ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
TECHNICAL UNIVERSITY OF CRETE

Σχολή Μηχανικών Παραγωγής και Διοίκησης

Πολυτεχνείο Κρήτης

Διπλωματική Εργασία

Ανάπτυξη μεθευρετικού αλγορίθμου για το πρόβλημα δρομολόγησης οχημάτων με σκοπό την μείωση των ρύπων

Ονοματεπώνυμο Φοιτητή: Δαρατσιανός Μιχάλης

Επιβλέπων Καθηγητής : Μαρινάκης Ιωάννης

Χανιά 2023

Ευχαριστίες

Η παρούσα εργασία εκπονήθηκε στα πλαίσια του προπτυχιακού κύκλου σπουδών, στο τμήμα Μηχανικών Παραγωγής και Διοίκησης στο Πολυτεχνείο Κρήτης. Αρχικά θα ήθελα να ευχαριστήσω τον καθηγητή και επιβλέποντα στην διπλωματική μου εργασία, κύριο Μαρινάκη Ιωάννη για την υποστήριξη του και την καθοδήγηση που μου παρείχε κατά την εκπόνηση της διπλωματικής μου εργασίας. Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου για την υποστήριξη τους σε κάθε μου απόφαση.

Περίληψη

Στον σύγχρονο κόσμο των παγκόσμιων logistics και των μεταφορών, το πρόβλημα της Δρομολόγησης Οχημάτων με σκοπό την Μείωση των Ρύπων (PRP) έχει αναδειχθεί ως μια κρίσιμη πρόκληση που απαιτεί καινοτόμες και βιώσιμες λύσεις. Καθώς ο κόσμος γίνεται ολοένα και πιο διασυνδεδεμένος, ο περιβαλλοντικός αντίκτυπος των συστημάτων μεταφορών έχει συγκεντρώσει αυξημένη προσοχή, απαιτώντας μια λεπτή ισορροπία μεταξύ οικονομικής απόδοσης και οικολογικής ευθύνης. Το PRP, το οποίο επιδιώκει να ελαχιστοποιήσει τον περιβαλλοντικό αντίκτυπο που σχετίζεται με τις αποφάσεις δρομολόγησης οχημάτων, ενσωματώνει αυτή τη διασταύρωση, απαιτώντας εξελιγμένες στρατηγικές βελτιστοποίησης για την αποτελεσματική επίλυσή του.

Αυτή η πτυχιακή εργασία αναλύει εις βάθος το PRP, παρουσιάζοντας μια προσέγγιση που αξιοποιεί τον συνδυασμό διαφορετικών τεχνικών βελτιστοποίησης. Ο προτεινόμενος αλγόριθμος ενσωματώνει τα δυνατά σημεία τεσσάρων διακριτών μεθόδων. Της μεθόδου του Πλησιέστερου Γείτονα, της μεθόδου 2-opt και 1-0 relocate καθώς και της μεθόδου της Προσομοιωμένης Ανόπτησης. Συνδυάζοντας αυτές τις προσεγγίσεις, στόχος είναι να δημιουργηθεί μια ευέλικτη και ισχυρή μεθοδολογία λύσης ικανή να αντιμετωπίσει τις πολύπλευρες προκλήσεις που ενυπάρχουν στο συγκεκριμένο πρόβλημα.

Abstract

In the modern world of global logistics and transportation, the Pollution Routing Problem (PRP) has emerged as a critical challenge that requires innovative and sustainable solutions. As the world becomes increasingly interconnected, the environmental impact of transportation systems has gained increased attention, requiring a delicate balance between economic efficiency and ecological responsibility. PRP, which seeks to minimize the environmental impact associated with vehicle routing decisions, incorporates this intersection, requiring sophisticated optimization strategies to effectively resolve it.

This thesis analyzes the PRP in depth, presenting an approach that exploits the combination of different optimization techniques. The proposed algorithm incorporates the strengths of four distinct methods. The Nearest Neighbor method, the 2-opt and 1-0 relocate methods as well as the Simulated Annealing method. By combining these approaches, the goal is to create a flexible and robust solution methodology capable of addressing the multifaceted challenges inherent in this problem.

Περιεχόμενα

1. Εισαγωγή στην Εφοδιαστική και Βασικές Έννοιες

Στον σημερινό παγκοσμιοποιημένο και διασυνδεδεμένο κόσμο, οι αλυσίδες εφοδιασμού γίνονται όλο και πιο περίπλοκες και αναγκαίες καλύπτοντας πολλαπλές γεωγραφικές τοποθεσίες και εμπλέκοντας πολλούς ενδιαφερόμενους φορείς. Επηρεάζονται από πολλούς παράγοντες, συμπεριλαμβανομένων των μεταβαλλόμενων απαιτήσεων των πελατών, της δυναμικής της αγοράς, των τεχνολογικών εξελίξεων και της περιβαλλοντικής βιωσιμότητας. Ως αποτέλεσμα, οι οργανισμοί πρέπει να υιοθετήσουν αποτελεσματικές πρακτικές διαχείρισης της εφοδιαστικής αλυσίδας για να παραμείνουν ανταγωνιστικοί, να βελτιώσουν την ικανοποίηση των πελατών τους, να βελτιστοποιήσουν τη λειτουργική αποτελεσματικότητα και να επιτύχουν βιώσιμη ανάπτυξη.

.

Εικόνα 1.1 *Εφοδιαστική Αλυσίδα*

1.1 Εφοδιαστική Αλυσίδα (Supply Chain)

Οι αλυσίδες εφοδιασμού είναι τα περίπλοκα δίκτυα οργανισμών, ανθρώπων, δραστηριοτήτων, πληροφοριών και πόρων που εμπλέκονται στην παραγωγή και διανομή αγαθών και υπηρεσιών. Λειτουργούν ως η ραχοκοκαλιά σχεδόν κάθε κλάδου, διασφαλίζοντας την έγκαιρη παράδοση των προϊόντων στους καταναλωτές.

Οδηγούμενοι από την παγκοσμιοποίηση και τις διαρκώς εκτεινόμενες απαιτήσεις των πελατών, η εφοδιαστική αλυσίδα παίζει καθοριστικό ρόλο στην δημιουργία πλεονεκτήματος για όλες τις επιχειρήσεις. Με τον όρο εφοδιαστική αλυσίδα εννοούμε όχι μόνο τη ροή υλικών από τον προμηθευτή πρώτων υλών ή τον κατασκευαστή μέχρι τον τελικό καταναλωτή, αλλά παράλληλα και την ροή πληροφοριών μεταξύ των μελών της ίδιας αλυσίδας. Αποτελείται απ‘ όλα τα στάδια που εμπλέκονται έμμεσα ή άμεσα, στην ικανοποίηση των απαιτήσεων των πελατών.

Εικόνα 1.2 *Ροή προϊόντων και πληροφοριών εντός της Εφοδιαστικής Αλυσίδας*

Μια αλυσίδα εφοδιασμού είναι το σύνολο των δραστηριοτήτων, των εγκαταστάσεων και των μέσων διανομής που απαιτούνται για τη διεξαγωγή ολόκληρης της διαδικασίας πώλησης ενός προϊόντος. Αυτό είναι, από την αναζήτηση πρώτων υλών, την επακόλουθη μετατροπή τους και την κατασκευή έως την μεταφορά και παράδοση στον τελικό καταναλωτή. Με άλλα λόγια, η εφοδιαστική αλυσίδα είναι μια στρατηγική και υλικοτεχνική λειτουργία που περιλαμβάνει όλες τις λειτουργίες που είναι απαραίτητες για ένα εμπόρευμα να φτάσει στον τελικό πελάτη σε βέλτιστες συνθήκες.

Ο κύριος στόχος της αλυσίδας εφοδιασμού είναι η ικανοποίηση των αναγκών του τελικού πελάτη με τον καλύτερο δυνατό τρόπο. Δηλαδή έχει ως σκοπό την παράδοση αγαθών και υπηρεσιών εγκαίρως, την αποφυγή περιττών απωλειών ή των απωλειών γενικότερα, την βελτιστοποίηση του χρόνου διανομής, την σωστή διαχείριση αποθεμάτων και αποθήκης, την καθιέρωση επαρκών καναλιών επικοινωνίας και συντονισμού και την αντιμετώπιση απρόβλεπτων αλλαγών στη ζήτηση, την προσφορά ή άλλων όρων.

1.2 Διαχείριση Εφοδιαστικής Αλυσίδας (Supply Chain Management)

Η διαχείριση της εφοδιαστικής αλυσίδας είναι ένας κρίσιμο κομμάτι που παίζει καθοριστικό ρόλο στην επιτυχία και την ανταγωνιστικότητα των οργανισμών σε διάφορους κλάδους. Περιλαμβάνει τον απρόσκοπτο συντονισμό και την ενοποίηση διαδικασιών, δραστηριοτήτων και πόρων για την παροχή προϊόντων και υπηρεσιών στους πελάτες με έγκαιρο, αποτελεσματικό και οικονομικά αποδοτικό τρόπο. Η διαχείριση της εφοδιαστικής αλυσίδας περιλαμβάνει ολόκληρο το ταξίδι ενός προϊόντος, από την έναρξή του ως πρώτες ύλες έως την τελική παράδοσή του στον τελικό καταναλωτή.

1.3 Εφοδιαστική (Logistics)

Τα Logistics περιλαμβάνουν τον προγραμματισμό και την εκτέλεση της αποθήκευσης και της διακίνησης αγαθών μεταξύ διαφορετικών σημείων της εφοδιαστικής αλυσίδας. Η διαχείριση της εφοδιαστικής αλυσίδας συντονίζει τις εγκαταστάσεις, τους ανθρώπους, τον εξοπλισμό και άλλους πόρους για να διασφαλίσει ότι τα προϊόντα μετακινούνται όταν πρέπει έτσι ώστε να υπάρχει χώρος για αυτά στην επόμενη στάση. Ο σχεδιασμός ανάλογα με την ζήτηση, τη μεταφορά (συμπεριλαμβανομένης της διαχείρισης στόλου), τη διαχείριση των αποθεμάτων, τον χειρισμό υλικών και την εκπλήρωση παραγγελιών είναι όλες διαδικασίες που εμπίπτουν στην εφοδιαστική.

1.3.1 Ο ρόλος των Logistics στη Διαχείριση της Εφοδιαστικής Αλυσίδας

Στη διαχείριση της εφοδιαστικής αλυσίδας, τα Logistics είναι υπεύθυνα για τη μετακίνηση και την αποθήκευση αγαθών και υπηρεσιών, μαζί με τα έγγραφα και τις αναφορές που καταγράφουν αυτές τις κινήσεις κατά τη διάρκεια της διαδρομής ενός αντικειμένου προς τον πελάτη.

Τα Logistics περιλαμβάνουν τις πολυάριθμες μεθόδους μεταφοράς που μεταφέρουν αποθέματα από τη μια τοποθεσία στην άλλη. Επομένως είναι υπεύθυνα για τον προσδιορισμό του σημείου που μπορούν να διατηρηθούν τα αγαθά σε κάθε στάδιο μέχρι να χρειαστούν σε άλλη τοποθεσία, κάτι που είναι απαραίτητο για την αποτελεσματική διαχείριση της εφοδιαστικής αλυσίδας.

1.3.2 Η σημαντικότητα των Logistics στις αλυσίδες εφοδιασμού

Τα Logistics είναι ένα κρίσιμο κομμάτι της εφοδιαστικής αλυσίδας επειδή διαχειρίζονται και παρακολουθούν το ανθρώπινο δυναμικό και τους πόρους που απαιτούνται για την αποθήκευση και τη μεταφορά αγαθών και υπηρεσιών. Τα Logistics διασφαλίζουν ότι τα υλικά και τα προϊόντα κινούνται αξιόπιστα τη σωστή στιγμή και εντός προϋπολογισμού.

Ορισμένες πτυχές της εφοδιαστικής που υποστηρίζουν τις αλυσίδες εφοδιασμού περιλαμβάνουν την παράδοση των σωστών προϊόντων την κατάλληλη στιγμή, την μείωση του κόστους και την βελτίωση της αποτελεσματικότητας, την βοήθεια στη διατήρηση των πελατών και στην αύξηση της αφοσίωσης τους, την παροχή μιας ξεχωριστής πρότασης αξίας για ορισμένες επιχειρήσεις και τέλος την παροχή ενός μέσου παράδοσης αγαθών από την πιο οικονομική τοποθεσία παραγωγής στην τοποθεσία του πελάτη.

Εικόνα 1.2 *Εφοδιαστική (Logistics)*

1.4 Εφοδιαστική Αλυσίδα εν Καιρό Πανδημίας Covid-19

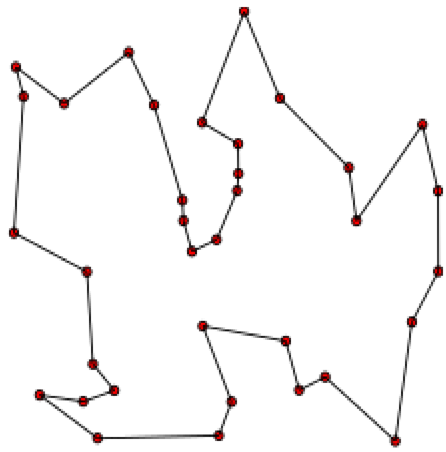
Η πανδημία COVID-19 υπογράμμισε την κρίσιμη σημασία μιας ανθεκτικής και αποτελεσματικής εφοδιαστικής αλυσίδας. Καθώς ο ιός εξαπλώθηκε παγκοσμίως, οι αλυσίδες εφοδιασμού αντιμετώπισαν πρωτόγνωρες προκλήσεις, που κυμαίνονταν από διακοπές στην παραγωγή και τη διανομή έως την αυξημένη ζήτηση για ορισμένα βασικά αγαθά. Η πανδημία εξέθεσε ευπάθειες στις αλυσίδες εφοδιασμού, υπογραμμίζοντας την ανάγκη για μεγαλύτερη ευελιξία και προσαρμοστικότητα. Οι αξιόπιστες και σωστά διαχειριζόμενες αλυσίδες εφοδιασμού κατάφεραν να δασώσουν την σταθερότητα της κοινωνίας, διασφαλίζοντας τη συνεχή ροή βασικών αγαθών, όπως ιατρικές προμήθειες, τρόφιμα και φαρμακευτικά προϊόντα. Οι εταιρείες που αντιμετώπισαν με επιτυχία την πολυπλοκότητα της αλυσίδας εφοδιασμού ήταν σε καλύτερη θέση για να ανταποκριθούν στις αυξήσεις της ζήτησης, να μετριάσουν τις διακοπές και γενικότερα να ανταποκριθούν στις συνολικές προκλήσεις του καιρού εκείνου.

Η πανδημία τόνισε τη διασύνδεση των παγκόσμιων αλυσίδων εφοδιασμού και την αναγκαιότητα προληπτικών στρατηγικών για την ενίσχυση της ανθεκτικότητας απέναντι σε απρόβλεπτες κρίσεις, ενισχύοντας τον κρίσιμο ρόλο της διαχείρισης της εφοδιαστικής αλυσίδας στη διατήρηση της κοινωνικής ευημερίας.

