ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Διπλωματική εργασία με θέμα:

«Ανάπτυξη μεθευρετικού αλγορίθμου για την επίλυση του προβλήματος Δρομολόγησης Οχημάτων με χρονικά διαστήματα και παραλαβές και διανομές»

υπό τον

Σερελέα Κωνσταντίνο

Επιβλέπων καθηγητής: Αθανάσιος Μυγδαλάς

Χανιά

Ιούλιος 2004

Ευχαριστίες

Ευχαριστώ θερμά όλους όσους βοήθησαν στην πραγμάτωση αυτής της διπλωματικής εργασίας και κυρίως τον καθηγητή μου Αθανάσιο Μυγδαλά για τις απαραίτητες κατευθύνσεις και οδηγίες του καθώς και τον υποψήφιο διδάκτωρ κ. Γιάννη Μαρινάκη για τις πολύτιμες υποδείξεις του. Επίσης θα ήθελα να ευχαριστήσω την οικογενειά μου και τους φίλους μου για την ανοχή και την υποστήριξη που μου παρείχανε όλα αυτά τα χρόνια.

Περίληψη εργασίας

Στην παρούσα εργασία γίνεται η παρουσίαση ενός αλγορίθμου επίλυσης του προβλήματος Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και Παραλαβές και Διανομές και βελτιστοποίση των αποτελεσμάτων του με τη βοήθεια των αλγορίθμων τοπικής αναζήτησης 2- opt και 3-opt.

Συνοπτικά :

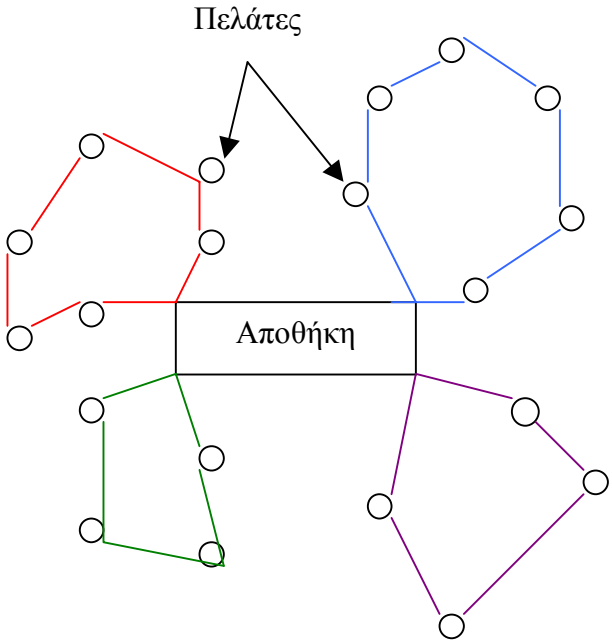
- Στο κεφάλαιο 1 γίνεται μια λεπτομερής εισαγωγή στο πρόβλημα Δρομολόγησης Οχημάτων.
- Στο κεφάλαιο 2 γίνεται αναλυτική παρουσίαση των προβλημάτων:
    1. Του περιπλανώμενου πωλητή
    2. Δρομολόγησης Οχημάτων και ορισμένων προεκτάσεων του, μια εκ των οποίων είναι και το πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και Παραλαβές και Διανομές
- Στο κεφάλαιο 3 γίνεται παρουσίαση ευρετικών και μεθευρετικών αλγορίθμων επίλυσης του προβλήματος Δρομολόγησης Οχημάτων.
- Στο κεφάλαιο 4 παρουσιάζεται αναλυτικά ο αλγόριθμος επίλυσης του προβλήματος Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και Παραλαβές και Διανομές και οι αλγορίθμοι τοπικής αναζήτησης 2- opt και 3-opt.
- Στο κεφάλαιο 5, τέλος, γίνεται αναλυτική εφαρμογή του αλγορίθμου σε δύο προβλήματα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και Παραλαβές και Διανομές.

Κεφάλαιο 1: Εισαγωγή στο πρόβλημα Δρομολόγησης Οχημάτων

1.1 Το πρόβλημα δρομολόγησης οχημάτων (VEHICLE ROUTING PROBLEM)

Το Πρόβλημα Δρομολόγησης Οχημάτων (VRP) αποτελεί ένα από τα σημαντικότερα και πιο μελετημένα προβλήματα διανομής της συνδυαστικής βελτιστοποίησης. Για την επίλυση του συγκεκριμένου προβλήματος πρέπει να υπολογισθούν οι βέλτιστες διαδρομές διανομής προϊόντων που πρέπει να εκτελεσθούν από ένα σύνολο οχημάτων έτσι ώστε να ικανοποιηθεί η ζήτηση των πελατών. Ο σκοπός του Προβλήματος Δρομολόγησης Οχημάτων είναι η ελαχιστοποίηση του κόστους μεταφοράς προϊόντων ενός στόλου οχημάτων που ξεκινούν και καταλήγουν σε μία αποθήκη. Για το συγκεκριμένο πρόβλημα υπάρχει ένας αρκετά μεγάλος αριθμός διαφορετικών πραγματικών περιπτώσεων που έχουν να κάνουν με την ύπαρξη κάποιων περιορισμών ως προς τη διανομή των προϊόντων και σε μερικές περιπτώσεις συσχετίζεται με το Πρόβλημα του Πλανόδιου Πωλητή (TSP).

Όπως αναφέρθηκε και παραπάνω, το Πρόβλημα Δρομολόγησης οχημάτων αφορά την εξυπηρέτηση ενός αριθμού πελατών σε μια συγκεκριμένη χρονική περίοδο και από ένα δεδομένο αριθμό οχημάτων. Τα οχήματα αυτά εκκινούν από ένα ή και περισσότερα κέντρα διανομής και μεταφέρουν τα εμπορεύματά τους σε ένα κατάλληλο οδικό δίκτυο. Για την επίλυση του συγκεκριμένου προβλήματος απαιτείται ο υπολογισμός των διαδρομών που θα εκτελέσει κάθε όχημα ξεχωριστά με αφετηρία και τέλος ένα συγκεκριμένο κέντρο διανομής (αποθήκη). Παρακάτω ακολουθεί ένα διάγραμμα το οποίο περιγράφει τη φύση του προβλήματος που εξετάζουμε.



Σχήμα 1: Αναπαράσταση Προβλήματος Δρομολόγησης Οχημάτων

Στο παραπάνω σχήμα παρατηρούμε ένα οδικό δίκτυο που έχει ως χαρακτηριστικά:

1. Την αποθήκη που βρίσκεται στο κέντρο του διαγράμματος και από την οποία εκκινούν τα οχήματα για να εκτελέσουν τα δρομολόγια τους. Συχνά συναντάται το φαινόμενο της ύπαρξης περισσότερων του ενός κέντρων διανομής το καθένα από τα οποία εκτελεί τις διεργασίες του αυτόνομα. Σε αυτές τις περιπτώσεις το συγκεκριμένο πρόβλημα αντιμετωπίζεται ως ένα σύνθετο πρόβλημα πολλών μεμονωμένων προβλημάτων δρομολόγησης οχημάτων. Τέλος, όταν υπάρχουν περισσότερες από μια αποθήκες παρατηρείται και το φαινόμενο τα οχήματα να εκκινούν και να καταλήγουν σε διαφορετική αποθήκη.
2. Οι διαδρομές που εκτελούν τα οχήματα και που παριστάνονται από τα τόξα του γραφήματος. Τα τόξα μπορεί να είναι προσανατολισμένα ή μη ανάλογα με το αν η διαδρομή μπορεί να διασχιστεί και προς τις δύο κατευθύνσεις ή είναι μονής κατεύθυνσης. Κάθε τόξο έχει ένα συγκεκριμένο κόστος το οποίο αντιπροσωπεύεται από το μήκος του και χαρακτηρίζεται από ένα δεδομένο χρόνο διαδρομής που εξαρτάται από την περίοδο διάσχισης του συγκεκριμένου τόξου και τον τύπο των οχημάτων .

3. Τέλος, οι κορυφές του γραφήματος αντικατοπτρίζουν τους πελάτες στους οποίους καταλήγουν τα οχήματα. Κάθε πελάτης έχει δεδομένες απαιτήσεις ως προς την ποσότητα και το είδος των προϊόντων που περιμένει να του διοχετευθούν από την αποθήκη. Επιπλέον κάθε πελάτης έχει συγκεκριμένες χρονικές περιόδους (κατά τη διάρκεια της ημέρας) στις οποίες μπορεί να εξυπηρετηθεί και χρειάζεται ένα δεδομένο χρόνο (ο οποίος εξαρτάται και από τον τύπο του οχήματος) για να συλλέξει ή να διανείμει τα εκάστοτε προϊόντα. Τέλος, τυπικά χαρακτηριστικά των πελατών αποτελούν τα είδη και τα τεχνικά χαρακτηριστικά των οχημάτων που χρησιμοποιούνται (πιθανότατα λόγω περιορισμών στο φορτίο ή περιορισμών στην κυκλοφορία). Υπάρχουν, ωστόσο αρκετές περιπτώσεις προβλημάτων στις οποίες είναι αδύνατη η ικανοποίηση της ζήτησης όλων των πελατών οπότε επιλέγεται είτε η μείωση των ποσοτήτων διανομής ή συλλογής, είτε η μη εξυπηρέτηση ενός μικρού αριθμού πελατών. Η επιλογή της κατάλληλης πολιτικής σε αυτήν την περίπτωση προσδιορίζεται από τις εκάστοτε προτεραιότητες ή και ποινές που σχετίζονται με τη μερική ή ολική έλλειψη εξυπηρέτησης.

Για την μεταφορά των προϊόντων χρησιμοποιείται ένας αριθμός οχημάτων των οποίων το μέγεθος εξαρτάται από τις απαιτήσεις των πελατών. Τα τυπικά χαρακτηριστικά των οχημάτων είναι:

- Το κέντρο διανομής κάθε οχήματος και η πιθανότητα το όχημα να καταλήξει στην ίδια αποθήκη από την οποία εκκίνησε.
- Η χωρητικότητα κάθε οχήματος που μπορεί να εκφρασθεί ως το μέγιστο βάρος ή ως ο μέγιστος όγκος ή ακόμη και ως ο μέγιστος αριθμός πακέτων που μπορεί το εκάστοτε όχημα να μεταφέρει.
- Ο διαχωρισμός του αποθηκευτικού χώρου του οχήματος σε περιπτώσεις που υπάρχουν περισσότερα από ένα προϊόντα που χαρακτηρίζεται από τον αριθμό, το είδος και την χωρητικότητα των διαφορετικών προϊόντων.
- Οι διαδρομές από τις οποίες μπορεί να μεταβεί το όχημα.
- Το κόστος χρήσης και κίνησης του οχήματος.
- Οι μέθοδοι φόρτωσης-εκφόρτωσης των προϊόντων.

Επιπλέον αναφοράς χρίζουν και οι περιορισμοί στους οποίους εμπίπτουν οι οδηγοί των οχημάτων. Οι περιορισμοί αυτοί έχουν να κάνουν με τις εργασιακές συμβάσεις που οι ίδιοι έχουν υπογράψει με την εταιρία και με τους κανονισμούς της εταιρίας (το ωράριο εργασίας, οι υπερωρίες, η μέγιστη διάρκεια οδήγησης και άλλα).

Κατά την επίλυση του Προβλήματος Δρομολόγησης Οχημάτων θα πρέπει οι διαδρομές που θα προκύψουν να ικανοποιούν ορισμένους περιορισμούς που εξαρτώνται από τα χαρακτηριστικά των πελατών και των οχημάτων, τη φύση και ευπάθεια των προϊόντων, καθώς και από την πολιτική της εταιρίας ως προς το επίπεδο εξυπηρέτησης που έχει επιλέξει να παρέχει. Έτσι λοιπόν σε κάθε διαδρομή το συνολικό φορτίο εμπορευμάτων δε θα πρέπει να υπερβαίνει τη χωρητικότητα του φορτηγού, οι πελάτες της συγκεκριμένης διαδρομής μπορούν να λαμβάνουν ή ακόμη και να διανέμουν προϊόντα και παράλληλα να εξυπηρετούνται σε δεδομένα χρονικά διαστήματα. Παράλληλα είναι αναγκαίο να ικανοποιούνται και περιορισμοί που έχουν να κάνουν με την προτεραιότητα ορισμένων πελατών έναντι άλλων είτε αυτοί είναι μεμονωμένοι είτε είναι ένα υποσύνολο πελατών. Χαρακτηριστικό παράδειγμα τέτοιας περίπτωσης είναι και το πρόβλημα συλλογής και διανομής (VRPPD) στο οποίο σε κάθε διαδρομή που κάνει ένα όχημα πραγματοποιούνται ταυτόχρονα και διανομές αλλά και συλλογές προϊόντων.

Άλλος ένας σημαντικός περιορισμός προτεραιότητας προκύπτει όταν πρέπει να τηρηθεί μια συγκεκριμένη σειρά κατά την εξυπηρέτηση ορισμένων πελατών που ανήκουν στο ίδιο υποσύνολο αλλά δεν είναι του ιδίου τύπου (αν για παράδειγμα παραλαμβάνουν ή παραδίδουν προϊόντα). Χαρακτηριστικό παράδειγμα τέτοιας περίπτωσης είναι το πρόβλημα δρομολόγησης οχημάτων με παραλαβές (VRPB) στο οποίο οι περιορισμοί έχουν να κάνουν με την παραλαβή και παράδοση των προϊόντων ελέγχοντας ωστόσο οι διανομές να γίνονται πριν από τις παραλαβές κατά την διάρκεια της διαδρομής.

Σημαντικοί περιορισμοί προκύπτουν και από το επίπεδο εξυπηρέτησης που θέλει η εκάστοτε εταιρία να παρέχει στους πελάτες της. Ο χρόνος εξυπηρέτησης των πελατών (το μέτρο εξυπηρέτησης πελατών) αποτελεί μια από τις σημαντικότερες παραμέτρους κατά την επίλυση του προβλήματος δρομολόγησης οχημάτων. Επειδή οι παραγγελίες των πελατών δεν είναι συνήθως σταθερές ούτε χρονικά, αλλά ούτε και ποσοτικά, κάθε εταιρία θα πρέπει να επιλύσει το πρόβλημα προσεγγιστικά ως προς

τις παραγγελίες για μια δεδομένη χρονική περίοδο. Οι προσεγγίσεις που επιλέγονται είναι:

- Τυπική περίπτωση: Σε αυτήν την περίπτωση η εταιρία έχει συγκεκριμένους πελάτες με δεδομένες απαιτήσεις (τυπικές) και σε δεδομένες χρονικές περιόδους. Οι πελάτες κάνουν παραγγελίες εμπορευμάτων σταθερά κάθε x μέρες (x< περιόδου που εξετάζουμε) και δέχονται επισκέψεις από την αποθήκη τόσες φορές όσες προκύπτουν από την διαίρεση της χρονικής περιόδου με το x. Οι διαδρομές που προκύπτουν από την επίλυση ουσιαστικά γνωστοποιούν το χρόνο επίσκεψης κάθε πελάτη από το κέντρο διανομής.
- Διακεκομμένου χρόνου: Κατά την επίλυση του προβλήματος δρομολόγησης οχημάτων στη συγκεκριμένη περίπτωση ακολουθείται η λογική κατά την οποία για παραγγελίες που γίνονται σε ένα ορισμένο χρονικό διάστημα x ημερών, αυτές να ικανοποιούνται στο επόμενο χρονικό διάστημα x ημερών. Ωστόσο σε αυτήν την περίπτωση δεν μπορούν να ικανοποιηθούν πελάτες των οποίων οι παραγγελίες πρέπει να διανεμηθούν την ίδια χρονική περίοδο στην οποία έγιναν και αυτές, αλλά ικανοποιούνται την επόμενη. Επομένως υπάρχουν περιπτώσεις κατά τις οποίες η επίλυση με αυτού του είδους την προσέγγιση δεν είναι εφικτή.
- Αργά μετακινούμενες προτεραιότητες πελατών: Στην συγκεκριμένη προσέγγιση θέτουμε προτεραιότητα σε κάθε πελάτη λαμβάνοντας υπόψιν το χρόνο που υπολείπεται μέχρι να δεχτεί ο εκάστοτε πελάτης την παραγγελία. Η λογική που ακολουθείται είναι πως όσο λιγότερος χρόνος απομένει, τόσο μεγαλύτερη είναι η προτεραιότητα να εξυπηρετηθεί. Σε κάθε χρονική στιγμή, λοιπόν, για την επίλυση του προβλήματος θα επιλέγεται μια πολύπλοκη αντικειμενική συνάρτηση με κόστη δρομολόγησης και προτεραιοτήτων για κάθε πελάτη ώστε να είναι εφικτή η εξυπηρέτηση των πελατών μέσα σε ένα δεδομένο αριθμό ημερών που ουσιαστικά υποδεικνύει τη μέγιστη καθυστέρηση.

1.2 Στόχοι κατά την επίλυση

Οι στόχοι προς επίτευξη κατά την επίλυση ενός προβλήματος δρομολόγησης οχημάτων είναι αρκετοί και πολλές φορές προκύπτουν και από τον συνδυασμό τους. Παρακάτω παρουσιάζονται μερικοί από τους πλέον συνήθης είναι:

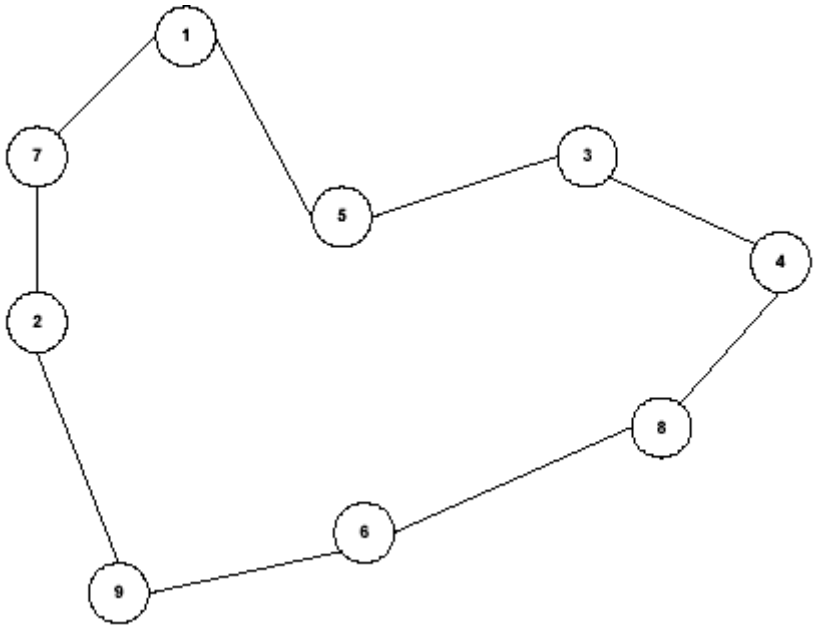
- Ελαχιστοποίηση του συνολικού κόστους μεταφοράς, το οποίο εξαρτάται από την ολική απόσταση των δρομολογίων (ή το χρόνο δρομολογίων) και τα κόστη χρήσης των οχημάτων και αμοιβής των οδηγών.
- Ελαχιστοποίηση του συνολικού αριθμού οχημάτων (και παράλληλα οδηγών) που χρειάζονται για την κάλυψη των αναγκών των πελατών.
- Ελαχιστοποίηση των ποινών-κυρώσεων που προκύπτουν από την μερική εξυπηρέτηση των πελατών.
- Εξισορρόπηση των διαδρομών των οχημάτων με βάση τον χρόνο μετάβασης και το φορτίο μεταφοράς.

Κεφάλαιο 2: Τα προβλήματα Διανομής και οι προεκτάσεις τους

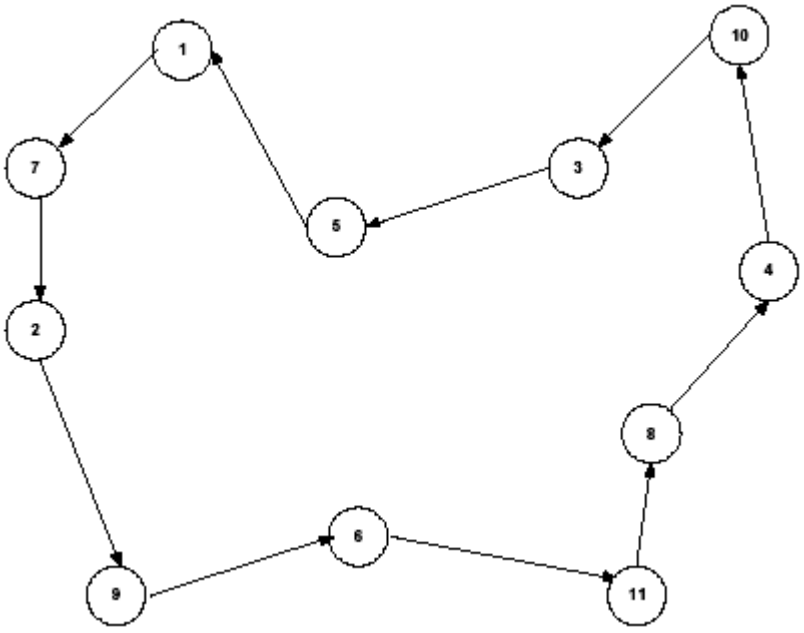
2.1 Το πρόβλημα του Περιπλανώμενου πωλητή (Traveling Salesman Problem)

Το πρόβλημα του περιπλανώμενου πωλητή αποτελεί ένα από τα σημαντικότερα προβλήματα συνδυαστικής βελτιστοποίησης. Αυτό οφείλεται κυρίως στην απλότητα του αλλά ταυτόχρονα και στη δυσκολία επίλυσης του.

Σε ένα πρόβλημα περιπλανώμενου πωλητή ένας πωλητής πρέπει να επισκεφθεί όλες τις πόλεις ξεχωριστά ξεκινώντας από και καταλήγοντας στην ίδια πόλη- αφετηρία. Το πρόβλημα αυτό απαιτεί τον καθορισμό ενός κύκλου ελαχίστου κόστους που περνά από κάθε κόμβο του συσχετιζόμενου γραφήματος ακριβώς μια φορά. Η απόσταση (κόστος) των πόλεων μεταξύ τους (για όλα τα ζεύγη των πόλεων) θεωρείται γνωστή. Πρέπει επιπλέον να είναι γνωστό αν το κόστος του ταξιδιού μεταξύ δύο τοποθεσιών εξαρτάται ή όχι από την κατεύθυνση του γραφήματος. Εάν δεν εξαρτάται έχουμε ένα συμμετρικό πρόβλημα (σχήμα 2.1) ενώ στην αντίθετη περίπτωση έχουμε ένα ασύμμετρο ή προσανατολισμένο πρόβλημα (σχήμα 2.2) περιπλανώμενου πωλητή.



Σχήμα 2.1:Συμμετρική περίπτωση προβλήματος περιπλανώμενου πωλητή (TSP)



Σχήμα 2.2: Μη συμμετρική περίπτωση προβλήματος περιπλανώμενου πωλητή (TSP)

Το συγκεκριμένο πρόβλημα έχει μορφοποιηθεί με πολλούς τρόπους. Ακολουθεί αμέσως παρακάτω μια αρκετά απλή μορφοποίηση του προβλήματος του περιπλανώμενου πωλητή:

Το TSP αποτελείται από ένα πλήρες, σταθμισμένο, μη προσανατολισμένο γράφημα G που καθορίζεται από ζεύγη (N,d), όπου N είναι το σύνολο των κόμβων- πόλεων και d η συνάρτηση απόστασης μεταξύ των κόμβων του γραφήματος.

Το γράφημα G πρέπει να ικανοποιεί τις παρακάτω σχέσεις:

- $d(i, j) = d(j, i)$ για κάθε i και j που ανήκει στο N. (1)
- $d(i, j) \geq 0$ για κάθε i και j που ανήκει στο N. (2)
- $d(i, j) + d(j, k) \geq d(i, k)$ για κάθε i, j και k που ανήκει στο N. (3)

Η συνθήκη 3 ονομάζεται τριγωνική ανισότητα. Ο αριθμός d (i, j) ονομάζεται μήκος ή βάρος του (i, j).

Κάνοντας χρήση ακέραιου προγραμματισμού μια πιο σύνθετη μορφοποίηση του προβλήματος TSP είναι η ακόλουθη:

n: ο αριθμός των πόλεων-κόμβων.

c_{ij}: το κόστος ταξιδιού από την πόλη i στην πόλη j.

$$x_{ij}^{k} = \begin{cases} 1, \text{ το όχημα κ χρησιμοποιεί το τόξο (i,j)} \\ 0, \text{ διαφορετικά} \end{cases}$$

Επειδή από κάθε πόλη i θα πρέπει το όχημα να φεύγει ακριβώς μια φορά,

$$\sum_{j=1}^{n} X_{ij} = 2, \quad \text{για ι=1,2,…..n}$$

Αντίστοιχα το όχημα θα πρέπει να πηγαίνει ακριβώς μια φορά σε κάθε πόλη,

$$\sum_{j=1}^{n} X_{1j} = 2,$$

και

$$\sum_{ij \in En} X_{ij} = 2,$$

Επιπλέον πρέπει να ισχύουν:

$X(C) \leq | C | -1$, για όλους τους κύκλους $C \in \{ 2,3,…..,n\}$

Τέλος, η αντικειμενική συνάρτηση για κάθε κύκλο είναι η ακόλουθη:

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

Για το πρόβλημα του περιπλανώμενου πωλητή υπάρχουν πάρα πολλές εφαρμογές και σε διάφορα πεδία. Έχει αναπτυχθεί ένας μεγάλος αριθμός αλγόριθμων επίλυσής του συγκεκριμένου προβλήματος τόσο ακριβείς, όσο και ευρετικοί. Οι ακριβείς αλγόριθμοι που στηρίζονται κυρίως σε μεθόδους διακλάδωσης και οριοθέτησης

(branch and bound), δίνουν τη βέλτιστη λύση ακόμα και σε προβλήματα με πάρα πολλές μεταβλητές (μεγάλης τάξης). Ωστόσο απαιτείται υπερβολικά μεγάλος χρόνος για τον υπολογισμό των λύσεων. Με την ανάπτυξη ευρετικών μεθόδων, από την άλλη μεριά, το πρόβλημα του χρόνου αντιμετωπίστηκε ενώ παράλληλα έδιναν πολύ καλές λύσεις. Παραδείγματα τέτοιων αλγορίθμων είναι ο αλγόριθμος Lin- Keninghan και οι αλγόριθμοι 2-opt και 3-opt που θα αναλυθούν διεξοδικότερα στο επόμενο κεφάλαιο.

2.2 Το πρόβλημα δρομολόγησης οχημάτων (V.R.P.)

Στο βασικό πρόβλημα δρομολόγησης οχημάτων, σε αντίθεση με το πρόβλημα του περιοδεύοντος πωλητή, έχουμε ως στόχο να δημιουργήσουμε παραπάνω από ένα δρομολόγια. Η μαθηματική διατύπωση του είναι η ακόλουθη:

Έστω ότι έχουμε n τοποθεσίες και με i= 1 συμβολίζουμε την κεντρική αποθήκη (κέντρο διανομής). Τότε με i=2,…,n συμβολίζονται οι πελάτες. Κάθε πελάτης (i) έχει μια δεδομένη ζήτηση ποσότητας προϊόντων που συμβολίζεται με q_i και το κόστος μετάβασης από τον πελάτη i στον πελάτη j συμβολίζεται με C_{ij}. Θεωρούμε ότι η εταιρία διαθέτει K οχήματα τα οποία εκτελούν τις διαδρομές με χωρητικότητα Q_K.

Παράλληλα σε κάθε όχημα θα αντιστοιχεί μια συγκεκριμένη διαδρομή που θα ξεκινά και θα καταλήγει στην κεντρική αποθήκη. Η απλούστερη μορφή με την οποία μπορεί να αποδώσει κανείς το πρόβλημα Δρομολόγησης Οχημάτων γίνεται βάσει των παρακάτω προϋποθέσεων:

- Ο στόλος των οχημάτων που εκτελούν τις διαδρομές αποτελείται από όμοια οχήματα (όμοια χωρητικότητα).
- Δεν υπάρχουν περιορισμοί με χρονικά διαστήματα(Time Windows).

Επομένως το πρόβλημα μπορεί να διατυπωθεί μαθηματικά ως εξής:
Έστω ότι i=1 είναι η κεντρική αποθήκη ή το κέντρο διανομής. Τότε i=2,3,4....,n θα είναι οι πελάτες μας. Θεωρούμε ότι κάθε πελάτης i έχει ζήτηση q_i ποσότητα προϊόντων και το κόστος μετάβασης από τον πελάτη i στον j ορίζετε ως c_{ij}. Εάν η

εταιρία διαθέτει K οχήματα που εκτελούν τις μεταφορές, η χωρητικότητα κάθε οχήματος θα είναι Q_K. Επιπλέον, γίνεται η υπόθεση πως όλοι οι πελάτες και τα οχήματα θα εξυπηρετούν με φθίνουσα σειρά τα q_i και Q_K. Τέλος, σε κάθε όχημα θα αντιστοιχεί μια διαδρομή η οποία θα ξεκινάει και θα καταλήγει στο κέντρο διανομής.

Η αντικειμενική συνάρτηση που έχουμε προς ελαχιστοποίηση θα είναι:

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

Υπό τους περιορισμούς:

$\sum d_i x_{ij}^v \le K$, για κάθε ν= 1,2,…., K

$X= [x_{ij}^v] \in S^*$

$$x_{ij}^v = \begin{cases} 1, \text{το όχημα ν χρησιμοποιεί το τόξο (i,j)} \\ 0, \text{διαφορετικά} \end{cases}$$

με c_{ij} συμβολίζεται η απόσταση που διανύει ένα όχημα (το κόστος της διαδρομής) προκειμένου να μεταβεί από τον πελάτη i στον πελάτη j. Επίσης τα d_i, Q, K και S^* συμβολίζουν τα εξής:

d_i = ζήτηση του πελάτη i.

Q = Χωρητικότητα οχήματος.

K = Το σύνολο οχημάτων της εταιρίας.

S^* = Το σύνολο όλων των Μ λύσεων του προβλήματος του περιπλανώμενου πωλητή.

Το πρόβλημα Δρομολόγησης Οχημάτων ανήκει στα σημαντικότερα προβλήματα που πραγματεύεται η Συνδυαστική Βελτιστοποίηση. Έχει ένα μεγάλο εύρος προεκτάσεων που οφείλεται στις πάρα πολλές εφαρμογές του σε πραγματικά προβλήματα. Για κάθε πραγματικό πρόβλημα Δρομολόγησης Οχημάτων τίθενται κάθε φορά διαφορετικοί περιορισμοί τέτοιοι ώστε οι λύσεις που θα προκύπτουν να ανταποκρίνονται στις επιθυμητές κατασκευές των διαδρομών. Παραδείγματα προεκτάσεων του προβλήματος είναι η εξυπηρέτηση να περιλαμβάνει εκτός από διανομές και παραλαβές, να διακινούνται παραπάνω από ένα προϊόντα, κάθε είδος προϊόντος να μην ξεπερνά μια δεδομένη χωρητικότητα, η εξυπηρέτηση κάθε πελάτη να πρέπει να γίνεται σε συγκεκριμένες χρονικές περιόδους, τα οχήματα να έχουν διαφορετική χωρητικότητα, οι απαιτήσεις των πελατών να μην είναι προκαθορισμένες, να υπάρχουν πιθανές σχέσεις προαπαίτησης εξυπηρέτησης μεταξύ των πελατών και πολλές ακόμη προεκτάσεις τόσο ως προς τις απαιτήσεις των πελατών, όσο και ως προς τις χρονικές απαιτήσεις .

Μερικές από τις σημαντικότερες προεκτάσεις του προβλήματος Δρομολόγησης Οχημάτων που θα αναλυθούν διεξοδικά είναι:

- Το πρόβλημα Δρομολόγησης Οχημάτων Περιορισμένης Χωρητικότητας (Capacitated Vehicle Routing problem).
- Το πρόβλημα Δρομολόγησης Οχημάτων με Παραλαβές (Vehicle Routing Problem with Backhauls).
- Το πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Διαστήματα (Vehicle Routing Problem with Time Windows).
- Το πρόβλημα Δρομολόγησης Οχημάτων με Παραλαβές και Διανομές (Vehicle Routing Problem with Pickup and Delivery).
- Το πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Διαστήματα και Παραλαβές και Διανομές (Vehicle Routing Problem with Time Windows and Pickup and Delivery).

2.3 Πρόβλημα Δρομολόγησης Οχημάτων με Περιορισμένη Χωρητικότητα

Στο πρόβλημα Δρομολόγησης Οχημάτων με Περιορισμένη χωρητικότητα έχουμε ένα σύνολο πελατών που η ζήτηση τους είναι προκαθορισμένη. Τα οχήματα είναι όμοια μεταξύ τους (ίδιας χωρητικότητας), εκκινούν από την κεντρική αποθήκη και εκτελούν το δρομολόγιο τους με μοναδικούς περιορισμούς, τους περιορισμούς χωρητικότητας. Στόχος της επίλυσης του συγκεκριμένου προβλήματος είναι η ελαχιστοποίηση του συνολικού κόστους εξυπηρετώντας τους πελάτες. Η περιγραφή του προβλήματος με βάση τη θεωρία των γραφημάτων έχει την εξής μορφή:

Έστω ότι έχουμε ένα πλήρες γράφημα G = (V, A), όπου V = {1,….,n} το σύνολο των κόμβων με τον κόμβο 1 να αντιστοιχεί στην κεντρική αποθήκη και τους υπόλοιπους να αντιστοιχούν στους πελάτες. Το σύμβολο c_{ij} δηλώνει το μη αρνητικό κόστος διαδρομής από τον κόμβο i στον κόμβο j. Αν τα τόξα του γραφήματος είναι μη προσανατολισμένα τότε το πρόβλημα μας είναι συμμετρικό ενώ αν είναι προσανατολισμένα, τότε το πρόβλημα μας είναι μη συμμετρικό.

Σε διάφορες πρακτικές περιπτώσεις ο πίνακας κόστους ικανοποιεί τη σχέση:

$c_{in} + c_{nj} \geq c_{ij}$ για κάθε i, j, n ∈ V.

Η παραπάνω τριγωνική ανισότητα ουσιαστικά υποδηλώνει ότι για ένα όχημα η καλύτερη επιλογή είναι να πάει κατευθείαν από τον κόμβο i στον κόμβο j, χωρίς να μεταβεί πρώτα στον n.

Το Πρόβλημα Δρομολόγησης Οχημάτων με περιορισμένη χωρητικότητα έχει ως στόχο την εύρεση K κύκλων με το ελάχιστο δυνατό κόστος (είναι ίσο με το άθροισμα όλων των κοστών των κόμβων που ανήκουν σε κάθε κύκλο) και με τα εξής χαρακτηριστικά:

- Κάθε κύκλος έχει ως σημείο αφετηρίας και τερματισμού την αποθήκη.
- Η ζήτηση κάθε πελάτη ικανοποιείται από ένα μόνο όχημα.

- Το άθροισμα της ζήτησης των κόμβών δεν πρέπει να ξεπερνά την χωρητικότητα του οχήματος.

2.4 Πρόβλημα Δρομολόγησης Οχημάτων με Παραλαβές (VRPB)

Στο συγκεκριμένο πρόβλημα Δρομολόγησης Οχημάτων έχουμε ένα σύνολο πελατών V που χωρίζεται σε δύο υποσύνολα. Θεωρούμε ότι το πρώτο αποτελείται από n πελάτες (υποσύνολο L),καθένας από τους οποίους απαιτεί να του παραδοθεί μια συγκεκριμένη ποσότητα προϊόντων από το Κέντρο Διανομής. Ονομάζουμε αυτούς του πελάτες, πελάτες παράδοσης (Line haul customers) ή προμηθευόμενους. Έστω τώρα ότι το δεύτερο υποσύνολο πελατών αποτελείται από m πελάτες από τους οποίους πρέπει να παραληφθεί μια συγκεκριμένη ποσότητα προϊόντων. Τους πελάτες αυτούς τους ονομάζουμε πελάτες παραλαβής (Backhaul Customers) ή προμηθευτές. Το σύνολο των πελατών είναι αριθμημένο έτσι ώστε L={1,.....,n} και B={n+1,....,n+m}.

