



## Διπλωματική Εργασία

Διαδικασία Άπληστης Τυχαιοποιημένης Προσαρμοστικής  
Αναζήτησης για το Πρόβλημα Ταυτόχρονης Δρομολόγησης  
και Αποθεματοποίησης

**Του:** Τρίμα Δημήτριου

**Επιβλέπων Καθηγητής:** Καθηγητής, Ιωάννης Μαρινάκης

**Μέλος Επιτροπής:** Καθηγητής, Ματσατσίνης Νικόλαος

**Μέλος Επιτροπής:** ΕΔΙΠ, Μαρινάκη Μαγδαληνή

Τμήμα Μηχανικών Παραγωγής και Διοίκησης

Πολυτεχνείο Κρήτης

Χανιά, Ελλάδα 2023



## Ευχαριστίες

Στην παρούσα παράγραφο θα ήθελα να ευχαριστήσω όλους αυτούς που στάθηκαν δίπλα μου κατά τη διάρκεια των φοιτητικών μου χρόνων, με κάθε υπαρκτό τρόπο. Ένα μεγάλο ευχαριστώ στην οικογένειά μου για όλα όσα μου έχουν προσφέρει και στους φίλους μου που μοιραστήκαμε τόσα. Επίσης θα ήθελα να ευχαριστήσω ειδικά τον διδακτορικό του κ. Μαρινάκη, κ. Νικόλαο – Αντώνιο Κυριακάκη (πλέον Δόκτωρ Νικόλαο – Αντώνιο Κυριακάκη), για την πολύτιμη βοήθεια που μου προσέφερε κατά τη διάρκεια εκπόνησης της συγκεκριμένης διπλωματικής εργασίας καθώς και για την υπομονή του και το χρόνο του.

# Περιεχόμενα

## Κεφάλαιο 1<sup>ο</sup>

ΕΙΣΑΓΩΓΗ	6
1.1 Ιστορική Αναδρομή	6
1.2 Ορισμός - Έννοια Εφοδιαστικής Αλυσίδας	7
1.3 Ορισμός - Έννοια των Logistics	9

## Κεφάλαιο 2<sup>ο</sup>

ΔΡΟΜΟΛΟΓΗΣΗ ΟΧΗΜΑΤΩΝ	10
2.1 Το Πρόβλημα Δρομολόγησης Οχημάτων	10
2.2 Επίλυση προβλημάτων δρομολόγησης οχημάτων και αλγορίθμων αρχικής λύσης	11
2.3 Αλγόριθμοι τοπικής αναζήτησης	12
2.4 Μεθευρετικοί αλγόριθμοι	14

## Κεφάλαιο 3<sup>ο</sup>

ΠΡΟΒΛΗΜΑΤΑ ΤΑΥΤΟΧΡΟΝΗΣ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΚΑΙ ΑΠΟΘΕΜΑΤΟΠΟΙΗΣΗΣ (IRP)	15
3.1 Γενικά	15
3.2 Μοντελοποίηση	19

## Κεφάλαιο 4<sup>ο</sup>

### ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

### ΤΑΥΤΟΧΡΟΝΗΣ ΔΡΟΜΟΛΟΓΗΣΗΣ ΚΑΙ ΑΠΟΘΕΜΑΤΟΠΟΙΗΣΗΣ

### ΜΕ ΧΡΗΣΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ GRASP 23

4.1	Διαδικασία Άπληστης Τυχαιοποιημένης Προσαρμοστικής Αναζήτησης (Greedy Randomized Adaptive Search Procedure) . . . . .	23
4.2	Περιγραφή του προβλήματος και λύσης του . . . . .	25
4.2.1	Αρχική Λύση . . . . .	26
4.2.2	Τοπικοί Αλγόριθμοι Αποθεματοποίησης . . . . .	30
4.3	Ψευδοκώδικας . . . . .	37

## Κεφάλαιο 5<sup>ο</sup>

### ΕΞΑΓΩΓΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΚΑΙ ΣΧΟΛΙΑΣΜΟΣ 38

5.1	Παραμετροποίηση . . . . .	39
5.2	Αποτελέσματα . . . . .	40
5.3	Συγκρίσεις . . . . .	50
5.4	Συμπεράσματα και παρατηρήσεις . . . . .	52
5.5	Πιθανή μελλοντική έρευνα . . . . .	53

## Βιβλιογραφία

## ΚΕΦΑΛΑΙΟ 1

### ΕΙΣΑΓΩΓΗ

#### 1.1 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

Ετυμολογικά ο όρος Logistics/Εφοδιαστική προέρχεται από την ελληνική λέξη «λόγος» που σημαίνει λογική, με την έννοια της εκλογίκευσης αποσκοπώντας στην επίτευξη καθαρά ορισμένων στόχων.

Ο όρος χρησιμοποιείται από την αρχαιότητα και συγκεκριμένα κατά κύριο λόγο σε πολεμικές εκστρατείες. Πιο συγκεκριμένα σύμφωνα με ιστορικούς, ο πρώτος που χρησιμοποίησε τον όρο ήταν ο Μέγας Αλέξανδρος καθώς ήταν υψίστης σημασίας ο ανεφοδιασμός των στρατευμάτων του κατά την εκστρατεία του στα βάθη της Περσίας.

Οι αρχαίοι Αιγύπτιοι μετέφεραν ολόκληρα μνημεία από καταστραμμένες πόλεις σε καινούργιες, προκειμένου να θαυμάζονται εκ νέου και να κοσμούν τις νεοσύστατες πόλεις.

Στο πιο πρόσφατο παρελθόν, μαζική χρήση της «Εφοδιαστικής» έλαβε χώρα κατά τη διάρκεια του Β' Παγκοσμίου Πολέμου από τους Αμερικάνους και τους Συμμάχους με κύριο στόχο τότε, τη μαζική μεταφορά στρατιωτικού προσωπικού και εξοπλισμού από τη μια άκρη του Ατλαντικού στην άλλη, καθώς επίσης και τη συνεχή κάλυψη των αναγκών σε τροφή, εξοπλισμό, υλικά και πυρομαχικά των στρατιωτικών μονάδων που επιχειρούσαν.

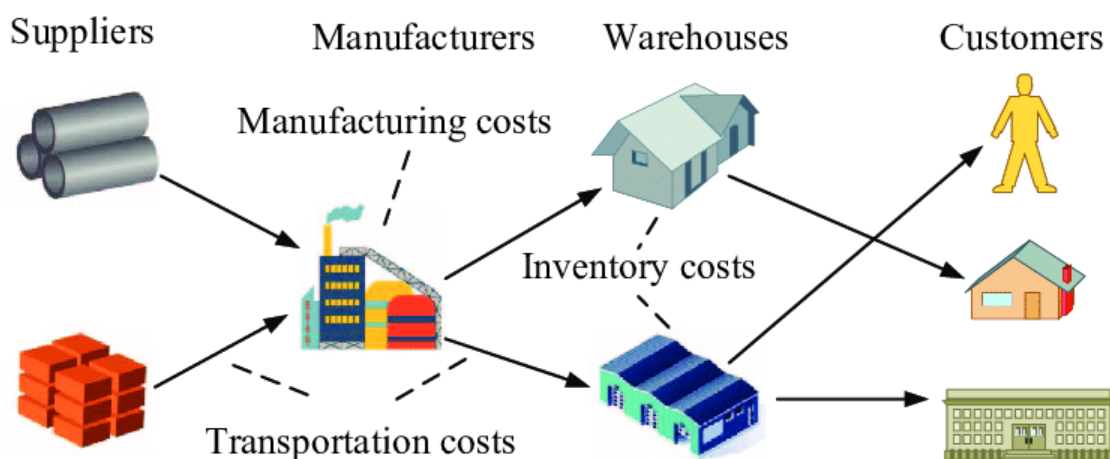
Μεταπολεμικά, η Εφοδιαστική Αλυσίδα άρχισε να ερευνάται επιστημονικά από τη δεκαετία του '60 με σκοπό να δώσει λύσεις και νέους τρόπους βιωσιμότητας στις επιχειρήσεις, αρχικά στον τομέα της διανομής, ένας κλάδος ο οποίος μέχρι τότε βρισκόνταν στο περιθώριο. Τα αποτελέσματα ήταν εντυπωσιακά, καθώς επιτυγχάνονταν πρωτόγνωρα θετικά για την εποχή

νούμερα όσον αφορά τη μείωση κόστους, ταχύτητα εξυπηρέτησης και συνεπώς αύξηση ανταγωνιστικότητας της επιχείρησης στο σύνολό της.

Τέλος, η ραγδαία ταχύτητα την παγκοσμιοποίησης δημιούργησε νέες ανάγκες σε πολλούς τομείς. Ανάγκες διανομής και αποθήκευσης προϊόντων, μέχρι συνεργασίας μεταξύ επιχειρήσεων, κρατών και αυτών μεταξύ τους.

## 1.2 ΟΡΙΣΜΟΣ – ΕΝΝΟΙΑ ΕΦΟΔΙΑΣΤΙΚΗΣ ΑΛΥΣΙΔΑΣ

«Η εφοδιαστική αλυσίδα είναι η διαδικασία του σχεδιασμού, της εφαρμογής και του ελέγχου της αποτελεσματικής ροής και αποθήκευσης πρώτων υλών, ημικατεργασμένων προϊόντων, τελικών προϊόντων και της σχετιζόμενης πληροφορίας από το σημείο προσέλευσης στο σημείο κατανάλωσης με σκοπό τη συμμόρφωση στις απαιτήσεις των πελατών.»  
( Μαρινάκης, Μαρινάκη, Μυγδαλάς 2019)



Εικόνα 1 Μία επισκόπηση της εφοδιαστικής αλυσίδας

[https://www.researchgate.net/figure/An-overview-of-the-logistics-network\\_fig2\\_312562721](https://www.researchgate.net/figure/An-overview-of-the-logistics-network_fig2_312562721)

Όπως φαίνεται από τον ορισμό που έχει δοθεί παραπάνω, αλλά και από το διάγραμμα, η εφοδιαστική αλυσίδα έχει ως σκοπό τη βελτίωση της απόδοσης της εταιρείας ούτως ώστε, τα προϊόντα και υπηρεσίες που παρέχονται, να φτάνουν στην καλύτερη δυνατή κατάσταση στον πελάτη-καταναλωτή, τη χρονική στιγμή που απαιτείται να φτάσουν.

Ορισμένες από τις βασικές λειτουργίες της διαχείρισης της εφοδιαστικής αλυσίδας είναι οι εξής:

- Επιλογή προμηθευτών
- Εξέταση αποθεμάτων
- Διαχείριση αποθηκών
- Επιλογή κατάλληλης αποθήκης
- Διανομή
- Εξυπηρέτηση των πελατών

Γενικά, υπάρχει η εντύπωση πως η εφοδιαστική αλυσίδα παρέχει μια ροή προϊόντων αποκλειστικά προς μια και μόνο κατεύθυνση, ωστόσο αυτό είναι λάθος. Η εφοδιαστική αλυσίδα τελειώνει με την παράδοση του προϊόντος στον πελάτη, ωστόσο τα προϊόντα μπορεί να είναι ελαττωματικά, να χαλάνε ή απλά να χρειάζονται επιστροφή επειδή πάλιωσαν. Έτσι ξεκινάει μια ροή από τον καταναλωτή προς τα πίσω. Αυτή η αντίστροφη ροή ωστόσο πρέπει να λαμβάνεται κανονικά υπόψη ως μέρος της εφοδιαστικής αλυσίδας κατά το σχεδιασμό και τον έλεγχο της και όχι ως κάτι ατομικά ξεχωριστό.



### 1.3 ΟΡΙΣΜΟΣ – ΕΝΝΟΙΑ ΤΩΝ LOGISTICS

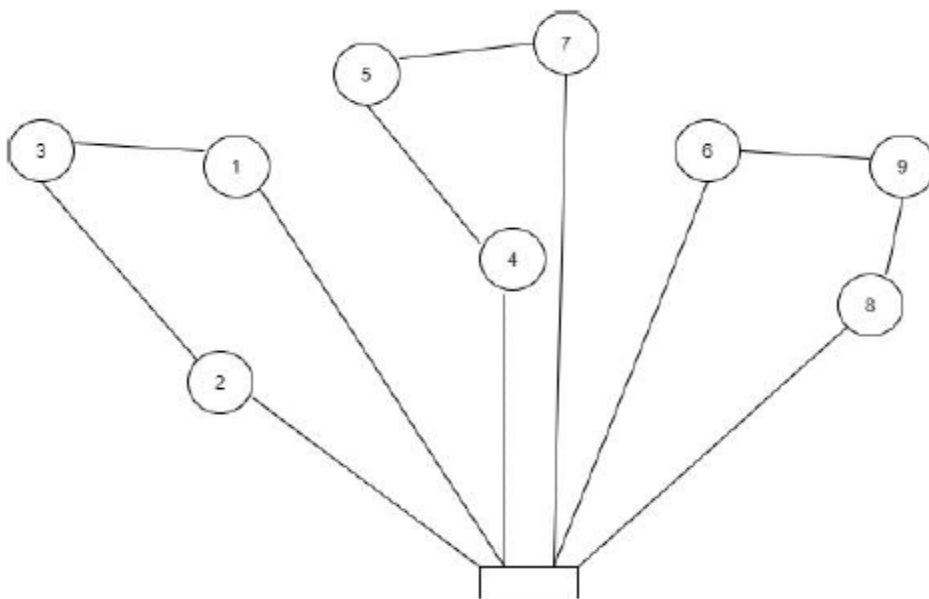
Σύμφωνα με την ελληνική εταιρεία logistics «Logistics είναι το τμήμα εκείνο της εφοδιαστικής αλυσίδας που σχεδιάζει, υλοποιεί και ελέγχει την αποδοτική και αποτελεσματική – κανονική και αντίστροφη- ροή και αποθήκευση των προϊόντων, υπηρεσιών και των σχετικών πληροφοριών από το σημείο προέλευσής τους έως το σημείο κατανάλωσής τους, ώστε να ικανοποιηθούν οι απαιτήσεις των πελατών». Όπως φαίνεται και από τον δοθέντα ορισμό, τα **Logistics** αποτελούν αναπόσπαστο κομμάτι της *Εφοδιαστικής Αλυσίδας*.