2. Προβλήματα Δρομολόγησης Οχημάτων

2.1 Πρόβλημα του Πλανόδιου Πωλητή

Το πρόβλημα του πλανόδιου πωλητή (Traveling Salesman Problem ή TSP) θέτει την ακόλουθη ερώτηση: "Δεδομένης μιας λίστας πόλεων και των αποστάσεων μεταξύ κάθε ζεύγους πόλεων, ποια είναι η συντομότερη δυνατή διαδρομή που επισκέπτεται κάθε πόλη ακριβώς μία φορά και επιστρέφει στην πόλη προέλευσης;" Είναι ένα NP-hard πρόβλημα στη συνδυαστική βελτιστοποίηση, σημαντικό στη θεωρητική επιστήμη των υπολογιστών και στην επιχειρησιακή έρευνα. Το πρόβλημα του ταξιδιώτη αγοραστή και το πρόβλημα δρομολόγησης οχημάτων είναι και οι δύο γενικεύσεις του TSP.



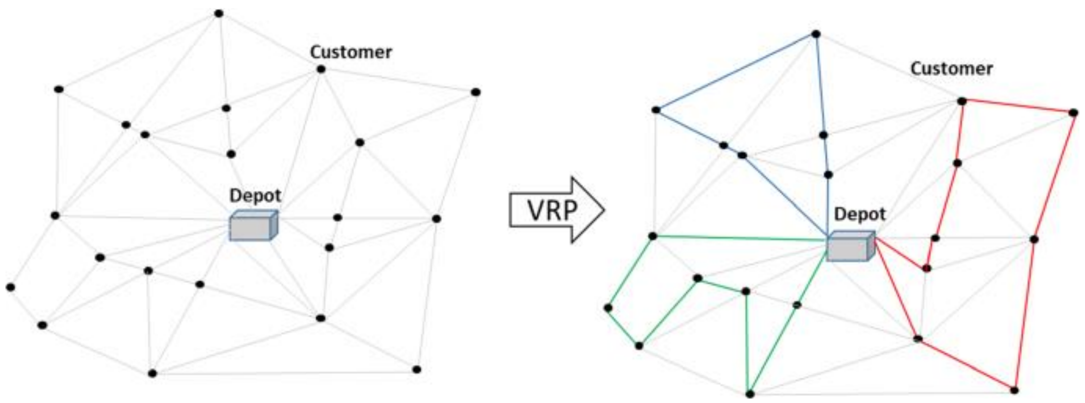
Εικόνα 2.1 *Παράδειγμα επίλυσης προβλήματος πλανόδιου πωλητή.*

Το πρόβλημα διατυπώθηκε για πρώτη φορά το 1930 και είναι ένα από τα πιο εντατικά μελετημένα προβλήματα βελτιστοποίησης. Χρησιμοποιείται ως σημείο αναφοράς για πολλές μεθόδους βελτιστοποίησης. Παρόλο που το πρόβλημα είναι υπολογιστικά δύσκολο, είναι γνωστοί πολλοί ευρετικοί και ακριβείς αλγόριθμοι, έτσι ώστε ορισμένες περιπτώσεις με δεκάδες χιλιάδες πόλεις να μπορούν να λυθούν πλήρως και ακόμη και προβλήματα με εκατομμύρια πόλεις μπορούν να προσεγγιστούν σε ένα μικρό κλάσμα του 1%.

Το TSP έχει πολλές εφαρμογές ακόμη και στην πιο καθαρή του σύνθεση, όπως για παράδειγμα στα logistics και στην κατασκευή μικροτσίπ. Ελαφρώς τροποποιημένο, εμφανίζεται ως υπό-πρόβλημα σε πολλούς τομείς, όπως η αλληλουχία DNA. Σε αυτές τις εφαρμογές, η έννοια πόλη αντιπροσωπεύει, για παράδειγμα, πελάτες, σημεία συγκόλλησης ή θραύσματα DNA και η έννοια της απόστασης αντιπροσωπεύει χρόνους ταξιδιού ή κόστος ή ένα μέτρο ομοιότητας μεταξύ των θραυσμάτων DNA. Το TSP εμφανίζεται επίσης στην αστρονομία, καθώς οι αστρονόμοι που παρατηρούν πολλές πηγές θα θέλουν να ελαχιστοποιήσουν τον χρόνο που αφιερώνουν μετακινώντας το τηλεσκόπιο μεταξύ των πηγών. Σε τέτοια προβλήματα, το TSP μπορεί να ενσωματωθεί σε ένα βέλτιστο πρόβλημα ελέγχου. Σε πολλές εφαρμογές, μπορεί να επιβληθούν πρόσθετοι περιορισμοί, όπως περιορισμένοι πόροι ή χρονικά παράθυρα.

2.2 Πρόβλημα Δρομολόγησης Οχημάτων

Το πρόβλημα δρομολόγησης οχημάτων (VRP) είναι ένα πρόβλημα συνδυαστικής βελτιστοποίησης και ακέραιου προγραμματισμού που θέτει την ερώτηση "Ποιο είναι το βέλτιστο σύνολο διαδρομών που πρέπει να διασχίσει ένας στόλος οχημάτων προκειμένου να παραδοθούν σε ένα δεδομένο σύνολο πελατών;" Γενικεύει το πρόβλημα του πλανόδιου πωλητή (TSP). Εμφανίστηκε για πρώτη φορά σε μια εργασία των George Dantzig και John Ramser το 1959, στην οποία γράφτηκε η πρώτη αλγοριθμική προσέγγιση και εφαρμόστηκε στις παραδόσεις βενζίνης. Συχνά το πλαίσιο είναι αυτό της παράδοσης αγαθών που βρίσκονται σε μια κεντρική αποθήκη σε πελάτες που έχουν κάνει παραγγελίες για τέτοια αγαθά. Ο στόχος του VRP είναι να ελαχιστοποιήσει το συνολικό κόστος διαδρομής. Το 1964, οι Clarke και Wright βελτίωσαν την προσέγγιση των Dantzig και Ramser χρησιμοποιώντας έναν αποτελεσματικό άπληστο αλγόριθμο που ονομάζεται αλγόριθμος αποταμίευσης.

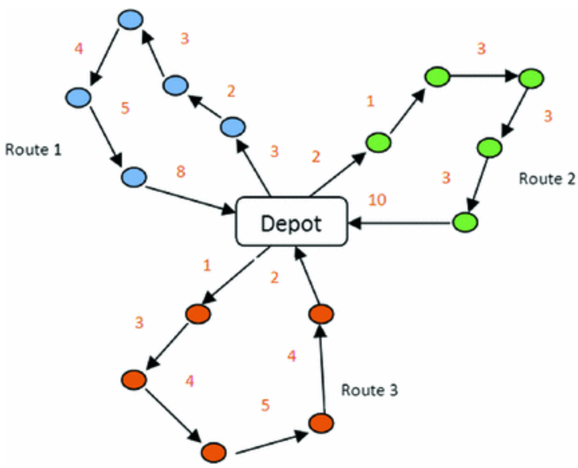


Εικόνα 2.2 *Εύρεση λύσης με βάση το πρόβλημα δρομολόγησης οχημάτων.*

Το πρόβλημα VRP είναι NP-hard, επομένως το μέγεθος των προβλημάτων που μπορούν να λυθούν βέλτιστα χρησιμοποιώντας μαθηματικό προγραμματισμό ή συνδυαστική βελτιστοποίηση μπορεί να είναι περιορισμένο. Επομένως, οι εμπορικοί λύτες τείνουν να χρησιμοποιούν ευρετικές μεθόδους λόγω του μεγέθους και της συχνότητας των VRP του πραγματικού κόσμου που πρέπει να επιλύσουν. Το VRP έχει πολλές άμεσες εφαρμογές στη βιομηχανία. Οι προμηθευτές εργαλείων δρομολόγησης VRP συχνά ισχυρίζονται ότι μπορούν να προσφέρουν εξοικονόμηση κόστους από 5% έως 30%.

Το VRP αφορά την εξυπηρέτηση μιας εταιρείας παράδοσης. Πώς παραδίδονται τα πράγματα από μία ή περισσότερες αποθήκες που διαθέτουν ένα δεδομένο σύνολο οχημάτων και λειτουργούν από ένα σύνολο οδηγών που μπορούν να κινηθούν σε ένα δεδομένο οδικό δίκτυο σε ένα σύνολο πελατών. Ζητά τον καθορισμό ενός συνόλου διαδρομών, S, (μία διαδρομή για κάθε όχημα που πρέπει να ξεκινά και να τελειώνει στη δική του αποθήκη) έτσι ώστε να ικανοποιούνται όλες οι απαιτήσεις και οι λειτουργικοί περιορισμοί των πελατών και να ελαχιστοποιείται το συνολικό κόστος μεταφοράς. Αυτό το κόστος μπορεί να ερμηνεύεται ως χρηματικό, απόστασης ή κάπως αλλιώς. Το οδικό δίκτυο μπορεί να περιγραφεί χρησιμοποιώντας ένα γράφημα όπου τα τόξα είναι δρόμοι και οι κορυφές είναι διασταυρώσεις μεταξύ τους. Τα τόξα μπορεί να είναι κατευθυνόμενα ή μη λόγω της πιθανής παρουσίας μονοδρομικών δρόμων ή διαφορετικού κόστους προς κάθε κατεύθυνση. Κάθε τόξο έχει ένα σχετικό κόστος που είναι γενικά το μήκος ή ο χρόνος ταξιδιού του που μπορεί να εξαρτάται από τον τύπο του οχήματος.

Για να γίνει γνωστό το συνολικό κόστος κάθε διαδρομής, πρέπει να είναι γνωστά το κόστος ταξιδιού και ο χρόνος ταξιδιού μεταξύ κάθε πελάτη και της αποθήκης. Για να γίνει αυτό, το αρχικό γράφημα μετατρέπεται σε ένα, όπου οι κορυφές είναι οι πελάτες και η αποθήκη και τα τόξα είναι οι δρόμοι μεταξύ τους. Το κόστος σε κάθε τόξο είναι το χαμηλότερο κόστος μεταξύ των δύο σημείων του αρχικού οδικού δικτύου. Αυτό είναι εύκολο να γίνει, καθώς τα προβλήματα της συντομότερης διαδρομής είναι σχετικά εύκολο να λυθούν. Αυτό μετατρέπει το αραιό αρχικό γράφημα σε πλήρες γράφημα. Για κάθε ζεύγος κορυφών i και j, υπάρχει ένα τόξο (i,j) του πλήρους γραφήματος του οποίου το κόστος γράφεται ως C{ij} και ορίζεται ότι είναι το κόστος της συντομότερης διαδρομής από το i στο j. Ο χρόνος ταξιδιού t{ij} είναι το άθροισμα των χρόνων διαδρομής των τόξων στη συντομότερη διαδρομή από το i έως το j στο αρχικό γράφημα δρόμου.



Εικόνα 2.3 *Απλό παράδειγμα δρομολόγησης οχημάτων και τα κόστη μεταξύ κάθε πελάτη και της αποθήκης.*

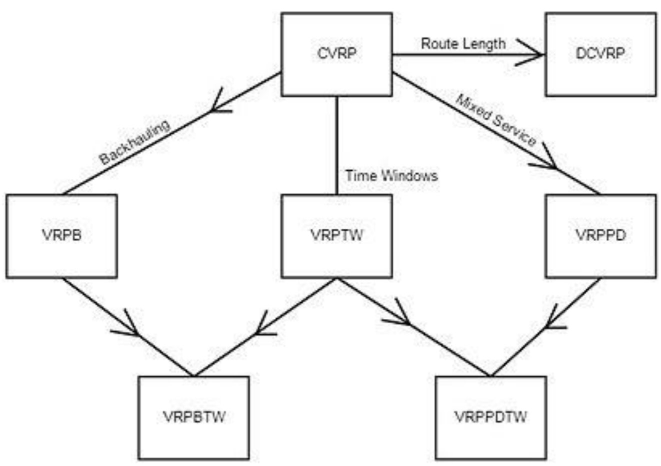
Μερικές φορές είναι αδύνατο να ικανοποιηθούν όλες οι απαιτήσεις ενός πελάτη και σε τέτοιες περιπτώσεις οι λύτες μπορεί να μειώσουν τις απαιτήσεις ορισμένων πελατών ή να αφήσουν ορισμένους πελάτες χωρίς εξυπηρέτηση. Για την αντιμετώπιση αυτών των καταστάσεων μπορεί να εισαχθεί μια μεταβλητή προτεραιότητας για κάθε πελάτη ή να δοθούν σχετικές κυρώσεις για τη μερική ή την έλλειψη εξυπηρέτησης για κάθε πελάτη.

Η αντικειμενική συνάρτηση ενός VRP μπορεί να είναι πολύ διαφορετική ανάλογα με τη συγκεκριμένη εφαρμογή του αποτελέσματος, αλλά μερικοί από τους πιο κοινούς στόχους είναι:

- Η ελαχιστοποίηση του συνολικού κόστους μεταφοράς με βάση την συνολική απόσταση που διανύθηκε καθώς και το πάγιο κόστος που σχετίζεται με τα χρησιμοποιούμενα οχήματα και τους οδηγούς
- Η ελαχιστοποίηση του αριθμού των οχημάτων που απαιτούνται για την εξυπηρέτηση όλων των πελατών
- Η ελαχιστοποίηση τις κυρώσεις για υπηρεσίες χαμηλής ποιότητας
- Η μεγιστοποίηση του συλλεγόμενου κέρδους.

2.3 Παραλλαγές στο Πρόβλημα Δρομολόγησης Οχημάτων

Υπάρχουν διάφορες παραλλαγές και εξειδικεύσεις του προβλήματος δρομολόγησης οχημάτων. Παρακάτω παρουσιάζονται μερικά υπό-προβλήματα του κοινού VRP ενώ στην εικόνα 2.3 φαίνεται και σχηματικά η σχέση των υπό-προβλημάτων αυτών.



Εικόνα 2.3 *Απεικόνιση της σχέσης μεταξύ κοινών υπό-προβλημάτων VRP.*

Πρόβλημα δρομολόγησης οχημάτων με κέρδη (VRPP)

Ένα πρόβλημα μεγιστοποίησης όπου δεν είναι υποχρεωτική η επίσκεψη σε όλους τους πελάτες. Ο στόχος είναι ο κάθε πελάτης να επισκέπτεται μία φορά μεγιστοποιώντας το άθροισμα των εισπραχθέντων κερδών με ταυτόχρονη τήρηση ενός χρονικού ορίου οχήματος. Τα οχήματα πρέπει να ξεκινούν και να τελειώνουν στο αμαξοστάσιο.

Μεταξύ των πιο γνωστών και μελετημένων VRPP είναι το πρόβλημα δρομολόγησης οχημάτων με παραλαβή και παράδοση (VRPPD). Στο συγκεκριμένο πρόβλημα ένας αριθμός αγαθών πρέπει να μετακινηθεί από συγκεκριμένες τοποθεσίες παραλαβής σε άλλες τοποθεσίες παράδοσης. Ο στόχος είναι να βρεθούν οι βέλτιστες διαδρομές για έναν στόλο οχημάτων που θα επισκεφθεί τις τοποθεσίες αυτές.

Πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα (VRPTW)

Οι τοποθεσίες παράδοσης έχουν χρονικά παράθυρα εντός των οποίων πρέπει να πραγματοποιηθούν οι παραδόσεις (ή οι επισκέψεις).

Πρόβλημα δρομολόγησης χωρητικότητας οχημάτων: CVRP

Τα οχήματα έχουν περιορισμένη μεταφορική ικανότητα των εμπορευμάτων που πρέπει να παραδοθούν.

