

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ



<<Αλγοριθμική Επίλυση του Προβλήματος Δρομολόγησης Μη-Στελεχωμένων Εναέριων Οχημάτων με Παραλαβές, Παραδόσεις και Σταθμούς Αντικατάστασης Μπαταριών>>

<<Algorithm Solution of the Drone Routing Problem with Pickups, Deliveries and Battery Replacement Stations>>

ΓΙΑΝΝΟΥΛΑΚΗΣ ΕΜΜΑΝΟΥΗΛ

A.M: 2017010071

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Δρ. Μαρινάκης Ιωάννης

XANIA 2023

ΕΥΧΑΡΙΣΤΙΕΣ

Για την ολοκλήρωση αυτού του κύκλου με την παρούσα διπλωματική εργασία θα ήθελα να ευχαριστήσω την οικογένεια μου για όλη της την στήριξη όλο αυτό το διάστημα. Χρωστάω επίσης ένα πολύ μεγάλο ευχαριστώ στον κ. Ιωάννη Μαρινάκη, καθώς και στον Δρ. Κυριακάκη Νικόλαο για την απίστευτη προθυμία του κάθε φορά για να βοηθήσει.

Πίνακας Περιεχομένων

ΚΕΦΑΛΑΙΟ 1.....	7
ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΦΟΔΙΑΣΤΙΚΗ ΑΛΥΣΙΔΑ	7
1.1 Εισαγωγή	7
1.2 Προγραμματισμός Εφοδιαστικής Αλυσίδας.....	8
1.3 Διαχείριση Εφοδιαστικής Αλυσίδας.....	9
1.4 Κόστη	11
1.5 Μέτρα Απόδοσης	12
ΚΕΦΑΛΑΙΟ 2.....	13
ΠΡΟΒΛΗΜΑΤΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ	13
2.1 Εισαγωγή	13
2.2 Το πρόβλημα Δρομολόγησης Οχημάτων (VRP).....	13
2.3 Αλγόριθμοι Προβλημάτων Δρομολόγησης:	14
2.3.1 Το περιορισμένης χωρητικότητας πρόβλημα δρομολόγησης.....	14
2.3.2 Πρόβλημα δρομολόγησης οχημάτων με πολλαπλές αποθήκες	15
2.3.3 Πρόβλημα δρομολόγησης οχημάτων για την εξυπηρέτηση πελατών μέσα σε δεδομένα χρονικά παράθυρα.....	15
2.3.4 Πρόβλημα δρομολόγησης οχημάτων με ταυτόχρονη διανομή και παραλαβή προϊόντων κατά την διάρκεια της διανομής.....	15
2.3.5 Πρόβλημα δρομολόγησης οχημάτων με δύο είδη πελατών κατά τη διάρκεια της διαδρομής	15
2.4 Προβλήματα Δρομολόγησης Οχημάτων με Drones	16
2.4.1 Μια μεγάλη προσαρμοστική μεταβλητής γειτονιά αναζήτησης για το πρόβλημα δρομολόγησης οχημάτων με drones	16
2.4.2 Ένας δυναμικός αλγόριθμος δρομολόγησης οχημάτων που βασίζεται στη βελτιστοποίηση για την παράδοση γεύματος κατά παραγγελία με drones	16
2.4.3 Βελτιστοποίηση της παράδοσης δεμάτων με τη βοήθεια drone για το πρόβλημα του ταξιδιώτη πωλητή που πετάει.....	17
ΚΕΦΑΛΑΙΟ 3.....	18
ΑΛΓΟΡΙΘΜΟΙ	18
3.1 Εισαγωγή	18
3.2 Ευρετικοί Αλγόριθμοι.....	18
3.2.1 Αλγόριθμοι απληστίας.....	19
3.2.2 Προσεγγιστικοί αλγόριθμοι	20
3.2.3 Αλγόριθμοι τοπικής αναζήτησης.....	20

3.3 Μεθευρετικοί Αλγόριθμοι	21
3.3.1 Διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης (Greedy Randomized Adaptive Search Procedure (Grasp))	21
3.3.2 Αλγόριθμος μεταβλητής γειτονιάς αναζήτησης (Variable Neighborhood Search (VNS))	23
3.3.3 Προσομοιωμένη ανόπτηση (Simulated Annealing (SA))	24
3.3.4 Μέθοδος αποδοχής κατωφλιού (Threshold Accepted (TA))	24
3.3.5 Περιορισμένη αναζήτηση (Tabu Search (TS))	24
3.3.6 Αλγόριθμος επανασύνδεσης διαδρομών (Path relinking (PR))	24
3.3.7 Αλγόριθμος επαναληπτικής τοπικής αναζήτησης (Iterated Local Search (ILS)).....	25
3.3.8 Αλγόριθμος καθοδηγούμενης τοπικής αναζήτησης (Guided Local Search (GLS))	25
3.3.9 Αλγόριθμος προσαρμοστικής αναζήτησης (Adaptive Memory (AM))	25
3.4 Γενετικοί Αλγόριθμοι.....	25
3.4.1 Βελτιστοποίηση μεταναστευτικών πτηνών (MBO)	26
3.4.2 Βελτιστοποίηση σωματιδίων (PSO).....	26
3.4.3 Αλγόριθμος τεχνητής αποικίας μελισσών (ABC).....	26
3.4.4 Firefly algorithm	27
3.4.5 Αλγόριθμος αναζήτησης κούκου (CSA)	27
ΚΕΦΑΛΑΙΟ 4.....	28
ΜΑΘΗΜΑΤΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΟΥ EM-SDRPPD-BRS	28
4.1 Μοντελοποίηση	28
ΚΕΦΑΛΑΙΟ 5.....	33
ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΜΕ ΤΟΝ ΥΒΡΙΔΙΚΟ ΑΛΓΟΡΙΘΜΟ GRASP/VNS.....	33
5.1 Δεδομένα	33
5.2 Αλγόριθμοι.....	34
5.3 Ψευδοκώδικες	42
ΚΕΦΑΛΑΙΟ 6.....	51
ΑΠΟΤΕΛΕΣΜΑΤΑ	51
6.1 Αποτελέσματα του Εκάστοτε Σετ Δεδομένων	51
6.2 Σύγκριση Σετ Δεδομένων.....	59
ΚΕΦΑΛΑΙΟ 7.....	73
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	73
ΒΙΒΛΙΟΓΡΑΦΙΑ	74

ΠΕΡΙΛΗΨΗ

Η εφοδιαστική αλυσίδα είναι η διαδικασία μεταφοράς ενός προϊόντος ή μιας υπηρεσίας από το σημείο παραγωγής στο σημείο παράδοσης με κύριο μέλημα τη μείωση του λειτουργικού κόστους της μεταφοράς των προϊόντων και την αύξηση της ικανοποίησης του πελάτη. Η εφοδιαστική αλυσίδα με σύμμαχο την τεχνολογία εξελίσσεται διαρκώς, με την εύρεση προηγμένων τρόπων και εναλλακτικών επίτευξης του σκοπού της. Στην παρούσα διπλωματική εργασία μοντελοποιείται κι επιλύεται το πρόβλημα Δρομολόγησης Μη- Στελεχωμένου Εναέριου Οχημάτος με Παραλαβές, Παραδόσεις και Σταθμούς Αντικατάστασης Μπαταριών. Σκοπός του προβλήματος είναι η βελτιστοποίηση της διαδρομής, ελαχιστοποιώντας την ενέργεια του drone. Το drone πραγματοποιεί την παραλαβή πακέτων με προκαθορισμένο βάρος από ένα σύνολο πωλητών, καθώς και την παράδοση σε ένα σύνολο αγοραστών, λαμβάνοντας υπόψη τους περιορισμούς χωρητικότητας και ενέργειας. Σε περίπτωση αξιοποίησης της εκάστοτε μπαταρίας, το drone οδηγείται στον κοντινότερο σταθμό αντικατάστασης μπαταρίας, προϋποθέτοντας ότι υπάρχει διαθεσιμότητα. Για την επίλυση του προβλήματος υλοποιείται ένας υβριδικός αλγόριθμος, ο οποίος αποτελείται από τον αλγόριθμο Απληστης Τυχαιοποιημένης Αναζήτησης για την αρχικοποίηση της γειτονιάς, γνωρίζοντας την θέση του κάθε πελάτη και των σταθμών και από τον αλγόριθμο Μεταβλητής Γειτονιάς Αναζήτησης για την βελτιστοποίηση της λύσης εφαρμόζοντας διάφορες μεθόδους τοπικής αναζήτησης.

ABSTRACT

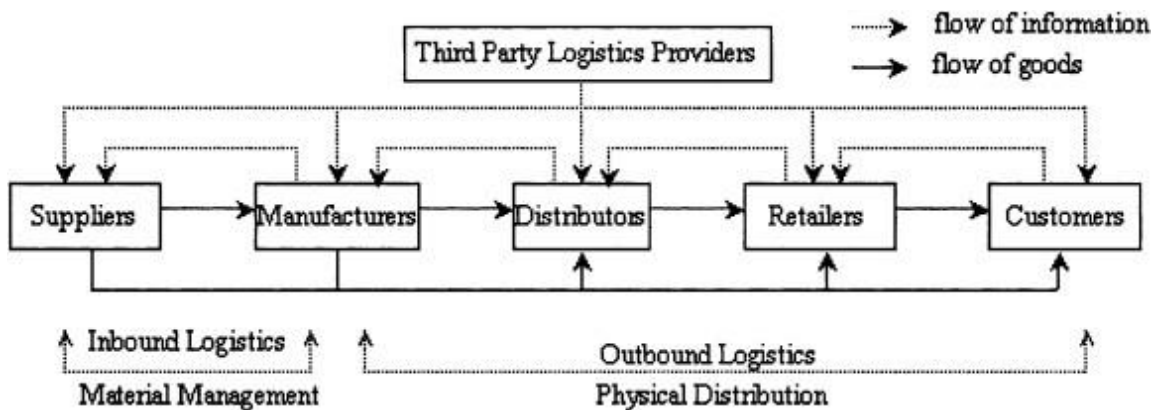
The supply chain is the process of transferring a product or one service from the point of production to the point of delivery, having as its main concern the reduction of the operational cost of transferring products and customer satisfaction. The supply chain allied with all the time evolving technology finds state-of-the-art ways and alternatives to achieve its purpose. In this senior thesis a drone routing problem is modeled and solved. The goal of this problem is the optimization of the path, minimizing the energy of one single drone. The drone carries out the pickup of the packages, which have predetermined weight, from a set of sellers and the delivery to a set of buyers, taking into consideration the capacity and energy constraints. In case of utilization of each battery, the drone is driven to the closest battery replacement station, if there is still availability. For the solution of the problem a hybrid algorithm is implemented, which consists of the algorithm of Greedy Randomized Adaptive Search Procedure for the initialization of the solution, knowing the location of each customer and each station and the Variable Neighborhood Search for the optimization of the solution applying a variety of local search methods.

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΦΟΔΙΑΣΤΙΚΗ ΑΛΥΣΙΔΑ

1.1 Εισαγωγή

Στην σημερινή παγκόσμια αγορά, οι εταιρίες δεν ανταγωνίζονται πλέον ως ανεξάρτητες οντότητες με ξεχωριστή επωνυμία, αλλά ως αναπόσπαστο μέρος των συνδέσμων της εφοδιαστικής αλυσίδας. Μια αλυσίδα εφοδιασμού αναφέρεται ως ένα ολοκληρωμένο σύστημα που συγχρονίζει μια σειρά αλληλένδετων επιχειρηματικών διαδικασιών για την απόκτηση πρώτων υλών και ανταλλακτικών, την μετατροπή αυτών σε τελικά προϊόντα, την προσθήκη τιμής σε αυτά, την διανομή και την προώθηση είτε σε λιανοπωλητές είτε σε πελάτες και την διευκόλυνση του σχηματισμού ανταλλαγών μεταξύ διάφορων επιχειρηματικών φορέων. Κύριος στόχος της είναι η ενίσχυση των λειτουργικών αποδοτικών κερδών.



Σχήμα 1.1. Εφοδιαστική αλυσίδα

Πιο συνοπτικά η διαχείριση της εφοδιαστικής αλυσίδας ορίζεται ως το κλειδί ολοκλήρωσης των επιχειρηματικών διαδικασιών από τους τελικούς χρήστες μέσω των αρχικών προμηθευτών που παρέχουν τα προϊόντα, τις υπηρεσίες, τις πληροφορίες και προσθέτουν αξία για τους πελάτες και τους άλλους ενδιαφερόμενους. Η εφοδιαστική χαρακτηρίζεται από μια προς τα εμπρός ροή αγαθών και μια προς τα πίσω ροή πληροφοριών όπως φαίνεται στην εικόνα 1. Τυπικά αποτελείται από δύο βασικές επιχειρηματικές διαδικασίες:

- Την διαχείριση των υλικών
- Φυσική κατανομή

Διαχείριση Υλικών

Η διαχείριση υλικών ασχολείται με την απόκτηση και αποθήκευση ανταλλακτικών και προμηθειών πρώτων υλών. Στηρίζει ολόκληρο τον κύκλο της ροής υλικών από την αγορά και τον εσωτερικό έλεγχο παραγωγής στον σχεδιασμό και τον έλεγχο της εργασιακής εξέλιξης προς τις αποθήκες, τα πλοία και την διανομή των τελικών προϊόντων.

Φυσική κατανομή

Από την άλλη πλευρά, η φυσική κατανομή περικλείει όλες τις εξερχόμενες δραστηριότητες που σχετίζονται με την παροχή υπηρεσιών στους πελάτες. Αυτές οι δραστηριότητες συμπεριλαμβάνουν παραλαβή και επεξεργασία παραγγελίας, ανάπτυξη καταστημάτων, αποθήκευση και χειρισμός, εξερχόμενες μεταφορές, ενοποίηση, τιμολόγηση, προωθητική υποστήριξη, διαχείριση προϊόντων επιστροφής και υποστήριξη τρόπου ζωής. Συνδυάζοντας αυτά τα δύο, η εφοδιαστική αλυσίδα δεν παρουσιάζει απλά μια γραμμική αλυσίδα επιχειρηματικών σχέσεων ένα προς ένα, αλλά ένα δίκτυο αυτών.

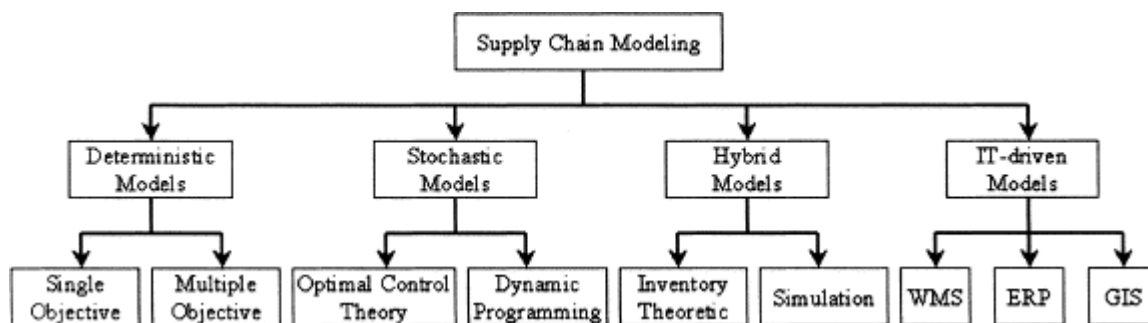
1.2 Προγραμματισμός Εφοδιαστικής Αλυσίδας

Κοιτάζοντας το ευρύ φάσμα της εφοδιαστικής αλυσίδας, κανένα μοντέλο δεν μπορεί να καλύψει όλες τις πλευρές των διαδικασιών. Σύμφωνα με τους [Chopra και Meindl \(2001\)](#) και [Stevens \(1989\)](#), [8], ο σκοπός ενός προβλήματος σε αυτόν τον τομέα βασίζεται σε τρία επίπεδα ιεραρχίας αποφάσεων, την ανταγωνιστική στρατηγική, τα τακτικά σχέδια και τις λειτουργικές ρουτίνες. Οι κατηγορίες των προβλημάτων που συναντώνται στην ανταγωνιστική στρατηγική περιλαμβάνουν αποφάσεις κατανομής τοποθεσίας, σχεδιασμού ζήτησης, σχεδιασμού διανομής, στρατηγικές συμμαχίες, ανάπτυξη νέων προϊόντων, εξωτερική ανάθεση, εκλογή προμηθευτών, τιμολόγηση και αναδιάρθρωση δικτύου. Υπάρχουν επίσης κάποια προβλήματα τακτικής. Αυτά συμπεριλαμβάνουν τις συντεταγμένες της παραγωγής και της διανομής, την διαχείριση υλικών, την επιλογή του εξοπλισμού και τα σχέδια. Τα προβλήματα που παρουσιάζονται κατά την αντιμετώπιση των λειτουργικών ρουτινών περιλαμβάνουν τα προγράμματα δρομολόγησης οχημάτων.

Γενικά μπορεί να υπάρχουν διάφορα σχήματα ταξινόμησης για την κατηγοριοποίηση των μοντέλων της εφοδιαστικής αλυσίδας. Για να ελαχιστοποιηθεί η σύγχυση αναπτύχθηκε αρχικά μια ταξινόμηση με κλασικές κατευθυντήριες γραμμές για τη διχοτόμηση των μαθηματικών μοντέλων: ντετερμινιστικό και στοχαστικό ([Bradley et al., 1977](#), [Mentzer and Schuster, 1983](#), [Budnick et al., 1988](#), [Beamon, Hillier, 199](#) [Lieberman, 2001](#)) ,[8]. Όπως σημειώνεται από τους [Silver \(1981\)](#), [Budnick et al., 1988](#), [Zipkin, 2000](#), [8], ορισμένα μοντέλα εφοδιαστικής αλυσίδας που βασίζονται στη θεωρία και την προσομοίωση αποθεμάτων περιέχουν τόσο ντετερμινιστικά όσο και στοχαστικά στοιχεία και κατά συνέπεια θα πρέπει να αντιμετωπίζονται ως υβριδικά. Μια άλλη κατηγορία που ονομάζεται μοντέλα που βασίζονται σε πληροφοριακά συστήματα (IT) προστέθηκε στην ταξινόμηση για να αντικατοπτρίζει τις τρέχουσες εξελίξεις στον τομέα της πληροφορικής για τη βελτίωση της αποτελεσματικότητας της εφοδιαστικής αλυσίδας.

Τα ντετερμινιστικά μοντέλα υποθέτουν ότι όλες οι παράμετροι του μοντέλου είναι γνωστές και σταθερές με βεβαιότητα, ενώ τα στοχαστικά μοντέλα λαμβάνουν υπόψη τις αβέβαιες και τυχαίες παραμέτρους. Τα ντετερμινιστικά μοντέλα διακρίνονται σε μονοσκοπικά και πολλαπλά αντικειμενικά μοντέλα. Αυτή η κατηγορία αναπτύχθηκε για να αντικατοπτρίζει την αυξανόμενη ανάγκη για εναρμόνιση αντικρουόμενων στόχων διαφορετικών εταίρων της εφοδιαστικής αλυσίδας. Τα στοχαστικά μοντέλα υποταξινομούνται σε μοντέλα βέλτιστου ελέγχου θεωρητικού και δυναμικού προγραμματισμού. Τα υβριδικά μοντέλα έχουν τόσο στοιχεία ντετερμινιστικών όσο και στοχαστικών μοντέλων. Αυτά τα μοντέλα περιλαμβάνουν μοντέλα απογραφής-θεωρίας και προσομοίωσης που είναι ικανά να αντιμετωπίσουν τόσο τη βεβαιότητα όσο και την αβεβαιότητα που περιλαμβάνει παραμέτρους του μοντέλου.

Ο [Shapiro \(2001\)](#), [8], παρατήρησε πρόσφατα ότι η ανάπτυξη της πληροφορικής ήταν η κινητήρια δύναμη για τις καινοτομίες της εφοδιαστικής αλυσίδας και τον επακόλουθο ανασχεδιασμό της επιχειρηματικής διαδικασίας. Τα μοντέλα IT στοχεύουν στην ενοποίηση και το συντονισμό διάφορων φάσεων του σχεδιασμού της εφοδιαστικής σε πραγματικό χρόνο χρησιμοποιώντας λογισμικό εφαρμογών, έτσι ώστε να μπορούν να ενισχύσουν την ορατότητα σε όλη την αλυσίδα εφοδιασμού. Αυτά τα μοντέλα περιλαμβάνουν WMS, συστήματα διαχείρισης μεταφορών (TMS), ολοκληρωμένη παρακολούθηση μεταφοράς, συνεργατικό σχεδιασμό και αναπλήρωση προβλέψεων (CPFR), σχεδιασμό απαιτήσεων υλικού (MRP), σχεδιασμό πόρων διανομής (DRP), ERP και συστήματα γεωγραφικών πληροφοριών (GIS).



Σχήμα 1.2. Ταξινόμηση των διάφορων τρόπων μοντελοποίησης της εφοδιαστικής αλυσίδας

1.3 Διαχείριση Εφοδιαστικής Αλυσίδας

Σε αυτό το αναδυόμενο ανταγωνιστικό περιβάλλον, η απόλυτη επιτυχία της μεμονωμένης επιχείρησης θα εξαρτάται από την ικανότητα της διοίκησης να ενσωματώνει το περίπλοκο δίκτυο επιχειρηματικών σχέσεων της εταιρείας. Η εφοδιαστική αλυσίδα ασχολείται με την συνολική αριστεία επιχειρηματικής διαδικασίας κι αντιπροσωπεύει έναν νέο τρόπο διαχείρισης της επιχείρησης και τις σχέσεις με άλλα μέλη της εφοδιαστικής αλυσίδας. Σε πολλές εταιρίες, η διαχείριση έχει καταλήξει στο συμπέρασμα ότι η βελτιστοποίηση του προϊόντος δεν μπορεί να πραγματοποιηθεί χωρίς την εφαρμογή μιας διαδικασίας προσέγγισης στην επιχείρηση. Οι διαδικασίες που προσδιορίζονται παρουσιάζονται αναλυτικά παρακάτω.

1. Διαδικασία διαχείρισης σχέσεων πελατών

Το πρώτο βήμα ως προς μια ολοκληρωμένη εφοδιαστική αλυσίδα είναι να στοχεύσει στους σωστούς πελάτες. Οι συμφωνίες των προϊόντων και των υπηρεσιών καθορίζουν τα επίπεδα αποδοτικότητας. Οι ομάδες εξυπηρέτησης δουλεύουν με τους πελάτες για να εξαλειφθεί η μεταβλητότητα της ζήτησης. Οι αξιολογήσεις απόδοσης πραγματοποιούνται για να παρέχουν πληροφορίες σχετικά με τις υπηρεσίες που παρέχονται στους πελάτες αλλά και του κέρδους που αναμένεται να έρθει.

2. Διαδικασία διαχείρισης εξυπηρέτησης πελατών

Η παροχή υπηρεσιών παρέχει στους πελάτες σε πραγματικό χρόνο την πληροφορία για την ημερομηνία αποστολής και την διαθεσιμότητα του προϊόντος. Επίσης βοηθάει τους πελάτες με τις εφαρμογές των προϊόντων.

3. Διαδικασία διαχείρισης της ζήτησης

Η ζήτηση των πελατών είναι μακράν η μεγαλύτερη πηγή μεταβλητότητας και πηγάζει από ακανόνιστα μοτίβα. Η διαδικασία διαχείρισης της ζήτησης πρέπει να εξισορροπεί τις απαιτήσεις του πελάτη με τις δυνατότητες της προμήθειας της επιχείρησης. Ένα καλό σύστημα χρησιμοποιεί σημεία πώλησης και βασικά δεδομένα πελατών για την μείωση της αβεβαιότητας και την παροχή αποτελεσματικής ροής σε όλη την αλυσίδα ανεφοδιασμού.

4. Διαδικασία εκπλήρωσης παραγγελίας πελάτη

Η αποτελεσματικότητα σε αυτό το κομμάτι έρχεται με την δυνατότητα της παροχής των υπηρεσιών προς τους πελάτες, τις ημερομηνίες που έχουν ανάγκη. Η επιτυχής εκπλήρωση της παραγγελίας απαιτεί την ενσωμάτωση των σχεδίων κατασκευής, διανομής και μεταφοράς της εταιρίας. Θα πρέπει να αναπτυχθεί ο τομέας της μεταφοράς των προϊόντων έτσι ώστε να ικανοποιούνται οι απαιτήσεις των πελατών και να μειώνεται το κόστος παράδοσης.

5. Διαδικασία διαχείρισης ροής κατασκευής

Στην παραγωγική διαδικασία γίνονται συχνά λάθη στον συνδυασμό των προϊόντων με αποτέλεσμα να υπάρχουν αχρειαστα αποθέματα, υπερβολικό κόστος μεταφοράς αποθεμάτων, μείωση σήμανσης και μεταφορτώσεις των προϊόντων. Οι διαδικασίες παραγωγής πρέπει να είναι ευέλικτες ώστε να ανταποκρίνονται στις αλλαγές της αγοράς. Οι παραγγελίες διεκπεραιώνονται σε τακτική βάση με ελάχιστα μεγέθη παρτιδών.

6. Διαδικασία σύναψης δημόσιων συμβάσεων

Αναπτύσσονται στρατηγικά σχέδια με προμηθευτές για να υποστηριχθεί η διαδικασία διαχείρισης και ανάπτυξης νέων προϊόντων. Οι προμηθευτές κατηγοριοποιούνται βάση διάφορων κριτηρίων, όπως η συνεισφορά τους και η κρισιμότητα τους στον οργανισμό. Η ύπαρξη έγκαιρου προμηθευτή μειώνει τον χρόνο που χρειάζεται ο συντονισμός μεταξύ της μηχανικής και της αγοράς πριν από την ολοκλήρωση του σχεδιασμού.

7. Ανάπτυξη προϊόντων και εμποριοποίηση

Η ψυχή της εταιρίας είναι τα νέα προϊόντα. Οι πελάτες και οι προμηθευτές πρέπει να είναι ενσωματωμένοι στην διαδικασία ανάπτυξης των προϊόντων, προκειμένου να μειώνεται ο χρόνος στην αγορά. Όσο τελειώνει ο κύκλος ζωής κάποιων προϊόντων, τότε πρέπει να παραχθούν γρήγορα και επιτυχώς τα σωστά προϊόντα σε ακόμα μικρότερη περίοδο, έτσι ώστε να είναι ανταγωνιστικά. Οι υπεύθυνοι της διαδικασίας πρέπει να συμβαδίζουν με τους πελάτες και να ξέρουν τις ανάγκες τους. Πρέπει να επιλέξουν υλικά και προμηθευτές σε συνδυασμό με την προμήθεια και πρέπει να αναπτύξουν την τεχνολογία παραγωγής και να ενσωματωθούν στην καλύτερη ροή εφοδιαστικής αλυσίδας για το προϊόν και την αγορά.

8. Διαδικασία επιστροφών

Η διαχείριση των επιστροφών ως διαδικασία σε μια εταιρία προσφέρει μια πολύ καλή ευκαιρία για την επίτευξη βιώσιμου ανταγωνισμού. Η αποτελεσματικότητα σε αυτόν τον τομέα επιτρέπει τον εντοπισμό ευκαιριών βελτίωσης της παραγωγικότητας κ πρωτοποριακών έργων. Οι επιστροφές διαχωρίζονται σε τέσσερις κατηγορίες: τον εξοπλισμό, τα ανταλλακτικά, τις προμήθειες και τις ανταγωνιστικές συναλλαγές.

1.4 Κόστη

Η εφοδιαστική αλυσίδα έχει πολλές δραστηριότητες, οι οποίες απαιτούν διάφορες δαπάνες. Κάποιες από αυτές είναι η εξυπηρέτηση του πελάτη, οι μεταφορές και οι διανομές, η εκπαίδευση του προσωπικού, η διαχείριση των εγκαταστάσεων, η επεξεργασία των παραγγελιών και των πληροφοριών, οι επικοινωνίες και η διαχείριση των αποθεμάτων. Η μείωση των εξόδων για κάθε μια από τις παραπάνω δαπάνες έχει ως αποτέλεσμα την συνολική ελαχιστοποίηση του κόστους και συνεπώς σε μια αποδοτικότερη εφοδιαστική αλυσίδα. Πιο αναλυτικά τα κόστη:

1. Κόστος εξυπηρέτησης πελατών:

Το κόστος αυτό περιλαμβάνει τα έξοδα για την ολοκλήρωση των παραγγελιών, για ανταλλακτικά και για την διαχείριση των επιστρεφόμενων προϊόντων.

2. Κόστος μεταφορών και διανομών:

Το κόστος αυτό αναφέρεται στα έξοδα παράδοσης των προϊόντων είτε στον πελάτη είτε στην εκάστοτε αποθήκη κι επηρεάζεται και μεταβάλλεται από διάφορα κριτήρια. Αυτά τα κριτήρια είναι η απόσταση του σημείου παραγωγής σε σχέση με τον πελάτη ή την αποθήκη, το μέσο μεταφοράς, ο όγκος αυτών, καθώς και το βάρος.

3. Κόστος διαχείρισης παραγγελιών:

Τα έξοδα αυτά έχουν να κάνουν με την πρόβλεψη των παραγγελιών και των τεχνολογικών συστημάτων με τα οποία αυτή πραγματοποιείται.

