ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

ΑΛΓΟΡΙΘΜΟΣ ΑΠΟΙΚΙΑΣ ΜΥΡΜΗΓΚΙΩΝ ΓΙΑ ΤΟ ΠΡΟΒΛΗΜΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΜΕ ΧΡΟΝΙΚΑ ΠΑΡΑΘΥΡΑ.

ΙΑΣΟΝΑΣ ΣΤΑΥΡΙΑΝΙΔΗΣ

2015010135

Χανιά,2024

Περιεχόμενα

Ευχαριστίες

Πρώτα από όλα, θα ήθελα να ευχαριστήσω την οικογένεια μου που με στηρίζει όλα αυτά τα χρόνια σε κάθε μου επιλογή και τον καθηγητή που με ανέλαβε για την πραγματοποίηση της διπλωματικής μου εργασίας κύριο Ιωάννη Μαρινάκη.

Εισαγωγή

Ο Αβραάμ Μάσλοου ήταν Αμερικανός ψυχολόγος του 20^{ου} αιώνα , το 1943 στο άρθρο του με τίτλο A theory of human motivation διατύπωσε την πυραμίδα των αναγκών. Ήταν μία προσπάθεια να κατηγοριοποιήσει τις ανάγκες του ανθρώπου σε κλιμακωτά επίπεδα όπου σύμφωνα με τον αυτόν οι ανάγκες αυτές αποτελούν την κινητήρια δύναμη για την ανθρώπινη συμπεριφορά .

Στον πάτο της πυραμίδας βρίσκονται οι βασικές ανάγκες του ανθρώπου για την επιβίωση ,π.χ. ανάγκη για τροφή και νερό. Στο δεύτερο επίπεδο είναι η ανάγκη για εύρεση εστίας καθώς και η διασφάλιση της ακεραιότητας του πρώτου επιπέδου. Στο τρίτο επίπεδο εμφανίζονται οι κοινωνικές ανάγκες, ανάγκη για διαπροσωπικές σχέσεις κτλ.. Στο τέταρτο οι εγωιστικές ανάγκες και σαν ύψιστο όριο και κορυφή της πυραμίδας βρίσκεται η αυτοπραγμάτωση του υποκειμένου.

Κάθε φορά που καλύπτονται οι ανάγκες ενός επιπέδου αμέσως εκφράζονται τα «θέλω» του επόμενου επιπέδου. Κατά τον Μάσλοου καθώς ο χρόνος τείνει προς στο άπειρο το άτομο δεν σταματάει ποτέ να έχει ανάγκες. Αν παρομοιαστεί η επιχείρηση ως ένας ζωντανός οργανισμός, θα μπορούσε να ισχυριστεί κάποιος πως η πυραμίδα των ανθρώπινων αναγκών του Μάσλοου καθρεπτίζεται και για τις επιχειρήσεις.

Σε μία αγορά που εξελίσσετε συνεχώς, ο ανταγωνισμός αυξάνεται ασταμάτητα μεταξύ των επιχειρήσεων, αυτό οδηγεί στην δημιουργία πολλών προβλημάτων για όλες τις εταιρίες. Συνεπώς, γεννιέται συνεχώς η ανάγκη για λήψη στρατηγικών αποφάσεων κάτω από πιεστικές συνθήκες, αποφάσεις οι οποίες απαιτείται να παρθούν άμεσα χρονικά και ταυτόχρονα να είναι λειτουργικές, αξιόπιστες, με εργονομικές παρεμβάσεις οι οποίες θα επιτρέψουν σε ένα σύστημα εργασίας, να αποδώσει το μέγιστο ποσοτικό και ποιοτικό αποτέλεσμα με την ελάχιστη δυνατή καταπόνηση. Υπάρχει λοιπόν μια συνεχόμενη ροή αναζήτησης βέλτιστων λύσεων σε υπάρχουσες ανάγκες.

Στα πλαίσια αυτής τη λογικής η παρούσα εργασία εστιάζει σε ένα πρόβλημα το οποίο ανήκει στο τμήμα της εφοδιαστικής αλυσίδας μιας επιχείρησης. Καταπιάνεται με ένα ζήτημα δρομολόγησης οχημάτων (VRP) για την διανομή έτοιμων προϊόντων .Υπάρχουν αρκετές παραλλαγές του συγκεκριμένου προβλήματος, αυτό που αναλύεται στην παρούσα εργασία

είναι πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα και ένα απλό πρόβλημα δρομολόγησης οχημάτων στο οποίο ζητάμε να βρούμε μία όσον το δυνατόν καλύτερη λύση με τη χρήση ενός αλγόριθμου εμπνευσμένου από τη φύση.

Κύριος στόχος του αλγορίθμου, είναι σχεδίαση ενός πλάνου μεταφορών προϊόντων με όσο το δυνατόν μικρότερο αριθμό οχημάτων και χαμηλών απαιτήσεων ενέργειας-κόστους για την πραγματοποίηση τους. Ο αλγόριθμος παίρνει μορφή στο προγραμματιστικό περιβάλλον της matlab.

Για τρεις ομάδες δεδομένων της βιβλιογραφίας, χρησιμοποιείται ο αλγόριθμος που κατασκευάστηκε και προκύπτουν κάποια συμπεράσματα ως προς την αποτελεσματικότητα του και συμπεράσματα για μελλοντικές έρευνες.

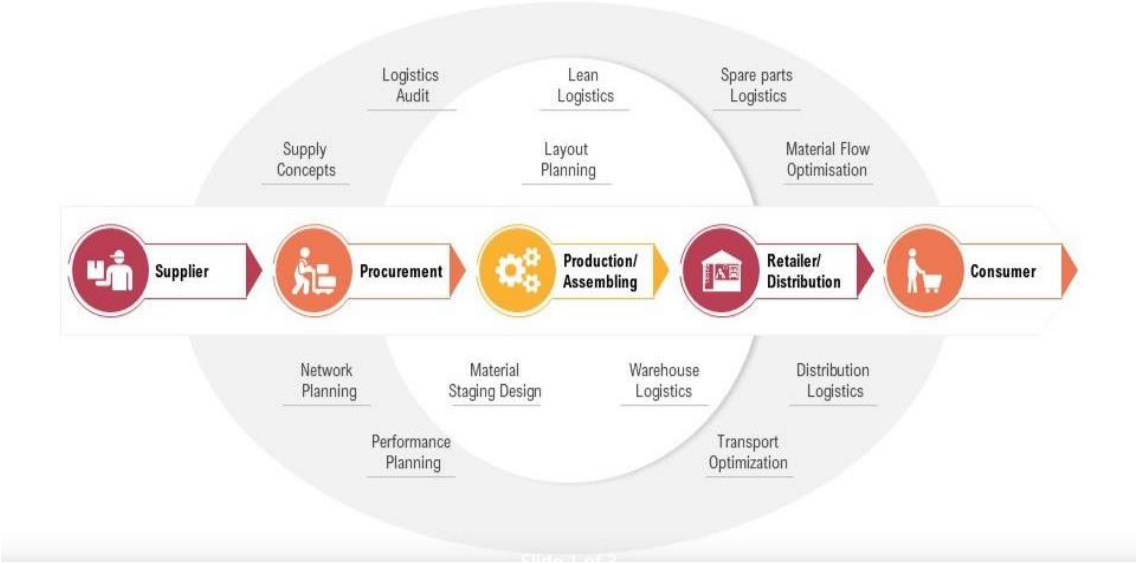
Μέσα λοιπόν από την ανάγκη για εύρεση ενός καλού αλγορίθμου επίλυσης, γίνεται μία απόπειρα απάντησης στο ερώτημα πόσο καλά ανταποκρίνονται αυτές οι μέθοδοι στο συγκεκριμένο πρόβλημα;.

Κεφάλαιο 1 : Εφοδιαστική αλυσίδα

1.1 Η επιστήμη της εφοδιαστικής αλυσίδας(supply chain)

Για να καταφέρει να αποκτήσει ανταγωνιστικό πλεονέκτημα μία επιχείρηση είναι σημαντικό να υπάρχει μια ομαλή λειτουργία στην ροή της παραγωγής, μαζί με σωστή συνεργασία με τους εξωτερικούς συνεργάτες της που θα την βοηθήσουν να αποδίδει στο 100% των δυνατοτήτων της. Για αυτό το λόγο γίνεται επιτακτική ανάγκη η σωστή διαχείριση της εφοδιαστικής αλυσίδας(supply chain). Τι είναι η εφοδιαστική αλυσίδα και από τι αποτελείται θα ρωτήσει κανείς;

Ένας σύντομος ορισμός για την εφοδιαστική αλυσίδα είναι πώς η εφοδιαστική αλυσίδα εστιάζει στην ροή του υλικού από το στάδιο του προμηθευτή σαν πρώτη ύλη μέχρι να φτάσει στα χέρια του καταναλωτή και ταυτόχρονα την ροή πληροφορίας μεταξύ των τμημάτων και υποτμημάτων που στελεχώνουν το συνολικό διάγραμμα ροής



Η εικόνα τραβήχτηκε από : https://www.quora.com/Is-supply-chain-management-important

Εικόνα 1.1

Πρακτικά η εφοδιαστική αλυσίδα εμπεριέχει όλες εκείνες τις ενέργειες που απαιτούνται να γίνουν για την παραλαβή της πρώτης ύλης σε μια εταιρία μέχρι και να διανεμηθεί το τελικό προϊόν στον πελάτη. Όπως για παράδειγμα την επιλογή προμηθευτών, διαχείριση αποθήκης, έλεγχο αποθεμάτων, μεταποιητική διαχείριση, επιλογή κατάλληλων σημείων αναμονής και τέλος μεταφορά στα καταστήματα μέχρι να φτάσει το τελικό προϊόν στα χέρια των καταναλωτών.

Οι βασικές αρχές και διεργασίες της εφοδιαστικής αλυσίδας επαναλαμβάνονται μέσα σε αυτήν αρκετές φορές, καθώς το σημείο αρχής της ροής με το τελικό σημείο δεν βρίσκονται στην ίδια γεωγραφική περιοχή που σημαίνει ότι μια τέτοια αποστολή συνήθως δεν εξαρτάται αποκλειστικά από μια εταιρία. Αφορά λοιπόν την οργάνωση και διαχείριση σε όλα τα υποσύνολα μιας μονάδας παραγωγής προκειμένου να εναρμονιστούν μεταξύ τους και με τους εξωτερικούς συνεργάτες για να επιτευχθεί ο απώτερος σκοπός ο οποίος σε εταιρικό περιβάλλον είναι η σωστή εξυπηρέτηση του πελάτη.

Στην ουσία η εφοδιαστική αλυσίδα εκτείνεται πίσω στους αρχαίους χρόνους, αν και οι βασικές αρχές της έχουν εξελιχθεί και εφαρμοστεί πιο ευρέως στους νεότερους αιώνες. Από τις αρχές του εμπορίου και της ανταλλαγής αγαθών στις αρχαίες πολιτισμένες κοινωνίες, η ανάγκη για αποτελεσματική διαχείριση της κίνησης και του αποθέματος των προϊόντων αποτέλεσε μια αρχή για την εφοδιαστική αλυσίδα.

Κατά την διάρκεια του Β' παγκοσμίου πολέμου η εφοδιαστική αλυσίδα έκρινε σημαντικά αποτελέσματα διότι είχε καθοριστικό ρόλο στην υλοποίηση των στρατιωτικών επιχειρήσεων και τον στρατηγικό σχεδιασμό διανομής στρατιωτικών μονάδων, τεχνολογιών και προμηθειών.

Στις δεκαετίες του 1960 και 1970, εμφανίστηκαν οι πρώτες μεθοδολογίες και τεχνικές για τον υπολογισμό των αποθεμάτων και την προγνωστική ανάλυση, οι οποίες βελτίωσαν τη δυνατότητα προβλέψεων για τη ζήτηση και την αναπλήρωση αποθεμάτων.

Με την ανάπτυξη της πληροφορικής και της τεχνολογίας, η επιστήμη της εφοδιαστικής αλυσίδας επωφελήθηκε από τη χρήση των υπολογιστών και των προηγμένων λογισμικών για την αυτοματοποίηση και την ενίσχυση της διαχείρισης των δραστηριοτήτων της αλυσίδας εφοδιασμού.

Σήμερα, η επιστήμη της εφοδιαστικής αλυσίδας αντιμετωπίζει πολύπλοκες προκλήσεις λόγω της παγκοσμιοποίησης, των γρήγορων και αβέβαιων μεταβολών στη ζήτηση, καθώς και της αυξημένης πίεσης για τη βιωσιμότητα και την ανταγωνιστικότητα. Η συνεχής εξέλιξη της τεχνολογίας, η ανάπτυξη των μεγάλων δεδομένων (big data) και η εφαρμογή της τεχνητής νοημοσύνης και της αυτοματοποίησης έχουν ανοίξει νέες προοπτικές για τη βελτίωση της απόδοσης και της αποτελεσματικότητας της εφοδιαστικής αλυσίδας.

Όπως γίνεται κατανοητό πρόκειται για έναν κλάδο ο οποίος εμπλέκει πολλές παραμέτρους για να διασφαλιστεί η σωστή λειτουργία σε όλα τα μήκη και πλάτη. Είναι σημαντικό λοιπόν να υπάρχει οργανωτική σκέψη και μελέτη για το πλήθος των ενεργειών που θα πραγματοποιηθούν.

1.2 Εφοδιαστική (logistics)

Ο όρος logistics έχει τις ρίζες του από την ελληνική λέξη «λογιστική» είναι μια έννοια αρχαιότερη της «εφοδιαστικής αλυσίδας». Αν και σήμερα οι δύο αυτές έννοιες για κάποιος επιστήμονες είναι ίδιες υπάρχει και η άποψη πως τα logistics δεν είναι ακριβώς ίδια έννοια με αυτή της εφοδιαστικής αλυσίδας διότι δεν είναι παρά ένα κομμάτι αυτής της αλυσίδας το οποίο εστιάζει στις λειτουργίες που αφορούν στον συντονισμό και τη διαχείριση των διαδικασιών, των ροών και των πόρων που απαιτούνται για την αποτελεσματική και αποδοτική παραγωγή, διάθεση και παράδοση των προϊόντων.

Η εφοδιαστική καλύπτει ένα ευρύ φάσμα δραστηριοτήτων, συμπεριλαμβανομένης της αποθήκευσης, της διαχείρισης αποθεμάτων, της μεταφοράς, της συσκευασίας, της διαχείρισης της παραγωγής, της επεξεργασίας παραγγελιών, της παρακολούθησης και του σχεδιασμού των δραστηριοτήτων. Αυτές οι δραστηριότητες βοηθούν στην επίτευξη μιας ομαλής, αποτελεσματικής και οικονομικά αποδοτικής ροής των προϊόντων από τον καταναλωτή προς τον προμηθευτή.

Επίσης συνδέεται στενά με τη διαχείριση της εφοδιαστικής αλυσίδας, καθώς οι δραστηριότητές της επηρεάζουν άμεσα την ροή των υλικών, των πληροφοριών και των πόρων σε όλη την αλυσίδα εφοδιασμού. Η αποτελεσματική λειτουργία της εφοδιαστικής επιτρέπει

τη σωστή διαχείριση των αποθεμάτων, την ελαχιστοποίηση των κοστών, τη βελτίωση της παραγωγικότητας και την εκπλήρωση των απαιτήσεων των πελατών.

Ένα απλό παράδειγμα για την διαφοροποίηση των δύο συστημάτων θα ήταν αν πάρουμε την παγκόσμια εφοδιαστική αλυσίδα σαν σύνολο που αποτελείται από διάφορες εταιρίες στον πλανήτη όπου συνεργάζονται για το τελικό αποτέλεσμα. Μέσα σε αυτή τη ροή μέχρι να φτάσει το τελικό προϊόν στον τελικό προορισμό θα χρειαστεί να σταματήσει σε κάποιους «σταθμούς» κατά την διάρκεια του ταξιδιού του. Σε εκείνα τα σημεία λοιπόν παίρνουν ζωή όλες εκείνες οι ενέργειες που αφορούν το κομμάτι των logistics έτσι ώστε να μπορεί να ανταποκριθεί ο κάθε «σταθμός» σωστά για την επόμενη «στάση» που ακολουθεί η ροή του υλικού στην εφοδιαστική αλυσίδα.

1.3 Μεταφορές

Το κομμάτι των μεταφορών στην εφοδιαστική αλυσίδα είναι τακτικό και αποτελεί από τα μεγαλύτερα μέρη του συνόλου που εκφράζει το τελικό κόστος της επιχείρησης. Έχει παρατηρηθεί ότι διανομή φορτίου κυμαίνεται μεταξύ του ενός τρίτου και των δύο τρίτων αυτού του συνόλου, καθώς οι μεταφορές είναι επίσης και ο συνδετικός κρίκος των σταθμών παραγωγής, των αποθηκών και των καταναλωτών.

Είναι ένα εξαιρετικής σημασίας ζήτημα το οποίο χωρίζεται σε 2 κατηγορίες :

Εσωτερικές μεταφορές: περιλαμβάνουν τις μεταφορές τόσο των πρώτων υλών από τους προμηθευτές προς τα εργοστάσια όσο και τις μεταφορές των ημιέτοιμων προϊόντων ανάμεσα στα εργοστάσια της εταιρίας ή μεταφορές έτοιμων προϊόντων στις αποθήκες και στα σημεία πώλησης.

Εξωτερικές μεταφορές: αυτές περιλαμβάνουν την άμεση διανομή έτοιμου προϊόντος προς το πελάτη είτε από την αποθήκη προς αυτόν είτε από κάποιο κέντρο διανομής προς σε αυτόν.

Τα προβλήματα σχεδιασμού των μεταφορών θέτουν ερωτήματα όπως:

Ποια θα είναι η επιλογή του στόλου οχημάτων της επιχειρήσεις αρχικά σε μέγεθος και σε τι τύπο οχήματος;

Ποια θα είναι η βέλτιστη διαδρομή που θα ακολουθηθεί από τον στόλο;

Ποιος θα είναι ο σωστός σχεδιασμός αυτού του δικτύου;

Τι εργατικό δυναμικό χρειάζομαι για την υλοποίηση του τελικού πλάνου;

Μέσα από αυτά τα ερωτήματα καταλήγουμε στο κομμάτι της εφοδιαστικής αλυσίδας που αφορά τα προβλήματα δρομολόγησης οχημάτων.

