



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ &

ΔΙΟΙΚΗΣΗΣ

Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων για το
Ομαδοποιημένο Πρόβλημα Δρομολόγησης Οχημάτων



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Βλαμάκης Βασίλειος

Επιβλέπων Καθηγητής: Μαρινάκης Ιωάννης

Χανιά 2022

Περιεχόμενα

Περίληψη.....	6
Κεφάλαιο 1: Εισαγωγή στην εφοδιαστική αλυσίδα	7
1.1 Εφοδιαστική αλυσίδα (Supply Chain)	7
1.2 Διαχείριση εφοδιαστικής αλυσίδας	8
1.3 Οι λειτουργίες της εφοδιαστικής αλυσίδας.....	10
1.4 Στόχοι της εφοδιαστικής αλυσίδας	11
1.5 Οφέλη της εφοδιαστικής αλυσίδας.....	12
Κεφάλαιο 2: Το πρόβλημα δρομολόγησης οχημάτων (VRP)	14
2.1 Εισαγωγή.....	14
2.2 Οι παράμετροι του προβλήματος δρομολόγησης	14
2.3 Οι περιορισμοί του προβλήματος	15
2.4 Στόχοι επίλυσης του προβλήματος	16
2.5 Βασικά προβλήματα δρομολόγησης οχημάτων.....	16
2.5.1 Το πρόβλημα του πλανόδιου πωλητή	16
2.5.2 Το περιορισμένης χωρητικότητας πρόβλημα δρομολόγησης οχημάτων (CVRP)	17
2.5.3 Το πρόβλημα δρομολόγησης οχημάτων με πολλαπλές επιστροφές	18
στην αποθήκη (MVRP)	18
2.5.4 Το ανοιχτό πρόβλημα δρομολόγησης οχημάτων.....	19
2.5.5 Το ανοιχτό κλειστό πρόβλημα δρομολόγησης.....	20
2.5.6 Το πρόβλημα δρομολόγησης οχημάτων με μόνο περιορισμό τον χρόνο	20
μετάβασης.....	20
2.5.7 Το πρόβλημα δρομολόγησης οχημάτων με περιορισμούς.....	21
χωρητικότητας και χρονικά παράθυρα	21
2.5.8 Το πρόβλημα δρομολόγησης οχημάτων με ύπαρξη πολλαπλών	22
αποθηκών	22

2.5.9 Το πρόβλημα δρομολόγησης οχημάτων με δύο είδη πελατών κατά την διάρκεια της διαδρομής.....	22
2.5.10 Πρόβλημα δρομολόγησης οχημάτων με διανομή και παραλαβή	23
προϊόντων κατά την διάρκεια της διαδρομής	23
2.5.11 Στοχαστικό προβλήμα δρομολόγησης	24
2.6 Το ομαδοποιημένο πρόβλημα δρομολόγησης	25
2.7 Ορίσματα του ομαδοποιημένου προβλήματος δρομολόγησης	26
2.7.1 Μεταβλητές	26
2.7.2 Περιορισμοί	27
2.7.3 Μοντελοποίηση του προβλήματος	28
Κεφάλαιο 3: Αλγόριθμοι επίλυσης προβλημάτων δρομολόγησης	30
3.1 Εισαγωγή.....	30
3.2 Ευρετικοί αλγόριθμοι	30
3.2.1 Αλγόριθμοι απληστίας	31
3.2.2 Προσεγγιστικοί αλγόριθμοι.....	33
3.2.3 Αλγόριθμοι τοπικής αναζήτησης	33
3.3 Μεθευρετικοί Αλγόριθμοι	34
3.4 Εξελικτικοί και εμπνευσμένοι από την φύση αλγόριθμοι.....	35
3.4.1 Εξελικτικοί αλγόριθμοι.....	35
3.4.2 Αλγόριθμοι εμπνευσμένοι από την φύση.....	36
3.5 Ανάλυση των αλγορίθμων που χρησιμοποιήθηκαν για την επίλυση του ομαδοποιημένου προβλήματος δρομολόγησης οχημάτων.....	37
3.5.1 Ο αλγόριθμος πλησιέστερου γείτονα (nearest neighbor algorithm)	37
3.5.2 Ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων (Particle Swarm Optimization).....	39
3.5.3 Αλγόριθμοι τοπικής αναζήτησης	42
Κεφάλαιο 4: Επίλυση του ομαδοποιημένου προβλήματος δρομολόγησης οχημάτων	45
4.1 Εισαγωγή.....	45
4.2 Περιγραφή προβλήματος.....	45

4.3 Επίλυση του ομαδοποιημένου προβλήματος δρομολόγησης οχημάτων (CluVRP).....	46
4.4 Συγκεντρωτικοί πίνακες λύσεων-αποκλίσεων	52
4.5 Συμπεράσματα	55

Ευχαριστίες

Θα ήθελα να ευχαριστήσω ιδιαίτερα τον επιβλέποντα καθηγητή μου κ. Ιωάννη Μαρινάκη , για την καθοδήγηση που μου προσέφερε και το χρόνο που μου αφιέρωσε.

Την εργασία την αφιερώνω στην οικογένειά μου που με στήριξε σε όλο το διάστημα των σπουδών.

Περίληψη

Η διαδρομή που θα ακολουθήσει ένα όχημα για την εξυπηρέτηση των εκάστοτε πελατών είναι απαραίτητο να είναι η βέλτιστη, και για τον εξυπηρετούμενο αλλά και τον εξυπηρετητή. Όσο αφορά τον εξυπηρετούμενο, η βέλτιστη διαδρομή θεωρείται η πιο σύντομη και γρήγορη ενώ για τον εξυπηρετητή θεωρείται η πιο οικονομική. Για αυτόν τον λόγο έχουν δημιουργηθεί διάφορες μέθοδοι βελτιστοποίησης διαδρομών που συνδυάζουν τις παραπάνω απαιτήσεις για το καλύτερο δυνατό αποτέλεσμα.

Η διπλωματική αυτή εργασία αφορά μια παραλλαγή του προβλήματος δρομολόγησης χωρητικότητας οχημάτων (CVRP) που ονομάζεται Clustered VRP (CluVRP). Στο CluVRP, οι πελάτες χωρίζονται σε προκαθορισμένες ομάδες. Οι πελάτες που αντιστοιχούν σε ένα μεμονωμένο σύμπλεγμα πρέπει να επισκέπτονται όλοι από το ίδιο όχημα πριν αυτό φύγει από το σύμπλεγμα. Η έννοια της ομαδοποίησης είναι ευρέως γνωστή λόγω των οικονομικών της επιπτώσεων και της μειωμένης πολυπλοκότητάς της στη μοντελοποίηση και την επίλυση μεγάλου φάσματος εφαρμογών.

Αρχικά, χρησιμοποιείται ένας αλγόριθμος αρχικής λύσης που στοχεύει στη δημιουργία μιας αρχικής διαδρομής, η οποία όμως δεν είναι βέλτιστη. Αυτή η διαδρομή είναι απαραίτητη επειδή οι ακόλουθοι αλγόριθμοι θα την χρησιμοποιήσουν ως δεδομένο για να την φέρουν στην βέλτιστη κατάσταση. Και στη συνέχεια χρησιμοποιείται ένας αλγόριθμος εμπνευσμένος από την φύση και δύο αλγόριθμοι τοπικής αναζήτησης για τη βελτιστοποίηση της αρχικής λύσης.

Οι αλγόριθμοι που χρησιμοποιήθηκαν σε αυτήν την εργασία είναι, αρχικά ο πλησιέστερος γείτονας, έπειτα ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων (PSO), και τέλος οι 1-1 exchange και 2-opt.

Κεφάλαιο 1: Εισαγωγή στην εφοδιαστική αλυσίδα

1.1 Εφοδιαστική αλυσίδα (Supply Chain)

Μια αλυσίδα εφοδιασμού είναι μια σειρά διαδικασιών που περιλαμβάνουν την προμήθεια, την αποτελεσματική ροή και αποθήκευση πρώτων υλών, εξαρτημάτων, ημικατεργασμένων και τελικών προϊόντων καθώς και τον σχεδιασμό, την υλοποίηση και τον έλεγχο της ροής των σχετικών πληροφοριών από την πηγή στο σημείο κατανάλωσης (αγορά) για την κάλυψη των απαιτήσεων των πελατών.

Μια αλυσίδα εφοδιασμού περιλαμβάνει όλες τις δραστηριότητες και διαδικασίες που σχετίζονται με το σχεδιασμό και την παραγωγή προϊόντων και υπηρεσιών, τη διανομή και την εξυπηρέτηση πελατών που εκτελούνται από δύο ή περισσότερες εταιρείες για την κάλυψη των αναγκών των πελατών. Επιπλέον, περιλαμβάνει συντονισμό και συνεργασία με όλους τους εταίρους της εφοδιαστικής αλυσίδας που μπορεί να είναι μεσάζοντες, προμηθευτές, εταιρείες παροχής υπηρεσιών και πελάτες.

Απ' ότι καταλαβαίνουμε λοιπόν η Εφοδιαστική αλυσίδα έχει πολύ μεγάλη βαρύτητα στον τομέα του μάνατζμεντ της επιχείρησης καθώς ασχολείται με τον σχεδιασμό, την υλοποίηση αποφάσεων και την στρατηγική ενός οργανισμού.



1.1 Σχηματική απεικόνιση της Εφοδιαστικής Αλυσίδας

1.2 Διαχείριση εφοδιαστικής αλυσίδας

Μια εφοδιαστική αλυσίδα ή ένα δίκτυο εφοδιαστικής (logistics network) σχηματίζεται απ'όλα τα στάδια που έχουν να κάνουν, έμμεσα ή άμεσα, με την ικανοποίηση των απαιτήσεων του πελάτη. Η διαχείριση της εφοδιαστικής αλυσίδας περιλαμβάνει όλη την διαδικασία μεταφοράς και αποθήκευσης των πρώτων υλών, και τελικών αγαθών από τα σημεία προέλευσης στα σημεία κατανάλωσης.

Η αποτελεσματική διαχείριση, μίας αλυσίδας εφοδιασμού έχει ως στόχο την ικανοποίηση των πελατών με όσο δυνατόν χαμηλότερο κόστος, με σωστή ποσότητα του σωστού προϊόντος και στον σωστό τόπο και χρόνο. Για να συμβεί αυτό πρέπει οι πρώτες ύλες, τα αποθέματα και τα έτοιμα προϊόντα, να είναι διαθέσιμα σε σωστές ποσότητες, στο σωστό τόπο και την σωστή στιγμή. Με άλλα λόγια η βελτιστοποίηση της εφοδιαστικής αλυσίδας, συντέλει στην δημιουργία αξίας και ανταγωνιστικού πλεονεκτήματος για την επιχείρηση.

Για να συμβούν τα παραπάνω και να εφαρμοστεί ένας ορθός τρόπος διαχείρισης είναι αναγκαίο να δημιουργηθεί ένα συστημικό μοντέλο που θα αναλύει τις παραμέτρους και τις

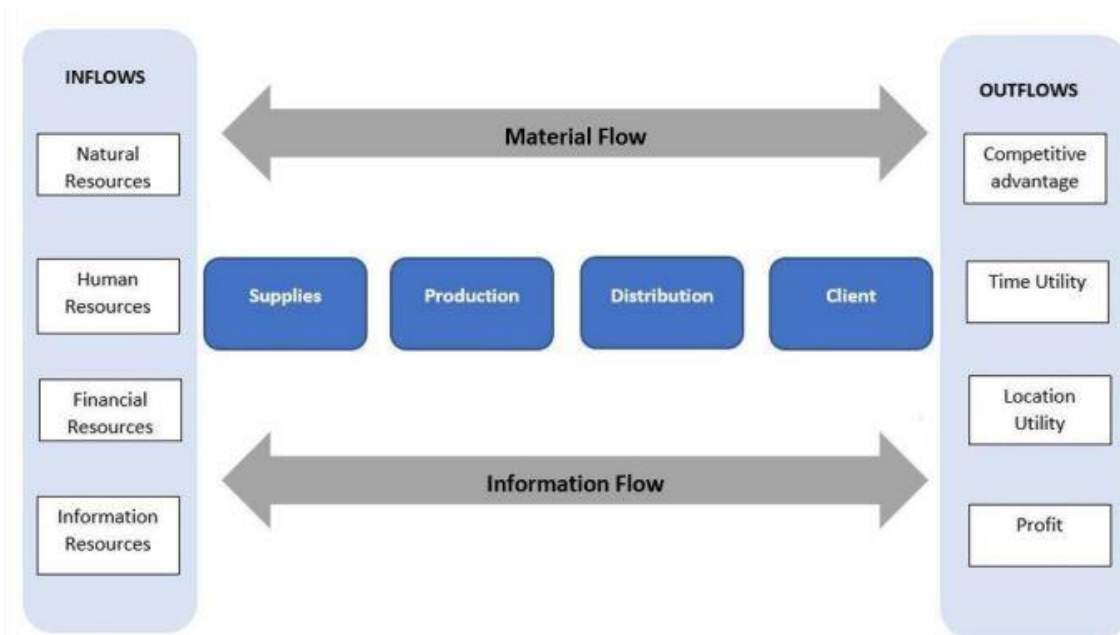
μεταβλητές που συνθέτουν μια εφοδιαστική αλυσίδα. Έγκειται για ένα δυναμικό σύστημα στο οποίο κάθε στάδιο αλληλοεπιδρά με τα υπόλοιπα και την ροή πληροφορίας αλλά και υλικών να κινείται αμφίπλευρα. Το σύστημα αυτό αποτελείται από εισερχόμενες ροές, εξερχόμενες ροές καθώς και τις διαδικασίες που πραγματοποιούνται για να παραδοθεί το προϊόν στον πελάτη.

Από εισερχόμενες ροές έχουμε:

- Φυσικούς πόρους
- Οικονομικοί πόροι
- Ανθρώπινο Δυναμικό
- Πληροφοριακοί Πόροι

Και από εξερχόμενες:

- Ανταγωνιστικό πλεονέκτημα
- Χρησιμότητα χρόνου
- Χρησιμότητα τόπου
- Η κερδοφορία



Εικόνα 1.2 Ανάλυση διαχείρισης της Εφοδιαστικής Αλυσίδας

Για να είναι μια επιχείρηση ανταγωνιστική, πολύ σημαντικό ρόλο παίζει η ορθή διαχείριση της εφοδιαστικής αλυσίδας. Τα δύο κυριότερα επίπεδα από τα οποία αποτελείται η βελτιστοποίηση της εφοδιαστικής είναι αυτό του προγραμματισμού και της εκτέλεσης.

- Προγραμματισμός: Πρόβλεψη ζήτησης βασιζόμενη σε παλαιότερα δεδομένα πωλήσεων και προμηθειών.
- Εκτέλεση: Εφαρμογή του σχεδίου που έχει δημιουργηθεί με βάση τον προγραμματισμό που έχει γίνει, το οποίο παρακολουθείται και ελέγχεται καθ' όλη την διάρκεια της εκτέλεσης.

1.3 Οι λειτουργίες της εφοδιαστικής αλυσίδας

Χωρίζονται σε δύο βασικές κατηγορίες:

Οι κύριες λειτουργίες: είναι αυτές που ξοδεύουν περισσότερο χρόνο για την επιχείρηση αλλά έχουν όμως και το μεγαλύτερο βάρος στην απόδοση της αλυσίδας. Οι διεργασίες αυτές είναι:

- οι μεταφορές
- η επεξεργασία των παραγγελιών
- η διαχείριση των αποθεμάτων

Οι υποστηρικτικές λειτουργίες: είναι οι διεργασίες που είναι απαραίτητες για τη διασφάλιση των κύριων λειτουργιών, ενώ παράλληλα στηρίζει η μία την άλλη. Αυτές είναι:

- Προμήθειες
- Αποθήκευση
- Διακίνηση υλικών
- Προγραμματισμός παραγωγής
- Προστατευτική συσκευασία
- Πληροφοριακή υποστήριξη
- Πρόβλεψη της ζήτησης

Για να είναι μια εφοδιαστική αλυσίδα ολοκληρωμένη και αποδοτική πρέπει οι λειτουργίες της να αλληλοεπιδρούν.

