

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
Σχολή Μηχανικών Παραγωγής και
Διοίκησης
Τομέας Επιστήμης Αποφάσεων



Διπλωματική μελέτη με θέμα:

Μοντελοποιήσεις και επίλυσις
γενικευμένου προβλήματος
δρομολόγησης οχημάτων χρήσει
αλγορίθμου περιορισμένης αναζήτησης



Συντάξας:
Φραντζεσκάκης Γεώργιος

Επιβλέπων:
Δρ. Μαρινάκης Ιωάννης
(επίκουρος καθηγητής)

Εξεταστική επιτροπή :

Αντωνιάδης Αριστομένης
Σταυρουλάκης Γεώργιος Ε.

Χανιά 2016

Ευχαριστίες

Είναι τιμή να αποφοιτώ από το Πολυτεχνείο Κρήτης και σ' αυτό ευχαριστώ όσους καθηγητές έδειξαν υπομονή και μεταδοτικότητα, ειδικές ευχαριστίες εις τον επιβλέποντα καθηγητή μου, Επίκουρο Δρ. Ιωάννη Μαρινάκη για τις γνώσεις που μου μετάδωσε καθ' όλη τη διάρκεια των σπουδών μου, επιπλέον ευχαριστώ τον Δρ. Αριστομένη Αντωνιάδη για τη μεταδοτικότητα και τις γνώσεις του στο Corel Draw , του αφιερώνω τα όμορφα σχέδια – διαγράμματα της εργασίας . Τέλος να ευχαριστήσω την οικογένεια μου για την οικονομική και ψυχική ενίσχυση που μου παρείχαν.

Περίληψις

Αντικειμενικός σκοπός της επιχειρησιακής ερεΰνης αποτελεί η υλοποίηση διαδικασιών και επίτευξης στόχων υπό περιορισμούς πόρων (χρηματικών, πρώτων υλών κ.α) . Στο γενικότερο αυτό πλαίσιο εντάσσεται και η λήψη αποφάσεων περίξ του προβλήματος διανομής προϊόντων – υπηρεσιών μέσω δρομολόγησης οχημάτων, για μια επιχείρηση ή εν γένει παραγωγική μονάδα. Η επαναστατική χρήση των Η/Υ δύναται να προσφέρει υποστήριξη στα κέντρα αποφάσεων. Καθολική χρήση συγκεκριμένων αλγοριθμικών μεθόδων όπως αυτή της μεθευρετικής αναζήτησης, αλλά και άλλων άπληστων τεχνικών εκ του πρακτέου έχουν αποδειχτεί ως οι πλέον αποτελεσματικότερες. Ο λόγος για αυτό είναι η συνδυαστική αφενός πολυπλοκότητα και αφετέρου η ανάγκη για δυναμική εναλλαγή των αποφάσεων σε πραγματικό χρόνο. Συνέπεια των ανωτέρω είναι ο μονόδρομος για ανάπτυξη ενός *ευέλικτου* λογισμικού για το πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα (Vehicle Routing Problem with Time Windows - VRPTW), με έμφαση στην *ακρίβεια* και την *ταχύτητα* εκτελέσεως. Σύγχρονες γλώσσες προγραμματισμού όπως η *Java* καθιστούν την προσπάθεια αυτή ακόμα πιο ανταγωνιστική. Στην εργασία ευρέθη μια καινοτόμος αλγοριθμική μεθοδολογία ονόματι « *Αναζήτηση μεταφορών με άπληστη αρχικοποιημένη δρομολόγηση – Greedy Initialized Routing Search On Transportations (GIRSOT)* ». Η εργασία απαρτίζεται από πέντε κεφάλαια με τη λογική το «διαίρει και βασίλευε» επιχειρείται η λύσις του προβλήματος υπό τη λύση πολλών μικρότερων προβλημάτων, έτσι στο πρώτο κεφάλαιο γίνεται παρουσίαση του προβλήματος, στο δεύτερο προτείνεται ένας καλός αλγόριθμος ομαδοποίησης των πελατών (3-PCC). Στο τρίτο κεφάλαιο γίνεται μια προσπάθεια κατασκευής μιας μεθόδου βελτιστοποίησης επί των δρομολογίων. Το αποτέλεσμα που προκύπτει δεν είναι ούτε απογοητευτικό αλλά και ούτε αρκετό. Στο τέταρτο κεφάλαιο προτείνεται ένας ακόμα αλγόριθμος, ο αλγόριθμος σποράς (seeding algorithm) ο οποίος δίνει στον αλγόριθμο GIRSOT την πολυεναρκτήρια ιδιότητα του. Επόμενο βήμα θα είναι μια ύστατη προσπάθεια μείωσης ακόμα περισσότερο του κόστους, με τη μέθοδο μαζικής ανάθεσης - group relocate. Αν όλες οι ανωτέρω διαδικασίες αποτύχουν εφαρμόζεται η *χρονική υποτίμησις* (Devaluation Time Algorithm – BTA), η οποία εξαναγκάζει τη 3-PCC να δώσει νέες αρχικές λύσεις με περισσότερα οχήματα. Τέλος στο κεφάλαιο πέντε επιλέγεται το δείγμα απ' τα προβλήματα του Solomon n100 και των Gehring & Homberger's για να δειχθεί η αποτελεσματικότητα του προτεινόμενου αλγορίθμου.



Κάτοχος των πνευματικών δικαιωμάτων παραμένει ο καταθέτης /συγγραφέας ή τρίτα πρόσωπα εφόσον τα πνευματικά δικαιώματα έχουν παραχωρηθεί από τον συγγραφέα, είτε είναι φοιτητής (προπτυχιακός ή μεταπτυχιακός), είτε μέλος ΔΕΠ, ο οποίος καταθέτης/συγγραφέας ορίζει την άδεια χρήσης του τεκμηρίου που καταθέτει και εκχωρεί στο Πολυτεχνείο Κρήτης το δικαίωμα διάθεσης του μέσα από το Διαδίκτυο (με την αποδοχή της άδειας διάθεσης του τεκμηρίου του στο ΙΑ).

Άδεια χρήσης τεκμηρίου:



CC BY-NC-ND: Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Όχι Παράγωγα Έργα 4.0 Διεθνές

Το Έργο είναι ελεύθερο:

- για Διανομή: για αναπαραγωγή, διανομή και παρουσίαση του Έργου στο κοινό

Υπό τις ακόλουθες προϋποθέσεις:

- Αναφορά Δημιουργού: Ο χρήστης θα πρέπει να κάνει αναφορά στο έργο με τον τρόπο που έχει οριστεί από το δημιουργό ή το χορηγούνται την άδεια.
- Μη Εμπορική Χρήση: Ο Χρήστης δεν μπορεί να χρησιμοποιήσει το έργο αυτό για εμπορικούς σκοπούς.
- Όχι Παράγωγα Έργα: Ο Χρήστης δεν μπορεί να αλλοιώσει, να τροποποιήσει ή να δημιουργήσει (πάνω από) αυτό τό έργο αυτό.

Περιεχόμενα

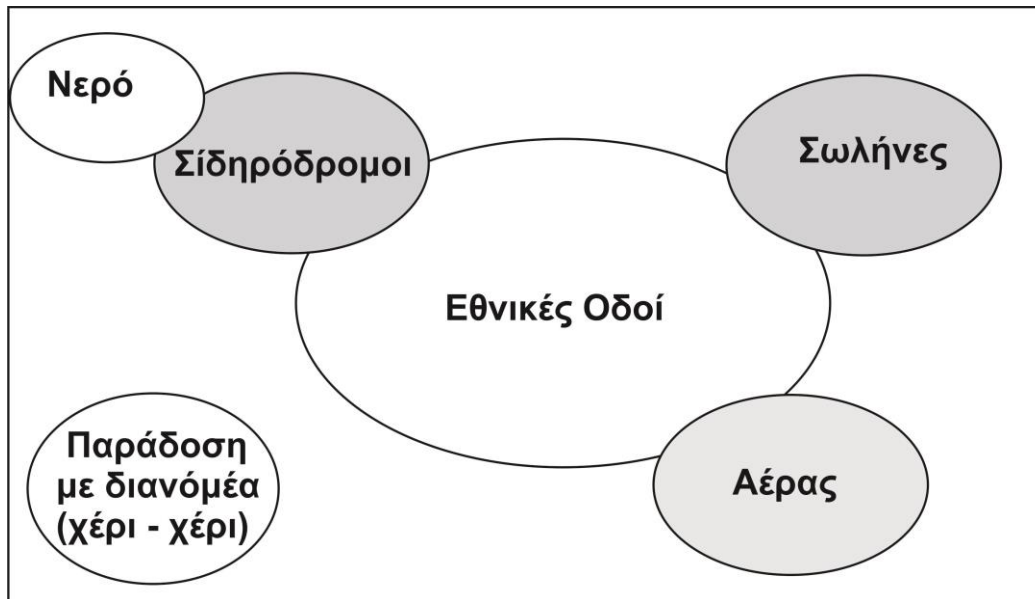
1. Εισαγωγή.....	7
1.1 Περί εφοδιασμού προϊόντων.....	7
1.2 Περί καθορισμού διαδρομών.....	9
1.3 Μαθηματική προτυποποίηση προβλήματος δρομολόγησης στόλου υποκείμενου χρονικών περιορισμών.....	11
2. Αλγόριθμος ομαδοποίησης 3-PCC.....	15
2.1 Εισαγωγικά.....	15
2.2 Ανάλυση της μεθόδου.....	17
2.2.1 Φάση construction.....	19
2.2.2 Φάση compression.....	22
2.2.3 Φάση adjustment.....	24
2.3 Αποτελέσματα – Συμπεράσματα.....	27
3. Βελτιστοποιήσεις δρομολογίου.....	31
3.1 Εισαγωγικά.....	31
3.2 Απαγορευμένη προσαρμοστικής μνήμης αναζήτηση.....	32
3.3 Άπληστη γενετική αναζήτηση.....	44
4. Αλγόριθμος GIRSOT.....	49
4.1 Μέθοδος σποράς seeding algorithm.....	50
4.2 Μέθοδος ομαδικής ανάθεσης group relocate.....	50
4.3 Μέθοδος χρονικής υποτίμησης DTA.....	53
4.4 Ενοποίηση όλων αλγόριθμος GIRSOT.....	53
5. Αποτελέσματα – Συμπεράσματα.....	59
5.1 Προβλήματα βιβλιογραφίας.....	59
5.2 Σύγκριση αποτελεσμάτων GIRSOT με βιβλιογραφία.....	60
5.3 Συμπεράσματα	73
6. Ευρετήριο.....	74
7. Παράρτημα.....	77
8. Βιβλιογραφία.....	80

Κεφάλαιο 1

Εισαγωγή

1.1 Περί Εφοδιασμού προϊόντων

Η παρούσα μελέτη υπάγεται εις το τομέα της επιχειρησιακής ερεύνης, συγκεκριμένα εις το τομέα λήψης αποφάσεων στη διαχείριση της εφοδιαστικής αλυσίδας (supply chain management) και κατ' επέκτασιν εις το τομέα της υπολογιστικής ερεύνης (computational research). Η κεντρική ιδέα της διαχείρισης εφοδιαστικής αλυσίδας είναι η εφαρμογή μιας ολικής προσέγγισης συστημάτων, δια τη διαχείρισης της ροής πληροφοριών, των υλικών και των υπηρεσιών από τους προμηθευτές πρώτων υλών μέσω εργοστασίων και αποθηκών ως τον τελικό αποδέκτη – καταναλωτή. Σχετικώς πρόσφατες τάσεις όπως η εξωτερική ανάθεση και η μαζική εξατομίκευση, οι οποίες αναγκάζουν τις επιχειρήσεις να αναζητούν ευέλικτους τρόπους ανταπόκρισης στη ζήτηση των πελατών. Αυτές εστιάζουν στη βελτιστοποίηση (optimization) των καθαρών δραστηριοτήτων για τη μεγιστοποίηση της ταχύτητας απόκρισης στις μεταβολές των πελατειακών προσδοκιών. Υποσύνολο της εφοδιαστικής αλυσίδας αποτελεί η διοίκηση διανομής προϊόντων και υπηρεσιών (logistics) .Το πρόβλημα έγκειται στο πως θα μεταφέρουμε αγαθά από τις εγκαταστάσεις στους καταναλωτές, είναι αρκετά περίπλοκο και επηρεάζει το κόστος του προϊόντος. Λεπτές οι ισορροπίες μεταξύ της τιμής μεταφοράς του προϊόντος και της ταχύτητας μεταφοράς αυτού, καθώς και της ευελιξίας να αντιδρούμε στις αλλαγές. Τα πληροφοριακά συστήματα (applications) παίζουν μεγάλο ρόλο στο συντονισμό των δραστηριοτήτων και περιλαμβάνουν πληροφορίες όπως η κατανομή πόρων, η διαχείριση των αποθεμάτων, ο προγραμματισμός και η παρακολούθηση των παραγγελιών. Η πλήρης ανάλυση αυτών των θεμάτων ξεπερνάει τα όρια της παρούσης μελέτης, ωστόσο αναλύεται επαρκώς μια νέα αποτελεσματική μεθοδολογία ανάπτυξης λογισμικού, κατάλληλου για υποστήριξη αποφάσεων πάνω στη διανομή προϊόντων. Μια απόφαση – κλειδί είναι πώς θα μεταφερθεί το προϊόν, στο σχήμα 1 απεικονίζονται οι εναλλακτικές .



Σχήμα.1 Διεργασίες Logistics

Εν γένει υπάρχουν έξι ευρέως αναγνωρίσιμα μέσα μεταφοράς: Αυτοκινητόδρομοι (φορτηγά) , νερό (πλοία) , αέρας (αεροπλάνα), σιδηρόδρομος (αμαξοστοιχίες) , αγωγοί και παραδώσεις χέρι – χέρι. Εδώ θα αναφερθούμε εις τους αυτοκινητοδρόμους (φορτηγά). Στη πραγματικότητα, λίγα προϊόντα μεταφέρονται χωρίς τη χρήση εθνικών οδών. Οι αυτοκινητόδρομοι προσφέρουν ευελιξία στη μεταφορά αγαθών εις οποιαδήποτε περιοχή, εκτός από εκείνες που περικλείονται από υγρό στοιχείο. Το μέγεθος , το βάρος και ο όγκος του προϊόντος είναι όλα διαχειρίσιμα με αυτό το μέσον μεταφοράς.

Στην άμεση διακόμιση (cross docking) [3] χρησιμοποιούνται ειδικές ενοποιημένες αποθήκες όταν τα φορτία από διάφορες πηγές συνδυάζονται μεταξύ των, με μεγαλύτερα φορτία σε κοινό προορισμό. Η στρατηγική χρησιμοποιείται σε αυτές τις ενοποιημένες αποθήκες, όπου , έναντι των μεγαλύτερων φορτίων, διαιρούνται εις μικρότερα για παράδοση στις γύρω περιοχές. Κάτι τέτοιο μπορεί να γίνει με τόσο συντονισμένο τρόπο, ώστε να μην χρειάζεται να κρατήσουμε τα προϊόντα σε απόθεμα (με αντίστοιχο αποθεματικό κόστος). Συνήθως οι λιανοπωλητές λαμβάνουν εμπορεύματα από πολλούς προμηθευτές στις τοπικές αποθήκες, όπου τα εμπορεύματα ταξινομούνται άμεσα σε φορτία για παράδοση σε διάφορα καταστήματα. Η χρήση των συστημάτων άμεσης διακόμισης έχει σαν αποτέλεσμα οι αποθήκες να διαθέτουν πάντα το ελάχιστον αποθεματικό επίπεδον. Τέλος η συντόνισης των αποθεμάτων υλοποιείται μέσω αυτοματοποιημένων συστημάτων ελέγχου (κατάλληλο λογισμικό) .

1.2 Περί καθορισμού δρομολογίων

Η περιγραφική ροή μέχρι στιγμής είναι πολύ γενική, λεπτομέρειες στην αναφορά διανομής προϊόντων εκκινούν από το πώς ένα μοναδικό όχημα θα εξυπηρετήσει ένα σύνολο συγκεκριμένων πελατών που οι απαιτήσεις των δεν ξεπερνούν τη χωρητικότητα του. Το πρόβλημα είναι ευρέως γνωστό ως πρόβλημα πλανόδιου πωλητού (traveling salesman problem with time windows - TSPTW), αντικειμενικά αναζητείται η συντομότερη διαδρομή που θα μπορεί να εξυπηρετήσει εις το μέγιστο βαθμό το σύνολο πελατών αφού το όχημα εκκινεί και καταλήγει στην αποθήκη εντός ωραρίου που θα εξυπηρετεί και το ίδιο το όχημα (οδηγός, συνοδηγός). Το όχημα θα πρέπει να εξυπηρετήσει απ ευθείας το πελάτη και όχι χρονικούς τμηματοποιημένα, η εξυπηρέτηση του πελάτη δηλαδή θα πρέπει να γίνεται σε μια φάση. Επέκτασις του προβλήματος πλανόδιου πωλητού αποτελεί το πρόβλημα δρομολόγησης οχημάτων (vehicle routing problem - VRP) . Η μετάβασις από το ένα όχημα εις ένα στόλο αυτών επηρεάζεται από δυο κατηγορίες περιορισμών, πρώτη ο χωρικός, αν δηλαδή η χωρητικότητα του ενός οχήματος δεν επαρκεί δια να καλύψει την ολότητα της ζήτησης των πελατών. Έπειτα ο χρονικός, αν δηλαδή η κάλυψη της ζήτησης ενός οχήματος υπερβαίνει το χρονικό ορίζοντα προγραμματισμού (ωράριο οχήματος) ή και βαθύτερα τα ατομικά χρονικά περιθώρια (time windows) των πελατών . Έτσι δια να πραγματοποιηθεί μια δρομολόγηση απαιτούνται η οχήματα , η οδηγοί ένα συγκεκριμένο οδικό δίκτυο και μια αποθήκη εκκίνησης – τερματισμού. Γενικά η σωστή επίλυση του προβλήματος καθορίζει ένα σύνολο από διαδρομές, ικανοποιώντας τις απαιτήσεις των πελατών, μη παραβιάζοντας δηλαδή τους περιορισμούς και έχοντας ελαχιστοποιήσει το κόστος διανομής (στόλος και απόσταση) . Ενδεικτικές πληροφορίες σχετικές με τις δραστηριότητες διανομής είναι οι ακόλουθες:

- Το μέγεθος του στόλου που χρησιμοποιεί η επιχείρηση
- Ο αριθμός των οδηγών
- Ο αριθμός των διαδρομών που γίνονται καθημερινά και ο μέσος αριθμός στάσεων ανά αυτών
- Οι διαδρομές εντός και εκτός πόλεως
- Το συνολικό ετήσιο κόστος των δραστηριοτήτων διανομής
- Το κόστος των πληρωμάτων
- Οι μελλοντικές απαιτήσεις και προβλέψεις στο τομέα ενδεχόμενων βλαβών
- Η τρέχουσα υπολογιστική δύναμη της επιχείρησης για τη δυνατότητα υποστήριξης του δικτύου διανομής
- Ο συνδυασμός δρομολογίων με άλλες δραστηριότητες

Επιπλέον πληροφορίες για τα χαρακτηριστικά των πελατών που ο υπεύθυνος διανομής θα πρέπει να λάβει υπόψη τα εξής:

- Το στίγμα του πελάτη στη γεωγραφική περιοχή διανομής
- Τη ποσότητα του/των προϊόντων που απαιτεί (demand) ένας πελάτης να του παραδοθούν ή συλλεχθούν
- Τα χρονικά περιθώρια(time windows) που ο πελάτης δύναται να εξυπηρετηθεί, επί παραδείγματι σε περιπτώσεις όπου η τοποθεσία δεν είναι προσβάσιμη βάσει συγκοινωνιακών περιορισμών.
- Ο απαιτούμενος χρόνος δια τη παράδοση – παραλαβή, εξαρτώμενος από το είδος του οχήματος και τα μέρη του οδικού δικτύου που θα προσπελάσει (περιορισμοί ταχύτητας - κ.ο.κ)

Το ανθρώπινο δυναμικό που χρησιμοποιεί τα οχήματα υπόκειται και αυτό σε περιορισμούς, όπως συνδικαλιστικούς κανόνες εργασίας, πόσα διαλλείματα απαιτούνται ανά χιλιομετρική απόσταση, ποιά η μισθοδοσία και η σχέση αυτής με τις υπερωρίες. Συγκεκριμένα:

- Οκτώ υποχρεωτικές ώρες ύπνου / ημέρα
- Συνεχόμενες ώρες οδήγησης να μην ξεπερνούν τις δέκα
- Υποχρεωτική μια ημέρα ρεπό / εβδομάδα εργασίας

Η ελαχιστοποίηση του κόστους διανομής για μια επιχείρηση εξαρτάται από τη διανυθείσα απόσταση (άμεσο κόστος αγοράς καυσίμων και χιλιομετρικές φθορές οχήματος), το χρόνο επιστροφής στην αποθήκη (τυχόν επιβαρύνσεις της επιχείρησης μέσω υπερωριών) καθώς και πάγια κόστη, όπως απόκτησης, συντήρησης οχημάτων/στόλου και φορολογικών επιβαρύνσεων που τα συνοδεύουν. Αν στο πρόβλημα VRP λαμβάνονται υπόψη και χρονικοί περιορισμοί τότε αυτό αναφέρεται ως Vehicle Routing Problem with Time Windows -VRPTW, έτσι όλα τα οχήματα εκκινούν από την αποθήκη τη στιγμή μηδέν και πρέπει να έχουν επιστρέψει το πολύ στο χρονικό ορίζοντα προγραμματισμού. Η δε εξυπηρέτηση κάθε πελάτη φέρει κάποια χρονοκαθυστέρηση (φόρτωση – εκφόρτωση, υπογραφή – παραλαβή παραστατικού πληρωμής κτλ) και απαιτείται από αυτόν η εξυπηρέτηση να έχει ξεκινήσει πριν από μια συγκεκριμένη ώρα (due time), ενώ απαιτεί επίσης να μην αρχίσει πριν από κάποια άλλη (start time). Περιπτώσει που κάποιο όχημα αφικνείται σένα πελάτη πριν το start time το όχημα υποχρεώνεται να αναμείνει τη διαφορά σε ακινησία. Υπάρχουν δυο είδη χρονικών παραθύρων το προηγούμενο που περιγραφικέ ήτο το αυστηρό, υπάρχουν όμως και τα χαλαρά όπου φτάνοντας ένα όχημα πριν το start time μπορεί να αρχίσει αυτομάτως η εξυπηρέτηση (πρακτικά είναι εκφυλισμένο στο μηδέν start time) ανάλογα θα μπορεί ένας πελάτης να εξυπηρετείται μετά το due time με αντίστοιχη κάθε φορά ποινή δυσφήμισης (που είναι δύσκολο να υπολογιστεί ποσοτικά). Το πώς όλα αυτά μπορούν να συμπτυχθούν εις ένα μοντέλο λειτουργίας θα φανεί παρακάτω.

1.3 Μαθηματική προτυποποίηση του προβλήματος δρομολόγησης στόλου υποκείμενου χρονικών περιορισμών

Το VRPTW ορίζεται από ένα στόλο οχημάτων V , ένα σύνολο πελατών C και ένα προσανατολισμένο γράφημα G . Θεωρείται ότι ο στόλος είναι ομοιογενής, δηλαδή όλα τα οχήματα είναι ίδια. Το γράφημα αποτελείται από $|C| + 2$ κόμβους, όπου οι πελάτες συμβολίζονται με $1, 2, \dots, n$. Η αποθήκη είναι ο κόμβος 0 (αποθήκη «αναχώρησης») και ο κόμβος $n + 1$ (αποθήκη «επιστροφής»). Το σύνολο όλων των κόμβων, δηλαδή $0, 1, \dots, n + 1$ συμβολίζεται με N . Το σύνολο των τόξων συμβολίζεται με A και ορίζει τις συνδέσεις μεταξύ αποθήκης - πελατών και ενός πελάτη με κάποιον άλλο πελάτη. Δεν υπάρχουν τόξα τα οποία καταλήγουν στον κόμβο 0 ή τόξα τα οποία εκκινούν από τον κόμβο $n + 1$. Κάθε τόξο (i, j) με $i \neq j$ έχει ένα κόστος c_{ij} και ένα χρόνο t_{ij} στον οποίο μπορεί να συμπεριλαμβάνεται και ο χρόνος εξυπηρέτησης του πελάτη i . Το κόστος c_{ij} ισούται με την Ευκλείδεια απόσταση των πελατών i και j και ο χρόνος t_{ij} ισούται με το άθροισμα της Ευκλείδειας απόστασης των πελατών i και j και του χρόνου εξυπηρέτησης του πελάτη i . Εδώ επισημαίνεται ότι το όχημα θεωρείται απλουστευμένα ότι κινείται με σταθερή μοναδιαία ταχύτητα [παράβαλε 2.1]. Κάθε όχημα έχει χωρητικότητα q προϊόντων και κάθε πελάτης i έχει ζήτηση d_i προϊόντων. Κάθε πελάτης i έχει ένα χρονικό παράθυρο $[a_i, b_i]$ και ένα όχημα πρέπει να τον επισκεφθεί πριν την χρονική στιγμή b_i . Εάν το όχημα επισκεφθεί τον πελάτη i πριν την χρονική στιγμή a_i τότε πρέπει να περιμένει μέχρι να «ανοίξει» το χρονικό του παράθυρο ώστε να τον εξυπηρετήσει. Τα χρονικά παράθυρα των δύο αποθηκών θεωρείται ότι είναι ίδια μεταξύ τους και ίσα με $[a_0, b_0]$ τα οποία συμβολίζουν τον χρονικό ορίζοντα προγραμματισμού (ωράριο εργασίας). Τα οχήματα απαγορεύεται να αναχωρήσουν από την αποθήκη πριν την χρονική στιγμή a_0 και πρέπει να επιστρέψουν στην αποθήκη το αργότερο την χρονική στιγμή b_{n+1} .

Θεωρείται ότι τα q, a_i, b_i, d_i και c_{ij} είναι μη αρνητικοί ακέραιοι αριθμοί και ότι τα t_{ij} είναι θετικοί ακέραιοι αριθμοί. Επίσης θεωρείται ότι η τριγωνική ανισότητα ισχύει για τα c_{ij} και t_{ij} (δηλαδή ισχύει ότι $c_{ij} \leq c_{ih} + c_{hj}$ και $t_{ij} \leq t_{ih} + t_{hj}, \forall h, i, j \in N$). Μπορεί να προστεθεί ένας αριθμός σε κάθε κόστος c_{ij} χωρίς να αλλάξει η βέλτιστη λύση του VRPTW αφού η τριγωνική ανισότητα θα εξακολουθεί να ισχύει.

Η μαθηματική περιγραφή του VRPTW χρησιμοποιεί δύο σύνολα μεταβλητών απόφασης, τα σύνολα x και s . Σε κάθε τόξο (i, j) με $i \neq j, i \neq n + 1, j \neq 0$ και σε κάθε όχημα k ορίζεται η μεταβλητή απόφασης x_{ijk} ως:

$$x_{ijk} = \begin{cases} 1, & \text{αν το όχημα } k \text{ μεταβαίνει από το } i \text{ στο } j \\ 0, & \text{διαφορετικά} \end{cases}$$

Η μεταβλητή απόφασης s_{ik} ορίζεται για κάθε κόμβο i και κάθε όχημα k συμβολίζοντας την χρονική στιγμή που το όχημα k ξεκινάει την εξυπηρέτηση του πελάτη i . Στην περίπτωση που το όχημα k δεν εξυπηρετεί τον πελάτη i η μεταβλητή απόφασης s_{ik} στερείται νοήματος και επομένως η τιμή της δεν λαμβάνεται υπόψη. Υποθέτουμε ότι $\alpha_0 = 0$ και συνεπώς $s_{0k} = 0, \forall k$.

Ο στόχος είναι να σχεδιαστεί ένα σύνολο διαδρομών που να ελαχιστοποιεί το συνολικό κόστος ώστε:

1. Έκαστος πελάτης να εξυπηρετείται εξολοκλήρου μια φορά (μονοφασική εξυπηρέτηση)
2. Κάθε όχημα να εκκινεί από το κόμβο 0 και να τερματίζει στον $n+1$
3. Το πλήθος του στόλου να μην υπερβεί το διαθέσιμο $|V|$
4. Τα χρονικά παράθυρα των πελατών και η χωρητικότητα των οχημάτων να ικανοποιούνται

Ακολουθεί αναλυτικά η μαθηματική περιγραφή του VRPTW :

Αντικειμενικός σκοπός^{**} :

$$\min_{x_{ijl}, s_{ik}} [\sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}] \quad (1)$$

Υποκείμενου περιορισμών :

- Μονοφασικής εξυπηρέτησης:

$$\sum_{k \in V} \sum_{i \in N} x_{ijk} = 1, \forall i \in C \quad (2)$$

- Πεπερασμένης χωρητικότητας οχήματος:

$$\sum_{i \in C} d_i \sum_{j \in N} x_{ijk} \leq Q, \forall k \in V \quad (3)$$

- Εκκίνησης από αποθήκη:

$$\sum_{j \in N} x_{0jk} = 1, \forall k \in V \quad (4)$$

- Εξυπηρέτησης πελάτη:

$$\sum_{i \in V} x_{ijk} - \sum_{i \in V} x_{jik} = 0, \forall j \in N, k \in K \quad (5)$$

- Επιστροφής στην αποθήκη:

$$\sum_{i \in N} x_{i, n+1, k} = 1, \forall k \in V \quad (6)$$

- Αναμονής οχήματος σε πελάτη:

$$s_{ik} + t_{ik} - M_{ij}(1 - x_{ijk}) \leq s_{jk}, \forall i, j \in N, \forall k \in V \quad (7)$$

- Τήρησης χρονικού παραθύρου πελάτη:

$$a_i \leq s_{ik} \leq b_i, \forall i \in N, \forall k \in V \quad (8)$$

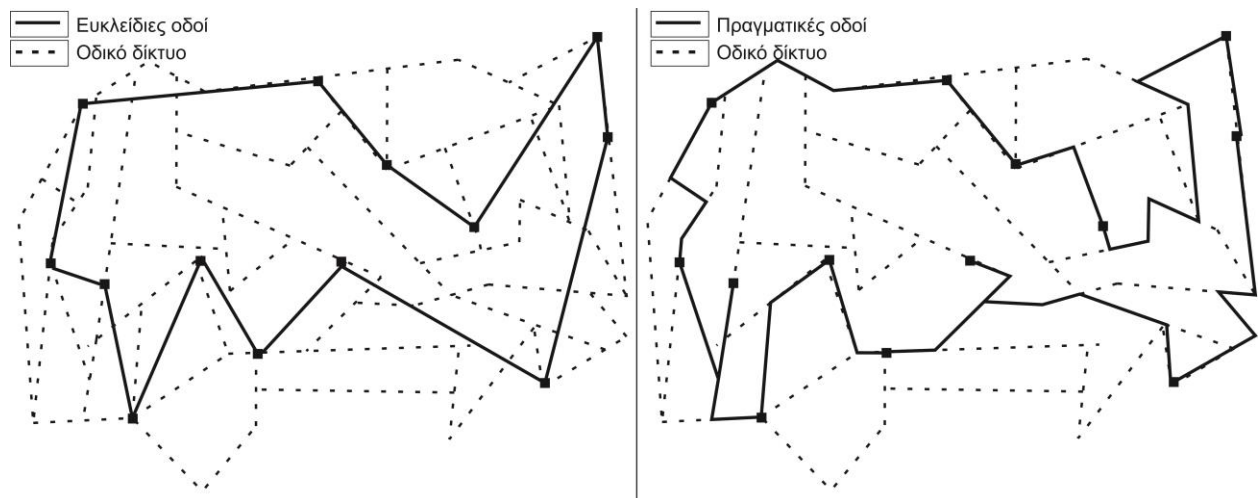
- Ακεραιότητας μεταβλητών απόφασης:

$$x_{ijk} \in \{0,1\} \quad (9)$$

- Πεπερασμένου διαθέσιμου στόλου:

$$\sum_{k \in N} \sum_{j \in N} x_{0jk} \leq |V|, \quad \forall k \in V, \forall j \in N \quad (10)$$

Πολύ σημαντικό στοιχείο δια να αναφερθεί πως στον αντικειμενικό σκοπό δεν υπάρχει η ελαχιστοποίηση του στόλου των οχημάτων, αλλά μόνον της διανυθείσας απόστασης. Το παραπάνω δεν επιφέρει πραγματική ελαχιστοποίηση στο κόστος διανομής. Μια τέτοια προσπάθεια θα εμπεριείχε και το δεύτερο αντικειμενικό κριτήριο που είναι το πλήθος των οχημάτων, ως αποτέλεσμα την αναγκαστική χρήση βαρών για έκαστο σκοπό. Η χρήση όμως βαρών είναι υποκειμενική από άποψης μαθηματικής ανάλυσης, στην πραγματικότητα τα βάρη αυτά υπολογίζονται μόνον με σοβαρή χρηματοοικονομική ανάλυση, και σε κάποιο κύκλο εργασιών – διανομών (πχ ετήσιο). Αυτός είναι και ένας λόγος όπως θα φανεί και στο κεφάλαιο 5 που τα παγκόσμια ελάχιστα σε απόσταση πολλές φορές απέχουν αρκετά από το βέλτιστο στόλο, είναι προφανές όμως ότι μια επιχείρηση θα προτιμήσει να διανύσει 100 χλμ. περισσότερα με ένα όχημα λιγότερο παρά το αντίθετο. Εις ότι αφορά τη χρήση ευκλείδειων αποστάσεων ως κόστος διανομής, αυτό στις περισσότερες περιπτώσεις δεν αποτελεί ρεαλιστική προσέγγιση [4]. Ο λόγος είναι ότι το πραγματικό οδικό δίκτυο μπορεί να απέχει πολύ σχέσει του αφηρημένου που θα προκύψει βάσει της απλοποίησης αυτής. Όπως φαίνεται και στο παρακάτω σχήμα οι πραγματικές διαδρομές θα είναι σχεδόν πάντα μεγαλύτερες από τις ευκλείδειες. Κάτι τέτοιο θεωρητικά θα μπορούσε να εξισωθεί μόνον αν υπήρχε ένα οδικό δίκτυο με πλήθος κόμβων να τείνει στο άπειρο και αποστάσεις μεταξύ αυτών να τείνουν στο μηδέν. Τονίζεται ότι η παρούσα μελέτη δεν ασχολείται με τη θεωρία δικτύων και τους αλγορίθμους που τη συνοδεύουν.



