

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ
ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ



Διπλωματική εργασία

Το Πρόβλημα δρομολόγησης με παραλαβές
και παραδόσεις με χρήση του αλγορίθμου
Περιορισμένης Αναζήτησης

Γονιδάκης Ιωάννης

Επιβλέπων Καθηγητής
Ιωάννης Μαρινάκης

30/6/2014
ΧΑΝΙΑ 2014

Ευχαριστίες

Ολοκληρώνοντας την εργασία μου θα ήθελα να ευχαριστήσω τους ανθρώπους που με βοήθησαν να βρίσκομαι στην ευχάριστη αυτή θέση.

Αρχικά, χάρη στην πολύτιμη βοήθεια του υπευθύνου καθηγητή κ. Μαρινάκη ξεκίνησα και τελικά ολοκλήρωσα μια διπλωματική εργασία η οποία με εισήγαγε και πρακτικά πλέον στην έννοια της εφοδιαστικής αλυσίδας, θα ήθελα λοιπόν να τον ευχαριστήσω για την βοήθεια τις επισημάνσεις καθώς και την καθοδήγηση στο έργο που είχα να εκπονήσω.

Επιπλέον, Ευχαριστώ όλους τους καθηγητές που μου έδωσαν τα κατάλληλα εφόδια για αυτόν τον τόσο αξιόλογο επαγγελματικό κλάδο που σπανίζει στην ελληνική τριτοβάθμια εκπαίδευση.

Τέλος, το σημαντικότερο στήριγμα σε όλη την προσπάθεια μου ήταν η οικογένεια μου στην οποία χρωστάω το μεγαλύτερο ευχαριστώ.

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1.Εισαγωγή.....	7
1.1 Περίληψη εργασίας.....	8
1.2Εισαγωγή στην εφοδιαστική αλυσίδα και τα logistics.....	8
ΚΕΦΑΛΑΙΟ 2. Το πρόβλημα δρομολόγησης με παραλαβές και παραδώσεις.....	11
2.1 Το πρόβλημα δρομολόγησης.....	12
2.2 Το πρόβλημα δρομολόγησης με παραλαβές και παραδώσεις.....	13
2.2.1 Μαθηματική απεικόνιση.....	13
2.2.1.1 Μοντελοποίηση.....	13
2.2.1.2 Αντικειμενική συνάρτηση.....	14
ΚΕΦΑΛΑΙΟ 3. Ο αλγόριθμος της περιορισμένης αναζήτησης.....	17
3.1 Μεθευριτικοί αλγόριθμοι.....	18
3.2 Ο αλγόριθμος της περιορισμένης αναζήτησης.....	19
3.2.1 Ανάλυση Αλγορίθμου.....	20
3.2.2 Μακροπρόθεσμες μνήμες.....	21
3.2.3 Ψευδοκώδικας.....	22

ΚΕΦΑΛΑΙΟ 4. Εφαρμογή στον κώδικα.....	25
4.1 Περιβάλλον επίλυσης.....	26
4.2 Δεδομένα.....	27
4.3 Κώδικας.....	28
4.3.1 Ανάλυση αλγορίθμου.....	29
4.3.2 Συναρτήσεις.....	29
4.3.2.1 Data.....	29
4.3.2.2 main.m.....	30
4.3.2.3 Kritfilod.....	31
4.3.2.4 Endiamesh_mesaia_mnhmh.m και Endiamesh_megalh_mnhmh.m.....	32
4.3.2.5 Mesaia_mnhmh.m.....	32
4.3.2.6 Megalh_mnhmh.m.....	33
4.3.2.7 Map.m.....	33
ΚΕΦΑΛΑΙΟ 5. Επίλυση.....	35
5.1 Αποτελέσματα.....	36
Α. προβλήματα τύπου eil με 51 κόμβους.....	36
Β. προβλήματα τύπου eil με 76 κόμβους.....	41
Γ. προβλήματα τύπου eil με 101 κόμβους.....	45
5.2 Ανάλυση αποτελεσμάτων και συμπεράσματα.....	49
Βιβλιογραφία.....	50

ΚΕΦΑΛΑΙΟ 1ο

ΕΙΣΑΓΩΓΗ

1.1 ΠΕΡΙΛΗΨΗ ΕΡΓΑΣΙΑΣ

Η εργασία έχει ως θέμα τη μελέτη του Προβλήματος δρομολόγησης με παραλαβές και παραδόσεις και τη δημιουργία κατάλληλου κώδικα σε προγραμματιστικό περιβάλλον matlab, που το επιλύει με τη χρήση του μεθευρετικού αλγορίθμου περιορισμένης αναζήτησης. Αρχικά παρουσιάζεται το πρόβλημα και γίνεται η μαθηματική του μοντελοποίηση.

Έπειτα γίνεται αναφορά στον αλγόριθμο περιορισμένης αναζήτησης, γίνεται η ιστορική του αναδρομή και περιγράφεται ο τρόπος με τον οποίο λειτουργεί. Στη συνέχεια, γίνεται η μαθηματική του εφαρμογή στο πρόβλημα της δρομολόγησης, η οποία θα χρησιμοποιηθεί στο πρόγραμμα επίλυσης. Το πρόγραμμα παρουσιάζεται και αναλύεται ως προς τη δομή του και ως προς τον κώδικα. Τέλος, παρουσιάζονται και αξιολογούνται τα αποτελέσματα.

1.2 Εισαγωγή στα logistics και την εφοδιαστική αλυσίδα

Η παρούσα διπλωματική εργασία έχει ως γενικότερο αντικείμενο την βελτιστοποίηση της εφοδιαστικής αλυσίδας. Είναι βασικό λοιπόν για αρχή να έχουμε κάποιους ορισμούς τόσο για την εφοδιαστική αλυσίδα όσο και για τα Logistics που έχουν βασικό ρολό σε κάθε τέτοιο δίκτυο. Logistics, λοιπόν, είναι η λειτουργία της επιχείρησης που ασχολείται με το σύνολο των δραστηριοτήτων για την παραγωγή, την εξασφάλιση και την διαθεσιμότητα όλων των ανθρώπινων και φυσικών πόρων, που είναι απαραίτητη για τη σωστή λειτουργία των διαδικασιών ενός συστήματος.

Ο πλέον ορισμός των logistics, όπως διατυπώθηκε το 1991 από το Council of Logistics Management των ΗΠΑ, είναι ο ακόλουθος: Logistics είναι η διαδικασία σχεδιασμού, εφαρμογής και ελέγχου μιας αποτελεσματικής και αποδοτικής ροής και αποθήκευσης αγαθών, υπηρεσιών και σχετικών πληροφοριών, από το σημείο παραγωγής στο σημείο κατανάλωσης, με σκοπό την ικανοποίηση των αναγκών και

των απαιτήσεων του πελάτη. Χαρακτηριστικός είναι ο ορισμός που έχει υιοθετήσει το Institute of Logistics της Μεγάλης Βρετανίας “Logistics is the management of the supply chain”, τα Logistics είναι η οργάνωση και η διαχείριση της εφοδιαστικής αλυσίδας. Ο ορισμός αυτός θεωρείται από πολλούς συγγραφείς θεμελιώδεις για την κατανόηση του όρου logistics. Από αυτόν τον ορισμό κατανοούμε επίσης ότι οι διαδικασίες των Logistics αποτελούν ένα τμήμα του συνόλου που λέγεται “Εφοδιαστική Αλυσίδα”.

Διακρίνουμε δύο κατηγορίες στοιχείων που απαρτίζουν αντίστοιχα τα logistics και την εκτέλεση αυτών. Η πρώτη κατηγορία αναφέρεται στα στάδια των logistics, δηλαδή στον σχεδιασμό, τον έλεγχο και την εκτέλεση. Η δεύτερη κατηγορία αναφέρεται στα μέρη της εκτέλεσης, Όπου συναντάμε τα στοιχεία των αγορών, της αποθήκευσης, της διαχείρισης των αποθεμάτων, της μεταφοράς και των διανομών.

Εφοδιαστική αλυσίδα

Ο όρος εφοδιαστική αλυσίδα(SUPPLY CHAIN) αναφέρεται σε ένα ολοκληρωμένο δίκτυο λειτουργιών που αφορά όλη τη διαδρομή του προϊόντος, από την πρώτη ύλη, την μεταποίηση του, την διανομή του και την πώληση του στον καταναλωτή. Η ΔΕΑ (Διαχείριση Εφοδιαστικής Αλυσίδας) αναφέρεται στο σχεδιασμό και τη διαχείριση όλων των ενεργειών-δραστηριοτήτων που σχετίζονται με τις διαδικασίες προμήθειας, την παραγωγή-μεταποίηση και όλες τις δραστηριότητες της διανομής. Επιπλέον, περιλαμβάνει το συντονισμό και τη συνεργασία με όλους τους εταίρους του δικτύου εφοδιασμού, που μπορεί να είναι προμηθευτές, ενδιαμέσοι κρίκοι, εταιρείες παροχής υπηρεσιών Third Party Logistics (3PL) και πελάτες. Κατ’ ουσία, η Διαχείριση Εφοδιαστικής Αλυσίδας ενοποιεί και ολοκληρώνει το σχεδιασμό, τις προμήθειες, την παραγωγή, την αποθήκευση, τη μεταφορά και τις πωλήσεις τόσο μέσα στις επιχειρήσεις όσο και μεταξύ αυτών.