Αναλυτικότερα, η εύρεση του βέλτιστου μονοπατιού μεταφοράς προϊόντων από την αποθήκη προς του πελάτες με στόχο το ελάχιστο κόστος και με βάση τους περιορισμούς που έχει το όχημα είναι μια από τις κύριες αποφάσεις που πρέπει να παρθούν για να λειτουργήσει σωστά και αποδοτικά η εφοδιαστική αλυσίδα. Επίσης, εξίσου σημαντική είναι και η συνεχής παρακολούθηση των αποθεμάτων σε συνδυασμό με την επεξεργασία των παραγγελιών έτσι ώστε να μην υπάρξει καμία έλλειψη στην αποθήκη και ταυτόχρονα να ικανοποιηθούν οι πελάτες.

Ο σημαντικότερος στόχος της επιχείρησης είναι η ικανοποίηση των πελατών της, οπότε μια ορθολογική και αξιόπιστη εφοδιαστική αλυσίδα επιφέρει ένα ανταγωνιστικό πλεονέκτημα, ελαχιστοποιεί το κόστος ενώ παράλληλα συνεισφέρει στην κερδοφορία της επιχείρησης.

1.4 Στόχοι της εφοδιαστικής αλυσίδας

Η εφοδιαστική αλυσίδα έχει ως σκοπό την ικανοποίηση των επιχειρησιακών στόχων με το μικρότερο δυνατό κόστος. Ή όπως έχει αναφέρει και ο Gattorna το 1997, η εφοδιαστική επιδιώκει να βρίσκεται το σωστό προϊόν, στη σωστή ποσότητα και ποιότητα, στον σωστό τόπο, στον σωστό χρόνο με το ελάχιστο δυνατό κόστος. Επομένως οι βασικοί στόχοι της εφοδιαστικής αλυσίδας είναι:

- 1) Η ικανοποίηση των πελατών
- 2) Η μείωση του κόστους

Πέραν όμως των βασικών στόχων, η εφοδιαστική έχει και επιμέρους στόχους. Ένας από αυτούς είναι η επιδίωξη υψηλού επιπέδου εξυπηρέτησης σε συνδυασμό με την σωστή διαχείριση των αποθεμάτων. Τα αποθέματα παίζουν έναν περίεργο ρόλο στην όλη διαδικασία, καθώς τα υψηλά αποθέματα ισοδυναμούν με υψηλό κόστος (συντήρησης, αποθήκευσης) ενώ παράλληλα τα χαμηλά αποθέματα έχουν υψηλό κίνδυνο σε ελλείψεις.

Για να υπάρχει λοιπόν μια ισορροπία στη διατήρηση των αποθεμάτων και στον ρυθμό παραγωγής, θα πρέπει η εφοδιαστική αλυσίδα να χρησιμοποιήσει τεχνικές έγκυρης και έγκαιρης πρόβλεψης της ζήτησης.

Ακόμη ένας στόχος είναι ο έλεγχος και η διασφάλιση της ποιότητας των υπηρεσιών. Γνωρίζοντας την δεδομένη ποιότητα σύμφωνα από την διαδικασία παραγωγής ο στόχος της εφοδιαστικής αλυσίδας είναι η διατήρηση της δεδομένης αυτής ποιότητας μέχρι να γίνει η παράδοση στον πελάτη. Η ποιότητα του προϊόντος παίζει πολύ σημαντικό ρόλο στην ακεραιότητα της επιχείρησης, για τον λόγο αυτό η εφοδιαστική αλυσίδα έχει κάποια στοιχεία με στόχο την επίτευξη της επιθυμητής ποιότητας στις υπηρεσίες της επιχείρησης. Δηλαδή:

- Δυναμικότητα: Η ικανότητα της επιχείρησης να μεταφέρει, μέσα στα χρονικά διαστήματα που έχει στην διαθεσή της, το προϊόν.
- Διαθεσιμότητα: Η ικανότητα της επιχείρησης να διαθέτει το σωστό απόθεμα για κάθε προϊόν.
- Συνέπεια: Η δυνατότητα να γίνονται τα παραπάνω σε καθημερινή βάση χωρίς λάθη.

1.5 Οφέλη της εφοδιαστικής αλυσίδας

Η εφοδιαστική αλυσίδα είναι απαραίτητο κομμάτι της επιχείρησης και είναι αναγκαία για την επίτευξη των στόχων της. Κάποια από τα οφέλη της είναι :

- Μείωση λειτουργικών εξόδων. Αυτό επιτυγχάνεται με την έγκαιρη παράδοση των προϊόντων καθώς και με την σωστή διαχείριση των αποθεμάτων. Παράλληλα μειώνει και το κόστος παραγωγής με την αγορά κατάλληλων ποσοτήτων πρώτων υλών.
- Δημιουργία ανταγωνιστικού πλεονεκτήματος. Η βελτίωση της χρηματοοικονομικής θέσης της επιχείρησης με την μείωση των εξόδων δημιουργεί ένα ανταγωνιστικό πλεονέκτημα.

- Ενθάρρυνση εξυπηρέτησης πελατών. Κύριος στόχος της εφοδιαστικής αλυσίδας είναι η ικανοποίηση των πελατών. Αυτό επιτυγχάνεται με την σωστή διανομή των προϊόντων και την κατάλληλη ποιότητα.

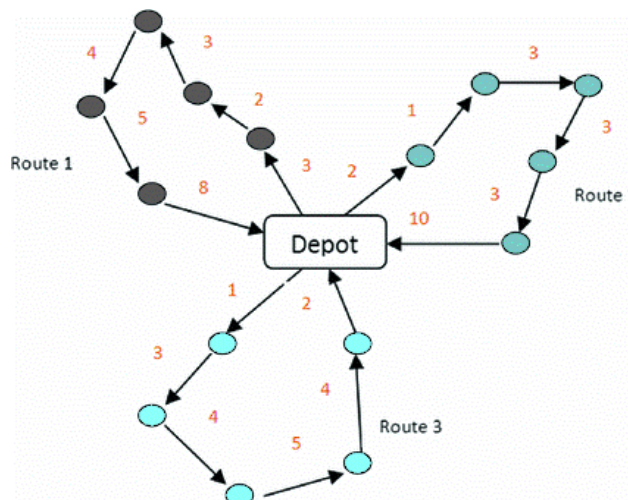
Η εφοδιαστική αλυσίδα αποτελεί ένα δυναμικό σύστημα που εξελίσσεται και προσαρμόζεται συνεχώς στις ανάγκες του περιβάλλοντος. Είναι διαδεδομένη τόσο λόγω της αποτελεσματικότητας και των στόχων της. Με την ορθή διαχείριση και την ολοκληρωμένη εφαρμογή της εφοδιαστικής αλυσίδα, η επιχείρηση έχει στην κατοχή της ένα πολύ ισχυρό όπλο απέναντι στο ανταγωνιστικό περιβάλλον.



Κεφάλαιο 2: Το πρόβλημα δρομολόγησης οχημάτων (VRP)

2.1 Εισαγωγή

Το πρόβλημα δρομολόγησης οχημάτων είναι ένα πρόβλημα συνδυαστικής βελτιστοποίησης και ακέραιου προγραμματισμού το οποίο βρίσκει ένα σύνολο διαδρομών για ένα στόλο οχημάτων με σκοπό να παραδώσουν τα προϊόντα σε ένα σύνολο πελατών. Τα δεδομένα του προβλήματος κατά βάση είναι οι πελάτες, τα οχήματα και οι αποθήκες. Η διανομή των προϊόντων επιτυγχάνεται μέσω ενός οδικού δικτύου. Το δίκτυο αυτό αναπαριστάται μέσω ενός γραφήματος, τόξων και κόμβων, όπως βλέπουμε και από το παρακάτω σχήμα.



Εικόνα 2.1.1: Αναπαράσταση δικτύου

Τα τόξα εδώ αναπαριστούν τις διαδρομές, και οι πελάτες είναι οι κόμβοι.

2.2 Οι παράμετροι του προβλήματος δρομολόγησης

Για τον πελάτη:

- Η τοποθεσία του.
- Τα χρονικά διαστήματα που μπορεί να εξυπηρετηθεί.
- Η ζήτηση του πελάτη, δηλαδή η ποσότητα προϊόντων που θέλει.

- Ο χρόνος εξυπηρέτησης του.
- Είδος εξυπηρέτησης που απαιτεί, διανομή προϊόντων, παραλαβή προϊόντων ή και τα δύο.

Για την αποθήκη:

- Η χωρητικότητα της.
- Η τοποθεσία της.
- Ο όγκος οχημάτων.

Για τα οχήματα:

- Το πλήθος των διαθέσιμων οχημάτων.
- Η χωρητικότητα του κάθε οχήματος.
- Το λειτουργικό κόστος (συντήρη, καύσιμα, κτλ).
- Την προσβασιμότητα του οχήματος.
- Η αποθήκη από την οποία ξεκινούν.

2.3 Οι περιορισμοί του προβλήματος

Οι περιορισμοί που θέτονται είναι και αυτοί που προσδιορίζουν το πρόβλημα, και αναφέρονται όπως και τα δεδομένα στους πελάτες, στα προϊόντα και στα οχήματα. Κάποιες κατηγορίες τους είναι:

- Οι χρονικοί περιορισμοί.
- Η σειρά με την οποία θα γίνει η επίσκεψη των πελατών.
- Οι περιορισμοί όσο αναφορά το είδος εξυπηρέτησης.
- Οι περιορισμοί όσο αναφορά τον αριθμό των οχημάτων που είναι διαθέσιμα.
- Οι περιορισμοί όσο αναφορά την χωρητικότητα των οχημάτων.
- Την διαθεσιμότητα μηχανημάτων φορτοεκφόρτωσης.

Γενικότερα οι περιορισμοί που μπορούν να τεθούν είναι πολλοί αλλά με μία σωστή διαχείρισή τους το πρόβλημα μπορεί να λυθεί.

2.4 Στόχοι επίλυσης του προβλήματος

Κυριότερος στόχος όπως έχει αναφερθεί και προηγουμένως είναι η ελαχιστοποίηση τους κόστους σε συνδιασμό με την ικανοποίηση των πελατών. Πιο συγκεκριμένα οι στόχοι επίλυσης του προβλήματος αποτυπώνονται ως εξής:

- Ελαχιστοποίηση του αριθμού των οχημάτων και των οδηγών που απαιτούνται για την κάλυψη όλων των ειδικών αναγκών του πελάτη.
- Δημιουργία διαδρομών χαμηλού κόστους.
- Ελαχιστοποιήστε τις κυρώσεις για έλλειψη εξυπηρέτησης πελατών.
- Εξισορρόπηση των διαδρομών του οχήματος που θα πραγματοποιηθούν με βάση τον χρόνο που απαιτείται για το ταξίδι και το φορτίο που αντιστοιχεί σε κάθε διαδρομή.

2.5 Βασικά προβλήματα δρομολόγησης οχημάτων

2.5.1 Το πρόβλημα του πλανόδιου πωλητή

Το πρόβλημα του πλανόδιου πωλητή αποτελεί την πιο απλή εκδοχή ενός προβλήματος δρομολόγησης και συγκροτεί την βάση για τα πιο σύνθετα προβλήματα δρομολόγησης. Ως δεδομένα έχουμε τα εξής:

- Το σύνολο των κόμβων που πρέπει να επισκεφθεί ο πωλητής.
- Την αφετηρία.
- Τις αποστάσεις των κόμβων.

Επίσης δεν υπάρχουν περιορισμοί ως προς τα παρακάτω:

- Το μέγεθος και την χωρητικότητα του οχήματος.
- Την ζήτηση των πελατών.
- Τον χρόνο άφιξης του πωλητή σε κάθε κόμβο και γενικότερα στον χρόνο που έχει στην διαθεσή του.

Το πρόβλημα αφορά την εύρεση της συντομότερης διαδρομής για έναν πωλητή ο οποίος ξεκινάει από την αφετηρία, στην συνέχεια θα πρέπει να επισκεφθεί όλους τους κόμβους από μία μόνο φορά και τέλος να επιστρέψει πάλι στην αφετηρία. Η διατύπωση του προβλήματος μπορεί να μοιάζει απλή αλλά στην πράξη δεν είναι, αφού μέχρι και σήμερα δεν έχει βρεθεί κάποιος τρόπος



επίλυσης, που να λύνει το πρόβλημα μέσα σε λογικά χρονικά πλαίσια. Παρόλα αυτά έχουν αναπτυχθεί κάποιοι αλγόριθμοι με σκοπό τη λύση του, και δίνουν ένα αποτέλεσμα αρκετά κοντά στο βέλτιστο.

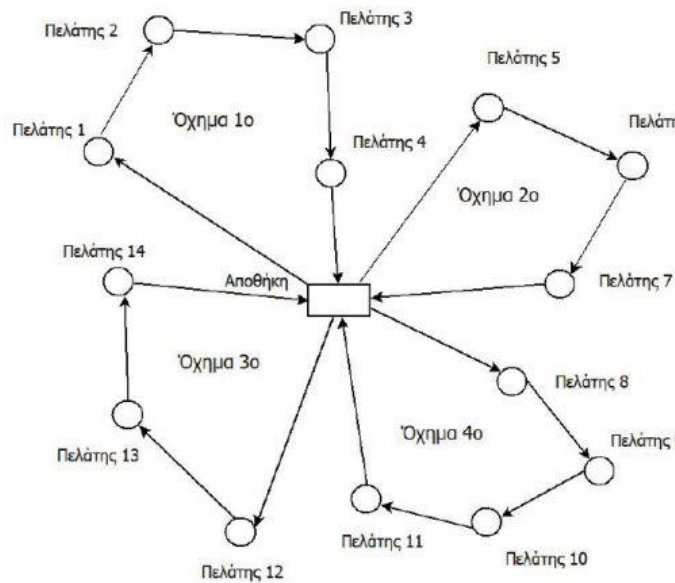
2.5.2 Το περιορισμένης χωρητικότητας πρόβλημα δρομολόγησης οχημάτων (CVRP)

Εδώ παρατηρούνται όλες τις περιπτώσεις που είναι απαραίτητη η χρήση πολλών οχημάτων. Αυτό συμβαίνει επειδή:

- Η συνολική ζήτηση πελατών είναι μεγαλύτερη από την απόδοση του οχήματος.
- Η ζήτηση πρέπει να ικανοποιηθεί εντός συγκεκριμένου χρονικού πλαισίου, κάτι που δεν είναι δυνατό με ένα μόνο όχημα.
- Δεν πρέπει να συγχέονται οι ανάγκες διαφορετικών πελατών για διαφορετικά προϊόντα.

Για αυτούς τους λόγους παράγονται πολλαπλές διαδρομές έτσι ώστε να καλυφθεί η ζήτηση με επιτυχία. Τα χαρακτηριστικά αυτών των διαδρομών είναι:

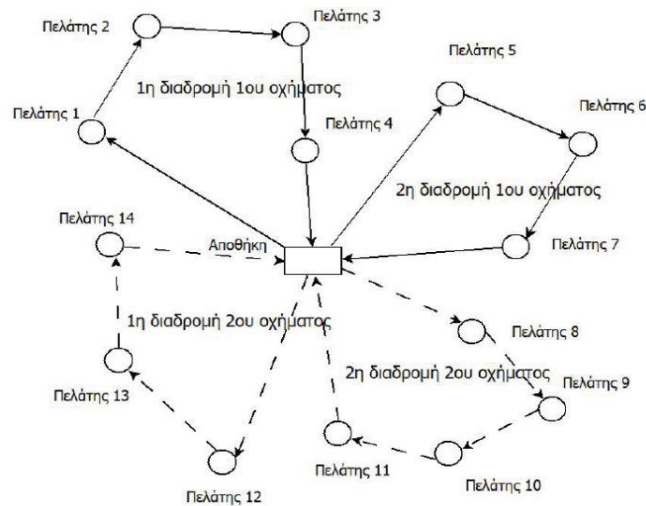
- Κάθε πελάτης εξυπηρετείται μία φορά.
- Πρέπει να εξυπηρετηθούν όλοι οι πελάτες.
- Κάθε διαδρομή αρχίζει από την αποθήκη και επιστρέφει σε αυτή μετά την ολοκλήρωσή της.
- Η ζήτηση της κάθε διαδρομής δεν μπορεί να υπερβεί την χωρητικότητα του οχήματος.



