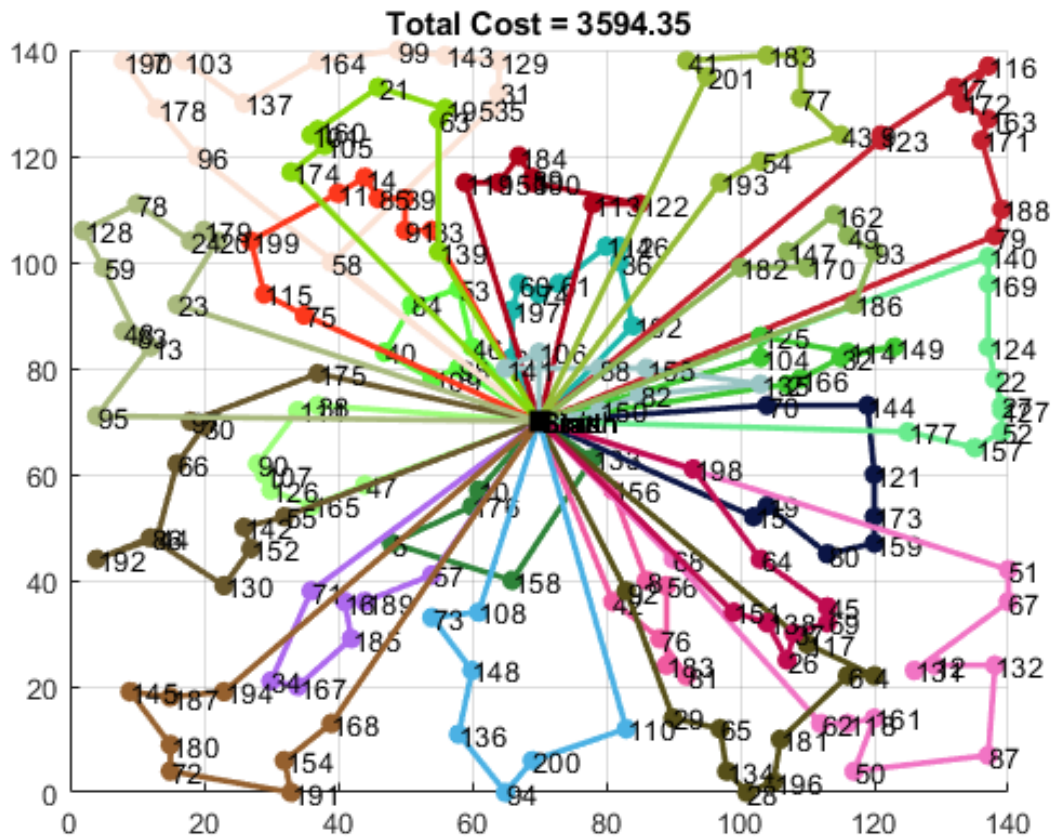


Γράφημα 6: Τελική λύση c201

Οι βασικότερες αλλαγές που παρατηρούνται εδώ είναι η 1-1 exchange ανταλλαγή του κόμβου 26 με τον 10, η 2-opt αλλαγή των κόμβων 35-37-40-39-38-36-32-34-33 και δεν παρατηρήθηκε κάποια 2-0 relocate μετακίνηση.

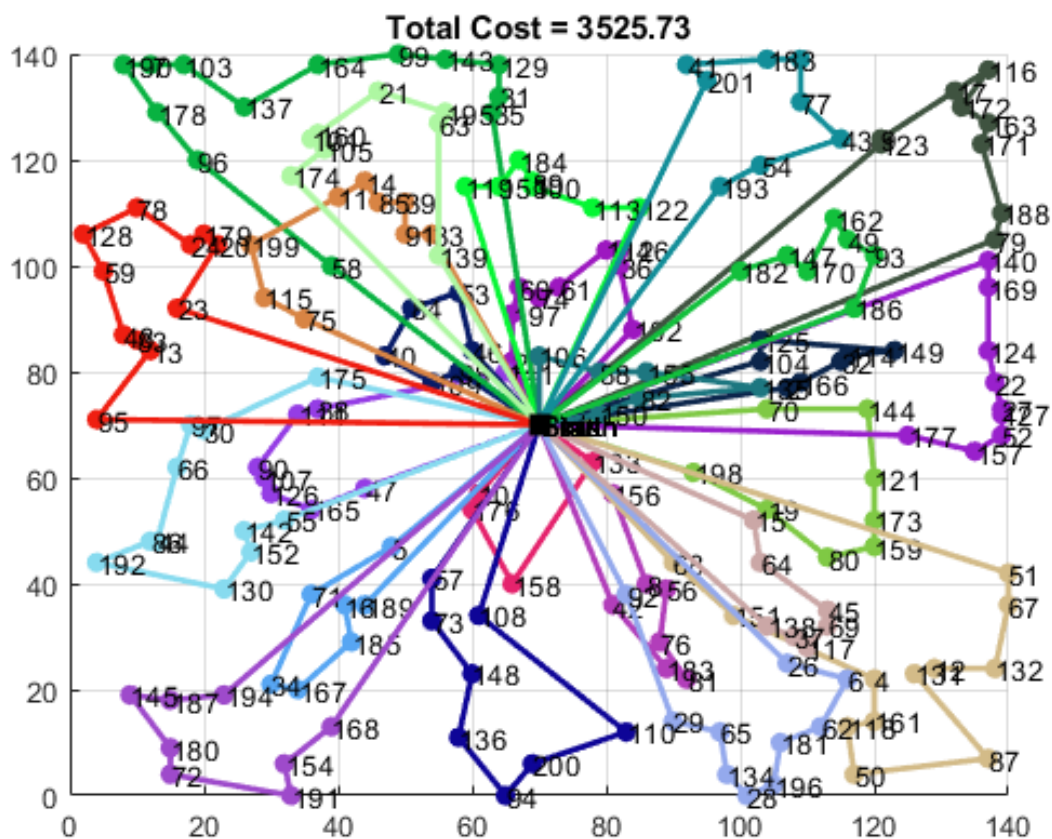
5.1.4 Πρόβλημα R1_2_1

Για το πρόβλημα R1_2_1 έχουμε 200 κόμβους με μέγιστο φορτίο του οχήματος 200 τεμάχια και συνολική ζήτηση 3513 προϊόντα και 4406 προϊόντα προς παραλαβή. οι αρχικές διαδρομές που προκύπτουν μέσω του αλγορίθμου εξοικονομήσεων Clarke & Wright απεικονίζονται στο παρακάτω γράφημα, έχουμε **24 διαδρομές** με συνολικό κόστος **3594,35 μονάδες**.



Γράφημα 7: Αρχική λύση R1_2_1

Οι βελτιωμένες διαδρομές και η τελική λύση του προβλήματος μέσω του αλγορίθμου της Μεταβλητής Γειτονιάς Αναζήτησης (VNS), απεικονίζεται στο παρακάτω πίνακα με συνολικό κόστος **3525,73 μονάδες**.