Πρόβλημα δρομολόγησης οχημάτων με μεταφορές

Τα αγαθά μπορούν να μεταφερθούν μεταξύ οχημάτων σε ειδικά καθορισμένους κόμβους μεταφοράς.

Πρόβλημα δρομολόγησης οχημάτων με δύο είδη πελατών (VRPB)

Το πρόβλημα αυτό είναι μία επέκταση του βασικού προβλήματος δρομολόγησης οχημάτων όπου οι πελάτες χωρίζονται σε δύο υποσύνολα. Το πρώτο υποσύνολο είναι οι πελάτες που απαιτούν την διανομή κάποιας ποσότητας προϊόντων (linehaul customers). Το δεύτερο υποσύνολο αποτελείται από πελάτες που ο κάθε ένας εξ ‘αυτών απαιτεί μία ποσότητα του προϊόντος να περισυλλεχθεί από αυτόν (backhaul customers). Μία επέκταση του VRPB είναι και το VRPBTW όπου σε αυτό το πρόβλημα προστίθενται και οι περιορισμού χρονικών παραθύρων για κάθε πελάτη.

3. Αλγόριθμοι Επίλυσης Προβλημάτων Δρομολόγησης Οχημάτων

Τα προβλήματα δρομολόγησης οχημάτων (VRP) είναι σύνθετα προβλήματα βελτιστοποίησης που περιλαμβάνουν τον καθορισμό των πιο αποτελεσματικών διαδρομών για έναν στόλο οχημάτων για την παράδοση αγαθών ή υπηρεσιών σε ένα σύνολο τοποθεσιών. Για την αποτελεσματική επίλυση των VRP, έχουν αναπτυχθεί αρκετοί αλγόριθμοι που στοχεύουν στην εύρεση βέλτιστων ή σχεδόν βέλτιστων λύσεων. Αυτοί οι αλγόριθμοι αξιοποιούν τεχνικές από την επιχειρησιακή έρευνα, τη θεωρία γραφημάτων και τη βελτιστοποίηση για την αποτελεσματική εξερεύνηση του χώρου λύσεων.

Στο συγκεκριμένο κεφάλαιο δεν θα γίνει αναφορά σε ακριβείς μεθόδους που εγγυώνται μία βέλτιστη λύση. Ενώ αυτοί οι αλγόριθμοι είναι σχεδιασμένοι να βρίσκουν την βέλτιστη λύση, μπορούν να γίνουν υπολογιστικά ακριβοί για μεγάλες περιπτώσεις προβλημάτων. Η επιλογή του αλγορίθμου εξαρτάται από παράγοντες όπως το μέγεθος του προβλήματος, τους χρονικούς περιορισμούς και την επιθυμητή ποιότητα λύσεων. Οι ερευνητές και οι επαγγελματίες αναπτύσσουν και βελτιώνουν συνεχώς αυτούς τους αλγόριθμους για να αντιμετωπίσουν τις προκλήσεις που τίθενται από διαφορετικές παραλλαγές VRP και περιορισμούς του πραγματικού κόσμου.

Συνολικά, οι αλγόριθμοι που χρησιμοποιούνται για την επίλυση των VRP διαδραματίζουν κρίσιμο ρόλο στη βελτιστοποίηση της κατανομής των οχημάτων και των πόρων, οδηγώντας τελικά σε εξοικονόμηση κόστους, βελτιωμένη απόδοση και βελτιωμένη ικανοποίηση των πελατών.

3.1 Ευρετικοί Αλγόριθμοι

Η ανάπτυξη των ευρετικών αλγορίθμων προέκυψε ως αποτέλεσμα της ανάγκης για μειωμένο χρόνο επίλυσης των προβλημάτων, δηλαδή στην εύρεση εφικτών λύσεων σε γρήγορο χρονικό διάστημα. . Βέβαια, η απόλυτη βελτιστοποίηση έχει θυσιαστεί για χάρη της ταχύτητας και δεν μπορεί να επιτευχθεί καθώς οι αλγόριθμοι αυτοί πραγματοποιούν περιορισμένη αναζήτηση στο διάστημα των πιθανών λύσεων. Όμως παρέχονται καλής ποιότητας λύσεις σε σύντομο χρονικό διάστημα. Χαρακτηριστικό των αλγορίθμων αυτών είναι ότι για το ίδιο πρόβλημα γίνεται να παραχθούν καλύτερες λύσεις από παραπάνω από ένα ευρετικό αλγόριθμο, ανάλογα με τις τιμές των παραμέτρων που δίνονται. Δηλαδή για ορισμένες τιμές να δίνονται καλύτερες λύσεις από έναν συγκεκριμένο αλγόριθμο, ενώ για άλλες τιμές από έναν άλλο. Η καταλληλόλητα ενός ευρετικού αλγορίθμου καθορίζεται από τον αν η λύση συγκλίνει στην βέλτιστη, αν προκύπτει εύκολα και γρήγορα και αν τίθενται λογικοί κανόνες που οδηγούν στη λύση.

Υπάρχουν τρεις κατηγορίες ευρετικών αλγορίθμων :

– Αλγόριθμοι Απληστίας (Greedy Algorithms)

Στόχος τους είναι η εύρεση εφικτών λύσεων, όμως επειδή είναι μυωπικοί αλγόριθμοι (κοιτούν μόνο μπροστά), απαιτείται αρκετά μεγάλος χρόνος.

– Αλγόριθμοι Τοπικής Αναζήτησης (Local Search Algorithms)

Προσπαθούν να βελτιώσουν μία αρχική εφικτή λύση με κάποια μέθοδο αναζήτησης στη γειτονιά της λύσης.

– Προσεγγιστικοί Αλγόριθμοι (Approximation Algorithms)

Λύνουν το πρόβλημα χρησιμοποιώντας περισσότερες πληροφορίες.

Στην συγκεκριμένη εργασία γίνεται χρήση των δύο πρώτων κατηγοριών, οι οποίες κατηγορίες αναλύονται περεταίρω στην συνέχεια.

3.1.1 Αλγόριθμοι Απληστίας

Ένας αλγόριθμος απληστίας είναι ένα αλγοριθμικό παράδειγμα που ακολουθεί την ευρετική επίλυση προβλημάτων κάνοντας σε κάθε βήμα την τοπικά βέλτιστη επιλογή με την ελπίδα να βρεθεί ένα ολικό βέλτιστο. Με άλλα λόγια, ένας αλγόριθμος απληστίας παίρνει την καλύτερη δυνατή απόφαση σε κάθε βήμα χωρίς να εξετάζει τις συνέπειες, με στόχο να φτάσει στη συνολικά βέλτιστη λύση.

Οι αλγόριθμοι απληστίας χρησιμοποιούνται συχνά σε προβλήματα βελτιστοποίησης, όπου ο στόχος είναι να βρεθεί η καλύτερη λύση ανάμεσα σε ένα σύνολο πιθανών λύσεων. Ωστόσο, είναι σημαντικό να σημειωθεί ότι οι αλγόριθμοι απληστίας δεν εγγυώνται πάντα το συνολικό βέλτιστο, καθώς μπορεί να κάνουν τοπικά βέλτιστες επιλογές που οδηγούν σε μη βέλτιστες λύσεις. Επομένως, η ορθότητα ενός άπληστου αλγορίθμου εξαρτάται από το πρόβλημα που αντιμετωπίζουμε και συχνά απαιτείται απόδειξη ορθότητας.

Παρακάτω θα αναφερθούν κάποιοι από αυτούς τους αλγορίθμους.

Αλγόριθμος πλησιέστερου γείτονα

Στην συγκεκριμένη πτυχιακή εργασία γίνεται χρήση του συγκεκριμένου άπληστου αλγορίθμου για την εύρεση μίας αρχικής λύσης. Τα βήματα που ακολουθεί είναι:

Βήμα 1. Επιλέγεται η αποθήκη ως αρχικός κόμβος της διαδρομής.

Βήμα 2. Με βάση τον τελευταίο κόμβο που έχει προστεθεί στην διαδρομή και από τους κόμβους που δεν έχουν επισκεφθεί ακόμα αναζητείται ο κοντινότερος κόμβος σε αυτόν. Αν δεν παραβιάζονται οι περιορισμοί ο ίδιος προστίθεται στην διαδρομή. Αντίθετα αν παραβιάζεται κάποιος από τους περιορισμούς τότε ελέγχεται ο αμέσως κοντινότερος κόμβος που δεν έχει εξυπηρετηθεί και ταυτόχρονα δεν παραβιάζει τους περιορισμούς. Αν δεν υπάρχει τέτοιος κόμβος τότε η διαδρομή κλείνει επιστρέφοντας στην αποθήκη και επαναλαμβάνοντας το πρώτο βήμα για τη δημιουργία νέας διαδρομής.

Βήμα 3. Το βήμα 2 επαναλαμβάνεται έως ότου όλοι οι κόμβοι να έχουν εξυπηρετηθεί.

Nearest Insertion

Μια διαδικασία εισαγωγής όπου επιλέγεται μια υπό-διαδρομή σε k κόμβους στην επανάληψη k και καθορίζει ποιος από τους υπόλοιπους n-k κόμβους θα εισαχθεί στην επόμενη υπό-διαδρομή (το βήμα επιλογής) και πού (μεταξύ ποιων δύο κόμβων) θα πρέπει να εισαχθεί (το βήμα εισαγωγής). Τα βήματα είναι τα εξής:

Βήμα 1. Η διαδικασία ξεκινάει από μία κυκλική διαδρομή που αποτελείται από έναν μόνο κόμβο i.

Βήμα 2. Εύρεση ενός κόμβου r που ελαχιστοποιεί το κόστος Cir και δημιουργείται η διαδρομή i-r-i.

Βήμα 3. (Βήμα επιλογής) Δεδομένης μιας υπό-διαδρομής, επιλέγεται ένας κόμβος r που δεν περιλαμβάνεται στην υπό-διαδρομή και είναι κοντινότερος σε κάποιο κόμβο j της υπό-διαδρομής, π.χ με μικρότερο κόστος Crj.

Βήμα 4. (Βήμα εισαγωγής) Αναζητείται το τόξο (i, j) στην υπό-διαδρομή που ελαχιστοποιεί το Cir + Crj – Cij. Ο κόμβος r προστίθεται μεταξύ των κόμβων i και j.

Βήμα 5. Αν όλοι οι κόμβοι έχουν προστεθεί στην διαδρομή τότε ο αλγόριθμος τερματίζεται, διαφορετικά επαναλαμβάνεται το βήμα 3.

Farthest Insertion

Βήμα 1. Η διαδικασία ξεκινάει από μία κυκλική διαδρομή που αποτελείται από έναν μόνο κόμβο i.

Βήμα 2. Εύρεση ενός κόμβου r που μεγιστοποιεί το κόστος Cir και δημιουργείται η διαδρομή i-r-i.

Βήμα 3. (Βήμα επιλογής) Δεδομένης μιας υπό-διαδρομής, επιλέγεται ένας κόμβος r που δεν περιλαμβάνεται στην υπό-διαδρομή και είναι πιο μακρινός σε κάποιο κόμβο j της υπό-διαδρομής, π.χ με μεγαλύτερο κόστος Crj.

Βήμα 4. (Βήμα εισαγωγής) Αναζητείται το τόξο (i, j) στην υπό-διαδρομή που ελαχιστοποιεί το Cir + Crj – Cij. Ο κόμβος r προστίθεται μεταξύ των κόμβων i και j.

Βήμα 5. Αν όλοι οι κόμβοι έχουν προστεθεί στην διαδρομή τότε ο αλγόριθμος τερματίζεται, διαφορετικά επαναλαμβάνεται το βήμα 3.

Αλγόριθμος Clarke & Wright

Κάθε βήμα αποτελείται υπολογίζοντας όλες τις εξοικονομήσεις όλων των πελατών και έπειτα με βάση την καλύτερη εξοικονόμηση δημιουργείται η διαδρομή η οποία πρέπει να είναι εφικτή και να μην παραβιάζεται κάποιος περιορισμός. Υπολογίζονται οι εξοικονομήσεις για όλα τα ζεύγη πελατών που ισούνται με Sij = C1i – Cij + Cj1.

3.1.2 Αλγόριθμοι Τοπικής Αναζήτησης

Αυτοί οι αλγόριθμοι ξεκινούν με μια αρχική εφικτή λύση και τη βελτιώνουν επαναληπτικά εξερευνώντας τη γειτονιά της τρέχουσας λύσης. Χρησιμοποιούν διάφορες τεχνικές όπως 2-opt, 3-opt και Or-opt για την αφαίρεση ή την αναδιάταξη των κόμβων μέσα στις διαδρομές. Οι τοπικές μέθοδοι αναζήτησης συχνά συμπληρώνουν άλλους αλγόριθμους για να βελτιώσουν την απόδοσή τους.

Στην συγκεκριμένη διπλωματική εργασία χρησιμοποιήθηκαν οι αλγόριθμοι :

2-opt

Αυτή η μέθοδος διαγράφει 2 ακμές και επανασυνδέει δύο μονοπάτια με διαφορετικό τρόπο έτσι ώστε να δημιουργηθεί μια καινούρια καλύτερη εφικτή διαδρομή.

1-0 επανατοποθέτηση (1-0 relocate)

Αυτή η μέθοδος διαγράφει έναν κόμβο από μία διαδρομή και τον μετακινεί/επανατοποθετεί σε ένα άλλο σημείο εφόσον δώσει καλύτερο συνολικό κόστος.

Άλλοι γνωστοί αλγόριθμοι τοπικής αναζήτησης είναι:

1-1 ανταλλαγή (1-1 Exchange)

Αυτή η μέθοδος ανταλλάσσει θέσεις ανάμεσα σε δύο κόμβους.

2-2 ανταλλαγή (2-2 Exchange)

Αυτή η μέθοδος ανταλλάσσει θέσεις μεταξύ δύο γειτονικών κόμβων με άλλους δύο διαφορετικούς γειτονικούς κόμβους.

3-3 ανταλλαγή (3-3 Exchange)

Αυτή η μέθοδος ανταλλάσσει θέσεις μεταξύ τριών γειτονικών κόμβων με άλλους τρείς διαφορετικούς γειτονικούς κόμβους.

3.2 Μεθευρετικοί Αλγόριθμοι

Κατά την εφαρμογή μίας μεθόδου τοπικής αναζήτησης είναι πολύ εύκολο αυτή να συγκλίνει σε ένα τοπικό ελάχιστο. Ο αλγόριθμος τοπικής αναζήτησης είναι πολύ ευαίσθητος στην αρχική λύση. Δηλαδή αν η αρχική λύση είναι πολύ καλή και βρίσκεται πολύ κοντά σε κάποιο τοπικό ελάχιστο τότε με την διαδικασία της τοπικής αναζήτησης θα βρεθεί πολύ εύκολα το τοπικό ελάχιστο αλλά υπάρχει πιθανότητα ο αλγόριθμος να παγιδευτεί σε αυτό και να μην μπορεί να βελτιώσει άλλο την λύση. Για την επίλυση αυτού του προβλήματος έχουν προταθεί πολλοί αλγόριθμοι που βοηθούν την αναζήτηση να ξεφύγει από κάποιο τοπικό ελάχιστο, οι οποίοι ονομάζονται μεθευρετικοί αλγόριθμοι.