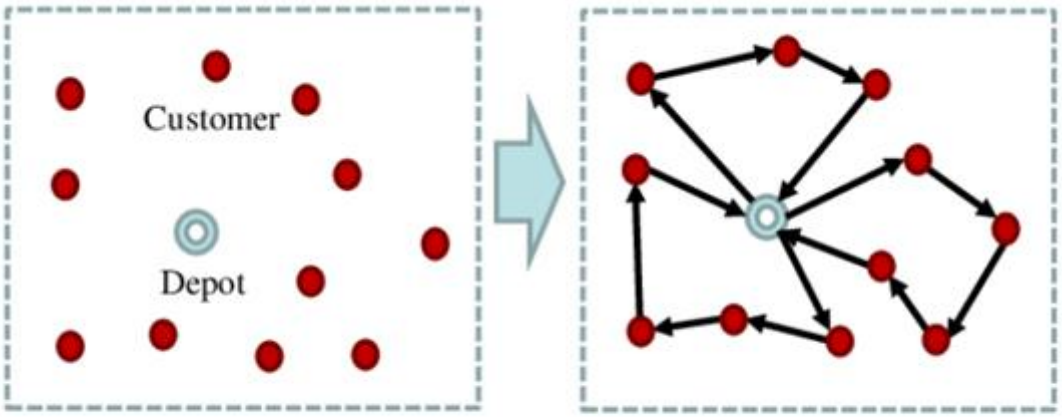
Κεφάλαιο 2: Το πρόβλημα δρομολόγησης οχημάτων

2.1 Ορισμός του προβλήματος δρομολόγησης οχημάτων (Vehicle routing problem-(VRP)).

Όπως μπορεί να γίνει κατανοητό μαζί με όσα έχουν ειπωθεί στις παραπάνω παραγράφους τα προβλήματα δρομολόγησης οχημάτων εστιάζουν στις μεταφορές και ορίζονται σαν ένα σύνολο όμοιων προβλημάτων το οποίο εστιάζει σε προβλήματα μεταφοράς των πρώτων υλών, ημικατεργασμένων η και τελικών προϊόντων, που αντιμετωπίζει μια επιχείρηση.

Υπάρχουν πολλές εκδοχές του συγκεκριμένου προβλήματος και καθορίζονται κάθε φορά από τα μέσα της επιχείρησης, τις ιδιαιτερότητες που θα θέσει το περιβάλλον ως προς την επιχείρηση, τους πελάτες ως προ την επιχείρηση και η επιχείρηση ως προς τον εαυτό της. Γενικά όποια και να είναι η φύση του προβλήματος, το σύνολο των διαδρομών που θα παραδοθεί σαν λύση, θα αποτελείτε από διαδρομές στις οποίες κάθε μια από αυτές, θα ξεκινάει από την αποθήκη της επιχείρησης και θα καταλήγει σε αυτήν, ταυτόχρονα θα πρέπει να ικανοποιεί όλους εκείνους τους περιορισμούς που ορίζουν το πρόβλημα που επιλύει.

Η λύση ενός τέτοιου προβλήματος αναπαρίσταται γραφικά ζωγραφίζοντας τους πελάτες με κόμβους στα σημεία που βρίσκονται στο χώρο, και οι καθορισμένες διαδρομές σχεδιάζονται με προσανατολισμένα βέλη. Η παρακάτω εικόνα δείχνει στο αριστερό μέλος με κόκκινο χρώμα τους πελάτες-κόμβους κατανεμημένους στο χώρο γύρο από την γαλάζια κεντρική αποθήκη, ενώ στο δεξί μέλος ένα σενάριο επίλυσης στο πρόβλημα εξυπηρέτησης τους.



Λύση για ένα πρόβλημα δρομολόγησης οχημάτων με μία αποθήκη :αριστερά οι πελάτες τοποθετημένοι στον χώρο και δεξιά οι διαδρομές εξυπηρέτησης τους.

Η παραπάνω εικόνα τραβήχτηκε από : https://www.researchgate.net/figure/Solution-scheme-for-single-depot-VRP-Left-location-of-members-Right-vehicle-routing_fig1_276092696

Εικόνα 1.2

Ένα αμετάβλητο κομμάτι του συνόλου που αφορά τα δεδομένα του κάθε προβλήματος δρομολόγησης οχημάτων είναι η γνώση του κόστους και του χρόνου που απαιτείται για να πραγματοποιηθεί η κίνηση από την αποθήκη προς πελάτη και από πελάτη σε πελάτη και ο χρόνος εξυπηρέτησης που θα χρειαστεί ο κάθε πελάτης μαζί με τις μονάδες χωρητικότητας που θα καταναλώσει από το συνολικό χώρο του οχήματος. Το υπόλοιπο σύνολο των δεδομένων μπορεί να ποικίλει καθώς εξαρτάται από τον εκάστοτε τύπο του προβλήματος.

Έστω ένα ζευγάρι κόμβων i και j του οποίου είναι γνωστά τα παραπάνω δεδομένα. Σε ένα ενδεχόμενο σενάριο επίλυσης στο οποίο οι παραπάνω κόμβοι θα συνδέονται μέσα από τις τελικές διαδρομές, ορίζεται το κόστος Cij το οποίο θα αντιστοιχεί στο κόστος συντομότερης διαδρομής που ξεκινάει από τον i και καταλήγει στον j αντίστοιχα και για τον χρόνο Tij. Επίσης αυτή η σύνδεση των δυο κόμβων στο τελικό γράφημα σχετίζεται με ένα τόξο (i,j) που παρουσιάζει τα δεδομένα του κόστους-χρόνου(Cij-Tij).Το άθροισμα των στοιχείων που σχετίζονται με τον χρόνο ή το κόστος όλων των τόξων που συντελούν μια διαδρομή μας δίνει το τελικό χρόνο ή κόστος της διαδρομής.

Βασικοί στόχοι που πρέπει να καλυφθούν σε ένα πρόβλημα δρομολόγησης οχημάτων είναι :

- Η ελαχιστοποίηση του συνολικού κόστους μεταφοράς των προϊόντων το οποίο είναι συνάρτηση της συνολικής διαδρομής που πρέπει να γίνει για να εξυπηρετηθούν όλοι οι πελάτες μαζί με τα κόστη που αφορούν την πληρωμή οδηγών και χρησιμοποίηση οχημάτων.
- Ελαχιστοποίηση των αριθμό των οχημάτων- οδηγών που θα χρειαστεί για την διαδρομή.

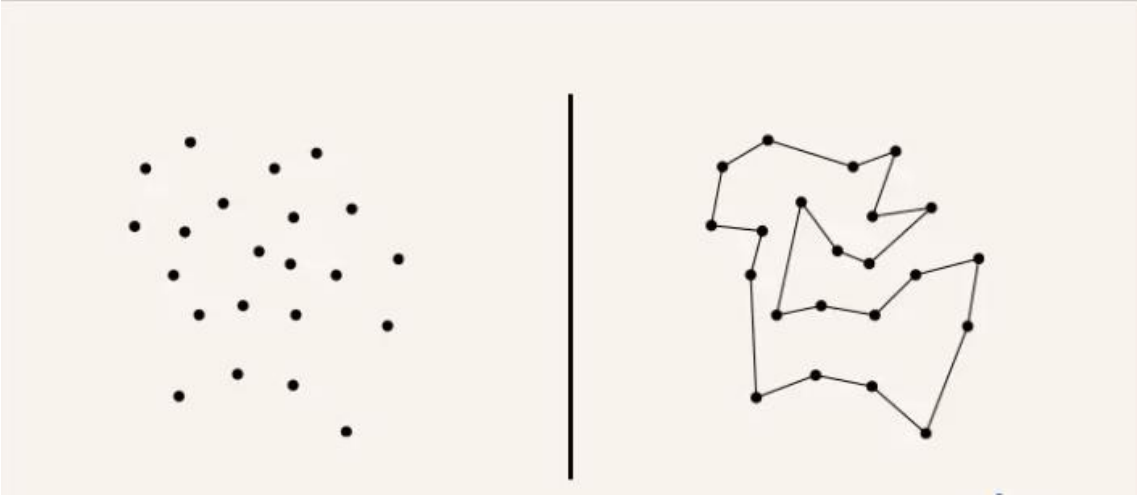
- Επίτευξη ισορροπίας μεταξύ των διαδρομών που θα προκύψουν σε θέματα τελικού φορτίου μαζί με χρόνο απόκρισης κάθε οχήματος.
- Η ελαχιστοποίηση των ποινών που αφορούν μερική ικανοποίηση των πελατών.

Για κάθε ένα από τους παραπάνω στόχους μπορεί να δοθεί ένας συντελεστής βαρύτητας έναντι των άλλων αναλόγως με την φύση του κάθε προβλήματος.

Στις παρακάτω παραγράφους θα αναφερθούν μερικές από τις εκδοχές του προβλήματος δρομολόγησης οχημάτων.

2.2 Πρόβλημα πλανόδιου πωλητή (Traveling salesman problem (TSP)).

Το πρόβλημα του πλανόδιου πωλητή(TSP) πρόκειται για το πιο γνωστό πρόβλημα στην κατηγορία προβλημάτων δρομολόγησης οχημάτων. Αποτελείτε από ένα μόνο όχημα (ή πλανόδιο πωλητή) οπού για συγκεκριμένη αφετηρία και για πεπερασμένο αριθμό πελατών στο χώρο, ψάχνει την βέλτιστη διαδρομή που θα διασχίσει πηγαίνοντας μόνο μία φορά τον καθένα με τελικό σημείο διαδρομής την αφετηρία.

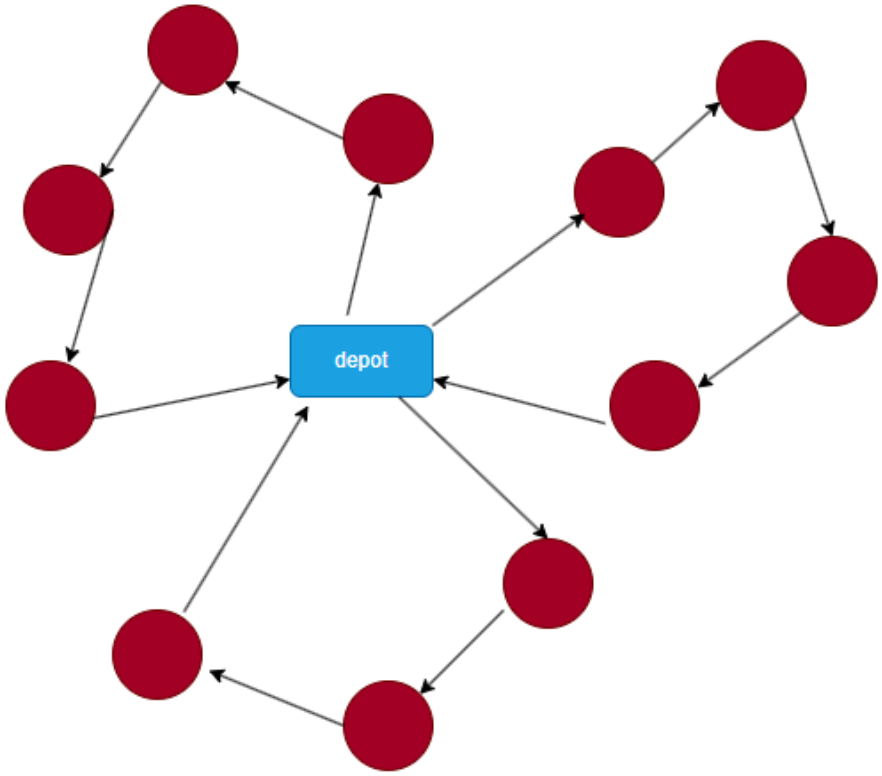


Εικόνα 2.1 τραβήχτηκε από: https://www.upperinc.com/guides/travelling-salesman-problem/

Στην εικόνα 2.1 φαίνεται μια λύση του προβλήματος πλανόδιου πωλητή (TSP). Η οποία ακολουθεί την παραπάνω αντίληψη. Ένας μοναδικός κύκλος(μοναδικός όχημα) που ξεκινάει από ένα σημείο(αφετηρία) και καταλήγει σε αυτό επισκέπτεται όλους τους πελάτες-κόμβους στο περιβάλλον μόνο μια φορά.

2.3 Το περιορισμένης χωρητικότητας πρόβλημα δρομολόγησης οχημάτων(Capacitated VRP)

Το περιορισμένης χωρητικότητας πρόβλημα δρομολόγησης οχημάτων(capacitated vehicle routing problem(CVRP)) ή απλούστερα **το πρόβλημα δρομολόγησης οχημάτων(vehicle routing problem(VRP)).** Πρόκειται για επέκταση του προβλήματος του πλανόδιου πωλητή και αναφέρεται για εκείνες τις περιπτώσεις που δεν είναι εφικτό να δεχτούν επίσκεψη όλοι οι πελάτες-κόμβοι με μόνο ένα όχημα. Αυτό συμβαίνει γιατί είτε η συνολική ζήτηση των πελατών θα ξεπερνάει τις μονάδες χωρητικότητας του ενός οχήματος. Είτε γιατί η εξυπηρέτηση αυτών θα πρέπει να γίνει με βάση κάποιων χρονικών περιορισμών που και πάλι δεν μπορούν να ικανοποιηθούν από ένα μόνο όχημα κ.α. Στην πραγματικότητα αυτός ο τύπος προβλήματος είναι η κύρια βάση για τα προβλήματα δρομολόγησης οχημάτων καθώς στις περισσότερες εκδοχές τα οχήματα έχουν πάντα πεπερασμένη χωρητικότητα, χώρια ό,τι άλλο περιορισμό μπορεί να φέρουν.



Εικόνα 2.3 κατασκευάστηκε από το draw.io

Στην εικόνα 2.3 αναπαρίσταται μία πιθανή λύση ενός τέτοιου προβλήματος. Αυτό που παρατηρείται συγκριτικά με το γράφημα-λύση του πλανόδιου πωλητή είναι ότι υπάρχει πάνω από μία διαδρομή-όχημα και υπάρχει μια κεντρική αποθήκη η οποία για κάθε διαδρομή ορίζεται σαν την αρχή της και τέλος της μαζί. Κατά αυτόν τον τρόπο λοιπόν με κυκλικές διαδρομές επιχειρείται να καλυφθεί το σύνολο των κόμβων στο χώρο για τις οποίες η συνολική ζήτηση της κάθε διαδρομής, δεν πρέπει να ξεπερνάει την χωρητικότητα του

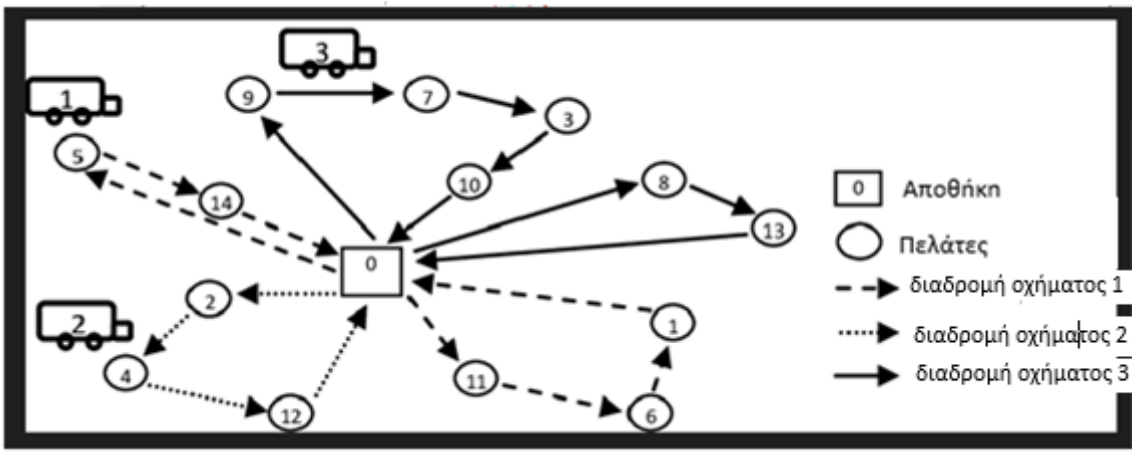
οχήματος, όλα τα οχήματα είναι ίδια, οι πελάτες πρέπει να παραλάβουν το φορτίο τους περνώντας με μία επανάληψη από αυτούς, άρα ο κάθε πελάτης-κόμβος θα ανήκει μόνο σε μια κυκλική διαδρομή και όπως και στο πρόβλημα του πλανόδιου πωλητή μέσα από αυτή την μορφή που θα πρέπει να εμφανίζεται η εκάστοτε λύση θα πρέπει επίσης να αναλογίζεται και το κόστος για αυτήν το οποίο επιθυμείτε να είναι το ελάχιστο συνολικό.

Παρακάτω θα αναφερθούν κάποια προβλήματα που βασίζονται σε πεπερασμένη χωρητικότητα οχήματος

2.4 Το πρόβλημα δρομολόγησης οχημάτων με πολλαπλές επιστροφές στην αποθήκη(Multitrip vehicle routing problem).

Στα περισσότερα προβλήματα δρομολόγησης οχημάτων όταν ανατίθεται σε ένα όχημα να καλύψει μία πιθανή διαδρομή αυτό σημαίνει ότι αυτή η διαδρομή θα καταναλώνει παράλληλα και τον διαθέσιμο χρόνο του εκάστοτε οχήματος σε μια εργάσιμη ημέρα. Τί γίνεται όμως όταν η διαδρομή αυτή αφήνει χρονικό κενό στην ολοκλήρωση του διαθέσιμου εργάσιμου χρόνου του οχήματος;. Σε αυτές τις περιπτώσεις τα περισσότερα οχήματα αν όχι όλα χρησιμοποιούνται παραπάνω από 1 φορές και έτσι προκύπτει το **Πρόβλημα δρομολόγησης οχημάτων με πολλαπλές επιστροφές στην αποθήκη (Multitrip vehicle routing problem).**

Σε αυτή την εκδοχή τα οχήματα επιστρέφουν στην αποθήκη αναλαμβάνουν εκ νέου φορτίο και ξεκινάνε καινούργια διαδρομή



Εικόνα 2.2 τραβήχτηκε από https://www.researchgate.net/figure/An-example-of-vehicle-routing-problem-with-multiple-trips_fig1_287264078.

Στην εικόνα 2.2 παρατηρείται ένα γράφημα-λύση της εκδοχής του προβλήματος, για τον λόγο περιορισμένης χωρητικότητας των οχημάτων σε σχέση με το φορτίο που πρέπει να εξυπηρετηθεί τα οχήματα 1 και 3 πραγματοποιούν από 2 διαδρομές. Στην ουσία οι διαδρομές αυτές είναι έτσι δομημένες προκειμένου να ικανοποιούνται όλοι οι περιορισμοί του προβλήματος. Για τα οχήματα που θα κάνουν από 2 διαδρομές έχει υπολογιστεί το μέγιστο φορτίο που μπορούν να αναλάβουν σε συνδυασμό με την κάλυψη του πλήρους φόρτου για κάθε πελάτη της κάθε διαδρομής, και ο χρόνος απόκρισης κάθε διαδρομής έτσι ώστε το άθροισμα των χρόνων να μην υπερβαίνει τον διαθέσιμο εργατικό χρόνο του κάθε οχήματος.

2.5 Τα προβλήματα δρομολόγησης με διανομές και παραλαβές προϊόντων.