Εικόνα 2.3.1.1 Απεικόνιση προβλήματος περιορισμένης χωρητικότητας

2.5.3 Το πρόβλημα δρομολόγησης οχημάτων με πολλαπλές επιστροφές στην αποθήκη (MVRP)

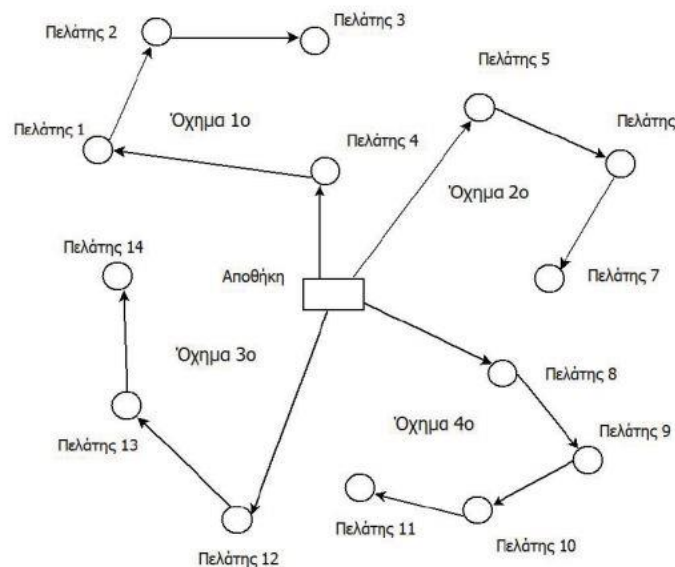
Τα περισσότερα προβλήματα δρομολόγησης οχημάτων επιτρέπουν μόνο μία διαδρομή στα οχήματα. Και αυτό διότι, η φόρτωση των οχημάτων και οι αποστάσεις που διανύονται με αυτά συνήθως περιλαμβάνουν ολόκληρο τον χρόνο που είναι διαθέσιμος στο όχημα σε μια εργάσιμη ημέρα. Ωστόσο, εάν ο χρονικός ορίζοντας είναι πολύ μεγάλος, είναι δυνατό να χρησιμοποιηθεί ένα όχημα για περισσότερες από μία διαδρομές.



Εικόνα 2.3.2.1 Με πολλαπλές επιστροφές
στην αποθήκη

2.5.4 Το ανοιχτό πρόβλημα δρομολόγησης οχημάτων

Το πρόβλημα αυτό ονομάζεται ανοιχτό επειδή τα οχήματα δεν επιστρέφουν στην αποθήκη μετά την εξυπηρέτηση των πελατών. Η παραλλαγή αυτή συμβαίνει όταν τα οχήματα της επιχείρησης δεν φτάνουν για να καλύψουν την ζήτηση και αναγκάζεται έτσι η εταιρεία να νοικιάσει οχήματα από εξωτερικό συνεργάτη. Οπότε μετά την εξυπηρέτηση των πελατών τα οχήματα επιστρέφουν στην αποθήκη της εταιρείας από όπου νοικιάστηκαν.

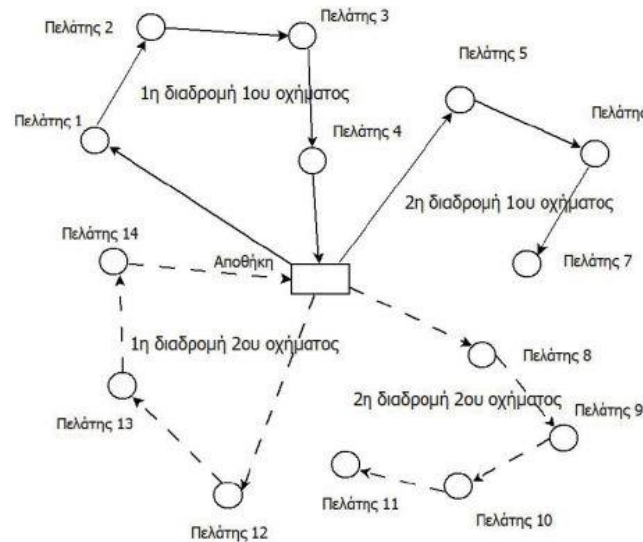


Εικόνα 2.3.3.1 Το ανοιχτό πρόβλημα

2.5.5 Το ανοιχτό κλειστό πρόβλημα δρομολόγησης

Εδώ έχουμε μια παραλλαγή του ανοιχτού προβλήματος δρομολόγησης οχημάτων. Τώρα υπολογίζεται η πιθανότητα το όχημα να έχει να χρόνο επιστρέψει στην αποθήκη, να ξεφορτώσει και στη συνέχεια να γυρίσει στον συνεργάτη από όπου έγινε η ενοικίαση.

Υπάρχει μία ακόμη περίπτωση εδώ, μία “παραλλαγή της παραλλαγής”, στην περίπτωση αυτή υπάρχουν τόσο ιδιόκτητα όσο και νοικιασμένα οχήματα. Το πρόβλημα μπορεί να λυθεί ως συνδιασμός του ανοιχτού προβλήματος δρομολόγησης για τα νοικιασμένα οχήματα και του περιορισμένου προβλήματος δρομολόγησης για τα ιδιόκτητα οχήματα.



Εικόνα 2.3.4.1 Ανοιχτό Κλειστό πρόβλημα δρομολόγησης οχημάτων

2.5.6 Το πρόβλημα δρομολόγησης οχημάτων με μόνο περιορισμό τον χρόνο μετάβασης

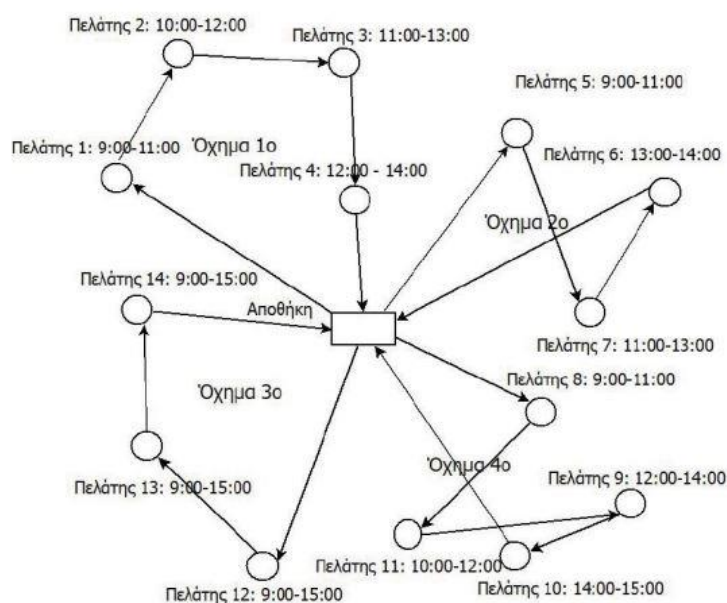
Στο πρόβλημα αυτό έχουμε σαν μοναδικό περιορισμό τον χρόνο μετάβασης, από την αποθήκη στον πελάτη ή από έναν πελάτη σε έναν άλλο, και όχι την ζήτηση. Στην πράξη αυτά τα προβλήματα μοντελοποιούν την εξυπηρέτηση από τεχνικούς, όπως ηλεκτρολόγους, υδραυλικούς κτλ.

2.5.7 Το πρόβλημα δρομολόγησης οχημάτων με περιορισμούς

χωρητικότητα και χρονικά παράθυρα

Το εν λόγω πρόβλημα αποτελεί ένα συνδυασμό δύο προβλημάτων, το περιορισμένης χωρητικότητας και το πρόβλημα δρομολόγησης με μόνο περιορισμό τον χρόνο μετάβασης. Σε αυτή την περίπτωση οι πελάτες έχουν συγκεκριμένα χρονικά περιθώρια στα οποία μπορούν να εξυπηρετηθούν. Ουσιαστικά προστίθεται ακόμη ένας περιορισμός, και τον θέτει είτε η επιχείρηση είτε ο πελάτης. Η χρονική περίοδος στην οποία μπορεί να εξυπηρετηθεί ο κάθε πελάτης ονομάζεται χρονικό παράθυρο. Τα πρόσθετα δεδομένα που απαιτούνται για τη λύση είναι ο χρόνος μετάβασης από κόμβο σε κόμβο και ο χρόνος εξυπηρέτησης για κάθε πελάτη.

Υπάρχουν δύο τύποι χρονικών παραθύρων, χαλαρά και σκληρά. Στη χαλαρή περίπτωση, όταν το όχημα φτάσει στον πελάτη κάποια στιγμή έξω από το παράθυρο του χρόνου, έχει τη δυνατότητα να τον εξυπηρετήσει. Σε άλλη περίπτωση, σε δύσκολες στιγμές, δεν γίνεται δεκτό σέρβις εκτός υπηρεσίας και το όχημα πρέπει να περιμένει την κατάλληλη στιγμή για να εξυπηρετήσει τον πελάτη.



Εικόνα 2.3.6 Με περιορισμούς χωρητικότητας και χρονικά παράθυρα

2.5.8 Το πρόβλημα δρομολόγησης οχημάτων με ύπαρξη πολλαπλών

αποθηκών

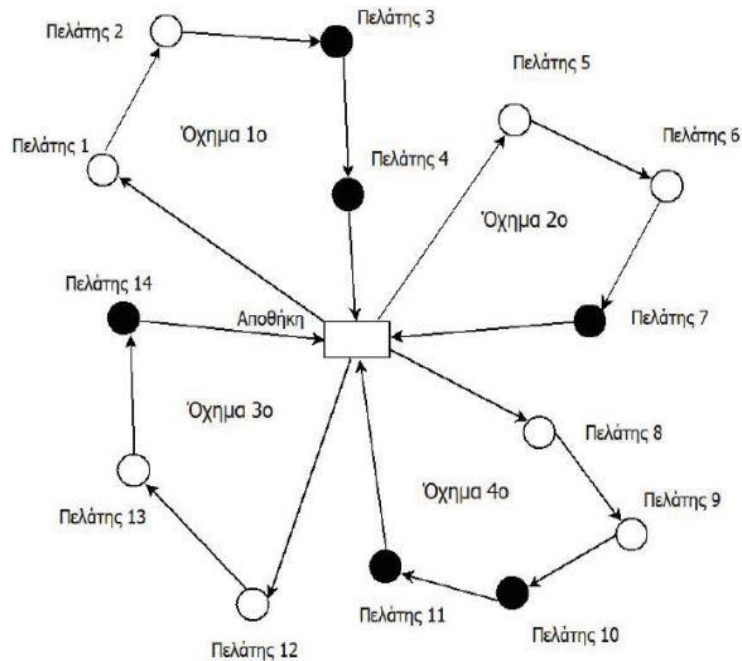
Το πρόβλημα αυτό αφορά εταιρείες που διαθέτουν πολλαπλές αποθήκες προκειμένου να εξυπηρετήσουν πελάτες. Υπάρχουν δύο τρόποι επίλυσης του προβλήματος. Στην πρώτη απλούστερη περίπτωση, κάθε αποθήκη έχει συγκεκριμένο αριθμό πελατών και οχημάτων, επομένως το πρόβλημα χωρίζεται σε πολλά μικρά προβλήματα δρομολόγησης για μία αποθήκη. Στη δεύτερη περίπτωση, το όχημα αναχωρεί από μια αποθήκη, είτε καταλήγει σε άλλη αποθήκη, είτε σταματά για καύσιμα σε άλλη αποθήκη και μετά συνεχίζει το δρόμο του.

Μπορεί επίσης να θεωρηθεί ως πρόβλημα ομαδοποίησης, καθώς στοχεύει στην εύρεση της διαδρομής των οχημάτων που ανήκουν σε μια αποθήκη. Σε αυτήν την ομαδοποίηση, οι πελάτες αρχικά τοποθετούνται σε αποθήκες και στη συνέχεια δημιουργούνται ταξίδια για κάθε αποθήκη και κάθε όχημα.

2.5.9 Το πρόβλημα δρομολόγησης οχημάτων με δύο είδη πελατών κατά την διάρκεια της διαδρομής

Στο πρόβλημα αυτό οι πελάτες χωρίζονται σε δύο υποσύνολα. Πρώτον, ο πελάτης ζητά να διανείμει το προϊόν και στη δεύτερη περίπτωση, ο πελάτης ζητά να συλλέξει μια συγκεκριμένη ποσότητα προϊόντος από αυτόν. Το πρόβλημα είναι να σχεδιάσουμε ένα σύνολο διαδρομών για ένα δεδομένο σύνολο οχημάτων που πρέπει να επισκεφτούν όλοι οι πελάτες. Ακολουθούν ορισμένοι επιπλέον περιορισμοί, οι οποίοι είναι:

- Εξυπηρετήστε πελάτες που χρειάζονται πρώτα διανομή και μετά πελάτες που χρειάζονται διαλογισμό.
- Κάθε πελάτης προέρχεται από έναν κύκλο.
- Μόνο οι πελάτες διαλογισμού επιτρέπεται να χρησιμοποιούν τη διαδρομή.

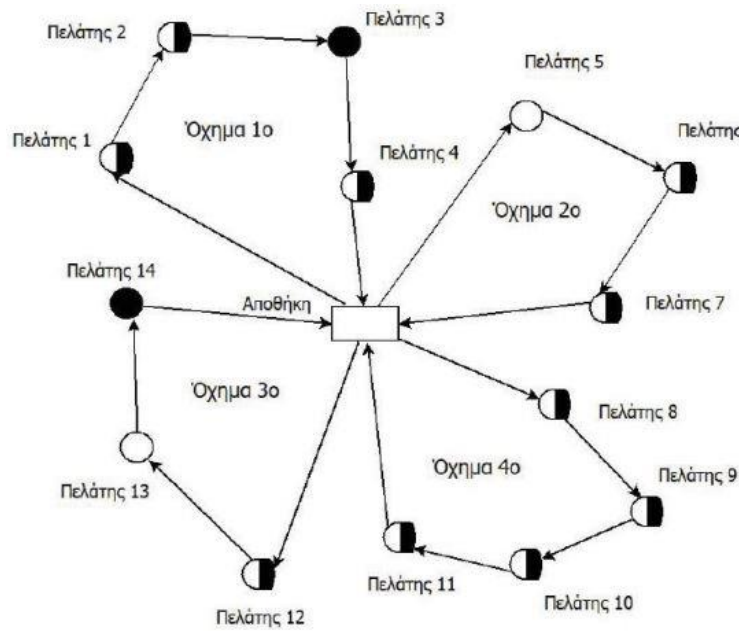


Εικόνα 2.3.8 Με δύο είδη πελατών κατά την διάρκεια της διαδρομής

Στην εικόνα εδώ, άσπρο χρώμα έχουν οι πελάτες που απαιτούν από το όχημα να τους παραδώσει μία ποσότητα προϊόντος και μαύρο εκείνοι που απαιτούν από το όχημα να παραλάβει από αυτούς μία ποσότητα προϊόντος.