Σχήμα.2 Παράδειγμα Ευκλείδειας – πραγματικής διαδρομής

Ο αλγόριθμος GIRSOT που αναπτύχθηκε και αναλύεται στα επόμενα κεφάλαια δίνει μεγάλη έμφαση εις την ελαχιστοποίηση του στόλου. Αναλυτικότερα για τους περιορισμούς:

- Η αντικειμενική συνάρτηση (1) ελαχιστοποιεί τη συνολική διανυθείσα απόσταση όλων των διαδρομών.
- Οι περιορισμοί (2) εξασφαλίζουν ότι κάθε πελάτης θα εξυπηρετηθεί σε μια φάση(μια κ'έξω).
- Οι περιορισμοί (3) δηλώνουν ότι ένα όχημα δεν μπορεί να εξυπηρετήσει περισσότερους πελάτες από όσους του επιτρέπει η χωρητικότητά του.
- Το σύνολο των περιορισμών (4), (5) και (6) αποτελεί τους ροϊκούς περιορισμούς του προβλήματος. Πιο συγκεκριμένα οι περιορισμοί (4) δηλώνουν ότι κάθε όχημα πρέπει να αναχωρήσει από την αποθήκη 0, οι περιορισμοί (5) δηλώνουν ότι όταν ένα όχημα εξυπηρετήσει κάποιο πελάτη πρέπει να αναχωρήσει για κάποιον άλλο προορισμό και τέλος οι περιορισμοί (6) δηλώνουν ότι όλα τα οχήματα πρέπει να επιστρέψουν στην αποθήκη $n + 1$.
- Οι περιορισμοί (7) δηλώνουν ότι ένα όχημα k δεν μπορεί να επισκεφθεί τον πελάτη j πριν την χρονική στιγμή $s_{ik} + t_{ij}$ εαν μεταβαίνει από τον πελάτη i στον πελάτη j . Οι σταθερές M_{ij} μπορούν να πάρουν μία αυθαίρετα μεγάλη και κοινή τιμή ή να χρησιμοποιηθεί η τιμή $\max\{b_i + t_{ij} - \alpha_j\}, (i, j) \in A$.
- Οι περιορισμοί (8) εξασφαλίζουν ότι τα χρονικά παράθυρα των πελατών θα ικανοποιηθούν.
- Οι περιορισμοί (9) εξασφαλίζουν ότι το σύνολο των μεταβλητών απόφασης x θα πάρει μόνο τις ακέραιες τιμές 0 και 1.
- Ο περιορισμός (10) αποτελεί ένα άνω φράγμα στον χρησιμοποιούμενο αριθμό οχημάτων. Τέλος πρέπει να αναφερθεί ότι ένα όχημα που δεν χρησιμοποιείται ακολουθεί την «κενή» διαδρομή $(0, n + 1)$.

Το πρόβλημα VRPTW αποτελεί γενίκευση των προβλημάτων πλανόδιου πωλητού (Traveling Salesman Problem – TSP) και VRP. Αν οι περιορισμοί (7) και (8) δεν ληφθούν υπόψη τότε το πρόβλημα χαλαρώνει στο VRP. Επιπλέον αν υπάρχει μόνο ένα όχημα διαθέσιμο τότε το πρόβλημα χαλαρώνει περαιτέρω στο TSP. Τέλος αν οι περιορισμοί χωρητικότητας των οχημάτων (3) δεν ληφθούν υπόψη τότε αχούμε το πρόβλημα m -TSPTW, ή αν υπάρχει μόνο ένα όχημα διαθέσιμο τότε έχουμε το πρόβλημα TSPTW.

Κεφάλαιο 2

Αλγόριθμος Ομαδοποίησης 3-PCC

2.1 Εισαγωγικά

Πρώτο και νευραλγικό στάδιο δια την δρομολόγηση ενός στόλου οχημάτων σένα πλήθος πελατών , αποτελεί η διακριτοποίηση αυτών βάσει κάποιου *μέτρου ομοιότητας* εν προκειμένω, της γεωγραφικής θέσεως. Η θέση αυτή δεν αποτελεί αντικείμενο ερεύνης ή απόδειξης, αλλά μια αναμφισβήτητη αλήθεια. Επί του παρόντος το πρόβλημα περιγράφεται ως:

Εύρισκε πλήθος m συστάδων, n πελατών , όπου έκαστη συστάδα να διακριτοποιείται σε συγκεκριμένη γεωγραφική περιοχή, περιλαμβάνουσα πελάτες εντός της. Διαισθητικά πελάτες στην ίδια συστάδα είναι «κοντινοί» ενώ άλλοι «μακρινοί»

Πρωταρχικά ερωτήματα που γεννιούνται:

- Ποίο το πλήθος πελατών για έκαστη συστάδα (ή ομάδα)
- Με ποιό τρόπο αυτοί θα επιλεγούν
- Ποίο το πλήθος των συστάδων

Τα ερωτήματα αποτελούν δυο όψεις του ίδιου νομίσματος. Τρείς οι περιορισμοί που καθορίζουν τις αποφάσεις (που όπως θα φανεί είναι δυο), ο πρώτος έχει ήδη αναφερθεί ως γεωγραφική περιοχή, αντιπροσωπεύει τη διανυθείσα απόσταση που θα χρησιμοποιηθεί ως μέτρο απόδοσης του όλου εγχειρήματος (σε συνδυασμό με το μέγεθος του στόλου). Υπερσύνολο του προηγούμενου είναι ο *χρόνος*, που στη παρούσα φάση θα ληφθεί υπόψη μόνον εις επίπεδο δρομολογίου. Διαφορετικά κάποιος θα μπορούσε να πει χρόνος προγραμματισμού (route time) ή απλούστερα ωράριο εργασίας του οχήματος (οδηγού, συνοδηγού) ή χρόνος επιστροφής στην αποθήκη. Εδώ επιτάσσεται να γίνει μια διευκρίνιση, χρόνος και απόσταση (από σημείο A στο B) είναι ίδιες μεταβλητές , συγκεκριμένα:

$$\chi\rho\acute{o}\nu\omicron\varsigma\ \omicron\chi\acute{\eta}\mu\alpha\tau\omicron\varsigma(A,B) = \frac{\text{Απόσταση}(A,B)}{\text{μέση ταχύτητα}} \xrightarrow{\text{ταχύτητα}=1} \chi\rho\acute{o}\nu\omicron\varsigma(A,B) = \text{απόσταση}(A,B) \quad (11)$$

Ότι ακολουθεί στη συνέχεια παραδέχεται τη μέση ταχύτητα ως μοναδιαία. Αυτό φυσικά θα μπορούσε να ανατρέψει σημαντικά μια δρομολόγηση καθώς όπως είναι φυσικό, κάθε γεωγραφική περιοχή ή/και κάθε οδός συμμετέχει γενικά με διαφορετική ταχύτητα (εθνικές οδοί, κατοικημένες περιοχές κτλ.). Η παρούσα μελέτη δεν ασχολείται με αυτό, καθώς δεν μελετάται η διάβασις των οχημάτων από οδικά/θαλάσσια/εναέρια δίκτυα. Το δίκτυο είναι ένα άπειρο σύνολο κόμβων με αμελητέες μεταξύ των αποστάσεις. Η θέση αυτή επιτρέπει σένα όχημα να κινηθεί απ' το σημείο $A(X_A, Y_A) \rightarrow B(X_B, Y_B)$ ευθέως με κάποια συνάρτηση μεταφοράς⁽¹⁾. Εφεξής σε κάθε υπολογισμό απόστασης, η συνάρτηση θα είναι γραμμική και η απόσταση θα ονομάζεται ευκλείδειος. Συγκεκριμένα :

$$\text{Απόσταση}(A,B) = \sqrt{(X_A - X_B)^2 + (Y_A - Y_B)^2} \quad (12)$$

Αν η συγκεκριμένη παραδοχή είναι και ρεαλιστική ή όχι, εξαρτάται από τη πυκνότητα του δικτύου μεταφοράς και σε κάθε περίπτωση δεν αποτελεί αντικείμενο της παρούσης μελέτης.

⁽¹⁾ Σε περιπτώσεις οδικών δικτύων η συνάρτηση μεταφοράς θα μπορούσε να είναι τετραγωνική ή κυβική, με τρόπο τέτοιο που να προσομοιώνει όσο γίνεται καλύτερα μια μέση οδική διάβαση της δρομολογηθείσας περιοχής.

Δεδομένης της ευκλείδειας απόστασης και της (μέσης)μοναδιαίας ταχύτητας, η μέθοδος που αναλύεται παρακάτω υπολογίζει το ωράριο του οχήματος ή route time υποθέτοντας μηδενικούς χρόνους παραμονής (delays) στο κάθε πελάτη δηλαδή ελαστικά time windows. Υπολογίζονται κανονικά ο ατομικός χρόνος εξυπηρέτησης (service time), Συγκεκριμένα:

$$\chi\rho\acute{o}\nu\omicron\varsigma\ \delta\rho\omicron\mu\omicron\lambda\omicron\gamma\acute{\iota}\omicron\upsilon = \text{Απόσταση δρομολογίου} + \sum_{i=1}^n \chi\rho\acute{o}\nu\omicron\varsigma\ \epsilon\chi\upsilon\pi\eta\rho\acute{\epsilon}\tau\eta\sigma\eta\varsigma_i \quad (13)$$

Τρίτος (κατ' ουσίαν δεύτερος) είναι ο περιορισμός χωρητικότητας των οχημάτων του στόλου. Εφεξής όλα τα οχήματα θα θεωρούνται πανομοιότυπα συγκεκριμένης χωρητικότητας αδιάστατου όγκου. Εκ παραδρομής να αναφερθεί ότι δεν λαμβάνεται υπόψη ο χρονικός περιορισμός για κάθε πελάτη (time window). Κάτι τέτοιο δεν θα είχε κανένα απολύτως νόημα καθώς οι παραχθείσες λύσεις δεν είναι προϊόν βελτιστοποίησης αλλά μιας άπληστης στρατηγικής. Που όπως θα φανεί παρακάτω σε επίπεδο δρομολογίου (πλανόδιου πωλητού) τουλάχιστον οι λύσεις αυτές είναι κακής ποιότητας .

Μεγάλη και ευρέως διαδεδομένη κατηγορία αλγορίθμων είναι οι άπληστοι (greedy algorithm) ή αλλιώς μυωπικοί αλγόριθμοι. Χαρακτηρίζονται και ως *προσεγγιστικοί καθώς – συνήθως- παράγουν υποβέλτιστες λύσεις*. Κάποιος θα μπορούσε να πει:

« Η απληστία... είναι καλή. Η απληστία είναι σωστή. Η απληστία αποδίδει (!) »

Υπάρχουν αρκετά υπέρ αλλά και κατά της λογικής αυτής, για αρχή εις ότι αφορά την ομαδοποίηση η εμπειρία δείχνει ότι αποδίδει καλά και γρήγορα. Είναι δύσκολο να ορίσει κάποιος την απληστία ένας ορισμός θα μπορούσε να είναι ο εξής:

Ορισμός: Απληστος αλγόριθμος δομεί τη λύση σε μικρά βήματα, περνώντας μυωπικές αποφάσεις σε κάθε βήμα για τη βελτιστοποίηση κάποιου υποκειμένου κριτηρίου.

Εδώ το υποκείμενο κριτήριο είναι η απόσταση και επιθυμείται η ελαχιστοποίηση της. Δεν είναι τυχαίο ότι άνθρωπος και απληστία ταυτίζονται ανά τους αιώνες. Είναι άπειρα τα παραδείγματα που αποδεικνύουν ότι ο άνθρωπος ενεργεί κοντόφθαλμα. Δεν είναι όμως πάντα κακό αυτό, σε πάρα πολλές περιπτώσεις αποδεικνύεται τουλάχιστον σωτήριο. Αφηρημένα λέμε:

$$\frac{\text{Αποδοτικότητα απληστίας}}{\text{χρόνο}} \gg 1 \quad (14)$$

Δια την δημιουργία μιας αρχικής ημιεφικτής [παράβαλε 2.3] λύσης (semi feasible) γίνεται αποκλειστική χρήση άπληστων τεχνικών, πλεονεκτήματα είναι τα εξής:

- Ταχύτατη σύγκλιση ήτοι σημαντικές σχετικές βελτιώσεις της λύσης σε κάθε βήμα του.
- Βέβαιη σύγκλιση ακόμα και αν οι λύσεις είναι υποβέλτιστες .
- Εύκολη υλοποίηση.
- Υπερτερεί έναντι πολλών άλλων σε προβλήματα χρονοπρογραμματισμού.
- Είναι αποδοτικός (φέρει πολυωνυμικό χρόνο εκτέλεσης).

Μειονεκτήματα:

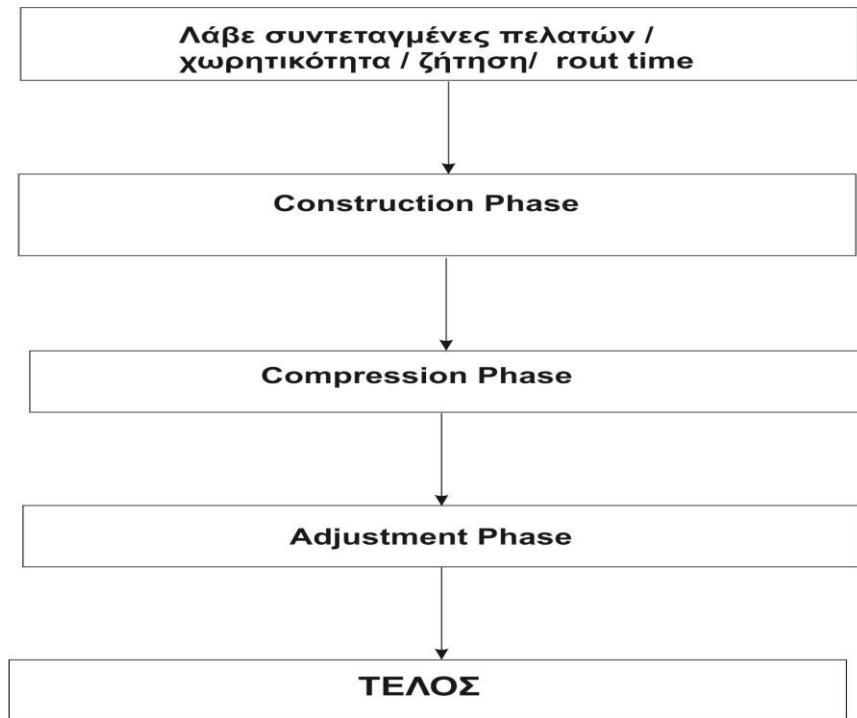
- Όχι εγγυημένα βέλτιστες λύσεις.
- Όσο η πολυπλοκότητα αυξάνεται τόσο η ποιότητα της λύσης μειώνεται.
- Άνιση σχέση πολυπλοκότητας και ποιότητας της λύσης.

2.2 Ανάλυση της μεθόδου

Η μέθοδος που ανεπτύχθη εμπνεύστηκε παλαιότερας μεθόδου [9] , σε εφαρμογή δρομολόγησης οχημάτων στερούμενων χρονικών περιορισμών. Είναι η Γεωγραφική

Ομαδοποίηση 3^{ων} Φάσεων (**3-Phased Centroid-based Clustering**) ή **3-PCC**. Συγκεκριμένα δομείται σε τρεις διαδοχικά εξαρτώμενες φάσεις απληστίας :

1. Κατασκευή – Construction (1st phase)
2. Συμπίεση – Compression (2nd phase)
3. Αναπροσαρμογή – Adjustment (3rd phase)



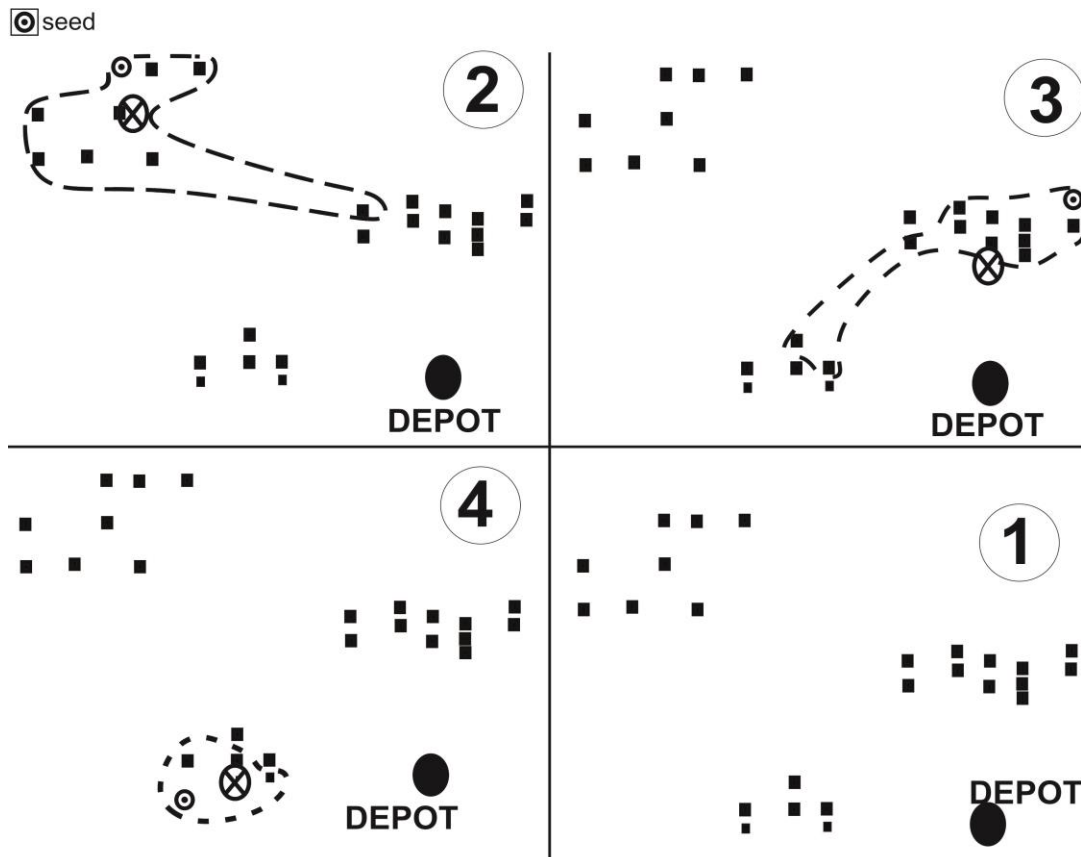
Σχήμα.3 Στάδια κατασκευής αρχικής λύσης

Λακωνικά, η φάση της κατασκευής γεννάει ένα πρώτο σπόρο - πελάτη και απ αυτόν εκκινεί τη πρώτη συστάδα, την οποία και τερματίζει όταν είτε το όχημα γεμίσει είτε αν ο χρόνος δρομολογίου παραβιαστεί. Η δεύτερη σε μια προσπάθεια «διόρθωσης» της απληστίας της προηγούμενης, προσπαθεί να επανατοποθετήσει (relocate) τους πελάτες, κάθε φορά της τελευταίας (χρονικά) συστάδας στις υπόλοιπες ούτως ώστε να την εξαλείψει. Τέλος η Τρίτη φάση χρήσι και πάλι relocate προσπαθεί να ανακατατάξει πελάτες στις ήδη υπάρχουσες συστάδες, ούτως ώστε σε κάθε κίνηση relocate να διακτιτοποιεί περισσότερο τις δυο εμπλεκόμενες συστάδες. Η φάση αναπροσαρμογής «διορθώνει» ακόμα περισσότερο την απληστία των δυο προηγούμενων. Παρακάτω γίνεται χρήση παραδείγματος διανομής 25 πελατών, χωρητικότητα οχήματος 200 μον.όγκου και route time 967. Οι πελάτες είναι τοποθετημένοι σε εμφανείς συστάδες, όπου ο 3-PCC αλγόριθμος θα κληθεί να εύρει.

2.2.1 Φάση Construction

Αναλυτικότερα για την φάση construction σημαντική κίνηση αποτελεί η επιλογή του πρώτου σπόρου (seed) ακολουθείται η λογική «έξω προς τα μέσα» κατά τούτο ο σπόρος κάθε φορά που ένα όχημα πρέπει να εκκινήσει από την αποθήκη (depot) επιλέγεται ο πιο απόμακρος διαθέσιμος πελάτης. Στην ερώτηση γιατί όχι ο πλησιέστερος, κάποιος πολύ εύκολα μπορεί να απαντήσει ότι επιλέγοντας κάποιον μακρινό στο δρόμο ενδιάμεσα, χωρίς αλλαγή της πορείας μπορεί το όχημα να εξυπηρετήσει και άλλους πελάτες. Έπειτα από το σπόρο αυτό κάθε φορά επιλέγει τον επόμενο πελάτη με το κριτήριο της πλησιέστερης γειτονικής – συστάδος (*nearest cluster - neighbour*). Σε αυτό το σημείο επιτάσσεται ο ορισμός ενός χαρακτηριστικού κάθε συστάδος. Μιλώντας πάντα για φυσικά μεγέθη επιλέγεται το γεωγραφικό κέντρο (*Geographical Canter*) ή **GC**. Μαθηματικά αυτό αποτελεί απλά το ζεύγος της μέσης συντεταγμένης και τετμημένης των πελατών της συστάδος ή κατά άλλους το κέντρο βάρους, συγκεκριμένα:

$$GC_i = (GC_x, GC_y) = \left(\frac{\sum_{i=1}^n x_i}{n}, \frac{\sum_{i=1}^n y_i}{n} \right) \quad (15)$$



Σχήμα.4 Μέθοδος 3-PCC / Στάδια πρώτης φάσης Construction

Δρομολόγηση:

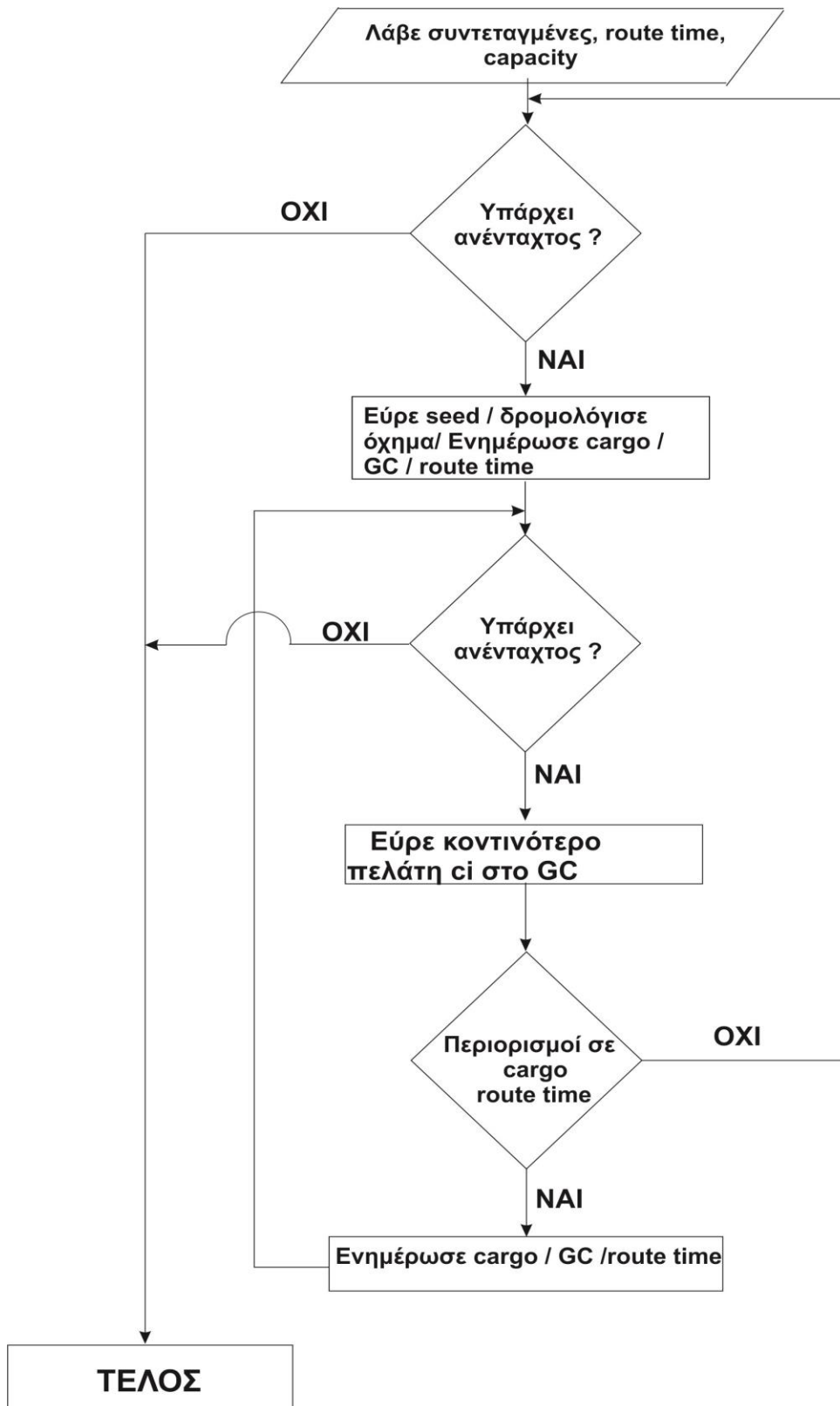
16	14	12	15	19	13	17	18	11	-	-	-
2	1	4	3	5	7	6	8	9	10	23	21
25	24	22	20	-	-	-	-	-	-	-	-

Πίνακας.1 Δρομολόγηση φάσης construction

Όπως φαίνεται παραπάνω θα επιλεγεί ως seed1 ο πελάτης 16 εν συνεχεία θα ενταχθούν οι 14, 12, 15, 19, 17, 18 και 11. Δεύτερο όχημα καλείται καθώς το πρώτο επιστρέφει σε χρόνο 923.51 και φορτίο 200. Ο δεύτερος seed2 = 2 και επόμενοι πελάτες οι 1, 4, 3, 5, 7, 6, 8, 9, 10, 23, 21, το όχημα επιστρέφει με φορτίο 180 καθώς ξεπεράστηκε το route time. Τέλος seed3 = 25 και επόμενοι οι 24, 22, 20. Επιστρέφει σε 393.57 και φορτίο 80. Ακολουθούν τα βήματα:

Βήμα 0 : Αν υπάρχει ανένταχτος → Βήμα 1 αλλιώς → Βήμα 4
Βήμα 1: Βρες τον σπόρο και δρομολόγησε νέο όχημα → Βήμα 1.2 1.2 : θέσε $GC \leftarrow (X_{seed}, Y_{seed})$, Φορτίο \leftarrow ζήτηση , Ενημέρωσε route time → Βήμα 3
Βήμα 3: Αν υπάρχει επόμενος πελάτης → Βήμα 3.1 αλλιώς → Βήμα 4 3.1: εντόπισε τον πλησιέστερο στο τρέχον GC αν πληροί περιορισμούς σε Φορτίο ,route time* → Βήμα 3.2 αλλιώς Βήμα 0 3.2: θέσε Φορτίο \leftarrow ζήτηση , ενημέρωσε GC, route time → Βήμα 3
Βήμα 4: <u>Τέλος</u>

Πίνακας.2 Ψευδοκώδικας φάσης construction



Σχήμα.5 Διάγραμμα ροής φάσης construction

2.2.2 Φάση Compression

Σε αυτό το σημείο θα υπολογιστεί το ελάχιστο μέγεθος του στόλου ή $fleet_{min}$ ως:

$$\lfloor fleet_{min} \rfloor = \frac{\text{συνολική ζήτηση}}{\text{χωρητικότητα οχήματος}} \quad (16)$$

Στο παράδειγμα η συνολική ζήτηση είναι $200 + 180 + 80 = 460$ μονάδες, με χωρητικότητα οχήματος στις 200, συνεπώς $460/200 = 2.3$ οχήματα. Εννοείται ότι σε περίπτωση δεκαδικού επιλέγεται το μεγαλύτερο προς τα άνω, εν προκειμένω 3 οχήματα. Το μέγεθος αυτό αποτελεί εκτίμηση του βέλτιστου (που δεν είναι ακόμα γνωστό) που όμως όπως θα φανεί [παράβαλε 2.3] είναι μια καλή εκτίμηση. Αυτό σημαίνει ότι η δρομολόγηση μπορεί να πραγματοποιηθεί με το λιγότερο $fleet_{min}$ οχήματα και είναι εύλογον η αναζήτηση να εκκινήσει από αυτό το κάτω όριο, καθότι είναι βαρύνουσας σημασίας η ελαχιστοποίηση τη συνάρτησης κόστους της δρομολόγησης [σχήμα 6], έκαστο όχημα φέρει τα κάτωθι κόστη:

- Απόκτησης οχήματος.
- Αποζημίωσης (ανά δρομολόγιο) οδηγού + κόστος ασφάλισης.
- Βλαβών και φθοράς του οχήματος.
- Παγίων φόρων επί του οχήματος.

Συνεπώς μια επιχείρηση σε καμία περίπτωση δεν θα ήθελε περιέχουσα χρήση κάποιου οχήματος.

Εις ότι αφορά τη φάση compression ελέγχεται αν ο παραχθείς στόλος της πρώτης φάσης είναι μεγαλύτερος του $fleet_{min}$. Αν ναι, πράττει όπως ήδη έχει αναφερθεί χωρίς εγγύηση ότι θα αποδώσει (καθότι είναι άπληστη) και δεδομένου ότι εδώ ελέγχεται αν ο χαλαρωμένος (κατά α%) περιορισμός ωραρίου παραβιάζεται.

Η χαλάρωση αυτή είναι εύλογη τα δρομολόγια είναι χαμηλής ποιότητας και υπάρχει μεγάλη πιθανότητα το ωράριο να μπορεί να βελτιωθεί σημαντικά. Διαφορετικά θα μπορούσε να ελεγχθεί ότι το ωράριο είναι υπέρ - εκτιμημένο συνέπεια της απληστίας. Στο συγκεκριμένο παράδειγμα δεν τίθεται θέμα συμπίεσης του στόλου. Στο βήμα 1 επιλέγεται για relocate ο πελάτης με τη μεγαλύτερη κάθε φορά ζήτηση (**Maximum Demand First**). Η σκέψη πίσω απ αυτό είναι ότι αν στα άλλα φορτηγά υπάρχει διαθέσιμος χώρος τότε τα κενά υπάρχει μεγαλύτερη πιθανότητα να μπορέσουν να καλυφθούν/μειωθούν όταν η ζήτηση που μεταφέρεται είναι φθίνουσα. Και πάλι άπληστη μέθοδος αναδεικνύει την ανωτερότητα της σε προβλήματα εμπλέκοντα το χρόνο. Ένα απλό παράδειγμα αποδεικνύει τη θέση αυτή.

