



**ΠΟΛΥΤΕΧΝΕΙΟ
ΚΡΗΤΗΣ**

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ
ΠΑΡΑΓΩΓΗΣ ΚΑΙ
ΔΙΟΙΚΗΣΗΣ**

**ΠΡΟΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
ΓΕΩΡΓΙΟΥ ΓΕΩΡΓΙΑ**

**ΔΡΟΜΟΛΟΓΗΣΗ ΟΧΗΜΑΤΩΝ ΜΕ ΠΑΡΑΛΑΒΗ ΚΑΙ
ΔΙΑΝΟΜΗ ΠΡΟΪΟΝΤΩΝ ΤΑΥΤΟΧΡΟΝΑ ΜΕ
ΔΕΔΟΜΕΝΑ ΧΡΟΝΙΚΑ ΠΑΡΑΘΥΡΑ (VEHICLE
ROOTING PROBLEM WITH PICKUP AND
DELIVERIES AND TIME WINDOWS)**

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ
ΜΑΡΙΝΑΚΗΣ ΙΩΑΝΝΗΣ**

Χανιά 2020

ΕΥΧΑΡΙΣΤΙΕΣ

Αρχικά, θα ήθελα να ευχαριστήσω τον καθηγητή μου, κ. Μαρινάκη, για την πολύτιμη βοήθειά του όλο αυτό το διάστημα. Άμεση σειρά προτεραιότητας έχουν οι φίλοι και η οικογένειά μου που με τον τρόπο τους με στήριξαν και ήταν πάντα δίπλα μου. Ευχαριστώ όλους τους καθηγητές του τμήματός μου για την καθοδήγηση, την γνώση και την επιμονή τους καθ' όλη τη διάρκεια της φοίτησής μου. Τέλος, θα ήθελα να ευχαριστήσω όλους τους συμφοιτητές, καθηγητές και κάθε μέλος του ιδρύματος για την όμορφη συνεργασία και συνύπαρξη στον χώρο του Πολυτεχνείου Κρήτης

Πίνακας περιεχομένων

| | |
|---|----|
| Κεφάλαιο 1ο | 5 |
| 1.1 Εισαγωγή | 5 |
| 1.1.1 Η Εφοδιαστική Αλυσίδα (SupplyChain)..... | 6 |
| 1.1.2 Logistics..... | 7 |
| Βασικές Αρχές των Logistics..... | 8 |
| Οι Δραστηριότητες των Logistics..... | 9 |
| Κεφάλαιο 2ο | 13 |
| 2.1 Εισαγωγή | 13 |
| 2.2 Το πρόβλημα της Δρομολόγησης Οχημάτων (VRP)..... | 13 |
| 2.3 Κατηγορίες Προβλημάτων Δρομολόγησης | 19 |
| 2.3.1 Το πρόβλημα του Πλανόδιου Πωλητή (TSP) (3) | 19 |
| 2.3.2 Πρόβλημα Δρομολόγησης Οχημάτων με Συγκεκριμένο Όριο Χωρητικότητας (Capacitated Vehicle Routing Problem – CVRP)..... | 20 |
| 2.3.3 Πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα (Vehicle Routing Problem with Time Windows – VRPTW)..... | 21 |
| 2.3.4 Πρόβλημα Δρομολόγησης Οχημάτων με Backhauls (Vehicle Routing Problem with Backhauls – VRPB)..... | 22 |
| 2.3.5 Πρόβλημα Δρομολόγησης Οχημάτων Πολλαπλών Αποθηκών (Vehicle Routing Problem with Multiple Depot – MDVRP)..... | 23 |
| 2.3.6 Πρόβλημα Δρομολόγησης Οχημάτων Περιόδου (Periodic Vehicle Routing Problem– PVRP)24 | |
| 2.3.7 Προβλήματα Δρομολόγησης Οχημάτων με Διασπαρμένες Παραδόσεις (Split Delivery Vehicle Routing Problem – SDVRP)..... | 25 |
| 2.3.8 Λοιπά Προβλήματα Δρομολόγησης (1)..... | 25 |
| Κεφάλαιο 3ο | 27 |
| 3.1 Σπουδαιότητα μελέτης του προβλήματος δρομολόγησης οχημάτων VRP (VehicleRoutingProblems) | 27 |
| 3.2 Πρόβλημα δρομολόγησης οχημάτων VRP (VehicleRoutingProblems) | 27 |
| 3.2.1 Πρόβλημα Δρομολόγησης Οχημάτων με Διανομή και Παραλαβή κατά τη Διάρκεια της Διαδρομής (Vehicle Routing Problem with Pickup and Delivery – VRPPD)..... | 28 |
| 3.2.2 Το μικτό πρόβλημα δρομολόγησης οχημάτων με δύο είδη πελατών κατά τη διάρκεια της διαδρομής (TheMixedVehicleRoutingProblemwithBackhauls (MVRPB)..... | 30 |
| 3.2.3 Πρόβλημα δρομολόγησης οχημάτων με χρονικά περιθώρια (VehicleRoutingProblemwithTimeWindows) | 31 |

| | |
|---|----|
| 3.2.4 Το μικτό πρόβλημα δρομολόγησης οχημάτων με δύο είδη πελατών κατά τη διάρκεια της διαδρομής και πολλές αποθήκες (TheMultipleDepotMixedVehicleRoutingProblemwithbackhauls - MDMVRPB)..... | 36 |
| Κεφάλαιο 4ο | 37 |
| 4.1 Αλγόριθμοι και το πρόβλημα δρομολόγησης οχημάτων | 37 |
| Ευρετικοί αλγόριθμοι..... | 40 |
| 4.1.1α Το πρόβλημα δρομολόγησης οχημάτων – Ο αλγόριθμος των εξοικονομήσεων των Clarke&Wright (Thesavingsalgorithm) | 42 |
| 4.1.1β Το πρόβλημα δρομολόγησης οχημάτων – Μέθοδοι δύο φάσεων..... | 43 |
| 4.1.1γ Η μέθοδος των δύο φάσεων των Fisher&Jaikumar..... | 44 |
| 4.1.1δ Η μέθοδος των δύο φάσεων των Mole&Jameson..... | 45 |
| 4.1.1ε Η μέθοδος των δύο φάσεων των Christofides, Mignozzi&Toth..... | 46 |
| 4.1.2 Αλγόριθμοι τοπικής αναζήτησης (Localsearchalgorithms) | 47 |
| 4.1.2α Το πρόβλημα δρομολόγησης οχημάτων και οι αλγόριθμοι τοπικής αναζήτησης | 49 |
| 4.1.3 Διαδικασία άπληστης Τυχοποιημένης Προσαρμοστικής Αναζήτησης – Μεθευρετικοί Αλγόριθμοι..... | 50 |
| 4.1.3.α Μεθευρετικοί Αλγόριθμοι και το πρόβλημα δρομολόγησης οχημάτων..... | 51 |
| 4.2 Αλγόριθμοι επίλυσης | 54 |
| 4.2.1 Αλγόριθμος Περιορισμένης Αναζήτησης (TabuSearch – TS) (4) | 54 |
| 4.2.2 Αλγόριθμος εμπνευσμένος από την φύση για ad-hoc δίκτυα και clouds(AntColonyOptimization - ACO) (6) | 60 |
| 4.2.3 Ο Αλγόριθμός Ant Colony System (ACS) (7)..... | 63 |
| 4.2.4 Ο Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων (ParticleSwarmOptimization(PSO) (8) | 65 |
| 4.2.5 Ο Αλγόριθμος Deathloff (9) | 70 |
| 4.2.6 Ο Αλγόριθμος AMM (10)..... | 71 |
| 4.2.7 Ο αλγόριθμος πλησιέστερου γείτονα (Largeneighborhoodsearch(LNS)) (11)..... | 72 |
| 4.2.8 Ο Αλγόριθμος TG (12) | 73 |
| 4.2.9 Ο Αλγόριθμος GTS (13) | 75 |
| ΚΕΦΑΛΑΙΟ 5 | 77 |
| Αποτελέσματα..... | 77 |
| Βιβλιογραφία | 79 |