Ο αντικειμενικός λοιπόν σκοπός της ΔΕΑ είναι η αύξηση της συνολικής κερδοφορίας κατά μήκος της αλυσίδας που συνεπάγεται την αύξηση της κερδοφορίας όλων των εταίρων της. Αυτό επιτυγχάνεται με την κατανόηση και ικανοποίηση των πελατειακών αναγκών στον απαιτούμενο χρόνο, και με την προσφορά προϊόντων υψηλής προστιθέμενης αξίας και ανταγωνιστικού κόστους. Για την επίτευξη των παραπάνω στόχων, απαραίτητα χαρακτηριστικά των εφοδιαστικών αλυσίδων που ανταγωνίζονται μέσα στο σύγχρονο παγκοσμιοποιημένο περιβάλλον είναι η ευελιξία και η ταχεία προσαρμοστικότητα τους στις δυναμικά μεταβαλλόμενες συνθήκες. Με άλλα λόγια η εφοδιαστική αλυσίδα αποτελείται από πολλά logistics. Βλέπουμε λοιπόν ότι τα Logistics και η Εφοδιαστική Αλυσίδα συνδέονται άμεσα, καθώς η εφοδιαστική αλυσίδα αποτελεί το βασικότερο πεδίο εφαρμογής των Logistics, τα οποία αποτελούν με τη σειρά τους

το βασικότερο μέσο για την αποτελεσματική οργάνωση και διαχείριση της Εφοδιαστικής Αλυσίδας.

ΚΕΦΑΛΑΙΟ 2ο

**Το πρόβλημα δρομολόγησης με παραλαβές
και παραδώσεις (VRPPD)**

2.1 Το πρόβλημα δρομολόγησης οχημάτων

Το πρόβλημα δρομολόγησης οχημάτων είναι ένα από τα σημαντικότερα προβλήματα βελτιστοποίησης και αποτελεί την ανάγκη για ελαχιστοποίηση του συνολικού μήκους διαδρομών που θα διασχίσει ένας στόλος οχημάτων ώστε να ικανοποιήσει τις ανάγκες των πελατών. Το πρόβλημα της δρομολόγησης είναι ο σημαντικότερος τομέας μιας εφοδιαστικής αλυσίδας με πληθώρα εφαρμογών του τόσο στον τομέα των επιχειρήσεων όσο και στην καθημερινή ζωή.

Το πρόβλημα δρομολόγησης παρουσιάστηκε για πρώτη φορά το 1959 από τους Dantzig και Ramser με στόχο την βελτιστοποίηση της διακίνησης βενζίνης. πέντε χρόνια αργότερα το 1964 οι εφάρμοσαν στο πρόβλημα έναν άπληστο μεθευρετικό αλγόριθμο που έφερε καλύτερα αποτελέσματα. ακλούθησαν από το τότε εκατοντάδες νέες προτάσεις επίλυσης αλλά και πολλές υποκατηγορίες του προβλήματος ανάλογα με τις ανάγκες του περιορισμούς και το προς βελτιστοποίηση ζητούμενο. Κάποιες από τις σημαντικότερες κατηγορίες παρατίθενται παρακάτω.

- Το πρόβλημα δρομολόγησης οχημάτων παραλαβές (Vehicle Routing Problem with Backhauls)
- Το πρόβλημα δρομολόγησης οχημάτων με Χρονικά διαστήματα (Vehicle Routing Problem with Time Windows).
- Το πρόβλημα δρομολόγησης με παραλαβές και παραδώσεις (Vehicle Routing Problem with Pickup and Delivery).
- Το πρόβλημα δρομολόγησης οχημάτων με Χρονικά διαστήματα με παραλαβές και παραδώσεις (Vehicle Routing Problem with Time Windows and Pickup and Delivery).
- Το στοχαστικό πρόβλημα δρομολόγησης οχημάτων (Stochastic Vehicle Routing Problem)
- Το δυναμικό πρόβλημα δρομολόγησης οχημάτων (Dynamic Vehicle Routing Problem)

Η συγκεκριμένη εργασία ασχολείται με το πρόβλημα δρομολόγησης με παραλαβές και παραδώσεις.

2.2 Το πρόβλημα δρομολόγησης με παραλαβές και παραδώσεις

Το πρόβλημα δρομολόγησης με παραλαβές και παραδώσεις αποτελεί μια επέκταση του κλασσικού προβλήματος δρομολόγησης που περιλαμβάνει πελάτες τόσο με θετική όσο και αρνητική ζήτηση. Δεδομένα του συγκεκριμένου προβλήματος είναι ο αριθμός μεταφορικών οχημάτων, η χωρητικότητά τους, ο αριθμός των πελατών k , η θέση τους στον χάρτη και η ζήτηση τους.

Τα κύρια χαρακτηριστικά του προβλήματος είναι τα ακόλουθα:

- Κάθε κύκλος περνά από την αποθήκη
- Κάθε πελάτης επισκέπτεται από μόνο ένα κύκλο.
- Κάθε όχημα αντιστοιχεί μόνο σε μια διαδρομή.
- Η συνολική που μεταφέρει ένα όχημα πρέπει να είναι μη αρνητική και να μην ξεπερνά την χωρητικότητα του οχήματος

Στο συγκεκριμένο πρόβλημα υπάρχουν πολλές υποκατηγορίες και εφαρμογές που εξαρτώνται από χαρακτηριστικά όπως ο αριθμός των διαφορετικών εμπορευμάτων και οχημάτων. Στην συγκεκριμένη εργασία τα προβλήματα που θα επιλύσουμε ανήκουν στην κατηγορία 1-PDTSP όπου υπάρχει ένα είδος προϊόντων. Γενίκευση του παραπάνω προβλήματος αποτελεί η κατηγορία m -PDTSP όπου αντιμετωπίζουμε m διαφορετικά προϊόντα. Συνεπώς οι πελάτες δεν μπορούν να χωριστούν σε κατηγορίες για παραλαβή ή παράδοση διότι πιθανώς να ανήκουν και στις δυο κατηγορίες για διαφορετικά προϊόντα. Το πρόβλημα στο οποίο μπορούμε να χρησιμοποιήσουμε αυτόν τον διαχωρισμό ονομάζεται ένα-προς-ένα m -PDTSP όπου το κάθε προϊόν έχει μια πηγή και έναν προορισμό. [1]

Μαθηματική απεικόνιση

Έστω πρόβλημα δρομολόγησης (1-PDTSP) με γράφημα $G = (V, A)$ όπου G είναι το σύνολο των κόμβων και A το σύνολο των τόξων. Ο πρώτος κόμβος αποτελεί και την αποθήκη, από την οποία ξεκίνα και στην οποία καταλήγει ο στόλος των οχημάτων με χωρητικότητα C , και τους υπόλοιπους κόμβους- πελάτες που πρέπει να εξυπηρετηθούν. Σε κάθε κόμβο- πελάτη αντιστοιχεί μια συγκεκριμένη ζήτηση d_i , ενώ σε κάθε τόξο A αντιστοιχεί και ένα κόστος μετακίνησης. Στόχος μας είναι η ελαχιστοποίηση του κόστους, δηλαδή της συνολικής απόστασης που θα διανύσει το

όχημα από την αποθήκη στους κόμβους- πελάτες και μετά ξανά πίσω στην αποθήκη.

Μοντελοποίηση

Σταθερές

K = αριθμός οχημάτων

N = αριθμός πελατών για παράδοση

M = αριθμός πελατών για παραλαβή

a_i = ζήτηση πελάτη προς παράδοση

b_i = φορτίο πελάτη για παραλαβή

C = χωρητικότητα οχήματος

c_{ij} = κόστος μετακίνησης από τον πελάτη i στον πελάτη j

Μεταβλητές

$$u_{ik} = \begin{cases} 1 & \text{αν πελατης προς παραδοση εξυπηρειται απο οχημα } k \\ 0 & \text{σε καθε αλλη περιπτωση} \end{cases}$$

$$v_{jk} = \begin{cases} 1 & \text{αν πελατης προς παραλαβη εξυπηρειται απο οχημα } k \\ 0 & \text{σε καθε αλλη περιπτωση} \end{cases}$$

$$x_{ijk} = \begin{cases} 1 & \text{αν οχημα } k \text{ μετακινηται απευθιας απο πελατη } i \text{ σε πελατη } j \\ 0 & \text{σε καθε αλλη περιπτωση} \end{cases}$$

$$S = \{\text{όλα τα } x_{ijk}\}$$

Αντικειμενική συνάρτηση

$$\sum_{k=1}^K \sum_{i=0}^{N+M} \sum_{j=0}^{N+M} c_{ij} x_{ijk}$$

Υπό τους περιορισμούς

$$\sum_{k=1}^N a_i u_{ik} \leq C, \quad k=1,2,\dots,K \quad (1)$$

$$\sum_{k=1}^K u_{ik} = 1, \quad i=1,2,\dots,n \quad (2)$$

$$u_{0k}=1, \quad k=1,2,\dots,K \quad (3)$$

$$u_{ik}=0 \text{ ή } 1, \quad i=1,2,\dots,N, \quad k=1,2,\dots,K \quad (4)$$

$$\sum_{i=N+1}^{N+M} b_i u_{ik} \leq C, \quad k=1,2,\dots,K \quad (5)$$

$$\sum_{k=1}^K u_{ik} = 1, \quad i=N+1,N+2,\dots,N+M \quad (6)$$

$$u_{0k}=1, \quad k=1,2,\dots,K \quad (7)$$

$$u_{ik}=0 \text{ ή } 1, \quad i=N+1,N+2,\dots,N+M, \quad k=1,2,\dots,K \quad (8)$$

$$\sum_{i=0}^{N+M} x_{ijk} = \begin{cases} u_{jk} & \text{αν } j = 1 \dots N \\ v_{jk} & \text{αν } j = N+1 \dots N+M \text{ και } j = 0 \end{cases} \quad (9)$$

$$\sum_{j=0}^{N+M} x_{ijk} = \begin{cases} u_{ik} & \text{αν } i = 1 \dots N \\ v_{ik} & \text{αν } i = N+1 \dots N+M \text{ και } j = 0 \end{cases} \quad (10)$$

$$\sum_{i=0}^N \sum_{\substack{j=N+1 \\ \text{και } j=0}}^{N+M} x_{ijk} = 1, \quad k=1,2,\dots,K \quad (11)$$

$$x_{ijk} \in S \quad (12)$$

[3],[4]