Έστω 3 δρομολόγια. Επιδιώκεται να συμπιεστεί το 3^ο στα υπόλοιπα 2, έκαστο όχημα χωράει 100 μονάδες

Όχημα 1	15	15	-	Όχημα 1	15	15	70
Όχημα 2	35	35	-	Όχημα 2	35	35	30
Όχημα 3	30	70	-	Όχημα 3	-	-	-

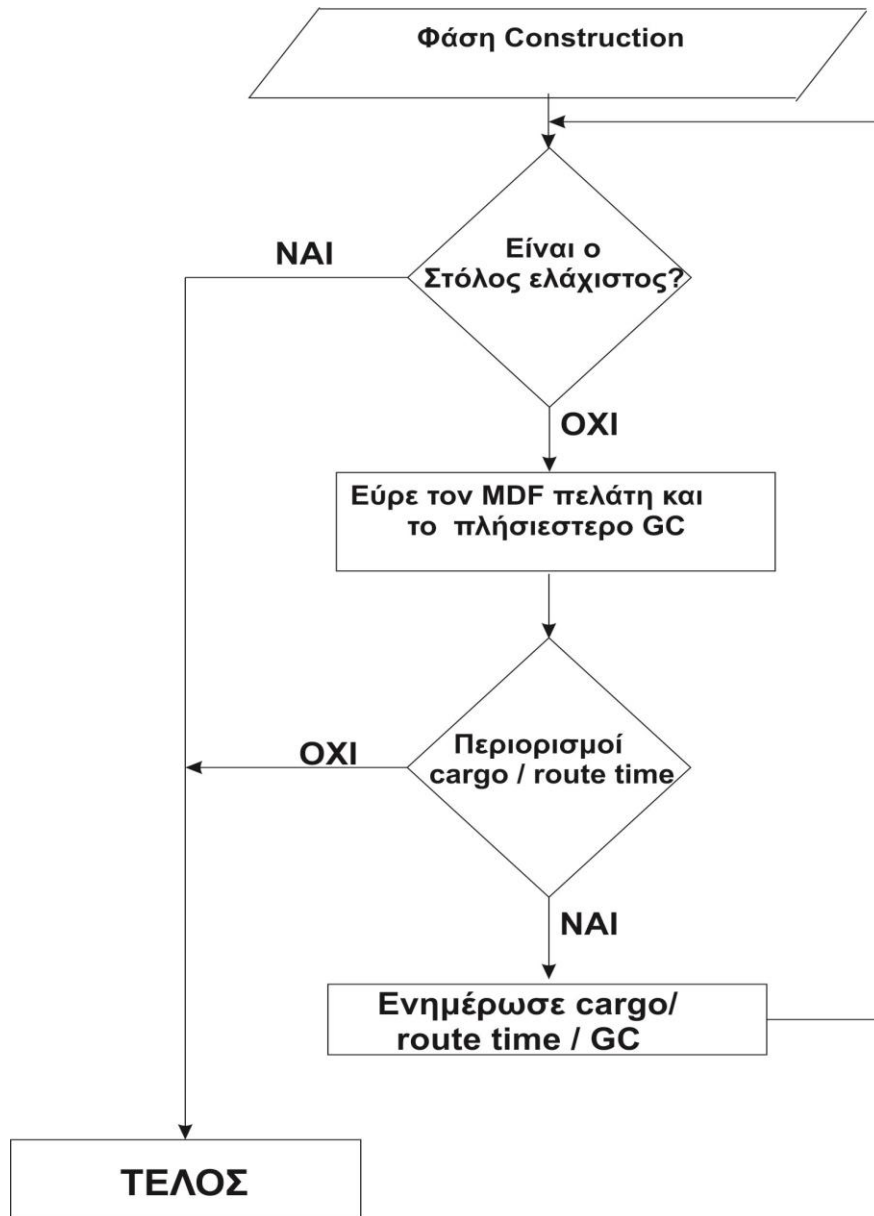
Πίνακας.3 Παράδειγμα επιλογής MDF

Αν πρώτα ο 1ος πελάτης του 3ου οχήματος πάει στο 1ο όχημα τότε η συμπίεση αποτυγχάνει. Αν όμως ληφθεί ο MDF πελάτης, που είναι ο δεύτερος με ζήτηση 70 η συμπίεση επιτυγχάνεται. Ακολουθούν τα βήματα της μεθόδου:

Βήμα 0: Αν υπάρχει επιπλέον δρομολόγιο → Βήμα 1 αλλιώς → Βήμα 4
Βήμα 1: Βρες MDF πελάτη → Βήμα 2
Βήμα 2: Βρες τη πλησιέστερη συστάδα και αν πληροί περιορισμούς σε φορτίο ,route time ⁽²⁾ → Βήμα 3 αλλιώς Βήμα 4
Βήμα 3: ενημέρωσε φορτίο , GC , route time → Βήμα 0
Βήμα 4: <u>Τέλος</u>

Πίνακας.4 Ψευδοκώδικας φάσης compression

(2) Ο Περιορισμός του χρόνου επιστροφής στην αποθήκη όπως υπολογίζεται από τη σχέση 3 χαλαρωμένος κατά ένα α%, έχει προταθεί 10% χαλάρωσις.

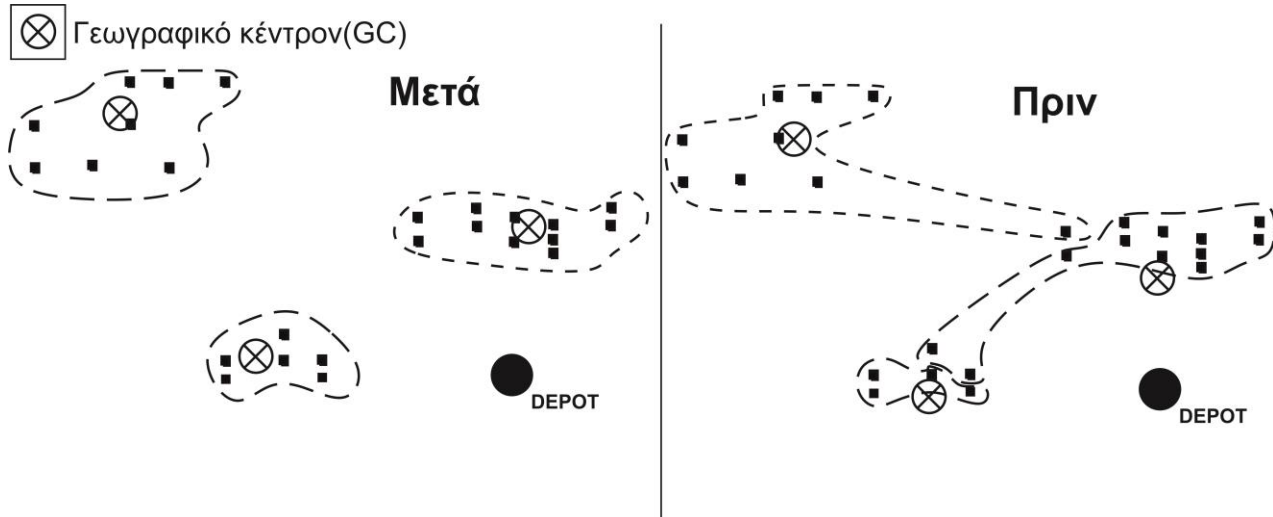


Σχήμα.6 Διάγραμμα ροής φάσης *compression*

2.2.3 Φάση *Adjustment*

Τέλος, στη φάση adjustment ο αλγόριθμος μεταφέρει (relocate) κάθε πελάτη (συστάδα δότης *giver*) στη πλησιέστερη κάθε φορά συστάδα (συστάδα λήπτης-receiver) λαμβάνοντας υπόψη το περιορισμό μόνον χωρητικότητας. Αν η μεταφορά – relocate είναι εφικτή και το κέντρο κάποιας άλλης συστάδας είναι πλησιέστερα στο πελάτη απ'οτι το κέντρο της δική του,

πραγματοποιείται. Η μέθοδος περατώνεται όσο δεν βρίσκει επόμενο relocate για 2 συνεχόμενους κύκλους επαναλήψεων. Όπως φαίνεται και στην εικόνα.2 είναι η σημαντικότερη φάση της ομαδοποίησης.



Σχήμα 7 Μέθοδος 3-PCC / Το πριν και το μετά της φάση adjustment

Αναλυτικότερα μετεφέρθησαν οι πελάτες :

- 11 από συστάδα 1 → 2
- 23 από συστάδα 2 → 3
- 21 από συστάδα 2 → 3

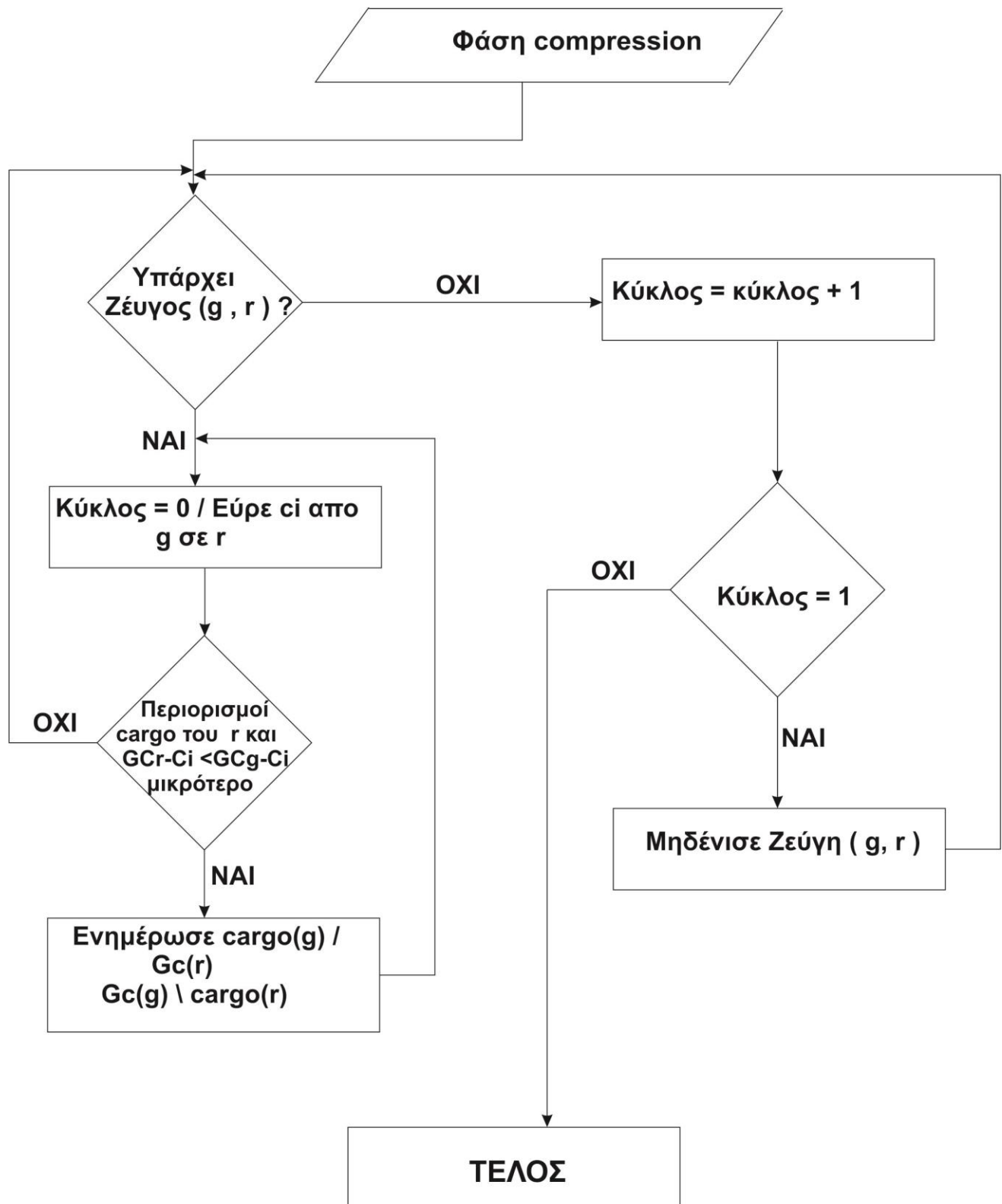
16	14	12	15	19	13	17	18	-	-	-	-
2	1	4	3	5	7	6	8	9	10	11	-
25	24	22	20	23	21	-	-	-	-	-	-

Πίνακας.5 Δρομολόγηση φάσης adjustment

Βήμα 0: Αν υπάρχει νέο ζεύγος συστάδων λήπτη – δότη → Βήμα 0.2 αλλιώς θέσε κύκλος \leftarrow κύκλος +1 → Βήμα 3 0.2: θέσε κύκλος \leftarrow 0 → Βήμα 1
Βήμα 1: Εύρε πελάτη από δότη σε λήπτη* αν χωράει στο λήπτη* και η απόσταση από το GC_g Είναι μεγαλύτερη από αυτή του GC_r → Βήμα 1.2 αλλιώς → Βήμα 0 1.2: Ενημέρωσε τα φορτία των 2 οχημάτων και τα GC → Βήμα 1
Βήμα 3: Αν κύκλος = 1 → Βήμα 3.1 αλλιώς → Βήμα 4 3.1: Μηδένισε τα προηγούμενα ζεύγη → Βήμα 0
Βήμα 4: <u>Τέλος</u>

Πίνακας.6 Ψευδοκώδικας φάσης adjustment

(*) $g \triangleq$ δότης , $r \triangleq$ λήπτης



Σχήμα.8 Διάγραμμα ροής φάσης adjustment

2.3 Αποτελέσματα - Συμπεράσματα

Είναι πλέον εμφανής η αποτελεσματικότητα της μεθόδου 3-PCC, τα κέντα GC των συστάδων είναι εμφανώς διαχωρισμένα. Οι αρχικές λύσεις που παράγει εν γένει για ένα περιορισμένο χρονικών παραθύρων πρόβλημα θα είναι μερικώς – εφικτές (ή ήμιεφικτή) .

Ορισμός: Ο όρος **ημιεφικτή λύση** (*semi feasible*) στο εξής θα χρησιμοποιείται δια να περιγράψει ένα δρομολόγιο που πληροί, τους περιορισμούς χωρητικότητας και χρόνου επιστροφής στην αποθήκη (σχέση 13) αλλά δεν θα πληροί τουλάχιστον ένα *time window*.

Κατά άλλους η ημιεφικτή αυτή λύση θα μπορούσε να χαρακτηριστεί και ως υποβελτιστη. Εκκινώντας από μια τέτοιας ποιότητας αρχική λύση, μια οποιαδήποτε μεθευρετική μέθοδος θα έχει μεγάλη πιθανότητα να συγκλίνει σχετικά εύκολα και γρήγορα εις το απόλυτον βέλτιστο. Αξίζει να τονιστεί ότι 3-PCC μέθοδος μπορεί να χρησιμοποιηθεί με οποιαδήποτε άλλη μεθευρετική μέθοδο, και σε προβλήματα Capacitated -VRP χαλαρώνοντας τον χρονικό περιορισμό. Ζητούμενο μετά από όλες αυτές τις διαδικασίες και φάσεις είναι να ικανοποιείται το μέτρο απόδοσης. Για τον αλγόριθμο 3-PCC ο οποίος φέρει την ευθύνη για μια καλή αρχική λύση (Initial solution), το μέτρο απόδοσης του αναφέρεται στο μέγεθος του στόλου (Vehicle Number ή N.V). Τα ερωτήματα που ανακύπτουν είναι τα εξής:

1. Ταυτίζεται το μέγεθος του παραχθέντος στόλου με το βέλτιστο μέγεθος (opt fleet), αν ναι με ποιά πιθανότητα ;
2. Αν όχι, πόσο απέχει από το βέλτιστο ;
3. Ποσό θα στοιχίσει σε χρόνο εκτέλεσης και ποιότητα λύσης αυτή η διαφορά ;

Ορισμός: **Απόδοση αρχικής λύσης** *initial efficiency – INEF*) μιας μεθόδου *K* εφεξής θα αναφέρεται ως:

$$\overline{INEF}_k = 1 - \frac{|\overline{opt\ fleet} - \overline{fleet}_K|}{\overline{opt\ fleet}} \rightarrow \{0,1\} \quad (17)$$

Ο λόγος της απολύτου διαφοράς του βέλτιστου στόλου ($\overline{opt\ fleet}$)* από στόλο της μεθόδου *K* (\overline{fleet}_K) προς το βέλτιστο στόλο.

* Η μέση τιμή αυτή θα πρέπει πάντα να αναφέρεται σε αντικειμενικό δείγμα προβλημάτων για να δύναται να γενικευτεί. Σε κάθε περίπτωση οι δυο αυτοί μέσοι αναφέρονται στο ίδιο δείγμα.

Παρατηρήσεις:

1. Αποτελεί αμερόληπτη εκτιμήτρια πιθανότητα (δυνατές τιμές από 0 έως 1).
2. Έχει νόημα για μέχρι το πολύ 2πλάσιο στόλο από το βέλτιστο.
3. Η πραγματική πιθανότητα δεν δύναται να είναι γνωστή.
4. Η πραγματική πιθανότητα δεν θα μπορεί ποτέ να φτάνει το 100% .

Είναι ανάγκη να υπάρχει ένας τέτοιος ορισμός. Συγκεκριμένα είναι η *εκτιμήτρια πιθανότητα*, θα πλησιάζει το μηδέν όσο ο στόλος πλησιάζει το διπλάσιο βέλτιστο, ενώ θα πλησιάζει τη μονάδα όσο ο στόλος πλησιάζει το βέλτιστο. Ακόμα μια ενδεχόμενη μεταβλητή χρόνου εκτέλεσης (run time) στο δείκτη απόδοσης, δεν θα είχε κανένα απολύτως νόημα καθώς εξορισμού οι μέθοδοι είναι ταχύτατες. Όπως έχει ήδη αναφερθεί αρχικές λύσεις κατασκευάζονται από άπληστους αλγορίθμους ή και τυχαιοποιημένα, που θεωρητικά με τη κατάλληλη στρατηγική θα μπορούσαν να φτάσουν σε απόδοση και το 0.99 . Η θέση αυτή θα μπορούσε να αποτελέσει αντικείμενο ερεύνης που όμως δεν απασχολεί τη παρούσα μελέτη. Εν προκειμένω όπως φαίνεται και απ' το πίνακα.7 :

Πίνακας Αποτελεσμάτων για αλγόριθμο 3-PCC											
a/a	Name	G.B N.V	3- PCC N.V	a/a	Name	G.B N.V	3- PCC N.V	a/a	Name	G.B N.V	3- PCC N.V
1	C101	10	10	19	R102	17	10	37	R208	2	3
2	C102	10	10	20	R103	13	10	38	R209	3	3
3	C103	10	10	21	R104	9	10	39	R210	3	3
4	C104	10	10	22	R105	14	10	40	R211	2	3
5	C105	10	10	23	R106	12	10	41	RC101	14	10
6	C106	10	10	24	R107	10	10	42	RC102	12	10
7	C107	10	10	25	R108	9	10	43	RC103	11	10
8	C108	10	10	26	R109	11	10	44	RC104	10	10
9	C109	10	10	27	R110	10	10	45	RC105	13	10
10	C201	3	3	28	R111	10	10	46	RC106	11	10
11	C202	3	3	29	R112	9	10	47	RC107	11	10
12	C203	3	3	30	R201	4	3	48	RC108	10	10
13	C204	3	3	31	R202	3	3	49	RC201	4	3
14	C205	3	3	32	R203	3	3	50	RC202	3	3
15	C206	3	3	33	R204	2	3	51	RC203	3	3
16	C207	3	3	34	R205	3	3	52	RC204	3	3
17	C208	3	3	35	R206	3	3	53	RC205	4	3
18	R101	19	10	36	R207	2	3	54	RC206	3	3
	AVERAGE	7.39	6.89		AVERAGE	8.00	7.28	55	RC107	3	3
opt fleet	7.50							56	RC108	3	3
fleet 3PCC	6.87								ERAGE	7.11	6.44
INEF	0.92										

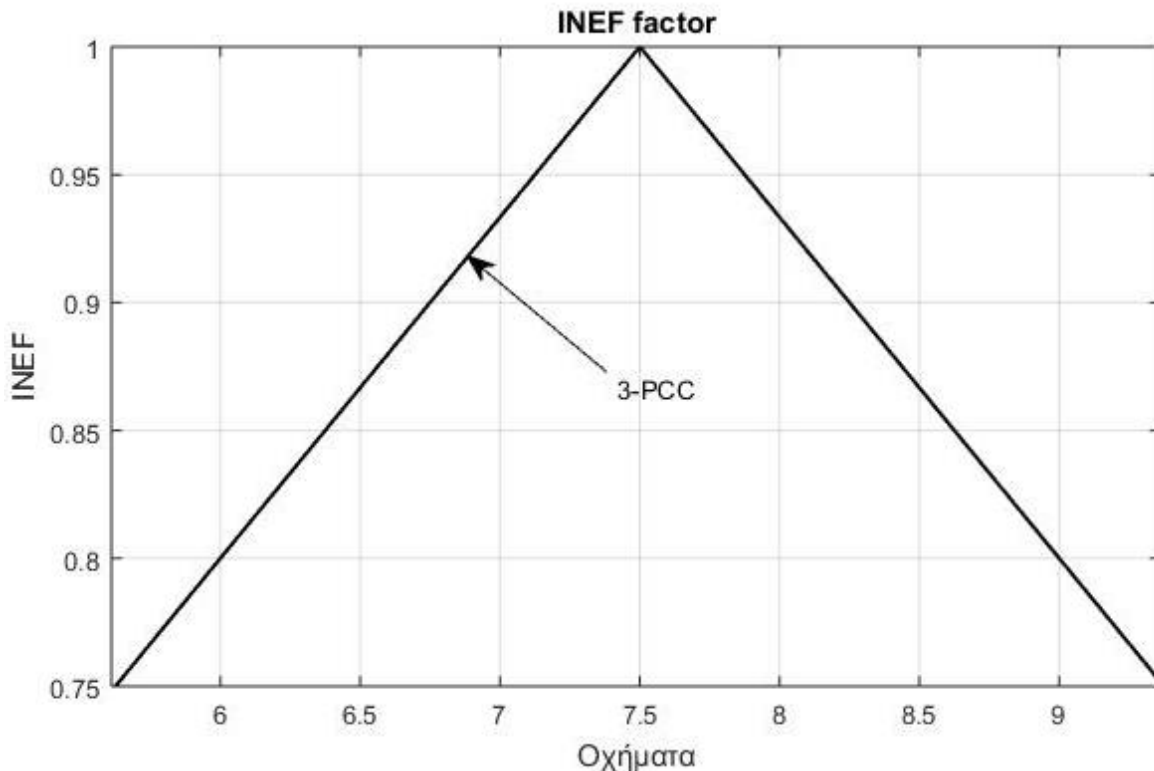
Πίνακας.7 Σύγκριση αποτελεσμάτων 3-PCC βιβλιογραφίας

$$\overline{opt\ fleet} = \frac{\Sigma(7.39+8+7.11)}{3} = 7.5 \text{ οχήματα}$$

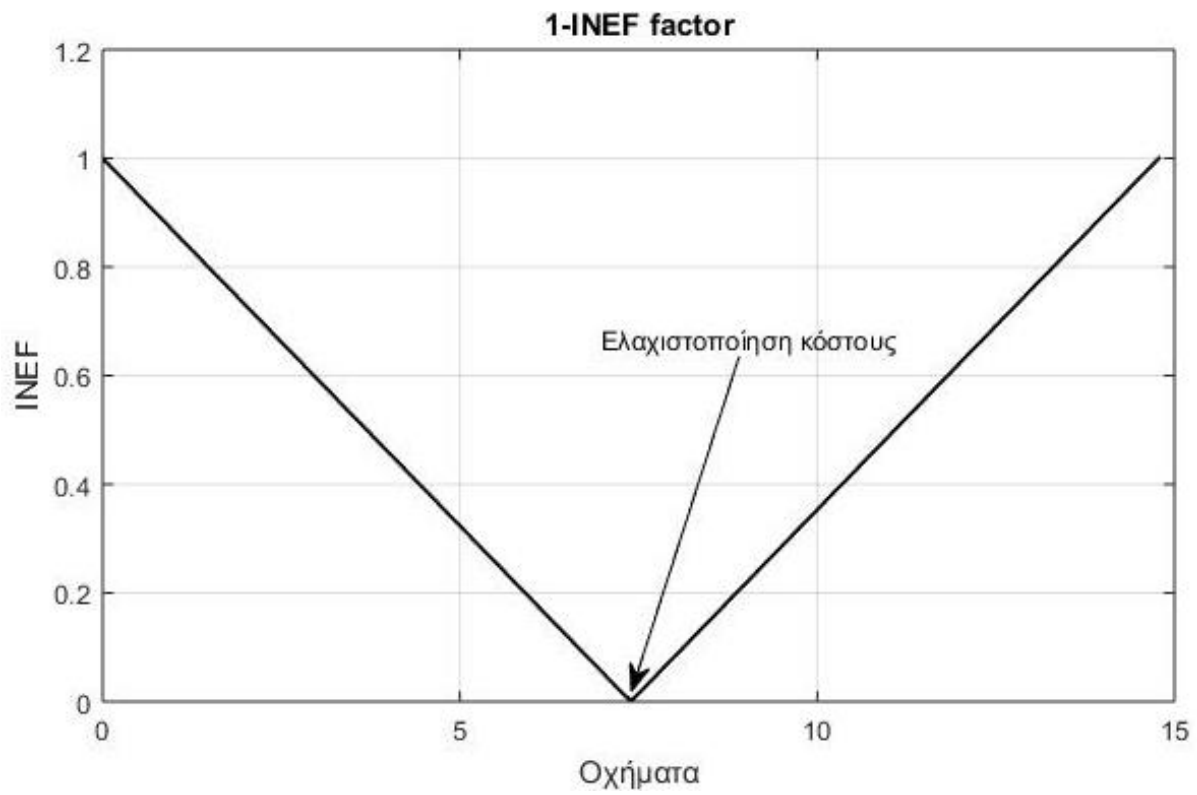
Αντίστοιχα,

$$\overline{fleet}_{3-PCC} = \frac{\Sigma(6.89 + 7.28 + 6.44)}{3} = 6.87 \text{ οχήματα}$$

Η διαφορά των δυο στόλων είναι στα 0,63 οχήματα ή **INEF_{3-PCC} = 0.92**, αυτό πρακτικά σημαίνει ότι ο μέσος στόλος που προκύπτει(μεθόδου 3-PCC) εκ του ανωτέρου δείγματος συμπίπτει σε ποσοστό 91% με το βέλτιστο στόλο. Διαφορετικά, αποκλίνει συμπληρωματικά με ποσοστό 9%. Πιο συγκεκριμένα σε 34 περατώσεις (60% του δείγματος) βρίσκει ακριβώς το βέλτιστο στόλο. Ένα περαιτέρω συμπέρασμα λιγότερο ακριβές –για την ώρα- είναι αναλογική και η μείωση του χρόνου εκτέλεσης (CPU time). Περισσότερα στοιχεία θα δοθούν στα επόμενα κεφάλαια. Παρακάτω φαίνεται η γραμμικότητα του βαθμού απόδοσης και πώς αυτός κορυφώνεται για το βέλτιστο στόλο στα 7,5 οχήματα. Οποιαδήποτε απόκλιση επιφέρει και μείωση. Η σημαντικότητα το προτεινόμενου βαθμού απόδοσης, φαίνεται άμεσα στο δυικό διάγραμμα (Σχήμα.6) που αντιπροσωπεύει μια κανονικοποιημένη συνάρτηση κόστους, για την επιχείρηση διανομής.



Σχήμα.9 Βαθμός απόδοσης INEF συναρτήσει στόλου για δείγμα Solomon n100



Σχήμα.6 Δυσικότητα συναρτήσει στόλου

Κεφάλαιο 3

Βελτιστοποιήσις δρομολογίου

3.1 Εισαγωγικά

Για τις ανάγκες βελτιστοποίησης τις όλης δρομολόγησης προτείνεται η χρήση ενός αλγορίθμου, που θα επιλύει ανεξάρτητα και χρονικώς διακριτά το πρόβλημα της διαδρομής του οχήματος. Γνωστό στη βιβλιογραφία ως πρόβλημα πλανόδιου πωλητού (traveling salesman problem). Εις ότι αφορά το τομέα της συνδυαστικής βελτιστοποίησης είναι μια απ' τις δυσκολότερες προκλήσεις ανήκουσα εις την κατηγορία των NP – πλήρεις (non polynomial complete) [8]. Δεν υπάρχει λύση σε πολυωνυμικό χρόνο και δεν μπορεί να επιβεβαιωθεί ότι δεν υπάρχει αυτή η λύση. Αντικείμενο της παρούσης μελέτης είναι η μεθευρετική μεθοδολογία για το VRPTW .

Η πλέον πολυχρησιμοποιημένη μέθοδος μεθευρετικής αναζήτησης είναι η *απαγορευμένη προσαρμοστικής μνήμης ή γνωστή ως περιορισμένη αναζήτηση (Tabu Search)* [5]. Όλα δομούνται στην λογική της τοπικής αναζήτησης (local search). Είναι μια πολύ γενική τεχνική που εξερευνά εις το χώρο των δυνατών λύσεων με σειριακό τρόπο, μετακινούμενη σένα βήμα από την τρέχουσα λύση εις μιαν «γειτονική» της χρήσει κάποιου μετασχηματισμού (relocate, 2-opt κτλ). Διαφορετικά είναι μια λογική δοκιμής και σφάλματος, η αρχαιότερα μέθοδος αναζήτησης. Μειονεκτεί εις ότι αφορά τη δυσκολία εφαρμογής της, ενώ πλεονεκτεί ως προς τη γνώση για την ποιότητα της παραχθείσας λύσης. Εν προκειμένω οι δυο ανωτέρω θέσεις δεν ισχύουν. Αφενός διότι το πρόβλημα είναι ιδιαιτέρως δύσκολο και χρήζει περίπλοκων στρατηγικών μεθευρετικής αναζήτησης, αφετέρου διότι υπάρχει η δυνατότητα επιβεβαίωσης της ποιότητας των λύσεων από τη βιβλιογραφία. Σε αυτό το σημείο ξεκαθαρίζεται ότι όροι «γειτονική λύση», «γειτονιά αναζήτησης», «περιοχή αναζήτησης», «τοπικό ελάχιστο» ως εργαλεία εικονογράφησης της αντικειμενικής συνάρτησης είναι αφηρημένες έννοιες και σε καμία περίπτωση δεν επιδέχονται μαθηματικής απόδειξης ή προτυποποίησης. Η αντικειμενική συνάρτηση σε προβλήματα δρομολόγησης οχημάτων φέρει πάρα πολλές μεταβλητές απόφασης, και είναι αδύνατον να παρασταθεί γραφικώς. Χρήση αυτής της αφηρημένης εικονογράφησης υπάρχει για να εμπνέει της στρατηγικές αναζήτησης και ως εύκολο μέσο

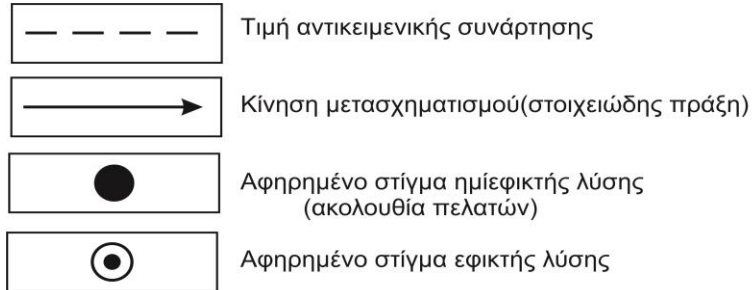
μετάδοτικότητας. Εδώ υπάρχει αντικειμενική συνάρτηση, το κόστος δρομολογίου ή διανυθείσα απόσταση οχήματος. Υποκείμενης χρονικών περιορισμών, αφενός χρόνου επιστροφής στην αποθήκη (route time), αφετέρου ατομικού παραθύρου του πελάτη (time window).