Το πρόβλημα Δρομολόγησης Οχημάτων με παραλαβές έχει ως στόχο την εύρεση K κύκλων με το ελάχιστο δυνατό κόστος (είναι ίσο με το άθροισμα των αποστάσεων μετάβασης των κόμβων που ανήκουν σε κάθε κύκλο) και με τα εξής χαρακτηριστικά:

- Κάθε όχημα πραγματοποιεί μια διαδρομή και κάθε διαδρομή επισκέπτεται το κέντρο διανομής (αποθήκη).
- Σε κάθε διαδρομή το συνολικό φορτίο ελέγχει και τους προμηθευόμενους και τους προμηθευτές ώστε να μην ξεπερνά την χωρητικότητα του οχήματος. Δηλαδή αν q_i είναι η ζήτηση τότε $Q_k \max\{q(L), q(B)\}$.
- Σε κάθε διαδρομή ορίζουμε έναν περιορισμό προτεραιότητας σύμφωνα με τον οποίο, κάθε φορά που σε μια διαδρομή εξυπηρετούνται και πελάτες παράδοσης και πελάτες παραλαβής, οι πελάτες του πρώτου υποσυνόλου πρέπει να εξυπηρετηθούν πριν από αυτούς του δεύτερου.

2.5 Πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα (VRPTW)

Στο πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα έχουμε ένα σύνολο πελατών που πρέπει να εξυπηρετηθούν από ένα σύνολο οχημάτων που εκκινούν και πάλι από την κεντρική αποθήκη της εταιρίας. Κάθε πελάτης έχει ένα φορτίο που πρέπει να περισυλλεχθεί μόνο που στο δεδομένο πρόβλημα ο ίδιος καθορίζει την χρονική περίοδο (χρονικό παράθυρο ή time window) στην οποία πρέπει να πραγματοποιηθεί η φόρτωση. Οι πελάτες εξυπηρετούνται από οχήματα δεδομένης χωρητικότητας και σκοπός της επίλυσης είναι να βρεθεί ένα σύνολο διαδρομών όπου κάθε όχημα εκκινεί και τελειώνει τη διαδρομή του στην κεντρική αποθήκη, η εξυπηρέτηση των πελατών γίνεται χωρίς να παραβιαστούν η χωρητικότητα και οι περιορισμοί που αφορούν τα χρονικά παράθυρα, ενώ παράλληλα ελαχιστοποιείται το συνολικό μήκος των διαδρομών.

Το πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα αποτελεί ουσιαστικά μια επέκταση του προβλήματος Δρομολόγησης Οχημάτων Περιορισμένης χωρητικότητας. Οι περιορισμοί χωρητικότητας εξακολουθούν να ισχύουν όμως στην προκείμενη περίπτωση ισχύει και ο επιπλέον περιορισμός σύμφωνα με τον οποίο κάθε πελάτης πρέπει να εξυπηρετηθεί μέσα σε μια χρονική περίοδο (α_i ,β_i). Τα επιπλέον δεδομένα που δίνονται για την επίλυση του προβλήματος είναι:

1. Η χρονική στιγμή που εκκινούν τα οχήματα από την κεντρική αποθήκη.
2. Ο χρόνος ταξιδιού από την πόλη i στην πόλη j.
3. Ο χρόνος εξυπηρέτησης για κάθε πελάτη s_i.
4. Τα χρονικά παράθυρα εξυπηρέτησης για κάθε πελάτη.

Η εξυπηρέτηση ενός πελάτη πρέπει να γίνει μέσα στο χρονικό παράθυρο που μπορεί ο πελάτης να εξυπηρετηθεί και το όχημα πρέπει να παραμείνει στον πελάτη για ένα χρονικό διάστημα ίσο με τον χρόνο εξυπηρέτησης του. Σε περίπτωση που το όχημα φτάσει σε έναν πελάτη νωρίτερα από τον προκαθορισμένο χρόνο, παραμένει στην τοποθεσία μέχρι να ξεκινήσει το χρονικό παράθυρο.

Συνήθως οι πίνακες κόστους και χρόνου μετάβασης συμπίπτουν και τα Χρονικά Διαστήματα καθορίζονται με βάση την υπόθεση ότι όλα τα οχήματα εκκινούν από την κεντρική αποθήκη την χρονική στιγμή 0. Επιπρόσθετα, οι απαιτήσεις των Χρονικών Παραθύρων απαιτούν πλήρη προσανατολισμό για κάθε διαδρομή ακόμα κι όταν πρόκειται για προβλήματα με τους αρχικούς πίνακες συμμετρικούς, οπότε στο πλήθος των περιπτώσεων το πρόβλημα μοντελοποιείται ως μη συμμετρικό πρόβλημα. Υπάρχουν δύο ειδών χρονικά παράθυρα. Τα χαλαρά, όπου αν ένα όχημα φτάσει σε κάποιον πελάτη σε χρόνο που δεν ανήκει στο χρονικό παράθυρο που του αντιστοιχεί, μπορεί να ξεκινήσει την εξυπηρέτηση του κανονικά εκείνη τη χρονική στιγμή. Στην περίπτωση των σκληρών Χρονικών Διαστημάτων, όμως, δεν επιτρέπεται η εξυπηρέτηση του πελάτη αν το όχημα φτάσει σε αυτόν αργότερα από τον αργότερο χρόνο εξυπηρέτησης. Αν όμως το όχημα φτάσει νωρίτερα από τον νωρίτερο χρόνο εξυπηρέτησης του πελάτη τότε μπορεί να περιμένει μέχρι ο χρόνος ταξιδιού να γίνει ίσος με τον νωρίτερο χρόνο εξυπηρέτησης του συγκεκριμένου πελάτη.

Η επίλυση του προβλήματος Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα συνίσταται στην εύρεση ακριβώς Κ διαδρομών με ελάχιστο κόστος τέτοιους ώστε:

- Η ζήτηση κάθε πελάτη ικανοποιείται από ένα μόνο όχημα.
- Κάθε κύκλος εκκινεί και καταλήγει στην κεντρική αποθήκη.
- Το άθροισμα της ζήτησης των πελατών κάθε κύκλου δεν υπερβαίνει τη χωρητικότητα του οχήματος.
- Σε κάθε πελάτη η εξυπηρέτηση εκκινεί μέσα στη χρονική περίοδο (α_i ,β_i) και το όχημα παραμένει εκεί για χρόνο που ισοδυναμεί με τον χρόνο εξυπηρέτησης του πελάτη.

Το πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα αποδίδεται σε ένα γράφημα G= (V, A) στο οποίο η αποθήκη συμβολίζεται από τους κόμβους 1 και n+1. Επομένως σε κάθε κύκλο ο αρχικός κόμβος είναι ο 1 ενώ ο τελικός είναι ο n+1. Τα χρονικά παράθυρα που αντιστοιχούν στους δύο αυτού κόμβους είναι μηδενισμένα όπως επίσης μηδενισμένη είναι και η ζήτηση καθώς και οι χρόνοι εξυπηρέτησης τους, δηλαδή $d_1 = d_{n+1} = s_1 = s_{n+1} = 0$.

Το πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα μπορεί να διατυπωθεί μαθηματικά ως εξής:

Περιορισμοί:

$$x_{ij}^{v} = \begin{cases} 1, \text{ το όχημα } \nu \text{ χρησιμοποιεί το τόξο } (i,j) \\ 0, \text{ διαφορετικά} \end{cases}$$

$$\sum_{\kappa \in E} \sum_{j \in V} x_{ijv} = 1 \qquad \text{για κάθε } i \in N \qquad (1)$$

$$\sum_{i \in V-\{0\}} x_{0jv} = 1 \qquad \text{για κάθε } j \in N, \nu \in K. \qquad (2)$$

$$\sum_{i \in V-\{j\}} x_{ijv} - \sum_{i \in V-\{j\}} x_{jiv} = 0 \qquad \text{για κάθε } j \in N, \nu \in K. \qquad (3)$$

$$\sum_{i \in V-\{n+1\}} x_{in+1v} = 1 \qquad \text{για κάθε } \nu \in K. \qquad (4)$$

$$x_{ijv} (w_{iv} + s_i + t_{ij} + w_{jv}) \leq 0 \qquad \text{για κάθε } \nu \in K, (i, j) \in A. \qquad (5)$$

$$\alpha_i \sum_{j \in V} x_{ijv} \leq w_{iv} \leq \beta_i \sum_{j \in V} x_{ijv} \qquad \text{για κάθε } i \in N, \nu \in K. \qquad (6)$$

$$E \leq w_{iv} \leq L \qquad \text{για κάθε } i \in (0, n+1), \nu \in K. \qquad (7)$$

$$\sum_{i \in N} d_i \sum_{j \in V} x_{ijv} \leq C \qquad \text{για κάθε } \nu \in K. \qquad (8)$$

$x_{ijv} \geq 0$ για κάθε ν ∈ K, (i, j) ∈ A.

Ενώ η αντικειμενική συνάρτηση είναι της μορφής:

$$C^* = \min \sum_{i,j} c_{ij} \sum_{v} x_{ijv}$$

Στην παραπάνω διατύπωση χρησιμοποιήθηκαν δύο είδη μεταβλητών, η x_{ijv} για τις μεταβλητές ροής και η w_{iv} ως χρονική μεταβλητή που καθορίζει πότε θα εξυπηρετηθεί ένα πελάτης i από ένα όχημα ν.

Η αντικειμενική συνάρτηση εκφράζει το συνολικό κόστος. Με τους περιορισμούς (2)- (4) χαρακτηρίζουμε τη ροή της διαδρομής που κάνει το όχημα ν. Οι περιορισμοί (5), (7) και (8) ελέγχουν αν είναι εφικτή η τοποθέτηση ενός πελάτη σε ένα κύκλο με βάση τα χρονικά διαστήματα και τη χωρητικότητα του οχήματος. Με τη χρήση του περιορισμού (6) θέτουμε το w_{iv} ίσο με μηδέν αν ένα όχημα δεν επισκέπτεται τους πελάτες i και j στη συγκεκριμένη διαδρομή.

2.6 Πρόβλημα Δρομολόγησης Οχημάτων με Παραλαβές και Διανομές (VRPPD)

Στο πρόβλημα Δρομολόγησης Οχημάτων με Παραλαβές και Διανομές έχουμε ένα σύνολο από πελάτες που πρέπει να εξυπηρετηθούν από έναν δεδομένο αριθμό οχημάτων που εκκινούν και πάλι από την κεντρική αποθήκη της εταιρίας. Στο συγκεκριμένο πρόβλημα όμως κάθε πελάτης σχετίζεται με δύο ποσότητες d_i και p_i που υποδηλώνουν τον αριθμό των προϊόντων που πρέπει να διανεμηθούν και να παραληφθούν από τον συγκεκριμένο πελάτη. Σε αρκετές περιπτώσεις χρησιμοποιείται μόνο μια ποσότητα για κάθε πελάτη $d_i = d_i - p_i$, που υπολογίζει τη διαφορά μεταξύ των ζητήσεων διανομής και παραλαβής (λαμβάνει και αρνητικές τιμές). Για κάθε πελάτη καθορίζονται δύο κόμβοι, ο O_i (κόμβος από τον οποίο εκκινούν τα προϊόντα που θα διανεμηθούν) και ο D_i (κόμβος στον οποίο καταλήγουν τα προϊόντα που συλλέγονται). Σε αρκετές περιπτώσεις οι δύο αυτοί κόμβοι αναφέρονται στην ίδια κορυφή, την κεντρική αποθήκη δηλαδή.

Στο παρών πρόβλημα, παράλληλα, πρέπει να ισχύει πάντα ο περιορισμός πως σε κάθε πελάτη η διανομή των προϊόντων πρέπει να γίνεται πάντα πριν από την παραλαβή, ώστε η συνολική φόρτωση του οχήματος τη χρονική στιγμή που φτάνει το όχημα σε έναν καινούριο πελάτη να είναι ίση με την αρχική φόρτωση από την αποθήκη μείον το άθροισμα των προϊόντων που διένειμε στους προηγούμενους πελάτες συν το άθροισμα των προϊόντων που ικανοποιήθηκαν με παραλαβές από τους προηγούμενους πελάτες.

Η επίλυση του προβλήματος Δρομολόγησης Οχημάτων με Παραλαβές και Διανομές συνίσταται στην εύρεση ακριβώς Κ διαδρομών με ελάχιστο κόστος τέτοιους ώστε:

- Κάθε κύκλος εκκινεί και καταλήγει στην κεντρική αποθήκη.
- Η ζήτηση κάθε πελάτη ικανοποιείται από ένα μόνο όχημα.
- Το φορτίο του οχήματος κατά τη διάρκεια ενός κύκλου δεν μπορεί να πάρει αρνητικές τιμές αλλά ούτε και να υπερβεί την χωρητικότητα του οχήματος.
- Για κάθε πελάτη i ο κόμβος O_i πρέπει να εξυπηρετηθεί από την ίδια διαδρομή και πριν από τον κόμβο D_i ,αρκεί να είναι διαφορετικός από την κεντρική αποθήκη.
- Για κάθε πελάτη i ο κόμβος D_i πρέπει να εξυπηρετηθεί από την ίδια διαδρομή και μετά από τον κόμβο O_i ,αρκεί να είναι διαφορετικός από την κεντρική αποθήκη.

2.7 Πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και με Παραλαβές και Διανομές (VRPTWPD)

Το πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και Παραλαβές και διανομές είναι η σύνθεση των δύο προβλημάτων που παρουσιάστηκαν στις δύο προηγούμενες παραγράφους, του προβλήματος Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και του προβλήματος Δρομολόγησης Οχημάτων με Παραλαβές και Διανομές. Στο συγκεκριμένο πρόβλημα έχουμε ένα σύνολο από πελάτες N={2,…,n} που πρέπει να εξυπηρετηθούν από έναν δεδομένο αριθμό οχημάτων Κ που εκκινούν και πάλι από την κεντρική αποθήκη της εταιρίας. Κάθε πελάτης χαρακτηρίζεται από την γεωγραφική του θέση, τον αριθμό προϊόντων παραλαβής και

διανομής d_i και p_i καθώς και από τα χρονικά παράθυρα (α_i, β_i) στα οποία πρέπει να εξυπηρετηθεί. Ένα όχημα μπορεί να φτάσει σε έναν πελάτη νωρίτερα από τον νωρίτερο χρόνο εξυπηρέτησης του και να παραμείνει εκεί χωρίς επιπλέον κόστος μέχρι να τον εξυπηρετήσει. Ωστόσο σε καμία περίπτωση δεν επιτρέπεται η εξυπηρέτηση κανενός πελάτη μετά το πέρας του βραδύτερου χρόνου εξυπηρέτησης του.

Το πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και με Παραλαβές και Διανομές αποδίδεται σε ένα γράφημα G= (V, A) στο οποίο η αποθήκη συμβολίζεται από τους κόμβους 1 και n+1. Επομένως σε κάθε κύκλο ο αρχικός κόμβος είναι ο 1 ενώ ο τελικός είναι ο n+1. Τα χρονικά παράθυρα που αντιστοιχούν στους δύο αυτού κόμβους είναι μηδενισμένα όπως επίσης μηδενισμένη είναι και η ζήτηση καθώς και οι χρόνοι εξυπηρέτησης τους, δηλαδή $d_1 = d_{n+1} = s_1 = s_{n+1} = 0$.

Με τους συμβολισμούς c_{ij} και t_{ij} δηλώνουμε το κόστος και χρόνο διαδρομής για να μετακινηθεί ένα όχημα από τον κόμβο i στον κόμβο j.

Σκοπός του προβλήματος είναι η εξυπηρέτηση των πελατών με ελαχιστοποίηση του κόστους και έχει τα εξής χαρακτηριστικά:

- Κάθε κύκλος εκκινεί και καταλήγει στην κεντρική αποθήκη.
- Κάθε πελάτης δέχεται επίσκεψη σε ένα μόνο κύκλο
- Το φορτίο του οχήματος κατά τη διάρκεια ενός κύκλου δεν μπορεί να πάρει αρνητικές τιμές αλλά ούτε και να υπερβεί την χωρητικότητα του οχήματος.
- Σε κάθε πελάτη η εξυπηρέτηση εκκινεί μέσα στη χρονική περίοδο (α_i, β_i) και το όχημα παραμένει εκεί για χρόνο που ισοδυναμεί με τον χρόνο εξυπηρέτησης του πελάτη.
- Για κάθε πελάτη i ο κόμβος O_i πρέπει να εξυπηρετηθεί από την ίδια διαδρομή και πριν από τον κόμβο D_i ,αρκεί να είναι διαφορετικός από την κεντρική αποθήκη.
- Για κάθε πελάτη i ο κόμβος D_i πρέπει να εξυπηρετηθεί από την ίδια διαδρομή και μετά από τον κόμβο O_i ,αρκεί να είναι διαφορετικός από την κεντρική αποθήκη.

Το πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και Παραλαβές και Διανομές μπορεί να διατυπωθεί μαθηματικά ως εξής:

Περιορισμοί:

$$x_{ij}^{v} = \begin{cases} 1, \text{ το όχημα } \nu \text{ χρησιμοποιεί το τόξο (i,j)} \\ 0, \text{ διαφορετικά} \end{cases}$$

$$\sum_{v \in E} \sum_{j \in V} x_{ijv} = 1 \qquad \text{για κάθε } i \in N \qquad (1)$$

$$\sum_{i \in V} x_{ip}^{v} = \sum_{j \in V} x_{pj}^{v} \qquad \text{για κάθε } p \in N, \nu \in K \qquad (2)$$

$$\sum_{j \in v} x_{1j}^{v} \leq 1 \qquad \text{για κάθε } \nu \in K \qquad (3)$$

$$\sum_{i \in N} x_{i,n+1}^{v} = \sum_{j \in N} x_{1,j}^{v} \qquad \text{για κάθε } \nu \in K \qquad (4)$$

$$D_i^v + P_i^v \leq Q \qquad \text{για κάθε } i \in V, \nu \in K \qquad (5)$$

$$D_{n+1}^v = 0 \qquad \text{για κάθε } \nu \in K \qquad (6)$$

$$D_1^k = \sum_{i \in N} \sum_{j \in N} x_{ij}^k d_i \qquad \text{για κάθε } \nu \in K \qquad (7)$$

$$P_{n+1}^v = \sum_{i \in N} \sum_{j \in N} x_{ij}^v p_i \qquad \text{για κάθε } \nu \in K \qquad (8)$$

$$P_1^v = 0 \qquad \text{για κάθε } \nu \in K \qquad (9)$$

$$x_{ij}^v (P_i^v + p_i - P_j^v) = 0 \qquad \text{για κάθε } i, j \in V, \nu \in K \qquad (10)$$

$x_{ij}^{v} (D_{i}^{v} - d_{i} - D_{j}^{v}) = 0$ για κάθε i, j ∈ V, ν ∈ K (11)

$x_{ij}^{v} (T_{i}^{v} + t_{i} - T_{j}^{v}) = 0$ για κάθε i, j ∈ V, ν ∈ K (12)

$a_{i} \leq T_{i} \leq b_{i}$ για κάθε i ∈ V, ν ∈ K (13)

$D_{i}^{v} \geq 0$ για κάθε i ∈ V, ν ∈ K (14)

$P_{i}^{v} \geq 0$ για κάθε i ∈ V, ν ∈ K (15)

Ενώ η αντικειμενική συνάρτηση είναι της μορφής:

$$C^{*} = \min \sum_{i,j} c_{ij} \sum_{v} x_{ijv}$$

όπου

D_{i}^{v}: ο αριθμός των διανομών που έχουν γίνει από το όχημα ν μόλις φεύγει από τον πελάτη i.

P_{i}^{v}: ο αριθμός των παραλαβών που έχουν γίνει από το όχημα ν μόλις φεύγει από τον πελάτη i.

T_{i}^{v}: ο χρόνος στον οποίο το όχημα ν φτάνει στον κόμβο i.

Στόχος της αντικειμενικής συνάρτησης είναι η ελαχιστοποίηση του συνολικού κόστους της εκάστοτε διαδρομής. Ο περιορισμός (1) ελέγχει το ότι κάθε πελάτης εξυπηρετείται από ένα μόνο όχημα ενώ ο (2) δεσμεύει πως το όχημα που πηγαίνει σε ένα κόμβο είναι το ίδιο με αυτό που εξέρχεται από τον ίδιο κόμβο. Οι περιορισμοί (3), (4) δεσμεύουν κάθε όχημα να χρησιμοποιείται μόνο μια φορά. Με τον περιορισμό (5) ελέγχουμε αν το φορτίο του οχήματος, όταν εξέρχεται από κάθε κόμβο, είναι μικρότερο της χωρητικότητας του φορτηγού. Οι περιορισμοί (7) και (9) δεσμεύουν κάθε όχημα να εκκινεί από την κεντρική αποθήκη γεμάτο με προϊόντα προς διάθεση ενώ το φορτίο παραλαβής είναι ίσο με μηδέν. Οι περιορισμοί (6) και (8) εγγυώνται πως όταν τα οχήματα επιστρέφουν στην κεντρική αποθήκη, έχουν

διανείμει και αντίστοιχα παραλάβει όλα τα προϊόντα από τους κόμβους που διήλθαν. Οι περιορισμοί- ισότητες (10) και (11)δηλώνουν πως αν ένα όχημα πάει σε έναν κόμβο, τότε η ποσότητα διανομής μειώνεται κατά ποσό ίσο με το ποσό που διένειμε στον συγκεκριμένο κόμβο και η ποσότητα παραλαβής αυξάνεται αντίστοιχα. Ο περιορισμός (12) ελέγχει το χρόνο που εξυπηρετεί ένα όχημα κάποιον κόμβο ώστε ο χρόνος εξυπηρέτησης να είναι τουλάχιστον ίσος με τον "ενωρίτερο" χρόνο του κόμβου. Ο περιορισμός (13), τέλος, δηλώνει τα χρονικά διαστήματα για κάθε κόμβο.

Κεφάλαιο 3: Ευρετικοί αλγόριθμοι επίλυσης των προβλημάτων Διανομής

3.1 Εισαγωγή

Για την επίλυση του προβλήματος Δρομολόγησης Οχημάτων αλλά και των προεκτάσεων του έχουν προταθεί μέχρι σήμερα πάρα πολλές μέθοδοι. Οι μέθοδοι αυτοί διαχωρίζονται σε δύο μεγάλες κατηγορίες αλγορίθμων οι οποίες είναι:

1. Ευρετικοί αλγόριθμοι (heuristics) που αναπτύχθηκαν κυρίως την περίοδο μεταξύ 1960 έως 1990.
2. Μεθευρετικοί αλγόριθμοι (metaheuristics) που αναπτύχθηκαν την τελευταία δεκαετία.

Οι κλασσικοί ευρετικοί αλγόριθμοι χρησιμοποιούνται περισσότερο από τους μεθευρετικούς αλγορίθμους. Παρουσιάζουν μια περιορισμένη σχετικά εξερεύνηση του χώρου όπου αναζητούνται οι λύσεις και τα αποτελέσματα που προκύπτουν είναι αρκετά καλά και χωρίς μεγάλο χρόνο υπολογισμού. Υπάρχει όμως και η δυνατότητα σε αρκετές από αυτές τις μεθόδους να επεκταθούν με τέτοιο τρόπο που να είναι δυνατός ο συνυπολογισμός διαφόρων περιορισμών που εμφανίζονται σε διάφορες προεκτάσεις του προβλήματος Δρομολόγησης Οχημάτων. Οι κυριότερες τεχνικές που χρησιμοποιούνται είναι:

- Η σύμπτυξη (merge) των διαδρομών που ήδη υπάρχουν με τη χρήση ενός κριτηρίου εξοικονόμησης.
- Ο βαθμιαίος προσδιορισμός κορυφών σε κάθε διαδρομή με τη χρήση ενός κριτηρίου παρεμβολής

Οι μεθευρετικές μέθοδοι, από την άλλη, δίνουν έμφαση στην παρουσίαση μιας εξερεύνησης σε μεγάλο βάθος των περιοχών όπου βρίσκεται η λύση και οι οποίες συγκεντρώνουν τη μεγαλύτερη πιθανότητα εντοπισμού της βέλτιστης λύσης. Οι μέθοδοι αυτοί χρησιμοποιούν κανόνες περιορισμένης αναζήτησης στην γειτονιά των λύσεων και επανασυνδυασμό των λύσεων. Τα αποτελέσματα που εξάγονται από τη

χρήση μεθευρητικών αλγορίθμων είναι συνήθως καλύτερα ποιοτικά συγκρινόμενα με τα αποτελέσματα των κλασσικών ευρετικών μεθόδων. Ωστόσο ο χρόνος που απαιτείται για τον υπολογισμό της λύσης είναι πάρα πολύ μεγάλος. Στις δύο επόμενες παραγράφους περιγράφονται ορισμένοι από τους πιο σημαντικούς αλγορίθμους των δύο ειδών που προαναφέρθηκαν.

3.2 Ευρετικοί αλγόριθμοι

Οι ευρετικοί μέθοδοι χωρίζονται σε:

- Κατασκευαστικές μεθόδους.
- Μεθόδους δύο φάσεων.

Χαρακτηριστικό παράδειγμα κατασκευαστικής μεθόδου είναι ο αλγόριθμος των εξοικονομήσεων των Clarke & Wright (The savings algorithm). Εφαρμόζεται σε προβλήματα όπου ο αριθμός των οχημάτων θεωρείται μεταβλητή απόφασης και λειτουργεί σωστά τόσο για προσανατολισμένα, όσο και για μη προσανατολισμένα προβλήματα. Εμφανίζεται με δύο εκδοχές, την παράλληλη και την ακολουθητική και λειτουργεί ως εξής:

Βήμα 1: Υπολογισμός των εξοικονομήσεων
Βήμα 2: Κατάταξη των εξοικονομήσεων σε φθίνουσα σειρά

<u>Παράλληλη εκδοχή</u>

Βήμα 3: Αν μετά τη δημιουργία ενός δεσμού δημιουργηθεί μια διαδρομή που να ικανοποιεί όλους τους περιορισμούς του προβλήματος Δρομολόγησης Οχημάτων, τότε δεχόμαστε τον δεσμό στη λύση ενώ σε αντίθετη περίπτωση τον απορρίπτουμε.
Βήμα 4: Ελέγχουμε τον επόμενο δεσμό στη λίστα και επαναλαμβάνουμε το προηγούμενο βήμα μέχρι να μην ξαναεπιλεγεί άλλος δεσμός.

Ακολουθητική εκδοχή

Βήμα 3: Βρίσκουμε τον πρώτο εφικτό δεσμό στη λίστα που θα μπορούσε να χρησιμοποιηθεί για την προέκταση ενός εκ των δύο άκρων της συγκεκριμένης διαδρομής.
Βήμα 4: Αν η διαδρομή δεν μπορεί να επεκταθεί άλλο κλείνουμε τη διαδρομή και βρίσκουμε τον πρώτο εφικτό δεσμό στη λίστα για να ξεκινήσουμε μια νέα διαδρομή.
Βήμα 5: Επανάληψη των δύο προηγούμενων βημάτων μέχρι να μην επαναεπιλεγεί άλλος δεσμός.

Όποια εκδοχή και να επιλεγεί, θα πρέπει κάθε φορά να ελέγχεται η εφικτότητα της μερικής λύσης για να μην υπάρξει παραβίαση των περιορισμών. Επιπλέον παρατηρείται πως όταν στην αρχική λύση οι πελάτες είναι σε διαφορετική διαδρομή, η λύση δεν είναι εφικτή. Υπάρχει όμως περίπτωση στο τέλος της διαδικασίας μερικοί πελάτες να μείνουν εκτός διαδρομών.

Στις μεθόδους των δύο φάσεων, από την άλλη πλευρά, στην πρώτη φάση γίνεται ομαδοποίηση των πελατών τοποθετώντας τους σε οχήματα- διαδρομές, και στη δεύτερη φάση γίνεται η δρομολόγηση των συγκεκριμένων διαδρομών. Χαρακτηριστικό παράδειγμα της μεθόδου δύο φάσεων αποτελούν οι αλγόριθμοι των δύο φάσεων των Fisher and Jaikumar και των Christofides, Mignozzi and Toth.

Μέθοδος των δύο φάσεων των Fisher and Jaikumar.

Η συγκεκριμένη ευρετική διαδικασία αποτελείται από δύο φάσεις. Στην πρώτη εφαρμόζεται παράλληλη ομαδοποίηση λύνοντας βέλτιστα ένα γενικευμένο πρόβλημα εκχώρησης. Στη φάση αυτή τα βήματα που υλοποιούνται είναι τρία. Αναλυτικά:

Φάση 1

Βήμα 1: Επιλέγουμε έναν αριθμό πελατών, έστω m, που είναι οι αρχικοί πελάτες των ομάδων και τοποθετούμε ένα όχημα για κάθε ένα εξ' αυτών.
Βήμα 2:Για κάθε πελάτη αλλά και για κάθε ομάδα υπολόγισε το κόστος εισόδου με βάση τον αρχικό πελάτη.

Βήμα 3: Επίλυση του γενικευμένου προβλήματος εκχώρησης.

Ακολουθεί, έπειτα, η δεύτερη φάση της μεθόδου που έχει ένα μόνο βήμα και στο οποίο γίνεται η επίλυση και μορφοποίηση των τελικών διαδρομών.

Φάση 2

Βήμα 4: Επίλυση του προβλήματος για όλα τα σύνολα πελατών που έχουν εισαχθεί σε κάποιο όχημα και μορφοποίηση των τελικών διαδρομών.

Μέθοδος των δύο φάσεων των Christofides, Mignozzi and Toth

Ο συγκεκριμένος αλγόριθμος αποτελείται και αυτός από δύο φάσεις. Στην πρώτη φάση εφαρμόζεται ένας μεγάλος αριθμός δοκιμών για την ομαδοποίηση των λύσεων χρησιμοποιώντας ένα κριτήριο εκχωρήσεων ελάχιστου κόστους. Στην δεύτερη φάση ουσιαστικά λύνουμε ένα TSP και κρατάμε την καλύτερη από όλες τις λύσεις. Παρακάτω παρουσιάζεται ο αλγόριθμος με μορφή βημάτων:

Φάση 1

Βήμα 1: (Ακολουθητικές δοκιμές). Επιλέγουμε έναν πελάτη που δεν ανήκει στην διαδρομή (εστιασμένος πελάτης). Έπειτα επιλέγουμε ένα όχημα για τον συγκεκριμένο πελάτη.

Βήμα 2: Τοποθετούμε τους πελάτες που δεν ανήκουν σε κάποια διαδρομή σε κάποια ομάδα. Η επιλογή και τοποθέτηση κάθε πελάτη γίνεται με βάση ένα κόστος εκχώρησης που σχετίζεται άμεσα με τον πελάτη κάθε ομάδας, μέχρι να γεμίσει η χωρητικότητα των οχημάτων.

Βήμα 3: (Παράλληλες δοκιμές). Χρησιμοποιώντας όλους τους αρχικούς πελάτες για μια διαδρομή που υπολογίστηκαν στο πρώτο βήμα, απελευθερώνουμε όλους τους πελάτες από την ομάδα στην οποία ανήκουν.

Βήμα 4: Για κάθε πελάτη που είναι ελεύθερος, υπολογίζεται το κόστος εισόδου σε σχέση με τους αρχικούς πελάτες κάθε ομάδας. Έπειτα ελέγχονται όλες οι ομάδες για την εφικτότητα τους και επιλέγεται η ομάδα με το καλύτερο κόστος για τον πελάτη.

Βήμα 5: Εκχώρηση του πελάτη που επιλέχθηκε στο προηγούμενο βήμα στην ομάδα που αντιστοιχεί.
Βήμα 6: Επανάληψη των δύο προηγούμενων βημάτων μέχρι να μην υπάρχουν άλλοι ελεύθεροι πελάτες.

Φάση 2
Βήμα 7: Λύνουμε ένα TSP για κάθε ομαδοποίηση και επιλέγουμε την καλύτερη από τις λύσεις.

3.3 Μοντέρνοι Ευρετικοί Αλγόριθμοι

Οι συγκεκριμένοι αλγόριθμοι που έχουν αναπτυχθεί για την επίλυση του προβλήματος Δρομολόγησης Οχημάτων στηρίζονται στην τοπική αναζήτηση. Σε αυτές τις μεθόδους εξερευνάται το πεδίο της λύσης για να βρεθεί μια καλύτερη λύση. Υπάρχουν πέντε είδη μοντέρνων ευρετικών αλγορίθμων οι οποίοι είναι:

1. Προσομοιωμένη ανόπτηση.
2. Περιορισμένη αναζήτηση.
3. Γενετικοί αλγόριθμοι.
4. Νευρωνικά δίκτυα.
5. Ο αλγόριθμος των μυρμηγκιών.

Τα παραπάνω είδη των αλγορίθμων χρησιμοποιούν τα εξής στοιχεία:

- Χρησιμοποιούν έναν μεγάλο αριθμό επαναλήψεων.
- Περιλαμβάνουν έναν ή και περισσότερους πράκτορες.
- Λειτουργούν βάση ενός μηχανισμού συνεργασίας και ανταγωνισμού.
- Περιλαμβάνουν διαδικασίες αυτό-τροποποιήσεων των ευρετικών παραμέτρων ή ακόμα και της αναπαράστασης του προβλήματος.

3.3.1 Προσομοιωμένη ανόπτηση (Simulated Annealing)

Το όνομα του αλγορίθμου ή καλύτερα της στρατηγικής αυτής προέρχεται από την αναλογία ανάμεσα στην προσομοίωση της ανόπτησης των υλικών και την στρατηγική

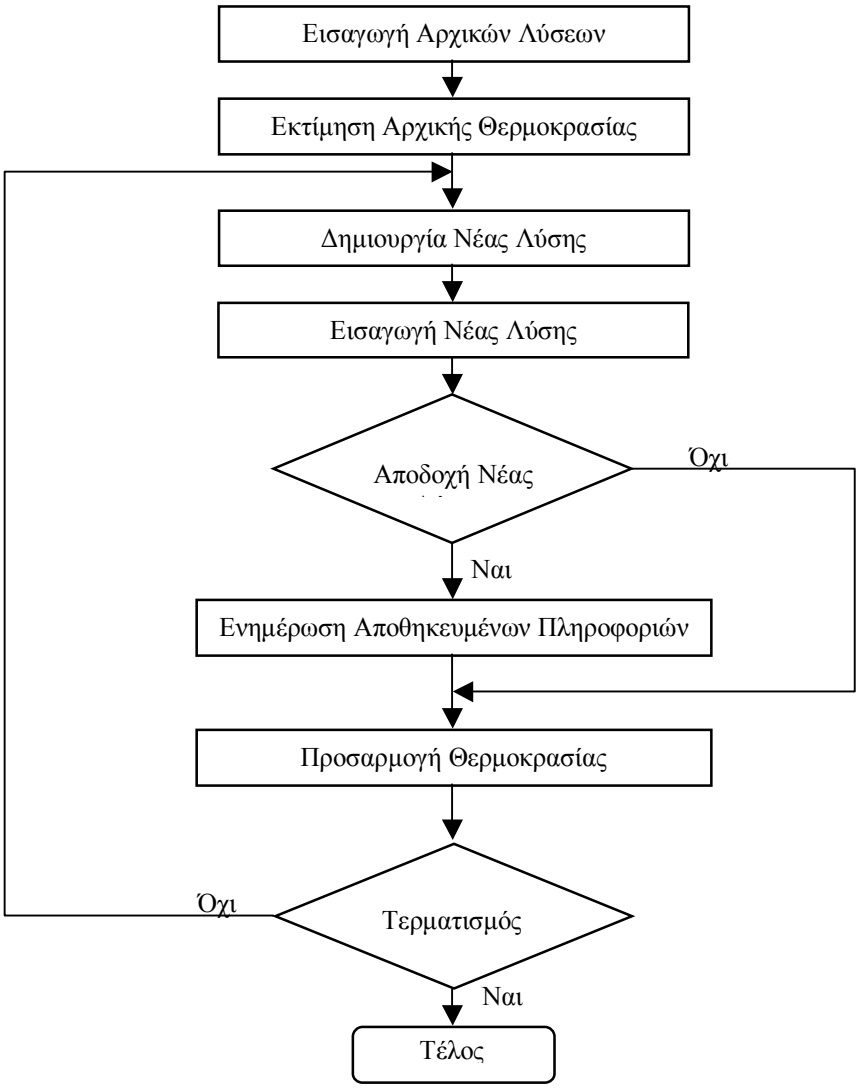
επίλυσης προβλημάτων συνδυαστικής βελτιστοποίησης. Οι συγκεκριμένοι αλγόριθμοι κινούνται συνεχώς στον εφικτό χώρο αναζήτησης των λύσεων και όταν σταματήσει η κίνηση έχουμε μια εφικτή λύση. Η προσομοιωμένη ανόπτηση εκκινεί από μια αρχική λύση και προχωρά σε κάθε επανάληψη σε μια καινούρια λύση στη γειτονιά της προηγούμενης λύσης μέχρι ένα κριτήριο τερματισμού να ικανοποιηθεί. Τα τρία βασικά κριτήρια τερματισμού που χρησιμοποιούνται είναι:

1. Η βέλτιστη τιμή f δεν έχει βελτιωθεί για ένα ποσοστό τουλάχιστον κ_1 συνεχών ανακυκλώσεων από Τ επαναλήψεις.
2. Ο αριθμός των αποδεχούμενων κινήσεων είναι μικρότερος από ένα ποσοστό τουλάχιστον κ_2 συνεχών ανακυκλώσεων από Τ επαναλήψεις.
3. Έχει εκτελεστεί ένας μεγάλος αριθμός συνεχόμενων ανακυκλώσεων από Τ επαναλήψεις.

