ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

Σχολή Μηχανικών Παραγωγής και Διοίκησης

Επίλυση του Προβλήματος Δρομολόγησης Οχημάτων Περιορισμένης Φόρτωσης και Χωρητικότητας με Χρήση Μεθευρετικού Αλγορίθμου Περιορισμένης Αναζήτησης (Tabu Search)

Διπλωματική Εργασία Μ.Δ.Ε

ΕΦΑΡΜΟΣΜΕΝΑ ΜΑΘΗΜΑΤΙΚΑ ΣΤΙΣ ΕΠΙΣΤΗΜΕΣ ΜΗΧΑΝΙΚΩΝ

ΚΡΑΣΑΚΗΣ ΠΑΝΑΓΙΩΤΗΣ

Επιβλέπων Καθηγητής : Δρ. Ιωάννης Μαρινάκης

ΧΑΝΙΑ, 2020

ΕΥΧΑΡΙΣΤΙΕΣ

Θέλω να ευχαριστήσω τον επιβλέποντα της διπλωματικής εργασίας μου, κ. Ιωάννη Μαρινάκη για το ενδιαφέρον θέμα που μου ανέθεσε, καθώς και για τη βοήθεια και καθοδήγηση του κατά τη διάρκεια της υλοποίησης της. Πάνω απ' όλα όμως θα ήθελα να ευχαριστήσω όλη την οικογένεια μου για τη στήριξη τους όλα αυτά τα χρόνια των σπουδών μου.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ

Στη σημερινή εποχή λόγω της παγκοσμιοποίησης και των διαρκώς αυξανόμενων απαιτήσεων των πελατών η ανάπτυξη της εφοδιαστικής αλυσίδας αποτελεί καθοριστικό παράγοντα για τη δημιουργία πλεονεκτημάτων στις επιχειρήσεις. Η μεταφορά και η διανομή προϊόντων είναι από τις σημαντικότερες δραστηριότητες της εφοδιαστικής αλυσίδας και συνήθως αντιστοιχούν σε μεγάλο ποσοστό των δαπανών μιας επιχείρησης. Η διαχείριση του στόλου οχημάτων για την μεταφορά και την διανομή αποτελεί κομβικό σημείο. Σήμερα τα καύσιμα είναι ακριβά και οι αστοχίες απαγορεύονται, η ορθή διαχείριση στοχεύει όχι μόνο στην απρόσκοπτη λειτουργία, αλλά και στον περιορισμό του κόστους. Αυτά σε συνδυασμό με την βελτίωση της εξυπηρέτησης των πελατών, εξασφαλίζουν ανταγωνιστικό πλεονέκτημα στις επιχειρήσεις. Για να επιτευχθεί αυτό απαιτείται βέλτιστη δρομολόγηση οχημάτων, η οποία μπορεί να καθοριστεί από αλγοριθμική επίλυση των αντίστοιχων προβλημάτων δρομολόγησης. Στην παρούσα διπλωματική εργασία επιλύουμε το πρόβλημα δρομολόγησης οχημάτων περιορισμένης φόρτωσης και χωρητικότητας. Έχοντας ένα σύνολο πελατών – κόμβων όπου κάθε ένας έχει την δική του ζήτηση και όντας γεωγραφικά διασκορπισμένοι προσπαθούμε μέσω αλγοριθμικών διαδικασιών να προσεγγίσουμε την βέλτιστη σειρά εξυπηρέτησής τους. Κάθε όχημα μπορεί να επισκεφτεί μονάχα μία φορά τον κάθε πελάτη και τα μήκη που διανύουν τα οχήματα σε κάθε φάση της εξυπηρέτησης αποτελούν το κόστος της συνολικής διαδικασίας. Αρχικά μέσω του αλγορίθμου του πλησιέστερου γείτονα κατασκευάζουμε μία αρχική εφικτή λύση. Στην συνέχεια, με τη χρήση των τοπικών αναζητήσεων των 1-0 relocate, 1-1 exchange (στην ίδια διαδρομή) και 1-1 exchange (σε διαφορετικές διαδρομές) γίνεται μία αρχική προσπάθεια να βελτιωθεί η αρχική λύση. Προκειμένου να αποφύγουμε την παγίδευση του αλγορίθμου σε τοπικά ελάχιστα και για να βελτιώσουμε και άλλο την αρχική μας λύση γίνεται εφαρμογή του μεθευρετικού αλγορίθμου περιορισμένης αναζήτησης (Tabu Search). Τέλος προκειμένου να έχουμε εφικτή φόρτωση των αντικειμένων στα οχήματα έτσι ώστε να μην παραβιάζονται οι περιορισμοί φόρτωσης, εφαρμόζουμε ένα σύνολο από μεθόδους οι οποίες την καταστούν πραγματοποιήσιμη. Κατά την εκπόνηση της παρούσας εργασίας αναπτύσσεται αλγόριθμος με χρήση της γλώσσας προγραμματισμού Fortran, και η αποτελεσματικότητα αυτού δοκιμάζεται κάνοντας χρήση παραδειγμάτων αναφοράς από τη βιβλιογραφία .

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Περιγραφή του προβλήματος

Στην παρούσα διπλωματική εργασία ασχολούμαστε με την επίλυση του προβλήματος δρομολόγησης οχημάτων περιορισμένης φόρτωσης και χωρητικότητας με χρήση αλγορίθμου περιορισμένης αναζήτησης (Tabu Search).

Τα τελευταία χρόνια τόσο η πρόοδος στις μεθόδους βελτιστοποίησης όσο και στα συστήματα πληροφορικής μας επιτρέπουν πλέον να εξετάσουμε σε πρακτικό κομμάτι τα προβλήματα βελτιστοποίησης που στο παρελθόν ήταν πολύ πιο δύσκολο και περίπλοκο να τα χειριστούμε. Ένα τέτοιο πρόβλημα βελτιστοποίησης αποτελεί και το πρόβλημα που παρουσιάζεται στη συγκεκριμένη διπλωματική εργασία, και έχει ως στόχο τη σωστή αποστολή ενός στόλου οχημάτων και συγχρόνως να εξασφαλίσει και το σωστό τρόπο τοποθέτησης ενός συνόλου αντικειμένων μέσα σε αυτά τα οχήματα. Το παρών πρόβλημα αποτελεί μια παραλλαγή του κλασικού προβλήματος δρομολόγησης οχημάτων (VRP) και μας παρουσιάζει την περίπτωση όπου τα αντικείμενα τα οποία είναι για παράδοση, και πρέπει να φορτωθούν στα οχήματα είναι δισδιάστατα και με συγκεκριμένο βάρος το κάθε ένα.

Το πρόβλημα δρομολόγησης οχημάτων με περιορισμένη φόρτωση και χωρητικότητα (2L-CVRP) είναι ένα πρακτικό πρόβλημα που συχνά εμφανίζεται στον τομέα της εφοδιαστικής αλυσίδας και αποτελεί συνδυασμό δύο από τα ποιο σημαντικά μη-πολυωνυμικά δύσκολα προβλήματα βελτιστοποίησης (NP-Hard) στον τομέα της μεταφοράς προϊόντων, του περιορισμένης χωρητικότητας προβλήματος δρομολόγησης οχημάτων (CVRP) και του δύο διαστάσεων προβλήματος συσκευασίας (2BPP).

Στο συγκεκριμένο πρόβλημα η ζήτηση των πελατών έχει να κάνει με αντικείμενα που όπως έχουμε αναφέρει έχουν συγκεκριμένες διαστάσεις (πλάτος και μήκος) και συγκεκριμένο βάρος, ενώ τα οχήματα έχουν την ίδια χωρητικότητα και την ίδια επιφάνεια φόρτωσης. Σκοπός μας είναι να δημιουργήσουμε διαδρομές με το μικρότερο δυνατό κόστος έτσι ώστε τα οχήματα να εξυπηρετήσουν τους πελάτες και συγχρόνως να έχουμε το σωστό τρόπο φορτώματος και ξεφορτώματος των αντικειμένων στα οχήματα αλλά και από τα οχήματα. Προκειμένου να μπορέσουμε να πετύχουμε το σκοπό αυτό θα πρέπει να τηρηθούν κάποιοι περιορισμοί τους οποίους και θα αναφέρουμε αναλυτικότερα στη συνέχεια της εργασίας.

Το 2L-CVRP πρόβλημα είναι ένα ιδιαίτερα σημαντικό πρόβλημα τόσο από θεωρητικής μεριάς καθώς είναι συνδυασμός δύο NP-Hard προβλημάτων και επομένως αυτό το κάνει ένα ακόμα πιο περίπλοκο NP-Hard πρόβλημα όσο και από πρακτικής μεριάς καθώς έχει μεγάλη εμπορική αξία. Είναι ένα πρόβλημα το οποίο έχει πολλές εφαρμογές στο τομέα των μεταφορών, και συγκεκριμένα στο κομμάτι της διαχείρισης των διανομών και των παραλαβών αντικειμένων με συγκεκριμένες διαστάσεις τα οποία δεν μπορούν να στοιβαχτούν είτε λόγο του μεγάλου βάρους τους ή του ότι μπορεί να είναι εύθραυστα ή για οποιονδήποτε άλλο λόγο (οικιακές συσκευές, έπιπλα, κ.α.).

Για την επίλυση του 2L-CVRP θα χρησιμοποιήσουμε τον μεθευρετικό αλγόριθμο περιορισμένης αναζήτησης Tabu Search τον οποίο και θα παρουσιάσουμε πιο αναλυτικά σε επόμενο κεφάλαιο. Ο συγκεκριμένος αλγόριθμος προτάθηκε πρώτη φορά για 2L-CVRP πρόβλημα από τους Gendreau και Martello, προκειμένου να μπορέσουν να λύσουν προβλήματα αρκετά μεγάλης κλίμακας και συγκεκριμένα με πάνω από 255 πελάτες και 700 αντικείμενα που οι μέχρι τότε αλγόριθμοι δεν μπορούσαν να επιλύσουν. Η διαδικασία της φόρτωσης των αντικειμένων του κάθε πελάτη στα οχήματα πραγματοποιείτε με την εφαρμογή μιας μεθόδου η οποία αφού δέχεται κάθε φορά τις διαστάσεις του αντικειμένου το οποίο είναι για φόρτωμα πραγματοποιεί μέσα από ένα σύνολο εντολών τον έλεγχο της επιφάνειας φόρτωσης του οχήματος προκειμένου να βρει τη σωστή θέση τοποθέτησης του αντικειμένου. Πιο αναλυτική περιγραφή όλων των διαδικασιών καθώς και του αλγορίθμου που κατασκευάσαμε θα γίνει στα επόμενα κεφάλαια.

Στο επόμενο κεφάλαιο της εργασίας γίνεται μια εκτενής αναφορά στα προβλήματα δρομολόγησης οχημάτων καθώς επίσης αναφέρουμε και μερικά από αυτά. Στη συνέχεια γίνεται αναφορά στους απλούς ευρετικούς αλγορίθμους αλλά και στους μεθευρετικούς αλγορίθμους όπου γίνεται και περιγραφή του αλγορίθμου περιορισμένης αναζήτησης (Tabu Search) τον οποίο και θα χρησιμοποιήσουμε στο πρόβλημα μας. Σε επόμενο κεφάλαιο γίνεται μια πιο αναλυτική παρουσίαση του προβλήματος σε συνδυασμό με κάποιες αναφορές στη βιβλιογραφία. Τέλος γίνεται παρουσίαση και περιγραφή του αλγορίθμου που κατασκευάσαμε για να επιλύσουμε το πρόβλημα μας .

ΚΕΦΑΛΑΙΟ 2

ΔΡΟΜΟΛΟΓΗΣΗ ΟΧΗΜΑΤΩΝ

2.1 Το Πρόβλημα Δρομολόγησης Οχημάτων

Η παγκοσμιοποίηση της οικονομίας έχει οδηγήσει στην συνεχιζόμενη ανάπτυξη του εμπορίου και των μεταφορών σε παγκόσμιο επίπεδο. Το υψηλό κόστος των μεταφορών, οι περιορισμοί που υπάρχουν η πολυπλοκότητα των προβλημάτων, καθώς και οι υψηλές απαιτήσεις που έχουν οι πελάτες οδήγησαν τις επιχειρήσεις στην ανάπτυξη μεθόδων και στρατηγικών επίλυσης, για τη βελτίωση του κόστους των προβλημάτων δρομολόγησης οχημάτων.

Τα προβλήματα που έχουν ως αντικείμενο την δρομολόγηση οχημάτων εντός των επιχειρησιακών δραστηριοτήτων και περιλαμβάνουν τόσο την παράδοση ,όσο και την παραλαβή προϊόντων από τους πελάτες, είναι γνωστά ως Προβλήματα Δρομολόγησης Οχημάτων (Vehicle Routing Problems). Τα προβλήματα αυτά έχουν να κάνουν με τη διανομή προϊόντων σε ένα συγκεκριμένο αριθμό πελατών που είναι διασκορπισμένοι γεωγραφικά στο χώρο, από μια ή περισσότερες αποθήκες, και με ένα συγκεκριμένο αριθμό διαθέσιμων οχημάτων και οδηγών για μια δεδομένη χρονική στιγμή και μέσα από ένα συγκεκριμένο οδικό δίκτυο.

Το πρόβλημα δρομολόγησης οχημάτων αποτελεί μια γενίκευση του προβλήματος του πλανόδιου πωλητή (Travelling Salesman Problem-TSP), και πρωτοεμφανίστηκε το 1959 από τους G.Dantzig και J.Ramser οι οποίοι εισήγαγαν μια αλγοριθμική προσέγγιση γραμμένη σε χαρτί και με πράξεις στο χέρι του προβλήματος για διανομές με σκοπό την ελαχιστοποίηση του κόστους την οποία και κατάφεραν να προσεγγίσουν επιτυχώς. Συνήθως το πρόβλημα αυτό απεικονίζεται ως μια αποθήκη στο κέντρο του χώρου δράσης, με τους πελάτες να βρίσκονται γύρω από την αποθήκη προκειμένου να εξυπηρετηθούν. Ο σκοπός της επίλυσης του προβλήματος δρομολόγησης οχημάτων είναι η εύρεση των διαδρομών εκείνων όπου η εφαρμογή τους θα μας δώσει το ελάχιστο συνολικό κόστος, δηλαδή την ελάχιστη συνολική απόσταση που θα διανύσουν όλα τα οχήματα περνώντας από όλους τους πελάτες. Συγχρόνως στόχος είναι και η ελαχιστοποίηση του αριθμού των οχημάτων, συνεπώς και τον οδηγών που θα χρησιμοποιηθούν. Όλοι οι πιθανοί περιορισμοί που ενδέχεται να υπάρχουν καθώς και όλες οι δυνατές πληροφορίες που είναι προς αξιοποίηση θα πρέπει να ικανοποιούνται από τη λύση του προβλήματος. Παρακάτω παρουσιάζονται οι περιορισμοί που πρέπει να λάβουμε υπόψη και αφορούν πληροφορίες για τα χαρακτηριστικά των πελατών, των οχημάτων, των διαδρομών καθώς και των οδηγών.

Χαρακτηριστικά των πελατών :

- Το σημείο του γραφήματος στο οποίο βρίσκεται ο πελάτης.

- Η ποσότητα των αγαθών (Demand), που πρέπει να παραλάβει ή να παραδώσει ο πελάτης.
- Τα χρονικά παράθυρα (Time Windows) κατά τη διάρκεια της ημέρας στα οποία ο πελάτης μπορεί να εξυπηρετηθεί.
- Ο χρόνος που απαιτείται για την παράδοση ή τη συλλογή των προϊόντων από τον πελάτη (Unloading or Loading time), πιθανότατα εξαρτώμενος από το είδος του οχήματος.
- Το είδος του οχήματος που μπορεί να χρησιμοποιηθεί από το στόλο της εταιρίας για την εξυπηρέτηση κάποιου πελάτη.

<u>Χαρακτηριστικά των οχημάτων :</u>

- Από ποια αποθήκη προέρχονται και αν υπάρχει πιθανότητα να τερματίσουν την διαδρομή τους σε άλλη αποθήκη από εκείνη που ξεκίνησαν.
- Η χωρητικότητα του οχήματος.
- Αν τα οχήματα είναι χωρισμένα σε τμήματα και πως θα φορτωθεί το κάθε τμήμα του οχήματος.
- Αν τα οχήματα είναι χωρισμένα σε ομάδες, κάθε μια από τις οποίες χαρακτηρίζεται από τη χωρητικότητα και από το είδος των προϊόντων που μπορεί να μεταφέρει.
- Αν υπάρχουν διαθέσιμα μηχανήματα για την φόρτωση και την εκφόρτωση των οχημάτων.
- Το σύνολο των δρόμων που είναι προσπελάσιμοι από το όχημα.
- Το κόστος που συσχετίζεται με την λειτουργία του κάθε οχήματος.

Σε αυτό το σημείο πρέπει να αναφέρουμε ότι και οι οδηγοί των οχημάτων πρέπει να ακολουθούν κάποιους κανόνες ,οι οποίοι δημιουργούν και αυτοί με τη σειρά τους κάποιους περιορισμούς στο πρόβλημα, και είναι βασικοί και για την υγεία τους.

Ενδεικτικά θα πρέπει να έχουν το ελάχιστο οχτώ ώρες ύπνου καθημερινώς, να μην οδηγούν συνεχόμενα παραπάνω από δέκα ώρες, καθώς και παραπάνω από δεκαπέντε ώρες οδήγηση μέσα σε μια μέρα, όπως και να μην δουλεύουν πάνω από έξι μέρες συνεχόμενα.

Όπως και με τους οδηγούς των οχημάτων που υπόκεινται σε κάποιους περιορισμούς έτσι και με τις διαδρομές έχουμε τους αντίστοιχους, οι οποίοι μπορεί να σχετίζονται με το προϊόν που διανέμεται ή να εξαρτώνται από τα χαρακτηριστικά των προηγούμενων κατηγοριών.

<u>Χαρακτηριστικά των διαδρομών.</u>

- Η συνολική ποσότητα που μεταφέρει το όχημα δεν πρέπει να ξεπερνάει τη χωρητικότητα του.
- Υπάρχουν πελάτες που ζητούν παράδοση ή παραλαβή ή και τα δυο.
- Η εξυπηρέτηση να πραγματοποιείτε σε συγκεκριμένη χρονική περίοδο, «χρονικά παράθυρα» και τις ώρες εργασίας των οδηγών του κάθε οχήματος.
- Τα οχήματα μπορεί να μεταφέρουν παραπάνω από ένα προϊόντα.
- Οι πελάτες να πρέπει να εξυπηρετηθούν με συγκεκριμένη σειρά.

Καταλήγοντας από όλα τα παραπάνω χαρακτηριστικά προκύπτουν και συγκεκριμένοι περιορισμοί που πρέπει να ληφθούν υπόψη κατά τη διάρκεια της υλοποίησης και της αναζήτησης της βέλτιστης λύσης, ώστε το τελικό αποτέλεσμα να είναι εφικτό και πραγματοποιήσιμο.

Οι στόχοι ορίζονται από την εκάστοτε επιχείρηση ανάλογα με το τι θέλει να πετύχει , τις απαιτήσεις των πελατών, το αγοραστικό κοινό στο οποίο απευθύνεται καθώς και στις προτεραιότητες που έχει η ίδια. Τέτοιοι στόχοι μπορεί να είναι :

- Η ελαχιστοποίηση του συνολικού κόστους μεταφοράς .
- Η ελαχιστοποίηση των οχημάτων ή και των οδηγών.
- Η ισορροπία μεταξύ των διαδρομών σε σχέση με τις απαιτούμενες ώρες που χρειάζονται .
- Η ελαχιστοποίηση των ποινών από τους μη ικανοποιημένους πελάτες.
- Η ελαχιστοποίηση των καυσίμων.
- Η ελαχιστοποίηση του συνολικού χρόνου στις διαδρομές.