Μια δεύτερη μέθοδος μεταερευτικής αναζήτησης είναι οι γενετικοί αλγόριθμοι [11] (genetic algorithms), είναι η ιδέα της «μίμησης» της γενετικής εξέλιξης στη φύση προς αναζήτηση καλύτερων λύσεων. Βασίζονται σε μια μίμηση της βιολογικής διαδικασίας στην οποία νέοι και καλύτεροι πληθυσμοί μεταξύ διαφορετικών ειδών αναπτύσσονται κατά τη διάρκεια της εξέλιξης. Έτσι εν αντιθέσει των περισσότερων ευρετικών αλγορίθμων, οι γενετικοί κατά τη διάρκεια της αναζήτησης μιας καλύτερης λύσης χρησιμοποιούν πληροφορίες από ένα πληθυσμό λύσεων ονόματι άτομα (individuals). Συνήθως χρησιμοποιούνται στοχαστικές στρατηγικές και πολλές παραλλαγές έχουν προταθεί. Η βασική ιδέα είναι το ταίριασμα δυο λύσεων με σκοπό τη δημιουργία μιας νέας, καλύτερης, χρήσει συνήθως δυο τελεστών, το δυαδικό (crossover - mating) και το μοναδιαίο (μετάλλαξη – mutation). Η διασταύρωση κάνει χρήση δυο ατόμων που ονομάζονται γονείς (parents) οι όποιοι παράγουν τους απογόνους ή τέκνα (offspring). Έκαστο άτομο του πληθυσμού είναι μια υποψήφια λύση στο πρόβλημα επιλογής υποσυνόλου χαρακτηριστικών. Αιτία του ότι οι γενετικοί αλγόριθμοι έχουν οικειοποιηθεί στρατηγικές γενετικής εξέλιξης είναι να υπάρχει ένας πληθυσμός λύσεων (population) από ξεχωριστά δηλαδή άτομα (individuals). Στη φύση αντίθετα κάθε βιολογικός οργανισμός φέρει κύτταρα που απαρτίζονται από ένα συγκεκριμένο αριθμό χρωμοσωμάτων, έτσι σένα γενετικό αλγόριθμο τα άτομα θα θεωρούνται και χρωμοσώματα. Κάποια από αυτά τα χρωμοσώματα θα ονομάζονται γονείς και θα επιλέγονται από τον τρέχων πληθυσμό προς αναπαραγωγή. Δυο μετασχηματισμοί λαμβάνουν χώρα, η διασταύρωση και η μετάλλαξις [παράβαλε 3.3]

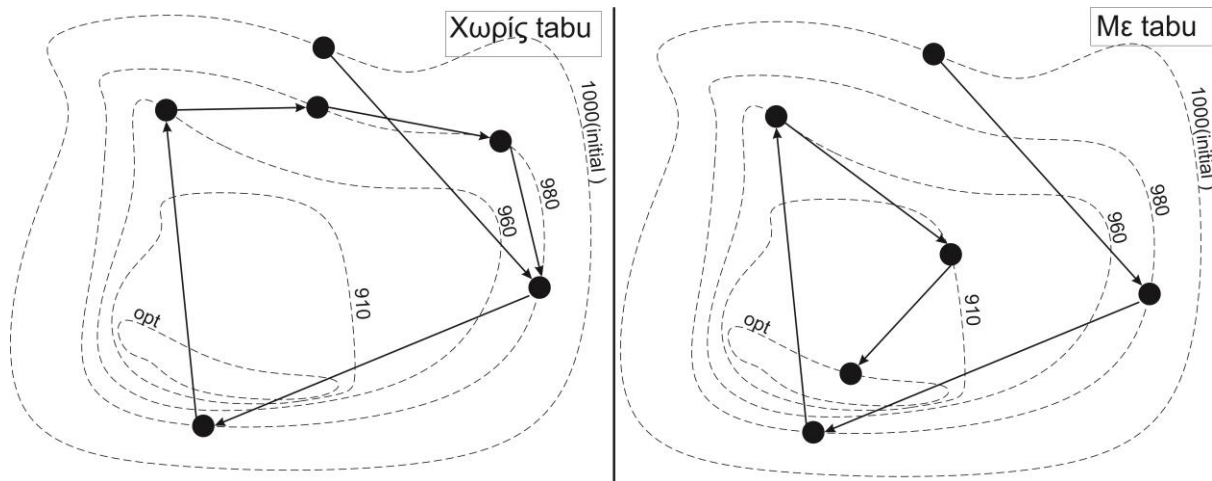
3.2 Απαγορευμένη , προσαρμοστικής μνήμης αναζήτηση

Αναλυτικότερα στον αλγόριθμο περιορισμένης αναζήτησης με προσαρμοστική μνήμη (Tabu Search with Adaptive Memory), η τοπική αναζήτηση γίνεται με δυο μετασχηματισμούς, μείζων τον 2-opt και ελάσσων το swap. Η μνήμη χωρίζεται σε βραχύχρονη (short – term) και μακρόχρονη (long – term). Χρήση της πρώτης γίνεται στο κομμάτι της τοπικής αναζήτησης, της δεύτερης στο κομμάτι της αλλαγής γειτονιών (neighbourhoods) αντίστοιχα. Οι απαγορευμένες κινήσεις προκύπτουν από τη short-term μνήμη. Το σκεπτικό πίσω από μια απαγόρευση είναι η αποφυγή επιστροφής στο ίδιο σημείο (κόστος αντικειμενικής συνάρτησης – [σχήμα 1]). Εφεξής οι αφηρημένοι όροι της τοπικότητας θα δεικνύονται με τοπογραφημένα 2 - διάστατα σχήματα. Κάθε σημείο της γεωγραφικής περιοχής θεωρείται ένας συνδυασμός της ημι - εφικτής λύσης.

Διακριτές υψομετρικές καμπύλες είναι το κόστος της λύσης (βουνά, κάμποι, χαράδρες), ακολουθούνται οι εξής συμβολισμοί:

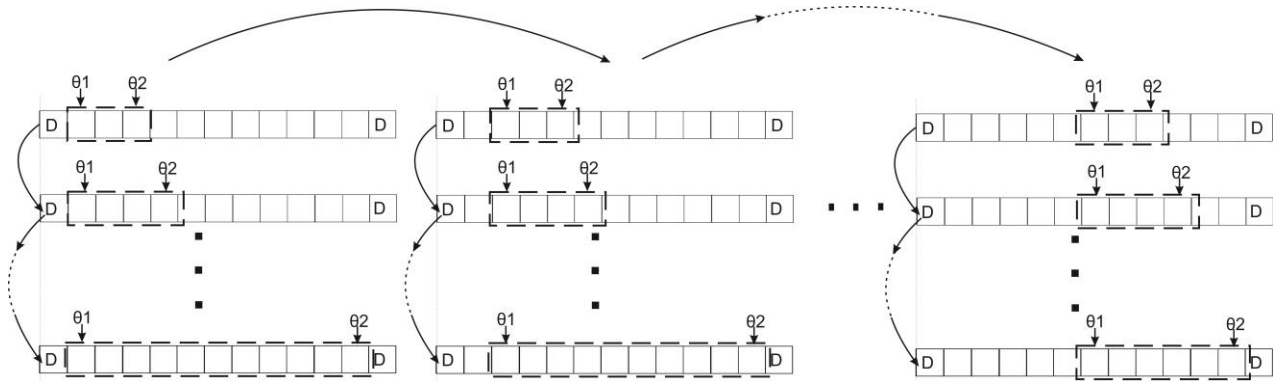


Σχήμα.11 Συμβολισμοί τοπογραφίας



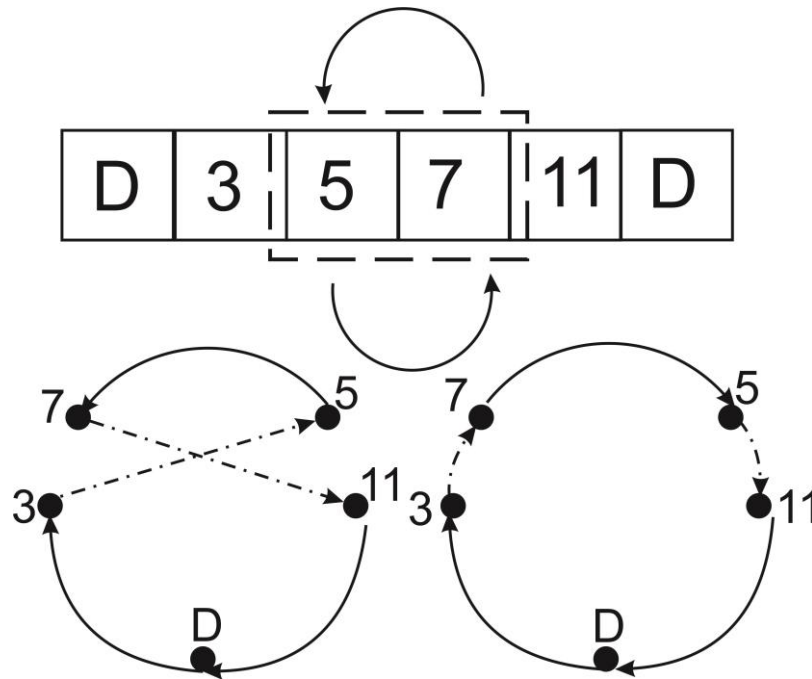
Σχήμα.12 Τοπική αναζήτηση με ή χωρίς κίνηση tabu

Στο σχήμα 1 φαίνεται ξεκάθαρα πως επιδράει μια απαγόρευση κίνησης. Έτσι αποφεύγεται μια αέναη ταλάντωση μεταξύ των τιμών 980 και 960 της αντικειμενικής συνάρτησης. Η στρατηγική που ακολουθείται δια τις στοιχειώδεις πράξεις είναι γραμμική (μεταξύ $\theta 1$ και $\theta 2$). Αυτό σημαίνει ότι εν γένει λαμβάνει χώρα από τα αριστερά προς τα δεξιά της λύσης. Είναι απλώς μια σύμβαση που κάλλιστα θα μπορούσε να ήταν διαφορετική, χωρίς αυτό να σημαίνει ότι αν ήταν διαφορετική θα άλλαζε κάτι. Είναι σαν κάποιος να προσπαθεί να εξερευνήσει ένα δάσος και να επιλέγει να προχωράει σε ευθεία ή ζιγκ – Ζακ.



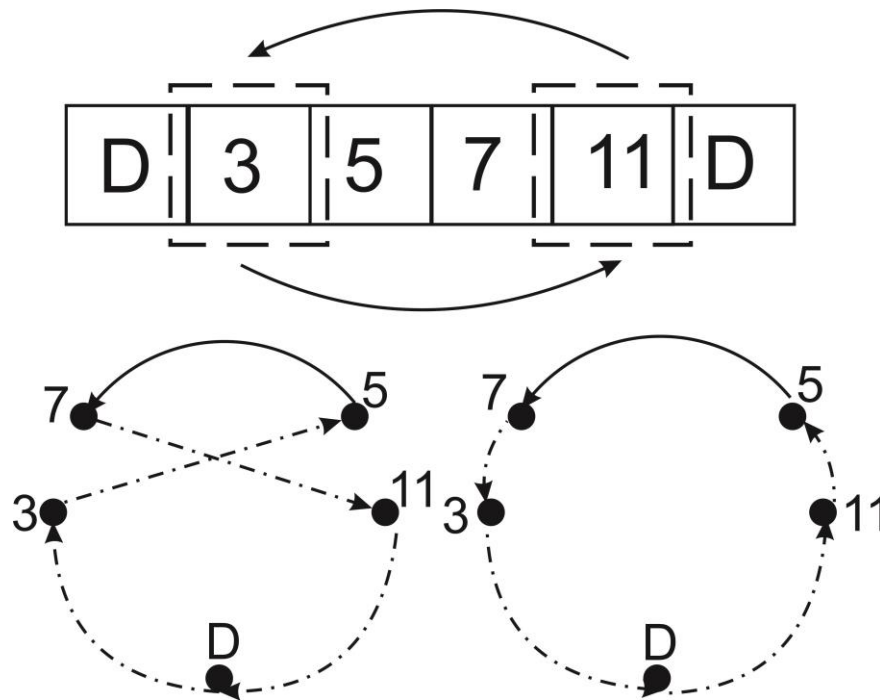
Σχήμα.13 Γραμμική στρατηγική μετασχηματισμού 2-opt

Η πρώτη στοιχειώδης πράξη που επιλέγονται είναι η 2-opt, κατά την οποία δυο τόξα εξέρχονται και δυο άλλα εισέρχονται στη λύση. Χαρακτηριστικά της είναι οι δυο θέσεις, θ_1 και θ_2 . Ένα παράδειγμα φαίνεται παρακάτω:



Σχήμα.14 Παράδειγμα 2-opt κίνησης

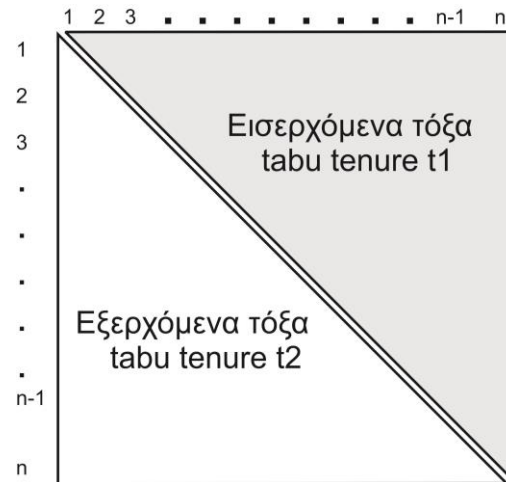
Στο σχήμα 3 φαίνεται μια διαδρομή τεσσάρων πελατών, όπου τα τόξα $\widehat{3-5}$, $\widehat{7-11}$ εξέρχονται, ενώ δυο άλλα τα $\widehat{3-7}$, $\widehat{5-11}$ εισέρχονται. Ο ελάχιστων μετασχηματισμός swap επιφέρει μεγαλύτερες αλλαγές στη διαδρομή, τέσσερα τόξα εξέρχονται και άλλα τέσσερα εισέρχονται.



Σχήμα.15 Παράδειγμα κίνησης swap

Για την ίδια διαδρομή με παραπάνω εξέρχονται τα τόξα $\widehat{D-3}$, $\widehat{3-5}$, $\widehat{7-11}$, $\widehat{11-D}$. Ταυτόχρονα εισέρχονται τα $\widehat{D-11}$, $\widehat{11-5}$, $\widehat{7-3}$, $\widehat{3-D}$

Όπως έχει ήδη αναφερθεί το πρόβλημα TSP είναι απ'τα δημοφιλέστερα της συνδυαστικής βελτιστοποίησης, κατά τούτο έχουν δημοσιευτεί εκατοντάδες διεθνή άρθρα προτείνοντα λύσεις. Έχει φανεί ότι στη πράξη ο 2-opt αποδίδει καλύτερα σχέσει άλλων και γ' αυτό το λόγο επιλέγεται ως μείζων μετασχηματισμός. Ο λόγος που χρησιμοποιείται ως ελάσσων (ή επικουρικός) ο swap είναι να προσδοθεί μεγαλύτερη «διαχυτικότητα» στον αλγόριθμο. Η επιλογή ανάμεσα στους δυο κάθε φορά που καλείται να πραγματοποιηθεί μια πράξη είναι τυχαία, με 80% 2-opt και 20% swap, χρήσει γεννήτριας τυχαίων αριθμών ομοιόμορφης κατανομής $U(0,1)$. Παρακάτω θα υπάρξουν περισσότερες τυχαίες μεταβλητές καθώς ο αλγόριθμος επιλέχτηκε να λειτουργεί στοχαστικά. Φυσικά η στοχαστική επιρροή είναι μικρή έναντι του ντετερμινισμού. Έχει ήδη αναφερθεί ότι οι απαγορεύσεις στις κινήσεις (αν υπάρξουν) αποφασίζονται βάσει της βραχύχρονης μνήμης. Έτσι κάθε φορά που μια πράξη 2-opt γίνεται αποθηκεύονται τα τέσσερα τόξα που την εμπλέκουν. Τα εισερχόμενα τόξα απ'τη στιγμή που θα καταχωρηθούν στη μνήμη, παραμένουν για τουλάχιστον διάρκεια απαγόρευσης tabu tenure - t1 πράξεις στη τρέχουσα λύση. Αντίστοιχα για τα εξερχόμενα τόξα με tabu tenure - t2. Στο σχήμα 11 δεικνύεται η δομή δεδομένων της short - term μνήμης.



Σχήμα.16 Δομή δεδομένων short – term μνήμης

Εις ότι αφορά τις χρονικές διάρκειες, κατά καιρούς έχουν προταθεί πολλά νούμερα, εν προκειμένω προτείνονται και πάλι τυχαίες μεταβλητές ομοιόμορφα κατανεμημένες και εις αναλογίαν του πλήθους των πελατών, έτσι :

$$\text{Εισερχόμενα τόξα tenure 1} = U(\sqrt{n}, 2\sqrt{n}) \quad (18)$$

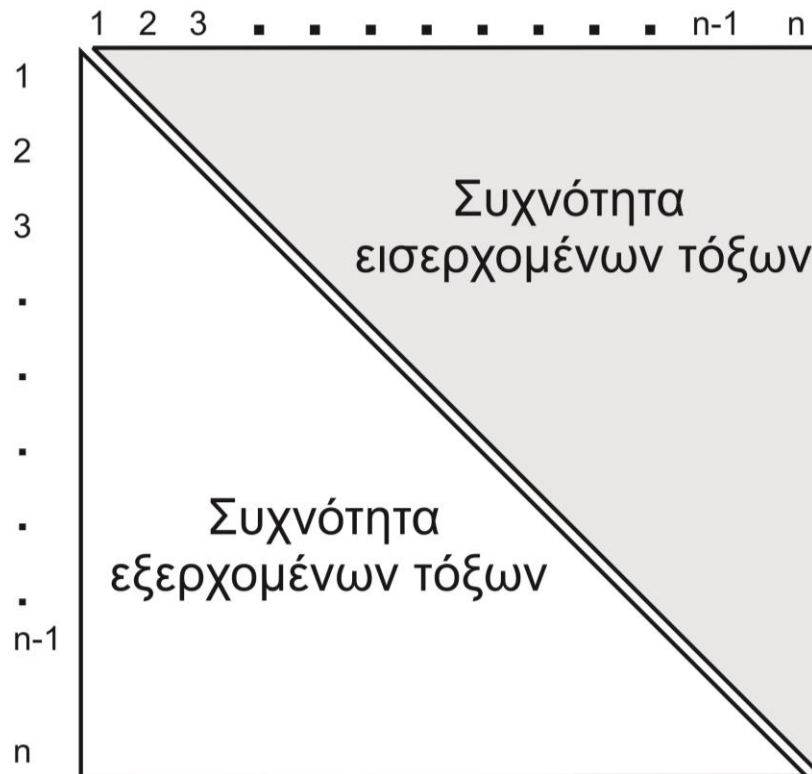
$$\text{Εξερχόμενα τόξα tenure 2} = U(2\sqrt{n}, 5\sqrt{n}) \quad (19)$$

Για το ανωτέρω παράδειγμα $n=4$, έχουμε $t1 = U(2, 4)$ ενώ $t2 = U(4, 10)$. Προκύπτει μεγαλύτερη πιθανότητα απαγόρευσης στα εξερχόμενα τόξα έναντι των εισερχομένων, κι έτσι θα μπορούσε κάποιος να πει ότι «προάγεται» η εισαγωγή νέων τόξων. Θεωρώντας τυχαία τώρα $t1 = 3$, $t2 = 7$ ο πίνακας για μια επανάληψη της αναζήτησης συμπληρώνεται ως :

	1	2	3	4	5	6	7	8	9	10	11
1	-	0	0	0	0	0	0	0	0	0	0
2	0	-	0	0	0	0	0	0	0	0	0
3	0	0	-	0	0	0	3	0	0	0	0
4	0	0	0	-	0	0	0	0	0	0	0
5	0	0	7	0	-	0	0	0	0	0	3
6	0	0	0	0	0	-	0	0	0	0	0
7	0	0	0	0	0	0	-	0	0	0	0
8	0	0	0	0	0	0	0	-	0	0	0
9	0	0	0	0	0	0	0	0	-	0	0
10	0	0	0	0	0	0	0	0	0	-	0
11	0	0	0	0	0	0	7	0	0	0	-

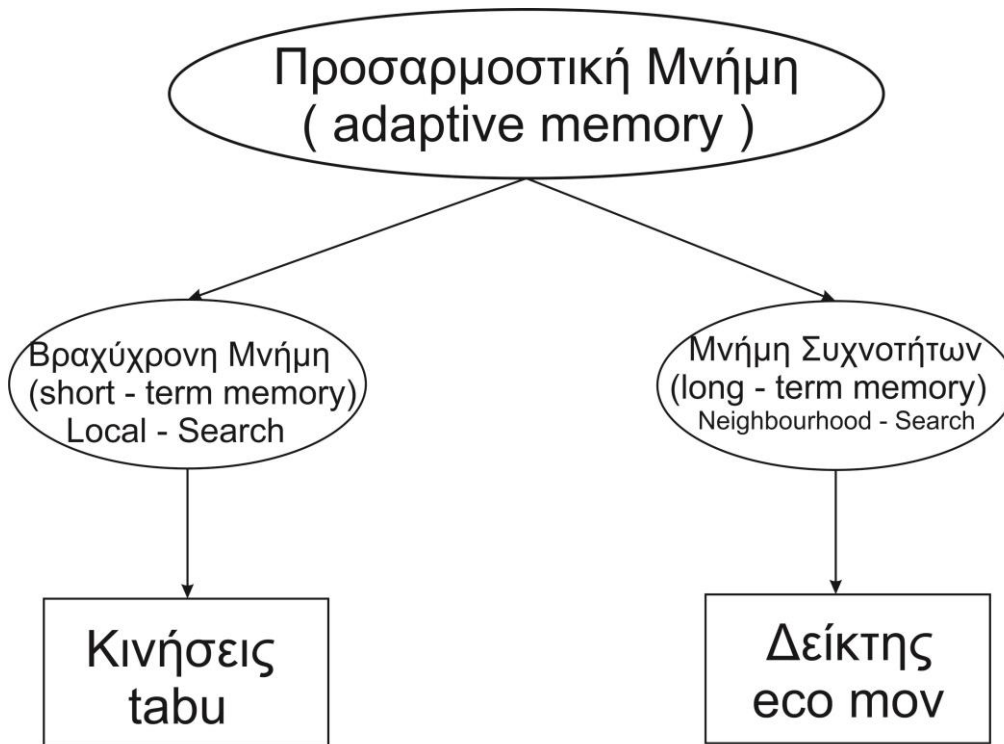
Πίνακας.8 short – term μνήμη παραδείγματος για $t1 = 3$, $t2 = 7$

Σε κάθε μετασχηματισμό της λύσης οι διάρκειες μειώνονται κατά ένα, εκτός και αν τα συγκεκριμένα τόξα ξαναεμφανιστούν στη λύση οπότε και ανανεώνεται η αντίστοιχη διάρκεια απαγόρευσης. Εις ότι αφορά την μακρόχρονη μνήμη (long – term) ή μνήμη συχνοτήτων, αποθηκεύει το «ιστορικό» της αναζήτησης, τις συχνότητες δηλαδή εμφάνισης του κάθε τόξου. Η δομή δεδομένων της μνήμης φαίνεται στο σχήμα 9 .



Σχήμα 17 Δομή δεδομένων μνήμης συχνοτήτων

Όπως έχει ήδη λεχθεί η μνήμη συχνοτήτων είναι το εργαλείο μετά – τοπικής αναζήτησης. Είναι η στρατηγική εύρεσης νέων περιοχών αναζήτησης και καθώς τοπικές αναζητήσεις λαμβάνουν χώρα η μνήμη εξελίσσεται μαζί μ' αυτές, αυτός είναι ο λόγος που χαρακτηρίζεται ως προσαρμοστική. Το περιεχόμενο αυτής της μνήμης χρησιμοποιείται δια τον υπολογισμό ενός δείκτη «οικονομίας της κίνησης» (*economy of a movement – eco mon*). Συνοπτικά η προσαρμοστική μνήμη φαίνεται στο σχήμα 12.



Σχήμα.12 Προσαρμοστική μνήμη

Ο δείκτης $eco\ of\ a\ movement$ υπολογίζεται με δυο τρόπους ανάλογα με το σκοπό που επιθυμείται να γίνει η αναζήτηση σε διαφορετική γειτονιά. Ο πρώτος αφορά την εντατικοποίηση της αναζήτησης, επιλέγεται δηλαδή μια λύση που προάγει τα συχνώς εισερχόμενα τόξα, ενώ αντίθετα αποθαρρύνει τα σπάνια εξαγόμενα τόξα. Συγκεκριμένα:

Εντατικοποίηση: $eco\ mon = \text{μήκος εξ.τόξου } 1 \times (1 - freq_{11}) \times \overline{\text{τόξο}}$
 $+ \text{μήκος εξ.τόξου } 2 \times (1 - freq_{21}) \times \overline{\text{τόξο}}$
 $- \text{μήκος εισ.τόξου } 1 \times freq_{12} \times \overline{\text{τόξο}}$
 $- \text{μήκος εισ.τόξου } 2 \times freq_{22} \times \overline{\text{τόξο}}$ (20)

Ο δεύτερος τη διάχυση, επιλέγονται λύσεις με συχνός εξερχόμενα τόξα, ενώ αποθαρρύνουν τα σπάνια εξερχόμενα τόξα:

Διάχυση: $eco\ mon = \text{μήκος εξ.τόξου } 1 \times freq_{11} \times \overline{\text{τόξο}}$
 $+ \text{μήκος εξ.τόξου } 2 \times freq_{21} \times \overline{\text{τόξο}}$
 $- \text{μήκος εισ.τόξου } 1 \times (1 - freq_{12}) \times \overline{\text{τόξο}}$
 $- \text{μήκος εισ.τόξου } 2 \times (1 - freq_{22}) \times \overline{\text{τόξο}}$ (21)

Το μέσο μήκος τόξου($\overline{\text{τόξο}}$) συνυπολογίζεται για να υπάρχει η σωστή τάξη μέγεθος, εκφράζει τη μέση απόσταση ενός οχήματος μεταξύ δυο πελατών. Ο υπολογισμός του γίνεται βάσει της δρομολόγησης του αλγορίθμου 3-PCC, συγκεκριμένα :

$$\overline{\text{τόξο}} = \frac{\text{Κόστος δρομολόγησης}_{3-PCC}}{\text{πλήθος πελατών} \times \text{πλήθος οχημάτων}} \quad (22)$$

Η επιλογή ανάμεσα σε διάχυση ή εντατικοποίηση γίνεται τυχαία, με έμφαση στην εντατικοποίηση, έτσι το 60% των γειτονιών επιλέγονται με δείκτη διάχυσης ενώ το υπόλοιπο 40% με δείκτη εντατικοποίησης.

Τα ανωτέρω αποτελούν τα δομικά εργαλεία του αλγορίθμου, όμως είναι ανάγκη να εφαρμοστεί και μια καλή στρατηγική. Αναλυτικότερα, ο προτεινόμενος αλγόριθμος θα επαναλαμβάνει τις στοιχειώδεις πράξεις του (2-opt, swap) όπως έχει ήδη αναφερθεί και δειχθεί στο σχήμα 8. Ο αριθμός επαναλήψεων είναι το άλφα και το ωμέγα, εις ότι αφορά τη τιμή της αντικειμενικής συνάρτησης λαμβάνονται υπόψη τα κάτωθι σενάρια :

- έχει κολλήσει σε μια τιμή.
- κάνει ταλάντωση μεταξύ δυο ακραίων τιμών.
- έχει ξεπεράσει σε πλήθος επαναλήψεων ένα άνω όριο.

Τα δυο πρώτα είναι δυναμικά σενάρια, εξαρτώνται καθαρά από τη φύση/ δομή του δρομολογίου, εκ των προτέρων είναι άγνωστο αν θα λάβουν χώρα ή όχι, παρ όλα αυτά θα προβλέπουν. Το τρίτο είναι ένα στατικό σενάριο περιπτώσει που τα άλλα δυο δεν συμβούν. Ο αλγόριθμος θα πρέπει να προκύψει όσο αποδοτικότερος γίνεται και αυτό εξαρτάται από τον τρόπο επιλογής των κριτηρίων (παραμέτρων) τερματισμού της αναζήτησης. Ως κριτήριο επιλογής ενός συνόλου παραμέτρων Α τερματισμού του αλγορίθμου προτείνεται ο δείκτης :

Ορισμός: **Δείκτης καθόδου (descent index)**, εκφράζει την ανά μονάδα χρόνου μείωση (εν γένει βελτίωση) της αντικειμενικής συνάρτησης, μιας μεθόδου υποκείμενης ενός συνόλου παραμέτρων Α .

$$DESIN_A = \frac{\text{initial cost} - \text{best cost}}{\text{run time}} \quad \left\{ \frac{\text{μον. κόστους}}{\text{μον. χρόνου}} \right\} \quad (23)$$

Παρατηρήσεις:

1. Δεν αποτελεί εργαλείο γενίκευσης της απόδοσης ενός αλγορίθμου.
2. Είναι ένας μέσος όρος ρυθμού καθόδου της αντικειμενικής. Συνάρτησης (ελαχιστοποίηση).
3. Εν γένει είναι αντιστρόφως ανάλογος με τη ποιότητα της λύσης.

4. Αποτελεί μέσο συγκριτικής εφαρμογής διαφορετικών παραμέτρων επί του ίδιου αλγορίθμου.

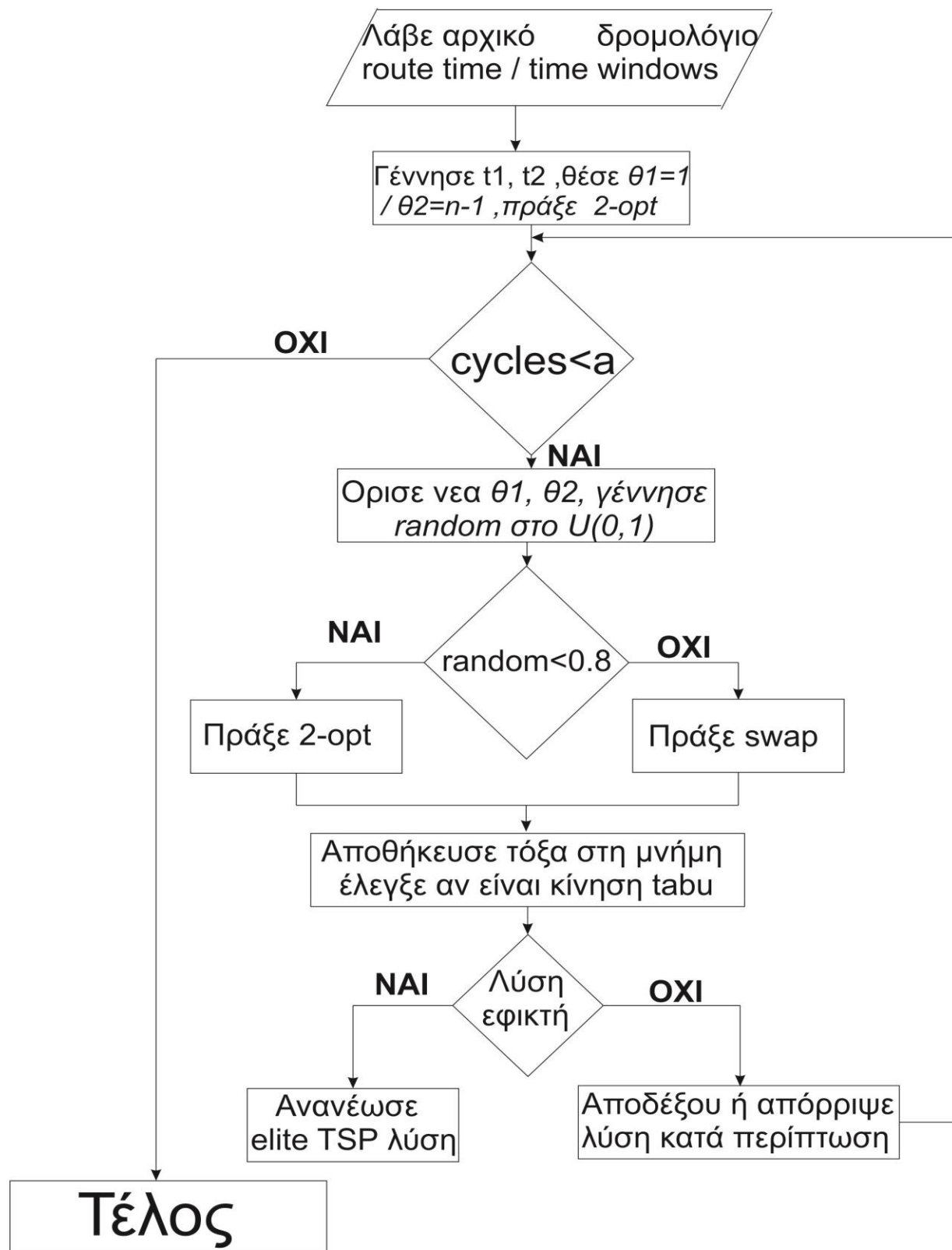
Σε καμία περίπτωση δεν αποτελεί αντικείμενο μαθηματικής συνεπαγωγής και δεν μπορεί να χρησιμοποιηθεί ως μέσο σύγκρισης άλλων αλγορίθμων. Αποκλειστική χρήση του ως κριτήριο σύγκρισης, διαφορετικών συνόλων παραμέτρων τερματισμού του ίδιου αλγορίθμου. Εναλλακτικά θα μπορούσε να λεχθεί ως μέσος ρυθμός καθόδου / βελτίωσης της συνάρτησης κόστους. Αν για παράδειγμα οι παράμετροι είναι τρείς, υπάρχουν δυο σύνολα παραμέτρων το $A_1 = \{ \alpha_1, \alpha_2, \alpha_3 \}$, $A_2 = \{ \alpha_4, \alpha_5, \alpha_6 \}$ τότε θα υπάρξουν $DESIN_{A_1}$, $DESIN_{A_2}$, επιλέγεται το σύνολο A που μεγιστοποιεί το δείκτη. Οι παράμετροι επιλέγονται πάντα αναλογικά του μεγέθους -n κόμβων του δρομολογίου. Τα ανωτέρω τρία σενάρια υπάρχουν τόσο σε επίπεδο τοπικής αλλά και μετά – τοπικής αναζήτησης, παρόλα αυτά θα ληφθούν υπόψη μόνον στη μετά – τοπική. Αναλυτικότερα για την τοπική αναζήτηση θα συμβαίνουν κάθε φορά linear cycles στοιχειώδης πράξεις επί του δρομολογίου. Οι κύκλοι αναφέρονται όπως δεικνύονται στο σχήμα 8. Κάθε φορά που ένας κύκλος πράξεων ολοκληρώνεται, πραγματοποιείται κίνηση 2-opt με θέσεις $\theta 1$, $\theta 2$ του πρώτου και τελευταίου κόμβου της τρέχουσας λύσης. Το σκεπτικό είναι να υπάρξει μεγαλύτερη διαχυτικότητα στην αναζήτηση. Κατά την ολοκλήρωση linear cycles επαναλήψεων τελειώνει η τοπική αναζήτηση και επαναπροσδιορίζονται οι διάρκειες της μνήμης $t1$, $t2$ όπως αυτές περιγράφονται στις σχέσεις 18, 19 αντίστοιχα. Σε κάθε στοιχειώδη πράξη 2-opt αποθηκεύονται τα τέσσερα εμπλεκόμενα τόξα, τρία σενάρια παρουσιάζονται:

1. Κίνηση tabu αλλά βελτιωτική (κριτήριο αισιοδοξίας / aspiration criteria).
2. Κίνηση όχι tabu αλλά μη – βελτιωτική (απόρριψη).
3. Κίνηση όχι tabu και βελτιωτική (αποδοχή).