Οι μεθευρετικοί αλγόριθμοι μπορούν να διαχωριστούν ανάλογα με το πόσες λύσεις χρησιμοποιούν. Υπάρχουν οι αλγόριθμοι που χρησιμοποιούν μία λύση και κάνουν αναζήτηση στη γειτονιά αυτής της λύσης, δηλαδή επικεντρώνονται στην τροποποίηση και βελτίωση μιας μόνο υποψήφιας λύσης. Επίσης υπάρχουν και οι αλγόριθμοι που έχουν ένα πληθυσμό λύσεων και προσπαθούν να κάνουν αναζήτηση σε όλο το χώρο λύσεων, βελτιώνουν δηλαδή πολλαπλές υποψήφιες λύσεις, συχνά χρησιμοποιώντας τα χαρακτηριστικά του πληθυσμού για να καθοδηγήσουν την αναζήτηση.

Οι αλγόριθμοι αυτοί χωρίζονται σε 4 κύριες κατηγορίες ανάλογα με τον τρόπο που επιλέγουν για να αποφύγουν το τοπικό ελάχιστο:

1. Επαναληπτικές διαδικασίες που ξεκινούν από διαφορετικές αρχικές λύσεις.

2. Αλγόριθμοι που δέχονται γειτονικές κινήσεις χωρίς να βελτιώνουν τη λύση, αλλά κάποια επόμενη κίνηση μπορεί να οδηγήσει στο να αποφευχθεί το τοπικό ελάχιστο καταλήγοντας σε κάποιο επόμενο τοπικό ελάχιστο το οποίο να είναι καλύτερο από το

τρέχον. Ένας τέτοιος αλγόριθμος είναι και αυτός που χρησιμοποιείται στην συγκεκριμένη εργασία όπου είναι ο αλγόριθμος προσομοιωμένης ανόπτησης (simulated annealing).

3. Αλγόριθμοι που αλλάζουν τη γειτονιά αναζήτησης εφόσον παγιδευτούν σε κάποιο τοπικό ελάχιστο.

4. Αλγόριθμοι που αλλάζουν την αντικειμενική συνάρτηση ή κάποια από τα δεδομένα του προβλήματος.

Προσομοιωμένη ανόπτηση (Simulated Annealing)

Ο αλγόριθμος προσομοιωμένης ανόπτησης προέρχεται από την αναλογία ανάμεσα στην προσομοίωση της ανόπτησης των υλικών και την στρατηγική επίλυση προβλημάτων συνδυαστικής βελτιστοποίησης.

Στην προσομοιωμένη ανόπτηση η γειτονιά N(s) καθορίζει καινούργιες καταστάσεις που μπορεί να φτάσει κάποιος από την τρέχουσα κατάσταση s. Από την αρχική κατάσταση s0 με μία τυχαία διαταραχή του συστήματος γεννιέται μία καινούργια λύση s' . Το αποτέλεσμα της αλλαγής αυτής στην τιμή της αντικειμενικής συνάρτησης υπολογίζεται ως εξής:

$$\delta = f(s') - f(s0) \qquad (1)$$

Εάν το $\delta < 0$ η καινούργια κατάσταση είναι πάντοτε αποδεκτή. Εάν η αλλαγή δεν είναι αρνητική, δηλαδή $\delta \geq 0$, η καινούργια κατάσταση είναι αποδεκτή με κάποια πιθανότητα.

Η λογική υπολογισμού της πιθανότητας προέρχεται από τους νόμους της Θερμοδυναμικής. Έτσι, έχουμε στη θερμοκρασία T, η πιθανότητα της αύξησης στην ενέργεια της ποσότητας δ δίνεται από την σχέση:

$$p(\delta) = e^{-(\delta/kt)} \qquad (2)$$

όπου k είναι η σταθερά Boltzmann (η οποία δεν έχει κανένα απολύτως νόημα στην βελτιστοποίηση και για αυτό το λόγο αγνοείται).

Αυτό σημαίνει ότι η προσομοιωμένη ανόπτηση προσφέρει έναν μηχανισμό για την αποδοχή μικρών αυξήσεων στην τιμή της αντικειμενικής συνάρτησης, ελέγχοντας την πιθανότητα της αποδοχής p(δ) της νέας αντικειμενικής τιμής δια μέσου της θερμοκρασίας. Επομένως σε υψηλότερες θερμοκρασίες είναι πιο πιθανό μία λύση με χειρότερη τιμή στην συνάρτηση κόστους να είναι αποδεκτή ως η καινούργια κατάσταση του προβλήματος. Σε κάθε επανάληψη η θερμοκρασία μειώνεται έτσι ώστε όσο προχωράει ο αλγόριθμος να μειώνονται οι πιθανότητες αποδοχής χειρότερης λύσης. Επομένως η πιθανότητα αποδοχής μίας λύσης είναι:

$p(\delta) = e^{-(\delta/t)}$ αν $\delta \geq 0$ ή 1 αν $\delta < 0$. (3)

Είναι πολύ σημαντική η επιλογή της αρχικής θερμοκρασίας. Αν η θερμοκρασία είναι πολύ μεγάλη ή γίνει $T = \infty$ τότε στην ουσία γίνεται αποδοχή όλων των λύσεων άρα η μέθοδος μετατρέπεται σε μία τυχαιοποιημένη μέθοδο. Αντίθετα αν η θερμοκρασία είναι πολύ μικρή ή γίνει $T = 0$ τότε στην ουσία δεν γίνεται αποδοχή καμίας λύσης που να χειροτερεύει την λύση, άρα ο αλγόριθμος μετατρέπεται σε αλγόριθμο τοπικής αναζήτησης.

Στην συνέχεια παρουσιάζεται ένας ψευδοκώδικας προσομοιωμένης ανόπτησης:

Αλγόριθμος Προσομοιωμένη Ανόπτηση

Αρχικοποίηση

Επέλεξε μια αρχική λύση s0

Επέλεξε μια αρχική θερμοκρασία t0

Επέλεξε μία συνάρτηση μείωσης της θερμοκρασίας α(t)

repeat

    repeat

        τυχαία επιλογή μίας γειτονίας s Ε N(s0)

        $\delta = f(s) - f(s0)$

        if $\delta < 0$ **then**

            s0 = s

        else

            δημιουργείται τυχαία x , ομοιόμορφα κατανεμημένα στην ακτίνα (0,1)

            if $x < e^{-(\delta/t)}$ **then**

                s0 = s

            endif

        endif

    until ο μέγιστος αριθμός επαναλήψεων ολοκληρωθεί

    t = α(t)

until κάποιο κριτήριο τερματισμού ικανοποιηθεί

Επέστρεψε τη βέλτιστη λύση

Περιορισμένη Αναζήτηση (Tabu Search)

Ο αλγόριθμος περιορισμένης αναζήτησης χρησιμοποιεί έναν αλγόριθμο τοπικής αναζήτησης για να μετακινηθεί από την μία λύση στην άλλη. Η στρατηγική που χρησιμοποιεί ο αλγόριθμος αν η λύση έχει παγιδευτεί σε τοπικό ελάχιστο είναι η χρήση μνήμης. Με την διαδικασία της μνήμης μικρής περιόδου ο αλγόριθμος “θυμάται” τις τελευταίες κινήσεις που έκανε καθώς αυτές αποθηκεύονται σε μία λίστα περιορισμένων κινήσεων και οι συγκεκριμένες κινήσεις απαγορεύεται να επιστρέψουν στην λύση για ένα συγκεκριμένο αριθμό επαναλήψεων. Σε κάθε επανάληψη προστίθεται στην λίστα η τελευταία κίνηση που έγινε και αν έχει συμπληρωθεί το μέγεθος της λίστας (όπου ορίζεται από μία μεταβλητή k) τότε η παλιότερη λύση αφαιρείται από την λίστα. Αν κάποια κίνηση που βρίσκεται στη λίστα έχει ως αποτέλεσμα μία καλύτερη συνάρτηση κόστους, τότε προφανώς αγνοείται η λίστα και η αγνόηση αυτή ονομάζεται κριτήριο απενεργοποίησης των περιορισμών.

Διαδικασία Άπληστης Τυχαιοποιημένης Προσαρμοστικής Αναζήτησης (Greedy Randomized Adaptive Search Procedure – GRASP)

Η διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης είναι μία επαναληπτική διαδικασία για την εύρεση προσεγγιστικών λύσεων σε προβλήματα συνδυαστικής βελτιστοποίησης. Αυτή η τυχαιοποιημένη τεχνική παρέχει μία εφικτή λύση σε κάθε επανάληψη. Οι επαναλήψεις σταματούν όταν κάποιο κριτήριο τερματισμού ικανοποιείται και το τελικό αποτέλεσμα είναι απλά η καλύτερη λύση που βρέθηκε από όλες τις επαναλήψεις.

Κάθε επανάληψη αποτελείται από δύο φάσεις, μία φάση κατασκευής μίας αρχικής λύσης και μία διαδικασία τοπικής αναζήτησης για βελτιστοποίηση αυτής της λύσης. Στη φάση κατασκευής, μία τυχαιοποιημένη συνάρτηση απληστίας χρησιμοποιείται για να κατασκευαστεί μία αρχική λύση με τη χρήση μίας στρατηγικής επιλογής του επόμενου στοιχείου που ονομάζεται λίστα περιορισμού των υποψηφίων. Αυτή η αρχική λύση στην συνέχεια βελτιώνεται με τη χρήση της διαδικασίας τοπικής αναζήτησης.

Αλγόριθμος Επανασύνδεσης Διαδρομών (Path Relinking - PR)

Η μέθοδος της επανασύνδεσης διαδρομών δημιουργεί καινούριες λύσεις με την εξέταση διαφορετικών τροχιών που συνδέουν υψηλής ποιότητας λύσεις. Ο αλγόριθμος ξεκινάει από μία από αυτές τις λύσεις που ονομάζεται εναρκτήρια λύση και λύση στόχου η τελική λύση. Οι ρόλοι των δύο λύσεων μπορούν να αλλάξουν. Έτσι, τις περισσότερες φορές η καλύτερη από τις δύο λύσεις παίζει το ρόλο της αρχικής λύσης και η χειρότερη παίζει το ρόλο της τελικής λύσης. Υπάρχει η πιθανότητα και οι δυο διαδρομές να εξερευνηθούν ταυτόχρονα. Το πολύ δυνατό χαρακτηριστικό του αλγορίθμου της επανασύνδεσης διαδρομών είναι ότι μπορεί να χρησιμοποιηθεί μαζί με οποιονδήποτε αλγόριθμο μπορεί να παράγει πάνω από δύο λύσεις.

Αλγόριθμος Μεταβλητής Γειτονίας Αναζήτησης (Variable Neighborhood Search - VNS)

Ο αλγόριθμος της μεταβλητής γειτονιάς αναζήτησης προτάθηκε από τους Hansen και Mladenovic. Η βασική ιδέα της μεθόδου είναι η επιτυχής αναζήτηση ενός συνόλου από γειτονιές για να βρεθεί μία καλύτερη λύση. Ο όρος γειτονίες αναφέρεται σε διαφορετικούς αλγορίθμους τοπικής αναζήτησης. Η αναζήτηση πραγματοποιείται είτε με τυχαίο είτε με πιο συστηματικό τρόπο με στόχο να ξεφύγει ο αλγόριθμος από κάποιο τοπικό ελάχιστο. Η μέθοδος αυτή εκμεταλλεύεται το γεγονός ότι διαφορετικές μέθοδοι τοπικής αναζήτησης μπορούν να οδηγήσουν σε διαφορετικά τοπικά ελάχιστα. Αν N είναι η γειτονιά που εξετάζεται και βρεθεί κάποια καλύτερη λύση τότε η διαδικασία ξεκινάει πάλι από την αρχή μέχρι να μην υπάρχει άλλο περιθώριο βελτίωσης. Αν δεν βρεθεί στην N γειτονιά καλύτερη λύση τότε η διαδικασία συνεχίζεται με την εξέταση στην N+1 που είναι συμπληρωματική της.

4. Το Πρόβλημα Δρομολόγησης Οχημάτων με στόχο την Αποφυγή της Ρύπανσης (The Pollution Routing Problem (PRP))

4.1 Εισαγωγή

Το Πρόβλημα Δρομολόγησης Οχημάτων με Στόχο την Αποφυγή της Ρύπανσης (The Pollution Routing Problem – PRP) είναι μία σχετικά πρόσφατη παραλλαγή του Προβλήματος Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα η οποία αφορά την δρομολόγηση ενός αριθμού οχημάτων για να εξυπηρετήσουν ένα σύνολο πελατών και τον καθορισμό της ταχύτητας τους σε κάθε τμήμα της διαδρομής έτσι ώστε να ελαχιστοποιηθεί μία συνάρτηση που αποτελείται από τα κόστη καυσίμων, εκπομπών και οδηγών.

4.2 Μοντελοποίηση

Το PRP καθορίζεται σε ένα πλήρες προσανατολισμένο γράφημα G = (N, A) όπου N = {0,..., η} είναι ένα σύνολο κόμβων, 0 είναι η αποθήκη και A = {(i , j) : i, j ∈ N και 1 ≠ j} είναι το σύνολο των τόξων. Η απόσταση από το i στο j υποδηλώνεται ως dij. Ενός καθορισμένου μεγέθους στόλος οχημάτων που δηλώνεται ως K = {1,..., m} είναι διαθέσιμος και κάθε όχημα έχει χωρητικότητα Q. Το σύνολο N0 = N \ {0} είναι το σύνολο πελατών και κάθε πελάτης i Ε N0 έχει μία μη αρνητική ζήτηση qi, καθώς και ένα χρονικό διάστημα [ai, bi] για εξυπηρέτηση. Ενωρίτερες αφίξεις επιτρέπονται αλλά το όχημα πρέπει να περιμένει μέχρι το χρόνο ai πριν ξεκινήσει η εξυπηρέτηση. Ο χρόνος εξυπηρέτησης του πελάτη i δηλώνεται ως ti.

Το PRP στηρίζεται σε ένα μοντέλο που λαμβάνει υπόψη τις εκπομπές CO2 το οποίο είναι ένα μοντέλο που υπολογίζει την κατανάλωση καυσίμου για μία δεδομένη χρονική στιγμή. Σύμφωνα με το μοντέλο αυτό, ο ρυθμός καυσίμου (fuel rate) δίνεται από:

$$FR = \xi*(k*N*V+ P/\eta) / \kappa \qquad (4)$$

όπου ξ είναι ο ρυθμός καυσίμου προς αέρα, k είναι ο συντελεστής τριβής της μηχανής, N είναι η ταχύτητα της μηχανής, V είναι το εκτόπισμα της μηχανής και η και κ είναι σταθερές. P είναι η έξοδος στης δύναμης της μηχανής (σε kilowatt) και υπολογίζεται ως:

$$P = Ptrack/\eta tf + Pacc \qquad (5)$$

όπου ηtf είναι η ικανότητα οδήγησης του οχήματος και Pacc είναι η δύναμη της μηχανής που σχετίζεται με τις απώλειες της μηχανής και την λειτουργία εξαρτημάτων του οχήματος όπως ο κλιματισμός. Pacc θεωρείται ίσο με 0. Η παράμετρος Ptrack είναι οι συνολικές απαιτήσεις της ελκτικής δύναμης (σε kilowatt) που υπάρχει στους τροχούς:

$$Ptrack = (M*\tau + M*g*\sin\theta + 0,5*Cd*\rho*A*u^2 + M*g*Cr*\cos\theta)\ u/1000 \qquad (6)$$

όπου M είναι το συνολικό βάρος του οχήματος (σε kilogram), u είναι η ταχύτητα του οχήματος (σε meter/second), τ είναι η επιτάχυνση (σε meter/(second)²), θ είναι η γωνία του δρόμου, g είναι η σταθερά της βαρύτητας, Cd και Cr είναι η σταθερά της αεροδυναμικής αντίστασης και η σταθερά της αντίστασης κύλισης, αντίστοιχα. Τέλος, ρ είναι η πυκνότητα του αέρα και A είναι η μετωπική επιφάνεια αέρα του οχήματος.