4. Κόστος αποθεμάτων:

Αυτά τα έξοδα είναι συγκεκριμένα. Το κόστος ευκαιρίας που αντιστοιχεί στην απόδοση του κεφαλαίου, το κόστος υπηρεσιών αποθέματος που αποτελείται από τα έξοδα της ασφάλισης και φορολόγησης, το κόστος αποθηκευτικού χώρου που περιλαμβάνει τα έξοδα των αποθηκών και τέλος το κόστος αποθεματικού κινδύνου που αυξάνεται με την παλαίωση, την ζημιά ή και την κλοπή των αποθεμάτων.

5. Κόστος παρτιδών:

Αυτά τα έξοδα αναφέρονται στην παραγωγή και την αγορά των πρώτων υλών και μπορούν να επηρεαστούν από την συνεχή αλλαγή των τιμών των πρώτων υλών αλλά και των τελικών προϊόντων στην αγορά.

1.5 Μέτρα Απόδοσης

Υπάρχουν διάφορα μέτρα απόδοσης τα οποία αν και ποιοτικά μπορούν εν μέρει να ποσοτικοποιηθούν. Μείζων παράγοντα παίζει η ικανοποίηση των πελατών με το προϊόν. Η ικανοποίηση των πελατών προέρχεται από την ικανοποίηση πριν από την συναλλαγή, η οποία σχετίζεται με τα στοιχεία της υπηρεσίας, την ικανοποίηση της συναλλαγής, η οποία σχετίζεται με φυσική διανομή των προϊόντων και την ικανοποίηση μετά την συναλλαγή, η οποία σχετίζεται με την υποστήριξη που παρέχεται για τα προϊόντα κατά τη χρήση. Άλλα μέτρα που επιτυγχάνουν την ικανοποίηση του πελάτη είναι η ευελιξία, που σχετίζεται στον βαθμό με τον οποίο η εφοδιαστική αλυσίδα μπορεί να ανταποκριθεί σε τυχαίες διακυμάνσεις στο πρότυπο ζήτησης. Η ενοποίηση ροής πληροφοριών, δηλαδή η ανταλλαγή πληροφοριών και η μεταφορά υλικών, η αποτελεσματική διαχείριση του κινδύνου και η απόδοση του προμηθευτή είναι παράγοντες που επηρεάζουν την κατάσταση που θα φτάσουν οι πρώτες ύλες στην παραγωγή.

Όσον αφορά τα μέτρα που βασίζονται στο κόστος, άκρως σημαντική είναι η ελαχιστοποίηση του κόστους και στην συνέχεια ακολουθούν η μεγιστοποίηση των πωλήσεων, η μεγιστοποίηση του κέρδους, η ελαχιστοποίηση των χρημάτων που απαιτούνται για τα αποθέματα και η μεγιστοποίηση της απόδοσης της επένδυσης, το οποίο συνεπάγεται αύξηση του καθαρού κέρδους και ελαχιστοποίηση του κόστους παραγωγής.

ΚΕΦΑΛΑΙΟ 2

ΠΡΟΒΛΗΜΑΤΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο αναλύονται κάποια από τα προβλήματα δρομολόγησης οχημάτων και στην συνέχεια αναφέρονται μερικές περιπτώσεις από τα προβλήματα δρομολόγησης οχημάτων με UAV. Τα UAV ή αλλιώς drones μπορεί να είναι είτε αυτόνομα είτε να είναι σε ζεύγη με φορτηγά.

2.2 Το πρόβλημα Δρομολόγησης Οχημάτων (VRP)

Με την συνεχώς αυξανόμενη τεχνολογία οι απαιτήσεις διαρκώς αυξάνονται με ρυθμούς που αναγκάζουν την εφοδιαστική αλυσίδα, τον πυρήνα των επιχειρήσεων, να εξελίσσεται. Η κάλυψη των απαιτήσεων των πελατών και η εξασφάλιση της αύξησης των μεριδίων της αγοράς και των καθαρών κερδών, σηματοδοτεί μια αναπτυγμένη εφοδιαστική αλυσίδα.

Το πρόβλημα της Δρομολόγησης Οχημάτων (VRP – VehicleRoutingProblem) είναι ένα πρόβλημα συνδυαστικής βελτιστοποίησης και προγραμματισμού που έχει ως σκοπό την εύρεση του βέλτιστου συνόλου διαδρομών που πρέπει να διανύσει ένας στόλος συγκεκριμένων οχημάτων έτσι ώστε να καλυφθεί ολόκληρη η ζήτηση των πελατών με το ελάχιστο δυνατό κόστος. Γενικεύει το πρόβλημα του πλανώδιου πωλητή που παρουσιάστηκε για πρώτη φορά από τους [Dantzing&Ramser to 1959, \[13\]](#), ενώ σήμερα αποτελεί ένα από τα σημαντικά και εφαρμόσιμα προβλήματα διανομής της εφοδιαστικής αλυσίδας. Οι Dantzing&Ramser δημιούργησαν την πρώτη αλγοριθμική προσέγγιση επίλυσης τέτοιων προβλημάτων και εφαρμόστηκε για την μεταφορά και παράδοση βενζίνης. Στην συνέχεια το έτος [1964, οι Clarke&Wright, \[13\]](#), βελτίωσαν την προσέγγιση των Dantzing&Ramser χρησιμοποιώντας έναν αποτελεσματικό άπληστο αλγόριθμο που ονομάζεται αλγόριθμος αποταμίευσης. Πλέον έχουν αναπτυχθεί πάρα πολλοί αλγόριθμοι για τέτοιου είδους προβλήματα, καθένας από τους οποίους βάση των δεδομένων που του δίνονται και των περιορισμών που θέτονται, επιδιώκει την βελτιστοποίηση του συνολικού κόστους.

Στα προβλήματα δρομολόγησης τόσο τα οχήματα όσο και οι πελάτες έχουν κάποια χαρακτηριστικά. Ξεκινώντας λοιπόν από τους πελάτες έχουμε ως δεδομένα τις συντεταγμένες τους στον χώρο, την ποσότητα και το βάρος των προϊόντων που πρέπει να διανεμηθούν ή να παραληφθούν από τον πελάτη, τα χρονικά διαστήματα μέσα στα οποία ο πελάτης μπορεί να εξυπηρετηθεί και ο χρόνος που απαιτείται για την εξυπηρέτηση. Από την άλλη πλευρά για τα οχήματα δίνονται οι αποθήκες ή η αποθήκη από την οποία ξεκινάνε και που ολοκληρώνουν την

διαδρομή, η χωρητικότητα τους και το συνολικό βάρος που μπορούν να διανέμουν, την διαθεσιμότητα μηχανήματος για την φόρτωση και εκφόρτωση των οχημάτων και το είδος του οχήματος, το οποίο συνεπάγεται στα λειτουργικά κόστη και σε ποιους δρόμους μπορεί να κατευθυνθεί.

Τα παραπάνω προβλήματα όπως προαναφέρθηκε αποτελούνται από περιορισμούς, το μέγεθος των οποίων καθορίζουν σε μεγάλο βαθμό την ποιότητα της λύσης. Τα είδη των περιορισμών για κάθε είδους πρόβλημα μεταβάλλεται. Οι πελάτες υπάρχει πιθανότητα να διαχωρίζονται σε σύνολα. Ένα τέτοιο σύνολο μπορεί να αποτελείται είτε από πελάτες που παραδίδουν ένα προϊόν, είτε από πελάτες που παραλαμβάνουν ένα προϊόν, είτε από πελάτες που έχουν και τις δύο παραπάνω ιδιότητες. Οι πελάτες μπορεί να ζητούν την εξυπηρέτησή τους σε συγκεκριμένο χρονικό διάστημα. Άλλοι περιορισμοί είναι ότι οι οδηγοί έχουν συγκεκριμένο ωράριο εργασίας και ότι το όχημα δεν έχει τη δυνατότητα να μεταφέρει φορτίο μεγαλύτερο της χωρητικότητας του. Ανάλογα το πρόβλημα μπορούν να υπάρξουν κι άλλοι περιορισμοί σχετικά με την σειρά εξυπηρέτησης πελατών.

Ως στόχο τα συγκεκριμένα προβλήματα έχουν να ελαχιστοποιηθεί το συνολικό κόστος, το οποίο επηρεάζεται και μεταβάλλεται ανάλογα της απόστασης που διανύει το όχημα ή τα οχήματα. Άλλος παράγοντας που επηρεάζει αυτό το κόστος είναι ο αριθμός των οχημάτων και των οδηγών, καθώς και ο χρόνος που απαιτείται για την εξυπηρέτηση όλων των πελατών.

2.3 Αλγόριθμοι Προβλημάτων Δρομολόγησης:

Σε αυτήν την ενότητα θα αναλυθούν μερικά από τα προβλήματα δρομολόγησης οχημάτων για την παράδοση ή και την παραλαβή προϊόντων. Τα προβλήματα αυτά εντάσσονται στα πλαίσια της ιεραρχικής διαχείρισης της εφοδιαστικής αλυσίδας.

2.3.1 Το περιορισμένης χωρητικότητας πρόβλημα δρομολόγησης

Το πρόβλημα δρομολόγησης οχημάτων περιορισμένης χωρητικότητας χρησιμοποιείται για τις περιπτώσεις που ένα όχημα λόγω περιορισμού του όγκου που περιέχει και μπορεί να μεταφέρει, δεν έχει την δυνατότητα να εξυπηρετήσει μόνο του όλους τους πελάτες. Όλοι οι πελάτες θα εξυπηρετηθούν μια φορά και οι διαδρομές που θα πραγματοποιηθούν από τα οχήματα θα έχουν αρχή και τέλος.

Το όχημα μπορεί να είναι αδύνατο να φέρει εις πέρας την ικανοποίηση όλων των πελατών μόνο του λόγω του ότι το όχημα δεν μπορεί να μεταφέρει όλη την ποσότητα του φορτίου, είτε λόγω όγκου είτε λόγω βάρους. Άλλος ένας λόγος θα μπορούσε να ήταν ότι υπάρχει περιορισμός χρόνου μέσα στον οποίο πρέπει να ικανοποιηθεί η ζήτηση.

2.3.2 Πρόβλημα δρομολόγησης οχημάτων με πολλαπλές αποθήκες

Το συγκεκριμένο πρόβλημα έχει παραπάνω από μια αποθήκη και έχει δύο τρόπους επίλυσης. Στην πρώτη εκδοχή, υπάρχει ένα όχημα που κάνει όλες τις επιμέρους διαδρομές. Ξεκινάει από μια αποθήκη και καταλήγει είτε στην ίδια είτε σε διαφορετική. Εκεί μπορεί να φορτώσει ξανά και να ξεκινήσει μια καινούρια διαδρομή. Στην δεύτερη εκδοχή κάθε αποθήκη έχει το δικό της αριθμό οχημάτων και κάθε ένα από αυτά εξυπηρετεί τους πελάτες που αντιστοιχούν στην κάθε αποθήκη.

2.3.3 Πρόβλημα δρομολόγησης οχημάτων για την εξυπηρέτηση πελατών μέσα σε δεδομένα χρονικά παράθυρα.

Σε αυτό το πρόβλημα πρέπει να εξυπηρετηθούν όλοι οι πελάτες μια φορά από ένα πλήθος οχημάτων, τα οποία έχουν ως σημείο εκκίνησης μια αποθήκη. Κάθε πελάτης περιμένει την παραλαβή ενός φορτίου μέσα σε ένα συγκεκριμένο χρονικό πλαίσιο. Τα οχήματα έχουν επίσης περιορισμό στην χωρητικότητα τους. Το όχημα για να μην παραβιάσει κάποιον περιορισμό πρέπει να έχει προλάβει να εξυπηρετήσει τον κάθε πελάτη πρωτού περάσει το χρονικό παράθυρο. Μέσα σε αυτό το χρονικό διάστημα εννοείται ότι το όχημα πρέπει να έχει φτάσει τον πελάτη και να προλάβει και να τον εξυπηρετήσει.

2.3.4 Πρόβλημα δρομολόγησης οχημάτων με ταυτόχρονη διανομή και παραλαβή προϊόντων κατά την διάρκεια της διανομής

Το πρόβλημα περιλαμβάνει διανομές και παραλαβές από τους πελάτες κατά τη διάρκεια μια διαδρομής. Κάθε πελάτης μπορεί να έχει δύο ιδιότητες. Η πρώτη είναι το να θέλει να διανέμει προϊόντα και η δεύτερη να θέλει να παραλάβει προϊόντα. Συνήθως κάθε πελάτης έχει μια από τις δύο απαιτήσεις. Επιπρόσθετα είναι εφικτό να υπάρχουν δύο κόμβοι από τους οποίους να ξεκινάνε τα προϊόντα χωρίς ωστόσο να αποκλείεται αυτοί οι δύο κόμβοι να αναφέρονται στον ίδιο κόμβο.

2.3.5 Πρόβλημα δρομολόγησης οχημάτων με δύο είδη πελατών κατά τη διάρκεια της διαδρομής

Οι πελάτες χωρίζονται σε δύο σύνολα, εκ των οποίων το πρώτο περιλαμβάνει τους πελάτες, οι οποίοι δίνουν κάποιο προϊόν στο όχημα και το δεύτερο σύνολο, το οποίο παίρνει κάποια συγκεκριμένη παραγγελία ενός πελάτη του πρώτου συνόλου. Σε αυτό το πρόβλημα υπάρχει ένας πάρα πολύ βασικός περιορισμός. Δεν μπορεί να εξυπηρετηθεί κάποιος πελάτης του δεύτερου κύκλου πρωτού εξυπηρετηθεί ο αντίστοιχος πελάτης του πρώτου κύκλου. Επίσης:

- κάθε όχημα ακολουθεί μόνο μια διαδρομή που έχει ως αρχικό και τελικό κόμβο την αποθήκη.
- Κάθε πελάτης επισκέπτεται μόνο μια φορά.

- Τα οχήματα δεν επιτρέπεται να υπερβούν την χωρητικότητα τους.
- Το κάθε όχημα δεν μπορεί να ξεπεράσει την απόσταση που του έχει τεθεί.
- Διαδρομές που περιλαμβάνουν πελάτες μόνο του δεύτερου συνόλου δεν γίνονται αποδεκτές.
- Σκοπός είναι η ελαχιστοποίηση της απόστασης που θα διανύσουν τα οχήματα.

2.4 Προβλήματα Δρομολόγησης Οχημάτων με Drones

Το πρόβλημα δρομολόγησης οχημάτων με drones είναι μια επέκταση του κλασικού προβλήματος δρομολόγησης με περιορισμό χωρητικότητας, όπου όχι μόνο φορτηγά αλλά και drones χρησιμοποιούνται για την παράδοση δεμάτων στους πελάτες.

2.4.1 Μια μεγάλη προσαρμοστική μεταβλητής γειτονιά αναζήτησης για το πρόβλημα δρομολόγησης οχημάτων με drones

Για το πρόβλημα των [\[10\] Sacramento, D., Pisinger, D., & Ropke, S. \(2019\)](#) δεδομένου ενός στόλου ομοιογενών φορτηγών, καθένα από τα οποία είναι εξοπλισμένο με ένα μόνο UAV ή drone, η αποστολή είναι να παραδωθούν τα πακέτα σε ένα σύνολο πελατών, καθέ ένας από τους οποίους πρέπει να εξυπηρετηθεί ακριβώς μια φορά είτε από τον οδηγό είτε από το UAV που λειτουργεί σε συντονισμό με το φορτηγό. Κάθε φορτηγό με το αντίστοιχο UAV πρέπει να έχει ως σημείο εκίνησης και επιστροφής μια αποθήκη. Όταν το drone δεν είναι σε λειτουργία, θα μεταφερθεί με το φορτηγό, εξοικονομώντας ενέργεια από την μπαταρία. Τα drones μπορούν να αποσταλούν από το φορτηγό και να παραληφθούν ξανά από το ίδιο φορτηγό από διαφορετική τοποθεσία πολλές φορές κατά την διάρκεια της διαδρομής του φορτηγού. Ωστόσο το drone μπορεί να επισκέπτεται έναν πελάτη την φορά λόγω της περιορισμένης χωρητικότητας. Υπάρχει ένας χρόνος που σχετίζεται με την εκίνηση και την ανάκτηση του drone και ένας άλλος χρόνος που σχετίζεται με τον χρόνο εξυπηρέτησης. Τα φορτηγά έχουν περιορισμένη χωρητικότητα και η διαδρομή των φορτηγών δεν πρέπει να υπερβαίνει ένα συγκεκριμένο χρονικό όριο κατά την διάρκεια της ημέρας. Σκοπός του προβλήματος είναι η ικανοποίηση όλων των πελατών και η ελαχιστοποίηση του συνολικού κόστους λειτουργίας, χωρίς την παραβίαση των περιορισμών.

2.4.2 Ένας δυναμικός αλγόριθμος δρομολόγησης οχημάτων που βασίζεται στη βελτιστοποίηση για την παράδοση γεύματος κατά παραγγελία με drones

Στον αλγόριθμο του [Liu \(2019\) \[6\]](#) κάθε drone φέρει ένα μόνο μονωμένο κουτί τροφίμων που μπορεί να χωρέσει πολλές παραγγελίες κανονικού μεγέθους. Δεν επιτρέπεται η μεταφορά διαφορετικών ειδών τροφίμων (συγκεκριμένα ζεστά γεύματα και κρύα ποτά) στο ίδιο κουτί. Η μονάδα χρόνου είναι το λεπτό. Η φόρτωση παραγγελιών σε drone και παραγγελιών εκφόρτωσης από ένα drone διαρκεί ένα ή περισσότερα διαδοχικά ολόκληρα λεπτά, μια παράμετρος ανεξάρτητη

από τον τύπο του drone, τον τύπο παραγγελίας και το μέγεθος παραγγελίας. Ο ρυθμός κατανάλωσης μπαταρίας εξαρτάται από το μέγεθος του ωφέλιμου φορτίου και το σωματικό βάρος του drone. Οι μπαταρίες μπορούν να αντικατασταθούν σε αποθήκες φόρτισης και οι μπαταρίες που ανταλλάσσονται είναι πλήρως φορτισμένες. Η αλλαγή μπαταρίας διαρκεί ένα ή πολλά ολόκληρα λεπτά για οποιοδήποτε drone. Οι αποθήκες φόρτισης δεν έχουν χωρητικότητα - κάθε αποθήκη έχει άφθονα σημεία προσγείωσης και ένα drone μπορεί να εξυπηρετηθεί αμέσως μετά την άφιξη σε μια αποθήκη. Υποθέτουμε ότι τα drones έχουν ικανότητες πλοήγησης και είναι ικανά να ακολουθούν τις εντολές αποστολής και να αποφεύγουν αυτόνομα συγκρούσεις στην πορεία πτήσης. Όλες οι παραγγελίες γευμάτων από προεπιλογή έχουν την ίδια προτεραιότητα και σε περίπτωση που απαιτείται ρητή ιεράρχηση προτεραιοτήτων για την επίλυση αδιέξοδων καταστάσεων, η παραγγελία που έχει ξεκινήσει νωρίτερα έχει υψηλότερη προτεραιότητα. Για να διατηρηθεί η επιχειρησιακή πραγματικότητα, το εύρος κινητικότητας του drone στον δυσδιάστατο χώρο αντιπροσωπεύεται από ένα πολύγωνο n όψεων όπου το n είναι μια επιλογή μοντελοποίησης. Ωστόσο, ο κανόνας L1 (η απόσταση του Μανχάταν) υιοθετείται για τη μέτρηση της εγγύτητας, δηλαδή για τον προσδιορισμό του εάν ένα drone έχει φτάσει σε μια θέση παραλαβής/απόθεσης/φόρτισης και για την τιμωρία της περιττής περιπλάνησης των drones.

2.4.3 Το πρόβλημα του Ιπτάμενου πλανώδιου πωλητή: Βελτιστοποίηση της παράδοσης δεμάτων με τη βοήθεια drone

Το FSTSP των [Murray και Chu \(2015\)](#), [9], εξετάζει ένα σύνολο πελατών c , καθένας από τους οποίους πρέπει να εξυπηρετηθεί ακριβώς μία φορά είτε από φορτηγό παράδοσης που λειτουργεί από τον οδηγό είτε από μη επανδρωμένο αεροσκάφος που λειτουργεί σε συντονισμό με το φορτηγό. Παρακάτω αναφέρονται οι συνθήκες λειτουργίας.

Παρόλο που το UAV μπορεί να επισκέπτεται μόνο έναν πελάτη ανά πτήση, το φορτηγό μπορεί να επισκεφθεί πολλούς πελάτες ενώ το UAV βρίσκεται σε πτήση. Το UAV θεωρείται ότι παραμένει σε συνεχή πτήση, εκτός από την παράδοση του δέματος σε πελάτη. Έτσι, κατά τον συντονισμό της επιστροφής στο φορτηγό, το UAV δεν μπορεί προσωρινά να προσγειωθεί κατά τη διαδρομή για εξοικονόμηση ενέργειας της μπαταρίας, εάν το UAV φτάσει πριν από το φορτηγό. Εάν το φορτηγό συλλέξει το UAV σε κάποιον κόμβο πελάτη i , το UAV μπορεί να εκτοξευθεί ξανά από το i . Ωστόσο, εάν το UAV εκτοξευτεί από το i , ενδέχεται να μην επιστρέψει στο φορτηγό στον κόμβο i . Εάν το τελευταίο σκέλος μιας πτήσης UAV περιλαμβάνει ένα ραντεβού με το φορτηγό, αυτό πρέπει να πραγματοποιηθεί στην τοποθεσία ενός πελάτη που εξυπηρετείται από το φορτηγό. Το UAV δεν μπορεί να επανασυνδεθεί με το φορτηγό σε κάποια ενδιάμεση θέση. Επιπλέον, το φορτηγό δεν επιτρέπεται να επισκέπτεται ξανά κανέναν κόμβο πελάτη για να ανακτήσει το UAV. Ούτε το UAV ούτε το φορτηγό επιτρέπεται να επισκεφθεί κανέναν κόμβο που δεν είναι πελάτης (εκτός φυσικά από την αποθήκη). Επιπλέον, κανένα όχημα δεν μπορεί να επισκεφθεί ξανά πελάτες. Σε περίπτωση που μια πτήση UAV τερματιστεί στην αποθήκη, το UAV τίθεται εκτός λειτουργίας (δηλαδή, δεν μπορεί να εκτοξευθεί ξανά από την αποθήκη).

ΚΕΦΑΛΑΙΟ 3

ΑΛΓΟΡΙΘΜΟΙ

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο παρουσιάζονται κάποιες κατηγορίες αλγορίθμων για την επίλυση των προβλημάτων που αναφέρθηκαν στο κεφάλαιο 2.

3.2 Ευρετικοί Αλγόριθμοι

Οι ευρετικοί αλγόριθμοι, σχετίζονται με τεχνικές βασισμένες στην εμπειρία για την επίλυση, την μάθηση και την ανακάλυψη. Όπου μια εξουθενωτική αναζήτηση δεν είναι πρακτική, οι ευρετικοί αλγόριθμοι χρησιμοποιούνται για να επιταχύνουν την διαδικασία εύρεσης μιας ικανοποιητικής λύσης. Παραδείγματα αυτής της μεθόδου συμπεριλαμβάνουν τον κανόνα του δεξιού χεριού, μια εικασία, μια διαισθητική κρίση ή την κοινή λογική. Με πιο ακριβείς όρους, οι ευρετικοί αλγόριθμοι είναι στρατηγικές, χρησιμοποιώντας εύκολα προσβάσιμες, αν και ελάχιστα εφαρμόσιμες πληροφορίες για τον έλεγχο της επίλυσης προβλημάτων στον άνθρωπο και τα ρομπότ. Στην επιστήμη των υπολογιστών, στην τεχνητή νοημοσύνη και στην μαθηματική βελτιστοποίηση ένας ευρετικός αλγόριθμος είναι μια τεχνική σχεδιασμένη για να λύνει προβλήματα πιο γρήγορα και για να βρει λύση προσεγγιστικά, όταν οι κλασικές μέθοδοι δεν μπορούν να βρουν καμία λύση. Ωστόσο δεν εγγυώνται ότι θα βρεθεί η καλύτερη λύση. Συνήθως αυτοί οι αλγόριθμοι βρίσκουν μια λύση γρήγορα κι εύκολα. Μερικές φορές μπορεί να είναι ακριβείς και να βρουν την καλύτερη δυνατή λύση, όμως ο αλγόριθμος θα αποκαλείται και πάλι ευρετικός μέχρι να αποδειχθεί ότι αυτή η λύση είναι όντως η καλύτερη. Οι κατηγορίες των ευρετικών αλγορίθμων είναι:

1. οι αλγόριθμοι απληστίας (greedy algorithms), οι οποίοι έχουν σκοπό την εύρεση εφικτής λύσης. Ωστόσο λόγω του ότι είναι μυωπικοί αλγόριθμοι, απαιτούν πολύ μεγάλο χρόνο.
2. οι προσεγγιστικοί αλγόριθμοι (approximation algorithms), οι οποίοι εστιάζουν στην λύση του πρόβληματος των αλγορίθμων απληστίας, αντλώντας παραπάνω πληροφορίες.
3. Αλγόριθμοι τοπικής αναζήτησης (local search algorithms). Οι αλγόριθμοι τοπικής αναζήτησης έχοντας μια αρχική εφικτή λύση επιδιώκουν την βελτίωση της, χρησιμοποιώντας κάποια μέθοδο αναζήτησης στην γειτονιά της λύσης.

3.2.1 Αλγόριθμοι απληστίας

Ένας άπληστος αλγόριθμος προσπαθεί να βρει τη βέλτιστη λύση σε ένα πρόβλημα εφαρμόζοντας μια ακολουθία επιλογών. Κάθε μια από τις επιλογές επιλέγει την καλύτερη λύση τη δεδομένη στιγμή σύμφωνα με κάποιο κριτήριο. Πιο συγκεκριμένα, τα περισσότερα προβλήματα αποτελούνται από τα δεδομένα.

Η πλειοψηφία των προβλημάτων έχει κάποια δεδομένα εισόδου και κάποιους περιορισμούς που πρέπει να ικανοποιούνται. Γενικά οι αλγόριθμοι απληστίας έχουν ως σκοπό να βρουν μια εφικτή λύση, η οποία να βελτιστοποιεί ή να ελαχιστοποιεί την αντικειμενική συνάρτηση. Αυτή η διαδικασία υλοποιείται μέσα από κάποια βήματα. Σε κάθε βήμα γίνεται έλεγχος ποιο από τα δεδομένα εισόδου θα χρησιμοποιηθεί, από το οποίο να προκύπτει μια εφικτή λύση. Η σειρά με την οποία χρησιμοποιούνται τα δεδομένα εισόδου εξαρτάται από κάποιο συγκεκριμένο κριτήριο. Κάποιοι αλγόριθμοι απληστίας παρουσιάζονται παρακάτω.

1. Πλησιέστερος γείτονας

Σε αυτή την διαδικασία ο αλγόριθμος ξεκινάει από έναν κόμβο, πηγαίνει σε κάθε επανάληψη στον αμέσως κοντινότερο κόμβο από αυτόν που βρίσκεται, τον οποίο δεν πρέπει να τον έχει ξαναεπισκεφθεί κι ο αλγόριθμος σταματάει όταν το όχημα ή τα οχήματα έχουν επισκεφθεί όλους τους κόμβους μια φορά. Τα βήματα που ακολουθεί:

Βήμα 1: Η διαδρομή ξεκινάει από έναν οποιονδήποτε κόμβο.

Βήμα 2: Πηγαίνουμε από τον κόμβο που βρισκόμαστε στον αμέσως κοντινότερο, εφόσον δεν τον έχουμε επισκεφθεί ξανά.

Βήμα 3: Επαναλαμβάνουμε το βήμα 2, έως ότου επισκεφθούμε όλους τους κόμβους.

2. Διαδικασία εισαγωγής κόμβων

Σε αυτόν τον αλγόριθμο απληστίας σχηματίζεται ένας κύκλος κόμβων. Ξεκινάμε από έναν οποιονδήποτε κόμβο και σε κάθε επανάληψη εισάγεται ένας νέος κόμβος, ο οποίος δεν είναι ήδη στην διαδρομή. Η επιλογή του επόμενου κόμβου, αλλά και σε ποιο σημείο του κύκλου θα τοποθετηθεί εξαρτάται από διάφορα κριτήρια.

3. Εγγύτερη συγχώνευση

Σε αυτή την περίπτωση έχουμε έναν αριθμό από κυκλικές διαδρομές k , όπου κάθε διαδρομή αποτελείται από έναν κόμβο. Στην συνέχεια κάθε μια κυκλική διαδρομή συγχωνεύεται με μια άλλη κυκλική διαδρομή, έχοντας ως κριτήριο την ελαχιστοποίηση του κόστους. Η ίδια διαδικασία συνεχίζεται έως ότου συγχωνευτούν όλες οι αρχικές κυκλικές διαδρομές, σχηματίζοντας μια μοναδική τελική κυκλική διαδρομή.

4. Εγγύτερη πρόσθεση

Η εγγύτερη πρόσθεση ξεκινάει από μια κυκλική διαδρομή ενός κόμβου. Σε κάθε επανάληψη επιλέγεται ένας κόμβος, ο οποίος θα ενωθεί με έναν υπάρχων κόμβο της κυκλικής διαδρομής με κριτήριο, την ελαχιστοποίηση του κόστους.