## ΚΕΦΑΛΑΙΟ 2

### ΔΡΟΜΟΛΟΓΗΣΗ ΟΧΗΜΑΤΩΝ

#### 2.1 ΤΟ ΠΡΟΒΛΗΜΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ (VRP)

«Το *Πρόβλημα Δρομολόγησης Οχημάτων* αφορά την παράδοση/παραλαβή προϊόντων στους/και από τους πελάτες τα οποία προκύπτουν στα πλαίσια των επιχειρησιακών δραστηριοτήτων μεταφοράς.» (Μαρινάκης, Μαρινάκη, Μυγδαλάς 2019)



Εικόνα 2 Απλό παράδειγμα δρομολόγησης οχημάτων. (Μια αποθήκη και εννέα πελάτες)

Η πιο απλή μορφή του συγκεκριμένου τύπου προβλημάτων φαίνεται στην παραπάνω εικόνα (εικ.2). Στη συγκεκριμένη εικόνα το **Πρόβλημα Δρομολόγησης Οχημάτων** αποτελείται από μια αποθήκη και εννέα

πελάτες τους οποίους προκειμένου να τους εξυπηρετήσουμε μπαίνουν σε 3 δρομολόγια. Σκοπός είναι να εξυπηρετηθούν οι πελάτες κάνοντας τη μικρότερη δυνατή διαδρομή. Οι διαδρομές που εκτελούνται έχουν σαν αφετηρία την αποθήκη και ως τελικό κόμβο επιστρέφουν στην αποθήκη. Μπορεί τις διαδρομές να τις έχει εκτελέσει μόνο ένα όχημα, ξεκινώντας ένα νέο δρομολόγιο κάθε φορά που τελειώνει ένα άλλο και γυρίζει στην αποθήκη, ή τρία διαφορετικά οχήματα ξεκινώντας ταυτόχρονα από την αποθήκη - αφετηρία.

Ωστόσο, τα προβλήματα που αντιμετωπίζονται στην πραγματικότητα από την *Εφοδιαστική Αλυσίδα* είναι πολύ πιο σύνθετα. Πρέπει να τηρούνται συγκεκριμένα χρονικά πλαίσια, υπάρχουν περιορισμοί ως προς τα διαθέσιμα οχήματα και τη χωρητικότητά τους, καθώς και συγκεκριμένα όρια που μπορεί να υπάρχουν στις αποθήκες καθώς και άλλα πολλά.

## 2.2 ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΩΝ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΚΑΙ ΑΛΓΟΡΙΘΜΟΙ ΑΡΧΙΚΗΣ ΛΥΣΗΣ

Προκειμένου να επεξεργαστούμε και να λύσουμε προβλήματα συνδυαστικής βελτιστοποίησης, δουλεύουμε σε δύο φάσεις. Στην πρώτη φάση δημιουργούμε μία **αρχική λύση** μέσω ενός αλγορίθμου απληστίας και στη δεύτερη χρησιμοποιούμε κάποιον ή κάποιους αλγορίθμους τοπικής αναζήτησης προκειμένου να μεγιστοποιήσουμε ή να ελαχιστοποιήσουμε μια αντικειμενική συνάρτηση υπό την τήρηση κάποιων περιορισμών. Στα *Προβλήματα Δρομολόγησης Οχημάτων* κάνουμε λόγο για ελαχιστοποίηση κόστους. Σύμφωνα με την αρχή της απληστίας, ο αλγόριθμος ξεκινάει, δημιουργώντας μια μερική λύση, μη εφικτή. Σε κάθε επόμενο βήμα, ο αλγόριθμος κοιτάει να βρει και να

κάνει τη βέλτιστη επιλογή. Αυτό, επαναλαμβάνεται έως ότου δημιουργηθεί μία εφικτή λύση η οποία θα είναι και η γενική του, η οποία κατά τη δημιουργία της θα τηρεί όλους τους πιθανούς περιορισμούς που έχουν τεθεί και θα ελαχιστοποιεί την αντικειμενική της συνάρτηση. Οι πιο συχνοί Αλγόριθμοι Κατασκευής Αρχικής Λύσης είναι οι παρακάτω:

- Αλγόριθμος Πλησιέστερου Γείτονα
- Αλγόριθμος Πλησιέστερης Εκχώρησης
- Αλγόριθμος Εξοικονομήσεων των Clarke και Wright

## 2.3 ΑΛΓΟΡΙΘΜΟΙ ΤΟΠΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ

Οι αλγόριθμοι τοπικής αναζήτησης εφαρμόζονται πάνω στην αρχική λύση που έχει δημιουργηθεί. Κοιτώντας στη γειτονιά της αρχικής λύσης ο αλγόριθμος, εφαρμόζει τις βελτιωτικές κινήσεις που έχουν επιλεγεί. Αυτές οι κινήσεις εφαρμόζονται επαναληπτικά έως ότου να μην μπορεί η λύση μας να βελτιωθεί άλλο, δηλαδή μέχρι να πέσει ο αλγόριθμος σε κάποιο τοπικό ελάχιστο. Όταν συμβεί αυτό, σημαίνει πως έχουμε πάρει τη βελτιωμένη λύση στο πρόβλημά μας. Πέραν του ότι θα πρέπει να επιλεγθεί κατάλληλη γειτονιά για να δώσει τα βέλτιστα αποτελέσματα ο αλγόριθμος τοπικής αναζήτησης, θα πρέπει να είναι και ο κατάλληλος για το πρόβλημα που καλούμαστε να βελτιστοποιήσουμε. Υπάρχουν αρκετοί αλγόριθμοι τοπικής αναζήτησης, ωστόσο στα Προβλήματα Δρομολόγησης Οχημάτων με σκοπό τη βελτιστοποίηση του κόστους διαδρομής, η «Αγία Τριάδα» αποτελείται από τους εξής παρακάτω:

### 1. Μέθοδος 2-opt

Στον συγκεκριμένο αλγόριθμο διαγράφονται δύο ακμές και επανασυνδέονται με τέτοιο τρόπο, ώστε να προκύψει μια καινούργια διαδρομή. Εφόσον το κόστος δρομολόγησης της διαδρομής μειώνεται, η διαδρομή αλλάζει σύμφωνα με τη διαγραφή και την επανασύνδεση των ακμών. Πιο συγκεκριμένα για κάθε κόμβο εξετάζουμε τις πιθανές 2-opt κινήσεις του. Εάν το κόστος μειώνεται, λαμβάνουν χώρα οι αλλαγές που έχουμε βρει. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να μην μειώνεται άλλο το κόστος.

### 2. Μέθοδος 1-1 exchange

Ο συγκεκριμένος αλγόριθμος κάνει ανταλλαγές κόμβων στην ίδια διαδρομή ή και μεταξύ διαδρομών της ίδιας χρονικής περιόδου, εφόσον τηρούνται οι περιορισμοί που έχουν τεθεί. Ειδικά, για κάθε κόμβο  $i$ , ψάχνουμε να βρούμε έναν κόμβο  $j$ , για τους οποίους εφόσον αλλάξουμε τις μεταξύ τους θέσεις θα ελαττωθεί το συνολικό κόστος δρομολόγησης. Όταν δεν είναι εφικτό, κάποια ανταλλαγή να μας δώσει καλύτερο αποτέλεσμα, η διαδικασία ολοκληρώνεται.

### 3. Μέθοδος 1-0 relocate

Η συγκεκριμένη μέθοδος αποτελείται από τη διαγραφή ενός κόμβου από μία διαδρομή και η επανατοποθέτησή του σε μία άλλη, ή από την επανατοποθέτηση του στην ίδια διαδρομή αλλά, σε διαφορετική θέση. Η διαδικασία είναι επαναλαμβανόμενη και σταματάει όταν πλέον δεν είναι δυνατόν να αλλάξει η θέση κάποιου κόμβου για να πάρουμε καλύτερο αποτέλεσμα.

## 2.4 ΜΕΘΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Οι αλγόριθμοι τοπικής αναζήτησης όπως αναφέραμε προηγουμένως σταματάνε όταν πέφτουν πάνω σε τοπικό ελάχιστο. Προκειμένου να λύσουμε αυτό το πρόβλημα και να ξεκολλήσουμε από κάποιο τοπικό ελάχιστο, χρησιμοποιούνται αλγόριθμοι που είναι γνωστοί ως **Μεθευρετικοί Αλγόριθμοι**. Ο διαχωρισμός τους γίνεται ανάλογα με το πόσες λύσεις χρησιμοποιούν. Συγκεκριμένοι άλλοι αλγόριθμοι χρησιμοποιούν μονάχα μία λύση και ενεργούν στη γειτονιά αυτής, ενώ άλλοι έχουν στη διάθεσή τους μια πληθώρα λύσεων και κάνουν αναζήτηση σε όλες αυτές. Μπορούμε να κατηγοριοποιήσουμε περαιτέρω τους αλγορίθμους που χρησιμοποιούν μία λύση ανάλογα τη μέθοδο που χρησιμοποιούν, ως εξής:

- Επαναληπτικές διαδικασίες που αρχίζουν από διαφορετικές αρχικές λύσεις, π.χ. Διαδικασία Άπληστης Τυχαιοποιημένης Προσαρμοστικής Αναζήτησης (GRASP).
- Αλγόριθμοι που αλλάζουν τη γειτονιά αναζήτησης, π.χ. Αλγόριθμος Μεταβλητής Γειτονιάς Αναζήτησης (VNS).
- Αλγόριθμοι που δέχονται γειτονικές κινήσεις που δε βελτιώνουν τη λύση, π.χ. Αλγόριθμος Περιορισμένης Αναζήτησης (Tabu Search).
- Αλγόριθμοι που αλλάζουν την αντικειμενική συνάρτηση ή κάποια από τα δεδομένα του προβλήματος, π.χ. Αλγόριθμος Καθοδηγούμενης Τοπικής Αναζήτησης (Guided Local Search).

## ΚΕΦΑΛΑΙΟ 3

### ΠΡΟΒΛΗΜΑΤΑ ΤΑΥΤΟΧΡΟΝΗΣ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΚΑΙ ΑΠΟΘΕΜΑΤΟΠΟΙΗΣΗΣ (IRP)

#### 3.1 ΓΕΝΙΚΑ

Κατά κύριο λόγο ο ανεφοδιασμός των αποθεμάτων ενός πελάτη συμβαίνει ως εξής: αφού τα αποθέματα προϊόντων ενός πελάτη πέσουν κάτω από ένα επίπεδο που έχει οριστεί από τον ίδιο, ο πελάτης στέλνει την παραγγελία του στον προμηθευτή με την ποσότητα προϊόντων που χρειάζεται. Ο προμηθευτής λαμβάνει όλες τις παραγγελίες του από τους πελάτες του και έπειτα, καθορίζει τα δρομολόγια που θα κάνει με τρόπο τέτοιο, ώστε το κόστος δρομολόγησης να είναι το μικρότερο δυνατό. Ουσιαστικά λύνει ένα πρόβλημα δρομολόγησης οχημάτων.

Ωστόσο, με αυτόν τον τρόπο ανατροφοδότησης των πελατών δημιουργείται ένα σημαντικό πρόβλημα στον προμηθευτή. Μη έχοντας τη δυνατότητα να ελέγξει ο ίδιος το πότε θα επισκεφτεί τους πελάτες του, είναι συχνό φαινόμενο να συσσωρεύονται παραγγελίες από πολλούς πελάτες την ίδια χρονική περίοδο. Αυτό σαν αποτέλεσμα οδηγεί σε υπερβολικά υψηλά κόστη δρομολόγησης, διότι πρέπει να εξυπηρετηθούν πολλοί πελάτες την ίδια χρονική στιγμή, συνεπώς τα δρομολόγια είναι πολύ μεγάλα και οι αποστάσεις τεράστιες.

Στις αρχές της δεκαετίας του '80 άρχισε να αναπτύσσεται ένα νέο μοντέλο αναφορικά με τη δρομολόγηση και την διαχείριση αποθεμάτων. Η προσέγγιση αυτή ονομάστηκε *Διαχείριση Αποθεμάτων από τους Πωλητές* (vendor manage inventory). Σύμφωνα με αυτό το μοντέλο, την

πλήρη διαχείριση των αποθεμάτων του κάθε πελάτη την έχει ο προμηθευτής, δηλαδή το πότε θα ανεφοδιαστεί ο πελάτης και με πόση ποσότητα. Με αυτόν τον τρόπο ο προμηθευτής-πωλητής, είναι σε θέση να φτιάξει έτσι τα δρομολόγια, ώστε να ελαχιστοποιήσει τα κόστη δρομολόγησης που προκύπτουν. Το μόνο που χρειάζεται να προσέξει ο πωλητής είναι να μη μείνει κάποιος πελάτης χωρίς απόθεμα για να λειτουργήσει.