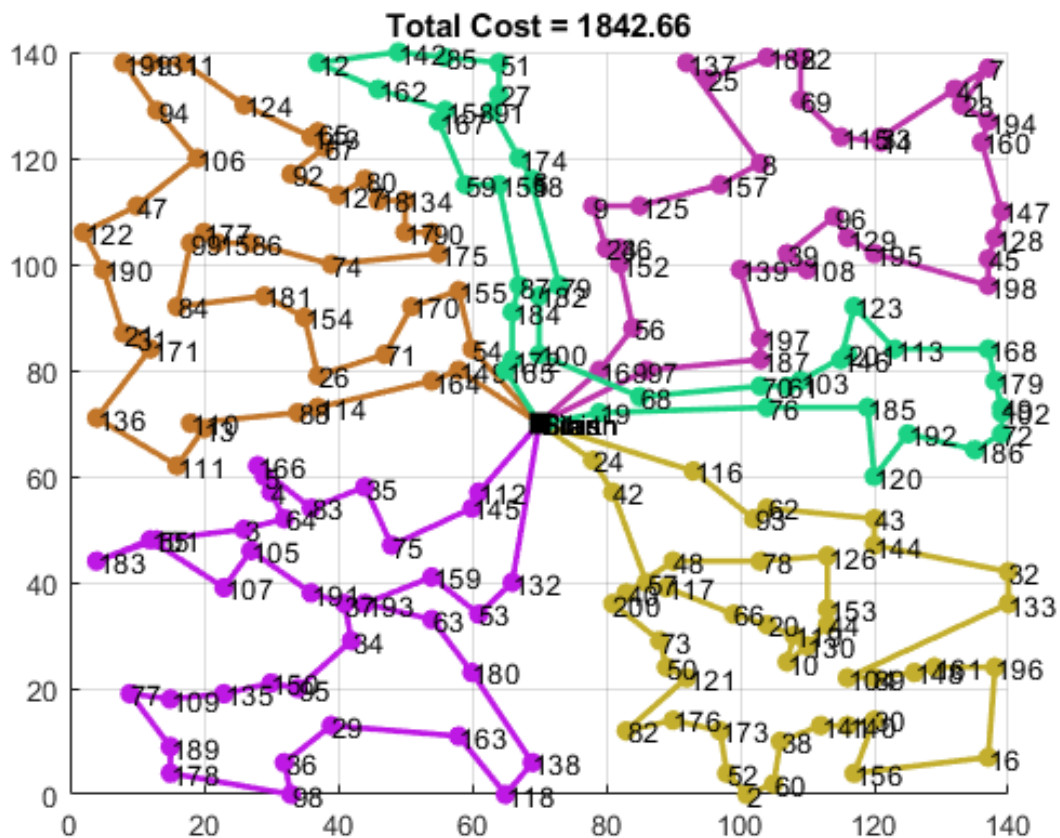


Γράφημα 8: Τελική λύση R1_2_1

Οι βασικότερες αλλαγές που παρατηρούνται εδώ είναι η 1-1 exchange ανταλλαγή του κόμβου 15 με τον 198, η 2-ort αλλαγή των κόμβων μετά τον 58 κόμβο όπου άλλαξε η σειρά εξυπηρέτησεων των επόμενων κόμβων και η 2-0 relocate μετακίνηση των κόμβων 5 και 57 .

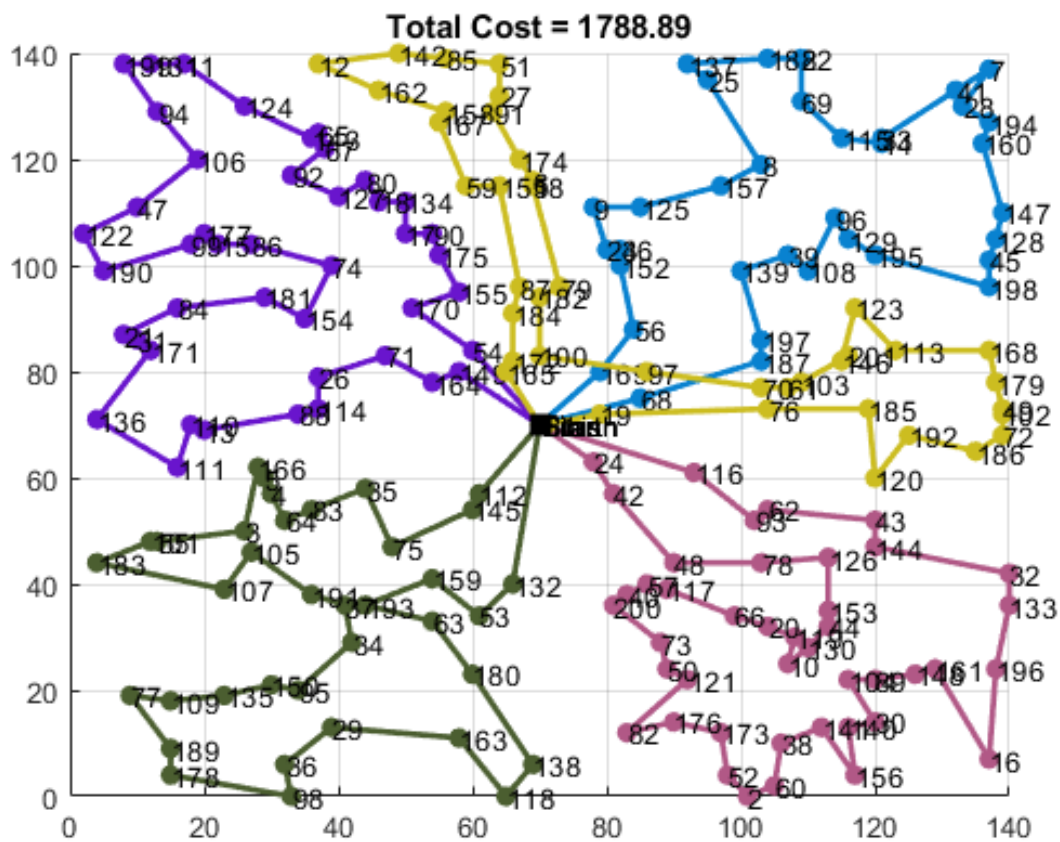
5.1.5 Πρόβλημα R2_2_1

Για το πρόβλημα R2_2_1 έχουμε 200 κόμβους με μέγιστο φορτίο του οχήματος 1000 τεμάχια και συνολική ζήτηση 3513 προϊόντα και 4406 προϊόντα προς παραλαβή. οι αρχικές διαδρομές που προκύπτουν μέσω του αλγορίθμου εξοικονομήσεων Clarke & Wright απεικονίζονται στο παρακάτω γράφημα, έχουμε **11 διαδρομές** με συνολικό κόστος **1842,66 μονάδες**.



Γράφημα 9: Αρχική λύση R2_2_1

Οι βελτιωμένες διαδρομές και η τελική λύση του προβλήματος μέσω του αλγορίθμου της Μεταβλητής Γειτονιάς Αναζήτησης (VNS), απεικονίζεται στο παρακάτω πίνακα με συνολικό κόστος **1788,89 μονάδες**.

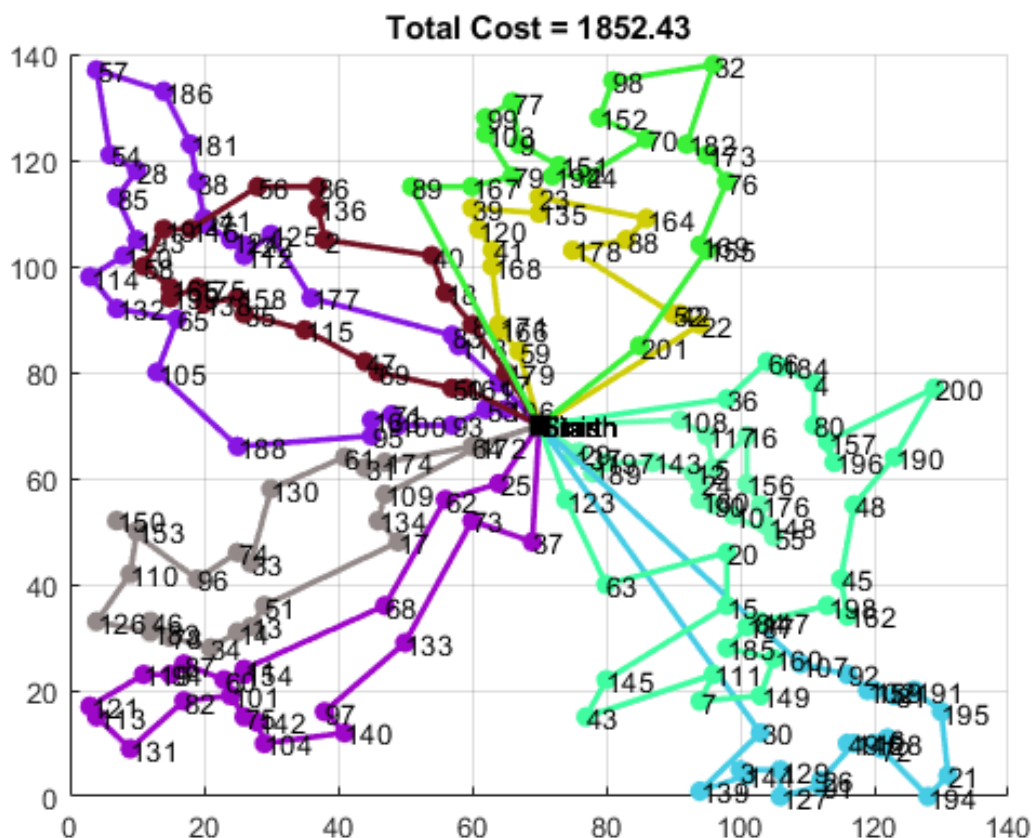


Γράφημα 10: Τελική λύση R2_2_1

Οι βασικότερες αλλαγές σε αυτό το πρόβλημα παρατηρούνται κυρίως στη μπλε και στη ροζ διαδρομή καθώς γίνονται αρκετές 2-ορτ και 1-1 exchange αλλαγές που βελτιώνουν το συνολικό κόστος.