3.2.2 Προσεγγιστικοί αλγόριθμοι

Οι προσεγγιστικοί αλγόριθμοι χρησιμοποιούν επιπλέον πληροφορία για την επίλυση του προβλήματος. Ξεκινώντας από ένα τανυόν δέντρο γίνεται η μετατροπή του σε κύκλο του Euler και στην συνέχεια σε κύκλο του Hamilton. Τα περισσότερα προβλήματα VRP επιλύονται με τις προσεγγίσεις που παρουσιάζονται παρακάτω.

1. Ομαδοποίηση των κόμβων κι έπειτα δημιουργία των διαδρομών.
2. Σχηματισμός της διαδρομής κι έπειτα διαχωρισμός της διαδρομής και δημιουργία άλλων μικρότερων διαδρομών.
3. Στις διαδικασίες επαναποθήκευσης ή καταχώρησης έχουμε μια διαδρομή, κατα πάσα πιθανότητα μη εφικτή, η οποία συγκρίνεται με μια διαφορετική διαδρομή, η οποία επίσης μπορεί να μην είναι εφικτή, αλλά έχει μικρότερο κόστος. Η τελική διαδρομή είναι εφικτή.
4. Πραγματοποιείται το ίδιο με την διαδικασία 3. Ωστόσο σε αυτήν την περίπτωση κάθε επανάληψη, επιλέγεται μια λύση, η οποία είναι πάντα εφικτή.
5. Η προσέγγιση του μαθηματικού προγραμματισμού παίρνει προβλήματα που απαιτούν τέτοιου είδους μορφοποίηση.
6. Η αλληλεπιδρών βελτιστοποίηση, στην οποία μια ομάδα έμπειρων ατόμων έχοντας την κατάλληλη εμπειρία, καθορίζουν ή αλλάζουν τις παραμέτρους και την αντικειμενική προσπαθώντας να επιδιώξουν την λύση του προβλήματος.
7. Οι ακριβείς διαδικασίες, οι οποίες περιλαμβάνουν τον αλγόριθμο διακλάδωσης και οριοθέτησης (branch and bound) και τον αλγόριθμο τομής επιπέδων (cutting planes).

3.2.3 Αλγόριθμοι τοπικής αναζήτησης

Τα σημαντικότερα προβλήματα που πρέπει να αντιμετωπιστούν κατά τη διάρκεια της φάσης της σχεδίασης ενός αλγόριθμου τοπικής αναζήτησης είναι η επιλογή της γειτονιάς και αν ικανοποιούνται οι περιορισμοί που έχουν τεθεί. Σε δεύτερη φάση ο έλεγχος της ποιότητας της αρχικής λύσης, καθώς όσο καλύτερη είναι τόσο πιο εύκολα είναι να οδηγηθούμε πιο γρήγορα στην βελτίωση της λύσης και στην μέθοδο που θα χρησιμοποιηθεί. Παρακάτω παρουσιάζονται κάποιοι μέθοδοι τοπικής αναζήτησης.

2-OPT

Αυτή η μέθοδος αποτελείται γενικά από τη διαγραφή 2 ακμών και την επανασύνδεση δύο μονοπατιών με διαφορετικό τρόπο για να καθορίσουμε μια καινούρια διαδρομή.

3-OPT

Προσφέρει μεγαλύτερη ευελιξία από την 2-opt διαχωρίζοντας τον αλγόριθμο σε 3 μέρη αντί για 2.

1-0 επανατοποθέτηση (1-0 relocation)

Σε αυτήν την μέθοδο ένας κόμβος επανατοποθετείται από το σημείο που βρίσκεται σε κάποιο άλλο σημείο ίδιας ή διαφορετικής διαδρομής, εφόσον η αλλαγή αυτή είναι εφικτή και καλύτερη της αρχικής.

2-0 επανατοποθέτηση (2-0 relocation)

Σε αυτήν την μέθοδο 2 κόμβοι επανατοποθετούνται από το σημείο που βρίσκονται σε κάποιο άλλο σημείο ίδιας ή διαφορετικής διαδρομής, εφόσον η αλλαγή αυτή είναι εφικτή και καλύτερη της αρχικής.

1-1 ανταλλαγή (1-1 reexchange)

Σε αυτήν την μέθοδο 1 κόμβος ανταλλάσσεται με 1 άλλον κόμβο από την ίδια ή άλλη διαδρομή, εφόσον η αλλαγή είναι εφικτή και καλύτερη της αρχικής.

2-2 ανταλλαγή (2-2 reexchange)

Σε αυτήν την μέθοδο 2 κόμβοι ανταλλάσσονται με 2 άλλους διαφορετικούς κόμβους από την ίδια ή άλλη διαδρομή, εφόσον η αλλαγή είναι εφικτή και καλύτερη της αρχικής.

3.3 Μεθευρετικοί Αλγόριθμοι

Οι μεθευρετικοί αλγόριθμοι είναι μια μέθοδος βελτιστοποίησης ενός προβλήματος έχοντας ως δεδομένο ένα μέτρο ποιότητας. Οι αλγόριθμοι αυτοί κάνουν είτε λίγες είτε καμία υπόθεση για την βελτιστοποίηση του προβλήματος και μπορούν να αναζητήσουν πολύ μεγάλες λίστες υποψηφίων. Ωστόσο δεν εγγυώνται ότι θα βρεθεί η βέλτιστη λύση. Πιο συγκεκριμένα χρησιμοποιούνται για να ξεκολλήσει μια λύση από το τοπικό ελάχιστο. Σε κάποια βήματα τους μπορεί να οδηγήσουν σε μη εφικτές λύσεις, αλλά με αυτόν τον τρόπο μπορεί μια λύση να αποφύγει κάποιο τοπικό ελάχιστο.

3.3.1 Διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης (Greedy Randomized Adaptive Search Procedure (Grasp))

Η διαδικασία άπληστης τυχαιοποιημένης αναζήτησης είναι μια επαναληπτική διαδικασία, η οποία αποτελείται από δύο φάσεις, την φάση κατασκευής, στην οποία παράγεται μια εφικτή λύση και

την φάση τοπικής αναζήτησης, στην οποία βρίσκεται το τοπικό βέλτιστο της κατασκευασμένης γειτονιάς. Το αποτέλεσμα της διαδικασίας είναι η καλύτερη λύση συνολικά.

Στην φάση κατασκευής, κατασκευάζεται επαναληπτικά μια εφικτή λύση, ένα στοιχείο την φορά. Ο τρόπος υλοποίησης αυτής της φάσης γίνεται περιορίζοντας την μεγάλη λίστα υποψηφίων που έχουμε σε μια μικρότερη, που ονομάζεται λίστα περιορισμού των υποψηφίων (Restricted Candidate List). Υπάρχουν δύο στρατηγικές κατασκευής αυτής της λίστας.

Στην πρώτη στρατηγική επιλέγεται τυχαία ένας αριθμός υποψηφίων από την αρχική λίστα και εισχωρείται στην μικρότερη λίστα. Έπειτα, από την μικρότερη λίστα υποψηφίων, επιλέγεται ο υποψήφιος με την μικρότερη απόσταση από τον τελευταίο κόμβο της γειτονιάς. Στην συνέχεια αφαιρείται ο υποψήφιος από την αρχική λίστα υποψηφίων και η διαδικασία επαναλαμβάνεται μέχρις ότου αδειάσει η αρχική λίστα.

Στην δεύτερη στρατηγική επιλέγεται από την αρχική λίστα υποψηφίων ένας αριθμός υποψηφίων με κριτήριο την απόσταση που έχουν από τον τελευταίο κόμβο της γειτονιάς και προστίθενται στην περιορισμένη λίστα υποψηφίων. Στην συνέχεια από την περιορισμένη λίστα, επιλέγεται τυχαία πλέον ένας από τους υποψηφίους. Έπειτα αφαιρείται ο υποψήφιος από την αρχική λίστα υποψηφίων και η διαδικασία επαναλαμβάνεται μέχρις ότου αδειάσει η αρχική λίστα

Η κατασκευή της λίστας περιορισμού των υποψηφίων είναι ένα πολύ σημαντικό κομμάτι. Αν, για παράδειγμα, η λίστα είναι μικρή τότε οι λύσεις που θα παράγονται παρουσιάζουν αρκετές ομοιότητες. Αντιθέτως, αν η λίστα είναι πολύ μεγάλη τότε η κατασκευή της λύσης θα περιλαμβάνει μεγάλο ποσοστό τυχαιοποίησης

Η φάση τοπικής αναζήτησης είναι αναγκαία, καθώς η φάση κατασκευής δεν εγγυάται ότι θα δώσει την βέλτιστη λύση. Για να εφαρμοστεί η φάση τοπικής αναζήτησης, καθορίζεται μια συνάρτηση που να κάνει αναζήτηση στη γειτονιά της αρχικής λύσης.

Κάποιες φορές μπορεί να χρησιμοποιηθεί μια τελευταία φάση όπου κάποιες από τις καλύτερες λύσεις συνδυάζονται με την καινούρια λύση, ώστε να βελτιωθεί η ποιότητα της λύσης.

Παρακάτω παρουσιάζεται μια γενική μορφή του αλγόριθμου:

$$x^* = \infty$$

Για όσο η συνθήκη τερματισμού του GRASP δεν ικανοποιείται

κάνε

Αρχικοποίηση μια λύσης x

Τοπική αναζήτηση στην x

Εάν $x < x^*$ **τότε**

$$x^* = x$$

Τέλος_Εάν

Τέλος_Για

Επέστρεψε x^*

3.3.2 Αλγόριθμος μεταβλητής γειτονιάς αναζήτησης (Variable Neighborhood Search (VNS))

Η μεταβλητή γειτονιά αναζήτησης εκμεταλεύεται συστηματικά την ιδέα της αλλαγής της γειτονιάς, τόσο σε κάθοδο στα τοπικά ελάχιστα όσο και σε αποφυγή από τις γειτονιές που τα περιέχουν. Για να εφαρμοστεί ο αλγόριθμος, πρέπει να έχει κατασκευαστεί μια λύση στον χώρο X . Για να ορίσουμε γειτονιές, χρειαζόμαστε μια συνάρτηση απόστασης μεταξύ δύο λύσεων. Ο αλγόριθμος μεταβλητής γειτονιάς αναζήτησης εκμεταλεύεται συστηματικά τις ακόλουθες παρατηρήσεις: (i) ένα τοπικό ελάχιστο μιας γειτονιάς δεν είναι απαραίτητα τοπικό ελάχιστο μιας άλλης, (ii) ένα ολικό ελάχιστο είναι τοπικό ελάχιστο σε όλες τις κατασκευασμένες γειτονιές, (iii) για πολλά προβλήματα τα τοπικά ελάχιστα μερικών λύσεων είναι κοντά το ένα με το άλλο.

Ο αλγόριθμος συνδυάζει ντετερμινιστική και στοχαστική αναζήτηση. Μια σειρά από κατασκευασμένες γειτονιές, οι οποίες καθορίζουν γειτονιές γύρω από οποιοδήποτε σημείο του χώρου, που επιλέχθηκε πρώτα. Μετά η τοπική αναζήτηση χρησιμοποιείται και οδηγεί σε ένα τοπικό βέλτιστο x . Ένα σημείο επιλέγεται τυχαία στην πρώτη γειτονιά και πραγματοποιείται μια κάθοδος, οδηγώντας σε ένα καινούριο τοπικό ελάχιστο.

Ο αλγόριθμος της μεταβλητής γειτονιάς αναζήτησης βασίζεται στον αλγόριθμο της μεταβλητής γειτονιάς καθόδου. Ο αλγόριθμος μεταβλητής γειτονιάς καθόδου χρησιμοποιεί διαδοχικές γειτονιές αναζήτησης σε κατάβαση προς ένα τοπικό ελάχιστο.

Αρχικά κάποιος πρέπει να καθορίσει το σύνολο των γειτονιών (N_i , όπου $i=1, \dots, imax$). Έστω ότι η N_1 είναι η πρώτη γειτονιά και s είναι η αρχική λύση. Εάν δεν μπορεί να υπάρξει βελτίωση στη λύση της s σε κάποια επανάληψη της N_i τότε η γειτονιά αναζήτησης αλλάζει και γίνεται N_{i+1} .

Εάν μια βελτίωση στην τρέχουσα λύση βρεθεί τότε αυτό που γίνεται είναι ότι η αναζήτηση ξεκινάει από την αρχή.

Αυτή η αναζήτηση θα είναι αποτελεσματική αν οι γειτονιές που χρησιμοποιούνται είναι συμπληρωματικές μεταξύ τους που σημαίνει ότι ένα τοπικό βέλτιστο της μιας γειτονιάς δεν πρέπει να είναι τοπικό βέλτιστο της άλλης γειτονιάς.

Η κάθε επανάληψη του αλγόριθμου αποτελείται από τρία βήματα: ανακίνηση(shaking), τοπική αναζήτηση (local search) και κίνηση (move). Κάθε επανάληψη δημιουργεί μια καινούρια γειτονιά αναζήτησης. Εάν το κόστος είναι μικρότερο τότε αντικαθίσταται η παλιά λύση από την καινούρια.

Στην συνέχεια η ίδια διαδικασία αρχίζει από την αρχή με τη χρήση της γειτονιάς N_1 και τη καινούρια λύση s'' που βρήκαμε προηγούμενα.

Αν δεν βρεθεί μία καλύτερη λύση τότε ο αλγόριθμος προχωρά στην επόμενη γειτονιά αναζήτησης

3.3.3 Προσομοιωμένη ανόπτηση (Simulated Annealing (SA))

Η προσομοιωμένη ανόπτηση είναι ένας μεθευρετικός αλγόριθμος που προέρχεται από μια αναλογία του τρόπου, με τον οποίο βρίσκεται μια διαμόρφωση βέλτιστου ατόμου στη στατιστική μηχανική. Χρησιμοποιεί τη θερμοκρασία ως ρητή στρατηγική για την καθοδήγηση στην αναζήτηση. Ο χώρος λύσεων συνήθως εξερευνάται κάνοντας τυχαίες προσπάθειες. Η διαδικασία της προσομοιωμένης ανόπτησης, παράγει τυχαία έναν μεγάλο αριθμό από τυχαίες λύσεις, κρατώντας και τις καλές και τις κακές λύσεις. Όσο η προσομοίωση εξελίσσεται, οι απαιτήσεις για την αντικατάσταση μια λύσης γίνεται όλο και πιο αυστηρές. Πιο συγκεκριμένα κατά την διάρκεια αυτής της διαδικασίας το υλικό θερμαίνεται μέχρι το σημείο τήξης και έπειτα ψύχεται αργά. Η αρχική θερμοκρασία δεν πρέπει να είναι πολύ υψηλή, διότι τότε όλες οι λύσεις γίνονται αποδεκτές. Ωστόσο δεν πρέπει να είναι ούτε και πολύ μικρή, διότι δεν γίνεται αποδεκτή καμία λύση που να χειροτερεύει την υπάρχουσα λύση, οπότε δεν μπορούμε να αποφύγουμε εύκολα κάποιο τοπικό ελάχιστο. Γενικά, η διαδικασία παίρνει ένα μικρό σύνολο βέλτιστων λύσεων. Το πλεονέκτημα αυτής της μεθόδου είναι ότι έχει την ικανότητα να αποφεύγει τα τοπικά ελάχιστα.

3.3.4 Μέθοδος αποδοχής κατωφλιού (Threshold Accepted (TA))

Ορίζεται έναν κατώφλι. Οι λύσεις που παράγονται και είναι πάνω από αυτό δεν γίνονται αποδεκτές, σε αντίθεση με αυτές που βρίσκονται κάτω από αυτό το κατώφλι. Καλό είναι στις αρχικές επαναλήψεις το κατώφλι να είναι υψηλό και στην πορεία να μικραίνει.

3.3.5 Περιορισμένη αναζήτηση (Tabu Search (TS))

Η περιορισμένη αναζήτηση είναι μια τεχνική που παρακολουθεί τις περιοχές του χώρου, οι οποίες έχουν ήδη αναζητηθεί έτσι ώστε να αποφευχθεί η αναζήτηση κοντά σε αυτές τις περιοχές. Ξεκινάει από μια τυχαία αρχική λύση και μετακινείται διαδοχικά σε έναν από τους γείτονες της τωρινής λύσης. Η διαφορά αυτής της μεθόδου βασίζεται στην λίστα tabu, που είναι μια ειδική βραχυπρόθεσμη μνήμη. Ουσιαστικά, αποθηκεύεται μόνο μερικές πληροφορίες των λύσεων και όχι όλες τις λύσεις. Έτσι αποφεύγεται η επίσκεψη ίδιων λύσεων και ο εγκλωβισμός σε κάποιον κύκλο αλλά και σε τοπικό βέλτιστο.

3.3.6 Αλγόριθμος επανασύνδεσης διαδρομών (Path relinking (PR))

Σε αυτόν τον αλγόριθμο βρίσκονται διαδρομές για την δημιουργία ενός μονοπατιού, το οποίο συνδέει μια αρχική λύση με μια τελική λύση. Η αρχική λύση είναι η καλύτερη τις περισσότερες

φορές, ενώ η τελική η χειρότερη και μπορεί να αντικατασταθεί από τις λύσεις που θα βρεθούν στην πορεία.

3.3.7 Αλγόριθμος επαναληπτικής τοπικής αναζήτησης (Iterated Local Search (ILS))

Η επαναληπτική τοπική αναζήτηση είναι μια απλή και γενικά εφαρμόσιμη μεθυσρετική μέθοδος που εφαρμόζει επαναληπτικά την τοπική αναζήτηση σε τροποποιήσεις του τρέχοντος σημείου. Για να εφαρμοστεί ο αλγόριθμος πρέπει να καθοριστούν τέσσερα στοιχεία. Υπάρχουν τρεις μηχανισμοί που δημιουργούν μια αρχική λύση. Πρώτον η διαδικασία Modify, τροποποιεί την τρέχουσα λύση και οδηγεί σε κάποια ενδιάμεση λύση. Δεύτερον η διαδικασία LocalSearch, που φτάνει στο τοπικό ελάχιστο και τρίτον η AcceptanceCriterion, η οποία αποφασίζει ποια λύση θα εφαρμοστεί την επόμενη φορά στην διαδικασία Modify. Σε αντίθεση με τις άλλες μεθόδους δεν ακολουθεί μια τροχιά στον χώρο αναζήτησης, αλλά τροποποιήσεις λύσεων που αντιστοιχούν σε άλματα στο χώρο αναζήτησης, έτσι ώστε να επιτραπεί η έξοδος από το τοπικό ελάχιστο.

3.3.8 Αλγόριθμος καθοδηγούμενης τοπικής αναζήτησης (Guided Local Search (GLS))

Σε αυτήν την μέθοδο ορίζεται ένα σύνολο χαρακτηριστικών για τις υποψήφιες λύσεις. Η καινοτομία αυτής της διαδικασίας είναι ο τρόπος που επιλέγει τα χαρακτηριστικά που θα περιοριστούν. Κατανέμει αποτελεσματικά την προσπάθεια αναζήτησης στον χώρο, ευνοώντας τις υποσχόμενες περιοχές. Οι τροποποιήσεις περιορισμών κανονικοποιούν τις λύσεις που έχουν βρεθεί από την τοπική αναζήτηση σύμφωνα με προηγούμενες πληροφορίες που θα βοηθήσουν στην επίλυση του προβλήματος.

3.3.9 Αλγόριθμος προσαρμοστικής αναζήτησης (Adaptive Memory (AM))

Ο αλγόριθμος της προσαρμοστικής μνήμης είναι μια από τις βελτιώσεις του αλγόριθμου της περιορισμένης αναζήτησης. Έχει συγκεκριμένα βήματα. Αρχικά υπάρχει μια μνήμη που έχει αποθηκευμένες τις καλές λύσεις. Έπειτα συνδυάζονται οι λύσεις αυτές για την δημιουργία μιας καινούριας λύσης. Αν η λύση δεν είναι εφικτή, τότε ακολουθείται μια διαδικασία βελτίωσης της λύσης και τέλος η λύση βελτιώνεται ξανά.

3.4 Γενετικοί Αλγόριθμοι

Οι γενετικοί αλγόριθμοι στοχεύουν στην εξεύρεση λύσεων σε προβλήματα NP-hard. Η βασική ιδέα αυτών των αλγόριθμων είναι αρχικά να παραχθεί ένας αρχικός πληθυσμός τυχαία που αποτελείται από λύσεις, που ονομάζονται άτομα ή Χρωμοσώματα και μετά να εξελίξει αυτόν τον πληθυσμό μετά από έναν αριθμό επαναλήψεων, οι οποίες ονομάζονται γενιές. Κατά την διάρκεια

κάθε γενιάς, κάθε χρωμόσωμα αξιολογείται, χρησιμοποιώντας κάποιο μέτρο φυσικής κατάστασης. Η δημιουργία της επόμενης γενιάς νέων χρωμοσωμάτων, που ονομάζονται απόγονοι, σχηματίζεται είτε με τη συγχώνευση δύο χρωμοσωμάτων από την τρέχουσα παραγωγή χρησιμοποιώντας έναν τελεστή διασταύρωσης ή τροποποιώντας ένα χρωμόσωμα χρησιμοποιώντας έναν τελεστή μετάλλαξης. Μια νέα γενιά σχηματίζεται με επιλογή των τιμών ικανότητας, ορισμένου από τους γονείς και τους απογόνους και την απόρριψη άλλων, έτσι ώστε να διατηρείται σταθερό το μέγεθος του πληθυσμού. Τα καταλληλότερα χρωμοσώματα έχουν μεγαλύτερες πιθανότητες να επιλεγθούν. Μετά από αρκετές γενιές, οι αλγόριθμοι πλησιάζουν στο καλύτερο χρωμόσωμα, το οποίο αντιπροσωπεύει τη βέλτιστη ή υποβέλτιστη λύση του προβλήματος.

Η αρχικοποίηση του πληθυσμού θα ήταν καλό να γίνει με έναν τρόπο, όπως είναι η τυχαιοποίηση στις τιμές για κάθε μεταβλητή του ατόμου του πληθυσμού, έτσι ώστε να είναι πιο ομοιόμορφα κατανομημένος.

Ο πληθυσμός δεν θα πρέπει να είναι ούτε μικρός, γιατί δεν θα είναι αντιπροσωπευτικός, αλλά ούτε και μεγάλος, διότι δεν θα είναι αποτελεσματικός και γρήγορος ο αλγόριθμος.

3.4.1 Βελτιστοποίηση μεταναστευτικών πτηνών (MBO)

Ο σχηματισμός V είναι ο πιο διάσημος σχηματισμός που τα αποδημητικά πουλιά χρησιμοποιούν για να πετούν μεγάλες αποστάσεις. Ένα πουλί οδηγεί το κοπάδι και δύο σειρές άλλων πτηνών που το ακολουθούν. Στο σχηματισμό αυτόν το πουλί ηγέτης είναι αυτό που ξοδεύει περισσότερη ενέργεια. Τα πουλιά στις άλλες θέσεις επωφελούνται από τα πουλιά μπροστά του. Η εξοικονόμηση ενέργειας των πουλιών που βρίσκονται πιο πίσω λέγεται ότι είναι μεγαλύτερη.

3.4.2 Βελτιστοποίηση σωματιδίων (PSO)

Αυτή η μέθοδος αναπτύχθηκε από τους [Ebenhart και Kennedy το 1995, \[2\]](#). Η βελτιστοποίηση προήλθε από την παρακολούθηση ορισμένων σμήνων. Ο αλγόριθμος δεν χρησιμοποιεί μόνο μια ορμή του μεμονωμένου σωματιδίου στην κατεύθυνση προς την οποία είναι έτοιμο να κινηθεί, αλλά χρησιμοποιεί επίσης μια πληροφορία της <<προηγούμενης καλύτερης λύσης>> αυτού του σωματιδίου που ονομάζεται προσωπική βέλτιστη.

3.4.3 Αλγόριθμος τεχνητής αποικίας μελισσών (ABC)

Ο αλγόριθμος προτάθηκε από τον [karaboga \[2\]](#) για την βελτιστοποίηση αριθμητικών προβλημάτων. Ο αλγόριθμος προσομοιώνει την έξυπνη συμπεριφορά αναζήτησης τροφής σμήνων μελισσών. Στον αλγόριθμο, η αποικία των τεχνητών μελισσών περιέχει τρεις ομάδες μελισσών: τις εργάτριες, τους θεατές και τους προσκόπους. Η μέλισσα που επιλέγει μια πηγή τροφής είναι η

μέλισσα θεατής, ενώ η εργάτρια είναι αυτή που πηγαίνει σε αυτήν την πηγή και τέλος η μέλισσα πρόσκοπος πραγματοποιεί τυχαία αναζήτηση για την ανακάλυψη νέων πηγών.

3.4.4 Firefly algorithm

Ο αλγόριθμος Firefly αναπτύχθηκε για πρώτη φορά από τον [Xin-She Yang \[2\]](#) και βασίστηκε στα μοτίβα που αναβοσβήνουν οι πυγολαμπίδες. Οι πυγολαμπίδες είναι unisex, έτσι ώστε μια πυγολαμπίδα να προσελκύεται κι από άλλες ανεξάρτητα από το φύλο τους. Η ελκυστικότητα είναι ανάλογη με τη φωτεινότητα και μειώνονται και οι δύο όσο αυξάνεται η απόσταση τους. Η λιγότερο φωτεινή πυγολαμπίδα οδηγείται στην πιο φωτεινή. Εάν δεν υπάρχει φωτεινότερη πυγολαμπίδα από μια συγκεκριμένη τότε αυτή κινείται τυχαία. Η φωτεινότητα μιας πυγολαμπίδας καθορίζεται από το τοπίο της αντικειμενικής συνάρτησης.

3.4.5 Αλγόριθμος αναζήτησης κούκου (CSA)

Ο αλγόριθμος αναζήτησης Cuckoo εμπνέεται από την συμπεριφορά αναπαραγωγής των κούκων που γεννούν τα αυγά στις φωλιές των πουλιών υποδοχής. Κάθε αυγό αντιπροσωπεύει μια λύση και κάθε νέο αυγό κούκου μια νέα λύση. Χρησιμοποιούνται οι παρακάτω κανόνες. Κάθε κούκος γεννά ένα αυγό τη φορά και το πετάει σε μια τυχαία φωλιά. Οι καλύτερες φωλιές με υψηλή ποιότητα αυγών θα μεταφερθούν στις επόμενες γενιές. Ο αριθμός των διαθέσιμων φωλιών είναι σταθερός κι ένας οικοδεσπότης μπορεί να ανακαλύψει ένα ξένο αυγό με πιθανότητα $Pa[0,1]$.

ΚΕΦΑΛΑΙΟ 4

ΜΑΘΗΜΑΤΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΟΥ EM-SDRPPD-BRS

4.1 Μοντελοποίηση

Έστω $S = \{1, \dots, n\}$ το σύνολο των κόμβων παραλαβής και $B = \{n+1, \dots, 2n\}$ το σύνολο κόμβων παράδοσης, που αντιστοιχεί στο σύνολο των πωλητών και των αγοραστών, αντίστοιχα. $R = \{2n+1, \dots, 2n+s\}$ είναι το σύνολο των σταθμών αντικατάστασης μπαταρίας. Η ένωση και των δύο κόμβων τύπου πελάτη είναι $N = SUB$. Οι κόμβοι 0 και $2n+1$ αντιπροσωπεύουν τις τοποθεσίες βάσης, από τις οποίες εκτοξεύεται και επιστρέφει το drone αντίστοιχα. Η απόσταση μεταξύ δύο οποιωνδήποτε κόμβων δίνεται από το D_{ij} . Έστω i το αίτημα για μεταφορά δεμάτων μεταξύ του κόμβου παραλαβής i και του κόμβου παράδοσης $n+i$. Κάθε αίτημα i αντιστοιχεί στην παραλαβή και παράδοση ενός δέματος το οποίο ζυγίζει P_i . Κάθε σταθμός αντικατάστασης μπαταριών έχει M διαθέσιμες μπαταρίες, που μπορεί να χρησιμοποιήσει το drone, ανταλλάσσοντας τη χρησιμοποιημένη μπαταρία του με μια πλήρως φορτισμένη.

Το EM-SDRPPD-BRS ορίζεται σε ένα κατευθυνόμενο γράφημα $G = (V, A)$ όπου $V = N \cup R \cup \{0, 2n+1\}$ είναι το σύνολο κόμβων και A είναι το σύνολο τόξων. Το drone ζυγίζει μονάδες U και μπορεί να μεταφέρει έως και Q δέματα ταυτόχρονα, το συνολικό βάρος των οποίων δεν μπορεί να υπερβαίνει το μέγιστο ωφέλιμο φορτίο, βάρος Φ . Η μέγιστη ενεργειακή χωρητικότητα του drone είναι E . Έστω f_{ij} το συνολικό βάρος του ωφέλιμου φορτίου του drone κατά τη διέλευση του από το τόξο (i,j) , t_i ο χρόνος αναχώρησης από τον κόμβο i και w_i , l_i το συνολικό βάρος και η συνολική ποσότητα, αντίστοιχα, των δεμάτων μετά την εξυπηρέτηση στον κόμβο $i \in V$. Μια περιεκτική περιγραφή των συνόλων, των παραμέτρων και των μεταβλητών που χρησιμοποιούνται, παρουσιάζονται στον Πίνακα 1.