Στο πρώτο εφαρμόζεται το *κριτήριο φιλοδοξίας*, αν δηλαδή η κίνηση είναι tabu αλλά παρόλα αυτά επιφέρει βελτίωση στη τρέχουσα λύση αίρεται η απαγόρευσις και γίνεται αποδεκτή. Στο δεύτερο απλά η κίνηση απορρίπτεται και στο τρίτο γίνεται αποδεκτή. Ακολουθούν τα βήματα :

Βήμα 0: Λάβε αρχικό δρομολόγιο / route time / time windows → Βήμα 2
Βήμα 2: Γέννησε $t1$, $t2$ από τις σχέσεις 7, 8 αντίστοιχα, πράξε 2-opt κίνηση μεταξύ $1^{ου}$ και $n^{στου}$ πελάτη → Βήμα 3
Βήμα 3: Αν cycles ολοκληρώθηκαν → Βήμα 8 αλλιώς όρισε νέες θέσεις $\theta 1$, $\theta 2$ → Βήμα 4
Βήμα 4: Γέννησε τυχαίο αριθμό random στο $U(0,1)$ → Βήμα 5
Βήμα 5: Αν random < 0.8 πράξε 2-opt , αλλιώς swap → Βήμα 6
Βήμα 6: Αποθήκευσε κίνηση στη μνήμη , έλεγξε αν η κίνηση είναι tabu ή όχι → Βήμα 7
Βήμα 7: Αν το δρομολόγιο είναι εφικτό → Βήμα 7.2 αλλιώς → Βήμα 7
7.2: Αν είναι καλύτερο εφικτό θέσε elite TSP ← τρέχουσα λύση → Βήμα 7
Βήμα 7: Αποδέξου ή απόρριψε κίνηση βάσει περιπτώσεων 1, 2, 3 → Βήμα 3
Βήμα 8: <u>Τέλος</u>

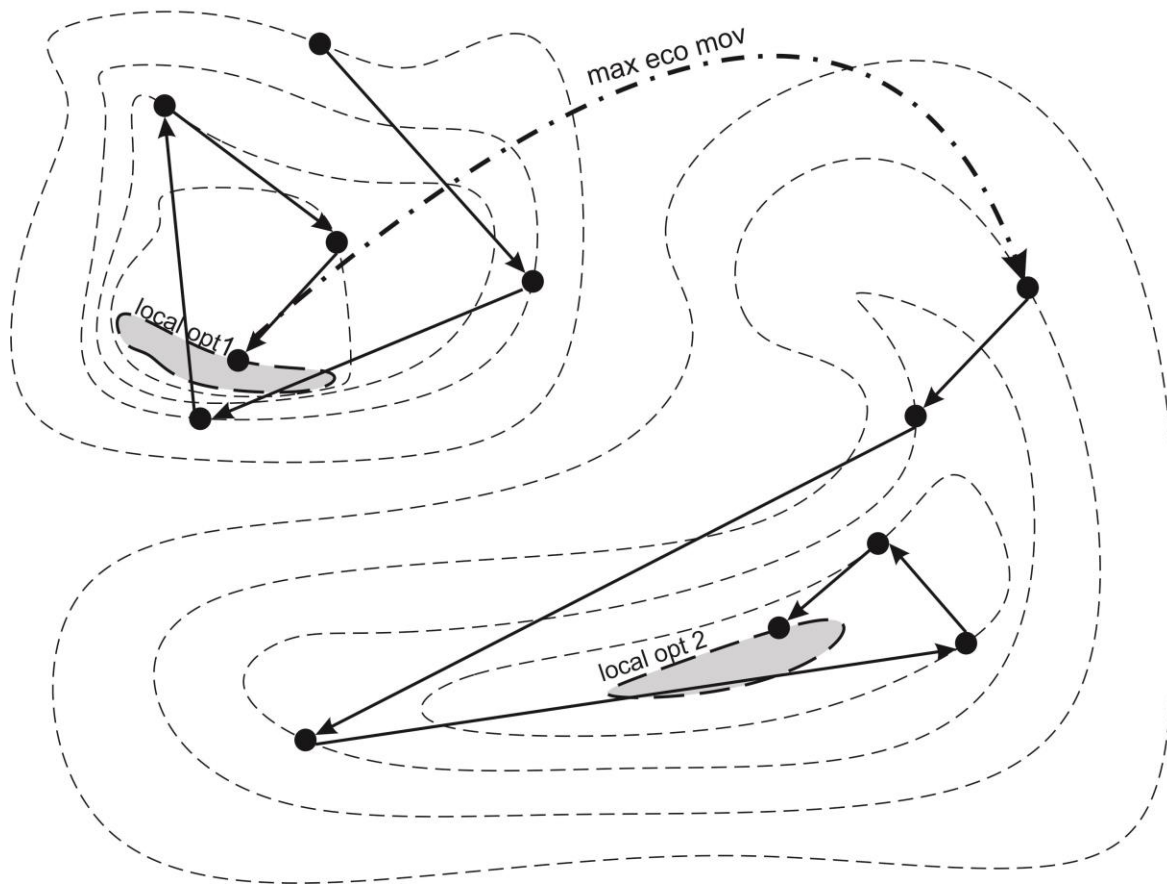
Πίνακας.9 Ψευδοκώδικας τοπικής αναζήτησης



Σχήμα.19 Διάγραμμα ροής τοπικής αναζήτησης

Εδώ επισημαίνεται ότι η αρχική σχεδίαση δεν λαμβάνει υπόψη τα time windows. Συνεπώς η στρατηγική αναζήτησης του αλγορίθμου αναφέρεται στη κάθοδο της αντικειμενικής συνάρτησης χωρίς να λαμβάνονται υπόψη χρονικά παράθυρα πελατών (χώρος ημιεφικτών λύσεων). Δεν υπάρχει κάποια στρατηγική που να προάγει πελάτες με συγκεκριμένα time windows ή να αποθαρρύνει άλλους. Παθητικά στη πορεία των επαναλήψεων ελέγχεται αν η λύση είναι εφικτή, και κάθε φορά ακολουθείται η λογική της καλύτερης εφικτής (best feasible). Το μειονέκτημα είναι ότι αν η εφικτή λύση δεν ευρίσκεται εις το δρόμο καθόδου της αντικειμενικής συνάρτησης, υπάρχει σοβαρή πιθανότητα να μην ανακαλυφθεί. Αυτό συνήθως συμβαίνει όταν η εφικτή λύση ευρίσκεται «μακριά» από τη βέλτιστη ημιεφικτή, στην επόμενη ενότητα θα δειχθεί πως αυτή η νοσηρότητα εν μέρει γιατρεύεται. Η μετά - τοπική αναζήτηση όπως έχει ήδη λεχθεί βασίζεται στο δείκτη *eco mon*, κατά τη τοπική αναζήτηση κάθε λύση που προκύπτει φέρει το δείκτη *eco mon*. Στο πέρας αυτής επιλέγεται ένα δρομολόγιο με το μεγαλύτερο *eco mon* και η διαδικασία επαναλαμβάνεται δυναμικά μέχρις ότου :

1. Επαναλαμβανόμενα ίδια τοπικά βέλτιστα ξεπεράσουν ένα συγκεκριμένο πλήθος.
2. Δυο κύκλοι συγκεκριμένου μεγέθους λάβουν χώρα.
3. Ένα άνω όριο ξεπεραστεί .

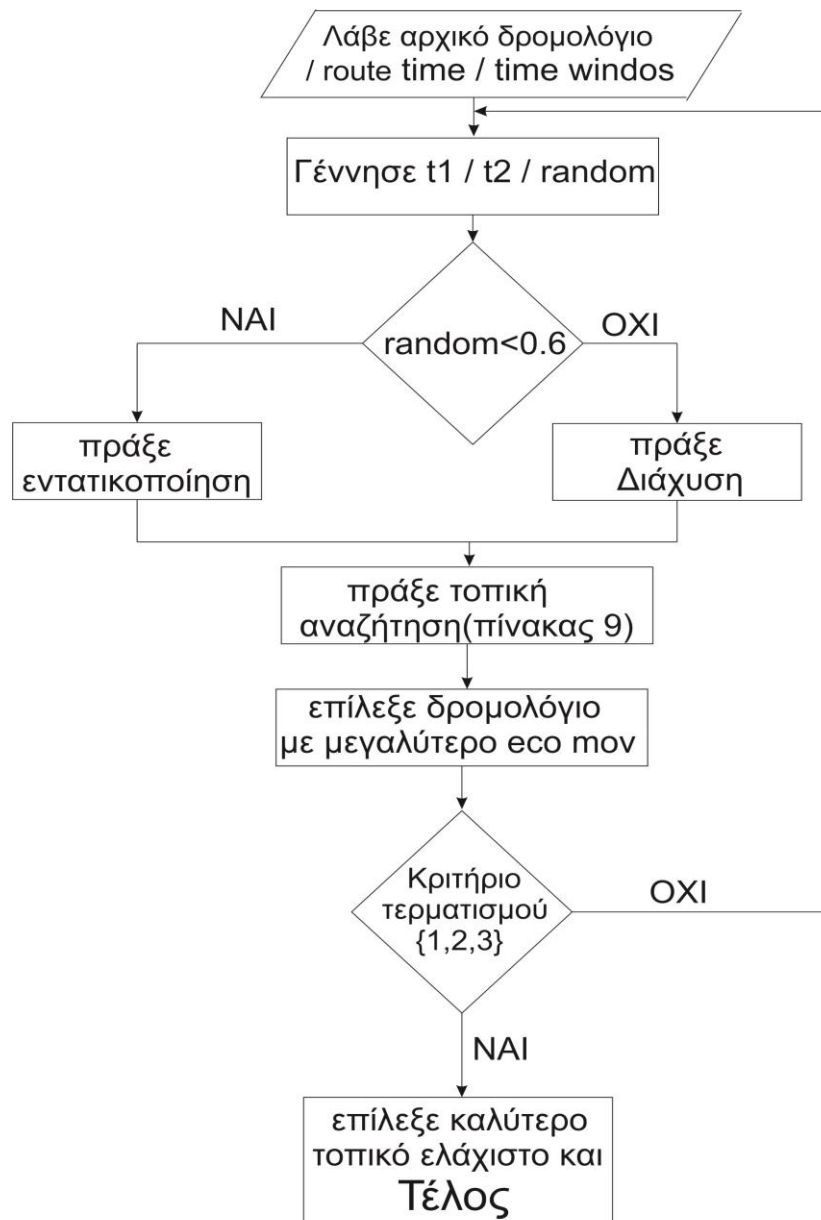


Σχήμα.20 Μετά - τοπική αναζήτηση

Ακολουθούν αθροιστικά τα βήματα :

Βήμα 0: Λάβε αρχικό δρομολόγιο / route time / time windows → Βήμα 1
Βήμα 1: Γέννησε t_1 , t_2 , random αν $\text{random} < 0.6$ $w \leftarrow$ εντατικοποίηση αλλιώς $w \leftarrow$ διάχυση → Βήμα 2
Βήμα 2: Πράξε τοπική αναζήτηση(πίνακας 9) → Βήμα 3
Βήμα 3: Επίλεξε δρομολόγιο με μεγαλύτερο eco μον → Βήμα 4
Βήμα 4: Αν κάποιο κριτήριο τερματισμού πληρείται → Βήμα 5 αλλιώς → Βήμα 1
Βήμα 5: Τέλος

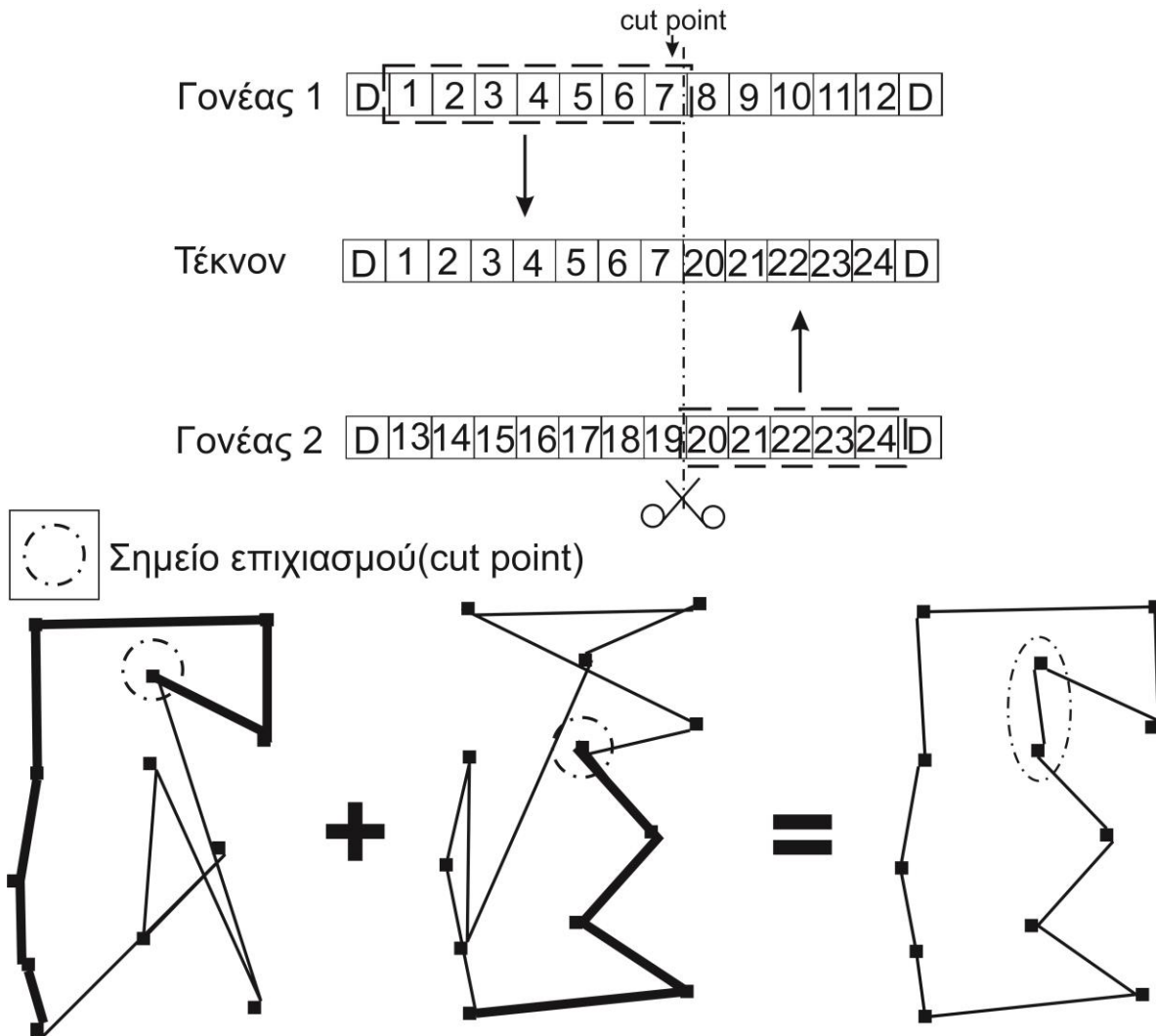
Πίνακας.10 Ψευδοκώδικας μετά - τοπικής αναζήτησης



Σχήμα.21 Διάγραμμα ροής Μέτα - τοπικής αναζήτησης

3.3 Γενετικά άπληστη τοπική αναζήτηση

Τελευταίο στάδιο είναι μια ταχύτατη διαδικασία εκ των γενετικών αλγορίθμων εμπνευσμένη [11]. Στα προηγούμενα στάδια του tabu search έχει ευρεθεί η καλύτερη ημιεφικτή λύση, θα την ονομάσουμε *semi elite* γονέα ή χρωμόσωμα. Ονομάζεται αλγόριθμος διασταύρωσης γονέων ή *crossover (sexual mating)*. Εφαρμόζεται ως μια μορφή εντατικότερης ευρετικής αναζήτησης πάνω στη λίστα με τα τοπικά ελάχιστα της tabu αναζήτησης που εδώ την ονομάζουμε πληθυσμό ατόμων (population) ή χρωμοσωμάτων ή υποψήφιων γονέων. Εξορισμού οι λύσεις αυτές ως προϊόντα αναζήτησης είναι καλές και μοιάζουν μεταξύ των. Βάσει της γενετικής φιλοσοφίας κάθε τοπικό ελάχιστο θα θεωρείται ένα άτομο (individual) ή χρωμόσωμα ή υποψήφιος γονέας, δεδομένου ότι οι υποψήφιοι γονείς είναι καλής ποιότητας είναι σαν να ανήκουν στο ίδιο «είδος» παραδείγματος χάριν γάτας. Επομένως θα υπάρχει και μεγάλη πιθανότητα αυτοί να δύναται να ζευγαρώσουν (sexual mating). Κριτήριο ζευγαρώματος είναι ο περιορισμός μονοφασικής εξυπηρέτησης κάθε πελάτη. Έκαστο προκύπτον ζεύγος γονέων θα παράγει ένα τουλάχιστον τέκνο, δηλαδή ένα διαφορετικό δρομολόγιο με την ελπίδα αυτό να είναι εφικτό (elite), ή και τουλάχιστον καλύτερης ποιότητας από το τρέχον καλύτερο ημιεφικτό (semi elite). Το σημείο επιχιασμού (cut point) θα επιλέγεται τυχαία με ομοιόμορφη κατανομή στο $U(\frac{n}{5}, \frac{n}{2})$, όπου $n \triangleq$ πλήθος πελατών. Η διαδικασία φαίνεται στο σχήμα 22.



Σχήμα.22 Διαδικασία διασταύρωσης γονέων - mating

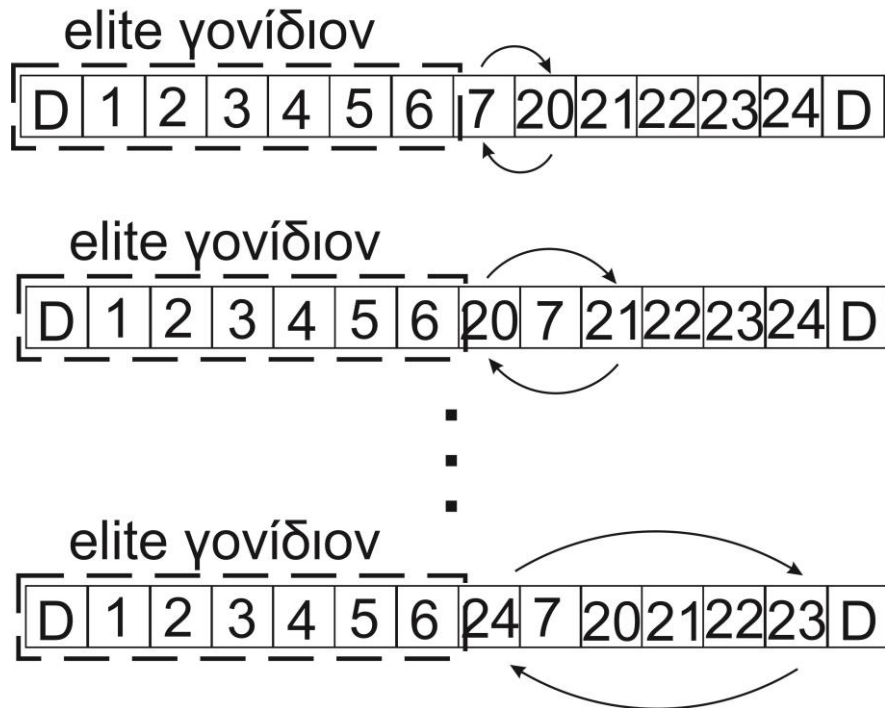
Η Διασταύρωση ή crossover συνήθως αποδίδει καρπούς όταν και οι δυο γονείς φέρουν μια τουλάχιστον «ανωμαλία» σε διαφορετικά μέρη της διαδρομής. Έτσι τα καλά χαρακτηριστικά των δυο γονέων μεταφέρονται εις το τέκνον, όπως παρόμοια συμβαίνει και στη φύση.

Η ανωτέρω διαδικασία θα παράξει είτε ένα elite δρομολόγιο είτε ένα semi elite γονέα, στη δεύτερη περίπτωση θα εφαρμοστεί η άφυλος μετάλλαξις (**asexual mutation** ή **mutation**). Δηλαδή θα κληθεί ο semi elite γονέας να μεταλλαχτεί χρήσει μοναδιαίου τελεστή swap, έτσι ώστε να παράξει ένα elite δρομολόγιο. Το σκεπτικό εδώ είναι ότι ο semi elite γονέας όντας πολύ καλής ποιότητας θα φέρει κάποιο γονίδιο (τμήμα του) εκκινόν από την αποθήκη το οποίο θα είναι εφικτό (elite γονίδιο). Το υπόλοιπο γονίδιο του semi elite γονέα θα θεωρηθεί γενικά ως semi elite γονίδιο, κάθε κόμβος του οποίου θα προσπαθείται να μεταλλαχτεί στο τρέχον κάθε φορά elite γονίδιο με γραμμικό τρόπο μέχρις ότου να παραχθεί το επίμαχον elite δρομολόγιο και η διαδικασία να τερματιστεί. Κριτήριο μετάλλαξης εδώ είναι η παραγωγή elite

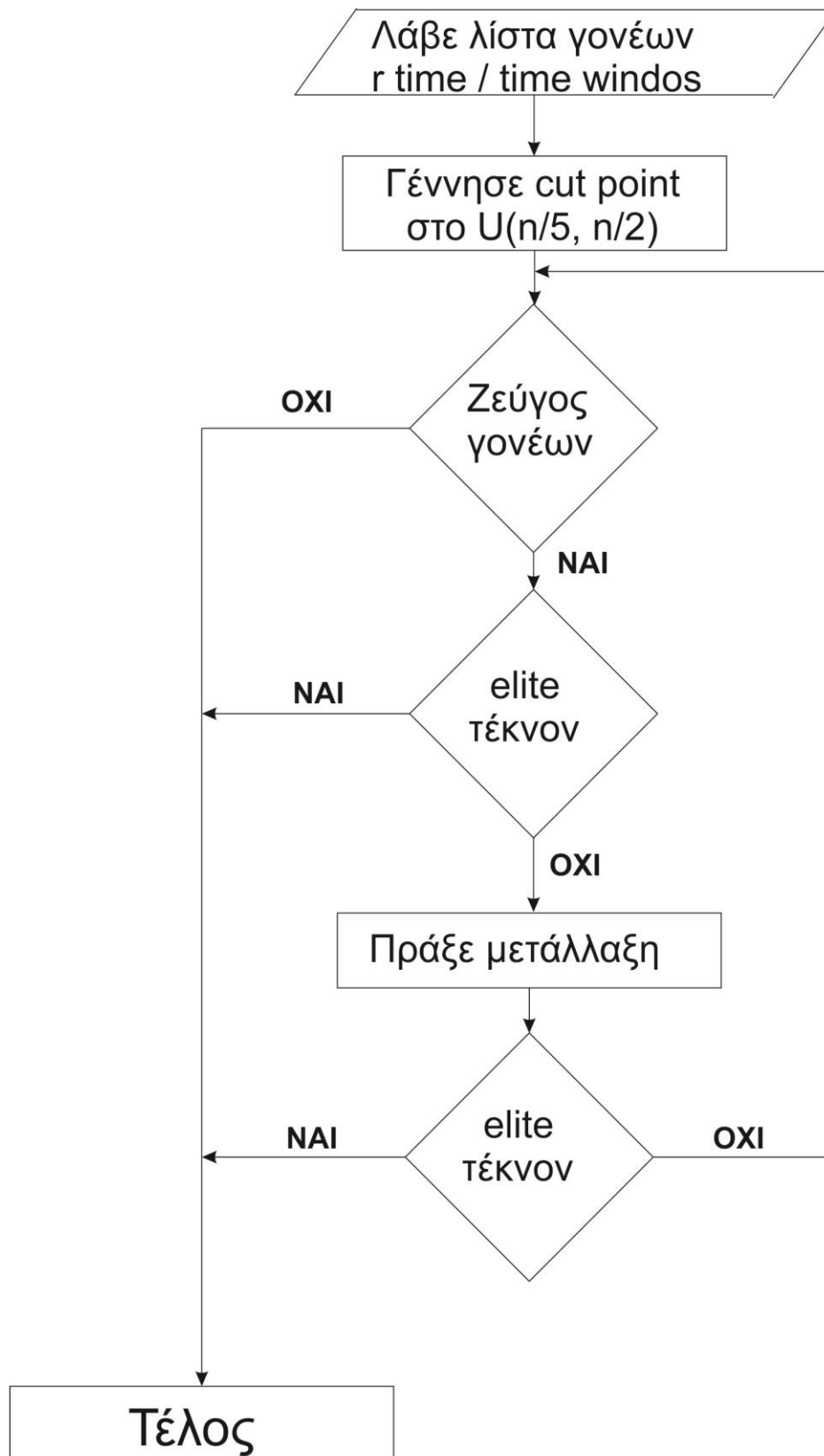
γονιδίου, η ποιότητα δεν λαμβάνεται υπόψη, διότι πρώτον το elite γονίδιο είναι ήδη καλής ποιότητας και δεύτερον διότι το semi elite γονίδιο, είναι και αυτό -αν και ανέφικτο- καλής ποιότητας, έτσι οποιαδήποτε μετάλλαξη και αν λάβει χώρα το προκύπτον κάθε φορά elite γονίδιο θα είναι και αυτό καλής ποιότητας (υποβέλτιστο ή και βέλτιστο). Η διαδικασία είναι άπληστη καθότι κοιτάζει μόνον μπροστά. Τα σενάρια είναι τα εξής: περιπτώσει που ο semi elite γονέας είναι η βέλτιστη ημιεφική λύση μετά τη μετάλλαξη να παραχθεί μια λίγο χειρότερης ποιότητας εφικτή λύση, θυσιάζουμε δηλαδή τη συντομότερη διαδρομή του οχήματος στο βωμό της συντομότερης εφικτότητας αυτής. Περιπτώσει που ο semi elite γονέας είναι μια διαδρομή κοντά στη βέλτιστη, αυτή είτε θα γίνει βέλτιστη και εφικτή, είτε υποβέλτιστη εφικτή κοντά στην υποβέλτιστη ημιεφική. Ακολουθούν τα βήματα :

Βήμα 0: Λάβε λίστα γονέων / r time / time windows
Βήμα 1: Γέννησε cut point στο $U(n/5, n/2)$ → Βήμα 2
Βήμα 2: Αν υπάρχει ζεύγος γονέων, ζευγάρωσε → Βήμα 4 αλλιώς → Βήμα 6
Βήμα 4: Αν το τέκνον είναι elite → Βήμα 6 αλλιώς → Βήμα 5
Βήμα 5: Πράξε μετάλλαξη στο τέκνον, αν υπάρξει elite → Βήμα 6 αλλιώς → Βήμα 2
Βήμα 6: Τέλος

Πίνακας.11 Ψευδοκώδικας mating – mutation



Σχήμα.17 Διαδικασία μετάλλαξης τέκνων - mating



Σχήμα.23 Διάγραμμα ροής mating - mutation

Κεφάλαιο 4

Αλγόριθμος GIRSOT

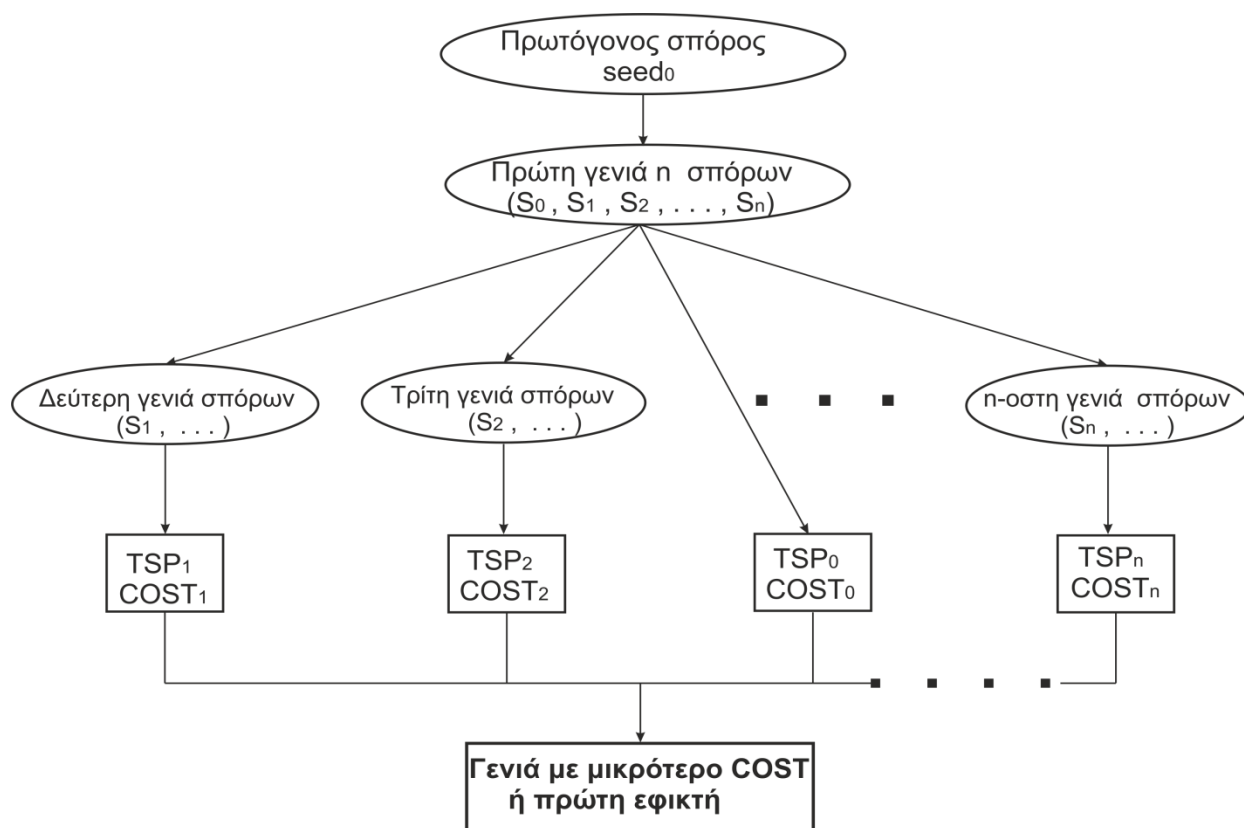
4.1 Εισαγωγικά

Κατά τα προηγούμενα κεφάλαια έγινε η παρουσίαση του προβλήματος, προτάθηκε ένας καλός αλγόριθμος ομαδοποίησης των πελατών (3-PCC) που όπως υπολογίστηκε στο δεύτερο κεφάλαιο αποδίδει $ENIF=0.92$. Στο τρίτο κεφάλαιο έγινε μια προσπάθεια να ελαχιστοποιηθεί όσο περισσότερο γίνεται η απόσταση έκαστου οχήματος και να ευρεθεί ένα εφικτό δρομολόγιο, με τους κόμβους που του έχει ορίσει ο 3-PCC. Το αποτέλεσμα που προκύπτει έως τώρα δεν είναι ούτε απογοητευτικό αλλά και ούτε αρκετό. Το ερώτημα που τίθεται είναι αν ο αλγόριθμος 3-PCC δίνει μια δρομολόγηση που να είναι εφικτή, αν είναι ημιεφικτή είναι η ελάχιστη δυνατή και αν δεν είναι, πόσο μακριά είναι από τη βέλτιστη (;). Δια την απάντηση στα ερωτήματα αυτά προτείνεται ένας ακόμα αλγόριθμος, ο αλγόριθμος σποράς (seeding algorithm).