ΚΕΦΑΛΑΙΟ 3ο

Περιορισμένη Αναζήτηση (Tabu Search)

3.1 Εισαγωγή στους Μεθευρετικούς Αλγόριθμους

Οι μεθευρετικοί αλγόριθμοι είναι μέθοδοι επίλυσης που συνδυάζουν διαδικασίες τοπικής αναζήτησης και υψηλότερου επιπέδου στρατηγικές για να δημιουργήσουν μια διαδικασία που είναι ικανή να ξεφύγει από κάποιο τοπικό ελάχιστο. Τα τελευταία χρόνια οι περισσότεροι αλγόριθμοι που έχουν αναπτυχθεί στο πεδίο της συνδυαστικής βελτιστοποίησης ανήκουν στη συγκεκριμένη κατηγορία. Τα δυο κύρια συστατικά κάθε μεθευρετικού αλγορίθμου είναι : η επίταση (intensification) και η διαφοροποίηση (diversification) ή εκμετάλλευση (exploitation) και εξερεύνηση (exploration).

Η **επίταση** είναι η εστίαση στην τοπική αναζήτηση εκμεταλλευόμενοι την πληροφορία ότι βρίσκουμε μια καλή λύση σε αυτήν την περιοχή. Αντιθέτως η **διαφοροποίηση** σημαίνει η εξερεύνηση του χώρου των λύσεων προς αναζήτηση νέας βέλτιστης λύσης. Η διαφοροποίηση εξασφαλίζει ότι ο αλγόριθμος δεν θα παγιδευτεί σε ένα τοπικό ελάχιστο. Με τη σωστή αναλογία των δύο παραπάνω στοιχείων του αλγορίθμου εξασφαλίζεται η εύρεση του ολικού βέλτιστου.

Οι μεθευρετικοί αλγόριθμοι συνήθως χρησιμοποιούν πιο παραδοσιακούς ευρετικούς αλγορίθμους σαν υποδιαδικασίες τους. Πολλές φορές μπορούν να επιτραπούν σε κάποιο μεθευρετικό αλγόριθμο, και βήματα που οδηγούν σε κάποια μη εφικτή ενδιάμεση λύση σε κάποιο βήμα του αλγορίθμου. Ο λόγος που επιτρέπεται αυτό είναι για να αποφθεχθεί κάποιο τοπικό ελάχιστο και ολική λύση που θα εξαχθεί από τον αλγόριθμο να είναι καλύτερη. Ένα επιπλέον χαρακτηριστικό αυτών των αλγορίθμων είναι ότι προσομοιάζουν μια διαδικασία που συνήθως έχει εφαρμογή στη φύση. τα στοιχεία που χρησιμοποιούν αυτοί οι αλγόριθμοι και μεταφορικά τα παρατηρούμε και στη φύση είναι τα εξής :

- Χρησιμοποιούν έναν αριθμό από επαναληπτικές δοκιμές
- Περιλαμβάνουν ένα οι περισσότερους πράκτορες (νευρώνες, μόρια, χρωμοσώματα, μυρμήγκια)
- Λειτουργούν (στην περίπτωση των πολύ-πρακτόρων) βάση ενός μηχανισμού συνεργασίας και ανταγωνισμού.
- Περιλαμβάνουν διαδικασίες αυτοτροποποιήσεων των ευρετικών παραμέτρων ή ακόμα και της αναπαράστασης το προβλήματος.

Τα χαρακτηριστικά των μεθευρετικών είναι τα εξής :

1. Μοντελοποιούν ένα φαινόμενο που υπάρχει στη φύση.
2. Μπορούν να μεταφερθούν εύκολα σε παράλληλη μορφή.
3. Είναι προσαρμοστικοί αλγόριθμοι.

3.2 Περιορισμένη Αναζήτηση (Tabu Search)

Εισαγωγή στην περιορισμένη αναζήτηση

ο μεθευρετικός αλγόριθμος της περιορισμένης αναζήτησης, είναι ίσως ο πιο γνωστός μεθευρετικός αλγόριθμος και αρχικά παρουσιάστηκε από τον Glover. Η περιορισμένη αναζήτηση χρησιμοποιεί ένα ευρετικό αλγόριθμο για να μετακινηθεί από την μια λύση στην άλλη. Όμως, όπως και στους άλλους μεθευρετικούς αλγορίθμους, υπάρχει η πιθανότητα η λύση να παγιδευτεί σε τοπικό ελάχιστο. Η σημαντική διαφορά της περιορισμένης αναζήτησης από τους υπόλοιπους μεθευρετικούς αλγορίθμους είναι ότι για να ξεφύγει η λύση από το τοπικό ελάχιστο χρησιμοποιείται μια στρατηγική για την επιλογή της επόμενης λύσης και δεν επιλέγεται τυχαία. Η συγκεκριμένη στρατηγική βασίζεται στη ύπαρξη μνήμης από τις προηγούμενες κινήσεις που έχουν πραγματοποιηθεί. Για να αποφευχθούν, δηλαδή, επαναλαμβανόμενοι κύκλοι γύρω από μια ομάδα λύσεων οι τελευταίες κινήσεις καταγράφονται σε μια λίστα, η οποία ονομάζεται λίστα περιορισμένων κινήσεων και οι συγκεκριμένες κινήσεις απαγορεύεται να επιστρέψουν στη λύση για ένα συγκεκριμένο αριθμό κινήσεων τον οποίο έχουμε καθορίσει από την αρχή.

Ανάλυση αλγορίθμου περιορισμένης αναζήτησης

Το πρώτο στάδιο για την ανάπτυξη του αλγορίθμου είναι όπως και σε κάθε άλλο αλγόριθμο η κατασκευή αρχικής λύσης. Έπειτα κατασκευάζεται η συνάρτηση

κόστους $c(s)$ και υπολογίζεται το ολικό κόστος ώστε να συγκριθεί με τις επόμενες κινήσεις. Αφού ολοκληρωθεί το πρώτο στάδιο του αλγορίθμου αρχίζει η βελτιστοποίηση της λύσης μέσω τοπικής αναζήτησης. Για κάθε κίνηση υπολογίζεται το κόστος και συγκρίνεται με το κόστος της προηγούμενης λύσης.

Για να αποφευχθεί η παγίδευση σε κάποιο τοπικό ελάχιστο χρησιμοποιούνται πληροφορίες από τις χ τελευταίες επαναλήψεις. Το σύνολο των τελευταίων χ επαναλήψεων μορφοποιούν την λίστα περιορισμένης αναζήτησης. Αυτή η διαδικασία ονομάζεται μνήμη μικρής διάρκειας και σύμφωνα με αυτή ο αλγόριθμος θυμάται τις τελευταίες κινήσεις και εμποδίζει την επαναχρησιμοποίηση τους. Οι απαγορευμένες αυτές κινήσεις ονομάζονται περιορισμένες κινήσεις και δεν μπορούν χρησιμοποιηθούν μέχρι να βγουν από την λίστα. Το μέγεθος της λίστας είναι ένας πολύ σημαντικός παράγοντας για την επιτυχία του αλγορίθμου καθώς επηρεάζει για πόσες επαναλήψεις θα περιορίζονται οι συγκεκριμένες κινήσεις. Η μεταβλητή χ ορίζεται από τον χρηστή στην αρχή του αλγορίθμου και αποτελεί το μέγεθος της μνήμης μικρής περιόδου. Σε κάθε επανάληψη προστίθεται στην λίστα η τελευταία κίνηση που έχει πραγματοποιηθεί και αν έχει συμπληρωθεί, η τελευταία κίνηση απελευθερώνεται.

Η ύπαρξη αυτής της μνήμης βοηθά στην αποφυγή των ίδιων λύσεων που προκύπτουν ανά τις επαναλήψεις.

Πάραυτα είναι πιθανό κάποια από τις απαγορευμένες λύσεις να μπορεί να βελτιώσει την συνάρτηση κόστους και λόγω της μνήμης να παρακάμπτεται. Για αυτόν τον λόγο εφαρμόζεται ένα κριτήριο απενεργοποίησης των περιορισμών. Με αυτόν τον τρόπο, όταν προκύπτει μια περιορισμένη κίνηση το κριτήριο υπολογίζει την συνάρτηση κόστους της λύσης, συγκρίνει με την προηγούμενη λύση και αποφασίζει αν είναι προτιμητέο το να αγνοηθούν οι περιορισμοί.