Η μαθηματική διατύπωση του EM-SDRPPD-BRS είναι η εξής:

$$\min F = \sum_{i \in V} \sum_{j \in V} (D_{ij} \times (U + f_{ij}) \times x_{ij}) \quad (1)$$

και υπακούει σε:

$$\sum_{j \in N \cup \{2n+s+1\}} x_{ij} = 1, \forall i \in S \cup R \quad (2)$$

$$\sum_{j \in N} x_{ij} - \sum_{j \in N} x_{j,n+i} = 0, \forall i \in S \quad (3)$$

$$\sum_{i \in NU\{0\}} x_{ij} \leq M, \forall j \in R \quad (4)$$

$$\sum_{j \in NURU\{2n+s+1\}} x_{0j} = 1 \quad (5)$$

$$\sum_{i \in NURU\{0\}} x_{ij} - \sum_{i \in NURU\{2n+s+1\}} x_{ji} = 0, \forall j \in N \quad (6)$$

$$\sum_{i \in BURU\{0\}} x_{i,2n+s+1} = 1 \quad (7)$$

$$x_{ij}(t_i + T_{ij} + t_j) \leq 0, \forall (i, j) \in A \quad (8)$$

$$t_i + T_{i,n+i} \leq t_{n+i}, \forall i \in S \quad (9)$$

$$x_{ij}(l_i + L_j - l_j) = 0, \forall (i, j) \in A \quad (10)$$

$$L_i \leq l_i \leq Q, \forall i \in S \quad (11)$$

$$0 \leq l_{n+1} \leq Q - Li, \forall n + i \in S \quad (12)$$

$$l_0 = 0 \quad (13)$$

$$x_{ij}(w_i + W_j - w_j) = 0, \forall (i, j) \in A \quad (14)$$

$$W_i \leq w_i \leq \Phi \quad \forall i \in S \quad (15)$$

$$0 \leq w_{n+1} \leq \Phi - W_i, \forall n + i \in S \quad (16)$$

$$w_0 = 0 \quad (17)$$

$$f_{ij} = w_j - W_j, \forall (i, j) \in A \quad (18)$$

$$0 \leq f_{ij} \leq \Phi \times x_{ij}, \forall (i, j) \in A \quad (19)$$

$$\sum_{i \in V} \sum_{j \in V} (D_{ij} \times (U + f_{ij}) \times x_{ij}) \leq E \times \sum_{r \in RU\{0\}} \sum_{j \in NURU\{2n+s+1\}} x_{ij} \quad (20)$$

$$e_j = \begin{cases} e_j, & \text{εάν } j \in N \\ E, & \text{εάν } j \in R \cup \{0\} \end{cases} \quad (21)$$

$$x_{ij} (e_i - (D_{ij} * (U + f_{ij})) + e_j) = 0, \forall (i, j) \in A \quad (22)$$

$$0 \leq e_j \leq E, \forall j \in V \quad (23)$$

$$x_{ij} \in \{0,1\}, \forall (i, j) \in A \quad (24)$$

$$Li = \begin{cases} -1, & \text{εάν } i \in B \\ 1, & \text{εάν } i \in S \\ 0, & \text{ειδήλως} \end{cases} \quad (25)$$

$$W_i = \begin{cases} -P_i, & \text{εάν } i \in B \\ P_i, & \text{εάν } i \in S \\ 0, & \text{εισάλλως} \end{cases} \quad (26)$$

Η αντικειμενική συνάρτηση (1) ελαχιστοποιεί την συνολική κατανάλωση ενέργειας. Οι περιορισμοί (2) και (3) εξασφαλίζουν ότι κάθε ζεύγος των κόμβων παραλαβής και παράδοσης εξυπηρετείται ακριβώς μία φορά από το ίδιο drone. Ο περιορισμός (4) επιτρέπει στους σταθμούς αντικατάστασης μπαταρίας να επισκέπτονται Μ φορές το πολύ. Οι περιορισμοί (5) – (7) αναγκάζουν το drone να ξεκινήσει τη διαδρομή από τη βάση και να επιστρέψει σε αυτή αφού ολοκληρώσει τη διαδρομή του. Οι περιορισμοί (8) και (9) αναγκάζουν το drone να επισκεφθεί τον κόμβο παραλαβής πριν από τον αντίστοιχο κόμβο παράδοσης. Οι περιορισμοί (10) – (12) υποβάλλουν τους ποσοτικούς περιορισμούς στο drone, ενώ ο (13) επιβάλλει να ξεκινήσει τη διαδρομή του άδειο. Ομοίως με τους ποσοτικούς περιορισμούς, οι (14) – (16) επιβάλλουν τους περιορισμούς βάρους του ωφέλιμου φορτίου στο drone ταυτόχρονα για τους κόμβους παραλαβής και παράδοσης ενώ ο (17) διασφαλίζει ότι αρχικά δεν φέρει ωφέλιμο φορτίο. Οι περιορισμοί (18) και (19) περιορίζουν το βάρος του ωφέλιμου φορτίου τόξου μεταξύ 0 και Φ για τα τόξα που χρησιμοποιούνται από το drone και το αναγκάζει να μηδενιστεί διαφορετικά, ενώ ο (20) επιβάλλει ένα όριο στη συνολική κατανάλωση ενέργειας του drone. Ο περιορισμός (21) απαιτεί ότι όταν το drone αναχωρεί από μια σταθμευμένη στάση φόρτισης, η ενέργειά του ε επαναφέρεται στη μέγιστη τιμή E, ενώ ο (22) επιβάλλει όρια στην υπολοιπόμενη ενέργεια του drone. Ο (23) περιορίζει τις τιμές ενέργειας του drone που φθάνει και αναχωρεί από κόμβους, μεταξύ του 0 και της μέγιστης χωρητικότητας της μπαταρίας E. Τέλος, ο (24) καθορίζει το πεδίο των δυαδικών μεταβλητών, ο (25) ορίζει τις ποσότητες σε κάθε κόμβο ανάλογα με τον τύπο του και ο (26) ορίζει το βάρος του πακέτου ανάλογα με το αν παραλαμβάνεται ή παραδίδεται.

Πίνακας 1: Σύνολα, παράμετροι και μεταβλητές της μοντελοποίησης του EM-SDRPPD-BRS.

Σύνολα και Χαρακτηριστικά Κόμβων

S Σύνολο πωλητών, $S = \{1, \dots, n\}$

B Σύνολο αγοραστών, $B = \{n + 1, \dots, 2n\}$

N Σύνολο πελατών, $N = S \cup B$

R Σύνολο σταθμών αντικατάστασης μπαταρίας, $R = \{2n + 1, \dots, 2n + s\}$

V Σύνολο κόμβων, $V = N \cup R \cup \{0, 2n + s + 1\}$

A Σύνολο τόξων, $A = \{(i, j) | i \in V, j \in V, i \neq j\}$

D_{ij} Απόσταση από τον κόμβο i στον κόμβο j

T_{ij} Χρόνος ταξιδιού από τον κόμβο i στον κόμβο j

Παράμετροι του Drone

Q Μέγιστη ποσότητα ωφέλιμου φορτίου του drone

Φ Μέγιστο βάρος ωφέλιμου φορτίου του drone

E Μέγιστη ενεργειακή χωρητικότητα του drone

U Το βάρος του drone, χωρίς ωφέλιμο φορτίο

Παράμετροι πελάτη

W_i Βάρος ωφέλιμου φορτίου που προστέθηκε ή αφαιρέθηκε στον κόμβο i

P_i Βάρος συσκευασίας που παραλήφθηκε ή παραδόθηκε στον κόμβο i

L_i Ποσότητα ωφέλιμου φορτίου που παραλήφθηκε ή παραδόθηκε στον κόμβο i

Μεταβλητές

x_{ij} Μεταβλητή απόφασης του drone που διασχίζει το τόξο (i, j)

w_i Συνολικό βάρος ωφέλιμου φορτίου του drone μετά την επίσκεψη στον κόμβο i

l_i Συνολική ποσότητα ωφέλιμου φορτίου του drone μετά την επίσκεψη στον κόμβο i

t_i Ο χρόνος αναχώρησης του drone από τον κόμβο i

e_i Υπολοιπόμενη ενέργεια του drone μετά την επίσκεψη στον κόμβο i

f_{ij} Συνολικό βάρος ωφέλιμου φορτίου του drone κατά τη διέλευση του τόξου (i, j)

ΚΕΦΑΛΑΙΟ 5

ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΜΕ ΤΟΝ ΥΒΡΙΔΙΚΟ ΑΛΓΟΡΙΘΜΟ GRASP/VNS

5.1 Δεδομένα

Για την επίλυση του παραπάνω προβλήματος (Energy Minimizing Single-Drone Routing Problem with Charging Stations) έχει χρησιμοποιηθεί ο υβριδικός αλγόριθμος GRASP/VNS. Η αρχικοποίηση της λύσης γίνεται μέσω της άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης (Grasp) και η βελτιστοποίηση της ενέργειας του drone μέσω κάποιων μεθόδων τοπικής αναζήτησης (Local Search 2-opt, 1-0, 2-0 relocation, 1-1, 2-2 reexchange και changing storage).

Για την επίλυση του παραπάνω προβλήματος (Energy Minimizing Single-Drone Routing Problem with Charging Stations) έχει χρησιμοποιηθεί ένας υβριδικός αλγόριθμος, ο οποίος αποτελείται από τον αλγόριθμο άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης (Grasp) για την αρχικοποίηση της γειτονιάς και από τον αλγόριθμο Μεταβλητής Γειτονιάς Αναζήτησης (VNS) για την βελτιστοποίηση της ενέργειας του drone μέσω κάποιων μεθόδων τοπικής αναζήτησης (Local Search 2-opt, 1-0, 2-0 relocation, 1-1, 2-2 reexchange και changing storage).

Ως δεδομένα του προβλήματος, τα οποία διαβάζονται από ένα αρχείο data.txt έχουμε:

N κόμβους, M αποθήκες, την μέγιστη ενέργεια της κάθε μπαταρίας του drone (1000), το μέγιστο βάρος (4), τον μέγιστο αριθμό πακέτων (3), τις συντεταγμένες των κόμβων και πληροφορίες για κάθε κόμβο-πελάτη linehaul (σε ποιον πελάτη προορίζεται το αντίστοιχο πακέτο και το βάρος του πακέτου).

Πιο αναλυτικά λοιπόν:

Δημιουργείται ο πίνακας αποστάσεων με την βοήθεια των συντεταγμένων που δίνονται και τον τύπο της ευκλείδειας απόστασης:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (27)$$

Έχουμε N κόμβους στον δυσδιάστατο χώρο, εκ των οποίων οι M από αυτούς είναι αποθήκες. Οι υπόλοιποι N-M είναι πελάτες, οι οποίοι χωρίζονται σε 2 υποσύνολα. Το πρώτο υποσύνολο είναι οι πελάτες που απαιτούν την διανομή του προϊόντος (linehauls customers). Το δεύτερο υποσύνολο αποτελείται από τους πελάτες που παραλαμβάνουν το προϊόν (backhauls customers) από ένα πελάτη του πρώτου συνόλου. Κάθε πελάτης του πρώτου συνόλου αντιστοιχεί σε έναν πελάτη του δεύτερου συνόλου.

Ένας πολύ βασικός περιορισμός του προβλήματος είναι ότι δεν μπορεί το drone να επισκεφθεί πρώτα έναν πελάτη του δεύτερου συνόλου, χωρίς πρώτα να έχει επισκεφθεί τον αντίστοιχο πελάτη του πρώτου συνόλου.

Το drone ακολουθεί μια διαδρομή ξεκινώντας πάντα από την αποθήκη 1 και καταλήγει στην ίδια αποθήκη. Η διαδρομή αποτελείται από επιμέρους διαδρομές.

Δεδομένου ότι το drone έχει συγκεκριμένο όριο ενέργειας (1000) που μπορεί να αξιοποιήσει σε κάθε επιμέρους διαδρομή, πρέπει πρώτου αυτό το όριο ξεπεραστεί να έχει ήδη επιστρέψει σε κάποια από τις Μ αποθήκες, έτσι ώστε να αντικαταστήσει την μπαταρία του και να συνεχίσει την αμέσως επόμενη επιμέρους διαδρομή.

Σημειώνεται ότι κάθε αποθήκη έχει συγκεκριμένο αριθμό διαθέσιμων μπαταριών (3) για αντικατάσταση μπαταρίας του drone, ο οποίος είναι δυναμικός, το οποίο σημαίνει ότι κάθε φορά που το drone φτάνει σε μια αποθήκη για να αλλάξει μπαταρία, τότε αφαιρείται μία μπαταρία από την αντίστοιχη αποθήκη. Σε περίπτωση που έχουν εξαντληθεί οι μπαταρίες μιας αποθήκης, τότε το drone θα επιλέξει έναν από τους υπόλοιπους σταθμούς για αλλαγή μπαταρίας. Όταν βρίσκεται στην τελευταία διαδρομή τότε η επιλογή της αποθήκης άφιξης περιορίζεται αναγκαστικά στην αποθήκη 1, όπου και τελειώνει η αρχικοποίηση της λύσης.

Επίσης το drone κατά την διάρκεια της διαδρομής του δεν μπορεί να κουβαλάει φορτίο μεγαλύτερο του 4, εξαιρουμένου του δικού του βάρους, το οποίο έχει οριστεί στην τιμή 1,

Ακόμα το drone δεν έχει την δυνατότητα να μεταφέρει την ίδια στιγμή παραπάνω απο 3 πακέτα.

5.2 Αλγόριθμοι

1. Αλγόριθμος Grasp

Έχουμε μια λίστα πελατών. Αποθηκεύονται οι αποστάσεις του κάθε κόμβου (πελάτη) με τον κόμβο στον οποίο βρίσκεται το drone την τρέχουσα στιγμή. Οι αποστάσεις αυτές ταξινομούνται σε αύξουσα σειρά. Στην συνέχεια αποθηκεύονται οι 3 μικρότερες αποστάσεις με τους αντίστοιχους πελάτες σε μια μικρότερη υποψήφια λίστα. Έπειτα επιλέγεται τυχαία με ίσοποση πιθανότητα ένας πελάτης από αυτήν την λίστα, ως ο επόμενος πελάτης. Θα ελεγχθεί σε πιο από τα 2 σύνολα ανήκει ο πελάτης που έχει επιλεγεί κι εφόσον ικανοποιεί επαρκώς όλους τους περιορισμούς τότε γίνεται αποδεκτός. Ο πελάτης αυτός θα αφαιρεθεί μόνιμα από την λίστα υποψηφίων και δεν μπορεί το drone να τον επισκεφθεί ξανά. Σε διαφορετική περίπτωση που δεν ικανοποιούνται οι περιορισμοί, ο πελάτης αυτός θα αφαιρεθεί προσωρινά από την μεγάλη λίστα πελατών και θα επαναληφθεί η ίδια διαδικασία, μέχρις ότου να βρεθεί πελάτης που να μην παραβιάζει κανέναν περιορισμό. Εφόσον λοιπόν βρεθεί ο επόμενος αποδεκτός πελάτης, επαναφέρονται στην αρχική μεγάλη λίστα οι πελάτες που είχαν αφαιρεθεί προσωρινά.

Σε περίπτωση που η λίστα πελατών έχει λιγότερους απο 3 πελάτες τότε θα επιλεγεί και πάλι τυχαία με ισόποση πιθανότητα ένας από αυτούς και θα πραγματοποιηθεί με την ίδια σειρά η διαδικασία που αναφέρθηκε παραπάνω.

Όταν σε κάποια επανάληψη αναζήτησης του επόμενου πελάτη το drone έχει φτάσει το όριο των πακέτων που μπορεί να μεταφέρει ή του βάρους που μπορεί να κουβαλήσει, τότε το drone οδηγείται στους πελάτες (backhauls) για να παραδώσει στον καθένα το αντίστοιχο πακέτο.

Αυτή την φορά η επιλογή του επόμενου πελάτη backhauls πραγματοποιείται με τον αλγόριθμο του πλησιέστερου γείτονα

2. Αλγόριθμος Πλησιέστερου Γείτονα

Ο πελάτης (backhauls), ο οποίος θα επιλεγεί είναι αυτός με το μικρότερο μήκος διαδρομής από τον κόμβο που βρίσκεται το drone. Ο αλγόριθμος αυτός συνεχίζει μέχρι το drone να έχει παραδώσει όλα τα πακέτα που κουβαλούσε μέχρι εκείνη την στιγμή. Ο ίδιος αλγόριθμος χρησιμοποιείται και στην περίπτωση που το drone πρέπει να οδηγηθεί σε κάποιο σταθμό για να αλλάξει μπαταρία. Επιλέγεται δηλαδή η κοντινότερη αποθήκη από τον τελευταίο πελάτη.

Ο αλγόριθμος αρχικοποίησης της λύσης ολοκληρώνεται όταν το drone έχει επισκεφθεί όλους τους πελάτες.

Υπολογισμός Ενέργειας

Η συνολική ενέργεια υπολογίζεται από μια συνάρτηση. Η συνολική ενέργεια χωρίζεται σε 2 μέρη. Το πρώτο μέρος υπολογίζεται από την απόσταση του drone προς τον επόμενο πελάτη πολλαπλασιασμένη με το βάρος του drone. Το βάρος του drone είναι το άθροισμα του βάρους του ίδιου του drone που ισούται με 1 με το βάρος των πακέτων που μεταφέρει.

Το δεύτερο μέρος υπολογισμού της ενέργειας είναι λόγω της μη αξιοποίησης όλου του ποσού ενέργειας που διαθέτει η κάθε μπαταρία. Προστίθεται λοιπόν στην συνολική ενέργεια κάθε φορά που το drone φτάνει σε κάποια αποθήκη ένα 20% της ενέργειας που έχει μείνει ανεκμετάλειτο.

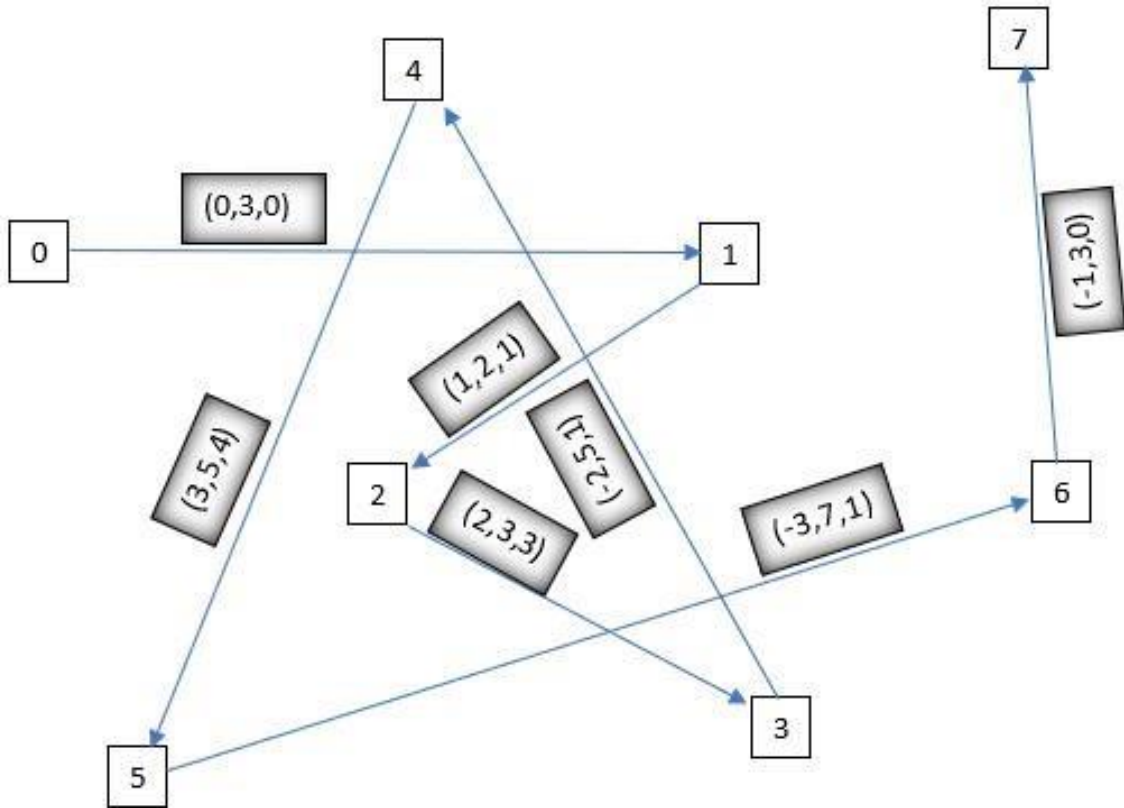
$$E_{ij} = D_{ij} \times (U + f_{ij}) + 0.2 \times (E - \sum_{i=0}^{n-1} \sum_{j=1}^n e_{ij}) \quad (28)$$

, όπου D_{ij} η απόσταση μεταξύ των κόμβων i και j , f_{ij} το συνολικό οφέλιμο φορτίο του drone, ενώ ταξιδεύει από τον κόμβο i στον κόμβο j , U το βάρος του drone, E η μέγιστη ενεργειακή χωρητικότητα του drone και $\sum_{i=0}^{n-1} \sum_{j=1}^n e_{ij}$ η ενέργεια της μπαταρίας που έχει αξιοποιηθεί από το drone. Η μεταβλητή n είναι το σύνολο των τόξων (i,j) της εκάστοτε επιμέρους διαδρομής.

Όσον αφορά τον υπολογισμό της ενέργειας της μεταφοράς του drone από έναν κόμβο i σε κάποιον κόμβο j , συμπεριλαμβάνεται μόνο το πρώτο σκέλος του τύπου (28):

$$e_{ij} = D_{ij} \times (U + f_{ij}) \quad (29)$$

Πιο αναλυτικά για τον (29):



Σχήμα 5.1 Παράδειγμα διαδρομής drone

Σε κάθε κουτάκι υπάρχουν 3 τιμές (i,j,k). Η μεταβλητή i αντιστοιχεί στο βάρος που παραδίδεται ή παραλαμβάνεται εκείνη την στιγμή. Η μεταβλητή j αντικατοπτρίζει την απόσταση κάθε τόξου και τέλος η μεταβλητή μνήμης k, η οποία κρατάει το συνολικό βάρος που μεταφέρει την δεδομένη στιγμή το drone. Ο υπολογισμός της ενέργειας για κάθε διαδρομή έχει ως εξής:

$$e_{01} = d_{01} \times (1 + 0) = 3 \times 1 = 3$$

$$e_{12} = d_{12} \times (1 + 1) = 2 \times 2 = 4$$

$$e_{23} = d_{23} \times (1 + 3) = 3 \times 4 = 12$$

$$e_{34} = d_{34} \times (1 + 1) = 5 \times 2 = 10$$

$$e_{45} = d_{45} \times (1 + 4) = 5 \times 5 = 25$$

$$e_{56} = d_{56} \times (1 + 1) = 7 \times 2 = 14$$

$$e_{67} = d_{67} \times (1 + 0) = 3 \times 1 = 3$$

Το drone ξεκινάει την διαδρομή του από την αποθήκη 1 και καταλήγει στην αποθήκη 7, από την οποία είτε θα τερματίσει είτε θα συνεχίσει με αντικατάσταση μπαταρίας και εξυπηρέτηση των υπόλοιπων πελατών. Το d_{ij} υποδηλώνει την απόσταση του τόξου ενώ οι τιμές της κάθε παρένθεσης είναι το σταθερό βάρος 1 του drone και το βάρος των τεμαχίων που μεταφέρει. Το συνολικό μεταφερόμενο βάρος αυξάνεται αν ο πελάτης παραδίδει ή μειώνεται αν ο πελάτης είναι παραλήπτης. Η συνολική ενέργεια υπολογίζεται με το άθροισμα των επιμέρους ενεργειών.

Vns

Εφόσον έχει πραγματοποιηθεί η αρχικοποίηση της λύσης, τότε εφαρμόζονται κάποιοι μέθοδοι τοπικής αναζήτησης με σκοπό την βελτιστοποίηση της ενέργειας. Κάθε μια από τις μεθόδους που παρουσιάζονται παρακάτω έχουν 1.000 συνολικά ευκαιρίες να δημιουργήσουν μια καινούρια εφικτή λύση και αυτή είναι η ακρίβεια που εφαρμόστηκε στον συγκεκριμένο αλγόριθμο. Σε περίπτωση που δεν βρεθεί εφικτή λύση μετά από 1.000 επαναλήψεις τότε ο κώδικας προχωράει είτε στην επόμενη επιμέρους γειτονιά, εφόσον υπάρχει, αλλιώς στην επόμενη μέθοδο τοπικής αναζήτησης. Εάν βρεθεί εφικτή λύση πριν ολοκληρωθούν οι 1.000 επαναλήψεις, οι επαναλήψεις μηδενίζονται, κι εάν αυτή η λύση είναι καλύτερη τότε αντικαθιστά την παλιά.

2-opt

Σε αυτήν την μέθοδο εφαρμόζουμε τοπική αναζήτηση 2-opt σε κάθε επιμέρους διαδρομή ξεχωριστά. Δεν θέλουμε κάποιος πελάτης backhauls να είναι πρώτα από τον αντίστοιχο πελάτη linehauls. Αυτό το πρόβλημα λύνεται με το να έχουμε μέσα σε κάθε επιμέρους διαδρομή ομάδες κόμβων και αυτές να αλλάζουν θέσεις με τις άλλες ομάδες της ίδιας επιμέρους διαδρομής. Η κατασκευή της κάθε ομάδας τερματίζει όταν κάθε linehaul πελάτης έχει παραδώσει στον αντίστοιχο backhaul πελάτη, είτε όταν μετά από τον οποιοδήποτε πελάτη το drone οδηγείται σε αποθήκη. Οι αποθήκες στην αρχή και στο τέλος της κάθε μιας από αυτές της διαδρομές παραμένουν ακέραιες. Έπειτα κόβουμε την διαδρομή σε 2 σημεία i και j και κάνουμε αντιστροφή (reverse). Το σημείο i παίρνει τιμές από $i=1, \dots, \text{size}()-1$ και το j παίρνει τιμές από $j=1, \dots, \text{size}()-i$. Εφόσον η αλλαγή είναι εφικτή και η ενέργεια που χρειάζεται συνολικά είναι μικρότερη τότε επιλέγεται ως η νέα λύση. Η ίδια διαδικασία πραγματοποιείται για κάθε επιμέρους γειτονιά.

Ένα παράδειγμα κατανόησης του αλγόριθμου:

επιμερους διαδρομή 1: 1 13 12 33 47 46 32 7 6 39 38... 1

Το 1d vector θα εισαχθεί σε ένα 2d vector και δημιουργούνται οι εξής ομάδες:

1 13 12 33 47 46 32 7 6 39 38 1

Έστω ότι $i=1$, τότε $j=size()-1$ και έχουμε την εξής αντιστροφή:

1 39 38 7 6 33 47 46 32 13 12 1

1-0 επανατοποθέτηση (1-0 relocation)

Αυτή η μέθοδος εφαρμόζεται για κάθε επιμέρους διαδρομή ξεχωριστά, αλλά και για όλη την διαδρομή συνολικά.

Στην **1^η περίπτωση** για κάθε γειτονιά ξεχωριστά επιλέγεται τυχαία ένας πελάτης, αφαιρείται από την γειτονιά και επανατοποθετείται τυχαία σε μια άλλη διαφορετική θέση εκτός από την πρώτη και την τελευταία. Εάν είναι εφικτή η λύση, τότε ελέγχεται η ενέργεια. Αν η ενέργεια είναι μικρότερη τότε αντικαθίσταται η παλιά γειτονιά με την νέα, αλλιώς επαναλαμβάνεται η ίδια διαδικασία για τις επόμενες επιμέρους γειτονιές μέχρι να τσεκαριστούν όλες από μια φορά. Σε περίπτωση που η λύση δεν είναι εφικτή επαναλαμβάνεται ο αλγόριθμος για την ίδια γειτονιά.

Ένα παράδειγμα κατανόησης του αλγόριθμου για την **1^η περίπτωση** (single_path):

επιμέρους διαδρομή : 1 13 12 33 47 7 6 46 32 ... 1

Έστω ότι έχει επιλεγθεί τυχαία ο αριθμός 47 της θέσης $i=4$. Αφαιρείται λοιπόν το στοιχείο αυτό από την επιμέρους διαδρομή κι έστω ότι επιλέγεται τυχαία η θέση $i=8$ ως η νέα θέση. Εάν η ανταλλαγή είναι εφικτή και η ενέργεια μικρότερη τότε έχουμε την νέα γειτονιά:

επιμέρους διαδρομή: 1 13 12 33 7 6 46 32 47 ... 1

Στην **2^η περίπτωση**, επιλέγεται έναν τυχαίο στοιχείο από μια τυχαία γραμμή και επανατοποθετείται σε μια διαφορετική τυχαία θέση τυχαίας γραμμής, εκτός από την πρώτη και την τελευταία θέση της συνολικής διαδρομής. Εάν είναι εφικτή η λύση, τότε ελέγχεται η ενέργεια. Αν η ενέργεια είναι μικρότερη τότε αντικαθίσταται η παλιά συνολική διαδρομή με την νέα. Σε περίπτωση που η λύση δεν είναι εφικτή επαναλαμβάνεται ο αλγόριθμος.