2.1.1 Μοντελοποίηση Προβλήματος Δρομολόγησης Οχημάτων

Η μαθηματική μοντελοποίηση του απλού προβλήματος δρομολόγησης οχημάτων αφορά ένα στόλο από πανομοιότυπα οχήματα που πραγματοποιεί διανομές σε πελάτες από μια κεντρική αποθήκη. Οι παράμετροι που ορίζουν το πρόβλημα είναι οι ακόλουθοι :

- k: ο αριθμός των οχημάτων, (k=1, 2…. ν).

- n: ο αριθμός των σημείων από τα οποία διέρχεται το όχημα (n=1 συμβολίζουμε την αποθήκη). Άρα n-1 ο αριθμός των πελατών.

- q_i: ζήτηση του πελάτη i.

- c_{ij}: κόστος διαδρομής από τον πελάτη i στον πελάτη j (υποθέτουμε ότι c_{ij}>0 και $c_{ij} = c_{ij}$ για κάθε i,j).

Από το πρόβλημα δρομολόγησης οχημάτων θα πρέπει να δημιουργηθούν k διαδρομές οχημάτων, όπου η κάθε μια διαδρομή θα ξεκινάει και θα καταλήγει στην αποθήκη και τα οχήματα θα περνούν από ένα υποσύνολο διαδρομών. Οι συγκεκριμένες διαδρομές θα επιλεχτούν έτσι ώστε να ελαχιστοποιηθεί το κόστος διανομής.

Επομένως από τα παραπάνω προκύπτει η μαθηματική διατύπωση του προβλήματος :

Αντικειμενική συνάρτηση :

$$c^* = min \sum_{v} \sum_{ij} c_{ij}\ x_{ij}^{v}$$

Περιορισμοί:

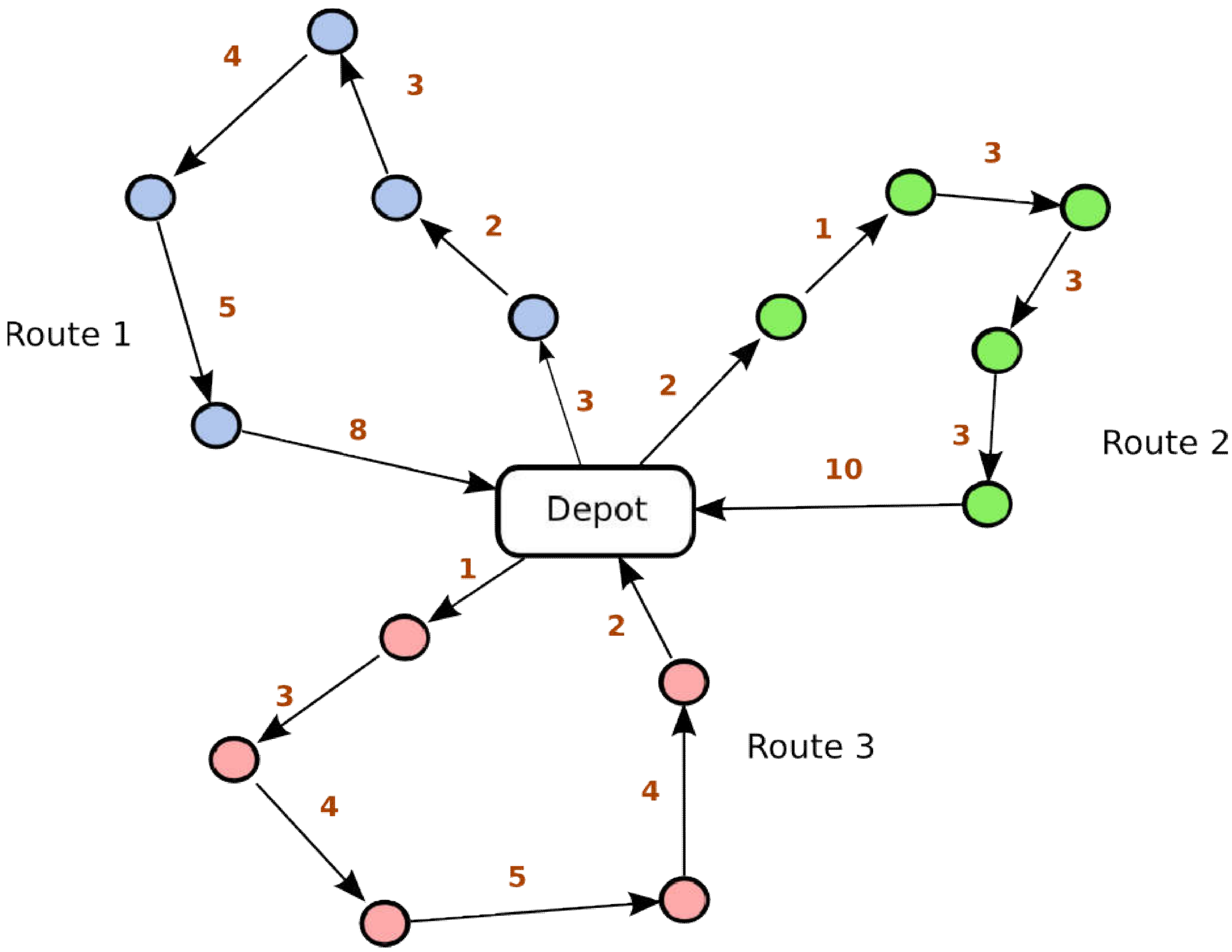
$$\sum q_i\, \chi_{ij} \leq k \qquad \text{για κάθε } k = 1,2, \ldots . m$$

$$\sum_{j} \chi_{ijk} = \sum_{j} \chi_{jik} = y_{ik} \qquad i = 1, \ldots . , n \ \text{ και } k = 1, \ldots , m$$

$$\sum_{k} y_{ik} = \begin{cases} 1, & i = 2, \ldots n \\ m, & i = 1 \end{cases}$$

X= [χ_{ij}^{v}]∈ S^*, όπου S^*το σύνολο όλων των λύσεων του προβλήματος.

$$\chi_{ij}^{v} = \begin{cases} 1, \text{το όχημα } v \text{ πραγματοποιεί τη διαδρομή } (i,j) \\ 0, \text{το όχημα } v \text{ δενπραγματοποιεί τη διαδρομή } (i,j) \end{cases}$$

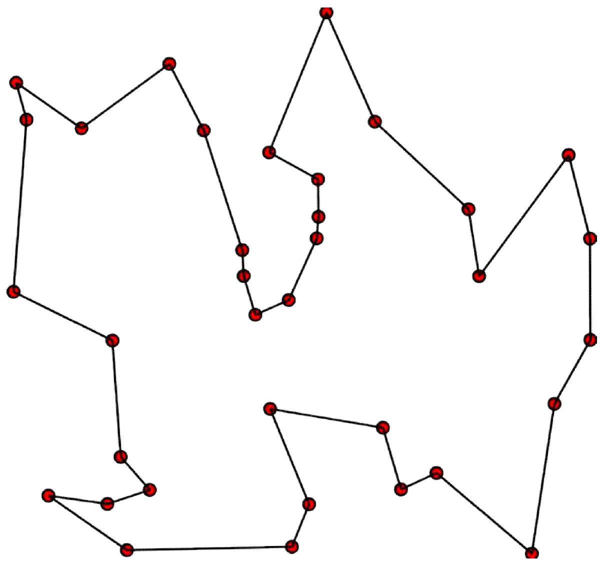


Σχήμα 2.1: Πρόβλημα Δρομολόγησης Οχημάτων (Πηγή :http://neo.lcc.uma.es/dynamic/vrp.html)

Το Πρόβλημα Δρομολόγησης Οχημάτων είναι ένα πρόβλημα ακέραιου προγραμματισμού το οποίο ανήκει στην κατηγορία των μη-πολυωνυμικών δύσκολων προβλημάτων (NP–Hard). Αυτό πρακτικά σημαίνει ότι η υπολογιστική προσπάθεια που απαιτείται για την επίλυση αυτού του προβλήματος αυξάνει εκθετικά ανάλογα με το μέγεθος του. Ανάλογα με τους περιορισμούς που περιγράφηκαν παραπάνω, υπάρχουν διάφορες παραλλαγές και θα γίνει αναφορά στα επόμενα κεφάλαια. Στην παρούσα διπλωματική εργασία θα ασχοληθούμε με το πρόβλημα δρομολόγησης οχημάτων περιορισμένης φόρτωσης και χωρητικότητας. Τέλος προκειμένου να βελτιωθεί η αποτελεσματικότητα των Προβλημάτων Δρομολόγησης Οχημάτων έχουν αναπτυχθεί διάφοροι αλγόριθμοι οι οποίοι θα παρουσιαστούν σε επόμενο κεφάλαιο.

2.1.2 Το Πρόβλημα του Πλανόδιου Πωλητή (TSP)

Ένα από τα πιο βασικά και διασημότερο πρόβλημα δρομολόγησης οχημάτων είναι αυτό του πλανόδιου πωλητή. Ουσιαστικά το πρόβλημα του πλανόδιου πωλητή (Traveling Salesman Problem) αποτέλεσε το πρωταρχικό πρόβλημα, όπου στη συνέχεια έγινε αναγωγή και δημιουργήθηκε το Πρόβλημα Δρομολόγησης Οχημάτων όπως αναφέρουμε παραπάνω. Το Πρόβλημα του Πλανόδιου Πωλητή αφορά την εύρεση της συντομότερης (σε χρόνο, απόσταση ή άλλο κόστος) διαδρομής για ένα όχημα (ή πωλητή) με αφετηρία κάποιο σημείο (π.χ. ένα κέντρο διανομής) και επιστροφή στο ίδιο σημείο αφού επισκεφτεί ένα σταθερό αριθμό πελατών ακριβώς μία φορά τον καθένα. Δεν υπάρχουν περαιτέρω περιορισμοί, δεδομένων των τοποθεσιών (κόμβων) που πρέπει να επισκεφτεί το όχημα και των αποστάσεων μεταξύ αυτών, το αποτέλεσμα του εν λόγω αλγορίθμου είναι μία συγκεκριμένη διαδρομή που επισκέπτεται όλους τους κόμβους από μία φορά και επιστρέφει στο σημείο εκκίνησης έχοντας διανύσει την ελάχιστη δυνατή απόσταση. Παρόλο που τα δεδομένα και οι περιορισμοί έχουν μια ιδιαίτερα απλή μορφή, στην πραγματικότητα η επίλυση του Προβλήματος του Πλανόδιου Πωλητή μπορεί να γίνει αρκετά δύσκολη και με μεγάλο υπολογιστικό φόρτο αν λάβουμε υπόψη ένα μεγάλο αριθμό κόμβων (πόλεων) προς επίσκεψη και αναλογιστούμε πόσοι πιθανοί συνδυασμοί προκύπτουν ή αλλιώς πόσες εναλλακτικές διαδρομές προς εξέταση. Συνεπώς, για μεγάλο αριθμό πόλεων κρίνεται δύσκολη έως αδύνατη η εξέταση όλων των πιθανών συνδυασμών. Για το λόγο αυτό επιλέγονται μέθοδοι που δίνουν μια προσεγγιστική λύση στο πρόβλημα και όχι τη βέλτιστη δυνατή, κάνοντας χρήση ευρετικών αλγορίθμων για την εύρεση του επόμενου κόμβου που θα επισκεφτεί το όχημα.



Σχήμα 2.2: Επίλυση του Πλανόδιου Πωλητή(Πηγή: https://en.wikipedia.org/wiki/Travelling_salesman_problem)

2.2 Πρόβλημα Δρομολόγησης Οχημάτων Με Περιορισμένη Χωρητικότητα (CVRP)

Το πρόβλημα της δρομολόγησης οχημάτων περιορισμένης χωρητικότητας (Capacitated Vehicle Routing Problem) είναι η πιο βασική εκδοχή του προβλήματος δρομολόγησης οχημάτων (VRP). Στο CVRP όλοι οι πελάτες δέχονται την παράδοση των προϊόντων, η ζήτηση τους είναι ντετερμινιστική και γνωστή εκ των προτέρων, ενώ δεν μπορεί να διασπαστεί. Τα οχήματα έχουν ίδια χωρητικότητα και ξεκινάνε από μια κεντρική αποθήκη και επιστρέφουν σε αυτή. Περιορισμός του προβλήματος αποτελεί το επίπεδο της ζήτησης σε κάθε όχημα. Στόχος είναι η ελαχιστοποίηση του κόστους για την εξυπηρέτηση όλων των πελατών. Το πρόβλημα αποτελεί μια επέκταση του προβλήματος του πλανόδιου πωλητή στην περίπτωση όπου ένα όχημα δεν φτάνει να εξυπηρετήσει όλους τους πελάτες. Έτσι δημιουργούνται κυκλικές διαδρομές με αρχή και τέλος την αποθήκη, όμως ο κάθε πελάτης θα καλύπτεται από μια και μόνο διαδρομή.

Παρακάτω περιγράφουμε το CVRP με όρους θεωρίας γραφημάτων. Έστω ότι έχουμε ένα πλήρες γράφημα G=(V,A), όπου V=(0,…….n) είναι το σύνολο των κόμβων και A το σύνολο των τόξων. Η αποθήκη αντιστοιχεί στο κόμβο 0 ή στο κόμβο 1. Το μη αρνητικό κόστος c_{ij} αντιστοιχεί σε κάθε τόξο (i,j) και αντιπροσωπεύει το κόστος του ταξιδιού από τον κόμβο i στο κόμβο j. Στις περισσότερες πρακτικές εφαρμογές οι κόμβοι συσχετίζονται με σημεία στο επίπεδο ,άρα τα δεδομένα είναι συντεταγμένες και κάθε κόστος c_{ij} υπολογίζεται από την Ευκλείδεια απόσταση ανάμεσα στα δυο σημεία που αντιστοιχούν στους κόμβους i, j. Στη συνέχεια θα δούμε μια βασική μορφοποίηση για την επίλυση του Προβλήματος Δρομολόγησης Οχημάτων Περιορισμένης Χωρητικότητας που βασίζεται στη ροή των οχημάτων έτσι όπως είχε παρουσιαστεί από τους Fisher και Jaikumar.

Μοντελοποίηση του CVRP:

$$\chi_{ijk}=\begin{cases} 1, & \text{εάν το όχημα } k \text{ επισκέπτεται τον πελάτη } j \text{ αμέσως μετά τον πελάτη } i \\ 0, & \text{αλλιώς} \end{cases}$$

$$y_{ik}=\begin{cases}1, & \varepsilon\acute{\alpha}\nu\ o\ \pi\varepsilon\lambda\acute{\alpha}\tau\eta\varsigma\ i\ \varepsilon\pi\iota\sigma\kappa\acute{\varepsilon}\pi\tau\varepsilon\tau\alpha\iota\ \alpha\pi o\ \tau o\ \acute{o}\chi\eta\mu\alpha\ k \\ 0, & \alpha\lambda\lambda\iota\acute{\omega}\varsigma\end{cases}$$

Ελαχιστοποιείται η αντικειμενική συνάρτηση κόστους:

$$min \sum_{ij} c_{ij} \sum_{k} \chi_{ijk}$$

Υπό:

$$\sum_{k} y_{ik} = \begin{cases}1, & i = 2, \dots n \\ m, & i = 1\end{cases} \qquad (2.1)$$

$$\sum_{i} q_i \, y_{ij} \leq Q_k \qquad k = 1, \dots, m \qquad (2.2)$$

$$\sum_{j} \chi_{ijk} = \sum_{j} \chi_{jik} = y_{ik} \qquad i = 1, \dots., n \ \ \kappa\alpha\iota\ k = 1, \dots, m \qquad (2.3)$$

$$\sum_{i,j \in S} \chi_{ijk} \leq |S| - 1 \quad \gamma\iota\alpha\ \acute{o}\lambda\alpha\ \tau\alpha\ S \subseteq \{2, \dots, n\}\ \kappa\alpha\iota\ \ k = 1, \dots, m \qquad (2.4)$$

$y_{ik} \in \{0,1\}$ i=1,…n και k=1,……m (2.5)

$\chi_{ijk} \in \{0,1\}$ i,j=1…., n και k =1…,m (2.6)

Ο περιορισμός (2.1) δείχνει ότι κάθε πελάτης εκχωρείται σε ένα μόνο όχημα , εκτός από την αποθήκη που την επισκέπτονται όλα τα οχήματα .

Ο περιορισμός (2.2) δείχνει ότι δεν πρέπει να παραβιαστεί η χωρητικότητα κάθε οχήματος σε ένα δρομολόγιο.

Ο περιορισμός (2.3) μας δείχνει ότι ένα όχημα που επισκέπτεται ένα πελάτη φεύγει και από αυτόν .

Τέλος ο περιορισμός (2.4) μας δείχνει ότι δεν επιτρέπει την υπερφόρτωση.

2.2.1 Πρόβλημα Δρομολόγησης Οχημάτων Περιορισμένης Φόρτωσης (δυο διαστάσεων) και Χωρητικότητας (2L-CVRP)

Το 2L-CVRP αποτελεί μια παραλλαγή του κλασικού CVRP το οποίο παρουσιάστηκε για πρώτη φορά από τους Lori, S.Gonzalez και Vigo. Το πρόβλημα αυτό μας παρουσιάζει την περίπτωση όπου τα αντικείμενα των πελατών έχουν δυο διαστάσεις (πλάτος και μήκος). Ένα 2L-CVRP μπορεί να έχει τη μορφή ενός απλού CVRP απλά και μόνο θέτοντας τις διαστάσεις των αντικειμένων ίσες με μηδέν και έχοντας ως χαρακτηριστικό μόνο το βάρος τους. Σκοπός του 2L-CVRP είναι να δημιουργήσει ένα σύνολο από διαδρομές όπου δεν παραβιάζονται οι περιορισμοί του απλού CVRP αλλά συγχρόνως να εγγυηθεί το σωστό φόρτωμα και ξεφόρτωμα των αντικειμένων από τα φορτηγά .Υπάρχουν δυο εκδοχές του 2L-CVRP η Unrestricted όπου μας ενδιαφέρει μόνο η σωστή φόρτωση των αντικειμένων στην επιφάνεια φόρτωσης των οχημάτων και η Sequential όπου μας ενδιαφέρει τόσο η φόρτωση όσο και το ξεφόρτωμα των αντικειμένων από τα οχήματα.

Πρέπει να αναφέρουμε ότι ο συγκεκριμένος τρόπος φόρτωσης αντικειμένων με δυο διαστάσεις σχετίζεται με το δυο διαστάσεων Bin Packing Problem (2BPP) το οποίο είναι ένα NP-Hard πρόβλημα συνδυαστικής βελτιστοποίησης, που έχει ως σκοπό τη φόρτωση ενός συνόλου αντικειμένων με δύο διαστάσεις σε όσο το δυνατών λιγότερα οχήματα ίδιας διάστασης και χωρητικότητας. Το 2L-CVRP είναι ένα ιδιαίτερα σημαντικό πρόβλημα τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο, από τη θεωρητική μεριά είναι ένα πρόβλημα το οποίο απαρτίζεται από δύο NP-Hard προβλήματα βελτιστοποίησης το CVRP και το 2BPP, καθώς επίσης είναι και ένα δύσκολο NP-Hard πρόβλημα με μεγάλη πολυπλοκότητα. Από την άλλη μεριά από πρακτικής άποψης το 2L-CVRP έχει μεγάλη εμπορική αξία, καθώς μπορεί να εφαρμοστεί σε πολλές περιπτώσεις στην κανονική ζωή που αφορούν τη σωστή λειτουργία της διανομής και της παραλαβής στο πλαίσιο της μεταφοράς που αφορά αντικείμενα συγκεκριμένων διαστάσεων και σχημάτων τα οποία δεν μπορούν να στοιβαχτούν ή λόγω του βάρους τους ή το ότι είναι εύθραυστα. Στη συνέχεια γίνεται περιγραφή του 2L-CVRP με όρους γραφημάτων.