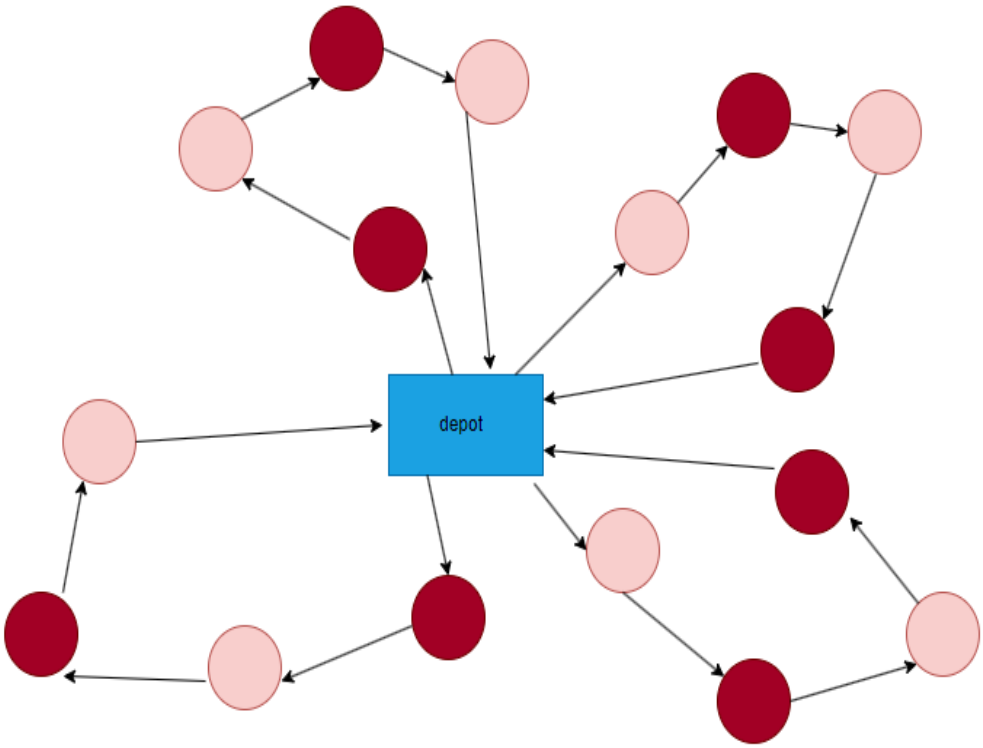
Τα **προβλήματα δρομολόγησης οχημάτων με διανομές και παραλαβές προϊόντων** είναι ένα υποσύνολο προβλημάτων δρομολόγησης το οποίο από μόνο του μπορεί να πάρει μεγάλες διαστάσεις καθώς μέσα σε αυτό το υποσύνολο υπάρχουν διαφορετικές εκδοχές του. Η βασική συνθήκη είναι ότι τα οχήματα πλέον δεν έχουν μόνο να παραδώσουν φορτίο αλλά έχουν και να παραλάβουν. Αρχικά τα προβλήματα αυτά ενδέχεται να αφορούν είτε προϊόντα είτε ανθρώπους, π.χ. την παραλαβή επιβατών από μια στάση μέχρι ο επιβάτης να φτάσει στην στάση που επιθυμεί για να κατέβει. Όταν το φορτίο είναι προϊόντα είναι εκεί που ξεκινάνε και οι διαφορετικές εκδοχές του προβλήματος.

Η πιο γνωστή μορφοποίηση του προβλήματος είναι εκείνη που συμβολίζεται ως **1-M-1 πρόβλημα παραδόσεων και παραλαβών** όπου οι διαδρομές των προϊόντων στους πελάτες και οι παραλαβές προϊόντων από αυτούς χωρίζονται σε 2 διακριτές κατηγορίες μια κατηγορία αφορά αυτούς που θέλουν να παραλάβουν προϊόντα από την αποθήκη μέσο των οχημάτων, και η άλλη κατηγορία είναι αυτή που αφορά όσους πελάτες θέλουν να παραδώσουν κάτι στα οχήματα έτσι ώστε αυτό να μεταφερθούν στην αποθήκη.

Μία απλή εκδοχή προβλήματος δρομολόγησης οχημάτων με διανομές και παραλαβές είναι αυτή που φέρει τίτλο **Πρόβλημα δρομολόγησης οχημάτων με παραλαβές και διανομές ενός μόνο εμπορεύματος (One-commodity pickup and delivery vehicle routing problem(1-PDVRP)).**

Όπως έχει γίνει κατανοητό, για το πρόβλημα αυτό υπάρχουν δύο εκδοχές στις ανάγκες των πελατών οι οποίες είναι είτε η απαίτηση να παραλάβει είτε να παραδώσει προϊόντα από τα οχήματα. Στην εικόνα 2.2 φαίνεται ένα γράφημα που αναπαριστά μια λύση του προβλήματος, οι κόμβοι με μαύρο χρώμα είναι πελάτες που επιθυμούν να παραλάβουν προϊόν και οι κόμβοι με κόκκινο χρώμα είναι πελάτες που επιθυμούν να παραδώσουν.

Το κάθε όχημα με όποιον τύπο πελάτη επιλέξει να ξεκινήσει θα ξεκινήσει είτε με κάποιο φορτίο είτε χωρίς είτε θα γεμίσει τον διαθέσιμο χώρο του με φορτίο. Αυτό εξαρτάται και από το ποιες είναι οι διαθέσιμες επιλογές του στην μετέπειτα πορεία.



Εικόνα 2.2 κατασκευάστηκε από το draw.io

Αυτό το γεγονός φέρνει πολυπλοκότητα στο πρόβλημα ως προς την στρατηγική που θα ακολουθηθεί καθώς πρέπει να υπολογισθεί εκ των προτέρων για την κάθε διαδρομή η σχέση ποσότητας-παραλαβής και ανάθεσης του κάθε οχήματος προς τον εκάστοτε πελάτη προκειμένου να βρεθεί μια βέλτιστη λύση όσον αφορά την χωρητικότητας του οχήματος και τον διαθέσιμο εργατικό χρόνο για όλη την διάρκεια της διαδρομής που θα ακολουθήσει, ταυτόχρονα με τον υπολογισμό των παραμέτρων του κόστους όπου όλα τα οχήματα πραγματοποιούν μόνο μια διαδρομή που ξεκινάει και καταλήγει στην αποθήκη, κάθε πελάτης-κόμβος δέχεται επίσκεψη μόνο μια φορά, και θα πρέπει το συνολικό κόστος να είναι όσο το δυνατό χαμηλό.

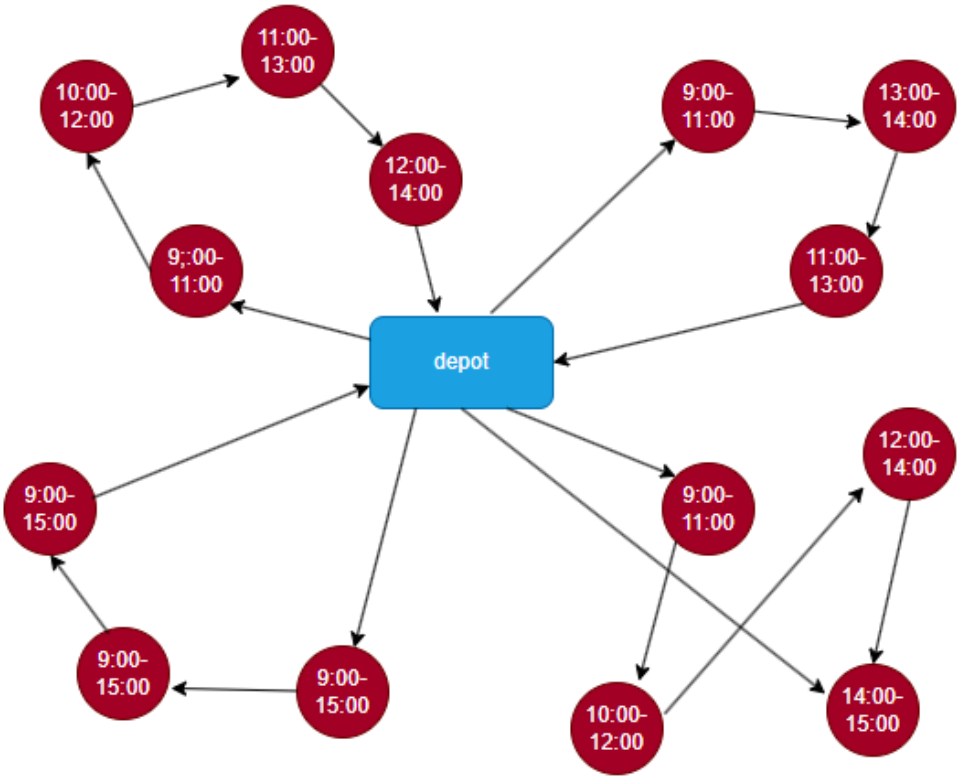
2.6 Προβλήματα δρομολόγησης οχημάτων για την εξυπηρέτηση πελατών μέσα σε συγκεκριμένα χρονικά περιθώρια(Vehicle routing problem with time windows).

Το πρόβλημα είναι γνωστό και ως **πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα**. Τι είναι τα χρονικά παράθυρα; Πρόκειται για μια επέκταση του προβλήματος πεπερασμένης χωρητικότητας όπου οι πελάτες που είναι διάσπαρτοι στον χώρο επιθυμούν να εξυπηρετηθούν εντός ενός χρονικού πλαισίου [αi,βi] και αυτό το χρονικό πλαίσιο ονομάζεται και χρονικό παράθυρο. Όπως σε πολλά προβλήματα περιορισμένης χωρητικότητας έτσι και σε αυτό όλα τα οχήματα ξεκινάνε από την αποθήκη και καταλήγουν σε αυτήν, όλοι οι πελάτες πρέπει να εξυπηρετηθούν με μια επίσκεψη, το συνολικό φορτίο της κάθε διαδρομής δεν γίνεται να ξεπερνάει την χωρητικότητα του οχήματος, όλα τα οχήματα είναι τα ίδια και για την συγκεκριμένη εκδοχή θα πρέπει ο κάθε πελάτης να εξυπηρετηθεί μέσα στο χρονικό παράθυρο που επιθυμεί. Επιπλέον σε αυτό να αναφερθεί ότι υπάρχουν δύο είδη χρονικών παραθύρων :

- **Τα χαλαρά χρονικά παράθυρα:** Όπου δίνεται η δυνατότητα εξυπηρέτησης πελάτη εκτός χρονικού ορίου για μικρές αποκλίσεις είτε πριν είτε μετά

- **Τα σκληρά χρονικά παράθυρα:** Όπου η εξυπηρέτηση του εκάστοτε πελάτη μπορεί μόνο να γίνει εντός του πλαισίου [αi,βi] που σημαίνει ότι αν το όχημα φτάσει νωρίτερα σε αυτόν τότε θα πρέπει να περιμένει μέχρι να «ανοίξει» το χρονικό παράθυρο.

Ο προκαθορισμένος χρόνος για την εξυπηρέτηση του κάθε πελάτη είναι ανεξάρτητος από το τέλος του χρονικού περιθωρίου βi. Το οποίο σημαίνει πως δεν υπολογίζεται το άθροισμα του χρόνου εξυπηρέτησης με τον χρόνο άφιξης για θεωρηθεί η επίσκεψη εντός ορίων μονάχα ο χρόνος άφιξης στον πελάτη-κόμβο.



Εικόνα 2.3 κατασκευάστηκε από το draw.io

Στην παραπάνω εικόνα 2.3 παρουσιάζεται μια εκδοχή εξυπηρέτησης 14 πελατών στο χώρο οπού παρατηρείται στους πελάτες-κόμβους τα χρονικά τους παράθυρα, καθώς και μια διαδρομή στην οποία υπάρχουν σταυρωτά τόξα, διότι έτσι όπως ήταν δομημένα τα χρονικά παράθυρα των πελατών στην περιοχή δεν γινόταν να ακολουθηθεί μια πιο εύκολη διαδρομή για το όχημα. Τέτοια φαινόμενα είναι τακτικά σε αυτή την εκδοχή του προβλήματος πεπερασμένης χωρητικότητας γεγονός που το κάνει εξαιρετικά δύσκολο να βρεθεί βέλτιστη λύση.

Ξεκινώντας την μορφοποίηση του προβλήματος θα οριστεί ένα γράφημα G=(V,A), όπου η αποθήκη συμβολίζεται με τους κόμβους 0 και v+1. Οποιαδήποτε εφικτή διαδρομή ξεκινάει από τον 0 και καταλήγει στον v+1.Επίσης υπάρχει και το χρονικό παράθυρο της αποθήκης το οποίο είτε αναφερόμαστε στον κόμβο 0 είτε στον v+1 είναι το ίδιο $\{\alpha_o,\beta_0\}=\{\alpha_{v+1},\beta_{v+1}\}=\{E,L\}$ το E είναι το ανώτατο χρονικό παράθυρο της αποθήκης και L το κατώτατο, για τις μονάδες ζήτησης d_i και χρόνου εξυπηρέτησης s_i των κόμβων αυτών είναι μηδενικές τιμές, δηλαδή $d_0=d_{v+1}=s_0=s_{v+1}=0$.

Στο πρότυπο που ακολουθεί περιλαμβάνονται δύο ειδών μεταβλητές, αυτές που αφορούν τις μεταβλητές ροής:

$$x_{ijk}=\begin{cases}1\\0\end{cases}$$

- 1 εάν το όχημα k επισκέπτεται τον πελάτη j αμέσως μετά τον i
- 0 αλλιώς

Το βασικό πρόβλημα δρομολόγησης οχημάτων μπορεί να καθοριστεί ως εξής:

$$\min \sum_{i,j} c_{ij} \sum_{k} x_i jk \tag{1}$$

υπό

$$\sum_{k \in K} \sum_{j \in V} x_{ijk} = 1 \qquad \forall i \in N \tag{2}$$

$$\sum_{i \in V-(0)} x_0 jk = 1 \qquad \forall j \in N, k \in K \tag{3}$$

$$\sum_{i \in V-(j)} x_i jk - \sum_{i \in V-(j)} x jik = 0 \qquad \forall j \in N, k \in K \tag{4}$$

$$\sum_{i \in V-(n+1)} x_{in+1k} = 1 \qquad \forall k \in K \tag{5}$$

$$x_i jk(w_i j + s_i + t_i j - w_j k) \leq 0 \qquad \forall k \in K (i,j) \in A \tag{6}$$

$$a_i \sum_{j \in V} x_i jk \leq w_i k \leq b_i \sum_{j \in V} x_i jk \qquad \forall i \in N, k \in K \tag{7}$$

$$E \leq w_i k \leq L \qquad \forall i \in (0, n+1), k \in K \tag{8}$$

$$\sum_{i \in N} d_i \sum_{j \in V} x_i jk \leq C \qquad \forall k \in K \tag{9}$$

$$x_i jk \geq 0 \qquad \forall k \in K, (i,j) \in A \tag{10}$$

$$x_i jk \in [0,1] \qquad \forall k \in K, (i,j) \in A \tag{11}$$

Η αντικειμενική συνάρτηση (1) εκφράζει το συνολικό κόστος .Ο περιορισμός (2) περιορίζει την εκχώρηση κάθε πελάτη σε ένα και μόνο όχημα ενώ οι περιορισμοί από (3) έως (5) χαρακτηρίζουν τη ροή στο μονοπάτι που ακολουθείται από το όχημα k. Οι περιορισμοί (6), (8) και (9) υπάρχουν για να εγγυώνται την εφικτότητα των διαδρομών με βάση τους περιορισμούς χωρητικότητας οχημάτων και χρόνου, και ο (7) εξαναγκάζει το wij για ένα δεδομένο k να γίνει ίσο με το 0 αν το όχημα δεν επισκέπτεται τους πελάτες i και j σε αυτή την διαδρομή.

Αυτές ήταν κάποιες βασικές εκδοχές του προβλήματος δρομολόγησης οχημάτων. Στην συνέχεια θα αναφερθούν τρόποι επίλυσης του προβλήματος μέσα από ένα εύρος αλγορίθμων από τους οποίους είναι εφικτό να προσεγγιστεί μια ικανοποιητική αν όχι βέλτιστη λύση στο εκάστοτε πρόβλημα.

Κεφάλαιο 3: Αλγόριθμοι επίλυσης προβλημάτων εφοδιαστικής αλυσίδας.

3.1 Απλοί Ευρετικοί αλγόριθμοι

Για μεγάλο πλήθος πελατών, μια βέλτιστη λύση στα προβλήματα δρομολόγησης είναι αρκετά δύσκολο να προσεγγιστεί και σχεδόν αδύνατο να βρεθεί μέσα σε γρήγορο χρόνο. Ωστόσο αν μια λύση από έναν ευρετικό αλγόριθμο δεν είναι βέλτιστη αλλά ικανοποιεί τα κριτήρια απόκλισης από αυτήν, γίνεται αποδεκτή. Επίσης χαρακτηριστικά που μπορούν να κάνουν αποδεκτή μια λύση είναι η ευκολία απόκτησης της και να εννοείται να είναι απόλυτα ορθή ως προς τα κριτήρια του προβλήματος.

Οι κατηγορίες των αλγορίθμων είναι οι εξής:

- **Αλγόριθμοι απληστίας (greedy algorithms):** Προσπαθούν να δώσουν μια αρχική εφικτή λύση στο πρόβλημα, αλλά σε περιπτώσεις που το πρόβλημα είναι δύσκολο δεν βρίσκουν καλή λύση διότι είναι μυωπικοί, δηλαδή βλέπουν μόνο μπροστά χωρίς να επεξεργάζονται κάποιες παραπάνω πληροφορίες για να γλιτώσουν χρόνο στην πορεία.

- **Προσεγγιστικοί αλγόριθμοι(approximation algorithms):** Προσπαθούν να επιλύσουν το πρόβλημα χρησιμοποιώντας επιπλέον πληροφορία για να φτιάξουν μια αρχική λύση.

- **Αλγόριθμοι τοπικής αναζήτησης (local search algorithms):** Αυτή η κατηγορία αναφέρεται στους αλγορίθμους οι οποίοι εφαρμόζονται πάνω στην αρχική λύση που έχει δημιουργηθεί ήδη έτσι ώστε να αναζητηθεί μια βελτίωση πάνω της.

Σε αυτό το σημείο να αναφερθεί πως δεν υπάρχει αποκλειστικότητα στην εύρεση της βέλτιστης λύσης μέσα από την εφαρμογή συγκεκριμένου αλγορίθμου. Υπάρχουν πολλοί τρόποι να για βρεθεί μια βέλτιστη λύση σε ένα συγκεκριμένο πρόβλημα.