Ένα παράδειγμα κατανόησης του αλγόριθμου για την **2^η περίπτωση** (total_path):

επιμερους διαδρομή 1: 1 13 12 33 47 7 6 46 32 ... 1

επιμέρους διαδρομή 2: 1 49 23 22 48 ... 52

επιμέρους διαδρομή 3: 52 35 34 39 38 ... 1

Έστω ότι έχει επιλεγεί τυχαία ο αριθμός 46 της θέσης $i=7$. Αφαιρείται λοιπόν το στοιχείο αυτό από την επιμέρους διαδρομή κι έστω ότι επιλέγεται τυχαία η γραμμή 2 και η θέση $i=4$. Εάν η ανταλλαγή είναι εφικτή και η ενέργεια μικρότερη τότε έχουμε την νέα γειτονιά:

επιμερους διαδρομή 1: 1 13 12 33 47 7 6 32 ... 1

επιμέρους διαδρομή 2: 1 49 23 22 46 48 ... 52

επιμέρους διαδρομή 3: 52 35 34 39 38 ... 1

2-0 επανατοποθέτηση (2-0 relocation)

Αυτή η μέθοδος εφαρμόζεται για κάθε επιμέρους διαδρομή ξεχωριστά, αλλά και για όλη την διαδρομή συνολικά.

Στην **1^η περίπτωση** για κάθε γειτονιά ξεχωριστά επιλέγονται τυχαία 2 γειτονικοί πελάτες i και $i+1$, αφαιρούνται από την γειτονιά και επανατοποθετούνται τυχαία σε μια άλλη διαφορετική θέση εκτός από την πρώτη και την τελευταία. Εάν είναι εφικτή η λύση, τότε ελέγχεται η ενέργεια. Αν η ενέργεια είναι μικρότερη τότε αντικαθίσταται η παλιά γειτονιά με την νέα, αλλιώς επαναλαμβάνεται η ίδια διαδικασία για τις επόμενες επιμέρους γειτονιές μέχρι να τσεκαριστούν όλες από μια φορά. Σε περίπτωση που η λύση δεν είναι εφικτή επαναλαμβάνεται ο αλγόριθμος για την ίδια γειτονιά.

Ένα παράδειγμα κατανόησης του αλγόριθμου για την **1^η περίπτωση** (single_path):

επιμέρους διαδρομή : 1 13 12 33 47 7 6 46 32 ... 1

Έστω ότι έχει επιλεγεί τυχαία το ζεύγος αριθμών 47,7 της θέσης $i=4,5$. Αφαιρούνται λοιπόν τα στοιχεία αυτά από την επιμέρους διαδρομή κι έστω ότι επιλέγεται τυχαία η θέση $i=1$ ως η νέα θέση. Εάν η ανταλλαγή είναι εφικτή και η ενέργεια μικρότερη τότε έχουμε την νέα γειτονιά:

επιμέρους διαδρομή: 1 47 7 13 12 33 6 46 32 ... 1

Στην **2^η περίπτωση**, επιλέγονται 2 τυχαίοι γειτονικοί πελάτες i και $i+1$ από μια τυχαία γραμμή και επανατοποθετούνται σε μια διαφορετική τυχαία θέση τυχαίας γραμμής, εκτός από την πρώτη και την τελευταία θέση της συνολικής διαδρομής. Εάν είναι εφικτή η λύση, τότε ελέγχεται η ενέργεια. Αν η ενέργεια είναι μικρότερη τότε αντικαθίσταται η παλιά συνολική διαδρομή με την νέα. Σε περίπτωση που η λύση δεν είναι εφικτή επαναλαμβάνεται ο αλγόριθμος.

Ένα παράδειγμα κατανόησης του αλγόριθμου για την **2^η περίπτωση** (total_path):

επιμερους διαδρομή 1: 1 13 12 33 47 33 6 46 32 ... 1

επιμέρους διαδρομή 2: 1 49 23 22 48 ... 52

επιμέρους διαδρομή 3: 52 35 34 39 38 ... 1

Έστω ότι έχει επιλεγεί τυχαία το ζεύγος αριθμών: 39,38 των θέσεων $i=3,4$. Αφαιρούνται λοιπόν αυτά τα 2 στοιχεία από την επιμέρους διαδρομή 3 κι έστω ότι επιλέγεται η γραμμή 2 και η θέση $i=1$. Εάν η ανταλλαγή είναι εφικτή και η ενέργεια μικρότερη τότε έχουμε την νέα γειτονιά:

επιμερους διαδρομή 1: 1 13 12 33 47 33 6 46 32 ... 1

επιμέρους διαδρομή 2: 1 39 38 49 23 22 48 ... 52

επιμέρους διαδρομή 3: 52 35 34 ... 1

1-1 ανταλλαγή (1-1 reexchange)

Αυτή η μέθοδος εφαρμόζεται για κάθε επιμέρους διαδρομή ξεχωριστά, αλλά και για όλη την διαδρομή συνολικά.

Στην **1^η περίπτωση** για κάθε γειτονιά ξεχωριστά επιλέγονται τυχαία 2 πελάτες και ανταλλάσσουν θέσεις. Εάν είναι εφικτή η λύση, τότε ελέγχεται η ενέργεια. Αν η ενέργεια είναι μικρότερη τότε αντικαθίσταται η παλιά γειτονιά με την νέα, αλλιώς επαναλαμβάνεται η ίδια διαδικασία για τις επόμενες επιμέρους γειτονιές μέχρι να τσεκαριστούν όλες από μια φορά. Σε περίπτωση που η λύση δεν είναι εφικτή επαναλαμβάνεται ο αλγόριθμος για την ίδια γειτονιά.

Ένα παράδειγμα κατανόησης του αλγόριθμου για την **1^η περίπτωση** (single_path):

επιμέρους διαδρομή : 1 13 12 **33** 47 **7** 6 46 32 ... 1

Έστω ότι έχουν επιλεγεί τυχαία οι αριθμοί: 33, της θέσης $i=3$ και 7 της θέσης $i=5$. Εάν η ανταλλαγή είναι εφικτή και η ενέργεια μικρότερη τότε έχουμε την νέα γειτονιά:

επιμέρους διαδρομή : 1 13 12 **7** 47 **33** 6 46 32 ... 1

Στην **2^η περίπτωση**, επιλέγονται 2 τυχαίοι πελάτες από μια τυχαία γραμμή ο καθένας και ανταλλάσσουν θέσεις. Εάν είναι εφικτή η λύση, τότε ελέγχεται η ενέργεια. Αν η ενέργεια είναι μικρότερη τότε αντικαθίσταται η παλιά συνολική διαδρομή με την νέα. Σε περίπτωση που η λύση δεν είναι εφικτή επαναλαμβάνεται ο αλγόριθμος.

Ένα παράδειγμα κατανόησης του αλγόριθμου για την **2^η περίπτωση** (total_path):

επιμέρους διαδρομή 1: 1 13 12 33 47 33 6 46 **32** ... 1

επιμέρους διαδρομή 2: 1 49 23 22 48 ... 52

επιμέρους διαδρομή 3: 52 **35** 34 39 38 ... 1

Έστω ότι έχουν επιλεγεί οι τυχαίοι αριθμοί: 32, της θέσης $i=8$ της πρώτης επιμέρους διαδρομής και 35, της θέσης $i=2$ της τρίτης επιμέρους διαδρομής. Εάν η ανταλλαγή είναι εφικτή και η ενέργεια μικρότερη τότε έχουμε την νέα γειτονιά:

επιμέρους διαδρομή 1: 1 13 12 33 47 33 6 46 **35** ... 1

επιμέρους διαδρομή 2: 1 49 23 22 48 ... 52

επιμέρους διαδρομή 3: 52 **32** 34 39 38 ... 1

2-2 ανταλλαγή (2-2 reexchange)

Αυτή η μέθοδος εφαρμόζεται για κάθε επιμέρους διαδρομή ξεχωριστά, αλλά και για όλη την διαδρομή συνολικά.

Στην **1^η περίπτωση** για κάθε γειτονιά ξεχωριστά επιλέγονται τυχαία 2 διαφορετικά ζεύγη γειτονικών πελατών i και $i+1$ και ανταλλάσσουν θέσεις. Εάν είναι εφικτή η λύση, τότε ελέγχεται η ενέργεια. Αν η ενέργεια είναι μικρότερη τότε αντικαθίσταται η παλιά γειτονιά με την νέα, αλλιώς επαναλαμβάνεται η ίδια διαδικασία για τις επόμενες επιμέρους γειτονιές μέχρι να τσεκαριστούν όλες από μια φορά. Σε περίπτωση που η λύση δεν είναι εφικτή επαναλαμβάνεται ο αλγόριθμος για την ίδια γειτονιά.

Ένα παράδειγμα κατανόησης του αλγόριθμου για την **1^η περίπτωση** (single_path):

επιμέρους διαδρομή : 1 13 12 **33 47** 7 6 **46 32** ... 1

Έστω ότι έχουν επιλεγεί τυχαία τα ζεύγη αριθμών: 33,47, των θέσεων $i=3,4$ και 46,32 της θέσης $i=7,8$. Εάν η ανταλλαγή είναι εφικτή και η ενέργεια μικρότερη τότε έχουμε την νέα γειτονιά:

επιμέρους διαδρομή : 1 13 12 **46 32** 7 6 **33 47**... 1

Στην **2^η περίπτωση**, επιλέγονται τυχαία 2 διαφορετικά ζεύγη γειτονικών πελατών i και $i+1$ από μια τυχαία γραμμή το κάθε ζεύγος και ανταλλάσσουν θέσεις. Εάν είναι εφικτή η λύση, τότε ελέγχεται η ενέργεια. Αν η ενέργεια είναι μικρότερη τότε αντικαθίσταται η παλιά συνολική διαδρομή με την νέα. Σε περίπτωση που η λύση δεν είναι εφικτή επαναλαμβάνεται ο αλγόριθμος.

Ένα παράδειγμα κατανόησης του αλγόριθμου για την **2^η περίπτωση** (total_path):

επιμέρους διαδρομή 1: 1 **13 12** 33 47 33 6 46 32 ... 1

επιμέρους διαδρομή 2: 1 49 23 22 48 ... 52

επιμέρους διαδρομή 3: 52 **35 34** 39 38 ... 1

Έστω ότι έχουν επιλεγεί τυχαία τα ζεύγη αριθμών 13,12, των θέσεων $i=1,2$ της πρώτης επιμέρους διαδρομής και 35,34 των θέσεων $i=1,2$ της τρίτης επιμέρους διαδρομής. Εάν η ανταλλαγή είναι εφικτή και η ενέργεια μικρότερη τότε έχουμε την νέα γειτονιά:

επιμέρους διαδρομή 1: 1 **35 34** 33 47 33 6 46 32 ... 1

επιμέρους διαδρομή 2: 1 49 23 22 48 ... 52

επιμέρους διαδρομή 3: 52 **13 12** 39 38 ... 1

Changing-Storage

Σε αυτήν την μέθοδο κάθε αποθήκη αλλαγής μπαταρίας αντικαθίσταται με την σειρά από τις υπόλοιπες αποθήκες που έχουν διαθέσιμη μπαταρία. Η αλλαγή εφόσον είναι εφικτή και βελτιώνει την ενέργεια αντικαθιστά την παλιά συνολική διαδρομή κι ο αλγόριθμος συνεχίζεται μέχρι να

πραγματοποιηθούν όλες οι επαναλήψεις. Αν η αλλαγή δεν είναι εφικτή ή δεν φέρει βελτίωση τότε συνεχίζεται ο αλγόριθμος μέχρι να πραγματοποιηθούν όλες οι επαναλήψεις.

Ένα παράδειγμα κατανόησης του αλγόριθμου:

1 13 12 33 47 46 32 53 49 23 22 48 52 21 51 17 50 16 20 1

Θα αντικατασταθεί η αποθήκη 53 με την 52 κι εφόσον η αλλαγή είναι εφικτή και καλύτερη πραγματοποιείται η αντικατάσταση.

1 13 12 33 47 46 32 52 49 23 22 48 52 21 51 17 50 16 20 1

Θα αντικατασταθεί η αποθήκη 52 με την 1 κι εφόσον η αλλαγή είναι εφικτή και καλύτερη πραγματοποιείται η αντικατάσταση.

1 13 12 33 47 46 32 1 49 23 22 48 52 21 51 17 50 16 20 1

Επαναλαμβάνεται ο αλγόριθμος για κάθε αποθήκη ξεχωριστά εκτός της αρχικής και τελικής της συνολικής διαδρομής.

Η σειρά που υλοποιούνται οι παραπάνω συναρτήσεις είναι

reexchange 1-1 (single_path), local search 2-opt, relocation 1-0 (single_path),

reexchange 2-2 (single_path), local-search-2opt, relocation 2-0 (single_path),

reexchange 1-1 (total_path), local search 2-opt, relocation 1-0 (total_path),

reexchange 2-2 (total_path), local-search-2opt, relocation 2-0 (total_path),

και τελευταία συνάρτηση η αλλαγή μπαταρίας changing-storage

Κάθε μια από τις παραπάνω τοπικές αναζητήσεις πραγματοποιείται τόσες φορές μέχρις ότου σε κάποια επανάληψη η λύση που θα βγάλουν να μην είναι καλύτερη. Συνολικά η βελτιστοποίηση της λύσης θα σταματήσει όταν καμία από τις παραπάνω τοπικές αναζητήσεις σε μια ολόκληρη επανάληψη δεν δώσει γειτονιά με μικρότερη ενέργεια.

5.3 Ψευδοκώδικες

Αλγόριθμος Άπληστης Τυχαιοποιημένης Αναζήτησης

Αρχικοποίηση μεταβλητών και πινάκων

ΓΙΑ όσο είναι το μέγεθος των υποψηφίων πελατών

 Αποθήκευσε τις αποστάσεις των πελατών από το drone

ΤΕΛΟΣ_ΓΙΑ

 Σόρταρε τις παραπάνω αποστάσεις

ΕΑΝ το μέγεθος το υποψήφιων πελατών ≥ 3 **ΤΟΤΕ**

Αποθήκευσε τους 3 κοντινότερους πελάτες με τις αποστάσεις τους

ΑΛΛΙΩΣ

Αποθήκευσε όλους τους πελάτες με τις αποστάσεις τους

ΤΕΛΟΣ_ΕΑΝ

Επέλεξε τυχαία έναν από τους αποθηκευμένους πελάτες

Επέστρεψε τον πελάτη

1-0 επανατοποθέτηση για κάθε επιμέρους διαδρομή

ΕΑΝ το μέγεθος της επιμέρους διαδρομής > 4

Επέλεξε τυχαίο πελάτη_1 από την επιμέρους_διαδρομή

Αφαίρεσε τον πελάτη_1 από την επιμέρους_διαδρομή

Εισχώρησε τον πελάτη_1 σε μια τυχαία θέση i της επιμέρους_διαδρομής ($i=1:\text{size}()-2$)

ΑΛΛΙΩΣ

Πήγαινε στην επόμενη επιμέρους_διαδρομή

ΤΕΛΟΣ_ΕΑΝ

ΚΑΝΕ_ΜΕΧΡΙ η λύση να είναι εφικτή

Επανάλαβε **1-0 επανατοποθέτηση για κάθε επιμέρους διαδρομή**

ΤΕΛΟΣ_ΚΑΝΕ

ΕΑΝ η λύση είναι καλύτερη

Αντικατέστησε την επιμέρους_διαδρομή

ΑΛΛΙΩΣ

Επέστρεψε την διαδρομή πριν τις αλλαγές

ΤΕΛΟΣ_ΕΑΝ

1-0 επανατοποθέτηση για όλη την διαδρομή

Επέλεξε μια επιμέρους_διαδρομή_1 τυχαία

ΟΣΟ το μέγεθος της επιμέρους_διαδρομής_1 < 4

Επέλεξε μια καινούρια επιμέρους_διαδρομή_1 τυχαία

ΤΕΛΟΣ_ΟΣΟ

Επέλεξε έναν τυχαίο πελάτη_1 από την επιμέρους_διαδρομή_1

Αφαίρεσε τον πελάτη_1

Επέλεξε μια επιμέρους_διαδρομή_2 τυχαία

Επέλεξε έναν τυχαίο πελάτη_2 από την επιμέρους_διαδρομή_2

ΟΣΟ (διαδρομή_1==διαδρομή_2 && πελάτης_1==πελάτης_2)

Επέλεξε καινούριο τυχαίο πελάτη_2

ΤΕΛΟΣ_ΟΣΟ

Εισχώρησε τον πελάτη_1 στην θέση του πελάτη_2 της επιμέρους_διαδρομής_2

Επέστρεψε όλη την διαδρομή

KANE_MEXPI η λύση να είναι εφικτή

Επανέλαβε **1-0 επανατοποθέτηση για όλη την διαδρομή**

ΤΕΛΟΣ_KANE

ΕΑΝ η λύση είναι καλύτερη

Αντικατέστησε την διαδρομή

ΑΛΛΙΩΣ

Επέστρεψε την διαδρομή πριν τις αλλαγές

ΤΕΛΟΣ_EAN

1-1 ανταλλαγή για κάθε επιμέρους διαδρομή

ΕΑΝ το μέγεθος της επιμέρους_διαδρομής >= 4

Επέλεξε τυχαίο πελάτη_1 της επιμέρους-διαδρομής

Επέλεξε τυχαίο πελάτη_2 της επιμέρους_διαδρομής

ΟΣΟ (πελάτης_1==πελάτη_2)

Επέλεξε τυχαίο πελάτη_2

ΤΕΛΟΣ_ΟΣΟ

Αντάλλαξε θέση τον πελάτη_1 με τον πελάτη_2

ΑΛΛΙΩΣ

Πήγαινε στην επόμενη επιμέρους_διαδρομή

ΤΕΛΟΣ_EAN

Επέστρεψε την επιμέρους_διαδρομή

KANE_MEXPI η λύση να είναι εφικτή

Επανέλαβε **1-1 ανταλλαγή για κάθε επιμέρους διαδρομή**

ΤΕΛΟΣ_KANE

EAN η λύση είναι καλύτερη

Αντικατέστησε την επιμέρους_διαδρομή

ΑΛΛΙΩΣ

Επέστρεψε την διαδρομή πριν τις αλλαγές

ΤΕΛΟΣ_EAN

1-1 ανταλλαγή για όλη την διαδρομή

Επέλεξε μια επιμέρους_διαδρομή_1 τυχαία

ΟΣΟ επιμέρους_διαδρομη_1 έχει μέγεθος <4

Επέλεξε καινούρια επιμέρους_διαδρομή_1 τυχαία

ΤΕΛΟΣ_ΟΣΟ

Επέλεξε πελάτη_1 στην επιμέρους_διαδρομή_1 τυχαία

Επέλεξε μια επιμέρους_διαδρομή_2 τυχαία

Επέλεξε πελάτη_2 στην επιμέρους_διαδρομή_2 τυχαία

ΟΣΟ (επιμέρους_διαδρομή_1==επιμέρους_διαδρομή_2 && πελάτης_1==πελάτης_2)

Επέλεξε καινούριο πελάτη_2 τυχαία

ΤΕΛΟΣ_ΟΣΟ

Αντάλλαξε θέση τον πελάτη_1 με τον πελάτη_2

Επέστρεψε την συνολική διαδρομή

KANE_MEXPI η λύση να είναι εφικτή

Επανάλαβε **1-1 ανταλλαγή για όλη την διαδρομή**

ΤΕΛΟΣ_KANE

ΕΑΝ η λύση είναι καλύτερη

Αντικατέστησε την διαδρομή

ΑΛΛΙΩΣ

Επέστρεψε την διαδρομή πριν τις αλλαγές

ΤΕΛΟΣ_EAN

2-0 επανατοποθέτηση για κάθε επιμέρους διαδρομή

ΕΑΝ το μέγεθος της επιμέρους_διαδρομής ≥ 4

Επέλεξε 2 γειτονικούς πελάτες, τον πελάτη_1 και τον πελάτη_2

Αφαίρεσε τον πελάτη_1 και τον πελάτη_2 από την επιμέρους_διαδρομή

Επέλεξε μια τυχαία θέση i , όπου $i=2:\text{size}()-1$

Εισχώρησε τον πελάτη_1 στην θέση i και τον πελάτη_2 στην θέση $i+1$

ΑΛΛΙΩΣ

Πήγαινε στην επόμενη επιμέρους_διαδρομή

ΤΕΛΟΣ_EAN

Επέστρεψε την επιμέρους_διαδρομή

KANE_MEXPI η λύση να είναι εφικτή

Επανάλαβε **2-0 επανατοποθέτηση για κάθε επιμέρους διαδρομή**

ΤΕΛΟΣ_KANE

ΕΑΝ η λύση είναι καλύτερη

Αντικατέστησε την επιμέρους_διαδρομή

ΑΛΛΙΩΣ

Επέστρεψε την διαδρομή πριν τις αλλαγές

ΤΕΛΟΣ_EAN

2-0 επανατοποθέτηση για όλη την διαδρομή

Επέλεξε μια επιμέρους_διαδρομή_1 τυχαία

ΟΣΟ επιμέρους_διαδρομή_1 έχει μέγεθος <5

Επέλεξε καινούρια επιμέρους_διαδρομή_1 τυχαία

ΤΕΛΟΣ_ΟΣΟ

Επέλεξε 2 γειτονικούς πελάτες, τον πελάτη_1 και τον πελάτη_2

Αφαίρεσε τον πελάτη_1 και τον πελάτη_2

Επέλεξε μια επιμέρους_διαδρομή_2 τυχαία

Επέλεξε μια τυχαία θέση i, όπου i=2:size()-3

ΟΣΟ (επιμέρους_διαδρομή_1==επιμέρους_διαδρομή_2 && θέση_πελάτη_1==θέση i)

Επέλεξε καινούρια θέση i τυχαία

ΤΕΛΟΣ_ΟΣΟ

Εισχώρησε τον πελάτη_1 στην θέση i της επιμέρους_διαδρομής_2

Εισχώρησε τον πελάτη_2 στην θέση i+1 της επιμέρους_διαδρομής_2

Επέστρεψε την συνολική διαδρομή

KANE_MEXPI η λύση να είναι εφικτή

Επανάλαβε **2-0 επανατοποθέτηση για όλη την διαδρομή**

ΤΕΛΟΣ_KANE

EAN η λύση είναι καλύτερη

Αντικατέστησε την διαδρομή

ΑΛΛΙΩΣ

Επέστρεψε την διαδρομή πριν τις αλλαγές

ΤΕΛΟΣ_EAN

2-2 ανταλλαγή για κάθε επιμέρους διαδρομή

ΕΑΝ το μέγεθος της επιμέρους_διαδρομής > 6

Επέλεξε 2 τυχαίες θέσεις i και j , όπου $i, j = 2:\text{size}()-3$

ΟΣΟ $(i == j \parallel (i-1) == j \parallel (i+1) == j)$

Επέλεξε καινούρια θέση j τυχαία

ΤΕΛΟΣ_ΟΣΟ

Αντάλλαξε τους πελάτες των θέσεων i και $i+1$ με τους πελάτες των θέσεων j και $j+1$

ΑΛΛΙΩΣ

Πήγαινε στην επόμενη επιμέρους_διαδρομή

ΤΕΛΟΣ_ΕΑΝ

Επέστρεψε την επιμέρους_διαδρομή

ΚΑΝΕ_MEXPI η λύση να είναι εφικτή

Επανάλαβε **2-2 ανταλλαγή για κάθε επιμέρους διαδρομή**

ΤΕΛΟΣ_ΚΑΝΕ

ΕΑΝ η λύση είναι καλύτερη

Αντικατέστησε την επιμέρους_διαδρομή

ΑΛΛΙΩΣ

Επέστρεψε την επιμέρους_διαδρομή πριν τις αλλαγές

ΤΕΛΟΣ_ΕΑΝ

2-2 Ανταλλαγή για όλη την διαδρομή

Επέλεξε μια επιμέρους_διαδρομή_1 τυχαία

ΟΣΟ επιμέρους_διαδρομή_1 έχει μέγεθος < 6

Επέλεξε καινούρια επιμέρους_διαδρομή_1 = k_1 τυχαία

ΤΕΛΟΣ_ΟΣΟ

Επέλεξε μια θέση i της επιμέρους_διαδρομής_1 τυχαία, όπου $i = 2:\text{size}()-3$

Επέλεξε μια επιμέρους_διαδρομή_2 = k_2 τυχαία

Επέλεξε μια θέση j της επιμέρους_διαδρομής_2 τυχαία, όπου $j = 2:\text{size}()-3$

ΟΣΟ ($k_1 == k_2 \ \&\& \ (i == j \ \parallel \ (i+1) == j \ \parallel \ (i-1) == j)$)

Επέλεξε καινούρια θέση j τυχαία

ΤΕΛΟΣ_ΟΣΟ

Αντάλλαξε τους πελάτες των θέσεων i και $i+1$ με τους πελάτες των θέσεων j και $j+1$

Επέστρεψε την διαδρομή

KANE_MEXPI η λύση να είναι εφικτή

Επανάλαβε **2-2 ανταλλαγή για όλη την διαδρομή**

ΤΕΛΟΣ_KANE

ΕΑΝ η λύση είναι καλύτερη

Αντικατέστησε την διαδρομή

ΑΛΛΙΩΣ

Επέστρεψε την διαδρομή πριν τις αλλαγές

ΤΕΛΟΣ_EAN

Local Search 2-opt για κάθε επιμέρους διαδρομή

Κάνε την 1d επιμέρους διαδρομή σε 2d

ΓΙΑ κάθε γραμμή $i=1:\text{size}()-1$ της 2d

ΓΙΑ κάθε γραμμή $j=1:\text{size}()-i$ της 2d

Γίνεται αντιστροφή του 2d vector

ΕΑΝ η λύση είναι εφικτή

ΕΑΝ η λύση είναι καλύτερη

Αντικατάσταση της επιμέρους διαδρομής

ΑΛΛΙΩΣ

Επέστρεψε την προηγούμενη διαδρομή

ΤΕΛΟΣ_EAN

ΑΛΛΙΩΣ

Επέστρεψε την προηγούμενη διαδρομή

ΤΕΛΟΣ_EAN

ΤΕΛΟΣ_ΓΙΑ

ΤΕΛΟΣ_ΓΙΑ

Changing storage_of_battery_replacement

ΓΙΑ κάθε στοιχείο της συνολικής διαδρομής

EAN είναι αποθήκη αντικατάστασης μπαταρίας

Γίνεται έλεγχος διαθεσιμότητας μπαταριών

EAN υπάρχει διαθεσιμότητα

Αλλαγή αποθήκης

EAN είναι εφικτή η αλλαγή

EAN η λύση είναι καλύτερη

Αντικατάσταση διαδρομής

ΑΛΛΙΩΣ

Επέστρεψε την προηγούμενη διαδρομή

ΤΕΛΟΣ_EAN

ΑΛΛΙΩΣ

Επέστρεψε την προηγούμενη διαδρομή

ΤΕΛΟΣ_EAN

ΤΕΛΟΣ_EAN

ΤΕΛΟΣ_EAN

ΤΕΛΟΣ_ΓΙΑ

ΚΕΦΑΛΑΙΟ 6

ΑΠΟΤΕΛΕΣΜΑΤΑ

Στην παρούσα διπλωματική εργασία επιτεύχθηκε βελτιστοποίηση ενέργειας σε 16 σετ δεδομένων. Κάθε ένα από αυτά τα σετ περιλαμβάνει έναν αριθμό κόμβων, τον αριθμό των σταθμών που μπορεί να οδηγηθεί το drone για την αντικατάσταση της μπαταρίας, εφόσον χρειαστεί, την μέγιστη ενέργεια που μπορεί το drone να εκμεταλευντεί από την εκάστοτε μπαταρία, το μέγιστο αριθμό του βάρους και των πακέτων που μπορεί να μεταφέρει. Κάθε 4 σετ ο αριθμός των συνολικών κόμβων είναι διαφορετικός. Πιο συγκεκριμένα αλλάζει και το μέγεθος των πελατών, αλλά και των αποθηκών αντικατάστασης μπαταρίας. Επίσης οι συντεταγμένες των κόμβων είναι διαφορετικές κάθε φορά που το μέγεθος των συνολικών κόμβων αλλάζει, με τους περιορισμούς για κάθε σενάριο να είναι οι ίδιοι.

Σκοπός του προβλήματος είναι η καλύτερη δυνατή βελτιστοποίηση της καταναλισκόμενης ενέργειας του drone εξυπηρετώντας όλους τους πελάτες, λαμβάνοντας υπόψη τα δεδομένα και τους περιορισμούς του προβλήματος.

Για κάθε ένα από τα σετ δεδομένων εκτελέστηκε ο κώδικας 5 φορές και παρακάτω παρουσιάζονται τα αποτελέσματα για κάθε περίπτωση, κατά μέσο όρο.

6.1 Αποτελέσματα του Εκάστοτε Σετ Δεδομένων

Για τα παραδείγματα των 53 κόμβων (50 πελάτες + 1 αποθήκη + 2 σταθμοί) έχουμε τους εξής πίνακες:

παράδειγμα 1 των 53 κόμβων (50 πελάτες + 1 αποθήκη + 2 σταθμοί)							
	Αρχική Ενέργεια	Βελτιωμέ νη Ενέργεια	Ενέργει α που μειώθη κε	Ποσοστό Βελτιωμέν ης Ενέργειας	Μπαταρίες που Αντικαταστήθη καν	Επαναλήψεις Βελτιστοποίη σης	Χρόνος εκτέλεσ ης (sec)
Μέσος Όρος 5 επαναλήψε ων	3114.7 4	2738.46	376.28	12.07%	2.60	2.80	1.26

Πίνακας 6.1 Αποτελέσματα βελτιστοποίησης του πρώτου σετ δεδομένων με 53 κόμβους

Σε αυτό το παράδειγμα η βελτίωση ανέρχεται σε ποσοστό 12.07% με τον χρόνο εκτέλεσης στα 1.26 δευτερόλεπτα. Κατά μέσο όρο για την ολοκλήρωση των διαδρομών χρειάστηκαν 2.6 μπαταρίες για αντικατάσταση. Η βελτιστοποίηση πραγματοποιήθηκε σε 2.8 επαναλήψεις κατά μέσο όρο.