2.5.10 Πρόβλημα δρομολόγησης οχημάτων με διανομή και παραλαβή προϊόντων κατά την διάρκεια της διαδρομής

Σε αυτή την περίπτωση οι πελάτες είναι πιο απαιτητικοί. Η διαφορά του από το προηγούμενο πρόβλημα είναι ότι ένας πελάτης μπορεί να χρειαστεί και διανομή και παραλαβή ενός προϊόντος, κάτι που προηγουμένως απαγορευόταν από τους περιορισμούς. Στην προηγούμενη περίπτωση ο πελάτης είχε είτε την μία δυνατότητα είτε την άλλη. Ένας βασικός περιορισμός εδώ είναι ότι τα προϊόντα διανέμονται σε κάθε πελάτη πριν από την παραλαβή τους, ώστε να μην υπερβαίνουν τα όρια χωρητικότητας του οχήματος. Επομένως, ο πελάτης συνδέεται με δύο ποσότητες, δηλαδή τη ζήτηση για το προϊόν που θα διανεμηθεί και θα παραληφθεί, αντίστοιχα.



Εικόνα 2.3.8 Με διανομή και παραλαβή
προϊόντων κατά την διάρκεια της διαδρομής

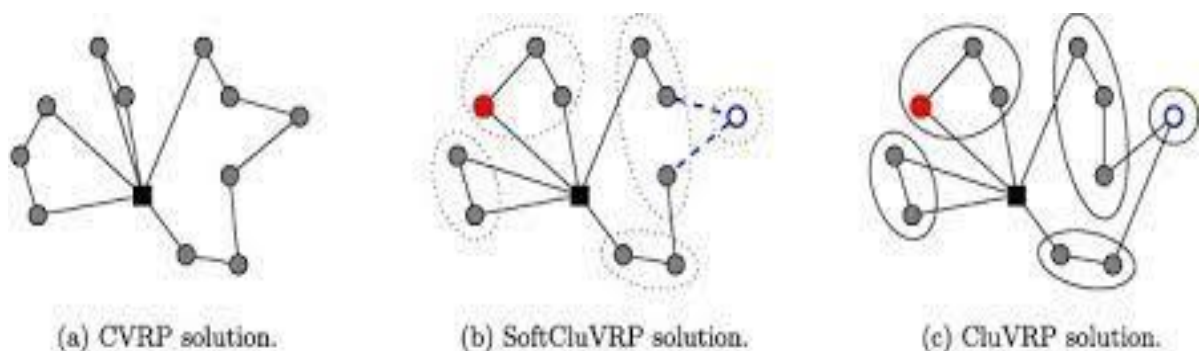
2.5.11 Στοχαστικό πρόβλημα δρομολόγησης

Στην κατηγορία αυτή, δεδομένα προβλημάτων όπως ο αριθμός των πελατών, οι ανάγκες τους και οι χρόνοι ταξιδιού μεταξύ των κόμβων μοντελοποιούνται ως τυχαίες διεργασίες. Αυτό σημαίνει ότι τα δεδομένα ορίζονται ως στοχαστικές μεταβλητές, δηλαδή μεταβλητές που μεταβάλλονται στον χρόνο, οπότε δεν είναι γνωστές οι τιμές τους εκ των προτέρων αλλά είναι γνωστές οι κατανομές τους. Για παράδειγμα, σε μια διαδρομή μπορεί να γίνει αλλαγή στην τοποθεσία παράδοσης ή στα χρονικά παράθυρα λόγω κάποιου απροόπτου θέματος του πελάτη. Επίσης οι πληροφορίες για την ζήτηση ουσιαστικά είναι εκτιμήσεις που γίνονται σύμφωνα με προηγούμενα δεδομένα, άρα και οι πληροφορίες αυτές είναι αναμφίβολες.

2.6 Το ομαδοποιημένο πρόβλημα δρομολόγησης

Πάνω σε αυτήν την κατηγορία προβλημάτων δρομολόγησης βασίζεται η διπλωματική αυτή. Πιο συγκεκριμένα έχουμε μια παραλλαγή του προβλήματος δρομολόγησης χωρητικότητας οχημάτων (CVRP), που αναλύθηκε προηγουμένως, και ονομάζεται Clustered VRP (CluVRP). Στο πρόβλημα αυτό, οι πελάτες χωρίζονται σε προκαθορισμένες ομάδες ή clusters. Οι πελάτες που ανήκουν σε μια από αυτές τις ομάδες θα πρέπει να επισκέπτονται όλοι από το ίδιο όχημα πριν φύγει από την ομάδα. Η μέθοδος της ομαδοποίησης στο πρόβλημα δρομολόγησης οχημάτων είναι ευρέως γνωστή λόγω των οικονομικών της επιπτώσεων και της μειωμένης πολυπλοκότητάς της στη μοντελοποίηση και την επίλυση μεγάλου όγκου προβλημάτων του πραγματικού κόσμου.

Υπάρχουν δύο εκδοχές του ομαδοποιημένου προβλήματος δρομολόγησης, τα προβλήματα με ισχυρούς περιορισμούς ομάδων (CluVRP) και με αδύναμους περιορισμούς ομάδων (SoftCluVRP). Στην πρώτη εκδοχή, όλοι οι πελάτες πρέπει να επισκέπτονται από το ίδιο όχημα χωρίς κάποια διακοπή. Τα οχήματα δεν επιτρέπεται να εισέρχονται και να εξέρχονται από τις ομάδες πολλές φορές ενώ εξυπηρετούν τους πελάτες. Στην περίπτωση όμως με τους αδύναμους περιορισμούς, οι πελάτες που ανήκουν στην ίδια ομάδα επισκέπτονται από το ίδιο όχημα, αλλά τώρα τα οχήματα μπορούν να εισέρχονται και να εξέρχονται από τις ομάδες παραπάνω από μία φορές κατά την διάρκεια της διαδρομής τους^{ΑΡΘΡΟ}.



Εικόνα 2.3.11 Το ομαδοποιημένο πρόβλημα δρομολόγησης
και οι δύο εκδοχές του.

2.7 Ορίσματα του ομαδοποιημένου προβλήματος δρομολόγησης

2.7.1 Μεταβλητές

Όνομα μεταβλητής	Ορισμός μεταβλητής
n	Ο συνολικός αριθμός πελατών.
c	Ο συνολικός αριθμός συστάδων (clusters).
0	Η αποθήκη.
n_i	Ο αριθμός των πελατών της i συστάδας.
m	Μεμονωμένο όχημα.
M	Συνολικός αριθμός διαθέσιμων οχημάτων στο δίκτυο.
r	Μεμονωμένο σύμπλεγμα (cluster). $r \in R$
R	Ομάδα των συστάδων.
d_r	Ζήτηση συμπλέγματος (συγκεντρωτική όλων των πελατών του cluster). $d_r > 0$
tc_{ij}	Το κόστος της διαδρομής από το i στο j . $tc_{ij} > 0$
Q	Μέγιστη χωρητικότητα κάθε οχήματος. $Q > 0$
C_r	Ομάδα πελατών μέσα σε ένα cluster.
V	Σύνολο κόμβων.
S	Οποιοδήποτε υποσύνολο από κόμβους πελατών. $\{1, 2, \dots, n\}$
E	Το σύνολο των τόξων που συνδέουν κάθε ζεύγος κόμβων (i,j) του συνόλου V.
$\delta^+(S)$	Σύνολο ακμών (i,j), όπου $(i \in S)$ και $(j \in V/S)$
$\delta^-(S)$	Σύνολο ακμών (i,j), όπου $(j \in S)$ και $(i \in V/S)$

Πίνακας 2.5.1 Οι μεταβλητές του ομαδοποιημένου προβλήματος
δρομολόγησης οχημάτων⁴

2.7.2 Περιορισμοί

Μαθηματική διατύπωση περιορισμών	Ανάλυση περιορισμών
$\sum_{m=1}^M y_{im} = 1 \quad \forall i \in \{1, 2, \dots, n\}$	Εξασφάλιση μίας ακριβώς επίσκεψης σε κάθε πελάτη.
$\sum_{m=1}^M y_{0m} \leq M$	Διασφάλιση του αριθμού των οχημάτων που χρησιμοποιούνται να μην υπερβαίνει τον αριθμό των διαθέσιμων οχημάτων.
$y_{0m} \geq y_{im} \\ \forall m \in \{1, 2, \dots, M\}, \forall i \in \{1, 2, \dots, n\}$	Κάθε όχημα που είναι σε κάποια διαδρομή πρέπει να επισκέπτεται την αποθήκη. Εάν κάποιο όχημα m δεν επισκεφθεί την αποθήκη τότε δεν θα μπορεί να επισκεφτεί κανέναν πελάτη.
$\sum_{j=1}^n x_{ijm} = \sum_{j=1}^n x_{jim} = y_{im} \\ \forall m \in \{1, 2, \dots, M\}, \forall i \in \{0, 1, 2, \dots, n\}$	Διασφάλιση ότι το όχημα που επισκέπτεται έναν πελάτη είναι και το ίδιο που αναχωρεί από τον συγκεκριμένο πελάτη.
$\sum_{i=0}^n d_i y_{im} \leq Q \quad \forall i \in \{0, 1, 2, \dots, M\}$	Ο περιορισμός αυτός δηλώνει ότι η μέγιστη χωρητικότητα φόρτωσης των οχημάτων είναι ικανοποιημένη.
$\sum_{i \in S} \sum_{j \in V/S} x_{ijm} \geq y_{hm} \\ \forall S \subseteq \{1, 2, \dots, n\}, h \in S, m \in \{0, 1, \dots, M\}$	Λειτουργεί σαν περιορισμός εξάλειψης της δετερεύουσας περιήγησης.
$\sum_{(i,j) \in \delta^+(C_r)} \sum_{m=1}^M x_{ijm} = \sum_{(i,j) \in \delta^-(C_r)} \sum_{m=1}^M x_{ijm} = 1 \\ \forall r \in R$	Διασφάλιση ότι το κάθε cluster επισκέπτεται ακριβώς μία φορά από ένα και μόνο όχημα.
$\sum_{i=1}^n d_i y_{im} \geq \sum_{i=1}^n d_i y_{im+1} \\ \forall m \in \{1, 2, \dots, M-1\}$	Εξασφάλιση μερικής συμμετρίας.

$x_{ijm} \in \{0,1\} \quad \forall (i,j) \in E,$ $\forall m \in \{1,2,...M\}$	Διαδική μεταβλητή απόφασης. Όταν παίρνει την τιμή 1 τότε το όχημα m ταξιδεύει από τον πελάτη i στον j. Αλλιώς παίρνει την τιμή 0.
$y_{im} \in \{0,1\} \quad \forall i \in \{0,1,2,...n\},$ $\forall m \in \{1,2,...M\}$	Διαδική μεταβλητή απόφασης. Όταν παίρνει την τιμή 1 τότε ο πελάτης i εξυπηρετείται από το όχημα m. Αλλιώς παίρνει την τιμή 0.

Πίνακας 2.5.2 Ανάλυση περιορισμών του ομαδοποιημένου προβλήματος⁴

2.7.3 Μοντελοποίηση του προβλήματος

Αντικειμενική συνάρτηση προς ελαχιστοποίηση⁴ :

$$\min \sum_{(i,j) \in E} \sum_{m=1}^M t_{Cij} x_{ijm}$$

Η αντικειμενική συνάρτηση έχει ως στόχο την ελαχιστοποίηση του κόστους ταξιδιού που διανύει ένα όχημα μέχρις ότου εξυπηρετήσει όλους τους πελάτες του. Και για να επιτευχθεί αυτή η ελαχιστοποίηση δεν θα πρέπει να παραβιάζεται κανένας από τους δέκα εξής περιορισμούς.

Υπό :

$$\sum_{m=1}^M y_{im} = 1 \quad \forall i \in \{1,2, \dots, n\}$$

$$\sum_{m=1}^M y_{0m} \leq M$$

$$\sum_{j=1}^n x_{ijm} = \sum_{j=1}^n x_{jim} = y_{im} \quad , \quad \forall m \in \{1,2,...,M\}, \forall i \in \{0,1,2,..., n\}$$

$$\sum_{i=0}^n d_i y_{im} \leq Q \quad , \quad \forall i \in \{0,1,2,...,M\}$$

$$\sum_{i \in S} \sum_{j \in V/S} x_{ijm} \geq y_{hm} \quad , \quad \forall S \subseteq \{1,2,...,n\}, h \in S, m \in \{0,1,...,M\}$$

$$\sum_{(i,j) \in \delta^+(C_r)} \sum_{m=1}^M x_{ijm} = \sum_{(i,j) \in \delta^-(C_r)} \sum_{m=1}^M x_{ijm} = 1 \quad , \quad \forall r \in R$$

$$\sum_{i=1}^n d_i y_{im} \geq \sum_{i=1}^n d_i y_{i,m+1} \quad , \quad \forall m \in \{1,2,...,M-1\}$$

$$x_{ijm} \in \{0,1\} \quad , \quad \forall (i,j) \in E, \forall m \in \{1,2,...,M\}$$

$$y_{im} \in \{0,1\} \quad , \quad \forall i \in \{0,1,2,...,n\}, \forall m \in \{1,2,...,M\}$$

$$y_{0m} \geq y_{im} \quad , \quad \forall m \in \{1,2,...,M\}, \forall i \in \{1,2,...,n\}$$

Κεφάλαιο 3: Αλγόριθμοι επίλυσης προβλημάτων δρομολόγησης

3.1 Εισαγωγή

Αυτό το κεφάλαιο συνοψίζει τις κατηγορίες αλγορίθμων που χρησιμοποιούνται για την επίλυση προβλημάτων δρομολόγησης οχημάτων και τους αλγόριθμους που εμπίπτουν σε αυτές τις κατηγορίες. Επειδή γενικά οι αλγόριθμοι αυτοί είναι πάρα πολλοί, θα γίνει αναφορά στους κυριότερους από αυτούς. Και στην συνέχεια, γίνεται μία ανάλυση των αλγορίθμων που χρειάστηκαν για την λύση του ομαδοποιημένου προβλήματος δρομολόγησης της συγκεκριμένης διπλωματικής.

3.2 Ευρετικοί αλγόριθμοι

Η εύρεση της βέλτιστης λύσης γίνεται όλο και πιο δύσκολη καθώς το μέγεθος του προβλήματος αυξάνεται και η προσπάθεια να βρεθεί το ολικό ελάχιστο ενός προβλήματος σε ένα σύντομο χρονικό διάστημα είναι σχεδόν αδύνατο. Το γεγονός αυτό οδήγησε στην δημιουργία μιας ομάδας αλγορίθμων που έχει ως σκοπό την εύρεση ενός τοπικού ελάχιστου, και όχι του ολικού, το οποίο, για να γίνει αποδεκτό, θα πρέπει να είναι εντός των ορίων που θέτει ο ίδιος ο αλγόριθμος. Τα όρια αυτά μπορεί να είναι η αποδοτικότητα της λύσης καθώς και ο χρόνος και η ευκολία απόκτησής της. Οι ευρετικοί αλγόριθμοι χωρίζονται σε τρεις υποκατηγορίες ανάλογα με την λογική στην οποία βασίζονται, αυτές είναι²:

- Οι Αλγόριθμοι απληστίας (Greedy algorithms)
- Οι Προσεγγιστικοί αλγόριθμοι (Approximation algorithms)
- Οι Αλγόριθμοι τοπικής αναζήτησης (Local search algorithms)

3.2.1 Αλγόριθμοι απληστίας

Οι αλγόριθμοι απληστίας έχουν σκοπό να δημιουργήσουν μία αρχική εφικτή λύση. Με τα δεδομένα του προβλήματος σαν είσοδο, βρίσκει μια λύση η οποία δεν είναι η βέλτιστη και απέχει πολύ από αυτήν αλλά στην συνέχεια η λύση αυτή χρησιμοποιείται σαν είσοδος σε άλλους αλγορίθμους με σκοπό να την βελτιστοποιήσουν.

Η λογική των αλγορίθμων απληστίας είναι απλή, αρχικά ο αλγόριθμος βρίσκει μια εν μέρει αδύνατη λύση, στη συνέχεια προσδιορίζει μία ή περισσότερες μεταβλητές σε κάθε βήμα μέχρι να βρεθεί μια πιθανή λύση. Αυτοί οι αλγόριθμοι εμπίπτουν στην κατηγορία των μωωπικών αλγορίθμων, δηλαδή επιλέγουν την καλύτερη εναλλακτική από τις διαθέσιμες επιλογές σε κάθε βήμα. Το πλεονέκτημα της μεθόδου είναι ότι αποδίδει μία λύση σε μικρό χρονικό διάστημα, αλλά το μειονέκτημα της είναι ότι η λύση αυτή απέχει από το ολικό ελάχιστο.