Οι σημαντικότεροι αλγόριθμοι προσομοιωμένης ανόπτησης είναι:

- Ο αλγόριθμος των Alfa, Heragu και Chen.
- Ο αλγόριθμος της προσομοιωμένης ανόπτησης του Osman.

Στο επόμενο σχήμα παρουσιάζεται το διάγραμμα ροής της συγκεκριμένης μεθοδολογίας.



Σχήμα 3.1: Αλγόριθμος Προσομειωμένης Ανόπτησης

3.3.2 Περιορισμένη αναζήτηση (Tabu search)

Η μέθοδος περιορισμένης αναζήτησης χρησιμοποιεί έναν ευρετικό αλγόριθμο για να μετακινηθεί από μια λύση σε μια άλλη. Υπάρχει ωστόσο η περίπτωση ο συγκεκριμένος αλγόριθμος, όπως και οι υπόλοιποι μεθευρετικοί αλγόριθμοι, να "κολλήσει" σε τοπικό ελάχιστο. Όμως αντίθετα από την περίπτωση της στρατηγικής της προσομοιωμένης ανόπτησης η προσπάθεια για να υπερνικηθεί το ρίσκο του να "κολλήσει" η μέθοδος σε ένα τοπικό ελάχιστο γίνεται όχι τυχαία, αλλά με στρατηγική. Η συγκεκριμένη μέθοδος, λοιπόν, οδηγεί στην αναζήτηση στο χώρο των εφικτών λύσεων με την κίνηση σε συνεχώς καλύτερες λύσεις.

Συνοπτικά ο αλγόριθμος περιορισμένης αναζήτησης λειτουργεί ως εξής. Πρώτα πρώτα κατασκευάζεται μια αρχική λύση αξιοποιώντας όλα τα ειδικά χαρακτηριστικά του προβλήματος. Στη συνέχεια υπολογίζεται η τιμή της συνάρτησης κόστους για τις υποψήφιες λύσεις στη γειτονιά της αρχικής λύσεις προσπαθώντας να κινηθεί η μέθοδος στη γειτονιά της καλύτερης λύσης. Συνήθως είναι η καλύτερη τιμή της συνάρτησης κόστους που καθορίζει το πόσο καλή είναι μια λύση. Για τον υπολογισμό του κόστους υπολογίζουμε μια τιμή κίνησης αντί του υπολογισμού του ολικού κόστους. Χρησιμοποιούμε, λοιπόν, μια συνάρτηση υπολογισμού των αλλαγών που σχετίζονται με την κίνηση στη θέση της συνάρτησης κόστους. Για την αποφυγή να "κολλήσει" ο αλγόριθμος σε ένα τοπικό ελάχιστο, χρησιμοποιούνται όλες οι πληροφορίες που έχουν συλλεχθεί στις προηγούμενες επαναλήψεις. Το σύνολο των λύσεων που καθορίζονται από τις συγκεκριμένες πληροφορίες κατασκευάζει τη λίστα περιορισμένης αναζήτησης (tabu list). Η λειτουργία αυτή, που ονομάζεται μνήμη μικρής διάρκειας, θυμάται τις κατευθύνσεις που έχουν τελευταία εξερευνηθεί. Αυτές οι απαγορευμένες πια κατευθύνσεις αναζήτησης ονομάζονται περιορισμένες κινήσεις και με κ θέτουμε το μέγεθος της λίστας. Με τη χρήση αυτής της λίστας αποφεύγεται η πιθανότητα λήψης των ίδιων λύσεων ανάμεσα στις λύσεις που προκύπτουν στις διάφορες επαναλήψεις. Επιπρόσθετα, πολλές φορές χρησιμοποιούνται περιορισμοί που βασίζονται στη συχνότητα. Πιο συγκεκριμένα, σε αυτήν την περίπτωση υπολογίζουμε πόσες φορές εμφανίζεται κάθε κίνηση στη διάρκεια της αναζήτησης και με βάση τους περιορισμούς συχνότητας εμποδίζεται η εμφάνιση επιπλέον λύσεων προς μια συγκεκριμένη κατεύθυνση. Αυτό επιτυγχάνεται με τη χρήση μιας μνήμης μακράς διαρκείας που βοηθά στην ποικιλομορφία των λύσεων.

Για την πλήρη εκμετάλλευση των περιορισμών συχνότητας μπορούν να χρησιμοποιηθούν κινήσεις με βάση τα ιδιαίτερα χαρακτηριστικά. Μεταβαίνοντας από μια λύση στην επόμενη, ένα χαρακτηριστικό της συγκεκριμένης κίνησης μπορεί να περικλείει οποιαδήποτε πιθανή λύση που είναι σε θέση να μεταβάλλει την κίνηση. Οι καταγεγραμμένες κινήσεις συνήθως χρησιμοποιούνται για να παρεμποδίσουν κινήσεις που θα αντιστρέψουν τις αλλαγές που αναπαριστούνται από αυτές από το να μην επιλεγούν. Κάτι που θα σήμαινε ότι αν μια καινούρια λύση που επιτυγχάνεται από μια δεδομένη κίνηση είναι η s', τότε η επόμενη λύση αποκλείεται να είναι πάλι η s που ήταν ίση με την αρχική. Περιορισμοί δεν έχουν να κάνουν μόνο με τις λύσεις αλλά και με τον αριθμό των επαναλήψεων. Υπάρχουν όμως και ορισμένες περιπτώσεις που θα ήταν καλύτερα να μη ληφθούν υπόψη οι παραπάνω περιορισμοί. Η "αγνόηση" των περιορισμών, που ονομάζεται κριτήριο ανασκόπησης γίνεται επειδή υπάρχει περίπτωση να οδηγηθούμε σε καλύτερη λύση. Εάν, όμως η λύση δεν είναι τόσο καλή όσο θα περιμέναμε τότε εφαρμόζουμε τους περιορισμούς και επιλύουμε ξανά το πρόβλημα. Επειδή, τέλος, η περιορισμένη αναζήτηση δε συγκλίνει με φυσικό τρόπο, για να σταματήσει ο αλγόριθμος κρίνεται απαραίτητη η χρήση κάποιων κριτηρίων τερματισμού. Συνήθως το κριτήριο που χρησιμοποιείται είναι ο αριθμός των επαναλήψεων από την τελευταία βελτιστοποίηση της λύσης. Παρακάτω αναγράφονται ορισμένοι από τους πιο συχνά χρησιμοποιούμενους αλγορίθμους περιορισμένης αναζήτησης.

- Ο αλγόριθμος του Willard.
- Ο αλγόριθμος περιορισμένης αναζήτησης του Osman.
- Ο αλγόριθμος TABUROUTE.
- Ο αλγόριθμος του Tailard.
- Ο αλγόριθμος των Xu και Kelly.
- Ο αλγόριθμος των Rego και Roucairol.
- Ο αλγόριθμος προσαρμοστικής μνήμης των Rochat και Taillard.

3.3.3 Γενετικοί Αλγόριθμοι

Οι Γενετικοί αλγόριθμοι είναι ουσιαστικά στρατηγικές αναζήτησης με τη χρήση τυχαιότητας που έχουν ως κύριο στόχο την προσομοίωση της φυσικής εξέλιξης των πληθυσμών όλων των ειδών. Στους αλγορίθμους αυτούς επιτρέπεται ο συνδυασμός

τμημάτων των πληροφοριών που λαμβάνονται από τους γονείς με σκοπό τη δημιουργία απογόνων. Η επιτυχία των γενετικών αλγορίθμων στηρίζεται στο συνδυασμό των κομματιών αυτών που αποτελούν και τον σημαντικότερο παράγοντα.

Για την περιγραφή ενός Γενετικού αλγορίθμου θεωρούμε έναν πληθυσμό X^t στον οποίο καθορίζεται μια συνάρτηση f. Στις περισσότερες των περιπτώσεων στόχο αποτελεί η μεγιστοποίηση αυτής της συνάρτησης, όμως στο πρόβλημα Δρομολόγησης Οχημάτων θέλουμε να επιτύχουμε την ελαχιστοποίηση της. Παρακάτω περιγράφεται βήμα προς βήμα ένας γενετικός αλγόριθμος όπου κ_1 είναι ο αριθμός των γενεών και κ_2 ο αριθμός των επιλογών ανά γενιά.

Βήμα 1: Τυχαία επιλογή των δύο γονέων.
Βήμα2: Δημιουργία δύο απογόνων από τους γονείς με ανταλλαγή κάποιων στοιχείων έπειτα από ένα τυχαίο σημείο όπως φαίνεται και στο παρακάτω παράδειγμα.

Γονέας 1	1	0	1	0	0
Γονέας 2	0	0	1	1	1
Απόγονος 1	1	0	1	1	1
Απόγονος 2	0	0	1	0	0

Βήμα 3: Εφαρμογή μιας τυχαίας μετάλλαξης σε κάθε απόγονο σε ορισμένα χαρακτηριστικά που έχουν πολύ μικρή πιθανότητα να συμβούν.
Βήμα 4: Δημιουργία της καινούριας λύσης με τη διαγραφή των 2 κ_2 χειρότερων λύσεων και την αντικατάστασή τους από 2 κ_2 απογόνους που μόλις δημιουργήθηκαν.

Η εφαρμογή γενετικών αλγορίθμων Στο πρόβλημα Δρομολόγησης Οχημάτων εμφανίζει αρκετές δυσκολίες.

- Είναι πολύ δύσκολο και περίπλοκο να αναπαρασταθεί ένα πρόβλημα Δρομολόγησης Οχημάτων και ειδικότερα οι διαδρομές του.
- Εύκολα δημιουργούνται μη εφικτοί απόγονοι από τους γονείς.

Παρακάτω παρουσιάζονται ορισμένες από τις σημαντικότερες εφαρμογές των γενετικών αλγορίθμων:

- Ο αλγόριθμος Gideon.
- Ο αλγόριθμος Generous.

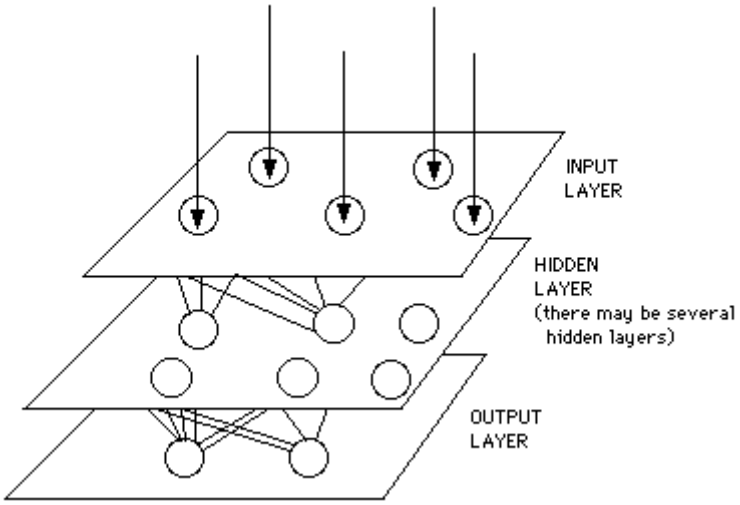
3.3.4 Νευρωνικά δίκτυα

Τα νευρωνικά δίκτυα αποτελούν ένα σύστημα το οποίο βασίζεται στον ανθρώπινο εγκέφαλο. Με τα νευρωνικά δίκτυα γίνεται μία απόπειρα προσομοίωσης μέσω εξειδικευμένου λογισμικού και hardware πολλαπλών στρωμάτων που αποτελούνται από απλά στοιχεία επεξεργασίας, τους λεγόμενους νευρώνες. Κάθε νευρώνας συνδέεται με κάποιους από τους γειτονικούς του νευρώνες με διάφορους συντελεστές (βάρη) οι οποίοι αναπαριστούν την ισχύ των δεσμών αυτών. Η εκμάθηση του νευρωνικού δικτύου ολοκληρώνεται ρυθμίζοντας τα βάρη αυτά έτσι ώστε το νευρωνικό δίκτυο να βγάλει καλά αποτελέσματα.

Για να επιτευχθεί η δημιουργία ενός νευρωνικού δικτύου το οποίο θα δίνει ικανοποιητικά αποτελέσματα θα πρέπει να ακολουθηθεί η τακτική δοκιμής- λάθους (trial and error), που σημαίνει πως πρέπει να γίνει ένας μεγάλος αριθμός δοκιμών μέχρι το νευρωνικό δίκτυο να εξάγει σωστά αποτελέσματα. Η σύνθεση ενός νευρωνικού δικτύου είναι ιδιαίτερα πολύπλοκη και απαιτεί την εφαρμογή των παρακάτω σταδίων:

- Τοποθέτηση των νευρώνων σε διάφορα στρώματα (layers).
- Επιλογή της σύνδεσης μεταξύ των νευρώνων ανάμεσα στα διάφορα στρώματα αλλά και ανάμεσα τους στο ίδιο στρώμα.
- Επιλογή του τρόπου με τον οποίο ένας νευρώνας δέχεται δεδομένα καθώς και του τρόπου με τον οποίο τα εξάγει.
- Καθορισμός της ισχύος της σύνδεσης μέσα στο δίκτυο επιτρέποντας στο δίκτυο να μάθει-διδαχθεί τις τιμές των βαρών χρησιμοποιώντας μία διαδικασία εκμάθησης

Τα τεχνητά νευρωνικά δίκτυα σχηματίζονται χρησιμοποιώντας στρώματα τα οποία συνδέονται μεταξύ τους αλλά με πολλούς διαφορετικούς τρόπους σύνδεσης μεταξύ τους. Συνήθως, όμως, όλα τα τεχνητά νευρωνικά δίκτυα ακολουθούν τον ίδιο τύπο κατασκευής. Κάποιοι από τους νευρώνες επικοινωνούν ουσιαστικά με το περιβάλλον για να λάβουν τα δεδομένα εισόδου, ενώ άλλοι νευρώνες εξάγουν στο περιβάλλον τα αποτελέσματα. Οι υπόλοιποι νευρώνες παραμένουν κρυφοί και επεξεργάζονται τα δεδομένα προκειμένου στη συνέχεια να εξαχθούν ικανοποιητικά αποτελέσματα. Μία απλή κατασκευή τεχνητού νευρωνικού δικτύου φαίνεται στο παρακάτω σχήμα:



Σχήμα 3.2: Αναπαράσταση ενός τεχνητού νευρωνικού δικτύου

Στο παραπάνω σχήμα οι νευρώνες υπάρχουν σε όλα τα στρώματα του δικτύου. Το στρώμα εισαγωγής (input layer) αποτελείται από νευρώνες οι οποίοι δέχονται τα δεδομένα εισόδου από το εξωτερικό περιβάλλον. Το εξωτερικό στρώμα αποτελείται από νευρώνες οι οποίοι επικοινωνούν με το εξωτερικό περιβάλλον για να δώσει τα αποτελέσματα. Μεταξύ των 2 αυτών στρωμάτων υπάρχουν ενδιάμεσα 'κρυφά' στρώματα.

Ο εγκέφαλος μαθαίνει από το παρελθόν. Τα νευρωνικά δίκτυα με τις εναλλαγές στα βάρη σύνδεσης προκαλούν την εκμάθηση του δικτύου ώστε να δώσει σωστά αποτελέσματα. Το σύστημα αποκτά νέες γνώσεις κάθε φορά που αλλάζουμε τα βάρη

αυτά. Η ικανότητα εκμάθησης ενός νευρωνικού δικτύου εξαρτάται από την αρχιτεκτονική του καθώς και από τον αλγόριθμο που χρησιμοποιείται για την εκμάθησή του. Η εκπαίδευση ενός νευρωνικού δικτύου γίνεται με 3 μεθόδους:

1. **Εκμάθηση χωρίς επιτήρηση:** Οι 'κρυφοί' νευρώνες στην περίπτωση αυτή οργανώνονται μόνοι τους χωρίς βοήθεια από το εξωτερικό τους περιβάλλον.
2. **Εκμάθηση με ενίσχυση:** Για τη λειτουργία της χρειάζεται συνεργασία-βοήθεια από το εξωτερικό περιβάλλον. Οι κρυφοί νευρώνες ενώνονται τυχαία και οι εναλλαγές στη συνδεσμολογία τους στηρίζονται στο πόσο κοντά βρίσκεται το νευρωνικό δίκτυο στη λύση του προβλήματος.
3. **Back propagation:** Η μέθοδος αυτή λειτουργεί όπως η μέθοδος εκμάθησης με ενίσχυση με τη διαφορά όμως ότι το δίκτυο επεξεργάζεται και πληροφορίες που αφορούν σφάλματα και τελικά γίνεται φιλτράρισμα των πληροφοριών αυτών αλλάζοντας έτσι τις συνδέσεις μεταξύ των στρωμάτων του δικτύου με αποτέλεσμα την εξαγωγή πολύ καλών αποτελεσμάτων.

3.3.5 Ο αλγόριθμος των μυρμηγκιών

Ο αλγόριθμος των μυρμηγκιών κατασκευάζει λύσεις με την επιτυχή επιλογή των πόλεων που θα επισκεφτούν μέχρι την ολοκλήρωση των επισκέψεων όλων των πόλεων. Κάθε φορά που επιλέγεται μια πόλη που παραβιάζει τους περιορισμούς, το όχημα γυρίζει αυτόματα στην αποθήκη και στη συνέχεια αρχίζει μια καινούρια διαδρομή. Στην επιλογή μιας πόλης δύο χαρακτηριστικά λαμβάνονται υπόψη:

1. Πόσο καλή είναι η πόλη που θα επιλεγεί. Μια πληροφορία που αποθηκεύεται στην αίσθηση που έχουν τα μυρμήγκια στο pheromone και που παρουσιάζεται από το διάνυσμα t_{ij} και σχετίζεται με κάθε τόξο.
2. Πόσο ελπιδοφόρα είναι η επιλογή της συγκεκριμένης πόλης.

Παρακάτω παρουσιάζεται βηματικά ο συγκεκριμένος αλγόριθμος:

Βήμα 1: Αρχικοποίηση

Βήμα 2: Για έναν δεδομένο, μέγιστο αριθμό επαναλήψεων:

Βήμα 2.1: Για κάθε μυρμήγκι δημιουργείται μια καινούρια λύση

Βήμα 2.2: Βελτίωση όλων των διαδρομών με τη χρήση ενός 2- opt αλγορίθμου.

Βήμα 2.3: Ενημέρωση του πρώτου κριτηρίου.

Κεφάλαιο 4: Ανάπτυξη αλγορίθμου

4.1 Εισαγωγή στην περιγραφή του προγράμματος

Στόχος της συγκεκριμένης διπλωματικής ήταν η δημιουργία ενός προγράμματος επίλυσης του προβλήματος Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και με Παραλαβές και Διανομές και η βελτίωση της λύσης με την χρήση αλγορίθμων Τοπικής Αναζήτησης. Ο προγραμματισμός έγινε με τη χρήση της γλώσσας FORTRAN vf90 και σε περιβάλλον Linux και πιο συγκεκριμένα με τη χρήση Suse Linux 8.2. Στο κεφάλαιο αυτό θα γίνει μια λεπτομερής ανάπτυξη του προγράμματος που συνοπτικά ακολούθησε την παρακάτω διαδρομή:

- Επίλυση του προβλήματος του Περιπλανώμενου Πωλητή με τη χρήση του αλγορίθμου Nearest Neighborhood (Πλησιέστερου Γείτονα).
- Μετατροπή του προγράμματος επίλυσης του προβλήματος του Περιπλανώμενου Πωλητή σε πρόγραμμα επίλυσης του προβλήματος Δρομολόγησης Οχημάτων με χρήση απλών περιορισμών χωρητικότητας οχημάτων.
- Μετατροπή του προγράμματος επίλυσης του προβλήματος Δρομολόγησης Οχημάτων σε πρόγραμμα επίλυσης του προβλήματος Δρομολόγησης Οχημάτων με Παραλαβές και Διανομές.
- Μετατροπή του προγράμματος επίλυσης του προβλήματος Δρομολόγησης Οχημάτων με παραλαβές και διανομές σε πρόγραμμα επίλυσης του προβλήματος Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και με Παραλαβές και Διανομές.
- Βελτίωση της λύσης που βρέθηκε στο προηγούμενο βήμα με την χρήση αλγορίθμου τοπικής αναζήτησης 2- opt.
- Βελτίωση της λύσης που βρέθηκε στο προηγούμενο βήμα με την χρήση αλγορίθμου τοπικής αναζήτησης 3- opt.
- Επαναληπτική εφαρμογή των δύο παραπάνω βημάτων.

4.2 Ο αλγόριθμος Nearest Neighborhood

Η διαδικασία του πλησιέστερου γείτονα χρησιμοποιήθηκε για τον υπολογισμό της διαδρομής που θα ακολουθούσε ο πωλητής. Σύμφωνα με αυτήν, ο πωλητής ξεκινά αρχικά από μια πόλη και στη συνέχεια επισκέπτεται κάθε φορά την πλησιέστερη πόλη από αυτήν που βρισκόταν προηγουμένως. Η συγκεκριμένη διαδικασία συνεχίζεται μέχρι ο πωλητής να έχει επισκεφθεί όλες τις πόλεις ανεξαιρέτως. Η τελευταία πόλη που επισκέπτεται είναι η αρχική από όπου εκκίνησε ώστε να κλείσει ο κύκλος. Παρακάτω παρουσιάζεται βηματικά η μέθοδος του πλησιέστερου γείτονα:

Βήμα 1: Εκκίνηση από έναν οποιονδήποτε κόμβο για τη δημιουργία του μονοπατιού.
Βήμα 2: Βρίσκουμε τον κόμβο που είναι πλησιέστερα στον τελευταίο κόμβο που προστέθηκε στη διαδρομή και τον προσθέτουμε στη διαδρομή μας.
Βήμα 3: Επανάληψη του προηγούμενου βήματος μέχρι να συμπεριληφθούν όλοι οι κόμβοι στην διαγραφόμενη διαδρομή.

Ο υπολογισμός των αποστάσεων των κόμβων μεταξύ τους γίνεται με βάση τον τύπο:

$D_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$, όπου

D_{ij} : η υπολογιζόμενη απόσταση του κόμβου j από τον κόμβο i.

x_i : η θέση του κόμβου i ως προς τον x- άξονα.

y_j : η θέση του κόμβου j ως προς τον y- άξονα.

4.3 Επίλυση του προβλήματος του Περιπλανώμενου Πωλητή με τη χρήση του αλγορίθμου Nearest Neighborhood.

Σε ένα πρόβλημα περιπλανώμενου πωλητή, ένας πωλητής πρέπει να επισκεφθεί όλες τις πόλεις ξεχωριστά ξεκινώντας από και καταλήγοντας στην ίδια πόλη- αφετηρία. Το πρόβλημα αυτό απαιτεί τον καθορισμό ενός κύκλου ελαχίστου κόστους που περνά από κάθε κόμβο του συσχετιζόμενου γραφήματος ακριβώς μια φορά.. Στην

συγκεκριμένη διπλωματική το κόστος του ταξιδιού μεταξύ δύο τοποθεσιών δεν εξαρτάται από την κατεύθυνση του γραφήματος (συμμετρικό πρόβλημα). Η απόσταση (κόστος) των πόλεων μεταξύ τους (για όλα τα ζεύγη των πόλεων) υπολογίζεται βάση του αλγορίθμου Nearest Neighborhood που περιγράφηκε στην αμέσως προηγούμενη παράγραφο και τα κόστη που υπολογίζονται, τα αποθηκεύουμε σε έναν μονοδιάστατο πίνακα. Η αποθήκευση των αποστάσεων γίνεται ακολουθώντας την εξής διαδικασία:

Εκκινώντας από την αποθήκη υπολογίζουμε τις αποστάσεις της D_{ij} (απόσταση από τον κόμβο i στον κόμβο j) από κάθε κόμβο ξεχωριστά και τις αποθηκεύουμε μια προς μια στον μονοδιάστατο πίνακα μας **d**. Μόλις γίνει αυτό προχωρούμε στην επόμενη πόλη και υπολογίζουμε κάθε φορά την απόστασή της από τις υπόλοιπες πόλεις και την αποθηκεύουμε στην επόμενη θέση του πίνακα. Αυτή η επαναληπτική διαδικασία συνεχίζεται μέχρι να υπολογιστούν όλες οι αποστάσεις μεταξύ των κόμβων. Ο πίνακας στο τέλος θα έχει αποθηκευμένα

n*(n-1)/2 στοιχεία, όπου με n συμβολίζεται ο συνολικός αριθμός των κόμβων. Παρακάτω παρουσιάζεται σε ένα παράδειγμα με πέντε κόμβους (η αποθήκη συμβολίζεται από τον κόμβο ένα) η μορφή του πίνακα **d**:

d(k)=

D(1,2)	D(1,3)	D(1,4)	D(1,5)	D(2,3)	D(2,4)	D(2,5)	D(3,4)	D(3,5)	D(4,5)

Για την αποφυγή, κατά τον υπολογισμό της διαδρομής, της περίπτωσης ο πωλητής να ξαναπεράσει από έναν κόμβο, χρησιμοποιήθηκε ένας μονοδιάστατος πίνακας **e** ιδίου μεγέθους με τον πίνακα κόστους **d**. Τα στοιχεία του αρχικά είναι ίσα με το μηδέν και κάθε φορά που εισάγεται ένας κόμβος στην διαδρομή, τα στοιχεία του **e** που αφορούν τον συγκεκριμένο κόμβο παίρνουν την τιμή ένα. Έτσι αν χρησιμοποιήσουμε το προηγούμενο παράδειγμα με τους πέντε κόμβους και υποθέσουμε ότι ο πρώτος κόμβος που θα επισκεφτούμε μόλις φύγουμε από την κεντρική αποθήκη (κόμβος 1) είναι ο κόμβος 3, ο πίνακας θα έχει τις εξής τιμές:

e(k)=

0	1	0	0	1	0	0	1	1	0

Αυτή η διαδικασία συνεχίζεται μέχρι ο πίνακας e να μην έχει άλλα μηδενικά στοιχεία. Για να γίνει όμως πιο εύκολη η συγκεκριμένη διαδικασία χρησιμοποιήσαμε δύο ακόμη μονοδιάστατους πίνακες **from(k)** και **to(k)** ίδιου μεγέθους με τους πίνακες **d** kai **e**. Ο πίνακας **from(k)** έχει ως στοιχεία τους κόμβους από όπου εκκινεί το όχημα σε κάθε στοιχείο του **d**, ενώ ο πίνακας **to(k)** έχει ως στοιχεία τους κόμβους όπου καταλήγει το όχημα. Παρακάτω φαίνονται τα στοιχεία των δύο πινάκων για το ίδιο παράδειγμα σύμφωνα με το οποίο γεμίσαμε και τους πίνακες **d** kai **e**.

from(k)=

1	1	1	1	2	2	2	3	3	4

to(k)=

2	3	4	5	3	4	5	4	5	5

Έχοντας, λοιπόν, κατασκευάσει αυτούς τους τέσσερις πίνακες, η επίλυση του προβλήματος του περιπλανώμενου πωλητή ακολούθησε την εξής διαδικασία:

Βήμα 1: Εκκινούμε τη διαδικασία τοποθετώντας σε έναν νέο πίνακα **from1(k1)** ως πρώτο στοιχείο (k1 =1) την κεντρική αποθήκη και έπειτα βρίσκουμε το μικρότερο κόστος- απόσταση από τον κόμβο k1 (κεντρική αποθήκη) προς τους υπόλοιπους κόμβους και αποθηκεύονται:

1. η θέση του πίνακα **d** σε μια μεταβλητή **m**.
2. το κόστος σε έναν πίνακα **cost** (μονοδιάστατο) υπολογισμού του κόστους της συνολικής διαδρομής.

Βήμα 2: Το **to(m)** μας υποδεικνύει ποιος θα είναι ο επόμενος κόμβος που θα μπει στην διαδρομή μας. Επομένως αποθηκεύεται η τιμή του σε έναν νέο πίνακα **to1(k1)**, αυξάνεται το **k1** κατά ένα και επαναλαμβάνουμε το βήμα 1 τοποθετώντας αρχικά στον πίνακα **from1(k1)** την τιμή που είναι αποθηκευμένη στον πίνακα **to(m)**. Σε αυτό το βήμα γίνεται και η αύξηση των στοιχείων του πίνακα **e** κατά 1, που έχουν ως στοιχείο τον κόμβο που εισήχθη στην καινούρια διαδρομή.

Οι πίνακες **from1** και **to1** αποθηκεύουν την διαδρομή ενώ ο πίνακας **cost** το κόστος της. Η διαδικασία αυτή συνεχίζεται μέχρι να εισαχθούν όλοι οι κόμβοι στον κύκλο, δηλαδή μέχρι ο πίνακας **e** να μην έχει άλλα μηδενικά στοιχεία. Τότε στην τελευταία θέση του πίνακα **to1** αποθηκεύεται ο κόμβος της κεντρικής αποθήκης για να κλείσει η διαδρομή. Παρακάτω παρουσιάζεται η μορφή των πινάκων **from1**, **to1** και **cost** για το παράδειγμα των πέντε πόλεων.

1
3
5
2
4

3
5
2
4
1

D(1,3)
d(k1-1)+D(3,5)
d(k1-1)+D(5,2)
d(k1-1)+D(2,4)
d(k1-1)+D(4,1)

From1 **to1** **cost**

4.4 Μετατροπή του προγράμματος επίλυσης του προβλήματος του Περιπλανώμενου Πωλητή σε πρόγραμμα επίλυσης του προβλήματος Δρομολόγησης Οχημάτων με χρήση απλών περιορισμών χωρητικότητας οχημάτων.

Στο βασικό πρόβλημα δρομολόγησης οχημάτων, όπως προαναφέρθηκε και στην εισαγωγή, σε αντίθεση με το πρόβλημα του περιοδεύοντος πωλητή, στόχος είναι η δημιουργία άνω του ενός δρομολογίων. Τα οχήματα είναι όμοια μεταξύ τους (ίδιας

χωρητικότητας), εκκινούν από την κεντρική αποθήκη και εκτελούν το δρομολόγιο τους με μοναδικούς περιορισμούς, τους περιορισμούς χωρητικότητας. Στόχος της επίλυσης του συγκεκριμένου προβλήματος είναι η ελαχιστοποίηση του συνολικού κόστους εξυπηρετώντας τους πελάτες.

Για τη μετατροπή του προγράμματος επίλυσης του περιπλανώμενου πωλητή σε πρόγραμμα επίλυσης του προβλήματος Δρομολόγησης Οχημάτων, η διαδικασία που ακολουθήθηκε ήταν: στην διαδρομή που είχε ήδη δημιουργηθεί κατά την επίλυση του προβλήματος του περιπλανώμενου πωλητή, να ελέγχεται αν η ποσότητα που φορτώνεται σε ένα όχημα κατά την εισαγωγή ενός κόμβου ξεπερνά την χωρητικότητα του οχήματος. Αν συμβαίνει αυτό το γεγονός, το συγκεκριμένο όχημα επιστρέφει στην αποθήκη ενώ ο κόμβος εξυπηρετείται από ένα άλλο όχημα που εκκινεί ταυτόχρονα από την αποθήκη.

Για την επίλυση του προβλήματος Δρομολόγησης Οχημάτων χρησιμοποιήθηκαν τέσσερις καινούριοι πίνακες, ο **from2**, ο **to2**, ο **cap** και ο **cost2**. Ο **from2** και ο **to2** χρησιμοποιούνται, όπως και προηγουμένως, για την αποθήκευση των κόμβων με βάση τις διαδρομές που εκτελούν τα οχήματα. Στους πίνακες αυτούς παρατηρείται ο κόμβος 1, που αντιστοιχεί στην κεντρική αποθήκη, παραπάνω από μια φορές. Το γεγονός αυτό οφείλεται στο ότι υπάρχουν παραπάνω από ένας κύκλοι. Ο πίνακας **cap** χρησιμοποιείται για τον υπολογισμό της χωρητικότητας των οχημάτων κατά την εισαγωγή κόμβων σε κάθε διαδρομή και για τον έλεγχο αν αυτή ξεπερνά τη μέγιστη χωρητικότητας τους, κάτι που θα σημάνει τη δημιουργία μιας καινούριας διαδρομής. Αντίστοιχα ο πίνακας **cost2** χρησιμοποιείται για την αποθήκευση του συνολικού κόστους των διαδρομών που υπολογίζονται κάθε φορά που τοποθετείται ένας νέος κόμβος στη διαδρομή.
Το μέγεθος των πινάκων είναι ίσο με τον αριθμό των πελατών προσθέτοντας τον αριθμό των κύκλων που υπολογίζονται κατά την επίλυση του προβλήματος Δρομολόγησης Οχημάτων .Παρακάτω παρουσιάζεται ο αλγόριθμος του προβλήματος υπό μορφή βημάτων:

Βήμα 1: Εκκινούμε τη διαδικασία τοποθετώντας σε έναν νέο πίνακα **from1(k1)** ως πρώτο στοιχείο (k1 =1) την κεντρική αποθήκη και έπειτα βρίσκουμε το μικρότερο

κόστος- απόσταση από τον κόμβο k1 (κεντρική αποθήκη) προς τους υπόλοιπους κόμβους και αποθηκεύονται:

1. η θέση του πίνακα **d** σε μια μεταβλητή **m**.
2. το κόστος σε έναν πίνακα **cost** (μονοδιάστατο) υπολογισμού του κόστους της συνολικής διαδρομής.

Βήμα 2: Το **to(m)** μας υποδεικνύει ποιος θα είναι ο επόμενος κόμβος που θα μπει στην διαδρομή μας. Επομένως αποθηκεύεται η τιμή του σε έναν νέο πίνακα **to1(k1)**, αυξάνεται το **k1** κατά ένα και επαναλαμβάνουμε το βήμα 1 τοποθετώντας αρχικά στον πίνακα **from1(k1)** την τιμή που είναι αποθηκευμένη στον πίνακα **to(m)**. Σε αυτό το βήμα γίνεται και η αύξηση των στοιχείων του πίνακα **e** κατά 1, που έχουν ως στοιχείο τον κόμβο που εισήχθη στην καινούρια διαδρομή.

Βήμα 3: Αρχικά τοποθετούμε στον πίνακα **from2(k)** ως πρώτο στοιχείο (k =1) την κεντρική αποθήκη και έπειτα τοποθετούμε στο **to2(k)** τον κόμβο που βρίσκεται στην ίδια θέση του πίνακα **to1** αφού ελεγχθεί πρώτα ότι η χωρητικότητα του οχήματος, με την προθήκη των εμπορευμάτων από τον κόμβο που είναι προς εισαγωγή, είναι μικρότερη από τη μέγιστη χωρητικότητα του. Σε αντίθετη περίπτωση το συγκεκριμένο όχημα (**to2(k)** =1) επιστρέφει στην αποθήκη και ένα νέο, άδειο όχημα συνεχίζει τη διαδρομή .

Βήμα 4: Επανάληψη του προηγούμενου βήματος , αυξάνοντας το k κατά ένα, μέχρι όλοι οι πελάτες να έχουν εξυπηρετηθεί.