Κεφάλαιο 1ο

1.1 Εισαγωγή

Το παρόν κεφάλαιο είναι το εισαγωγικό της παρούσας διπλωματικής εργασίας και για αυτό το λόγο στη συνέχεια παρουσιάζονται οι έννοιες της εφοδιαστικής αλυσίδας και των Logistics καθώς επίσης και οι βασικές αρχές και δραστηριότητές τους. Σκοπός είναι η κατανόηση των εννοιών αυτών, καθώς στη συνέχεια να αναλυθούν εκτενέστερα αλγόριθμοι που σχετίζονται με το Πρόβλημα της Δρομολόγησης Οχημάτων, που αποτελεί σημαντική δραστηριότητα των Logistics.

Το Πρόβλημα της Δρομολόγησης Οχημάτων (VRP – VehicleRoutingProblem) παρουσιάστηκε για πρώτη φορά από τους Dantzing&Ramser το 1959, ενώ σήμερα αποτελεί ένα από τα πιο σημαντικά και εφαρμόσιμα προβλήματα διανομής της εφοδιαστικής αλυσίδας. Οι Dantzing&Ramser δημιούργησαν την πρώτη αλγοριθμική προσέγγιση επίλυσης τέτοιων προβλημάτων και την εφάρμοσαν για την διανομή γκαζολίνης σε έναν αριθμό σταθμών τροφοδοσίας. Στη συνέχεια, το έτος 1964, οι Clarke&Wright πρότειναν έναν ευρετικό αλγόριθμο εξοικονόμησης, ο οποίος παρουσιάζει σημαντικές βελτιώσεις σε σχέση με αυτόν των Dantzing&Ramser.

Πρόκειται για ένα πρόβλημα διανομής, το οποίο όταν λυθεί θα πρέπει να επιφέρει όφελος για μια επιχείρηση μέσα από την εξοικονόμηση κόστους. Με σκοπό την επίλυση τέτοιου είδους προβλημάτων, θα πρέπει αρχικά να υπολογιστούν οι βέλτιστες διαδρομές, τις οποίες θα εκτελέσει ένας συγκεκριμένος αριθμός οχημάτων, με γνώμονα πάντα να καλύπτεται η ζήτηση των πελατών. Κύριος σκοπός του Προβλήματος της Δρομολόγησης Οχημάτων είναι η ελαχιστοποίηση του κόστους μεταφοράς των προϊόντων από ένα συγκεκριμένο αριθμό οχημάτων, τα οποία έχουν ως αφετηρία και τερματισμό το ίδιο σημείο, ήτοι την αποθήκη. Στην πραγματικότητα, υπάρχουν αρκετές περιπτώσεις που θέτουν διαφορετικούς περιορισμούς ως προς τη διανομή των προϊόντων για την επίλυση του εν λόγω προβλήματος. (1)

1.1.1 Η Εφοδιαστική Αλυσίδα (SupplyChain)

Ο όρος Εφοδιαστική Αλυσίδα (SupplyChain) αναφέρεται σε ένα ολοκληρωμένο δίκτυο που περιλαμβάνει στενά συνεργαζόμενες επιχειρηματικές μονάδες, όπως τους παραγωγούς πρώτων υλών, τους κατασκευαστές, τους εμπόρους, τους λιανοπωλητές και τους καταναλωτές. Η Εφοδιαστική Αλυσίδα επιτρέπει την ροή υλικών, υπηρεσιών και πληροφοριών από τους προμηθευτές πρώτων υλών και τους παραγωγούς των τελικών προϊόντων μέχρι τον τελικό πελάτη.

Η Εφοδιαστική Αλυσίδα είναι ένα δυναμικό σύστημα που εξελίσσεται μέσα στο χρόνο. Κύριος σκοπός της είναι η ικανοποίηση του πελάτη. Σε όλα τα στάδια δημιουργούνται κόστη, τα οποία προσauζάνονται στο τελικό προϊόν, το οποίο καλείται να πληρώσει ο τελικός καταναλωτής. Για το λόγο αυτό, επιτακτική ανάγκη είναι η βέλτιστη διαχείριση του συνόλου της Εφοδιαστικής Αλυσίδας, ώστε ο τελικός καταναλωτής να παραλάβει τα αγαθά ή υπηρεσίες στο καλύτερο δυνατό χρόνο αλλά και στην καλύτερη τιμή.



Σχήμα 1.1: Εφοδιαστική Αλυσίδα

1.1.2 Logistics

Ο όρος «Logistics» αναφέρεται συγκεκριμένα στο τμήμα της Διαχείρισης Εφοδιαστικής Αλυσίδας, το οποίο σχεδιάζει, υλοποιεί και ελέγχει την αποδοτική και αποτελεσματική, κανονική και αντίστροφη ροή και αποθήκευση των προϊόντων, υπηρεσιών και των σχετικών πληροφοριών από το σημείο προέλευσης τους έως το σημείο κατανάλωσης τους, ώστε να ικανοποιηθούν οι απαιτήσεις των πελατών – καταναλωτών. Αντικειμενικός σκοπός είναι η αύξηση της συνολικής κερδοφορίας κατά μήκος της αλυσίδας, η οποία και συνεπάγεται την αύξηση της κερδοφορίας όλων όσων συμμετέχουν σε αυτή. Αυτό δύναται να επιτευχθεί με την ορθή και έγκαιρη κατανόηση και ικανοποίηση των αναγκών των τελικών καταναλωτών, καθώς και με την προσφορά προϊόντων υψηλής προστιθέμενης αξίας. Επομένως, τα «Logistics» βρίσκουν εφαρμογή σε δύο πεδία, τα οποία είναι:

- Το πρώτο πεδίο είναι η επιχείρηση, η οποία πρέπει να οργανώσει την εισροή, την εσωτερική διακίνηση και την εκροή υλικών και προϊόντων κατά τέτοιον τρόπο, έτσι ώστε να εξασφαλίζει τη μέγιστη ικανοποίηση των πελατών της.
- Το δεύτερο πεδίο είναι η εφοδιαστική αλυσίδα, η οποία αποτελείται από όλες εκείνες τις επιχειρήσεις και οργανισμούς που είναι απαραίτητοι έτσι ώστε ένα προϊόν από πρώτες ύλες να καταλήξει στον τελικό πελάτη. Η αποτελεσματική οργάνωση και διοίκηση της ροής προϊόντων και πληροφοριών σε αυτήν την αλυσίδα αποτελεί επιτακτική ανάγκη σε μία παγκοσμιοποιημένη και ψηφιακή οικονομία, όπου ο ανταγωνισμός από ατομικός (επιχείρηση εναντίον επιχείρησης) γίνεται συλλογικός (εφοδιαστική αλυσίδα εναντίον εφοδιαστικής αλυσίδας). (Μαλινδρέτος, 2015)

Όπως γίνεται εύκολα αντιληπτό από τα παραπάνω, Logistics και Εφοδιαστική Αλυσίδα συνδέονται άρρηκτα μεταξύ τους, αφού στην ουσία τα Logistics αποτελούν τη Διοίκηση της Εφοδιαστικής Αλυσίδας.