Στο κεφάλαιο δυο περιγράφεται πως ο 3-PCC ξεκινώντας από ένα σπόρο(τον μακρύτερο πελάτη απτήν αποθήκη), δημιουργεί όλες τις υπόλοιπες συστάδες. Το σπόρο αυτό θα τον ονομάσουμε *πρωτόγονο (primitive seed)*. Η δρομολόγηση που προκύπτει από το πρωτόγονο σπόρο θα λέγεται *σπορά (seeding)*. Τι θα συνέβαινε αν ο σπόρος αυτός ήταν διαφορετικός (;). Δεν υπάρχει απάντηση, παρόλα αυτά λόγω των φάσεων απληστίας υπάρχει η πιθανότητα θέτοντας κάποιο άλλο σπόρο ως πρωτόγονο να προκύψει μια καλύτερη δρομολόγηση, έτσι προκύπτει ο αλγόριθμος σποράς [παράβαλε 4.2]. Επόμενο βήμα θα είναι μια ύστατη προσπάθεια μείωσης ακόμα περισσότερο του κόστους, καθώς θα γίνει προσπάθεια αρχικά τμήματα δρομολογίων να ανατεθούν σε άλλα οχήματα γειτονικών δρομολογίων (group relocate). Τέλος αν όλες οι ανωτέρω διαδικασίες δεν επιφέρουν εφικτή δρομολόγηση αναλαμβάνει η μέθοδος *χρονικής υποτίμησης (Devaluation Time Algorithm – DTA)*, η οποία εξαναγκάζει τη 3-PCC να δώσει νέα αρχική λύση (διατηρώντας τις καλές ιδιότητες και χαρακτηριστικά της) με ένα όχημα περισσότερο, όλη η διαδικασία επαναλαμβάνεται έως ότου ο στόλος φτάσει το μέγιστο διαθέσιμο ή μέχρι να βρεθεί εφικτή δρομολόγηση. Η ανωτέρω μεθοδολογία είναι ιεραρχική, προσπαθεί πρώτα να ελαχιστοποιήσει το στόλο και έπειτα τη διανυθείσα απόσταση.

4.2.1 Αλγόριθμος σποράς(seeding algorithm)

Ο αλγόριθμος 3-PCC όταν εκτελείται και παραγάγει μια ομαδοποίηση , προκύπτει και ένα διάλυμα σπόρων όπως περιγράφεται στο κεφάλαιο δυο [παράβαλε 2.2.1]. Εν συνεχεία στην ομαδοποίηση αυτή βελτιστοποιούνται ανεξάρτητα τα δρομολόγια(m-TSP) και προκύπτει το κόστος. Ο 3-PCC επαναλαμβάνεται μετά τη πρώτη κλήση του και κάθε φορά ως πρωτόγονος σπόρος τίθεται ο επόμενος της πρώτης σποράς έως ότου χρησιμοποιηθεί όλη η πρώτη σπορά. Αν για παράδειγμα η πρώτη κλήση του 3-PCC δίδει τη σπορά $\{ s_1, s_2, s_3, s_4 \}$, αν δεν δώσει εφικτή δρομολόγηση θα κληθεί άλλες τρεις το πολύ φορές με πρωτόγονους s_2, s_3, s_4 αντίστοιχα. Συνέπεια τεσσάρων καλών διαφορετικών αρχικών λύσεων. Το παιχνίδι της αναζήτησης γίνεται πολυεναρκτήριο, η κάθε μια βελτιστοποιείται σε επίπεδο δρομολογίου (m - TSP) και επιλέγεται η καλύτερη από αυτές βάσει κόστους ή η εφικτή. Η επιλογή του πρωτόγονου κάθε φορά σπόρου από τη πρωτόγονη σπορά με γεννήτρια τυχαίων αριθμών και με τη προϋπόθεση ότι αυτός δεν έχει ξαναχρησιμοποιηθεί. Ακολουθούν τα βήματα:

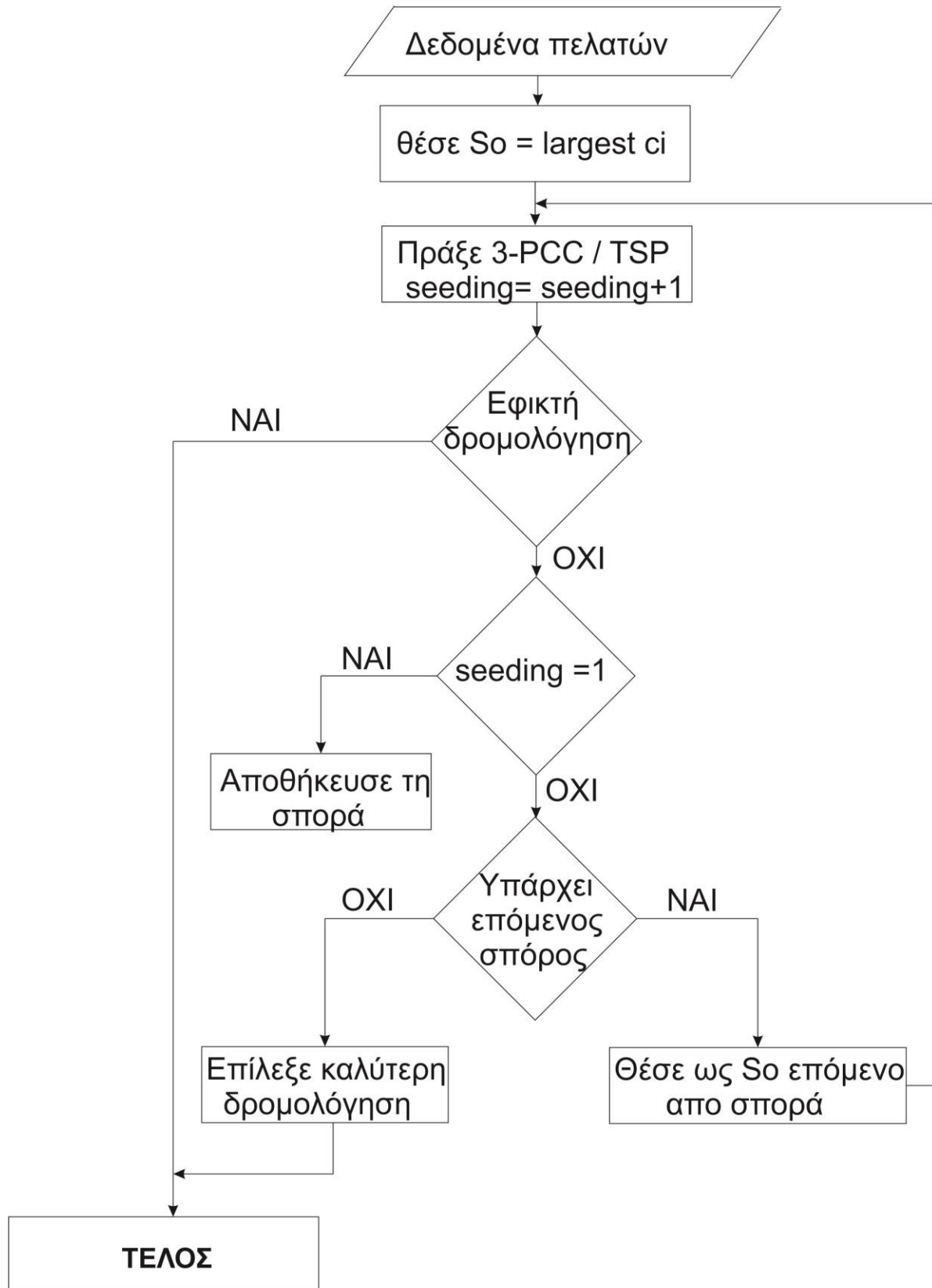


Σχήμα.24 Αλγόριθμος σποράς

Βήμα 0: Λάβε δεδομένα πελατών → Βήμα 1
Βήμα 1: Θέσε πρωτόγονο σπόρο $S_0 \leftarrow$ μακρύτερο C_i από depot → Βήμα 2
Βήμα 2: Πράξε 3-PCC , m-TSP , αν δρομολόγηση εφικτή → Βήμα 5 αλλιώς αν πρώτη σπορά, Αποθήκευσε τη → Βήμα 3 αλλιώς → Βήμα 3
Βήμα 3: Αν υπάρχει επόμενος σπόρος διάλεξε τυχαία θέσε το ως S_0 → Βήμα 2 αλλιώς → Βήμα 4
Βήμα 4: Επίλεξε τη καλύτερη δρομολόγηση → Βήμα 5
Βήμα 5: Τέλος

Πίνακας.12 Ψευδοκώδικας σποράς(seeding algorithm)

Είναι προφανές ότι η διαδικασία θα είναι ταχύτατη όταν από την πρώτη σπορά ευρίσκεται εφικτή λύση. Ωστόσο δεν θα είναι και τόσο άσχημα τα πράγματα όσο αναζητούνται και άλλες αρχικές λύσεις στο πρόβλημα. Επισημαίνεται ότι στο στάδιο αυτό όλες οι γενεές αρχικών λύσεων θα περιέχουν σχεδόν με βεβαιότητα ίδιο πλήθος οχημάτων. Διαφέρει η κατανομή των πελατών στον ήδη ευρεθέντα στόλο. Στη συνέχεια θα δειχθεί πως όταν το στάδιο αυτό αποτύχει στο να εύρη εφικτή λύση η 3-PCC εξαναγκάζεται στο να αυξήσει τα οχήματα του στόλου.



Σχήμα.25 Διάγραμμα ροής αλγορίθμου σποράς

4.2.2 Μέθοδος μαζικής ανάθεσης - Group Relocate

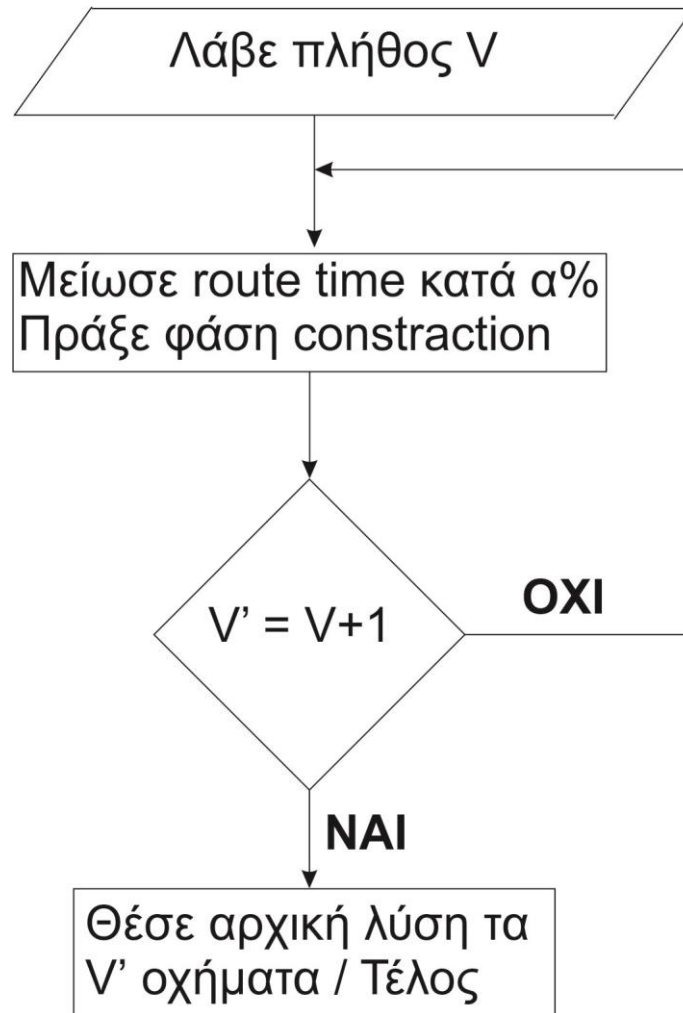
Τελευταίος αλγόριθμος βελτιστοποίησης είναι ο Group Relocate, μια διαδικασία εν γένει μαζικής ανάθεσης πελατών από ένα όχημα εις ένα άλλο που πραγματοποιεί γειτονικό δρομολόγιο. Είναι η πρώτη και τελευταία προσπάθεια ανάμειξης των δρομολογίων με ευρετική τεχνική, η προηγούμενη προσπάθεια της φάσης adjustment [παράβαλε 2.2.3] ήταν καθαρά άπληστη αλλά και αποδίδουσα. Τα κριτήρια της μεθόδου είναι δυο, αναμφίβολα το χωρικό, της μη παραβίασης χωρητικότητας των οχημάτων, το δεύτερον η μείωση της διανυθείσας απόστασης και για τα δυο εμπλεκόμενα κάθε φορά δρομολόγια. Η ανάθεση θα λαμβάνει χώρα για τουλάχιστον ένα πελάτη ($1 - 0$ relocate) ή και για n πλήθος πελατών ($n - 0$ relocate), από την αρχή ή το τέλος του δρομολογίου δότη στην αρχή ή το τέλος του δρομολογίου δέκτη, για κάθε αποδοχή relocate. Αφού και ο περιορισμός χωρητικότητας πληρείται πραγματοποιείται βελτιστοποίηση του τμήματος του δρομολογίου δέκτη (receiver route) με αλγόριθμο πλησιέστερου Γείτονος (nearest neighbor), και εν συνεχεία ελέγχεται κατά πόσο οι αποστάσεις των δυο (receiver, giver route) έχουν μεταβληθεί. Ο λόγος που επιλέγεται πλησιέστερος Γείτων δεν είναι τυχαίος, αφενός είναι ένας αμιγώς άπληστος (και ταχύτατος) αλγόριθμος, αφετέρου αποδίδει πολύ καλά αφού τα προς βελτιστοποίηση τμήματα εξορισμού είναι μικρά σε μέγεθος. Επομένως η μέθοδος δεν είναι εξολοκλήρου ευρετική αφού φέρει και άπληστη φάση.

4.2.3 Μέθοδος χρονικής υποτίμησης

Τελευταίος αλγόριθμος που χρησιμοποιείται είναι η μέθοδος χρονικής υποτίμησης (Devaluation Time Algorithm – DTA), με σκοπό την παραγωγή αρχικής λύσης με ένα περισσότερο όχημα απ' τον τρέχων κάθε φορά στόλο. Ο αλγόριθμος 3-PCC όπως περιγράφηκε στο κεφάλαιο δυο φέρει αρκετές παραμέτρους και κάνει χρήση των περιορισμών χωρητικότητας των οχημάτων και του μέγιστου χρόνου δρομολογίου (route time), όσο ο χρόνος αυτός μειώνεται τόσο ο παραχθείς στόλος αυξάνεται (βλέπε σχήμα 20) καθώς τα οχήματα που δρομολογούνται τον ξεπερνούν και καλούνται νέα οχήματα μέχρις ότου ενταχθούν όλοι οι πελάτες. Έτσι επιλέγεται ένα βήμα μείωσης του route time 2% μέχρις ότου ο στόλος αποκτήσει κάθε φορά τα οχήματα που έχουν ζητηθεί από τον αλγόριθμο GIRSOT. Πλεονέκτημα της μεθόδου DTA είναι ότι παράγει αρχικές λύσεις με περισσότερα οχήματα διατηρώντας όλα τα καλά χαρακτηριστικά του 3-PCC. Η μέθοδος είναι πολύ απλή, ακολουθούν τα βήματα:

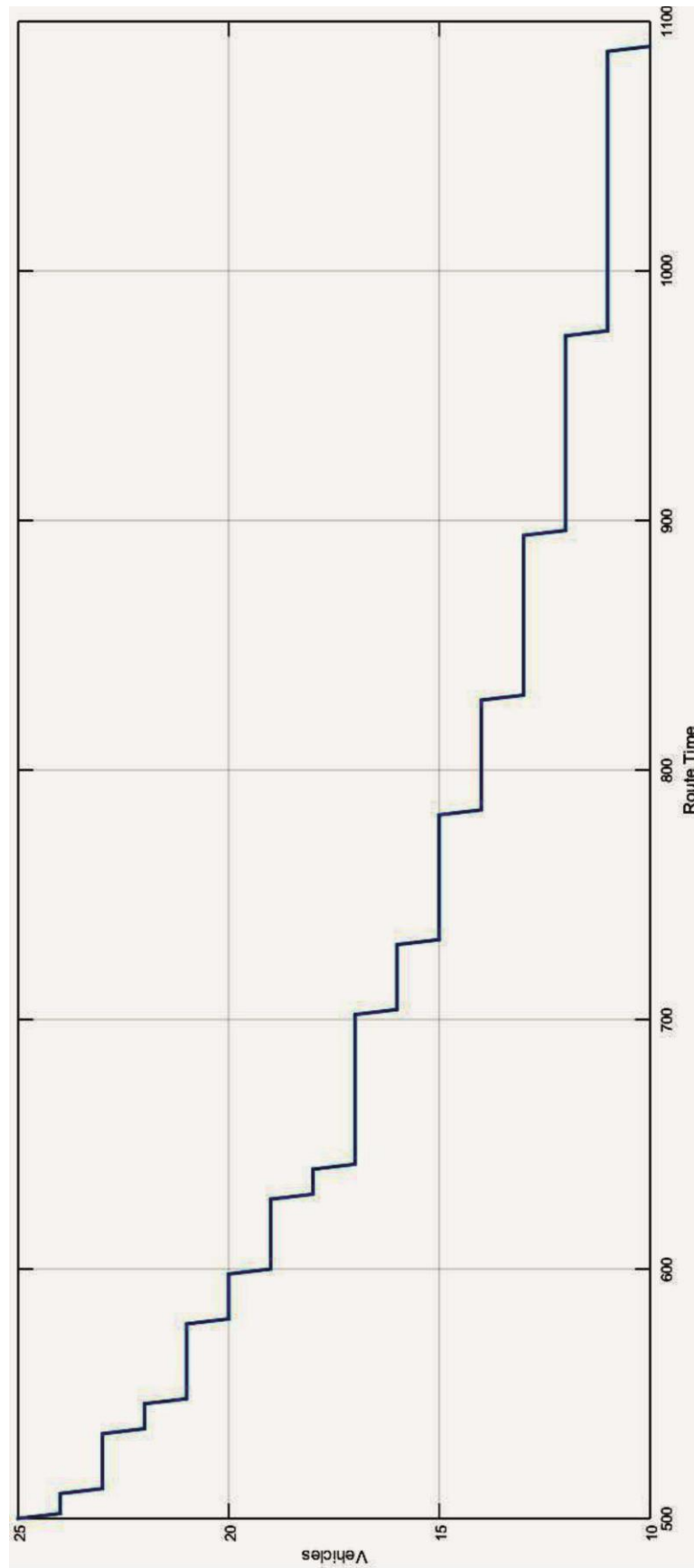
Βήμα 0: Λάβε πλήθος οχημάτων $V \rightarrow$ Βήμα 1
Βήμα 1: Θέσε $\text{route time}' = \text{route time} - 2\%$ / πράξε construction(πίνακας 2) \rightarrow Βήμα 2
Βήμα 2: Αν $V' = V + 1 \rightarrow$ Βήμα 3 αλλιώς \rightarrow Βήμα 1
Βήμα 3: Θέσε αρχική λύση με V' οχήματα / Τέλος

Πίνακας.13 Αλγόριθμος χρονικής υποτίμησης DTA



Σχήμα.19 Διάγραμμα ροής αλγορίθμου χρονικής υποτίμησης

Στο παρακάτω σχήμα παρουσιάζεται πώς ο στόλος της 3-PCC (φάση construction) μεταβάλλεται συναρτήσει του ωραρίου των οχημάτων για το πρόβλημα C101. Παρατηρείται ότι ταυτίζεται με τη γενικευμένη συνάρτηση *Heaviside* ή τυχαίου περιπάτου (random walk). Όπως φαίνεται ο τυχαίος περίπατος εκκινεί από το χρόνο 1.136 όπου δίνει και τον ελάχιστο δυνατό στόλο $fleet_{min}$ [παράβαλε 2.2.2] για το πρώτο περιορισμό δηλαδή χωρητικότητας των οχημάτων στις 200 μον. Όγκου, ενώ τερματίζει για χρόνο 500 με το δεύτερο περιορισμό που είναι ο μέγιστος διαθέσιμος στόλος στα 25 οχήματα.



Σχήμα.20 Μέγεθος στόλου 3-PCC συναρτήσει route time

2.2.4 Ενοποίηση όλων, αλγόριθμος GIRSOT

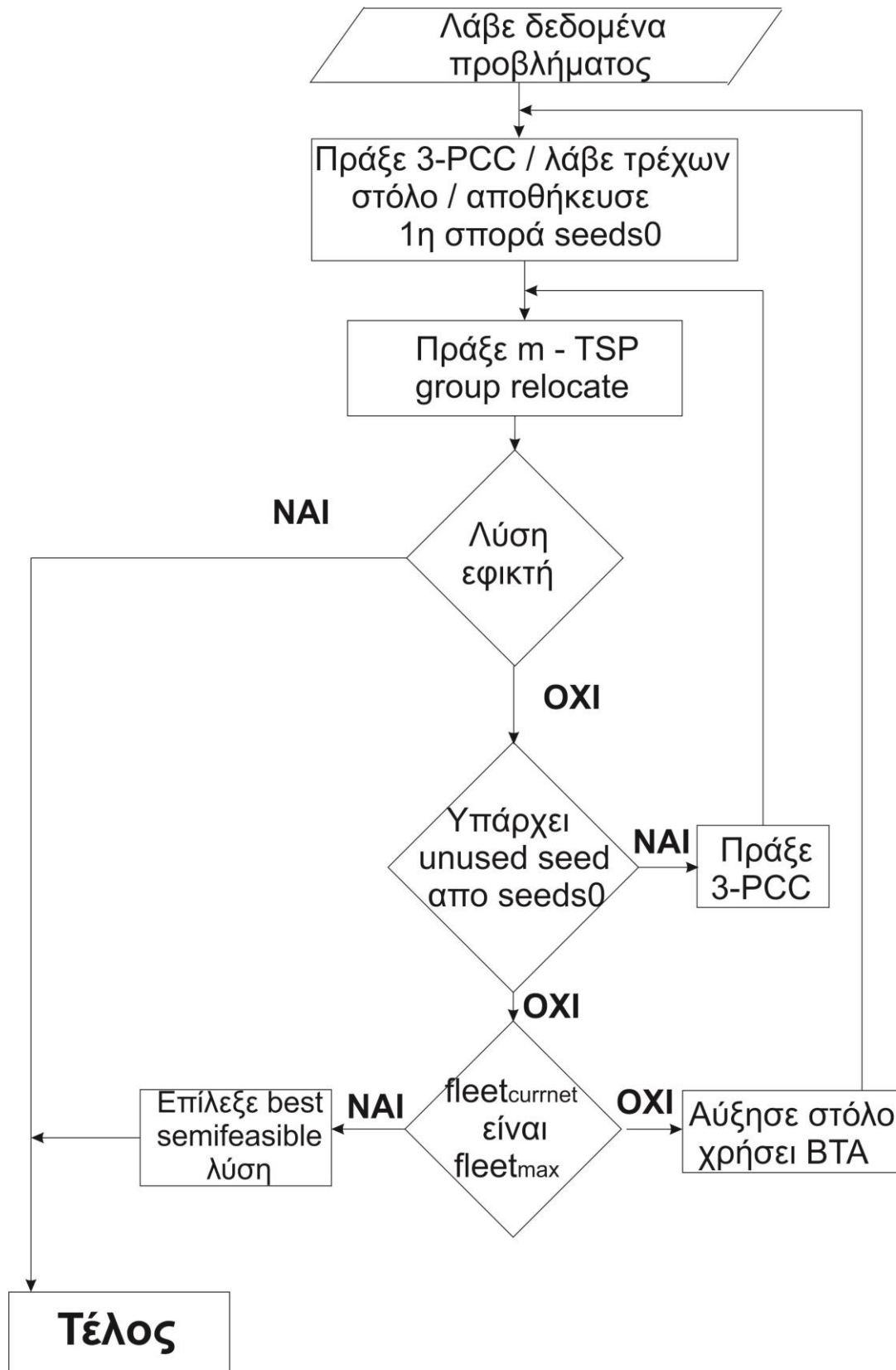
Σύμφωνα με τα όσα αναλύθηκαν στα προηγούμενα κεφάλαια συνοψίζονται εις τους κάτωθι αλγορίθμους / μεθόδους:

1. Δημιουργίας αρχικής λύσης – αλγόριθμος 3-PCC
2. Βελτιστοποίηση επιμέρους διαδρομών – αλγόριθμος βελτιστοποίησης δρομολογίου
3. Δημιουργία διαφορετικών αρχικών λύσεων – αλγόριθμος σποράς
4. Βελτιστοποίηση μεταξύ δρομολογίων – αλγόριθμος ομαδικής ανάθεσης
5. Δημιουργίας αρχικών λύσεων με περισσότερα οχήματα – αλγόριθμος χρονικής υποτίμησης

Σε αυτό το σημείο προτείνεται ένας αλγόριθμος ενοποίησης των ανωτέρω ονόματι « *Αναζήτηση μεταφορών με άπληστη αρχικοποιημένη δρομολόγηση – Greedy Initialized Routing Search On Transportations (GIRSOT)* ». Είναι μια ιεραρχική μέθοδος αναζήτησης ελαχιστοποιώντας πρώτα το στόλο και έπειτα τη διανυθείσα απόσταση, πλεονέκτημα της είναι η υψηλή διακριτοποίηση καθώς απαρτίζεται από διαφορετικά μέρη και η αποτελεσματικότητα του εξαρτάται από την αποτελεσματικότητα των επιμέρους αλγορίθμων 1 - 5 . Ακολουθούν τα βήματα:

Βήμα 0: Λάβε δεδομένα προβλήματος → Βήμα 1
Βήμα 1: Πράξε 3-PCC / λάβε τρέχων στόλο / αποθήκευσε πρωτόγονη σπορά Seeds0 → Βήμα 1.2
1.2: Πράξε $m - \text{TSP} / \text{group relocate}$, αν δρομολόγηση εφικτή → Βήμα 6 Αλλιώς → Βήμα 2
Βήμα 2: Αν υπάρχει αχρησιμοποίητος σπόρος από seeds0 διάλεξε ένα τυχαία Ως πρωτόγονο → Βήμα 3 αλλιώς → Βήμα 4
Βήμα 3: Πράξε 3-PCC → Βήμα 1.2
Βήμα 4: Αν τρέχων στόλος $\leq \text{fleet}_{\max}$ αύξησε κατά ένα όχημα χρήσει DTA → Βήμα 1 Αλλιώς → Βήμα 5
Βήμα 5: Επίλεξε καλύτερη ημιεφικτή δρομολόγηση → Βήμα 6
Βήμα 6: Τέλος

Πίνακας.14 Ψευδοκώδικας αλγορίθμου GIRSOT



Σχήμα.21 Διάγραμμα ροής αλγορίθμου GIRSOT

Κεφάλαιο 5

Αποτελέσματα - Συμπεράσματα

5.1 Προβλήματα βιβλιογραφίας

Οι δυνατότητες, τα πλεονεκτήματα και μειονεκτήματα του προτεινόμενου αλγορίθμου GIRSOT διερευνήθηκαν χρήσει κάποιων προβλημάτων από τη βιβλιογραφία, συγκεκριμένα ανάμεσα σε δυο μελετητών του Solomon [24] και των Gehring & Homberger's [18]. Τα προβλήματα του Solomon είναι κλίμακας εκατό πελατών ενώ των Gehring & Homberger's διακοσίων. Τα προβλήματα αυτά ενσωματώνουν αρκετά χαρακτηριστικά πραγματικών προβλημάτων δρομολόγησης και μπορούν να θεωρηθούν ως αμερόληπτο δείγμα για την λειτουργία ενός επιλυτικού μοντέλου. Παρόλα αυτά φέρουν και νοσηρά σημεία όπως παραδείγματος χάρη δεν ανταποκρίνονται στη πραγματικότητα εις ότι αφορά τους χρόνους εξυπηρέτησης των πελατών, καθώς αυτοί παρουσιάζονται ως ίδιοι για κάθε πελάτη παρόλο που εν γένει έκαστος πελάτης ζητεί και διαφορετική ποσότητα προϊόντων. Είναι προφανές ότι υπάρχει μια αναλογική σχέση μεταξύ ζήτησης και χρόνου εξυπηρέτησης. Εις ότι αφορά τη διανομή των πελατών εις τη γεωγραφική περιοχή είναι αρκετά αντιπροσωπευτικά, επίσης τα χρονικά παράθυρα είναι εις αναλογίαν του χρόνου προγραμματισμού με ποικιλία εις ότι αφορά τη στενότητα ή χαλαρότητα των. Αναλυτικότερα δια τα προβλήματα του Solomon μπορούν να λεχθούν τα εξής:

- 56 διαθέσιμα προβλήματα με πλήθος πελατών 100 και κατανομή αυτών εις 10000 τετραγωνικές μονάδες
- Οι κατηγορίες προβλημάτων είναι 6 και οι κλάσεις ανάμεσα σ' αυτές 3, οι κατηγορίες R1, R2 ανήκουν στη κλάση R, οι κατηγορίες C1, C2 στη κλάση C και οι κατηγορίες RC1, RC2 στη κλάση RC, το πλήθος έκαστης κατηγορίας κυμαίνεται από 8 έως 12 προβλήματα
- Για τη κλάση R (random) οι θέσεις των πελατών είναι ομοιόμορφα τυχαία κατανεμημένες

- Για τη κλάση RC (random – cluster) είναι κατά το ήμισυ ομοιόμορφα τυχαίοι και κατά το υπόλοιπο ήμισυ κατανεμημένοι σε ομάδες
- Τέλος η κλάση C (cluster) όλοι οι πελάτες είναι κατανεμημένοι σε ομάδες με τα κέντρα αυτών ομοιόμορφα τυχαία κατανεμημένα.
- Εις ότι αφορά το χρονικό ορίζοντα προγραμματισμού οι κατηγορίες C1, R1, RC1 έχουν σύντομο χρόνο και κατά μέσον όρο 5 με 10 πελάτες ανά όχημα ενώ τα οχήματα έχουν χωρητικότητα 200 μονάδες όγκου.
- Οι κατηγορίες C2, R2, RC2 αντίθετα έχουν μεγάλο ορίζοντα προγραμματισμού και χωρητικότητα 700 μονάδων όγκου, κατά μέσον όρο 30 πελάτες δύναται να εξυπηρετούνται από ένα όχημα.
- Εις ότι αφορά τα χρονικά παράθυρα σε κάθε υποκατηγορία από 1 έως 4 αυτά γίνονται από πολύ αυστηρά σε χαλαρότερα.
- Για κάθε κλάση οι πελάτες είναι κατανεμημένοι στις ίδιες θέσεις ενώ διαφοροποιούνται τα χρονικά παράθυρα και ο ορίζοντας προγραμματισμού.

Εις ότι αφορά τα προβλήματα των Gehring & Homberger's κυμαίνονται εις το ίδιο μήκος κύματος με τα προβλήματα του Solomon με βασική ειδοποιό διαφορά το διπλάσιο πλήθος πελατών. Η χρήση μεγαλύτερων προβλημάτων επιτρέπει ένα ασφαλή συμπέρασμα εις ότι αφορά τους χρόνους εκτέλεσης του προγράμματος καθώς και τον δείκτη καθόδου [παραβάλε 3.2]. Τέλος αναφέρεται ότι σε όλα τα προβλήματα αυτών των δυο μελετητών η θέση της αποθήκης ευρίσκεται εις το μέσον της γεωγραφικής περιοχής των πελατών. Η υπόθεση αυτή δεν είναι πάντοτε ρεαλιστική -είναι η ιδανική- παρόλα αυτά η πιστότητα του προτεινόμενου αλγορίθμου εις ότι αφορά διαφορετικές θέσεις της αποθήκης, παραμένει σταθερή.

Ο προτεινόμενος αλγόριθμος GIRSOT υλοποιήθηκε σε προγραμματιστικό περιβάλλον java [10], για την ακρίβεια των δεκαδικών αριθμών επιλέχθηκε ο τύπος μεταβλητής double μεγέθους 64 bit, οι εκτελέσεις έλαβαν χώρα στον επεξεργαστή της Intel® Core™ i5 – 4460 CPU@ με συχνότητα ρολογιού στα 3,2 GHz και μνήμη τύπου RAM στα 4 GB, με λειτουργία του επεξεργαστή στα 64 bit και λειτουργικό σύστημα windows 10 home. Ο μέσος χρόνος εκτέλεσης όπως φαίνεται και από το παρακάτω πίνακα ήταν 1,97 λεπτά.