Για ένα δεδομένο τόξο (i, j) μήκους d, έστω u είναι η ταχύτητα του οχήματος που διασχίζει το τόξο. Αν όλες οι μεταβλητές στο FR εκτός από την ταχύτητα του οχήματος u παραμένουν σταθερές στο τόξο (i, j), η κατανάλωση καυσίμου (στο L) σε αυτό το τόξο μπορεί να υπολογιστεί ως:

$$F(u) = k*N*V*\lambda*\ d/u \qquad (7)$$

$$+P*\lambda*\gamma*\ d/u \qquad (8)$$

όπου λ = ξ/κ*ψ και γ = 1/1000*ηtf*η είναι σταθερές και ψ είναι ο συντελεστής μετατροπής του καυσίμου από gram/second σε lter/second. Επιπλέον, έστω M είναι το φορτίο που μεταφέρεται μεταξύ των κόμβων i και j. Πιο συγκεκριμένα, M = w + f, όπου w είναι το βάρος του άδειου οχήματος και f είναι το φορτίο του οχήματος. Έστω α = τ + g*sinθ + g*Cr*cosθ είναι η συγκεκριμένη σταθερά του οχήματος-τόξου και β = 0,5*Cd*ρ*A είναι μία συγκεκριμένη σταθερά του οχήματος. Παραλείπουμε τους δείκτες (i,j) στις μεταβλητές υ, d, f, και α για να απλοποιήσουμε την παρουσίαση. Τότε το F(u) μπορεί να ξαναγραφεί ως:

$$F(u) = \lambda*(k*N*V + w*\gamma*\alpha*u + \gamma*\alpha*f*u + \beta*\gamma*u^2)*\ d/u \qquad (9)$$

Στη συνέχεια παρουσιάζεται η μορφοποίηση του προβλήματος. Το μοντέλο χρησιμοποιεί μια συνάρτηση ταχύτητας που καθορίζεται από R μη-μειούμενα επίπεδα ταχύτητας $\bar{u}^r$(r = 1,..., R). Οι δυαδικές μεταβλητές xij παίρνουν την τιμή 1 αν και μόνο αν το τόξο (i, j) εμφανίζεται στη λύση. Οι συνεχείς μεταβλητές fij αντιστοιχούν στη συνολική ποσότητα ροής σε κάθε τόξο (i, j) Ε A. Οι συνεχείς μεταβλητές yj, αντιστοιχούν στη χρονική στιγμή που η εξυπηρέτηση στον κόμβο j ξεκινάει. Επιπλέον, οι μεταβλητές sj αντιστοιχούν στο συνολικό χρόνο που ένα όχημα περνάει από κάποιους κόμβους που του έχουν ανατεθεί, συμπεριλαμβανομένου και του κόμβου j πριν επιστρέψει στην αποθήκη. Τέλος, οι δυαδικές μεταβλητές z^rij δείχνουν αν (ή όχι) το τόξο (i, j) διασχίζεται με ταχύτητα r.

$$\min \quad \sum_{(i,j)\in A} k * N * V * \lambda * d_{ij} * \sum_{r=1}^{R} z_{ij}^{r} * \bar{u}^{r} \tag{10}$$

$$+ \sum_{(i,j)\in A} w * \gamma * \lambda * a_{ij} * d_{ij} * x_{ij} \tag{11}$$

$$+ \sum_{(i,j)\in A} \gamma * \lambda * a_{ij} * d_{ij} * f_{ij} \tag{12}$$

$$+ \sum_{(i,j)\in A} \beta * \gamma * \lambda * d_{ij} * \sum_{r=1}^{R} z_{ij}^{r} * (\bar{u}^{r})^{2} \tag{13}$$

$$+ \sum_{j\in N_0} f_d * s_j \tag{14}$$

Υπό

$$\sum_{j\in N} x_{0j} = m \tag{15}$$

$$\sum_{j\in N} x_{ij} = 1 \ , \forall\, i \in N_0 \tag{16}$$

$$\sum_{i\in N} x_{ij} = 1 \ , \forall\, j \in N_0 \tag{17}$$

$$\sum_{j\in N} f_{ji} - \sum_{j\in N} f_{ij} = q_i \ , \forall\ i \in N_0 \tag{18}$$

$$q_j * x_{ij} \leq f_{ij} \leq (Q - q_i) * x_{ij}, \forall\, (i,j) \in A \tag{19}$$

$$y_i - y_j + t_i + \sum_{r\in R} \frac{d_{ij} * z_{ij}^{r}}{\bar{u}^{r}} \leq K_{ij} * \left(1 - x_{ij}\right), \forall\, i \in N,\ j \in N_0 , i \neq j \tag{20}$$

$$a_i \leq y_i \leq b_i, \forall\, i \in N_0 \tag{21}$$

$$y_j + t_j - s_j + \sum_{r\in R} \frac{d_{j0} * z_{j0}^{r}}{\bar{u}^{r}} \leq L * \left(1 - x_{j0}\right), \forall\, j \in N_0 \tag{22}$$

$$\sum_{r=1}^{R} z_{ij}^{r} = x_{ij} \ , \forall\, (i,j) \in A \tag{23}$$

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

$$f_{ij} \geq 0, \forall\, (i,j) \in A \tag{25}$$

$$y_i \geq 0, \forall\ i \in N_0 \tag{26}$$

$$z_{ij}^{r} \in \{0,1\}, \forall\, (i,j) \in A, \mathrm{r} = 1, \dots, \mathrm{R} \tag{27}$$

Η αντικειμενική συνάρτηση (10-13) έχουν προέλθει από την εξίσωση (9). Οι εξισώσεις (11) και (12) υπολογίζουν το κόστος που προέρχεται από το απόβαρο και το φορτίο του οχήματος αντίστοιχα. Τέλος, ο όρος (14) υπολογίζει το μισθολογικό κόστος των οδηγών. Οι περιορισμοί (15) καθορίζουν ότι κάθε όχημα θα φύγει από την αποθήκη. Οι περιορισμοί (16) και (17) καθορίζουν ότι κάθε πελάτης εξυπηρετείται από κάποιο όχημα μόνο μια φορά. Οι περιορισμοί (18) και (19) είναι οι περιορισμοί διατήρησης της ροής στα τόξα. Οι περιορισμοί (20 - 22), όπου $K_{ij} = \max\{0, b_i + s_i + d_{ij}/l_{ij} – \alpha_j)$, και το L είναι ένας μεγάλος αριθμός, αντιστοιχούν στους χρονικούς περιορισμούς του οχήματος. Οι περιορισμοί (23) εξασφαλίζουν ότι μόνο ένα επίπεδο ταχύτητας επιλέγεται για κάθε τόξο, οπότε σε αυτή την περίπτωση ισχύει ότι $z^r_{ij} = 1$ εάν $x_{ij} = 1$.

5. Διαδικασία Επίλυσης του Προβλήματος PRP

5.1 Δεδομένα

Για την λύση του συγκεκριμένου προβλήματος ήταν απαραίτητα κάποια δεδομένα, ενώ για να εξασφαλιστεί η προσαρμοστικότητα της μεθοδολογίας επίλυσης του προβλήματος για διάφορα σετ δεδομένων, τα ίδια εισάγονταν από ένα αρχείο txt, η μορφή του οποίου παρουσιάζεται στην εικόνα 5.1. Τα σετ δεδομένων πέρα από τις διαφορές που είχαν στον αριθμό των πελατών, παρουσίαζαν διαφορές και στον τρόπο που επιλέγονταν τα χρονικά παράθυρα του κάθε πελάτη. Στα σετ δεδομένων τύπου B η τιμή των χρονικών παραθύρων επιλέγονταν τυχαία με την ομοιόμορφη κατανομή σε ένα διάστημα 2.000 – 5.000 δευτερολέπτων, ενώ στα δεδομένα τύπου C το διάστημα αυτό ήταν από 2.000 – 15.000 δευτερόλεπτα.

```
10

6350    3650

20      90

0       41150   25680   54200   95380   15910   88960   74120   26010   88181   66070
40660   0       51980   32800   99870   42210   75660   63880   24350   72070   26250
25010   51780   0       61520   74050   12890   69270   52590   42910   73400   76700
54270   32750   61560   0       77030   51930   42930   31920   49480   39500   29500
94930   100030  74070   76930   0       81260   55600   46100   111960  61700   106350
15830   42600   12880   52340   81050   0       78000   61320   33730   82130   67520
88751   75700   69300   43030   55210   78040   0       17200   90550   6520    68800
73340   63440   52480   31830   46430   61220   17130   0       75520   21260   61250
25990   24350   43780   49530   111730  34010   90550   75740   0       88960   48920
88411   71740   73420   39430   61390   82160   6550    21320   88830   0       64010
65440   26250   76760   29330   106070  66990   68760   61180   48920   64080   0

0                                       Kingston_upon_Hull      0       0       32400   0

1                                              Pocklington      721     16298   19703   1442
2                                                   Brough      814     22946   26858   1628
3                                                    Selby      620     8674    12807   1240
4                                                 Boughton      311     19460   21987   622
5                                       Barton_upon_Humber      167     21211   23649   334
6                                                Darfield       513     7968    10968   1026
7                                                  Bentley      568     22076   25101   1136
8                                                   Watton      763     2560    4905    1526
9                                                 Cudworth      558     19874   23977   1116
10                                                   Haxby      636     15638   18155   1272
```

Εικόνα 5.1 *Παράδειγμα Δεδομένων για 10 κόμβους*

Αρχικά ορίστηκε ο αριθμός των πελατών που έπρεπε να εξυπηρετηθούν. Στην συνέχεια έγινε οριστικοποίηση σε κάποια από τα δεδομένα του οχήματος που θα χρησιμοποιούταν. Πιο αναλυτικά ως "6350" ορίστηκε το καθαρό βάρος του οχήματος σε κιλά, "3650" το μέγιστο επιτρεπτό φορτίο που μπορεί να μεταφέρει σε κιλά, ενώ "20" και "90" είναι η ελάχιστη και μέγιστη ταχύτητα αντίστοιχα (σε km/h) που μπορεί να αναπτύξει το όχημα. Αξίζει να σημειωθεί ότι στην συγκεκριμένη έρευνα χρησιμοποιήθηκε μόνο ένας τύπος οχήματος οπότε αυτά τα δεδομένα παρέμειναν σταθερά για όλα τα σετ δεδομένων.

Τα υπόλοιπα δεδομένα του σετ αφορούσαν τους πελάτες - κόμβους. Αρχικά σε έναν πίνακα δίνονταν οι αποστάσεις (σε μέτρα) μεταξύ των πελατών, τόσο μεταξύ τους όσο και μεταξύ της αποθήκης. Τέλος σε έναν ακόμα πίνακα δίνονταν δεδομένα που όριζαν, τον αριθμό του κάθε κόμβου (πάντα ως 0 οριζόταν η αποθήκη), το όνομα της κάθε πόλης (κάθε πελάτης - κόμβος ήταν και μία πόλη), την ζήτηση σε κιλά του κάθε κόμβου, τον νωρίτερο χρόνο άφιξης (σε δευτερόλεπτα), τον αργότερο χρόνο άφιξης (σε δευτερόλεπτα), και τον χρόνο εξυπηρέτησης (σε δευτερόλεπτα). Όσο αφορά την αποθήκη, ως αργότερος χρόνος ορίζεται η συνολική ώρα που μπορεί να απασχολήσει η εταιρεία τον οδηγό του οχήματος.

Για την υλοποίηση των αλγορίθμων που χρησιμοποιήθηκαν στην συγκεκριμένη έρευνα έγιναν οι εξής παραδοχές. Ως ταχύτητα χρησιμοποιήθηκε ο μέσος όρος του δοσμένου εύρους, καθώς θεωρήθηκε ότι για να διανύσει κάποια απόσταση το όχημα η ταχύτητα του δεν θα είναι σταθερή καθ' όλη την διάρκεια της απόστασης αυτής. Επομένως ο μέσος όρος ήταν ο καλύτερος τρόπος για να μπορεί να βρεθεί μια τιμή για την ταχύτητα του οχήματος στο πρόβλημα. Τέλος έγινε και μετατροπή της ταχύτητας του οχήματος από km/h σε m/s για να μπορούν να είναι εφικτοί οι υπολογισμοί.

Για να γίνει εφικτή η επίλυση του Pollution Routing Problem (PRP) χρειάστηκαν επιπλέον και ορισμένα δεδομένα που αφορούσαν το όχημα που χρησιμοποιήθηκε. Τα δεδομένα αυτά ήταν αναγκαία ώστε να μπορεί να υπολογιστεί το συνολικό κόστος. Στον Πίνακα 5.1 παρουσιάζονται τα δεδομένα αυτά με τις αντίστοιχες τιμές τους και την περιγραφή της κάθε μεταβλητής.

ξ	Fuel-to-air-mass ratio	1
k	Engine friction factor (kJ/rev/l)	0.2
N	Engine speed (rev/s)	33
V	Engine displacement (l)	5
g	Gravitational constant (m/s2)	9.81
Cd m	Coefficient of aerodynamic drag	0.7
r	Air density (kg/m3)	12.041
A	Frontal surface area (m2)	3.912
Cr	Coefficient of rolling resistance	0.01
η tf	Vehicle drive train efficiency	0.4
η	Efficiency parameter for diesel engines	0.9
C	Fuel and CO2 emission cost per liter (£)	1.4
κ	Heating value of a typical diesel fuel (kJ/g)	44
ψ	Conversion factor (g/second to liter/s)	737
λ	*λ* = ξ/κψ	3.08E−05
γ	g = 1/1000 *htfh*	0.002777
α	*α* = τ + *g* sin θ + *gCr* cos θ	0.098100
β	*β* = 0.5*CdrA*	1.648.653
Vc	Variable cost (£/m)	0.0002

Πίνακας 5.1 *Δεδομένα Οχήματος*

5.2 Μεθοδολογία Επίλυσης

Για την επίλυση του Pollution Routing Problem υλοποιήθηκε και εκτελέστηκε κώδικας σε Python που είχε ως βάση τρεις αλγόριθμους. Αυτοί ήταν ο αλγόριθμος του πλησιέστερου γείτονα, για την εύρεση μίας αρχικής λύσης, ο αλγόριθμος 2-opt 1-0 relocate και ο αλγόριθμος προσομοιωμένης ανόπτησης. Στο κεφάλαιο αυτό, επομένως, θα γίνει η ανάλυση της λειτουργίας του κάθε αλγόριθμου στον κώδικα.