Ένας άλλος ορισμός για τα Logistics αναφέρει ότι είναι η διαδικασία της στρατηγικής διαχείρισης των προμηθειών, της κίνησης και αποθήκευσης πρώτων υλών, εξαρτημάτων, τελικών αποθεμάτων, καθώς και των σχετικών πληροφοριών για τις ροές τους, μέσα στη επιχείρηση και τα κανάλια του Marketing με τέτοιο

τρόπο, ώστε η τρέχουσα και η μελλοντική κερδοφορία να μεγιστοποιούνται με την εκπλήρωση των παραγγελιών σύμφωνα με τις αρχές της αποτελεσματικότητας του κόστους. (MartinChristopher, 2007)

Τέλος, ένας ευρύτερα αποδεκτός ορισμός, ο οποίος διατυπώθηκε από το GlobalSupplyChainForum του Πανεπιστημίου TheOhioStateUniversity, Η.Π.Α. (Lambert, 2004) αναφέρει ότι «Logistics είναι η ολοκληρωμένη διαδικασία σχεδιασμού, εφαρμογής και ελέγχου βασικών διαδικασιών που μετατρέπουν τις εισροές από τους προμηθευτές σε προϊόντα και υπηρεσίες που προσθέτουν αξία στους πελάτες».

Βασικές Αρχές των Logistics

Τα Logistics, ως σύνθετο σύστημα, περιλαμβάνουν διάφορες επιμέρους διαδικασίες, οι οποίες στη συνέχεια και οι ίδιες αποτελούνται από επιμέρους δραστηριότητες. Οι βασικότεροι στόχοι του συστήματος των Logistics, με σκοπό την ορθή λειτουργία του, είναι οι παρακάτω:

- Ελαχιστοποίηση του λειτουργικού κόστους
- Ελαχιστοποίηση του κόστους επενδύσεων
- Βελτιστοποίηση της ποιότητας των παραγόμενων προϊόντων και υπηρεσιών
- Υψηλό επίπεδο εξυπηρέτησης πελατών

Επιπροσθέτως, θα πρέπει να γίνει κατανοητό ότι το κόστος των επιμέρους δραστηριοτήτων των Logistics, προστίθεται στο συνολικό κόστος των προϊόντων και διαμορφώνει την τελική τιμή αυτών. Αναλυτικά, ελαχιστοποίηση του κόστους θα πρέπει να αξιολογείται και να εξετάζεται το συνολικό κόστος και όχι μόνο τα επιμέρους κόστη των διαφόρων δραστηριοτήτων. Αυτό θα πρέπει να συμβαίνει, καθώς η ελαχιστοποίηση του κόστους μίας δραστηριότητας ίσως επιφέρει σημαντική αύξηση του κόστους σε μία άλλη δραστηριότητα και έτσι να μην επιτυγχάνεται τελικά η συνολική ελαχιστοποίηση του κόστους.

Εκτός από το κόστος, θα πρέπει να λαμβάνεται υπόψη και η ποιότητα των παραγόμενων προϊόντων και υπηρεσιών. Η βελτιστοποίηση της ποιότητας των προϊόντων και υπηρεσιών περιλαμβάνει ποιοτικά στοιχεία, τα βασικότερα από τα οποία είναι τα παρακάτω:

- Διαθεσιμότητα (Availability)

Η διαθεσιμότητα σχετίζεται με την διατήρηση του απαιτούμενου αριθμού αποθεμάτων, έτσι ώστε να είναι δυνατή η κάλυψη των αναγκών, τόσο των πελατών, όσο και της παραγωγής ανά πάσα στιγμή.

- Δυναμικότητα (Capacity)

Η δυναμικότητα σχετίζεται με την ταχύτητα εκτέλεσης των παραγγελιών.

- Συνέπεια (Consistency)

Η συνέπεια σχετίζεται με την έγκαιρη παράδοση των σωστών προϊόντων, σε καλή κατάσταση και δεδομένη ποιότητα, ώστε ο πελάτης να είναι σίγουρος ότι έχει παραλάβει τα προϊόντα που έχει παραγγείλει.

Οι δυο προαναφερόμενοι παράμετροι, ήτοι η ελαχιστοποίηση του κόστους και η βελτιστοποίηση της ποιότητας των προϊόντων και υπηρεσιών, συμβάλλουν στην επίτευξη της καλύτερης εξυπηρέτησης των πελατών, που είναι και ο βασικότερος σκοπός μας. Με σκοπό την παροχή της καλύτερης εξυπηρέτησης και κατ' επέκταση της ικανοποίησης των αναγκών του τελικού καταναλωτή, κάθε επιχείρηση θα πρέπει να παρέχει τα παρακάτω, τα λεγόμενα 7Σ:

- Σωστό προϊόν
- Σωστός τρόπος αποστολής
- Σωστός τόπος προορισμού
- Σωστό χρονοδιάγραμμα
- Σωστή ποσότητα
- Σωστή ποιότητα
- Σωστή τιμή(1)

Οι Δραστηριότητες των Logistics

Αρχικά θα πρέπει να τονιστεί το γεγονός ότι κάθε σύστημα Logistics σχεδιάζεται ανάλογα με τις ανάγκες και τη μορφή της κάθε επιχείρησης και για το λόγο αυτό δύναται να διαφέρει από επιχείρηση σε επιχείρηση. Γενικότερα όμως, οι βασικές δραστηριότητες ενός συστήματος Logistics, οι οποίες λαμβάνουν χώρα καθημερινά και οι οποίες συνθέτουν το κύκλωμα εφοδιασμού σε μία επιχείρηση είναι αναλυτικά οι εξής:

- Αγορές – Προμήθειες

Η δραστηριότητα που αφορά τις Αγορές και Προμήθειες μιας επιχείρησης αφορά κυρίως την απόκτηση τόσο προϊόντων, όσο και υπηρεσιών από τρίτους, δηλαδή από πηγές εκτός επιχείρησης (προμηθευτές). Οι εν λόγω αγορές αφορούν προϊόντα, τα οποία θα ενσωματωθούν στο παραγόμενο προϊόν και κατά περίπτωση μπορεί να αποτελούν πρώτες και βοηθητικές ύλες, υλικά συσκευασίας, ακόμα και καύσιμα τα οποία απαιτούνται για την λειτουργία της παραγωγικής μονάδας. Τέλος, να σημειωθεί ότι εκτός από τα παραπάνω, στις αγορές συγκαταλέγονται και οι υπηρεσίες από τρίτους. Η διαδικασία των αγορών, ήτοι η απόκτηση των αναγκαίων αγαθών και υπηρεσιών, με σκοπό την εύρυθμη λειτουργία μιας επιχείρησης, θα πρέπει να μελετηθεί και να σχεδιαστεί εξ αρχής, καθώς και να ικανοποιεί τους στόχους και τη στρατηγική της επιχείρησης.

Στο σημείο αυτό θα πρέπει να τονιστεί το γεγονός ότι το βασικότερο κριτήριο των αγορών δεν αποτελούν οι τιμές των προϊόντων. Σημαντικότερο παράγοντα αποτελεί η στενή συνεργασία μεταξύ των επιχειρήσεων και προμηθευτών. Η αξιολόγηση και επιλογή των πλέον κατάλληλων προμηθευτών, έγκειται κυρίως στην συνέπεια αυτών, στην ταχύτητα των παραδόσεων, καθώς και στην άμεση ανταπόκριση της μη προβλεπόμενης ζήτησης.