Τα προβλήματα ταυτόχρονης δρομολόγησης και αποθεματοποίησης (inventory routing problems) έχουν ομαδοποιηθεί σε τρεις μεγάλες κατηγορίες σύμφωνα με το χρονικό ορίζοντα που διαρκούν.

1. **Προβλήματα απλής περιόδου (single period problems):** Η ζήτηση των πελατών είναι στοχαστική και το κυρίως πρόβλημα σε αυτήν την κατηγορία είναι η εξισορρόπηση των κόστων μεταφοράς, αποθεματοποίησης και έλλειψης κάποιου προϊόντος.
2. **Προβλήματα πολλαπλών περιόδων (multi period problems):** Σε αυτήν την κατηγορία η ζήτηση των πελατών είναι είτε στοχαστική είτε διακριτή. Το γενικό πρόβλημα εδώ είναι η ελαχιστοποίηση του κόστους κρατώντας παράλληλα μια ποσότητα αποθεμάτων στις αποθήκες του πελάτη.
3. **Προβλήματα με άπειρο χρονικό ορίζοντα:** Σε αυτήν την κατηγορία η ζήτηση του κάθε πελάτη είναι προκαθορισμένη. Το γενικό πρόβλημα σε αυτή την κατηγορία είναι να οριστούν πολιτικές ανεφοδιασμού για τον κάθε πελάτη, καθώς και να βρεθούν οι καταλληλότερες διαδρομές για τα οχήματα.  
(Μαρινάκης, Μαρινάκη, Μυγδαλάς 2019)



Υπάρχουν επτά κατηγορίες με βάση τις οποίες μπορούμε να ταξινομήσουμε τις μορφές του:

- 1) Χρονικός ορίζοντας
  - 2) Δομή
  - 3) Αποφάσεις δρομολόγησης
  - 4) Αριθμός οχημάτων
  - 5) Χωρητικότητα οχημάτων
  - 6) Πολιτική αποθεματοποίησης
  - 7) Αποφάσεις αποθεματοποίησης
- (Μαρινάκης, Μαρινάκη, Μυγδαλάς 2019)

Σύμφωνα με τα παραπάνω γίνεται εύκολα αντιληπτό ότι υπάρχουν πάρα πολλές παραλλαγές του συγκεκριμένου προβλήματος, συνεπώς δεν μπορεί να οριστεί μια γενική μορφή του. Στα περισσότερα αυτού του είδους προβλήματα, τίθεται μια συγκεκριμένη πολιτική αποθεματοποίησης. Οι δύο πιο συχνά χρησιμοποιούμενες πολιτικές είναι η **πολιτική παραγγελίας μέχρι ενός συγκεκριμένου επιπέδου (*order up to level*)** και η άλλη είναι η **πολιτική μεγίστου επιπέδου αποθεμάτων (*maximum level*)**. Στην πρώτη πολιτική, όταν ένα όχημα επισκέπτεται έναν πελάτη τότε αυτό αφήνει ποσότητα τέτοια, ώστε να συμπληρώσει την ποσότητα που βρίσκεται στην αποθήκη του πελάτη μέχρι το σημείο που έχει οριστεί. Στη δεύτερη, το ποσό που θα αναπληρωθεί στον πελάτη είναι πιο ευέλικτο αλλά υπάρχει ο περιορισμός της χωρητικότητας της αποθήκης του.

Στη συγκεκριμένη διπλωματική εργασία εξετάστηκε το μοντέλο IRP σε έναν πεπερασμένο ορίζοντα προγραμματισμού του χρόνου  $T$  ( $t = 1, 2, \dots, T$ ) χωρίς κάποια συγκεκριμένη πολιτική αποθεματοποίησης. Αυτός ο ορίζοντας επαναλαμβάνεται περιοδικά στο άπειρο, με χρόνο κύκλου ίσο με το μήκος του  $T$ . Αυτή η υπόθεση οδηγεί στη δημιουργία πρακτικών

και αναγνωρίσιμων χρονοδιαγραμμάτων προγραμματισμού για κάποια βασική περίοδο μιας εβδομάδας, μήνα, έτους κ.λπ. Το μοντέλο ορίζεται σε ένα γράφημα  $G = (V, A)$ , όπου  $V = \{v_0, v_1, \dots, v_n\}$  είναι το σύνολο κορυφής και  $A = \{(v_i, v_j): v_i, v_j \in V, i \neq j\}$  είναι το σύνολο τόξων. Η κορυφή  $v_0$  αντιπροσωπεύει τον κεντρικό κόμβο (αποθήκη), όπου δεν κρατείται κάποια συγκεκριμένη ποσότητα εμπορεύματος. Αυτή η αποθήκη λειτουργεί ως σταθμός για ένα στόλο οχημάτων  $K$ . Κάθε όχημα  $veh_i$  ( $i = 1, \dots, K$ ) έχει χωρητικότητα ίση με  $Q$ . Οι κορυφές του  $V/\{v_0\}$  σχηματίζουν το σετ πελάτη (πωλητής λιανικής). Με κάθε πελάτη  $v_i$  ( $i = 1, 2, \dots, n$ ) συσχετίζεται μια καθορισμένη ζήτηση προϊόντος  $d_i$  ανά μονάδα χρόνου  $t$ , και ένα μοναδιαίο κόστος διατήρησης αποθέματος ίσο με  $h_i$  ( $d_i$  και  $h_i$  έχουν ακέραιες τιμές). Το κόστος που απαιτείται για τη μετακίνηση μεταξύ ενός ζευγαριού τοποθεσιών  $(v_i, v_j) \in A$  είναι  $c_{ij}$ . Όλοι οι έμποροι λιανικής έχουν επαρκή χωρητικότητα στις αποθήκες προκειμένου να καλύψουν ποσότητα ίση με τη συνολική ζήτηση σε έναν κύκλο  $D_i$  ( $D_i = T^* d_i$ ). Στην κάθε χρονική στιγμή  $t$ , κάθε πελάτης μπορεί να επισκέπτεται μόνο μία φορά – από οποιοδήποτε από τα  $K$  διαθέσιμα οχήματα – για να λάβει κάποια ποσότητα προϊόντος. Η συνολική ποσότητα που παραδίδεται σε κάθε πελάτη κατά τη διάρκεια του χρονικού ορίζοντα πρέπει να ισούται με τη συνολική του ζήτηση  $D_i$ . Το φορτίο που μεταφέρεται από το κάθε όχημα δεν μπορεί να υπερβεί τη χωρητικότητα  $Q$  του οχήματος. Ο στόχος του προβλήματος είναι να προσδιοριστούν οι διαδρομές που θα γίνουν από τα οχήματα, για κάθε μία από τις χρονικές μονάδες του χρονικού ορίζοντα  $T$ , μαζί με τις σχετικές ποσότητες παράδοσης στους πελάτες, έτσι ώστε να ελαχιστοποιηθεί το συνολικό κόστος μεταφοράς και το κόστος αποθεματοποίησης συνδυαστικά.

### 3.2 ΜΟΝΤΕΛΟΠΟΙΗΣΗ

Για να μοντελοποιήσουμε το πρόβλημα μαθηματικά, εισάγουμε τα ακόλουθα: έστω  $x_{ijtk}$  μια δυαδική μεταβλητή που ισούται με 1, εάν και μόνο εάν η κορυφή  $v_j$  επισκέπτεται αμέσως μετά το  $v_i$  ( $i \neq j$ ) τη χρονική στιγμή  $t$ , με το όχημα  $k$ . Η μεταβλητή  $q_{it}$  αντιπροσωπεύει την ποσότητα του προϊόντος που παραδίδεται στον πελάτη  $i$ , τη χρονική στιγμή  $t$ . Το απόθεμα ενός πελάτη  $v_i$ , τη χρονική μονάδα  $t$ , πριν την πιθανή αναπλήρωσή του, ισούται με  $sb_{it}$ . Η δυαδική μεταβλητή  $z_{it}$  είναι ίση με 1, εάν και μόνο εάν λάβει ο πελάτης  $v_i$  κάποια ποσότητα προϊόντος τη στιγμή  $t$  ( $q_{it} > 0$ ). Η μεταβλητή  $e_{it}$  είναι ίση με 1, εάν το επίπεδο αποθέματος πριν από την αναπλήρωση του πελάτη  $v_i$  τη στιγμή  $t$  είναι μεγαλύτερο από 0 ( $sb_{it} > 0$ ) και 0 διαφορετικά ( $sb_{it} = 0$ ). Τέλος, στο πάνω μέρος δεσμεύοντας τις ποσότητες παράδοσης, χρησιμοποιούμε  $M = \max(D_i) + 1$ , ( $i = 1, \dots, n$ ).

Η μοντελοποίηση του προβλήματος έχει ως εξής:

$$\text{Min} \quad \left( \sum_{t=1}^T \sum_{i=1}^n \left( sb_{it} + q_{it} - \frac{d_i}{2} \right) \cdot h_i + \sum_{k=1}^K \sum_{t=1}^T \sum_{i=1}^n \sum_{j=1}^n x_{ijkt} \cdot c_{ij} \right) \quad (1)$$

$$\text{Υπό: } \sum_{i=0}^n x_{ijtk} - \sum_{p=0}^n x_{jp tk} = 0, \quad j = (0, \dots, n); k = (1, \dots, K); t = (1, \dots, T) \quad (2)$$

$$\sum_{j=1}^n x_{0jtk} \leq 1, \quad k = (1, \dots, K); t = (1, \dots, T) \quad (3)$$

$$q_{it} \leq M \cdot z_{it}, \quad i = (0, \dots, n); t = (1, \dots, T) \quad (4)$$

$$q_{it} \geq z_{it}, \quad i = (0, \dots, n); t = (1, \dots, T) \quad (5)$$

$$\sum_{k=1}^K \sum_{j=0}^n x_{ijtk} - z_{it} = 0, \quad i = (0, \dots, n); t = (1, \dots, T) \quad (6)$$

$$\sum_{i=0}^n \sum_{j=0}^n x_{ijtk} \cdot q_{it} \leq Q, \quad k = (1, \dots, K); t = (1, \dots, T) \quad (7)$$

$$sb_{i(t+1)} = sb_{it} + q_{it} - d_i, \quad i = (0, \dots, n); t = (1, \dots, T) \quad (8)$$

$$sb_{i1} = sb_{iT} + q_{iT} - d_i, \quad i = (1, \dots, n) \quad (9)$$

$$sb_{it} \leq M \cdot e_{it}, \quad i = (0, \dots, n); t = (1, \dots, T) \quad (10)$$

$$sb_{it} \geq e_{it}, \quad i = (0, \dots, n); t = (1, \dots, T) \quad (11)$$

$$\sum_{t=1}^T e_{it} < T, \quad i = (1, \dots, n) \quad (12)$$

$$\sum_{v_i \in B} \sum_{v_j \in B} x_{ijtk} \leq |B| - 1, \quad k = (1, \dots, K); t = (1, \dots, T); B \subseteq V \setminus \{v_o\}; |B| > 1; \quad (13)$$

$$x_{ijtk} \in \{0, 1\}, \quad i = (1, \dots, n); j = (0, \dots, n); k = (1, \dots, K); t = (1, \dots, T) \quad (14)$$

$$z_{it} \in \{0, 1\}, \quad i = (0, \dots, n); t = (1, \dots, T) \quad (15)$$

$$e_{it} \in \{0, 1\}, \quad i = (0, \dots, n); t = (1, \dots, T) \quad (16)$$

$$q_{it} \text{ integer} \quad i = (0, \dots, n); t = (1, \dots, T) \quad (17)$$

Η αντικειμενική συνάρτηση (1) αποτελείται από δύο όρους. Η πρώτη αξιολογεί το συνολικό κόστος αποθεμάτων, ενώ το δεύτερο αντιστοιχεί στο συνολικό κόστος δρομολόγησης κατά την περίοδο προγραμματισμού.

Οι περιορισμοί (2) βεβαιώνουν ότι όταν ένα όχημα επισκέπτεται μια τοποθεσία, μετά αποχωρεί επίσης από αυτή.

Ο περιορισμός (3) μας εξασφαλίζει ότι κάθε χρονική στιγμή κάθε όχημα θα κάνει μία και μόνο διαδρομή.

Οι σχέσεις (4) και (5) εγγυώνται ότι  $z_{it}=1$ , αν και μόνο αν  $q_{it} > 0$ .

Οι περιορισμοί (6) δηλώνουν ότι όποτε ένας πελάτης λαμβάνει κάποια ποσότητα προϊόντος, αυτός ο πελάτης λαμβάνει προϊόν μόνο από ένα όχημα, ενώ ο (7) είναι ο περιορισμός χωρητικότητας των οχημάτων.

Το επίπεδο αποθεμάτων ενός πελάτη τη στιγμή  $t+1$  εκφράζεται ως συνάρτηση του επιπέδου αποθέματος και του ποσού αναπλήρωσης στην

προηγούμενη μονάδα χρόνου  $t$ , από τις σχέσεις (8) και (9). Οι περιορισμοί (10) και (11) ορίζουν ότι  $e_{it}=1$ , εάν και μόνο εάν  $sb_{it} > 0$ .

Ο περιορισμός (11) επίσης, δεν επιτρέπει καμία εξάντληση αποθεμάτων για τις αποθήκες.

Ο περιορισμός (12) ορίζει ότι για κάθε πελάτη  $v_i$  υπάρχει τουλάχιστον μία μονάδα χρόνου  $t^*$ , για το οποίο  $sb_{it^*} = 0$ . Αυτοί οι περιορισμοί επιβάλλουν το ελάχιστο απόθεμα κάθε πελάτη να είναι στο 0.