3.2 Κλασσικοί ευρετικοί αλγόριθμοι

Μια πολύ απλή μεθοδολογία που είναι ικανή να δώσει ένα στρατηγικό πλάνο ως προς την εξυπηρέτηση πελατών(μια αρχική λύση στο πρόβλημα) είναι η **μέθοδος του πλησιέστερου γείτονα.** Έχοντας σαν αφετηρία την αποθήκη η οποία σε μορφή δεδομένων είναι ορισμένη σαν τον κόμβο 1, αναζητείτε κάθε φορά ο πλησιέστερος κόμβος σε απόσταση από το σημείο που βρίσκεται το όχημα κάθε χρονική στιγμή. Οπότε αν για παράδειγμα υπάρχουν 50 πελάτες σκορπισμένοι στον χώρο θα ξεκινήσει μια επαναληπτική διαδικασία μέχρι να βρεθεί ένα μοντέλο που εξυπηρετεί αυτό το σύνολο ικανοποιώντας τους περιορισμούς του προβλήματος. Παρακάτω θα δοθεί ένας πιθανός κύκλος :

1-25-30-2-4-51-43-23-1

Όπως αναφέρθηκε πιο πάνω οι κόμβοι προς εξυπηρέτησης είναι από τον 2 έως και τον 51 αν υπάρξει η παραδοχή ότι ο παραπάνω κύκλος είναι ένας εφικτός κύκλος για το σύστημα και πως βρέθηκε με την μέθοδο του πλησιέστερου γείτονα αυτό σημαίνει ότι έγιναν 7 βήματα μέχρι η μεθοδολογία να φτάσει στην παραβίαση περιορισμών για τον συγκεκριμένο κύκλο. Πρακτικά τι ερωτήματα έθεσε η μέθοδος για να φτάσει σε αυτό το αποτέλεσμα;

1. **Αρχικά ποιος είναι ο κοντινότερος κόμβος από την σημείο που βρίσκομαι(που δεν έχει εξυπηρετηθεί) ;**
   Για την πρώτη επανάληψη σημαίνει πως ξεκινάει από την αποθήκη (1) άρα η απάντηση σε αυτό το ερώτημα ήταν ο κόμβος 25.
2. **Αν εισέλθει ο 25 τι μεταβολές θα προσφέρει; Πχ στην χωρητικότητα του οχήματος, στον χρόνο και πώς αυτά επηρεάζουν τους περιορισμούς του προβλήματος;**
   Αν αυτοί οι περιορισμοί του προβλήματος δεν επιτρέπουν τις μεταβολές που θα προσφέρει η εισαγωγή του νέου κόμβου στον κύκλο, τότε ο κύκλος θα πρέπει να κλείσει επιστρέφοντας στην αποθήκη.

Απάντηση σε αυτά τα δύο ερωτήματα δίνεται κάθε φορά μέσα από μια επαναληπτική διαδικασία έως ότου να εξυπηρετηθούν όλοι. Συνήθως αυτή η μέθοδος δεν αποτελεί εργαλείο για να βρεθεί μια βέλτιστη λύση διότι ανάμεσα στις διαδρομές των οχημάτων υπάρχει μια παρέκκλιση όσον αφορά την ποιότητα. Δηλαδή τείνουν τα πρώτα οχήματα να παρουσιάζουν καλές διαδρομές και οι διαδρομές των επόμενων να είναι αρκετά ανορθόδοξες καθώς είναι μια μυωπική μέθοδος που κοιτάει μόνο μπροστά.

Αντιστρόφως η μέθοδος **των εξοικονομήσεων των Clarke and Wright** είναι καλύτερη σε αυτό το κομμάτι αλλά λίγο πιο πολύπλοκη. Για να δοθεί παράδειγμα για αυτήν την μέθοδο θα χρειαστούν αναλυτικότερα δεδομένα. Αρχικά η μεθοδολογία λειτουργεί μέσα από έναν μαθηματικό τύπο υπολογισμού των εξοικονομήσεων **S=Cik+Ckj-Cij (3.1)**

Όπου το σύμβολο **S** είναι η εξοικονόμηση του κύκλου και το σύμβολο **C** είναι το κόστος του εκάστοτε τόξου. Τα σύμβολα i j k είναι οι εκάστοτε κόμβοι που φτιάχνουν ένα τόξο, το σύμβολο k αναφέρεται στον καινούργιο υποψήφιο κόμβο για εξυπηρέτηση στον κύκλο.

Για να γίνει πιο σαφές ο τρόπος με τον οποίον εξελίσσεται ο αλγόριθμος θα υποθέσουμε πώς ξεκινάμε την αναζήτηση νέου κόμβου από μια φάση που έχουν ήδη

εισέλθει ο πρώτος και ο δεύτερος κόμβος στον κύκλο, έστω λοιπόν: ***1-2-8-1*** ο κύκλος μέχρι στιγμής. Από αυτό το σημείο θα αναζητηθεί η κοντινότερη απόσταση όχι μόνο από το σημείο το οποίο βρίσκεται το όχημα(δηλαδή τον πελάτη 8) αλλά από όλους του κόμβους στον κύκλο μέχρι στιγμής. Έστω ότι τα τόξα (1,6) ,(2,3) και (8,7) είναι τα τόξα τα οποία θα εκφράσουν την κοντινότερη απόσταση για τον κόμβο 1, 2 και 8 αντίστοιχα, επομένως ο κοντινότερος κόμβος από τον 1 είναι ο 6, από τον 2 είναι ο 3 και από τον 8 είναι ο 7. Έχοντας από τα δεδομένα πως:

Η τιμή του κόστους για το τόξο (1,6) είναι: ***16 μονάδες κόστους***

Η τιμή του κόστους για το τόξο (2,3) είναι: ***15 μονάδες κόστους***

Η τιμή του κόστους για το τόξο (8,7) είναι: ***14 μονάδες κόστους***

Για τον λόγο του ότι το τόξο (8,7) έχει την χαμηλότερη τιμή κόστους ανάμεσα στις άλλες επιλογές, αυτόματος σημαίνει πώς ο κόμβος **7** θα είναι ο καινούργιος υποψήφιος κόμβος προς εξυπηρέτηση. Έτσι η μέθοδος θα γεννάει κάθε φορά διαφορετικά σενάρια ανάλογα με το πλήθος του κύκλου για το σύστημα, όπου οι εξοικονομήσεις του νέου κόμβου θα υπολογίζονται από όλες τις πιθανές θέσεις του στον κύκλο και από αυτές θα διαλέγεται η καλύτερη για το συγκεκριμένο παράδειγμα τα σενάρια είναι τα εξής:

1. 1-**7**-2-8-1
2. 1-2-**7**-8-1
3. 1-2-8-**7**-1

Θέλοντας να υπολογίσω το S για τον κάθε κύκλο θα πρέπει να ξεχωρίσω μέσα από τον τύπο (3.1) ποια είναι τα *i j k* του σεναρίου. Για το σενάριο 1 το *i=1* το *k=7* και το *j=2* αυτό σημαίνει ότι η εξοικονόμηση S_1 ισούται με το κόστος του τόξου (1,2) συν το κόστος του τόξου (7,2) πλην το κόστος του τόξου (1,2), $\mathbf{S_1=C_{17}+C_{72}-C_{12}}$. Δηλαδή προσθέτει τα κόστη των τόξων που θα προκύψουν την τοποθέτηση του 7 στην συγκεκριμένη θέση στον κύκλο και αφαιρεί το τόξο το οποίο θα «χαλαστεί» από τον κύκλο αν πρόκειται να μπει ο 7 σε αυτήν την θέση. Έτσι υπολογίζεται το συνολικό κόστος που θα προσθέσει στον υπάρχον κύκλο η έλευση του νέο κόμβου σε μια συγκεκριμένη θέση. Υπολογίζοντας με τον ίδιο τρόπο για τα υπόλοιπα σενάρια τα S_2 και S_3 .Θα επιλεχθεί εκείνο το σενάριο με την χαμηλότερη τιμή S_i άρα εκείνο το σενάριο που θα προσφέρει την μεγαλύτερη εξοικονόμηση στο σύστημα. Όλα τα παραπάνω θα εφαρμοσθούν αν και μόνο αν είναι εφικτό το σενάριο με βάση τους περιορισμούς του προβλήματος αλλιώς γεννιέται νέος κύκλος.

Να αναφερθεί ότι όταν γεννιέται νέος κύκλος ο πρώτος κόμβος που θα μπει στον κύκλο γίνεται με την επιλογή κοντινότερου σε απόσταση από την αποθήκη και από τον δεύτερο ξεκινάει και εφαρμόζεται η διαδικασία του αλγορίθμου. Στην συνέχεια θα δούμε μεθόδους που βοηθάνε στην βελτίωση της αρχικής λύσης.

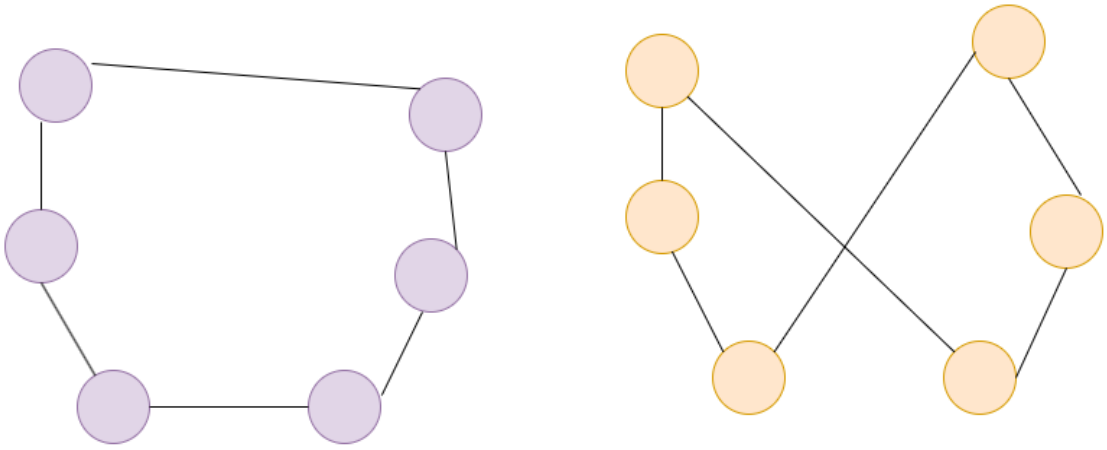
3.3 Τοπική αναζήτηση(local search)

Η τοπική αναζήτηση (local search) είναι μια διαδικασία που εφαρμόζεται αφού υπάρχει μια αρχική λύση για το πρόβλημα και σκοπός της είναι να την βελτιστοποιήσει μέσα από κάποιους αλγορίθμους. Στην ουσία η τοπική αναζήτηση εστιάζει σε ένα συγκεκριμένο σημείο το οποίο έχει περιθώρια βελτιστοποίησης. Πολλές αρχικές λύσεις παρουσιάζουν κύκλους οι οποίοι έχουν διασταυρώσεις των τόξων μεταξύ τους, αυτό το φαινόμενο φέρνει

υψηλότερα κόστη σαν λύση. Ένας αλγόριθμος που έχει αναπτυχθεί προκειμένου να δώσει λύση σε αυτό είναι ο **2opt**

Ο αλγόριθμος 2opt εστιάζει σε έναν συγκεκριμένο κύκλο όπου ελέγχει την εφικτότητα ανταλλαγής τόξων μέσα σε αυτόν με σκοπό την κατασκευή κύκλου που θα προσφέρει χαμηλότερο κόστος

Στην παρακάτω εικόνα φαίνεται μια εφαρμογή 2opt αλγορίθμου ο οποίος αλλάζει 2 τόξα τα οποία διασταυρώνονται και δημιουργεί ένα νέο κύκλο.



Εικόνα 3.1 κατασκευάστηκε από το draw.io

Γενικά τα βήματα που ακολουθεί είναι:

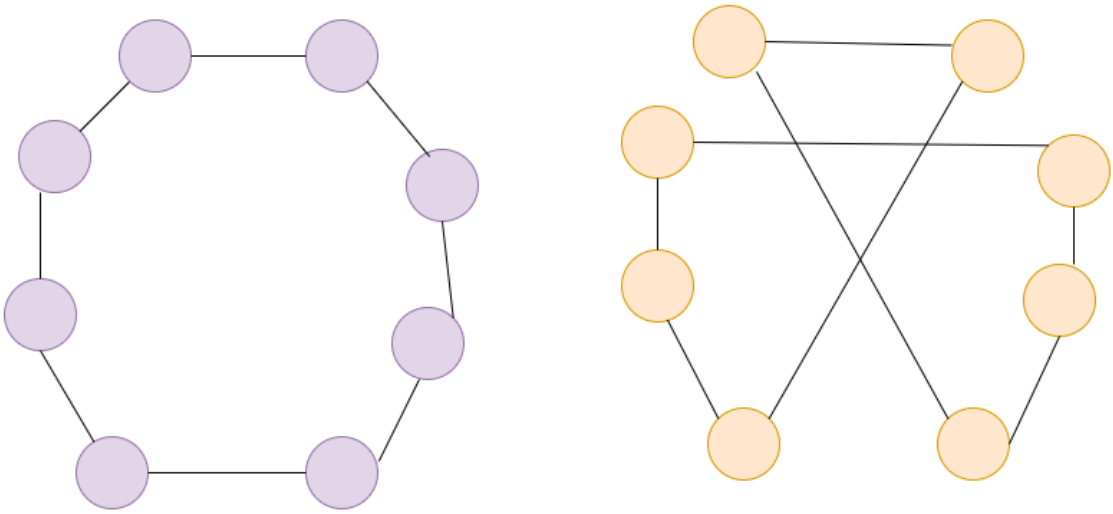
Βήμα 1. Έστω Τ η τρέχουσα διαδρομή.

Βήμα 2. Για κάθε κόμβο i=1,….,ν: Εξετάζονται όλες οι εφικτές προς τους περιορισμούς επιλογές για την αλλαγή 2 τόξων μέσα στον κύκλο και ανάμεσα σε αυτές τις επιλογές θα εφαρμοστεί εκείνη που προσφέρει την μεγαλύτερη μείωση του κόστους.

Βήμα 3. Αν ανάμεσα στις εφικτές επιλογές δεν υπάρχει καμία που να μειώνει το κόστος τότε δεν θα πραγματοποιηθεί καμία αλλαγή στον κύκλο και σταματάει η διαδικασία για τον συγκεκριμένο κύκλο.

Επίσης μια επέκταση του αλγορίθμου 2opt είναι ο αλγόριθμος **3opt** ο οποίος ακολουθεί ακριβώς τα ίδια βήματα με τον πρώτο αλλά η βασική διαφορά τους είναι ότι ο δεύτερος πραγματοποιεί αλλαγές για 3 τόξα μέσα σε ένα κύκλο, αυτό προϋποθέτει πως ο αριθμός των κόμβων σε έναν κύκλο πρέπει να αποτελείται από τουλάχιστον 8 πελάτες και πάνω για να εφαρμοστεί ο αλγόριθμος. Για αυτό τον λόγο συνηθίζεται η μέθοδος 3opt να εφαρμόζεται σε προβλήματα πλανόδιου πωλητή όπου η αρχική τους λύση εμπεριέχει κύκλους με μεγαλύτερο πλήθος πελατών συγκριτικά με τα προβλήματα δρομολόγησης οχημάτων, τα οποία λόγο των περιορισμών που έχουν για την κάθε τους εκδοχή είναι πιο δύσκολο για τα οχήματα να εκμεταλλευτούν το 100% της χωρητικότητας τους.

Η παρακάτω εικόνα δείχνει μια αλλαγή του αλγορίθμου 3opt οπού βγάζει 3 τόξα και τα αλλάζει με 3 καινούργια.



Εικόνα 3.1 κατασκευάστηκε από το draw.io

Μια δυσκολία που έρχεται μαζί με την επιθυμία για την χρήση των παραπάνω αλγορίθμων είναι η επιλογή των τόξων. Ποια τόξα θα πρέπει να διαγραφούν ώστε να προσαχθούν άλλα καλύτερα;

Οι επιλογές ποικίλουν, Αν για παράδειγμα χρησιμοποιείται η 2opt, μια από αυτές είναι **1)**να διαγραφούν 2 τόξα τυχαία, **2)**να διαγράφονται τα 2 χειρότερα ή **3)**να διαγράφεται το χειρότερο και τα 1 τυχαίο, το αντίστοιχο ισχύει για τον 3opt .Ανάλογα με την στρατηγική θα υπάρχουν διαφορετικά πλεονεκτήματα και μειονεκτήματα, π.χ. αν επιλεχθεί η στρατηγική με την απόλυτη τυχαιότητα μεν θα είναι πιο εύκολος στην κατασκευή του και πιο γρήγορος καθώς επιλέγει τυχαία αλλά θα χρειαστεί παραπάνω επαναλήψεις για να πετύχει μια καλύτερη λύση καθώς οι πιθανότητες να πέσεις σε εφικτό κόμματι που χωράει βελτίωση είναι πολύ μικρές. Αν επιλεχθούν τα χειρότερα τότε μπορεί εν μέρει να στοχεύεις σε κομμάτι που φαίνεται ότι χωράει βελτίωση αλλά υπάρχει κίνδυνος αλγόριθμος να παγιδεύεται σε μια τοπική ελάχιστη λύση και να μην μπορεί να βγει από εκεί καθώς κάθε φορά επιλέγει τα ίδια. Σαν καλύτερη στρατηγική είναι η $3^{η}$ η οποία συνδυάζει και τα δύο.

Πέραν από τους αλγόριθμους 2opt και 3opt υπάρχουν και μερικοί που εφαρμόζουν αλλαγές ανάμεσα σε διαφορετικούς κύκλους για την βελτιστοποίηση της αρχικής λύσης. Παρακάτω θα παρουσιαστούν τέσσερις διαδικασίες που εφαρμόζονται ανάμεσα σε δύο η παραπάνω κύκλους.

1-0 Επανατοποθέτηση(1-0 relocate):Η διαδικασία αυτή βγάζει έναν κόμβο από έναν κύκλο και τον τοποθετεί σε έναν άλλο με σκοπό την βελτίωση του κόστους. Συνήθως αυτή η διαδικασία έχει εφαρμογή όταν η αρχική λύση δίνει κύκλους με μικρό αριθμό πελατών, Έτσι γίνεται έρευνα για να διαλυθεί ο κύκλος ώστε να καταφέρουν να εξυπηρετηθούν κύκλοι που δεν έχουν και τόσο μεγάλο αριθμό πελατών

.

1-1 Ανταλλαγή (1-1 exchange):Ανταλλάσσει έναν κόμβο από μια διαδρομή με έναν άλλο από μια άλλη διαδρομή με σκοπό την μείωση του κόστους.

1-2 Ανταλλαγή (1-2 exchange):Ανταλλάσσει έναν κόμβο από μία διαδρομή με δύο κόμβους από μια άλλη.