- Αποθήκευση

Σημαντική δραστηριότητα των Logistics είναι και αυτή της διαχείρισης της αποθήκης, η οποία απαιτεί στρατηγικό σχεδιασμό και οργάνωση. Αρχικά, ως αποθήκη ορίζεται ο χώρος τοποθέτησης και φύλαξης προϊόντων. Η οργάνωση μιας αποθήκης θα πρέπει να επιτρέπει την άμεση και συνεχή ροή των προϊόντων με τη βοήθεια ειδικού εξοπλισμού σε ράφια, μεταφορικά μέσα και μηχανογραφικά συστήματα. Τέλος, να σημειωθεί ότι στις κύριες δραστηριότητες μιας αποθήκης συγκαταλέγονται τα παρακάτω:

- Παραλαβή των προϊόντων
- Τοποθέτηση των προϊόντων στα κατάλληλα ράφια
- Φύλαξη των προϊόντων
- Άμεση ανεύρεση και ροή εντός επιχείρησης
- Παρακολούθηση του ύψους των διαθέσιμων αποθεμάτων ανά κωδικό
- Προετοιμασία των εμπορευμάτων για τη διανομή

- Διαχείριση Αποθεμάτων:

Η σημασία των αποθεμάτων σε μια επιχείρηση είναι σημαντική, καθώς συμβάλει στην ομαλή και απρόσκοπτη λειτουργία της. Ο όρος «Διαχείριση

Αποθεμάτων» αφορά τον έλεγχο και διατήρηση του κατάλληλου επιπέδου αποθεμάτων με σκοπό η επιχείρηση να μπορεί να ανταποκριθεί στη ζήτηση, τόσο της παραγωγής, όσο και των τελικών καταναλωτών. Το κατάλληλο ύψος αποθεμάτων δημιουργείται εφόσον απαντηθούν δύο ερωτήματα. Το ένα ερώτημα αφορά τον αριθμό των προϊόντων που πρέπει να παραγγείλει μια επιχείρηση, ενώ το δεύτερο σχετίζεται με τη χρονική στιγμή που θα πρέπει να συμβεί αυτό. Ο υπολογισμός του βέλτιστου επιπέδου αποθεμάτων απαιτεί αποτελεσματική λύση, καθώς επηρεάζει άμεσα το κόστος λειτουργίας μιας επιχείρησης και όπως έχει προαναφερθεί σκοπός είναι η ελαχιστοποίηση του.

- Διακίνηση (Μεταφορές - Διανομές):

Η διακίνηση αφορά στον εντοπισμό της βέλτιστης λύσης, όσων αφορά τη φυσική μετακίνηση των προϊόντων μεταξύ των επιχειρήσεων και των προμηθευτών. Το κόστος της διακίνησης αποτελεί ένα από τα σημαντικότερα στοιχεία που προσδιορίζουν το συνολικό κόστος κατά το σχεδιασμό ενός συστήματος Logistics και θα πρέπει να μελετηθεί σε βάθος. Η δραστηριότητα της διακίνησης εμπεριέχει την εύρεση των καταλληλότερων μέσων μεταφοράς, ενώ συνδράμει στην απόφαση μιας επιχείρησης για αγορά ή ενοικίαση οχημάτων ή ακόμη και στην ανάθεση των διανομών των προϊόντων της σε τρίτους. Στο σημείο αυτό αξίζει να γίνει διαχωρισμός των εννοιών «Διανομές» και «Μεταφορές». Η διαφορά τους έγκειται στο γεγονός ότι οι Διανομές έχουν ως αντικείμενο την διακίνηση των τελικών προϊόντων στους πελάτες της επιχείρησης, ενώ οι Μεταφορές αναφέρονται στην μεταφορά προϊόντων μεταξύ παραγωγής, αποθήκης, καθώς και στην μεταφορά Α΄ υλών στα διάφορα εργοστάσια. Με σκοπό την άμεση και χωρίς καθυστερήσεις εξυπηρέτηση της ζήτησης θα πρέπει να υπολογιστεί το κόστος της διακίνησης και να ληφθούν υπόψη διάφοροι παράμετροι με σκοπό την ελαχιστοποίηση του, ήτοι για παράδειγμα ο τρόπος εγκατάστασης μιας επιχείρησης, τα μεταφορικά μέσα που θα χρησιμοποιηθούν, οι συντομότερες διαδρομές προς τους πελάτες.

- Πληροφόρηση

Τέλος, σημαντική δραστηριότητα ενός συστήματος Logistics είναι και η Πληροφόρηση. Οι τεχνολογικές εξελίξεις, επιτρέπουν τόσο την εύκολη επικοινωνία των απομακρυσμένων εγκαταστάσεων, όσο και τη δημιουργία βάσεων δεδομένων, που επιτρέπουν την καταγραφή και τον έλεγχο των αποθεμάτων και των διανομών. (1)

Στο σημείο αυτό να σημειωθεί ότι η παρούσα διπλωματική εργασία θα επικεντρωθεί στην παρουσίαση των αλγορίθμων που σχετίζονται με την δραστηριότητα της διακίνησης των προϊόντων ενός συστήματος Logistics καθώς επίσης και την παρουσίαση των αποτελεσμάτων αυτών. Τέλος, θα πραγματοποιηθεί σύγκριση των αλγορίθμων και αναφορά της καλύτερης για το δικό μας πρόγραμμα.

Κεφάλαιο 2ο

2.1 Εισαγωγή

Στο δεύτερο κεφάλαιο αναλύεται το Πρόβλημα της Δρομολόγησης Οχημάτων. Αρχικά, παρουσιάζεται συνοπτικά το Πρόβλημα του Πλανόδιου Πωλητή (TSP – TravellingSalesmanProblem) που αποτελεί το πιο απλό πρόβλημα δρομολόγησης, στη συνέχεια παρουσιάζεται το Πρόβλημα της Δρομολόγησης Οχημάτων (VRP – VehicleRoutingProblem), τα βασικά χαρακτηριστικά του, ενώ γίνεται αναφορά στις διάφορες προεκτάσεις του εν λόγω προβλήματος, οι οποίες δημιουργούνται από τους διάφορους περιορισμούς που μπορεί να θέτει το πρόβλημα. (2)

2.2 Το πρόβλημα της Δρομολόγησης Οχημάτων (VRP)

Το Πρόβλημα της Δρομολόγησης Οχημάτων (VRP – VehicleRoutingProblem) παρουσιάστηκε για πρώτη φορά από τους Dantzing&Ramser το έτος 1959, ενώ σήμερα αποτελεί ένα από τα πιο σημαντικά και εφαρμόσιμα προβλήματα διανομής της εφοδιαστικής αλυσίδας. Οι Dantzing&Ramser δημιούργησαν την πρώτη αλγοριθμική προσέγγιση επίλυσης τέτοιων προβλημάτων και την εφάρμοσαν για την διανομή γκαζολίνης σε έναν αριθμό σταθμών τροφοδοσίας. Στη συνέχεια, το έτος 1964, οι Clarke&Wright πρότειναν έναν ευρετικό αλγόριθμο εξοικονόμησης, ο οποίος παρουσιάζει σημαντικές βελτιώσεις σε σχέση με αυτόν των Dantzing&Ramser. Ο εν λόγω αλγόριθμος παρουσιάζεται αναλυτικά στο τέταρτο Κεφάλαιο.