5.2.1 Αλγόριθμος Εύρεσης Αρχικής Λύσης

Ως αλγόριθμος εύρεσης αρχικής λύσης επιλέχθηκε ο αλγόριθμος του κοντινότερου γείτονα. Ο αλγόριθμος αυτός παρόλη την απλότητα της λειτουργίας του είναι αρκετός για να δώσει μία αρχική λύση ώστε αυτή μετά να τροφοδοτηθεί στον μεθευρετικό αλγόριθμο. Τα βήματα που ακολουθήθηκαν ήταν τα εξής. Ως αρχικός κόμβος κάθε διαδρομής ορίζονταν πάντα η αποθήκη (δηλαδή ο κόμβος 0). Στην συνέχεια επιλεγόταν ο κοντινότερος κόμβος και ελέγχονταν αν παραβιάζει κάποιον από τους περιορισμούς. Αν δεν τους παραβίαζε τότε προστίθονταν στην διαδρομή και οριζόταν ως τρέχον κόμβος. Διαφορετικά η διαδρομή έκλεινε, επιστρέφοντας πίσω στην αποθήκη. Η διαδικασία αυτή συνεχιζόταν μέχρις ότου όλοι οι κόμβοι να έχουν εξυπηρετηθεί, δηλαδή να έχουν προστεθεί σε κάποια διαδρομή.

5.2.2 Αλγόριθμος Τοπικής Αναζήτησης

Έχοντας την αρχική λύση από τον αλγόριθμο του πλησιέστερου γείτονα, έγινε προσπάθεια βελτίωσης της με την χρήση μεθόδων τοπικής αναζήτησης. Οι μέθοδοι αυτοί που χρησιμοποιήθηκαν στην συγκεκριμένη έρευνα ήταν η 2-opt και 1-0 relocate.

1-0 relocate

Σε αυτή την μέθοδο επιλεγόταν ένας κόμβος από μια διαδρομή και επανατοποθετούνταν σε μία άλλη με στόχο την μείωση του συνολικού κόστους. Ο αλγόριθμος εξέταζε ποια διαδρομή περιείχε τους λιγότερους κόμβους για να επιλέξει κάποιον κόμβο τυχαία από εκεί, ενώ σε περίπτωση που οι διαδρομές αυτές ήταν παραπάνω από μία ο αλγόριθμος επέλεγε εκείνη με το μεγαλύτερο κόστος. Ο κόμβος αφού επανατοποθετούνταν σε μία διαδρομή, γινόταν έλεγχος τόσο για το αν παραβιάζει κάποιον/κάποιους από τους περιορισμούς αλλά και για το αν το συνολικό κόστος βελτιωνόταν. Αν δεν παραβίαζε κάποιον περιορισμό και βελτίωνε το κόστος ο κόμβος αυτός παρέμενε στην διαδρομή αυτή, ενώ σε διαφορετική περίπτωση τοποθετούνταν πίσω στην διαδρομή από την οποία είχε αρχικά διαγραφεί.

2-opt

Σε αυτή την μέθοδο, αφού εκτελούνταν η μέθοδος 1-0 relocate για έναν αριθμό επαναλήψεων, επιλεγόταν κάθε διαδρομή , από το αποτέλεσμα που προέκυπτε, που περιείχε πάνω από δύο κόμβους, πέραν της αποθήκης, και γινόταν αλλαγή στην θέση δύο κόμβων μέσα στην διαδρομή αυτή. Αν η αλλαγή αυτή ήταν εφικτή και μείωνε το συνολικό κόστος τότε γινόταν αποδεκτή και ο αλγόριθμος προχωρούσε στην προσπάθεια αλλαγής θέσης των επόμενων κόμβων. Διαφορετικά επέστρεφε τους κόμβους που είχε γίνει η αλλαγή στην αρχική τους θέση και προχωρούσε πάλι στην προσπάθεια αλλαγής θέσης επόμενων κόμβων.

5.2.3 Αλγόριθμος Προσομοιωμένης Ανόπτησης

Σε αυτή την μέθοδο αρχικά δημιουργήθηκε, με την βοήθεια του αλγορίθμου του κοντινότερου γείτονα, μία αρχική λύση η οποία στην συνέχεια βελτιωνόταν κάπως με την μέθοδο 2-opt και 1-0 relocate. Την βελτιωμένη αυτή λύση την χρησιμοποιούσε τελικά η μέθοδος ως αρχική λύση για την μέθοδο της προσομοιωμένης ανόπτησης. Αρχικά ορίστηκε μία αρχική θερμοκρασία (T = 10) και ένας ρυθμός ψύξης (cooling rate = 0.99), ο οποίος σε κάθε επανάληψη έριχνε την θερμοκρασία κατά ένα ποσοστό.

Στην συνέχεια δημιουργήθηκε μία δομή επανάληψης στην οποία μειωνόταν σε κάθε βήμα της η θερμοκρασία (βάση του ρυθμού μείωσης) και είχε ως κριτήρια τερματισμού είτε την ελάχιστη θερμοκρασία είτε το δοσμένο όριο επαναλήψεων. Δηλαδή η δομή επανάληψης τερμάτιζε είτε μόλις η θερμοκρασία έφτανε τους 0.5 βαθμούς είτε μόλις είχαν ολοκληρωθεί 500 επαναλήψεις. Μέσα σε αυτή την δομή επανάληψης υπήρχε μία δεύτερη δομή επανάληψης, η οποία εκτελούνταν 250 φορές, και στην οποία γινόντουσαν οι εξής εντολές. Αρχικά γινόταν μία τυχαία αλλαγή, η οποία όμως δεν παραβίαζε τους περιορισμούς. Πιο συγκεκριμένα διαλέγονταν τυχαία δύο διαδρομές και δύο κόμβοι, ένας από κάθε τυχαία διαδρομή, και γινόταν αλλαγή στους δύο αυτούς κόμβους. Έπειτα αν η αλλαγή αυτή επέφερε καλύτερο αποτέλεσμα στην τιμή του συνολικού κόστους τότε γινόταν η καλύτερη λύση. Διαφορετικά, αν δηλαδή το κόστος ήταν ίδιο ή μεγαλύτερο, τότε με βάση την κανονική κατανομή επιλέγονταν τυχαία μία τιμή και αν αυτή ήταν μικρότερη από το αποτέλεσμα $e^{[(\text{βέλτιστο κόστος – κόστος της τυχαίας αλλαγής})/T]}$, όπου T είναι η θερμοκρασία σε εκείνη στην επανάληψη, τότε η λύση γινόταν αποδεκτή, δηλαδή γινόταν η καλύτερη λύση. Στην αντίθετη περίπτωση όμως η λύση δεν γινόταν αποδεκτή και ο αλγόριθμος προχωρούσε στην επόμενη επανάληψη, δηλαδή στην επόμενη τυχαία αλλαγή.

Μόλις ολοκληρωνόντουσαν και οι δύο δομές επανάληψης, στην τελική λύση που προέκυπτε γινόταν προσπάθεια περαιτέρω βελτίωσης με την χρήση πάλι των μεθόδων 2-opt και 1-0 relocate.

6. Αποτελέσματα

Σε αυτό το κεφάλαιο θα γίνει η παρουσίαση των αποτελεσμάτων που προέκυψαν από την υλοποίηση του κώδικα που δημιουργήθηκε στο περιβάλλον της python, για την επίλυση του προβλήματος δρομολόγησης οχημάτων με σκοπό την μείωση των ρύπων.

Η υλοποίηση πραγματοποιήθηκε σε 9 παραδείγματα αναφοράς, όπου κάθε παράδειγμα περιείχε διαφορετικό αριθμό κόμβων. Τα αποτελέσματα αποσκοπούν στο να φανεί το ποσοστό μείωσης του κόστους της αρχικής λύσης, αλλά και ο χρόνος εκτέλεσης που χρειάστηκε για να το πετύχει αυτό. Ο αλγόριθμος για κάθε σετ δεδομένων εκτελέστηκε 5 φορές με σκοπό την άντληση ενός μέσου όρου για κάθε σετ.

6.1 Αποτελέσματα Για Διάφορα Σετ Δεδομένων

Αποτελέσματα για 10 κόμβους

	Αριθμός Οχημάτων	**Κόστος**
Nearest Neighbor	5	409,89
2-opt After Nearest Neighbor	4	363,37
Simulated Annealing	4	326,928
2-opt After Simulated Annealing	4	326,928
Ποσοστό Μείωσης του Κόστους		20,24%

Πίνακας 6.1 *Αποτελέσματα για 10 κόμβους*

Τα αποτελέσματα για το σετ δεδομένων με τους 10 κόμβους παρουσιάζουν ένα ικανοποιητικό αποτέλεσμα. Το αρχικό κόστος μειώθηκε κατά 20,24% ενώ παράλληλα μειώθηκε και ο απαιτούμενος αριθμός οχημάτων από 5 σε 4. Παρατηρείται όμως ότι μετά τον αλγόριθμο της προσομοιωμένης ανόπτησης δεν μπορεί να βελτιωθεί άλλο η λύση κάτι που πιθανότατα οφείλεται στον μικρό αριθμό κόμβων.

Αποτελέσματα για 15 κόμβους

	Αριθμός Οχημάτων	Κόστος
Nearest Neighbor	9	641,76
2-opt After Nearest Neighbor	7	528,52
Simulated Annealing	7	528,52
2-opt After Simulated Annealing	7	528,52
Ποσοστό Μείωσης του Κόστους		17,65%

Πίνακας 6.2 *Αποτελέσματα για 15 κόμβους*

Τα αποτελέσματα για το σετ δεδομένων με τους 15 κόμβους παρουσιάζουν ένα επαρκώς ικανοποιητικό αποτέλεσμα. Το αρχικό κόστος μειώθηκε κατά 17,65% ενώ παράλληλα μειώθηκε και ο απαιτούμενος αριθμός οχημάτων από 9 σε 7. Παρατηρείται όμως ότι μετά τον αλγόριθμο της προσομοιωμένης ανόπτησης, η λύση πάλι δεν μπορεί να βελτιωθεί περαιτέρω κάτι που πιθανότατα οφείλεται στον μικρό αριθμό κόμβων.

Αποτελέσματα για 20 κόμβους

	Αριθμός Οχημάτων	Κόστος
Nearest Neighbor	7	554,86
2-opt After Nearest Neighbor	7	554,62
Simulated Annealing	7	546,994
2-opt After Simulated Annealing	7	546,994
Ποσοστό Μείωσης του Κόστους		1,42%

Πίνακας 6.3 *Αποτελέσματα για 20 κόμβους*

Τα αποτελέσματα για το σετ δεδομένων με τους 20 κόμβους παρουσιάζουν ένα ιδιαίτερο αποτέλεσμα. Το αρχικό κόστος μειώθηκε μόλις κατά 1,42% ενώ παράλληλα ο απαιτούμενος αριθμός οχημάτων παρέμεινε ο ίδιος. Αν και αυτό εκ πρώτης όψεως φαίνεται να σημαίνει ότι το αποτέλεσμα δεν είναι καθόλου ικανοποιητικό, αυτό όμως αλλάζει αν γίνει σύγκριση στην αρχική λύση με το σετ δεδομένων με τους 15 κόμβους. Αυτό που παρατηρείται είναι ότι σε αυτό το σετ δεδομένων η αρχική λύση είναι 554,86 έναντι 641,76 που είχε το σετ με τους 15 κόμβους. Μια διαφορά της τάξης του 13.54% από την αρχική λύση και 4,75% από την τελική. Επομένως το αποτέλεσμα εξακολουθεί να είναι ικανοποιητικό, απλά το μικρό ποσοστό μείωσης οφείλεται στο γεγονός ότι ο αλγόριθμος του κοντινότερου γείτονα προσεγγίζει αρκετά την τελική λύση, δηλαδή το αποτέλεσμα που δίνει είναι εξαιρετικά καλό.

Παρατηρείται επιπλέον ότι μετά τον αλγόριθμο της προσομοιωμένης ανόπτησης, η λύση πάλι δεν μπορεί να βελτιωθεί περαιτέρω κάτι που πιθανότατα σημαίνει ότι ο αριθμός των κόμβων είναι ακόμα μικρός για να μπορούν να γίνουν εφικτές αλλαγές μετά την προσομοιωμένη ανόπτηση.

Αποτελέσματα για 25 κόμβους

	Αριθμός Οχημάτων	**Κόστος**
Nearest Neighbor	14	1005,79
2-opt After Nearest Neighbor	8	726,73
Simulated Annealing	8	580,114
2-opt After Simulated Annealing	7	574,928
Ποσοστό Μείωσης του Κόστους		42,84%

Πίνακας 6.4 *Αποτελέσματα για 25 κόμβους*

Τα αποτελέσματα για το σετ δεδομένων με τους 25 κόμβους παρουσιάζουν ένα πολύ ικανοποιητικό αποτέλεσμα. Το αρχικό κόστος μειώθηκε κατά 42,84% ενώ παράλληλα μειώθηκε και ο απαιτούμενος αριθμός οχημάτων στο μισό, από 14 σε 7. Παρατηρείται επίσης ότι μετά τον αλγόριθμο της προσομοιωμένης ανόπτησης, η λύση βελτιώνεται και άλλο μειώνοντας έτσι τον αριθμό των απαιτούμενων οχημάτων από 8 σε 7 καθώς και το συνολικό κόστος κατά 0,9%. Αξίζει να σημειωθεί ότι αυτό το σετ δεδομένων είχε το μεγαλύτερο ποσοστό μείωσης του συνολικού κόστους.

Αποτελέσματα για 50 κόμβους

	Αριθμός Οχημάτων	**Κόστος**
Nearest Neighbor	26	1960,29
2-opt After Nearest Neighbor	16	1452,69
Simulated Annealing	16	1228,02
2-opt After Simulated Annealing	15	1210,52
Ποσοστό Μείωσης του Κόστους		38,25%

Πίνακας 6.5 *Αποτελέσματα για 50 κόμβους*

Τα αποτελέσματα για το σετ δεδομένων με τους 50 κόμβους παρουσιάζουν ένα πολύ ικανοποιητικό αποτέλεσμα. Το αρχικό κόστος μειώθηκε κατά 38,25% ενώ παράλληλα μειώθηκε και ο απαιτούμενος αριθμός οχημάτων από 26 σε 15. Παρατηρείται επίσης ότι μετά τον αλγόριθμο της προσομοιωμένης ανόπτησης, η λύση βελτιώνεται και άλλο μειώνοντας έτσι τον αριθμό των απαιτούμενων οχημάτων από 16 σε 15 καθώς και το συνολικό κόστος κατά 1,4%.

Αποτελέσματα για 75 κόμβους

	Αριθμός Οχημάτων	**Κόστος**
Nearest Neighbor	49	3649,85
2-opt After Nearest Neighbor	39	3046,98
Simulated Annealing	39	3046,98
2-opt After Simulated Annealing	29	2510,15
Ποσοστό Μείωσης του Κόστους		31,23%

Πίνακας 6.6 *Αποτελέσματα για 75 κόμβους*

Τα αποτελέσματα για το σετ δεδομένων με τους 75 κόμβους παρουσιάζουν ένα αρκετά ικανοποιητικό αποτέλεσμα. Το αρχικό κόστος μειώθηκε κατά 31,23% ενώ παράλληλα μειώθηκε και ο απαιτούμενος αριθμός οχημάτων από 49 σε 29. Παρατηρείται επίσης ότι μετά τον αλγόριθμο της προσομοιωμένης ανόπτησης, η λύση βελτιώνεται και άλλο μειώνοντας έτσι τον αριθμό των απαιτούμενων οχημάτων από 39 σε 29 καθώς και το συνολικό κόστος κατά 17,6%.