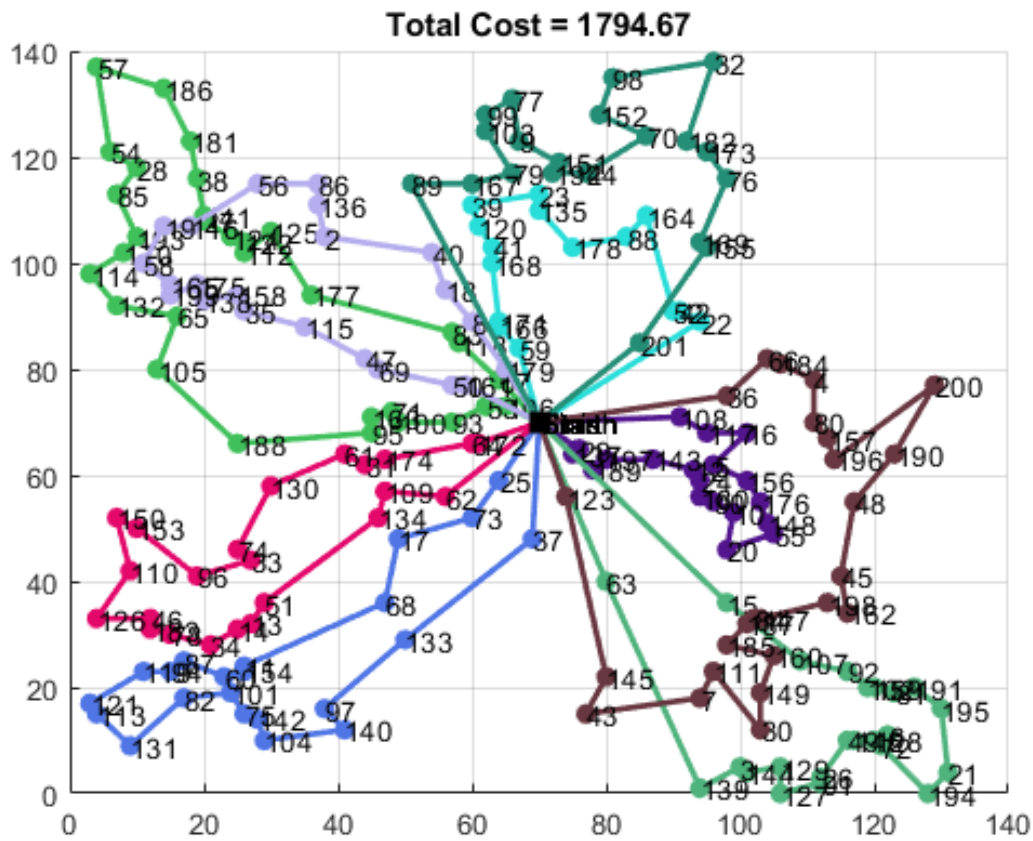
5.1.6 Πρόβλημα C2_2_1

Για το πρόβλημα C2_2_1 έχουμε 200 κόμβους με μέγιστο φορτίο του οχήματος 700 τεμάχια και συνολική ζήτηση 3770 προϊόντα και 6010 προϊόντα προς παραλαβή. οι αρχικές διαδρομές που προκύπτουν μέσω του αλγορίθμου εξοικονομήσεων Clarke & Wright απεικονίζονται στο παρακάτω γράφημα, έχουμε **9 διαδρομές** με συνολικό κόστος **1852,43 μονάδες**.



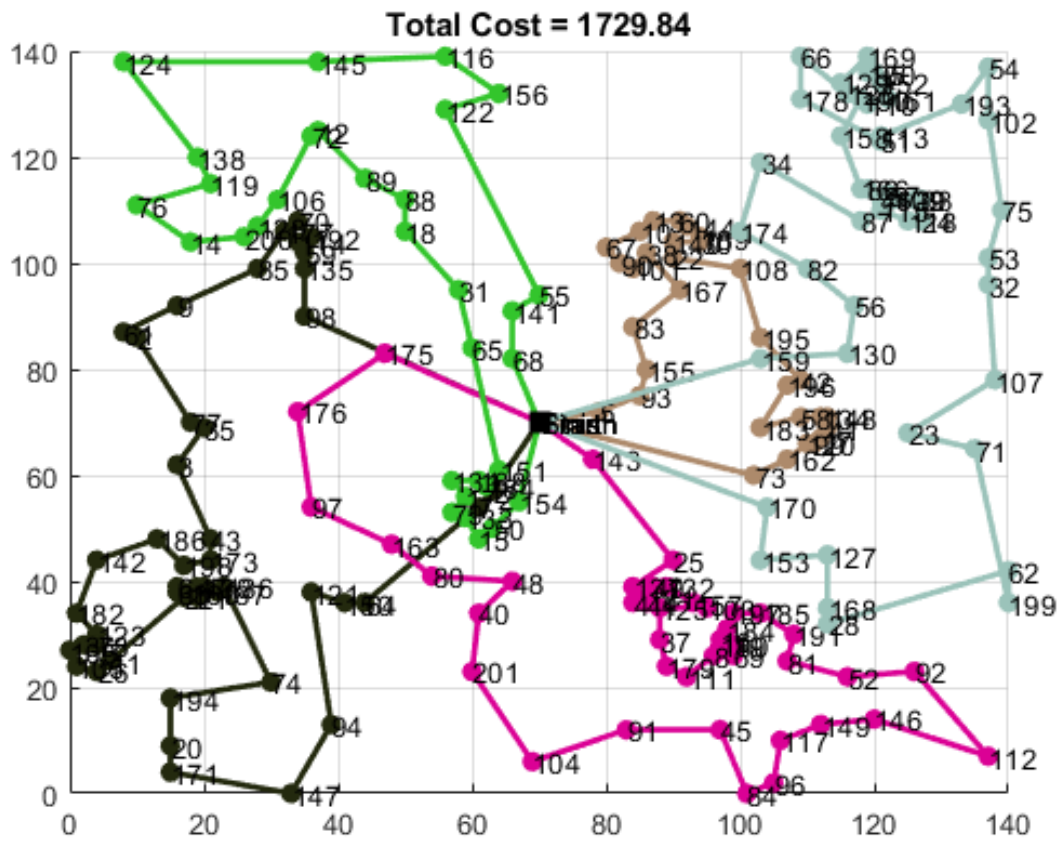
Γράφημα 11: Αρχική λύση C2_2_1

Οι βελτιωμένες διαδρομές και η τελική λύση του προβλήματος μέσω του αλγορίθμου της Μεταβλητής Γειτονιάς Αναζήτησης (VNS), απεικονίζεται στο παρακάτω πίνακα με συνολικό κόστος **1794,67 μονάδες**.



Γράφημα 12: Τελική λύση C2_2_1

Σε αυτό το πρόβλημα παρατηρούνται κυρίως 2-0 relocate μετακινήσεις όπως των κόμβων 63, 20, 15 μερικές 1-1 exchange ανταλλαγή όπως των κόμβων 17 και 62 και αρκετές μικρότερες 2-ort αλλαγές που βελτιώνουν το συνολικό κόστος.