2-2 Ανταλλαγή(2-2 exchange):Ανταλλάσσει δύο κόμβους από μια διαδρομή με δύο κόμβους από μια άλλη

3.4 Μεθευρετικοί αλγόριθμοι

Το πόσο πιθανό είναι να φτάσει σε μια βέλτιστη λύση ένας αλγόριθμος τοπικής αναζήτησης, εξαρτάτε πάρα πολύ από την αρχική λύση την οποία επιθυμεί να βελτιστοποιήσει. Αν για παράδειγμα η αρχική λύση είναι ικανοποιητική και ταυτόχρονα συγκλίνει σε ένα τοπικό ελάχιστο, τότε ο αλγόριθμος της τοπικής αναζήτησης θα καταφέρει να πέσει σε αυτό με κίνδυνο την παγίδευση του σε εκείνο το σημείο ενώ στην πραγματικότητα ενδέχεται να υπάρχει προοπτική για κάτι καλύτερο. Λόγο αυτού του προβλήματος έχει αναπτυχθεί μια ακόμη κατηγορία αλγορίθμων, οι **Μεθευρετικοί αλγόριθμοι** μέσα από τους οποίους δίνεται η δυνατότητα αποφυγής της πιθανότητας να κολλήσει μια αναζήτηση σε τοπικό ελάχιστο.

Μέσα σε αυτό το σύνολο αλγορίθμων οι στρατηγικές ποικίλουν. Υπάρχουν αλγόριθμοι που χρησιμοποιούν μία λύση και μέσα από αυτήν αναζητούν καλύτερες εκδοχές αυτής, ενώ κάποιοι άλλοι χρησιμοποιούν μια οικογένεια από λύσης και επακολουθεί αναζήτηση μέσα σε αυτό το σύνολο. Για την πρώτη υποκατηγορία ενισχύεται περισσότερο το κομμάτι της εκμετάλλευσης ή εντατικοποίησης και στην δεύτερη εδρεύει περισσότερο το χαρακτηριστικό της εξερεύνησης ή διάχυσης.

Σε τέσσερις κύριες κατηγορίες βασίζεται το σύνολο των μεθευρετικών αλγορίθμων:

- Επαναληπτικές διαδικασίες που ξεκινάνε από διαφορετικές αρχικές λύσεις
- Αλγόριθμοι οι οποίοι δεν απαιτούν το κάθε επόμενο βήμα να είναι η καλύτερη επιλογή ανάμεσα στις άλλες καθώς αυτό μπορεί μακροπρόθεσμα να δώσει καλύτερη λύση.
- Αλγόριθμοι που όταν κολλήσουν σε ένα τοπικό ελάχιστο τότε αλλάζουν το σημείο αναζήτησης λύσεων.
- Αλγόριθμοι οι οποίοι αλλάζουν είτε την αντικειμενική συνάρτηση του προβλήματος είτε τους περιορισμούς του.

Αναφορικά θα δοθούν μερικοί τίτλοι μεθευρετικών αλγορίθμων:

- Προσομοιωμένη Ανόπτηση (Simulated Anneling)
- Περιορισμένη Αναζήτηση (Tabu search)
- Διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής(Greedy Randomized Adaptive Search Procedure)
- Αλγόριθμος Επανασύνδεσης Διαδρομών(Path Relinking(PR))
- Αλγόριθμος Μεταβλητής Γειτονιάς Αναζήτησης (Variable Neighborhood Search)

3.5 Αλγόριθμοι εμπνευσμένοι από την φύση

Σε αυτή την παράγραφο θα γίνει αναφορά ίσως στους πιο ενδιαφέρον αλγορίθμους επίλυσης προβλημάτων συνδυαστικής βελτιστοποίησης. Αλγόριθμοι εμπνευσμένοι από την φύση παρουσιάζονται ως **γενετικοί, εξελικτικοί και μιμητικοί αλγόριθμοι.**

Η έμπνευση των **γενετικών αλγορίθμων** έρχεται μέσα από την εξελικτική διαδικασία που υπάρχει στην φύση. Αρχικά, δημιουργείτε μια αρχική πληθυσμιακή συλλογή από λύσεις(**άτομα**). Αυτές οι λύσεις αποτελούνται από γονίδια που αντιπροσωπεύουν τα χαρακτηριστικά του προβλήματος που προσπαθούμε να βελτιστοποιήσουμε.

Μέσα από αυτές τις λύσεις θα ξεχωρίσουν κάποιες που φέρουν καταλληλότερα χαρακτηριστικά για τον στόχο του προβλήματος(**γενιά**). Από αυτήν την γενιά ένα ποσοστό λύσεων θα χρησιμοποιηθεί για συνδυαστούν δύο λύσεις μεταξύ τους(**γονείς**) ώστε να δημιουργήσουν ένα νέο σύνολο από καινούργιες λύσεις(**απογόνους**). Στο υπόλοιπο ποσοστό λύσεων θα πραγματοποιηθούν διαδικασίες αναζήτησης για μια καλύτερη λύση(**μετάλλαξη**). Έτσι όσες λύσεις γεννιούνται και είναι πιο καλές θα παίρνουν τις θέσεις των λύσεων της προηγούμενη γενιάς με σκοπό να βρεθεί η καλύτερη λύση.

Αναφορικά κάποιοι γενετικοί αλγόριθμοι:

- **Γενετικοί αλγόριθμοι πολλαπλών πληθυσμών-νησιών(Island genetic Algorithms).**
- **Αλγόριθμος της Διαφορικής Εξέλιξης (Differential evolution).**
- **Αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων(Particle swarm optimization).**
- **Αλγόριθμος βελτιστοποίησης Ζευγαρώματος μελισσών(Honey bees Mating Optimization).**
- **Αλγόριθμος βελτιστοποίησης αποικίας μυρμηγκιών (Ant colony optimization)**

Κεφάλαιο 4: Αλγόριθμος αποικίας μυρμηγκιών για το πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα.

4.1 Εισαγωγή στο πρόβλημα της εργασίας

Η παρούσα διπλωματική εργασία εστιάζει σε ένα πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα. Το οποίο είναι όπως ακριβώς έχει προαναφερθεί στην παράγραφο 2.6 στο κεφάλαιο 2.

Δηλαδή τα δεδομένα και περιορισμοί είναι τα εξής :

- Όλα τα οχήματα είναι ίδια και πεπερασμένης χωρητικότητας.
- Η επίσκεψη σε κάθε πελάτη-κόμβο γίνεται μόνο μια φορά.
- Όλα τα οχήματα ξεκινάνε από την κεντρική αποθήκη και καταλήγουν σε αυτήν.
- Για τον κάθε πελάτη στο χώρο υπάρχει ένα συγκεκριμένο χρονικό περιθώριο στο οποίο επιθυμεί να εξυπηρετηθεί.
- Η συνολική διαδρομή του κάθε οχήματος πρέπει να είναι εντός χρονικών παραθύρων της αποθήκης.
- Μονάδες φορτίου που πρέπει να παραδοθούν σε κάθε πελάτη.
- Χρονικές μονάδες εξυπηρέτησης μετά την άφιξη του οχήματος στον κάθε πελάτη.
- Μονάδες απόστασης-χρόνου μετακίνησης-κόστους για την μετακίνηση του εκάστοτε οχήματος από οποιοδήποτε σημείο προς οποιοδήποτε σημείο στο χώρο.

Η τελευταία βούλα στα δεδομένα σημαίνει πως η απόσταση, ο χρόνος μετακίνησης και το κόστος είναι 1 προς 1 μεταξύ τους πράγμα που σημαίνει ότι για ένα τόξο (i,j) αν έχει κόστος Cij=10 τότε θα έχει χρειάζεται αντίστοιχα 10 μονάδες για τον χρόνο t και είναι 10 μονάδες απόστασης d.Επίσης για τους περιορισμούς των χρονικών παραθύρων δεν υπάρχει καμία ανοχή στην παράδοση φορτίου σε σχέση με τον χρόνο. Αν το όχημα φτάσει νωρίτερα από την έναρξη του παραθύρου τότε περιμένει για να εξυπηρετήσει και δεν γίνεται να παραδώσει αν φτάσει μετά την λήξη του παραθύρου.

Αυτό το πρόβλημα θα επιλυθεί με την χρήση ενός συνδυασμού των αλγορίθμων **Clark n Wright και αποικίας μυρμηγκιών(ant colony).** Έχει αναφερθεί ο τρόπος λειτουργίας του Clark n Wrigth αλγορίθμου, παρακάτω θα αναφερθεί και η μαθηματική μοντελοποίηση του αλγορίθμου αποικίας μυρμηγκιών. Αρχικά να αναφερθεί πως ο αλγόριθμος στην ουσία

μιμείται την διαδικασία που ακολουθούν τα μυρμήγκια για την αναζήτηση τροφής τους. Αναπτύσσουν μία τεχνική η οποία τα βοηθά να βρουν την συντομότερη διαδρομή από την φωλιά τους προς την τροφή τους. Στην αρχή της αναζήτησης τους τα μυρμήγκια κινούνται τυχαία γύρο από την φωλιά τους για την εύρεση τροφής, αυτές οι τυχαίες κινήσεις όμως θα οδηγήσουν σιγά σιγά προς την εύρεση της βέλτιστης ή σχεδόν βέλτιστης διαδρομής καθώς το κάθε μυρμήγκι μετά την ολοκλήρωση της εκάστοτε διαδρομής αφήνει πίσω του μια ουσία, την ***φερομόνη*** η οποία αποτελεί ένα μέσο επικοινωνίας για τα μυρμήγκια. Η ποσότητα της φερορμόνης που υπάρχει σε ένα μονοπάτι επηρεάζει την πιθανότητα επιλογής του από τα μυρμήγκια. Όσο πιο σύντομη είναι η διαδρομή και όσο πιο μεγάλη είναι η ποσότητα της τροφής που βρέθηκε μέσα από αυτό, τόσο πιο πολύ φερορμόνη θα αφήσει στο μονοπάτι το μυρμήγκι. Επίσης αυτή η ουσία δεν μένει για πάντα στο μονοπάτι, μετά από κάποιο χρόνο εξατμίζεται. Επομένως, μέσα από μια συνεχόμενη και επαναληπτική διαδικασία θα προκύψει μια μεγάλη ποσότητα φερορμόνης στο καλύτερο ή σχεδόν καλύτερο μονοπάτι το οποίο θα αποτελεί τον μοναδικό πόλο έλξης για τα μυρμήγκια. Ταυτόχρονα τα υπόλοιπα μονοπάτια θα πάψουν έχουν την όποια ποσότητα φερορμόνης, καθώς λόγο έλλειψης επισκεψιμότητας δεν θα ανανεώνεται η φερορμόνη σε αυτά, και σιγά σιγά θα εξατμιστεί παντελώς.

Η μοντελοποίηση του αλγορίθμου για τα προβλήματα δρομολόγησης οχημάτων, ξεκινάει με την μεταβλητή τ_{ij} η οποία δηλώνει την ποσότητα της φερομόνης στα τόξα για να πάει από το i στον j. Η ευρετική πληροφορία που χρησιμοποιείται στο συγκεκριμένο πρόβλημα έρχεται από την συνάρτηση :

$$n_{ij}=1/c_{ij}$$

Όπου το c_{ij} είναι η απόσταση του τόξου i j. Η παραπάνω σχέση λέει πως η επιθυμία του οχήματος να επισκεφθεί μετά τον πελάτη i τον j είναι αντιστρόφως ανάλογη με την απόσταση του τόξου i j. Σε αυτό το σημείο θα δειχθεί ο τύπος από τον οποίο υπολογίζεται η φερομόνη τ_{ij} :

$$T_{ij}=m/TC.$$

Όπου m είναι ο αριθμός των μυρμηγκιών TC είναι το κόστος ενός κύκλου ο οποίος έχει κατασκευαστεί με έναν πιο απλό τρόπο για παράδειγμα με την μέθοδο του πλησιέστερου γείτονα. Κατά την κατασκευή κύκλων στον αλγόριθμος αποικίας μυρμηγκιών κάθε φορά που υπάρχει το πρόβλημα επιλογής του επόμενου κόμβου προς εξυπηρέτηση εφαρμόζεται ένας πιθανοτικός κανόνας ο οποίος θα επηρεάσει την επιλογή του οχήματος-μυρμηγκιού.

Αν για παράδειγμα το εκάστοτε μυρμήγκι έχει να επιλέξει ανάμεσα σε ένα πλήθος πελατών *M* τότε θα πρέπει να υπολογισθεί η πιθανότητα επιλογής κόμβου για κάθε πελάτη στο σύνολο *M*. Αυτό θα γίνει χρησιμοποιώντας την σχέση:

$$p_{ij} = \frac{[\tau_{ij}]^{\alpha}[n_{ij}]^{\beta}}{\sum_{l=1}^{M}[\tau_{ij}]^{\alpha}[n_{ij}]^{\beta}} \qquad (4.1)$$

Αρχικά τα *α* και *β* είναι εκθέτες οι οποίοι προσδίδουν βαρύτητα στις αντίστοιχές μεταβλητές αν οριστεί σαν μεταβλητή ελκυστικότητας $\xi ij = [\tau ij]^{\alpha}[nij]^{\beta}$ δηλαδή να ακολουθηθεί ο κόμβος j μετά τον i τότε σε περίπτωση που *α>β* αυτό θα σημαίνει ότι η ελκυστικότητας **ξ**$_{ij}$, θα επηρεάζεται από την φερομόνη περισσότερο σε αντίθεση με την ευρετική μεταβλητή n. Επομένως ο ορισμός της πιθανότητας θα μπορούσε να χαρακτηριστεί και έτσι :

$$p_{ij} = \frac{\xi_{ij}}{\sum_{l=1}^{M} \xi_{ij}} \qquad (4.2)$$

Έχοντας υπολογίσει την πιθανότητα για τον κάθε πιθανό κόμβο προς εξυπηρέτηση το μυρμήγκι-όχημα θα ακολουθήσει την διαδρομή που θα του τύχει. Στην συνέχεια αφού βρεθεί μια λύση θα πρέπει όλα εκείνα τα δεδομένα που προέκυψαν από την παραπάνω διαδικασία, δηλαδή η ποσότητας της φερομόνης που υπολογίσθηκε, να προστεθεί στις διαδρομές που εκτέλεσε το μυρμήγκι. Πριν γίνει αυτή η πρόσθεση θα γίνει η διαδικασία της εξάτμισης με την χρήση του παρακάτω τύπου.

$$\tau_{ij} = (1 - \rho)\tau_{ij} \qquad (4.3)$$

Η σχέση (4.3) πρακτικά πηγαίνει σε όλα τα τόξα που υπάρχουν και μειώνει την φερομόνη καθώς το *0<ρ<1*. Έτσι για όσα τόξα δεν θα προστεθεί φερομόνη καθώς μεσα από την διαδικασία του αλγορίθμου μέχρι στιγμής δεν κρίθηκαν ικανά, οι τιμές της φερομόνης τους μειώνεται εκθετικά. Για τα τόξα τα οποία κρίηκαν ικανά να παρουσιαστούν σαν μια πιθανή λύση θα υπάρχει αύξηση της ποσοτητάς της φερομόνης μέσα από τον παρακάτω τύπο.

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^{m} \Delta[\tau_{ij}]^{k} \qquad (4.4)$$

Εδώ η αύξηση γίνεται αφού υπολογισθεί το ***Δτ*** το οποίο είναι η ποσότητα της φερομόνης που παράχθηκε από το μυρμήγκι k. Όσο χαμηλότερο είναι το κόστος του κύκλου τόσο πιο μεγάλη θα είναι η ποσότητα της φερομόνης. Αυτό εκφράζεται από την σχέση (4.5) ,στον παρονομαστή είναι το κόστος C^k του κύκλου που δημιούργησε το εκάστοτε μυρμήγκι *k.*

$$\Delta[\tau_{ij}]^{k} = \frac{1}{C^{k}} \qquad (4.5)$$

Ουσιαστικά, με βάση την ποιότητα του κύκλου που θα σχηματιστεί υπολογίζεται η ποσότητα της φερομόνης που παράχθηκε από το μυρμήγκι (σχέση 4.5) και στην συνέχεια αυτή προστίθεται σε κάθε τόξο του κύκλου από τον οποίο προήλθε (σχέση 4.4). Όπως προαναφέρθηκε πιο πάνω στην παράγραφο, η λύση του προβλήματος της εργασίας δίνεται μέσα από ένα συνδυασμό των αλγορίθμων Clark n Wright με τον αλγόριθμο αποικίας μυρμηγκιών. Πως θα συνδυαστούν οι δύο αλγόριθμοι για να παρουσιάσουν ένα αποτέλεσμα;

Πάνω σε αυτό θα φανεί στην συνέχεια πιο αναλυτικά ο τρόπος με τον οποίο συνδυάζονται. Γενικά η λειτουργία αλγορίθμου υπολογίζει πρώτα τα ***savings*** S_{ij} και στην συνέχεια για τον υπολογισμό του συντελεστή ελκυστικότητας $\xi ij = [\tau ij]^{\alpha}[nij]^{\beta}$ αντί για την μεταβλητή n_{ij} θα εισέλθει στην θέση της το S_{ij}.

4.2 Ο αλγόριθμος σε βάθος.

Ο αλγόριθμος θα πραγματοποιηθεί στο περιβάλλον της MATLAB, οι αρχικοποιήσεις των δεδομένων θα διαβάζονται κάθε φορά από ένα αρχείο excel.

3	VEHICLE						
4	NUMBER	CAPACITY					
5	25	200					
6							
7	CUSTOMER						
8	CUST NO.	XCOORD.	YCOORD.	DEMAND	READY TIME	DUE DATE	SERVICE TIME
9							
10	0	40	50	0	0	1236	0
11	1	45	68	10	912	967	90
12	2	45	70	30	825	870	90
13	3	42	66	10	65	146	90
14	4	42	68	10	727	782	90
15	5	42	65	10	15	67	90
16	6	40	69	20	621	702	90
17	7	40	66	20	170	225	90
18	8	38	68	20	255	324	90
19	9	38	70	10	534	605	90
20	10	35	66	10	357	410	90
21	11	35	69	10	448	505	90
22	12	25	85	20	652	721	90
23	13	22	75	30	30	92	90
24	14	22	85	10	567	620	90

data | C101 | C102 | C103 | C104

Εικόνα 4.1

Η εικόνα 4.1 δείχνει ένα μέρος από κάποια δεδομένα τα οποία επεξεργάζεται ο αλγόριθμος. Τα παραπάνω δεδομένα θα αποθηκευτούν μέσα σε κάποια διανύσματα στην MATLAB, με τις συντεταγμένες XCOORD και YCOORD θα σχηματιστεί ένας πίνακας 101x101 ο οποίος θα έχει περιέχει για όλα τα πιθανά τόξα την απόσταση, τον χρόνο και το κόστος. Αυτά

τα στοιχεία είναι 1 προς 1 μεταξύ τους, αυτό σημαίνει πως για οποιοδήποτε τόξο *i->j* οι μονάδες κόστους-χρόνου-απόστασης πραγματοποίησης της διαδρομής είναι ίσες. Για να σχηματιστεί αυτός ο πίνακας θα χρησιμοποιηθεί ο τύπος της ευκλείδειας απόστασης μεταξύ τους ο οποίος εκτός από τετραγωνικός θα είναι και συμμετρικός και η συμμετρία του σημαίνει πως είναι ίδιες οι τιμές είτε να πας από τον κόμβο i->j είτε από τον j->i. Έστω ο δισδιάστατος πίνακας *A* ο οποίος θα προκύψει από:

$$A(i,j) = \sqrt{(XCOORD(i) - XCOORD(j))^2 + (YCOORD(i) - YCOORD(j))^2}$$

Έχοντας δημιουργήσει και τον πίνακα του κόστους-απόστασης-χρόνου ο αλγόριθμος έχει πλέον όλα εκείνα τα δεδομένα που χρειάζεται για να επεξεργαστεί. Αρχικά θα παρουσιαστεί η γενικότερη εικόνα του αλγορίθμου και στην συνέχεια θα γίνεται πιο συγκεκριμένη περιγραφή σε κομμάτια του.