Το Πρόβλημα της Δρομολόγησης Οχημάτων αφορά στην ουσία ένα πρόβλημα διανομής, το οποίο όταν λυθεί θα πρέπει να επιφέρει όφελος για μια επιχείρηση μέσα από την εξοικονόμηση κόστους. Με σκοπό την επίλυση τέτοιου είδους προβλήματος, θα πρέπει αρχικά να υπολογιστούν οι βέλτιστες διαδρομές, τις οποίες θα εκτελέσει ένας συγκεκριμένος αριθμός οχημάτων, με γνώμονα πάντα να καλύπτεται η ζήτηση των πελατών.

Κύριος σκοπός του Προβλήματος της Δρομολόγησης Οχημάτων είναι η ελαχιστοποίηση του κόστους μεταφοράς των προϊόντων από ένα συγκεκριμένο αριθμό οχημάτων, τα οποία έχουν ως αφετηρία και τερματισμό το ίδιο σημείο, ήτοι την αποθήκη. Στην πραγματικότητα υπάρχουν αρκετές περιπτώσεις, που θέτουν διαφορετικούς περιορισμούς ως προς τη διανομή των προϊόντων για την επίλυση του εν λόγω προβλήματος

Ορισμός του Προβλήματος

Το πρόβλημα VRP(VehicleRoutingProblem) είναι ένα πρόβλημα ακέραιου γραμμικού προγραμματισμού το οποίο ανήκει στην κατηγορία των μη-πολυωνυμικών δύσκολων προβλημάτων (NP – Hard). Αυτό πρακτικά σημαίνει ότι η υπολογιστική προσπάθεια που απαιτείται για την επίλυση αυτού του προβλήματος αυξάνει εκθετικά με το μέγεθος του προβλήματος.

Το πρόβλημα

Σε ένα βασικό πρόβλημα δρομολόγησης οχημάτων έχουμε τα παρακάτω δεδομένα:

- Έναν συγκεκριμένο αριθμό πελατών που πρέπει να εξυπηρετηθούν, οι οποίοι αποτελούν τους κόμβους
- Τη ζήτηση του κάθε πελάτη
- Έναν συγκεκριμένο αριθμό οχημάτων, τα οποία θα πρέπει να ικανοποιήσουν τη ζήτηση των πελατών
- Δεν υπάρχουν χρονικοί περιορισμοί
- Δεν υπάρχουν περιορισμοί ως προς την προτεραιότητα εξυπηρέτησης

Ο στόχος

Ο στόχος επίλυσης του προβλήματος αποτυπώνεται ως εξής:

- Η δημιουργία ενός συνόλου διαδρομών χαμηλού κόστους. Αυτό δύναται να επιτευχθεί μέσα από τον υπολογισμό της διαδρομής εκείνης που ελαχιστοποιεί την διανυθείσα απόσταση ή το χρόνο παράδοσης των προϊόντων.
- Η ελαχιστοποίηση του αριθμού των οχημάτων και κατ' επέκταση και των οδηγών που απαιτούνται για την κάλυψη της δεδομένης ζήτησης του συνόλου των πελατών.
- Η εξισορρόπηση των διαδρομών των οχημάτων που θα προκύψουν με βάση τις ώρες που απαιτούνται για να διανυθούν αυτές, καθώς και με τα φορτία που αντιστοιχούν σε κάθε διαδρομή.
- Η ελαχιστοποίηση των κυρώσεων που ίσως προκύψουν από την μερική εξυπηρέτηση των πελατών. (Ombuki – Berman&Hanshar, 2009)

Μαθηματική Μοντελοποίηση του Προβλήματος

Η μαθηματική μοντελοποίηση του κλασικού προβλήματος της δρομολόγησης οχημάτων (VRP) έχει ως εξής:

Η αντικειμενική συνάρτηση:

$$\min z = \sum_{i=0}^N \sum_{j=1}^N \sum_{k=1}^K x_{ij}^k \cdot c_{ij} \quad (1)$$

Υπό τους περιορισμούς:

$$\sum_{i=0}^N \sum_{k=1}^K x_{ij}^k = 1 \quad \forall j \in \{1, \dots, N\} \quad (2)$$

$$\sum_{j=0}^N \sum_{k=1}^K x_{ij}^k = 1 \quad \forall i \in \{1, \dots, N\} \quad (3)$$

$$\sum_{j=0}^N x_{ij}^k + \sum_{j=0}^N x_{ij}^k = 1 \quad \forall i \in \{1, \dots, N\}, \forall k \in \{1, \dots, K\} \quad (4)$$

$$\sum_{i=0}^N \sum_{k=1}^K x_{ij}^k \cdot d_i \leq Q \quad \forall k \in \{1, \dots, K\} \quad (5)$$

$$\sum_{j=0}^N x_{0j}^k \leq 1 \quad \forall k \in \{1, \dots, K\} \quad (6)$$

$$\sum_{i=0}^N x_{i0}^k \leq 1 \quad \forall k \in \{1, \dots, K\} \quad (7)$$

Αναλυτικά, η αντικειμενική συνάρτηση (1) αναφέρεται στην ελαχιστοποίηση του συνολικού κόστους μεταφοράς, το οποίο εξαρτάται από την συνολική διανυθείσα απόσταση όλων των οχημάτων.

Οι περιορισμοί (2) και (3) εκφράζουν ότι κάθε πελάτης – κόμβος επισκέπτεται μόνο μία φορά από το όχημα k .

Ο περιορισμός (4) εκφράζει ότι εάν ένα όχημα επισκεφτεί έναν πελάτη πρέπει να φύγει από αυτόν.

Ο περιορισμός (5) εκφράζει ότι κανένα όχημα δεν δύναται να εξυπηρετήσει έναν πελάτη του οποίου η ζήτηση υπερβαίνει τη χωρητικότητα του οχήματος.

Οι περιορισμοί (6) και (7) διασφαλίζουν ότι κάθε όχημα δύναται να κάνει μία μόνο διαδρομή.

Τέλος, ο περιορισμός (8) εκφράζει ότι όλες οι μεταβλητές απόφασης είναι δυαδικές (0 ή 1).

Περιορισμοί του Προβλήματος της Δρομολόγησης Οχημάτων

Το Πρόβλημα της Δρομολόγησης Οχημάτων δύναται να περιλαμβάνει αρκετούς περιορισμούς στις διαδρομές που πρόκειται να ακολουθήσουν τα οχήματα, καθώς και περιορισμούς που σχετίζονται με τους πελάτες. Οι πρόσθετοι περιορισμοί διαμορφώνουν και τις παραλλαγές του προβλήματος που συναντάμε στην πράξη και που θα παρουσιαστούν αναλυτικά στην επόμενη ενότητα. Στην συνέχεια θα παρουσιαστούν διάφοροι περιορισμοί που τροποποιούν το κατά περίπτωση πρόβλημα δρομολόγησης και οι οποίοι θα πρέπει να λαμβάνονται υπόψη.

➤ Κέντρα Διάθεσης

Στην πιο απλή μορφή του προβλήματος, υπάρχει ένα σημείο διάθεσης (αποθήκη), από το οποίο εξυπηρετείται ένας συγκεκριμένος αριθμός πελατών με τη χρήση ενός συγκεκριμένου αριθμού οχημάτων. Ένας περιορισμός που μπορεί να τεθεί είναι η ύπαρξη περισσότερων του ενός κέντρων διάθεσης, καθώς επίσης και τα προϊόντα που θα πρέπει να διανεμηθούν να υπάρχουν διαθέσιμα σε συγκεκριμένες, ακόμη και σε περισσότερες από μία αποθήκες.