Οι αλγόριθμοι απληστίας βασίζονται σε διάφορες στρατηγικές επίλυσης προβλημάτων. Αυτές οι στρατηγικές χωρίζονται στις ακόλουθες ομάδες²:

1) Ομαδοποίηση πρώτα-δρομολόγηση έπειτα

Αρχικά, ομαδοποιούμε τους κόμβους με βάση την απόσταση και στη συνέχεια δημιουργούμε διαδρομές με το χαμηλότερο δυνατό κόστος..

2) Δρομολόγηση πρώτα- ομαδοποίηση έπειτα

Η αρχή αυτής της λογικής είναι να δημιουργηθεί ένας κύκλος που θα περιέχει όλους τους κόμβους του προβλήματος. Στο επόμενο βήμα, αυτή η διαδρομή χωρίζεται σε μεμονωμένες μικρότερες διαδρομές, με αποτέλεσμα ένα σύνολο εφικτών κύκλων με το χαμηλότερο κόστος.

3) Εξοικονομήσεις/καταχώρηση

Η στρατηγική αυτή επιτυγχάνεται με την σύγκριση δύο διαδρομών, οι οποίες μπορεί να είναι ή και να μην είναι εφικτές, και γίνεται καταχώρηση εκείνης με το

μικρότερο κόστος. Ο αλγόριθμος τερματίζει επιστρέφοντας μία εφικτή λύση και γενικότερα έχει ως στόχο την βελτιστοποίηση του κόστους.

4) Βελτίωση ή ανταλλαγή

Αντίθετα με προηγουμένως, χρησιμοποιούνται μόνο εφικτές λύσεις και συγκρίνονται ως προς το κόστος. Στο τελικό στάδιο παραμένει η εφικτή λύση με το ελάχιστο κόστος.

5) Προσέγγιση μαθηματικού προγραμματισμού

Γίνεται χρήση μαθηματικών μοντέλων και μαθηματικού προγραμματισμού μέσω αλγορίθμων που βασίζονται στην λογική αυτή.

6) Αλληλοεπιδρών βελτιστοποίηση

Στην συγκεκριμένη κατηγορία εισέρχονται έμπειροι αποφασίζοντες οι οποίοι συμμετέχουν στην κατασκευή της λύσης εκφέροντας την άποψη τους, βασιζόμενοι στην εμπειρία και τις γνώσεις τους.

7) Ακριβής διαδικασία

Η ακριβής διαδικασία είναι να λυθεί το πρόβλημα δρομολόγησης αναζητώντας όλες τις λύσεις. Οι διαδικασίες αυτές περιλαμβάνουν εξειδικευμένους αλγορίθμους όπως ο αλγόριθμος τομής επιπέδων και ο αλγόριθμος διακλάδωσης και οριοθέτησης.

Οι πιο σημαντικοί αλγόριθμοι απληστίας είναι²:

- Ο αλγόριθμος του πλησιέστερου γείτονα (Nearest Neighborhood Algorithm).
- Ο αλγόριθμος της διαδικασίας εισαγωγής κόμβων (Nearest Insertion Algorithm).
- Ο αλγόριθμος εξοικονομήσεων των Clarke and Write.
- Ο αλγόριθμος εγγύτερης συγχώνευσης.
- Ο αλγόριθμος εγγύτερης πρόσθεσης.

3.2.2 Προσεγγιστικοί αλγόριθμοι

Οι αλγόριθμοι προσέγγισης προσπαθούν να λύσουν το πρόβλημα χρησιμοποιώντας τις παραπάνω πληροφορίες.. Με λίγα λόγια οι προσεγγιστικοί αλγόριθμοι υπολογίζουν σε πολυωνμικό χρόνο μία λύση η οποία είναι σίγουρα κοντά στην βέλτιστη. Συνήθως χρησιμοποιούνται είτε όταν δεν μπορεί να βρεθεί λύση στο πρόβλημα με την χρήση κάποιας άλλης μεθόδου είτε όταν η λύση των υπόλοιπων μεθόδων δεν είναι αρκετά αποδοτική και χρονοβόρα. Τέλος με τους αλγορίθμους αυτούς εισάγεται και η έννοια του λόγου προσέγγισης. Ο λόγος αυτός ορίζεται ως το κόστος της λύσης του προσεγγιστικού αλγορίθμου προς το κόστος της βέλτιστης δυνατής λύσης.

3.2.3 Αλγόριθμοι τοπικής αναζήτησης

Οι αλγόριθμοι τοπικής αναζήτησης αποτελούν μια πολύ απλή μέθοδο καθώς χρησιμοποιούν την μέθοδο, δοκιμή και σφάλμα, και τα αποτελέσματα τους είναι πολύ ικανοποιητικά. Επίσης χρησιμοποιούνται διαχρονικά λόγω της απλότητας και της αποτελεσματικότητάς τους. Οι αλγόριθμοι αυτοί ουσιαστικά βελτιστοποιούν την ήδη υπάρχουσα λύση, που προκύπτει είτε από την χρήση αλγορίθμων απληστίας είτε από προσεγγιστικών αλγορίθμων.

Η λογική της τοπικής αναζήτησης είναι πολύ απλή. Αρχικά επιλέγεται μια γειτονιά λύσεων από ένα ευρύτερο σύνολο και υλοποιείται μία επαναληπτική μέθοδος αναζήτησης για να βρεθεί μια εφικτή λύση. Κάθε επανάληψη βρίσκει μια διαφορετική λύση από την πρώτη και εφόσον γίνει η σύγκριση αυτών των δύο επιλέγεται η λύση με το ελάχιστο κόστος. Η επαναληπτική διαδικασία σταματάει με κάποιο κριτήριο τερματισμού. Ένα τέτοιο κριτήριο μπορεί να είναι ένας αριθμός επαναλήψεων όπου η λύση δεν βελτιώνεται άλλο.

Οι κυριότεροι αλγόριθμοι τοπικής αναζήτησης είναι:

- 2-opt
- 3-opt
- 1-0 επανατοποθέτηση (relocate)
- swap
- Or-opt
- 1-1 exchange

3.3 Μεθευρετικοί Αλγόριθμοι

Το πιο βασικό μειονέκτημα των ευρετικών αλγορίθμων είναι ότι βασίζονται πολύ στην αρχική λύση που τους δίνεται. Πιο συγκεκριμένα, αν αυτή η αρχική λύση είναι κοντά σε τοπικό ελάχιστο θα <<παγιδευτεί>> σε αυτό, με αποτέλεσμα να θεωρηθεί αυτή η λύση ως βέλτιστη και να μην βρεθεί το ολικό ελάχιστο του προβλήματος. Για το σκοπό αυτό, υπάρχουν μεθευρετικοί μέθοδοι που μπορούν να αποφύγουν τα τοπικά ελάχιστα χρησιμοποιώντας διάφορες τεχνικές.

Οι μεθευρετικοί αλγόριθμοι χωρίζονται σε δύο κατηγορίες, αυτούς που χρησιμοποιούν μια αρχική λύση και αυτούς που χρησιμοποιούν μεγάλο αριθμό λύσεων. Και στις δύο περιπτώσεις ο σκοπός της αναζήτησης είναι η εύρεση καλύτερων αποτελεσμάτων σε ολόκληρο το φάσμα λύσεων. Αρχικά, όσοι χρησιμοποιούν μόνο μία λύση μπορούν να κάνουν καλύτερη χρήση του ιστότοπου και να αυξήσουν την προσπάθεια αναζήτησής τους. Ενώ οι αλγόριθμοι που χρησιμοποιούν πληθυσμό λύσεων, έχουν μεγαλύτερη δυνατότητα διάχυσης της πληροφορίας γύρω από πολλά σημεία στο χώρο λύσεων και επίσης έχουν το πλεονέκτημα της εξερεύνησης. Οι αλγόριθμοι αυτοί λειτουργούν, δημιουργώντας πρώτα μία αρχική λύση (ή πλήθος αρχικών λύσεων) και έπειτα μέσω τοπικής αναζήτησης προσπαθούν να βελτιώσουν την λύση αυτή. Να μην ξεχάσουμε βέβαια ότι υπάρχουν και υβριδικές μορφές αυτών των δύο κατηγοριών για καλύτερο αποτέλεσμα.

Οι αλγόριθμοι που χρησιμοποιούν μία λύση χωρίζονται σε 4 κατηγορίες. Αυτές είναι²:

- i. Επαναληπτικές διαδικασίες που αρχίζουν από διαφορετικές αρχικές λύσεις. Ορισμένοι αλγόριθμοι που λειτουργούν με αυτόν τον τρόπο είναι οι αλγόριθμοι επαναληπτικής τοπικής αναζήτησης και οι αλγόριθμοι πολυεναρκτήριας τοπικής αναζήτησης.
- ii. Αλγόριθμοι που δέχονται γειτονικές κινήσεις που δεν βελτιώνουν τη λύση. Στις μεθόδους αυτές ένα βήμα που δεν βελτιώνει την λύση μπορεί να γίνει αποδεκτό εαν ικανοποιεί μερικούς περιορισμούς που θέτονται. Αυτό έχει σαν αποτέλεσμα, ο αλγόριθμος να ξεφεύγει από το τοπικό ελάχιστο και να οδηγείται σε κάποιο

επόμενο τοπικό ελάχιστο που μπορεί να είναι καλύτερο από το προηγούμενο ή βέλτιστη λύση. Οι πιο γνωστοί αλγόριθμοι της κατηγορίας αυτής είναι η περιορισμένη αναζήτηση (tabu search) και προσομοιωμένη απόπτηση.

- iii. Αλγόριθμοι που αλλάζουν την γειτονιά αναζήτησης. Σε αυτή την κατηγορία, οι αλγόριθμοι που επιμένουν στα τοπικά ελάχιστα αλλάζουν τον αλγόριθμο που χρησιμοποιούν και βρίσκουν λύσεις σε άλλες γειτονιές στο χώρο λύσης. Οι πιο δημοφιλείς αλγόριθμοι σε αυτήν την κατηγορία είναι ο αλγόριθμος επέκτασης γειτονιάς αναζήτησης και ο αλγόριθμος αναζήτησης μεταβλητής γειτονιάς (VNS).
- iv. Αλγόριθμοι που τροποποιούν την αντικειμενική συνάρτηση ή κάποια δεδομένα προβλημάτων. Σε αυτή την περίπτωση, ο αλγόριθμος τροποποιεί το πρόβλημα αλλάζοντας την αντικειμενική συνάρτηση ή τους περιορισμούς του προβλήματος. Ο πιο τυπικός αλγόριθμος σε αυτήν την κατηγορία είναι η καθοδηγούμενη τοπική αναζήτηση.

3.4 Εξελικτικοί και εμπνευσμένοι από την φύση αλγόριθμοι

3.4.1 Εξελικτικοί αλγόριθμοι

Οι αλγόριθμοι αυτοί έχουν ως βάση για τον υπολογισμό λύσεων, τις αρχές της φυσικής επιλογής και της επιβίωσης του ικανότερου. Πιο ειδικά, επιλέγεται από έναν πληθυσμό λύσεων η καλύτερη δυνατή (επιβίωση του ικανότερου) και στην συνέχεια συνδυάζεται με τις αλλαγές που έχουν γίνει στην αρχική λύση (μεταλλάξεις) με στόχο την εύρεση της βέλτιστης λύσης. Με άλλα λόγια οι αλγόριθμοι της κατηγορίας αυτής, αντιγράφουν τις φυσικές διαδικασίες που πραγματοποιούνται στην φύση και τις χρησιμοποιούν για να επιλύσουν αποτελεσματικά προβλήματα βελτιστοποίησης και αναζήτησης.

Οι σημαντικότεροι εξελικτικοί αλγόριθμοι είναι:

- Οι γενετικοί αλγόριθμοι (Genetic Algorithms)
- Οι γενετικοί αλγόριθμοι πολλαπλών πληθυσμών-νησιών (Island genetic algorithms)
- Οι μιμητικοί αλγόριθμοι (Memetic algorithms)

- Οι αλγόριθμοι της διαφορικής εξέλιξης (Differential evolution)

3.4.2 Αλγόριθμοι εμπνευσμένοι από την φύση

Αλγόριθμοι που εμπίπτουν σε αυτή την κατηγορία έχουν προκύψει μετά από μοντελοποίηση της συμπεριφοράς ορισμένων οργανισμών και ορισμένων φυσικών φαινομένων. Υπάρχουν πολλά είδη τέτοιων αλγορίθμων και είναι διαφορετικοί μεταξύ τους. Σύμφωνα με τους Fister Jr. και Xin-She Yang το 2013, οι αλγόριθμοι εμπνευσμένοι από τη φύση μπορούν να χωριστούν σε τέσσερις κατηγορίες¹, οι οποίες είναι:

- Αλγόριθμοι σμήνους (Swarm intelligence) : Αλγόριθμοι που προσομοιώνουν την νοημοσύνη σμήνους.
- Βιο-εμπνευσμένοι αλγόριθμοι (Bio-inspired algorithms) : Αλγόριθμοι βασισμένοι σε βιολογικές διαδικασίες.
- Φυσικο-χημικοί αλγόριθμοι (Physical Phenomena and laws of science) : Αλγόριθμοι εμπνευσμένοι από τους νόμους φυσικής και χημείας.
- Άλλοι : Αλγόριθμοι που δεν ανήκουν σε κάποια από τις προηγούμενες κατηγορίες.

Οι πιο διαδεδομένοι αλγόριθμοι είναι:

- ❖ Ο αλγόριθμος βελτιστοποίησης αποικίας μυρμηγκιών (Ant colony optimization).
- ❖ Ο αλγόριθμος τεχνητής αποικίας μυρμηγκιών (Artificial bee colony optimization algorithm).
- ❖ Ο αλγόριθμος της πυγολαμπίδας (Fire-fly algorithm).

- ❖ Ο αλγόριθμος αναζήτησης κούκων (Cuckoo search algorithm).
- ❖ Ο αλγόριθμος βελτιστοποίησης ζευγαρώματος μελισσών (Honey bees mating optimization).
- ❖ Ο αλγόριθμος αναζήτησης νυχτερίδας (Bat algorithm).
- ❖ Ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων (Particle swarm optimization).

3.5 Ανάλυση των αλγορίθμων που χρησιμοποιήθηκαν για την επίλυση του ομαδοποιημένου προβλήματος δρομολόγησης οχημάτων

Για την επίλυση του ομαδοποιημένου προβλήματος δρομολόγησης οχημάτων υλοποιήθηκαν τέσσερις διαφορετικοί αλγόριθμοι. Αρχικά ο αλγόριθμος πλησιέστερου γείτονα κατασκευάζει τις αρχικές λύσεις. Έπειτα για την βελτιστοποίηση των αρχικών αυτών λύσεων χρησιμοποιήθηκε ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων και δύο αλγόριθμοι τοπικής αναζήτησης, ο 1-1 exchange και 2-opt. Στην συνέχεια, περιγράφεται αναλυτικά η λειτουργία των αλγορίθμων αυτών.

3.5.1 Ο αλγόριθμος πλησιέστερου γείτονα (nearest neighbor algorithm)

Στα περισσότερα προβλήματα δρομολόγησης που συναντάμε, είναι αναγκαία η εύρεση μίας αρχικής λύσης. Αυτή μπορεί σε ορισμένες περιπτώσεις να ικανοποιεί τις ανάγκες του εκάστοτε προβλήματος αλλά συνήθως χρησιμοποιείται σαν αφετηρία για την βελτιστοποίηση της λύσης που ακολουθεί.