Το 2L-CVRP μπορεί να προσδιοριστεί ως εξής:

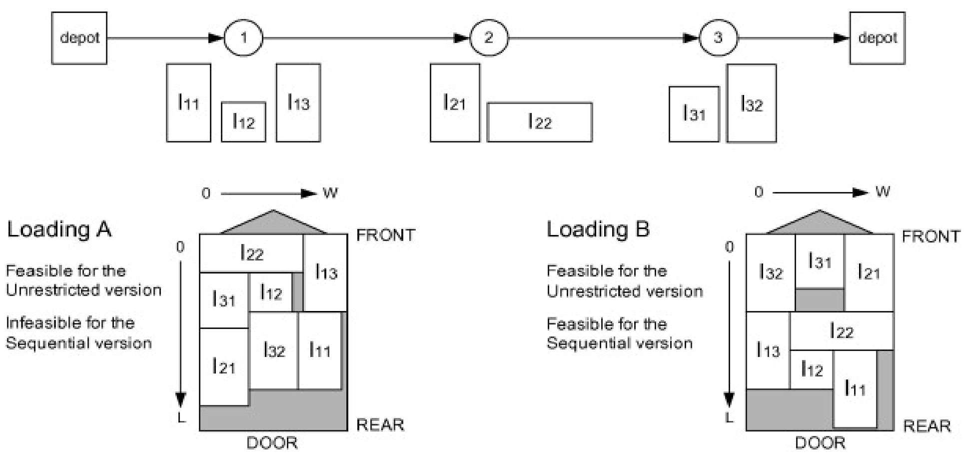
Έστω ότι έχουμε ένα πλήρες γράφημα G=(V,E) όπου V=(0,…..,n) είναι το σύνολο των κόμβων, με την αποθήκη να αντιστοιχεί στον κόμβο 0 ή στον κόμβο 1 και E είναι το σύνολο των τόξων. Για κάθε τόξο e_{ij} υπάρχει ένα κόστος c_{ij} όπου είναι το κόστος μεταφοράς από τον κόμβο i στον κόμβο j. Εντωμεταξύ έχουμε K πανομοιότυπα οχήματα με το κάθε ένα να έχει χωρητικότητα D και οι διαστάσεις της επιφάνειας φόρτωσης είναι W για το πλάτος και L για το μήκος. Ο συνολικός χώρος της επιφάνειας του οχήματος ισούται με A=L∗W.

Κάθε πελάτης i (i=1,…n) έχει ένα σύνολο I_i=(1,2,…m_i) από αντικείμενα m_i όπου το συνολικό βάρος των αντικειμένων του πελάτη i είναι ίσο με d_i. Κάθε αντικείμενο t∈ I_i έχει επίσης συγκεκριμένο πλάτος w_{it} και μήκος l_{it}.

Το συνολικό εμβαδόν όλων των αντικειμένων κάθε πελάτη ισούται με $a_i = \sum_{t=1}^{m_i} l_{it}\, w_{it}$

Ο σκοπός του 2L-CVRP είναι να τοποθετηθούν οι πελάτες i=1,2…,n στις διαδρομές R_k κ=1,2,…K έτσι ώστε να μειωθεί όσο το δυνατών περισσότερο το κόστος μεταφοράς και να είναι εφικτή η φόρτωση των αντικειμένων στην επιφάνεια του οχήματος κάθε διαδρομής. Για να είναι εφικτή η φόρτωση των οχημάτων πρέπει να τηρούνται κάποιοι περιορισμοί.

1. Όλα τα αντικείμενα του κάθε πελάτη πρέπει να φορτωθούν στο ίδιο όχημα, δηλαδή δεν γίνεται να μοιραστούν σε πάνω από ένα οχήματα.
2. Τα αντικείμενα έχουν συγκεκριμένες διαστάσεις και πρέπει να φορτωθούν με τις πλευρές τους παράλληλα στην επιφάνεια φόρτωσης.
3. Κάθε διαδρομή πρέπει να ξεκινάει και να τελειώνει στην αποθήκη.
4. Οι πελάτες πρέπει να επισκέπτονται από το όχημα μια και μόνο φορά.
5. Το συνολικό βάρος των αντικειμένων μίας διαδρομής δεν πρέπει να ξεπερνάει τη χωρητικότητα του οχήματος.
6. Όλα τα αντικείμενα των πελατών θα πρέπει να βρίσκονται εντός της επιφάνειας φόρτωσης του οχήματος.
7. Δεν πρέπει κάποιο αντικείμενο να βρίσκετε πάνω από κάποιο άλλο.



Εικόνα 2.1: Παράδειγμα Unrestricted και Sequential φόρτωσης.

Όπως είχαμε αναφέρει υπάρχουν δυο εκδοχές του 2L-CVRP το Unrestricted 2L-CVRP και το Sequential 2L-CVRP. Η αναφορά που έγινε παραπάνω αφορά την ικανοποίηση της πρώτης εκδοχής, δηλαδή το Unrestricted 2L-CVRP, ενώ για να ικανοποιηθεί και η δεύτερη εκδοχή εκτός από όλα τα παραπάνω θα πρέπει να ισχύει ένας ακόμα περιορισμός. Στο συγκεκριμένο περιορισμό αυτό που μας ενδιαφέρει είναι το ξεφόρτωμα των αντικειμένων του κάθε πελάτη από τα οχήματα, θα πρέπει όλα τα αντικείμενα του πελάτη να μπορούν να ξεφορτωθούν με κινήσεις παράλληλες στο μήκος της επιφάνειας φόρτωσης, με άλλα λόγια δεν θα πρέπει να χρειαστεί να μετακινήσουμε τα αντικείμενα κάποιου άλλου πελάτη j όταν το όχημα θα βρίσκεται στον πελάτη i. Η συγκεκριμένη εκδοχή του 2L-CVRP είναι και αυτή που θα χρησιμοποιήσουμε στο πρόβλημα μας.

2.3 Το Πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα (VRPTW)

Το πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα (VRPTW) αφορά την ικανοποίηση ενός συνόλου πελατών γεωγραφικά διασκορπισμένων, των οποίων η εξυπηρέτηση πρέπει να πραγματοποιείται μέσα στα πλαίσια ενός προκαθορισμένου χρονικού διαστήματος το οποίο ονομάζεται και χρονικό παράθυρο. Κάθε πελάτης πρέπει να εξυπηρετηθεί από ένα πλήθος οχημάτων περιορισμένης χωρητικότητας που τη χρονική στιγμή t=0 βρίσκονται στην αποθήκη. Κάθε πελάτης έχει ένα φορτίο που πρέπει να παραλάβει και την απαίτηση αυτό να πραγματοποιηθεί μέσα σε συγκεκριμένο χρονικό παράθυρο το οποίο θέτει ο ίδιος. Σκοπός είναι να βρεθεί ένα σύνολο διαδρομών για τα οχήματα που να ικανοποιεί τις απαιτήσεις των πελατών αλλά και τους περιορισμούς, ελαχιστοποιώντας το συνολικό κόστος των διαδρομών. Πρόκειται ουσιαστικά για μία επέκταση του CVRP, με ίδια αντικειμενική συνάρτηση ελαχιστοποίησης κόστους, αλλά περιλαμβάνονται επιπλέον οι χρόνοι κατά τους οποίους το όχημα αναχωρεί από την αποθήκη, οι χρόνοι μετάβασης από έναν πελάτη σε κάποιον άλλο, οι χρόνοι εξυπηρέτησης για κάθε πελάτη και οι χρονικές περίοδοι μέσα στις οποίες πρέπει να εξυπηρετηθεί ο κάθε πελάτης.

2.4 Το Πρόβλημα Δρομολόγησης Οχημάτων με Διανομή και Παραλαβή κατά τη Διάρκεια της Διαδρομής (VRPPD)

Το Πρόβλημα Δρομολόγησης Οχημάτων με Διανομή και Παραλαβή κατά τη Διάρκεια της Διαδρομής προκύπτει όταν ο κάθε πελάτης προκειμένου να εξυπηρετηθεί απαιτεί και να παραλάβει αλλά και να παραδώσει αντικείμενα από και προς το όχημα αντίστοιχα. Στο συγκεκριμένο πρόβλημα σε κάθε πελάτη αντιστοιχούν δυο ποσότητες, η ποσότητα που θα παραλάβει ο πελάτης και αυτή που θα παραδώσει. Επιπλέον για κάθε πελάτη καθορίζονται δύο κόμβοι ο κόμβος O_i $και$ D_i , που είναι οι κόμβοι από τους οποίους ξεκινάνε τα προϊόντα που διανέμονται στον πελάτη και οι κόμβοι που καταλήγουν τα προϊόντα που συλλέγονται από τον πελάτη. Ένας περιορισμός του προβλήματος αυτού είναι ότι ο πελάτης πρέπει πρώτα να παραλάβει τα αντικείμενα και στη συνέχεια να παραδώσει τα άλλα, επομένως προκύπτει ότι για κάθε πελάτη i ο κόμβος O_i αν δεν είναι η αποθήκη πρέπει να εξυπηρετηθεί στην ίδια διαδρομή και πριν από τον πελάτη i , καθώς και για κάθε πελάτη i ο κόμβος D_i αν δεν είναι η αποθήκη πρέπει να εξυπηρετηθεί στην ίδια διαδρομή και μετά από τον πελάτη i.

2.5 Το Πρόβλημα Δρομολόγησης Οχημάτων Χωρίς Επιστροφή Στην Αποθήκη (OVRP)

Σε όλα τα προηγούμενα προβλήματα δρομολόγησης μετά την εκτέλεση των δρομολογίων το όχημα πρέπει να επιστρέψει στην αποθήκη μετά την εξυπηρέτηση των πελατών. Στη συγκεκριμένη κατηγορία προβλημάτων το όχημα δεν επιστρέφει στην αποθήκη, συνεχίζει να εκτελεί το δρομολόγιο μέχρι να αδειάσει όλο το φορτίο του και εκεί ολοκληρώνεται η διαδρομή του. Οι διαδρομές που πρόκειται να διανύσουν τα οχήματα είναι προκαθορισμένες, δηλαδή γνωρίζουμε εκ των προτέρων τους πελάτες που θα επισκεφτεί κάθε ένα από αυτά. Συνεπώς, κάθε ένα όχημα ξεκινάει από την αποθήκη διαθέτοντας ποσότητα ίση με το άθροισμα των απαιτήσεων των πελατών που πρόκειται να επισκεφτεί, ενώ ολοκληρώνει τη διαδρομή του με την επίσκεψη και εξυπηρέτηση του τελευταίου προγραμματισμένου πελάτη στην εν λόγω διαδρομή. Ο λόγος που συμβαίνει αυτό είναι ότι τα οχήματα δεν ανήκουν στην εταιρία, οπότε προσπαθεί να χρησιμοποιήσει όσο το δυνατό λιγότερα οχήματα για την εξυπηρέτηση των πελατών με στόχο να μπορέσει να ελαχιστοποιήσει το κόστος ενοικίασης από κάποια εξειδικευμένη εταιρία. Αξίζει να αναφερθεί ότι πολλές φορές κρίνεται προτιμότερη η ενοικίαση οχημάτων από τους προμηθευτές, καθώς επιτυγχάνεται μικρότερο κόστος κατά τη διανομή των αγαθών.

ΚΕΦΑΛΑΙΟ 3

ΑΛΓΟΡΙΘΜΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΠΡΟΒΛΗΜΑΤΩΝ

3.1 Απλοί Ευρετικοί Αλγόριθμοι (Heuristics)

Οι ευρετικοί αλγόριθμοι χρησιμοποιούνται συνήθως όταν οι γνωστές μέθοδοι είναι πολύ δύσκολο να βρουν τη βέλτιστη λύση του προβλήματος λόγω της πολύ μεγάλης πολυπλοκότητας του προβλήματος. Οι δύο στόχοι είναι η υλοποίηση αλγορίθμων με αποδεκτούς (καλούς) χρόνους τρεξίματος αλλά και την εύρεση καλής ποιότητας λύσεων. Οι ευρετικοί αλγόριθμοι παρουσιάζουν περιορισμένη σχετικά εξερεύνηση του χώρου όπου αναζητούνται οι λύσεις και τα αποτελέσματα που προκύπτουν δεν είναι τόσο καλά αλλά δεν έχουν μεγάλο χρόνο υπολογισμού.

Για κάθε πρόβλημα βελτιστοποίησης δεν υπάρχει μόνο ένας ευρετικός αλγόριθμος που δίνει τη βέλτιστη λύση ,ενδέχεται όμως στο ίδιο πρόβλημα για ορισμένες τιμές των παραμέτρων να παρέχει καλύτερες λύσεις κάποιος άλλος ευρετικός αλγόριθμος. Σε ορισμένα προβλήματα είναι αδύνατον να βρούμε τη βέλτιστη λύση σε ικανοποιητικό χρόνο.

Διάφορες κατηγορίες ευρετικών αλγορίθμων είναι:

3.1.1 Αλγόριθμος απληστίας (Greedy Algorithms)

Οι αλγόριθμοι απληστίας προσπαθούν να οδηγήσουν σε μια εφικτή λύση του προβλήματος, αλλά πολλές φορές χωρίς πολύ καλά αποτελέσματα γιατί είναι μυωπικοί αλγόριθμοι, δηλαδή βλέπουν μόνο μπροστά.

Τα περισσότερα προβλήματα δρομολόγησης οχημάτων έχουν n κόμβους, ένας αλγόριθμος απληστίας λειτουργεί σταδιακά, χρησιμοποιώντας ένα από τα δεδομένα (κόμβους) κάθε φορά, με στόχο να το εισάγει σε ένα υποσύνολο που ικανοποιεί τους περιορισμούς δημιουργώντας έτσι μια εφικτή λύση. Η λύση κατασκευάζεται τμηματικά, βρίσκοντας δηλαδή την καλύτερη επιλογή από το σημείο που βρισκόμαστε μια δεδομένη στιγμή και δεν ελέγχει άλλες πιθανές επιλογές κόμβου που πιθανώς να οδηγούσαν σε καλύτερο αποτέλεσμα.

Ένα παράδειγμα αλγόριθμου απληστίας είναι ο αλγόριθμος του πλησιέστερου γείτονα ο οποίος θα χρησιμοποιηθεί για την εύρεση μιας αρχικής λύσης και στο πρόβλημα που θα εξετάσουμε.

Αλγόριθμος Του Πλησιέστερου Γείτονα:

Σε αυτή την ευρετική διαδικασία, ο πωλητής ξεκινά από κάποιο αρχικό κόμβο και μετά επισκέπτεται τον κόμβο που είναι πλησιέστερος στον αρχικό. Από εκεί επισκέπτεται τον πλησιέστερο κόμβο που δεν έχει ακόμη επισκεφτεί, μέχρις ότου όλοι οι κόμβοι περιληφθούν στην διαδρομή του, και τότε επιστρέφει στον αρχικό κόμβο. Τα βασικά βήματα του αλγόριθμου είναι:

1. Ξεκίνα με οποιοδήποτε αρχικό κόμβο (αποθήκη) σαν ξεκίνημα της διαδρομής.
2. Βρίσκουμε τον πλησιέστερο κόμβο που δεν συμπεριλαμβάνεται στη διαδρομή και τον προσθέτουμε στο μονοπάτι.
3. Επαναλαμβάνουμε το 2 μέχρι όλοι οι κόμβοι να ανήκουν στην διαδρομή.

3.1.2 Προσεγγιστικοί Αλγόριθμοι (Approximation Algorithms)

Οι προσεγγιστικοί αλγόριθμοι λειτουργούν σαν τους αλγόριθμους απληστίας, μόνο που για να λύσουν το πρόβλημα χρησιμοποιούν επιπλέον πληροφορία.

3.1.3 Αλγόριθμοι Τοπικής Αναζήτησης (Local Search Algorithms)

Η τοπική αναζήτηση βασίζεται στην αρχαιότερη μέθοδο βελτιστοποίησης, στην μέθοδο δοκιμής και σφάλματος. Η μέθοδος είναι απλή και έχει αποδειχθεί πολύ επιτυχημένη στην πράξη σε ένα πολύ μεγάλο αριθμό προβλημάτων συνδυαστικής βελτιστοποίησης. Οι αλγόριθμοι τοπικής αναζήτησης ξεκινάνε από μία αρχική τυχαία λύση και συνεχίζουν τη διαδικασία αναζήτησης εκτελώντας διαδοχικές παραλλαγές στην αρχική λύση. Μία κίνηση εφαρμόζεται στην ήδη υπάρχουσα λύση για να παραχθεί μία καινούρια από την γειτονιά της υπάρχουσας λύσης. Η απόφαση για το αν θα επιτραπεί κάποια κίνηση εξαρτάται από την ποιότητα των παραγόμενων λύσεων. Σε περίπτωση θετικής απόφασης τότε η υπάρχουσα λύση θα αντικατασταθεί από κάποια γειτονική η οποία θα χρησιμοποιηθεί ως αφετηρία για τις επακόλουθες δοκιμές, διαφορετικά η αναζήτηση συνεχίζεται με την υπάρχουσα λύση.

Δύο βασικές μορφές της Τοπικής Αναζήτησης είναι:

1. λ-opt: ο πιο γνωστός από αυτούς είναι ο 2opt. Αυτή η μέθοδος αποτελείται γενικά από τη διαγραφή 2 ακμών και την επανασύνδεση δύο μονοπατιών με διαφορετικό τρόπο για να καθορίσουμε μια καινούρια διαδρομή.

2. (λ-n)-Exchange: ο πιο γνωστός από αυτούς είναι ο 1-1 ανταλλαγή (Exchange) στο συγκεκριμένο αλγόριθμο 2 κόμβοι αλλάζουν θέση μεταξύ τους. Σημαντική είναι και η υποκατηγορία λ-0 ανταλλαγή γνωστή ως λ-0 επανατοποθέτηση (relocate). Χαρακτηριστικό παράδειγμα είναι 1-0 επανατοποθέτηση, όπου στη συγκεκριμένη μέθοδο ένας κόμβος επανατοποθετείται από το σημείο που βρίσκεται σε κάποιο άλλο σημείο.

3.2 Μεθευρετικοί Αλγόριθμοι (Metaheuristics)

3.2.1 Εισαγωγή στους Μεθευρετικούς αλγόριθμους

Οι μεθευρετικοί αλγόριθμοι με τη χρήση της τοπικής αναζήτησης και ενός υψηλότερου επιπέδου στρατηγικών επιτυγχάνουν να ξεφύγουν από κάποιο τοπικό ελάχιστο για την εύρεση της βέλτιστης λύσης σε σύνθετα προβλήματα. Οι αλγόριθμοι αυτού του τύπου χρησιμοποιούνται κυρίως για την επίλυση πολύπλοκων προβλημάτων που τα δεδομένα είναι λίγα, η βέλτιστη λύση δεν είναι ορατή και η πλήρης καταγραφή των λύσεων είναι αδύνατη, λόγω του μεγάλου εύρους των λύσεων. Όταν όμως δοθεί μια οποιαδήποτε λύση μπορούμε να τη δοκιμάσουμε και να κρίνουμε πόσο ικανοποιητική είναι. Κοινό στοιχείο σχεδόν όλων των μεθευρετικών αλγορίθμων είναι η επεξεργασία τυχαίων αναζητήσεων και γενικότερα παραδοσιακών αλγορίθμων σαν υπό διαδικασίες τους.