Χρησιμοποιούνται για να ορίσουν κόστος αποθέματος του πελάτη  $v_i$  ξεχωριστά από τους άλλους, για μια δεδομένη σειρά αναπληρώσεων αποθεμάτων ποσότητας  $q_{it}$  ( $t = 1, \dots, T$ ).

Οι περιορισμοί (13) εγγυώνται ότι η αποθήκη υπάρχει σε κάθε διαδρομή που δημιουργείται.

Τέλος οι περιορισμοί (14)–(17) προσδιορίζουν τον τύπο των μεταβλητών του μοντέλου.

## ΚΕΦΑΛΑΙΟ 4

### **ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΤΑΥΤΟΧΡΟΝΗΣ ΔΡΟΜΟΛΟΓΗΣΗΣ ΚΑΙ ΑΠΟΘΕΜΑΤΟΠΟΙΗΣΗΣ ΜΕ ΧΡΗΣΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ GRASP**

Η συγκεκριμένη διπλωματική εργασία έχει ως σκοπό, να επιλύσει ένα Πρόβλημα Ταυτόχρονης Δρομολόγησης και Αποθεματοποίησης με χρήση του αλγορίθμου Grasp. Τα παραδείγματα αναφοράς που χρησιμοποιήθηκαν, πάρθηκαν από τη δημοσίευση «An integrated local search method for inventory and routing decisions» των Emmanouil E. Zachariadis, Christos D. Tarantilis, Chris T. Kiranoudis [3], οι οποίοι έλυσαν το συγκεκριμένο πρόβλημα χρησιμοποιώντας τον μεθευρετικό αλγόριθμο Tabu. Προκειμένου να δούμε την αποτελεσματικότητα του αλγορίθμου για το συγκεκριμένο πρόβλημα, θα συγκρίνουμε τα αποτελέσματα του προγράμματός, με τα αποτελέσματα της συγκεκριμένης επιστημονικής δημοσίευσης.

#### **4.1 ΔΙΑΔΙΚΑΣΙΑ ΑΠΛΗΣΤΗΣ ΤΥΧΑΙΟΠΟΙΗΜΕΝΗΣ ΠΡΟΣΑΡΜΟΣΤΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ (Greedy Randomized Adaptive Search Procedure)**

Η Διαδικασία Άπληστης Τυχαιοποιημένης Προσαρμοστικής Αναζήτησης, είναι μια επαναληπτική διαδικασία, με σκοπό την εύρεση προσεγγιστικών λύσεων. Κάθε επανάληψη σημαίνει και πιθανή λύση, ενώ κάθε επανάληψη χωρίζεται σε δύο φάσεις. Στην πρώτη φάση

κατασκευάζεται μια άπληστη τυχαιοποιημένη αρχική λύση μέσω χρήσης της “λίστας περιορισμού των υποψηφίων” (restricted candidate list). Σε αυτήν τη λίστα μπαίνει ένα πλήθος καλύτερων πιθανών υποψηφίων στόχων, ο οποίος επιλέγεται μέσα από χρήση μιας συνάρτησης απληστίας. Μπορεί να είναι όλοι οι υποψήφιοι στόχοι ισοπίθανοι προς επιλογή από τη λίστα ή να διαφοροποιούνται ανάλογα με τους περιορισμούς. Μόλις δημιουργηθεί η αρχική λύση συνεχίζουμε στη δεύτερη φάση η οποία αποτελείται από τη βελτίωση της αρχικής λύσης με χρήση κάποιων αλγορίθμων τοπικής αναζήτησης. Η όλη διαδικασία επαναλαμβάνεται όσες φορές έχουμε θέσει και στο τέλος κρατάμε την καλύτερη λύση από όλες τις επαναλήψεις που κάναμε.

Ο ψευδοκώδικας του αλγόριθμου GRASP ο οποίος περιγράφηκε προηγουμένως είναι ο εξής:

Ανάγνωση Δεδομένων

**Διαδικασία** άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης

Αρχικοποίηση

$$c(s^*) = \infty$$

**Repeat**

Κατασκευή μιας αρχικής λύσης  $s$

Εφαρμογή τοπικής αναζήτησης στην  $s$

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

$$s^* = s$$

**Endif**

**Until** για όσο το κριτήριο τερματισμού δεν ικανοποιείται

**Επέστρεψε** τη βέλτιστη λύση ( $s^*$ )



## 4.2 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΚΑΙ ΛΥΣΗΣ ΤΟΥ

Αρχικά, τα δεδομένα που παρέχονται αποτελούνται από πελάτες-κόμβους οι οποίοι βρίσκονται σε θέσεις που δίνονται από συντεταγμένες σε άξονες  $X,Y$ . Κάθε πελάτης έχει και μία συγκεκριμένη ζήτηση η οποία είναι σταθερή για κάθε χρονική στιγμή. Η ζήτηση πρέπει να καλυφθεί στο σύνολό της μέσα στο χρονικό ορίζοντα και κάθε χρονική στιγμή  $t$  δεν επιτρέπεται η αποθηκευμένη ποσότητα του πελάτη να γίνει μικρότερη του 0. Ο κόμβος νούμερο 1 αποτελεί την αποθήκη του κάθε σετ δεδομένων, η οποία είναι και το σημείο εκκίνησης και τερματισμού της κάθε διαδρομής και δεν έχει κάποια ζήτηση ή περιορισμό χωρητικότητας καθώς επίσης δε λαμβάνεται υπόψη ως πελάτης. Ανάλογα το σετ δεδομένων υπάρχει και διαφορετικός αριθμός διαθέσιμων οχημάτων και καθορισμένης χωρητικότητας, ίδια για κάθε όχημα σε κάθε σετ. Ο χρονικός ορίζοντας είναι ίδιος σε κάθε σετ και έχει διάρκεια  $T_{Hor}=7$ , δηλαδή για κάθε  $t = (1,...,T_{Hor})$  υποδηλώνεται και μία μέρα της εβδομάδας. Όπως αναφέρθηκε προηγουμένως το πρόβλημα είναι σύνθετο. Εκτός από την εύρεση των συντομότερων διαδρομών, σκοπός αποτελεί και η εύρεση της κατάλληλης χρονικής στιγμής για την αποθεματοποίηση ενός πελάτη, αλλά και το ποσό αυτής, κάθε φορά μέσα στη διαδρομή. Είναι αντιληπτό ότι η χρήση απλών αλγορίθμων ελαχιστοποίησης κόστους μέσω μείωσης του κόστους διαδρομών δεν θα δώσει τα επιθυμητά αποτελέσματα. Για αυτό θα χρησιμοποιηθούν και οι αλγόριθμοι τοπικής αναζήτησης που δημιουργήθηκαν και χρησιμοποιήθηκαν στη δημοσίευση που αναφέρθηκε προηγουμένως οι οποίοι στοχεύουν συνδυαστικά το κόστος αποθεματοποίησης και αυτό των διαδρομών. Ξεκινώντας θα δημιουργηθεί μία αρχική λύση με χρήση της Λίστας Περιορισμού και αφού θα έχουμε πάρει τις αρχικές διαδρομές και κόστη, τότε θα εφαρμοστούν οι αλγόριθμοι τοπικής αναζήτησης

προκειμένου να μειωθούν τα κόστη δρομολόγησης και αποθεματοποίησης. Το πρόγραμμα θα τρέξει για μέγεθος Λίστας Περιορισμού Υποψηφίων ίσο με 4, 5 και ίσο με το σύνολο των πελατών. Παρακάτω περιγράφονται αναλυτικά τα βήματα και τα κομμάτια του προγράμματος.

#### 4.2.1 ΑΡΧΙΚΗ ΛΥΣΗ

Η αρχική λύση προκύπτει με έναν συνδυαστικό τρόπο. Αρχικά ελέγχονται πόσοι και εφόσον υπάρχουν, ποιοι πελάτες δεν θα έχουν απόθεμα προκειμένου να λειτουργήσουν την χρονική στιγμή  $t$ . Οι πελάτες δίχως καθόλου απόθεμα μπαίνουν σε μια «δεξαμενή». Από εκεί, γίνεται έλεγχος για το όχημα  $k$ , για έναν συγκεκριμένο αριθμό πελατών οι οποίοι είναι οι πιο κοντινοί πελάτες από τον κόμβο στον οποίο βρίσκεται το όχημα, δηλαδή δημιουργείται μια Λίστα Περιορισμού των Υποψηφίων. Η επιλογή του επόμενου πελάτη γίνεται με τυχαίο και ισοπίθανο τρόπο για κάθε υποψήφιο της λίστας. Εφόσον μπορούμε να του αποδώσουμε ποσότητα ίση με μία φορά της ζήτησής του τότε ο συγκεκριμένος πελάτης επισκέπτεται από το όχημα  $k$ , καλύπτεται η ζήτησή του τη χρονική στιγμή  $t$  και γίνεται ο έλεγχος για το επόμενο όχημα. Τώρα θα ξαναγίνει έλεγχος για το ποιοι δεν έχουν εμπόρευμα για να λειτουργήσουν τη χρονική στιγμή  $t$ , χωρίς να υπάρχει αυτή τη φορά ο προηγούμενος πελάτης. Ο έλεγχος τώρα αφορά επίσκεψη από το επόμενο όχημα  $k$ . Με αυτόν τον τρόπο εξασφαλίζεται ότι τα οχήματά μας ξεκινάνε ταυτόχρονα από την αποθήκη. Η συγκεκριμένη διαδικασία επαναλαμβάνεται για κάθε χρονική στιγμή  $t$  του χρονικού ορίζοντα στην αρχή κάθε χρονικής στιγμής και αν τύχει να μην μπορεί να καλυφθεί η ζήτηση τότε το όχημα επιστρέφει στην αποθήκη. Αφότου η συγκεκριμένη δεξαμενή αδειάσει και τα οχήματα δεν έχουν επιστρέψει στην αποθήκη τότε λαμβάνει χώρα η παρακάτω

διαδικασία. Το κάθε όχημα του στόλου μας, έχει επισκεφθεί κάποιους πελάτες και κάποιοι ενδεχομένως να μην έχουν λάβει κάποια επίσκεψη ακόμα. Οι πελάτες τώρα ελέγχονται πάλι ένας για το κάθε όχημα ξεχωριστά.

Οι έλεγχοι που γίνονται είναι οι εξής:

- Αν έχει γίνει ήδη μια επίσκεψη από κάποιο άλλο όχημα.
- Αν έχουν γίνει συνολικά 7 επισκέψεις. Παραπάνω απαγορεύεται
- Αν έχει λάβει ποσότητα συνολικά ίση με συνολική του ζήτηση.

Οι πελάτες που τηρούν αυτές τις προϋποθέσεις μπαίνουν σε μια λίστα για το κάθε όχημα. Η λίστα αυτή ανανεώνεται σε κάθε επανάληψη της υπάρχουσας χρονικής στιγμής σύμφωνα με το ποιοι πελάτες λαμβάνουν επίσκεψη και από ποιο όχημα. Η επιλογή πλέον γίνεται τυχαία και ισοπίθανα καθώς τα κόστη είναι πιο σύνθετα να υπολογιστούν. Η διαδικασία υπολογισμού του κόστους θα εξηγηθεί αναλυτικά σε παρακάτω παράγραφο. Η όλη διαδικασία σταματάει για μία χρονική στιγμή, όταν όλα τα οχήματα επιστρέψουν στην αποθήκη.

Για παράδειγμα, έστω ο πελάτης 2 για στόλο 3 οχημάτων. Έστω ότι ισχύουν οι προϋποθέσεις και μπαίνει στη λίστα του πρώτου οχήματος αλλά δεν επισκέπτεται από αυτό. Έστω ότι ισχύουν οι προϋποθέσεις και μπαίνει στη λίστα του δεύτερου οχήματος αλλά δεν λαμβάνει επίσκεψη από αυτό. Έστω ότι ισχύουν και στο 3<sup>ο</sup> όχημα οι προϋποθέσεις και τελικά ο πελάτης 2 λαμβάνει επίσκεψη από το 3<sup>ο</sup> όχημα. Στις επόμενες επαναλήψεις για την υπάρχουσα χρονική στιγμή

ο πελάτης 2 θα είναι πλέον μονάχα στη λίστα υποψηφίων του 3<sup>ου</sup> οχήματος.