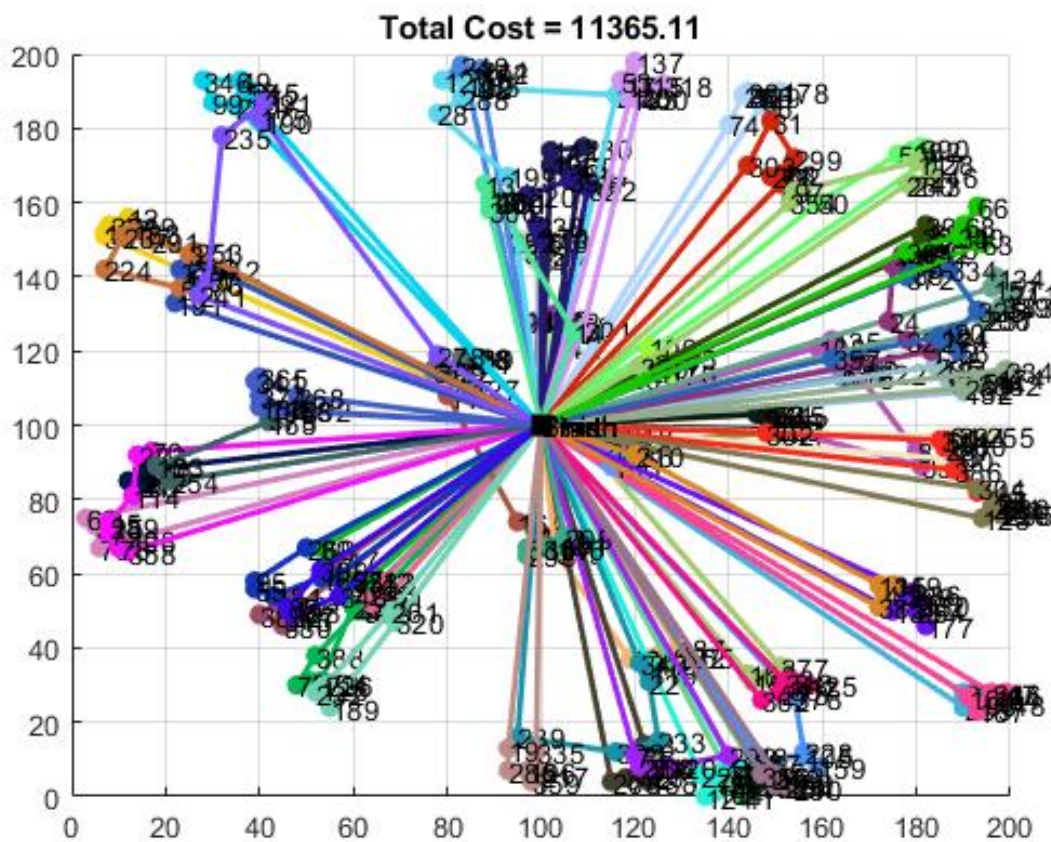


Γράφημα 14: Τελική λύση RC2_2_1

Οι βασικότερες αλλαγές σε αυτό το πρόβλημα παρατηρούνται κυρίως στη πράσινη, στη ροζ και στη μαύρη διαδρομή καθώς γίνονται αρκετές 2-opt και μερικές 1-1 exchange αλλαγές καθώς και η 2-0 relocate μετακίνηση του κόμβου 23 που βελτιώνουν το συνολικό κόστος εξυπηρέτησης.

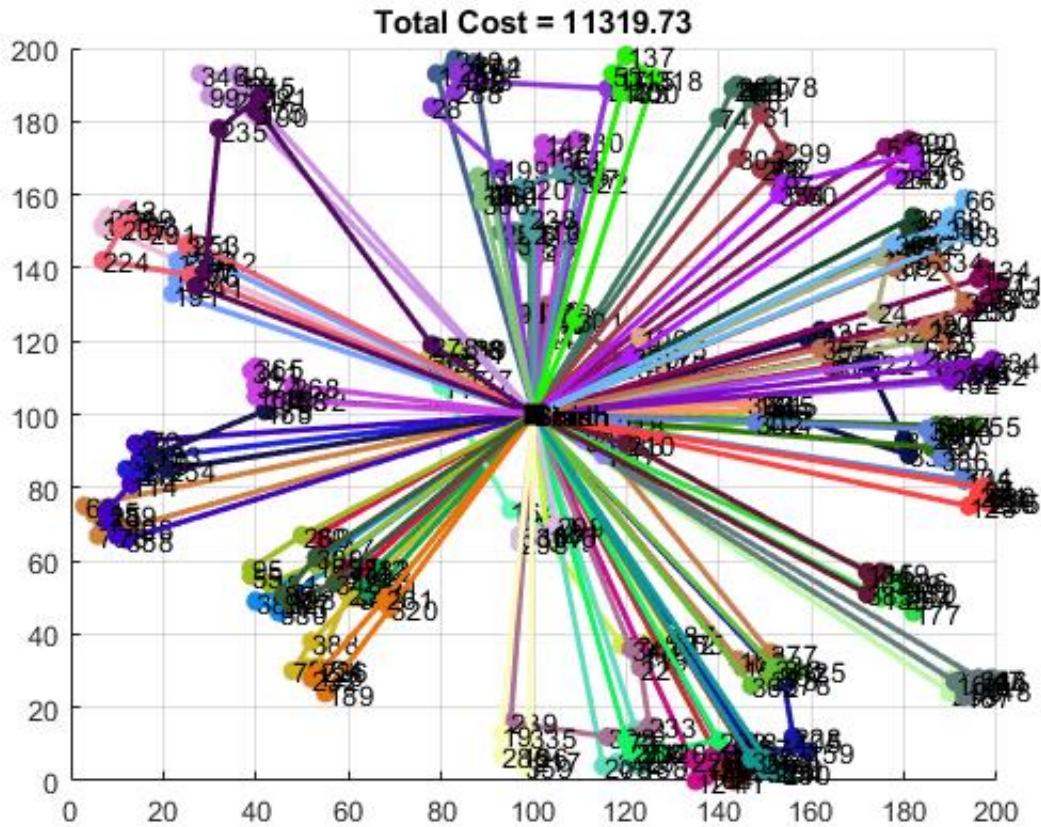
5.1.8 Πρόβλημα C1_4_1

Για το πρόβλημα C1_4_1 έχουμε 400 κόμβους με μέγιστο φορτίο του οχήματος 200 τεμάχια και συνολική ζήτηση 7109 προϊόντα και 10433 προϊόντα προς παραλαβή. οι αρχικές διαδρομές που προκύπτουν μέσω του αλγορίθμου εξοικονομήσεων Clarke & Wright απεικονίζονται στο παρακάτω γράφημα, έχουμε **54 διαδρομές** με συνολικό κόστος **11365,11 μονάδες**.



Γράφημα 15: Αρχική λύση C1_4_1

Οι βελτιωμένες διαδρομές και η τελική λύση του προβλήματος μέσω του αλγορίθμου της Μεταβλητής Γειτονιάς Αναζήτησης (VNS), απεικονίζεται στο παρακάτω πίνακα με συνολικό κόστος **11319,73 μονάδες**.



Γράφημα 16: Τελική λύση C1_4_1

Από τα γραφήματα των 400 κόμβων λόγω των πολλών διαδρομών και κόμβων εξυπηρέτησης δεν μπορούμε να διακρίνουμε τις πιθανές αλλαγές που έχουν γίνει για τη βελτίωση του κόστους των διαδρομών. Στο συγκεκριμένο πρόβλημα ο αλγόριθμος μας, VNS επέστρεψε τα καλύτερα αποτελέσματα, μόλις 1,75% απόκλιση από τα βέλτιστα της μεθόδου APA. Γι' αυτό το λόγο είχε σημασία να προβληθούν οι διαδρομές και οι κόμβοι ώστε να καταλάβουμε τι μάς οδήγησε σε μια τόσο καλή λύση. Αυτό που γίνεται εύκολα αντιληπτό από το παραπάνω γράφημα είναι ότι η εξαιρετική προσέγγιση της βέλτιστης λύσης, οφείλεται στην τοποθεσία των κόμβων, που βρίσκονται σε συμπλέγματα και είναι πολύ εύκολο να δημιουργηθούν οι γειτονιές εξυπηρέτησης.