Επιπλέον, τα διάφορα είδη των μεθευρετικών αλγορίθμων εφαρμόζονται και αποτελούν αντικείμενο μελέτης σε πολλά διαφορετικά επιστημονικά πεδία. Η κατηγοριοποίησή τους πραγματοποιείται ανάλογα με τις λύσεις τις οποίες χρησιμοποιούν. Δηλαδή, υπάρχουν αλγόριθμοι που βασίζονται σε μία λύση και κάνουν αναζήτηση στη γειτονιά αυτής της λύσης και αλγόριθμοι που έχουν έναν πληθυσμό από λύσεις οι οποίες προσπαθούν να κάνουν αναζήτηση σε όλο τον χώρο των λύσεων. Φυσικά υπάρχουν και υβριδικές μορφές των παραπάνω κατηγοριών (hybrid algorithms). Οι αλγόριθμοι που χρησιμοποιούν μια λύση και κάνουν εξερεύνηση στη γειτονιά αναζήτησης γύρω από τη λύση έχουν πάρα πολύ ισχυρές δυνατότητες εκμετάλλευσης ή εντατικοποίησης (exploitation or intensification) της περιοχής γύρω από τη λύση. Από την άλλη μεριά οι αλγόριθμοι που έχουν πληθυσμό

λύσεων , έχουν πολύ ισχυρές δυνατότητες εξερεύνησης ή διάχυσης (exploration or diversification) σε όλο το χώρο λύσεων .
Η ποιότητα των λύσεων που παράγουν οι μεθευρετικές μέθοδοι είναι αρκετά πιο υψηλή συγκριτικά με τις ευρετικές μεθόδους. Οι μεθευρετικοί αλγόριθμοι που επικεντρώνονται στην αναζήτηση γύρω από κάποιο σημείο που έχει βρεθεί το τοπικό ελάχιστο με τη μέθοδο της τοπικής αναζήτησης μπορούν να διακριθούν σε τέσσερις βασικές κατηγορίες, ανάλογα με τον τρόπο που χρησιμοποιούν για να αποφύγουν το τοπικό ελάχιστο:

1. Επαναληπτικές διαδικασίες που αρχίζουν από διαφορετικές αρχικές λύσεις.

    - Η μέθοδος της διαδικασίας άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης (Greedy Randomized Adaptive Search Procedure).
    - Αλγόριθμοι πολυεναρκτήριας τοπικής αναζήτησης (multistart local search).
    - Αλγόριθμοι της επαναληπτικής τοπικής αναζήτησης (iterated local search).

2. Αλγόριθμοι που δέχονται, υπό κάποιες συνθήκες, γειτονικές κινήσεις που δεν βελτιώνουν τη λύση. Με αυτόν τον τρόπο μπορεί στις επόμενες κινήσεις να βρούμε ένα τοπικό ελάχιστο καλύτερο από το τρέχων.

    - Προσομοιωμένη ανόπτηση (Simulated Annealing).
    - Περιορισμένη αναζήτηση (Tabu search).

3. Αλγόριθμοι που αλλάζουν τη γειτονιά αναζήτησης. Όταν βρεθούν σε τοπικό ελάχιστο αλλάζουν τον αλγόριθμο που χρησιμοποιούν.

    - Αλγόριθμος μεταβλητής γειτονιάς αναζήτησης (Variable Neighborhood Search - VNS).
    - Αλγόριθμος επέκτασης της γειτονιάς αναζήτησης (Expanding Neighborhood Search - ENS).

4. Αλγόριθμοι που αλλάζουν την αντικειμενική συνάρτηση ή τους περιορισμούς του προβλήματος.

    - Αλγόριθμος καθοδηγούμενης τοπικής αναζήτησης (Guided Local Search).

3.2.2 Αλγόριθμος Περιορισμένης Αναζήτησης (Tabu Search Algorithm)

Ο αλγόριθμος της περιορισμένης αναζήτησης είναι μία μέθοδος επίλυσης που συνδυάζει τη διαδικασία τοπικής αναζήτησης και στρατηγική υψηλότερου επιπέδου για την αποφυγή παγίδευσης σε τοπικό ελάχιστο, δηλαδή ανήκει στην κατηγορία των μεθευρετικών αλγορίθμων. Η μέθοδος αρχικά παρουσιάστηκε από τον Fred W. Glover το 1986.

Ο αλγόριθμος περιορισμένης αναζήτησης χρησιμοποιεί ένα ευρετικό αλγόριθμο για να μετακινηθεί από την μία λύση στην άλλη. Υπάρχει όμως ο κίνδυνος να εγκλωβιστεί σε κάποιο τοπικό ελάχιστο και να μην είναι δυνατή η εξερεύνηση άλλων περιοχών για τη βελτίωση της λύσης καθώς είναι πιθανό το ενδεχόμενο να πραγματοποιούνται οι ίδιες εναλλαγές. Το παραπάνω πρόβλημα αποφεύγεται με τη χρήση μνήμης για τις προηγούμενες πραγματοποιηθέντες κινήσεις ώστε να ξεφύγει από το τοπικό ελάχιστο. Για να αποφευχθεί η επανάληψη των προηγούμενων λύσεων, οι κινήσεις που πραγματοποιούνται καταγράφονται σε μία λίστα, η οποία ονομάζεται λίστα περιορισμένων κινήσεων (Tabu List). Οι κινήσεις που θα εισαχθούν στην λίστα απαγορεύεται να επαναληφθούν για συγκεκριμένο αριθμό επαναλήψεων και έτσι αποφεύγεται η επανάληψη των ίδιων λύσεων. Οι συγκεκριμένες κινήσεις παραμένουν στη λίστα για ένα συγκεκριμένο αριθμό επαναλήψεων που το ορίζει ο χρήστης ή προσαρμόζεται δυναμικά ανάλογα με τις απαιτήσεις του προβλήματος. Η ανανέωση της λίστας απαγορευμένων κινήσεων γίνεται δυναμικά και λειτουργεί σύμφωνα με το σύστημα FIFO (First In First Out). Η διαδικασία αυτή ονομάζεται μνήμη μικρής περιόδου (Short Term Memory). Μία άλλη στρατηγική που μπορεί να προστεθεί, είναι η εισαγωγή περιορισμών που αφορούν τη συχνότητα. Σύμφωνα με αυτή, μετράτε ο αριθμός των φορών που μια συγκεκριμένη κίνηση εμφανίζεται κατά τη διάρκεια της αναζήτησης και με βάση τον περιορισμό συχνότητας εμφάνισης μίας κίνησης, γίνεται προσπάθεια να εμποδιστεί η επανάληψη αυτής της κίνησης. Αυτή η διαδικασία ονομάζεται μνήμη μακράς διάρκειας (Long Term Memory), και βοηθάει να εξερευνηθούν κάποιοι αναξιοποίητοι χώροι. Η στρατηγική αυτή, ονομάζεται στρατηγική διάχυσης (diversification). Υπάρχουν όμως και άλλες στρατηγικές. Μία από αυτές έχει στόχο, να παραμείνουν οι μεταβλητές που εμφανίζονται πολύ συχνά, εντός της τελικής λύσης. Η λογική αυτής της στρατηγικής είναι ότι η βέλτιστη λύση θα βρίσκεται κοντά στο υποσχόμενο σημείο, και μπορεί να βρεθεί αν γίνουν κάποιες μικρό αλλαγές με αναζήτηση στη γειτονιά των μεταβλητών που κρατάμε. Η στρατηγική αυτή ονομάζεται στρατηγική εντατικοποίησης (intensification strategies) και η μνήμη ονομάζεται μνήμη μεσαίας περιόδου (Medium Term Memory). Μερικές φορές είναι ευνοϊκότερο να αγνοήσουμε τους περιορισμούς. Αυτό μπορεί να γίνει, όταν μία κίνηση που θα πραγματοποιηθεί να επιφέρει καλύτερο αποτέλεσμα στη συνάρτηση κόστους. Σε αυτή την περίπτωση θα αγνοήσουμε τον περιορισμό και θα πραγματοποιήσουμε την κίνηση αφού θα μας επιφέρει καλύτερο αποτέλεσμα. Η αγνόηση αυτή των περιορισμών ονομάζεται κριτήριο απενεργοποίησης των περιορισμών (aspiration cretirion) και λέμε ότι ενεργοποιήθηκε το κριτήριο φιλοδοξίας. Στην περίπτωση που το αποτέλεσμα όλων των κινήσεων επιφέρει μία κατώτερη λύση, τότε θα επιλέξουμε την πρώτη μη απαγορευμένη κίνηση, όπως περιγράφεται και παραπάνω.

Η περιορισμένη αναζήτηση δε συγκλίνει με φυσικό τρόπο για αυτό απαιτείται ο καθορισμός κάποιων κριτηρίων τερματισμού, τα σημαντικότερα των οποίων είναι:

- Δεν υπάρχουν περαιτέρω εφικτές κινήσεις.
- Μετά την ολοκλήρωση ενός προκαθορισμένου πλήθους επαναλήψεων του αλγορίθμου.
- Μετά την ολοκλήρωση ενός προκαθορισμένου πλήθους επαναλήψεων κατά τις οποίες η βέλτιστη λύση δεν έχει αλλάξει.
- Η τιμή της αντικειμενικής συνάρτησης φτάσει ένα προκαθορισμένο κατώφλι .

Ψευδοκώδικας του Αλγορίθμου Περιορισμένης Αναζήτησης.

Αρχικοποίηση
Κατασκευή μιας αρχικής λύσης S_0
Υπολογισμός κόστους λύσης
S*= S_0 !   αρχικοποίηση βέλτιστης λύσης
F(S*)=f(S_0)
Κύρια φάση
Do whileκάποιο κριτήριο σταματήματος δεν έχει ικανοποιηθεί
Υπολογισμός μια γειτονικής λύσης S’
If f(S’)<f(S*)**then**
S*=S’
f* =f(S’)
endif
Αποθήκευσε την τελευταία κίνηση στη λίστα περιορισμένων υποψηφίων(ταυτόχρονα αν έχει συμπληρωθεί το μέγεθος της λίστας διέγραψε την παλαιότερη)

Κάλεσε κάθε k1 επαναλήψεις την στρατηγική εντατικοποίησης
If f($S_{εντατικοποίησης}$)< f(S*) **then**
S*= $S_{εντατικοποίησης}$
f* =f($S_{εντατικοποίησης}$)
endif
Κάλεσε κάθε k2 επαναλήψεις την στρατηγική διάχυσης
If f($S_{διάχυσης}$)< f(S*)**then**
S*= $S_{διάχυσης}$
f* =f($S_{διάχυσης}$)
endif
enddo
Επέστρεψε τη βέλτιστη λύσηS*

3.2.3 Άλλοι Αλγόριθμοι Βελτιστοποίησης

.

3.2.3.1 Διαδικασία Άπληστης Τυχαιοποιημένης Προσαρμοστικής Αναζήτησης (GRASP)

Η διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης (GRASP) χρησιμοποιείται στην εύρεση προσεγγιστικών λύσεων σε προβλήματα συνδυαστικής βελτιστοποίησης. Η τεχνική αυτή παρέχει μια εφικτή λύση σε κάθε επανάληψη. Οι επαναλήψεις της διαδικασίας σταματούν όταν κάποιο κριτήριο τερματισμού ικανοποιείται. Το τελικό αποτέλεσμα είναι η πιο βέλτιστη λύση που βρέθηκε από όλες τις επαναλήψεις. Κάθε επανάληψη αποτελείτε από δύο φάσεις μια φάση κατασκευής μιας αρχικής λύσης (Construction Phase), και μια διαδικασία τοπικής αναζήτησης (Local Search Phase) για βελτιστοποίηση αυτής της λύσης. Στη φάση κατασκευής, μια τυχαιοποιημένη συνάρτηση απληστίας χρησιμοποιείται για να κατασκευαστεί μια λύση. Αυτή η αρχική λύση στη συνέχεια βελτιώνεται με τη χρήση της διαδικασίας τοπικής αναζήτησης. Η στρατηγική επιλογής του επόμενου στοιχείου βασίζεται στην τυχαία επιλογή από μια λίστα υποψηφίων, που ονομάζεται λίστα περιορισμού των υποψηφίων (Restricted Candidate List) για εισαγωγή στη λύση την οποία κάθε στοιχείο κατατάσσεται βάσει μιας συνάρτησης απληστίας. Ένα μειονέκτημα του GRASP είναι η ανεξαρτησία των επαναλήψεων, ο βασικός αλγόριθμος απορρίπτει πληροφορίες για οποιαδήποτε λύση βρέθηκε και δεν είναι η βέλτιστη.

3.2.3.2 Προσομοιωμένη Ανόπτηση (Simulated Annealing)

Η προσομοιωμένη ανόπτηση (SimulatedAnnealing) προτάθηκε για πρώτη φορά για τη χρησιμοποίηση σε προβλήματα συνδυαστικής βελτιστοποίησης προς επίλυση από τον Kirkpatrick. Η μεθοδολογία του αλγορίθμου είναι η εξής. Ξεκινάμε από μία αρχική λύση, και σε κάθε επανάληψη επιλέγεται μία τυχαία κίνηση. Εάν η κίνηση αυτή βελτιώνει το αποτέλεσμα τότε είναι πάντοτε αποδεκτή. Σε διαφορετική περίπτωση η καινούργια λύση γίνεται αποδεκτή με κάποια πιθανότητα που προέρχεται από τους νόμους της θερμοδυναμικής και πιο συγκεκριμένα από τον τύπο $p(\delta)=e^{-\frac{\delta}{kt}}$, με k να είναι η σταθερά Boltzmann, t η θερμοκρασία και δ η ποσότητα της ενέργειας. Αυτό σημαίνει ότι η προσομοιωμένη ανόπτηση αποδέχεται μία μικρή αύξηση στην αντικειμενική συνάρτηση ελέγχοντας την πιθανότητα αποδοχής p(δ), μέσω της θερμοκρασίας .

Ο αλγόριθμος χρησιμοποιεί ένα πρόγραμμα ανόπτησης για τη συστηματική μείωση της θερμοκρασίας. Καθώς μειώνεται η θερμοκρασία, μειώνεται και η έκταση αναζήτησης του αλγορίθμου ώστε τελικά να συγκλίνει σε κάποιο τοπικό ακρότατο. Οι βασικές στρατηγικές μείωσης της θερμοκρασίας είναι:

- Εκθετική μείωση .
- Δυναμική επιλογή της θερμοκρασίας, σύμφωνα με την προοδευτική ανακάλυψη νέων χαρακτηριστικών στην περιοχή αναζήτησης.
- Καθορισμένου μήκους συνάρτηση, σταθερής μορφής (a(t)=a*t, a σταθερό).
- Λογαριθμική προσέγγιση.

Τυπικά κριτήρια σύγκλισης του αλγορίθμου είναι:

- Η θερμοκρασία έχει φτάσει ένα προκαθορισμένο χαμηλό επίπεδο.
- Ένας αριθμός επαναλήψεων με συνεχή μείωση της θερμοκρασίας έχει περάσει χωρίς αποδοχή της λύσης.
- Η αναλογία των αποδεχτών κινήσεων έχει πέσει κάτω από μία δεδομένη τιμή.
- Ένας μέγιστος αριθμός από επαναλήψεις έχει ολοκληρωθεί.

3.2.3.3 Αλγόριθμος μεταβλητής γειτονιάς αναζήτησης (Variable Neighborhood Search)

Ο αλγόριθμος μεταβλητής γειτονιάς αναζήτησης (VNS) προτάθηκε από τους Hansen και Mladenovic το 1997. Η βασική ιδέα είναι η χρήση πολλών μεθόδων τοπικής αναζήτησης για να βρεθεί μια καλύτερη λύση ή για να ξεφύγει ο αλγόριθμος από κάποιο τοπικό βέλτιστο. Σχεδιάστηκε για την προσέγγιση των λύσεων διακριτών και συνεχών προβλημάτων βελτιστοποίησης και σύμφωνα με αυτό στοχεύει στη λύση προβλημάτων γραμμικού προγραμματισμού, ακέραιου, μικτών προβλημάτων ακέραιου προγραμματισμού, μη γραμμικών προβλημάτων και πολλών ακόμα.

Η μέθοδος αυτή εκμεταλλεύεται το γεγονός ότι διαφορετικές μέθοδοι τοπικής αναζήτησης μπορούν να οδηγήσουν σε διαφορετικά τοπικά βέλτιστα. Είναι ένας στοχαστικός αλγόριθμος, ο οποίος επιλέγει ένα σύνολο γειτονιών (Ni όπου i=1,…,i_{max}) και στη συνέχεια η κάθε επανάληψη αλγορίθμου ακολουθεί τα εξής βήματα: την ανακίνηση (shaking), την τοπική αναζήτηση (local search) και την κίνηση (move). Στην ουσία σε κάθε επανάληψη μία αρχική λύση s΄ δημιουργείται από την τρέχουσα γειτονιά αναζήτησης (για παράδειγμα με τυχαίο τρόπο). Μία διαδικασία τοπικής αναζήτησης εφαρμόζεται στη λύση s΄ με στόχο να παράγουμε τη λύση s΄΄ Έπειτα ελέγχεται η αντικειμενική συνάρτηση και αν η λύση οδηγεί σε βελτίωση του κόστους τότε η καινούρια λύση αντικαθιστά την αρχική λύση. Στη συνέχεια η ίδια διαδικασία αρχίζει από την αρχή με τη χρήση της γειτονιάς N_1 και τη καινούρια λύση s΄΄ που βρήκαμε προηγουμένως. Αν δεν βρεθεί μία καλύτερη λύση τότε ο αλγόριθμος προχωρά στην επόμενη γειτονιά αναζήτησης.

Οι εφαρμογές της VNS αυξάνονται με ταχείς ρυθμούς και αφορούν πολλούς τομείς. Κάποιες από αυτές είναι: η χωροθέτηση εγκαταστάσεων, η ανάλυση συστάδων, ο προγραμματισμός, η δρομολόγηση οχημάτων, ο σχεδιασμός των δικτύων, η τεχνητή νοημοσύνη, η μηχανική, η ομαδοποίηση των προβλημάτων, η βιολογία, η γεωμετρία, ο σχεδιασμός των τηλεπικοινωνιών.

ΚΕΦΑΛΑΙΟ 4

ΑΝΑΛΥΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΚΑΙ ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΈΡΕΥΝΑ

4.1 Γενικά

Το πρόβλημα δρομολόγησης οχημάτων περιορισμένης φόρτωσης και χωρητικότητας το οποίο θα επιλύσουμε με χρήση αλγορίθμου περιορισμένης αναζήτησης (Tabu Search) είναι μια παραλλαγή του κλασικού προβλήματος δρομολόγησης οχημάτων VRP που παρουσιάστηκε για πρώτη φορά από τους Toth και Vigo 2002. Το 2L-CVRP είναι ένας συνδυασμός δύο από τα ποιο σημαντικά NP-hard προβλήματα βελτιστοποίησης στον τομέα τις συνδυαστικής βελτιστοποίησης, του περιορισμένης χωρητικότητας προβλήματος δρομολόγησης οχημάτων και του δυο διαστάσεων προβλήματος συσκευασίας (Bin Packing Problem-2BPP). Στο 2L-CVRP η ζήτηση του κάθε πελάτη καθορίζεται από ένα σύνολο αντικειμένων τα οποία εκτός από συγκεκριμένο βάρος έχουν και συγκεκριμένο πλάτος και μήκος, ενώ και τα οχήματα είναι πανομοιότυπα με την ίδια χωρητικότητα σε βάρος αλλά και με ίδια επιφάνεια φόρτωσης. Σκοπός μας είναι να βρούμε για όλα τα οχήματα τις διαδρομές εκείνες οι οποίες θα μας δώσουν το ελάχιστο κόστος προκειμένου να εξυπηρετηθούν όλοι οι πελάτες αλλά και να φορτωθούν όλα τα αντικείμενα στα οχήματα με επιτυχία.