Η εισαγωγή ενός πελάτη σε μια διαδρομή  $rt_k^t$  επιφέρει και ένα κόστος το οποίο υπολογίζεται όπως φαίνεται παρακάτω:

- Αν ο πελάτης που έτυχε έχει ήδη μπει σε κάποια διαδρομή της χρονικής στιγμής  $t$ , τότε όπως αναφέρθηκε, μπορεί να ξαναμπεί μόνο στην ίδια διαδρομή. Επειδή έχουμε αναφέρει πως κάθε χρονική στιγμή μπορούμε να επισκεφτούμε μονάχα μία φορά τον κάθε πελάτη την κάθε χρονική στιγμή, πρέπει να κατανοήσουμε ότι εφόσον βρεθεί στη διαδρομή μας 2 ή παραπάνω φορές ένας πελάτης, οι επιπλέον φορές που βρίσκεται στη διαδρομή μας, αποτελούν **εικονικές επισκέψεις**. Αυτή η εικονική επίσκεψη δεν αυξάνει το κόστος δρομολόγησης, καθώς η κανονική επίσκεψη έχει γίνει ήδη. Ωστόσο το συνολικό κόστος θα υποστεί μία αύξηση λόγω αύξησης του κόστους αποθεματοποίησης, καθώς αφήνουμε επιπλέον προϊόν. Έστω  $AD_i$  είναι η ποσότητα του πελάτη  $i$  που του έχουμε ήδη αφήσει στη διαδρομή  $rt_k^t$ . Η εικονική επίσκεψη οδηγεί σε αύξηση ίση με  $(AD_i + d_i/2) \cdot h_i$ . Όπου  $d_i$  η ζήτηση του πελάτη  $i$  και  $h_i$  το κόστος που υπόκειται κάθε μονάδα προϊόντος στην αποθήκη του πελάτη  $i$ .
- Αν ο πελάτης που έτυχε, είναι να μπει για πρώτη φορά στη διαδρομή  $rt_k^t$  τότε υπάρχει κόστος δρομολόγησης αλλά και αποθεματοποίησης. Το κόστος δρομολόγησης είναι απλά το κόστος του να πάει το όχημα  $k$  από τη θέση που βρίσκεται, στον πελάτη που έτυχε  $i$ , ενώ το κόστος αποθεματοποίησης ισούται με  $d_i \cdot h_i/2$ . Όπου  $d_i$  η ζήτηση του πελάτη  $i$  και  $h_i$  το κόστος που υπόκειται κάθε μονάδα προϊόντος στην αποθήκη του πελάτη  $i$ .

Προγραμματιστικά, όταν συμβαίνει μια εικονική επίσκεψη, η θέση του οχήματος  $k$ , δεν αλλάζει και παρέμενε σταθερή. Επειδή όμως κοιτάμε κάθε όχημα, επαναληπτικά, προκειμένου να θεωρηθούν οι κινήσεις που γίνονται, ταυτόχρονες, θα έμπαινε σαν “επόμενη” θέση στη διαδρομή, ο τελευταίος πελάτης ο οποίος έλαβε προϊόν για πρώτη φορά στη συγκεκριμένη διαδρομή. Προκειμένου να δούμε πόσες επισκέψεις έγιναν στον κάθε πελάτη, δηλαδή πόση ποσότητα έλαβε κατά τη δημιουργία της αρχικής λύσης, δημιουργήθηκε ένας πίνακας προκειμένου να κρατάμε τον αριθμό των εικονικών επισκέψεων και στη συνέχεια σβήνουμε από τις διαδρομές τους διπλότυπους πελάτες.

Παρακάτω φαίνεται ένα παράδειγμα 3 οχημάτων, 30 πελατών σχετικά με την παραπάνω διαδικασία.

$val(:, :, 2) =$

1	2	15	28	7	26	27	18	9	21	24	24	10	1	1	0	0	0
1	20	13	14	5	29	12	22	30	<b>16</b>	<b>16</b>	<b>16</b>	<b>16</b>	31	1	0	0	0
1	8	17	19	3	11	25	23	6	6	1	1	1	1	1	0	0	0

Πίνακας Διαδρομών Ο πελάτης 16 φαίνεται ότι μπαίνει 4 φορές στη διαδρομή του 2ου οχήματος

Ο αριθμός 2 στο  $val(:, :, 2)$  δείχνει τη χρονική στιγμή  $t=2$

Κάθε σειρά αποτελεί και διαδρομή διαφορετικού οχήματος.

1	2	3	4	5	6	7	8	9	10	11	<b>12</b>	13	14	15	<b>16</b>	17	18	19	20	21	<b>22</b>	23	24	25	26	27	28	29	<b>30</b>	31
0	1	2	0	1	1	1	1	1	1	1	<b>2</b>	1	1	1	<b>1</b>	1	1	1	1	2	<b>2</b>	1	1	1	1	1	1	1	<b>2</b>	1

Πίνακας Εικονικών Επισκέψεων. Στον συγκεκριμένο πίνακα φαίνεται πως ο πελάτης 16 στην πραγματικότητα του έγινε μόνο μία επίσκεψη και όχι 4 εικονικές επισκέψεις. Οι επιπλέον 3 έγινες στους πελάτες 12, 22 και 30 που προηγούνταν του 16.

val(:, :, 2) =

1	2	15	28	7	26	27	18	9	21	24	10	1	1	1	0	0	0
1	20	13	14	5	29	12	22	30	16	31	1	1	1	1	0	0	0
1	8	17	19	3	11	25	23	6	1	1	1	1	1	1	0	0	0

Πίνακας Διαδρομών Πως φαίνονται οι διαδρομές στο τέλος της αρχικής λύσης

#### 4.2.2 ΤΟΠΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΑΠΟΘΕΜΑΤΟΠΟΙΗΣΗΣ

Στην εργασία των Emmanouil E. Zachariadis, Christos D. Tarantilis, Chris T. Kiranoudis [3] γίνεται λόγος για τη δημιουργία 2 αλγορίθμων τοπικής αναζήτησης οι οποίοι στοχεύουν και οι 2 συνδυαστικά στη μείωση του inventory cost αλλά και του routing cost. Ο πρώτος αλγόριθμος, προσθέτει στο χρονικό πρόγραμμα της αρχικής μας λύσης μία επιπλέον επίσκεψη αναπλήρωσης αποθέματος ( *insertion of replenishment point* ) σε κάποιον πελάτη, ενώ ο δεύτερος αλγόριθμος αφαιρεί μια επίσκεψη από το πρόγραμμα κάποιου πελάτη ( *removal of replenishment point* ).

#### 4.2.2.1 Πρόσθεση επίσκεψης αναπλήρωσης αποθέματος (Insertion of Replenishment Point)

Ο αλγόριθμος εισαγωγής αποτελείται από την προσθήκη μιας νέας αναπλήρωσης σε κάποια χρονική μονάδα  $t_{ins}$  του χρονοδιαγράμματος του πελάτη  $v_i$ . Η νέα αυτή αναπλήρωση εκχωρείται στη διαδρομή  $rt_k^{t_{ins}}$ . Προκειμένου να είμαστε σε θέση, ώστε να γίνει αυτή η κίνηση, πρέπει να ισχύουν δύο προϋποθέσεις:

- (α) ο πελάτης δεν εξυπηρετείται σε καμία διαδρομή τη χρονική στιγμή  $t_{ins}$ ,
- (β) Το όχημα  $k$  της διαδρομής  $rt_k^{t_{ins}}$  δεν είναι πλήρως φορτωμένο.

Έστω  $S(t)$ , το απόθεμα ενός πελάτη τη στιγμή  $t$ . Ο χειριστής εισαγωγής χρησιμοποιεί έναν άπληστο μηχανισμό για την αναδιάταξη των αναπληρώσεων: κινούμενοι προς τα αριστερά του  $t_{ins}$ , η πρώτη χρονική στιγμή για την οποία το απόθεμα του πελάτη είναι μηδέν, ορίζεται ως  $t_{base}$ ,  $S(t_{base}) = 0$ . Στη συνέχεια, γίνεται έλεγχος στο χρονικό διάστημα  $(t_{base}, t_{ins}]$ , για να βρεθεί η στιγμή που αντιστοιχεί στο χαμηλότερο επίπεδο αποθεμάτων, και συμβολίζεται με  $t_{target}$ . Εάν η υπολειπόμενη χωρητικότητα του οχήματος  $k$  στη διαδρομή  $rt_k^{t_{ins}}$  δεν είναι χαμηλότερη από τη  $S(t_{target})$ , μια ποσότητα  $quantMoved = S(t_{target})$ , αφαιρείται από την αναπλήρωση του  $t_{base}$  και εκχωρείται στο σημείο αναπλήρωσης των  $t_{ins}$ . Από την άλλη, αν η υπολειπόμενη χωρητικότητα του  $k$  της  $rt_k^{t_{ins}}$  είναι μικρότερη από το  $S(t_{target})$ , η ποσότητα  $quantMoved$  παίρνει τιμή ίση με την υπολειπόμενη χωρητικότητα του οχήματος. Με το να προσθέσουμε μία νέα αναπλήρωση στο υπάρχον πρόγραμμα αναπληρώσεων ενός

πελάτη, τροποποιείται τόσο το κόστος αποθεματοποίησης όσο και η σειρά επισκέψεων των πελατών μιας διαδρομής, με αποτέλεσμα να επηρεάζεται και το κόστος δρομολόγησης. Το κόστος δρομολόγησης αυξάνεται σύμφωνα με την εισαγωγή του ελάχιστου κόστους (minimum insertion cost algorithm) του  $v_i$  στη διαδρομή  $rt_k^{t_{ins}}$ , ενώ το κόστος αποθεματοποίησης μειώνεται κατά  $h_i \cdot quantMoved \cdot |t_{ins} - t_{base}|$ , όπου  $|t_{ins} - t_{base}|$  υποδηλώνει την απόσταση μεταξύ  $t_{base}$  και  $t_{ins}$ , ξεκινώντας από το προηγούμενο και κινούμενοι κατά μήκος του χρονικού ορίζοντα.

#### 4.2.2.2 Αφαίρεση επίσκεψης αναπλήρωσης (Removal of Replenishment Point)

Ο αλγόριθμος αφαίρεσης αναπληρώσεων διαγράφει μια αναπλήρωση αποθέματος από τη στιγμή  $t_{rem}$  του χρονοδιαγράμματος, του πελάτη  $v_i$ . Η αφαιρεθείσα ποσότητα  $q_{i_{t_{rem}}}$  δίνεται σε μία υπάρχουσα επίσκεψη του πελάτη  $v_i$  στη στιγμή  $t_{target}$ , δεδομένου ότι η ποσότητα  $q_{i_{t_{rem}}}$  δεν υπερβαίνει το υπόλοιπο χωρητικότητας αυτής της διαδρομής. Προκειμένου να μπορούμε να χρησιμοποιήσουμε αυτόν τον αλγόριθμο, ο πελάτης πρέπει να έχει λάβει επίσκεψη, τουλάχιστον 2 φορές κατά τη διάρκεια του χρονικού ορίζοντα. Με τη χρήση αυτού του αλγορίθμου μειώνεται το κόστος μεταφοράς, καθώς ο πελάτης  $v_i$  εξαιλείται από το δρομολόγιο που τον εξυπηρετεί τη στιγμή  $t_{rem}$ . Η επίδραση στο κόστος αποθεματοποίησης εξαρτάται από τις λεπτομέρειες της μετακίνησης και περιγράφεται στη συνέχεια. Η εφαρμογή του αλγορίθμου *removal of replenishment point* υπαγορεύεται από έναν άπληστο μηχανισμό: ξεκινώντας από το σημείο  $t_{rem}$  και προχωρώντας προς τα εμπρός, κατά μήκος του άξονα του



χρόνου, ελέγχονται όλες στιγμές αναπλήρωσης αποθέματος του πελάτη  $v_i$ . Έστω  $t_{base}$  δείχνει την πρώτη στιγμή αναπλήρωσης για την οποία ισχύει  $sb_{it_{base}} - q_{it_{rem}} < 0$ .

Στη συνέχεια, ξεκινώντας από τη χρονική μονάδα  $t_{base}$  και προχωρώντας προς την αντίθετη κατεύθυνση, η χρονική στιγμή  $t_{target}$  ορίζεται ως η χρονική στιγμή που συναντάμε την πρώτη αναπλήρωση. Δύο είναι τα ενδεχόμενα που μπορούν να προκύψουν με τη χρήση αυτού του αλγορίθμου:

a) *Μετατόπιση προς τα πίσω της αφαιρεθείσας αναπλήρωσης*

Εάν η στιγμή  $t_{target}$  συμπίπτει με τη  $t_{rem}$ , τότε η  $t_{target}$  τίθεται ίση με τη χρονική στιγμή της πρώτης αναπλήρωσης που πραγματοποιήθηκε πριν από την  $t_{rem}$ . Στην περίπτωση αυτή, όπου η  $t_{target}$  προηγείται της  $t_{rem}$ , το κόστος απογραφής του συστήματος αυξάνεται κατά:

$$h_i \cdot q_{i\_t\_rem} \cdot |t_{rem} - t_{target}|,$$

όπου  $|t_{rem} - t_{target}|$  ισούται με τα χρονικά διαστήματα μεταξύ  $t_{target}$  και  $t_{rem}$ .

Εάν  $sb_{it_{rem}} = 0$ , τότε εμφανίζεται μια πρόσθετη επίδραση στο κόστος απογραφής. Έστω  $t_{sl}$  είναι η στιγμή εξυπηρέτησης με τη δεύτερη χαμηλότερη τιμή του αποθέματος πριν δεχτεί την ποσότητα αναπλήρωσης, τότε το κόστος αποθεματοποίησης μειώνεται κατά  $sb_{it_{sl}} \cdot T\_Hor$ .

Αυτή η μείωση του κόστους οφείλεται στους περιορισμούς (12) οι οποίοι επιβάλλουν το ελάχιστο επίπεδο αποθέματος να είναι το 0.

b) *Μετατόπιση προς τα εμπρός της αφαιρεθείσας αναπλήρωσης*

Εάν η στιγμή  $t_{target}$  δεν συμπίπτει με την  $t_{rem}$ , τότε η ποσότητα που αφαιρείται από τη  $t_{rem}$  μετατοπίζεται προς τα εμπρός, στη χρονική στιγμή της αναπλήρωσης  $t_{target}$ . Στην αυτή την περίπτωση, το κόστος αποθεματοποίησης μειώνεται κατά:

$$h_i \cdot q_{i_{t_{rem}}} \cdot |t_{target} - t_{rem}|,$$

όπου  $|t_{target} - t_{rem}|$  είναι η απόσταση μεταξύ  $t_{rem}$  και  $t_{target}$ , ξεκινώντας από την πρώτη και κινούμενη κατά μήκος του χρονικού ορίζοντα.