Παρακάτω ακολουθεί ένα παράδειγμα της επίλυσης του αλγορίθμου Δρομολόγησης για δέκα πόλεις. Θεωρούμε ότι οι πίνακες **from1** και **to1** είναι οι ακόλουθοι:

1
3
5
4
7
9
2
8
6
10

from1

3
5
4
7
9
2
8
6
10
1

to1

Παράλληλα, θεωρούμε πως το αρχικό μας όχημα όταν φθάνει στον κόμβο 9, δεν μπορεί να τον εισάγει στην διαδρομή του γιατί παραβιάζονται οι περιορισμοί χωρητικότητας. Οι πίνακες **from2** και **to2** λοιπόν θα έχουν τη μορφή:

1
3
5
4
7
1
9
2
8
6
10

from2

3
5
4
7
1
9
2
8
6
10
1

to2

Από τα αποτελέσματα που εμφανίζονται στους πίνακες **from2** και **to2** παρατηρούμε ότι για την επίλυση του συγκεκριμένου οχήματος χρειαζόμαστε δύο οχήματα που θα εκτελούν τις εξής διαδρομές:

Όχημα 1: 1→ 3→ 5→ 4→7→1

Όχημα 2: 1→ 9→2→ 8→ 6→10→1

4.5 Μετατροπή του προγράμματος επίλυσης του προβλήματος Δρομολόγησης Οχημάτων σε πρόγραμμα επίλυσης του προβλήματος Δρομολόγησης Οχημάτων με Παραλαβές και Διανομές.

Στο πρόβλημα Δρομολόγησης Οχημάτων με Παραλαβές και Διανομές ένα σύνολο από πελάτες πρέπει να εξυπηρετηθούν από έναν δεδομένο αριθμό οχημάτων που εκκινούν και πάλι από την κεντρική αποθήκη της εταιρίας. Στο συγκεκριμένο πρόβλημα όμως κάθε πελάτης σχετίζεται με δύο ποσότητες d_i και p_i που υποδηλώνουν τον αριθμό των προϊόντων που πρέπει να διανεμηθούν και να παραληφθούν από τον συγκεκριμένο πελάτη.

Παράλληλα πρέπει να ισχύει πάντα ο περιορισμός πως σε κάθε πελάτη η διανομή των προϊόντων θα γίνεται πάντα πριν από την παραλαβή, ώστε η συνολική φόρτωση του οχήματος τη χρονική στιγμή που φτάνει το όχημα σε έναν καινούριο πελάτη να είναι ίση με την αρχική φόρτωση από την αποθήκη μείον το άθροισμα των προϊόντων που είχε διανύμει στους προηγούμενους πελάτες συν το άθροισμα των προϊόντων που ικανοποιήθηκαν με παραλαβές από τους προηγούμενους πελάτες.

Για τη μετατροπή του προγράμματος επίλυσης Δρομολόγησης Οχημάτων σε πρόγραμμα επίλυσης Δρομολόγησης Οχημάτων με Παραλαβές και Διανομές χρησιμοποιήσαμε έναν ακόμη πίνακα **cap2** που θα αποθηκεύονται σε αυτόν οι ποσότητες που το όχημα θα παραλαμβάνει από τον κάθε πελάτη. Οι πίνακες **from2** και ο **to2** χρησιμοποιούνται, όπως και προηγουμένως, για την αποθήκευση των κόμβων με βάση τις διαδρομές που εκτελούν τα οχήματα. Στους πίνακες αυτούς

παρατηρείται και πάλι το φαινόμενο ο κόμβος 1, που αντιστοιχεί στην κεντρική αποθήκη, να εμφανίζεται παραπάνω από μια φορές, κάτι που οφείλεται στο γεγονός ότι υπάρχουν παραπάνω από ένας κύκλοι. Αντίστοιχα ο πίνακας **cost2** χρησιμοποιείται για την αποθήκευση του συνολικού κόστους των διαδρομών που υπολογίζονται κάθε φορά που τοποθετείται ένας νέος κόμβος στη διαδρομή. Στη συνέχεια παρουσιάζεται ο αλγόριθμος επίλυσης του προβλήματος Δρομολόγησης Οχημάτων με Παραλαβές και Διανομές:

Βήμα 1: Εκκινούμε τη διαδικασία τοποθετώντας σε έναν νέο πίνακα **from1(k1)** ως πρώτο στοιχείο (k1 =1) την κεντρική αποθήκη και έπειτα βρίσκουμε το μικρότερο κόστος- απόσταση από τον κόμβο k1 (κεντρική αποθήκη) προς τους υπόλοιπους κόμβους και αποθηκεύονται:

1. η θέση του πίνακα **d** σε μια μεταβλητή **m**.
2. το κόστος σε έναν πίνακα **cost** (μονοδιάστατο) υπολογισμού του κόστους της συνολικής διαδρομής.

Βήμα 2: Το **to(m)** μας υποδεικνύει ποιος θα είναι ο επόμενος κόμβος που θα μπει στην διαδρομή μας. Επομένως αποθηκεύεται η τιμή του σε έναν νέο πίνακα **to1(k1)**, αυξάνεται το **k1** κατά ένα και επαναλαμβάνουμε το βήμα 1 τοποθετώντας αρχικά στον πίνακα **from1(k1)** την τιμή που είναι αποθηκευμένη στον πίνακα **to(m)**. Σε αυτό το βήμα γίνεται και η αύξηση των στοιχείων του πίνακα **e** κατά 1, που έχουν ως στοιχείο τον κόμβο που εισήχθη στην καινούρια διαδρομή.

Βήμα 3: Αρχικά τοποθετούμε στον πίνακα **from2(k)** ως πρώτο στοιχείο (k =1) την κεντρική αποθήκη και έπειτα τοποθετούμε στο **to2(k)** τον κόμβο που βρίσκεται στην ίδια θέση του πίνακα **to1** αφού πρώτα ελεγχθούν:

1. η χωρητικότητα του οχήματος, με την προσθήκη των εμπορευμάτων από τον κόμβο που είναι προς εισαγωγή, να είναι μικρότερη από τη μέγιστη χωρητικότητα του. Σε αντίθετη περίπτωση το συγκεκριμένο όχημα (**to2(k)** =1) επιστρέφει στην αποθήκη και ένα νέο, άδειο όχημα συνεχίζει τη διαδρομή .
2. η χωρητικότητα των προς παράδοση εμπορευμάτων του οχήματος να είναι μεγαλύτερη από τις απαιτήσεις του πελάτη που θα εισαχθεί στην διαδρομή. Σε αντίθετη περίπτωση το συγκεκριμένο όχημα (**to2(k)** =1) επιστρέφει στην αποθήκη και ένα νέο, άδειο όχημα συνεχίζει τη διαδρομή .

3. Αν το σύνολο των εμπορευμάτων που έχουν συλλεχθεί μέχρι τον πελάτη που εξετάζουμε συν τα προϊόντα που θα παραληφθούν από τον συγκεκριμένο προστιθέμενα στο σύνολο των εμπορευμάτων που δεν έχουν ακόμη παραδοθεί μείον τα εμπορεύματα που θα παραδοθούν σε αυτόν, είναι λιγότερα από τη μέγιστη χωρητικότητα του οχήματος. Σε αντίθετη περίπτωση το συγκεκριμένο όχημα (**to2(k)** =1) επιστρέφει στην αποθήκη και ένα νέο, άδειο όχημα συνεχίζει τη διαδρομή .

Βήμα 4: Επανάληψη του προηγούμενου βήματος αυξάνοντας το k κατά ένα μέχρι όλοι οι πελάτες να έχουν εξυπηρετηθεί.

Παρακάτω ακολουθεί ένα παράδειγμα δέκα κόμβων. Θεωρούμε ότι οι πίνακες **from1** και **to1** είναι οι ακόλουθοι:

1
3
5
4
7
9
2
8
6
10

from1

3
5
4
7
9
2
8
6
10
1

to1

Οι πίνακες **cap1** και **cap2** που ακολουθούν υποδηλώνουν τις διανομές και παραλαβές που αντιστοιχούν στις απαιτήσεις κάθε πελάτη- κόμβου:

0
11
18
17
22
10
10
11
20
15

cap1

0
6
6
4
1
9
8
12
19
8

cap2

Έστω ότι η χωρητικότητα του οχήματός είναι 50. Στους κόμβους 4 και 8 παρατηρείται παραβίαση των ελέγχων χωρητικότητας του οχήματος επειδή στην πρώτη περίπτωση το όχημα δεν μπορεί να καλύψει τη ζήτηση του πελάτη 7 που ακολουθεί, ενώ στη δεύτερη περίπτωση παραβιάζεται η συνολική χωρητικότητα του οχήματος. Τα αποτελέσματα των διαδρομών φαίνονται στους πίνακες που ακολουθούν:

1
3
5
4
1
7
9
2
8
1
6
10

from2

3
5
4
1
7
9
2
8
1
6
10
1

to2

Από τα αποτελέσματα που εμφανίζονται στους πίνακες **from2** και **to2** παρατηρούμε ότι για την επίλυση του συγκεκριμένου οχήματος χρειαζόμαστε τρία οχήματα που θα εκτελούν τις εξής διαδρομές:

Όχημα 1: 1→ 3→ 5→ 4→1

Όχημα 2: 1→ 7→ 9→ 2→ 8→ 1

Όχημα 3: 1→ 6→ 10→ 1

4.6 Μετατροπή του προγράμματος επίλυσης του προβλήματος Δρομολόγησης Οχημάτων με παραλαβές και διανομές σε πρόγραμμα επίλυσης του προβλήματος Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και με Παραλαβές και Διανομές.

Το πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και Παραλαβές και διανομές είναι η σύνθεση των δύο προβλημάτων:

1. του προβλήματος Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και
2. του προβλήματος Δρομολόγησης Οχημάτων με Παραλαβές και Διανομές.

Στο συγκεκριμένο πρόβλημα έχουμε ένα σύνολο από πελάτες N={1,…,n} που πρέπει να εξυπηρετηθούν από έναν δεδομένο αριθμό οχημάτων K που εκκινούν και πάλι από την κεντρική αποθήκη της εταιρίας. Κάθε πελάτης χαρακτηρίζεται από την γεωγραφική του θέση, τον αριθμό προϊόντων παραλαβής και διανομής d_i και p_i καθώς και από τα χρονικά παράθυρα (α_i, β_i) στα οποία πρέπει να εξυπηρετηθεί. Ένα όχημα μπορεί να φτάσει σε έναν πελάτη νωρίτερα από τον νωρίτερο χρόνο εξυπηρέτησης του και να παραμείνει εκεί χωρίς επιπλέον κόστος μέχρι να τον εξυπηρετήσει. Ωστόσο σε καμία περίπτωση δεν επιτρέπεται η εξυπηρέτηση κανενός πελάτη μετά το πέρας του βραδύτερου χρόνου εξυπηρέτησης του.

Για τη μετατροπή του προγράμματος επίλυσης Δρομολόγησης Οχημάτων με Παραλαβές και Διανομές σε πρόγραμμα επίλυσης Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και με Παραλαβές και Διανομές χρησιμοποιήσαμε τρεις ακόμη πίνακες, τους **dtime, rtime** και **stime**. Και τα τρία διανύσματα είναι μονοδιάστατα μεγέθους όσοι είναι οι κόμβοι του προβλήματος και χρησιμοποιούνται για την αποθήκευση των καινούριων δεδομένων των κόμβων που αφορούν τον νωρίτερο χρόνο εξυπηρέτησης κάθε πελάτη (**rtime)**, τον βραδύτερο χρόνο εξυπηρέτησης (**dtime**) και το χρόνο εξυπηρέτησης κάθε πελάτη (**stime**). Οι πίνακες **from2** και ο **to2** χρησιμοποιούνται, όπως και προηγουμένως, για την αποθήκευση των κόμβων με βάση τις διαδρομές που εκτελούν τα οχήματα. Στους πίνακες αυτούς παρατηρείται και πάλι το φαινόμενο ο κόμβος 1, που αντιστοιχεί στην κεντρική αποθήκη, να εμφανίζεται παραπάνω από μια φορές, κάτι που οφείλεται στο γεγονός ότι υπάρχουν

παραπάνω από ένας κύκλοι. Αντίστοιχα ο πίνακας **cost2** χρησιμοποιείται για την αποθήκευση του συνολικού κόστους των διαδρομών που υπολογίζονται κάθε φορά που τοποθετείται ένας νέος κόμβος στη διαδρομή. Στη συνέχεια παρουσιάζεται ο αλγόριθμος επίλυσης του προβλήματος Δρομολόγησης Οχημάτων με Παραλαβές και Διανομές:

Βήμα 1: Εκκινούμε τη διαδικασία τοποθετώντας σε έναν νέο πίνακα **from1(k1)** ως πρώτο στοιχείο (k1 =1) την κεντρική αποθήκη και έπειτα βρίσκουμε το μικρότερο κόστος- απόσταση από τον κόμβο k1 (κεντρική αποθήκη) προς τους υπόλοιπους κόμβους και αποθηκεύονται:

3. η θέση του πίνακα **d** σε μια μεταβλητή **m**.
4. το κόστος σε έναν πίνακα **cost** (μονοδιάστατο) υπολογισμού του κόστους της συνολικής διαδρομής.

Βήμα 2: Το **to(m)** μας υποδεικνύει ποιος θα είναι ο επόμενος κόμβος που θα μπει στην διαδρομή μας. Επομένως αποθηκεύεται η τιμή του σε έναν νέο πίνακα **to1(k1)**, αυξάνεται το **k1** κατά ένα και επαναλαμβάνουμε το βήμα 1 τοποθετώντας αρχικά στον πίνακα **from1(k1)** την τιμή που είναι αποθηκευμένη στον πίνακα **to(m)**. Σε αυτό το βήμα γίνεται και η αύξηση των στοιχείων του πίνακα **e** κατά 1, που έχουν ως στοιχείο τον κόμβο που εισήχθη στην καινούρια διαδρομή.

Βήμα 3: Αρχικά τοποθετούμε στον πίνακα **from2(k)** ως πρώτο στοιχείο (k =1) την κεντρική αποθήκη και έπειτα τοποθετούμε στο **to2(k)** τον κόμβο που βρίσκεται στην ίδια θέση του πίνακα **to1** αφού πρώτα ελεγχθούν:

1. η χωρητικότητα του οχήματος, με την προσθήκη των εμπορευμάτων από τον κόμβο που είναι προς εισαγωγή, να είναι μικρότερη από τη μέγιστη χωρητικότητα του. Σε αντίθετη περίπτωση το συγκεκριμένο όχημα (**to2(k)** =1) επιστρέφει στην αποθήκη και ένα νέο, άδειο όχημα συνεχίζει τη διαδρομή .
2. η χωρητικότητα των προς παράδοση εμπορευμάτων του οχήματος να είναι μεγαλύτερη από τις απαιτήσεις του πελάτη που θα εισαχθεί στην διαδρομή. Σε αντίθετη περίπτωση το συγκεκριμένο όχημα (**to2(k)** =1) επιστρέφει στην αποθήκη και ένα νέο, άδειο όχημα συνεχίζει τη διαδρομή .
3. Αν το σύνολο των εμπορευμάτων που έχουν συλλεχθεί μέχρι τον πελάτη που εξετάζουμε συν τα προϊόντα που θα παραληφθούν από τον συγκεκριμένο

προστιθέμενα στο σύνολο των εμπορευμάτων που δεν έχουν ακόμη παραδοθεί μείον τα εμπορεύματα που θα παραδοθούν σε αυτόν, είναι λιγότερα από τη μέγιστη χωρητικότητα του οχήματος. Σε αντίθετη περίπτωση το συγκεκριμένο όχημα (**to2(k)** =1) επιστρέφει στην αποθήκη και ένα νέο, άδειο όχημα συνεχίζει τη διαδρομή .

4. Αν το όχημα επισκέπτεται έναν κόμβο νωρίτερα από τον νωρίτερο χρόνο εξυπηρέτησης του οπότε και το όχημα παραμένει εκεί μέχρι ο χρόνος διαδρομής του να γίνει ίσος με τον νωρίτερο χρόνο του κόμβου. Σε αντίθετη περίπτωση το όχημα εξυπηρετεί χωρίς καθυστέρηση τον πελάτη.
5. Αν το όχημα επισκέπτεται έναν κόμβο νωρίτερα από τον βραδύτερο χρόνο εξυπηρέτησης του οπότε και το εξυπηρετεί. Σε αντίθετη περίπτωση το συγκεκριμένο όχημα (**to2(k)** =1) επιστρέφει στην αποθήκη και ένα νέο, άδειο όχημα συνεχίζει τη διαδρομή .

Βήμα 4: Επανάληψη του προηγούμενου βήματος αυξάνοντας το k κατά ένα μέχρι όλοι οι πελάτες να έχουν εξυπηρετηθεί.

Παρακάτω ακολουθεί ένα παράδειγμα δέκα κόμβων. Θεωρούμε ότι οι πίνακες **from1** και **to1** είναι οι ακόλουθοι:

1
3
5
4
7
9
2
8
6
10

from1

3
5
4
7
9
2
8
6
10
1

to1

Αντίστοιχα οι πίνακες Οι πίνακες **cap1** και **cap2** που ακολουθούν υποδηλώνουν τις διανομές και παραλαβές που αντιστοιχούν στις απαιτήσεις κάθε πελάτη- κόμβου:

0
11
18
17
22
10
10
11
20
15

cap1

0
6
6
4
1
9
8
12
19
8

cap2

Ακολουθούν οι πίνακες **dtime, rtime** και **stime** για κάθε κόμβο ξεχωριστά, που βάση αυτών θα γίνει ο έλεγχος αν χρονικά μπορεί να εξυπηρετηθεί κάθε πελάτης.

0
1
34
28
35
60
10
11
20
15

dtime

0
30
48
87
94
112
24
50
77
99

rtime

0
6
6
6
6
6
6
6
6
6

stime

Παρατηρούμε πως στον κόμβο 2 το όχημα αντιμετωπίζει πρόβλημα χρονικού περιορισμού, δηλαδή φθάνει στον πελάτη αργότερα από τον βραδύτερο χρόνο εξυπηρέτησης του. Επομένως το όχημα επιστρέφει στην κεντρική αποθήκη και η εξυπηρέτηση του συγκεκριμένου πελάτη θα γίνει από άλλο όχημα. Παρακάτω ακολουθούν τα αποτελέσματα των διαδρομών υπό μορφή πινάκων:

1
3
5
4
1
7
9
1
2
8
1
6
10

from2

3
5
4
1
7
9
1
2
8
1
6
10
1

to2

Από τα αποτελέσματα που εμφανίζονται στους πίνακες **from2** και **to2** παρατηρούμε ότι για την επίλυση του συγκεκριμένου οχήματος χρειαζόμαστε τέσσερα οχήματα που θα εκτελούν τις εξής διαδρομές:

Όχημα 1: 1→ 3→ 5→ 4→1

Όχημα 2: 1→ 7→ 9→ 1

Όχημα 3: 1→ 2→ 8→ 1

Όχημα 4: 1→ 6→ 10→ 1

4.7 Βελτίωση της λύσης που βρέθηκε στο προηγούμενο βήμα με την χρήση αλγορίθμου τοπικής αναζήτησης 2- opt.

Οι αλγόριθμος 2-opt, αλλά και ο 3-opt που θα περιγραφεί στη επόμενη παράγραφο, ως αλγόριθμοι τοπικής αναζήτησης λειτουργούν πάνω σε μία ήδη υπάρχουσα λύση, δηλαδή πάνω σε ένα ήδη υπάρχον δρομολόγιο και λειτουργούν σταδιακά βελτιώνοντας τη λύση. Έτσι μειώνεται το μήκος του δρομολογίου μέχρι να μην υπάρχουν πλέον αλλαγές που να οδηγούν σε περαιτέρω βελτίωση.

Ο αλγόριθμος 2-opt επιτυγχάνει την ακόλουθη κίνηση:

Διαγράφει 2 ακμές σπάζοντας το δρομολόγιο σε 2 και στη συνέχεια ενώνει τα 2 κομμάτια αυτά με διαφορετικό τρόπο, ώστε να έχουμε μείωση του συνολικού κόστους.

Η παραπάνω αλλαγή γίνεται μόνο στην περίπτωση που έχουμε μείωση του κόστους, αλλιώς σε αντίθετη περίπτωση δεν πραγματοποιείται καμία αλλαγή. Ο έλεγχος γίνεται μεταξύ τεσσάρων κόμβων κάθε φορά και όταν διαπιστώνεται μείωση κόστους, εναλλάσσουμε τους υπό εξέταση κόμβους με τη μορφή που παρουσιάζεται στο παρακάτω παράδειγμα:

Στους πίνακες **from2** και **to2** παρουσιάζονται οι διαδρομές των οχημάτων:

1
3
5
4
7
9
1
2
8
6
10

from2

3
5
4
7
9
1
2
8
6
10
1

to2

Όχημα 1: 1→ 3→ 5→ 4→7→ 9→1

Όχημα 2: 1→ 2→ 8→ 6→ 10→1

Για κάθε διαδρομή που περιέχει τουλάχιστον τέσσερις διαφορετικούς κόμβους (για το όχημα 1) γίνεται η ακόλουθη διαδικασία:

Υπολογισμός :

1. του αθροίσματος των κοστών των μεταβάσεων από τον κόμβο1 στον κόμβο 3 και από τον κόμβο 5 στον κόμβο 4 και αποθήκευση του στην μεταβλητή α.
2. του αθροίσματος των κοστών των μεταβάσεων από τον κόμβο1 στον κόμβο 5 και από τον κόμβο 3 στον κόμβο 4 και αποθήκευση του στην μεταβλητή β.

Έλεγχος :

Αν **(α < β)** τότε αλλάζει η διαδρομή μας αφού με εναλλαγή των κόμβων 3 και 5 πετυχαίνουμε μείωση του κόστους και η νέα διαδρομή θα είναι:

Όχημα 1: 1→ 5→ 3→ 4→7→ 9→1

Αν όμως ίσχυε **(α > β)** τότε στη θέση της μεταβλητής β θα αποθηκεύαμε το κόστος των κόμβων 4 και 7 και θα επαναλαμβάναμε τον έλεγχο. Ο συγκεκριμένος έλεγχος γίνεται μέχρι να ελεγχθούν όλοι οι κόμβοι που ανήκουν στην ίδια διαδρομή.

Παράλληλα για να διατηρηθεί ο προσανατολισμός του δρομολογίου θα πρέπει να αντιστραφούν οι πόλεις που βρίσκονται μεταξύ των κόμβων ανάμεσα στους οποίους γίνεται η εναλλαγή. Αν, δηλαδή, στη παρούσα διαδρομή γινόταν εναλλαγή των κόμβων 3 και 9, η καινούρια διαδρομή θα ήταν η εξής:

Όχημα 1: 1→ 9→ 7→ 4→5→ 3→1

4.8 Βελτίωση της λύσης που βρέθηκε στο προηγούμενο βήμα με την χρήση αλγορίθμου τοπικής αναζήτησης 3- opt.

Μία βελτίωση του αλγορίθμου 2-opt αποτελεί ο 3-opt. Με τον συγκεκριμένο αλγόριθμο η διαδρομή διακόπτεται σε τρία κομμάτια για μεγαλύτερη ευελιξία στην τροποποίηση του δρομολογίου και ακολουθεί την ίδια λογική με τον αλγόριθμο 2-opt. Ο αλγόριθμος 3-opt είναι πολύ πιο αποτελεσματικός από τον 2 –opt και προσεγγίζει καλύτερα τη βέλτιστη λύση αρκεί σε κάθε διαδρομή να υπάρχουν τουλάχιστον έξι διαφορετικοί κόμβοι.

Ο αλγόριθμος 3-opt μπορεί να πραγματοποιηθεί με 8 συνδυασμούς 3 ζευγαριών. Στη συγκεκριμένη διπλωματική εργασία χρησιμοποιήθηκαν οι τέσσερις εξ' αυτών. Στη συνέχεια παρουσιάζεται ο τρόπος λειτουργίας αυτών των συνδυασμών:

Έστω ότι έχουμε 10 πόλεις, οι οποίες ενώνονται με τη σειρά. Όπως παρουσιάζεται στους επόμενους πίνακες:

1
2
3
4
5
6
7
8
9
10

from2

2
3
4
5
6
7
8
9
10
1

to2

Ο πρώτος συνδυασμός τριών ζευγαριών του αλγορίθμου 3- opt λειτουργεί ως εξής:

Υπολογισμός :

1. του αθροίσματος των κοστών των μεταβάσεων από τον κόμβο 2 στον κόμβο 3, από τον κόμβο 5 στον 6 και από τον κόμβο 9 στον 10,και αποθήκευση του στην μεταβλητή α.
2. του αθροίσματος των κοστών των μεταβάσεων από τον κόμβο 2 στον κόμβο 5, από τον κόμβο 3 στον 9 και από τον κόμβο 6 στον 10, και αποθήκευση του στην μεταβλητή β.

Έλεγχος :

Αν **(α < β)** τότε αλλάζει η διαδρομή μας αφού με εναλλαγή των κόμβων 3, 5, 6 και 9 πετυχαίνουμε μείωση του κόστους και η νέα διαδρομή θα είναι:

1
2
5
4
3
9
8
7
6
10

from2

2
5
4
3
9
8
7
6
10
1

to2

Παράλληλα για να διατηρηθεί ο προσανατολισμός του δρομολογίου θα πρέπει να αντιστραφούν οι πόλεις που βρίσκονται μεταξύ των κόμβων ανάμεσα στους οποίους γίνεται η εναλλαγή όπως έγινε με τους κόμβους 7 και 8.

Ο δεύτερος συνδυασμός τριών ζευγαριών του αλγορίθμου 3- opt λειτουργεί ως εξής:

Υπολογισμός :

1. του αθροίσματος των κοστών των μεταβάσεων από τον κόμβο 2 στον κόμβο 3, από τον κόμβο 5 στον 6 και από τον κόμβο 9 στον 10,και αποθήκευση του στην μεταβλητή α.
2. του αθροίσματος των κοστών των μεταβάσεων από τον κόμβο 2 στον κόμβο 5, από τον κόμβο 3 στον 9 και από τον κόμβο 6 στον 10, και αποθήκευση του στην μεταβλητή β.

Έλεγχος :

Αν **(α < β)** τότε αλλάζει η διαδρομή μας κάνοντας τις εναλλαγές κόμβων που παρουσιάζονται στους παρακάτω πίνακες:

1
2
7
8
9
6
5
4
3
10

from2

2
7
8
9
6
5
4
3
10
1

to2

Ο τρίτος συνδυασμός τριών ζευγαριών του αλγορίθμου 3- opt λειτουργεί ως εξής:

Υπολογισμός :

1. του αθροίσματος των κοστών των μεταβάσεων από τον κόμβο 2 στον κόμβο 3, από τον κόμβο 5 στον 6 και από τον κόμβο 9 στον 10,και αποθήκευση του στην μεταβλητή α.
2. του αθροίσματος των κοστών των μεταβάσεων από τον κόμβο 2 στον κόμβο 5, από τον κόμβο 3 στον 9 και από τον κόμβο 6 στον 10, και αποθήκευση του στην μεταβλητή β.

Έλεγχος :

Αν **(α < β)** τότε αλλάζει η διαδρομή μας κάνοντας τις εναλλαγές κόμβων που παρουσιάζονται στους παρακάτω πίνακες:

1
2
7
8
9
3
4
5
6
10

from2

2
7
8
9
3
4
5
6
10
1

to2

Τέλος, ο τέταρτος συνδυασμός τριών ζευγαριών του αλγορίθμου 3- opt λειτουργεί ως εξής:

Υπολογισμός :

1. του αθροίσματος των κοστών των μεταβάσεων από τον κόμβο 2 στον κόμβο 3, από τον κόμβο 5 στον 6 και από τον κόμβο 9 στον 10,και αποθήκευση του στην μεταβλητή α.
2. του αθροίσματος των κοστών των μεταβάσεων από τον κόμβο 2 στον κόμβο 5, από τον κόμβο 3 στον 9 και από τον κόμβο 6 στον 10, και αποθήκευση του στην μεταβλητή β.

Έλεγχος :

Αν **(α < β)** τότε αλλάζει η διαδρομή μας κάνοντας τις εναλλαγές κόμβων που παρουσιάζονται στους παρακάτω πίνακες:

1
2
9
8
7
3
4
5
6
10

from2

2
9
8
7
3
4
5
6
10
1

to2

Κεφάλαιο 5: Εφαρμογές

5.1 Εισαγωγή

Το πρόγραμμα επίλυσης του προβλήματος Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και με Παραλαβές και Διανομές εφαρμόστηκε αρχικά για ένα πραγματικό πρόβλημα στην εταιρία ταχυμεταφορών **ACS** (παράγραφος 5.4) και στη συνέχεια, για να φανεί η αποτελεσματικότητα του αλγορίθμου, εφαρμόστηκε σε κάποια προβλήματα των οποίων τα δεδομένα υπήρχαν στο διαδίκτυο (παράγραφος 5.5). Στις επόμενες δύο παραγράφους παρουσιάζονται οι μεταβλητές και οι περιορισμοί του προγράμματος και ακολουθούν οι εφαρμογές και τα αποτελέσματά τους.

5.2 Μεταβλητές του προγράμματος

Ορίζουμε τις μεταβλητές του Μαθηματικού Μοντέλου που χρησιμοποιείται για την επίλυση του Προβλήματος Δρομολόγησης Οχημάτων της ACS.

- C_{ij} = η απόσταση από τον πελάτη i στον πελάτη j η οποία είναι ίση και με την απόσταση του πελάτη j από τον πελάτη i
- D_i^v: ο αριθμός των διανομών που έχουν γίνει από το όχημα v μόλις φεύγει από τον πελάτη i.
- P_i^v: ο αριθμός των παραλαβών που έχουν γίνει από το όχημα v μόλις φεύγει από τον πελάτη i.
- T_i^v: ο χρόνος στον οποίο το όχημα v φτάνει στον κόμβο i.

- K = ο αριθμός των οχημάτων
- M_v = η χωρητικότητα των οχημάτων
- x_{ij} = δυαδική μεταβλητή για την οποία ισχύει

$$x_{ij}^{v} = \begin{cases} 1, \text{ το όχημα } \nu \text{ χρησιμοποιεί το τόξο (i,j)} \\ 0, \text{ διαφορετικά} \end{cases}$$

5.3 Περιορισμοί του προγράμματος- αντικειμενική συνάρτηση

Περιορισμοί:

$$\sum_{v \in E} \sum_{j \in V} x_{ijv} = 1 \qquad \text{για κάθε i} \in \text{N} \qquad (1)$$

$$\sum_{i \in V} x_{ip}^{v} = \sum_{j \in V} x_{pj}^{v} \qquad \text{για κάθε p} \in \text{N, } \nu \in \text{K} \qquad (2)$$

$$\sum_{j \in v} x_{0j}^{v} \leq 1 \qquad \text{για κάθε } \nu \in \text{K} \qquad (3)$$

$$\sum_{i \in N} x_{i,n+1}^{v} = \sum_{j \in N} x_{o,j}^{v} \qquad \text{για κάθε } \nu \in \text{K} \qquad (4)$$

$$D_{i}^{v} + P_{i}^{v} \leq Q \qquad \text{για κάθε i} \in \text{V, } \nu \in \text{K} \qquad (5)$$

$$D_{n+1}^{v} = 0 \qquad \text{για κάθε } \nu \in \text{K} \qquad (6)$$

$$D_{1}^{k} = \sum_{i \in N} \sum_{j \in N} x_{ij}^{k} d_i \qquad \text{για κάθε } \nu \in \text{K} \qquad (7)$$

$$P_{n+1}^{v} = \sum_{i \in N} \sum_{j \in N} x_{ij}^{v} p_i \qquad \text{για κάθε } \nu \in \text{K} \qquad (8)$$

$P_1^v = 0$ για κάθε $v \in K$ (9)

$x_{ij}^v (P_i^v + p_i - P_j^v) = 0$ για κάθε $i, j \in V, v \in K$ (10)

$x_{ij}^v (D_i^v - d_i - D_j^v) = 0$ για κάθε $i, j \in V, v \in K$ (11)

$x_{ij}^v (T_i^v + t_i - T_j^v) = 0$ για κάθε $i, j \in V, v \in K$ (12)

$a_i \leq T_i \leq b_i$ για κάθε $i \in V, v \in K$ (13)

$D_i^v \geq 0$ για κάθε $i \in V, v \in K$ (14)

$P_i^v \geq 0$ για κάθε $i \in V, v \in K$ (15)

Αντικειμενική συνάρτηση:

$$C^* = \min \sum_{i,j} c_{ij} \sum_{v} x_{ijv}$$

Στόχος της αντικειμενικής συνάρτησης είναι η ελαχιστοποίηση του συνολικού κόστους της εκάστοτε διαδρομής. Ο περιορισμός (1) ελέγχει το ότι κάθε πελάτης εξυπηρετείται από ένα μόνο όχημα ενώ ο (2) δεσμεύει ότι το όχημα που πηγαίνει σε ένα κόμβο είναι το ίδιο με αυτό που εξέρχεται από τον ίδιο κόμβο. Οι περιορισμοί (3), (4) δεσμεύουν κάθε όχημα να χρησιμοποιείται μόνο μια φορά. Με τον περιορισμό (5) ελέγχουμε αν το φορτίο του οχήματος, όταν εξέρχεται από κάθε κόμβο, είναι μικρότερο της χωρητικότητας του φορτηγού. Οι περιορισμοί (7) και (9) δεσμεύουν κάθε όχημα να εκκινεί από την κεντρική αποθήκη γεμάτο με προϊόντα προς διάθεση ενώ το φορτίο παραλαβής είναι ίσο με μηδέν. Οι περιορισμοί (6) και (8) εγγυώνται πως όταν τα οχήματα επιστρέφουν στην κεντρική αποθήκη, έχουν διανείμει και αντίστοιχα παραλάβει όλα τα προϊόντα από τους κόμβους που πέρασαν. Οι περιορισμοί- ισότητες (10) και (11)δηλώνουν πως αν ένα όχημα πάει σε έναν κόμβο, τότε η ποσότητα διανομής μειώνεται κατά ποσό ίσο με το ποσό που διένειμε

στον συγκεκριμένο κόμβο και η ποσότητα παραλαβής αυξάνεται αντίστοιχα. Ο περιορισμός (12) ελέγχει το χρόνο που εξυπηρετεί ένα όχημα κάποιον κόμβο ώστε ο χρόνος εξυπηρέτησης να είναι τουλάχιστον ίσος με τον νωρίτερο χρόνο του κόμβου. Ο περιορισμός (13), τέλος, δηλώνει τα χρονικά διαστήματα για κάθε κόμβο.

5.4 Πραγματική εφαρμογή του προγράμματος επίλυσης του προβλήματος Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και με Παραλαβές και Διανομές.

Το πρόγραμμα επίλυσης του προβλήματος Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και με Παραλαβές και Διανομές εφαρμόστηκε για το υποκατάστημα της εταιρίας ταχυμεταφορών **ACS** των Χανίων. Τα δεδομένα που χρησιμοποιήθηκαν στην πρακτική εφαρμογή είναι του μήνα Νοέμβρη του 2002, ενός μήνα αντιπροσωπευτικού για την εταιρεία από πλευράς ζητήσεων των πελατών. Η κεντρική αποθήκη της εταιρίας **ACS** βρίσκεται στην Μ. Μπότσαρη.