παράδειγμα 2 των 53 κόμβων (50 πελάτες + 1 αποθήκη + 2 σταθμοί)							
	Αρχική Ενέργεια	Βελτιωμένη Ενέργεια	Ενέργεια που Βελτιώθηκε	Ποσοστό Βελτιωμένης Ενέργειας	Μπαταρίες που Αντικαταστήθηκαν	Επαναλήψεις Βελτιστοποίησης	Χρόνος εκτέλεσης (sec)
Μέσος Όρος 5 επαναλήψεων	3007.88	2520.11	487.77	15.91%	2.40	4.00	1.25

Πίνακας 6.2 Αποτελέσματα βελτιστοποίησης του δεύτερου σετ δεδομένων με 53 κόμβους

Σε αυτό το παράδειγμα η βελτίωση ανέρχεται σε ποσοστό 15.91% με τον χρόνο εκτέλεσης στα 1.25 δευτερόλεπτα. Κατά μέσο όρο για την ολοκλήρωση των διαδρομών χρειάστηκαν 2.4 μπαταρίες για αντικατάσταση. Η βελτιστοποίηση πραγματοποιήθηκε σε 4 επαναλήψεις κατά μέσο όρο.

παράδειγμα 3 των 53 κόμβων (50 πελάτες + 1 αποθήκη + 2 σταθμοί)							
	Αρχική Ενέργεια	Βελτιωμένη Ενέργεια	Ενέργεια που Βελτιώθηκε	Ποσοστό Βελτιωμένης Ενέργειας	Μπαταρίες που Αντικαταστήθηκαν	Επαναλήψεις Βελτιστοποίησης	Χρόνος εκτέλεσης (sec)
Μέσος Όρος 5 επαναλήψεων	3494.71	3056.81	437.90	12.50%	3.00	3.00	1.35

Πίνακας 6.3 Αποτελέσματα βελτιστοποίησης του τρίτου σετ δεδομένων με 53 κόμβους

Σε αυτό το παράδειγμα η βελτίωση ανέρχεται σε ποσοστό 12.5% με τον χρόνο εκτέλεσης στα 1.35 δευτερόλεπτα. Κατά μέσο όρο για την ολοκλήρωση των διαδρομών χρειάστηκαν 3 μπαταρίες για αντικατάσταση. Η βελτιστοποίηση πραγματοποιήθηκε σε 3 επαναλήψεις κατά μέσο όρο.

παράδειγμα 4 των 53 κόμβων (50 πελάτες + 1 αποθήκη + 2 σταθμοί)							
	Αρχική Ενέργεια	Βελτιωμένη Ενέργεια	Ενέργεια που Βελτιώθηκε	Ποσοστό Βελτιωμένης Ενέργειας	Μπαταρίες που Αντικαταστήθηκαν	Επαναλήψεις Βελτιστοποίησης	Χρόνος εκτέλεσης (sec)
Μέσος Όρος 5 επαναλήψεων	3404.90	3046.46	358.44	10.52%	3.00	3.20	1.31

Πίνακας 6.4 Αποτελέσματα βελτιστοποίησης του τέταρτου σετ δεδομένων με 53 κόμβους

Σε αυτό το παράδειγμα η βελτίωση ανέρχεται σε ποσοστό 10.52% με τον χρόνο εκτέλεσης στα 1.31 δευτερόλεπτα. Κατά μέσο όρο για την ολοκλήρωση των διαδρομών χρειάστηκαν 3 μπαταρίες για αντικατάσταση. Η βελτιστοποίηση πραγματοποιήθηκε σε 3.2 επαναλήψεις κατά μέσο όρο.

Για τα παραδείγματα των 104 κόμβων (100 πελάτες + 1 αποθήκη + 3 σταθμοί) έχουμε τους εξής πίνακες:

παράδειγμα 1 των 104 κόμβων (100 πελάτες + 1 αποθήκη + 3 σταθμοί)							
	Αρχική Ενέργεια	Βελτιωμένη Ενέργεια	Ενέργεια που Βελτιώθηκε	Ποσοστό Βελτιωμένης Ενέργειας	Μπαταρίες που Αντικαταστήθηκαν	Επαναλήψεις Βελτιστοποίησης	Χρόνος εκτέλεσης (sec)
Μέσος Όρος 5 επαναλήψεων	5702.84	4708.30	994.54	17.27%	5.20	4.20	8.96

Πίνακας 6.5 Αποτελέσματα βελτιστοποίησης του πρώτου σετ δεδομένων με 104 κόμβους

Σε αυτό το παράδειγμα η βελτίωση ανέρχεται σε ποσοστό 17.27% με τον χρόνο εκτέλεσης στα 8.96 δευτερόλεπτα. Κατά μέσο όρο για την ολοκλήρωση των διαδρομών χρειάστηκαν 5.2 μπαταρίες για αντικατάσταση. Η βελτιστοποίηση πραγματοποιήθηκε σε 4.2 επαναλήψεις κατά μέσο όρο.

παράδειγμα 2 των 104 κόμβων (100 πελάτες + 1 αποθήκη + 3 σταθμοί)							
	Αρχική Ενέργεια	Βελτιωμένη Ενέργεια	Ενέργεια που Βελτιώθηκε	Ποσοστό Βελτιωμένης Ενέργειας	Μπαταρίες που Αντικαταστήθηκαν	Επαναλήψεις Βελτιστοποίησης	Χρόνος εκτέλεσης (sec)
Μέσος Όρος 5 επαναλήψεων	5257.83	4491.20	766.63	14.53%	5.00	2.60	5.58

Πίνακας 6.6 Αποτελέσματα βελτιστοποίησης του δεύτερου σετ δεδομένων με 104 κόμβους

Σε αυτό το παράδειγμα η βελτίωση ανέρχεται σε ποσοστό 14.53% με τον χρόνο εκτέλεσης στα 5.58 δευτερόλεπτα. Κατά μέσο όρο για την ολοκλήρωση των διαδρομών χρειάστηκαν 5 μπαταρίες για αντικατάσταση. Η βελτιστοποίηση πραγματοποιήθηκε σε 2.6 επαναλήψεις κατά μέσο όρο.

παράδειγμα 3 των 104 κόμβων (100 πελάτες + 1 αποθήκη + 3 σταθμοί)							
	Αρχική Ενέργεια	Βελτιωμένη Ενέργεια	Ενέργεια που Βελτιώθηκε	Ποσοστό Βελτιωμένης Ενέργειας	Μπαταρίες που Αντικαταστήθηκαν	Επαναλήψεις Βελτιστοποίησης	Χρόνος εκτέλεσης (sec)
Μέσος Όρος 5 επαναλήψεων	6405.41	5642.10	763.31	11.87%	5.8	5.20	9.77

Πίνακας 6.7 Αποτελέσματα βελτιστοποίησης του τρίτου σετ δεδομένων με 104 κόμβους

Σε αυτό το παράδειγμα η βελτίωση ανέρχεται σε ποσοστό 11.87% με τον χρόνο εκτέλεσης στα 9.77 δευτερόλεπτα. Κατά μέσο όρο για την ολοκλήρωση των διαδρομών χρειάστηκαν 5.8 μπαταρίες για αντικατάσταση. Η βελτιστοποίηση πραγματοποιήθηκε σε 5.2 επαναλήψεις κατά μέσο όρο.

παράδειγμα 4 των 104 κόμβων (100 πελάτες + 1 αποθήκη + 3 σταθμοί)							
	Αρχική Ενέργεια	Βελτιωμένη Ενέργεια	Ενέργεια που Βελτιώθηκε	Ποσοστό Βελτιωμένης Ενέργειας	Μπαταρίες που Αντικαταστήθηκαν	Επαναλήψεις Βελτιστοποίησης	Χρόνος εκτέλεσης (sec)
Μέσος Όρος 5 επαναλήψεων	5977.20	5231.47	745.73	12.38%	5.80	4.40	7.94

Πίνακας 6.8 Αποτελέσματα βελτιστοποίησης του τέταρτου σετ δεδομένων με 104 κόμβους

Σε αυτό το παράδειγμα η βελτίωση ανέρχεται σε ποσοστό 12.38% με τον χρόνο εκτέλεσης στα 7.94 δευτερόλεπτα. Κατά μέσο όρο για την ολοκλήρωση των διαδρομών χρειάστηκαν 5.8 μπαταρίες για αντικατάσταση. Η βελτιστοποίηση πραγματοποιήθηκε σε 4.4 επαναλήψεις κατά μέσο όρο.

Για τα παραδείγματα των 155 κόμβων (150 πελάτες + 1 αποθήκη + 4 σταθμοί) έχουμε τους εξής πίνακες:

παράδειγμα 1 των 155 κόμβων (150 πελάτες + 1 αποθήκη + 4 σταθμοί)							
	Αρχική Ενέργεια	Βελτιωμένη Ενέργεια	Ενέργεια που Βελτιώθηκε	Ποσοστό Βελτιωμένης Ενέργειας	Μπαταρίες που Αντικαταστήθηκαν	Επαναλήψεις Βελτιστοποίησης	Χρόνος εκτέλεσης (sec)
Μέσος Όρος 5 επαναλήψεων	7488.41	6449.10	1039.31	13.88%	7	5.6	23

Πίνακας 6.9 Αποτελέσματα βελτιστοποίησης του πρώτου σετ δεδομένων με 155 κόμβους

Σε αυτό το παράδειγμα η βελτίωση ανέρχεται σε ποσοστό 13.88% με τον χρόνο εκτέλεσης στα 23 δευτερόλεπτα. Κατά μέσο όρο για την ολοκλήρωση των διαδρομών χρειάστηκαν 7 μπαταρίες για αντικατάσταση. Η βελτιστοποίηση πραγματοποιήθηκε σε 5.6 επαναλήψεις κατά μέσο όρο.

παράδειγμα 2 των 155 κόμβων (150 πελάτες + 1 αποθήκη + 4 σταθμοί)							
	Αρχική Ενέργεια	Βελτιωμένη Ενέργεια	Ενέργεια που Βελτιώθηκε	Ποσοστό Βελτιωμένης Ενέργειας	Μπαταρίες που Αντικαταστήθηκαν	Επαναλήψεις Βελτιστοποίησης	Χρόνος εκτέλεσης (sec)
Μέσος Όρος 5 επαναλήψεων	8193.7	6836.28	1357.43	16.45%	7.8	4.2	19.54

Πίνακας 6.10 Αποτελέσματα βελτιστοποίησης του δεύτερου σετ δεδομένων με 155 κόμβους

Σε αυτό το παράδειγμα η βελτίωση ανέρχεται σε ποσοστό 16.45% με τον χρόνο εκτέλεσης στα 19.54 δευτερόλεπτα. Κατά μέσο όρο για την ολοκλήρωση των διαδρομών χρειάστηκαν 7.8 μπαταρίες για αντικατάσταση. Η βελτιστοποίηση πραγματοποιήθηκε σε 4.2 επαναλήψεις κατά μέσο όρο.

παράδειγμα 3 των 155 κόμβων (150 πελάτες + 1 αποθήκη + 4 σταθμοί)							
	Αρχική Ενέργεια	Βελτιωμένη Ενέργεια	Ενέργεια που Βελτιώθηκε	Ποσοστό Βελτιωμένης Ενέργειας	Μπαταρίες που Αντικαταστήθηκαν	Επαναλήψεις Βελτιστοποίησης	Χρόνος εκτέλεσης (sec)
Μέσος Όρος 5 επαναλήψεων	9641.84	8448.2	1193.64	12.23%	9.2	4.6	24.32

Πίνακας 6.11 Αποτελέσματα βελτιστοποίησης του τρίτου σετ δεδομένων με 155 κόμβους

Σε αυτό το παράδειγμα η βελτίωση ανέρχεται σε ποσοστό 12.23% με τον χρόνο εκτέλεσης στα 24.32 δευτερόλεπτα. Κατά μέσο όρο για την ολοκλήρωση των διαδρομών χρειάστηκαν 9.2 μπαταρίες για αντικατάσταση. Η βελτιστοποίηση πραγματοποιήθηκε σε 4.6 επαναλήψεις κατά μέσο όρο.

παράδειγμα 4 των 155 κόμβων (150 πελάτες + 1 αποθήκη + 4 σταθμοί)							
	Αρχική Ενέργεια	Βελτιωμέ νη Ενέργεια	Ενέργεια που Βελτιώθη κε	Ποσοστό Βελτιωμέ νης Ενέργειας	Μπαταρίες που Αντικαταστήθη καν	Επαναλήψεις Βελτιστοποίη σης	Χρόνος εκτέλεσ ης (sec)
Μέσος Όρος 5 επαναλήψε ων	10273. 33	9047.35	1225.98	11.9%	10	3.2	17.89

Πίνακας 6.12 Αποτελέσματα βελτιστοποίησης του τέταρου σεν δεδομένων με 155 κόμβους

Σε αυτό το παράδειγμα η βελτίωση ανέρχεται σε ποσοστό 11.9% με τον χρόνο εκτέλεσης στα 17.89 δευτερόλεπτα. Κατά μέσο όρο για την ολοκλήρωση των διαδρομών χρειάστηκαν 10 μπαταρίες για αντικατάσταση. Η βελτιστοποίηση πραγματοποιήθηκε σε 3.2 επαναλήψεις κατά μέσο όρο.

Για τα παραδείγματα των 206 κόμβων (200 πελάτες + 1 αποθήκη + 5 σταθμοί) έχουμε τους εξής πίνακες:

παράδειγμα 1 των 206 κόμβων (200 πελάτες + 1 αποθήκη + 5 σταθμοί)							
	Αρχική Ενέργεια	Βελτιωμέ νη Ενέργεια	Ενέργεια που Βελτιώθη κε	Ποσοστό Βελτιωμέ νης Ενέργειας	Μπαταρίες που Αντικαταστήθη καν	Επαναλήψεις Βελτιστοποίη σης	Χρόνος εκτέλεσ ης (sec)
Μέσος Όρος 5 επαναλήψε ων	10156. 84	8712.94	1443.9	14.25%	9.8	7.2	61.64

Πίνακας 6.13 Αποτελέσματα βελτιστοποίησης του πρώτου σεν δεδομένων με 206 κόμβους

Σε αυτό το παράδειγμα η βελτίωση ανέρχεται σε ποσοστό 14.25% με τον χρόνο εκτέλεσης στα 61.64 δευτερόλεπτα. Κατά μέσο όρο για την ολοκλήρωση των διαδρομών χρειάστηκαν 9.8 μπαταρίες για αντικατάσταση. Η βελτιστοποίηση πραγματοποιήθηκε σε 7.2 επαναλήψεις κατά μέσο όρο.

παράδειγμα 2 των 206 κόμβων (200 πελάτες + 1 αποθήκη + 5 σταθμοί)							
	Αρχική Ενέργεια	Βελτιωμένη Ενέργεια	Ενέργεια που Βελτιώθηκε	Ποσοστό Βελτιωμένης Ενέργειας	Μπαταρίες που Αντικαταστήθηκαν	Επαναλήψεις Βελτιστοποίησης	Χρόνος εκτέλεσης (sec)
Μέσος Όρος 5 επαναλήψεων	10455.5	8962.28	1493.22	14.23%	10.2	6.2	55.32

Πίνακας 6.14 Αποτελέσματα βελτιστοποίησης του δεύτερου σετ δεδομένων με 206 κόμβους

Σε αυτό το παράδειγμα η βελτίωση ανέρχεται σε ποσοστό 14.23% με τον χρόνο εκτέλεσης στα 55.32 δευτερόλεπτα. Κατά μέσο όρο για την ολοκλήρωση των διαδρομών χρειάστηκαν 10.2 μπαταρίες για αντικατάσταση. Η βελτιστοποίηση πραγματοποιήθηκε σε 6.2 επαναλήψεις κατά μέσο όρο.

παράδειγμα 3 των 206 κόμβων (200 πελάτες + 1 αποθήκη + 5 σταθμοί)							
	Αρχική Ενέργεια	Βελτιωμένη Ενέργεια	Ενέργεια που Βελτιώθηκε	Ποσοστό Βελτιωμένης Ενέργειας	Μπαταρίες που Αντικαταστήθηκαν	Επαναλήψεις Βελτιστοποίησης	Χρόνος εκτέλεσης (sec)
Μέσος Όρος 5 επαναλήψεων	11863.94	10682.62	1181.32	9.89%	11.6	3.2	37.22

Πίνακας 6.15 Αποτελέσματα βελτιστοποίησης του τρίτου σετ δεδομένων με 206 κόμβους

Σε αυτό το παράδειγμα η βελτίωση ανέρχεται σε ποσοστό 9.89% με τον χρόνο εκτέλεσης στα 37.22 δευτερόλεπτα. Κατά μέσο όρο για την ολοκλήρωση των διαδρομών χρειάστηκαν 11.6 μπαταρίες για αντικατάσταση. Η βελτιστοποίηση πραγματοποιήθηκε σε 3.2 επαναλήψεις κατά μέσο όρο.

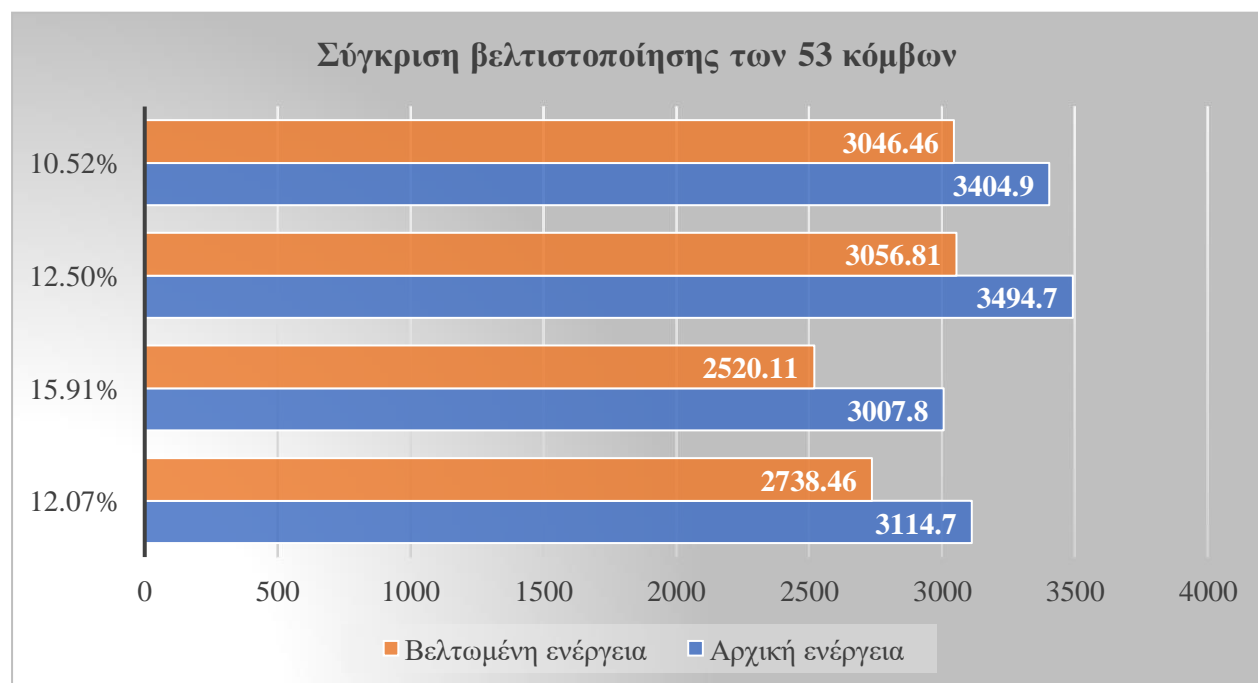
παράδειγμα 4 των 206 κόμβων (200 πελάτες + 1 αποθήκη + 5 σταθμοί)							
	Αρχική Ενέργεια	Βελτιωμέ νη Ενέργεια	Ενέργεια που Βελτιώθη κε	Ποσοστό Βελτιωμέ νης Ενέργειας	Μπαταρίες που Αντικαταστήθη καν	Επαναλήψεις Βελτιστοποίη σης	Χρόνος εκτέλεσ ης (sec)
Μέσος Όρος 5 επαναλήψε ων	12709. 68	11422.48	1287.2	10.13%	12.6	6.8	88.85

Πίνακας 6.16 Αποτελέσματα βελτιστοποίησης του τέταρου σετ δεδομένων με 206 κόμβους

Σε αυτό το παράδειγμα η βελτίωση ανέρχεται σε ποσοστό 10.13% με τον χρόνο εκτέλεσης στα 88.85 δευτερόλεπτα. Κατά μέσο όρο για την ολοκλήρωση των διαδρομών χρειάστηκαν 12.6 μπαταρίες για αντικατάσταση. Η βελτιστοποίηση πραγματοποιήθηκε σε 6.8 επαναλήψεις κατά μέσο όρο.

6.2 Σύγκριση Σετ Δεδομένων

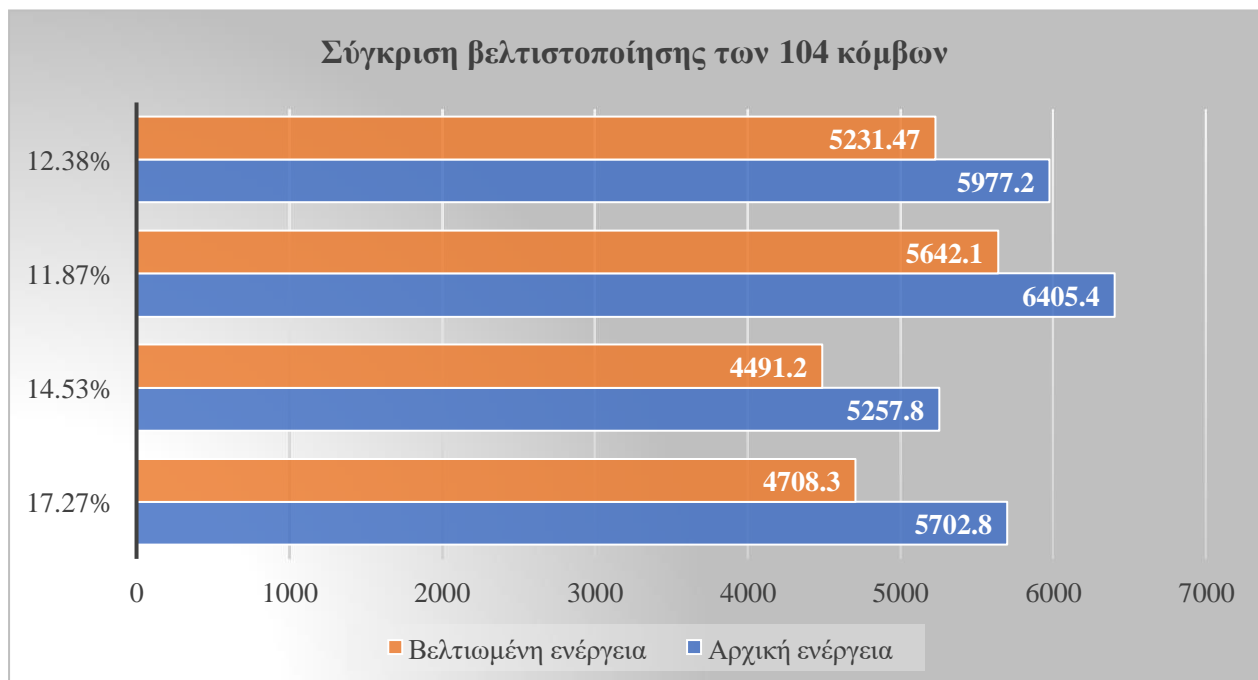
Στην προηγούμενη υποενότητα παρουσιάζονται οι μέσοι όροι 5 επαναλήψεων για κάθε ένα από τα σετ δεδομένων. Αρχικά θα συγκριθούν τα σετ δεδομένων που έχουν τον ίδιο αριθμό κόμβων και στην συνέχεια τα σετ δεδομένων με διαφορετικό αριθμό κόμβων. Τα παρακάτω γραφήματα δημιουργήθηκαν με την βοήθεια του excel, από το αρχείο energy_of_drones.xlsx.



Γράφημα 1. Σύγκριση βελτιστοποίησης των σετ δεδομένων με 53 κόμβους

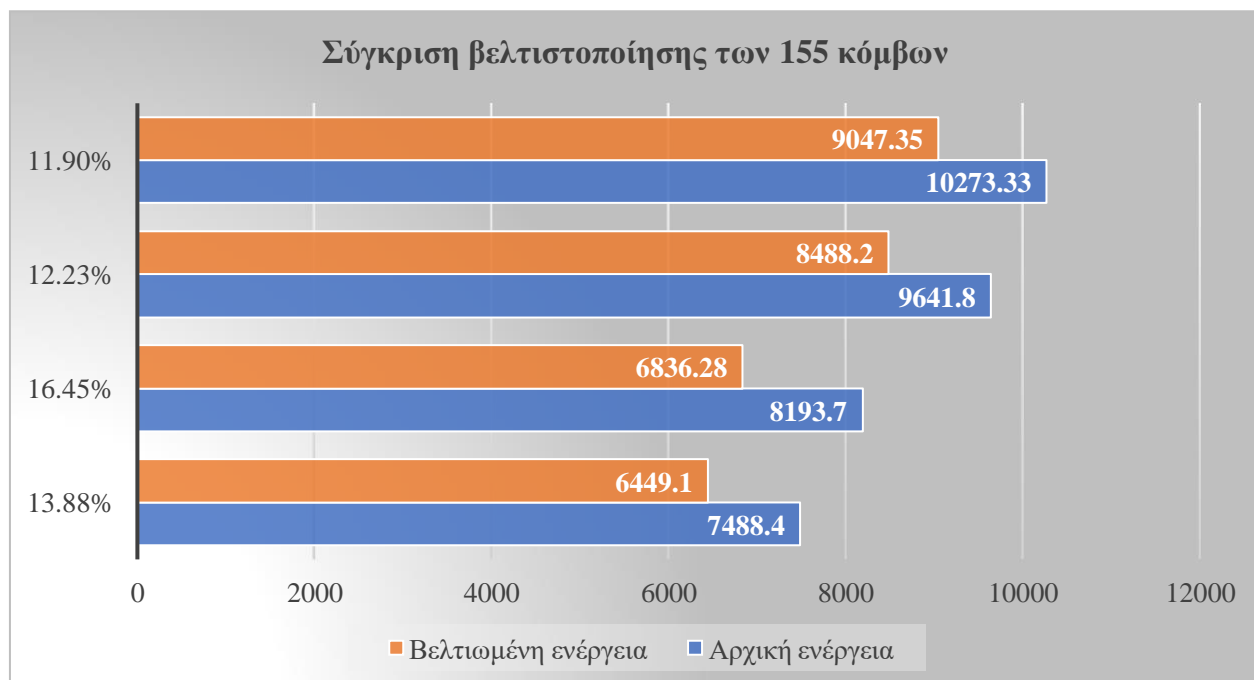
Σε αυτού του είδους τα γραφήματα ο άξονας x αναφέρεται στην ενέργεια. Το χρώμα μπλε αντιστοιχεί στην αρχική ενέργεια, ενώ το πορτοκαλί στην τελική. Ο άξονας y αντιστοιχεί στα ποσοστά βελτιστοποίησης κάθε μιας από τις 4 περιπτώσεις.

Στο παραπάνω γράφημα πραγματοποιείται σύγκριση των σετ δεδομένων με τους 53 κόμβους. Παρατηρείται ότι η βελτιστοποίηση που πραγματοποιήθηκε στην μικρότερη αρχική ενέργεια είναι καλύτερη σε ποσοστό από τις άλλες και οδήγησε και στην μικρότερη τελική ενέργεια. Αξίζει να σημειωθεί ότι η βελτιστοποίηση 12.5% έναντι του 10.52% δεν αποδίδει σε καλύτερο τελικό αποτέλεσμα λόγω της διαφοράς της αρχικής ενέργειας που έχει προκύψει από τους μέσους όρους.