5.2 Σύγκριση αποτελεσμάτων GIRSOT με προβλήματα βιβλιογραφίας

Το δείγμα 25 προβλημάτων επιλέχθηκε με κριτήριο τη συνέπεια του αλγορίθμου σε ποιότητα και χρόνο εκτέλεσης, περιλαμβάνονται 20 προβλήματα του Solomon μεγέθους 100 πελατών και 5 προβλήματα των Gehring & Homberger's μεγέθους 200 πελατών, για κάθε πρόβλημα υπήρξαν μια σειρά 5 εκτελέσεων και απ'αυτές ελήφθησαν τα καλύτερα

αποτελέσματα με κριτήριο πρώτα το μέγεθος του στόλου και έπειτα της διανυθείσας απόστασης, αναλυτικά για κάθε πρόβλημα:

<i>a/a</i>	<i>Name</i>	<i>G.B Score</i>	<i>G.B N.V</i>	<i>GIRSAT score</i>	<i>GISOT N.V</i>	<i>G.B score error(%)</i>	<i>Δ(N.V)</i>	<i>CPU(min.)</i>
100 customers	1 C101	828.94	10	828.94	10	0.00	0	0.04
	2 C102	828.95	10	828.94	10	0.00	0	0.03
	3 C103	828.06	10	828.94	10	0.11	0	0.08
	4 C104	824.78	10	828.94	10	0.50	0	0.04
	5 C105	828.94	10	828.94	10	0.00	0	0.03
	6 C106	828.95	10	828.94	10	0.00	0	0.05
	7 C107	828.96	10	828.94	10	0.00	0	0.04
	8 C108	828.97	10	830.31	10	0.16	0	0.03
	9 C109	828.98	10	831.9	10	0.35	0	0.03
	10 C203	591.17	3	674.66	4	12.38	1	3.16
100 customers	11 C204	590.6	3	660.15	4	10.54	1	5.3
	12 R108	960.88	9	1104.77	12	13.02	3	0.3
	13 R104	1007.24	9	1155.5	13	12.83	4	0.4
	14 R204	825.52	2	923.74	6	10.63	4	8.3
	15 R211	885.71	2	1075.84	5	17.67	3	5.5
	16 R208	726.75	2	991.5	5	26.70	3	7.09
	17 RC104	1135.48	10	1281.52	11	11.40	1	0.1
	18 RC108	1139.82	10	1222.1	11	6.73	1	0.02
	19 RC204	798.41	3	965.88	5	17.34	2	5.5
	20 RC208	726.75	2	950.88	5	23.57	3	3.7
200 customers	21 C1_2_3	2707.35	18	2774.28	21	2.41	3	2.75
	22 C1_2_4	2643.31	18	2732.64	20	3.27	2	0.77
	23 C1_2_9	2687.83	18	2740.28	20	1.91	2	1.38
	24 C1_2_10	2643.51	18	2724.58	20	2.98	2	1.05
	25 RC1_2_4	2851.68	18	3426.51	19	16.78	1	1.12
	<i>mean</i>	1215.10	9.40	1314.78	10.84	7.65	1.44	2.03

Πίνακας.14 Αποτελέσματα GIRSAT

Εις ότι αφορά τη Τρίτη και τέταρτη στήλη του πίνακος 14 αναγράφονται τα παγκοσμίως γνωστά (Global Best) σε διανυθείσα απόσταση (Cost) και πλήθους οχημάτων (Number of Vehicles) που αναφέρονται σε μεθευρετικούς αλγορίθμους και όχι απλώς τα παγκοσμίως βέλτιστα διότι αυτά δεν αντιπροσωπεύουν το πραγματικό ελάχιστο του κόστους διανομής [παράβαλε 1.3].

Παρακάτω έχουν επιλεχθεί προς εικονογράφηση τα εξής προβλήματα από το δείγμα του Solomon:

1. Κλάσης C υποκατηγορίας 104.
2. Κλάσης C υποκατηγορίας 204.
3. Κλάσης R υποκατηγορίας 104.
4. Κλάσης R υποκατηγορίας 204.
5. Κλάσης RC υποκατηγορίας 104.
6. Κλάσης RC υποκατηγορίας 204.

Ενώ από το δείγμα των Gehring & Homberger's τα:

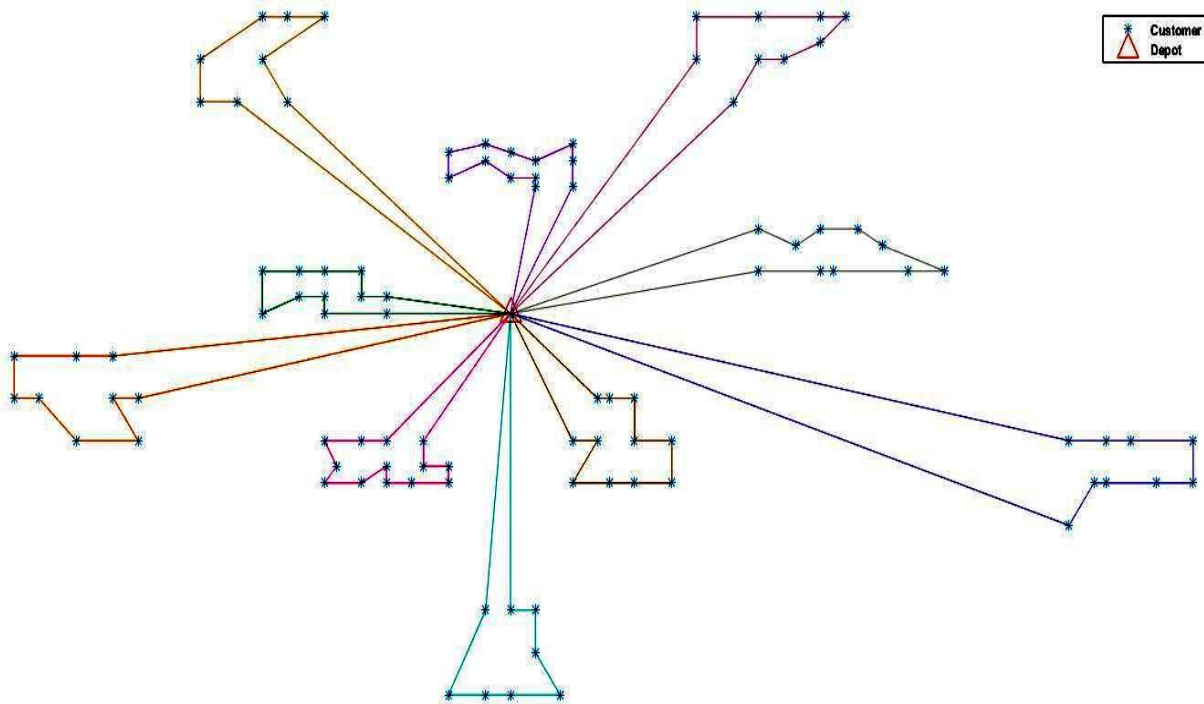
1. Κλάσης C υποκατηγορίας 1_2_4.
2. Κλάσης RC υποκατηγορίας 1_2_4.

Έκαστη εικονογράφηση συνοδεύεται με αναλυτικά στοιχεία για τα δρομολόγια όπως αυτά εκτυπώνονται από το λογισμικό, τα στοιχεία αυτά είναι :

1. Διανυθείσα απόσταση οχήματος (Distance).
2. Διανυθείσα απόσταση στόλου (Total Distance).
3. Χρόνος επιστροφής οχήματος στην αποθήκη (Route time).
4. Φορτίο οχήματος εκκινόν από την αποθήκη (Cargo).
5. Χρόνος αναμονής οχήματος στους πελάτες (delay).

Εις τη δομή δεδομένων του μητρώου των δρομολογίων η σήμανση της αποθήκης γίνεται με το n+1 πελάτη, για τα προβλήματα Solomon αποθήκη είναι το 101, ενώ για τα υπόλοιπα 201. Η εικονογράφηση των δρομολογήσεων έγινε με πρόγραμμα που αναπτύχθηκε για τις ανάγκες αυτής της μελέτης σε περιβάλλον MatLab, ο πηγαίος κώδικας και οδηγίες χρήσης του προγράμματος αυτού ευρίσκονται εις το παράρτημα 1.

C104

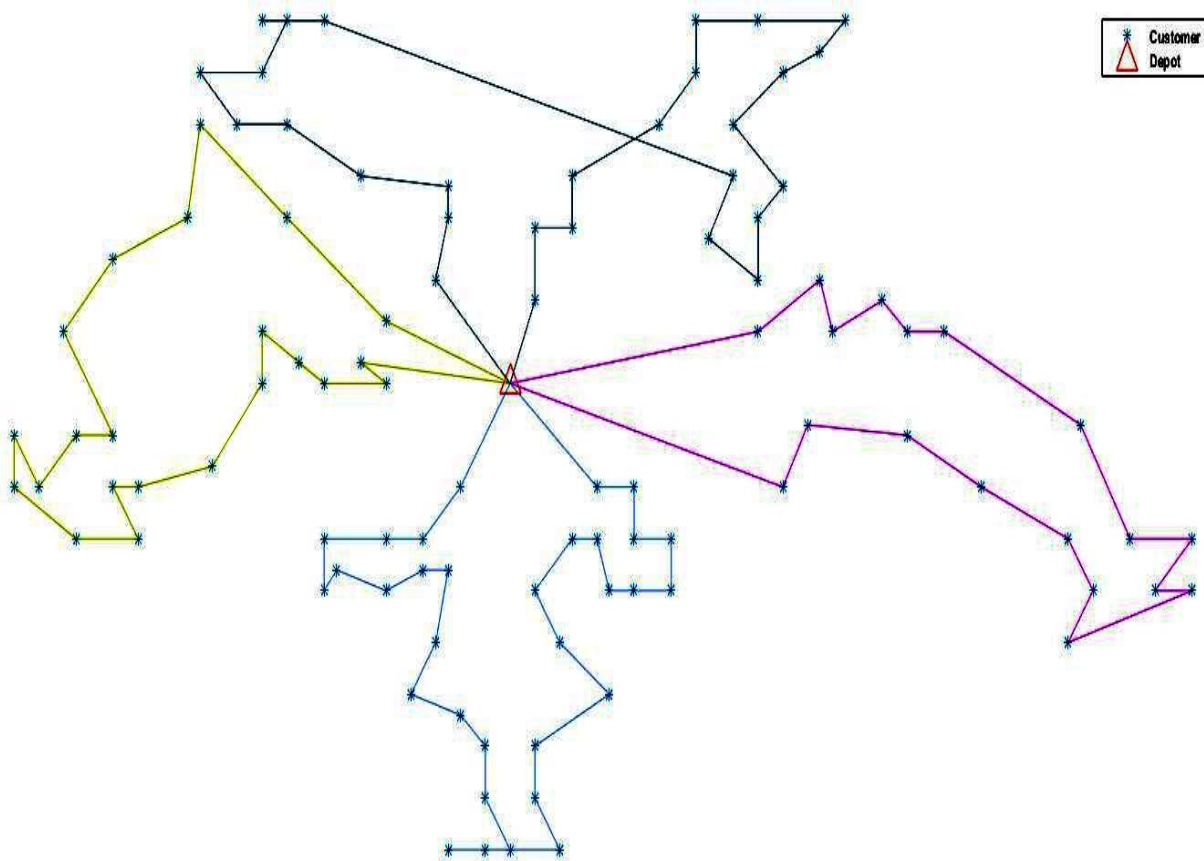


Σχήμα 22 Δρομολόγηση C104

Route #1: Feasible Distance: 127.30 | Cargo: 150 | Route Time: 937.30 | Delay: 0.00
 Route #2: Feasible Distance: 101.88 | Cargo: 200 | Route Time: 821.88 | Delay: 0.00
 Route #3: Feasible Distance: 95.94 | Cargo: 190 | Route Time: 905.94 | Delay: 0.00
 Route #4: Feasible Distance: 97.23 | Cargo: 200 | Route Time: 1070.84 | Delay: 163.61
 Route #5: Feasible Distance: 96.63 | Cargo: 190 | Route Time: 1074.44 | Delay: 257.81
 Route #6: Feasible Distance: 76.07 | Cargo: 170 | Route Time: 976.07 | Delay: 0.00
 Route #7: Feasible Distance: 64.81 | Cargo: 160 | Route Time: 1234.81 | Delay: 0.00
 Route #8: Feasible Distance: 59.40 | Cargo: 200 | Route Time: 1049.40 | Delay: 0.00
 Route #9: Feasible Distance: 50.80 | Cargo: 170 | Route Time: 1040.80 | Delay: 0.00
 Route #10: Feasible Distance: 59.62 | Cargo: 180 | Route Time: 1139.62 | Delay: 0.00

Route# 1: [101 81 78 76 71 70 73 77 79 80 101]
 Route# 2: [101 57 55 54 53 56 58 60 59 101]
 Route# 3: [101 98 96 95 94 92 93 97 100 99 101]
 Route# 4: [101 34 36 39 38 37 35 31 33 32 101]
 Route# 5: [101 13 15 12 14 16 19 18 17 101]
 Route# 6: [101 90 87 86 83 82 84 85 88 89 91 101]
 Route# 7: [101 43 42 41 40 44 46 45 48 51 50 52 49 47 101]
 Route# 8: [101 67 65 63 62 74 72 61 64 68 66 69 101]
 Route# 9: [101 20 24 25 27 29 30 28 26 23 22 21 101]
 Route# 10: [101 5 3 7 8 10 11 9 6 4 2 1 75 101]
 ##Total Distance:829.69

C204



Σχήμα 23 Δρομολόγηση C204

Route #1: Feasible Distance: 161.06 | Cargo: 310 | Route Time: 3307.49 | Delay: 1526.44
 Route #2: Feasible Distance: 162.95 | Cargo: 560 | Route Time: 3052.83 | Delay: 9.88
 Route #3: Feasible Distance: 196.18 | Cargo: 530 | Route Time: 2988.82 | Delay: 272.63
 Route #4: Feasible Distance: 156.98 | Cargo: 410 | Route Time: 2708.47 | Delay: 571.49

Route# 1:

[101 90 88 86 84 83 82 85 76 71 73 70 80 79 81 78 77 87 96 101]

Route# 2:

[101 67 63 62 74 72 61 64 66 69 68 65 49 55 54 53 58 60 56 59 57 40 44 46 41 42 45 50 51 52 47 43 48 101]

Route# 3:

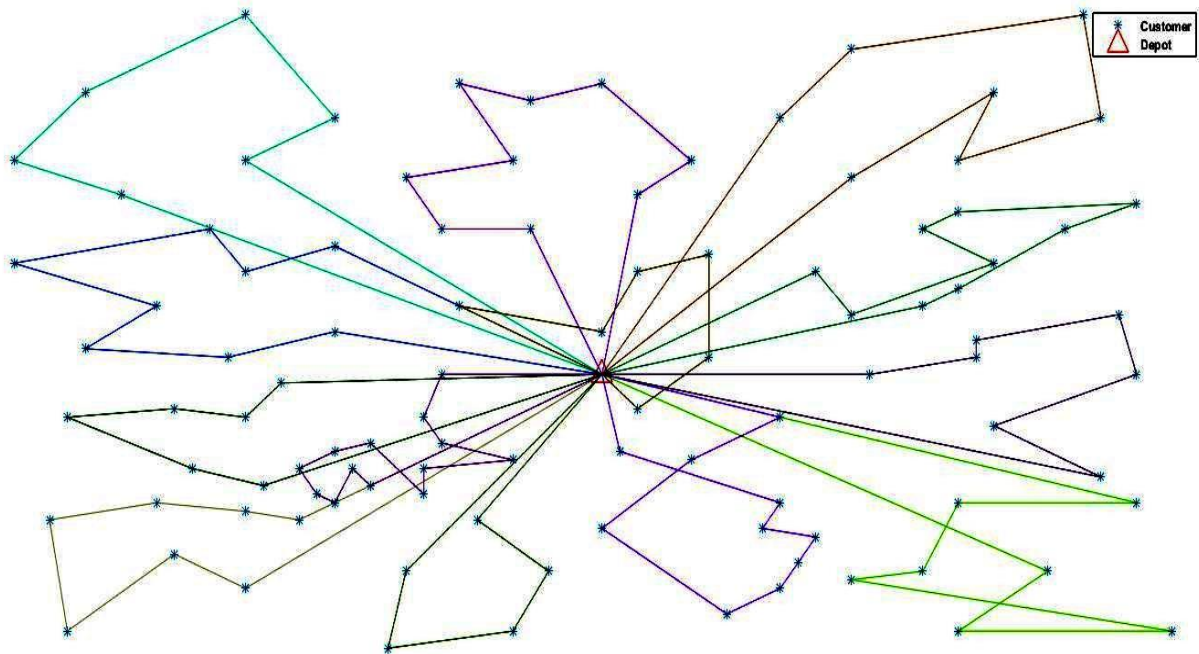
[101 93 5 75 2 1 99 100 97 92 94 95 98 3 4 91 89 7 12 16 14 15 19 17 13 9 11 10 8 101]

Route# 4:

[101 22 20 24 27 30 29 6 32 33 31 35 38 39 37 36 34 28 26 23 18 25 21 101]

##Total Distance:677.17

R104



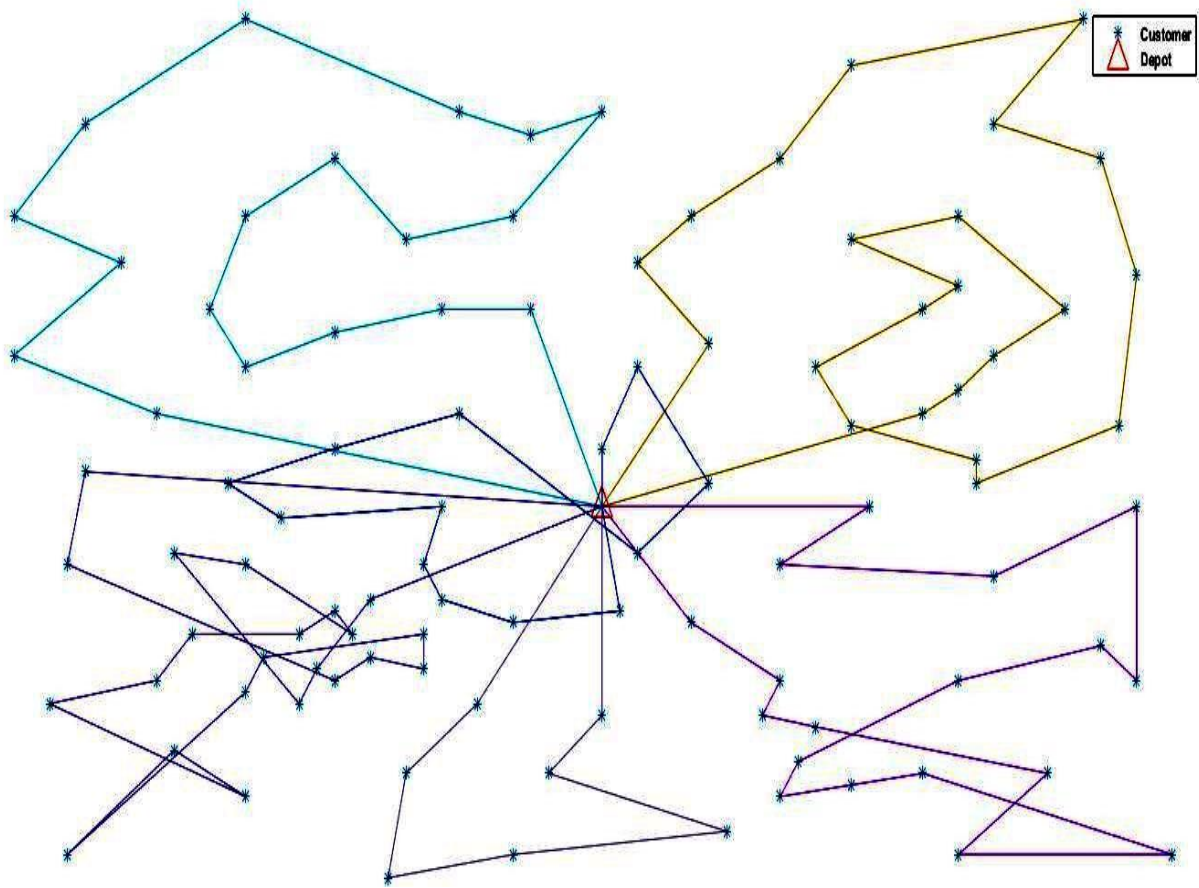
Σχήμα 24 Δρομολόγηση R104

Route #1: Feasible Distance: 74.71 | Cargo: 53 | Route Time: 124.71 | Delay: 0.00
 Route #2: Feasible Distance: 123.62 | Cargo: 103 | Route Time: 222.64 | Delay: 29.02
 Route #3: Feasible Distance: 115.18 | Cargo: 100 | Route Time: 193.09 | Delay: 17.91
 Route #4: Feasible Distance: 114.06 | Cargo: 134 | Route Time: 215.54 | Delay: 31.48
 Route #5: Feasible Distance: 100.16 | Cargo: 126 | Route Time: 219.04 | Delay: 48.89
 Route #6: Feasible Distance: 110.51 | Cargo: 130 | Route Time: 227.99 | Delay: 27.47
 Route #7: Feasible Distance: 90.55 | Cargo: 106 | Route Time: 171.98 | Delay: 1.43
 Route #8: Feasible Distance: 69.42 | Cargo: 92 | Route Time: 157.62 | Delay: 28.20
 Route #9: Feasible Distance: 89.94 | Cargo: 133 | Route Time: 189.15 | Delay: 9.21
 Route #10: Feasible Distance: 87.89 | Cargo: 93 | Route Time: 176.46 | Delay: 18.58
 Route #11: Feasible Distance: 79.08 | Cargo: 127 | Route Time: 220.58 | Delay: 41.50
 Route #12: Feasible Distance: 52.47 | Cargo: 71 | Route Time: 118.16 | Delay: 5.69
 Route #13: Feasible Distance: 70.84 | Cargo: 190 | Route Time: 220.73 | Delay: 19.90

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

##Total Distance:1178.42

R204

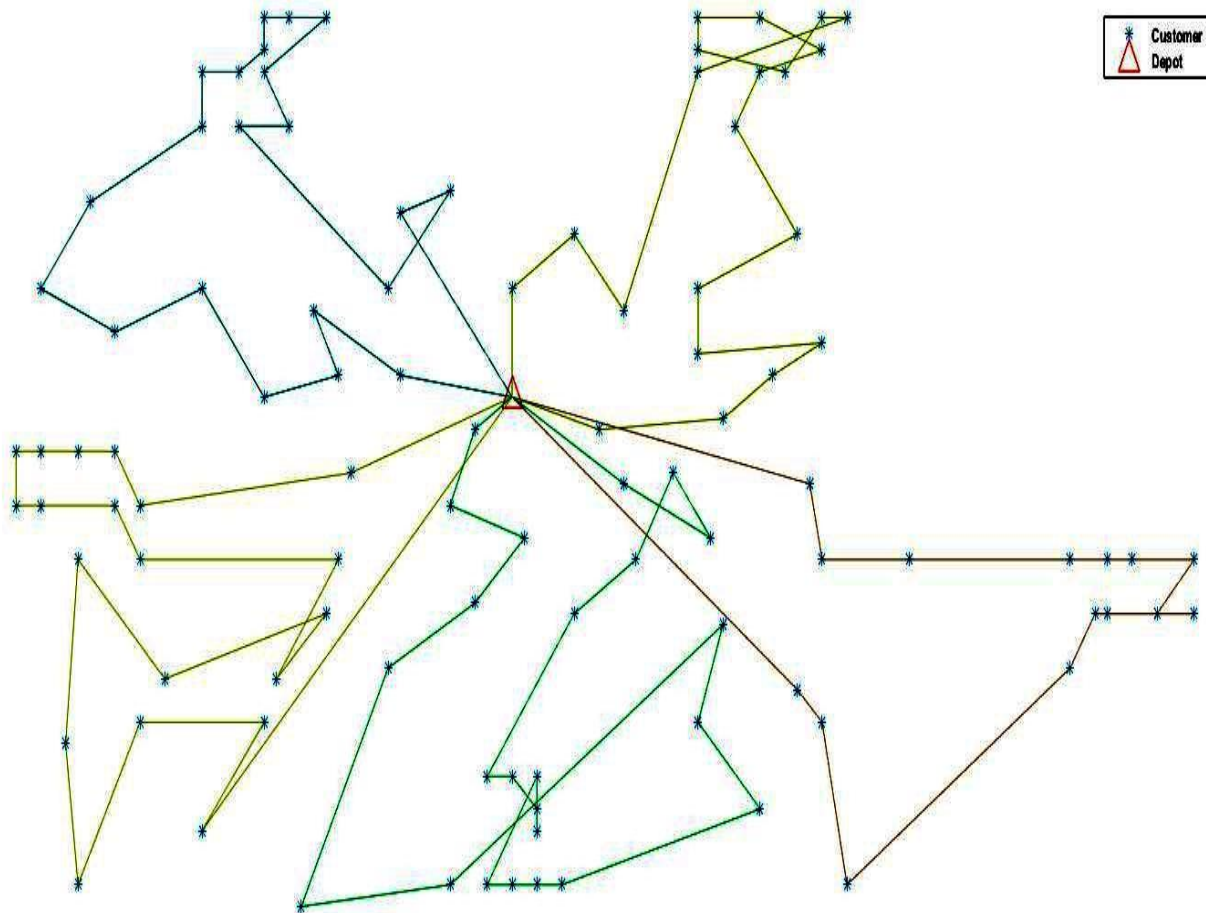


Σχήμα 25 Δρομολόγηση R204

Route #1: Feasible Distance: 188.55 | Cargo: 320 | Route Time: 846.70 | Delay: 438.15
 Route #2: Feasible Distance: 177.94 | Cargo: 274 | Route Time: 448.62 | Delay: 90.68
 Route #3: Feasible Distance: 168.50 | Cargo: 273 | Route Time: 926.11 | Delay: 577.61
 Route #4: Feasible Distance: 198.27 | Cargo: 353 | Route Time: 853.13 | Delay: 434.87
 Route #5: Feasible Distance: 101.38 | Cargo: 173 | Route Time: 868.06 | Delay: 636.68
 Route #6: Feasible Distance: 89.11 | Cargo: 65 | Route Time: 448.38 | Delay: 289.28

Route# 1: [101 1 70 30 20 66 65 71 35 34 29 80 68 76 50 33 81 51 9 78 79 3 77 101]
 Route# 2: [101 31 88 7 82 48 19 11 62 10 32 90 63 64 49 36 47 46 8 101]
 Route# 3: [101 40 21 73 72 39 23 67 56 75 22 74 4 55 25 24 54 26 12 101]
 Route# 4: [101 45 17 37 92 97 95 85 91 38 44 14 86 16 61 93 99 59 5 84 100 98 96 101]
 Route# 5: [101 27 69 28 53 52 18 83 60 89 6 94 13 58 101]
 Route# 6: [101 2 57 41 15 43 42 87 101]
 ##Total Distance:923.74

RC204



Σχήμα 26 Δρομολόγηση RC204

Route #1: Feasible Distance: 240.68 | Cargo: 366 | Route Time: 839.17 | Delay: 398.49

Route #2: Feasible Distance: 165.70 | Cargo: 225 | Route Time: 558.40 | Delay: 242.70

Route #3: Feasible Distance: 227.49 | Cargo: 419 | Route Time: 923.91 | Delay: 456.43

Route #4: Feasible Distance: 165.64 | Cargo: 343 | Route Time: 416.77 | Delay: 51.13

Route #5: Feasible Distance: 166.39 | Cargo: 371 | Route Time: 839.24 | Delay: 462.85

Route# 1: [101 82 10 12 14 47 17 16 15 11 9 99 86 52 87 13 97 75 59 74 58 101]

Route# 2: [101 67 62 50 34 31 29 27 28 26 30 32 33 89 63 85 101]

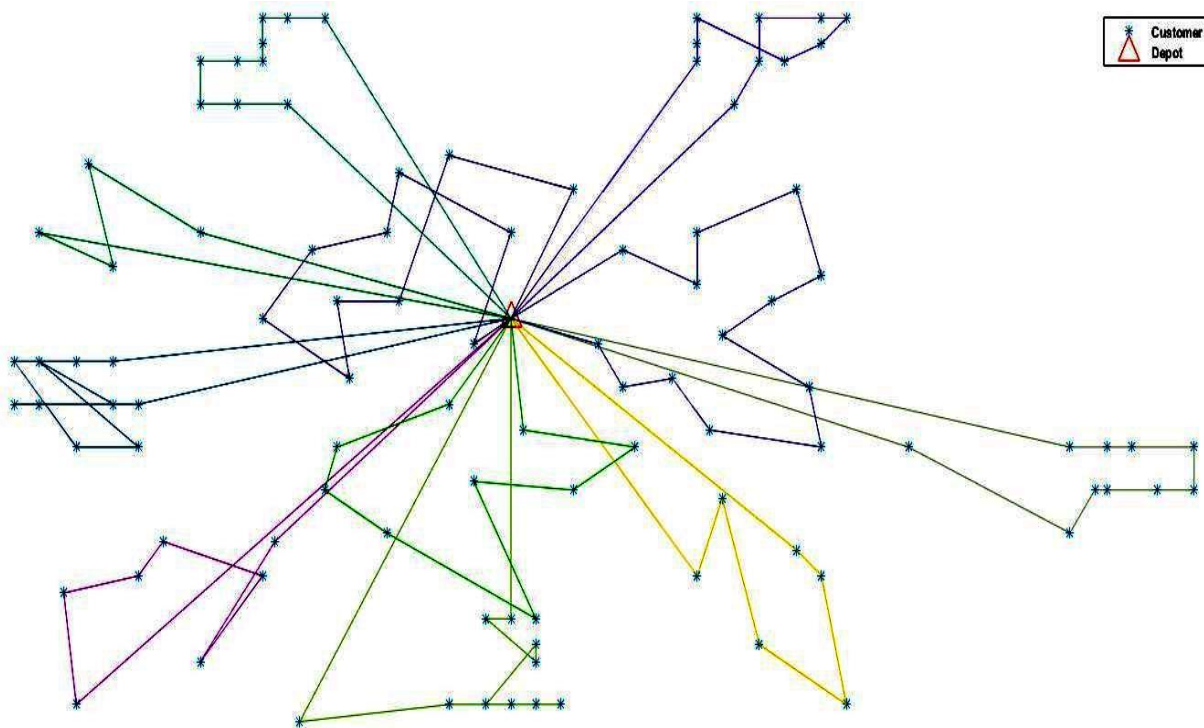
Route# 3: [101 91 95 92 56 64 24 22 49 19 20 23 21 48 18 76 51 84 25 77 57 83 66 65 90 101]

Route# 4: [101 80 94 93 71 96 54 72 41 39 37 40 43 44 38 36 35 42 81 61 68 101]

Route# 5: [101 69 88 98 53 60 78 73 79 7 8 46 45 5 3 1 4 2 6 55 70 100 101]

##Total Distance:965.88

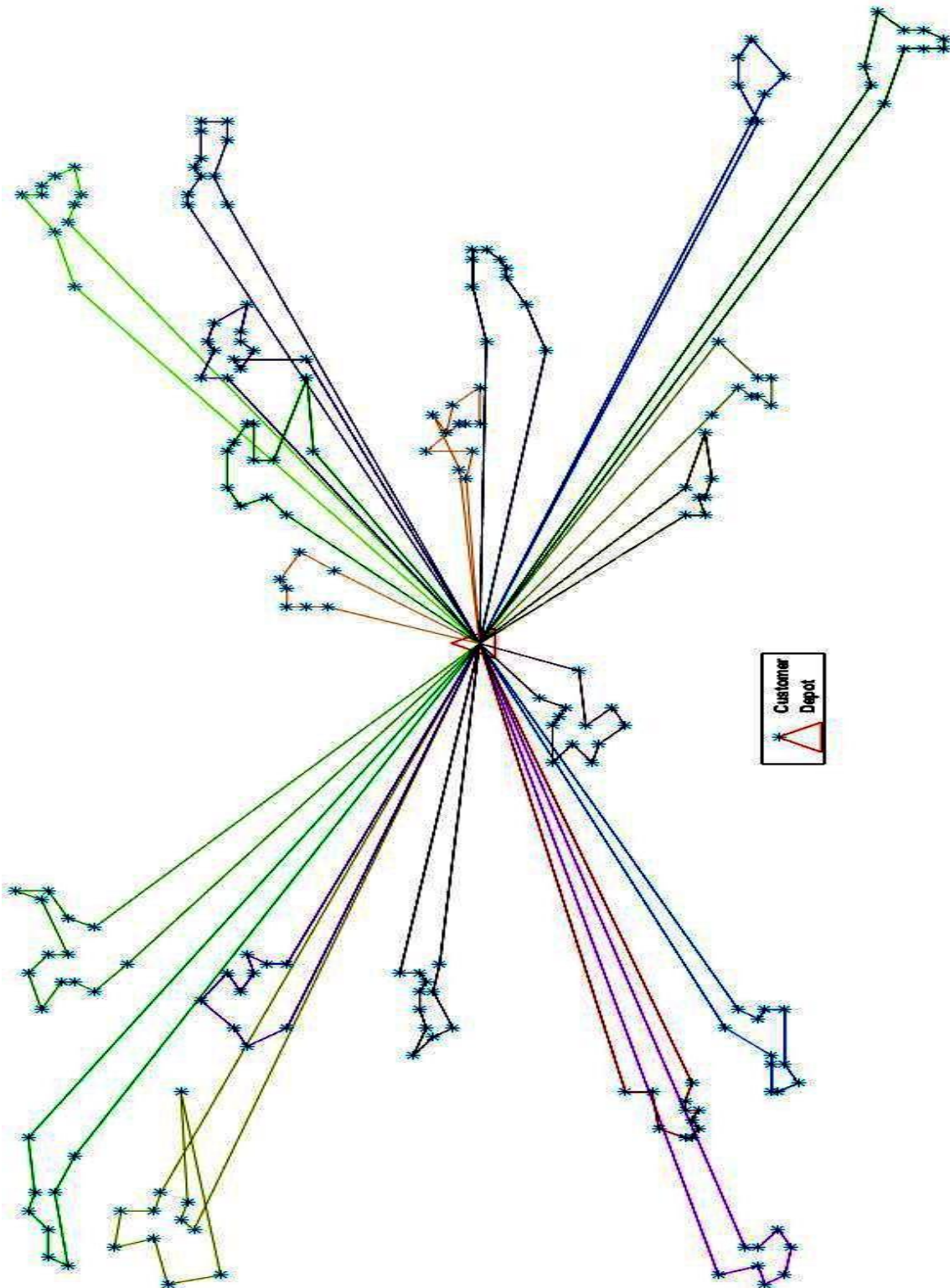
RC104



Σχήμα 27 Δρομολόγηση RC104

Route #1: Feasible Distance: 97.64 | Cargo: 79 | Route Time: 177.67 | Delay: 40.03
 Route #2: Feasible Distance: 127.56 | Cargo: 180 | Route Time: 227.56 | Delay: 0.00
 Route #3: Feasible Distance: 148.40 | Cargo: 139 | Route Time: 229.61 | Delay: 11.21
 Route #4: Feasible Distance: 124.13 | Cargo: 101 | Route Time: 184.13 | Delay: 0.00
 Route #5: Feasible Distance: 104.88 | Cargo: 158 | Route Time: 234.88 | Delay: 0.00
 Route #6: Feasible Distance: 130.61 | Cargo: 197 | Route Time: 236.15 | Delay: 5.53
 Route #7: Feasible Distance: 102.55 | Cargo: 200 | Route Time: 221.00 | Delay: 18.46
 Route #8: Feasible Distance: 126.81 | Cargo: 200 | Route Time: 233.57 | Delay: 6.76
 Route #9: Feasible Distance: 95.88 | Cargo: 190 | Route Time: 239.08 | Delay: 43.19
 Route #10: Feasible Distance: 103.42 | Cargo: 133 | Route Time: 238.94 | Delay: 45.52
 Route #11: Feasible Distance: 119.33 | Cargo: 147 | Route Time: 238.99 | Delay: 9.66