Μακροπρόθεσμες μνήμες

Εκτός από την από την μνήμη μικρής διάρκειας ο αλγόριθμος της περιορισμένης αναζήτησης βασίζεται και σε κάποιες άλλες στρατηγικές- μνήμες για την βελτιστοποίηση της λύσης του. Η σημαντικότερη από αυτές είναι η στρατηγική εντατικοποίησης. η συγκεκριμένη στρατηγική χρησιμοποιεί μια μνήμη στην οποία αποθηκεύει τον συνδυασμό κόμβων που περιέχει το μονοπάτι κάθε λύσης. Ο αλγόριθμος θεωρεί ότι είναι πιθανότερο οι πιο συχνοί συνδυασμοί κόμβων να είναι μέρος της βέλτιστης λύσης. Έτσι μέσω της μνήμης εντοπίζονται οι συχνοί συνδυασμοί και παραμένουν μέσα στην λύση, έπειτα πραγματοποιείται νέα

αναζήτηση στην γειτονιά της νέας λύσης. Η συγκεκριμένη μνήμη ονομάζεται **μνήμη μεσαίας περιόδου** έχει μέγεθος όσο και τα πιθανά τόξα των κόμβων και ενημερώνεται σε κάθε επανάληψη. Ο αριθμός επαναλήψεων που αποθηκεύει είναι ανάλογος με τις συνολικές επαναλήψεις και μεγαλύτερος από αυτόν της μικρής μνήμης.

Η δεύτερη στρατηγική που χρησιμοποιεί η περιορισμένη αναζήτηση είναι η στρατηγική της διάχυσης. ομοίως με την στρατηγική της εντατικοποίησης χρησιμοποιείται μια μνήμη που αποθηκεύει τα τόξα των λύσεων. η διαφορά τώρα είναι ότι ο αλγόριθμος θέλει να ερευνήσει σε πιο σπάνια εμφανισμένους συνδυασμούς ώστε να επανεκκινήσει την διαδικασία εύρεσης από την λιγότερο πιθανή λύση. Η μνήμη που χρησιμοποιείται ονομάζεται μνήμη μεγάλης έχει την ίδια δομή και λειτουργία με την μεσαία μνήμη αλλά ενημερώνεται για μεγαλύτερο πλήθος επαναλήψεων.

Η χρήση των συγκεκριμένων στρατηγικών μας διασφαλίζει ότι ο αλγόριθμος δεν θα εγκλωβιστεί σε κάποιο τοπικό ελάχιστο καθώς καλώντας τις δυο αυτές μνήμες μετά από έναν αριθμό επαναλήψεων είναι πιθανότερο να εξερευνηθούν όλες οι περιοχές λύσεων.

Αλγόριθμος

Αρχικοποίηση

Κατασκευή μιας αρχικής λύσης S_0

Υπολογισμός κόστους λύσης

$S^* = S_0$! αρχικοποίηση βέλτιστης λύσης

$F(S^*) = f(S_0)$

Κύρια φάση

KANE ΟΣΟ κάποιο κριτήριο σταματήματος δεν έχει ικανοποιηθεί

Υπολογισμός μια γειτονικής λύσης S'

AN $f(S') < f(S^*)$ **TOTE**

$S^* = S'$

$f^* = f(S')$

ΤΕΛΟΣ ΑΝ

Αποθήκευσε την τελευταία κίνηση στη λίστα περιορισμένων υποψηφίων(ταυτόχρονα αν έχει συμπληρωθεί το μέγεθος της λίστας διέγραψε την παλαιότερη)

Κάλεσε κάθε k_1 επαναλήψεις την στρατηγική εντατικοποίησης

ΑΝ $f(S_{\text{εντατικοποίησης}}) < f(S^*)$ **ΤΟΤΕ**

$$S^* = S_{\text{εντατικοποίησης}}$$

$$f^* = f(S_{\text{εντατικοποίησης}})$$

ΤΕΛΟΣ ΑΝ

Κάλεσε κάθε k_2 επαναλήψεις την στρατηγική διάχυσης

ΑΝ $f(S_{\text{διάχυσης}}) < f(S^*)$ **ΤΟΤΕ**

$$S^* = S_{\text{διάχυσης}}$$

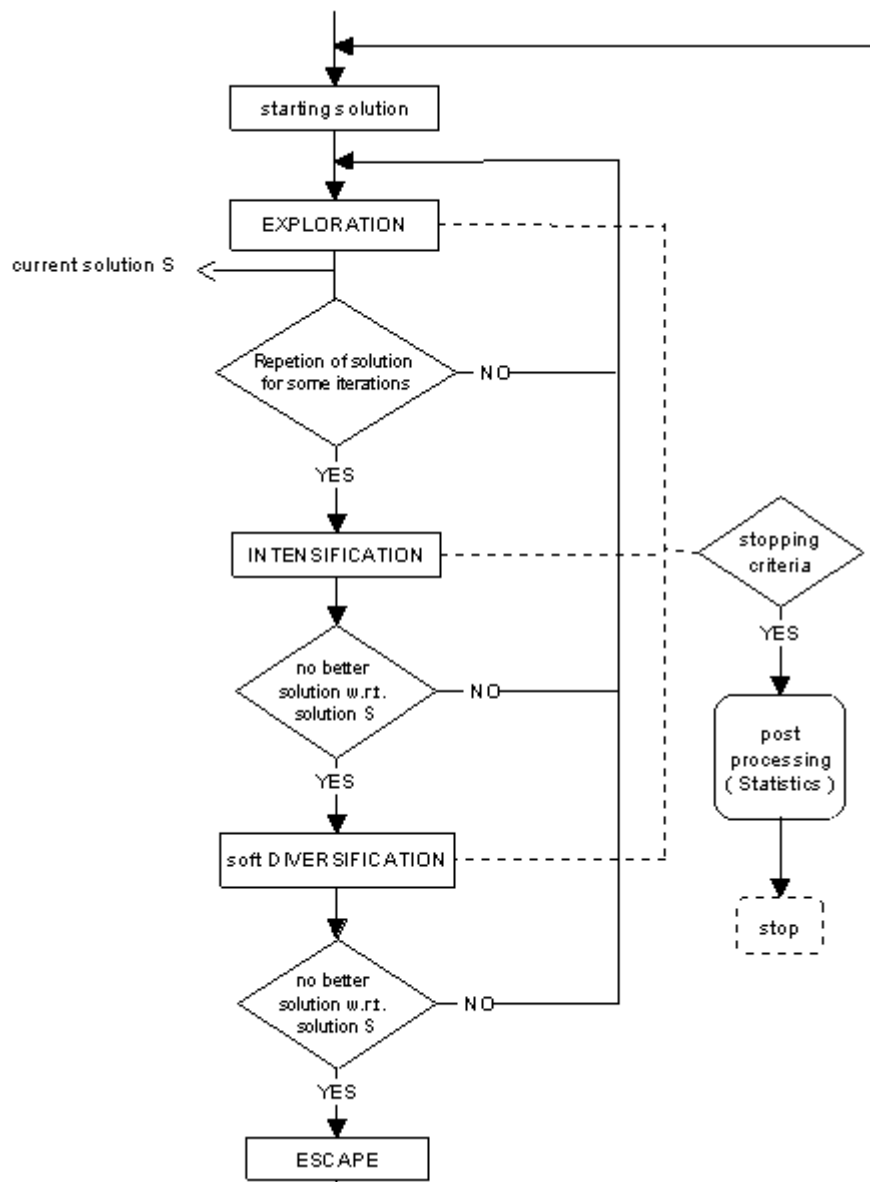
$$f^* = f(S_{\text{διάχυσης}})$$

ΤΕΛΟΣ ΑΝ**ΤΕΛΟΣ ΚΑΝΕ**

Επέστρεψε τη βέλτιστη λύση.

[1],[6]

Στην παρακάτω εικόνα βλέπουμε τον τρόπο λειτουργίας ενός αλγορίθμου περιορισμένης αναζήτησης:



ΚΕΦΑΛΑΙΟ 4ο

Εφαρμογή στον κώδικα

4.1 Επιλογή περιβάλλοντος επίλυσης

Το περιβάλλον Matlab είναι ένα διαδραστικό περιβάλλον για αριθμητικούς υπολογισμούς, απεικόνιση δεδομένων και ταυτόχρονα μια γλώσσα προγραμματισμού υψηλού επιπέδου.

Χρησιμοποιείται για την ανάλυση δεδομένων, την ανάπτυξη αλγορίθμων, τη δημιουργία μοντέλων και εφαρμογών. Η γλώσσα σε συνδυασμό με τα εργαλεία και τις έτοιμες συναρτήσεις επιτρέπουν την επίτευξη της λύσης ευκολότερα σε σχέση με άλλες κλασσικές γλώσσες προγραμματισμού.

Χρησιμοποιείται ευρέως στον επιστημονικό κοινό κυρίως στη μηχανική, τις φυσικές επιστήμες και τα οικονομικά.

Ο λόγος που επελέγη το περιβάλλον Matlab έναντι κάποιας άλλης γλώσσας έγκυται στα παρακάτω πλεονεκτήματα:

- Το βασικό δομικό στοιχείο δεδομένων είναι ο πίνακας. Ένας απλός ακέραιος θεωρείται πίνακας 1X1. Αρκετές μαθηματικές πράξεις σε πίνακες περιλαμβάνονται στο Matlab, όπως το εξωτερικό γινόμενο, η ορίζουσα και η αντιστροφή πινάκων.
- Περιλαμβάνει πράξεις διανυσμάτων, για παράδειγμα η ένωση δυο διανυσμάτων γίνεται με μια εντολή αντί για μια δομή επανάληψης.
- Η ευκολία στην γραφική απεικόνιση της λύσης
- Η ευκολία εισαγωγής και εξαγωγής δεδομένων από διάφορους τύπους δεδομένων όπως το αρχείο excel.