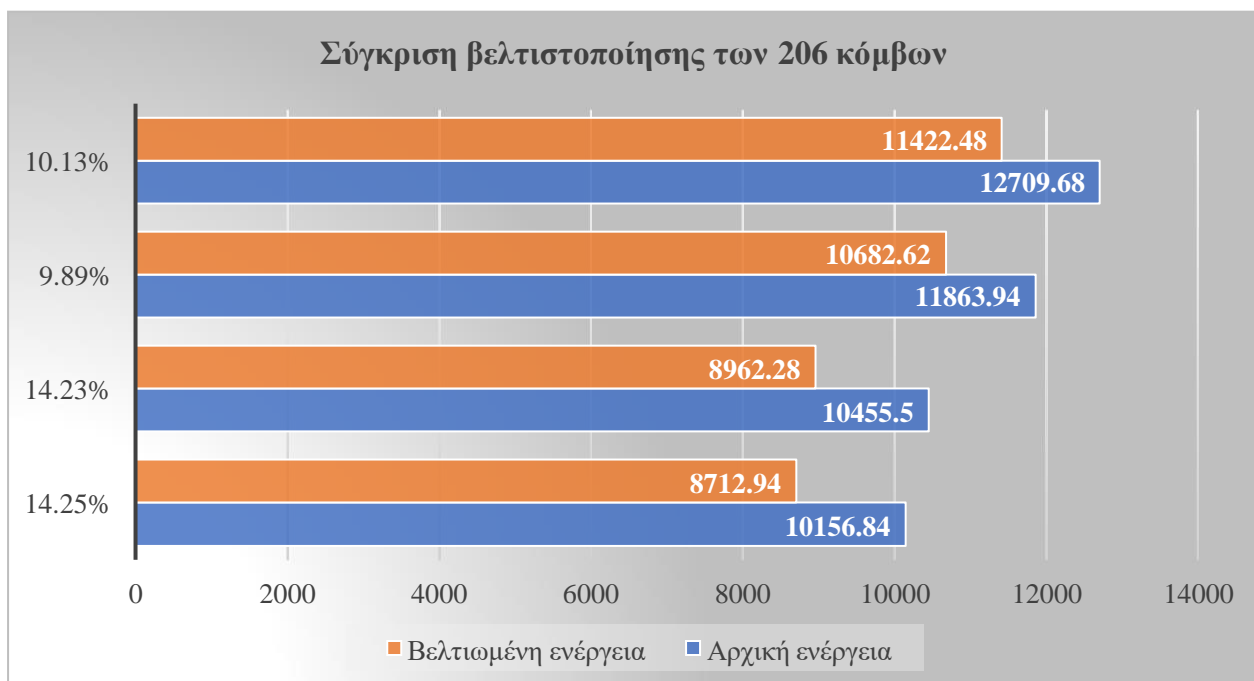
Γράφημα 2. Σύγκριση βελτιστοποίησης των σετ δεδομένων με 104 κόμβους

Στο παραπάνω γράφημα για άλλη μια φορά παρατηρείται ότι η μικρότερη αρχική ενέργεια οδηγεί σε καλύτερη τελική βελτιστοποίηση. Επίσης το ποσοστό βελτίωσης της αρχικής λύσης είναι μεγαλύτερο για τις 2 μικρότερες αρχικές ενέργειες, σε αντίθεση με τις 2 μεγαλύτερες που το ποσοστό αυτό αγγίζει το 11.87% και 12.38%.



Γράφημα 3. Σύγκριση βελτιστοποίησης των σετ δεδομένων με 155 κόμβους

Όπως και στα προηγούμενα γραφήματα έτσι και σε αυτό η ενέργεια με την μικρότερη αρχική ενέργεια έχει την καλύτερη τελική ενέργεια μετά την βελτιστοποίηση. Τα ποσοστά βελτιστοποίησης για τις 2 μεγαλύτερες αρχικές ενέργειες είναι τα 2 πιο μικρά. Εν αντιθέση οι καλύτερες αρχικές λύσεις έχουν μεγαλύτερα ποσοστά βελτιστοποίησης.

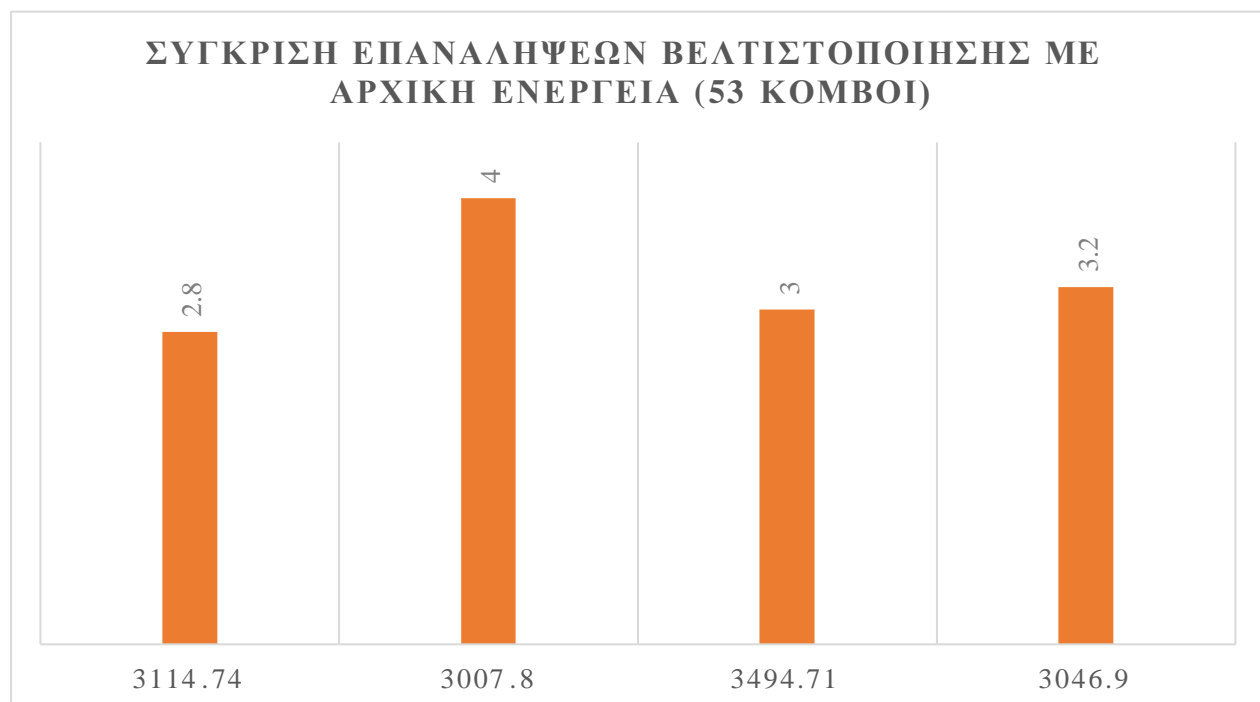


Γράφημα 4. Σύγκριση βελτιστοποίησης των σετ δεδομένων με 206 κόμβους

Στο γράφημα 4, η μικρότερη αρχική ενέργεια έχει το μεγαλύτερο ποσοστό βελτιστοποίησης οδηγώντας και στην βέλτιστη λύση συγκρίνοντας με τις υπόλοιπες 3 περιπτώσεις μέσω των όρων. Οι 2 καλύτερες αρχικές λύσεις παρουσιάζουν μεγαλύτερο ποσοστό βελτίωσης, σε αντίθεση με τις 2 μεγαλύτερες αρχικές ενέργειες, οι οποίες εμφανίζουν μικρότερο ποσοστό.

Βάσει των πινάκων που αναγράφονται οι μέσοι όροι, σημειώνεται ότι ο χρόνος εκτέλεσης του αλγόριθμου εξαρτάται της πολυπλοκότητας του προβλήματος. Πιο συγκεκριμένα για τους 53 κόμβους ο χρόνος εκτέλεσης των 4 σετ λαμβάνοντας τον μέσο όρο αυτών ανέρχεται στα 1.29 δευτερόλεπτα ενώ για τους 104, 155 και 206 κόμβους ανέρχεται στα 8.06, 21.18 και στα 60.75 δευτερόλεπτα αντίστοιχα. Επίσης όσο μικρότερος ο αριθμός των κόμβων, για τα σετ δεδομένων με τον ίδιο αριθμό κόμβων, τόσο μικρότερη η απόκλιση τερματισμού της βελτιστοποίησης. Αντίθετα, όσο μεγαλύτερος αυτός ο αριθμός τόσο πιο πολύ μπορεί να διαφοροποιείται ο χρόνος εκτέλεσης του αλγόριθμου.

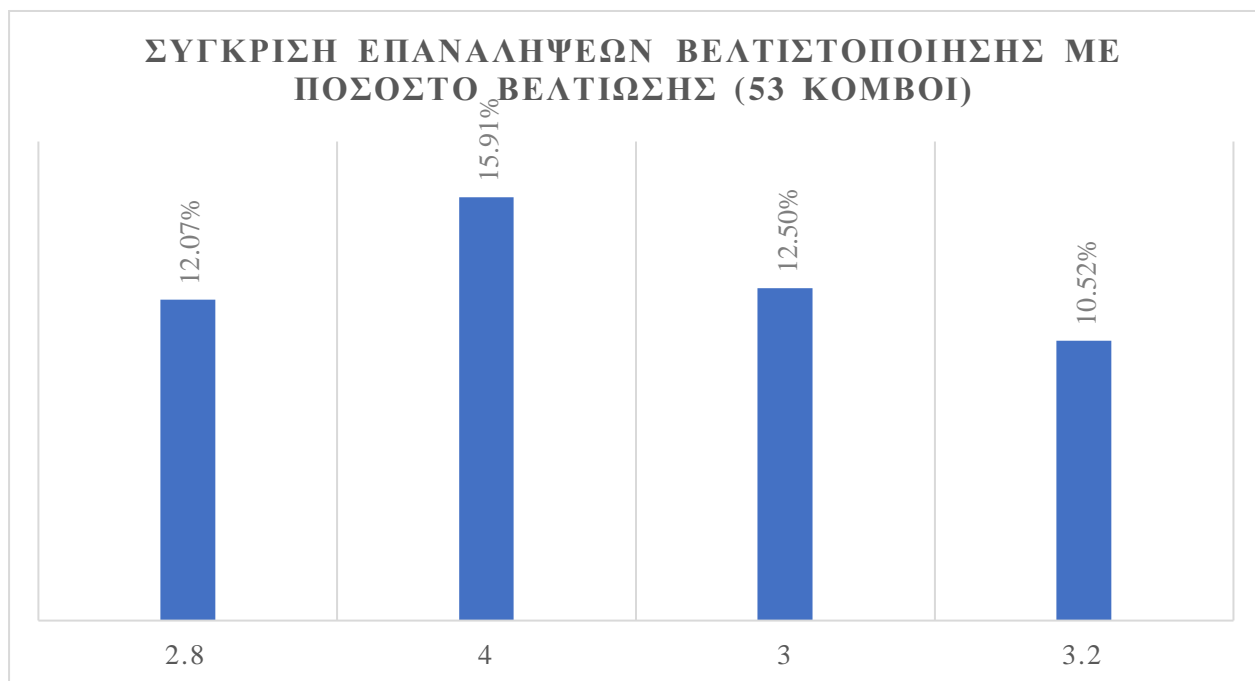
Όσον αφορά τις μπαταρίες, ο αριθμός των μπαταριών που απαιτείται για αντικατάσταση είναι ανάλογος του αριθμού των συνολικών πελατών που πρέπει να ικανοποιηθούν. Όσο μεγαλύτερος λοιπόν αυτός ο αριθμός τόσο περισσότερες μπαταρίες πρέπει να είναι διαθέσιμες καθολικά στο σύνολο των σταθμών. Όπως και στον χρόνο εκτέλεσης, έτσι και σε αυτήν την περίπτωση όσο μεγαλύτερος ο αριθμός κόμβων τόσο μεγαλύτερη μπορεί να είναι η απόκλιση των μπαταριών που απαιτούνται για 2 σετ δεδομένων με τον ίδιο αριθμό κόμβων.



Γράφημα 5. Σύγκριση επαναλήψεων βελτιστοποίησης με την αρχική ενέργεια για τους 53 κόμβους

Σε αυτού του είδους τα γραφήματα ο άξονας x αναφέρεται στην αρχική ενέργεια, ενώ στην κορυφή της μπάρας παρουσιάζεται ο μέσος όρος του αριθμού των επαναλήψεων για κάθε μέσο των αρχικών ενεργειών.

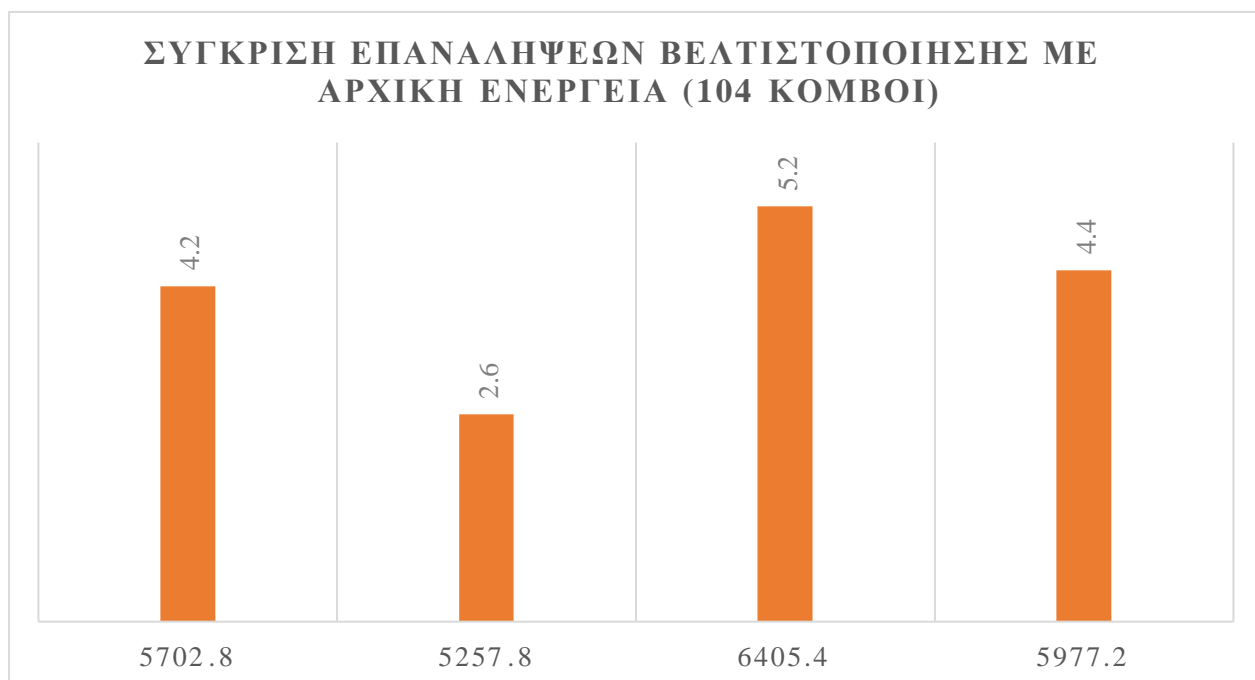
Στο παραπάνω γράφημα αξίζει να σημειωθεί ότι ο μεγαλύτερος αριθμός επαναλήψεων πραγματοποιείται στην μικρότερη αρχική ενέργεια.



Γράφημα 6. Σύγκριση επαναλήψεων βελτιστοποίησης με το ποσοστό βελτίωσης για τους 53 κόμβους

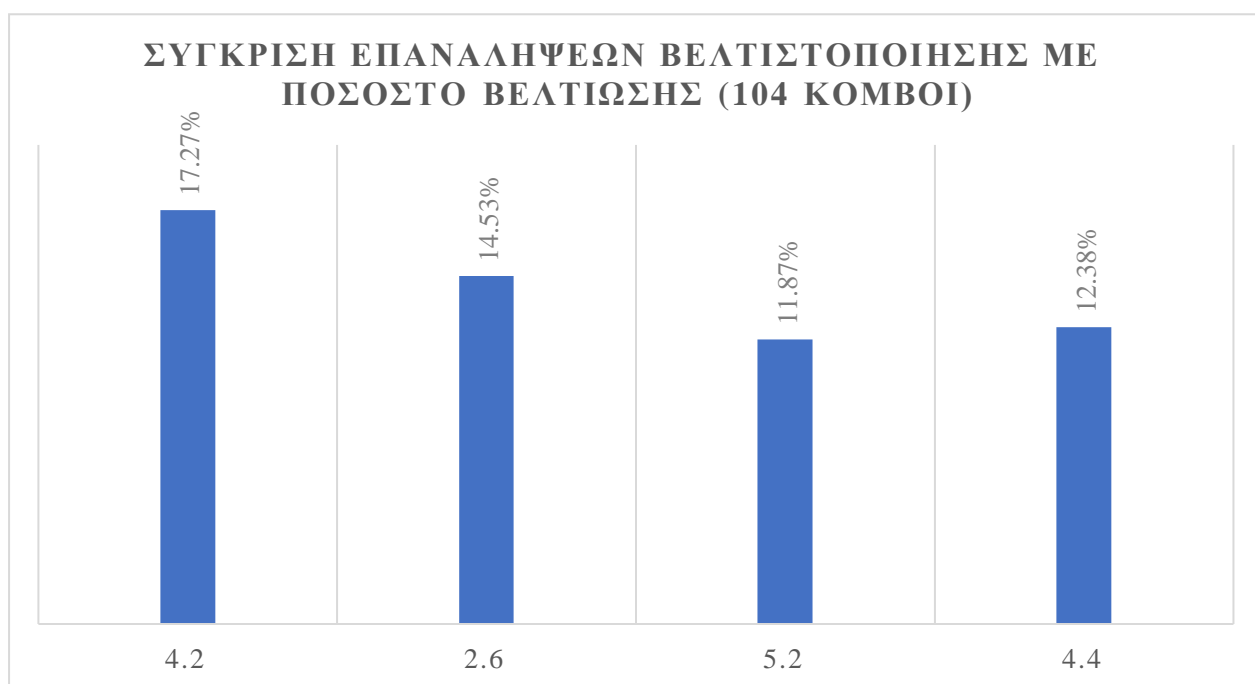
Σε αυτού του είδους τα γραφήματα ο άξονας x αναφέρεται στις επαναλήψεις, ενώ στην κορυφή της εκάστοτε μπάρας παρουσιάζονται τα ποσοστά της βελτιστοποίησης.

Στο γράφημα 6. ο μεγαλύτερος αριθμός επαναλήψεων έχει και το καλύτερο ποσοστό βελτίωσης.



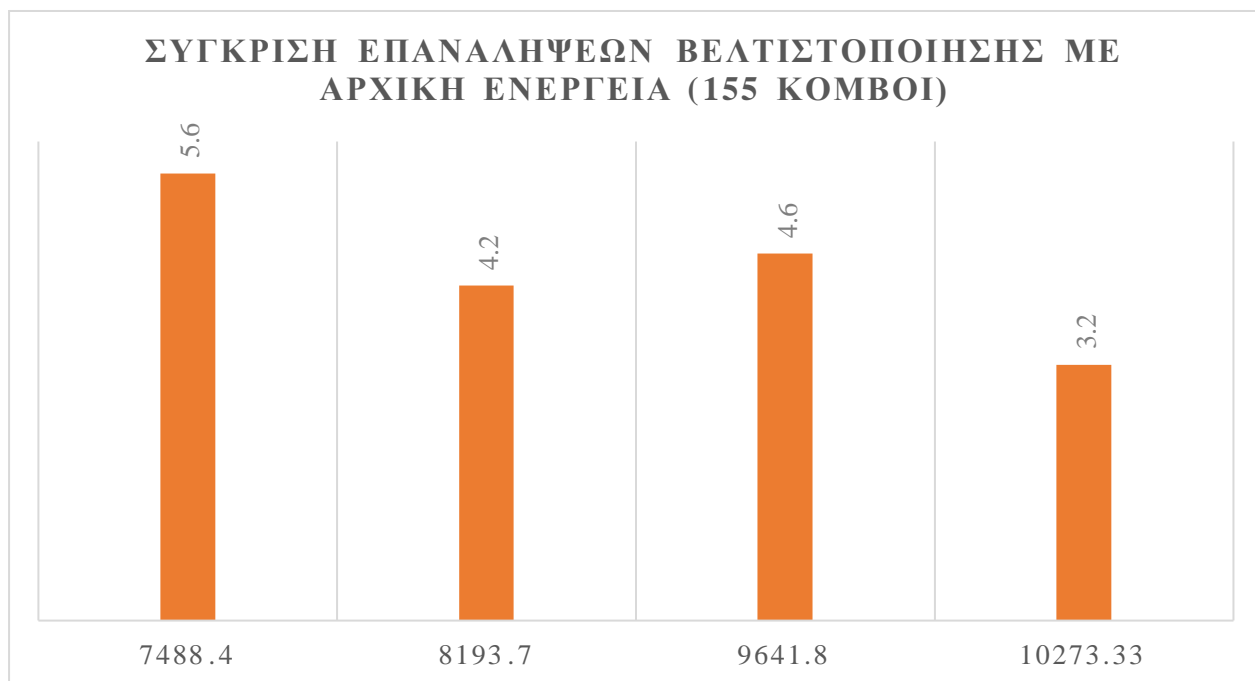
Γράφημα 7. Σύγκριση επαναλήψεων βελτιστοποίησης με την αρχική ενέργεια για τους 104 κόμβους

Στο παραπάνω γράφημα ο μεγαλύτερος αριθμός επαναλήψεων πραγματοποιείται στην μεγαλύτερη αρχική ενέργεια. Αντίθετα η καλύτερη αρχική λύση σημειώνει μικρότερο αριθμό επαναλήψεων.



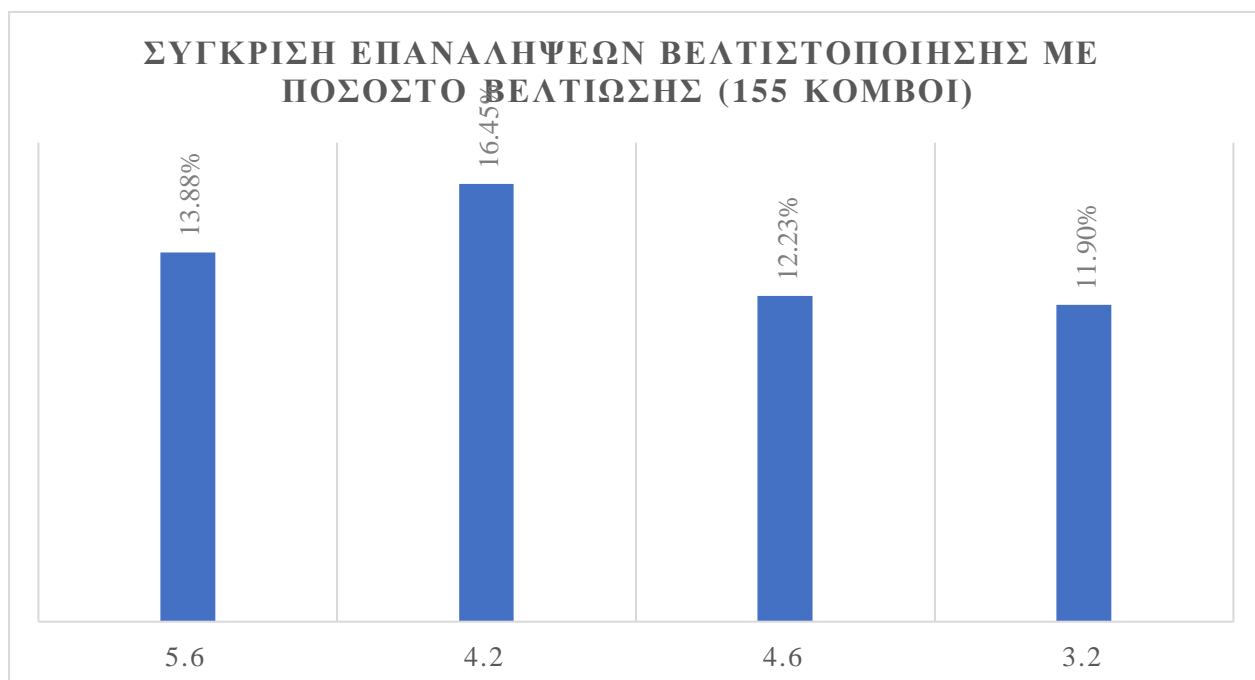
Γράφημα 8. Σύγκριση επαναλήψεων βελτιστοποίησης με το ποσοστό βελτίωσης για τους 104 κόμβους

Στο γράφημα 8. υπογραμμίζεται ότι οι περισσότερες επαναλήψεις αποδίδουν στην μικρότερη βελτιστοποίηση.



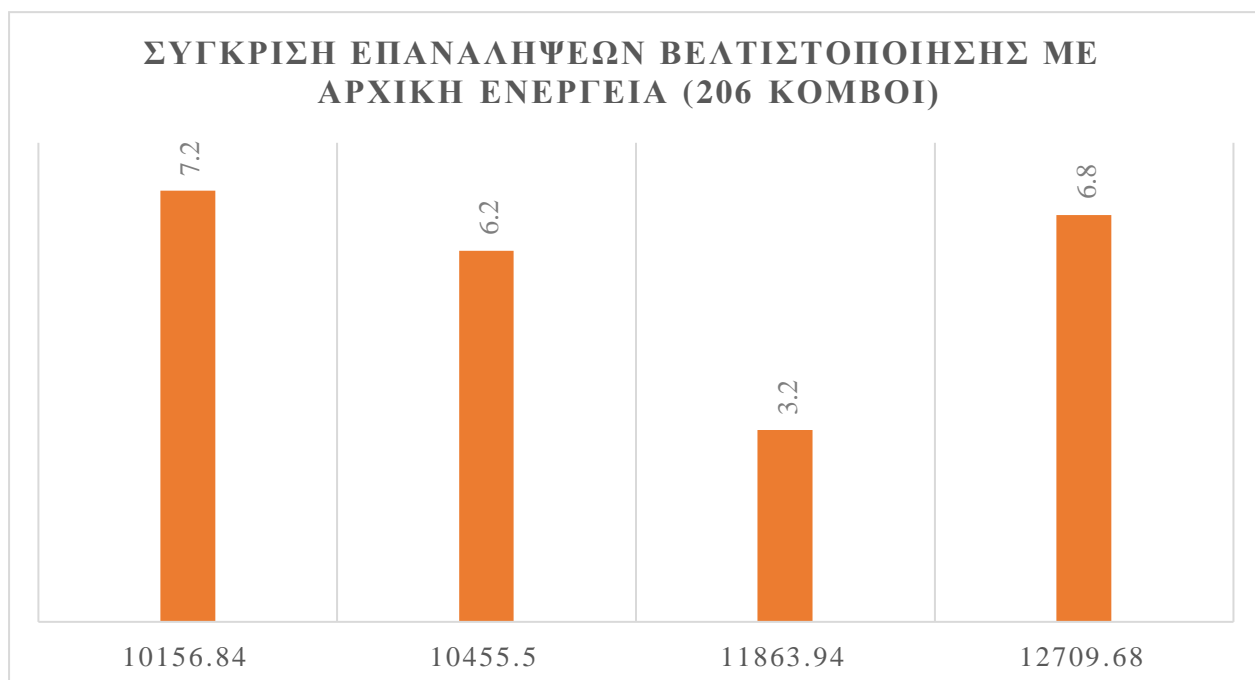
Γράφημα 9. Σύγκριση επαναλήψεων βελτιστοποίησης με την αρχική ενέργεια για τους 155 κόμβους

Στο προαναφερθέν γράφημα οι περισσότερες επαναλήψεις βελτιστοποίησης γίνονται στην μικρότερη αρχική ενέργεια και οι λιγότερες στην μεγαλύτερη



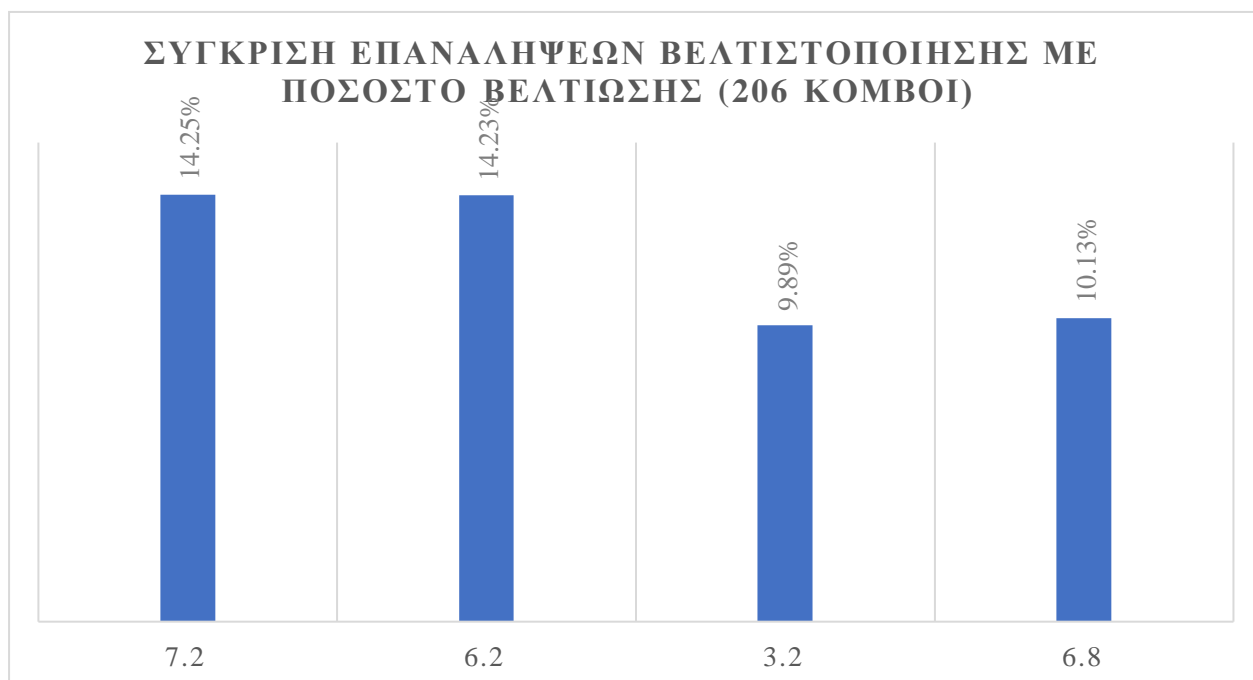
Γράφημα 10. Σύγκριση επαναλήψεων βελτιστοποίησης με το ποσοστό βελτίωσης για τους 155 κόμβους

Σε αυτό το γράφημα οι λιγότερες επαναλήψεις οδηγούν στην μικρότερη βελτιστοποίηση.