Κεφάλαιο 6: Συγκρίσεις Αποτελεσμάτων και Συμπεράσματα

Σε αυτό το κεφάλαιο θα παρουσιαστούν τα σημαντικότερα σημεία του αλγορίθμου, οι δοκιμές που έγιναν για τη βελτίωση των αποτελεσμάτων, τα προβλήματα που εμφανίστηκαν κατά τη διάρκεια κατασκευής της βέλτιστης λύσης, ο τρόπος και οι τεχνικές που χρησιμοποιήθηκαν για την επίλυση αυτών των προβλημάτων καθώς και η αξιολόγηση των αποτελεσμάτων και η σύγκριση τους με τα βέλτιστα αποτελέσματα που προκύπτουν από την εφαρμογή της μεθόδου APA. (Emmanouil E. Zachariadis)

Αρχικά, η μέθοδος APA είναι ένας μεθευρετικός αλγόριθμος για το πρόβλημα δρομολόγησης οχημάτων με Ταυτόχρονη διανομή και παραλαβή (VRPSPD), το οποίο αποτελεί ένα σημαντικό πρόβλημα δρομολόγησης οχημάτων με πολλές εφαρμογές στα προβλήματα της αντίστροφης εφοδιαστικής αλυσίδας. Ο προτεινόμενος αλγόριθμος είναι ικανός να εξερευνήσει μια ευρεία λύση γειτονιών με κωδικοποίηση στατικών δοκιμαστικών κινήσεων σε ένα ειδικό σύνολο δεδομένων. Η εξερεύνηση του διαστήματος λύσεων συντονίζεται από τη χρήση ενός στοιχείου της στρατηγικής της διάχυσης (diversification), που ονομάζεται «υποσχέσεις» και βασίζεται στο κριτήριο απενεργοποίησης των περιορισμών (aspiration criteria) του αλγορίθμου περιορισμένης αναζήτησης (Tabu search). Η ιδέα των υποσχέσεων παρουσιάστηκε αρχικά για το πρόβλημα δρομολόγησης οχημάτων με δύο είδη πελατών κατά τη διάρκεια της διαδρομής (VRPB), και σχεδιάστηκε για να θεωρεί ολοκληρωμένες διαδρομές ως χαρακτηριστική λύση υπό εξέταση. Σε αντίθεση με αυτόν τον προαναφερθέντα σχεδιασμό, ο μηχανισμός των «υποσχέσεων», που προτείνεται στην παρούσα μελέτη, εκμεταλλεύεται περισσότερα βασικά χαρακτηριστικά, που ονομάζονται διανύσματα λύσης (solution Arcs). Ο συγκεκριμένος αλγόριθμος εφαρμόστηκε σε 18 VRPSPD προβλήματα προερχόμενα από τη βιβλιογραφία. Ο αλγόριθμος αποδείχθηκε τόσο ισχυρός όσο και αποτελεσματικός, βελτιώνοντας τις λύσεις των περισσότερων προβλημάτων. (Emmanouil E. Zachariadis)

6.1 Αλγόριθμος των εξοικονομήσεων των Clarke & Wright

Πρωτίστως, για κάθε πρόβλημα έγινε εισαγωγή των δεδομένων και κατασκευάστηκε ο πίνακας distance με όλες τις αποστάσεις μεταξύ των κόμβων και στη συνέχεια εφαρμόζουμε τη μέθοδο των εξοικονομήσεων των Clarke and Wright.

Αρχικά δημιουργούμε τις ελάχιστες διαδρομές-κύκλους ώστε κάθε πελάτης να εξυπηρετείται από ένα όχημα, δηλαδή, οι ελάχιστες αρχικές διαδρομές είναι τόσες όσοι και οι πελάτες. Επίσης υπολογίζονται οι συνολικές ζητήσεις (demand) και παραλαβές (pickup) για κάθε διαδρομή, το κόστος των διαδρομών- κύκλων, οι εξοικονομήσεις για τις ενώσεις μεταξύ των διαδρομών καθώς ελέγχεται και η εφικτότητα της λύσης, δηλαδή, εάν παραβιάζεται ο περιορισμός της χωρητικότητας του οχήματος. Τέλος για τις εξοικονομήσεις που είναι εφικτές επιλέγεται η μέγιστη εξοικονόμηση και προχωράμε στη συγχώνευση των δύο αυτών διαδρομών, οπότε πλέον έχουμε μια λιγότερη διαδρομή και με αυτόν τον τρόπο μειώνεται ο αριθμός των διαδρομών-κύκλων και το κόστος τους. Η διαδικασία αυτή συνεχίζεται μέχρις ότου να μην υπάρχει άλλη εφικτή ένωση διαδρομών.

Σε αυτό το σημείο λοιπόν παρατηρείται το πρώτο σημαντικό πρόβλημα του αλγορίθμου μας, το οποίο δεν είναι άλλο από το χρονικό διάστημα που χρειάζεται ο αλγόριθμος των Clarke & Wright για να βγάλει αποτελέσματα, δηλαδή τις αρχικές μας λύσεις. Αναλυτικά, για τα παραδείγματα των 100 κόμβων χρειάζονται κατά μέσο όρο 2 δευτερόλεπτα και τα παραδείγματα των 200 κόμβων χρειάζονται 53 δευτερόλεπτα αλλά το πραγματικό πρόβλημα βρίσκεται στα παραδείγματα των 400 κόμβων που χρειάζονται κατά μέσο όρο 15.068 δευτερόλεπτα, περίπου 4,5 ώρες για να εμφανιστούν αποτελέσματα. Έτσι, γίνεται εύκολα αντιληπτό ότι αυτοί οι χρόνοι είναι αναποτελεσματικοί και σε καμία περίπτωση ο συγκεκριμένος αλγόριθμος δεν θα μπορούσε να βοηθήσει σε ένα πρόβλημα υπό πραγματικών συνθηκών.

Οπότε, έπρεπε να βρεθεί μία λύση για τη μείωση του χρόνου υλοποίησης του αλγορίθμου. Αρχικά βρέθηκε το πρόβλημα, το οποίο ήταν ότι ο αλγόριθμος υπολόγιζε ξανά από την αρχή όλες τις εξοικονομήσεις. Στην πραγματικότητα όμως αυτό δεν χρειάζεται καθώς η εξοικονόμηση 2 διαδρομών δεν επηρεάζεται αν δεν έχει αλλάξει μια από τις δύο διαδρομές. Στη συνέχεια έγιναν οι διορθώσεις στον αλγόριθμο και πλέον όλες οι εξοικονομήσεις υπολογίζονται στην πρώτη επανάληψη και στις επόμενες επαναλήψεις δεν υπολογίζονται πάλι όλες αλλά μόνο οι εξοικονομήσεις που σχετίζονται με τη νέα διαδρομή, αυτή που δημιουργήθηκε από την συγχώνευση των 2 προηγούμενων διαδρομών. Αυτό είχε ως αποτέλεσμα τη δραστική μείωση των χρόνων υλοποίησης του αλγορίθμου. Στα παραδείγματα με τους 100 κόμβους χρειάζονται 0,4 δευτερόλεπτα, με τους 200 κόμβους χρειάζονται 2 δευτερόλεπτα και για τους 400 κόμβους παρατηρείται η μεγαλύτερη βελτίωση, όπου μειώθηκε ο χρόνος που έτρεχε ο αλγόριθμος από τα 15.068 δευτερόλεπτα στα 120 δευτερόλεπτα. Παρακάτω φαίνονται αναλυτικά οι λύσεις και οι χρόνοι υλοποίησης για κάθε παράδειγμα στον αρχικό Clarke & Wright (C&W old) και στον διορθωμένο Clarke & Wright (C&W new).

	C&W old			C&W new			gap	
	Cost	t(s)		Cost	t(s)		(%) Cost	%t
r101	1088,87	1,86		1088,87	0,37		0,00%	80,34%
r201	706,01	1,97		706,01	0,41		0,00%	79,39%
c101	1272,88	1,77		1272,88	0,35		0,00%	80,22%
c201	732,24	1,98		732,24	0,43		0,00%	78,34%
rc101	1138,30	1,86		1138,30	0,36		0,00%	80,64%
rc201	725,54	1,99		725,54	0,37		0,00%	81,48%
R1_2_1	3594,35	51,30		3594,35	1,97		0,00%	96,16%
R2_2_1	1842,66	54,88		1842,66	2,14		0,00%	96,09%
C1_2_1	3826,75	50,34		3826,75	2,01		0,00%	96,02%
C2_2_1	1852,43	55,98		1852,43	2,15		0,00%	96,15%
RC1_2_1	3572,26	50,16		3572,26	1,98		0,00%	96,04%
RC2_2_1	1795,40	55,20		1795,40	2,18		0,00%	96,05%
R1_4_1	10225,92	14463,67		10225,92	112,21		0,00%	99,22%
R2_4_1	4057,06	16061,18		4057,06	126,78		0,00%	99,21%
C1_4_1	11365,11	14648,26		11365,11	105,48		0,00%	99,28%
C2_4_1	4003,09	14591,11		4003,09	124,71		0,00%	99,15%
RC1_4_1	10033,86	15093,44		10033,86	119,17		0,00%	99,21%
RC2_4_1	3818,40	15493,92		3818,40	135,59		0,00%	99,12%

Cost: Συνολικό κόστος, **t(s):** χρόνος υλοποίησης του αλγορίθμου σε δευτερόλεπτα, **%Cost:** επί της εκατό απόκλιση του κόστους των δύο αυτών λύσεων, **%t:** επί της εκατό βελτίωση του χρόνου υλοποίησης του νέου αλγορίθμου.