4.2.2.3 *Αλγόριθμοι τοπικής αναζήτησης μείωσης κόστους δρομολόγησης.*

Οι αλγόριθμοι που χρησιμοποιήθηκαν είναι οι 2-opt, 1-1 exchange και 1-0 relocate όπως περιγράφηκαν στο αντίστοιχο κεφάλαιο. Η χρήση τους έγινε παράλληλα με τη χρήση των αλγορίθμων που μόλις περιγράφηκαν αλλά και ξεχωριστά, μετά το τέλος τους, ανάλογα το σετ δεδομένων που γινόταν χρήση. Κάθε συνδυασμός τους σε διαφορετικό dataset έδινε και διαφορετικά αποτελέσματα.

#### 4.2.2.4 Τρόπος λειτουργίας προγράμματος και δομή

Το πρόγραμμα κάθε φορά που το τρέχουμε κάνει 10 επαναλήψεις και κάθε φορά συγκρίνει τη λύση που βρίσκει στην εκάστοτε επανάληψη με την καλύτερη που υπάρχει μέχρι εκείνη τη στιγμή. Αν βρεθεί καλύτερη λύση, τότε κρατάει τα κόστη δρομολόγησης και αποθεματοποίησης, καθώς και το πρόγραμμα δρομολογήσεων που δημιουργήθηκε. Στο τέλος βγάζει έναν μέσο όρο συνολικού κόστους, κόστος δρομολογήσεων και αποθεματοποίησης. Συνολικά τρέχουμε το πρόγραμμα για κάθε σετ δεδομένων 10 φορές και κρατάμε τα καλύτερα κόστη από την κάθε φορά και στη συνέχεια βγάζουμε τον μέσο όρο από τα καλύτερα αποτελέσματα που πήραμε.

Όσον αφορά τη δομή του κώδικα, αρχικά δημιουργείται μια αρχική λύση με τον τρόπο που περιεγράφηκε προηγουμένως. Αφού ολοκληρωθεί αυτή η διαδικασία, έχουμε κρατήσει τις διαδρομές των οχημάτων σε κάθε χρονική στιγμή του χρονικού ορίζοντα, τις ποσότητες που τροφοδοτούμε τον κάθε πελάτη σε κάθε επίσκεψη, πόση ποσότητα παραμένει διαθέσιμη σε κάθε όχημα κάθε χρονική στιγμή και πόση ποσότητα έχει διαθέσιμη στην αρχή της κάθε χρονικής στιγμής κάθε πελάτης. Οι ποσότητες που τροφοδοτούμε τον κάθε πελάτη δίνεται από το γινόμενο των εικονικών επισκέψεων επί τη ζήτηση του πελάτη.

Στη συνέχεια επιτελείται μία επαναληπτική διαδικασία βελτιστοποίησης η οποία αποτελείται από τους αλγορίθμους που περιεγράφηκαν στο 4.2.2.1 και 4.2.2.2 συνδυαστικά με τον αλγόριθμο 2-opt και 1-0 relocate. Η επαναληπτική διαδικασία σταματάει όταν οι αλγόριθμοι σύνθετου χειρισμού των επαναλήψεων δεν κάνουν

βελτιώσεις ενώ έχουν εκτελεστεί έναν συγκεκριμένο αριθμό και οι 2 ταυτοχρόνως, ο οποίος αριθμός έχει προκαθοριστεί εξ αρχής. Εφόσον βελτιώνουν τη λύση μας, έστω μία φορά – μέσα στον συγκεκριμένο αριθμό εκτελέσεων - είτε ο ένας είτε ο άλλος, η διαδικασία θα ξαναεκτελεστεί. Οι 2 συγκεκριμένοι αλγόριθμοι σε κάθε τους επανάληψη ελέγχουν όλους τους πελάτες – οι οποίοι τηρούν τις προϋποθέσεις – και όλες τις πιθανές αλλαγές που μπορούν να κάνουν. Στο τέλος θα κάνουν μόνο μια αλλαγή στο πρόγραμμά μας, για έναν πελάτη, αυτήν η οποία θα βελτιώσει το κόστος του συστήματος στο μέγιστο βαθμό.

Σε πρώτη φάση εκτελούνται με τη σειρά και επαναληπτικά οι αλγόριθμοι πρόσθεσης και αφαίρεσης επισκέψεων έως ότου να μη δίνουν περαιτέρω βελτίωση. Έπειτα εκτελείται ο 2-opt αλγόριθμος για συγκεκριμένο αριθμό επαναλήψεων. Σε ορισμένα σετ δεδομένων χρησιμοποιήθηκε, αμέσως μετά τον 2-opt, ο 1-0 relocate. Εφόσον οι αλγόριθμοι πρόσθεσης και αφαίρεσης επισκέψεων βελτίωσαν έστω και μία φορά τη λύση, όλη η διαδικασία θα ξαναεκτελεστεί με τη σειρά και τον τρόπο που περιεγράφηκε. Στη συνέχεια σταματάει η επαναληπτική διαδικασία και εκτελούνται με τη σειρά οι αλγόριθμοι τοπικής αναζήτησης 2-opt, 1-1 exchange και 1-0 relocate για προκαθορισμένο αριθμό επαναλήψεων.

Αυτή η διαδικασία αποτέλεσε μια επανάληψη του προγράμματός που κατασκευάστηκε και δίνει μια πιθανή λύση. Συνολικά θα τρέξει 10 επαναλήψεις και θα μας δώσει 10 πιθανές λύσεις. Από αυτές κρατάμε την καλύτερη και βγάζουμε έναν μέσο όρο κόστους από αυτές τις επαναλήψεις.

Σε περίπτωση που το πρόγραμμά μας πέσει σε κάποιον περιορισμό, ξεπερνώντας τα όρια των αποθεμάτων, είναι αναγκαίο να τρέξουμε το πρόγραμμα από την αρχή, κρατώντας ωστόσο την καλύτερη λύση σε όσες επαναλήψεις έκανε και συγκρίνοντάς τη με την καλύτερη από τις επόμενες συμπληρωματικές επαναλήψεις.

#### 4.3 ΨΕΥΔΟΚΩΔΙΚΑΣ

##### **GreedyRandomizedAdaptiveSearchProcedure**

Ανάγνωση Δεδομένων

Αρχικοποίηση

$c(s^*) = \infty$

**Repeat**

Κατασκευή Αρχικής Λύσης

**Repeat**

Εφαρμογή Αλγορίθμων R.I. & R.R. \*

Εφαρμογή Αλγορίθμων Τοπικής Αναζήτησης

**Until** όσο οι R.I. και R.R. μειώνουν το κόστος

**Repeat**

Εφαρμογή Αλγορίθμων Τοπικής Αναζήτησης

**Until** το κριτήριο τερματισμού δεν ικανοποιείται

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

$s^* = s$

**Endif**

**Until** το κριτήριο τερματισμού δεν ικανοποιείται

**Επέστρεψε** τη βέλτιστη λύση ( $s^*$ )

\* R.I, R.R. = Replenishment Insertion, Replenishment Removal

## ΚΕΦΑΛΑΙΟ 5

### ΕΞΑΓΩΓΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΚΑΙ ΣΧΟΛΙΑΣΜΟΣ

Στο συγκεκριμένο κεφάλαιο θα γίνει παρουσίαση των αποτελεσμάτων του προγράμματος που δημιουργήθηκε και σύγκριση με αυτά από την εργασία των Emmanouil E. Zachariadis, Christos D. Tarantilis, Chris T. Kiranoudis [3], καθώς και στη συνέχεια θα παρουσιαστούν τα συμπεράσματα που προκύπτουν από τα αποτελέσματα και τις συγκρίσεις.

Ο κώδικας για το συγκεκριμένο θέμα γράφτηκε σε προγραμματιστικό περιβάλλον MATLAB. Λόγω των μεγάλων χρόνων που χρειάζονταν προκειμένου να τρέξει το πρόγραμμα, έγινε δοκιμή του προγράμματος σε σετ δεδομένων με 30,50,70,90 και 110 πελάτες. Ενδεικτικά αναφέρεται πως προκειμένου να τρέξει μία φορά για 30 πελάτες χρειαζόταν 28-30 λεπτά και όσο αυξάνονταν οι πελάτες και τα διαθέσιμα οχήματα, αφετέρου και οι συγκρίσεις, τόσο περισσότερο αυξάνονταν και ο χρόνος εκτέλεσης του προγράμματός μας. Από την άλλη, το πρόγραμμα που γίνεται σύγκριση, γράφτηκε σε C++ και ενδεικτικά για τους 30 πελάτες, εκτελούνταν σε χρόνο 170 sec κατά μ.ο, ενώ για το μεγαλύτερο σετ δεδομένων - 210 πελατών - εκτελούνταν σε χρόνο 2298 sec κατά μέσο όρο.

Στα αποτελέσματα παρουσιάζονται το κόστος αποθεματοποίησης (inventory cost), το κόστος δρομολόγησης (routing cost) και το συνολικό κόστος που προκύπτει από το άθροισμα τους (total cost). Όπως αναφέρθηκε, κάθε φορά που έτρεχε το πρόγραμμα εκτελούσε 10

επαναλήψεις και εμφάνιζε 10 πιθανές λύσεις. Από αυτές κρατούσαμε την καλύτερη και το πρόγραμμα εκτελέστηκε 10 φορές για κάθε αριθμό πελατών. Επίσης παρουσιάζονται τα δεδομένα των διαθέσιμων οχημάτων.

## 5.1 ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ

Ο αλγόριθμος που παρουσιάστηκε στο κεφάλαιο 4 έχει υποστεί την παρακάτω παραμετροποίηση. Καταρχάς μία εκτέλεση του αλγορίθμου αποτελείται από 10 πιθανές λύσεις του προβλήματος και από αυτές κρατάμε την καλύτερη. Στη Λίστα Περιορισμού των υποψηφίων στην πρώτη φάση υπάρχουν πιθανοί πελάτες ίσοι με 4, 5 ή το σύνολο των πελατών που υπάρχουν στο παράδειγμα αναφοράς, ενώ στη συνέχεια όταν όλοι οι πελάτες λάβουν τουλάχιστον μία εικονική επίσκεψη, ο κάθε πελάτης τοποθετείται σε μία «δεξαμενή» υποψήφιων πελατών ανάλογα με το όχημα που τον επισκέφτηκε. Και στις 2 φάσεις το ενδεχόμενο επίσκεψης ενός πελάτη είναι ισοπίθανο. Οι αλγόριθμοι 4.2.2.1 και 4.2.2.2 τρέχουν ταυτόχρονα έως ότου δεν μας δώσουν καλύτερα αποτελέσματα τουλάχιστον 5 φορές, έκαστος. Ο 2-opt αλγόριθμος ο οποίος βρίσκεται αμέσως μετά τον 4.2.2.2 και εντός της επαναληπτικής διαδικασίας των 4.2.2.1 και 4.2.2.2 εκτελείται κάθε φορά 250 φορές. Οι τοπικοί αλγόριθμοι στο 4.2.2.3 εκτελούνται από 1000 φορές.

## 5.2 ΑΠΟΤΕΛΕΣΜΑΤΑ

### 5.2.1 ΠΑΡΑΔΕΙΓΜΑ ΑΝΑΦΟΡΑΣ 30 ΠΕΛΑΤΩΝ

Αριθμός Πελατών 30	Διαθέσιμα Οχήματα 3	Χωρητικότητα 300	Χρονικός Ορίζοντας 7
-----------------------	------------------------	---------------------	-------------------------

Εικόνα 3 Δεδομένα οχημάτων για 30 πελάτες

ΜΕΓΕΘΟΣ=n			
Τρεξίματα/Αποτελέσματα	Συνολικό Κόστος	Κόστος Δρομολόγησης	Κόστος Αποθεματοποίησης
1ο τρέξιμο	39833,04	27771,54	12061,5
2ο τρέξιμο	37779,31	26630,81	11148,5
3ο τρέξιμο	39212	27216,5	11995,5
4ο τρέξιμο	37387,93	26143,43	11244,5
5ο τρέξιμο	38632,9	25911,4	12721,5
6ο τρέξιμο	39680,8	27308,3	12372,5
7ο τρέξιμο	39293,82	27365,32	11928,5
<b>8ο τρέξιμο</b>	<b>35068,87</b>	23432,37	11636,5
9ο τρέξιμο	38326,22	26349,72	11976,5
10ο τρέξιμο	38636,83	27365,33	11271,5
<b>Μ.Ο.</b>	<b>38385,172</b>	<b>26549,472</b>	<b>11835,7</b>

Πίνακας αποτελεσμάτων για 30 πελάτες και μέγεθος n



ΜΕΓΕΘΟΣ=4			
Τρεξίματα/Αποτελέσματα	Συνολικό Κόστος	Κόστος Δρομολόγησης	Κόστος Αποθεματοποίησης
1ο τρέξιμο	37914,95	26354,45	11560,5
2ο τρέξιμο	38887,02	27332,52	11554,5
3ο τρέξιμο	38877,06	27591,56	11285,5
4ο τρέξιμο	39742,63	29308,13	10434,5
5ο τρέξιμο	39603,94	28109,44	11494,5
6ο τρέξιμο	<b>36700,36</b>	24537,86	12162,5
7ο τρέξιμο	39424,47	28504,97	10919,5
8ο τρέξιμο	38999,8	27709,3	11290,5
9ο τρέξιμο	39387,61	27840,11	11547,5
10ο τρέξιμο	39155,1	28084,6	11070,5
<b>Μ.Ο.</b>	<b>38869,294</b>	<b>27537,294</b>	<b>11332</b>