➤ Στόλος Οχημάτων

Ο σημαντικότερος περιορισμός σε ένα πρόβλημα VRP αναφορικά με τα οχήματα είναι αυτός της χωρητικότητας τους. Υπάρχει περίπτωση να υπάρχει ανομοιογένεια ως προς τη χωρητικότητα ενός στόλου οχημάτων, ήτοι κάθε όχημα να έχει διαφορετική χωρητικότητα. Περιορισμός δύναται να υπάρχει και ως προς τον τρόπο κοστολόγησης, ήτοι εάν τα οχήματα είναι ιδιόκτητα ή μισθωμένα. Τέλος, σοβαρά υπόψη πρέπει να λαμβάνονται και οι διανυόμενες αποστάσεις, καθώς και τα χρονικά διαστήματα κατά τα οποία τα οχήματα θα πρέπει να παραμένουν άεργα ή εντός των οποίων θα πρέπει να έχουν ολοκληρώσει υποχρεωτικά τις διανομές.

➤ Χρόνος Επίσκεψης

Κάθε κόμβος του δικτύου, ήτοι πελάτης, θα πρέπει να εξυπηρετείται εντός δεδομένων χρονικών ορίων (Χρονικά Παράθυρα). Τα χρονικά παράθυρα είναι χρήσιμα για τους πελάτες, καθώς έτσι τους δίνεται η δυνατότητα να προγραμματίσουν τις εργασίες τους, εν αντιθέσει με τους προμηθευτές που δεν διαθέτουν την απαιτούμενη ευελιξία για την παράδοση των προϊόντων.

➤ Χρόνος Εξυπηρέτησης

Ο χρόνος εξυπηρέτησης, ήτοι ο απαιτούμενος χρόνος παραμονής του οχήματος σε κάθε πελάτη, τις περισσότερες φορές είναι σταθερός. Πρακτικά όμως ο εν λόγω χρόνος εξαρτάται από τον πελάτη, την παραγγελθείσα ποσότητα προς παράδοση και την ευκολία πρόσβασης του οχήματος στις εγκαταστάσεις του πελάτη.

➤ Παραλαβή Προϊόντων:

Σε αρκετές περιπτώσεις εκτός από την παράδοση εμπορευμάτων στους πελάτες, υπάρχει και η περίπτωση παραλαβής επιστρεφόμενων προϊόντων. Στις περιπτώσεις όπου οι ποσότητες των επιστρεφόμενων προϊόντων είναι γνωστές ο προγραμματισμός της δρομολόγησης γίνεται πιο απλός, ενώ το πρόβλημα δυσκολεύει όταν η πληροφόρηση για τις εν λόγω ποσότητες είναι περιορισμένη ή και άγνωστη.

➤ Εξάρτηση των Κέντρων Διάθεσης (Αποθήκη) με τα Οχήματα:

Σε αρκετά προβλήματα δρομολόγησης αντιμετωπίζεται το πρόβλημα των διαφορετικών ειδών οχημάτων. Επιπρόσθετος περιορισμός μπορεί να υπάρξει και ως προς το ποιος πελάτης θα εξυπηρετηθεί από ποιο όχημα.

Τέλος, μπορεί να υπάρχει η δυνατότητα τα οχήματα να έχουν ως αφετηρία και τερματισμό της διαδρομής τους διαφορετικά σημεία, ήτοι αποθήκες.

➤ Εξάρτηση Πελατών - Οχημάτων:

Υπάρχουν προβλήματα δρομολόγησης, όπου το σύνολο των οχημάτων δεν δύναται να εξυπηρετήσει το σύνολο των πελατών. Τέτοιοι περιορισμοί δύναται να οφείλονται στην αδυναμία των οχημάτων να προσεγγίσουν τα σημεία παράδοσης, όπως για παράδειγμα λόγω του όγκου τους. Επιπροσθέτως, περιορισμοί μπορεί να δημιουργούνται και από τη σχέση που έχει δημιουργηθεί μεταξύ των οδηγών των οχημάτων και των πελατών, όπου οι τελευταίοι μπορεί να εκφράζουν προτίμηση σε κάποιον συγκεκριμένο οδηγό είτε επειδή έχουν αναπτύξει καλές σχέσεις, είτε λόγω του ότι ο οδηγός γνωρίζει τις ιδιαιτερότητες τους.

➤ Ασυμβατότητες μεταξύ Πελατών:

Αρκετές φορές περιορισμούς στα προβλήματα δρομολόγησης θέτουν οι ασυμβατότητες μεταξύ πελατών. Αρκετοί πελάτες λόγω του αντικειμένου τους έχουν εγκατασταθεί στην ίδια περιοχή και γειτονεύουν. Λόγω του ανταγωνισμού που επικρατεί, η παράδοση των προϊόντων σε κάθε πελάτη είναι προτιμότερο να δρομολογείται από διαφορετικό δρομολόγιο.

➤ Σημαντικότητα Πελατών:

Στην περίπτωση αυτή όλοι οι πελάτες δεν αντιμετωπίζονται με τον ίδιο τρόπο κατά τον σχεδιασμό των δρομολογίων. Σε κάποιες περιπτώσεις λόγω του ότι δεν υπάρχει αρκετό διαθέσιμο απόθεμα για να εξυπηρετηθεί το σύνολο της ζήτησης ή ακόμη και ο αριθμός των οχημάτων δεν είναι επαρκής, η επιχείρηση θέτει κριτήρια, βάσει των οποίων επιλέγονται οι πελάτες εκείνοι, οι οποίοι θα εξυπηρετηθούν με προτεραιότητα. Συνήθως επιλέγονται εκείνοι που συνεισφέρουν περισσότερο στο τζίρο της επιχείρησης.

➤ Χαρακτηριστικά Δικτύου:

Με σκοπό την μοντελοποίηση της πραγματικότητας με τον καλύτερο δυνατό τρόπο, θα πρέπει να ληφθεί υπόψη το γεγονός ότι οι χρόνοι μεταφοράς εξαρτώνται από εξωγενείς παράγοντες και διαφέρουν σε διαφορετικές χρονικές στιγμές της ημέρας. Ένα σημαντικό χαρακτηριστικό του δικτύου είναι για παράδειγμα η κίνηση στους δρόμους σε ώρες αιχμής.

2.3 Κατηγορίες Προβλημάτων Δρομολόγησης

Καθημερινά παρουσιάζονται διαφορετικά προβλήματα δρομολόγησης, τα οποία καλούνται να λάβουν υπόψη τους διαφορετικά δεδομένα, καθώς και διαφορετικούς περιορισμούς. Για το λόγο αυτό έχουν αναπτυχθεί διαφορετικά είδη VRP, τα οποία λύνουν τα κατά περίπτωση προβλήματα δρομολόγησης που δημιουργούνται. Στη συνέχεια αναλύονται οι σημαντικότερες κατηγορίες προβλημάτων δρομολόγησης.

2.3.1 Το πρόβλημα του Πλανόδιου Πωλητή (TSP) (3)

Το Πρόβλημα του Πλανόδιου Πωλητή (TSP - TravellingSalesmanProblem) αποτελεί την πιο απλή εκδοχή ενός προβλήματος δρομολόγησης, το οποίο διατυπώθηκε για πρώτη φορά το έτος 1930 και αποτελεί τη βάση για πιο σύνθετα προβλήματα βελτιστοποίησης. Το Πρόβλημα του Πλανόδιου Πωλητή είναι ένα κλασικό πρόβλημα βελτιστοποίησης. Τα δεδομένα του εν λόγω προβλήματος είναι τα εξής:

- Ένας αριθμός πόλεων που ο πωλητής θα πρέπει να επισκεφθεί.
- Μια πόλη αφετηρία.
- Τις αποστάσεις μεταξύ των πόλεων.