Το 2L-CVRP είναι ιδιαίτερα σημαντικό τόσο από πρακτικής όσο και από θεωρητικής άποψης. Μπορεί το CVRP να αποτελεί μια ποιο απλή έκδοση από πρακτική άποψη παρόλα αυτά όμως στην καθημερινότητα οι άνθρωποι που ασχολούνται με την εφοδιαστική αλυσίδα πρέπει να λύσουν ταυτόχρονα εκτός από τα προβλήματα της δρομολόγησης και τα προβλήματα της φορτοεκφόρτωσης των προϊόντων-αντικειμένων. Συνήθως λόγω του βάρους των αντικειμένων ή του μεγάλου ύψους τους καθώς και το ότι μπορεί κάποια αντικείμενα να είναι εύθραυστα έχει σαν αποτέλεσμα τα αντικείμενα να μην μπορούν να φορτωθούν το ένα πάνω στο άλλο παρά μόνο το ένα δίπλα στο άλλο. Επομένως βλέπουμε ότι το 2L-CVRP έχει προφανώς εμπορική αξία .

Από θεωρητική άποψη το 2L-CVRP είναι συνδυασμός δυο NP-hard προβλημάτων του CVRP και του 2BPP κάτι το οποίο από μόνο του αποτελεί μια πρόκληση. Υπάρχουν δύο παραλλαγές του 2L-CVRP, στην πρώτη μέθοδο αυτό που μας ενδιαφέρει μόνο είναι ο σωστός τρόπος που θα φορτωθούν τα αντικείμενα στα οχήματα και ονομάζεται Unrestricted, ενώ στη δεύτερη η οποία ονομάζεται Sequential μας ενδιαφέρει τόσο ο σωστός τρόπος φορτώματος όσο και ξεφορτώματος των αντικειμένων από τα οχήματα. Η δεύτερη περίπτωση είναι και αυτή με την οποία θα ασχοληθούμε. Η διαδικασία φόρτωσης των αντικειμένων στα οχήματα έχει πολλά κοινά στοιχεία με το δυο διαστάσεων πρόβλημα συσκευασίας (Bin Packing Problem-2BPP) το οποίο και είναι ένα μη-πολυωνυμικό δύσκολο πρόβλημα (NP-hard) συνδυαστικής βελτιστοποίησης που έχει σαν στόχο να φορτωθούν ένα σύνολο από αντικείμενα συγκεκριμένου πλάτους και μήκους σε όσο το δυνατόν λιγότερα κοντέινερ ίδιου πλάτους και μήκους .

Το πρόβλημα δρομολόγησης οχημάτων περιορισμένης φόρτωσης και χωρητικότητας παρουσιάστηκε για πρώτη φορά από τους Iori, Salazar-Gonzalez, και Vigo (2007) όπου και χρησιμοποίησαν μια μέθοδο η οποία εφάρμοζε τον αλγόριθμο διακλάδωσης και τομής (branch and cut) μέσω του οποίου δημιουργούσαν της διαδρομές των οχημάτων, και έναν αλγόριθμο διακλάδωσης και οριοθέτησης (branch and bound) ο οποίος εγγυόταν τη σωστή φόρτωση των αντικειμένων στα οχήματα. Παρόλα αυτά επειδή η παραπάνω μέθοδος δεν μπορούσε να εφαρμοστεί σωστά σε περιπτώσεις με πάνω από 30 πελάτες και πάνω από 90 αντικείμενα και επομένως όπως καταλαβαίνουμε σε πραγματικές συνθήκες οι κλίμακες των προβλημάτων τείνουν να είναι μεγαλύτερες ο Gendreau (2007) επικεντρώθηκε να λύσει τέτοια προβλήματα με τη χρήση μεθευρετικών αλγορίθμων. Πιο αναλυτικά η μέθοδος που εφάρμοσε χρησιμοποιούσε έναν μεθευρετικό αλγόριθμο περιορισμένης αναζήτησης (Tabu Search) για την δημιουργία των διαδρομών του προβλήματος και για να έχει εφικτή φόρτωση των αντικειμένων έκανε εφαρμογή του δυο διαστάσεων Strip Packing Problem καταφέρνοντας με αυτόν τον τρόπο να λύσει πάνω από 180 διαφορετικές περιπτώσεις προβλημάτων. Τέλος μια ακόμα διαφορετική προσέγγιση έγινε από τους Zachariadi ,Tarantili, και Kiranoudi οι οποίοι πρότειναν έναν μεθευρετικό αλγόριθμο που βασίζεται στις αρχές της Περιορισμένης Αναζήτησης (Tabu Search) και της κατευθυνόμενης τοπικής αναζήτησης (Guided Local Search) προκειμένου να δημιουργήσει ένα σύνολο βέλτιστων διαδρομών, καθώς επίσης χρησιμοποιεί και ένα σύνολο ευρετικών αλγορίθμων για την επίλυση προβλημάτων συσκευασίας (packing heuristics) για να εγγυηθεί τη σωστή φόρτωση των αντικειμένων στα οχήματα.

Καταλήγοντας στο πρόβλημα μας προκειμένου να βρούμε τις διαδρομές εκείνες που θα ακολουθήσουν τα οχήματα και θα έχουν το μικρότερο κόστος κάνουμε χρήση τριών διαφορετικών μεθόδων τοπικής αναζήτησης. Οι μέθοδοι αυτοί σε συνδυασμό με την εφαρμογή του μεθευρετικού αλγορίθμου περιορισμένης αναζήτησης (Tabu Search) θα μας δημιουργήσουν τις διαδρομές αυτές. Επίσης όσο αφορά το κομμάτι της τοποθέτησης των αντικειμένων στα οχήματα τα οποία και θα πρέπει να τοποθετηθούν κατάλληλα προκειμένου να μην παραβιάζονται οι περιορισμοί της φόρτωσης που έχουμε αναφέρει σε προηγούμενο κεφάλαιο ,αυτό πραγματοποιείται με την εφαρμογή μιας μεθόδου που χρησιμοποίει έναν πίνακα δυο διαστάσεων και οι διαστάσεις αυτές είναι ίδιες με αυτές των οχημάτων. Πιο αναλυτικά ο πίνακας αυτός έχει σε κάθε θέση του τον αριθμό μηδέν, όταν πάμε να τοποθετήσουμε ένα αντικείμενο κάποιου πελάτη στο όχημα-πίνακα αφού δούμε πόσες θέσεις θα χρειαστεί να καταλάβει το συγκεκριμένο αντικείμενο στον πίνακα και με την προϋπόθεση οι θέσεις αυτές να μην είναι πιασμένες από κάποιο άλλο αντικείμενο τότε και μόνο τότε μπορούμε να τοποθετήσουμε το αντικείμενο αυτό σε αυτές τις θέσεις οι ο οποίες γεμίζουν με τον αριθμό ένα και με αυτόν το τρόπο γνωρίζουμε ότι στις θέσεις αυτές υπάρχει ήδη κάποιο αντικείμενο. Η διαδικασία επαναλαμβάνεται για όλα τα αντικείμενα που είναι να φορτωθούν σε κάποιο όχημα.

ΚΕΦΑΛΑΙΟ 5

ΠΑΡΟΥΣΙΑΣΗ ΚΑΙ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

5.1 Γενικά

Όπως έχουμε αναφέρει και στα προηγούμενα κεφάλαια στη παρούσα διπλωματική εργασία ασχολούμαστε με την επίλυση του προβλήματος δρομολόγησης οχημάτων περιορισμένης φόρτωσης και χωρητικότητας. Σκοπός του προβλήματος είναι ο καθορισμός των βέλτιστων διαδρομών ενός στόλου οχημάτων που ξεκινούν από μια αποθήκη και εξυπηρετούν όλους τους πελάτες, ώστε να ελαχιστοποιηθεί το συνολικό κόστος. Τα οχήματα του στόλου είναι ομογενή, έχουν δηλαδή ίδια πεπερασμένη χωρητικότητα και διαστάσεις ενώ ξεκινούν από την ίδια αποθήκη. Κάθε πελάτης έχει πεπερασμένη ζήτηση που πρέπει να καλυφθεί. Θεωρούμε ότι το κόστος της διαδρομής προκύπτει αθροιστικά από τις επιμέρους αποστάσεις μεταξύ των πελατών που ανήκουν στην διαδρομή. Πρόκειται για ένα κλειστό πρόβλημα δρομολόγησης οχημάτων όπου τα οχήματα έχουν κοινή αφετηρία αλλά και τερματισμό μια κεντρική αποθήκη. Το συγκεκριμένο πρόβλημα δρομολόγησης οχημάτων όπως έχουμε αναφέρει αποτελεί έναν συνδυασμό του περιορισμένης χωρητικότητας προβλήματος δρομολόγησης οχημάτων (CVRP) και του δύο διαστάσεων προβλήματος συσκευασίας (Bin Packing Problem). Ο συνδυασμός των δυο αυτών προβλημάτων κάνει το πρόβλημα μας ακόμα ποιο ιδιαίτερο και πολύπλοκο και επομένως καθιστά και ποιο δύσκολη τη κατασκευή του αλγορίθμου που θα παρουσιάσουμε παρακάτω.

Το παραπάνω πρόβλημα θα επιλυθεί με εφαρμογή της μεθόδου της Περιορισμένης Αναζήτησης (Tabu Search), η οποία χωρίζετε σε δύο φάσεις. Στη πρώτη φάση βρίσκουμε μια αρχική εφικτή λύση του προβλήματος κάνοντας χρήση του αλγορίθμου του πλησιέστερου γείτονα ενώ στη δεύτερη φάση επέρχεται σταδιακή βελτίωση με τη χρήση μεθόδων τοπικής αναζήτησης. Στη συνέχεια και αφού έχουμε βρει τις διαδρομές των οχημάτων θα πραγματοποιηθεί η φόρτωση των αντικειμένων του κάθε πελάτη στο αντίστοιχο όχημα ελέγχοντας συγχρόνως και τη σωστή τήρηση των περιορισμών που αφορούν το κομμάτι της φόρτωσης και έχουν αναφερθεί σε προηγούμενο κεφάλαιο.

5.2 Αρχικοποίηση Μεταβλητών Και Επεξεργασία Δεδομένων

Για την επίλυση του προβλήματος που περιγράφηκε αναπτύχθηκε κατάλληλος αλγόριθμος με χρήση της γλώσσα προγραμματισμού Fortran, ο οποίος βασίζεται στις αρχές της μεθόδου της περιορισμένης αναζήτησης και η αποτελεσματικότητα αυτού ελέγχεται κάνοντας χρήση παραδειγμάτων από τη βιβλιογραφία. Πιο συγκεκριμένα έγινε εισαγωγή δεδομένων από παραδείγματα για τα οποία υπάρχει δημοσιευμένη μια λύση. Τα δεδομένα αυτά αφορούν το πλήθος και τις συντεταγμένες των θέσεων των πελατών, το πλήθος των οχημάτων που θα χρησιμοποιήσουμε καθώς και την χωρητικότητα αλλά και τις διαστάσεις (πλάτος και μήκος) αυτών, επίσης τη ζήτηση του κάθε πελάτη αλλά και το πλήθος των αντικειμένων του κάθε ενός μαζί με τις διαστάσεις (πλάτος και μήκος) κάθε αντικειμένου. Αναλυτικό παράδειγμα φαίνεται παρακάτω.

```
   20 --- number of customers (no depot)
    6 --- number of vehicles
   32 --- number of items
Capacity - height - width of vehicles
    58       40       20

   Node   - x -     -y -    demand
    0     30.0     40.0      0.0
    1     42.0     41.0     19.0
    2     31.0     32.0     29.0
    3      5.0     25.0     23.0
    4     12.0     42.0     21.0
    5     36.0     16.0     10.0
    6     52.0     41.0     15.0
    7     27.0     23.0      3.0
    8     17.0     33.0     41.0
    9     13.0     13.0      9.0
   10     57.0     58.0     28.0
   11     62.0     42.0      8.0
   12     42.0     57.0      8.0
   13     16.0     57.0     16.0
   14      8.0     52.0     10.0
   15      7.0     38.0     28.0
   16     27.0     68.0      7.0
   17     30.0     48.0     15.0
   18     43.0     67.0     14.0
   19     58.0     48.0      6.0
   20     58.0     27.0     19.0
  Node - num.of items - h - w for each item

    0    0
    1    2         8     7    17     3
    2    1         31     4
    3    2         8    10     7    16
    4    1         20     2
    5    1         22     3
    6    2         8    17    12     8
    7    2         29     2    34     2
    8    1         8    10
    9    2         20     8    23     3
   10    2         22     3     5    15
   11    2         13     9     8     4
   12    1         16     9
   13    2         10     5    14     5
   14    1         31     3
   15    2         30     2    13     7
   16    1         5    13
   17    2         18    10    24     3
   18    2         21     3     8    16
   19    1         6    14
   20    2         27     4    17     4
```

Εικόνα 5.1: Παράδειγμα Δεδομένων .

Αρχικά πρέπει να εξετάσουμε το πρόβλημα και να δούμε ποιες μεταβλητές πρέπει να αρχικοποιήσουμε και ποίες χρειάζεται να κατασκευάσουμε στη συνέχεια. Ένα από τα δεδομένα που έχουμε είναι η αρίθμηση των πελατών, με τον αριθμό 0 να είναι η αποθήκη. Στο πρόβλημα μας όμως η αποθήκη θα αναπαρίσταται με τον αριθμό 1, ο πελάτης 1 με τον αριθμό 2 κ.ο.κ.. Επίσης οι συντεταγμένες του κάθε κόμβου-πελάτη καθώς και η ζήτηση του κάθε ενός βρίσκονται αποθηκευμένες σε αρχεία .txt με ονόματα SYNTETAGMENES και ZHTHSH αντίστοιχα. Με τα συγκεκριμένα αρχεία και τη βοήθεια από δυο υπορουτίνες που έχουμε κατασκευάσει δημιουργούμε τον πίνακα COORD με τις συντεταγμένες των κόμβων- πελατών καθώς και ένα διάνυσμα με το όνομα DEMAND που περιέχει τη ζήτηση του κάθε πελάτη αλλά και της αποθήκης που είναι ίση με μηδέν. Τέλος έχουμε δυο ακόμα αρχεία .txt τα οποία περιέχουν τις διαστάσεις των αντικειμένων του κάθε πελάτη με ονομασίες X και Y με

το πρώτο να περιέχει τα μήκη και το δεύτερο τα πλάτη των αντικειμένων των πελατών.
Αντίστοιχα αποθηκεύουμε και τα υπόλοιπα δεδομένα του προβλήματος στις παρακάτω μεταβλητές.

Na : Αριθμός πελατών

D : Χωρητικότητα οχήματος

L_V : Το μήκος του οχήματος

W_V : Το πλάτος του οχήματος

Τέλος δημιουργήσαμε πίνακα που περιέχει τις αποστάσεις μεταξύ όλων των πελατών με τη βοήθεια του πίνακα COORD που είχαμε κατασκευάσει στην αρχή. Ο πίνακας συμβολίζεται ως COSTS(i , j) όπου i ο προηγούμενος πελάτης και j ο επόμενος. Η απόσταση που υπολογίζεται είναι ευκλείδεια και ο πίνακας είναι συμμετρικός, που σημαίνει ότι η απόσταση που χρειαζόμαστε για να πάμε από ένα πελάτη i στον επόμενο j ισούται με την αντίστροφη από τον j στον i. Επίσης σημαντικό είναι να μην ληφθεί υπόψη η απόσταση του πελάτη από τον εαυτό του και επομένως τα σημεία στον πίνακα COSTS(i,j) η διαγώνιος δηλαδή απειρίζονται. Ο πίνακας γεμίζει με την παρακάτω εξίσωση.

$$\text{COSTS(i,j)}=\sqrt{\left(COORD(i,1)-COORD(j,1)\right)^2+\left(COORD(i,2)-COORD(j,2)\right)^2}$$

5.3 Δημιουργία Αρχικής Εφικτής Λύσης Με Τον Αλγόριθμο Του Πλησιέστερου Γείτονα

Αφού έχουμε εισάγει τα απαραίτητα προς την επίλυση του προβλήματος δεδομένα, το επόμενο βήμα είναι να ξεκινήσει η υλοποίησή του. Όπως προαναφέρθηκε, η πρώτη φάση του αλγορίθμου της Περιορισμένης Αναζήτησης αφορά την εύρεση μιας αρχικής εφικτής λύσης στο πρόβλημα, το οποίο επιτυγχάνεται κάνοντας χρήση κάποιου απλού αλγόριθμου απληστίας. Στην παρούσα διπλωματική εργασία εφαρμόστηκε ο αλγόριθμος του πλησιέστερου γείτονα για την εύρεση της επιθυμητής αρχικής λύσης, δηλαδή ενός συνόλου διαδρομών που θα ικανοποιεί τις απαιτήσεις των πελατών πληρώντας τους περιορισμούς χωρητικότητας.

Βάσει λοιπόν του αλγορίθμου αυτού κάθε φορά επισκεπτόμαστε τον πλησιέστερο πελάτη από το εκάστοτε σημείο που βρισκόμαστε τον οποίο εξυπηρετούμε και συνεχίζουμε με την ίδια λογική στον επόμενο έως ότου εξυπηρετηθούν όλοι οι πελάτες και να επιστρέψουμε τελικώς στο σημείο εκκίνησης. Λόγω των περιορισμών δε δύναται σε μία διαδρομή να εξυπηρετηθούν όλοι οι πελάτες, συνεπώς διαμορφώνεται ένα πλήθος από διαδρομές που εκτελούν διαφορετικά οχήματα. Κάθε φορά που πρόκειται να επισκεφτούμε έναν πελάτη, πέραν του ότι οι απαιτήσεις του δε δύναται να επιφέρουν τον όγκο του φορτίου του οχήματος εκτός ορίων, λαμβάνεται υπόψη το γεγονός ότι χρειάζεται μετά την εξυπηρέτησή του να πραγματοποιηθεί επιστροφή στην αποθήκη, για το λόγο αυτό υπολογίζεται η απόσταση επιστροφής στην αποθήκη από τον εν λόγω πελάτη και αν καθίσταται εφικτή μόνο τότε επιτυγχάνεται η επίσκεψη και κατ' επέκταση εξυπηρέτησή του. Τη στιγμή που ελέγχεται και δεν είναι εφικτή η εξυπηρέτηση του επόμενου πελάτη, πραγματοποιείται έλεγχος για το αν κάποιος από τους υπολειπόμενους μη εξυπηρετημένους πελάτες δύναται να εξυπηρετηθεί προτού επιστρέψει το όχημα στην αποθήκη. Αν πληροί τους περιορισμούς η εν λόγω κίνηση τότε πραγματοποιείται, διαφορετικά επιστρέφουμε απευθείας στην αποθήκη από την θέση που βρισκόμαστε τη δεδομένη χρονική στιγμή.

Το πρώτο βήμα στον αλγόριθμο, είναι η αρχικοποίηση των μεταβλητών που θα χρειαστούμε για την επίλυση του προβλήματος. Αρχικά, δηλώνονται οι μεταβλητές που δηλώνουν την αρχική κατάσταση που επικρατεί πριν ξεκινήσει η διαδικασία επίλυσης του προβλήματος. Όλα τα οχήματα βρίσκονται στην αποθήκη με πλήρη διαθέσιμη χωρητικότητα και χωρίς να έχει εξυπηρετηθεί κανένας πελάτης. Επομένως ορίζουμε τις μεταβλητές .

H_U : Διάνυσμα με την απόσταση της αποθήκης από κάθε πελάτη.

U: Διάνυσμα με την απόσταση (μικρότερη προς μεγαλύτερη) κάθε πελάτη από τον επόμενο.

diadromes: Πίνακας με τις διαδρομές των οχημάτων.

demandp: Μεταβλητή που ελέγχει τη ζήτηση σε κάθε διαδρομή.

routecost: Μεταβλητή που μετράει το κόστος κάθε διαδρομής.