Πίνακας 6.1: Σύγκριση αρχικού Clarke & Wright με το βελτιωμένο Clarke & Wright

Αρχικά, όπως έχει αναφερθεί και προηγουμένως, η μέθοδος των εξοικονομήσεων του Clarke & Wright έχει ως αποτέλεσμα την δημιουργία μιας αρκετά καλής αρχικής λύσης. Παρατηρούμε ότι από τα 18 προβλήματα μόνο σε 6 απέχουμε πάνω από 10% από τη βέλτιστη λύση της μεθόδου APA, σε 6 βρισκόμαστε μεταξύ του 6% και 8% και στα υπόλοιπα 6 η λύση απέχει λιγότερο από 6% από τα βέλτιστα. Τέλος, κατά μέσο όρο για όλα τα προβλήματα έχουμε **8,23%** απόκλιση από τα βέλτιστα αποτελέσματα της μεθόδου APA.

Στο παρακάτω πίνακα παρουσιάζονται το κόστος των αρχικών λύσεων και ο αριθμός των διαδρομών των προβλημάτων που προέκυψαν από τον αλγόριθμο Clarke & Wright (**Αρχικές C&W**), το κόστος των βέλτιστων λύσεων και των διαδρομών μέσω της μεθόδου APA (**APA**) και οι αποκλίσεις των δύο αυτών λύσεων (**gap**).

	Αρχικές (C&W)			APA			gap	
	Cost	n		Cost	n		(%) Cost	n
r101	1088,87	13		1009,95	12		7,81%	1
r201	706,01	3		666,20	3		5,98%	0
c101	1272,88	17		1220,18	16		4,32%	1
c201	732,24	5		662,07	5		10,60%	0
rc101	1138,30	11		1059,32	10		7,46%	1
rc201	725,54	3		672,92	3		7,82%	0
R1_2_1	3594,35	24		3375,19	23		6,49%	1
R2_2_1	1842,66	5		1665,58	5		10,63%	0
C1_2_1	3826,75	29		3641,89	28		5,08%	1
C2_2_1	1852,43	9		1726,73	9		7,28%	0
RC1_2_1	3572,26	25		3316,94	23		7,70%	2
RC2_2_1	1795,40	5		1560,00	5		15,09%	0
R1_4_1	10225,92	56		9668,18	53		5,77%	3
R2_4_1	4057,06	11		3560,73	10		13,94%	1
C1_4_1	11365,11	64		11125,14	63		2,16%	1
C2_4_1	4003,09	15		3549,20	15		12,79%	0
RC1_4_1	10033,86	54		9520,06	51		5,40%	3
RC2_4_1	3818,40	11		3414,90	11		11,82%	0

Cost: Συνολικό κόστος, n: αριθμός διαδρομών, %Cost: επί της εκατό απόκλιση του κόστους των δύο αυτών λύσεων.

Πίνακας 6.2: Σύγκριση Αρχικών λύσεων με τη μέθοδο APA

6.2 Αλγόριθμος μεταβλητής γειτονίας αναζήτησης (VNS)

Το επόμενο στάδιο είναι η κατασκευή του αλγορίθμου μεταβλητής γειτονίας αναζήτησης (VNS) με σκοπό τη βελτίωση τη αρχικής λύσης. Αρχικά ο VNS μας αποτελείται από τον αλγόριθμο τοπικής αναζήτησης 1-1 exchange και 2-0 relocate. Συγκεκριμένα, εφαρμόζεται πρώτα ο 1-1 exchange, όπου σε κάθε επανάληψη ελέγχεται η ανταλλαγή ενός κόμβου με όλους τους άλλους κόμβους είτε της ίδιας διαδρομής είτε κάποιας άλλης, η προσπέλαση μεταξύ των κόμβων γίνεται σειριακά, δηλαδή ξεκινάμε από το πρώτο κόμβο της πρώτης διαδρομής και στη τελευταία μας επανάληψη ελέγχουμε πιθανές αλλαγές για το τελευταίο κόμβο της τελευταίας διαδρομής. Σε κάθε επανάληψη για τα ζευγάρια κόμβων που η ανταλλαγή τους είναι εφικτή, δηλαδή δεν παραβιάζει κάποιο περιορισμό, υπολογίζεται το νέο συνολικό κόστος της διαδρομής (ένα πρόκειται για ανταλλαγή κόμβων της ίδιας διαδρομής) ή των διαδρομών (ένα πρόκειται για ανταλλαγή κόμβων διαφορετικών διαδρομών) και συγκρίνεται με το προηγούμενο συνολικό κόστος. Αν το νέο κόστος είναι μικρότερο, δηλαδή έχουμε μείωση του κόστους εξυπηρέτησης με την ανταλλαγή, αυτή η

λύση αποθηκεύεται σε έναν πίνακα. Η διαδικασία αυτή γίνεται για όλες τις πιθανές ανταλλαγές του κόμβου με τους υπόλοιπους. Στο τέλος της επαναληπτικής διαδικασίας για αυτό το κόμβο επιλέγεται η βέλτιστη λύση, αυτή που μειώνει το συνολικό κόστος περισσότερο και γίνεται η αλλαγή στις διαδρομές. Έπειτα συνεχίζουμε με τη νέα λύση στον επόμενο κόμβο και με αυτό το τρόπο ελέγχονται όλοι οι κόμβοι και οι πιθανές ανταλλαγές τους.

Μετά τον έλεγχο όλων των κόμβων ο αλγόριθμος 1-1 exchange έχει τελειώσει και μας έχει επιστρέψει μια βελτιωμένη λύση. Στην συνέχεια στη νέα λύση εφαρμόζεται ο 2-0 relocate όπου πάλι με τον ίδιο τρόπο για όλους τους κόμβους ελέγχονται οι πιθανές μετακινήσεις κάθε κόμβου, υπολογίζονται και αποθηκεύονται οι μετακινήσεις που επιφέρουν μείωση του συνολικού κόστους, αλλά πάλι σε κάθε επανάληψη εφαρμόζεται μία αλλαγή στις διαδρομές και αυτή είναι η μετακίνηση του κόμβου που επιφέρει την μεγαλύτερη μείωση του συνολικού κόστους. Η διαδικασία αυτή γίνεται για όλους τους κόμβους και με το πέρας όλων των κόμβων ο αλγόριθμος 2-0 relocate τελειώνει και έχουμε τη βελτιωμένη λύση του προβλήματος μας.

Σε αυτό το σημείο παρατηρήθηκε ότι ο 2-0 relocate στα περισσότερα παραδείγματα βελτιώνει ελάχιστα τη λύση που έχει προκύψει από τον 1-1 exchange και αυτό μας οδήγησε στο συμπέρασμα ότι η επιλογή στο να γίνονται οι βέλτιστες αλλαγές για κάθε κόμβο και ότι πρώτα εφαρμόζεται ο 1-1 exchange και έπειτα ο 2-0 relocate, έχει ως αποτέλεσμα ο 1-1 exchange να μας οδηγεί σε τοπικά ελάχιστα και για αυτό το λόγο τα αποτελέσματα μας δεν ήταν ιδιαίτερα ικανοποιητικά.

Έπειτα, έγιναν μερικές αλλαγές στον αλγόριθμο και πλέον οι 2 τοπικές αναζητήσεις συνδυάζονται σε κάθε επανάληψη για την εύρεση της βελτιωμένης λύσης. Ο νέος αλγόριθμος ονομάστηκε VNS(1). Πιο συγκεκριμένα, αρχικά εφαρμόζεται πάλι πρώτα ο 1-1 exchange αλλά αυτή τη φορά δεν ελέγχονται όλες οι πιθανές ανταλλαγές ώστε να εφαρμοστεί η βέλτιστη, αλλά η πρώτη εφικτή ανταλλαγή που δεν παραβιάζει τους περιορισμούς μας εφαρμόζεται άμεσα στη λύση. Στην συνέχεια είτε έχει γίνει κάποια ανταλλαγή μέσω του αλγορίθμου 1-1 exchange είτε όχι εφαρμόζεται για τον ίδιο κόμβο ο αλγόριθμος 2-0 και ελέγχεται αν μπορεί να γίνει κάποια μετακίνηση του κόμβου αυτού. Άρα σε μια επανάληψη μπορούν να γίνουν 2 κινήσεις ενός κόμβου και αυτό βοηθάει τον αλγόριθμο 2-0 relocate να είναι πιο δραστήσιος και εξυπηρετεί ώστε ο αλγόριθμος μας να μην πέφτει άμεσα σε τοπικά βέλτιστα και παρέχει μια πιο βελτιωμένη λύση.