Να σημειωθεί ότι το Πρόβλημα του Πλανόδιου Πωλητή δεν έχει περιορισμούς ως προς τα παρακάτω:

- Χρόνος: Ο πωλητής έχει στη διάθεσή του απεριόριστο χρόνο και δεν υπάρχει περιορισμός ως προς το χρόνο άφιξης του σε κάθε πόλη.
- Μέγεθος και Χωρητικότητα του οχήματος.
- Ζήτηση των πελατών.

Ο πωλητής θα πρέπει να ξεκινήσει την διαδρομή του από την πόλη αφετηρία, θα πρέπει να επισκεφθεί όλες τις πόλεις από μία μόνο φορά και τέλος θα πρέπει να επιστρέψει στην αφετηρία. Στόχος του προβλήματος είναι η εύρεση της συντομότερης διαδρομής και κατ' επέκταση η ελαχιστοποίηση της συνολικά διανυθείσας απόστασης. Να σημειωθεί ότι οι πόλεις χρησιμοποιούνται για να γίνει πιο απλή και κατανοητή η περιγραφή του προβλήματος. Η διατύπωση του

προβλήματος παρουσιάζεται απλή αλλά στην πράξη δεν ισχύει, καθότι έως και σήμερα δεν έχει βρεθεί μια μέθοδος επίλυσης, η οποία λύνει το πρόβλημα μέσα σε λογικά χρονικά πλαίσια. Το Πρόβλημα του Πλανόδιου Πωλητή ανήκει στην κατηγορία των NP – Hard προβλημάτων. Αυτό σημαίνει ότι δεν υπάρχει κάποιος συγκεκριμένος αλγόριθμος, ο οποίος να λύνει το πρόβλημα σε πολυωνυμικό χρόνο, σε χρόνο εξαρτώμενο πολυωνυμικά από τον αριθμό των πόλεων που εξετάζονται στο πρόβλημα.

Συμπερασματικά, όσο αυξάνεται ο αριθμός των πόλεων, τόσο αυξάνεται εκθετικά και ο χρόνος που απαιτείται για την επίλυση του προβλήματος. Στην περίπτωση εκατοντάδων πόλεων ίσως απαιτούνται και χρόνια για τον εντοπισμό της βέλτιστης λύσης. Παρά το γεγονός ότι το συγκεκριμένο πρόβλημα παρουσιάζει αυξημένη δυσκολία ως προς τον υπολογισμό και την πολυπλοκότητά του, πολλοί αλγόριθμοι έχουν προταθεί με σκοπό τη λύση του, η οποία δίνει ένα αποτέλεσμα κοντά στη βέλτιστη. Οι εν λόγω αλγόριθμοι διακρίνονται σε τρεις κατηγορίες και είναι οι εξής:

- Αλγόριθμοι Τοπικής Αναζήτησης
- Αλγόριθμοι Κατασκευής Δρομολογίων
- Μεθευρετικοί Αλγόριθμοι

2.3.2 Πρόβλημα Δρομολόγησης Οχημάτων με Συγκεκριμένο Όριο Χωρητικότητας (Capacitated Vehicle Routing Problem – CVRP)

Το Πρόβλημα της Δρομολόγησης Οχημάτων με περιορισμένη χωρητικότητα συνιστά την συνηθέστερη εκδοχή τέτοιου είδους προβλημάτων. Για το εν λόγω πρόβλημα, ένας συγκεκριμένος αριθμός οχημάτων με δεδομένη χωρητικότητα, θα πρέπει να εξυπηρετήσει την ήδη γνωστή ζήτηση των πελατών από μία αποθήκη. Στόχος είναι η ελαχιστοποίηση του κόστους παράδοσης των προϊόντων στους πελάτες, μέσα από την ελαχιστοποίηση της διαδρομής των οχημάτων καθώς και του συνολικού χρόνου της μεταφοράς. Το εν λόγω πρόβλημα, αποτελεί επέκταση του Προβλήματος του Πλανόδιου Πωλητή (TSP - Traveling Salesman Problem), με τη διαφορά ότι όλοι οι πελάτες δεν εξυπηρετούνται σε μία μόνο διαδρομή του οχήματος. Ο λόγος που συμβαίνει αυτό

είναι ο περιορισμός που θέτει η μέγιστη χωρητικότητα του εκάστοτε οχήματος. Αναλυτικότερα, ως δεδομένα έχουμε:

- Έναν αριθμό n πελατών, οι οποίοι εξυπηρετούνται από μια αποθήκη.
- Την ζητούμενη ποσότητα d_i προϊόντων για κάθε πελάτη, όπου $i=1, 2, 3, \dots, n$.
- Τον αριθμό των οχημάτων χωρητικότητας K .

Με δεδομένη την περιορισμένη χωρητικότητα των οχημάτων, αυτά πραγματοποιούν πολλαπλές κυκλικές διαδρομές (R_i), οι οποίες σε κάθε περίπτωση έχουν ως αφετηρία και τερματισμό της διαδρομής το ίδιο σημείο, ήτοι την αποθήκη. Θα πρέπει να σημειωθεί ότι κάθε πελάτης εξυπηρετείται από μία και μοναδική διαδρομή, ενώ δεν επιτρέπεται η ζητούμενη ποσότητα να μοιραστεί σε περισσότερα του ενός οχήματα. Επομένως, η λύση ενός προβλήματος CVRP θα πρέπει να προσδιορίζει το σύνολο των απαιτούμενων διαδρομών (R_i), όπου κάθε πελάτης θα επισκέπτεται μια μόνο φορά, ενώ η συνολική ζήτηση των προϊόντων προς παράδοση δεν θα πρέπει σε καμία περίπτωση να ξεπερνά τη χωρητικότητα του οχήματος.

$$K : \sum_{i=1}^m d_i \leq K$$

Συνοψίζοντας, στόχος είναι η ελαχιστοποίηση του κόστους, μέσα από την ελαχιστοποίηση της συνολικά διανυθείσας απόστασης, καθώς και τον αριθμό των οχημάτων.

2.3.3 Πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα (Vehicle Routing Problem with Time Windows – VRPTW)

Το εν λόγω πρόβλημα αποτελεί επέκταση του προβλήματος CVRP, το οποίο και αναλύθηκε παραπάνω. Για την επίλυση του συγκεκριμένου προβλήματος, εξακολουθούν να ισχύουν οι περιορισμοί που οφείλονται στην χωρητικότητα των οχημάτων, επιπροσθέτως όμως το πρόβλημα καλείται να λάβει υπόψη το γεγονός ότι η εξυπηρέτηση του κάθε πελάτη θα πρέπει να λάβει χώρα εντός συγκεκριμένου χρονικού διαστήματος, το οποίο ονομάζεται χρονικό παράθυρο (Time Window). Το υπό εξέταση πρόβλημα, εκτός από τα χαρακτηριστικά που παρουσιάζει ένα CVRP πρόβλημα, περιλαμβάνει τόσο για την αποθήκη, όσο και για κάθε πελάτη C_i ($i=1, 2, 3, \dots, n$) ένα χρονικό παράθυρο $[b_i, e_i]$, κατά τη διάρκεια του οποίου ο

κάθε πελάτης θα πρέπει να εξυπηρετηθεί. Προγενέστερα ή μεταγενέστερα του συγκεκριμένου χρονικού παραθύρου δεν μπορεί να πραγματοποιηθεί εξυπηρέτηση. Για το λόγο αυτό ο νωρίτερος χρόνος εκκίνησης από την αποθήκη (σημείο 0) είναι ο b_0 , ενώ ο αργότερος χρόνος επιστροφής του οχήματος σε αυτή είναι το e_0 .