Πίνακας αποτελεσμάτων για 30 πελάτες και μέγεθος 4

ΜΕΓΕΘΟΣ=5			
Τρεξίματα/Αποτελέσματα	Συνολικό Κόστος	Κόστος Δρομολόγησης	Κόστος Αποθεματοποίησης
1ο τρέξιμο	41068,42	28352,92	12715,5
2ο τρέξιμο	39337,24	28191,74	11145,5
3ο τρέξιμο	37726,4	25649,9	12076,5
4ο τρέξιμο	<b>35843,54</b>	24280,04	11563,5
5ο τρέξιμο	38247,87	26448,37	11799,5
6ο τρέξιμο	37188,57	25653,07	11535,5
7ο τρέξιμο	38058,69	26453,19	11605,5
8ο τρέξιμο	37522,02	26393,52	11128,5
9ο τρέξιμο	38907,51	27346,01	11561,5
10ο τρέξιμο	37176,84	25962,34	11214,5
<b>Μ.Ο.</b>	<b>38107,71</b>	<b>26473,11</b>	<b>11634,6</b>

Πίνακας αποτελεσμάτων για 30 πελάτες και μέγεθος 5

## 5.2.2 ΠΑΡΑΔΕΙΓΜΑ ΑΝΑΦΟΡΑΣ 50 ΠΕΛΑΤΩΝ

Αριθμός Πελατών 50	Διαθέσιμα Οχήματα 4	Χωρητικότητα 500	Χρονικός Ορίζοντας 7
-----------------------	------------------------	---------------------	-------------------------

Εικόνα 4 Δεδομένα οχημάτων για σετ 50 πελατών

ΜΕΓΕΘΟΣ=n			
Τρεξίματα/Αποτελέσματα	Συνολικό Κόστος	Κόστος Δρομολόγησης	Κόστος Αποθεματοποίησης
1ο τρέξιμο	57873,35	36791,35	21082
2ο τρέξιμο	58811,26	37458,26	21353
3ο τρέξιμο	59280,55	37194,55	22086
4ο τρέξιμο	58490,35	36881,35	21609
5ο τρέξιμο	58477,03	37302,03	21175
<b>6ο τρέξιμο</b>	<b>56492,07</b>	35734,07	20758
7ο τρέξιμο	59240,92	37477,92	21763
8ο τρέξιμο	58654,31	37158,31	21496
9ο τρέξιμο	59503,98	37541,98	21962
10ο τρέξιμο	58388,7	37446,7	20942
<b>Μ.Ο.</b>	<b>58521,252</b>	<b>37098,652</b>	<b>21422,6</b>

Πίνακας αποτελεσμάτων για 50 πελάτες και μέγεθος n

ΜΕΓΕΘΟΣ=4			
Τρεξίματα/Αποτελέσματα	Συνολικό Κόστος	Κόστος Δρομολόγησης	Κόστος Αποθεματοποίησης
1ο τρέξιμο	59056,62	38146,62	20910
2ο τρέξιμο	58911,54	37915,54	20996
3ο τρέξιμο	59308,85	38856,85	20452
4ο τρέξιμο	61549,22	37975,22	23574
5ο τρέξιμο	60422,66	38228,66	22194
6ο τρέξιμο	62352,42	41439,42	20913
7ο τρέξιμο	58499,7	37072,7	21427
8ο τρέξιμο	59769,84	39038,84	20731
9ο τρέξιμο	59280,22	38667,22	20613
10ο τρέξιμο	57159,97	35919,97	21240
<b>Μ.Ο.</b>	<b>59631,104</b>	<b>38326,104</b>	<b>21305</b>

Πίνακας αποτελεσμάτων για 50 πελάτες και μέγεθος 4

ΜΕΓΕΘΟΣ=5			
Τρεξίματα/Αποτελέσματα	Συνολικό Κόστος	Κόστος Δρομολόγησης	Κόστος Αποθεματοποίησης
1ο τρέξιμο	58510,17	37460,17	21050
2ο τρέξιμο	57684,43	36971,43	20713
3ο τρέξιμο	57423,62	36401,62	21022
4ο τρέξιμο	58823,26	37680,26	21143
5ο τρέξιμο	58379,9	36668,9	21711
6ο τρέξιμο	57894,4	36889,4	21005
7ο τρέξιμο	58858,3	37797,3	21061
8ο τρέξιμο	59945,66	37775,66	22170
9ο τρέξιμο	58623,8	37690,8	20933
10ο τρέξιμο	58553,71	37767,71	20786
<b>Μ.Ο.</b>	<b>58469,725</b>	<b>37310,325</b>	<b>21159,4</b>

Πίνακας αποτελεσμάτων για 50 πελάτες και μέγεθος 5

### 5.2.3 ΠΑΡΑΔΕΙΓΜΑ ΑΝΑΦΟΡΑΣ 70 ΠΕΛΑΤΩΝ

Αριθμός Πελατών 70	Διαθέσιμα Οχήματα 3	Χωρητικότητα 700	Χρονικός Ορίζοντας 7
-----------------------	------------------------	---------------------	-------------------------

Πίνακας Δεδομένων. Δεδομένα οχημάτων για 70 πελάτες

ΜΕΓΕΘΟΣ=n			
Τρεξίματα/Αποτελέσματα	Συνολικό Κόστος	Κόστος Δρομολόγησης	Κόστος Αποθεματοποίησης
1ο τρέξιμο	62872,66	40282,66	22590
2ο τρέξιμο	61974,53	39487,53	22487
3ο τρέξιμο	62120,94	38736,94	23384
4ο τρέξιμο	62351,75	39186,75	23165
<b>5ο τρέξιμο</b>	<b>61065,93</b>	38376,93	22689
6ο τρέξιμο	61204,72	39372,72	21832
7ο τρέξιμο	61823,05	39162,05	22661
8ο τρέξιμο	61352,57	38155,57	23197
9ο τρέξιμο	61897,68	39210,68	22687
10ο τρέξιμο	62544,91	39339,91	23205
<b>Μ.Ο.</b>	<b>61920,874</b>	<b>39131,174</b>	<b>22789,7</b>

Πίνακας αποτελεσμάτων για 70 πελάτες και μέγεθος n

ΜΕΓΕΘΟΣ=4			
Τρεξίματα/Αποτελέσματα	Συνολικό Κόστος	Κόστος Δρομολόγησης	Κόστος Αποθεματοποίησης
1ο τρέξιμο	62720,45	40079,45	22641
2ο τρέξιμο	65216,72	42889,72	22327
3ο τρέξιμο	63890,61	41503,61	22387
4ο τρέξιμο	<b>62720,45</b>	40079,45	22641
5ο τρέξιμο	64311,13	41878,13	22433
6ο τρέξιμο	64487,88	42697,88	21790
7ο τρέξιμο	63885,56	41178,56	22707
8ο τρέξιμο	63868,5	42100,5	21768
9ο τρέξιμο	64208,66	41686,66	22522
10ο τρέξιμο	64281,34	42321,34	21960
<b>Μ.Ο.</b>	<b>63959,13</b>	<b>41641,53</b>	<b>22317,6</b>

Πίνακας αποτελεσμάτων για 70 πελάτες και μέγεθος 4

ΜΕΓΕΘΟΣ=5			
Τρεξίματα/Αποτελέσματα	Συνολικό Κόστος	Κόστος Δρομολόγησης	Κόστος Αποθεματοποίησης
1ο τρέξιμο	64200,07	41705,07	22495
2ο τρέξιμο	64341,72	42185,72	22156
3ο τρέξιμο	<b>63397,31</b>	40739,31	22658
4ο τρέξιμο	64677,78	43132,78	21545
5ο τρέξιμο	65150,28	42822,28	22328
6ο τρέξιμο	64587,74	42613,74	21974
7ο τρέξιμο	64851,48	42650,48	22201
8ο τρέξιμο	64296,81	42483,81	21813
9ο τρέξιμο	64329,82	41898,82	22431
10ο τρέξιμο	65639,83	43256,83	22383
<b>Μ.Ο.</b>	<b>64547,284</b>	<b>42348,884</b>	<b>22198,4</b>

Πίνακας αποτελεσμάτων για 70 πελάτες και μέγεθος 5

## 5.2.4 ΠΑΡΑΔΕΙΓΜΑ ΑΝΑΦΟΡΑΣ 90 ΠΕΛΑΤΩΝ

Αριθμός Πελατών 90	Διαθέσιμα Οχήματα 4	Χωρητικότητα 800	Χρονικός Ορίζοντας 7
-----------------------	------------------------	---------------------	-------------------------

Πίνακας Δεδομένων. Δεδομένα οχημάτων σε 90 πελάτες

ΜΕΓΕΘΟΣ=n			
Τρεξίματα/Αποτελέσματα	Συνολικό Κόστος	Κόστος Δρομολόγησης	Κόστος Αποθεματοποίησης
1ο τρέξιμο	85345,12	53321,12	32024
2ο τρέξιμο	83734,85	52695,85	31039
3ο τρέξιμο	82718,94	51581,94	31137
4ο τρέξιμο	83742,31	52102,31	31640
<b>5ο τρέξιμο</b>	<b>82133,98</b>	50686,98	31447
6ο τρέξιμο	83754,91	51756,91	31998
7ο τρέξιμο	84263,41	51912,41	32351
8ο τρέξιμο	83985,34	52279,34	31706
9ο τρέξιμο	84763,82	51946,82	32817
10ο τρέξιμο	82648,1	50071,1	32577
<b>Μ.Ο.</b>	<b>83709,078</b>	<b>51835,478</b>	<b>31873,6</b>

Πίνακας αποτελεσμάτων για 90 πελάτες και μέγεθος n

ΜΕΓΕΘΟΣ=4			
Τρεξίματα/Αποτελέσματα	Συνολικό Κόστος	Κόστος Δρομολόγησης	Κόστος Αποθεματοποίησης
1ο τρέξιμο	83087,8	50948,8	32139
2ο τρέξιμο	83033,73	51603,73	31430
3ο τρέξιμο	85805,79	53824,79	31981
4ο τρέξιμο	85934,2	54079,2	31855
5ο τρέξιμο	83833,47	51505,47	32328
6ο τρέξιμο	<b>82061,74</b>	49605,74	32456
7ο τρέξιμο	84033,94	52343,94	31690
8ο τρέξιμο	83914,8	51582,8	32332
9ο τρέξιμο	86801,15	54386,15	32415
10ο τρέξιμο	83940,16	52320,16	31620
<b>Μ.Ο.</b>	<b>84244,678</b>	<b>52220,078</b>	<b>32024,6</b>

Πίνακας αποτελεσμάτων για 90 πελάτες και μέγεθος 4

ΜΕΓΕΘΟΣ=5			
Τρεξίματα/Αποτελέσματα	Συνολικό Κόστος	Κόστος Δρομολόγησης	Κόστος Αποθεματοποίησης
1ο τρέξιμο	83245,31	51113,31	32132
2ο τρέξιμο	85272,4	52937,4	32335
3ο τρέξιμο	84385,63	53025,63	31360
4ο τρέξιμο	82264,84	50334,84	31930
5ο τρέξιμο	84740,68	52814,68	31926
6ο τρέξιμο	83534,68	51709,68	31825
7ο τρέξιμο	82732,01	50865,01	31867
8ο τρέξιμο	84690,79	52729,79	31961
9ο τρέξιμο	82866,19	51309,19	31557
10ο τρέξιμο	84575,43	52772,43	31803
<b>Μ.Ο.</b>	<b>83830,796</b>	<b>51961,196</b>	<b>31869,6</b>

Πίνακας αποτελεσμάτων για 90 πελάτες και μέγεθος 5

## 5.2.5 ΠΑΡΑΔΕΙΓΜΑ ΑΝΑΦΟΡΑΣ 110 ΠΕΛΑΤΩΝ

Αριθμός Πελατών 110	Διαθέσιμα Οχήματα 4	Χωρητικότητα 900	Χρονικός Ορίζοντας 7
------------------------	------------------------	---------------------	-------------------------

Πίνακας Δεδομένων Δεδομένα οχημάτων για 110 πελάτες

ΜΕΓΕΘΟΣ=n			
Τρεξίματα/Αποτελέσματα	Συνολικό Κόστος	Κόστος Δρομολόγησης	Κόστος Αποθεματοποίησης
1ο τρέξιμο	88876,48	49496,48	39380
2ο τρέξιμο	88085,1	50283,1	37802
3ο τρέξιμο	88352,16	50400,16	37952
4ο τρέξιμο	87543,52	51101,52	36442
5ο τρέξιμο	87651,43	50368,43	37283
6ο τρέξιμο	87550,11	50457,11	37093
7ο τρέξιμο	89479,09	50776,09	38703
8ο τρέξιμο	89124,14	50929,14	38195
<b>9ο τρέξιμο</b>	<b>87543,52</b>	51101,52	36442
10ο τρέξιμο	88134,71	50494,71	37640
<b>Μ.Ο.</b>	<b>88234,026</b>	<b>50540,826</b>	<b>37693,2</b>