Ωστόσο παρ όλες τις ευκολίες για τον προγραμματιστή/χρήστη, εμπεριέχει δύο κύρια μειονεκτήματα:

1. Καταλαμβάνει πολύ χώρο στη μνήμη και σε αργούς υπολογιστές αργεί να βγάλει αποτελέσματα.
2. Αποτελεί εφαρμογή του λειτουργικού και επομένως χρησιμοποιεί όσο χρόνο CPU το λειτουργικό του επιτρέπει. Αυτό κάνει εφαρμογές πραγματικού χρόνου πολύ δύσκολες.

Η ευκολία κατά τον προγραμματισμό ήταν η αιτία που χρησιμοποιήθηκε η Matlab έναντι κάποιας άλλης γλώσσας η οποία θα ήταν πιο γρήγορη κατά την εκτέλεση.

4.2 Δεδομένα

Τα δεδομένα του προβλήματος που επιλύεται στην παρούσα εργασία δόθηκαν σε μορφή excel και είχαν την παρακάτω μορφή:

- eil51Q41

Χωρητικότητα οχήματος		41	
Αριθμός κόμβων		51	
Συντεταγμένες στον χάρτη χ,ψ του κάθε κόμβου	1	30	40
	2	37	52

	51	56	37
Ζήτηση κάθε κόμβου	1	-25	
	2	7	
		
	51	-10	

Τα δεδομένα μορφοποιούνται κατάλληλα και εισάγονται στον κώδικα με την ειδική συνάρτηση (DATA.m).

4.3 Κώδικας

Η επίλυση του προβλήματος στο περιβάλλον matlab πραγματοποιήθηκε ύστερα από δημιουργία ενός βασικού προγράμματος και κάποιων συναρτήσεων που αυτό καλεί.

Ανάλυση αλγορίθμου

Στόχος κάθε αλγορίθμου είναι υπολογίσει την συνάρτηση κόστους και να βελτιστοποιήσει το αποτέλεσμα της. Στην προκείμενη περίπτωση, ο στόχος είναι η ελαχιστοποίηση της συνολικής απόστασης που θα διανύσει το μεταφορικό όχημα για να ικανοποιήσει το σύνολο των πελατών.

Πρώτο στάδιο είναι η δημιουργία μιας τυχαίας αρχικής λύσης. Είναι σημαντικό αυτή η λύση να τηρεί τους περιορισμούς του προβλήματος ώστε να μπορεί να χρησιμοποιηθεί μετέπειτα για σύγκριση. Η λύση αυτή επιτυγχάνεται, όπως θα δούμε και παρακάτω, με την χρήση του αλγορίθμου του πλησιέστερου γείτονα. Αφού υπάρχει μια αρχική λύση αρχίζουν οι επαναλήψεις τοπικής αναζήτησης. Ο κώδικας εναλλάσσει δυο κόμβους ανά επανάληψη και υπολογίζει τη νέα διαδρομή αλλά και το κόστος της. Παράλληλα η εναλλαγή τοποθετείται στην λίστα περιορισμένων κινήσεων. Σε περίπτωση που η κίνηση ήταν ήδη περιορισμένη, καλείται το κριτήριο φιλοδοξίας.

Ανά έναν προκαθορισμένο αριθμό κινήσεων καλούνται η στρατηγικές εντατικοποίησης και διάχυσης, για κάθε μια από υπολογίζεται η διαδρομή και το κόστος της. Σε περίπτωση που το κόστος της είναι μικρότερο από αυτό της αρχικής λύσης ή κοντινό σε αυτήν, αρχίζει νέος κύκλος επαναλήψεων τοπικής αναζήτησης από την συγκεκριμένη λύση.

Τέλος, μετά την ολοκλήρωση των προκαθορισμένων επαναλήψεων, ο αλγόριθμος επιστρέφει την βέλτιστη λύση καθώς και τις καλύτερες λύσεις των υπόλοιπων στρατηγικών.

Συναρτήσεις

Παρακάτω, παρατίθενται οι συνεντίσεις που απαρτίζουν τον κώδικα με τον οποίο επιλύσαμε τα προβλήματα. Σε κάθε συνάρτηση υπάρχει επεξήγηση το ιδιοτήτων και του σκοπού της ενώ στις σημαντικότερες παρατίθεται και ψευδοκώδικας για καλύτερη κατανόηση της λειτουργίας της.

Data.m

Η συνάρτηση εισάγει στο πρόγραμμα τα δεδομένα του προβλήματος. Με την χρήση των κατάλληλων εντολών ο αλγόριθμος διαβάζει τα δεδομένα από το αρχείο excel και τα τοποθετεί σε πίνακες τους οποίους χρησιμοποιεί το βασικό κομμάτι του κώδικα.

Main.m

Αποτελεί το βασικό κομμάτι του κώδικα το οποίο και καλεί τις υπόλοιπες συναρτήσεις. Αρχικά ο αλγόριθμος καλεί την συνάρτηση data ώστε να χρησιμοποιήσει το μορφοποιημένα δεδομένα του προβλήματος. Έπειτα αρχικοποιεί, βρίσκει μια αρχική εφικτή λύση και υπολογίζει το κόστος της. Για να επιχθεί αυτό το βήμα υλοποιείται ο αλγόριθμος του πλησιέστερου γείτονα ως εξής:

Αρχικοποίηση

ΟΣΘ το άθροισμα των κόμβων που τα οχήματα έχουν επισκεφτεί < αριθμού των κόμβων

-Βρίσκω τον κοντινότερο οικονομικά κόμβο

ΑΝ το απόθεμα του οχήματος φτάνει για την ικανοποίηση της ζήτησης

-Σημείωσε τη μετάβαση

-Υπολόγισε τη ροή στη μετάβαση αυτή

-Καταχώρησε τον κόμβο στην διαδρομή

-Πρόσθεσε στη συνολική απόσταση του οχήματος την απόσταση της μετάβασης

-Θέσε ως επόμενο κόμβο αφητηρίας τον κόμβο προορισμού.

ΤΕΛΟΣ_ΑΝ

ΑΝ δεν υπάρχει επιτρεπτή κίνηση **Η** έχουμε επισκεφτεί όλους τους κόμβους

-Επέστρεψε στην αποθήκη

-Σημείωσε τη μετάβαση

-Υπολόγισε τη ροή στη μετάβαση αυτή

-Πρόσθεσε στη συνολική απόσταση του οχήματος την απόσταση της μετάβασης

-Θέσε ως επόμενο κόμβο αφητηρίας την αποθήκη.

ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΟΣΟ

ΤΥΠΩΣΕ την διαδρομή του φορτηγού και την συνολική απόσταση.

Μετά αρχίζουν οι επαναλήψεις τοπικής αναζήτησης με παράλληλη χρήση της μνήμης μικρής περιόδου. Για κάθε επανάληψη υπολογίζεται το κόστος και αναζητείται η διαδρομή που θα το ελαχιστοποιεί. Τέλος καλεί τις παρακάτω συναρτήσεις ώστε να ικανοποιηθούν το κριτήριο φιλοδοξίας και οι στρατηγικές εντατικοποίησης και διάχυσης.

Αρχικοποίηση

ΓΙΑ ένα αριθμό επαναλήψεων

-κάνε αλλαγή μεταξύ 2 τυχαίων κόμβων

-βρες το κομμάτι της διαδρομής μέχρι τον κόμβο της πρώτης σε σειρά αλλαγής

-βρες το φορτίο του οχήματος στον τελευταίο σταθμό της συγκεκριμένης διαδρομής

ΟΣΟ δεν έχεις επισκεφτεί όλες τις πόλεις –κόμβους

-βρες τον κοντινότερο κόμβο

ΑΝ δεν τηρείται ο περιορισμός χωρητικότητας

-βρες επόμενο κοντινό κόμβο

ΤΕΛΟΣ ΑΝ

ΑΝ τηρείται ο περιορισμός χωρητικότητας

ΑΝ δεν τηρείται ο περιορισμός περιορισμένων κινήσεων

-κάλεσε κριτήριο φιλοδοξίας (*Kritfilod.m*)

ΑΛΛΙΩΣ

-πρόσθεσε τον κόμβο στη διαδρομή

- Αφαίρεσε τον κόμβο από τη λίστα

-Θέσε επόμενο κόμβο αφετηρίας τον κόμβο που πήγες

-ενημέρωσε πίνακα περιορισμένων κινήσεων

ΤΕΛΟΣ ΑΝ

ΑΝ έχεις επισκεφθεί όλους τους κόμβους **Η** δεν υπάρχει κόμβος να τηρεί τους περιορισμούς

-Καταχώρησε τη μετάβαση από τον κόμβο που είσαι στην αποθήκη

-Θέσε επόμενη αφετηρία την αποθήκη
-Αρχικοποίησε το απόθεμα

ΤΕΛΟΣ ΑΝ

-αποθήκευσε την μετάβαση στους πινάκες μνημών

ΤΕΛΟΣ_ΟΣΟ

-Αρχικοποίηση

ΤΕΛΟΣ ΓΙΑ

ΤΥΠΩΣΕ αποτελέσματα

Kritfilod.m

Η συγκεκριμένη συνάρτηση αποσκοπεί στην δημιουργία του κριτηρίου φιλοδοξίας-απενεργοποίησης περιορισμών. Όπως αναφέρθηκε και στην αρχή, το κομμάτι αυτό του κώδικα αποτελεί ένα μηχανισμό του αλγορίθμου να απενεργοποιεί τους περιορισμούς σε περίπτωση που η περιορισμένη επιλαχούσα κίνηση πρόκειται να βελτιώσει την λύση. Η συνάρτηση, λοιπόν, καλείται σε περίπτωση που η κίνηση που επιλέχτηκε είναι περιορισμένη. Το πρόγραμμα υπολογίζει με τον αλγόριθμο του πλησιέστερου γείτονα υποθετικά το κόστος της κίνησης και αν αυτό είναι καλύτερο το κριτήριο ενεργοποιείται.