Γράφημα 11. Σύγκριση επαναλήψεων βελτιστοποίησης με την αρχική ενέργεια για τους 206 κόμβους

Σε αυτό το γράφημα οι περισσότερες επαναλήψεις πραγματοποιούνται στην μικρότερη αρχική ενέργεια.



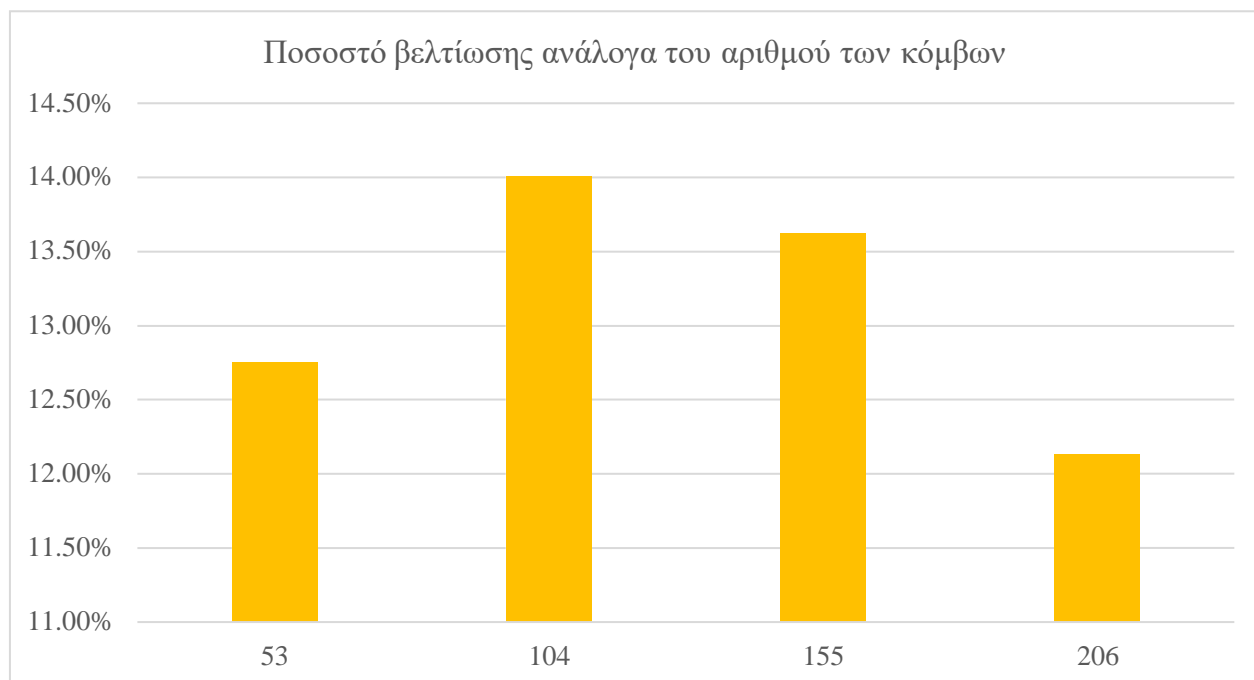
Γράφημα 12. Σύγκριση επαναλήψεων βελτιστοποίησης με το ποσοστό βελτίωσης για τους 206 κόμβους

Σε αυτό το γράφημα ο μεγαλύτερος αριθμός επαναλήψεων φέρει και μεγαλύτερο ποσοστό βελτιστοποίησης. Αντίθετα ο μικρότερος έχει ως αποτέλεσμα την χειρότερη βελτιστοποίηση συγκριτικά με τα άλλα παραδείγματα.

Γενικά βάσει των παραπάνω γραφημάτων, συνήθως ο μεγαλύτερος αριθμός επαναλήψεων φέρνει και το καλύτερο ποσοστό βελτιστοποίησης, ενώ από την άλλη ο μικρότερος αριθμός επαναλήψεων οδηγεί στο μικρότερο ποσοστό βελτιστοποίησης. Αυτό δεν ισχύει κατά κανόνα, ωστόσο προτιμάται μεγαλύτερος αριθμός επαναλήψεων, διότι στατιστικά υπάρχει μεγαλύτερη πιθανότητα να επιτευχθεί καλύτερη απόδοση.

Όσον αφορά τον αριθμό επαναλήψεων και την αρχική ενέργεια παρατηρείται ότι η μικρότερη αρχική ενέργεια έχει τις περισσότερες φορές μεγάλο αριθμό επαναλήψεων και σε αρκετές περιπτώσεις τον μεγαλύτερο, το οποίο όπως και πριν δεν ισχύει κατά κανόνα, αλλά στατιστικά.

Συμπερασματικά λοιπόν, σύμφωνα με όλα τα παραπάνω η μικρότερη αρχική ενέργεια στα δοθέντα παραδείγματα και στα συγκεκριμένα τρεξίματα του κώδικα οδηγεί κάθε φορά σε καλύτερη τελική βελτιωμένη ενέργεια. Επίσης, όσο περισσότερες επαναλήψεις έχουν πραγματοποιηθεί αυξάνεται η πιθανότητα να υλοποιηθεί μεγαλύτερο ποσοστό βελτιστοποίησης.

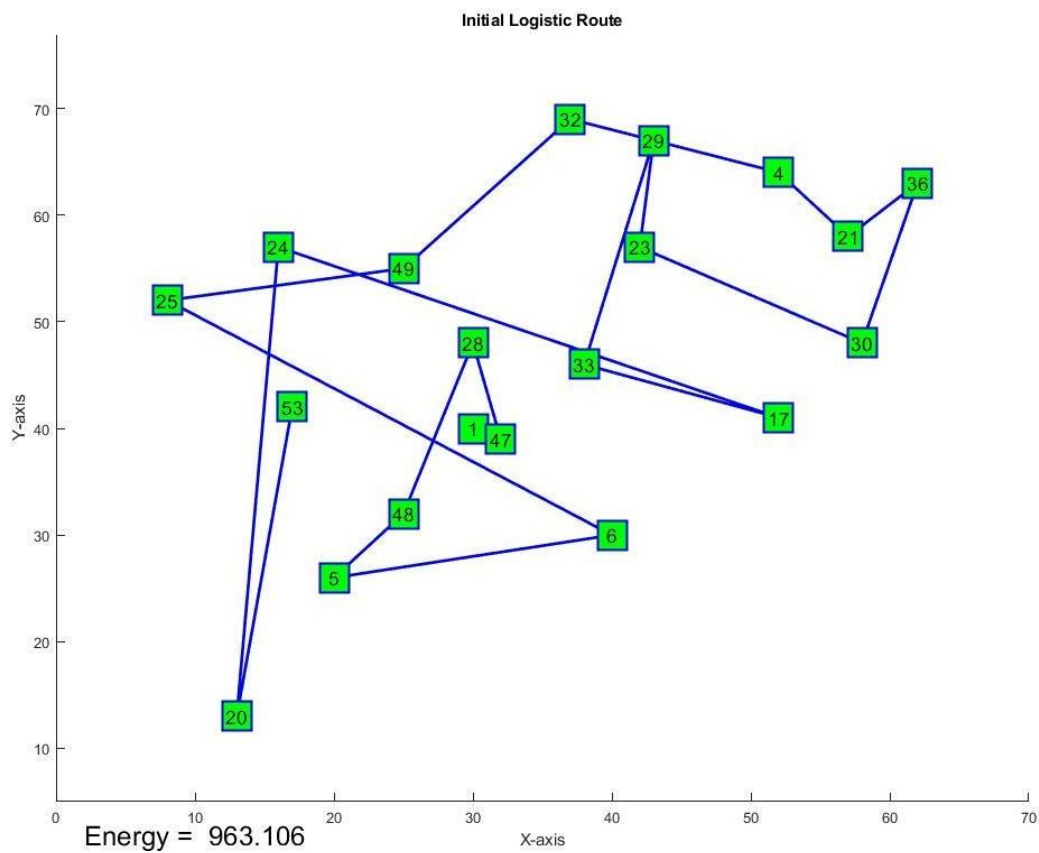


Γράφημα 13 Σύγκριση της βελτιστοποίησης των σετ δεδομένων με διαφορετικό αριθμό κόμβων

Λαμβάνοντας υπόψη το παραπάνω γράφημα, έχοντας προσμετρήσει τους μέσους όρους για κάθε 4 σετ δεδομένων έχουμε ως αποτέλεσμα ότι ο αλγόριθμος φαίνεται να λειτουργεί καλύτερα για τους 104 κόμβους έπειτα για τους 155, στην συνέχεια για τους 53 και τελευταία για τους 206. Κατά μέσο όρο λοιπόν, υπάρχει μια ικανοποιητική βελτιστοποίηση για κάθε αριθμό κόμβων κι ενδεχομένως ο κώδικας να έχει καλύτερη αντιμετώπιση στην βελτιστοποίηση ενός αριθμού κόμβων κοντά στους 104, ενώ όσο μικρότερος ή μεγαλύτερος είναι αυτός ο αριθμός να μην η λύση να είναι ικανοποιητική αλλά δεν φτάνει τα ίδια μεγάλα ποσοστά.

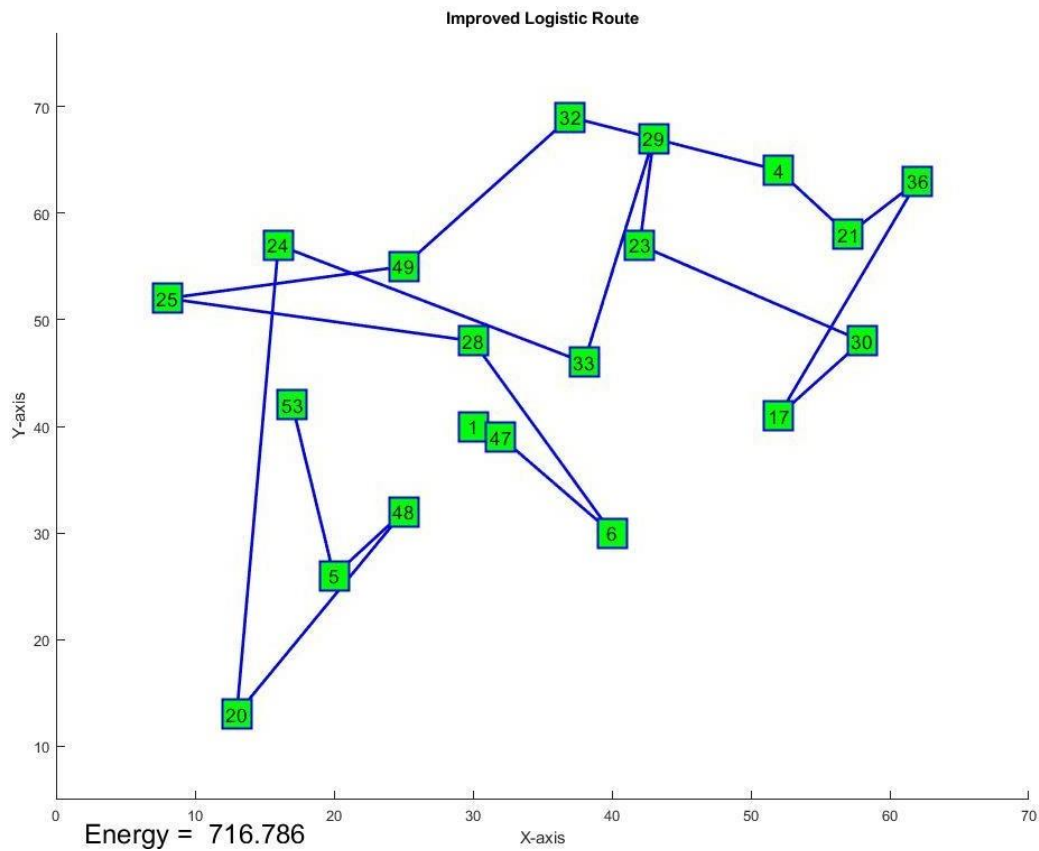
Ο χρόνος για κάθε βελτιστοποίηση δεν ξεπερνάει ποτέ τα 2 λεπτά και στις περισσότερες περιπτώσεις είναι μικρότερη των 30 δευτερολέπτων, οπότε θα μπορούσε να αποτελέσει πρακτική εφαρμογή.

Παρακάτω θα παρουσιαστούν 2 ζεύγη γραφημάτων. Στο πρώτο σκέλος εμφανίζεται η αρχική γειτονιά, ενώ στο δεύτερο η βελτιωμένη διαδρομή. Τα παρακάτω γραφήματα δημιουργήθηκαν μέσω των δεδομένων και των αποτελεσμάτων που λαμβάνει ένας κώδικας της matlab, `energy_of_drones.m` από το τον κώδικα που πραγματοποιείται η υλοποίηση του προβλήματος της παρούσας διπλωματικής εργασίας, `energy_of_drones.cpp`.



Γράφημα 14. Αρχική επιμέρους διαδρομή 1 ενός παραδείγματος των 53 κόμβων

Παραπάνω σκιαγραφείτε μια αρχικοποιημένη επιμέρους διαδρομή και φαίνεται ότι διάφορα τόξα είναι μεγάλου μήκους, προκαλώντας την διασταύρωση με άλλα τόξα και δημιουργώντας κύκλους στην διαδρομή. Αρκετά από αυτά επηρεάζουν αρνητικά την καταναλισκόμενη ενέργεια του drone.



Γράφημα 15. Βελτιωμένη επιμέρους διαδρομή 1 του γραφήματος 14.

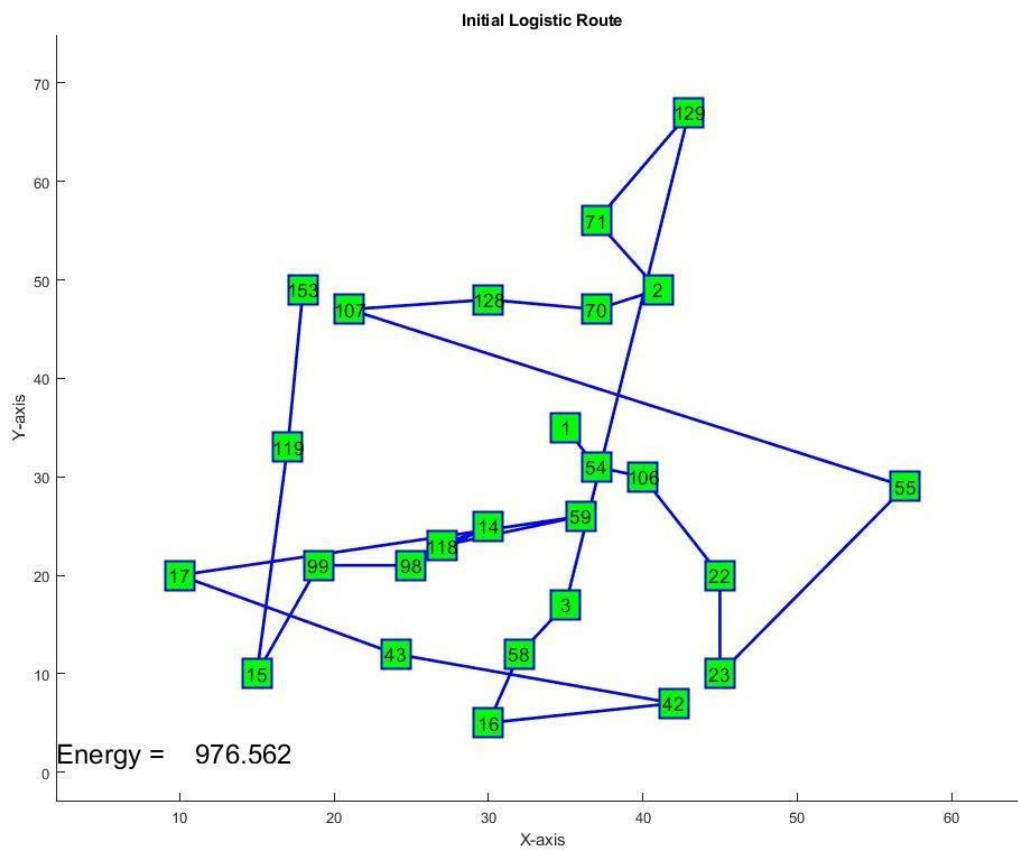
Και στα δύο γραφήματα αναγράφεται κάτω αριστερά η απαιτούμενη ενέργεια του drone. Η βελτίωση ανέρχεται στις 246.32 μονάδες. Η αρχική επιμέρους διαδρομή είναι η:

1 47 28 48 5 6 25 49 32 4 21 36 30 23 29 33 17 24 20 53

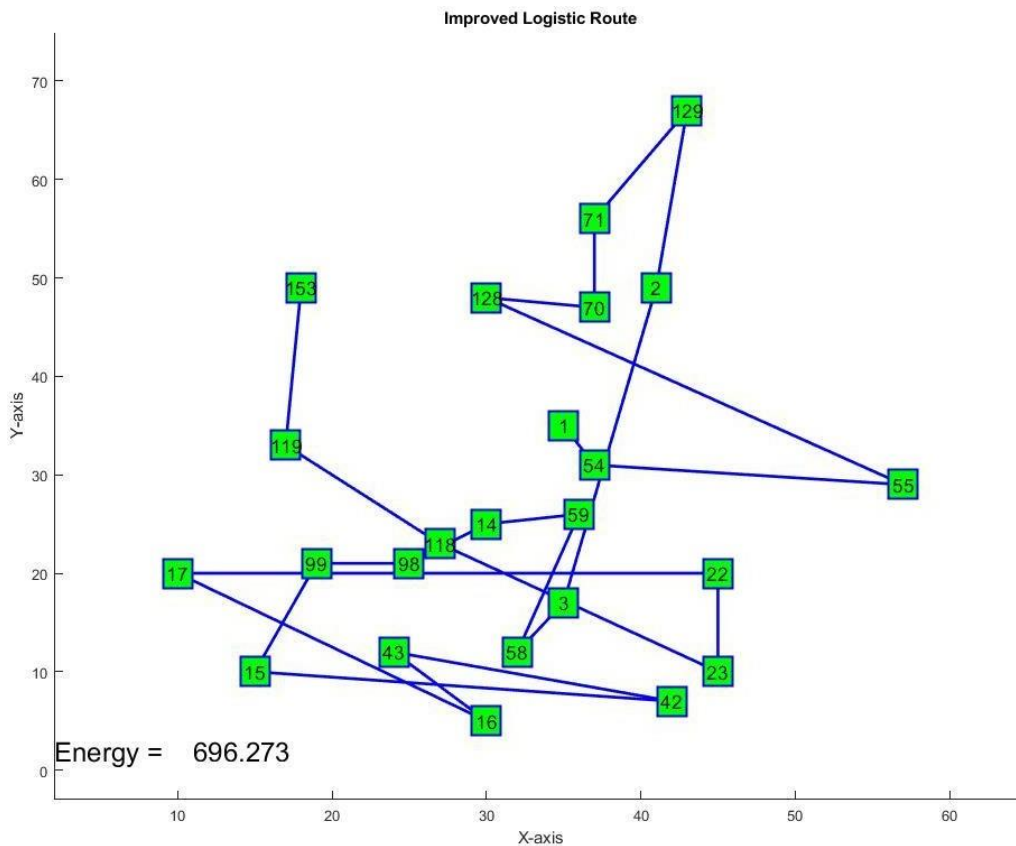
Και η βελτιωμένη:

1 47 6 28 25 49 32 4 21 36 17 30 23 29 33 24 20 48 5 53

Παρατηρείται ότι έχουν γίνει 2 επανατοποθετήσεις 1-0 και 1 επανατοποθέτηση 2-0. Με αυτές τις μεθόδους τοπικής αναζήτησης έχουν αποφευχθεί κάποιοι κύκλοι που δημιουργούνται και αρκετές μεταβάσεις από τον τρέχων κόμβο στον επόμενο είναι μικρότερης απόστασης. Ωστόσο αξίζει να σημειωθεί ότι κάποιες φορές προτιμάται να διανύσει το drone μεγαλύτερη απόσταση, διότι η ενέργεια που δαπανάται εξαρτάται κι από άλλους παράγοντες όπως το βάρος που μεταφέρεται την δεδομένη στιγμή.



Γράφημα 16. Αρχική επιμέρους διαδρομή 1 ενός παραδείγματος των 155 κόμβων



Γράφημα 17. Βελτιωμένη επιμέρους διαδρομή 1 του γραφήματος 16.

Στα γραφήματα 16 και 17 είναι αισθητή η αλλαγή στην διαδρομή με το μέγεθος των τόξων να είναι αρκετά μειωμένο, το οποίο αντικατοπτρίζεται κιόλας από τις 277.142 μονάδες μείωσης στην ενέργεια.

Η αρχική διαδρομή με 973.415 ενέργεια είναι η εξής:

1 54 106 22 23 55 107 128 70 2 71 129 3 58 16 42 43 17 59 118 14 98 99 15 119 153

Η βελτιωμένη λύση με 696.273 ενέργεια

1 54 55 128 70 71 129 2 3 58 59 14 98 99 15 42 43 16 17 22 23 118 119 153

Αξίζει να σημειωθεί ότι σε αυτήν την περίπτωση 2 κόμβοι (106 και 107) φεύγουν τελείως από την διαδρομή και εξυπηρετούνται σε κάποια άλλη επιμέρους διαδρομή επιτυγχάνοντας ομαλότερες μετακινήσεις του drone και μια αρκετά μεγάλη βελτιστοποίηση.

ΚΕΦΑΛΑΙΟ 7

ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην παρούσα διπλωματική εργασία εφαρμόστηκε ένας υβριδικός αλγόριθμος συνδυάζοντας την άπληστη τυχαιοποιημένη αναζήτηση με την εφαρμογή του αλγόριθμου Grasp για την αρχικοποίηση της λύσης και την μεταβλητή γειτονιάς αναζήτησης VNS για την βελτιστοποίηση με διάφορες μεθόδους τοπικής αναζήτησης.

Δοκιμάσαμε τον αλγόριθμο για 16 σετ δεδομένων εκ των οποίων συμπεριλαμβάνονται τέσσερις τετράδες με διαφορετικό αριθμό κόμβων η κάθε μια. Επίσης, κάθε σετ δεδομένων έχει διαφορετικούς πελάτες στο πρώτο και στο δεύτερο σύνολο. Πραγματοποιήθηκε η σύγκριση σε πρώτο στάδιο για τα αποτελέσματα των αρχείων με τον ίδιο αριθμό κόμβων και στην συνέχεια τα αποτελέσματα των αρχείων με διαφορετικό αριθμό κόμβων.

Αρχικά, παρατηρήθηκε ότι η καλύτερη αρχική λύση θα οδηγήσει και σε καλύτερη βελτιωμένη λύση ανεξάρτητα του ποσοστού βελτίωσης αυτής. Επιπρόσθετα, η ποιότητα της βελτιωμένης λύσης δείχνει να επηρεάζεται σε μεγάλο βαθμό από την αρχική ενέργεια και από τις επαναλήψεις βελτιστοποίησης που θα πραγματοποιήσει ο αλγόριθμος. Σημειώθηκαν περισσότερες επαναλήψεις στις χαμηλότερες αρχικές ενέργειες, στις περισσότερες περιπτώσεις, και κατά πλειοψηφία αυτός ο αριθμός επαναλήψεων οδηγούσε σε καλύτερο ποσοστό βελτιστοποίησης.

Ο χρόνος εκτέλεσης κατά μέσο όρο σημειώθηκε στα 1.29, 8.06, 21.19 και 60.75 δευτερόλεπτα αντίστοιχα για τους 53, 104, 155 και 206 κόμβους.

Ο αριθμός των μπαταριών που απαιτείται αυξάνεται αναλογικά σε σχέση με την τιμή της αρχικής ενέργειας της αρχικής συνολικής διαδρομής. Κατά μέσο όρο για τους προαναφερθέντες αριθμούς κόμβων κατά αύξουσα σειρά απαιτούνται 2.75, 5.45, 8.5 και 11.05 μπαταρίες, εξαιρουμένης της αρχικής εγκατεστημένης μπαταρίας του drone.

Όσον αφορά την σύγκριση των μέσων όρων κάθε μιας από τις τετράδες, βγήκε το συμπέρασμα ότι ο αλγόριθμος συμπεριφέρεται καλύτερα στους 104 κόμβους με έναν μέσο όρο στην βελτιστοποίηση της λύσης που αγγίζει το 14.01%, ενώ για τις άλλες περιπτώσεις το ποσοστό αυτό είναι μικρότερο. Πιο συγκεκριμένα, για τους υπόλοιπους μέσους όρους έχουμε μια βελτιστοποίηση της αρχικής λύσης της τάξεως του 12.75% για τους 53 κόμβους, 13.62% για τους 155 και 12.13% για τους 206, ποσοστά αρκετά ικανοποιητικά.

Τέλος, αξίζει να επισημανθεί ότι η βελτιστοποίηση της ενέργειας μπορεί να πραγματοποιηθεί με την μεταφορά ή την ανταλλαγή ενός ή παραπάνω κόμβων στην ίδια είτε σε κάποια άλλη επιμέρους διαδρομή. Συνήθως, η μεταφορά του drone από τον τρέχων κόμβο προτιμάται να είναι προς τον κοντινότερο δυνατό κόμβο, χωρίς ωστόσο αυτό να είναι πάντα εφικτό λόγω των περιορισμών του προβλήματος. Η βελτιστοποίηση θα μειώσει όσο το δυνατόν περισσότερο τις διασταυρώσεις τόξων και τις μεγάλες αποστάσεις, για όσο αυτό επιτρέπεται από τους περιορισμούς.

BIBΛΙΟΓΡΑΦΙΑ

- [1] Alsheddy, A., Voudouris, C., Tsang, K., P., E., & Alhindi, A. (2016). Guided local search. https://www.researchgate.net/profile/Ahmad-Alhindi/publication/312516867_Guided_Local_Search/links/5d194236a6fdcc2462b49043/Guided-Local-Search.pdf
- [2] Desale, S., Rasool, A., Andhale., S., & Rane, P. (2015). Heuristic and Meta-Heuristic Algorithms and Their relevance to the Real World: A Survey. International Journal of Computer Engineering in Research Trends, Vol. 2, 296-304. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=039da802e218c89a3c49e235af8e555b2fdfa063>
- [3] Hansen, P., Mladenovic, N., & Urošević, D. (2006). Variable neighborhood search and local branching. Computers & Operations Research, Elsevier Ltd, Vol. 33, 3034-3045. <https://www.sciencedirect.com/science/article/pii/S0305054805000894>
- [4] Kyriakakis, A., N., Stamadianos, T., Marinaki M., & Marinakis, Y. (2022). The electric vehicle routing problem with drones: An energy minimization approach for aerial deliveries. Cleaner Logistics and Supply Chain, Elsevier Ltd, Vol. 4, 100041. [The electric vehicle routing problem with drones: An energy minimization approach for aerial deliveries - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S2468013222000041)
- [5] Kokash, N. An introduction to heuristic algorithms. Department of Informatics and Telecommunications. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=8314bf30780871868076775ba62759f1faf8c9f0>
- [6] Liu, Y. (2019). An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones. Computers & Operations Research, Elsevier Ltd, Vol. 111, 1-20. [An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S0305054819000041)
- [7] Lambert, M., D., & Cooper, C., M. (2000). Issues in supply chain management. Industrial Marketing Management, Elsevier Science Inc, Vol. 29, 65-83. <https://www.sciencedirect.com/science/article/pii/S0019850199001133>
- [8] Min, H., & Zhou, G. (2002). Supply chain modeling: past, present and future. Computers and industrial engineering, Elsevier Science Ltd, Vol. 43, 231-249. <https://www.sciencedirect.com/science/article/pii/S0360835202000669>

[9] Murray, C., C., & Chu, G., A. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. Transportation Research Part C: Emerging Technologies, Elsevier Ltd, Vol. 54, 86-109.

[The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery - ScienceDirect](#)

[10] Sacramento, D., Pisinger, D., & Ropke, S. (2019). An adaptive large neighborhood search metaheuristic for the vehicle routing problem. Transportation Research Part C: Emerging Technologies, Elsevier Ltd, Vol. 102, 289-315.

[An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones - ScienceDirect](#)

[11] Stutzle, T. Applying iterated local search to the permutation flow shop problem. Department of Computer Science. 10, D-64283 Darmstadt.

<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=6c5b717fbc4ac191746238235bc6ac7b3f47b9ac>

[12] Μαρινάκης, Ι., & Μυγδαλάς, Α. (2008). Σχεδιασμός και Βελτιστοποίηση της Εφοδιαστικής Αλυσίδας, Εκδόσεις “σοφία”.

[13] https://en.wikipedia.org/wiki/Vehicle_routing_problem