Για μεγαλύτερη διευκόλυνση δεν υπολογίζονται οι αποστάσεις από τους πελάτες αλλά από τα κέντρα των οδών τους, υποθέτοντας ότι ο διανομέας- παραλήπτης φτάνει στο κέντρο της οδού και από εκεί εκτελεί τις αποστολές του. Οι κόμβοι, λοιπόν, είναι τα κέντρα των οδών και στην συγκεκριμένη περιοχή υπάρχουν 48 κόμβοι, οι οποίοι και δίνονται στον **Πίνακα 5.1.** Τον μήνα όμως Νοέμβρη υπήρξε ζήτηση από πελάτες των 27 μονάχα κόμβων. Οι 27 αυτοί κόμβοι είναι οι χρωματισμένοι οδοί του Πίνακα 5.1:

Αν. Γογονή	Αν. Μάντακα	Κριαράδων	Πολυρρηνίας
Ν. Μαριακάκη	Μ. Μπότσαρη	Γεωργιλάδων	Νικολακάκη Επισκ.
Π. Κορνάρου	Γρηγορίου Ε	Δικτύνης	Ξυρουχάκη
Θρ. Καλαφάτη	Υψηλαντών	Φαλασάρνης	Μάλεμε
Μ. Κατράκη	Κυδωνίας	Απτέρων	Ηρακλείου
Ουνέσκο	Πλ. Κοτσάμπαση	Σαμώνα	Νεάρχου
Παπαδάκη	Καραϊσκάκη	Βολάνης	Σολωμού
Μυλωνογιάννη	Περίδου	Ελύρου	Ερωφίλης
Εφ. Πολεμιστών	Αποκορώνου	Λυσσού	Ι. Κορνάρου
Ν. Σκουλά	Σφακίων	Μαλαθίρου	Μπονιαλή
Βολουδάκιδων	Μαρκορά	Πειραιώς	Πλ. Κολοκοτρώνη
Ν. Πλαστήρα	Εμ. Αντωνιάδου	Αναπαύσεως	Ερωτόκριτου

Πίνακας 5.1: Πίνακας περιοχών εξυπηρέτησης

Χρειάστηκε σημαντικός χρόνος για τον ορισμό των μονάδων χωρητικότητας και την θεωρητική τοποθέτηση των φακέλων και των μικροδεμάτων μέσα στα οχήματα. Τελικά επιλέχθηκε ως τύπος προϊόντων μεταφοράς τα μικροδέματα και ως τύπος οχήματος για την μεταφορά των προϊόντων το μοτοποδήλατο που μεταφέρει σε κάθε διαδρομή περίπου δέκα μικροδέματα (εξαρτάται από το μέγεθος). Ως χρονικά διαστήματα επιλέχθηκαν ενιαία, σταθερά χρονικά παράθυρα λειτουργίας του καταστήματος καθημερινά από τις οχτώ το πρωί έως τις τέσσερις το απόγευμα. Στον πίνακα 5.2 παρουσιάζονται οι παραλαβές και διανομές που κατά μέσο όρο έχει απαίτηση ο κάθε πελάτης. Τέλος, στον πίνακα 5.3 παρουσιάζονται οι αποστάσεις μεταξύ των κόμβων- πελατών:

Περιοχή	Διανομές	Παραλαβές
Μ. Μπότσαρη	0	0
Αποκορώνου	10,88	4,16
Γογωνή	1,64	0,96
Εφ. Πολεμιστών	2,08	1,16
Μπονιαλή	3,56	1,8
Περίδου	4,12	1,84
Πλ. Κολοκοτρώνη	0,24	0,16
Δικτύνης	0,48	0,2
Ν. Πλαστήρα	2,72	1,56
Εμ. Αντωνιάδου	0,28	0,2
Πειραιώς	0,48	0,2
Άπτερων	0,52	0,08
Ερωφίλης	0,44	0,2
Σολωμού	0,84	0,28
Φαλασαρνής	0,48	0,24
Κυδωνίας	1,2	0,68
Ν. Σκουλά	2,2	1,16
Νεάρχου	1,76	1,08
Καραϊσκάκη	4,2	2,76
Ηρακλείου	2	0,96
Υψηλάντων	2,16	1,84
Αναπαύσεως	2,44	1
Νικολακάκη	0,64	0,24
Λισσού	0,52	0,2
Πλ. Κοτσιάμπαση	0,56	0,28
Μυλονογιάννη	2,24	1,04
Ελύρου	0,68	0,4

Πίνακας 5.2: Πίνακας Διανομών- Παραλαβών για κάθε πελάτη

	ΜΠΟΤ	ΑΠΟΚ	ΓΟΓ	ΠΟΛ	ΜΠΟΝ	ΠΕΡ	ΚΟΛΟ	ΔΙΚΤ	ΠΛΑΣ	ΑΝΤ	ΠΕΙΡ	ΑΠΤ	ΕΡΩ	ΣΟΛ	ΦΑΛ	ΚΥΔ	ΣΚΟΥ	ΝΕΑ	ΚΑΡ	ΗΡΑ	ΥΨΗ	ΑΝΑΠ	ΝΙΚΟ	ΛΙΣ	ΚΟΤΣ	ΜΥΛ	ΕΛΥ
ΜΠΟΤ	0	350	740	390	370	210	210	580	330	990	620	680	390	350	680	270	215	565	195	875	140	370	835	855	290	155	700
ΑΠΟΚ	350	0	625	310	230	350	230	370	390	620	855	330	115	210	485	525	290	215	445	545	430	525	700	525	425	370	545
ΓΟΓ	740	625	0	505	840	780	760	290	915	620	1340	485	740	855	115	950	485	720	855	445	835	330	210	290	935	580	365
ΠΟΛ	390	310	505	0	490	430	410	200	560	720	970	390	440	400	350	660	165	370	525	600	505	135	445	445	565	220	350
ΜΠΟΝ	370	230	840	490	0	180	80	580	230	895	700	600	310	270	680	760	350	465	290	795	290	465	855	760	230	270	720
ΠΕΡ	210	350	780	430	180	0	115	640	135	915	600	680	390	370	800	270	290	545	110	875	115	410	875	895	155	410	775
ΚΟΛΟ	210	230	760	410	80	115	0	620	235	855	670	545	270	230	410	390	270	430	305	760	235	410	790	700	245	350	660
ΔΙΚΤ	580	370	290	200	580	640	620	0	680	700	1225	390	370	485	175	835	350	390	700	580	700	330	330	350	770	390	260
ΠΛΑΣ	330	390	915	560	230	135	235	680	0	1050	580	740	430	395	875	250	465	600	155	970	250	560	1010	855	40	505	835
ΑΝΤ	990	620	620	720	895	915	855	700	1050	0	1535	390	680	795	525	1185	855	525	1090	250	1090	855	410	350	1050	1050	410
ΠΕΙΡ	620	855	1340	970	700	600	670	1225	580	1535	0	1185	1070	895	1280	330	835	1245	505	1420	505	990	1440	1500	565	740	1420
ΑΠΤ	680	330	485	390	600	680	545	390	740	390	1185	0	390	545	370	855	545	210	775	290	780	525	270	195	800	620	135
ΕΡΩ	390	115	740	440	310	390	270	370	430	680	1070	390	0	75	875	600	350	180	530	660	560	700	620	640	485	450	505
ΣΟΛ	350	210	855	400	270	370	230	485	395	795	895	545	75	0	660	640	425	270	525	680	475	545	720	660	450	505	700
ΦΑΛ	680	485	115	350	680	800	410	175	875	525	1280	370	875	660	0	970	485	565	840	410	775	310	155	175	875	505	245
ΚΥΔ	270	525	950	660	760	270	390	835	250	1185	330	855	600	640	970	0	465	875	175	1080	175	640	1090	1130	200	440	1050
ΣΚΟΥ	215	290	485	165	350	290	270	350	465	855	835	545	350	425	485	465	0	430	390	740	330	165	600	620	430	75	565
ΝΕΑ	565	215	720	370	465	545	430	390	600	525	1245	210	180	270	565	875	430	0	760	410	750	515	545	390	700	525	340
ΚΑΡ	195	445	855	525	290	110	305	700	155	1090	505	775	530	525	840	175	390	760	0	970	75	525	1010	1050	115	360	915
ΗΡΑ	875	545	445	600	795	875	760	580	970	250	1420	290	660	680	410	1080	740	410	970	0	1000	720	320	235	970	895	310
ΥΨΗ	140	430	835	505	290	115	235	700	250	1090	505	780	560	475	775	175	330	750	75	1000	0	485	935	970	190	290	875
ΑΝΑΠ	370	525	330	135	465	410	410	330	560	855	990	525	700	545	310	640	165	515	525	720	485	0	465	505	575	215	410
ΝΙΚΟ	835	700	210	445	855	875	790	330	1010	410	1440	270	620	720	155	1090	600	545	1010	320	935	465	0	60	1185	720	145
ΛΙΣ	855	525	290	445	760	895	700	350	855	350	1500	195	640	660	175	1130	620	390	1050	235	970	505	60	0	1310	730	125
ΚΟΤΣ	290	425	935	565	230	155	245	770	40	1050	565	800	485	450	875	200	430	700	115	970	190	575	1185	1310	0	580	990
ΜΥΛ	155	370	580	220	270	410	350	390	505	1050	740	620	450	505	505	440	75	525	360	895	290	215	720	730	580	0	600
ΕΛΥ	700	545	365	350	720	775	660	260	835	410	1420	135	505	700	245	1050	565	340	915	310	875	410	145	125	990	600	0

Πίνακας 5.3: Πίνακας αποστάσεων των 27 περιοχών

Αποτελέσματα της εφαρμογής

Τα αποτελέσματα που εξήχθησαν για το υποκατάστημα της εταιρίας ταχυμεταφορών **ACS** των Χανίων εμφανίζονται αναλυτικά στο παράρτημα της εργασίας. Παρακάτω παρουσιάζονται οι διαδρομές προέκυψαν από την επίλυση του προβλήματος Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και με Παραλαβές και Διανομές καθώς και το συνολικό κόστος των διαδρομών:

Όχημα 1

Μ. Μπότσαρη→ Υψηλάντων→ Καραϊσκάκη→ Περίδου→ Μ. Μπότσαρη

Όχημα 2

Μ. Μπότσαρη→ Πλατεία Κολοκοτρώνη→ Μπονιαλή→ Αποκορώνου→ Ερωφίλης→ Σολωμού→Νεάρχου→Άπτερων→ Μ. Μπότσαρη

Όχημα 3

Μ. Μπότσαρη→ Ελύρου→ Λισσού→ Νικολακάκη→ Φαλασσαρνής→ Μ. Μπότσαρη

Όχημα 4

Μ. Μπότσαρη→ Γογωνή→ Μ. Μπότσαρη

Όχημα 5

Μ. Μπότσαρη→ Δικτύνης→ Εφ. Πολεμιστών→ Αναπαύσεως→ Ν. Σκουλά→ Μυλονογιάννη→ Κυδωνίας→ Πλατεία Κοτσιάμπαση→ Ν. Πλαστήρα→ Πειραιώς→ Μ. Μπότσαρη

Όχημα 6

Μ. Μπότσαρη→ Ηρακλείου→ Εμμ. Αντωνιάδου → Μ. Μπότσαρη

Το συνολικό κόστος της διαδρομής είναι 10755 μέτρα. Ακολουθούν οι διαδρομές και το συνολικό κόστος έπειτα από τη βελτίωση των αποτελεσμάτων βάση του αλγορίθμου 2- opt:

Όχημα 1

Μ. Μπότσαρη→ Υψηλάντων→ Καραϊσκάκη→ Περίδου→ Μ. Μπότσαρη

Όχημα 2

Μ. Μπότσαρη→ Πλατεία Κολοκοτρώνη→ Μπονιαλή→ Αποκορώνου→ Άπτερων → Νεάρχου→ Ερωφίλης → Σολωμού → Μ. Μπότσαρη

Όχημα 3

Μ. Μπότσαρη→ Ελύρου→ Λισσού→ Νικολακάκη→ Φαλασσαρνής→ Μ. Μπότσαρη

Όχημα 4

Μ. Μπότσαρη→ Γογωνή→ Μ. Μπότσαρη

Όχημα 5

Μ. Μπότσαρη→ Αναπαύσεως→ Εφ. Πολεμιστών→ Δικτύνης→ Ν. Σκουλά→ Μυλονογιάννη→ Πειραιώς→ Κυδωνίας→ Πλατεία Κοτσιάμπαση→ Ν. Πλαστήρα→ Μ. Μπότσαρη

Όχημα 6

Μ. Μπότσαρη→ Ηρακλείου→ Εμμ. Αντωνιάδου → Μ. Μπότσαρη

Το συνολικό κόστος της διαδρομής είναι 10285 μέτρα, έχουμε δηλαδή μείωση της διαδρομής κατά 470 μέτρα. Πρέπει να σημειωθεί, ωστόσο, πως περαιτέρω μείωση των διαδρομών λόγω του αλγορίθμου 3-opt δεν υπήρξε λόγω του μικρού αριθμού κόμβων ανά κύκλο.

Εκτός από την αναλυτική περιγραφή των αποτελεσμάτων, στο παράρτημα υπάρχουν τα αποτελέσματα δύο ακόμη σεναρίων για διαφορετικές χωρητικότητες του οχήματος:

1. για χωρητικότητα οχήματος ίση με 15 μικροδέματα.
2. για χωρητικότητα οχήματος ίση με 8 μικροδέματα.

5.5 Εφαρμογή του προγράμματος επίλυσης του προβλήματος Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και με Παραλαβές και Διανομές σε πρόβλημα ελέγχου της αποτελεσματικότητας του αλγορίθμου.

Τα δεδομένα του προβλήματος που χρησιμοποιήθηκαν για τον έλεγχο της αποτελεσματικότητας του αλγορίθμου πάρθηκαν από το διαδίκτυο και πιο συγκεκριμένα από τη διεύθυνση:

http://neo.lcc.uma.es/radi-aeb/WebVRP/index.html?/Problem%20Instances/instances.html

Στο πρόβλημα αυτό οι πελάτες- κόμβοι προς εξυπηρέτηση είναι διακόσιοι προσθέτωντας και τον κόμβο 1 που συμβολίζει την κεντρική αποθήκη. Τα χρονικά παράθυρα είναι σταθερά και στη συγκεκριμένη εφαρμογή και οι αποστάσεις υπολογίζονται με τον αλγόριθμο Nearest Neighborhood.

Αποτελέσματα της εφαρμογής

Στον παρακάτω πίνακα παρουσιάζονται:

1. το συνολικό κόστος διαδρομής που προέκυψε από την επίλυση του προβλήματος.
2. τα βελτιωμένα συνολικά κόστη που προέκυψαν από τον 2- opt και 3- opt.

Επίλυση μέσω:	Συνολικό κόστος	Μείωση κόστους
VRPTWPD	2.381,708	
2- opt	2.336,044	45,663
3- opt φάση 1	2.324,662	57,046
3- opt φάση 2	2.302,980	78,727
3- opt φάση 3	2.273,537	108,171
3- opt φάση 4	2.251,432	130,275

Πίνακας 5.4: Πίνακας αποτελεσμάτων εφαρμογής 2

Όπως διαφαίνεται από τον παραπάνω πίνακα η αχική επίλυση του προβλήματος έδωσε ως ελάχιστη διαδρομή 2381,708 μέτρα με χρήση 8 οχημάτων. Η μείωση του κόστους, ωστόσο, που προέκυψε από τους αλγορίθμους 2- opt και 3- opt ήταν 130,275 μέτρα, δηλαδή το συνολικό κόστος βελτιώθηκε κατά 5,5 % περίπου. Τα δεδομένα του προβλήματος αλλά και τα αποτελέσματα του προβλήματος παρουσιάζονται αναλυτικά στο παράρτημα.

Παράρτημα

Αποτελέσματα εφαρμογής 1

<u>Σενάριο 1: Για χωρητικότητα = 10</u>

Τα αποτελέσματα του TSP

i = 1 from2(i) = 1. to2(i) = 21. cost1 = 140.
i = 2 from2(i) = 21. to2(i) = 19. cost1 = 215.
i = 3 from2(i) = 19. to2(i) = 6. cost1 = 325.
i = 4 from2(i) = 6. to2(i) = 7. cost1 = 440.
i = 5 from2(i) = 7. to2(i) = 5. cost1 = 520.
i = 6 from2(i) = 5. to2(i) = 2. cost1 = 750.
i = 7 from2(i) = 2. to2(i) = 13. cost1 = 865.
i = 8 from2(i) = 13. to2(i) = 14. cost1 = 940.
i = 9 from2(i) = 14. to2(i) = 18. cost1 = 1210.
i = 10 from2(i) = 18. to2(i) = 12. cost1 = 1420.
i = 11 from2(i) = 12. to2(i) = 27. cost1 = 1555.
i = 12 from2(i) = 27. to2(i) = 24. cost1 = 1680.
i = 13 from2(i) = 24. to2(i) = 23. cost1 = 1740.
i = 14 from2(i) = 23. to2(i) = 15. cost1 = 1895.
i = 15 from2(i) = 15. to2(i) = 3. cost1 = 2010.
i = 16 from2(i) = 3. to2(i) = 8. cost1 = 2300.
i = 17 from2(i) = 8. to2(i) = 4. cost1 = 2500.
i = 18 from2(i) = 4. to2(i) = 22. cost1 = 2635.
i = 19 from2(i) = 22. to2(i) = 17. cost1 = 2800.
i = 20 from2(i) = 17. to2(i) = 26. cost1 = 2875.
i = 21 from2(i) = 26. to2(i) = 16. cost1 = 3315.
i = 22 from2(i) = 16. to2(i) = 25. cost1 = 3515.
i = 23 from2(i) = 25. to2(i) = 9. cost1 = 3555.
i = 24 from2(i) = 9. to2(i) = 11. cost1 = 4135.
i = 25 from2(i) = 11. to2(i) = 20. cost1 = 5555.
i = 26 from2(i) = 20. to2(i) = 10. cost1 = 5805.
i = 27 from2(i) = 10. to2(i) = 1. cost1 = 6795.

Τα αποτελέσματα του VRP

i = 1 from3(i) = 1. to3(i) = 21. cost2 = 140.
i = 2 from3(i) = 21. to3(i) = 19. cost2 = 215.
i = 3 from3(i) = 19. to3(i) = 6. cost2 = 325.
i = 4 from3(i) = 6. to3(i) = 1. cost2 = 535.
i = 5 from3(i) = 1. to3(i) = 7. cost2 = 745.

i = 6 from3(i) = 7. to3(i) = 5. cost2 = 825.
i = 7 from3(i) = 5. to3(i) = 2. cost2 = 1055.
i = 8 from3(i) = 2. to3(i) = 13. cost2 = 1170.
i = 9 from3(i) = 13. to3(i) = 14. cost2 = 1245.
i = 10 from3(i) = 14. to3(i) = 18. cost2 = 1515.
i = 11 from3(i) = 18. to3(i) = 12. cost2 = 1725.
i = 12 from3(i) = 12. to3(i) = 1. cost2 = 2405.
i = 13 from3(i) = 1. to3(i) = 27. cost2 = 3105.
i = 14 from3(i) = 27. to3(i) = 24. cost2 = 3230.
i = 15 from3(i) = 24. to3(i) = 23. cost2 = 3290.
i = 16 from3(i) = 23. to3(i) = 15. cost2 = 3445.
i = 17 from3(i) = 15. to3(i) = 1. cost2 = 4125.
i = 18 from3(i) = 1. to3(i) = 3. cost2 = 4865.
i = 19 from3(i) = 3. to3(i) = 1. cost2 = 5605.
i = 20 from3(i) = 1. to3(i) = 8. cost2 = 6185.
i = 21 from3(i) = 8. to3(i) = 4. cost2 = 6385.
i = 22 from3(i) = 4. to3(i) = 22. cost2 = 6520.
i = 23 from3(i) = 22. to3(i) = 17. cost2 = 6685.
i = 24 from3(i) = 17. to3(i) = 26. cost2 = 6760.
i = 25 from3(i) = 26. to3(i) = 16. cost2 = 7200.
i = 26 from3(i) = 16. to3(i) = 25. cost2 = 7400.
i = 27 from3(i) = 25. to3(i) = 9. cost2 = 7440.
i = 28 from3(i) = 9. to3(i) = 11. cost2 = 8020.
i = 29 from3(i) = 11. to3(i) = 1. cost2 = 8640.
i = 30 from3(i) = 1. to3(i) = 20. cost2 = 9515.
i = 31 from3(i) = 20. to3(i) = 10. cost2 = 9765.
i = 32 from3(i) = 10. to3(i) = 1. cost2 = 10755.

Τα αποτελέσματα του 2-opt

i = 1 from3(i) = 1. cost1 = 140.
i = 2 from3(i) = 21. cost1 = 215.
i = 3 from3(i) = 19. cost1 = 325.
i = 4 from3(i) = 6. cost1 = 535.
i = 5 from3(i) = 1. cost1 = 745.
i = 6 from3(i) = 7. cost1 = 825.
i = 7 from3(i) = 5. cost1 = 1055.
i = 8 from3(i) = 2. cost1 = 1385.
i = 9 from3(i) = 12. cost1 = 1595.
i = 10 from3(i) = 18. cost1 = 1775.
i = 11 from3(i) = 13. cost1 = 1850.
i = 12 from3(i) = 14. cost1 = 2200.
i = 13 from3(i) = 1. cost1 = 2900.
i = 14 from3(i) = 27. cost1 = 3025.
i = 15 from3(i) = 24. cost1 = 3085.
i = 16 from3(i) = 23. cost1 = 3240.
i = 17 from3(i) = 15. cost1 = 3920.
i = 18 from3(i) = 1. cost1 = 4660.

i = 19 from3(i) = 3. cost1 = 5400.
i = 20 from3(i) = 1. cost1 = 5770.
i = 21 from3(i) = 22. cost1 = 5905.
i = 22 from3(i) = 4. cost1 = 6105.
i = 23 from3(i) = 8. cost1 = 6455.
i = 24 from3(i) = 17. cost1 = 6530.
i = 25 from3(i) = 26. cost1 = 7270.
i = 26 from3(i) = 11. cost1 = 7600.
i = 27 from3(i) = 16. cost1 = 7800.
i = 28 from3(i) = 25. cost1 = 7840.
i = 29 from3(i) = 9. cost1 = 8170.
i = 30 from3(i) = 1. cost1 = 9045.
i = 31 from3(i) = 20. cost1 = 9295.
i = 32 from3(i) = 10. cost1 = 10285.

Σενάριο 2: Για χωρητικότητα = 15

Τα αποτελέσματα του TSP

i = 1 from2(i) = 1. to2(i) = 21. cost1 = 140.
i = 2 from2(i) = 21. to2(i) = 19. cost1 = 215.
i = 3 from2(i) = 19. to2(i) = 6. cost1 = 325.
i = 4 from2(i) = 6. to2(i) = 7. cost1 = 440.
i = 5 from2(i) = 7. to2(i) = 5. cost1 = 520.
i = 6 from2(i) = 5. to2(i) = 2. cost1 = 750.
i = 7 from2(i) = 2. to2(i) = 13. cost1 = 865.
i = 8 from2(i) = 13. to2(i) = 14. cost1 = 940.
i = 9 from2(i) = 14. to2(i) = 18. cost1 = 1210.
i = 10 from2(i) = 18. to2(i) = 12. cost1 = 1420.
i = 11 from2(i) = 12. to2(i) = 27. cost1 = 1555.
i = 12 from2(i) = 27. to2(i) = 24. cost1 = 1680.
i = 13 from2(i) = 24. to2(i) = 23. cost1 = 1740.
i = 14 from2(i) = 23. to2(i) = 15. cost1 = 1895.
i = 15 from2(i) = 15. to2(i) = 3. cost1 = 2010.
i = 16 from2(i) = 3. to2(i) = 8. cost1 = 2300.
i = 17 from2(i) = 8. to2(i) = 4. cost1 = 2500.
i = 18 from2(i) = 4. to2(i) = 22. cost1 = 2635.
i = 19 from2(i) = 22. to2(i) = 17. cost1 = 2800.
i = 20 from2(i) = 17. to2(i) = 26. cost1 = 2875.
i = 21 from2(i) = 26. to2(i) = 16. cost1 = 3315.
i = 22 from2(i) = 16. to2(i) = 25. cost1 = 3515.
i = 23 from2(i) = 25. to2(i) = 9. cost1 = 3555.
i = 24 from2(i) = 9. to2(i) = 11. cost1 = 4135.
i = 25 from2(i) = 11. to2(i) = 20. cost1 = 5555.
i = 26 from2(i) = 20. to2(i) = 10. cost1 = 5805.
i = 27 from2(i) = 10. to2(i) = 1. cost1 = 6795.

Τα αποτελέσματα του VRP

i = 1 from3(i) = 1. to3(i) = 21. cost2 = 140.
i = 2 from3(i) = 21. to3(i) = 19. cost2 = 215.
i = 3 from3(i) = 19. to3(i) = 6. cost2 = 325.
i = 4 from3(i) = 6. to3(i) = 7. cost2 = 440.
i = 5 from3(i) = 7. to3(i) = 5. cost2 = 520.
i = 6 from3(i) = 5. to3(i) = 2. cost2 = 750.
i = 7 from3(i) = 2. to3(i) = 13. cost2 = 865.
i = 8 from3(i) = 13. to3(i) = 14. cost2 = 940.
i = 9 from3(i) = 14. to3(i) = 1. cost2 = 1290.
i = 10 from3(i) = 1. to3(i) = 18. cost2 = 1855.
i = 11 from3(i) = 18. to3(i) = 12. cost2 = 2065.
i = 12 from3(i) = 12. to3(i) = 27. cost2 = 2200.
i = 13 from3(i) = 27. to3(i) = 24. cost2 = 2325.
i = 14 from3(i) = 24. to3(i) = 23. cost2 = 2385.
i = 15 from3(i) = 23. to3(i) = 15. cost2 = 2540.
i = 16 from3(i) = 15. to3(i) = 1. cost2 = 3220.
i = 17 from3(i) = 1. to3(i) = 3. cost2 = 3960.
i = 18 from3(i) = 3. to3(i) = 8. cost2 = 4250.
i = 19 from3(i) = 8. to3(i) = 4. cost2 = 4450.
i = 20 from3(i) = 4. to3(i) = 22. cost2 = 4585.
i = 21 from3(i) = 22. to3(i) = 1. cost2 = 4955.
i = 22 from3(i) = 1. to3(i) = 17. cost2 = 5170.
i = 23 from3(i) = 17. to3(i) = 26. cost2 = 5245.
i = 24 from3(i) = 26. to3(i) = 16. cost2 = 5685.
i = 25 from3(i) = 16. to3(i) = 25. cost2 = 5885.
i = 26 from3(i) = 25. to3(i) = 9. cost2 = 5925.
i = 27 from3(i) = 9. to3(i) = 11. cost2 = 6505.
i = 28 from3(i) = 11. to3(i) = 20. cost2 = 7925.
i = 29 from3(i) = 20. to3(i) = 10. cost2 = 8175.
i = 30 from3(i) = 10. to3(i) = 1. cost2 = 9165.

Τα αποτελέσματα του 2-opt

i = 1 from3(i) = 1. cost1 = 140.
i = 2 from3(i) = 21. cost1 = 215.
i = 3 from3(i) = 19. cost1 = 325.
i = 4 from3(i) = 6. cost1 = 440.
i = 5 from3(i) = 7. cost1 = 520.
i = 6 from3(i) = 5. cost1 = 750.
i = 7 from3(i) = 2. cost1 = 865.
i = 8 from3(i) = 13. cost1 = 940.
i = 9 from3(i) = 14. cost1 = 1290.
i = 10 from3(i) = 1. cost1 = 1855.
i = 11 from3(i) = 18. cost1 = 2065.
i = 12 from3(i) = 12. cost1 = 2200.

i = 13 from3(i) = 27. cost1 = 2325.
i = 14 from3(i) = 24. cost1 = 2385.
i = 15 from3(i) = 23. cost1 = 2540.
i = 16 from3(i) = 15. cost1 = 3220.
i = 17 from3(i) = 1. cost1 = 3610.
i = 18 from3(i) = 4. cost1 = 3810.
i = 19 from3(i) = 8. cost1 = 4100.
i = 20 from3(i) = 3. cost1 = 4430.
i = 21 from3(i) = 22. cost1 = 4800.
i = 22 from3(i) = 1. cost1 = 4955.
i = 23 from3(i) = 26. cost1 = 5030.
i = 24 from3(i) = 17. cost1 = 5495.
i = 25 from3(i) = 16. cost1 = 5825.
i = 26 from3(i) = 11. cost1 = 6390.
i = 27 from3(i) = 25. cost1 = 6430.
i = 28 from3(i) = 9. cost1 = 7400.
i = 29 from3(i) = 20. cost1 = 7650.
i = 30 from3(i) = 10. cost1 = 8640.

Σενάριο3: Για χωρητικότητα = 8

Τα αποτελέσματα του TSP

i = 1 from2(i) = 1. to2(i) = 21. cost1 = 140.
i = 2 from2(i) = 21. to2(i) = 19. cost1 = 215.
i = 3 from2(i) = 19. to2(i) = 6. cost1 = 325.
i = 4 from2(i) = 6. to2(i) = 7. cost1 = 440.
i = 5 from2(i) = 7. to2(i) = 5. cost1 = 520.
i = 6 from2(i) = 5. to2(i) = 2. cost1 = 750.
i = 7 from2(i) = 2. to2(i) = 13. cost1 = 865.
i = 8 from2(i) = 13. to2(i) = 14. cost1 = 940.
i = 9 from2(i) = 14. to2(i) = 18. cost1 = 1210.
i = 10 from2(i) = 18. to2(i) = 12. cost1 = 1420.
i = 11 from2(i) = 12. to2(i) = 27. cost1 = 1555.
i = 12 from2(i) = 27. to2(i) = 24. cost1 = 1680.
i = 13 from2(i) = 24. to2(i) = 23. cost1 = 1740.
i = 14 from2(i) = 23. to2(i) = 15. cost1 = 1895.
i = 15 from2(i) = 15. to2(i) = 3. cost1 = 2010.
i = 16 from2(i) = 3. to2(i) = 8. cost1 = 2300.
i = 17 from2(i) = 8. to2(i) = 4. cost1 = 2500.
i = 18 from2(i) = 4. to2(i) = 22. cost1 = 2635.
i = 19 from2(i) = 22. to2(i) = 17. cost1 = 2800.
i = 20 from2(i) = 17. to2(i) = 26. cost1 = 2875.
i = 21 from2(i) = 26. to2(i) = 16. cost1 = 3315.
i = 22 from2(i) = 16. to2(i) = 25. cost1 = 3515.
i = 23 from2(i) = 25. to2(i) = 9. cost1 = 3555.
i = 24 from2(i) = 9. to2(i) = 11. cost1 = 4135.

i = 25 from2(i) = 11. to2(i) = 20. cost1 = 5555.
i = 26 from2(i) = 20. to2(i) = 10. cost1 = 5805.
i = 27 from2(i) = 10. to2(i) = 1. cost1 = 6795.

Τα αποτελέσματα του VRP

i = 1 from3(i) = 1. to3(i) = 21. cost2 = 140.
i = 2 from3(i) = 21. to3(i) = 19. cost2 = 215.
i = 3 from3(i) = 19. to3(i) = 6. cost2 = 325.
i = 4 from3(i) = 6. to3(i) = 1. cost2 = 535.
i = 5 from3(i) = 1. to3(i) = 7. cost2 = 745.
i = 6 from3(i) = 7. to3(i) = 5. cost2 = 825.
i = 7 from3(i) = 5. to3(i) = 2. cost2 = 1055.
i = 8 from3(i) = 2. to3(i) = 13. cost2 = 1170.
i = 9 from3(i) = 13. to3(i) = 14. cost2 = 1245.
i = 10 from3(i) = 14. to3(i) = 1. cost2 = 1595.
i = 11 from3(i) = 1. to3(i) = 18. cost2 = 2160.
i = 12 from3(i) = 18. to3(i) = 12. cost2 = 2370.
i = 13 from3(i) = 12. to3(i) = 27. cost2 = 2505.
i = 14 from3(i) = 27. to3(i) = 24. cost2 = 2630.
i = 15 from3(i) = 24. to3(i) = 23. cost2 = 2690.
i = 16 from3(i) = 23. to3(i) = 1. cost2 = 3525.
i = 17 from3(i) = 1. to3(i) = 15. cost2 = 4205.
i = 18 from3(i) = 15. to3(i) = 1. cost2 = 4885.
i = 19 from3(i) = 1. to3(i) = 3. cost2 = 5625.
i = 20 from3(i) = 3. to3(i) = 1. cost2 = 6365.
i = 21 from3(i) = 1. to3(i) = 8. cost2 = 6945.
i = 22 from3(i) = 8. to3(i) = 4. cost2 = 7145.
i = 23 from3(i) = 4. to3(i) = 22. cost2 = 7280.
i = 24 from3(i) = 22. to3(i) = 17. cost2 = 7445.
i = 25 from3(i) = 17. to3(i) = 26. cost2 = 7520.
i = 26 from3(i) = 26. to3(i) = 16. cost2 = 7960.
i = 27 from3(i) = 16. to3(i) = 25. cost2 = 8160.
i = 28 from3(i) = 25. to3(i) = 9. cost2 = 8200.
i = 29 from3(i) = 9. to3(i) = 11. cost2 = 8780.
i = 30 from3(i) = 11. to3(i) = 1. cost2 = 9400.
i = 31 from3(i) = 1. to3(i) = 20. cost2 = 10275.
i = 32 from3(i) = 20. to3(i) = 10. cost2 = 10525.
i = 33 from3(i) = 10. to3(i) = 1. cost2 = 11515.

Τα αποτελέσματα του 2-opt

i = 1 from3(i) = 1. cost1 = 140.
i = 2 from3(i) = 21. cost1 = 215.
i = 3 from3(i) = 19. cost1 = 325.
i = 4 from3(i) = 6. cost1 = 535.

i = 5 from3(i) = 1. cost1 = 745.
i = 6 from3(i) = 7. cost1 = 825.
i = 7 from3(i) = 5. cost1 = 1055.
i = 8 from3(i) = 2. cost1 = 1170.
i = 9 from3(i) = 13. cost1 = 1245.
i = 10 from3(i) = 14. cost1 = 1595.
i = 11 from3(i) = 1. cost1 = 2160.
i = 12 from3(i) = 18. cost1 = 2370.
i = 13 from3(i) = 12. cost1 = 2505.
i = 14 from3(i) = 27. cost1 = 2630.
i = 15 from3(i) = 24. cost1 = 2690.
i = 16 from3(i) = 23. cost1 = 3525.
i = 17 from3(i) = 1. cost1 = 4205.
i = 18 from3(i) = 15. cost1 = 4885.
i = 19 from3(i) = 1. cost1 = 5625.
i = 20 from3(i) = 3. cost1 = 6365.
i = 21 from3(i) = 1. cost1 = 6945.
i = 22 from3(i) = 8. cost1 = 7145.
i = 23 from3(i) = 4. cost1 = 7280.
i = 24 from3(i) = 22. cost1 = 7445.
i = 25 from3(i) = 17. cost1 = 7520.
i = 26 from3(i) = 26. cost1 = 7960.
i = 27 from3(i) = 16. cost1 = 8160.
i = 28 from3(i) = 25. cost1 = 8200.
i = 29 from3(i) = 9. cost1 = 8780.
i = 30 from3(i) = 11. cost1 = 9400.
i = 31 from3(i) = 1. cost1 = 10275.
i = 32 from3(i) = 20. cost1 = 10525.
i = 33 from3(i) = 10. cost1 = 11515.