ΥΠΟΛΟΓΙΣΕ λύση με τη μέθοδο του βασικού προγράμματος
(Main.m)

ΑΝ κόστος λύσης < κόστος καλύτερης μέχρι τώρα λύσης

-επέτρεψε την κίνηση

-ενημέρωσε την μικρή μνήμη

ΤΕΛΟΣ ΑΝ

ΑΝ κόστος λύσης >= κόστος καλύτερης μέχρι τώρα λύσης

-μην επιτρέψεις την κίνηση

ΤΕΛΟΣ ΑΝ

ΑΡΧΙΚΟΠΟΙΗΣΕ

Endiamesh_mesaia_mnhmh.m και endiamesh_megalh_mnhmh.m

Οι συναρτήσεις αυτές αποτελούν ένα ενδιάμεσο στάδιο για την ενεργοποίηση των στρατηγικών εντατικοποίησης και διάχυσης. Πιο αναλυτικά, αυτές καλούνται σε κάθε επανάληψη και γεμίζουν την μεγάλης και μεσαίας περιόδου μνήμη αντίστοιχα, όταν περάσει ο προκαθορισμένος αριθμός οπού η κάθε στρατηγική καλείται η αντίστοιχη μνήμη μηδενίζεται και διαδικασία αρχίζει από την αρχή. Ο αριθμός των αλλαγών που τοποθετούνται στον πίνακα εξαρτάται από τις παραμέτρους μεγέθους της μεσαίας και της μεγάλης μνήμης αντίστοιχα.

ΑΝ μέγεθος μνήμης = μέγιστος μέγεθος

-μηδένισε τον πίνακα μνήμης

ΤΕΛΟΣ ΑΝ

-αποθήκευσε για κάθε επανάληψη την αλλαγή κόμβων

-Αύξησε κατά ένα βαθμό την τομή των 2 κόμβων στον πίνακα

-αύξησε κατά 1 το μέγεθος της μνήμης

ΑΡΧΙΚΟΠΟΙΗΣΕ

Mesaia_mnhmh.m

Αποτελεί την στρατηγική της εντατικοποίησης. Ανά έναν αριθμό επαναλήψεων η συνάρτηση καλείται, ελέγχει την μνήμη μεσαίας περιόδου και δημιουργεί ένα μονοπάτι διαδρομής βασιζόμενη στον μέγιστο συνδυασμό κόμβων που περιέχει η μνήμη. Έπειτα υπολογίζει και το κόστος της συγκεκριμένης λύσης.

ΔΙΑΒΑΣΕ τον πίνακα της ενδιάμεσης μεσαίας μνήμης

θέσε αρχικό κόμβο την αποθήκη

ΟΣΟ δεν έχεις επισκεφτεί όλες τις πόλεις –κόμβους

-βρες τον κόμβο με τις περισσότερες αλλαγές με την αφετηρία από τον πίνακα

-AN δεν τηρείται ο περιορισμός χωρητικότητας

-βρες επόμενο καλύτερο κόμβο

-ΤΕΛΟΣ AN

AN τηρείται ο περιορισμός χωρητικότητας

-πρόσθεσε τον κόμβο στη διαδρομή

Αφαίρεσε τον κόμβο από τον πίνακα

-Θέσε επόμενο κόμβο αφετηρίας τον κόμβο που πήγες

ΤΕΛΟΣ AN

AN έχεις επισκεφθεί όλους τους κόμβους Ή δεν υπάρχει κόμβος να τηρεί τους περιορισμούς

-Καταχώρησε τη μετάβαση από τον κόμβο που είσαι στην αποθήκη

-Θέσε επόμενη αφετηρία την αποθήκη

-Αρχικοποίησε το απόθεμα

ΤΕΛΟΣ_AN

-ΤΕΛΟΣ_ΟΣΟ

-ΤΥΠΩΣΕ το μονοπάτι και το μήκος της διαδρομής του

Megalh_mnhmh.m

Αποτελεί την στρατηγική της διάχυσης. Όπως και η παραπάνω, καλείται ανά έναν προκαθορισμένο αριθμό επαναλήψεων, ελέγχει την μνήμη μεγάλης περιόδου όπου στην οποία αναζητά τα λιγότερο δημοφιλή ανά τις επαναλήψεις ζευγάρια κόμβων και δημιουργεί μια νέα λύση. Τέλος υπολογίζει το κόστος της. Ο κώδικας της συνάρτησης είναι όμοιος με αυτόν της μεσαίας μνήμης, με μονή διάφορα την αναζήτηση των ελαχίστων αλλαγών.

Map.m

Η παραπάνω συνάρτηση δημιουργεί γράφημα στο οποίο απεικονίζεται η αποθήκη καθώς και οι υπόλοιποι κόμβοι. Επίσης, σχεδιάζει τα τόξα τα οποία απαρτίζουν την βέλτιστη διαδρομή.

ΚΕΦΑΛΑΙΟ 4^ο

ΕΠΙΛΥΣΗ

ΑΠΟΤΕΛΕΣΜΑΤΑ

Στην παρούσα εργασία επιλύθηκαν δέκα προβλήματα δρομολόγησης τα όποια ποικιλοούσαν ως αναφορά το πλήθος των κόμβων αλλά και την χωρητικότητα των οχημάτων. Παραδείγματος χάριν τα παρακάτω :

- Eil51Q41
- Eil73Q134
- Eil101Q82

Τα προβλήματα επιλύθηκαν πολλές φορές ώστε να πραγματοποιηθούν δόκιμες στον κώδικα και στις τιμές των παραμέτρων που θα χρησιμοποιηθούν ώστε να επιτεθούν τα καλύτερα δυνατά αποτελέσματα. οι παράμετροι αφορούν το μέγεθος της κάθε μνήμης αλλά και τον αριθμό των επαναλήψεων. Έπειτα από αρκετές δοκιμές και αφού καταλήξαμε στις τιμές που θα έχει η κάθε παράμετρος πραγματοποιήθηκαν 5 τελικές διμεις σε κάθε πρόβλημα ώστε να υπάρχει μια τελική εικόνα. Θα πρέπει να σημειώσουμε ότι οι παράμετροι δεν διατηρήθηκαν ίδιες σε κάθε πρόβλημα καθώς, ανάλογα το μέγεθος του προβλήματος αυτές έπρεπε να αλλάξουν. Για παράδειγμα, για μεγαλύτερο πλήθος κόμβων έπρεπε να μεγαλώσει και το μέγεθος των μακροπρόθεσμων μνημών για καλύτερη χρήση των στρατηγικών, ενώ ο αριθμός επαναλήψεων έπρεπε να μειωθεί ώστε να μην υπάρχει μεγάλη απόκλιση στον χρόνο επίλυσης του κάθε προβλήματος.

Τέλος, γίνεται ανάλυση των αποτελεσμάτων και σύγκριση με αυτά της βιβλιογραφίας.

A. Προβλήματα τύπου eil με 51 κόμβους

Οι παράμετροι που επιλέχτηκαν στα συγκεκριμένα προβλήματα:

Μέγεθος μικρής μνήμης	10
Μέγεθος μεσαίας μνήμης	200
Μέγεθος μεγάλης μνήμης	300
επαναλήψεις	4000

- **eil51Q41**

Χαρακτηριστικά του προβλήματος:

χωρητικότητα	41
κομβόι	51

Κόστος αρχικής λύσης	712
Κόστος αρχικής λύσης μεσαίας μνήμης	842
Κόστος αρχικής λύσης μεγάλης μνήμης	2166
Κόστος καλύτερης λύσης	587

Όπως βλέπουμε σε αυτή την περίπτωση το κόστος των λύσεων των στρατηγικών ήταν μεγαλύτερο από εκείνο της αρχικής λύσης. Συνεπώς η βέλτιστη λύση προσήλθε από τοπική αναζήτηση στην αρχική λύση.

Τελικές δοκιμές:

Επανάληψη	Αποτέλεσμα μικρής μνήμης	Καλύτερη γνωστή λύση
1 ^η	587	
2 ^η	590	
3 ^η	588	
4 ^η	587	
5 ^η	587	
M.O	587,8	504

- **eil51Q70**

Χαρακτηριστικά του προβλήματος:

χωρητικότητα	70
κομβόι	51

Κόστος αρχικής λύσης	615
Κόστος αρχικής λύσης μεσαίας μνήμης	594
Κόστος αρχικής λύσης μεγάλης μνήμης	1,931
Κόστος καλύτερης λύσης	505

Σε αυτήν την περίπτωση η λύση της μεσαίας μνήμης ήταν καλύτερη από εκείνη της αρχικής οπότε προτιμήθηκε να γίνει περεταίρω τοπική αναζήτηση στη λύση της μεσαίας μνήμης.