Όπως περιγράφηκε και νωρίτερα, ο αλγόριθμος του πλησιέστερου γείτονα ανήκει στην κατηγορία των αλγορίθμων απληστίας. Η μεθοδολογία του αλγορίθμου είναι πολύ απλή. Αρχικά το όχημα ξεκινάει από την αποθήκη (αρχικός κόμβος) με δεδομένο την θέση των πελατών του, επισκέπτεται τον κοντινότερό του πελάτη, δηλαδή τον πελάτη με το μικρότερο κόστος σύμφωνα με το κόστος του προηγούμενου κόμβου. Στην συνέχεια

εξυπηρετεί τον πελάτη, τον διαγράφει από την λίστα και επαναλαμβάνει την ίδια διαδικασία έως ότου εξυπηρετηθούν όλοι οι πελάτες.

Τα βήματα της μεθόδου παρουσιάζονται στην συνέχεια²:

Βήμα 1^ο : Εκκίνηση από τον αρχικό κόμβο του συστήματος (κόμβος 1, $n=1$).

Βήμα 2^ο : Αναζήτηση και τοποθέτηση του πελάτη με το μικρότερο κόστος (κόμβος αναφοράς είναι ο αρχικός n) , σαν κόμβο νούμερο $n=n+1$.

Βήμα 3^ο : Επανάληψη του δεύτερου βήματος μέχρι να έχουν καταγραφεί όλοι οι πελάτες (κόμβοι) στην διαδρομή. (τελευταίος κόμβος πρέπει να είναι ο αρχικός (αποθήκη))

Επανα- λήψεις	Κόμβοι											Κόστος
1η	1											-
2η	1 8											11.66
3η	1 8 7											14.03
4η	1 8 7 6											16.49
5η	1 8 7 6 4											21.02
6η	1 8 7 6 4 5											20.39
7η	1 8 7 6 4 5 9											12.36
8η	1 8 7 6 4 5 9 10											12.04
9η	1 8 7 6 4 5 9 10 2											28.07
10η	1 8 7 6 4 5 9 10 2 3											15.29
11η	1 8 7 6 4 5 9 10 2 3 1											19.2

Εικόνα 3.5.1 Παράδειγμα απεικόνισης του πλησιέστερου γείτονα

3.5.2 Ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων (Particle Swarm Optimization)

Ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων είναι εμπνευσμένος από τις κοινωνικές συμπεριφορές και τους βιολογικούς μηχανισμούς κάποιων οργανισμών όπως κοπάδια ψαριών, αποικίες εντόμων και σμήνη πτηνών που κινούνται και ταξιδεύουν ομαδικά. Η αρχική ιδέα για να σχεδιαστεί ο αλγόριθμος ήταν η καταγραφή της κίνησης ενός σμήνους πουλιών, και να δοθεί απάντηση στο πώς τα πουλιά μπορούν να αλλάζουν ξαφνικά κατεύθυνση και να μην χαλάει ο σχηματισμός τους καθώς και το πώς μπορούν και ταξιδεύουν ως ένα σμήνος. Ένα ιδιαίτερο χαρακτηριστικό του εξελικτικού αυτού μηχανισμού είναι ότι, οι οργανισμοί του σμήνους ή αλλιώς τα σωματίδια (particles) έχουν την ιδιότητα να θυμούνται την καλύτερη θέση που έχουν επισκεφθεί και μπορούν να μεταδίδουν αυτή την πληροφορία και στο υπόλοιπο σμήνος. Αυτό σημαίνει ότι τα σωματίδια παίρνουν και ατομικές αποφάσεις αλλά και ομαδικές μιας και υπάρχει επικοινωνία μεταξύ τους.

Σκοπός του αλγορίθμου είναι η εύρεση της βέλτιστης λύσης με την χρήση του σμήνους σωματιδίων για την αναζήτηση μέσα στον χώρο του προβλήματος. Τα σωματίδια αυτά διαμορφώνονται σε διανύσματα και δηλώνουν θέσεις-λύσεις. Το κάθε ένα από τα σωματίδια αυτά καταλαμβάνει μία θέση που απεικονίζει μια πιθανή λύση στο πρόβλημα βελτιστοποίησης. Η ποιότητα κάθε λύσης αξιολογείται από μια συνάρτηση καταλληλότητας-προσαρμογής (fitness function $f(x_{ij})$), ώστε να βρεθεί η βέλτιστη λύση.

Ας δούμε λοιπόν την λειτουργία αυτού του αλγορίθμου. Αρχικά σε κάθε σωματίδιο του σμήνους δίνεται μια τυχαία θέση (x_{ij}) στον χώρο και αποκτά ακόμη μια τυχαία ταχύτητα (u_{ij}). Η θέση και η ταχύτητα είναι συναρτήσεις του χρόνου t , όπου t τα βήματα των επαναλήψεων του αλγορίθμου. Επίσης για κάθε σωματίδιο υπολογίζεται η συνάρτηση καταλληλότητας του. Ακόμα το σωματίδιο σε κάθε χρονική στιγμή t , αποθηκεύει πληροφορία για την θέση του, την προηγούμενη ταχύτητά του, την καλύτερη θέση που έχει μέχρι στιγμής (personal best), δηλαδή τη θέση που έχει αποδόσει την καλύτερη τιμή στην αντικειμενική συνάρτηση, καθώς και την θέση που έχει το βέλτιστο σωματίδιο στον πληθυσμό (global best), δηλαδή το σωματίδιο που έχει επιτύχει την καλύτερη τιμή για την

αντικειμενική συνάρτηση. Η ταχύτητα προσδιορίζει την επόμενη θέση του σωματιδίου στο χώρο των λύσεων και αποτελείται από ένα άθροισμα τριών όρων:

- Της προηγούμενης ταχυτητάς του. Η οποία χρησιμοποιείται ως μνήμη για να μην μπορέσει το σωματίδιο να κάνει μεγάλη αλλαγή στην κατεύθυνσή του κατά την διάρκεια μιας επανάληψης.
- Ο δεύτερος ποσοτικοποιεί την απόδοση που έχει κάθε σωματιδίου σε σύγκριση με τις αποδόσεις που είχε προηγουμένως. Ουσιαστικά ο όρος αυτός προσπαθεί να επαναφέρει το σωματίδιο σε παλιές πιο αποδοτικές καταστάσεις.
- Ο τρίτος όρος ποσοτικοποιεί την απόδοση που έχει κάθε σωματιδίου σε σχέση με τις αποδόσεις του πληθυσμού ή μιας ομάδας σωματιδίων. Ουσιαστικά αυτός ο όρος βασίζεται στη ανατροφοδότηση που δέχεται από τα γειτονικά σωματίδια και την βέλτιστη λύση που είχαν βρεί.

Στο επόμενο βήμα του αλγορίθμου, έχουμε αλλαγή της θέσης και της ταχύτητας των σωματιδίων σύμφωνα με τις παρακάτω εξισώσεις:

- $v_{ij}(t+1) = v_{ij}(t) + c_1 * rand_1 * (pbest_{ij} - x_{ij}(t)) + c_2 * rand_2 * (gbest_j - x_{ij}(t))$
- $x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)$

Όπου: $x_{ij}(t)$ η θέση του σωματιδίου i

$v_{ij}(t)$ η ταχυτητά του

c_1, c_2 μεταβλητές επιτάχυνσης, οι οποίες παίρνουν συνήθως την τιμή 2

r_1, r_2 δύο τυχαίες μεταβλητές στο διάστημα [0,1]

Εφόσον βρεθούν οι τιμές για την ταχύτητα και την θέση των σωματιδίων τότε δίνονται ξανά νέες τιμές στα σωματίδια και η διαδικασία που μόλις περιγράφηκε επαναλαμβάνεται μέχρι να πληρούνται τα κριτήρια για τη διακοπή του αλγορίθμου, δηλαδή να βρεθεί η επιθυμητή τιμή της συνάρτησης, ή να φτάσουμε στον μέγιστο αριθμό επαναλήψεων.

Τα πλεονεκτήματα του αλγορίθμου βελτιστοποίησης σμήνους σωματιδίων που τον κάνουν να υπερέχει σε σχέση με άλλες μεθόδους επίλυσης είναι⁵:

- Η ταχύτητα του. Αποδίδει τις ίδιες λύσεις σε πολύ λιγότερο χρόνο σε σχέση με άλλες μεθόδους, καθώς λειτουργεί με λιγότερες συναρτήσεις υπολογισμού.
- Η ποιότητα του. Έχει την ικανότητα να βρίσκει τόσο τοπικά όσο και ολικά βέλτιστα.
- Η ευελιξία του. Μπορεί και επιλύει πληθώρα προβλημάτων, διότι προσαρμόζεται γρήγορα σε αλλαγές του περιβάλλοντος του και σε εναλλαγές από στατικά σε δυναμικά περιβάλλοντα.
- Η απλότητα του σαν κώδικας. Λειτουργεί με την χρήση απλών και βασικών εξισώσεων και μπορεί να ολοκληρωθεί σε λίγες γραμμές κώδικα.

Βασικός ψευδοκώδικας του αλγορίθμου PSO:

Αλγόριθμος Βελτιστοποίηση Σμήνους Σωματιδίων

Αρχικοποίηση

Επιλογή του αριθμού των σμηνών

Επιλογή του αριθμού των σωματιδίων σε κάθε σμήνος

Αρχικοποίηση της θέσης και της ταχύτητας κάθε σωματιδίου

Υπολογισμός του αρχικού κόστους του κάθε σωματιδίου

Εύρεση Βέλτιστου σωματιδίου ολόκληρου του σμήνους

Εύρεση Βέλτιστης λύσης κάθε σωματιδίου

Κύρια Φάση

Do until δεν έχει φθάσει ο μέγιστος αριθμός επαναλήψεων

Υπολογισμός της ταχύτητας του κάθε σωματιδίου

Υπολογισμός της νέας θέσης του κάθε σωματιδίου

Υπολογισμός της νέας συνάρτησης ποιότητας του κάθε σωματιδίου

Ενημέρωση της βέλτιστης λύσης του κάθε σωματιδίου

Εύρεση του βέλτιστου σωματιδίου ολόκληρου του σμήνους

Enddo

Επιστροφή βέλτιστου σωματιδίου (βέλτιστης λύσης).

3.5.3 Αλγόριθμοι τοπικής αναζήτησης

3.5.3.1 Αλγόριθμος 1-1 exchange

Η μέθοδος αυτή χρησιμοποιεί σαν είσοδο μια αρχική λύση που έχει υπολογιστεί από κάποια προηγούμενη μέθοδο, και λειτουργεί με την ταυτόχρονη ανταλλαγή δύο πελατών από διαφορετικές όμως διαδρομές. Επίσης ο αλγόριθμος 1-1 exchange λειτουργεί για ολόκληρη τη λύση και όχι για μεμονωμένες διαδρομές.

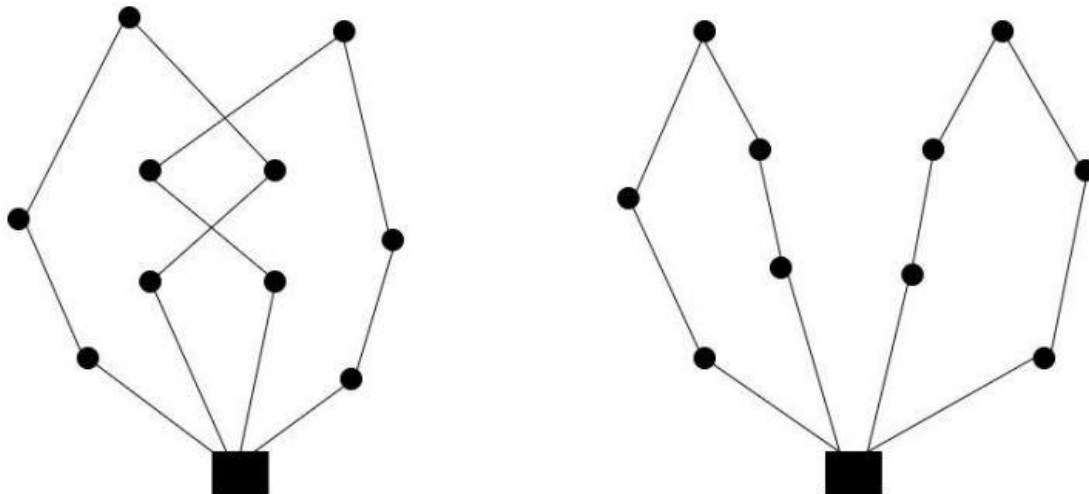
Η διαδικασία επιλογής των πελατών για τις ανταλλαγές μπορεί να είναι είτε τυχαία είτε σκόπιμη. Η τυχαία επιλογή, όμως, θα έχει ως αποτέλεσμα να γίνουν πάρα πολλές άσκοπες επαναλήψεις, που είτε δεν θα βελτιώνουν τη λύση, είτε θα την βελτιώνουν, αλλά οι καλές διαδρομές θα καταστραφούν. Οπότε για να είναι πιο αποδοτική η μέθοδος θα πρέπει η επιλογή των πελατών να γίνει στοχευμένα, δηλαδή να επιλεγθεί ο κόμβος που επιφέρει το μεγαλύτερο κόστος στην διαδρομή. Μετά την στοχευμένη επιλογή και ανταλλαγή πελατών, υπολογίζεται το κόστος και δεδομένου ότι οι δύο τροποποιημένες διαδρομές είναι φθηνότερες από τις αρχικές, η αλλαγή παραμένει. Ο αλγόριθμος αυτός επίσης τερματίζει όταν ικανοποιηθεί το κριτήριο τερματισμού².

Τα βήματα του αλγορίθμου:

Βήμα 1^ο : Επιλογή δύο πελατών (με τυχαίο τρόπο ή αυτόν με το μεγαλύτερο κόστος).

Βήμα 2^ο : Ανταλλαγή των θέσεων των δύο αυτών κόμβων και υπολογισμός του κόστους των δύο νέων διαδρομών. Εάν μειωθεί το κόστος η αλλαγή αυτή παραμένει.

Βήμα 3^ο : Η διαδικασία επαναλαμβάνεται μέχρις ότου ικανοποιηθεί το κριτήριο τερματισμού.



Σχήμα 3.5.3.1 Εφαρμογή αλγορίθμου 1-1 exchange

3.5.3.2 Αλγόριθμος 2-Opt

Ως είσοδο λαμβάνει επίσης μια αρχική λύση από κάποια προηγούμενη μέθοδο και προσπαθεί να τη βελτιστοποιήσει. Η αποδεκτή λύση της προηγούμενης μεθόδου χωρίζεται σε διαδρομές. Πιο συγκεκριμένα, το πιο πιθανόν ενδεχόμενο είναι να μην ικανοποιηθούν όλοι οι πελάτες σε μία μόνο διαδρομή. Και για τον λόγο αυτό η αρχική λύση χωρίστηκε σε όλες τις διαδρομές που έχουν ήδη δημιουργηθεί. Ο αλγόριθμος 2-opt εκτελείται σε κάθε πιθανή διαδρομή ξεχωριστά και όχι στην συνολική λύση.

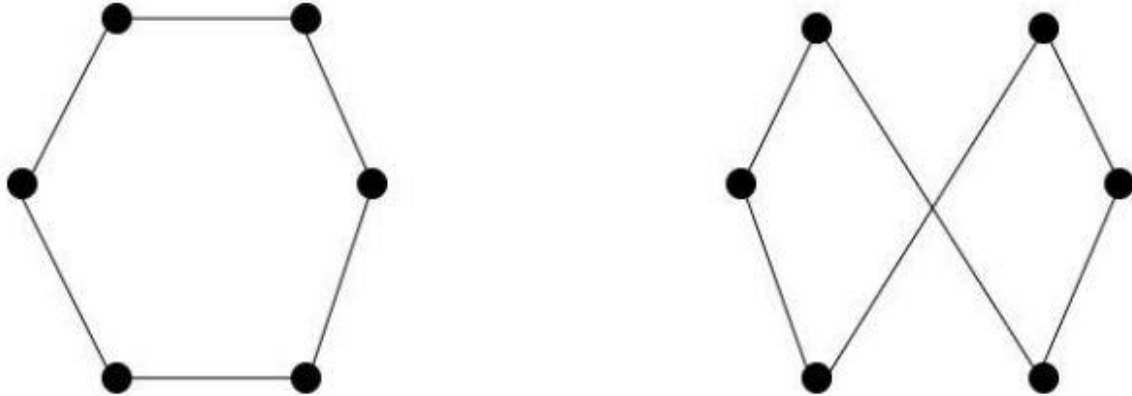
Αυτή η μέθοδος αφαιρεί τις δύο άκρες και επανασυνδέει τις δύο διαδρομές με άλλο τρόπο για να καθορίσει έτσι μία νέα διαδρομή. Η προϋπόθεση για τη διατήρηση μιας νέας λύσης είναι το κόστος να είναι μικρότερο από την αρχική λύση.