Αποτελέσματα εφαρμογής 2

Δεδομένα

Αριθμός κόμβων = 201
Χωρητικότητα οχήματος = 700

Κόμβ.	X	Y	Διανομές	Νωρίτερος Χρόνος	Αργότερος Χρόνος	Χρ. Εξυπ.	Παραλαβ.
1	70	70	0	7	18	0	0
2	38	105	20	7	18	90	30
3	100	5	20	7	18	90	20
4	111	78	10	7	18	90	20
5	96	62	10	7	18	90	30
6	122	11	30	7	18	90	40
7	94	18	30	7	18	90	20
8	60	89	10	7	18	90	10
9	67	123	20	7	18	90	30
10	99	53	20	7	18	90	10
11	26	24	20	7	18	90	20
12	93	61	10	7	18	90	10
13	27	32	20	7	18	90	20
14	25	31	20	7	18	90	20
15	98	36	10	7	18	90	10
16	101	68	10	7	18	90	10
17	49	48	10	7	18	90	40
18	56	95	10	7	18	90	30
19	14	107	20	7	18	90	20
20	98	46	20	7	18	90	20
21	131	4	40	7	18	90	20
22	94	89	20	7	18	90	10
23	70	113	30	7	18	90	30
24	94	59	30	7	18	90	20
25	64	59	10	7	18	90	10
26	112	3	10	7	18	90	10
27	20	108	10	7	18	90	20
28	10	118	20	7	18	90	20
29	76	65	40	7	18	90	10
30	103	12	20	7	18	90	20
31	44	62	30	7	18	90	30
32	96	138	10	7	18	90	30
33	27	44	20	7	18	90	20
34	21	28	40	7	18	90	10
35	26	91	10	7	18	90	40
36	98	75	10	7	18	90	20
37	69	48	20	7	18	90	10
38	19	116	10	7	18	90	30
39	60	111	10	7	18	90	10
40	54	102	20	7	18	90	20
41	63	103	20	7	18	90	10
42	91	91	10	7	18	90	20
43	77	15	10	7	18	90	10
44	77	117	30	7	18	90	30
45	115	41	20	7	18	90	20
46	12	33	10	7	18	90	10
47	44	82	20	7	18	90	20

48	117	55	20	7	18	90	20
49	116	10	20	7	18	90	30
50	57	77	20	7	18	90	20
51	29	36	20	7	18	90	10
52	90	91	10	7	18	90	10
53	62	73	20	7	18	90	20
54	6	121	20	7	18	90	20
55	105	49	10	7	18	90	10
56	28	115	10	7	18	90	10
57	4	137	30	7	18	90	20
58	11	100	20	7	18	90	20
59	67	84	30	7	18	90	10
60	23	22	20	7	18	90	30
61	41	64	20	7	18	90	20
62	56	56	40	7	18	90	20
63	80	40	20	7	18	90	20
64	60	66	30	7	18	90	10
65	16	90	20	7	18	90	20
66	104	82	30	7	18	90	10
67	64	78	10	7	18	90	30
68	47	36	20	7	18	90	20
69	46	80	20	7	18	90	10
70	86	124	20	7	18	90	30
71	48	72	20	7	18	90	20
72	121	9	10	7	18	90	20
73	60	52	20	7	18	90	10
74	25	46	20	7	18	90	20
75	26	15	20	7	18	90	10
76	98	116	30	7	18	90	20
77	66	131	10	7	18	90	20
78	15	30	20	7	18	90	10
79	66	117	30	7	18	90	20
80	111	70	20	7	18	90	10
81	123	19	20	7	18	90	20
82	17	18	30	7	18	90	20
83	57	87	10	7	18	90	30
84	102	33	30	7	18	90	10
85	7	113	10	7	18	90	10
86	37	115	20	7	18	90	20
87	17	25	10	7	18	90	20
88	83	105	30	7	18	90	10
89	51	115	10	7	18	90	10
90	96	55	20	7	18	90	20
91	112	2	20	7	18	90	10
92	116	23	20	7	18	90	40
93	57	70	10	7	18	90	20
94	15	23	30	7	18	90	10
95	45	68	30	7	18	90	20
96	19	41	10	7	18	90	10
97	38	16	10	7	18	90	20
98	81	135	20	7	18	90	20
99	62	128	20	7	18	90	20
100	49	70	20	7	18	90	10
101	24	19	10	7	18	90	20
102	119	20	10	7	18	90	20
103	62	125	30	7	18	90	20
104	29	10	30	7	18	90	10
105	13	80	30	7	18	90	10
106	65	73	10	7	18	90	20
107	109	25	10	7	18	90	10
108	91	71	10	7	18	90	20

109	47	57	30	7	18	90	30
110	9	42	20	7	18	90	30
111	96	23	30	7	18	90	20
112	26	102	40	7	18	90	10
113	4	15	20	7	18	90	40
114	3	98	10	7	18	90	20
115	35	88	20	7	18	90	10
116	18	107	10	7	18	90	20
117	95	68	20	7	18	90	20
118	58	85	10	7	18	90	10
119	11	23	10	7	18	90	20
120	61	107	10	7	18	90	10
121	3	17	10	7	18	90	20
122	26	104	10	7	18	90	10
123	74	56	10	7	18	90	10
124	24	105	10	7	18	90	20
125	30	106	30	7	18	90	10
126	4	33	20	7	18	90	20
127	106	0	10	7	18	90	10
128	120	10	20	7	18	90	40
129	106	5	30	7	18	90	20
130	30	58	20	7	18	90	20
131	9	9	10	7	18	90	30
132	7	92	20	7	18	90	20
133	50	29	10	7	18	90	20
134	46	52	10	7	18	90	10
135	70	110	30	7	18	90	30
136	37	111	20	7	18	90	40
137	75	64	20	7	18	90	20
138	20	93	10	7	18	90	30
139	94	1	10	7	18	90	20
140	41	12	10	7	18	90	10
141	20	109	10	7	18	90	10
142	28	14	20	7	18	90	20
143	87	63	30	7	18	90	30
144	101	4	10	7	18	90	20
145	80	22	10	7	18	90	10
146	117	10	10	7	18	90	30
147	103	33	10	7	18	90	20
148	104	51	20	7	18	90	20
149	103	19	30	7	18	90	20
150	7	52	20	7	18	90	20
151	73	119	30	7	18	90	10
152	79	128	30	7	18	90	20
153	10	50	20	7	18	90	10
154	26	23	50	7	18	90	20
155	95	103	20	7	18	90	30
156	101	59	30	7	18	90	20
157	113	67	20	7	18	90	10
158	25	94	10	7	18	90	10
159	120	20	20	7	18	90	20
160	105	26	30	7	18	90	30
161	59	77	20	7	18	90	20
162	116	34	20	7	18	90	20
163	45	71	10	7	18	90	10
164	86	109	10	7	18	90	10
165	15	96	10	7	18	90	30
166	64	88	30	7	18	90	20
167	60	115	20	7	18	90	20
168	63	100	10	7	18	90	10
169	94	104	30	7	18	90	30

170	8	102	10	7	18	90	20
171	64	89	10	7	18	90	30
172	61	66	10	7	18	90	10
173	95	121	10	7	18	90	10
174	47	63	30	7	18	90	10
175	19	96	10	7	18	90	20
176	103	55	20	7	18	90	10
177	36	94	30	7	18	90	40
178	75	103	20	7	18	90	20
179	65	80	10	7	18	90	30
180	94	56	30	7	18	90	20
181	18	123	20	7	18	90	10
182	92	123	10	7	18	90	10
183	12	31	20	7	18	90	20
184	106	81	20	7	18	90	10
185	98	28	10	7	18	90	30
186	14	133	30	7	18	90	20
187	101	32	20	7	18	90	10
188	25	66	10	7	18	90	30
189	78	61	40	7	18	90	20
190	123	64	20	7	18	90	10
191	126	20	20	7	18	90	30
192	72	117	10	7	18	90	10
193	10	105	20	7	18	90	20
194	128	0	10	7	18	90	20
195	130	16	30	7	18	90	10
196	114	63	10	7	18	90	30
197	80	63	20	7	18	90	10
198	113	36	10	7	18	90	20
199	15	94	30	7	18	90	30
200	129	77	10	7	18	90	10
201	85	85	20	7	18	90	30

Τα αποτελέσματα του TSP

i = 1 from2(i) = 1. to2(i) = 106. cost1 = 5.83095169
i = 2 from2(i) = 106. to2(i) = 53. cost1 = 8.83095169
i = 3 from2(i) = 53. to2(i) = 161. cost1 = 13.8309517
i = 4 from2(i) = 161. to2(i) = 50. cost1 = 15.8309517
i = 5 from2(i) = 50. to2(i) = 93. cost1 = 22.8309517
i = 6 from2(i) = 93. to2(i) = 64. cost1 = 27.8309517
i = 7 from2(i) = 64. to2(i) = 172. cost1 = 28.8309517
i = 8 from2(i) = 172. to2(i) = 25. cost1 = 36.4467239
i = 9 from2(i) = 25. to2(i) = 73. cost1 = 44.5089798
i = 10 from2(i) = 73. to2(i) = 62. cost1 = 50.1658325
i = 11 from2(i) = 62. to2(i) = 109. cost1 = 59.2212181
i = 12 from2(i) = 109. to2(i) = 134. cost1 = 64.3202362
i = 13 from2(i) = 134. to2(i) = 17. cost1 = 69.3202362
i = 14 from2(i) = 17. to2(i) = 68. cost1 = 81.4857635
i = 15 from2(i) = 68. to2(i) = 133. cost1 = 89.1015396
i = 16 from2(i) = 133. to2(i) = 97. cost1 = 106.79335
i = 17 from2(i) = 97. to2(i) = 140. cost1 = 111.79335
i = 18 from2(i) = 140. to2(i) = 104. cost1 = 123.958878

i = 19 from2(i) = 104. to2(i) = 142. cost1 = 128.081985
i = 20 from2(i) = 142. to2(i) = 75. cost1 = 130.318054
i = 21 from2(i) = 75. to2(i) = 101. cost1 = 134.790192
i = 22 from2(i) = 101. to2(i) = 60. cost1 = 137.952469
i = 23 from2(i) = 60. to2(i) = 154. cost1 = 141.114746
i = 24 from2(i) = 154. to2(i) = 11. cost1 = 142.114746
i = 25 from2(i) = 11. to2(i) = 34. cost1 = 148.517868
i = 26 from2(i) = 34. to2(i) = 14. cost1 = 153.517868
i = 27 from2(i) = 14. to2(i) = 13. cost1 = 155.753937
i = 28 from2(i) = 13. to2(i) = 51. cost1 = 160.226074
i = 29 from2(i) = 51. to2(i) = 33. cost1 = 168.47229
i = 30 from2(i) = 33. to2(i) = 74. cost1 = 171.30072
i = 31 from2(i) = 74. to2(i) = 96. cost1 = 179.110977
i = 32 from2(i) = 96. to2(i) = 110. cost1 = 189.160858
i = 33 from2(i) = 110. to2(i) = 153. cost1 = 197.223114
i = 34 from2(i) = 153. to2(i) = 150. cost1 = 200.828659
i = 35 from2(i) = 150. to2(i) = 126. cost1 = 220.064041
i = 36 from2(i) = 126. to2(i) = 46. cost1 = 228.064041
i = 37 from2(i) = 46. to2(i) = 183. cost1 = 230.064041
i = 38 from2(i) = 183. to2(i) = 78. cost1 = 233.226318
i = 39 from2(i) = 78. to2(i) = 87. cost1 = 238.611481
i = 40 from2(i) = 87. to2(i) = 94. cost1 = 241.439911
i = 41 from2(i) = 94. to2(i) = 119. cost1 = 245.439911
i = 42 from2(i) = 119. to2(i) = 82. cost1 = 253.250168
i = 43 from2(i) = 82. to2(i) = 131. cost1 = 265.291748
i = 44 from2(i) = 131. to2(i) = 113. cost1 = 273.10199
i = 45 from2(i) = 113. to2(i) = 121. cost1 = 275.338043
i = 46 from2(i) = 121. to2(i) = 130. cost1 = 324.42981
i = 47 from2(i) = 130. to2(i) = 188. cost1 = 333.8638
i = 48 from2(i) = 188. to2(i) = 61. cost1 = 349.988312
i = 49 from2(i) = 61. to2(i) = 31. cost1 = 353.593872
i = 50 from2(i) = 31. to2(i) = 174. cost1 = 356.756165
i = 51 from2(i) = 174. to2(i) = 95. cost1 = 362.141327
i = 52 from2(i) = 95. to2(i) = 163. cost1 = 365.141327
i = 53 from2(i) = 163. to2(i) = 71. cost1 = 368.303619
i = 54 from2(i) = 71. to2(i) = 100. cost1 = 370.539673
i = 55 from2(i) = 100. to2(i) = 69. cost1 = 380.97998
i = 56 from2(i) = 69. to2(i) = 47. cost1 = 383.808411
i = 57 from2(i) = 47. to2(i) = 115. cost1 = 394.625061
i = 58 from2(i) = 115. to2(i) = 177. cost1 = 400.707825
i = 59 from2(i) = 177. to2(i) = 35. cost1 = 411.148132
i = 60 from2(i) = 35. to2(i) = 158. cost1 = 414.310425
i = 61 from2(i) = 158. to2(i) = 138. cost1 = 419.409454
i = 62 from2(i) = 138. to2(i) = 175. cost1 = 422.571747
i = 63 from2(i) = 175. to2(i) = 165. cost1 = 426.571747
i = 64 from2(i) = 165. to2(i) = 199. cost1 = 428.571747
i = 65 from2(i) = 199. to2(i) = 65. cost1 = 432.694855
i = 66 from2(i) = 65. to2(i) = 132. cost1 = 441.914398
i = 67 from2(i) = 132. to2(i) = 114. cost1 = 449.125488
i = 68 from2(i) = 114. to2(i) = 170. cost1 = 455.528625

i = 69 from2(i) = 170. to2(i) = 58. cost1 = 459.134186
i = 70 from2(i) = 58. to2(i) = 193. cost1 = 464.233215
i = 71 from2(i) = 193. to2(i) = 19. cost1 = 468.705353
i = 72 from2(i) = 19. to2(i) = 116. cost1 = 472.705353
i = 73 from2(i) = 116. to2(i) = 27. cost1 = 474.941406
i = 74 from2(i) = 27. to2(i) = 141. cost1 = 475.941406
i = 75 from2(i) = 141. to2(i) = 124. cost1 = 481.598267
i = 76 from2(i) = 124. to2(i) = 122. cost1 = 483.83432
i = 77 from2(i) = 122. to2(i) = 112. cost1 = 485.83432
i = 78 from2(i) = 112. to2(i) = 125. cost1 = 491.49118
i = 79 from2(i) = 125. to2(i) = 2. cost1 = 499.553436
i = 80 from2(i) = 2. to2(i) = 136. cost1 = 505.6362
i = 81 from2(i) = 136. to2(i) = 86. cost1 = 509.6362
i = 82 from2(i) = 86. to2(i) = 56. cost1 = 518.63623
i = 83 from2(i) = 56. to2(i) = 38. cost1 = 527.691589
i = 84 from2(i) = 38. to2(i) = 181. cost1 = 534.762634
i = 85 from2(i) = 181. to2(i) = 28. cost1 = 544.196594
i = 86 from2(i) = 28. to2(i) = 54. cost1 = 549.196594
i = 87 from2(i) = 54. to2(i) = 85. cost1 = 557.25885
i = 88 from2(i) = 85. to2(i) = 186. cost1 = 578.448486
i = 89 from2(i) = 186. to2(i) = 57. cost1 = 589.218811
i = 90 from2(i) = 57. to2(i) = 89. cost1 = 641.112915
i = 91 from2(i) = 89. to2(i) = 167. cost1 = 650.112915
i = 92 from2(i) = 167. to2(i) = 39. cost1 = 654.112915
i = 93 from2(i) = 39. to2(i) = 120. cost1 = 658.236023
i = 94 from2(i) = 120. to2(i) = 41. cost1 = 662.70813
i = 95 from2(i) = 41. to2(i) = 168. cost1 = 665.70813
i = 96 from2(i) = 168. to2(i) = 18. cost1 = 674.310425
i = 97 from2(i) = 18. to2(i) = 8. cost1 = 681.521545
i = 98 from2(i) = 8. to2(i) = 83. cost1 = 685.127075
i = 99 from2(i) = 83. to2(i) = 118. cost1 = 687.363159
i = 100 from2(i) = 118. to2(i) = 166. cost1 = 694.07135
i = 101 from2(i) = 166. to2(i) = 171. cost1 = 695.07135
i = 102 from2(i) = 171. to2(i) = 59. cost1 = 700.902283
i = 103 from2(i) = 59. to2(i) = 179. cost1 = 705.37439
i = 104 from2(i) = 179. to2(i) = 67. cost1 = 707.610474
i = 105 from2(i) = 67. to2(i) = 29. cost1 = 725.302307
i = 106 from2(i) = 29. to2(i) = 137. cost1 = 726.716492
i = 107 from2(i) = 137. to2(i) = 189. cost1 = 730.959106
i = 108 from2(i) = 189. to2(i) = 197. cost1 = 733.787537
i = 109 from2(i) = 197. to2(i) = 143. cost1 = 740.787537
i = 110 from2(i) = 143. to2(i) = 12. cost1 = 747.112122
i = 111 from2(i) = 12. to2(i) = 24. cost1 = 749.348206
i = 112 from2(i) = 24. to2(i) = 180. cost1 = 752.348206
i = 113 from2(i) = 180. to2(i) = 90. cost1 = 754.58429
i = 114 from2(i) = 90. to2(i) = 10. cost1 = 758.189819
i = 115 from2(i) = 10. to2(i) = 176. cost1 = 762.661926
i = 116 from2(i) = 176. to2(i) = 148. cost1 = 766.785034
i = 117 from2(i) = 148. to2(i) = 55. cost1 = 769.021118
i = 118 from2(i) = 55. to2(i) = 20. cost1 = 776.636902

i = 119 from2(i) = 20. to2(i) = 15. cost1 = 786.636902
i = 120 from2(i) = 15. to2(i) = 84. cost1 = 791.636902
i = 121 from2(i) = 84. to2(i) = 147. cost1 = 792.636902
i = 122 from2(i) = 147. to2(i) = 187. cost1 = 794.872986
i = 123 from2(i) = 187. to2(i) = 185. cost1 = 799.872986
i = 124 from2(i) = 185. to2(i) = 111. cost1 = 805.258179
i = 125 from2(i) = 111. to2(i) = 7. cost1 = 810.643372
i = 126 from2(i) = 7. to2(i) = 149. cost1 = 819.69873
i = 127 from2(i) = 149. to2(i) = 30. cost1 = 826.69873
i = 128 from2(i) = 30. to2(i) = 3. cost1 = 834.314514
i = 129 from2(i) = 3. to2(i) = 144. cost1 = 835.728699
i = 130 from2(i) = 144. to2(i) = 129. cost1 = 840.827698
i = 131 from2(i) = 129. to2(i) = 127. cost1 = 845.827698
i = 132 from2(i) = 127. to2(i) = 91. cost1 = 852.152283
i = 133 from2(i) = 91. to2(i) = 26. cost1 = 853.152283
i = 134 from2(i) = 26. to2(i) = 49. cost1 = 861.214539
i = 135 from2(i) = 49. to2(i) = 146. cost1 = 862.214539
i = 136 from2(i) = 146. to2(i) = 128. cost1 = 865.214539
i = 137 from2(i) = 128. to2(i) = 72. cost1 = 866.628723
i = 138 from2(i) = 72. to2(i) = 6. cost1 = 868.864807
i = 139 from2(i) = 6. to2(i) = 81. cost1 = 876.927063
i = 140 from2(i) = 81. to2(i) = 159. cost1 = 880.089355
i = 141 from2(i) = 159. to2(i) = 102. cost1 = 881.089355
i = 142 from2(i) = 102. to2(i) = 92. cost1 = 885.33197
i = 143 from2(i) = 92. to2(i) = 107. cost1 = 892.612061
i = 144 from2(i) = 107. to2(i) = 160. cost1 = 896.735168
i = 145 from2(i) = 160. to2(i) = 198. cost1 = 909.541443
i = 146 from2(i) = 198. to2(i) = 162. cost1 = 913.146973
i = 147 from2(i) = 162. to2(i) = 45. cost1 = 920.218018
i = 148 from2(i) = 45. to2(i) = 48. cost1 = 934.360168
i = 149 from2(i) = 48. to2(i) = 196. cost1 = 942.904175
i = 150 from2(i) = 196. to2(i) = 157. cost1 = 947.027283
i = 151 from2(i) = 157. to2(i) = 80. cost1 = 950.632812
i = 152 from2(i) = 80. to2(i) = 4. cost1 = 958.632812
i = 153 from2(i) = 4. to2(i) = 184. cost1 = 964.463745
i = 154 from2(i) = 184. to2(i) = 66. cost1 = 966.699829
i = 155 from2(i) = 66. to2(i) = 36. cost1 = 975.919373
i = 156 from2(i) = 36. to2(i) = 16. cost1 = 983.535156
i = 157 from2(i) = 16. to2(i) = 117. cost1 = 989.535156
i = 158 from2(i) = 117. to2(i) = 108. cost1 = 994.535156
i = 159 from2(i) = 108. to2(i) = 5. cost1 = 1004.83081
i = 160 from2(i) = 5. to2(i) = 156. cost1 = 1010.66174
i = 161 from2(i) = 156. to2(i) = 190. cost1 = 1033.22278
i = 162 from2(i) = 190. to2(i) = 200. cost1 = 1047.54065
i = 163 from2(i) = 200. to2(i) = 22. cost1 = 1084.54065
i = 164 from2(i) = 22. to2(i) = 42. cost1 = 1088.14624
i = 165 from2(i) = 42. to2(i) = 52. cost1 = 1089.14624
i = 166 from2(i) = 52. to2(i) = 201. cost1 = 1096.95654
i = 167 from2(i) = 201. to2(i) = 88. cost1 = 1117.05627
i = 168 from2(i) = 88. to2(i) = 164. cost1 = 1122.05627

i = 169 from2(i) = 164. to2(i) = 169. cost1 = 1131.49023
i = 170 from2(i) = 169. to2(i) = 155. cost1 = 1132.90442
i = 171 from2(i) = 155. to2(i) = 76. cost1 = 1146.24609
i = 172 from2(i) = 76. to2(i) = 173. cost1 = 1152.07703
i = 173 from2(i) = 173. to2(i) = 182. cost1 = 1155.68262
i = 174 from2(i) = 182. to2(i) = 70. cost1 = 1161.76538
i = 175 from2(i) = 70. to2(i) = 152. cost1 = 1169.82764
i = 176 from2(i) = 152. to2(i) = 98. cost1 = 1177.10779
i = 177 from2(i) = 98. to2(i) = 32. cost1 = 1192.40479
i = 178 from2(i) = 32. to2(i) = 44. cost1 = 1220.72437
i = 179 from2(i) = 44. to2(i) = 151. cost1 = 1225.19653
i = 180 from2(i) = 151. to2(i) = 192. cost1 = 1227.43262
i = 181 from2(i) = 192. to2(i) = 23. cost1 = 1231.90479
i = 182 from2(i) = 23. to2(i) = 135. cost1 = 1234.90479
i = 183 from2(i) = 135. to2(i) = 79. cost1 = 1242.96704
i = 184 from2(i) = 79. to2(i) = 9. cost1 = 1249.0498
i = 185 from2(i) = 9. to2(i) = 103. cost1 = 1254.43494
i = 186 from2(i) = 103. to2(i) = 99. cost1 = 1257.43494
i = 187 from2(i) = 99. to2(i) = 77. cost1 = 1262.43494
i = 188 from2(i) = 77. to2(i) = 178. cost1 = 1291.84583
i = 189 from2(i) = 178. to2(i) = 40. cost1 = 1312.86963
i = 190 from2(i) = 40. to2(i) = 105. cost1 = 1359.39917
i = 191 from2(i) = 105. to2(i) = 37. cost1 = 1423.89722
i = 192 from2(i) = 37. to2(i) = 123. cost1 = 1433.33118
i = 193 from2(i) = 123. to2(i) = 63. cost1 = 1450.41919
i = 194 from2(i) = 63. to2(i) = 145. cost1 = 1468.41919
i = 195 from2(i) = 145. to2(i) = 43. cost1 = 1476.03491
i = 196 from2(i) = 43. to2(i) = 139. cost1 = 1498.05762
i = 197 from2(i) = 139. to2(i) = 194. cost1 = 1532.07227
i = 198 from2(i) = 194. to2(i) = 21. cost1 = 1537.07227
i = 199 from2(i) = 21. to2(i) = 195. cost1 = 1549.11389
i = 200 from2(i) = 195. to2(i) = 191. cost1 = 1554.77075
i = 201 from2(i) = 191. to2(i) = 1. cost1 = 1629.84399

Τα αποτελέσματα του VRP

i = 1 from3(i) = 1. to3(i) = 106. cost2 = 5.83095169
i = 2 from3(i) = 106. to3(i) = 53. cost2 = 8.83095169
i = 3 from3(i) = 53. to3(i) = 161. cost2 = 13.8309517
i = 4 from3(i) = 161. to3(i) = 50. cost2 = 15.8309517
i = 5 from3(i) = 50. to3(i) = 93. cost2 = 22.8309517
i = 6 from3(i) = 93. to3(i) = 1. cost2 = 35.8309517
i = 7 from3(i) = 1. to3(i) = 64. cost2 = 46.6012802
i = 8 from3(i) = 64. to3(i) = 172. cost2 = 47.6012802
i = 9 from3(i) = 172. to3(i) = 25. cost2 = 55.2170525
i = 10 from3(i) = 25. to3(i) = 73. cost2 = 63.2793121
i = 11 from3(i) = 73. to3(i) = 62. cost2 = 68.9361649
i = 12 from3(i) = 62. to3(i) = 109. cost2 = 77.9915466

i = 13 from3(i) = 109. to3(i) = 134. cost2 = 83.0905685
i = 14 from3(i) = 134. to3(i) = 17. cost2 = 88.0905685
i = 15 from3(i) = 17. to3(i) = 68. cost2 = 100.256096
i = 16 from3(i) = 68. to3(i) = 133. cost2 = 107.871872
i = 17 from3(i) = 133. to3(i) = 97. cost2 = 125.563675
i = 18 from3(i) = 97. to3(i) = 140. cost2 = 130.563675
i = 19 from3(i) = 140. to3(i) = 104. cost2 = 142.729202
i = 20 from3(i) = 104. to3(i) = 142. cost2 = 146.85231
i = 21 from3(i) = 142. to3(i) = 75. cost2 = 149.088379
i = 22 from3(i) = 75. to3(i) = 101. cost2 = 153.560516
i = 23 from3(i) = 101. to3(i) = 60. cost2 = 156.722794
i = 24 from3(i) = 60. to3(i) = 154. cost2 = 159.885071
i = 25 from3(i) = 154. to3(i) = 11. cost2 = 160.885071
i = 26 from3(i) = 11. to3(i) = 34. cost2 = 167.288193
i = 27 from3(i) = 34. to3(i) = 14. cost2 = 172.288193
i = 28 from3(i) = 14. to3(i) = 13. cost2 = 174.524261
i = 29 from3(i) = 13. to3(i) = 51. cost2 = 178.996399
i = 30 from3(i) = 51. to3(i) = 33. cost2 = 187.242615
i = 31 from3(i) = 33. to3(i) = 74. cost2 = 190.071045
i = 32 from3(i) = 74. to3(i) = 96. cost2 = 197.881302
i = 33 from3(i) = 96. to3(i) = 110. cost2 = 207.931183
i = 34 from3(i) = 110. to3(i) = 153. cost2 = 215.993439
i = 35 from3(i) = 153. to3(i) = 150. cost2 = 219.598984
i = 36 from3(i) = 150. to3(i) = 1. cost2 = 285.119965
i = 37 from3(i) = 1. to3(i) = 126. cost2 = 360.783691
i = 38 from3(i) = 126. to3(i) = 46. cost2 = 368.783691
i = 39 from3(i) = 46. to3(i) = 183. cost2 = 370.783691
i = 40 from3(i) = 183. to3(i) = 78. cost2 = 373.945984
i = 41 from3(i) = 78. to3(i) = 87. cost2 = 379.331146
i = 42 from3(i) = 87. to3(i) = 1. cost2 = 448.858124
i = 43 from3(i) = 1. to3(i) = 94. cost2 = 521.204529
i = 44 from3(i) = 94. to3(i) = 119. cost2 = 525.204529
i = 45 from3(i) = 119. to3(i) = 82. cost2 = 533.014771
i = 46 from3(i) = 82. to3(i) = 131. cost2 = 545.056335
i = 47 from3(i) = 131. to3(i) = 113. cost2 = 552.866577
i = 48 from3(i) = 113. to3(i) = 121. cost2 = 555.102661
i = 49 from3(i) = 121. to3(i) = 130. cost2 = 604.194397
i = 50 from3(i) = 130. to3(i) = 188. cost2 = 613.628357
i = 51 from3(i) = 188. to3(i) = 61. cost2 = 629.752869
i = 52 from3(i) = 61. to3(i) = 31. cost2 = 633.358398
i = 53 from3(i) = 31. to3(i) = 174. cost2 = 636.520691
i = 54 from3(i) = 174. to3(i) = 95. cost2 = 641.905884
i = 55 from3(i) = 95. to3(i) = 163. cost2 = 644.905884
i = 56 from3(i) = 163. to3(i) = 71. cost2 = 648.068176
i = 57 from3(i) = 71. to3(i) = 100. cost2 = 650.30426
i = 58 from3(i) = 100. to3(i) = 69. cost2 = 660.744568
i = 59 from3(i) = 69. to3(i) = 47. cost2 = 663.572998
i = 60 from3(i) = 47. to3(i) = 115. cost2 = 674.389648
i = 61 from3(i) = 115. to3(i) = 177. cost2 = 680.472412
i = 62 from3(i) = 177. to3(i) = 35. cost2 = 690.91272

i = 63 from3(i) = 35. to3(i) = 158. cost2 = 694.075012
i = 64 from3(i) = 158. to3(i) = 138. cost2 = 699.174011
i = 65 from3(i) = 138. to3(i) = 175. cost2 = 702.336304
i = 66 from3(i) = 175. to3(i) = 165. cost2 = 706.336304
i = 67 from3(i) = 165. to3(i) = 199. cost2 = 708.336304
i = 68 from3(i) = 199. to3(i) = 65. cost2 = 712.459412
i = 69 from3(i) = 65. to3(i) = 132. cost2 = 721.678955
i = 70 from3(i) = 132. to3(i) = 114. cost2 = 728.890076
i = 71 from3(i) = 114. to3(i) = 170. cost2 = 735.293213
i = 72 from3(i) = 170. to3(i) = 58. cost2 = 738.898743
i = 73 from3(i) = 58. to3(i) = 193. cost2 = 743.997742
i = 74 from3(i) = 193. to3(i) = 19. cost2 = 748.469849
i = 75 from3(i) = 19. to3(i) = 116. cost2 = 752.469849
i = 76 from3(i) = 116. to3(i) = 27. cost2 = 754.705933
i = 77 from3(i) = 27. to3(i) = 141. cost2 = 755.705933
i = 78 from3(i) = 141. to3(i) = 124. cost2 = 761.362793
i = 79 from3(i) = 124. to3(i) = 122. cost2 = 763.598877
i = 80 from3(i) = 122. to3(i) = 1. cost2 = 819.204651
i = 81 from3(i) = 1. to3(i) = 112. cost2 = 873.610535
i = 82 from3(i) = 112. to3(i) = 125. cost2 = 879.267395
i = 83 from3(i) = 125. to3(i) = 2. cost2 = 887.329651
i = 84 from3(i) = 2. to3(i) = 136. cost2 = 893.412415
i = 85 from3(i) = 136. to3(i) = 86. cost2 = 897.412415
i = 86 from3(i) = 86. to3(i) = 56. cost2 = 906.412415
i = 87 from3(i) = 56. to3(i) = 38. cost2 = 915.467773
i = 88 from3(i) = 38. to3(i) = 181. cost2 = 922.538818
i = 89 from3(i) = 181. to3(i) = 28. cost2 = 931.972778
i = 90 from3(i) = 28. to3(i) = 54. cost2 = 936.972778
i = 91 from3(i) = 54. to3(i) = 85. cost2 = 945.035034
i = 92 from3(i) = 85. to3(i) = 186. cost2 = 966.22467
i = 93 from3(i) = 186. to3(i) = 57. cost2 = 976.994995
i = 94 from3(i) = 57. to3(i) = 1. cost2 = 1071.04285
i = 95 from3(i) = 1. to3(i) = 89. cost2 = 1119.88953
i = 96 from3(i) = 89. to3(i) = 167. cost2 = 1128.88953
i = 97 from3(i) = 167. to3(i) = 39. cost2 = 1132.88953
i = 98 from3(i) = 39. to3(i) = 120. cost2 = 1137.01257
i = 99 from3(i) = 120. to3(i) = 41. cost2 = 1141.48474
i = 100 from3(i) = 41. to3(i) = 168. cost2 = 1144.48474
i = 101 from3(i) = 168. to3(i) = 18. cost2 = 1153.08704
i = 102 from3(i) = 18. to3(i) = 8. cost2 = 1160.2981
i = 103 from3(i) = 8. to3(i) = 83. cost2 = 1163.90369
i = 104 from3(i) = 83. to3(i) = 118. cost2 = 1166.13977
i = 105 from3(i) = 118. to3(i) = 166. cost2 = 1172.84802
i = 106 from3(i) = 166. to3(i) = 171. cost2 = 1173.84802
i = 107 from3(i) = 171. to3(i) = 59. cost2 = 1179.67896
i = 108 from3(i) = 59. to3(i) = 179. cost2 = 1184.15112
i = 109 from3(i) = 179. to3(i) = 67. cost2 = 1186.38721
i = 110 from3(i) = 67. to3(i) = 29. cost2 = 1204.07898
i = 111 from3(i) = 29. to3(i) = 137. cost2 = 1205.49316
i = 112 from3(i) = 137. to3(i) = 189. cost2 = 1209.73584

i = 113 from3(i) = 189. to3(i) = 197. cost2 = 1212.56421
i = 114 from3(i) = 197. to3(i) = 143. cost2 = 1219.56421
i = 115 from3(i) = 143. to3(i) = 12. cost2 = 1225.88879
i = 116 from3(i) = 12. to3(i) = 24. cost2 = 1228.12488
i = 117 from3(i) = 24. to3(i) = 180. cost2 = 1231.12488
i = 118 from3(i) = 180. to3(i) = 90. cost2 = 1233.36096
i = 119 from3(i) = 90. to3(i) = 10. cost2 = 1236.96655
i = 120 from3(i) = 10. to3(i) = 176. cost2 = 1241.43872
i = 121 from3(i) = 176. to3(i) = 148. cost2 = 1245.56177
i = 122 from3(i) = 148. to3(i) = 55. cost2 = 1247.79785
i = 123 from3(i) = 55. to3(i) = 20. cost2 = 1255.41357
i = 124 from3(i) = 20. to3(i) = 15. cost2 = 1265.41357
i = 125 from3(i) = 15. to3(i) = 84. cost2 = 1270.41357
i = 126 from3(i) = 84. to3(i) = 147. cost2 = 1271.41357
i = 127 from3(i) = 147. to3(i) = 187. cost2 = 1273.64966
i = 128 from3(i) = 187. to3(i) = 185. cost2 = 1278.64966
i = 129 from3(i) = 185. to3(i) = 111. cost2 = 1284.03479
i = 130 from3(i) = 111. to3(i) = 7. cost2 = 1289.41992
i = 131 from3(i) = 7. to3(i) = 149. cost2 = 1298.47534
i = 132 from3(i) = 149. to3(i) = 1. cost2 = 1359.2207
i = 133 from3(i) = 1. to3(i) = 30. cost2 = 1425.95154
i = 134 from3(i) = 30. to3(i) = 3. cost2 = 1433.56726
i = 135 from3(i) = 3. to3(i) = 144. cost2 = 1434.98145
i = 136 from3(i) = 144. to3(i) = 129. cost2 = 1440.08044
i = 137 from3(i) = 129. to3(i) = 127. cost2 = 1445.08044
i = 138 from3(i) = 127. to3(i) = 91. cost2 = 1451.40503
i = 139 from3(i) = 91. to3(i) = 26. cost2 = 1452.40503
i = 140 from3(i) = 26. to3(i) = 49. cost2 = 1460.46729
i = 141 from3(i) = 49. to3(i) = 146. cost2 = 1461.46729
i = 142 from3(i) = 146. to3(i) = 128. cost2 = 1464.46729
i = 143 from3(i) = 128. to3(i) = 72. cost2 = 1465.88147
i = 144 from3(i) = 72. to3(i) = 6. cost2 = 1468.11755
i = 145 from3(i) = 6. to3(i) = 81. cost2 = 1476.17981
i = 146 from3(i) = 81. to3(i) = 159. cost2 = 1479.34204
i = 147 from3(i) = 159. to3(i) = 102. cost2 = 1480.34204
i = 148 from3(i) = 102. to3(i) = 92. cost2 = 1484.58472
i = 149 from3(i) = 92. to3(i) = 107. cost2 = 1491.86487
i = 150 from3(i) = 107. to3(i) = 160. cost2 = 1495.98792
i = 151 from3(i) = 160. to3(i) = 198. cost2 = 1508.79419
i = 152 from3(i) = 198. to3(i) = 162. cost2 = 1512.39978
i = 153 from3(i) = 162. to3(i) = 45. cost2 = 1519.47083
i = 154 from3(i) = 45. to3(i) = 48. cost2 = 1533.61292
i = 155 from3(i) = 48. to3(i) = 196. cost2 = 1542.15686
i = 156 from3(i) = 196. to3(i) = 157. cost2 = 1546.27991
i = 157 from3(i) = 157. to3(i) = 80. cost2 = 1549.8855
i = 158 from3(i) = 80. to3(i) = 4. cost2 = 1557.8855
i = 159 from3(i) = 4. to3(i) = 184. cost2 = 1563.71643
i = 160 from3(i) = 184. to3(i) = 66. cost2 = 1565.95251
i = 161 from3(i) = 66. to3(i) = 36. cost2 = 1575.17212
i = 162 from3(i) = 36. to3(i) = 16. cost2 = 1582.78784

i = 163 from3(i) = 16. to3(i) = 117. cost2 = 1588.78784
i = 164 from3(i) = 117. to3(i) = 108. cost2 = 1593.78784
i = 165 from3(i) = 108. to3(i) = 1. cost2 = 1614.81165
i = 166 from3(i) = 1. to3(i) = 5. cost2 = 1642.01465
i = 167 from3(i) = 5. to3(i) = 156. cost2 = 1647.84558
i = 168 from3(i) = 156. to3(i) = 190. cost2 = 1670.40662
i = 169 from3(i) = 190. to3(i) = 200. cost2 = 1684.72449
i = 170 from3(i) = 200. to3(i) = 22. cost2 = 1721.72449
i = 171 from3(i) = 22. to3(i) = 42. cost2 = 1725.33008
i = 172 from3(i) = 42. to3(i) = 52. cost2 = 1726.33008
i = 173 from3(i) = 52. to3(i) = 201. cost2 = 1734.14038
i = 174 from3(i) = 201. to3(i) = 88. cost2 = 1754.24011
i = 175 from3(i) = 88. to3(i) = 164. cost2 = 1759.24011
i = 176 from3(i) = 164. to3(i) = 169. cost2 = 1768.67407
i = 177 from3(i) = 169. to3(i) = 155. cost2 = 1770.08826
i = 178 from3(i) = 155. to3(i) = 76. cost2 = 1783.42993
i = 179 from3(i) = 76. to3(i) = 173. cost2 = 1789.26086
i = 180 from3(i) = 173. to3(i) = 182. cost2 = 1792.86646
i = 181 from3(i) = 182. to3(i) = 70. cost2 = 1798.94922
i = 182 from3(i) = 70. to3(i) = 152. cost2 = 1807.01147
i = 183 from3(i) = 152. to3(i) = 98. cost2 = 1814.29163
i = 184 from3(i) = 98. to3(i) = 32. cost2 = 1829.58862
i = 185 from3(i) = 32. to3(i) = 44. cost2 = 1857.9082
i = 186 from3(i) = 44. to3(i) = 151. cost2 = 1862.38037
i = 187 from3(i) = 151. to3(i) = 192. cost2 = 1864.61646
i = 188 from3(i) = 192. to3(i) = 23. cost2 = 1869.08862
i = 189 from3(i) = 23. to3(i) = 135. cost2 = 1872.08862
i = 190 from3(i) = 135. to3(i) = 79. cost2 = 1880.15088
i = 191 from3(i) = 79. to3(i) = 9. cost2 = 1886.23364
i = 192 from3(i) = 9. to3(i) = 103. cost2 = 1891.61877
i = 193 from3(i) = 103. to3(i) = 99. cost2 = 1894.61877
i = 194 from3(i) = 99. to3(i) = 1. cost2 = 1953.16785
i = 195 from3(i) = 1. to3(i) = 77. cost2 = 2014.29883
i = 196 from3(i) = 77. to3(i) = 178. cost2 = 2043.70972
i = 197 from3(i) = 178. to3(i) = 40. cost2 = 2064.7334
i = 198 from3(i) = 40. to3(i) = 105. cost2 = 2111.26294
i = 199 from3(i) = 105. to3(i) = 37. cost2 = 2175.76099
i = 200 from3(i) = 37. to3(i) = 123. cost2 = 2185.19507
i = 201 from3(i) = 123. to3(i) = 63. cost2 = 2202.28296
i = 202 from3(i) = 63. to3(i) = 145. cost2 = 2220.28296
i = 203 from3(i) = 145. to3(i) = 43. cost2 = 2227.89868
i = 204 from3(i) = 43. to3(i) = 139. cost2 = 2249.92139
i = 205 from3(i) = 139. to3(i) = 194. cost2 = 2283.93604
i = 206 from3(i) = 194. to3(i) = 21. cost2 = 2288.93604
i = 207 from3(i) = 21. to3(i) = 195. cost2 = 2300.97754
i = 208 from3(i) = 195. to3(i) = 191. cost2 = 2306.63428
i = 209 from3(i) = 191. to3(i) = 1. cost2 = 2381.70752