Τελικές δοκιμές:

Επανάληψη	Αποτέλεσμα μικρής μνήμης	Καλύτερη γνωστή λύση
1 ^η	505	
2 ^η	507	
3 ^η	505	
4 ^η	511	
5 ^η	506	
Μ.Ο	506,8	434

- **eil51Q80**

Χαρακτηριστικά του προβλήματος:

χωρητικότητα	41
κομβοί	80

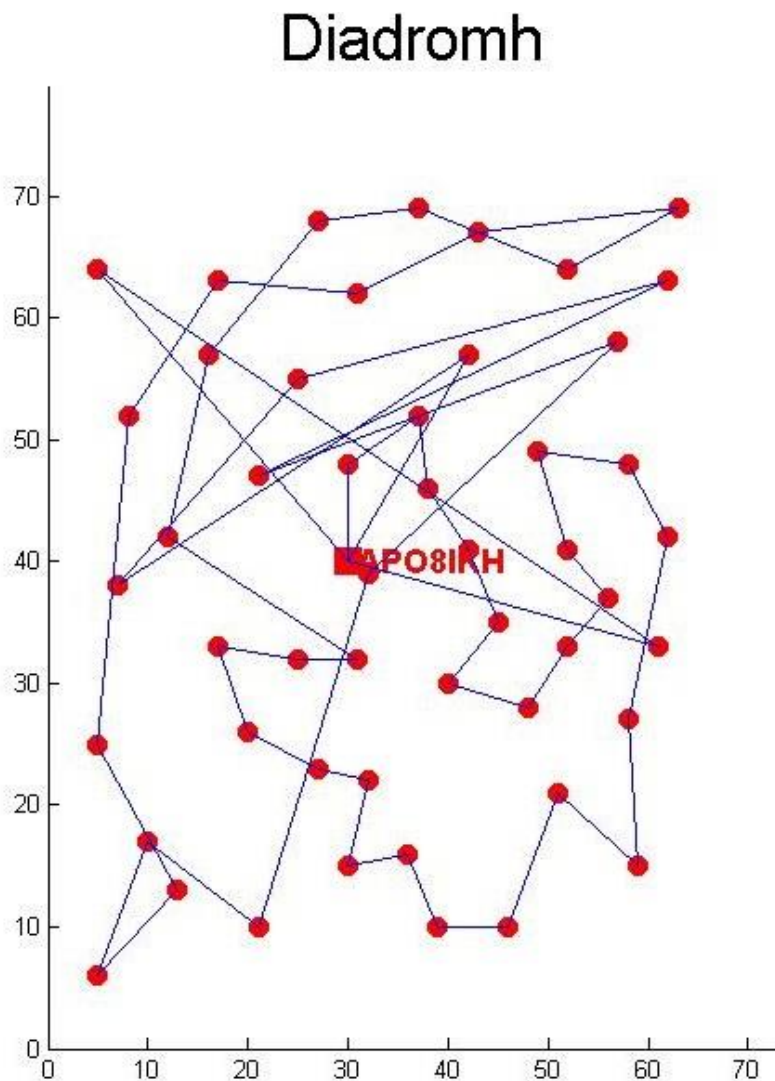
Κόστος αρχικής λύσης	615
Κόστος αρχικής λύσης μεσαίας μνήμης	594
Κόστος αρχικής λύσης μεγάλης μνήμης	1797
Κόστος καλύτερης λύσης	505

Και σε αυτήν την περίπτωση η λύση της μεσαίας μνήμης ήταν καλύτερη οπότε έγινε περεταίρω τοπική αναζήτηση στη λύση της μεσαίας μνήμης.

Τελικές δοκιμές:

Επανάληψη	Αποτέλεσμα μικρής μνήμης	Καλύτερη γνωστή λύση
1 ^η	505	
2 ^η	514	
3 ^η	505	
4 ^η	506	
5 ^η	505	
Μ.Ο	507	434

Σχεδιάγραμμα α. μια διαδρομή με 51 κόμβους:



Σημείωση: Τα παραπάνω προβλήματα ήταν και τα μικρότερα σε πλήθος κόμβων που επιλύθηκαν, οπότε υπήρχε η δυνατότητα να χρησιμοποιηθεί μεγαλύτερος αριθμός επαναλήψεων τοπικής αναζήτησης στον ίδιο χρόνο επίλυσης.

B. Προβλήματα τύπου eil με 76 κόμβους

Οι παράμετροι που επιλέχτηκαν στα συγκεκριμένα προβλήματα:

Μέγεθος μικρής μνήμης	10
Μέγεθος μεσαίας μνήμης	200
Μέγεθος μεγάλης μνήμης	300
επαναλήψεις	3000

- **eil76Q134**

Χαρακτηριστικά του προβλήματος:

χωρητικότητα	134
κόμβοι	76

Κόστος αρχικής λύσης	1052
Κόστος αρχικής λύσης μεσαίας μνήμης	917
Κόστος αρχικής λύσης μεγάλης μνήμης	2617
Κόστος καλύτερης λύσης	613

Τοπική αναζήτηση στη λύση της μεσαίας μνήμης.

Τελικές δοκιμές

Επανάληψη	Αποτέλεσμα μικρής μνήμης	Καλύτερη γνωστή λύση
1 ^η	628	
2 ^η	632	
3 ^η	613	
4 ^η	618	
5 ^η	613	
Μ.Ο	620,7	547

- **eil76Q138**

Χαρακτηριστικά του προβλήματος:

Χωρητικότητα	138
Κόμβοι	76

Κόστος αρχικής λύσης	785
Κόστος αρχικής λύσης μεσαίας μνήμης	738
Κόστος αρχικής λύσης μεγάλης μνήμης	2800
Κόστος καλύτερης λύσης	617

Τοπική αναζήτηση στη λύση της μεσαίας μνήμης.

Τελικές δοκιμές:

Επανάληψη	Αποτέλεσμα μικρής μνήμης	Καλύτερη γνωστή λύση
1 ^η	617	
2 ^η	640	
3 ^η	617	
4 ^η	617	
5 ^η	624	
M.O	623	547

- **eil76Q160**

Χαρακτηριστικά του προβλήματος:

χωρητικότητα	160
κόμβοι	76

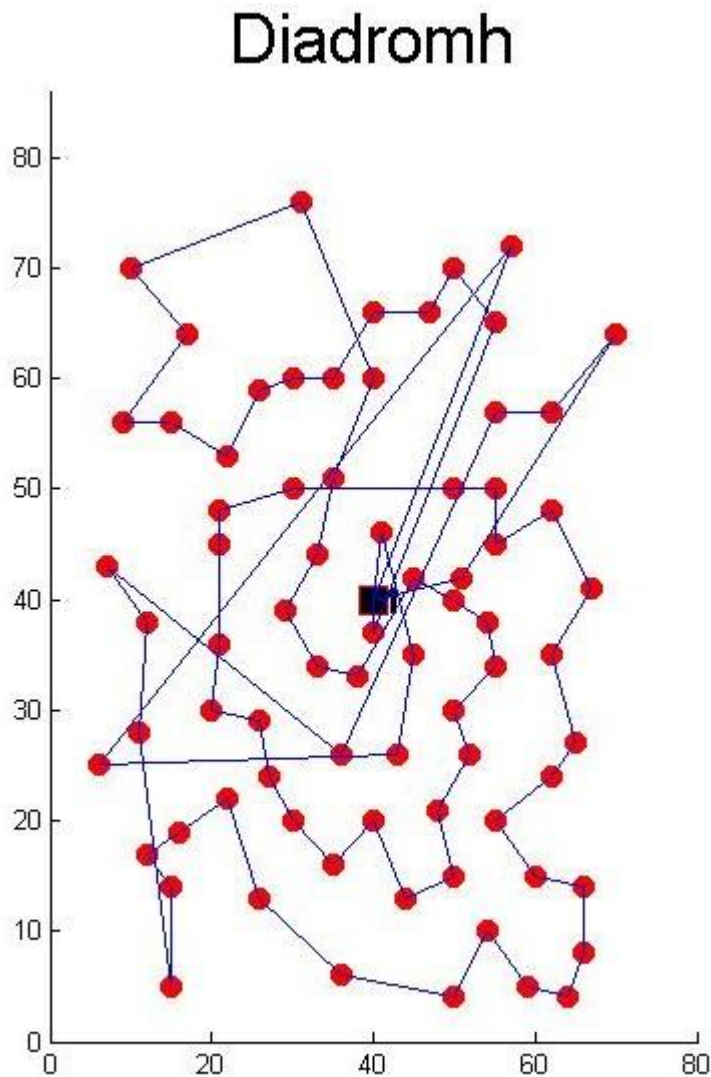
Κόστος αρχικής λύσης	770
Κόστος αρχικής λύσης μεσαίας μνήμης	743
Κόστος αρχικής λύσης μεγάλης μνήμης	2936
Κόστος καλύτερης λύσης	605

Τοπική αναζήτηση στη λύση της μεσαίας μνήμης.

Τελικές δοκιμές:

Επανάληψη	Αποτέλεσμα μικρής μνήμης	Καλύτερη γνωστή λύση
1 ^η	605	
2 ^η	609	
3 ^η	612	
4 ^η	609	
5 ^η	609	
Μ.Ο	609	539

Σχεδιάγραμμα β. μια διαδρομή με 76 κόμβους:



Σημείωση: Στα παραπάνω προβλήματα ο αριθμός των κόμβων ήταν κατά σχεδόν 50% μεγαλύτερος από ότι στην προηγούμενη κατηγορία. Συνεπώς υπήρξαν και οι αντίστοιχες αλλαγές και στις παραμέτρους. Αρχικά μειώθηκε ο αριθμός των επαναλήψεων ώστε να μην επιβαρύνεται περισσότερο το πρόγραμμα, έπειτα αυξήθηκε το μέγεθος της μικρής και μεγάλης μνήμης. Αυτό συνέβη διότι, έχοντας μεγαλύτερο πλήθος κόμβων προς αλλαγή στον ίδιο αριθμό επαναλήψεων υπήρχε ο κίνδυνος το δείγμα να μην είναι αντιπροσωπευτικό. Πράγμα που παρατηρήθηκε και κατά τις δοκιμές. Τέλος, το μέγεθος της μικρής μνήμης δεν χρειάστηκε να μεταβληθεί.