Αποτελέσματα για 100 κόμβους

	Αριθμός Οχημάτων	**Κόστος**
Nearest Neighbor	53	3817,93
2-opt After Nearest Neighbor	43	3284,24
Simulated Annealing	43	2871,11
2-opt After Simulated Annealing	33	2612,91
Ποσοστό Μείωσης του Κόστους		31,56%

Πίνακας 6.7 *Αποτελέσματα για 100 κόμβους*

Τα αποτελέσματα για το σετ δεδομένων με τους 100 κόμβους παρουσιάζουν ένα αρκετά ικανοποιητικό αποτέλεσμα. Το αρχικό κόστος μειώθηκε κατά 31,56% ενώ παράλληλα μειώθηκε και ο απαιτούμενος αριθμός οχημάτων από 53 σε 33. Παρατηρείται επίσης ότι μετά τον αλγόριθμο της προσομοιωμένης ανόπτησης, η λύση βελτιώνεται και άλλο μειώνοντας έτσι τον αριθμό των απαιτούμενων οχημάτων από 43 σε 33 καθώς και το συνολικό κόστος κατά 8,99%.

Αποτελέσματα για 150 κόμβους

	Αριθμός Οχημάτων	**Κόστος**
Nearest Neighbor	75	5171,18
2-opt After Nearest Neighbor	65	4539,02
Simulated Annealing	65	3863,63
2-opt After Simulated Annealing	55	3621,7
Ποσοστό Μείωσης του Κόστους		29,96%

Πίνακας 6.8 *Αποτελέσματα για 150 κόμβους*

Τα αποτελέσματα για το σετ δεδομένων με τους 150 κόμβους παρουσιάζουν ένα αρκετά ικανοποιητικό αποτέλεσμα. Το αρχικό κόστος μειώθηκε κατά 29,96% ενώ παράλληλα μειώθηκε και ο απαιτούμενος αριθμός οχημάτων από 75 σε 55. Παρατηρείται επίσης ότι μετά τον αλγόριθμο της προσομοιωμένης ανόπτησης, η λύση βελτιώνεται και άλλο μειώνοντας έτσι τον αριθμό των απαιτούμενων οχημάτων από 65 σε 55 καθώς και το συνολικό κόστος κατά 6,3%.

Αποτελέσματα για 200 κόμβους

	Αριθμός Οχημάτων	**Κόστος**
Nearest Neighbor	103	7031,65
2-opt After Nearest Neighbor	93	6419,9
Simulated Annealing	93	5527,4
2-opt After Simulated Annealing	83	5230,37
Ποσοστό Μείωσης του Κόστους		25,62%

Πίνακας 6.9 *Αποτελέσματα για 200 κόμβους*

Τα αποτελέσματα για το σετ δεδομένων με τους 200 κόμβους παρουσιάζουν ένα αρκετά ικανοποιητικό αποτέλεσμα. Το αρχικό κόστος μειώθηκε κατά 25,62% ενώ παράλληλα μειώθηκε και ο απαιτούμενος αριθμός οχημάτων από 103 σε 83. Παρατηρείται επίσης ότι μετά τον αλγόριθμο της προσομοιωμένης ανόπτησης, η λύση βελτιώνεται και άλλο μειώνοντας έτσι τον αριθμό των απαιτούμενων οχημάτων από 93 σε 83 καθώς και το συνολικό κόστος κατά 5,4%.

6.2 Συγκεντρωτικά Αποτελέσματα

Στο κεφάλαιο αυτό θα παρουσιαστούν τα συγκεντρωτικά αποτελέσματα από την εκτέλεση των αλγορίθμων για όλα τα σετ δεδομένων. Τα αποτελέσματα αυτά αφορούν τον χρόνο εκτέλεσης για κάθε σετ και το πόσο αυτός αυξήθηκε από το ένα σετ στο άλλο, την αρχική λύση καθώς και το ποσοστό μείωσής της.

Χρόνος Εκτέλεσης

	10	**15**	**20**	**25**	**50**	**75**	**100**	**150**	**200**
ΧΡΟΝΟΣ ΕΚΤΕΛΕΣΗΣ	9,15717	14,13112	19,35864	26,63976	85,64933	194,89821	330,67228	714,56262	1192,94264

Πίνακας 6.10 *Χρόνος εκτέλεσης αλγορίθμου για κάθε σετ δεδομένων*

Ο χρόνος εκτέλεσης (μετρούμενος σε δευτερόλεπτα) για κάθε σετ δεδομένων παρουσιάζει μια αυξητική τάση. Όπως φαίνεται και στο γράφημα 6.1, η τάση αυτή είναι εκθετικής μορφής. Αυτό σημαίνει ότι όσο ο αριθμός των κόμβων είναι μικρός, μια αύξηση στον αριθμό αυτό επιφέρει μικρή αύξηση στον χρόνο εκτέλεσης. Όμως σε μεγάλους αριθμούς κόμβων, μια αύξηση στην τιμή αυτή επιφέρει μια αρκετά μεγάλη αύξηση στην τιμή του χρόνου εκτέλεσης. Αυτό μπορεί να παρατηρηθεί και στον πίνακα 6.10. Αυξάνοντας τους κόμβους από 10 σε 50 ο χρόνος εκτέλεσης αυξάνεται κατά 76,49216 δευτερόλεπτα, μια αύξηση του 835,33%. Όμως αυξάνοντας από τους 50 στους 100 κόμβους, ο χρόνος εκτέλεσης αυξάνεται κατά 245,02295 δευτερόλεπτα δηλαδή μια αύξηση της τάξης του 286.08% περίπου.

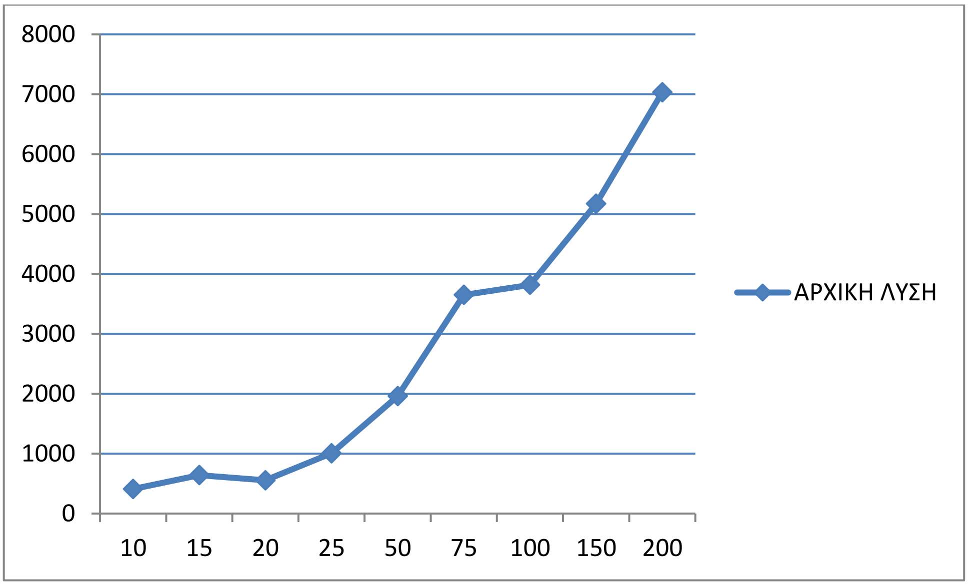
Γράφημα 6.1 *Χρόνος εκτέλεσης αλγορίθμου για κάθε σετ δεδομένων*

Αρχική Λύση

	10	15	20	25	50	75	100	150	200
ΑΡΧΙΚΗ ΛΥΣΗ	409,89	641,76	554,86	1005,79	1960,29	3649,85	3817,93	5171,18	7031,65

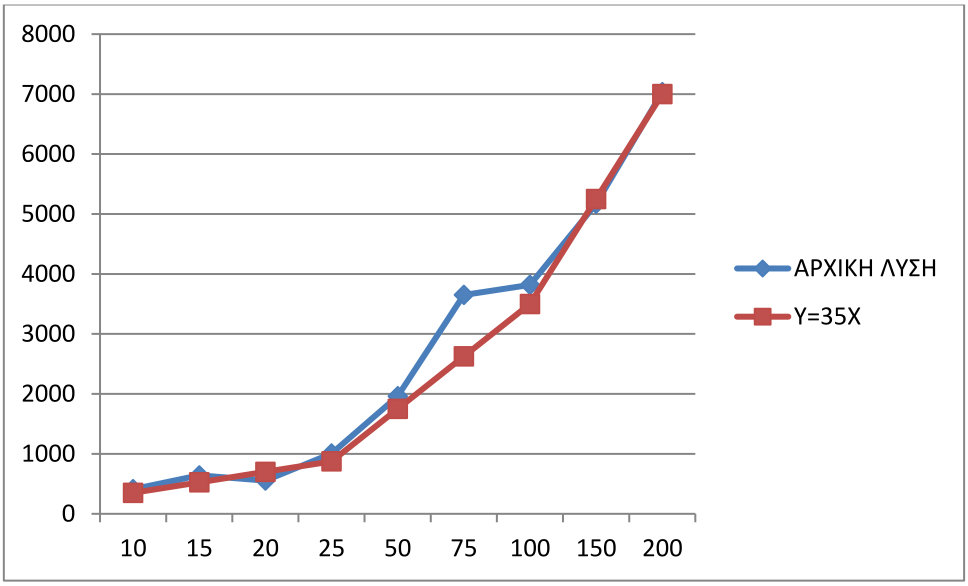
Πίνακας 6.11 *Τιμή της αρχικής λύσης για κάθε σετ δεδομένων*

Η αρχική λύση για κάθε σετ δεδομένων είναι το αποτέλεσμα του αλγορίθμου του κοντινότερου γείτονα (Nearest Neighbor Algorithm). Όπως φαίνεται και στον πίνακα 6.11 κάθε αύξηση στον αριθμό των κόμβων συνεπάγεται και μεγαλύτερη τιμή στην αρχική λύση. Μοναδική εξαίρεση αποτελεί το σετ δεδομένων με τους 20 κόμβους όπου ο αλγόριθμος του κοντινότερου γείτονα δίνει αποτέλεσμα πολύ κοντά στο τελικό με αποτέλεσμα η λύση του να είναι μικρότερη από αυτή του σετ δεδομένων με τους 15 κόμβους.



Γράφημα 6.2 *Τιμή της αρχικής λύσης για κάθε σετ δεδομένων*

Η αυξητική αυτή τάση της αρχικής λύσης είναι σχετικά γραμμική. Παρόλο που στο γράφημα 6.2 η τάση αυτή φαίνεται να είναι εκθετική, η ψευδαίσθηση αυτή δημιουργείται καθώς οι δυο άξονες δεν είναι αναλογικοί μεταξύ τους. Αυτό μπορεί να φανεί καλύτερα όταν συγκριθεί η γραφική παράσταση της αρχικής λύσης με την ευθεία Y=35X στους ίδιους άξονες (Γράφημα 6.3).



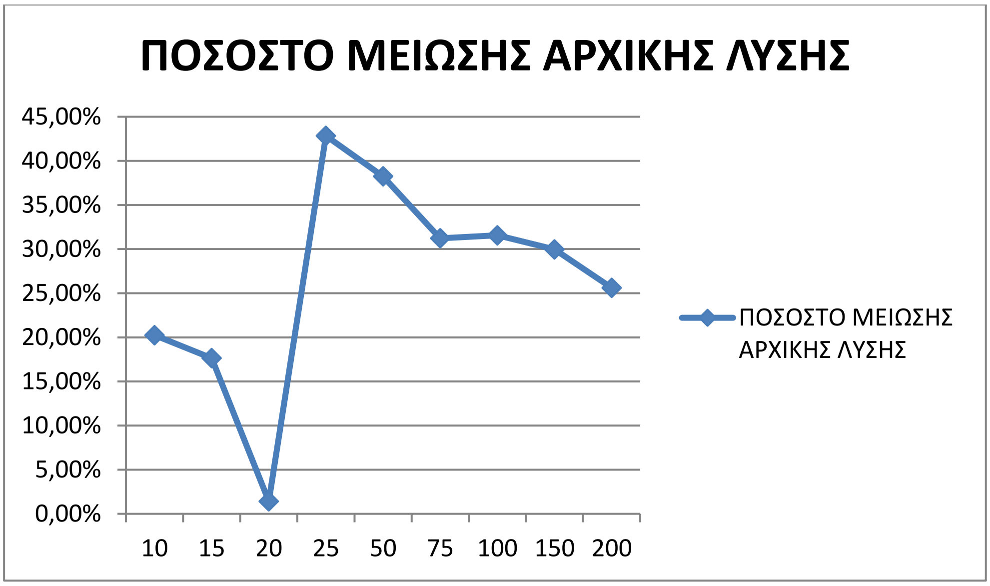
Γράφημα 6.3 *Σύγκριση της τιμής της αρχικής λύσης με την ευθεία Y=35X*

Ποσοστό Μείωσης της Αρχικής Λύσης

	10	**15**	**20**	**25**	**50**	**75**	**100**	**150**	**200**
ΠΟΣΟΣΤΟ ΜΕΙΩΣΗΣ ΑΡΧΙΚΗΣ ΛΥΣΗΣ	20,24 %	17,65 %	1,42 %	42,84 %	38,25 %	31,23 %	31,56 %	29,96 %	25,62 %

Πίνακας 6.12 *Ποσοστό μείωσης της αρχικής λύσης*

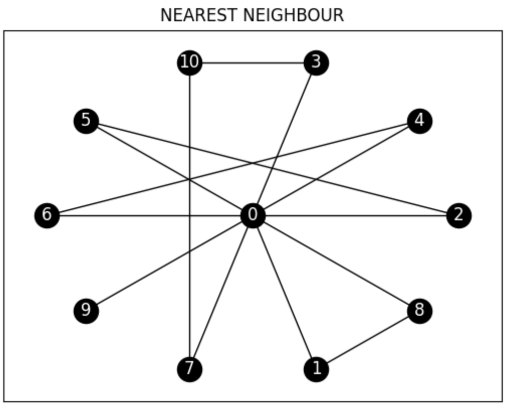
Το ποσοστό μείωσης της αρχικής λύσης εκφράζει την μείωση που παρουσίαζε η λύση της μεθόδου 2-opt, μετά την εφαρμογή της στον αλγόριθμο της προσομοιωμένης ανόπτησης, ως προς την λύση της μεθόδου του κοντινότερου γείτονα. Όπως φαίνεται στο γράφημα 6.3 αλλά και στον πίνακα 6.12, το μεγαλύτερο ποσοστό μείωσης παρουσιάζεται στο σετ δεδομένων με τους 25 κόμβους, όπου το ποσοστό μείωσης είναι σχεδόν 43%. Αντιθέτως το μικρότερο ποσοστό μείωσης παρουσιάζεται στο σετ δεδομένων με τους 20 κόμβους με ποσοστό μόλις 1,42%. Όμως, όπως εξηγήθηκε παραπάνω, ο μικρός αυτός αριθμός οφείλεται στην πολύ καλή προσέγγιση του αλγορίθμου του κοντινότερου γείτονα της τελικής λύσης και όχι στην ανεπάρκεια του προγράμματος να βελτιώσει την αρχική λύση. Κατά μέσο όρο η μείωση της αρχικής λύσης είναι της τάξης του 26,53%, μία αρκετά ικανοποιητική μείωση.