Τα αποτελέσματα του 2- opt

i = 1 from3(i) = 1. cost1 = 5.83095169
i = 2 from3(i) = 106. cost1 = 8.83095169
i = 3 from3(i) = 53. cost1 = 13.8309517
i = 4 from3(i) = 161. cost1 = 15.8309517
i = 5 from3(i) = 50. cost1 = 22.8309517
i = 6 from3(i) = 93. cost1 = 35.8309517
i = 7 from3(i) = 1. cost1 = 45.6798096
i = 8 from3(i) = 172. cost1 = 46.6798096
i = 9 from3(i) = 64. cost1 = 54.7420654
i = 10 from3(i) = 25. cost1 = 62.8043213
i = 11 from3(i) = 73. cost1 = 68.461174
i = 12 from3(i) = 62. cost1 = 77.5165558
i = 13 from3(i) = 109. cost1 = 82.6155777
i = 14 from3(i) = 134. cost1 = 87.6155777
i = 15 from3(i) = 17. cost1 = 99.781105
i = 16 from3(i) = 68. cost1 = 107.396881
i = 17 from3(i) = 133. cost1 = 125.088684
i = 18 from3(i) = 97. cost1 = 130.088684
i = 19 from3(i) = 140. cost1 = 142.254211
i = 20 from3(i) = 104. cost1 = 146.377319
i = 21 from3(i) = 142. cost1 = 148.613388
i = 22 from3(i) = 75. cost1 = 153.085526
i = 23 from3(i) = 101. cost1 = 156.247803
i = 24 from3(i) = 60. cost1 = 159.41008
i = 25 from3(i) = 154. cost1 = 160.41008
i = 26 from3(i) = 11. cost1 = 166.813202
i = 27 from3(i) = 34. cost1 = 171.813202
i = 28 from3(i) = 14. cost1 = 174.049271
i = 29 from3(i) = 13. cost1 = 178.521408
i = 30 from3(i) = 51. cost1 = 186.767624
i = 31 from3(i) = 33. cost1 = 189.596054
i = 32 from3(i) = 74. cost1 = 197.406311
i = 33 from3(i) = 96. cost1 = 207.456192
i = 34 from3(i) = 110. cost1 = 215.518448
i = 35 from3(i) = 153. cost1 = 219.123993
i = 36 from3(i) = 150. cost1 = 284.644989
i = 37 from3(i) = 1. cost1 = 353.441772
i = 38 from3(i) = 46. cost1 = 361.441772
i = 39 from3(i) = 126. cost1 = 369.687988
i = 40 from3(i) = 183. cost1 = 372.850281
i = 41 from3(i) = 78. cost1 = 378.235443
i = 42 from3(i) = 87. cost1 = 447.762421
i = 43 from3(i) = 1. cost1 = 520.108826
i = 44 from3(i) = 94. cost1 = 524.108826
i = 45 from3(i) = 119. cost1 = 534.108826
i = 46 from3(i) = 121. cost1 = 536.34491
i = 47 from3(i) = 113. cost1 = 544.155151

i = 48 from3(i) = 131. cost1 = 556.196716
i = 49 from3(i) = 82. cost1 = 598.256226
i = 50 from3(i) = 130. cost1 = 607.690186
i = 51 from3(i) = 188. cost1 = 623.814697
i = 52 from3(i) = 61. cost1 = 627.420227
i = 53 from3(i) = 31. cost1 = 630.58252
i = 54 from3(i) = 174. cost1 = 635.967712
i = 55 from3(i) = 95. cost1 = 638.967712
i = 56 from3(i) = 163. cost1 = 643.09082
i = 57 from3(i) = 100. cost1 = 645.326904
i = 58 from3(i) = 71. cost1 = 653.57312
i = 59 from3(i) = 69. cost1 = 656.40155
i = 60 from3(i) = 47. cost1 = 667.218201
i = 61 from3(i) = 115. cost1 = 673.300964
i = 62 from3(i) = 177. cost1 = 683.741272
i = 63 from3(i) = 35. cost1 = 686.903564
i = 64 from3(i) = 158. cost1 = 692.002563
i = 65 from3(i) = 138. cost1 = 695.164856
i = 66 from3(i) = 175. cost1 = 699.164856
i = 67 from3(i) = 165. cost1 = 701.164856
i = 68 from3(i) = 199. cost1 = 705.287964
i = 69 from3(i) = 65. cost1 = 714.507507
i = 70 from3(i) = 132. cost1 = 721.718628
i = 71 from3(i) = 114. cost1 = 728.121765
i = 72 from3(i) = 170. cost1 = 731.727295
i = 73 from3(i) = 58. cost1 = 736.826294
i = 74 from3(i) = 193. cost1 = 741.298401
i = 75 from3(i) = 19. cost1 = 745.298401
i = 76 from3(i) = 116. cost1 = 747.534485
i = 77 from3(i) = 27. cost1 = 748.534485
i = 78 from3(i) = 141. cost1 = 754.191345
i = 79 from3(i) = 124. cost1 = 756.427429
i = 80 from3(i) = 122. cost1 = 812.033203
i = 81 from3(i) = 1. cost1 = 866.439087
i = 82 from3(i) = 112. cost1 = 907.779114
i = 83 from3(i) = 57. cost1 = 918.549438
i = 84 from3(i) = 186. cost1 = 932.971619
i = 85 from3(i) = 54. cost1 = 941.033875
i = 86 from3(i) = 85. cost1 = 946.864807
i = 87 from3(i) = 28. cost1 = 956.298767
i = 88 from3(i) = 181. cost1 = 963.369812
i = 89 from3(i) = 38. cost1 = 972.425171
i = 90 from3(i) = 56. cost1 = 981.425171
i = 91 from3(i) = 86. cost1 = 985.425171
i = 92 from3(i) = 136. cost1 = 994.027466
i = 93 from3(i) = 125. cost1 = 1002.08972
i = 94 from3(i) = 2. cost1 = 1049.51331
i = 95 from3(i) = 1. cost1 = 1059.51331
i = 96 from3(i) = 67. cost1 = 1061.74939
i = 97 from3(i) = 179. cost1 = 1070.35168

i = 98 from3(i) = 118. cost1 = 1072.58777
i = 99 from3(i) = 83. cost1 = 1076.19336
i = 100 from3(i) = 8. cost1 = 1083.40442
i = 101 from3(i) = 18. cost1 = 1104.0199
i = 102 from3(i) = 89. cost1 = 1113.0199
i = 103 from3(i) = 167. cost1 = 1117.0199
i = 104 from3(i) = 39. cost1 = 1121.14294
i = 105 from3(i) = 120. cost1 = 1125.61511
i = 106 from3(i) = 41. cost1 = 1128.61511
i = 107 from3(i) = 168. cost1 = 1139.66052
i = 108 from3(i) = 171. cost1 = 1140.66052
i = 109 from3(i) = 166. cost1 = 1145.66052
i = 110 from3(i) = 59. cost1 = 1166.68433
i = 111 from3(i) = 29. cost1 = 1168.09851
i = 112 from3(i) = 137. cost1 = 1172.34119
i = 113 from3(i) = 189. cost1 = 1175.16956
i = 114 from3(i) = 197. cost1 = 1182.16956
i = 115 from3(i) = 143. cost1 = 1188.49414
i = 116 from3(i) = 12. cost1 = 1190.73022
i = 117 from3(i) = 24. cost1 = 1193.73022
i = 118 from3(i) = 180. cost1 = 1195.96631
i = 119 from3(i) = 90. cost1 = 1199.5719
i = 120 from3(i) = 10. cost1 = 1204.04407
i = 121 from3(i) = 176. cost1 = 1208.16711
i = 122 from3(i) = 148. cost1 = 1210.4032
i = 123 from3(i) = 55. cost1 = 1218.01892
i = 124 from3(i) = 20. cost1 = 1228.01892
i = 125 from3(i) = 15. cost1 = 1233.01892
i = 126 from3(i) = 84. cost1 = 1234.01892
i = 127 from3(i) = 147. cost1 = 1236.255
i = 128 from3(i) = 187. cost1 = 1241.255
i = 129 from3(i) = 185. cost1 = 1246.64014
i = 130 from3(i) = 111. cost1 = 1252.02527
i = 131 from3(i) = 7. cost1 = 1261.08069
i = 132 from3(i) = 149. cost1 = 1321.82605
i = 133 from3(i) = 1. cost1 = 1388.55688
i = 134 from3(i) = 30. cost1 = 1396.17261
i = 135 from3(i) = 3. cost1 = 1397.58679
i = 136 from3(i) = 144. cost1 = 1402.68579
i = 137 from3(i) = 129. cost1 = 1407.68579
i = 138 from3(i) = 127. cost1 = 1414.01038
i = 139 from3(i) = 91. cost1 = 1415.01038
i = 140 from3(i) = 26. cost1 = 1423.07263
i = 141 from3(i) = 49. cost1 = 1424.07263
i = 142 from3(i) = 146. cost1 = 1427.07263
i = 143 from3(i) = 128. cost1 = 1428.48682
i = 144 from3(i) = 72. cost1 = 1430.7229
i = 145 from3(i) = 6. cost1 = 1438.78516
i = 146 from3(i) = 81. cost1 = 1441.94739
i = 147 from3(i) = 159. cost1 = 1442.94739

i = 148 from3(i) = 102. cost1 = 1447.19006
i = 149 from3(i) = 92. cost1 = 1454.47021
i = 150 from3(i) = 107. cost1 = 1458.59326
i = 151 from3(i) = 160. cost1 = 1471.39954
i = 152 from3(i) = 198. cost1 = 1475.00513
i = 153 from3(i) = 162. cost1 = 1482.07617
i = 154 from3(i) = 45. cost1 = 1496.21826
i = 155 from3(i) = 48. cost1 = 1504.76221
i = 156 from3(i) = 196. cost1 = 1508.88525
i = 157 from3(i) = 157. cost1 = 1512.49084
i = 158 from3(i) = 80. cost1 = 1520.49084
i = 159 from3(i) = 4. cost1 = 1526.32178
i = 160 from3(i) = 184. cost1 = 1528.55786
i = 161 from3(i) = 66. cost1 = 1537.77747
i = 162 from3(i) = 36. cost1 = 1545.39319
i = 163 from3(i) = 16. cost1 = 1551.39319
i = 164 from3(i) = 117. cost1 = 1556.39319
i = 165 from3(i) = 108. cost1 = 1577.41699
i = 166 from3(i) = 1. cost1 = 1604.61987
i = 167 from3(i) = 5. cost1 = 1610.45081
i = 168 from3(i) = 156. cost1 = 1633.01184
i = 169 from3(i) = 190. cost1 = 1647.32971
i = 170 from3(i) = 200. cost1 = 1684.32971
i = 171 from3(i) = 22. cost1 = 1687.9353
i = 172 from3(i) = 42. cost1 = 1688.9353
i = 173 from3(i) = 52. cost1 = 1696.74561
i = 174 from3(i) = 201. cost1 = 1716.84534
i = 175 from3(i) = 88. cost1 = 1721.84534
i = 176 from3(i) = 164. cost1 = 1731.2793
i = 177 from3(i) = 169. cost1 = 1732.69348
i = 178 from3(i) = 155. cost1 = 1746.03516
i = 179 from3(i) = 76. cost1 = 1751.86609
i = 180 from3(i) = 173. cost1 = 1755.47168
i = 181 from3(i) = 182. cost1 = 1770.99585
i = 182 from3(i) = 32. cost1 = 1786.29297
i = 183 from3(i) = 98. cost1 = 1793.57312
i = 184 from3(i) = 152. cost1 = 1801.63538
i = 185 from3(i) = 70. cost1 = 1813.03711
i = 186 from3(i) = 44. cost1 = 1817.50928
i = 187 from3(i) = 151. cost1 = 1819.74536
i = 188 from3(i) = 192. cost1 = 1824.21753
i = 189 from3(i) = 23. cost1 = 1827.21753
i = 190 from3(i) = 135. cost1 = 1835.27979
i = 191 from3(i) = 79. cost1 = 1841.36255
i = 192 from3(i) = 9. cost1 = 1846.74768
i = 193 from3(i) = 103. cost1 = 1849.74768
i = 194 from3(i) = 99. cost1 = 1908.29675
i = 195 from3(i) = 1. cost1 = 1941.67334
i = 196 from3(i) = 178. cost1 = 1971.08423
i = 197 from3(i) = 77. cost1 = 2002.46899

i = 198 from3(i) = 40. cost1 = 2048.99854
i = 199 from3(i) = 105. cost1 = 2114.55005
i = 200 from3(i) = 123. cost1 = 2123.98413
i = 201 from3(i) = 37. cost1 = 2137.58569
i = 202 from3(i) = 63. cost1 = 2155.58569
i = 203 from3(i) = 145. cost1 = 2163.20142
i = 204 from3(i) = 43. cost1 = 2185.22412
i = 205 from3(i) = 139. cost1 = 2222.4397
i = 206 from3(i) = 191. cost1 = 2228.09644
i = 207 from3(i) = 195. cost1 = 2240.13794
i = 208 from3(i) = 21. cost1 = 2245.13794
i = 209 from3(i) = 194. cost1 = 2336.04443

Τα αποτελέσματα του 3- opt φάση 1

i = 1 from3(i) = 1. cost1 = 5.83095169
i = 2 from3(i) = 106. cost1 = 8.83095169
i = 3 from3(i) = 53. cost1 = 13.8309517
i = 4 from3(i) = 161. cost1 = 15.8309517
i = 5 from3(i) = 50. cost1 = 22.8309517
i = 6 from3(i) = 93. cost1 = 35.8309517
i = 7 from3(i) = 1. cost1 = 45.6798096
i = 8 from3(i) = 172. cost1 = 46.6798096
i = 9 from3(i) = 64. cost1 = 54.7420654
i = 10 from3(i) = 25. cost1 = 62.8043213
i = 11 from3(i) = 73. cost1 = 68.461174
i = 12 from3(i) = 62. cost1 = 77.5165558
i = 13 from3(i) = 109. cost1 = 82.6155777
i = 14 from3(i) = 134. cost1 = 87.6155777
i = 15 from3(i) = 17. cost1 = 99.781105
i = 16 from3(i) = 68. cost1 = 107.396881
i = 17 from3(i) = 133. cost1 = 125.088684
i = 18 from3(i) = 97. cost1 = 130.088684
i = 19 from3(i) = 140. cost1 = 142.254211
i = 20 from3(i) = 104. cost1 = 146.377319
i = 21 from3(i) = 142. cost1 = 148.613388
i = 22 from3(i) = 75. cost1 = 153.085526
i = 23 from3(i) = 101. cost1 = 156.247803
i = 24 from3(i) = 60. cost1 = 159.41008
i = 25 from3(i) = 154. cost1 = 160.41008
i = 26 from3(i) = 11. cost1 = 166.813202
i = 27 from3(i) = 34. cost1 = 171.813202
i = 28 from3(i) = 14. cost1 = 174.049271
i = 29 from3(i) = 13. cost1 = 178.521408
i = 30 from3(i) = 51. cost1 = 186.767624
i = 31 from3(i) = 33. cost1 = 189.596054
i = 32 from3(i) = 74. cost1 = 197.406311
i = 33 from3(i) = 96. cost1 = 207.456192

i = 34 from3(i) = 110. cost1 = 215.518448
i = 35 from3(i) = 153. cost1 = 219.123993
i = 36 from3(i) = 150. cost1 = 284.644989
i = 37 from3(i) = 1. cost1 = 353.441772
i = 38 from3(i) = 46. cost1 = 361.441772
i = 39 from3(i) = 126. cost1 = 369.687988
i = 40 from3(i) = 183. cost1 = 372.850281
i = 41 from3(i) = 78. cost1 = 378.235443
i = 42 from3(i) = 87. cost1 = 447.762421
i = 43 from3(i) = 1. cost1 = 520.108826
i = 44 from3(i) = 94. cost1 = 524.108826
i = 45 from3(i) = 119. cost1 = 534.108826
i = 46 from3(i) = 121. cost1 = 536.34491
i = 47 from3(i) = 113. cost1 = 544.155151
i = 48 from3(i) = 131. cost1 = 556.196716
i = 49 from3(i) = 82. cost1 = 598.256226
i = 50 from3(i) = 130. cost1 = 607.690186
i = 51 from3(i) = 188. cost1 = 623.814697
i = 52 from3(i) = 61. cost1 = 627.420227
i = 53 from3(i) = 31. cost1 = 630.58252
i = 54 from3(i) = 174. cost1 = 635.967712
i = 55 from3(i) = 95. cost1 = 638.967712
i = 56 from3(i) = 163. cost1 = 643.09082
i = 57 from3(i) = 100. cost1 = 645.326904
i = 58 from3(i) = 71. cost1 = 653.57312
i = 59 from3(i) = 69. cost1 = 656.40155
i = 60 from3(i) = 47. cost1 = 667.218201
i = 61 from3(i) = 115. cost1 = 673.300964
i = 62 from3(i) = 177. cost1 = 683.741272
i = 63 from3(i) = 35. cost1 = 686.903564
i = 64 from3(i) = 158. cost1 = 692.002563
i = 65 from3(i) = 138. cost1 = 695.164856
i = 66 from3(i) = 175. cost1 = 699.164856
i = 67 from3(i) = 165. cost1 = 701.164856
i = 68 from3(i) = 199. cost1 = 705.287964
i = 69 from3(i) = 65. cost1 = 714.507507
i = 70 from3(i) = 132. cost1 = 721.718628
i = 71 from3(i) = 114. cost1 = 728.121765
i = 72 from3(i) = 170. cost1 = 731.727295
i = 73 from3(i) = 58. cost1 = 736.826294
i = 74 from3(i) = 193. cost1 = 741.298401
i = 75 from3(i) = 19. cost1 = 745.298401
i = 76 from3(i) = 116. cost1 = 747.534485
i = 77 from3(i) = 27. cost1 = 748.534485
i = 78 from3(i) = 141. cost1 = 754.191345
i = 79 from3(i) = 124. cost1 = 756.427429
i = 80 from3(i) = 122. cost1 = 812.033203
i = 81 from3(i) = 1. cost1 = 866.439087
i = 82 from3(i) = 112. cost1 = 888.393555
i = 83 from3(i) = 85. cost1 = 912.580322

i = 84 from3(i) = 57. cost1 = 923.350647
i = 85 from3(i) = 186. cost1 = 937.772827
i = 86 from3(i) = 54. cost1 = 942.772827
i = 87 from3(i) = 28. cost1 = 952.206787
i = 88 from3(i) = 181. cost1 = 959.277832
i = 89 from3(i) = 38. cost1 = 968.333191
i = 90 from3(i) = 56. cost1 = 977.333191
i = 91 from3(i) = 86. cost1 = 981.333191
i = 92 from3(i) = 136. cost1 = 989.935547
i = 93 from3(i) = 125. cost1 = 997.997803
i = 94 from3(i) = 2. cost1 = 1045.42139
i = 95 from3(i) = 1. cost1 = 1055.42139
i = 96 from3(i) = 67. cost1 = 1057.65747
i = 97 from3(i) = 179. cost1 = 1066.25977
i = 98 from3(i) = 118. cost1 = 1068.49585
i = 99 from3(i) = 83. cost1 = 1072.10144
i = 100 from3(i) = 8. cost1 = 1079.3125
i = 101 from3(i) = 18. cost1 = 1099.92798
i = 102 from3(i) = 89. cost1 = 1108.92798
i = 103 from3(i) = 167. cost1 = 1112.92798
i = 104 from3(i) = 39. cost1 = 1117.05103
i = 105 from3(i) = 120. cost1 = 1121.52319
i = 106 from3(i) = 41. cost1 = 1124.52319
i = 107 from3(i) = 168. cost1 = 1135.5686
i = 108 from3(i) = 171. cost1 = 1136.5686
i = 109 from3(i) = 166. cost1 = 1141.5686
i = 110 from3(i) = 59. cost1 = 1162.59241
i = 111 from3(i) = 29. cost1 = 1164.00659
i = 112 from3(i) = 137. cost1 = 1168.24927
i = 113 from3(i) = 189. cost1 = 1171.07764
i = 114 from3(i) = 197. cost1 = 1178.07764
i = 115 from3(i) = 143. cost1 = 1184.40222
i = 116 from3(i) = 12. cost1 = 1186.63831
i = 117 from3(i) = 24. cost1 = 1189.63831
i = 118 from3(i) = 180. cost1 = 1191.87439
i = 119 from3(i) = 90. cost1 = 1195.47998
i = 120 from3(i) = 10. cost1 = 1199.95215
i = 121 from3(i) = 176. cost1 = 1204.0752
i = 122 from3(i) = 148. cost1 = 1206.31128
i = 123 from3(i) = 55. cost1 = 1213.927
i = 124 from3(i) = 20. cost1 = 1223.927
i = 125 from3(i) = 15. cost1 = 1228.927
i = 126 from3(i) = 84. cost1 = 1229.927
i = 127 from3(i) = 147. cost1 = 1232.16309
i = 128 from3(i) = 187. cost1 = 1237.16309
i = 129 from3(i) = 185. cost1 = 1242.54822
i = 130 from3(i) = 111. cost1 = 1247.93335
i = 131 from3(i) = 7. cost1 = 1256.98877
i = 132 from3(i) = 149. cost1 = 1317.73413
i = 133 from3(i) = 1. cost1 = 1384.46497

i = 134 from3(i) = 30. cost1 = 1392.08069
i = 135 from3(i) = 3. cost1 = 1393.49487
i = 136 from3(i) = 144. cost1 = 1398.59387
i = 137 from3(i) = 129. cost1 = 1403.59387
i = 138 from3(i) = 127. cost1 = 1409.91846
i = 139 from3(i) = 91. cost1 = 1410.91846
i = 140 from3(i) = 26. cost1 = 1418.98071
i = 141 from3(i) = 49. cost1 = 1419.98071
i = 142 from3(i) = 146. cost1 = 1422.98071
i = 143 from3(i) = 128. cost1 = 1424.3949
i = 144 from3(i) = 72. cost1 = 1426.63098
i = 145 from3(i) = 6. cost1 = 1434.69324
i = 146 from3(i) = 81. cost1 = 1437.85547
i = 147 from3(i) = 159. cost1 = 1438.85547
i = 148 from3(i) = 102. cost1 = 1443.09814
i = 149 from3(i) = 92. cost1 = 1450.3783
i = 150 from3(i) = 107. cost1 = 1454.50134
i = 151 from3(i) = 160. cost1 = 1467.30762
i = 152 from3(i) = 198. cost1 = 1470.91321
i = 153 from3(i) = 162. cost1 = 1477.98425
i = 154 from3(i) = 45. cost1 = 1492.12634
i = 155 from3(i) = 48. cost1 = 1500.67029
i = 156 from3(i) = 196. cost1 = 1504.79333
i = 157 from3(i) = 157. cost1 = 1508.39893
i = 158 from3(i) = 80. cost1 = 1516.39893
i = 159 from3(i) = 4. cost1 = 1522.22986
i = 160 from3(i) = 184. cost1 = 1524.46594
i = 161 from3(i) = 66. cost1 = 1533.68555
i = 162 from3(i) = 36. cost1 = 1541.30127
i = 163 from3(i) = 16. cost1 = 1547.30127
i = 164 from3(i) = 117. cost1 = 1552.30127
i = 165 from3(i) = 108. cost1 = 1573.32507
i = 166 from3(i) = 1. cost1 = 1600.52808
i = 167 from3(i) = 5. cost1 = 1606.35901
i = 168 from3(i) = 156. cost1 = 1628.92004
i = 169 from3(i) = 190. cost1 = 1643.23792
i = 170 from3(i) = 200. cost1 = 1680.23792
i = 171 from3(i) = 22. cost1 = 1683.84351
i = 172 from3(i) = 42. cost1 = 1684.84351
i = 173 from3(i) = 52. cost1 = 1692.65381
i = 174 from3(i) = 201. cost1 = 1712.75354
i = 175 from3(i) = 88. cost1 = 1717.75354
i = 176 from3(i) = 164. cost1 = 1727.1875
i = 177 from3(i) = 169. cost1 = 1728.60168
i = 178 from3(i) = 155. cost1 = 1741.94336
i = 179 from3(i) = 76. cost1 = 1747.77429
i = 180 from3(i) = 173. cost1 = 1751.37988
i = 181 from3(i) = 182. cost1 = 1766.90405
i = 182 from3(i) = 32. cost1 = 1782.20117
i = 183 from3(i) = 98. cost1 = 1789.48132

i = 184 from3(i) = 152. cost1 = 1797.54358
i = 185 from3(i) = 70. cost1 = 1808.94531
i = 186 from3(i) = 44. cost1 = 1813.41748
i = 187 from3(i) = 151. cost1 = 1815.65356
i = 188 from3(i) = 192. cost1 = 1820.12573
i = 189 from3(i) = 23. cost1 = 1823.12573
i = 190 from3(i) = 135. cost1 = 1831.18799
i = 191 from3(i) = 79. cost1 = 1837.27075
i = 192 from3(i) = 9. cost1 = 1842.65588
i = 193 from3(i) = 103. cost1 = 1845.65588
i = 194 from3(i) = 99. cost1 = 1904.20496
i = 195 from3(i) = 1. cost1 = 1962.07544
i = 196 from3(i) = 105. cost1 = 2008.60498
i = 197 from3(i) = 40. cost1 = 2039.98975
i = 198 from3(i) = 77. cost1 = 2069.40063
i = 199 from3(i) = 178. cost1 = 2150.55469
i = 200 from3(i) = 145. cost1 = 2158.17041
i = 201 from3(i) = 43. cost1 = 2180.19312
i = 202 from3(i) = 139. cost1 = 2214.20776
i = 203 from3(i) = 194. cost1 = 2219.20776
i = 204 from3(i) = 21. cost1 = 2231.24927
i = 205 from3(i) = 195. cost1 = 2236.90601
i = 206 from3(i) = 191. cost1 = 2287.06567
i = 207 from3(i) = 63. cost1 = 2300.66724
i = 208 from3(i) = 37. cost1 = 2310.10132
i = 209 from3(i) = 123. cost1 = 2324.66162

Τα αποτελέσματα του 3- opt φάση 2

i = 1 from3(i) = 1. cost1 = 5.83095169
i = 2 from3(i) = 106. cost1 = 8.83095169
i = 3 from3(i) = 53. cost1 = 13.8309517
i = 4 from3(i) = 161. cost1 = 15.8309517
i = 5 from3(i) = 50. cost1 = 22.8309517
i = 6 from3(i) = 93. cost1 = 35.8309517
i = 7 from3(i) = 1. cost1 = 45.6798096
i = 8 from3(i) = 172. cost1 = 46.6798096
i = 9 from3(i) = 64. cost1 = 54.7420654
i = 10 from3(i) = 25. cost1 = 62.8043213
i = 11 from3(i) = 73. cost1 = 68.461174
i = 12 from3(i) = 62. cost1 = 77.5165558
i = 13 from3(i) = 109. cost1 = 82.6155777
i = 14 from3(i) = 134. cost1 = 87.6155777
i = 15 from3(i) = 17. cost1 = 99.781105
i = 16 from3(i) = 68. cost1 = 107.396881
i = 17 from3(i) = 133. cost1 = 125.088684
i = 18 from3(i) = 97. cost1 = 130.088684
i = 19 from3(i) = 140. cost1 = 142.254211

i = 20 from3(i) = 104. cost1 = 146.377319
i = 21 from3(i) = 142. cost1 = 148.613388
i = 22 from3(i) = 75. cost1 = 153.085526
i = 23 from3(i) = 101. cost1 = 156.247803
i = 24 from3(i) = 60. cost1 = 159.41008
i = 25 from3(i) = 154. cost1 = 160.41008
i = 26 from3(i) = 11. cost1 = 166.813202
i = 27 from3(i) = 34. cost1 = 171.813202
i = 28 from3(i) = 14. cost1 = 174.049271
i = 29 from3(i) = 13. cost1 = 178.521408
i = 30 from3(i) = 51. cost1 = 186.767624
i = 31 from3(i) = 33. cost1 = 189.596054
i = 32 from3(i) = 74. cost1 = 197.406311
i = 33 from3(i) = 96. cost1 = 207.456192
i = 34 from3(i) = 110. cost1 = 215.518448
i = 35 from3(i) = 153. cost1 = 219.123993
i = 36 from3(i) = 150. cost1 = 284.644989
i = 37 from3(i) = 1. cost1 = 353.441772
i = 38 from3(i) = 46. cost1 = 361.441772
i = 39 from3(i) = 126. cost1 = 369.687988
i = 40 from3(i) = 183. cost1 = 372.850281
i = 41 from3(i) = 78. cost1 = 378.235443
i = 42 from3(i) = 87. cost1 = 447.762421
i = 43 from3(i) = 1. cost1 = 520.108826
i = 44 from3(i) = 94. cost1 = 524.108826
i = 45 from3(i) = 119. cost1 = 534.108826
i = 46 from3(i) = 121. cost1 = 536.34491
i = 47 from3(i) = 113. cost1 = 544.155151
i = 48 from3(i) = 131. cost1 = 556.196716
i = 49 from3(i) = 82. cost1 = 598.256226
i = 50 from3(i) = 130. cost1 = 607.690186
i = 51 from3(i) = 188. cost1 = 623.814697
i = 52 from3(i) = 61. cost1 = 627.420227
i = 53 from3(i) = 31. cost1 = 630.58252
i = 54 from3(i) = 174. cost1 = 635.967712
i = 55 from3(i) = 95. cost1 = 638.967712
i = 56 from3(i) = 163. cost1 = 643.09082
i = 57 from3(i) = 100. cost1 = 645.326904
i = 58 from3(i) = 71. cost1 = 653.57312
i = 59 from3(i) = 69. cost1 = 656.40155
i = 60 from3(i) = 47. cost1 = 667.218201
i = 61 from3(i) = 115. cost1 = 673.300964
i = 62 from3(i) = 177. cost1 = 683.741272
i = 63 from3(i) = 35. cost1 = 686.903564
i = 64 from3(i) = 158. cost1 = 692.002563
i = 65 from3(i) = 138. cost1 = 695.164856
i = 66 from3(i) = 175. cost1 = 699.164856
i = 67 from3(i) = 165. cost1 = 701.164856
i = 68 from3(i) = 199. cost1 = 705.287964
i = 69 from3(i) = 65. cost1 = 714.507507

i = 70 from3(i) = 132. cost1 = 721.718628
i = 71 from3(i) = 114. cost1 = 728.121765
i = 72 from3(i) = 170. cost1 = 731.727295
i = 73 from3(i) = 58. cost1 = 736.826294
i = 74 from3(i) = 193. cost1 = 741.298401
i = 75 from3(i) = 19. cost1 = 745.298401
i = 76 from3(i) = 116. cost1 = 747.534485
i = 77 from3(i) = 27. cost1 = 748.534485
i = 78 from3(i) = 141. cost1 = 754.191345
i = 79 from3(i) = 124. cost1 = 756.427429
i = 80 from3(i) = 122. cost1 = 812.033203
i = 81 from3(i) = 1. cost1 = 867.836426
i = 82 from3(i) = 86. cost1 = 871.836426
i = 83 from3(i) = 136. cost1 = 881.685303
i = 84 from3(i) = 56. cost1 = 890.740662
i = 85 from3(i) = 38. cost1 = 897.811707
i = 86 from3(i) = 181. cost1 = 908.582031
i = 87 from3(i) = 186. cost1 = 919.352356
i = 88 from3(i) = 57. cost1 = 935.476868
i = 89 from3(i) = 54. cost1 = 940.476868
i = 90 from3(i) = 28. cost1 = 946.3078
i = 91 from3(i) = 85. cost1 = 968.262329
i = 92 from3(i) = 112. cost1 = 973.919189
i = 93 from3(i) = 125. cost1 = 981.981445
i = 94 from3(i) = 2. cost1 = 1029.40503
i = 95 from3(i) = 1. cost1 = 1039.40503
i = 96 from3(i) = 67. cost1 = 1041.64111
i = 97 from3(i) = 179. cost1 = 1050.24341
i = 98 from3(i) = 118. cost1 = 1052.47949
i = 99 from3(i) = 83. cost1 = 1056.08508
i = 100 from3(i) = 8. cost1 = 1063.29614
i = 101 from3(i) = 18. cost1 = 1083.91162
i = 102 from3(i) = 89. cost1 = 1092.91162
i = 103 from3(i) = 167. cost1 = 1096.91162
i = 104 from3(i) = 39. cost1 = 1101.03467
i = 105 from3(i) = 120. cost1 = 1105.50684
i = 106 from3(i) = 41. cost1 = 1108.50684
i = 107 from3(i) = 168. cost1 = 1119.55225
i = 108 from3(i) = 171. cost1 = 1120.55225
i = 109 from3(i) = 166. cost1 = 1125.55225
i = 110 from3(i) = 59. cost1 = 1146.57605
i = 111 from3(i) = 29. cost1 = 1147.99023
i = 112 from3(i) = 137. cost1 = 1152.23291
i = 113 from3(i) = 189. cost1 = 1155.06128
i = 114 from3(i) = 197. cost1 = 1162.06128
i = 115 from3(i) = 143. cost1 = 1168.38586
i = 116 from3(i) = 12. cost1 = 1170.62195
i = 117 from3(i) = 24. cost1 = 1173.62195
i = 118 from3(i) = 180. cost1 = 1175.85803
i = 119 from3(i) = 90. cost1 = 1179.46362