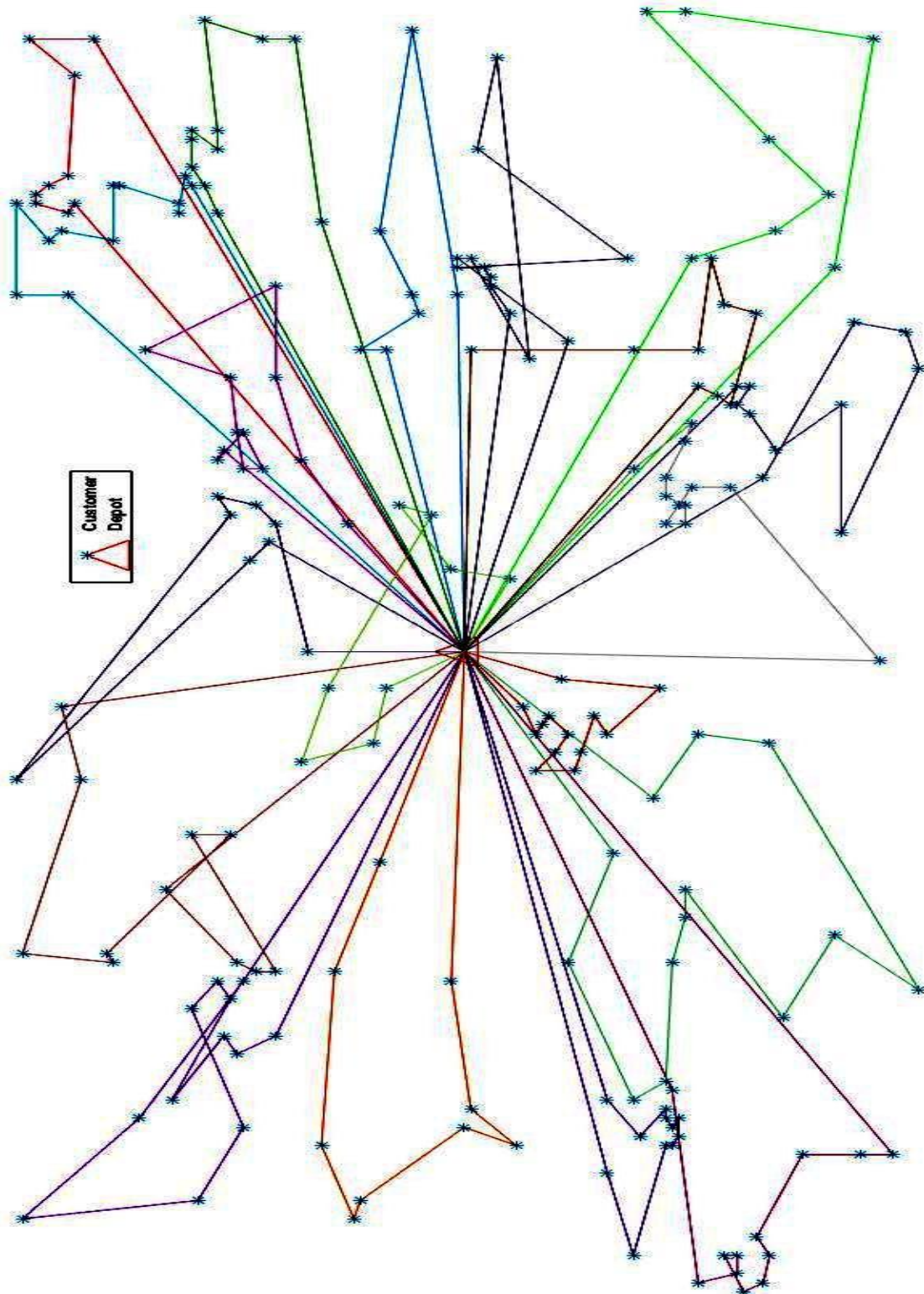
Route# 1: [101 60 79 78 73 101] Route# 7: [101 42 44 43 38 37 35 36 40 39 41 101]
 Route# 2: [101 34 31 29 27 26 28 30 32 33 50 101]
 Route# 3: [101 75 97 59 87 74 58 86 101]
 Route# 4: [101 85 63 89 76 84 51 101] Route# 11: [101 61 70 69 98 82 53 88 55 100 68 90 101]
 Route# 5: [101 80 91 92 95 62 67 94 93 71 72 54 96 81 101]
 Route# 6: [101 22 24 19 49 23 21 48 18 25 77 101] Route# 9: [101 2 6 7 8 46 4 45 5 3 1 101]
 Route# 8: [101 10 16 15 11 47 9 13 17 14 12 101]
 ##Total Distance:1281.21

C1_2_4

Route #1: Feasible Distance: 202.63 | Cargo: 180 | Route Time: 1102.63 | Delay: 0.00
 Route #2: Feasible Distance: 200.97 | Cargo: 160 | Route Time: 920.97 | Delay: 0.00
 Route #3: Feasible Distance: 178.92 | Cargo: 200 | Route Time: 1078.92 | Delay: 0.00
 Route #4: Feasible Distance: 222.23 | Cargo: 190 | Route Time: 1212.23 | Delay: 0.00
 Route #5: Feasible Distance: 178.08 | Cargo: 140 | Route Time: 898.08 | Delay: 0.00
 Route #6: Feasible Distance: 165.46 | Cargo: 140 | Route Time: 795.46 | Delay: 0.00
 Route #7: Feasible Distance: 180.29 | Cargo: 200 | Route Time: 1350.29 | Delay: 0.00
 Route #8: Feasible Distance: 130.53 | Cargo: 170 | Route Time: 1095.51 | Delay: 64.98
 Route #9: Feasible Distance: 151.07 | Cargo: 190 | Route Time: 1141.07 | Delay: 0.00
 Route #10: Feasible Distance: 144.02 | Cargo: 180 | Route Time: 1134.02 | Delay: 0.00
 Route #11: Feasible Distance: 134.13 | Cargo: 200 | Route Time: 1124.13 | Delay: 0.00
 Route #12: Feasible Distance: 126.54 | Cargo: 200 | Route Time: 1269.46 | Delay: 62.92
 Route #13: Feasible Distance: 114.53 | Cargo: 190 | Route Time: 883.48 | Delay: 48.95
 Route #14: Feasible Distance: 110.01 | Cargo: 190 | Route Time: 1190.01 | Delay: 0.00
 Route #15: Feasible Distance: 102.06 | Cargo: 170 | Route Time: 1092.06 | Delay: 0.00
 Route #16: Feasible Distance: 93.59 | Cargo: 200 | Route Time: 993.59 | Delay: 0.00
 Route #17: Feasible Distance: 64.37 | Cargo: 180 | Route Time: 1293.69 | Delay: 149.32
 Route #18: Feasible Distance: 65.80 | Cargo: 150 | Route Time: 695.80 | Delay: 0.00
 Route #19: Feasible Distance: 89.65 | Cargo: 120 | Route Time: 787.25 | Delay: 67.60
 Route #20: Feasible Distance: 78.01 | Cargo: 180 | Route Time: 1138.10 | Delay: 70.09

Route# 1: [201 32 171 65 86 115 94 51 174 136 189 201]
 Route# 2: [201 177 3 88 8 186 127 98 157 201]
 Route# 3: [201 21 23 182 75 163 194 145 195 52 92 201]
 Route# 4: [201 181 117 49 58 7 132 185 54 18 104 161 201]
 Route# 5: [201 60 82 180 84 191 125 4 72 201]
 Route# 6: [201 148 197 124 141 69 200 103 201]
 Route# 7: [201 114 159 38 22 150 151 16 140 187 142 111 63 56 201]
 Route# 8: [201 93 55 135 184 199 37 81 138 137 183 201]
 Route# 9: [201 170 134 50 156 112 168 79 29 87 42 123 201]
 Route# 10: [201 190 5 10 193 46 128 106 167 34 95 158 201]
 Route# 11: [201 57 118 83 143 176 36 33 121 165 188 108 201]
 Route# 12: [201 59 15 105 89 169 153 40 152 26 14 198 48 201]
 Route# 13: [201 70 47 160 147 66 164 91 12 201]
 Route# 14: [201 30 120 19 192 196 97 96 130 28 74 133 149 201]
 Route# 15: [201 101 144 119 166 35 126 71 9 1 99 53 201]
 Route# 16: [201 20 41 85 80 31 25 172 77 110 162 201]
 Route# 17: [201 113 175 13 43 2 90 67 17 39 78 155 107 201]
 Route# 18: [201 62 131 44 102 146 68 76 201]
 Route# 19: [201 73 116 129 11 6 122 139 201]
 Route# 20: [201 45 173 27 154 24 61 100 64 178 179 109 201]
 ##Total Distance:2732.90

RC1_2_4



Route #1:	Feasible Distance: 157.31		Cargo: 199		Route Time: 491.10		Delay: 223.79
Route #2:	Feasible Distance: 207.45		Cargo: 188		Route Time: 297.47		Delay: 0.02
Route #3:	Feasible Distance: 238.38		Cargo: 195		Route Time: 517.25		Delay: 158.87
Route #4:	Feasible Distance: 243.81		Cargo: 155		Route Time: 373.30		Delay: 49.49
Route #5:	Feasible Distance: 214.62		Cargo: 195		Route Time: 520.91		Delay: 186.30
Route #6:	Feasible Distance: 195.93		Cargo: 199		Route Time: 451.10		Delay: 135.17
Route #7:	Feasible Distance: 176.80		Cargo: 185		Route Time: 296.15		Delay: 9.35
Route #8:	Feasible Distance: 226.97		Cargo: 185		Route Time: 536.02		Delay: 179.05
Route #9:	Feasible Distance: 217.12		Cargo: 190		Route Time: 369.09		Delay: 31.98
Route #10:	Feasible Distance: 194.90		Cargo: 195		Route Time: 390.79		Delay: 85.89
Route #11:	Feasible Distance: 228.78		Cargo: 198		Route Time: 510.33		Delay: 171.55
Route #12:	Feasible Distance: 174.50		Cargo: 153		Route Time: 419.18		Delay: 164.69
Route #13:	Feasible Distance: 151.21		Cargo: 194		Route Time: 371.11		Delay: 149.91
Route #14:	Feasible Distance: 155.66		Cargo: 137		Route Time: 595.06		Delay: 349.40
Route #15:	Feasible Distance: 165.99		Cargo: 200		Route Time: 285.99		Delay: 0.00
Route #16:	Feasible Distance: 154.75		Cargo: 192		Route Time: 395.55		Delay: 120.80
Route #17:	Feasible Distance: 151.09		Cargo: 198		Route Time: 241.09		Delay: 0.00
Route #18:	Feasible Distance: 82.59		Cargo: 200		Route Time: 212.59		Delay: 0.00
Route #19:	Feasible Distance: 99.35		Cargo: 200		Route Time: 179.35		Delay: 0.00

```

Route# 1: [201 42 197 138 46 159 20 26 62 5 141 185 201]
Route# 2: [201 109 189 99 94 151 160 192 53 101 201]
Route# 3: [201 84 199 127 118 176 69 105 13 75 123 137 113 201]
Route# 4: [201 148 111 198 61 91 145 51 167 201]
Route# 5: [201 186 181 35 77 122 164 104 25 10 193 19 170 201]
Route# 6: [201 45 56 165 85 50 112 157 125 128 168 65 177 201]
Route# 7: [201 86 114 108 38 32 117 23 74 52 31 55 201]
Route# 8: [201 162 96 172 135 120 149 63 73 93 146 200 39 79 201]
Route# 9: [201 169 147 40 72 70 22 126 133 16 98 119 161 201]
Route# 10: [201 18 68 179 7 110 44 90 83 95 116 178 201]
Route# 11: [201 88 191 58 134 87 17 11 71 144 121 155 201]
Route# 12: [201 54 100 37 12 9 115 66 89 201]
Route# 13: [201 57 106 129 41 195 194 158 201]
Route# 14: [201 174 97 8 60 1 76 2 34 175 201]
Route# 15: [201 24 102 156 131 48 43 3 136 180 124 36 103 201]
Route# 16: [201 143 59 29 21 139 188 173 33 81 107 166 82 201]
Route# 17: [201 182 152 184 27 190 80 15 183 196 201]
Route# 18: [201 150 187 163 28 6 171 130 78 132 49 14 47 153 201]
Route# 19: [201 142 4 154 92 140 30 64 67 201]
##Total Distance:3437.21

```

5.3 Συμπεράσματα

Κλείνοντας αυτή τη μελέτη θα ήτο παραγωγικόν να σχολιαστούν τα αποτελέσματα του προτεινόμενου αλγορίθμου, καθώς αυτά δεν είναι τα καλύτερα δυνατά. Εν γένει στο δείγμα των 25 προβλημάτων της βιβλιογραφίας που επιλέχθηκε (πίνακας 14) επιτυγχάνεται μέσο σφάλμα 7,65 % στη διανυθείσα απόσταση των οχημάτων και ταυτόχρονα μέσο στόλο 1,44 οχήματα περισσότερα. Συγκεκριμένα ο βέλτιστος στόλος είναι 235 οχήματα και ο προτεινόμενος αλγόριθμος δίνει 271 ήτοι 36 περισσότερα οχήματα. Αναλυτικότερα για την απόσταση η συνολική φτάνει τις 30.377,54 μον. μήκους ενώ ο αλγόριθμος διανύει 32.869,62 ήτοι 2 492,08 μον.μήκους περισσότερες. Αυτά είναι τα στοιχεία που δίνονται από τη βιβλιογραφία. Αναλυτικότερα για κάθε κατηγορία :

- Είναι προφανές ότι για τη κλάση C1 εις όλα τα προβλήματα επετεύχθη εντυπωσιακό αποτέλεσμα εις όλα τα μέτρα απόδοσης (διανυθείσα απόσταση, πλήθος οχημάτων, χρόνος εκτέλεσης), συγκεκριμένα το σφάλμα για τα προβλήματα C101 – C109 ανήλθε μόλις στο 0,12%, ο παραχθείς στόλος είναι ακριβώς ο βέλτιστος και ο μέσος χρόνος εκτέλεσης εντυπωσιακός μόλις στα 2,5 δευτερόλεπτα. Αυτό ήταν αναμενόμενο καθώς η αρχική λύση που δομεί ο 3-PCC αλγόριθμος ταυτίζεται ακριβώς με τη βέλτιστη λύση, το μόνο που χρειάστηκε να πράξει περαιτέρω ο GIRSOT είναι να εύρει τα βέλτιστα δρομολόγια χρήσει m – TSP.
- Η κλάση C2 εμπεριέχει οχήματα μεγάλης χωρητικότητας, συνέπεια ανάθεσης πολλών σχετικά πελατών ανά δρομολόγιο αυτό δημιουργεί οργανικό πρόβλημα στον συνολικό αλγόριθμο καθώς ο TSP όπως αναλύθηκε και στο 3.2 αδυναμεί εις το να εύρει εφικτές λύσεις ειδικά σε στενά χρονικά παράθυρα, συνέπεια αυτού είναι ο GIRSOT να αναζητάει εφικτή λύση με περισσότερα οχήματα γεγονός που μειώνει πολύ την απόδοση του. Έτσι στα δυο επιλεχθέντα C204, C208 επιτυγχάνεται σφάλμα 11,46% και στόλος υποβέλτιστος κατά 2 οχήματα. Στο ίδιο μήκος κύματος κινούνται και τα προβλήματα της κλάσης R2.
- Κατερχόμενοι του πίνακος 14 συναντώνται τα 2 προβλήματα της κλάσης R1, τα οχήματα είναι μικρά (200 μονάδες) και απαιτείται μεγαλύτερος στόλος. Λόγω της τυχαιότητας διασποράς των θέσεων των πελατών ο 3-PCC δεν είναι τόσο αποδοτικός. Συνεπεία αυτού ο group relocate θα πρέπει να πράξει μια πιο περίπλοκη διαδικασία αναζήτησης δια να εύρη τις κατάλληλες μεταθέσεις.
- Στη κατηγορία RC1 τα αποτελέσματα είναι κάπως καλύτερα με σφάλμα στο 9% και 2 οχήματα περισσότερα, αναμενόμενο καθώς αυτή η κατηγορία κυμαίνεται κάπου

ανάμεσα στην C1 και R1. Ταυτόχρονα ο χρόνος εκτέλεσης εντυπωσιάζει και εδώ αφού είναι μόλις στα 3,6 δευτερόλεπτα.

- Εις ότι αφορά τα 5 προβλήματα των Gehring & Homberger's που επιλέχθηκαν ανήκουν αναλογικά τα τέσσερα από αυτά στη κατηγορία C1 και εδώ φαίνεται η πιστότητα του αλγορίθμου για αυτή τη κατηγορία προβλημάτων αφού το σφάλμα στα C1_2_3, 4, 9, 10 είναι μόλις 2,64% και ο μέσος στόλος απέχει 2,25 οχήματα απ' το βέλτιστο, το πρόβλημα RC1_2_4 παρόμοιο με το RC104 φέρει αρκετά κακή ποιότητα λύσης με 16% σφάλμα. Τα πράγματα όμως είναι καλύτερα στο στόλο καθώς αυτός ξεπερνάει το βέλτιστο μόλις κατά 1 όχημα. Ο μέσος χρόνος εκτέλεσης αυτών των 5 προβλημάτων είναι πάρα πολύ ικανοποιητικός για κλίμακας 200 πελατών στα 1,4 λεπτά.

Περαιτέρω θα μπορούσε κάποιος να λάβει σημαντικές πληροφορίες από τους χρόνους αναμονής (delays) συνοδεύοντας τις εικονογραφήσεις. Τα συγκεκριμένα προβλήματα της βιβλιογραφίας αναφέρονται σε συγκεκριμένες περιπτώσεις δρομολογήσεων όπου οι λύσεις βγαίνουν βέλτιστες με ελάχιστα ή και ανύπαρκτα σε πολλές περιπτώσεις delays. Η ανωτέρω πρόταση δεν ανταποκρίνεται προφανώς στη πραγματικότητα, εκτός και αν τύχει κάτι τέτοιο να συμβαίνει. Γενική αλήθεια είναι πως όσο μικρότερα είναι αυτά τα delays τόσο πιο κοντά στη βέλτιστη θα βρίσκεται η δρομολόγηση. Πολλές φορές συμβαίνει ένας και μόνον πελάτης να δημιουργεί πρόβλημα χρονοπρογραμματισμού σ' όλο το δρομολόγιο, διότι η εξυπηρέτηση του μπορεί είτε να απαιτεί ένα ακόμα όχημα είτε να διανυθούν αρκετά περισσότερα χιλιόμετρα. Οι πληροφορίες λοιπόν που παρέχει αυτό το λογισμικό θα μπορούσαν να βοηθήσουν τον υπεύθυνο δρομολόγησης στο να «χαλαρώσει» τα παράθυρα κάποιων πελατών και έτσι να μειώσει και το κόστος διανομής. Γίνεται σ' αυτό το σημείο εμφανές ότι όσο τέλειο και αν είναι ένα λογισμικό ποτέ δεν μπορεί να υποκαταστήσει το ανθρώπινο δυναμικό, δύναται όμως να παράσχει υποστήριξη εις τα κέντα αποφάσεων.

Παράρτημα

1. Πρόγραμμα εικονογράφησης Δρομολογήσεων σε MatLab

Ο παρακάτω κώδικας :

```
clear all
close all
newplot;
%Συντεταγμένες αποθήκης
dx=40;
dy=50;
r_num=10;%Πλήθος δρομολογίων
max_cust=13;%Μέγιστο πλήθος πελατών στο μητρώο δρομολογίων
fid1=fopen('x_cord.txt','r');
x=fscanf(fid1, '%g\n',[1 inf]);
fclose(fid1);
fid2=fopen('y_cord.txt','r');
y=fscanf(fid2, '%g\n',[1 inf]);
fclose(fid2);
fid3=fopen('Routes.txt','r');
Routes=fscanf(fid3, '%g\n',[1 inf]);
fclose(fid3);
Routes=reshape(Routes,max_cust,r_num);
Routes=Routes';
n = length(x);
[blin, bcol] = size(Routes);
c=1:n;%Σειριακός αριθμός πελατών από 1 έως n
c=c';
c=int2str(c);
hold all
plot(x,y, '*',dx,dy, 'r^');
%text(x,y,c,'fontsize',10);
legend('Customer','Depot');
grid on;
for i=1:blin,
    color = [rand, rand, rand];
    plot([dx,x(Routes(i,1))],[dy,y(Routes(i,1))],'color',color);
    j=1;
    while (j<bcol) && (Routes(i,j+1)~=0),
        plot([x(Routes(i,j)),x(Routes(i,j+1))],[y(Routes(i,j)),y(Routes(i,j+1))],'color',color);
        j = j+1;
    end
    plot([x(Routes(i,j)),dx],[y(Routes(i,j)),dy],'color',color);
end
axis([min(x)-2 max(x)+2 min(y)-2 max(y)+2]);
```

Οδηγίες χρήσης:

- Δέχεται 3 αρχεία εισόδου σε μορφή(format) με κατάληξη “.txt”
- Τα αρχεία δεδομένων πρέπει να ευρίσκονται εις τον ίδιο φάκελο με το αρχείο του κώδικος
- Τα δυο πρώτα αρχεία είναι τη τετμημένη “x.txt” και τεταγμένη “y.txt” αντίστοιχα
- Το τρίτο αρχείο ονόματι “Routes.txt” πρέπει να είναι ένας τετραγωνικός πίνακας με γραμμές όσα και τα δρομολόγια και στήλες όσες και το μέγιστο πλήθος πελατών (μεγαλύτερο δρομολόγιο) τα κενά στις γραμμές με τα υπόλοιπα δρομολόγια θα πρέπει να είναι συμπληρωμένα με το 0.

Παράδειγμα αρχείου δεδομένων Routes.txt με 10 δρομολόγια και μέγιστο πλήθος πελατών 13

90	87	86	83	82	84	85	88	89	91	0	0	0
81	78	76	71	70	73	77	79	80	0	0	0	0
57	55	54	53	56	58	60	59	0	0	0	0	0
98	96	95	94	92	93	97	100	99	0	0	0	0
34	36	39	38	37	35	31	33	32	0	0	0	0
13	15	12	14	16	19	18	17	0	0	0	0	0
43	42	41	40	44	45	46	48	51	50	52	49	47
67	65	63	62	74	72	61	64	68	66	69	0	0
20	24	25	27	29	30	28	26	23	22	21	0	0
5	3	7	8	10	11	9	6	4	2	1	75	0

Ευρετήριο

Τα νούμερα που αναγράφονται είναι οι σελίδες

Σχήματα

Κεφάλαιο 1

Σχ.1 Διεργασίες logistics 8
Σχ.2 Παράδειγμα ευκλείδειας – πραγματικής απόστασης 13

Κεφάλαιο 2

Σχ.3 Στάδια κατασκευής αρχικής λύσης 18
Σχ.4 Μέθοδος 3-PCC ,στάδια φάσης construction 19
Σχ.5 Διάγραμμα ροής φάσης construction 21
Σχ.6 Διάγραμμα ροής φάσης compression 24
Σχ.7 Μέθοδος 3-PCC , το πριν και το μετά της φάσης adjustment 25
Σχ.8 Διάγραμμα ροής ροής φάσης adjustment 26
Σχ.9 Βαθμός απόδοσης INEF συναρτήσει στόλου για δείγμα Solomon n100 29
Σχ.10 Δυσικότητα INEF συναρτήσει στόλου 30

Κεφάλαιο 3

Σχ.11 Συμβολισμοί τοπογραφίας 33
Σχ.12 Τοπική αναζήτηση με ή χωρίς tabu 33
Σχ.13 Γραμμική στρατηγική μετασχηματισμού 2-opt 34
Σχ.14 Παράδειγμα κίνησης 2-opt 34
Σχ.15 Παράδειγμα κίνησης swap 35
Σχ.16 Δομή δεδομένων short – term μνήμης 36
Σχ.17 Δομή δεδομένων μνήμης συχνοτήτων 37
Σχ.18 Προσαρμοστική μνήμη 38
Σχ.19 Διάγραμμα ροής τοπικής αναζήτησης 41
Σχ.20 Μετά – τοπική αναζήτηση 42
Σχ.21 Διάγραμμα ροής μετά – τοπικής αναζήτησης 43
Σχ.22 Διαδικασία διασταύρωσης γονέων 45
Σχ.23 Διάγραμμα ροής mating – mutation 47

Κεφάλαιο 4

Πίνακες

Κεφάλαιο 2

Πιν.1 Δρομολόγηση φάσης construction 20
Πιν.2 Ψευδοκώδικας φάσης construction 20
Πιν.3 Παράδειγμα επιλογής MDF 23
Πιν.4 Ψευδοκώδικας φάσης compression 23
Πιν.5 Δρομολόγηση φάσης adjustment 25
Πιν.6 Ψευδοκώδικας φάσης adjustment 25
Πιν.7 Σύγκριση αποτελεσμάτων 3-PCC βιβλιογραφίας 28

Κεφάλαιο 3

Πιν.8 short – term μνήμη παραδείγματος
Πιν.9 Ψευδοκώδικας τοπικής αναζήτησης 40
Πιν.10 Ψευδοκώδικας μετά – τοπικής αναζήτησης 43
Πιν.11 Ψευδοκώδικας mating – mutation

Κεφάλαιο 4

Πιν.12 Ψευδοκώδικας αλγορίθμου σποράς 51
Πιν.13 Αλγόριθμος χρονικής υποτίμησης 54
Πιν.14 Ψευδοκώδικας GIRSOT 56

Κεφάλαιο 5

Πιν.15 Αποτελέσματα GIRSOT 62

Σχ.24	Αλγόριθμος σποράς	50
Σχ.25	Διάγραμμα ροής αλγορίθμου σποράς	52
Σχ.19	Διάγραμμα ροής αλγορίθμου χρονικής υποτίμησης	54
Σχ.20	Μέγεθος στόλου 3-PCC συναρτήσει route time	55
Σχ.21	Διάγραμμα ροής αλγορίθμου GIRSOT	57

Κεφάλαιο 5

Σχ.22	Δρομολόγηση C104	64
Σχ.23	Δρομολόγηση C204	65
Σχ.24	Δρομολόγηση R104	66
Σχ.25	Δρομολόγηση R204	67
Σχ.26	Δρομολόγηση RC204	68
Σχ.27	Δρομολόγηση RC104	69
Σχ.28.	Δρομολόγηση C1_2_4	70
Σχ.29	Δρομολόγηση RC1_2_4	72

Βιβλιογραφία

- [1] Abrie de Feitos Jonathan, (2012). *A Tabu search metaheuristic algorithm for the multiple depot vehicle routing problem with time windows*. Faculty of Engineering, build environment and informationTechnology.
- [2] Basu Sumanta, (2012). *Tabu search implementation on Traveling salesman problem and its variations: A literature story*. American Journal of Operations Research, 2, 163 – 173.
- [3] Chase Richard B., & Jacobs Robert F., (2012). *Operations and supply management – Διοίκηση λειτουργιών εφοδιαστικής αλυσίδας*. Εκδόσεις Π. Χ. Πασχαλίδης.
- [4] Chard Hart, (2013). *Graph theory topics in computer networking*. In Partial Fullfillment of Math 4395.
- [5] Fermin Alfredo Tang Montone, & Garvao Dieguez Roberto, (2006). *A tabu search algorithm for vehicle routing problem with simultaneous pick-up and delivery service*. Computers & Operations Research, 33, 595 – 619.
- [6] Hao Tang, & Miller – Hooks Elise, (2005). *A tabu search heuristic for the team orienteering problem*. Computers & Operations Research, 33, 1379 – 1407.
- [7] Laporte Gilbert, (1992). *The vehicle routing problem An overview of exctact and approximate algorithms*. European Journal of Operational Research, 59, 345 – 358.
- [8] Kleinberg Jon, & Tardos Eva, (2008). *Algorithm design – Σχεδιασμός Αλγορίθμων*. Εκδόσεις Κλειμάριθμος.
- [9] Kwangcheal Shin, & Sangyong Ham, (2011). *A centroid – based heuristic algorithm for the capacitated vehicle routing problem*. Computins and Informatics, 30, 721 – 732.
- [10] Savitch Walter, (2012). *Absolute Java*. PERSON.
- [11] Wang Hsiao – Fan, & Chen Ying – Yen, (2012). *A genetic algorithm for the simultaneous delivery and pick – up problems with time windows*. Computers & Industrial Engineering, 62, 84 – 95.

Βιβλιογραφία

- [12] Ακριβής Γ. Δ., & Δούγαλης Β. Α., (2010). *Εισαγωγή στην αριθμητική ανάλυση*. Πανεπιστημιακές εκδόσεις Κρήτης.
- [13] Γκόγκος Χρήστος Γ., (2009). *Αλγόριθμοι Συνδυαστικής Βελτιστοποίησης σε μεταερευνητικές τεχνικές*. Πανεπιστήμιο Πατρών, αρ. διατριβής – 255.
- [14] Δημητρουλάς Διονύσιος, (2012). *Επίλυση του προβλήματος δρομολόγησης οχημάτων με χρονικά παράθυρα μέσω μιμητικού αλγορίθμου*. Μεταπτυχιακή διατριβή.
- [15] Μαργαρίτης Κωνσταντίνος, (2004). *Εισαγωγή στη Java*. GNU FDL.
- [16] Μυγδαλάς Αθανάσιος, & Ιωάννης Μαρινάκης, (2009). *Σημειώσεις Συνδυαστικής Βελτιστοποίησης*.
- [17] Μυγδαλάς Αθανάσιος, & Ιωάννης Μαρινάκης, (2008). *Σχεδιασμός και Βελτιστοποίηση της Εφοδιαστικής Αλυσίδας*. Εκδόσεις σοφία.
- [18] <https://www.sintef.no/projectweb/top/vrptw/homberger-benchmark/200-customers/>
- [19] <http://web.cba.neu.edu/~msolomon/problems.html>
- [20] <https://books.google.gr/books?id=qKJSqw6J7AgC&pg=PA314&lpg=PA314&dq=gardener%27s+algorithm&source=bl&ots=eRssZSGVO4&sig=vOZWapA-X0EHtOpoEkWp8zKHGw8&hl=el&sa=X&ved=0ahUKEwjIjafw1uXKAhVBBoKHfN-AesQ6AEIHTAA#v=onepage&q=gardener's%20algorithm&f=false>
- [21] <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/STSP.html>
- [22] https://en.wikipedia.org/wiki/Vehicle_routing_problem
- [23] <http://neo.lcc.uma.es/vrp/vrp-flavors/vrp-with-time-windows/>
- [24] <http://sun.aei.polsl.pl/~zjc/best-solutions-solomon.html#C101>