Πίνακας αποτελεσμάτων για 110 πελάτες και μέγεθος n

ΜΕΓΕΘΟΣ=4			
Τρεξίματα/Αποτελέσματα	Συνολικό Κόστος	Κόστος Δρομολόγησης	Κόστος Αποθεματοποίησης
1ο τρέξιμο	90088,34	51528,34	38560
2ο τρέξιμο	86806,1	52607,1	34199
3ο τρέξιμο	88504,53	48845,53	39659
4ο τρέξιμο	90392,87	50677,87	39715
5ο τρέξιμο	90691,58	54231,58	36460
6ο τρέξιμο	<b>86170,51</b>	51389,51	34781
7ο τρέξιμο	87070,47	52891,47	34179
8ο τρέξιμο	89904,08	51233,08	38671
9ο τρέξιμο	89592,65	52807,65	36785
10ο τρέξιμο	88318,16	51438,16	36880
<b>Μ.Ο.</b>	<b>88753,929</b>	<b>51765,029</b>	<b>36988,9</b>

Πίνακας αποτελεσμάτων για 110 πελάτες και μέγεθος 4



ΜΕΓΕΘΟΣ=5			
Τρεξίματα/Αποτελέσματα	Συνολικό Κόστος	Κόστος Δρομολόγησης	Κόστος Αποθεματοποίησης
1ο τρέξιμο	89984,03	52030,03	37954
2ο τρέξιμο	86839,34	51421,34	35418
3ο τρέξιμο	86977,35	50571,35	36406
4ο τρέξιμο	89888,5	50934,5	38954
5ο τρέξιμο	86752,95	50865,95	35887
6ο τρέξιμο	<b>85564,29</b>	50633,29	34931
7ο τρέξιμο	89710,42	50509,42	39201
8ο τρέξιμο	87465,95	52776,95	34689
9ο τρέξιμο	89736,46	50134,46	39602
10ο τρέξιμο	87404,93	49711,93	37693
<b>Μ.Ο.</b>	<b>88032,422</b>	<b>50958,922</b>	<b>37073,5</b>

Πίνακας αποτελεσμάτων για 110 πελάτες και μέγεθος 5

### 5.3 ΣΥΓΚΡΙΣΕΙΣ

Στο συγκεκριμένο κεφάλαιο θα γίνει σύγκριση μεταξύ των αποτελεσμάτων της εργασίας Emmanouil E. Zachariadis, Christos D. Tarantilis, Chris T. Kiranoudis [3] και των αποτελεσμάτων που προέκυψαν τρέχοντας το πρόγραμμά για τα 3 μεγέθη που πάρθηκαν για τη Λίστα Περιορισμού των Υποψηφίων.

Στον παρακάτω πίνακα παρουσιάζονται τα καλύτερα αποτελέσματα - για Λίστα Περιορισμού Υποψηφίων ίση με τον αριθμό των πελατών - για τα αντίστοιχα παραδείγματα αναφοράς και γίνεται μια σύγκριση μεταξύ τους.

Παράδειγμα Αναφοράς	Συνολικό Κόστος		Κόστος Δρομολόγησης		Κόστος Αποθεματοποίησης		Διαφορά	Διαφορά Κοσ.Δρομ.
	Τρίμας	Εργασία	Τρίμας	Εργασία	Τρίμας	Εργασία	%	%
30 πελάτες	37387,93	33004,4	26143,43	20909,9	11244,5	12094,5	13,28	25,03
50 πελάτες	56492,07	52061,5	35734,07	30688,5	20758	21373	8,51	16,44
70 πελάτες	61065,83	53478,7	38376,83	28629,7	22689	24849	14,19	34,05
90 πελάτες	82133,98	71678,1	50686,98	38489,1	31447	33189	14,59	31,69
110 πελάτες	87543,52	73858,8	51101,52	40196,8	36442	33662	18,53	27,13

Πίνακας συγκρίσεων

Όπως παρατηρείται η χρήση του αλγορίθμου Grasp δεν έφερε καλύτερα αποτελέσματα από ότι η χρήση του αλγορίθμου Tabu. Η ποσοστιαία διαφορά μεταξύ των καλύτερων αποτελεσμάτων σύγκρισης φαίνεται να είναι σημαντική και το κυριότερο πρόβλημα παρατηρείται πως είναι το κόστος δρομολόγησης. Η ποσοστιαία διαφορά είναι αρκετά μεγάλη και η αύξηση στα περισσότερα παραδείγματα ξεπερνάει το 25%. Το κόστος αποθεματοποίησης

φαίνεται πως είναι αρκετά κοντά και για τα περισσότερα παραδείγματα αναφοράς τα αποτελέσματα είναι καλύτερα στο δικό μας πρόγραμμα. Με βάση αυτά τα αποτελέσματα για τους συγκεκριμένους αριθμούς πελατών είναι εύλογο να συμπεράνουμε πως για ακόμα μεγαλύτερο αριθμό πελατών η ποσοστιαία διαφορά μεταξύ των καλύτερων αποτελεσμάτων θα αυξηθεί και άλλο, ξεπερνώντας ακόμη και ενδεχομένως το 40%.

Παράδειγμα Αναφοράς	Συνολικό Κόστος			Κόστος Δρομολόγησης			Κόστος Αποθεματοποίησης		
	n	4	5	n	4	5	n	4	5
30 πελάτες	38385,172	38869,294	<b>38107,71</b>	26549,472	27537,294	26473,11	11835,7	11332	11634,6
50 πελάτες	58521,252	59631,104	<b>58469,725</b>	37098,652	38326,104	37310,325	21422,6	21305	21159,4
70 πελάτες	<b>61920,874</b>	63959,13	64547,284	39131,174	41641,53	42348,884	22789,7	22317,6	22198,4
90 πελάτες	<b>83709,078</b>	84244,678	83830,796	51835,478	52220,078	51961,196	31873,6	32024,6	31869,6
110 πελάτες	88234,026	88753,929	<b>88032,422</b>	50540,826	51765,029	50958,922	37693,2	36988,9	37073,5

Πίνακας Συγκρίσεων μέσων όρων με διαφορετικά μεγέθη Λίστας Περιορισμού Υποψηφίων

Στον παραπάνω πίνακα γίνεται σύγκριση των μέσων όρων των αποτελεσμάτων που προέκυψαν. Τα αποτελέσματα φαίνεται πως είναι κοντά μεταξύ τους, ωστόσο το μικρότερο μέγεθος Λίστας δε φαίνεται να υπερτερεί σε κανένα από τα άλλα μεγέθη σε οποιοδήποτε παράδειγμα αναφοράς. Ακόμα και όταν η Λίστα Περιορισμού των Υποψηφίων παίρνει μέγεθος ίση με τον αριθμό των πελατών, που σημαίνει ότι η επιλογή για τον επόμενο πελάτη γίνεται με τυχαίο τρόπο, τα αποτελέσματα είναι καλύτερα συγκρίνοντάς τα με αποτελέσματα για μέγεθος Λίστας ίση με 4.

## 5.4 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΑΡΑΤΗΡΗΣΕΙΣ

Εξετάζοντας τα παραπάνω αποτελέσματα και τον πίνακα συγκρίσεων παρατηρούνται ορισμένες σημαντικές διαφορές στα 2 προγράμματα.

Στα 2 παραδείγματα με το μικρότερο πλήθος πελατών έχουμε τη μικρότερη ποσοστιαία διαφορά. Αυτό συμβαίνει γιατί για τους 30 πελάτες είναι σχετικά μικρές οι διαδρομές και για τους 50 προστίθεται ένα επιπλέον όχημα σε σχέση με τους 30. Όμως στους 70 πελάτες χρησιμοποιούμε πάλι 3 οχήματα και συνδυαστικά με την αύξηση των πελατών και επακόλουθα των μεγαλύτερων διαδρομών που γίνονται, παρατηρείται μια μεγάλη αύξηση του ποσοστού του κόστους δρομολόγησης συγκρίνοντας τα παραδείγματα πελατών μεταξύ τους. Για τους 90 και 110 πελάτες καθώς χρησιμοποιούνται 4 οχήματα φαίνεται πως υπάρχει μια σταθεροποίηση στη ποσοστιαία διαφορά του κόστους δρομολόγησης. Για τους 110 πελάτες φαίνεται πως είναι το μοναδικό παράδειγμα που έχουμε αύξηση του συγκρίσιμου κόστους αποθεματοποίησης.

Τέλος, σημαντικό πρόβλημα αποτέλεσε το περιβάλλον προγραμματισμού. Ο κώδικας γράφτηκε σε Matlab R2016a. Οι χρόνοι που απαιτούνταν προκειμένου να τρέξει το πρόγραμμά μας αυτά τα 5 παραδείγματα ήταν τεράστιοι συγκριτικά με εκείνους της εργασίας που συγκρίναμε τα αποτελέσματά μας και όσο αυξάνονταν ο αριθμός πελατών και οχημάτων τόσο αυξάνονταν και οι απαιτούμενοι χρόνοι.

## 5.5 ΠΙΘΑΝΗ ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ

Όπως περιεγράφηκε στο 4.2.2.4 οι αλγόριθμοι που χρησιμοποιούνται προκειμένου να βελτιώσουμε τις διαδρομές είναι οι 1-1 exchange, 1-0 opt και 1-0 relocate. Αυτοί οι αλγόριθμοι με τον τρόπο που γράφτηκαν και χρησιμοποιήθηκαν, βελτίωσαν σημαντικά τα κόστη που παίρναμε από τις αρχικές λύσεις. Ωστόσο, δεν παίρναμε κάθε φορά την καλύτερη δυνατή αλλαγή που μπορούσε να γίνει. Ίσως με αυτόν τον τρόπο θα μπορούσαμε να πάρουμε καλύτερα αποτελέσματα. Μία άλλη πρόταση σχετική με τη χρήση του Grasp στο συγκεκριμένο πρόβλημα είναι ο πειραματισμός στη Λίστας Περιορισμού ως προς τα αποτελέσματα που θα έδινε στα κόστη δρομολόγησης. Επίσης θα μπορούσαμε να τρέξουμε επαναληπτικά τα κομμάτια κώδικα των τοπικών αλγορίθμων για έναν «χ» αριθμό επαναλήψεων και με τυχαία σειρά προκειμένου να προκύψουν καλύτερα αποτελέσματα δρομολόγησης, ακόμα και συνδυαστικά με τους αλγορίθμους 4.2.2.1 και 4.2.2.2. Επιπρόσθετα, θα μπορούσε να υπάρξει βελτίωση αν δημιουργούσαμε τις διαδρομές για το κάθε όχημα με σειρά προτεραιότητας και όχι ταυτόχρονα όπως κάναμε στην παρούσα εργασία. Με αυτόν τον τρόπο ενδεχομένως τα συνολικά κόστη δρομολόγησης να ήταν καλύτερα αν και με αυτόν τον τρόπο ενδεχομένως οι διαδρομές να μην ήταν ισορροπημένες για το κάθε όχημα ως προς το μέγεθός τους.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Ιωάννης Μαρινάκης | Μαγδαληνή Μαρινάκη | Αθανάσιος Μυγδαλάς (2019) *Προβλήματα Δρομολόγησης Οχημάτων στη Διαχείριση της Εφοδιαστικής Αλυσίδας*
- [2] Μαρινάκης, Ι., Μυγδαλάς, Α. (2008) *Σχεδιασμός και Βελτιστοποίηση της Εφοδιαστικής Αλυσίδας, Εκδόσεις “σοφία”*.
- [3] An integrated local search method for inventory and routing decisions Emmanouil E. Zachariadis, Christos D. Tarantilis, Chris T. Kiranoudis
- [4] Anily, S., & Federgruen, A. (1990). One warehouse multiple retailer systems with vehicle routing costs. *Management Science*, 36(1), 92–114.
- [5] Anily, S., & Federgruen, A. (1993). Two-echelon distribution systems with vehicle routing costs and central inventories. *Operations Research*, 41, 37–47.
- [6] Archetti, C., Bertazzi, L., Laporte, G., & Speranza, M. G. (2007). A Branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41, 382–391.
- [7] Bramel, J., & Simichi-Levi, D. (1995). A location based heuristic for general routing problems. *Operations Research*, 43, 649–660.
- [8] Cordeau, J. F., Gendreau, M., & Laporte, G. (1997). A Tabu Search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30, 105–119.

- [9] Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13, 533–549.
- [10] Herer, Y. T., & Roundy, R. (1997). Heuristics for a one-warehouse multiretailer distribution problem with performance bounds. *Operations Research*, 45, 102–115.
- [11] Jung, J., & Mathur, K. (2007). An efficient heuristic algorithm for a two-echelon joint inventory and routing problem. *Transportation Science*, 41, 55–73.
- [12] Tarantilis, C. D., Zachariadis, E. E., & Kiranoudis, C. T. (2008). A hybrid guided local search for the vehicle-routing problem with intermediate replenishment facilities. *INFORMS Journal of Computing*, 20, 154–168.
- [13] Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. Philadelphia: SIAM Monographs on Discrete Mathematics and Applications.
- [14] Viswanathan, S., & Mathur, K. (1997). Integrating routing and inventory decisions in one-warehouse multiretailer multiproduct distribution systems. *Management Science*, 43, 294–312.
- [15] Zhao, Q.-H., Wang, S., & Lai, K.-K. (2007). A partition approach to the inventory/routing problem. *European Journal of Operations Research*, 177, 786–802.
- [16] [https://repository.kallipos.gr/bitstream/11419/5392/2/02\\_chapter01.pdf](https://repository.kallipos.gr/bitstream/11419/5392/2/02_chapter01.pdf)
- [17] <https://el.wikipedia.org>

- [18] [https://www.researchgate.net/figure/An-overview-of-the-logistics-network\\_fig2\\_312562721](https://www.researchgate.net/figure/An-overview-of-the-logistics-network_fig2_312562721)
- [19] <https://www.tandfonline.com/doi/pdf/10.1080/16484142.2005.9638005>
- [20] <https://www.supplychain.gr>