Αφότου εφαρμοστεί ο αλγόριθμος μας στα 18 παραδείγματα μπορούμε να εξάγουμε το συμπέρασμα ότι πλέον οι αλλαγές μας στο κώδικα έχουν εφαρμογή και μπόρεσαν να βελτιώσουν τις αρχικές μας λύσεις. Πιο αναλυτικά, τα αποτελέσματα του αλγορίθμου μας VNS(1), καθώς και η σύγκριση τους με τα βέλτιστα της μεθόδου APA παρουσιάζονται στο παρακάτω πίνακα.

	VNS(1)			APA			gap	
	Cost	n		Cost	n		(%) Cost	n
r101	1077,67	13		1009,95	12		6,71%	1
r201	693,89	3		666,20	3		4,16%	0
c101	1271,24	17		1220,18	16		4,18%	1
c201	721,52	5		662,07	5		8,98%	0
rc101	1133,62	11		1059,32	10		7,01%	1
rc201	721,71	3		672,92	3		7,25%	0
R1_2_1	3557,36	24		3375,19	23		5,40%	1
R2_2_1	1833,12	5		1665,58	5		10,06%	0
C1_2_1	3799,49	29		3641,89	28		4,33%	1
C2_2_1	1810,24	9		1726,73	9		4,84%	0
RC1_2_1	3539,40	25		3316,94	23		6,71%	2
RC2_2_1	1755,12	5		1560,00	5		12,51%	0
R1_4_1	10187,20	56		9668,18	53		5,37%	3
R2_4_1	3959,26	11		3560,73	10		11,19%	1
C1_4_1	11344,33	64		11125,14	63		1,97%	1
C2_4_1	3932,97	15		3549,20	15		10,81%	0
RC1_4_1	10005,98	54		9520,06	51		5,10%	3
RC2_4_1	3745,80	11		3414,90	11		9,69%	0

Cost: Συνολικό κόστος, **n:** αριθμός διαδρομών, **%Cost:** επί της εκατό απόκλιση του κόστους των δύο αυτών λύσεων.

Πίνακας 6.3: Σύγκριση αλγορίθμου VNS(1) με τη μέθοδο APA

Ο αλγόριθμος VNS(1) κατά μέσο όρο βελτίωσε τη αρχική μας λύση κατά 1,15% και πλέον η απόκλιση μας από τη βέλτιστη λύση της μεθόδου APA είναι **7,01%**.

Στη συνέχεια παρά το γεγονός ότι μια απόκλιση γύρω στο 7% είναι μια καλή προσέγγιση του προβλήματος, υπάρχουν πράγματα που δεν έχουν δοκιμαστεί ακόμα και πιθανότατα να βελτιώσουν τη λύση περαιτέρω, όπως, η αλλαγή της σειράς που γίνονται οι τοπικές αναζητήσεις.

Συγκεκριμένα, έγινε η αλλαγή στη σειρά και πλέον γίνεται πρώτα ο 2-0 relocate και στη συνέχεια ο 1-1 exchange. Ο αλγόριθμος αυτός ονομάστηκε VNS(2). Αυτό που παρατηρήθηκε όμως είναι ότι σε μερικές προβλήματα ο VNS(2) βελτιώνει ελάχιστα τη λύση μας, σε κάποια άλλα η λύση είναι ίδια με τον VNS(1) και σε κάποια άλλα η λύση είναι ελαφρώς χειρότερη.

Έτσι κατασκευάστηκε ένας νέος αλγόριθμος όπου τρέχει ξεχωριστά ο VNS(1) και ο VNS(2) στην αρχική μας λύση, αποθηκεύονται οι λύσεις των δύο διαδικασιών και στο τέλος επιλέγεται η λύση που επιφέρει το μικρότερο συνολικό κόστος, δηλαδή η καλύτερη λύση.

Αυτός ο αλγόριθμος ονομάστηκε VNS(1,2) και τα αποτελέσματα και η σύγκριση τους με τα βέλτιστα αποτελέσματα της μεθόδου APA φαίνονται στο παρακάτω πίνακα.

	VNS(1,2)			APA			gap	
	Cost	n		Cost	n		(%) Cost	n
r101	1077,67	13		1009,95	12		6,71%	1
r201	693,89	3		666,20	3		4,16%	0
c101	1271,24	17		1220,18	16		4,18%	1
c201	721,52	5		662,07	5		8,98%	0
rc101	1133,47	11		1059,32	10		7,00%	1
rc201	721,71	3		672,92	3		7,25%	0
R1_2_1	3545,36	24		3375,19	23		5,04%	1
R2_2_1	1830,69	5		1665,58	5		9,91%	0
C1_2_1	3799,49	29		3641,89	28		4,33%	1
C2_2_1	1804,60	9		1726,73	9		4,51%	0
RC1_2_1	3525,29	25		3316,94	23		6,28%	2
RC2_2_1	1751,82	5		1560,00	5		12,30%	0
R1_4_1	10159,04	56		9668,18	53		5,08%	3
R2_4_1	3919,30	11		3560,73	10		10,07%	1
C1_4_1	11325,11	64		11125,14	63		1,80%	1
C2_4_1	3911,61	15		3549,20	15		10,21%	0
RC1_4_1	9968,53	54		9520,06	51		4,71%	3
RC2_4_1	3721,10	11		3414,90	11		8,97%	0

Cost: Συνολικό κόστος, **n:** αριθμός διαδρομών, **%Cost:** επί της εκατό απόκλιση του κόστους των δύο αυτών λύσεων.

Πίνακας 6.4: Σύγκριση αλγορίθμου VNS(1,2) με τη μέθοδο APA

Παρατηρούμε ότι η αλλαγή της σειράς των 2 τοπικών αναζητήσεων επηρεάζει κυρίως τα μεγαλύτερα προβλήματα των 400 κόμβων καθώς ο 2-0 relocate δουλεύει ως αφαιρετικό εργαλείο και δίνει περισσότερο χώρο σε ορισμένες διαδρομές με σκοπό να εισέρθουν κόμβοι ή να γίνουν ανταλλαγές με κόμβους που έχουν μεγάλο αριθμό ζητήσεων (demand) και παραλαβών (pickup) που θα βελτιώσουν τη λύση μας. Έτσι ο αλγόριθμος VNS(1,2) βελτιώνει τα αποτελέσματα κατά 0,25% από τον VNS(1) και η απόκλιση μας από τα βέλτιστα αποτελέσματα της μεθόδου APA είναι 6,75%.

Επίσης κάτι ακόμα που θα μπορούσε να μας βοηθήσει να φτάσουμε στο 6% απόκλιση είναι η εφαρμογή ενός ακόμη αλγορίθμου τοπικής αναζήτησης και αυτός δεν είναι άλλος από τον 2-opt. Η επιλογή του αλγορίθμου 2-opt δεν έγινε τυχαία, συγκεκριμένα επιλέχθηκε γιατί ο 1-1 exchange και ο 2-0 relocate έχουν κάνει τις αλλαγές μεταξύ διαφορετικών διαδρομών και ίσως πρέπει να επικεντρωθούμε σε κάθε διαδρομή ξεχωριστά και στη σειρά εξυπηρέτησης των κόμβων. Έτσι λοιπόν καταλήξαμε στον αλγόριθμο 2-opt όπου είναι ο πιο γνωστός αλγόριθμος στην επίλυση τέτοιων ζητημάτων.

Πλέον, ο αλγόριθμος μας αποτελείται από τις τοπικές αναζητήσεις 1-1 exchange, 2-0 relocate και τον 2-opt. Όμως και σε αυτή τη περίπτωση η σειρά που θα εφαρμοστούν παίζει σημαντικό

ρόλο. Έτσι δοκιμάστηκαν όλοι οι πιθανοί συνδυασμοί και συγκρίθηκαν τα αποτελέσματα μεταξύ τους. Εκεί που καταλήξαμε είναι ότι τα βέλτιστα αποτελέσματα προκύπταν κυρίως από την εφαρμογή πρώτα του 2-0 relocate έπειτα του 2-opt και τέλος του 1-1 exchange. Αυτός ο αλγόριθμος αποτελεί τον τελικό αλγόριθμο VNS.