Ο αλγόριθμος ξεκινάει τη διαδικασία κατασκευής των διαδρομών με μόνο περιορισμό να μην παραβιάζεται η χωρητικότητα των οχημάτων.
Κάθε φορά που ένας πελάτης τοποθετηθεί σε μία διαδρομή τότε το κόστος μετάβασης σε αυτόν τον πελάτη από όλους τους υπόλοιπους απειρίζεται προκειμένου να μην ξανά ελεγχτεί. Η διαδικασία επαναλαμβάνεται ως ότου όλοι οι πελάτες τοποθετηθούν στα οχήματα. Από το πίνακα των διαδρομών diadromes, κατασκευάζεται ένας πίνακας diadromes_telikes(i,j), ο οποίος σε κάθε γραμμή του περιλαμβάνει και μία διαδρομή. Για διαδρομές με λιγότερους κόμβους-πελάτες προστίθενται μηδενικά στο τέλος. Επίσης κατασκευάζεται ένα διάνυσμα ROUTE_COST(i) που περιέχει τα κόστη

κάθε διαδρομής, καθώς και μια μεταβλητή με το όνομα totalcost με το συνολικό κόστος όλων των διαδρομών.

5.4 Εύρεση Βέλτιστης Λύσης Με Χρήση Αλγορίθμου Περιορισμένης Αναζήτησης.

Η λύση που βρέθηκε προηγουμένως είναι εφικτή αλλά δεν είναι και η βέλτιστη. Για να βελτιστοποιηθεί το συνολικό κόστος γίνεται χρήση του αλγορίθμου Περιορισμένης Αναζήτησης (Tabu Search). Η μέθοδος παρότι είναι αρκετά περίπλοκη και δύσκολο να εφαρμοστή εξ'ολοκλήρου αποδίδει δίνοντας αρκετά καλά αποτελέσματα. Η μέθοδος στηρίζεται σε εφαρμογές της τοπικής αναζήτησης όπως είναι οι εναλλαγές κόμβων ίδιας διαδρομής, η μεταφορά ενός κόμβου από μία διαδρομή σε μία άλλη και οι ανταλλαγές κόμβων μεταξύ διαφορετικών διαδρομών. Το κύριο χαρακτηριστικό της Περιορισμένης Αναζήτησης είναι ότι χρησιμοποιεί μνήμη για την αποφυγή επανάληψης ίδιων κινήσεων. Για να εφαρμοστεί η μέθοδος απαιτείται η ύπαρξη μιας αρχικής εφικτής λύσης την οποία και καλούμαστε να βελτιώσουμε με διάφορες στρατηγικές .
Σε πρώτο βήμα ορίζουμε τις μεταβλητές από τα υπάρχοντα δεδομένα που θα χρειαστούν στη συνέχεια. Επιπλέων θα χρειαστεί και ένας πίνακας tabu_list τον οποίο μηδενίζουμε και είναι διαστάσεων (l_t,2) και θα τον χρησιμοποιήσουμε σαν ουρά (FIFO). Αυτός είναι ο πίνακας απαγορευμένων κινήσεων στον οποίο αποθηκεύονται οι πρόσφατες κινήσεις οι οποίες απαγορεύεται να πραγματοποιηθούν για ένα συγκεκριμένο πλήθος επαναλήψεων το οποίο ορίζει ο χρήστης .
Στην περίπτωση μας μέρος της Περιορισμένης Αναζήτησης αποτελεί ο 1-1 exchange στην ίδια διαδρομή, ο οποίος δεδομένου μιας αρχικής λύσης που έχει προκύψει από τον αλγόριθμο του Πλησιέστερου Γείτονα εφαρμόζει τοπικές αναζητήσεις επιδιώκοντας την βελτίωση του αποτελέσματος.

5.4.1 Εφαρμογή Τοπικής Αναζήτησης 1-1Exchange.

Η συγκεκριμένη μέθοδος τοπικής αναζήτησης έχει ως βασική ιδέα να αλλάζει θέση μεταξύ δυο τυχαίων πελατών που βρίσκονται στην ίδια διαδρομή προκειμένου να δούμε αν μειώνεται το κόστος της διαδρομής. Πιο συγκεκριμένα με τη βοήθεια δυο εντολών random_number και για συγκεκριμένο αριθμό επαναλήψεων που ορίζεται από τον χρήστη ανάλογα το μέγεθος του προβλήματος, παίρνουμε δυο τυχαίους πελάτες που βρίσκονται μέσα στην διαδρομή και τους αλλάζουμε θέσεις προκειμένου να δούμε αν θα μειωθεί το κόστος της διαδρομής. Ο περιορισμός της ζήτησης δεν λαμβάνεται υπόψη διότι από τη στιγμή που και οι δύο πελάτες βρίσκονται στην ίδια διαδρομή δεν μεταβάλετε.

Αφού γίνει υπολογισμός του καινούργιου κόστους της διαδρομής και το καινούργιο κόστος είναι μικρότερο από το μέχρι τότε βέλτιστο κόστος τότε ελέγχουμε αν η αλλαγή που πάμε να πραγματοποιήσουμε βρίσκεται στην tabu_list. Σε περίπτωση που βρίσκεται στη λίστα απαγορευμένων κινήσεων και το κόστος είναι βέλτιστο τότε ενεργοποιείται το κριτήριο φιλοδοξίας, αλλιώς η αλλαγή απορρίπτεται. Η διαδικασία επαναλαμβάνεται για όλες τις διαδρομές για ένα συγκεκριμένο αριθμό επαναλήψεων όπως και έχουμε αναφέρει.

5.5 Άλλες Μέθοδοι Τοπικής Αναζήτησης Που Εφαρμόσαμε.

Μέθοδος τοπικής αναζήτησης 1-0 επανατοποθέτηση (1-0 relocate)

Με τη μέθοδο 1-0 relocate μπορούμε να πάρουμε έναν κόμβο από τη θέση του και να τον τοποθετήσουμε μεταξύ δύο άλλων. Η μέθοδος relocate βασίζεται στην απλή ιδέα της διαγραφής ενός πελάτη από μια διαδρομή και επανατοποθέτηση του σε μια άλλη διαδρομή με καλύτερο κόστος. Πιο αναλυτικά αφού επιλέξουμε με τη βοήθεια της εντολής random_number δυο τυχαίες διαδρομές από τον πίνακα diadromes τότε με μια άλλη εντολή random_number γίνεται τυχαία επιλογή ενός πελάτη από κάθε μια από αυτές τις διαδρομές. Στη συνέχεια ελέγχουμε αν η ζήτηση της δεύτερης τυχαίας διαδρομής μαζί με τη ζήτηση του πελάτη από τη πρώτη διαδρομή των οποίο και θα μεταφέρουμε δεν ξεπερνάει το D τη χωρητικότητα δηλαδή του οχήματος. Αν η ζήτηση είναι μεγαλύτερη επιστρέφουμε για να πάρουμε δυο άλλες τυχαίες διαδρομές . Σε περίπτωση που η ζήτηση είναι μικρότερη του D τότε γίνεται υπολογισμός του καινούργιου κόστους των διαδρομών και αν το κόστος αυτό είναι μικρότερο από το είδη βέλτιστο τότε πραγματοποιείτε η επανατοποθέτηση. Σε αντίθετη περίπτωση επιστρέφουμε πίσω και ξανά επαναλαμβάνουμε τη διαδικασία. Η διαδικασία επαναλαμβάνεται για ένα συγκεκριμένο αριθμό επαναλήψεων που έχει οριστεί από το χρήστη και εξαρτάται από το μέγεθος του προβλήματος .

Μέθοδος τοπικής αναζήτησης 1-1ανταλλαγή (1-1exchange).

Στο αλγοριθμικό κομμάτι αυτό θα πραγματοποιηθεί η ανταλλαγή κόμβων μεταξύ διαφορετικών διαδρομών. Η ανταλλαγή αυτή είναι μία διαδικασία της τοπικής αναζήτησης 1-1 ανταλλαγή. Στη διαδικασία αυτή επιλέγονται δυο κόμβοι διαφορετικών διαδρομών που ικανοποιούν τους περιορισμούς και γίνεται ταυτόχρονη ανταλλαγή.

Πιο αναλυτικά όπως και στις προηγούμενες περιπτώσεις τοπικής αναζήτησης έτσι και εδώ με τη βοήθεια των εντολών random_number επιλέγουμε τυχαία δυο διαδρομές καθώς και από ένα τυχαίο πελάτη από τις δυο αυτές διαδρομές. Στη συνέχεια αφού ελέγξουμε ότι η ζήτηση στις δυο καινούργιες διαδρομές δεν είναι μεγαλύτερη του D και επομένως δεν παραβιάζεται η χωρητικότητα τότε υπολογίζουμε το καινούργιο κόστος των διαδρομών. Αν το καινούργιο κόστος είναι μεγαλύτερο από το είδη βέλτιστο τότε επιστρέφουμε πίσω για να πάρουμε νέες τυχαίες διαδρομές. Σε αντίθετη περίπτωση αποθηκεύουμε το καινούργιο κόστος κάθε διαδρομής στο διάνυσμα ROUTE_COSTS και τις διαδρομές στον πίνακα diadromes_telikes που έχουν το βέλτιστο κόστος μέχρι εκείνη τη στιγμή. Το συγκεκριμένο κομμάτι του αλγορίθμου επαναλαμβάνεται για συγκεκριμένο αριθμό επαναλήψεων τον οποίο έχει ορίσει ο χρήστης .

5.6 Διαδικασία Φόρτωσης Των Αντικειμένων.

Όπως έχουμε αναφέρει στο συγκεκριμένο πρόβλημα δρομολόγησης οχημάτων εκτός από τον περιορισμό της χωρητικότητας έχουμε και τον περιορισμό της φόρτωσης. Δηλαδή του σωστού τρόπου εισαγωγής και εξαγωγής των αντικειμένων του κάθε πελάτη από τα οχήματα.
Έχοντας ολοκληρώσει το τμήμα του αλγορίθμου με τις εφαρμογές των μεθόδων της τοπικής αναζήτησης αλλά και την εφαρμογή του αλγορίθμου περιορισμένης αναζήτησης πρέπει να κατασκευάσουμε το κομμάτι του αλγορίθμου που αφορά τη φόρτωση των αντικειμένων. Για τη κατασκευή αυτή θα χρησιμοποιήσουμε τον πίνακα diadromes_telikes ο οποίος περιέχει τις τελικές διαδρομές των οχημάτων και οι οποίες μας έδωσαν και το ποιο βέλτιστο κόστος από την εφαρμογή των μεθόδων που εφαρμόσαμε.
Πιο αναλυτικά με τη βοήθεια των αρχείων X και Y τα οποία περιέχουν τις διαστάσεις των αντικειμένων δημιουργούμε τους πίνακες W_axis και L_axis με τις διαστάσεις των αντικειμένων. Στη συνέχεια του αλγορίθμου δημιουργούμε έναν πίνακα diadromes_n3 ο οποίος περιέχει τους πελάτες με τη σειρά που θα μπουν στα οχήματα και αυτό γιατί τα αντικείμενα του πρώτου πελάτη που θα επισκεφτεί το όχημα θα πρέπει να φορτωθούν τελευταία του δεύτερου προτελευταία κ.ο.κ. Επιπλέων γίνεται δημιουργία και ενός πίνακα TRUCK με διαστάσεις L_V και W_V, δηλαδή όσο είναι και οι διαστάσεις των οχημάτων και ο πίνακας γεμίζει με μηδενικά. Ο πίνακας αυτός θα αντιπροσωπεύει το όχημα μας και κάθε φορά που ένα αντικείμενο τοποθετείτε στον πίνακα οι θέσεις που θα καταλαμβάνει στον πίνακα θα παίρνουν τον αριθμό 1. Κάθε φορά που κάποιο αντικείμενο τοποθετείτε στον πίνακα έχει προηγηθεί ο έλεγχος των θέσεων που θα χρειαστούν με τη βοήθεια κάποιων εντολών if οι οποίες ελέγχουν τις θέσεις αυτές για να δουν αν περιέχουν όλες τον αριθμό 0. Όταν όλα τα αντικείμενα ενός πελάτη τοποθετηθούν στον πίνακα-όχημα τότε ο αλγόριθμος μας προχωράει στον επόμενο πελάτη της διαδρομής και επαναλαμβάνει τη διαδικασία.

Προκειμένου ένας πελάτης να μπορέσει να παραμείνει στη διαδρομή που έχει τοποθετηθεί θα πρέπει όλα του αντικείμενα να μπορούν να φορτωθούν στο ίδιο όχημα. Στη περίπτωση που αυτό δεν καταστεί δυνατό τότε ο αλγόριθμος μας μεταφέρει τον πελάτη αυτόν καθώς και όσους είναι μετά από αυτόν σε ένα διάνυσμα diad_den το οποίο και περιέχει τους πελάτες που τα αντικείμενα τους δεν χώρεσαν όλα στα οχήματα. Σε αυτό το σημείο ο αλγόριθμος μας υπολογίζει το κόστος των διαδρομών με τους πελάτες που τα αντικείμενα τους φορτώθηκαν στα οχήματα και ξανά φτιάχνει τον πίνακα diadromes_telikes με τις νέες διαδρομές. Τέλος προκειμένου να δημιουργήσουμε νέες διαδρομές για τους πελάτες που δεν χώρεσαν στα οχήματα ο αλγόριθμος μας επιστρέφει στην αρχή του προγράμματος και επαναλαμβάνει τη διαδικασία από την αρχή. Πιο συγκεκριμένα βρίσκουμε μια νέα αρχική λύση με τη μέθοδο του πλησιέστερου γείτονα για τους πελάτες που βρίσκονται στο διάνυσμα diad_den και γίνεται βελτίωση αυτής της λύσης με τις μεθόδους τοπικής αναζήτησης που έχουμε αναφέρει, καθώς και εφαρμογή της περιορισμένης αναζήτησης. Για τις καινούργιες διαδρομές ο αλγόριθμός μας επαναλαμβάνει πάλι τη διαδικασία της φόρτωσης, η διαδικασία επαναλαμβάνεται ως ότου όλοι οι πελάτες τοποθετηθούν σε κάποια διαδρομή. Καταλήγοντας ο αλγόριθμός μας επιστρέφει τον πίνακα diadromes_telikes με όλες τις διαδρομές που θα χρειαστεί να γίνουν καθώς και το συνολικό βέλτιστο κόστος των διαδρομών αυτόν.

ΚΕΦΑΛΑΙΟ 6

ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ

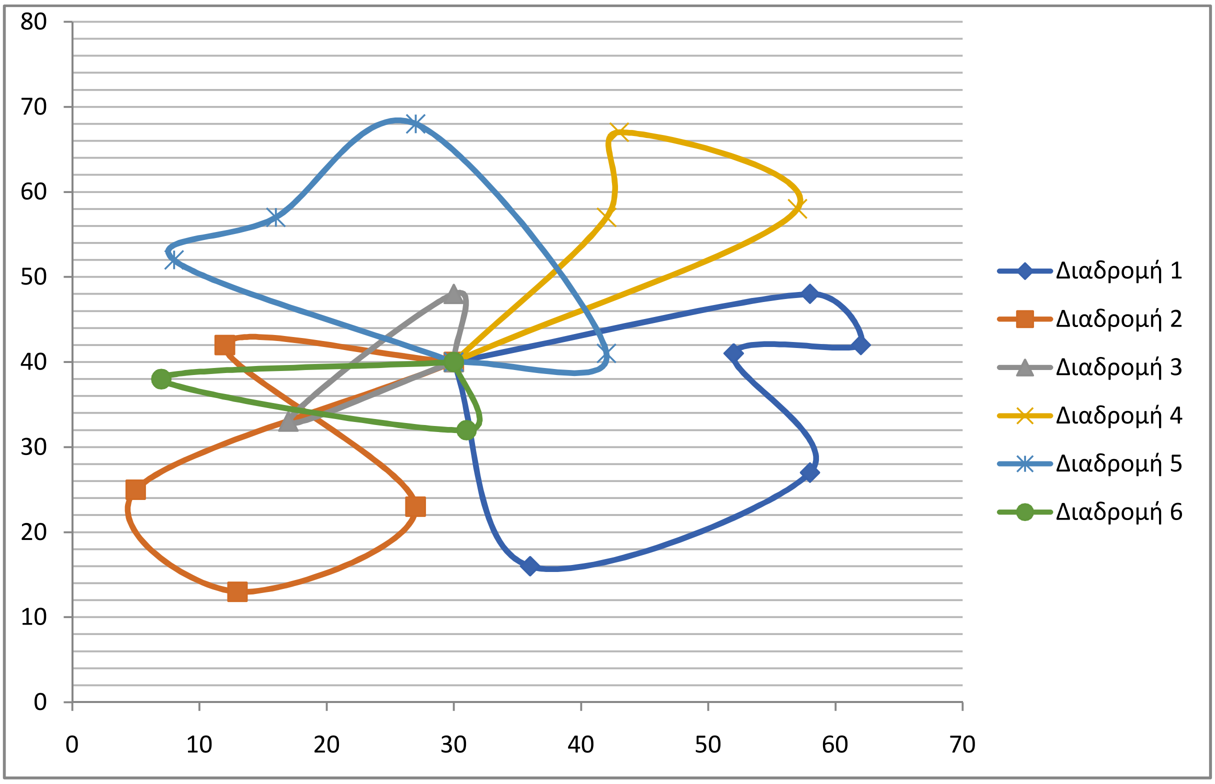
6.1 Παρουσίαση και Ανάλυση Αποτελεσμάτων.

Προκειμένου να αξιολογήσουμε τα αποτελέσματα του αλγορίθμου τρέξαμε μια σειρά από πρότυπα προβλήματα αναφοράς (benchmark problems) και πιο συγκεκριμένα επιλύθηκαν 9 προβλήματα δρομολόγησης από ένα σύνολο 180 διαφορετικών προβλημάτων που αφορούν προβλήματα περιορισμένης φόρτωσης και χωρητικότητας και τα οποία κατασκευάστηκαν από τους Iori και Gendreau. Σε κάθε πρόβλημα η τοποθεσία των κόμβων καθορίζεται από τις καρτεσιανές τους συντεταγμένες. Ακόμα σε όλα τα προβλήματα τα οχήματα έχουν περιορισμένη χωρητικότητα , όπως επίσης έχουν και το ίδιο πλάτος και μήκος. Τέλος σε κάθε κόμβο αντιστοιχεί ένα σύνολο από αντικείμενα με συγκεκριμένα πλάτη και μήκη.

Δεδομένα Προβλημάτων				
Πρόβλημα	**Πελάτες**	**Χωρητικότητα**	**Αντικείμενα**	**Οχήματα**
Πρόβλημα 1	**20**	**58**	**32**	**6**
Πρόβλημα 2	**21**	**6000**	**37**	**4**
Πρόβλημα 3	**25**	**48**	**61**	**8**
Πρόβλημα 4	**32**	**8000**	**65**	**7**
Πρόβλημα 5	**40**	**60**	**60**	**14**
Πρόβλημα 6	**50**	**160**	**82**	**11**
Πρόβλημα 7	**75**	**100**	**152**	**17**
Πρόβλημα 8	**100**	**112**	**152**	**19**
Πρόβλημα 9	**120**	**200**	**183**	**23**

6.2 Αναλυτικά Αποτελέσματα των Προβλημάτων.