Γ. προβλήματα τύπου eil με 101 κόμβους

Οι παράμετροι που επιλέχτηκαν στα συγκεκριμένα προβλήματα:

Μέγεθος μικρής μνήμης	10
Μέγεθος μεσαίας μνήμης	400
Μέγεθος μεγάλης μνήμης	500
επαναλήψεις	2500

- **eil101Q95**

Χαρακτηριστικά του προβλήματος:

χωρητικότητα	95
κόμβοι	101

Κόστος αρχικής λύσης	889
Κόστος αρχικής λύσης μεσαίας μνήμης	1016
Κόστος αρχικής λύσης μεγάλης μνήμης	3574
Κόστος καλύτερης λύσης	754

Το κόστος της λύσης μεσαίας και μεγάλης μνήμης είναι αρκετά μεγαλύτερο από αυτό της αρχικής οπότε επιλέχτηκε να μην γίνει επιπλέον τοπική αναζήτηση.

Επανάληψη	Αποτέλεσμα μικρής μνήμης	Καλύτερη γνωστή λύση
1 ^η	756	
2 ^η	768	
3 ^η	754	
4 ^η	763	
5 ^η	763	
Μ.Ο	760	654

- **eil101Q185**

Χαρακτηριστικά του προβλήματος:

χωρητικότητα	185
κόμβοι	101

Κόστος αρχικής λύσης	807
Κόστος αρχικής λύσης μεσαίας μνήμης	801
Κόστος αρχικής λύσης μεγάλης μνήμης	3722
Κόστος καλύτερης λύσης	736

Τοπική αναζήτηση στη λύση της μεσαίας μνήμης.

Τελικές δοκιμές

επανάληψη	Αποτέλεσμα μικρής μνήμης	Καλύτερη γνωστή λύση
1 ^η	736	
2 ^η	736	
3 ^η	750	
4 ^η	736	
5 ^η	742	
Μ.Ο	740	629

- **eil101Q85**

Χαρακτηριστικά του προβλήματος:

χωρητικότητα	85
κόμβοι	101

Τοπική αναζήτηση στη λύση της μεσαίας μνήμης.

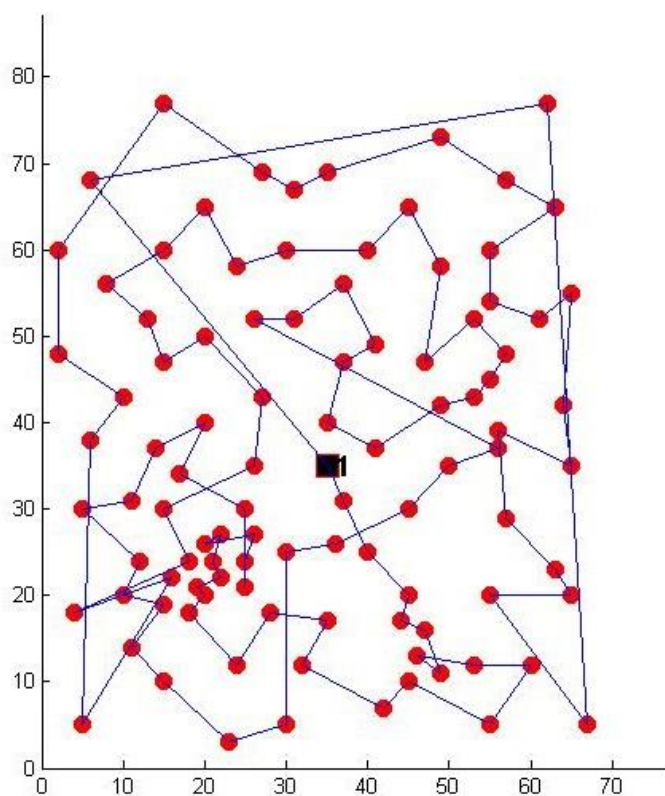
Τελικές δοκιμές

Κόστος αρχικής λύσης	916
Κόστος αρχικής λύσης μεσαίας μνήμης	885
Κόστος αρχικής λύσης μεγάλης μνήμης	3584
Κόστος καλύτερης λύσης	768

Επανάληψη	Αποτέλεσμα μικρής μνήμης	Καλύτερη γνωστή λύση
1 ^η	768	
2 ^η	773	
3 ^η	768	
4 ^η	771	
5 ^η	768	
M.O	769,6	662

Σχεδιάγραμμα γ. μια διαδρομή με 101 κόμβους:

Diadromh



Σημείωση: Παρατηρούμε πως υπάρχει περεταίρω αύξηση στον αριθμό των κόμβων συνεπώς προχωρήσαμε σε αλλαγές όπως και στα προβλήματα των 76 κόμβων, αυξάνοντας περισσότερο το μέγεθος των μνημών και μειώνοντας τις επαναλήψεις.

Θα πρέπει να παρατηρήσουμε ότι παρά της αλλαγές στις παραμέτρους ο αλγόριθμος έχει την ίδια, σχετικά, αποδοτικότητα και στις 3 κατηγορίες προβλημάτων.

Ανάλυση αποτελεσμάτων και συμπεράσματα

Ο αλγόριθμος της περιορισμένης αναζήτησης είναι εκ του αποτελέσματος ένας αρκετά αποδοτικός αλγόριθμος. Σημαντικό, βεβαία, είναι και η χρήση των κατάλληλων παραμέτρων έτσι ώστε να επιτεχθούν και τα βέλτιστα αποτελέσματα. Όπως είδαμε παραπάνω σε κάθε πρόβλημα που επιλύθηκε υπήρχαν διαφοροποιήσεις κυρίως ως προς τον αριθμό των κόμβων με αποτέλεσμα να χρειαστεί κάποιες παράμετροι του προβλήματος να αλλάξουν.

Επίσης, πρέπει να παρατηρήσουμε ότι το περιβάλλον της matlab είναι αρκετά πιο αργό από άλλα προγράμματα όπως η C. Αυτό είχε ως αποτέλεσμα να περιοριστεί αρκετά ο αριθμός των επαναλήψεων ειδικά στα προβλήματα με τους 101 κόμβους και συνεπώς να μην βελτιωθούν περεταίρω τα αποτελέσματα.

Όσων αφορά, τα αποτελέσματα, παρατηρούμε ότι κατά τις στρατηγικές που χρησιμοποιούμε τα αποτελέσματα της μεσαίας αλλά και της μεγάλης μνήμης απέχουν αρκετά από την αντίστοιχη βέλτιστη. Αυτό συμβαίνει διότι, κατά την κατασκευή αυτών των λύσεων οι περιορισμοί του προβλήματος δεν επιτρέπουν την ακριβή ακολουθία της κάθε στρατηγικής. παρόλα αυτά, οι συγκεκριμένες λύσεις χρησιμοποιούνται ως αρχικές και τοποθετούνται με την σειρά τους στον αλγόριθμο. Με αυτόν τον τρόπο ο στόχος των στρατηγικών που αποτελεί την αναζήτηση σε νέες γειτονίες λύσεων επιταχύνεται.

Για να συνοψίσουμε, τα αποτελέσματα της εργασίας επέτυχαν τον αρχικό τους στόχο πλησιάζοντας αρκετά τα βέλτιστα. Κάποιες προτάσεις για περεταίρω βελτίωση θα ήταν η χρήση κάποιας εξελιγμένης τοπικής αναζήτησης, η χρήση του αλγορίθμου με προσθήκη περιορισμών χρόνου και απόστασης, η χρήση παραθύρων χρόνου καθώς και η αντιμετώπιση και εστίαση σε κάθε πρόβλημα ξεχωριστά.

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΕΛΛΗΝΙΚΗ

1. Ιωάννης Μαρινάκης, Μυγδάλας Αθανάσιος, 2008, “Σχεδιασμός και Βελτιστοποίηση της εφοδιαστικής αλυσίδας”

ΞΕΝΗ

- 2.M. W. P. Savelsbergh and M. Sol. The general pickup and delivery problem. *Transportation Science*, 1995.
3. H. Hernández-Pérez and J. J. Salazar-González. The one-commodity pickup-and-delivery travelling salesman problem
- 4.S. Anily and J. Bramel. Approximation algorithms for the capacitated traveling salesman problem with pickups and deliveries. 1999
5. K. Ganesh, T.T. Narendran, 2006, ‘A cluster-and-search heuristic to solve the vehicle routing problem with delivery and pick-up’.
- 6.José Brandao, 2005, ‘A new tabu search algorithm for the vehicle routing problem with backhauls’
7. S. Anily and J. Bramel. Approximation algorithms for the capacitated traveling salesman problem with pickups and deliveries. *Naval Research Logistics*, 1999.
8. S. Anily and G. Mosheiov. The traveling salesman problem with delivery and backhauls. *Operations Research Letters*, 1994.
9. G. Mosheiov. The traveling salesman problem with pickup and delivery. *European Journal of Operational Research*, 1994

10. M. W. P. Savelsbergh and M. Sol. The general pickup and delivery problem. *Transportation Science*, 1995.
11. R. Baldacci, E. Hadjiconstantinou and A. Mingozzi. "An exact algorithm for the traveling salesman problem with deliveries and collections", 2003.