Γράφημα 6.3 *Ποσοστό μείωσης της αρχικής λύσης*

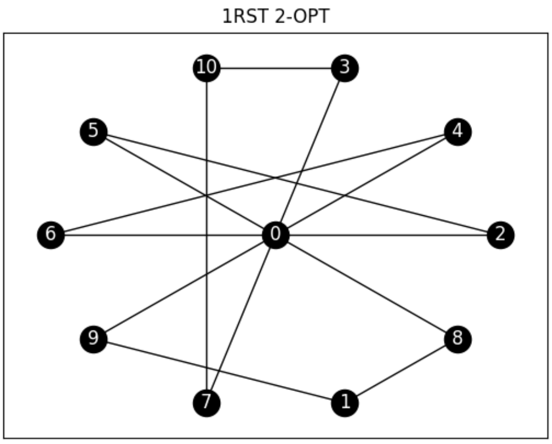
6.2.1 Γραφική Αναπαράσταση Αποτελεσμάτων

Στις εικόνες 6.1 έως 6.4 παρουσιάζονται οι γραφικές απεικονίσεις των αλγορίθμων που χρησιμοποιήθηκαν και που αναλύθηκαν στο κεφάλαιο 5. Οι αλγόριθμοι εφαρμόστηκαν στο σετ δεδομένων για τους 10 κόμβους ενώ πρέπει να σημειωθεί ότι θέσεις των κόμβων τοποθετήθηκαν τυχαία καθώς δεν υπήρχαν πληροφορίες, σε κανένα σετ δεδομένων, σχετικά με τις γεωγραφικές συντεταγμένες των κόμβων. Έτσι ο κόμβος 0, δηλαδή η αποθήκη, τοποθετήθηκε στο κέντρο ενώ οι υπόλοιποι κόμβοι στοιχήθηκαν ομοιόμορφα περιμετρικά του.



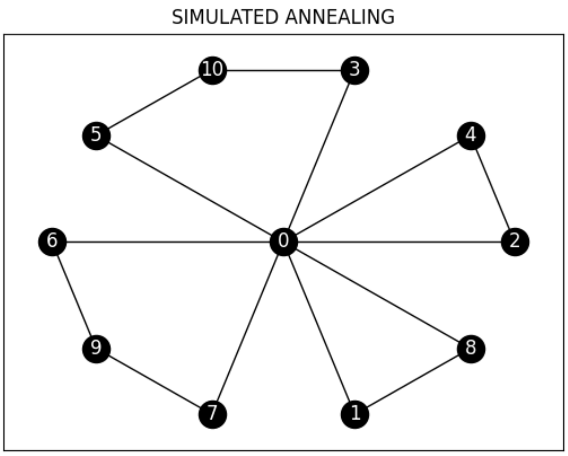
Εικόνα 6.1 *Γραφική απεικόνιση του αλγορίθμου Nearest Neighbor*

Όπως φαίνεται στην εικόνα 6.1 αν και ο αλγόριθμος του κοντινότερου γείτονα έχει δημιουργήσει διαδρομές που δεν φέρουν διακλάδωση μεταξύ της ίδιας διαδρομής, φέρουν όμως διακλάδωση μεταξύ τους ενώ παράλληλα έχει δημιουργήσει διαδρομή όπου περιέχει μονάχα έναν κόμβο. Αυτό αυξάνει το συνολικό κόστος και άρα η λύση αυτή επιδέχεται βελτίωση. Το αποτέλεσμα αυτό ήταν αναμενόμενο καθώς ο συγκεκριμένος αλγόριθμος χρησιμοποιείται μονάχα για την δημιουργία μίας αρχικής λύσης και όχι για την δημιουργία μίας βέλτιστης.



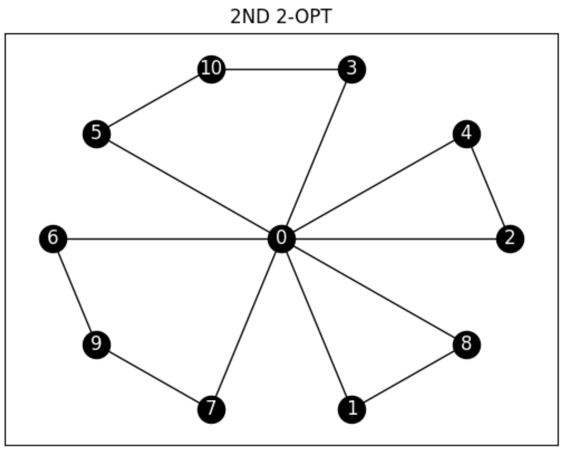
Εικόνα 6.2 *Γραφική απεικόνιση του αλγορίθμου 2-opt*

Την βελτίωση αυτή του κόστους θα προσπαθήσει να την κάνει ο αλγόριθμος 2-opt όπου όπως φαίνεται στην εικόνα 6.2 καταφέρνει να τοποθετήσει τον κόμβο 9 στην διαδρομή μαζί με τους κόμβους 8 και 1. Αυτό βελτιώνει το συνολικό κόστος, όμως η λύση επιδέχεται περεταίρω βελτίωση καθώς παραμένουν οι διακλαδώσεις μεταξύ των διαδρομών.



Εικόνα 6.3 *Γραφική απεικόνιση του αλγορίθμου Simulated Annealing*

Όπως φαίνεται στην εικόνα 6.3 ο αλγόριθμος της Προσομοιωμένης Ανόπτησης καταφέρνει να διορθώσει το πρόβλημα της διακλάδωσης των διαδρομών κάτι που βελτιώνει και άλλο το συνολικό κόστος, αν όχι να το κάνει βέλτιστο.



Εικόνα 6.4 *Γραφική απεικόνιση του αλγορίθμου 2-opt*

Η εφαρμογή του αλγορίθμου 2-opt στο αποτέλεσμα της Προσομοιωμένης Ανόπτησης φαίνεται ότι δεν μπορεί να βελτιώσει άλλο την λύση (εικόνα 6.4). Αυτό πιθανότατα συμβαίνει λόγω του μικρού αριθμού κόμβων που είχε το σετ καθώς παρατηρείται ότι μέχρι τους 20 κόμβους δεν μπορεί ο αλγόριθμος να βελτιώσει την λύση, όμως από τους 25 και μετά ο ίδιος την βελτιώνει σε κάθε σετ.

7. Συμπεράσματα

Συμπερασματικά, η παρούσα πτυχιακή εργασία διερευνά το Πρόβλημα Δρομολόγησης Οχημάτων με Σκοπό την Μείωση των Ρύπων (PRP) μέσα από την χρήση διαφόρων τεχνικών βελτιστοποίησης, δηλαδή του αλγόριθμου Πλησιέστερου Γείτονα, του αλγόριθμου 2-opt, του αλγόριθμου 1-0 relocate και της Προσομοιωμένης Ανόπτησης. Ο στόχος ήταν να διερευνηθεί η αποτελεσματικότητα αυτών των αλγορίθμων στην αντιμετώπιση του PRP, ένα κρίσιμο ζήτημα στη διασταύρωση των logistics και των μεταφορών με την περιβαλλοντική βιωσιμότητα.

Η εφαρμογή του αλγόριθμου του Πλησιέστερου Γείτονα παρόλη την απλότητά του αποδείχθηκε αποτελεσματικός στην δημιουργία αρχικών λύσεων για το PRP, θέτοντας τις βάσεις για τις επόμενες διαδικασίες βελτιστοποίησης. Ο αλγόριθμος τοπικής αναζήτησης 2-opt επέτρεψε τη βελτίωση των διαδρομών με την επαναληπτική εναλλαγή ζευγών κόμβων, αποδεικνύοντας την ικανότητά του να βελτιώνει την ποιότητα της λύσης μέσω της τοπικής αναζήτησης. Η ενσωμάτωση του αλγόριθμου 1-0 relocate στην 2-opt, με την ικανότητά του να μεταφέρει μεμονωμένους κόμβους σε μια διαδρομή, συνέβαλε περαιτέρω στη βελτίωση της ποιότητας της λύσης εξερευνώντας εναλλακτικές διαδρομές. Η Προσομοιωμένη Ανόπτηση, χρησιμοποιώντας ως αρχική λύση το τελικό αποτέλεσμα των παραπάνω αλγορίθμων, εισήγαγε μια συνολική προοπτική βελτιστοποίησης επιτρέποντας την πιθανολογική αποδοχή μη βέλτιστων λύσεων, αποφεύγοντας τον εγκλωβισμό σε τοπικό ελάχιστο και διευκολύνοντας την εξερεύνηση του χώρου λύσεων. Τέλος η επιπλέον χρήση του αλγορίθμου 2-opt, φάνηκε ότι σε αρκετές περιπτώσεις βοηθούσε στην περαιτέρω βελτίωση του αποτελέσματος της Προσομοιωμένης Ανόπτησης.

Η υλοποίηση του αλγορίθμου πραγματοποιήθηκε σε 9 σετ δεδομένων, όπου κάθε παράδειγμα περιείχε διαφορετικό αριθμό κόμβων. Ο αλγόριθμος για κάθε σετ δεδομένων εκτελέστηκε 5 φορές με σκοπό την άντληση ενός μέσου όρου για κάθε σετ.

Από τα αποτελέσματα της έρευνας προκύπτει ότι οι αλγόριθμοι κατάφεραν να ανταποκριθούν στις ανάγκες του προβλήματος μειώνοντας το κόστος της αρχικής λύσης έως και 43% από την λύση του αλγορίθμου του Κοντινότερου Γείτονα. Κατά μέσο όρο η μείωση αυτή ήταν της τάξης του 26,53%. Μόνο σε μία περίπτωση η μείωση αυτή ήταν πολύ μικρή (1,42%) που όμως αποδίδεται στην πολύ καλή αρχική λύση που κατασκεύαζε ο αλγόριθμος του Κοντινότερου Γείτονα.

Συνοψίζοντας, η ανάλυση του Προβλήματος Δρομολόγησης Οχημάτων με Σκοπό την Μείωση των Ρύπων (PRP) μέσω των αλγορίθμων Πλησιέστερου Γείτονα, 2-opt, 1-0 relocate και Προσομοιωμένης Ανόπτησης παρέχει πολύτιμες γνώσεις για τη εξαγωγή αποφάσεων όσο αναφορά την βελτιστοποίηση δρομολόγησης οχημάτων ελαχιστοποιώντας παράλληλα τις περιβαλλοντικές επιπτώσεις. Αυτή η έρευνα συμβάλλει στη συνεχιζόμενη συζήτηση για τη βιώσιμη εφοδιαστική και τις μεταφορές, ανοίγοντας το δρόμο για την ανάπτυξη πρακτικών και αποτελεσματικών λύσεων για την αντιμετώπιση των προκλήσεων που θέτει η ρύπανση στη δρομολόγηση.

8. Βιβλιογραφία

1. Μαρινάκης Ιωάννης, Μαρινάκη Μαγδαληνή & Μυγδαλάς, Αθανάσιος (2019). Προβλήματα Δρομολόγησης Οχημάτων στη Διαχείριση της Εφοδιαστικής Αλυσίδας (2η έκδοση). Χανιά: Εκδόσεις Νέων Τεχνολογιών.

2. Vincent F.Yu, A.A.N. Perwira Redi, Parida Jewpanya, Artya Lathifah, Meilinda F. N. Maghfiroh & Nur Aini Masruroh (2018*). A Simulated Annealing Heuristic for the Heterogeneous Fleet Pollution Routing Problem.* Environmental Sustainability in Asian Logistics and Supply Chains (pp 171–204).

3. Tolga Bektaş, Gilbert Laporte (2011). *The Pollution-Routing Problem*. Transportation Research Part B: Methodological, Volume 45, Issue 8, (pp 1232-1250).

4. Emrah Demir, Tolga Bektas , Gilbert Laporte (2012). *An adaptive large neighborhood search heuristic for the Pollution-Routing Problem.* European Journal of Operational Research, Volume 223, Issue 2, (pp 346-359).

5. Raphael Kramer, Anand Subramanian, Thibaut Vidal, Lucídio dos Anjos F. Cabral (2015). *A matheuristic approach for the Pollution-Routing Problem.* European Journal of Operational Research, Volume 243, Issue 2, (pp 523-539).

6. Anna Franceschetti, Dorothée Honhon, Tom Van Woensel, Tolga Bektaş, Gilbert Laporte (2013). *The time-dependent pollution-routing problem.* Transportation Research Part B: Methodological, Volume 56, (pp 265-293).

7. Μαλινδρέτος, Γ. (2015). Εφοδιαστική αλυσίδα, logistics και εξυπηρέτηση πελατών. Προπτυχιακό εγχειρίδιο. Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδόσεις.

8. Καμπέρη Αγγελική (2023). *Επίλυση του Συσσωρευτικού Προβλήματος Δρομολόγησης Οχημάτων με τον Μιμητικό Αλγόριθμο*. Διπλωματική Εργασία. Τμήμα Μηχανικών Παραγωγής και Διοίκησης. Πολυτεχνείο Κρήτης, Χανιά.

9. Σκευοφύλαξ Παναγιώτης (2019). *Εξελικτικοί αλγόριθμοι για το ανοικτό – κλειστό πρόβλημα δρομολόγησης οχημάτων*. Διπλωματική Εργασία. Τμήμα Μηχανικών Παραγωγής και Διοίκησης. Πολυτεχνείο Κρήτης, Χανιά.

10. Αγγελή Βασιλική (2015). *Αλγόριθμος Μεταβλητής Γειτονιάς Αναζήτησης για το Πρόβλημα Δρομολόγησης Οχημάτων Περιορισμένης Απόστασης*. Διπλωματική Εργασία. Τμήμα Μηχανικών Παραγωγής και Διοίκησης. Πολυτεχνείο Κρήτης, Χανιά.

11. Δημήτρης Ασκούνης, Δημήτρης Πανόπουλος. *Εφοδιαστική (Logistics)*. Συστήματα Διοίκησης. Εργαστήριο Συστημάτων Αποφάσεων και Διοίκησης. Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών. Εθνικό Μετσόβιο Πολυτεχνείο.

Ηλεκτρονική Βιβλιογραφία

12. https://en.wikipedia.org/wiki/Travelling_salesman_problem

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

14. https://www.netsuite.com/portal/resource/articles/erp/supply-chain-management-vs-logistics.shtml?fbclid=IwAR3JOriVcW7W6hU1NMtyEn5fFB5Ejb5kHyRdp5pSdzDJZEEXRsKcjFVqpA8

15. https://el.wikipedia.org/wiki/Διαχείριση_εφοδιαστικής_αλυσίδας?fbclid=IwAR31n3sYKfWU7aQVh76-x5CO5AQ9bacnRmwukKTKtxH6zjV9UVhD-Q0AA0w

Πηγές Εικόνων

1. https://www.bankingnews.gr/analyseis-ektheseis/articles/593656/to-xaos-stin-efodiastiki-alysida-plittei-tin-pagkosmia-anaptyksi-i-katastasi-tha-epideinothei

2. https://en.wikipedia.org/wiki/Vehicle_routing_problem

3. https://www.sigma-consultants.eu/sectors/12-Διοίκηση_Εφοδιαστικής_Αλυσίδας