Το αρχικό φύλλο κώδικα έχει όλες τις απαραίτητες αρχικοποιήσεις μαζί με τον πίνακα της φερορμόνης ο οποίος αρχικοποιείται με μονάδες. Έπειτα ξεκινάει ένας βρόχος επανάληψης *for* οποίος έχει σκοπό να πραγματοποιήσει από 1-60 επαναλήψεις, μέσα σε αυτήν την κύρια *for* θα υπάρχει μια πεπλεγμένη *for* η οποία θα εκτελείται από 1 έως 5 φορές και αυτό γιατί θα καλεί κάθε φορά την κύρια συνάρτηση(function) που κατασκευάστηκε για να δημιουργεί λύσεις. Τα αποτελέσματα από αυτές τις 5 λύσεις που προκύπτουν αποθηκεύονται σε ανάλογες λίστες και cell-arrays της MATLAB, οι λύσεις που προκύπτουν είναι διαφορετικές μεταξύ τους καθώς όπως έχει προαναφερθεί για τον αλγόριθμο λειτουργεί με τυχαιότητα. Ο τρόπος με τον οποίο η συνάρτηση κατασκευής της λύσης εφαρμόζει την τυχαιότητα θα παρουσιαστεί στην συνέχεια. Ο αριθμός των επαναλήψεων επηρεάζει κατά πολύ τον τελικό χρόνο απόκρισης του αλγορίθμου και για αυτό τον λόγο είναι πολύ συγκεκριμένος ως προς το επιθυμητό αποτέλεσμα σε σχέση με τον χρόνο. Για κάθε μια από αυτές τις 5 λύσεις θα εφαρμοστεί πάνω τους ένας αλγόριθμος που διερευνά πιθανές ενώσεις κύκλων μεταξύ τους. Στην συνέχεια διαλέγεται ο καλύτερος εκ των 5 και σε αυτόν εφαρμόζεται η τοπική αναζήτηση 1-1 exchange και 2opt. Αν και εφόσον υπάρχουν βελτιώσεις, αυτές περνάνε στα κατάλληλα διανύσματα που αποθηκεύουν τις εκάστοτε πληροφορίες της λύσης και αφού υπάρχει η καλύτερη δυνατή λύση που δίνετε από την πεντάδα τότε επηρεάζεται ο πίνακας της φερορμόνης που αρχικοποιήθηκε πιο πάνω. Παρακάτω δίνεται ένας ψευδοκώδικας του αλγορίθμου:

Για 1 έως 100 επαναλήψεις

Για 1 έως 5 επαναλήψεις

Κάλεσε συνάρτηση κατασκευής λύσης

Κάλεσε συνάρτηση ένωση κύκλων

Τέλος

Από τις παραπάνω λύσεις διάλεξε την τοπική καλύτερη

Εφάρμοσε τοπική αναζήτηση στην τοπική καλύτερη

Συγκρίνω κάθε φόρα την τοπική καλύτερη λύση με την ολική καλύτερη λύση.

Αν είναι καλύτερη από την ολική καλύτερη τότε παίρνει την θέση της.

Ανανέωση φερορμόνης με βάση την τοπική καλύτερη λύση.

Τέλος.

Ο παραπάνω ψευδοκώδικας μπορεί να χαρακτηριστεί και ως κύριος κώδικας. Παρόλο που δεν υπάρχει τιμή κόστους στα δεδομένα για την εισχώρηση νέου οχήματος στον στόλο θεωρείτε καλύτερη μια λύση με μικρότερο αριθμό οχημάτων έναντι μιας άλλης. Επομένως κάθε φορά που συγκρίνετε μια λύση με την ολική καλύτερη εξετάζεται ο αριθμός των οχημάτων της μαζί με το ολικό κόστος. Στις επόμενες παραγράφους θα αναφερθεί αναλυτικότερα ο τρόπος λειτουργίας των συναρτήσεων που καλούνται στον κύριο κώδικα.

4.2.1 Η συνάρτηση εύρεσης λύσης

Η συνάρτηση εύρεσης λύσης την οποία καλεί ο κύριος κώδικας αποτελεί το μεγαλύτερο μέρος του συνόλου στον χρόνο απόκρισης του αλγορίθμου.Η συνάρτηση για να δουλέψει χρησιμοποιεί όλα τα δεδομένα του προβλήματος μαζί με τον πίνακα της φερορμόνης και ως αποτέλεσμα επιστρέφει μια λύση. Δηλαδή ένα στρατηγικό πλάνο εκτέλεσης εξυπηρέτησης πελατών, μαζί με δεδομένα το συνολικό κόστος του πλάνου τον χρόνο εκτέλεσης κάθε φάσης του κάθε κύκλου ξεχωριστά, το κόστος και το φορτίο που έχει αναλάβει κάθε κύκλος ξεχωριστά.

Ο αλγόριθμος της συνάρτησης ξεκινάει με μια κύρια

While

η οποία θα έχει ο σκοπό την δημιουργία αρχικής λύσης (δηλαδή να εξυπηρετηθούν όλοι οι πελάτες).

Όσο υπάρχουν πελάτες που δεν έχουν εξυπηρετηθεί δεν σταματάει.

Στην συνέχεια συναντάει μια πεπλεγμένη

While

η οποία φροντίζει στο να δημιουργηθεί ένας κύκλος αυτός ο βρόχος έχει 3 επίπεδα τα οποία διαχωρίζονται με μία if.

If

(1)

Elseif

(2)

Else

(3)

end

κάθε κομμάτι κώδικα από αυτά τα (1),(2),(3) εκτελεί μια σειρά εντολών προκειμένου να βρεθεί συμβατός κόμβος προς εξυπηρέτηση.

(1):

Όταν εκτελείται το (1) σημαίνει ότι ο κύκλος που διαμορφώνεται είναι σε αρχικό στάδιο, δηλαδή ξεκινάει από την αποθήκη για να βρεθεί ο πρώτος πελάτης του κύκλου {1-ψαχνει_νεο_κόμβο}.Σε αυτό το κομμάτι απλά κοιτάει να βρεθεί ο πιο κοντινός στην αποθήκη πελάτης για να τον εξυπηρετήσει, αποθηκεύει σε χώρους μνήμης το κόστος, τις μονάδες χωρητικότητας που καταναλώνει και σε ένα παράλληλο διάνυσμα τον χρόνο που χρειάζεται μέχρι και να εξυπηρετήσει τον πελάτη που επισκέπτεται. Πάνω σε αυτούς τους χώρους μνήμης θα αποθηκεύονται οι αλλαγές σε περίπτωση που ο κύκλος θα μεγαλώσει. Όλους αυτούς τους χώρους μνήμης τους ¨δανείζεται¨ (καθώς όταν ξεκινάει επανάληψη >2 της κύριας while οι μεταβλητές αρχικοποιούνται ξανά =0) κάθε κύκλος που κατασκευάζεται μέχρι να συγκεντρωθούν τα επιθυμητά δεδομένα σε άλλους χώρους μνήμης οι οποίοι θα δουλευτούν μετά την κατασκευή αρχικής λύσης. Ο κόμβος που εξυπηρετείται σβήνεται από τον κεντρικό πίνακα του κόστους - χρόνου- απόστασης A. Σε αυτό το κομμάτι κώδικα πάντα εξυπηρετείται ένας κόμβος, αφού βρεθεί εξυπηρετήσιμος πελάτης με τα κατάλληλα χαρακτηριστικά(κοντινότερος στην αποθήκη σε αυτό το σημείο) τότε συνεχίζει στην επόμενη επανάληψη έχοντας δημιουργηθεί έως τώρα ένας κύκλος της μορφής {1-21-}.

(2):

Όταν εκτελείται το (2) σημαίνει ότι ο κύκλος μέχρις στιγμής έχει ήδη 1 κόμβο μέσα και ψάχνω τον $2^{ο}$ για τον κύκλο , σε αυτό το σημείο υπάρχει μια ακόμα *while* η οποία έχει ως σκοπό να βρεθεί εκείνος ο πελάτης που είναι ταυτόχρονα κοντινότερος (κατά *Clark n Wrigth*) και μπορούν να ικανοποιηθούν τα χρονικά παράθυρά του ,δηλαδή όσο θα ισχύει αυτή η *while* σημαίνει ότι δεν έχει βρεθεί συμβατός κόμβος και συνεχίζεται η αναζήτηση. Ταυτόχρονα εφαρμόζεται η μέθοδος *Clark n Wrigth* με αποτέλεσμα ο αλγόριθμος να κοιτάει τα κοντινότερα βέλη που φεύγουν από κάθε κόμβο μέσα στον κύκλο ,να βρίσκει το μικρότερο και ανάλογα την προέλευση του βέλους ,στην προκειμένη περίπτωση οι εκδοχές είναι δύο είτε από {1-καινουργιος_κομβος-21} είτε από {1-21-καινουργιος_κομβος}, στην συνέχεια πραγματοποιούνται οι κατάλληλες συγκρίσεις για να αποφασιστεί αν αξίζει να μπει στον κύκλο η όχι.Αν εν τέλει αξίζει ,θα σπάσει η *while* που αναφέρθηκε πιο πάνω, αλλιώς ο κόμβος αυτός θα διαγραφεί προσωρινά και μέσω του βρόχου επανάληψης θα εφαρμοστεί το ίδιο κομμάτι κώδικα μέχρι να βρεθεί κόμβος. Όταν βρεθεί θα επιστρέψουν τα στοιχεία των κόμβων που απορρίφθηκαν προσωρινά πίσω στον πίνακα *A* .Αν εν τέλει δεν βρεθεί κανένας συμβατός κόμβος τότε θα σχηματιστεί κύκλος που εξυπηρετεί μόνο έναν πελάτη.

(3):

Αν βρεθεί $2^{ος}$ κόμβος να εξυπηρετηθεί τότε ο αλγόριθμός περνάει στο $3^{ο}$ κομμάτι κώδικα το οποίο είναι και το κομμάτι που εφαρμόζεται η μέθοδος αποικίας μυρμηγκιών. Σε αυτό το κομμάτι το κριτήριο για το ποιος κόμβος θα επιλεγεί προκειμένου να εξεταστεί για το αν θα εισαχθεί στον κύκλο είναι πάλι ο κοντινότερος όπως και στο (2) κομμάτι κώδικα. Έστω ότι ο κύκλος μέχρι τώρα έχει σχηματιστεί ως εξής {1-21-23-}. Αφού έχω βρει τον κοντινότερο κατά Clark n Wrigth τότε προκύπτουν τα σενάρια τοποθέτησής του μέσα στον κύκλο, στην προκειμένη φάση θα είναι τρία και θα είναι τα εξής {1-νεος_κομβος-21-23},{1-21-νεος_κομβος-23},{1-21-23-νεος_κομβος}.Αυτά τα σενάρια τοποθετούνται σε ένα διάνυσμα,

ένα άλλο παράλληλο διάνυσμα με αυτό αποθηκεύει τα υπολογισμένα savings s_{ij} του κάθε σεναρίου αφού έχουν υπολογισθεί, το ίδιο συμβαίνει και για το $\xi ij = [sij]^{\alpha}[\varphi ij]^{\beta}$ δηλαδή για εκθέτη *α* έχω βάση το saving s_{ij} του σεναρίου και για εκθέτη *β* έχω βάση την ποσότητα της φερορμόνης από τον προηγούμενο στο νέο κόμβο του κάθε σεναρίου πχ έστω για σενάριο {1-νεος_κομβος-21-23} η φερορμόνη που έχω σαν βάση είναι από τον πίνακα της φερορμόνης *Φ* το στοιχείο $\Phi_{1,νεος\_κομβος}$. Έτσι υπολογίζονται οι συντελεστές ελκυστικότητας ξ_{ij} του κάθε σεναρίου, με βάση τα ξ_{ij} του κάθε σεναρίου υπολογίζονται οι πιθανότητες αυτού μέσα από την σχέση *4.2* που αναφέρετε πιο πάνω, στην συνέχεια υπολογίζονται τα διαστήματα των πιθανοτήτων στα οποία ανήκει το κάθε σενάριο. Ζητείται από τη MATLAB μια τυχαία πραγματική τιμή από το 0-1 και σε όποιο διάστημα ανήκει αυτή επιλέγεται και το σενάριο το οποίο αντιστοιχεί και θα εξεταστεί.. Σε αυτό το σημείο να αναφερθεί πως το χαρακτηριστικό των χρονικών παραθύρων στο πρόβλημα αποτελεί ισχυρό εμπόδιο στην χρονική εκτέλεση του αλγορίθμου διότι για οποιαδήποτε ενδεχόμενη αλλαγή σε έναν κύκλο θα πρέπει να εξεταστεί η ακεραιότητα των χρονικών παραθύρων για όλους τους πελάτες σε αυτόν. Το παρακάτω παράδειγμα θα βοηθήσει στην κατανόηση του εμποδίου αυτού.

Έστω ο κύκλος μέχρις στιγμής *1-2-22-35-11-1* και το επόμενο σενάριο στο οποίο θα εξεταστεί η εφικτότητα του είναι το *1-**5**-2-22-35-11-1*. Η νέα προσθήκη του κόμβου 5 έτυχε να εξεταστεί μέσα από την θέση που παίρνει στο παραπάνω σενάριο, για να εξεταστεί η εφικτότητα του σεναρίου δεν αρκεί μόνο να υπολογισθεί αν παραβιάζετε το χρονικό παράθυρο του επόμενου κόμβου του 5 δηλαδή τον 2, γιατί έτσι υπάρχει ενδεχόμενο στην συνέχεια της αναζήτησης να διαιωνίζετε ένα χρονικό σφάλμα μέσα στον κύκλο και στην ουσία η παραχθέντα λύση να μην είναι εφικτή. Η χρήση αυτής της απλής συνθήκης θα ήταν λειτουργική αν και μόνο αν το όχημα φτάσει ενωρίτερα από την έναρξη του χρονικού παραθύρου του μεθεπόμενου από τον 2 δηλαδή για το παραπάνω παράδειγμα τον 22. Η φύση του προβλήματος έχει παραπάνω ενδεχόμενα από αυτά, η πιο σίγουρη μέθοδος για να εξεταστεί η εφικτότητα του εκάστοτε σεναρίου είναι να υπολογίζονται από την αρχή οι χρόνοι του οχήματος για να εξεταστούν οι συνθήκες των χρονικών παραθύρων όλων των πελατών στον κύκλο.

Όπως γίνεται κατανοητό μια τέτοια στρατηγική προσφέρει ασφάλεια ως προς την εφικτότητα της λύσης, αλλά μπορεί να εκτελεί παραπάνω επαναλήψεις ο αλγόριθμος χωρίς λόγο, και έτσι να είναι άσκοπα χρονοβόρος. Επομένως, ο αλγόριθμος αποθηκεύει πάντα σε ένα παράλληλο διάνυσμα τους χρόνους που θα εκτελέσει το όχημα για κάθε φάση του κύκλου άρα οι τιμές του διανύσματος έχουν την ώρα της ημέρας που θα έχει τελειώσει το όχημα με την εξυπηρέτηση του κάθε πελάτη στον κύκλο. Αυτή η πληροφορία συνδυάζεται μαζί με την θέση που λαμβάνει ο νέος κόμβος στον κύκλο ώστε σε περίπτωση που θα εξέταζε ένα σενάριο της μορφής *1-2-22-35-**5**-11-1* να υπολογίζεται ο χρόνος από τον κόμβο 35 και μετά έτσι ώστε να εκτελεστεί πιο γρήγορα η διαδικασία.

Το χαρακτηριστικό της πεπλεγμένης *while* που είναι στο (2) κομμάτι κώδικα ισχύει και εδώ που σημαίνει πως ο αλγόριθμος ακόμα και αν συναντήσει μη εφικτά σενάρια θα τα συνεχίζει να ψάχνει μέσα από καινούργιο κόμβο εκείνο το εφικτό σενάριο με γνώμονα την μεθοδολογία που ακολουθεί, όταν έχει επιλέξει όλους του κόμβους που μπορούσε να επιλέξει και δεν έχει καταφέρει να μεγαλώσει περαιτέρω τον κύκλο που κατασκευάζει τότε ο

κύκλος θα κλείσει και θα συνεχίσει στην δημιουργία καινούργιου μέχρι να βγάλει το πλάνο εξυπηρέτησης όλων των πελατών.

Να σημειωθεί πώς όταν συνεχίζει να ψάχνει για τον ίδιο κύκλο εφικτά σενάρια διαγράφει από τις επιλογές του τους κόμβους με τους οποίους προσπάθησε και απέτυχε. Αυτό δεν σημαίνει πως ο κόμβος που απορρίπτει σαν επιλογή δεν είναι ικανός να υπάρξει σε ένα εφικτό σενάριο μέσα στο κύκλο αλλά αν δεν απορριφθεί ο κόμβος σαν επιλογή και απορρίπτεται κάθε φορά το τόξο τότε πρόκειται για έναν αλγόριθμο ο οποίος θα εξετάσει όλες τις πιθανές επιλογές στην εκάστοτε φάση και ο χρόνος που θα χρειάζεται για να παράξει αποτέλεσμα θα είναι πάρα πολύ μεγάλος.