Επίσης σε αυτό το σημείο δοκιμάστηκε, η προσπέλαση των κόμβων να γίνεται τυχαία μέσω της εξίσωσης `rand()` αλλά στις περισσότερες περιπτώσεις οι λύσεις που καταλήγαμε ήταν χειρότερες από αυτές που είχαμε μέχρι τώρα μέσω της σειριακής αναζήτησης, στις ελάχιστες περιπτώσεις όπου υπήρχαν λίγο καλύτερες λύσεις οι επαναλήψεις που χρειαζόταν και ο χρόνος μέχρι να μας βγάλει τα αποτελέσματα ήταν αρκετός και γι' αυτό το λόγο η τυχαία προσπέλαση θεωρήθηκε μη αποτελεσματική και παραμείναμε στη σειριακή επιλογή των επόμενων κόμβων.

Τέλος, με την ολοκλήρωση του αλγορίθμου της Μεταβλητής Γειτονιάς Αναζήτησης (VNS) τα αποτελέσματα μας σαφώς βελτιώθηκαν και κυρίως τα προβλήματα που είχαν μεγάλες αποκλίσεις. Συγκεκριμένα, μόνο σε 1 πρόβλημα είμαστε πάνω από 10%, σε 2 προβλήματα είμαστε κοντά στο 9%, σε 7 προβλήματα είμαστε ανάμεσα στο 6% με 8%, σε 7 είμαστε μεταξύ 3% και 4% και για το πρόβλημα C1_4_1 έχουμε τη καλύτερη λύση, στο 1,75%. Τέλος, κατά μέσο όρο για όλα τα προβλήματα απέχουμε κατά **6,15%** από τα βέλτιστα αποτελέσματα.

Αναλυτικά τα τελικά αποτελέσματα του VNS και η σύγκριση τους με τα βέλτιστα αποτελέσματα της μεθόδου APA παρουσιάζονται στο παρακάτω πίνακα.

	Τελικές			APA			gap	
	Cost	n		Cost	n		(%) Cost	n
r101	1077,67	13		1009,95	12		6,71%	1
r201	693,89	3		666,20	3		4,16%	0
c101	1271,24	17		1220,18	16		4,18%	1
c201	713,31	5		662,07	5		7,74%	0
rc101	1131,81	11		1059,32	10		6,84%	1
rc201	721,71	3		672,92	3		7,25%	0
R1_2_1	3525,73	24		3375,19	23		4,46%	1
R2_2_1	1788,89	5		1665,58	5		7,40%	0
C1_2_1	3796,06	29		3641,89	28		4,23%	1
C2_2_1	1794,67	9		1726,73	9		3,93%	0
RC1_2_1	3523,60	25		3316,94	23		6,23%	2
RC2_2_1	1729,84	5		1560,00	5		10,89%	0
R1_4_1	10123,59	56		9668,18	53		4,71%	3
R2_4_1	3886,66	11		3560,73	10		9,15%	1
C1_4_1	11319,73	64		11125,14	63		1,75%	1
C2_4_1	3879,59	15		3549,20	15		9,31%	0
RC1_4_1	9949,29	54		9520,06	51		4,51%	3
RC2_4_1	3660,44	11		3414,90	11		7,19%	0

Cost: Συνολικό κόστος, **n:** αριθμός διαδρομών, **%Cost:** επί της εκατό απόκλιση του κόστους των δύο αυτών λύσεων.

Πίνακας 6.5: Σύγκριση Τελικών λύσεων (VNS) με τη μέθοδο APA

6.3 Τελικά Συμπεράσματα

Έχοντας εκτελέσει τον αλγόριθμο στα 18 προβλήματα που αναφέρονται στο προηγούμενο κεφάλαιο έχουν παρατηρηθεί τα παρακάτω:

1. Ο αλγόριθμος των εξοικονομήσεων των Clarke & Wright δημιουργεί μια πολύ καλή αρχική λύση, μόλις 8,23% από τα βέλτιστα αποτελέσματα.
2. Στα προβλήματα που χρησιμοποιούνται λιγότερα οχήματα δηλαδή, όταν η χωρητικότητα των οχημάτων είναι μεγαλύτερη. Έχουμε μεγαλύτερες αποκλίσεις σε σχέση με τις αποκλίσεις των άλλων προβλημάτων.
3. Ο αλγόριθμος τοπικής αναζήτησης που συμβάλλει περισσότερο στη βελτίωση της λύσης είναι ο 2-opt.
4. Ο αλγόριθμος Μεταβλητής Γειτονίας Αναζήτησης VNS βελτιώνει την αρχική μας λύση κατά 2,08%, η μικρή ποσοστιαία βελτίωση οφείλεται στη καλή αρχική λύση, καθώς βρισκόμαστε πολύ κοντά στα βέλτιστα.

Για τις τελικές μας λύσεις παρατηρείται ότι:

1. Η καλύτερη εφαρμογή του αλγορίθμου βρίσκεται στο πρόβλημα **C1_4_1** με μόλις **1,75%** απόκλιση από τη βέλτιστη λύση.
2. Η χειρότερη εφαρμογή του αλγορίθμου βρίσκεται στο πρόβλημα **RC2_2_1** με **10,89%** απόκλιση από τη βέλτιστη λύση.
3. Κατά μέσο όρο σε όλα τα προβλήματα η απόκλιση από τα βέλτιστα είναι **6,15%**.

Συνοψίζοντας, ο κύριος στόχος της παρούσας εργασίας ήταν να δημιουργηθεί ένας αλγόριθμος για το πρόβλημα δρομολόγησης οχημάτων για ταυτόχρονες διανομές και παράλαβες προϊόντων (VRPSPD) ώστε ο μέσος όρος των λύσεις να μην αποκλίνει περισσότερο από το 6% από τα βέλτιστα και καμία λύση να μην υπερβαίνει το 10%.

Έπειτα λοιπόν από τις προηγούμενες παρατηρήσεις οφείλουμε να πούμε ότι τα αποτελέσματα του αλγορίθμου είναι απόλυτα ικανοποιητικά και εκπλήρωσαν πλήρως τις προσδοκίες μας. Αυτό όμως δεν σημαίνει ότι δεν επιδέχεται επιπλέον βελτιώσεις.

Οι αλλαγές και οι παραλλαγές που θα μπορούσαν να εφαρμοστούν είναι:

1. Να χρησιμοποιηθούν επιπλέον αλγόριθμοι τοπικοί αναζήτησης όπως είναι ο 2-2 exchange ή ο 3-opt.
2. Να επικεντρωθούμε σε συγκεκριμένα προβλήματα που έχουν μεγάλο κόστος με σκοπό τη περαιτέρω μείωση τους.

Βιβλιογραφία

1. <https://alfapeople.com/transform-manufacturing-supply-chain-operations-dynamics-365-safely-quickly/>
2. <https://www.terra.gr/el/%CE%B5%CF%86%CE%BF%CE%B4%CE%B9%CE%B1%CF%83%CF%84%CE%B9%CE%BA%CE%AE-%CE%B1%CE%BB%CF%85%CF%83%CE%AF%CE%B4%CE%B1/>
3. <https://www.mdpi.com/2227-9717/8/11/1363>
4. https://en.wikipedia.org/wiki/Vehicle_routing_problem
5. Chopra, S., & Meindl, P. (2007). Supply chain management. Strategy, planning & operation. In *Das summa summarum des management* (pp. 265-275). Gabler.
6. Zachariadis, E. E., & Kiranoudis, C. T. (2011). A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries. *Expert Systems with Applications*, 38(3), 2717-2726.
7. Μαρινάκης, Ι., & Μυγδαλάς, Α. (2008). Σχεδιασμός και βελτιστοποίηση της εφοδιαστικής αλυσίδας. *Εκδόσεις Σοφία, Θεσσαλονίκη*.
8. Μαρινάκης, Ι., & Μαρινάκη, Μ., & Μυγδαλάς, Α., (2019). *Προβλήματα Δρομολόγησης Οχημάτων στη Διαχείριση της εφοδιαστικής Αλυσίδας. Εκδόσεις Νέες τεχνολογίες, Αθήνα*.