6.2.1 Αποτελέσματα Προβλήματος 1

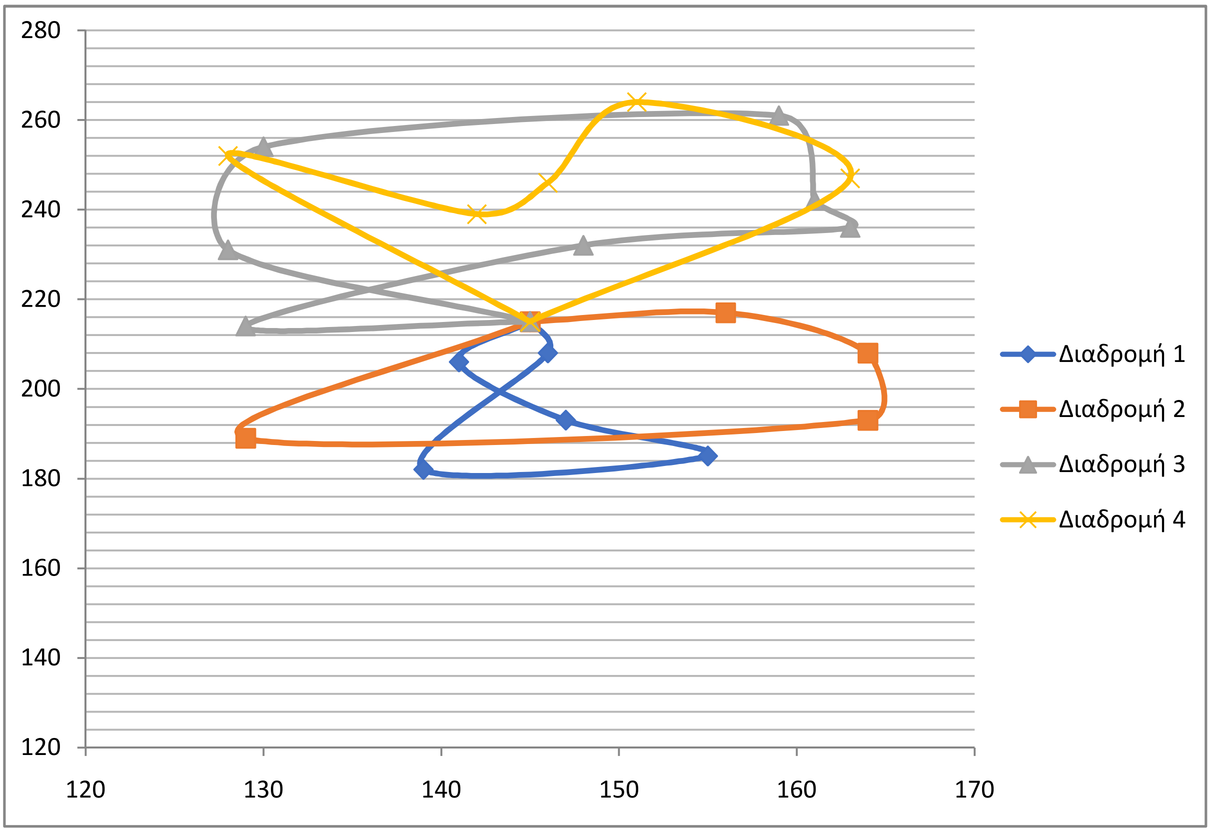


Συνολική Απόσταση: 485,48

Διαδρομές

1	20	12	7	21	6	1
1	4	10	8	5	1	
1	18	9	1			
1	11	19	13	1		
1	2	17	14	15	1	
1	16	3	1			

6.2.2 Αποτελέσματα Προβλήματος 2

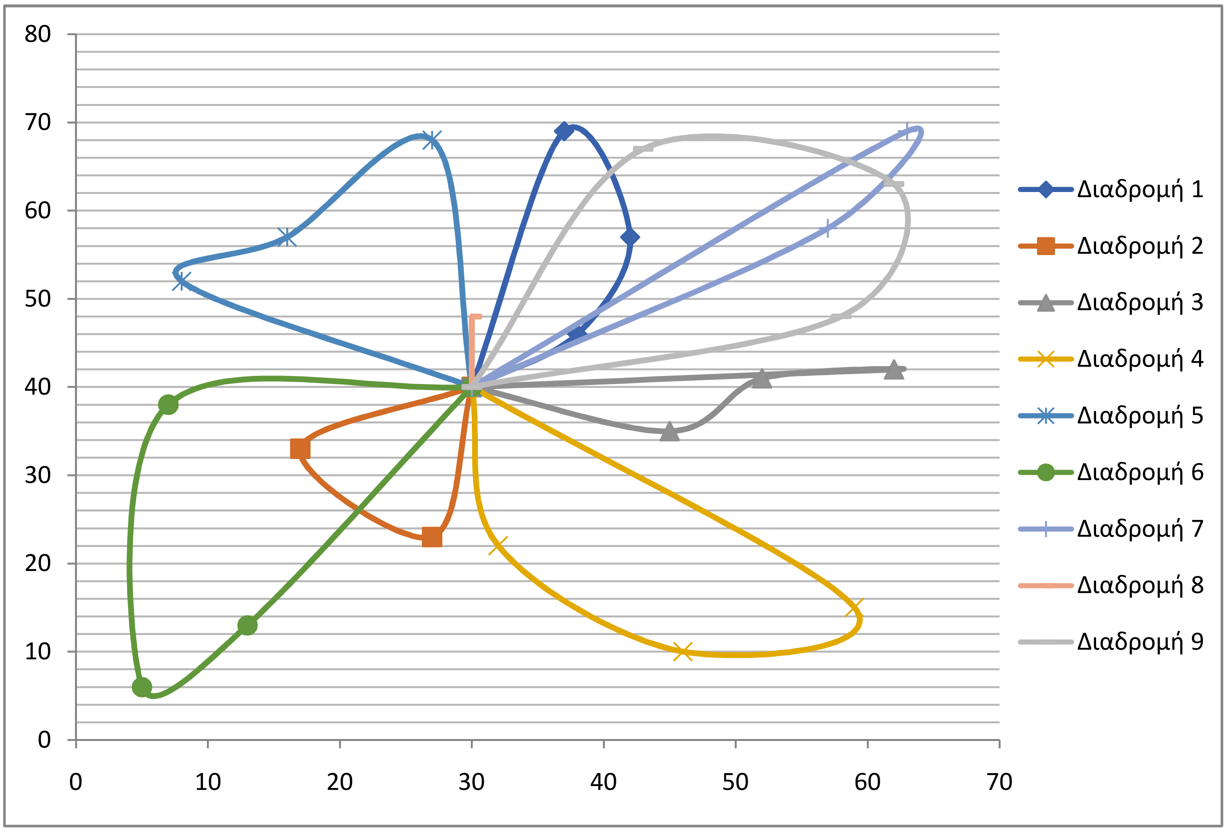


Συνολική Απόσταση: 493,24

Διαδρομές

1	15	22	21	18	17	1		
1	20	19	16	13	1			
1	14	11	10	8	3	4	12	1
1	5	9	7	2	6	1		

6.2.3 Αποτελέσματα Προβλήματος 3

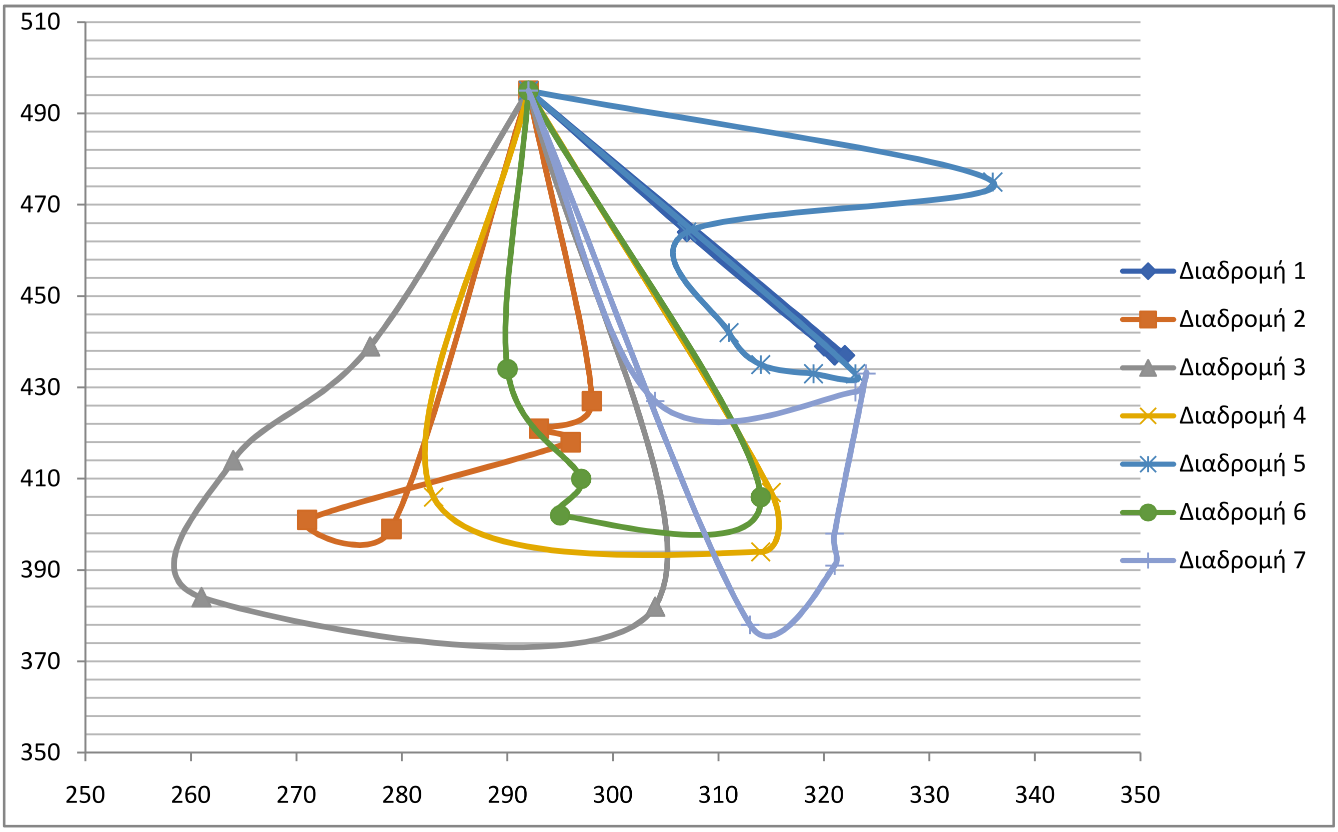


Συνολική Απόσταση: 710,81

Διαδρομές

1	24	2	7	1
1	23	19	25	1
1	10	9	12	1
1	11	26	5	1
1	6	22	1	
1	13	1		
1	15	21	14	1

6.2.4 Αποτελέσματα Προβλήματος 4

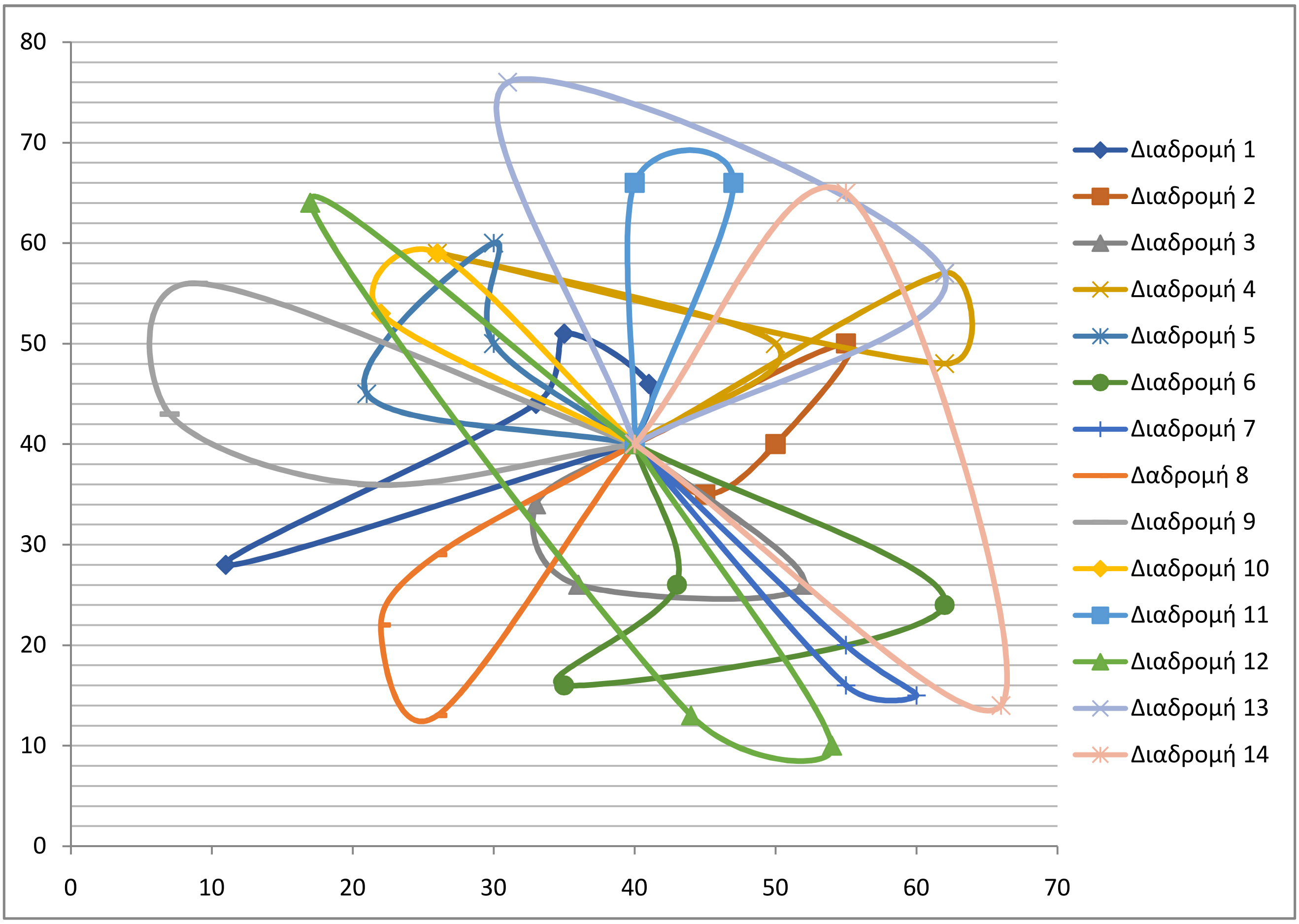


Συνολική Απόσταση: 1499.03

Διαδρομές

1	4	6	7	8	1		
1	2	15	16	29	28	1	
1	31	30	17	25	1		
1	27	23	19	1			
1	5	3	13	12	33	9	1
1	32	18	26	20	1		
1	14	11	10	22	21	24	1

6.2.5 Αποτελέσματα Προβλήματος 5

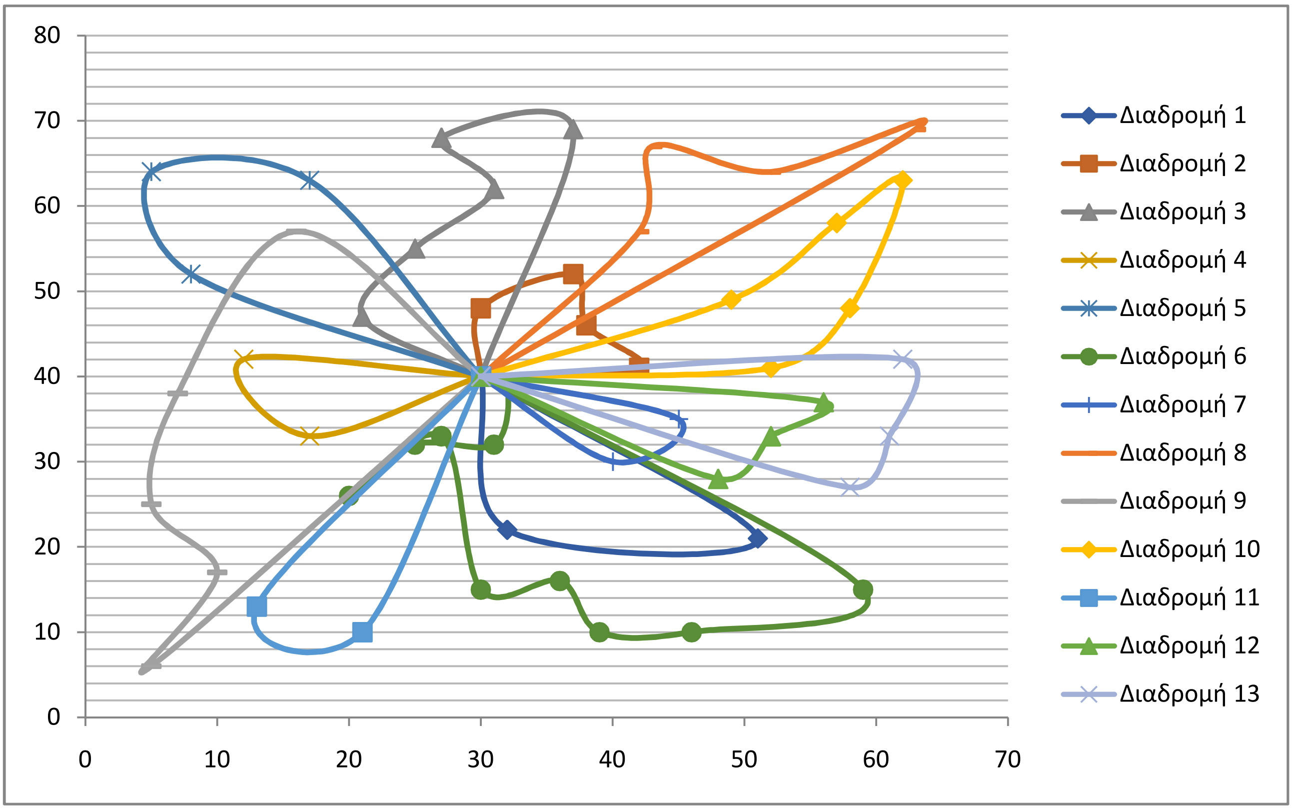


Συνολική Απόσταση: 1067,38

Διαδρομές

1	27	13	18	24	1
1	36	35	5	1	
1	30	3	7	1	
1	8	9	20	14	1
1	4	40	41	1	
1	16	29	31	1	
1	28	38	6	1	
1	23	2	34	1	
1	17	25	19	1	
1	33	10	1		
1	11	39	1		
1	26	22	37	1	
1	15	32	1		
1	12	21	1		

6.2.6 Αποτελέσματα Προβλήματος 6

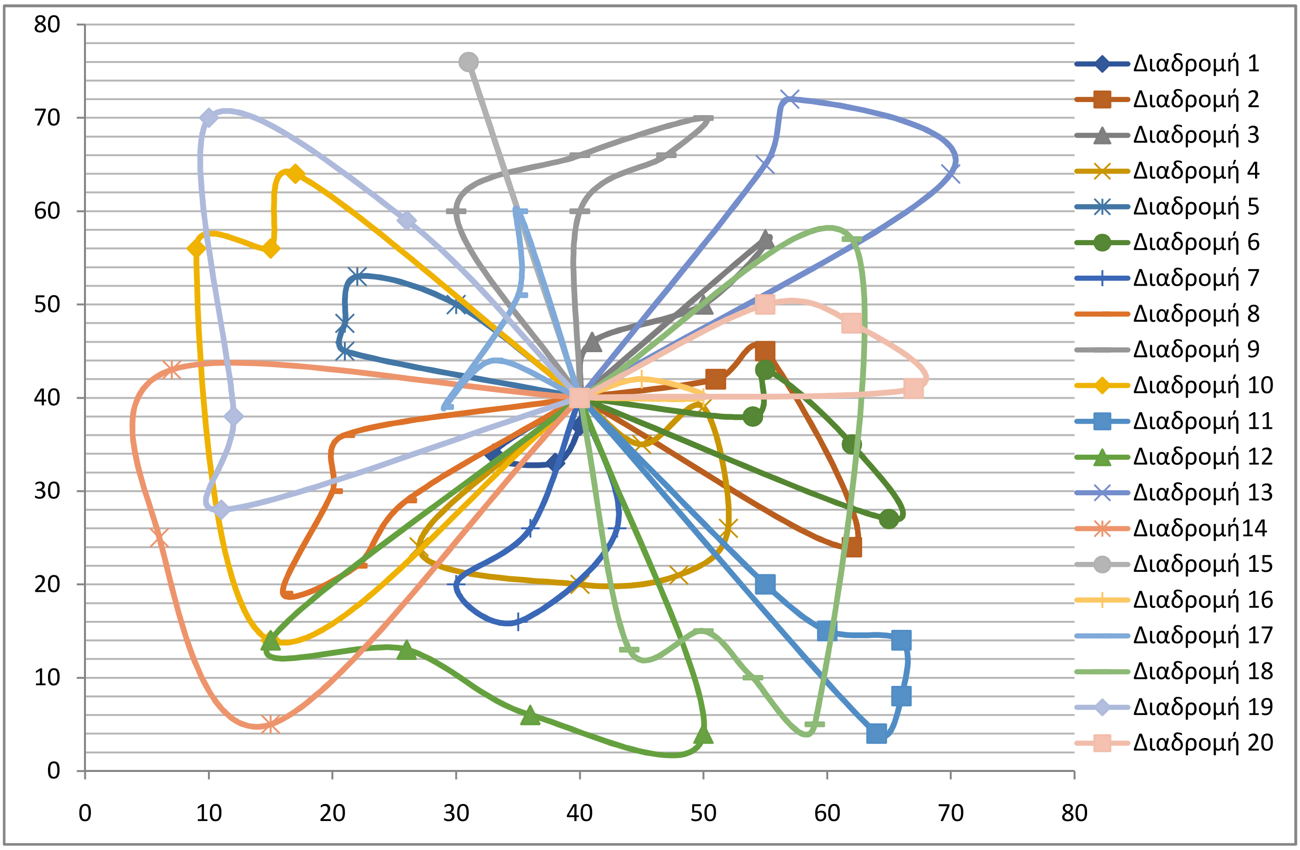


Συνολική Απόσταση: 963,59

Διαδρομές

1	38	11	1								
1	28	2	33	12	1						
1	7	49	9	27	32	1					
1	19	15	1								
1	25	44	8	1							
1	47	13	48	5	18	45	16	46	34	40	1
1	6	39	1								
1	23	29	4	37	1						
1	24	26	14	42	41	1					
1	3	21	36	30	17	1					
1	20	43	1								
1	50	10	51	1							
1	31	35	22	1							