4.2.2 Συνάρτηση τοπικής αναζήτησης πού ενώνει κύκλους

Για κάθε μια λύση από την πεντάδα ο εφαρμόζεται πάνω της τοπική αναζήτηση *1-0 relocate* η οποία είναι μια διαδικασία αναζήτησης συγχώνευσης κύκλων με σκοπό την μείωση του αριθμού των οχημάτων για την λύση. Η κλήση της συνάρτησης αυτής γίνεται μετά την κατασκευή της κάθε λύσης και λαμβάνει περισσότερα δεδομένα από τα υπάρχοντα τα οποία είναι μια ομαδοποιημένη κατανομή των κύκλων με βάση τον αριθμό των οχημάτων. Οι ομάδες των κύκλων που θα προκύψουν μετά από μια συνάρτηση κατασκευής της λύσης χωρίζονται σε τρία μέρη. Η πρώτη ομάδα είναι όσοι κύκλοι εξυπηρετούν μόνο έναν πελάτη και γυρίζουν πίσω στην αποθήκη, η δεύτερη ομάδα είναι όσοι κύκλοι εξυπηρετούν 2 πελάτες και η τελευταία είναι κύκλοι που εξυπηρετούν πάνω από 3 πελάτες.

Οι Συγχωνεύσεις γίνονται πάντα με συγκεκριμένη λογική, πρώτα θα εξεταστεί η συγχώνευση αρχικών κύκλων της $3^{ης}$ ομάδας με όσους κύκλους έχουν δημιουργηθεί στην $2^{η}$ ομάδα και στην συνέχεια αν και εφόσον υπάρχουν κύκλοι της $2^{ης}$ ομάδας που δεν έχουν συγχωνευθεί με κάποιους τότε εξετάζεται ένωση της $2^{ης}$ ομάδας με κύκλους από την $1^{η}$ ομάδα. Οι ενώσεις και για τις δυο περιπτώσεις κοιτάνε τον χρόνο του κύκλου μέχρι το σημείο του τελευταίου πελάτη-κόμβου που ανήκει στην ομάδα με του μεγαλύτερος σε μήκος κύκλους και στην συνέχεια κοιτάνε την έναρξη του χρονικού παραθύρου του πρώτου πελάτη κόμβου του κύκλου που ανήκει στην μικρότερη σε μήκος κατηγορία κύκλον. Η συνθήκη η οποία πραγματοποιείτε εξετάζει αν ο χρόνος για να πάει από τον τελευταίο πελάτη-κόμβο του μεγάλου κύκλου στο πρώτο πελάτη-κόμβο του μικρότερου κύκλου είναι μικρότερος από το χρονικό παράθυρο του τελευταίου, τότε σημαίνει ότι μπορούν οι κύκλοι να ενωθούν χωρίς να χρειάζεται νέος υπολογισμός γιατί σε αυτή την περίπτωση το όχημα θα πρέπει να περιμένει την έναρξη του χρονικού παραθύρου του νέου κόμβου και οι υπόλοιποι χρόνοι του οχήματος είναι υπολογισμένοι καθώς το κομμάτι που προστίθεται αποτελούσε έναν κύκλο στην μη ανανεωμένη λύση.

Η διαδικασία που ακολουθείτε για την συγχώνευση των κύκλων ξεκινάει με την συλλογή δεδομένων η οποία συλλέγει από όλους τους κύκλους τους απαραίτητους κόμβους που βρίσκονται στις άκρες ανάλογα με την ένωση που επιθυμεί να πραγματοποιήσει. Έπειτα κοιτάει για κάθε έναν κύκλο που ανήκει στην ομάδα με το μεγαλύτερο μήκος αν υπάρχει κάποια συμβατότητα με κύκλο της άλλης ομάδας. Αν υπάρχει συμβατότητα με πάνω από έναν κύκλους τότε διαλέγετε εκείνος ο κύκλος του οποίου το νέο τόξο που θα χρειαστεί για την ένωση προσφέρει το λιγότερο κόστος.

Σαν αποτέλεσμα η παραπάνω τοπική αναζήτηση πραγματοποιεί συγχωνεύσεις με επιτυχία στις αρχικές επαναλήψεις του κύριου κώδικα καθώς τότε παρουσιάζονται περισσότεροι κύκλοι που εξυπηρετούν έναν και δυο πελάτες μόνο. Όσο προχωράει σε αριθμό επαναλήψεων ο κύριος κώδικας τότε η συνάρτηση βρίσκει όλο και λιγότερες συγχωνεύσεις διότι σταματάνε να παρουσιάζονται τέτοιοι κύκλοι

4.2.3 Συνάρτηση τοπικής αναζήτησης 1-1 ανταλλαγή (1-1 exchange)

Αφού επιλεχθεί η καλύτερη λύση από την πεντάδα τότε μέσα από τον κύριο κώδικα θα καλεστεί η συνάρτηση τοπικής αναζήτησης που εφαρμόζει την 1-1 exchange μέθοδο. Για να εφαρμοστεί η μεθοδολογία θα χρειαστεί δυο κύκλους για κάθε αναζήτηση, οπότε ο αλγόριθμος διαλέγει τυχαία δυο διαφορετικούς κύκλους και στην συνέχεια διαλέγει πάλι τυχαία από ένα κόμβο για τον κάθε κύκλο. Για αυτούς του κόμβους που επιλέχθηκαν τυχαία θα δημιουργηθούν οι νέοι κύκλοι αλλάζοντας θέσει τους κόμβους μεταξύ τους ανάμεσα στους κύκλους. Έχοντας δημιουργήσει την ενδεχόμενη αλλαγή γίνεται εξέταση εφικτότητας για τους περιορισμούς χρονικών παραθύρων και ζήτησης ταυτόχρονα με τον υπολογισμό της εξοικονόμησης κόστους από αυτήν την ενδεχόμενη αλλαγή. Αν ισχύουν και υπάρχει εξοικονόμηση και δεν παραβιάζονται οι περιορισμοί τότε πραγματοποιείτε η αλλαγή για την λύση.

Η παραπάνω διαδικασία αφορά μια μονάχα αλλαγή ανάμεσα σε 2 κύκλους. Για αυτό τον λόγο το κομμάτι κώδικα που εκτελεί τα παραπάνω είναι μέσα σε έναν μεγάλο βρόχο επανάληψης *while* ο οποίος τρέχει για προκαθορισμένο αριθμό επαναλήψεων συνήθως για 50 επαναλήψεις κάθε φορά που καλείτε. Αυτό σημαίνει πως όταν ο κύριος κώδικας εκτελεστεί 50 φορές το παραπάνω κομμάτι θα κληθεί 50 φορές άρα στο σύνολο θα εκτελεστεί 2.500 φορές, με βάση αυτά τα νούμερα υπολογίστηκε από την εντολή χρονομέτρησης της MATLAB πως η συνάρτηση που εκτελεί το 1-1exchange καταναλώνει 1 λεπτό του συνολικού χρόνου του αλγορίθμου.

Παρακάτω θα δοθεί ένα παράδειγμα ψευδοκώδικα /με βάση την λειτουργία της συνάρτησης για την πραγματοποίηση της μεθόδου 1-1 ανταλλαγής:

Για 1 έως 100 επαναλήψεις

Διάλεξε έναν κύκλο τυχαία

Διάλεξε έναν 2ο κύκλο τυχαία

Αν δεν είναι ίδιοι

Διάλεξε έναν κόμβο τυχαία από τον 1ο κύκλο

Διάλεξε έναν κόμβο τυχαία από τον 2^{ο} κύκλο

Κατασκεύασε τους νέους κύκλους προς εξέταση αλλάζοντας τους κόμβους μεταξύ τους

Αν βελτιώνουν την λύση και είναι πραγματοποιήσιμες οι αλλαγές

Καταχώρισε τις αλλαγές

Αλλιώς

Εμφάνισε (δεν γίνεται η αλλαγή)

Τέλος

Τέλος

Τέλος

Αυτή η συνάρτηση βρίσκει βελτιώσεις με επιτυχία αλλά για να βελτιώσει μια τοπικώς ελάχιστη λύση αρκετά χρειάζεται πολύ παραπάνω αριθμό επαναλήψεων. Αυτό έχει ως αποτέλεσμα μέσα από την επαναληπτική διαδικασία του κύριου αλγορίθμου η συνάρτηση να αποτελεί όλο και μεγαλύτερο όγκο του τελικού χρόνου και να ανεβαίνει κατά πολύ ο συνολικός χρόνος εκτέλεσης του αλγορίθμου το οποίο είναι ανεπιθύμητο. Κάθε φορά που καλείτε είναι άγνωστο αν θα βρεθεί τουλάχιστον μια βελτίωση της λύσης, το γεγονός ότι δουλεύει με απόλυτη τυχαιότητα δεν εγγυάται σίγουρη βελτίωση σε ορισμένο εύρος επαναλήψεων. Προφανώς όσο πιο μεγάλο είναι το εύρος τόσο πιο αποδοτικό θα είναι. Μέσα από πειραματική διαδικασία ο αριθμός των επαναλήψεων ορίστηκε σε 50 επαναλήψεις διότι υπάρχει ισχυρότερη πιθανότητα να βρίσκει τουλάχιστον μια εφικτή αλλαγή που βελτιώνει το κόστος χωρίς να είναι μεγάλο βάρος για τον συνολικό χρόνο.

4.2.4 Συνάρτηση τοπικής αναζήτησης 2opt

Το τελευταίο φίλτρο που περνάει η τοπικώς ελάχιστη λύση είναι η συνάρτηση τοπικής αναζήτησης 2opt, σε αυτή την περίπτωση αναζητούνται αλλαγές μέσα στον ίδιο κύκλο. Η μέθοδος 2opt έχει εφαρμογή μόνο σε κύκλους που εξυπηρετούν 3 πελάτες και πάνω καθώς ψάχνει αλλαγές των τόξων μέσα στον ίδιο τον κύκλο κάθε φορά και για έναν κύκλο που εξυπηρετεί 2 πελάτες δεν μπορούν να υπάρξουν αλλαγές.

Επομένως ο αλγόριθμος για κύκλους με τα κατάλληλα χαρακτηριστικά θα εφαρμόσει αναζήτηση 2opt. Σε αντίθεση με την τυχαία λειτουργία της 1-1 exchange συνάρτησης η συνάρτηση 2opt ακολουθεί μια καθόλου τυχαία λογική για την επιλογή του τόξου που θα αναζητήσει να αλλάξει. Αφού επιλεχθεί ένας κύκλος στον οποίον είναι δυνατή η εφαρμογή

της μεθόδου, το επόμενο βήμα του κώδικα είναι να συλλέξει από τον κύκλο όλα τα τόξα του. Στην συνέχεια διαλέγει εκείνο το τόξο με το μεγαλύτερο κόστος στον κύκλο, για να ολοκληρωθεί η μέθοδος πρέπει να εξεταστεί το ενδεχόμενο εφικτότητα για τους περιορισμούς χρονικών παραθύρων ενός νέου κύκλου, δεν χρειάζεται να γίνει καμία αλλαγή η έλεγχος όσον αφορά την ζήτηση που θα εξυπηρετήσει ο νέος κύκλος καθώς οι κόμβοι μέσα στον κύκλο παραμένουν οι ίδιοι άρα και το φορτίο του νέου κύκλου παραμένει σταθερό. Η επιλογή του δεύτερου τόξου για την αλλαγή μέσα στον κύκλο γίνεται με βάση την θέση που έχει το πιο κοστοβόρο τόξο μέσα στο κύκλο. Αν το τόξο αυτό βρίσκεται μετά την μέση του κύκλου τότε θα εξεταστεί ενδεχόμενη αλλαγή με ένα τόξο δυο θέσης πίσω από την θέση που βρίσκεται το πιο ακριβό τόξο αλλιώς ,αν δηλαδή βρίσκεται στο πρώτο μισό του κύκλου θα κοιτάξει ενδεχόμενη αλλαγή δυο θέσης μπροστά. Ο λόγος για τον οποίο οι αποστάσεις είναι μόνο δύο θέσεις μπροστά η δυο θέσεις πίσω είναι γιατί η μέθοδος εφαρμόζεται στο πρόβλημα με τα χρονικά παράθυρα τα οποία αποτελούν μεγάλο εμπόδιο. Τέλος δεν υπάρχει λόγος να εξεταστεί ενδεχόμενη αλλαγή για μεγαλύτερη απόσταση από αυτή που προαναφέρθηκε διότι όσο πιο μεγάλη είναι η απόσταση από το τόξο τόσο πιο μεγάλη θα είναι και η χρονική απόσταση μεταξύ των κόμβων μαζί με των χρονικών παραθύρων ταυτόχρονα.

Ο ψευδοκώδικας εφαρμογής της μεθόδου 2opt:

Για 1 έως το πλήθος όλων των κύκλων κύκλων

Αν το μήκος του κύκλου είναι κατάλληλο

Διάλεξε το μεγαλύτερο τόξο στον κύκλο

Ανάλογα με την θέση του κατασκεύασε τον νέο κύκλο που θα ελεγχθεί για βελτίωση της λύσης

Αν βελτιώνει την λύση και είναι πραγματοποιήσιμες οι αλλαγές

Καταχώρησε τις αλλαγές

Αλλιώς

Εμφάνισε (δεν γίνεται αλλαγή).

Τέλος

Τέλος

Τέλος

Κεφάλαιο 5:Παρουσίαση αποτελεσμάτων

5.1 Αναφορά στα δεδομένα και αποτελέσματα

Τα δεδομένα αφορούν το πρόβλημα δρομολόγησης οχημάτων με πεπερασμένη χωρητικότητα. Ο αλγόριθμος εκτελέστηκε σε 14 δεδομένα της βιβλιογραφίας με το κάθε ένα αρχείο δεδομένων να έχει διαφορετικά χαρακτηριστικά σε αριθμό πελάτων, συνολικό χρόνο εκτέλεσης κύκλου και χωρητικότητας. Ανάμεσα τους δεν υπάρχουν δεδομένα με άπειρο χρόνο εκτέλεσης κύκλου και μηδενική τιμή στον χρόνο εξυπηρέτησης των πελατών. Στην συνέχεια παρουσιάζονται οι λύσεις που έδωσε .

	Βέλτιστες λύσεις	Λύσεις αλγορίθμου	Απόκλιση από τις βέλτιστες
par1	524,61	609,9798	16,2730028
par2	845,24	1121,24	32,65344754
par3	832,91	910,3062	9,29226447
par4	1049,24	1180,7852	12,53718882
par5	1361,24	1855,0458	36,27617466
par6	555,43	687,3721	23,75494662
par7	918,27	1124,6583	22,4757751
par8	889,12	1295,6367	45,72124123
par9	1197,23	1711,811	42,98096439
par10	1428,23	1979,8889	38,62535446
par11	1057,01	1186,0391	12,20698953
par12	827,21	1003,6	21,32348497
par13	1551,21	1948,5277	25,61340502
par14	876,45	1151,5361	31,38639968

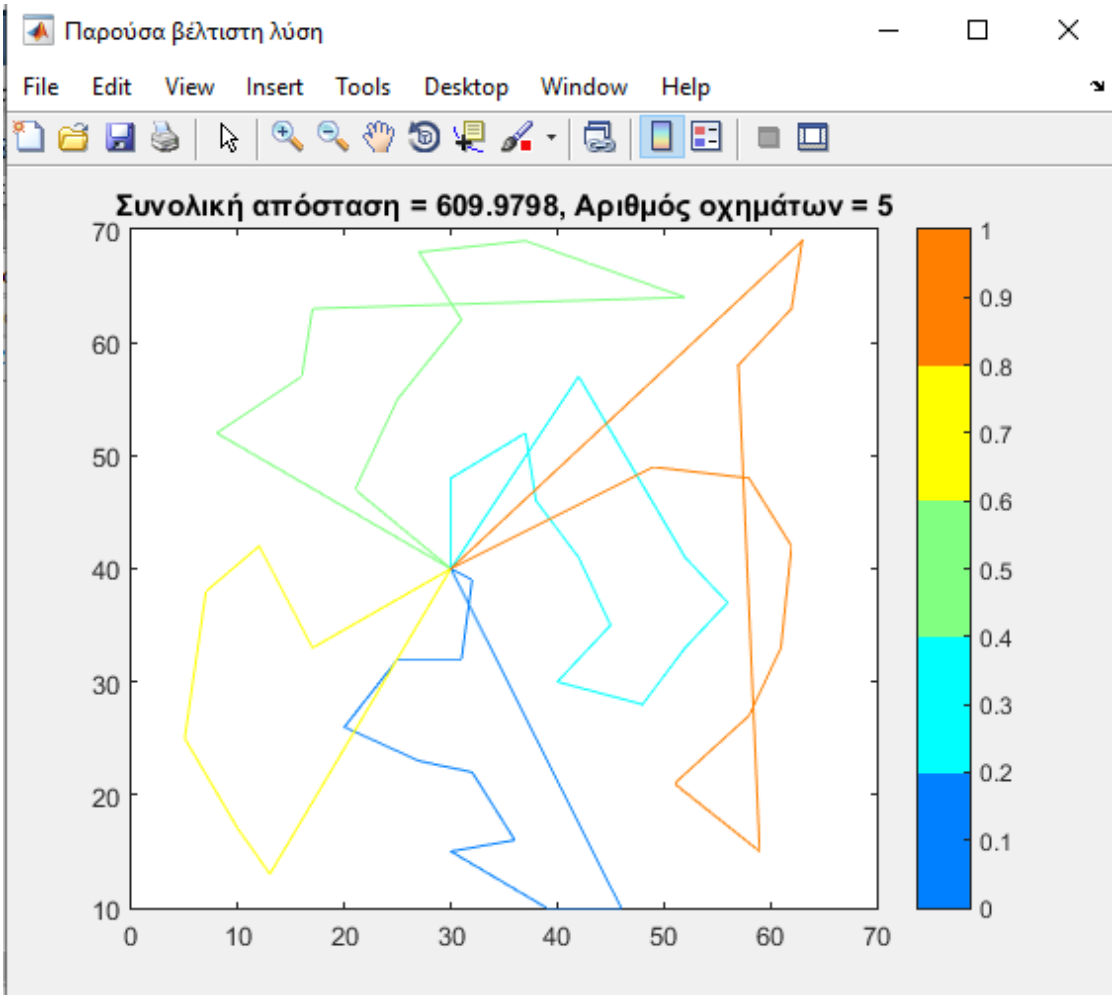
Πίνακας 5.1

Ο παραπάνω πίνακας δείχνει τα αποτελέσματα που εμφάνισε ο αλγόριθμος και τις αποκλίσεις του από τις βέλτιστες λύσεις που υπάρχουν.