i = 120 from3(i) = 10. cost1 = 1183.93579
i = 121 from3(i) = 176. cost1 = 1188.05884
i = 122 from3(i) = 148. cost1 = 1190.29492
i = 123 from3(i) = 55. cost1 = 1197.91064
i = 124 from3(i) = 20. cost1 = 1207.91064
i = 125 from3(i) = 15. cost1 = 1212.91064
i = 126 from3(i) = 84. cost1 = 1213.91064
i = 127 from3(i) = 147. cost1 = 1216.14673
i = 128 from3(i) = 187. cost1 = 1221.14673
i = 129 from3(i) = 185. cost1 = 1226.53186
i = 130 from3(i) = 111. cost1 = 1231.91699
i = 131 from3(i) = 7. cost1 = 1240.97241
i = 132 from3(i) = 149. cost1 = 1301.71777
i = 133 from3(i) = 1. cost1 = 1368.44861
i = 134 from3(i) = 30. cost1 = 1376.06433
i = 135 from3(i) = 3. cost1 = 1377.47852
i = 136 from3(i) = 144. cost1 = 1382.57751
i = 137 from3(i) = 129. cost1 = 1387.57751
i = 138 from3(i) = 127. cost1 = 1393.9021
i = 139 from3(i) = 91. cost1 = 1394.9021
i = 140 from3(i) = 26. cost1 = 1402.96436
i = 141 from3(i) = 49. cost1 = 1403.96436
i = 142 from3(i) = 146. cost1 = 1406.96436
i = 143 from3(i) = 128. cost1 = 1408.37854
i = 144 from3(i) = 72. cost1 = 1410.61462
i = 145 from3(i) = 6. cost1 = 1418.67688
i = 146 from3(i) = 81. cost1 = 1421.83911
i = 147 from3(i) = 159. cost1 = 1422.83911
i = 148 from3(i) = 102. cost1 = 1427.08179
i = 149 from3(i) = 92. cost1 = 1434.36194
i = 150 from3(i) = 107. cost1 = 1438.48499
i = 151 from3(i) = 160. cost1 = 1451.29126
i = 152 from3(i) = 198. cost1 = 1454.89685
i = 153 from3(i) = 162. cost1 = 1461.9679
i = 154 from3(i) = 45. cost1 = 1476.10999
i = 155 from3(i) = 48. cost1 = 1484.65393
i = 156 from3(i) = 196. cost1 = 1488.77698
i = 157 from3(i) = 157. cost1 = 1492.38257
i = 158 from3(i) = 80. cost1 = 1500.38257
i = 159 from3(i) = 4. cost1 = 1506.2135
i = 160 from3(i) = 184. cost1 = 1508.44958
i = 161 from3(i) = 66. cost1 = 1517.66919
i = 162 from3(i) = 36. cost1 = 1525.28491
i = 163 from3(i) = 16. cost1 = 1531.28491
i = 164 from3(i) = 117. cost1 = 1536.28491
i = 165 from3(i) = 108. cost1 = 1557.30872
i = 166 from3(i) = 1. cost1 = 1584.51172
i = 167 from3(i) = 5. cost1 = 1590.34265
i = 168 from3(i) = 156. cost1 = 1612.90369
i = 169 from3(i) = 190. cost1 = 1627.22156

i = 170 from3(i) = 200. cost1 = 1664.22156
i = 171 from3(i) = 22. cost1 = 1667.82715
i = 172 from3(i) = 42. cost1 = 1668.82715
i = 173 from3(i) = 52. cost1 = 1676.63745
i = 174 from3(i) = 201. cost1 = 1696.73718
i = 175 from3(i) = 88. cost1 = 1701.73718
i = 176 from3(i) = 164. cost1 = 1711.17114
i = 177 from3(i) = 169. cost1 = 1712.58533
i = 178 from3(i) = 155. cost1 = 1725.927
i = 179 from3(i) = 76. cost1 = 1731.75793
i = 180 from3(i) = 173. cost1 = 1735.36353
i = 181 from3(i) = 182. cost1 = 1750.8877
i = 182 from3(i) = 32. cost1 = 1766.18481
i = 183 from3(i) = 98. cost1 = 1773.46497
i = 184 from3(i) = 152. cost1 = 1781.52722
i = 185 from3(i) = 70. cost1 = 1792.92896
i = 186 from3(i) = 44. cost1 = 1797.40112
i = 187 from3(i) = 151. cost1 = 1799.63721
i = 188 from3(i) = 192. cost1 = 1804.10938
i = 189 from3(i) = 23. cost1 = 1807.10938
i = 190 from3(i) = 135. cost1 = 1815.17163
i = 191 from3(i) = 79. cost1 = 1821.25439
i = 192 from3(i) = 9. cost1 = 1826.63953
i = 193 from3(i) = 103. cost1 = 1829.63953
i = 194 from3(i) = 99. cost1 = 1888.1886
i = 195 from3(i) = 1. cost1 = 1946.05908
i = 196 from3(i) = 105. cost1 = 1992.58862
i = 197 from3(i) = 40. cost1 = 2023.97339
i = 198 from3(i) = 77. cost1 = 2053.38428
i = 199 from3(i) = 178. cost1 = 2156.31152
i = 200 from3(i) = 195. cost1 = 2161.96826
i = 201 from3(i) = 191. cost1 = 2178.7312
i = 202 from3(i) = 21. cost1 = 2183.7312
i = 203 from3(i) = 194. cost1 = 2217.74585
i = 204 from3(i) = 139. cost1 = 2239.76855
i = 205 from3(i) = 43. cost1 = 2247.38428
i = 206 from3(i) = 145. cost1 = 2265.38428
i = 207 from3(i) = 63. cost1 = 2278.98584
i = 208 from3(i) = 37. cost1 = 2288.41992
i = 209 from3(i) = 123. cost1 = 2302.98022

Τα αποτελέσματα του 3- opt φάση3

i = 1 from3(i) = 1. cost1 = 5.83095169
i = 2 from3(i) = 106. cost1 = 8.83095169
i = 3 from3(i) = 53. cost1 = 13.8309517
i = 4 from3(i) = 161. cost1 = 15.8309517
i = 5 from3(i) = 50. cost1 = 22.8309517
i = 6 from3(i) = 93. cost1 = 35.8309517
i = 7 from3(i) = 1. cost1 = 45.6798096
i = 8 from3(i) = 172. cost1 = 46.6798096
i = 9 from3(i) = 64. cost1 = 54.7420654
i = 10 from3(i) = 25. cost1 = 62.8043213
i = 11 from3(i) = 73. cost1 = 68.461174
i = 12 from3(i) = 62. cost1 = 77.5165558
i = 13 from3(i) = 109. cost1 = 82.6155777
i = 14 from3(i) = 134. cost1 = 87.6155777
i = 15 from3(i) = 17. cost1 = 99.781105
i = 16 from3(i) = 68. cost1 = 107.396881
i = 17 from3(i) = 133. cost1 = 125.088684
i = 18 from3(i) = 97. cost1 = 130.088684
i = 19 from3(i) = 140. cost1 = 142.254211
i = 20 from3(i) = 104. cost1 = 146.377319
i = 21 from3(i) = 142. cost1 = 148.613388
i = 22 from3(i) = 75. cost1 = 153.085526
i = 23 from3(i) = 101. cost1 = 156.247803
i = 24 from3(i) = 60. cost1 = 159.41008
i = 25 from3(i) = 154. cost1 = 160.41008
i = 26 from3(i) = 11. cost1 = 166.813202
i = 27 from3(i) = 34. cost1 = 171.813202
i = 28 from3(i) = 14. cost1 = 174.049271
i = 29 from3(i) = 13. cost1 = 178.521408
i = 30 from3(i) = 51. cost1 = 186.767624
i = 31 from3(i) = 33. cost1 = 189.596054
i = 32 from3(i) = 74. cost1 = 197.406311
i = 33 from3(i) = 96. cost1 = 207.456192
i = 34 from3(i) = 110. cost1 = 215.518448
i = 35 from3(i) = 153. cost1 = 219.123993
i = 36 from3(i) = 150. cost1 = 284.644989
i = 37 from3(i) = 1. cost1 = 353.441772
i = 38 from3(i) = 46. cost1 = 361.441772
i = 39 from3(i) = 126. cost1 = 369.687988
i = 40 from3(i) = 183. cost1 = 372.850281
i = 41 from3(i) = 78. cost1 = 378.235443
i = 42 from3(i) = 87. cost1 = 447.762421
i = 43 from3(i) = 1. cost1 = 477.376617
i = 44 from3(i) = 61. cost1 = 489.906586
i = 45 from3(i) = 130. cost1 = 527.985474
i = 46 from3(i) = 94. cost1 = 533.370667
i = 47 from3(i) = 82. cost1 = 545.412231

i = 48 from3(i) = 131. cost1 = 553.222473
i = 49 from3(i) = 113. cost1 = 555.458557
i = 50 from3(i) = 121. cost1 = 565.458557
i = 51 from3(i) = 119. cost1 = 610.680237
i = 52 from3(i) = 188. cost1 = 635.700256
i = 53 from3(i) = 35. cost1 = 638.862549
i = 54 from3(i) = 158. cost1 = 643.961548
i = 55 from3(i) = 138. cost1 = 647.12384
i = 56 from3(i) = 175. cost1 = 651.12384
i = 57 from3(i) = 165. cost1 = 653.12384
i = 58 from3(i) = 199. cost1 = 657.246948
i = 59 from3(i) = 65. cost1 = 666.466492
i = 60 from3(i) = 132. cost1 = 673.677612
i = 61 from3(i) = 114. cost1 = 680.08075
i = 62 from3(i) = 170. cost1 = 683.686279
i = 63 from3(i) = 58. cost1 = 688.785278
i = 64 from3(i) = 193. cost1 = 693.257385
i = 65 from3(i) = 19. cost1 = 697.257385
i = 66 from3(i) = 116. cost1 = 699.493469
i = 67 from3(i) = 27. cost1 = 700.493469
i = 68 from3(i) = 141. cost1 = 706.15033
i = 69 from3(i) = 124. cost1 = 708.386414
i = 70 from3(i) = 122. cost1 = 722.528564
i = 71 from3(i) = 177. cost1 = 728.611328
i = 72 from3(i) = 115. cost1 = 739.427979
i = 73 from3(i) = 47. cost1 = 742.256409
i = 74 from3(i) = 69. cost1 = 750.502625
i = 75 from3(i) = 71. cost1 = 752.738708
i = 76 from3(i) = 100. cost1 = 756.861816
i = 77 from3(i) = 163. cost1 = 759.861816
i = 78 from3(i) = 95. cost1 = 765.94458
i = 79 from3(i) = 31. cost1 = 769.106873
i = 80 from3(i) = 174. cost1 = 793.148499
i = 81 from3(i) = 1. cost1 = 846.963013
i = 82 from3(i) = 125. cost1 = 852.619873
i = 83 from3(i) = 112. cost1 = 874.574341
i = 84 from3(i) = 85. cost1 = 880.405273
i = 85 from3(i) = 28. cost1 = 885.405273
i = 86 from3(i) = 54. cost1 = 901.529785
i = 87 from3(i) = 57. cost1 = 912.30011
i = 88 from3(i) = 186. cost1 = 923.070435
i = 89 from3(i) = 181. cost1 = 930.141479
i = 90 from3(i) = 38. cost1 = 939.196838
i = 91 from3(i) = 56. cost1 = 948.196838
i = 92 from3(i) = 86. cost1 = 952.196838
i = 93 from3(i) = 136. cost1 = 958.279602
i = 94 from3(i) = 2. cost1 = 1005.70325
i = 95 from3(i) = 1. cost1 = 1015.70325
i = 96 from3(i) = 67. cost1 = 1017.93933
i = 97 from3(i) = 179. cost1 = 1026.54163

i = 98 from3(i) = 118. cost1 = 1028.77771
i = 99 from3(i) = 83. cost1 = 1032.3833
i = 100 from3(i) = 8. cost1 = 1039.59436
i = 101 from3(i) = 18. cost1 = 1060.20984
i = 102 from3(i) = 89. cost1 = 1069.20984
i = 103 from3(i) = 167. cost1 = 1073.20984
i = 104 from3(i) = 39. cost1 = 1077.33289
i = 105 from3(i) = 120. cost1 = 1081.80505
i = 106 from3(i) = 41. cost1 = 1084.80505
i = 107 from3(i) = 168. cost1 = 1095.85046
i = 108 from3(i) = 171. cost1 = 1096.85046
i = 109 from3(i) = 166. cost1 = 1101.85046
i = 110 from3(i) = 59. cost1 = 1122.87427
i = 111 from3(i) = 29. cost1 = 1124.28845
i = 112 from3(i) = 137. cost1 = 1128.53113
i = 113 from3(i) = 189. cost1 = 1131.3595
i = 114 from3(i) = 197. cost1 = 1138.3595
i = 115 from3(i) = 143. cost1 = 1144.68408
i = 116 from3(i) = 12. cost1 = 1146.92017
i = 117 from3(i) = 24. cost1 = 1149.92017
i = 118 from3(i) = 180. cost1 = 1152.15625
i = 119 from3(i) = 90. cost1 = 1155.76184
i = 120 from3(i) = 10. cost1 = 1160.23401
i = 121 from3(i) = 176. cost1 = 1164.35706
i = 122 from3(i) = 148. cost1 = 1166.59314
i = 123 from3(i) = 55. cost1 = 1174.20886
i = 124 from3(i) = 20. cost1 = 1184.20886
i = 125 from3(i) = 15. cost1 = 1189.20886
i = 126 from3(i) = 84. cost1 = 1190.20886
i = 127 from3(i) = 147. cost1 = 1192.44495
i = 128 from3(i) = 187. cost1 = 1197.44495
i = 129 from3(i) = 185. cost1 = 1202.83008
i = 130 from3(i) = 111. cost1 = 1208.21521
i = 131 from3(i) = 7. cost1 = 1217.27063
i = 132 from3(i) = 149. cost1 = 1278.01599
i = 133 from3(i) = 1. cost1 = 1334.23877
i = 134 from3(i) = 160. cost1 = 1338.36182
i = 135 from3(i) = 107. cost1 = 1352.67969
i = 136 from3(i) = 30. cost1 = 1360.29541
i = 137 from3(i) = 3. cost1 = 1361.70959
i = 138 from3(i) = 144. cost1 = 1366.80859
i = 139 from3(i) = 129. cost1 = 1371.80859
i = 140 from3(i) = 127. cost1 = 1378.13318
i = 141 from3(i) = 91. cost1 = 1379.13318
i = 142 from3(i) = 26. cost1 = 1387.19543
i = 143 from3(i) = 49. cost1 = 1388.19543
i = 144 from3(i) = 146. cost1 = 1391.19543
i = 145 from3(i) = 128. cost1 = 1392.60962
i = 146 from3(i) = 72. cost1 = 1394.8457
i = 147 from3(i) = 6. cost1 = 1402.90796

i = 148 from3(i) = 81. cost1 = 1406.07019
i = 149 from3(i) = 159. cost1 = 1407.07019
i = 150 from3(i) = 102. cost1 = 1411.31287
i = 151 from3(i) = 92. cost1 = 1424.65454
i = 152 from3(i) = 198. cost1 = 1428.26013
i = 153 from3(i) = 162. cost1 = 1435.33118
i = 154 from3(i) = 45. cost1 = 1449.47327
i = 155 from3(i) = 48. cost1 = 1458.01721
i = 156 from3(i) = 196. cost1 = 1462.14026
i = 157 from3(i) = 157. cost1 = 1465.74585
i = 158 from3(i) = 80. cost1 = 1473.74585
i = 159 from3(i) = 4. cost1 = 1479.57678
i = 160 from3(i) = 184. cost1 = 1481.81287
i = 161 from3(i) = 66. cost1 = 1491.03247
i = 162 from3(i) = 36. cost1 = 1498.64819
i = 163 from3(i) = 16. cost1 = 1504.64819
i = 164 from3(i) = 117. cost1 = 1509.64819
i = 165 from3(i) = 108. cost1 = 1530.672
i = 166 from3(i) = 1. cost1 = 1557.875
i = 167 from3(i) = 5. cost1 = 1563.70593
i = 168 from3(i) = 156. cost1 = 1586.26697
i = 169 from3(i) = 190. cost1 = 1600.58484
i = 170 from3(i) = 200. cost1 = 1637.58484
i = 171 from3(i) = 22. cost1 = 1641.19043
i = 172 from3(i) = 42. cost1 = 1642.19043
i = 173 from3(i) = 52. cost1 = 1650.00073
i = 174 from3(i) = 201. cost1 = 1670.10046
i = 175 from3(i) = 88. cost1 = 1675.10046
i = 176 from3(i) = 164. cost1 = 1684.53442
i = 177 from3(i) = 169. cost1 = 1685.94861
i = 178 from3(i) = 155. cost1 = 1699.29028
i = 179 from3(i) = 76. cost1 = 1705.12122
i = 180 from3(i) = 173. cost1 = 1708.72681
i = 181 from3(i) = 182. cost1 = 1724.25098
i = 182 from3(i) = 32. cost1 = 1739.5481
i = 183 from3(i) = 98. cost1 = 1746.82825
i = 184 from3(i) = 152. cost1 = 1754.8905
i = 185 from3(i) = 70. cost1 = 1779.22156
i = 186 from3(i) = 99. cost1 = 1782.22156
i = 187 from3(i) = 103. cost1 = 1787.60669
i = 188 from3(i) = 9. cost1 = 1793.68945
i = 189 from3(i) = 79. cost1 = 1804.68945
i = 190 from3(i) = 44. cost1 = 1809.16162
i = 191 from3(i) = 151. cost1 = 1811.39771
i = 192 from3(i) = 192. cost1 = 1815.86987
i = 193 from3(i) = 23. cost1 = 1818.86987
i = 194 from3(i) = 135. cost1 = 1858.86987
i = 195 from3(i) = 1. cost1 = 1916.74048
i = 196 from3(i) = 105. cost1 = 1963.27002
i = 197 from3(i) = 40. cost1 = 1994.65479

i = 198 from3(i) = 77. cost1 = 2024.06567
i = 199 from3(i) = 178. cost1 = 2071.07642
i = 200 from3(i) = 123. cost1 = 2080.5105
i = 201 from3(i) = 37. cost1 = 2094.11206
i = 202 from3(i) = 63. cost1 = 2112.11206
i = 203 from3(i) = 145. cost1 = 2119.72778
i = 204 from3(i) = 43. cost1 = 2141.75049
i = 205 from3(i) = 139. cost1 = 2175.76514
i = 206 from3(i) = 194. cost1 = 2180.76514
i = 207 from3(i) = 21. cost1 = 2192.80664
i = 208 from3(i) = 195. cost1 = 2198.46338
i = 209 from3(i) = 191. cost1 = 2273.53662

Τα αποτελέσματα του 3- opt φάση 4

i = 1 from3(i) = 1. cost1 = 5.83095169
i = 2 from3(i) = 106. cost1 = 8.83095169
i = 3 from3(i) = 53. cost1 = 13.8309517
i = 4 from3(i) = 161. cost1 = 15.8309517
i = 5 from3(i) = 50. cost1 = 22.8309517
i = 6 from3(i) = 93. cost1 = 35.8309517
i = 7 from3(i) = 1. cost1 = 48.3609161
i = 8 from3(i) = 25. cost1 = 56.423172
i = 9 from3(i) = 73. cost1 = 62.0800247
i = 10 from3(i) = 62. cost1 = 71.1354065
i = 11 from3(i) = 109. cost1 = 76.2344284
i = 12 from3(i) = 134. cost1 = 81.2344284
i = 13 from3(i) = 17. cost1 = 93.3999557
i = 14 from3(i) = 68. cost1 = 101.015732
i = 15 from3(i) = 133. cost1 = 118.707535
i = 16 from3(i) = 97. cost1 = 123.707535
i = 17 from3(i) = 140. cost1 = 135.873062
i = 18 from3(i) = 104. cost1 = 139.99617
i = 19 from3(i) = 142. cost1 = 142.232239
i = 20 from3(i) = 75. cost1 = 146.704376
i = 21 from3(i) = 101. cost1 = 149.866653
i = 22 from3(i) = 60. cost1 = 153.028931
i = 23 from3(i) = 154. cost1 = 154.028931
i = 24 from3(i) = 11. cost1 = 160.432053
i = 25 from3(i) = 34. cost1 = 165.432053
i = 26 from3(i) = 14. cost1 = 167.668121
i = 27 from3(i) = 13. cost1 = 172.140259
i = 28 from3(i) = 51. cost1 = 180.386475
i = 29 from3(i) = 33. cost1 = 183.214905
i = 30 from3(i) = 74. cost1 = 191.025162
i = 31 from3(i) = 96. cost1 = 201.075043
i = 32 from3(i) = 110. cost1 = 209.137299
i = 33 from3(i) = 153. cost1 = 212.742844

i = 34 from3(i) = 150. cost1 = 268.528137
i = 35 from3(i) = 172. cost1 = 269.528137
i = 36 from3(i) = 64. cost1 = 280.298462
i = 37 from3(i) = 1. cost1 = 349.095276
i = 38 from3(i) = 46. cost1 = 357.095276
i = 39 from3(i) = 126. cost1 = 365.341492
i = 40 from3(i) = 183. cost1 = 368.503784
i = 41 from3(i) = 78. cost1 = 373.888947
i = 42 from3(i) = 87. cost1 = 443.415924
i = 43 from3(i) = 1. cost1 = 465.506653
i = 44 from3(i) = 71. cost1 = 467.742706
i = 45 from3(i) = 100. cost1 = 471.865814
i = 46 from3(i) = 163. cost1 = 474.865814
i = 47 from3(i) = 95. cost1 = 480.948578
i = 48 from3(i) = 31. cost1 = 484.11087
i = 49 from3(i) = 174. cost1 = 490.193634
i = 50 from3(i) = 61. cost1 = 502.723602
i = 51 from3(i) = 130. cost1 = 540.80249
i = 52 from3(i) = 94. cost1 = 546.187683
i = 53 from3(i) = 82. cost1 = 558.229248
i = 54 from3(i) = 131. cost1 = 566.03949
i = 55 from3(i) = 113. cost1 = 568.275574
i = 56 from3(i) = 121. cost1 = 578.275574
i = 57 from3(i) = 119. cost1 = 623.497253
i = 58 from3(i) = 188. cost1 = 648.517273
i = 59 from3(i) = 35. cost1 = 651.679565
i = 60 from3(i) = 158. cost1 = 656.778564
i = 61 from3(i) = 138. cost1 = 659.940857
i = 62 from3(i) = 175. cost1 = 663.940857
i = 63 from3(i) = 165. cost1 = 665.940857
i = 64 from3(i) = 199. cost1 = 670.063965
i = 65 from3(i) = 65. cost1 = 679.283508
i = 66 from3(i) = 132. cost1 = 686.494629
i = 67 from3(i) = 114. cost1 = 692.897766
i = 68 from3(i) = 170. cost1 = 696.503296
i = 69 from3(i) = 58. cost1 = 701.602295
i = 70 from3(i) = 193. cost1 = 706.074402
i = 71 from3(i) = 19. cost1 = 710.074402
i = 72 from3(i) = 116. cost1 = 712.310486
i = 73 from3(i) = 27. cost1 = 713.310486
i = 74 from3(i) = 141. cost1 = 718.967346
i = 75 from3(i) = 124. cost1 = 721.20343
i = 76 from3(i) = 122. cost1 = 735.345581
i = 77 from3(i) = 177. cost1 = 741.428345
i = 78 from3(i) = 115. cost1 = 752.244995
i = 79 from3(i) = 47. cost1 = 755.073425
i = 80 from3(i) = 69. cost1 = 781.073425
i = 81 from3(i) = 1. cost1 = 834.887939
i = 82 from3(i) = 125. cost1 = 840.5448
i = 83 from3(i) = 112. cost1 = 862.499268

i = 84 from3(i) = 85. cost1 = 868.3302
i = 85 from3(i) = 28. cost1 = 873.3302
i = 86 from3(i) = 54. cost1 = 889.454712
i = 87 from3(i) = 57. cost1 = 900.225037
i = 88 from3(i) = 186. cost1 = 910.995361
i = 89 from3(i) = 181. cost1 = 918.066406
i = 90 from3(i) = 38. cost1 = 927.121765
i = 91 from3(i) = 56. cost1 = 936.121765
i = 92 from3(i) = 86. cost1 = 940.121765
i = 93 from3(i) = 136. cost1 = 946.204529
i = 94 from3(i) = 2. cost1 = 993.628174
i = 95 from3(i) = 1. cost1 = 1003.62817
i = 96 from3(i) = 67. cost1 = 1005.86426
i = 97 from3(i) = 179. cost1 = 1014.46655
i = 98 from3(i) = 118. cost1 = 1016.70264
i = 99 from3(i) = 83. cost1 = 1020.30817
i = 100 from3(i) = 8. cost1 = 1027.51929
i = 101 from3(i) = 18. cost1 = 1048.13477
i = 102 from3(i) = 89. cost1 = 1057.13477
i = 103 from3(i) = 167. cost1 = 1061.13477
i = 104 from3(i) = 39. cost1 = 1065.25781
i = 105 from3(i) = 120. cost1 = 1069.72998
i = 106 from3(i) = 41. cost1 = 1072.72998
i = 107 from3(i) = 168. cost1 = 1083.77539
i = 108 from3(i) = 171. cost1 = 1084.77539
i = 109 from3(i) = 166. cost1 = 1089.77539
i = 110 from3(i) = 59. cost1 = 1110.79919
i = 111 from3(i) = 29. cost1 = 1112.21338
i = 112 from3(i) = 137. cost1 = 1116.45605
i = 113 from3(i) = 189. cost1 = 1119.28442
i = 114 from3(i) = 197. cost1 = 1126.28442
i = 115 from3(i) = 143. cost1 = 1132.60901
i = 116 from3(i) = 12. cost1 = 1134.84509
i = 117 from3(i) = 24. cost1 = 1137.84509
i = 118 from3(i) = 180. cost1 = 1140.08118
i = 119 from3(i) = 90. cost1 = 1143.68677
i = 120 from3(i) = 10. cost1 = 1148.15894
i = 121 from3(i) = 176. cost1 = 1152.28198
i = 122 from3(i) = 148. cost1 = 1154.51807
i = 123 from3(i) = 55. cost1 = 1162.13379
i = 124 from3(i) = 20. cost1 = 1172.13379
i = 125 from3(i) = 15. cost1 = 1177.13379
i = 126 from3(i) = 84. cost1 = 1178.13379
i = 127 from3(i) = 147. cost1 = 1180.36987
i = 128 from3(i) = 187. cost1 = 1185.36987
i = 129 from3(i) = 185. cost1 = 1190.755
i = 130 from3(i) = 111. cost1 = 1196.14014
i = 131 from3(i) = 7. cost1 = 1205.19556
i = 132 from3(i) = 149. cost1 = 1265.94092
i = 133 from3(i) = 1. cost1 = 1322.1637

i = 134 from3(i) = 160. cost1 = 1326.28674
i = 135 from3(i) = 107. cost1 = 1340.60461
i = 136 from3(i) = 30. cost1 = 1348.22034
i = 137 from3(i) = 3. cost1 = 1349.63452
i = 138 from3(i) = 144. cost1 = 1354.73352
i = 139 from3(i) = 129. cost1 = 1359.73352
i = 140 from3(i) = 127. cost1 = 1366.05811
i = 141 from3(i) = 91. cost1 = 1367.05811
i = 142 from3(i) = 26. cost1 = 1375.12036
i = 143 from3(i) = 49. cost1 = 1376.12036
i = 144 from3(i) = 146. cost1 = 1379.12036
i = 145 from3(i) = 128. cost1 = 1380.53455
i = 146 from3(i) = 72. cost1 = 1382.77063
i = 147 from3(i) = 6. cost1 = 1390.83289
i = 148 from3(i) = 81. cost1 = 1393.99512
i = 149 from3(i) = 159. cost1 = 1394.99512
i = 150 from3(i) = 102. cost1 = 1399.23779
i = 151 from3(i) = 92. cost1 = 1412.57947
i = 152 from3(i) = 198. cost1 = 1416.18506
i = 153 from3(i) = 162. cost1 = 1423.2561
i = 154 from3(i) = 45. cost1 = 1437.39819
i = 155 from3(i) = 48. cost1 = 1445.94214
i = 156 from3(i) = 196. cost1 = 1450.06519
i = 157 from3(i) = 157. cost1 = 1453.67078
i = 158 from3(i) = 80. cost1 = 1461.67078
i = 159 from3(i) = 4. cost1 = 1467.50171
i = 160 from3(i) = 184. cost1 = 1469.73779
i = 161 from3(i) = 66. cost1 = 1478.9574
i = 162 from3(i) = 36. cost1 = 1486.57312
i = 163 from3(i) = 16. cost1 = 1492.57312
i = 164 from3(i) = 117. cost1 = 1497.57312
i = 165 from3(i) = 108. cost1 = 1518.59692
i = 166 from3(i) = 1. cost1 = 1545.7998
i = 167 from3(i) = 5. cost1 = 1551.63074
i = 168 from3(i) = 156. cost1 = 1574.19177
i = 169 from3(i) = 190. cost1 = 1588.50964
i = 170 from3(i) = 200. cost1 = 1625.50964
i = 171 from3(i) = 22. cost1 = 1629.11523
i = 172 from3(i) = 42. cost1 = 1630.11523
i = 173 from3(i) = 52. cost1 = 1637.92554
i = 174 from3(i) = 201. cost1 = 1658.02527
i = 175 from3(i) = 88. cost1 = 1663.02527
i = 176 from3(i) = 164. cost1 = 1672.45923
i = 177 from3(i) = 169. cost1 = 1673.87341
i = 178 from3(i) = 155. cost1 = 1687.21509
i = 179 from3(i) = 76. cost1 = 1693.04602
i = 180 from3(i) = 173. cost1 = 1696.65161
i = 181 from3(i) = 182. cost1 = 1712.17578
i = 182 from3(i) = 32. cost1 = 1727.4729
i = 183 from3(i) = 98. cost1 = 1734.75305

i = 184 from3(i) = 152. cost1 = 1742.81531
i = 185 from3(i) = 70. cost1 = 1754.21704
i = 186 from3(i) = 44. cost1 = 1758.68921
i = 187 from3(i) = 151. cost1 = 1772.90186
i = 188 from3(i) = 99. cost1 = 1775.90186
i = 189 from3(i) = 103. cost1 = 1781.28699
i = 190 from3(i) = 9. cost1 = 1787.36975
i = 191 from3(i) = 79. cost1 = 1793.36975
i = 192 from3(i) = 192. cost1 = 1797.84192
i = 193 from3(i) = 23. cost1 = 1800.84192
i = 194 from3(i) = 135. cost1 = 1840.84192
i = 195 from3(i) = 1. cost1 = 1855.4021
i = 196 from3(i) = 123. cost1 = 1864.83606
i = 197 from3(i) = 37. cost1 = 1878.4375
i = 198 from3(i) = 63. cost1 = 1896.4375
i = 199 from3(i) = 145. cost1 = 1904.05322
i = 200 from3(i) = 43. cost1 = 1926.07593
i = 201 from3(i) = 139. cost1 = 1960.09058
i = 202 from3(i) = 194. cost1 = 1965.09058
i = 203 from3(i) = 21. cost1 = 1977.1322
i = 204 from3(i) = 195. cost1 = 1982.78906
i = 205 from3(i) = 191. cost1 = 2110.73047
i = 206 from3(i) = 105. cost1 = 2157.26001
i = 207 from3(i) = 40. cost1 = 2188.64478
i = 208 from3(i) = 77. cost1 = 2218.05566
i = 209 from3(i) = 178. cost1 = 2251.43237

Βιβλιογραφία

Άρθρα - Βιβλία

[1] E. Aarts and J.K. Lentra. Local Search in Combinatorial Optimization. Wiley and Sons, 1997.
[2] L.D. Bodin, B.L. Golcen, A.A. Assad, and Michael O. Ball. Routing and Scheduling of vehicles and crews. Computers and Operations Research, 10:63-211, 1983.
[3] N. Christofides. Vehicle routing. In E.L. Lawer, J.K. Lenstra, A.H.G. Rinnoy Kan, and D.B. Shmoys, editors, The Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization, pages 431-448. Wiley and Sons, 1985.
[4] M.Gendrau, G. Laporte, and J.Y. Potvin. Vehicle routing : Modern heuristics. In E. Aarts and J.K. Lenstra, editors, Local Search in Combinatorial Optimization, pages 311-336. Wiley and Sons, 1997.
[5] G. Kontoravdis and J.F. Bard. A grasp for the vehicle routing problem with time windows. ORSA Journal on Computing, 7 (1): 10-23, 1995.
[6] G. Laporte, M. Gendreau, J.Y. Potvin, and F. Semet. Classical and modern heuristics for the vehicle routing problem. International Transactions in Operational Research, 7: 285-300, 2000.
[7] G. Mosheiov. Vehicle routing with pickup and delivery: Tour- partitioning heuristics. Computers and Industrial Engineering, 34: 669- 684, 1998.
[8] K.S. Ruland and E.Y. Robin. The pickup ane delivery problem: Faces and Branch-and- Cut algorithm. Computers Mathematical Applications, 33 (12): 1- 13, 1997.
[9] S.R Thangiah, J.Y. Potvin, and T. Sung. Heuristics approaches to vehicle routing with backhauls and time windows. Computers and Operations Research, 23: 1043-1057, 1996.
[10] Αθανάσιος Μυγδαλάς, Ιωάννης Μαρινάκης, Αθανασία Μαυρομάτη. Σημειώσεις μαθήματος Διαχείρησης Εφοδιαστικής αλυσίδας.
[11] Αθανάσιος Μυγδαλάς, Ιωάννης Μαρινάκης. Σημειώσεις μαθήματος Συνδυαστικής Βελτιστοποίησης.
[12] Ζαφειρίδης Χρήστος. Διπλωματική εργασία με θέμα: «Βελτιστοποίηση διανομής προϊόντων κεντρικής θέρμανσης και κλιματισμού στην περιοχή της δυτικής Μακεδονίας».
[13] Καρακιοζόπουλος Κωνσταντίνος. Διπλωματική εργασία με θέμα: «Μελέτη και Βελτιστοποίηση των Διανομών των προϊόντων της εταιρίας Μπισκότα Α.Β.Ε.Ε».
[14] Θεοδωρίδου Ανατολή. Διπλωματική εργασία με θέμα «Βελτιστοποίηση διανομής φακέλων και δεμάτων της εταιρίας ACS στα Χανιά».
[15] Athanasios Migdalas, Giannis Marinakis. Heuristics Solutions of Vehicle Routing Problems in Supply Chain Management In Combinatorial and Global Optimization, pp.205- 236.

http://neo.lcc.uma.es/radi-aeb/WebVRP/index.html?/Problem%20Instances/instances