6.2.7 Αποτελέσματα Προβλήματος 7



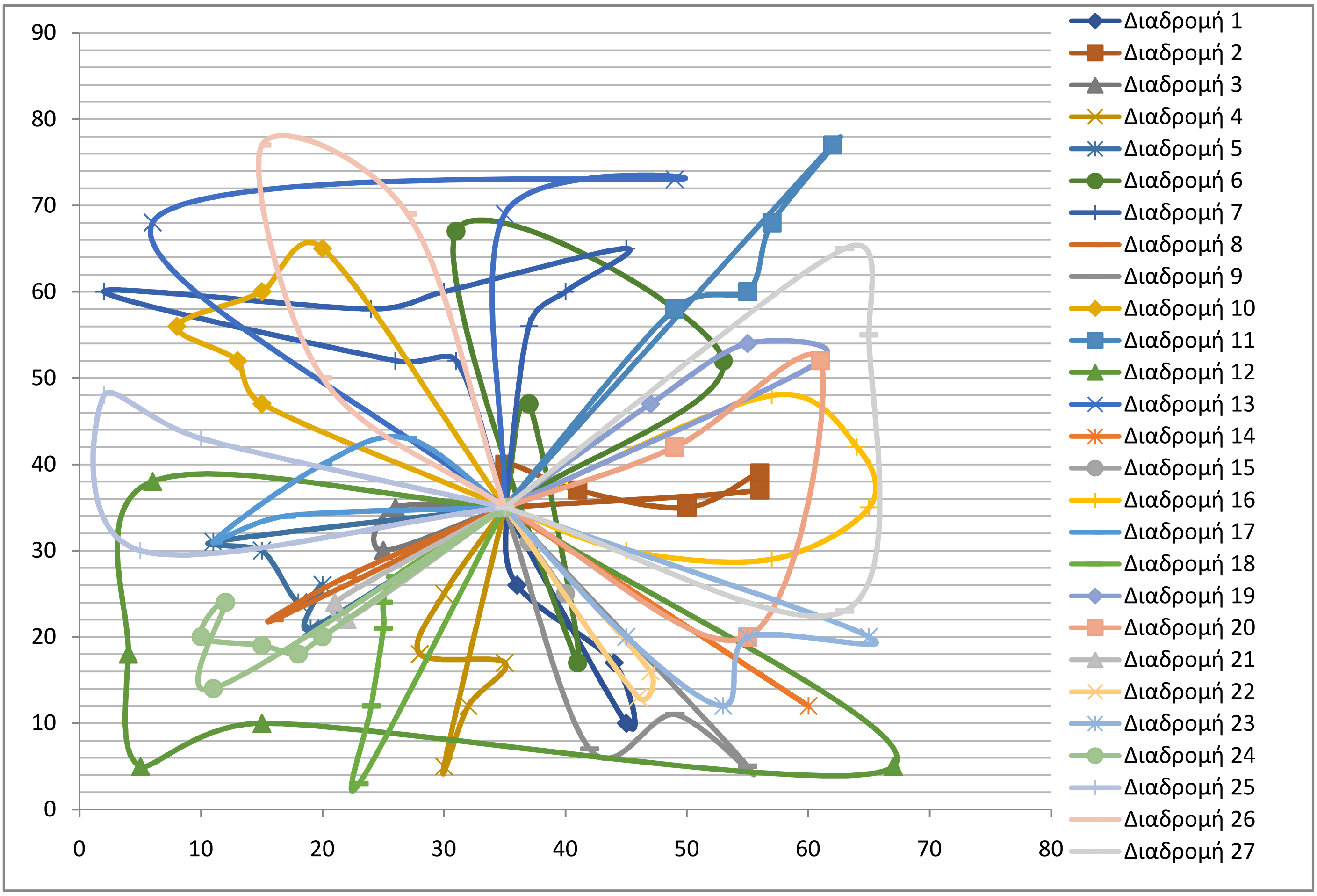
Συνολική Απόσταση: 1508,35

Διαδρομές

1	76	69	7	1			
1	16	9	47	1			
1	27	8	54	1			
1	74	75	49	30	46	5	1
1	41	33	45	4	1		
1	53	28	14	58	1		
1	31	29	63	3	1		
1	17	64	44	2	34	1	
1	59	39	66	11	40	1	
1	42	19	51	26	1		
1	6	38	21	71	61	1	
1	43	23	62	70	1		
1	12	67	60	1			
1	65	57	25	1			
1	32	1					
1	68	35	1				

1	18	52	13	73	1		
1	15	72	37	48	22	1	
1	10	56	50	24	1		
1	55	20	36	1			

6.2.8 Αποτελέσματα Προβλήματος 8



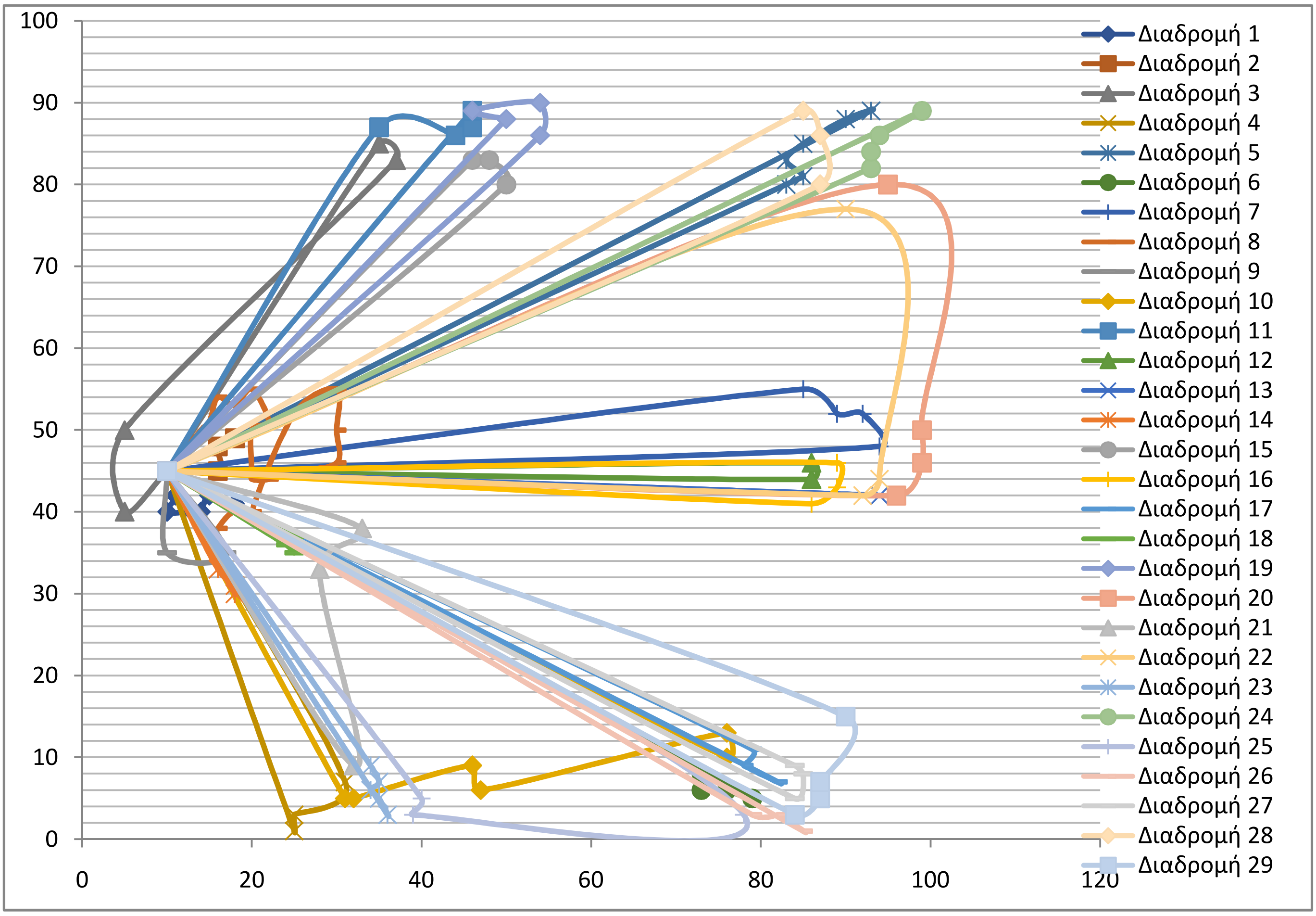
Συνολική Απόσταση: 2035,56

Διαδρομές

1	59	74	23	1					
1	28	29	13	69	81	1			
1	90	7	1						
1	14	88	3	58	16	1			
1	99	100	94	6	85	1			
1	70	2	91	34	1				
1	32	89	37	63	11	21	31	71	1
1	97	86	1						
1	42	76	24	1					
1	83	49	48	20	12	1			
1	52	10	72	66	1				

1	68	15	39	87	46	1			
1	50	67	33	1					
1	40	1							
1	54	41	1						
1	27	55	25	30	80	1			
1	53	19	84	61	1				
1	95	96	98	43	44	1			
1	51	82	79	1					
1	77	78	4	1					
1	60	93	1						
1	73	75	1						
1	22	57	5	26	1				
1	38	101	92	17	62	45	1		
1	18	47	9	1					
1	8	65	64	1					
1	56	35	36	1					

6.2.9 Αποτελέσματα Προβλήματος 9



Συνολική Απόσταση: 3856,30

Διαδρομές

1	89	83	112	87	88	93	1
1	96	103	102	1			
1	120	121	68	70	1		
1	6	3	2	1			
1	53	55	54	56	61	64	1
1	18	17	20	26	1		
1	41	44	46	49	1		
1	106	107	108	105	104	101	100
97	94	95	98	116	111	99	117
92	91	90	86	113	1		
1	82	85	1				
1	4	5	9	13	22	21	1
1	71	72	75	73	1		
1	39	40	1				
1	47	1					
1	118	114	84	1			
1	74	77	69	1			
1	38	42	43	1			
1	27	24	29	1			
1	19	119	115	1			
1	78	79	76	81	80	1	
1	66	52	51	50	1		
1	110	109	7	1			
1	45	48	60	1			
1	8	10	11	12	16	1	
1	62	63	65	67	1		
1	14	15	23	1			
1	25	28	34	1			
1	33	36	32	1			
1	57	59	58	1			
1	31	35	37	30	1		

6.3 Συμπεράσματα

Στη προηγούμενη ενότητα έγινε παρουσίαση των αποτελεσμάτων του αλγορίθμου για τα διάφορα προβλήματα αναφοράς που επιλύσαμε. Παρακάτω ακολουθεί η σύγκριση τους με τις αντίστοιχες παγκόσμιες βέλτιστες τιμές και παρουσιάζεται σε ποσοστιαίες μονάδες η απόκλιση από τα δικά μας αποτελέσματα καθώς και η διαφορά στον αριθμό των οχημάτων που χρειάστηκαν.

Πρόβλημα	Πελάτες	Αντικείμενα	Αποτελέσματα Αλγορίθμου	Οχήματα στον Αλγόριθμο	Παγκοσμίως Βέλτιστες Τιμές	Βέλτιστα Οχήματα	Απόκλιση %
Πρόβλημα 1	**20**	**32**	**485,48**	**6**	**444,62**	**6**	**9,18%**
Πρόβλημα 2	**21**	**37**	**493,24**	**4**	**396,36**	**4**	**24,40%**
Πρόβλημα 3	**25**	**61**	**710,81**	**10**	**624,06**	**8**	**13,90%**
Πρόβλημα 4	**32**	**65**	**1499,03**	**7**	**1101,49**	**7**	**36,09%**
Πρόβλημα 5	**40**	**60**	**1067,38**	**14**	**866,5**	**14**	**23,13%**
Πρόβλημα 6	**50**	**82**	**963,59**	**13**	**802.94**	**11**	**20%**
Πρόβλημα 7	**75**	**152**	**1508,35**	**20**	**1248,43**	**17**	**20,80%**
Πρόβλημα 8	**100**	**152**	**2035,56**	**27**	**1524,22**	**19**	**33,50%**
Πρόβλημα 9	**120**	**183**	**3856,3**	**29**	**2858,53**	**23**	**34,90%**

Από τον παραπάνω πίνακα με τη σύγκριση των βέλτιστων τιμών του αλγορίθμου παρατηρούμε ότι οι λύσεις που παράγονται από τον αλγόριθμο δεν είναι πάρα πολύ ικανοποιητικές, με την απόκλιση σε κάποιο παράδειγμα να φτάνει στο 36,09%. Αναλυτικότερα παρατηρούμε ότι :

- Ο μέσος όρος απόκλισης από τις παγκοσμίως βέλτιστες λύσης είναι στο 24%, με το εύρος απόκλισης να κυμαίνεται από 9,18% στο ελάχιστο έως 36,09% το μέγιστο.
- Τα οχήματα που χρησιμοποιήσαμε είναι κατά μέσο όρο 2,3 οχήματα παραπάνω από το παγκοσμίως βέλτιστο αριθμό οχημάτων .

Εκεί που πραγματικά ο αλγόριθμος μας αντιμετωπίζει το μεγαλύτερο πρόβλημα με αποτέλεσμα σε κάποια παραδείγματα η απόκλιση από τη βέλτιστη τιμή , καθώς και ο αριθμός των οχημάτων που χρειαστήκαμε να είναι μεγαλύτερα είναι το σημείο που πραγματοποιείται η φόρτωση των αντικειμένων στα οχήματα. Δυστυχώς η διαδικασία της φόρτωσης δεν πραγματοποιείται όπως θα θέλαμε με αποτέλεσμα διάφορες περιοχές της επιφάνειας φόρτωσης να μένουν αχρησιμοποίητες. Αυτό έχει ως αποτέλεσμα τα αντικείμενα πολλών πελατών να μην χωρούν στις διαδρομές στις οποίες και έχει δρομολογηθεί να τοποθετηθούν μετά και την εφαρμογή της μεθόδου της Περιορισμένης Αναζήτησης . Έτσι αναγκαζόμαστε να μετακινήσουμε τους πελάτες- κόμβους σε άλλες διαδρομές με αποτέλεσμα να δημιουργούνται παραπάνω διαδρομές κάτι το οποίο αυξάνει το τελικό κόστος .

Καταλήγοντας για πιθανή μελλοντική βελτίωση της ποιότητας του αλγορίθμου περιορισμένης αναζήτησης μπορούμε να προτείνουμε να γίνει χρήση μερικών ακόμα αλγορίθμων τοπικής αναζήτησης σε συνδυασμό με τους ήδη υπάρχοντες, για παράδειγμα 2-2 exchange, 3-opt, 2-0 relocate. Επίσης θα μπορούσαμε να χρησιμοποιήσουμε κάποιον άλλο αλγόριθμο για τη δημιουργία της αρχικής λύσης. Στο κομμάτι όμως που πρέπει να δοθεί περισσότερη σημασία για να έχουμε καλύτερα αποτελέσματα είναι αυτό της φόρτωσης των αντικειμένων. Θα πρέπει να γίνεται καλύτερη διαχείριση της επιφάνειας φόρτωσης προκειμένου να αποφεύγονται οι μετακινήσεις των πελατών – κόμβων σε άλλες διαδρομές .

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Ιωάννης Μαρινάκης, Μαγδαληνή Μαρινάκη, Το Πρόβλημα Δρομολόγησης Οχημάτων στην Εφοδιαστική Αλυσίδα. 2017

2. Ευρετικοί και Μεθευρετικοί αλγόριθμοι , Ι. Μαρινάκης.

3. Ιωάννης Μαρινάκης, Μυγδάλας Αθανάσιος, Σχεδιασμός και Βελτιστοποίηση της εφοδιαστικής αλυσίδας. 2008

4. Iori, M., Salazar Gonzalez, J.J., Vigo, D., 2007. An exact approach for the Vehicle Routing Problem with two dimensional loading constraints. Transportation Science 41 (2), 253–264

5. Emmanouil E. Zachariadis , Christos D.Tarantilis , Chris T.Kiranoudis , A Guided Tabu Search for the Vehicle Routing Problem with two-dimensional loading constraints. European Journal of Operational Research 195 (2009) 729–743

6. Michel Gendreau, Manuel Iori, Gilbert Laporte, Silvaro Martello, A Tabu Search Heuristic for the Vehicle Routing Problem with Two-Dimensional Loading Constraints. 2004

7. M. Iori. Metaheuristic algorithms for combinatorial optimization problems. 2004

8. Emmanouil E. Zachariadis , Christos D.Tarantilis , Chris T.Kiranoudis, The Vehicle Routing Problem with Simultaneous Pick-ups and Deliveries and Two Dimensional Loading Constraints. European Journal of Operational Research 251 (2016) 369-386)

9. Stephen C.H. Leung, Xiyue Zhou, Defu Zhang Jiemin Zheng, Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem. European Journal of Operational Research 38 (2011) 205-215

10. Lijun Wei, Zhenzhen Zhang, Defu Zhang, Andrew Lim , A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints. European Journal of Operational Research 243(2015) 798-814

11. Fisher, M. L., Jaikumar, R., and Wassenhove, L. N. V. (1986). A Multiplier Adjustment Method for the Generalized Assignment Problem, Management Science, 32(9), 1095-1103.

12. Fred Glover (1990). "Tabu Search: A Tutorial". Interfaces.

13. Glover F. , Laguna M. , 1995, 'Tabu Search Modern heuristic Techniques for Combinatorial Problems.

14. Kirkpatrick S., Gelatt Jr. C. D., Vecchi M. P., 1983, "Optimization by Simulated Annealing ", Science, Volume 220, Number 4598

15. Hansen, P., Mladenovic, N., (2001). Variable Neighborhood Search: Principles and applications, European Journal of Operational Research, 130, 3, 449-467

16. P. Toth and D. Vigo. An overview of vehicle routing problems, in The Vehicle Routing Problem (P. Toth and D. Vigo eds.). SIAM Monographs on Discrete Mathematics and Applications, 2002