Η διαδικασία του 2-opt είναι η ακόλουθη² :

Βήμα 1^ο : Έστω μια τρέχουσα διαδρομή.

Βήμα 2^ο : Για κάθε κόμβο της διαδρομής, δοκιμάζουμε όλες τις πιθανές 2-opt κινήσεις που είναι εφικτό να γίνουν. Υπολογίζουμε έπειτα το κόστος κάθε νέας διαδρομής και εάν παρατηρηθεί μείωση του κόστους η αλλαγή διατηρείται.

Βήμα 3^ο : Η διαδικασία επαναλαμβάνεται μέχρις ότου ικανοποιηθεί το κριτήριο τερματισμού, δηλαδή δεν μπορεί να βρεθεί περαιτέρω βελτίωση.



Σχήμα 3.5.3.2 Εφαρμογή αλγορίθμου 2-opt

Κεφάλαιο 4: Επίλυση του ομαδοποιημένου προβλήματος δρομολόγησης οχημάτων

4.1 Εισαγωγή

Αυτό το κεφάλαιο θα παρέχει μια λεπτομερή περιγραφή της μεθοδολογίας επίλυσης του προβλήματος και των αλγορίθμων που χρησιμοποιήθηκαν για την επίτευξη των αποτελεσμάτων. Για την λύση του προβλήματος χρησιμοποιήθηκε η γλώσσα προγραμματισμού C++ στο προγραμματιστικό περιβάλλον του Codeblocks. Ακόμη στο κεφάλαιο αυτό παρατίθενται και οι πίνακες των αποτελεσμάτων.

4.2 Περιγραφή προβλήματος

Το πρόβλημα ανήκει στην κατηγορία των ομαδοποιημένων προβλημάτων δρομολόγησης οχημάτων, και η οποία έχει αναλυθεί σε προηγούμενη ενότητα. Το ομαδοποιημένο αυτό πρόβλημα αποτελείται από 50 σετ δεδομένων όπου κάθε σετ περιέχει, ένα σύνολο πελατών που εξυπηρετούνται, ορίζοντας την αποθήκη από την οποία προέρχονται τα οχήματα ως κόμβο 1, τον αριθμό των διαθέσιμων οχημάτων, τη χωρητικότητα κάθε οχήματος και τις συντεταγμένες x και y όπου βρίσκεται κάθε πελάτης. Επιπλέον η ζήτηση του κάθε πελάτη θεωρείται γνωστή, και ο πελάτης νούμερο 1 έχει μηδενική ζήτηση σε όλα τα σετ δεδομένων καθώς είναι η αποθήκη όπως προαναφέρθηκε. Ακόμη τα σετ είναι χωρισμένα σε δύο ομάδες, την Α ομάδα με 27 σετ και την Β ομάδα με 23 σετ. Επίσης για να υπάρχει ευκολία στην αναγνώριση των σετ τα ονοματά τους έχουν συγκεκριμένη μορφή. Για παράδειγμα το πρώτο σετ με όνομα **A-n32-k5** δηλώνει ότι βρίσκεται στην Α ομάδα, έχει 32 κόμβους ή πελάτες και διαθέτει 5 οχήματα.

Ως δεδομένα επιπλέον έχουμε την βέλτιστη λύση του κάθε σετ. Δηλαδή το ελάχιστο κόστος και τις k βέλτιστες διαδρομές που ικανοποιούν την ζήτηση, όπου k είναι ο αριθμός των οχημάτων που διαθέτει κάθε σετ και παίρνει τιμές από 5 μέχρι και 10. Οπότε με τα δεδομένα αυτά και με την χρήση των αλγορίθμων, που έχουν προαναφερθεί και θα

αναλυθούν στην συνέχεια, θα γίνει μια προσέγγιση-σύγκριση στις λύσεις του κάθε σετ δεδομένων.

4.3 Επίλυση του ομαδοποιημένου προβλήματος δρομολόγησης οχημάτων (CluVRP)

Για την επίλυση του προβλήματος ήταν απαραίτητο να εισάγουμε τα δεδομένα του κάθε σετ στον κώδικα, επαναλαμβάνοντας την διαδικασία αυτή 6 φορές. Η επάναληψη της διαδικασίας επιτελείται για την παραγωγή ενός ικανοποιητικού εύρους τιμών με σκοπό τον προσδιορισμό της ευστάθειας των αλγορίθμων, και στην συνέχεια με γνώμονα αυτές τις τιμές θα παρουσιαστεί ένα συγκεντρωτικό πινακάκι που θα περιλαμβάνει για κάθε σετ δεδομένων :

- τον μέσο όρο των αποτελεσμάτων που βρέθηκαν από τον κώδικα,
- την απόκλιση τους από την βέλτιστη τιμή (που υπάρχει ως δεδομένο).

Για την λύση ενός σετ δεδομένων, βασική προϋπόθεση είναι η δημιουργία αρχικών λύσεων χρησιμοποιώντας τη μέθοδο του πλησιέστερου γείτονα. Έπειτα, με την χρήση του αλγορίθμου σμήνους σωματιδίων (PSO) και αλγορίθμων τοπικής αναζήτησης, επιτυγχάνεται βελτιστοποίηση των αρχικών αυτών λύσεων. Οι αλγόριθμοι τοπικής αναζήτησης που χρησιμοποιήθηκαν είναι ο 1-1 exchange και 2-Opt. Έπειτα επαναλαμβάνεται η ίδια μεθοδολογία επίλυσης και για τα υπόλοιπα σετ δεδομένων, με σκοπό την κατασκευή των συγκεντρωτικών πινάκων που παρουσιάζονται στην συνέχεια.

Πιο αναλυτικά, ο αλγόριθμος του πλησιέστερου γείτονα είναι ο πρώτος που συναντάμε στον κώδικα καθώς είναι αυτός που κατασκευάζει τις αρχικές λύσεις. Πριν όμως χρησιμοποιήσουμε τον αλγόριθμο αυτό, εισαγουμε τα δεδομένα του σετ στον κώδικα, προκειμένου να κατασκευαστούν οι κόμβοι – πελάτες του προβλήματος που αναπαριστά καθένα αρχείο. Δηλαδή, για κάθε κόμβο το όνομά του (ένας αριθμός), τις συντεταγμένες x και y της θέσης του, καθώς και τη ζήτησή του, το πλήθος των κόμβων του προβλήματος και την μέγιστη χωριτηκότητα που διαθέτει κάθε όχημα. Έπειτα υπολογίζουμε τις αποστάσεις όλων των κόμβων και τις τοποθετούμε σε ένα διάνυσμα διανυσμάτων ή πίνακα. Επίσης, ο

πίνακας των αποστάσεων χρησιμεύει για να βρίσκουμε το κόστος καθενός δρομολογίου που θα κατασκευάζουμε, αθροίζοντας τις αποστάσεις μεταξύ των αντίστοιχων κόμβων του. Μετά από αυτή την υλοποίηση είμαστε σε θέση να χρησιμοποιήσουμε τον αλγόριθμο του πλησιέστερου γείτονα. Ο αλγόριθμος αυτός μας παράγει μία αρχική λύση που όμως δεν είναι αρκετή για την σωστή χρήση του αλγορίθμου PSO, που συναντάμε αμέσως μετά. Για τον λόγο αυτό χρησιμοποιούμε την εντολή *random_shuffle* η οποία με τυχαίο ανακάτεμα της αρχικής μας λύσης δημιουργεί τόσες λύσεις-δρομολόγια όσα είναι τα *particles_number* - 1. Κατά αυτόν τον τρόπο έχουμε συνολικά τόσες αρχικές λύσεις όσα είναι τα *particles_number* και έτσι παράγονται επιπρόσθετες αρχικές λύσεις, οι οποίες σε συνδυασμό με αυτές του πλησιέστερου γείτονα δημιουργούν το εφικτό εύρος λύσεων που χρειάζεται για να υλοποιηθεί σωστά ο αλγόριθμος PSO. Πριν την χρήση του PSO υπολογίζουμε την συνάρτηση κόστους και μετατρέπουμε τις αρχικές λύσεις που υπολογίσαμε από διακριτή σε συνεχή μορφή, για να είναι εφικτό να εφαρμοστούν οι εξισώσεις του. Επιπλέον, για να αποφευχθούν κυκλικές διαδρομές, ένας κόμβος δεν πρέπει να εμφανίζεται περισσότερες από μία φορές σε μια διαδρομή.

Για την εκτέλεση του αλγορίθμου PSO, η αρχικοποίηση των παραμέτρων ορίζεται στην αρχή της εκτέλεσης του αλγορίθμου και παραμένει σταθερή καθ' όλη τη διάρκεια. Για την αρχικοποίηση έχουμε, *c1* και *c2* μεταβλητές επιτάχυνσης με τιμή 2.0, *rand1* και *rand2* δύο τυχαίες μεταβλητές που στην περίπτωση μας έχουν τιμή 0.5 και 0.8 αντίστοιχα, όπου οι σταθερές αυτές συμμετέχουν στην εξίσωση της ταχύτητας του αλγορίθμου. Επίσης επιλέγουμε τον αριθμό του σμήνους(*swarm*) και των σωματιδίων κάθε σμήνους(*particles*), η αρχική ταχύτητα κάθε σωματιδίου θεωρείται μηδενική και η αρχική του θέση θεωρείται η συνεχής του αναπαράσταση. Ακόμη η βέλτιστη θέση (*pbest*) καθενός σωματιδίου θεωρείται η αρχική του θέση και βέλτιστο σωματίδιο (*gbest*) αρχικά θεωρείται το πρώτο σωματίδιο του σμήνους.

Ακολούθως, ξεκινάει ο βασικός κορμός του PSO με κριτήριο τερματισμού το πλήθος των επαναλήψεων *N*, όπου για καθένα σωματίδιο έχουμε τα ακόλουθα βήματα :

- 1) Υπολογίζεται με τις εξισώσεις του PSO η νέα ταχύτητα και η νέα θέση του σωματιδίου.

- 2) Με βάση τη νέα θέση που έχει το κάθε σωματίδιο μετατρέπεται σε διακριτή αναπαράσταση και οι κόμβοι επανατοποθετούνται στη σωστή σειρά.
- 3) Υπολογίζεται το κόστος της λύσης.
- 4) Εφαρμόζεται ο αλγόριθμος τοπικής αναζήτησης 1-1 exchange προσπαθώντας να βελτιωθεί η παραγόμενη λύση.
- 5) Υπολογίζεται το κόστος της λύσης του 1-1 exchange.
- 6) Εφόσον η λύση του 1-1 exchange έχει μικρότερο κόστος από το κόστος του σωματιδίου :
 - η λύση του 1-1 exchange μετατρέπεται σε συνεχή μορφή,
 - και ενημερώνεται η θέση (x) του σωματιδίου.
- 7) Εφόσον το κόστος της λύσης του 1-1exchange είναι μικρότερο από τη βέλτιστη θέση (pbest) του σωματιδίου, ενημερώνεται η βέλτιστη θέση του σωματιδίου.
- 8) Εφόσον το κόστος της λύσης του 1-1exchange είναι μικρότερο από τη βέλτιστη θέση όλων των σωματιδίων (gbest), ενημερώνεται η βέλτιστη θέση όλων των σωματιδίων.
- 9) Εφαρμόζεται ο αλγόριθμος τοπικής αναζήτησης 2-opt προσπαθώντας να βελτιώσει περαιτέρω τη λύση.
- 10) Υπολογίζεται το κόστος της λύσης του 2-opt.
- 11) Εφόσον η λύση του 2-opt έχει μικρότερο κόστος:
 - η λύση του 2-opt μετατρέπεται σε συνεχή μορφή,
 - ενημερώνεται η θέση (x) του σωματιδίου.
- 12) Εφόσον το κόστος της λύσης του 2-opt είναι μικρότερο από τη βέλτιστη θέση (pbest) του σωματιδίου, ενημερώνεται η βέλτιστη θέση του σωματιδίου.
- 13) Εφόσον το κόστος της λύσης του 2-opt είναι μικρότερο από τη βέλτιστη θέση όλων των σωματιδίων (gbest), ενημερώνεται η βέλτιστη θέση όλων των σωματιδίων.

Όταν τελειώσει ο PSO, το βέλτιστο σωματίδιο μετατρέπεται σε διακριτή αναπαράσταση και οι κόμβοι της λύσης επανατοποθετούνται στη σωστή σειρά. Επίσης, κατασκευάζονται τα επιμέρους δρομολόγια που συνθέτουν τη συνολική λύση και υπολογίζεται το κόστος της συνολικής λύσης.

Στο σημείο αυτό επισημαίνεται ότι η υλοποίηση των αλγορίθμων τοπικής αναζήτησης 1-1 exchange και 2-opt έγινε έτσι ώστε οι αλγόριθμοι να αναζητούν εξαντλητικά τη βέλτιστη αλλαγή των κόμβων της λύσης με βάση το ελάχιστο κόστος. Αυτό έχει σαν αποτέλεσμα οι αλγόριθμοι να δίνουν μεν το βέλτιστο αποτέλεσμα, αλλά ο χρόνος εκτέλεσης αυξάνεται σημαντικά.

Ακολουθεί ενδεικτικά η εκτέλεση των παραπάνω βημάτων σε ορισμένα από τα παραδείγματα. Το πλήθος των σωματιδίων και των επαναλήψεων του αλγορίθμου που παρουσιάζονται στην συνέχεια, είναι αυτά που μας έφεραν τα καλύτερα δυνατά αποτελέσματα και με την πιο μικρή απόκλιση από τα βέλτιστα, και σε λογικά χρονικά πλαίσια. Και ακόμη τα c_1 , c_2 , $rand_1$, $rand_2$ παραμένουν σταθερά σε όλες τις υλοποιήσεις.

Υπενθυμίζεται ότι :

- το πλήθος των σωματιδίων(**particles**) του PSO ορίζεται σε 50,
- το πλήθος των επαναλήψεων(**N**) του PSO ορίζεται σε 200,
- $c_1=c_2=2.0$, $rand_1=0.5$ και $rand_2=0.8$, χωριτικότητα=100.