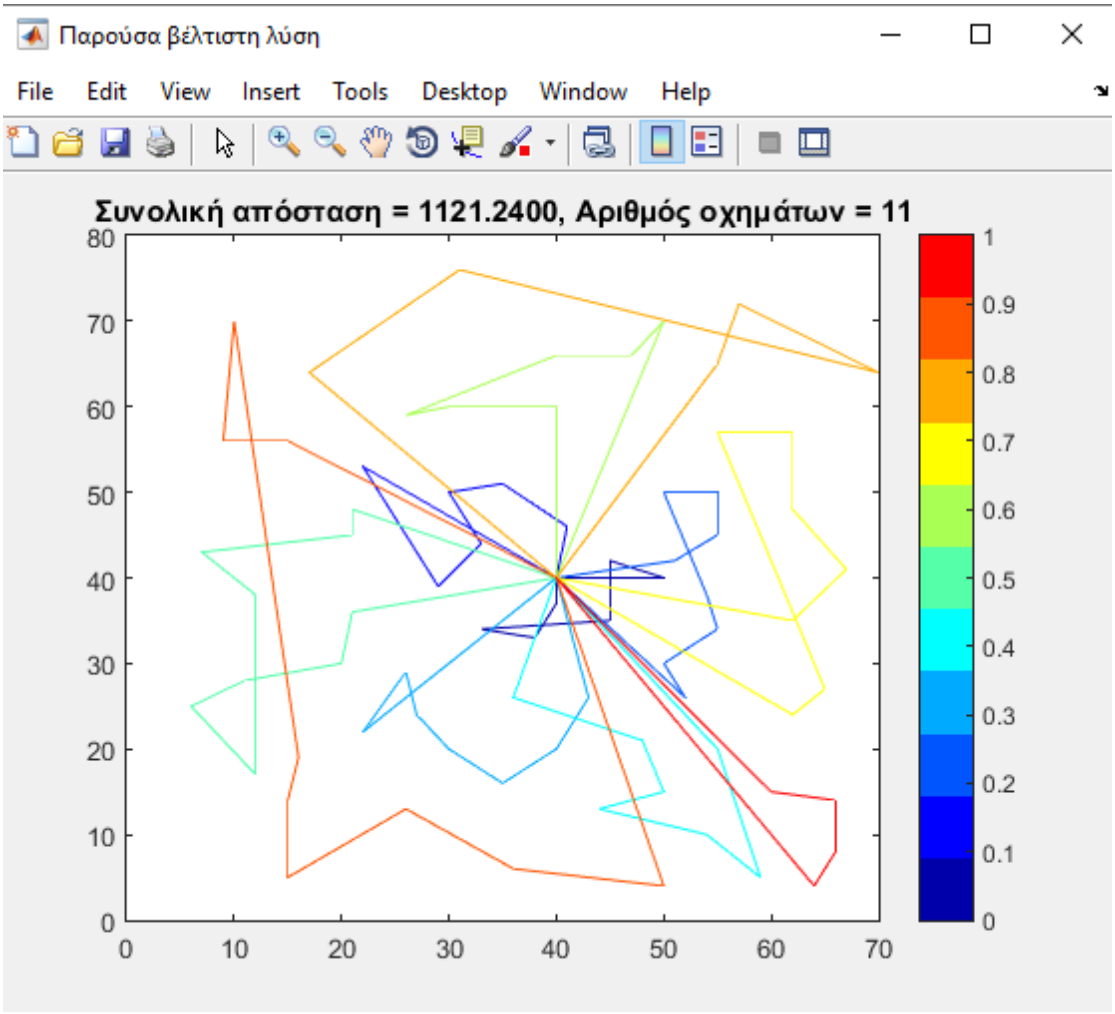
5.2 Γραφήματα αποτελεσμάτων

Παρακάτω παρουσιάζονται γραφικά μερικά από τα αποτελέσματα που προέκυψαν :

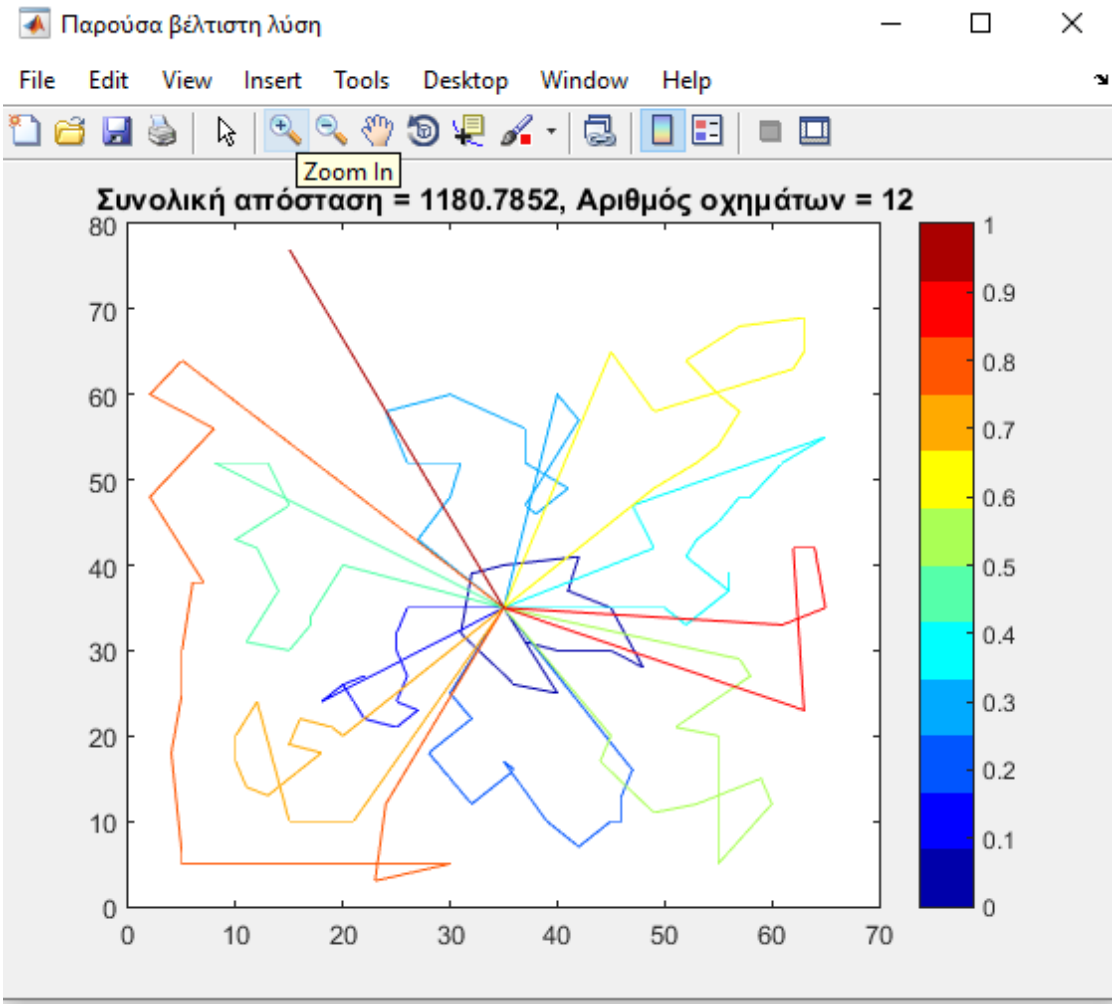
Par1:



Par2:


Παρούσα βέλτιστη λύση
File Edit View Insert Tools Desktop Window Help
Συνολική απόσταση = 1121.2400, Αριθμός οχημάτων = 11
0 10 20 30 40 50 60 70
0 10 20 30 40 50 60 70 80
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

Par4:

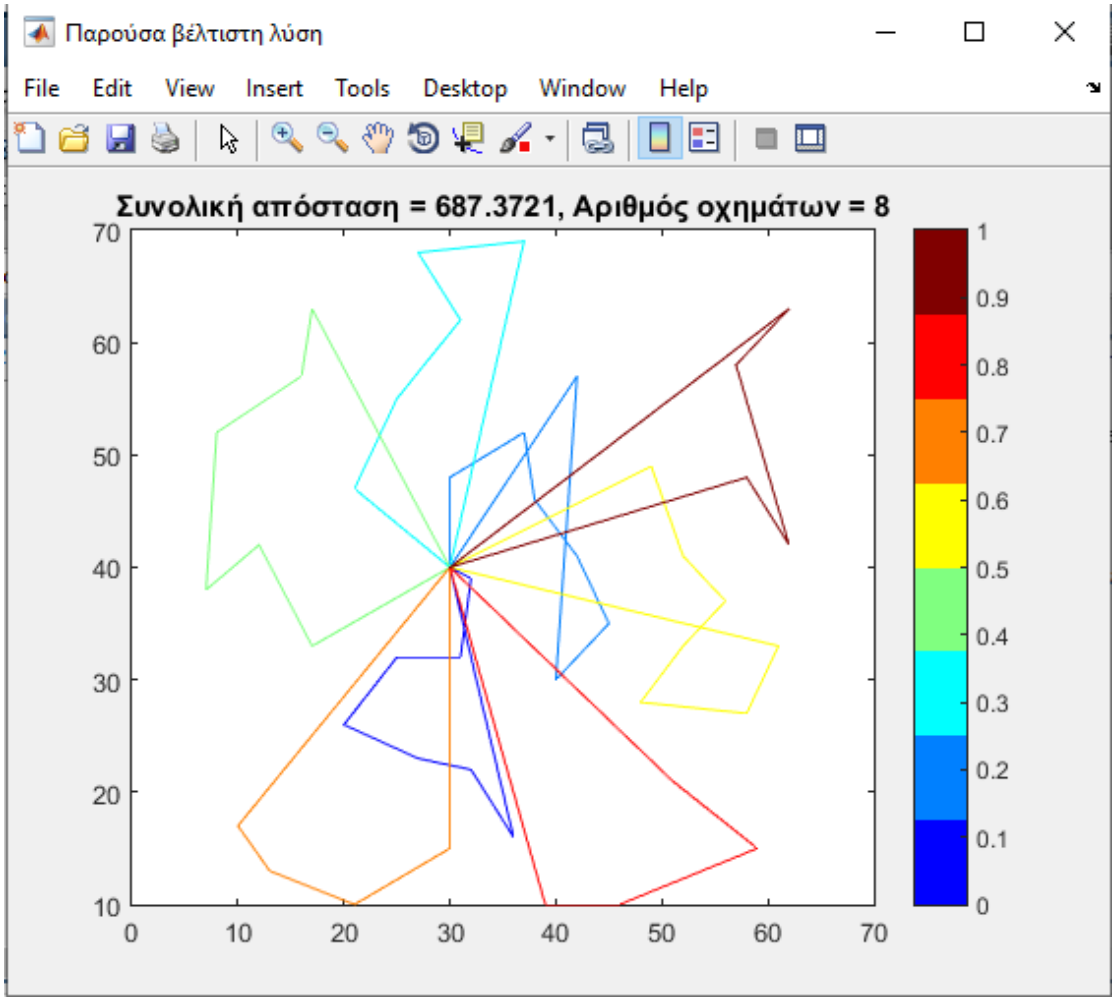


Αναλυτικότερα οι κύκλοι για αυτό το παράδειγμα:

1	54	106	27	150	139	29	112	28	147	113	59
	41	1									
1	90	148	7	95	96	118	98	93	60	100	97
	105	94	1								
1	14	138	88	145	58	116	3	146	42	23	134
	75	73	1								
1	53	128	32	89	149	63	11	71	102	2	133
	70	123	31	1							
1	13	110	81	69	151	117	78	4	80	130	79
	35	51	77	1							
1	19	61	119	6	85	84	115	9	83	49	125
	1										

1	22 1	74	76	57	140	40	24	5	111	131	55
1	103 21	34 1	82	121	10	104	72	137	36	136	52
1	38 143	99 1	86	92	101	120	45	142	17	62	15
1	43 48	44 37	16 144	39 1	141	87	114	18	46	126	47
1	135	25	30	122	56	1					
1	65	1									

Par6:

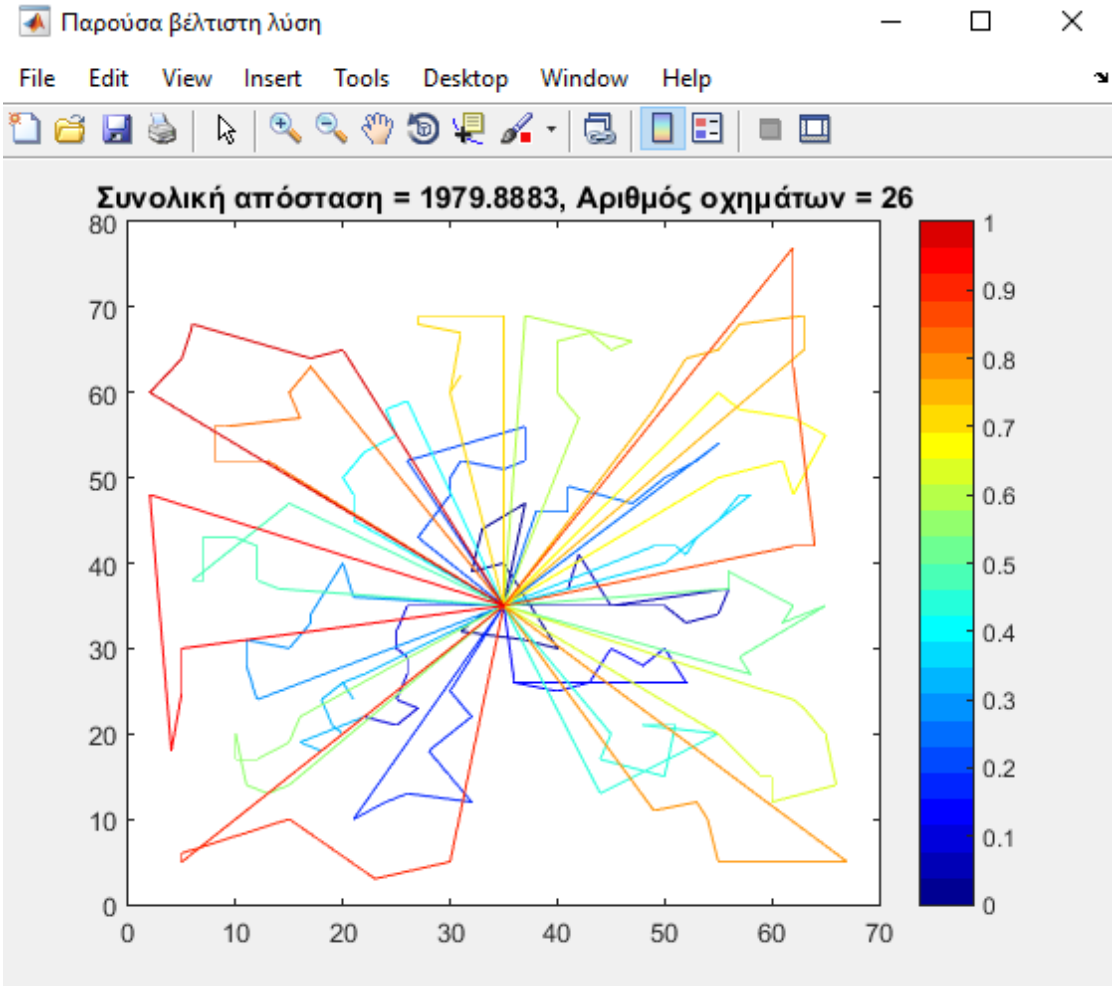


Αναλυτικότερα οι κύκλοι:

1	47 12	13 39	48 6	5 23	18 1	38	16	11	28	2	33

1	28	2	33	12	39	6	23	1
1	7	49	9	27	32	1		
1	19	15	26	25	24	8	1	
1	3	17	51	10	50	31	35	1
1	45	43	20	42	1			
1	11	40	34	46	1			
1	30	22	21	36	1			

Par10



Κεφάλαιο 6:Συμπεράσματα

Τελειώνοντας αυτή τη διπλωματική θα πούμε μερικά γενικά συμπεράσματα για όλη την διαδικασία. Αρχικά προσπαθήσαμε να τροποποιήσουμε τον αλγόριθμο και να χρησιμοποιήσουμε αντί για ένα κλασικό αλγόριθμου πλησιέστερου γείτονα για την τροφοδοσία του αλγόριθμου αποικίας μυρμηγκιών, ένα αλγόριθμο εξοικονομήσεων Clark and Wright. Ο τρόπος που τον χρησιμοποιήσαμε αύξησε την πολυπλοκότητα, χωρίς να βοηθήσει στην βελτίωση των αποτελεσμάτων τόσο πολύ ώστε να δώσει τις βέλτιστες λύσεις. Θα μπορούσαμε να αλλάξουμε τον τρόπο δημιουργίας των λύσεων και να ακολουθήσουμε την βιβλιογραφία, αλλά η λογική της συγκεκριμένης διπλωματικής ήταν να δούμε αν μπορούν αυτοί οι 2 αλγόριθμοι να συνδυαστούν αποτελεσματικά.

Επίσης επειδή θέλαμε να έχουμε ένα αλγόριθμο που είναι όσο το δυνατό πιο γενικώς γίνεται και που μπορεί να χρησιμοποιηθεί σε όσο το δυνατό περισσότερες παραλλαγές, τον δοκιμάσαμε σε προβλήματα δρομολόγησης χωρίς χρονικά παράθυρα, χωρίς όμως να απενεργοποιήσουμε τα χρονικά παράθυρα από τον αλγόριθμο. Έτσι δίνουμε τη δυνατότητα στο χρήστη να μπορεί να προσθέτει ή να αφαιρεί τα χρονικά παράθυρα ανάλογα με το αν υπάρχουν ή όχι στην εφαρμογή που θέλει να τρέξει. Αυτό το τελευταίο ήταν που οδήγησε και στο να μην φτάσουμε σε πάρα πολύ καλές λύσεις, γιατί εμείς θέλαμε εξαρχής αν δούμε αν θα μπορούσαμε να δώσουμε ένα πιο γενικευμένο μοντέλο. Σε αυτή τη φάση που βρίσκεται ο αλγόριθμος είναι αποτελεσματικός για μία ρεαλιστική εφαρμογή με πραγματικά δεδομένα, άρα μπορεί να χρησιμοποιηθεί εύκολα από κάποιον που έχει ένα πραγματικό πρόβλημα, αλλά δεν είναι τόσο ανταγωνιστικός με άλλες υλοποιήσεις της συγκεκριμένης μεθοδολογίας για αντίστοιχου τύπου προβλήματα.

Βιβλιογραφία

[1]: Μαρινάκης Ι., Μ. Α. (2008). *Σχεδιασμός και Βελτιστοποίηση της Εφοδιαστικής Αλυσίδας.* Θεσσαλονίκη: Σοφία.

[2]: D-Ants: Savings Based Ants divide and conquer the vehicle
routing problem
Marc Reimann∗, Karl Doerner, Richard F. Hartl

[3]: Γ., Μουρκούσης. (2008). *Μεθοδολογίες Ανάπτυξης Πληροφοριακών Συστημάτων Υποστήριξης Λήψης.* Patras.

[4]: Toth, P., & Vigo, D. (2002). *The Vehicle Routing Problem.* Society for Industrial and Applied Mathematics.

[5]: Papadimitrioy , C., & Steiglitz, K. (1982). *Combinatorial Otimization - Aglorithms and Complexity.* New Jersey: Prentice - Hall.

[6]: Festa P., Resende. M. (2002). *Essays and surveys in metaheuristics.* Springer US.

[7]: S., Lin. (1965). Computer Solutions of the Traveling Salesman Problem. *Technical Journal, 44*,