ΟΜΑΔΑ Α

Με 32 κόμβους και 5 οχήματα:

Βέλτιστη λύση (αρχείο A-n32-k5.sol):	Η λύση που προέκυψε:
Route #1: 21 31 19 17 13 7 26 Route #2: 12 1 16 30 Route #3: 27 24 Route #4: 29 18 8 9 22 15 10 25 5 20 Route #5: 14 28 11 4 23 3 2 6	Route#1 : 20 5 25 10 15 9 22 29 , Cost: 237 Route#2 : 2 3 23 28 4 11 8 18 14 , Cost: 232 Route#3 : 30 26 16 7 1 12 , Cost: 95 Route#4 : 21 31 19 17 13 6 , Cost: 183 Route#5 : 24 27 , Cost: 59.3
Optimal Cost = 784	Total Cost= 806
Διαφορά λύσεων = 22 ή 2,8%	

Με 33 κόμβους και 6 οχήματα:

Βέλτιστη λύση (αρχείο A-n33-k6.sol):	Η λύση που προέκυψε:
Route #1: 5 2 20 15 9 3 8 4 Route #2: 31 24 23 26 22 Route #3: 17 11 29 19 7 Route #4: 10 12 21 Route #5: 28 27 30 16 25 32 Route #6: 13 6 18 1 14 Optimal Cost=742	Route#1 : 31 23 24 22 26 4 8 , Cost: 195 Route#2 : 19 29 11 17 10 , Cost: 160 Route#3 : 21 12 , Cost: 72.3 Route#4 : 32 25 16 30 27 28 , Cost: 99.5 Route#5 : 14 7 20 9 15 2 3 5 , Cost: 162 Route#6 : 1 18 6 13 , Cost: 67.2 Total Cost= 755
Διαφορά λύσεων = 13 ή 1,75%	

Με 45 κόμβους και 7 οχήματα:

Βέλτιστη λύση (αρχείο A-n45-k7.sol):	Η λύση που προέκυψε:
Route #1: 8 42 1 37 30 5 22 Route #2: 13 38 17 23 25 15 10 Route #3: 18 7 16 20 40 Route #4: 39 21 24 44 14 9 32 Route #5: 33 29 19 36 31 Route #6: 27 41 43 11 3 28 6 2 Route #7: 12 4 26 34 35 Optimal Cost= 1146	Route #1 : 27 29 41 43 11 3 2 , Cost: 225 Route #2 : 14 33 31 36 19 5 21 , Cost: 198 Route #3 : 4 26 34 35 , Cost: 70.2 Route #4 : 22 30 37 1 42 8 39 , Cost: 226 Route #5 : 10 15 25 23 17 38 , Cost: 160 Route #6 : 18 40 7 20 16 13 32 9 , Cost: 145 Route #7 : 24 44 28 6 12 , Cost: 175 Total Cost=1199
Διαφορά λύσεων = 53 ή 4,62%	

Με 55 κόμβους και 9 οχήματα:

Βέλτιστη λύση (αρχείο A-n55-k9.sol):	Η λύση που προέκυψε:
Route #1: 4 7 42 31 20 46 26 Route #2: 36 11 15 51 2 17 14 Route #3: 37 3 34 33 21 Route #4: 1 45 6 8 Route #5: 25 41 29 Route #6: 23 52 24 44 50 48 18 Route #7: 32 38 16 40 53 5 10 12 Route #8: 30 22 19 27 13 54 28 Route #9: 47 39 49 9 35 43 Optimal Cost=1073	Route#1 : 41 28 13 27 19 , Cost: 89.5 Route#2 : 6 45 43 1 , Cost: 142 Route#3 : 50 48 44 24 52 23 54 22 , Cost: 168 Route#4 : 30 4 2 51 11 15 36 , Cost: 187 Route#5 : 14 46 17 34 3 37 , Cost: 73.8 Route#6 : 26 20 31 42 7 , Cost: 89.8 Route#7 : 25 29 18 12 21 , Cost: 88.4 Route#8 : 32 10 5 53 38 16 40 33 , Cost: 156 Route#9 : 35 9 49 39 47 8 , Cost: 172 Total Cost=1166
Διαφορά λύσεων = 93 ή 8,6%	

ΟΜΑΔΑ Β

Με 31 κόμβους και 5 οχήματα:

Βέλτιστη λύση (αρχείο B-n31-k5.sol):	Η λύση που προέκυψε:
Route #1: 30 23 8 12 28 26 Route #2: 21 16 18 25 5 4 29 Route #3: 7 17 13 6 9 22 Route #4: 20 27 10 2 Route #5: 14 15 11 24 19 1 3 Optimal Cost =672	Route #1 : 26 28 12 23 30 8 , Cost: 108 Route #2 : 14 15 11 24 19 1 3 , Cost: 155 Route #3 : 29 4 25 5 18 16 21 , Cost: 113 Route #4 : 22 6 9 17 13 7 , Cost: 111 Route #5 : 20 27 10 2 , Cost: 190 Total Cost=676
Διαφορά λύσεων = 4 ή 0,59%	

Με 33 κόμβους και 6 οχήματα:

Βέλτιστη λύση (αρχείο B-n38-k6.sol):	Η λύση που προέκυψε:
Route #1: 36 17 8 34 24 29 12 7 25 Route #2: 31 30 2 35 37 13 3 11 Route #3: 27 5 4 33 18 23 Route #4: 26 16 Route #5: 14 21 22 6 19 9 Route #6: 20 10 1 15 28 32 Optimal Cost =805	Route #1 : 36 17 8 34 24 29 12 7 25 , Cost: 167 Route #2 : 15 20 10 1 28 32 , Cost: 121 Route #3 : 3 13 37 35 2 30 19 9 , Cost: 214 Route #4 : 23 18 33 4 5 27 , Cost: 177 Route #5 : 26 16 22 11 21 14 , Cost: 74 Route #6 : 6 31 , Cost: 61.5 Total Cost=813
Διαφορά λύσεων = 8 ή 0,99%	

Με 44 κόμβους και 7 οχήματα:

Βέλτιστη λύση (αρχείο B-n44-k7.sol):	Η λύση που προέκυψε:
Route #1: 32 38 22 42 30 8 4 Route #2: 18 11 23 10 Route #3: 29 16 7 Route #4: 5 17 2 36 14 39 24 26 9 Route #5: 20 33 43 31 27 35 15 Route #6: 40 1 12 3 28 25 Route #7: 6 21 34 41 37 13 19 Optimal Cost= 909	Route #1 : 17 1 12 3 25 20 40 , Cost: 123 Route #2 : 10 38 22 30 42 8 , Cost: 176 Route #3 : 5 29 16 7 , Cost: 65.2 Route #4 : 23 11 9 6 , Cost: 170 Route #5 : 28 33 43 31 27 15 , Cost: 131 Route #6 : 35 2 36 14 39 24 26 4 32 , Cost: 229 Route #7 : 18 19 13 37 41 34 21 , Cost: 75.4 Total Cost=969
Διαφορά λύσεων = 60 ή 6,6%	

Με 57 κόμβους και 9 οχήματα:

Βέλτιστη λύση (αρχείο B-n57-k9.sol):	Η λύση που προέκυψε:
Route #1: 31 55 9 28 18 27 1 Route #2: 32 40 3 50 46 22 Route #3: 4 30 49 35 48 38 12 Route #4: 53 29 41 21 36 47 5 Route #5: 43 54 52 24 45 17 10 Route #6: 42 44 16 23 2 20 7 Route #7: 33 11 37 25 51 15 8 Route #8: 6 56 Route #9: 14 34 19 39 26 13 Cost 1598	Route #1 : 5 47 41 51 25 15 8 , Cost: 199 Route #2 : 35 14 39 13 44 11 7 , Cost: 226 Route #3 : 22 10 17 50 3 40 32 , Cost: 132 Route #4 : 46 9 28 18 27 45 54 , Cost: 222 Route #5 : 31 55 12 49 30 4 , Cost: 167 Route #6 : 33 34 19 26 23 42 , Cost: 228 Route #7 : 53 29 20 16 2 37 21 36 , Cost: 218 Route #8 : 38 48 43 52 24 1 , Cost: 203 Route #9 : 56 6 , Cost: 103 Total Cost=1697
Διαφορά λύσεων = 99 ή 6,19%	

4.4 Συγκεντρωτικοί πίνακες λύσεων-αποκλίσεων

Παρακάτω παρουσιάζονται τα αποτελέσματα σε πίνακες όπου η δεύτερη στήλη αναφέρετε στην βέλτιστη λύση που βρέθηκε στην βιβλιογραφία. Στη συνέχεια φαίνονται αναλυτικά τα αποτελέσματα με χρήση μόνο του αλγορίθμου PSO με 50 επαναλήψεις και 20 particles και έπειτα του PSO σε συνδυασμό με 1-1exchange και 2-Opt με συγκεκριμένους συνδυασμούς N και particles. Τέλος, στους παρακάτω πίνακες είναι οι αποκλίσεις των αποτελεσμάτων για όλες τις εναλλακτικές και για τις δύο ομάδες, καθώς και η συνολική απόκλιση όλων των εναλλακτικών για τα 50 παραδείγματα.

Α Ομάδα		Λύσεις με PSO, 1-1exchange και 2-opt			
Παραδείγματα	Βέλτιστη λύση	PSO	Με N=50, particles=20	Με N=100, particles=40	Με N=200, particles=50
A-n32-k5	784	1136	826	822	806
A-n33-k5	661	858	697	693	682
A-n33-k6	742	991	773	743	755
A-n34-k5	778	946	816	809	810
A-n36-k5	799	1065	838	828	841
A-n37-k5	669	864	727	701	678
A-n37-k6	949	1213	986	1001	994
A-n38-k5	730	1008	763	772	774
A-n39-k5	822	1073	904	862	873
A-n39-k6	831	1118	905	877	886
A-n44-k6	937	1289	1025	1016	978
A-n45-k6	944	1421	1060	1056	1036
A-n45-k7	1146	1442	1235	1201	1199
A-n46-k7	914	1312	1054	1022	969
A-n48-k7	1073	1374	1218	1176	1190
A-n53-k7	1010	1304	1163	1122	1152
A-n54-k7	1167	1512	1319	1265	1285
A-n55-k9	1073	1433	1209	1192	1166
A-n60-k9	1354	1748	1522	1533	1508
A-n61-k9	1034	1376	1163	1198	1173
A-n62-k8	1288	1650	1405	1489	1469
A-n63-k10	1314	1796	1509	1498	1409
A-n63-k9	1616	2366	1847	1785	1784
A-n64-k9	1401	1810	1599	1579	1561
A-n65-k9	1174	1762	1463	1376	1397
A-n69-k9	1159	1572	1401	1341	1325
A-n80-k10	1763	2193	2076	2071	2047
Αποκλίση από βέλτιστα		33,8%	11,12%	9,37%	8,44%

B Ομάδα			Λύσεις με PSO, 1-1exchange και 2-opt		
Παραδείγματα	Βέλτιστη λύση	PSO	Με N=50, particles=20	Με N=100, particles=40	Με N=200, particles=50
B-n31-k5	672	772	690	683	676
B-n34-k5	788	868	802	797	793
B-n35-k5	955	1225	973	973	974
B-n38-k6	805	922	825	833	813
B-n39-k5	549	746	572	572	568
B-n41-k6	829	1088	881	850	854
B-n43-k6	742	969	795	794	759
B-n44-k7	909	1113	1005	982	969
B-n45-k5	751	1011	804	807	795
B-n45-k6	678	922	777	726	735
B-n50-k7	741	970	791	776	806
B-n50-k8	1312	1537	1356	1355	1351
B-n51-k7	1032	1332	1187	1104	1101
B-n52-k7	747	945	829	807	843
B-n56-k7	707	1107	765	783	755
B-n57-k7	1153	1730	1282	1270	1273
B-n57-k9	1598	1877	1738	1698	1697
B-n63-k10	1496	2067	1717	1688	1644
B-n64-k9	861	1183	1027	1023	977
B-n66-k9	1316	1645	1486	1405	1445
B-n67-k10	1032	1455	1146	1195	1181
B-n68-k9	1272	1538	1452	1386	1409
B-n78-k10	1221	1475	1404	1429	1397
Αποκλίση από βέλτιστα		29,11%	9,13%	7,6%	7%

		Λύσεις με PSO, 1-1exchange και 2-opt		
Α&Β ομάδα	PSO	N=50, particles=20	N=100, particles=40	N=200, particles=50
Συνολική απόκλιση	31,66%	10,2%	8,55%	7,78%

4.5 Συμπεράσματα

Στη παρούσα διπλωματική ζητήθηκε το ομαδοποιημένο πρόβλημα δρομολόγησης οχημάτων να επιλυθεί με τη διαδικασία βελτιστοποίηση σμήνους σωματιδίων σε συνδυασμό με τοπικές αναζητήσεις. Παρατηρείται αρχικά σε όλες τις περιπτώσεις ότι όσο αυξάνονται οι κόμβοι και τα οχήματα στα παραδείγματα οι αποκλίσεις μεγαλώνουν, το οποίο είναι απόλυτα λογικό καθώς η λύση των παραδειγμάτων αυτών είναι πιο περίπλοκη και σύμφωνα με αυτά μπορούμε να παρατηρήσουμε πόσο αποδοτικός και ικανοποιητικός είναι ο αλγόριθμος μας.

Σε αυτό το σημείο πρέπει να τονιστεί ότι, έπειτα από πολλές δοκιμές, πρακτικά δεν γίνεται να επιτευχθούν τα ίδια αποτελέσματα με εκείνα της βιβλιογραφίας, καθώς στην περίπτωση αυτή έχουν χρησιμοποιηθεί στοχευμένοι αλγόριθμοι τοπικής αναζήτησης οι οποίοι σε συνδυασμό με τον αλγόριθμο PSO δίνουν τα βέλτιστα αποτελέσματα. Ενώ στην περίπτωση μας χρησιμοποιήθηκαν τα εργαλεία που έχουμε διδαχθεί στο μάθημα της εφοδιαστικής αλυσίδας. Επίσης η χρήση των αλγορίθμων τοπικής αναζήτησης έγινε με στοχευμένη εξαντλική αναζήτηση στους κόμβους και όχι τυχαία, και κατά αυτόν το τρόπο έχουμε μεν τα καλύτερα δυνατά αποτελέσματα αλλά αυξάνεται αρκετά ο χρόνος εκτέλεσης.

Όπως παρουσιάστηκε και προηγουμένως, χρησιμοποιήθηκαν διάφορα ζευγάρια εναλλακτικών τιμών για το πλήθος των επαναλήψεων του PSO (N) και των σωματιδίων (particles). Έπειτα λοιπόν από πολλές δοκιμές, επιλέχθηκαν οι εναλλακτικές εκείνες με την μικρότερη απόκλιση και παράλληλα τον μικρότερο χρόνο εκτέλεσης και είναι αυτές που παρουσιάστηκαν στους προηγούμενους πίνακες. Έδω να ειπωθεί ότι, η αύξηση των επαναλήψεων και του πλήθους των σωματιδίων βελτιώνει τις λύσεις του προβλήματος αυξάνοντας όμως τον χρόνο εκτέλεσης, και συγκεκριμένα έχουμε πολύ μικρή βελτίωση των λύσεων και τεράστια αύξηση του χρόνου επίλυσης.

Συνοψίζοντας τα καλύτερα αποτελέσματα παρατηρούνται στις 200 επαναλήψεις και τα 50 particles με συνολική απόκλιση 7,78% και χρόνο εκτέλεσης του κώδικα για τα 50 σετ 6 με 7 ώρες.

Βιβλιογραφία

1)Iztok Fister jr. (2013). A Brief Review of Nature-Inspired Algorithms for Optimization. Article. University of Maribor.

2)Μαρινάκης Ιωάννης, Μαρινάκη Μαγδαληνή, Μυγδάλας Αθανάσιος (2019). Προβλήματα Δρομολόγησης Οχημάτων στη διαχείριση της Εφοδιαστικής Αλυσίδας. Εκδόσεις Νέων Τεχνολογιών.

3)Μαρινάκης Ιωάννης (2020), διαλέξεις από το μάθημα: « Σχεδιασμός και Βελτιστοποίηση της Εφοδιαστικής Αλυσίδας», Πολυτεχνείο Κρήτης, Τμήμα Μηχανικών Παραγωγής και Διοίκησης.

4) Hybrid particle swarm optimization algorithm for solving the clustered vehicle routing problem, Md. Anisul Islam , Yuvraj Gajpal , Tarek Y. ElMekkawy

5) ΤΣΙΑΚΟΥΛΗ ΣΠΥΡΙΔΟΥΛΑ (2016), Χρήση του Αλγορίθμου Βελτιστοποίησης Σμήνους Σωματιδίων για την Επιτήρηση Θερμοκρασίας μιας Περιοχής, Υπεύθυνος Καθηγητής: ΠΑΠΟΥΤΣΙΔΑΚΗΣ ΜΙΧΑΛΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ ΤΜΗΜΑ ΑΥΤΟΜΑΤΙΣΜΟΥ