Το κάθε όχημα θα πρέπει να επισκεφθεί κάθε κόμβο (πελάτη) της διαδρομής που θα ακολουθήσει εντός ενός καθορισμένου χρονικού παραθύρου. Σε περίπτωση που η άφιξη του οχήματος στον κόμβο είναι προγενέστερη του νωρίτερου χρόνου b_i , τότε αυτό είναι υποχρεωμένο να αναμείνει έως τη χρονική στιγμή που ορίζει το παράθυρο. Αναφορικά με το χρόνο αναχώρησης από τον πελάτη, αυτός θα πρέπει υποχρεωτικά να είναι μικρότερος ή ίσος από το e_i .

Συμπερασματικά, ο στόχος επίλυσης τέτοιου είδους προβλημάτων, είναι η εύρεση ενός συνόλου διαδρομών, όπου κάθε διαδρομή θα έχει ως αρχή και τέλος την αποθήκη, ενώ δεν δύναται να παραβιάζονται οι περιορισμοί της χωρητικότητας και των χρονικών παραθύρων. Απώτερος σκοπός των παραπάνω είναι η ελαχιστοποίηση του συνολικού μήκους της κάθε διαδρομής και κατά συνέπεια και του κόστους αυτής.

2.3.4 Πρόβλημα Δρομολόγησης Οχημάτων με Backhauls (VehicleRoutingProblemwithBackhauls – VRPB)

Το Πρόβλημα Δρομολόγησης Οχημάτων με Backhauls (VRPB) συνιστά και αυτό ένα VRP πρόβλημα, στο οποίο οι πελάτες δύναται να παραλαμβάνουν αλλά και να επιστρέφουν προϊόντα. Στο πρόβλημα VRPPD, που αναλύθηκε παραπάνω, σημαντικό περιορισμό για την επίλυση του προβλήματος αποτελεί η επιστρεφόμενη ποσότητα των προϊόντων, καθώς θα πρέπει να προβλεφθεί χώρος στο εκάστοτε όχημα, ώστε αυτή να μπορεί να παραληφθεί. Στην περίπτωση των προβλημάτων δρομολόγησης οχημάτων με «Backhauls», και με σκοπό την αποφυγή της μη συμφέρουσας ανακατάταξης των εμπορευμάτων σε κάθε επισκεπτόμενο κόμβο – πελάτη, κάθε όχημα σε κάθε διαδρομή πραγματοποιεί πρώτα το σύνολο των παραδόσεων και στη συνέχεια πριν επιστρέψει στην

αποθήκη επισκέπτεται τους πελάτες από τους οποίους έχει να παραλάβει τα επιστρεφόμενα προϊόντα. Για την επίλυση του προβλήματος, οι πελάτες διακρίνονται σε δύο κατηγορίες. Στην πρώτη κατηγορία ανήκουν οι πελάτες που απαιτούν παραλαβή προϊόντων και ονομάζονται LinehaulsCustomers. Στην δεύτερη κατηγορία ανήκουν οι πελάτες που επιθυμούν την επιστροφή των προϊόντων, ήτοι οι επονομαζόμενοι BackhaulsCustomers.

Να σημειωθεί ότι οι ποσότητες των προϊόντων που παραδίδονται και επιστρέφονται είναι καθορισμένες εκ των προτέρων. Συνεπώς, και σε αυτή τη κατηγορία προβλημάτων, ισχύουν οι περιορισμοί της χωρητικότητας, των αποστάσεων, καθώς και η μοναδική επίσκεψη του κάθε κόμβου – πελάτη. Επιπρόσθετο περιορισμό αποτελεί το γεγονός ότι οι BackhaulsCustomers, θα πρέπει αυστηρά να επισκέπτονται μετά τους LinehaulsCustomers. Συμπερασματικά, στόχος είναι ο σχεδιασμός της βέλτιστης διαδρομής που θα οδηγήσει στην ελαχιστοποίηση της συνολικά διανυθείσας απόστασης, όπου όλες οι παραδόσεις πραγματοποιούνται πριν την περισυλλογή των επιστροφών. Τέλος, θα πρέπει να σημειωθεί ότι διαδρομές που περιλαμβάνουν μόνο πελάτες Backhauls δεν είναι επιτρεπτές.

2.3.5 Πρόβλημα Δρομολόγησης Οχημάτων Πολλαπλών Αποθηκών (VehicleRoutingProblemwithMultipleDepot – MDVRP)

Το εν λόγω πρόβλημα, αφορά τις επιχειρήσεις εκείνες που διαθέτουν περισσότερες από μια αποθήκες με σκοπό την εξυπηρέτηση των πελατών τους. Για την λύση του προβλήματος υπάρχουν οι εξής δύο τρόποι. Αναλυτικά: 1. Κάθε αποθήκη διαθέτει τον δικό της αριθμό οχημάτων, καθώς και πελατών που εξυπηρετεί, οπότε στην ουσία δημιουργούνται πολλαπλά, ανεξάρτητα VRP προβλήματα. 2. Κάθε όχημα έχει ως αφετηρία μία αποθήκη, ως τερματισμό μια άλλη, ενώ δύναται να διέρχεται από μία τρίτη αποθήκη για ανεφοδιασμό.

Γενικότερα, το εν λόγω πρόβλημα μπορεί να θεωρηθεί ως ένα πρόβλημα ομαδοποίησης, αφού ως στόχο έχει την ανεύρεση των διαδρομών των οχημάτων που ανήκουν σε κάθε αποθήκη. Στην ομαδοποίηση που δημιουργείται αρχικά

εντοπίζονται οι πελάτες που ανήκουν σε κάθε αποθήκη και στην συνέχεια δημιουργούνται τα δρομολόγια για κάθε όχημα.

2.3.6 Πρόβλημα Δρομολόγησης Οχημάτων Περιόδου (Periodic Vehicle Routing Problem – PVRP)

Σε αντίθεση με συνηθισμένα προβλήματα VRP, όπου οι παραδόσεις ρυθμίζονται σε ημερήσιο επίπεδο, το πρόβλημα δρομολόγησης οχημάτων περιόδου είναι πιο περίπλοκο, καθώς επεκτείνει το διάστημα του προγραμματισμού σε M ημέρες. Στην ουσία πρόκειται για ένα Πολύ – Επίπεδο Συνδυαστικό Πρόβλημα Βελτιστοποίησης, τα επίπεδα του οποίου αναλύονται στη συνέχεια:

- Πρώτο Επίπεδο: Στο επίπεδο αυτό καθορίζεται η περίοδος σύμφωνα με την οποία πραγματοποιούνται οι διανομές για κάθε πελάτη, ήτοι κάθε X ημέρες.
- Δεύτερο Επίπεδο: Στο εν λόγω επίπεδο πραγματοποιείται επιλογή από τις διάφορες εναλλακτικές για κάθε πελάτη, τηρώντας όμως τους περιορισμούς που έχουν τεθεί για τις διανομές. Συνεπώς, πραγματοποιείται ομαδοποίηση των πελατών που θα πρέπει να επισκεφτεί το όχημα την ίδια ημέρα της περιόδου.
- Τρίτο Επίπεδο: Σε αυτό το επίπεδο το πρόβλημα επιλύεται ως απλό πρόβλημα VRP για κάθε μία ημέρα της υπό εξέταση περιόδου.

Στόχος επίλυσης του υπό ανάλυση προβλήματος δρομολόγησης είναι η ελαχιστοποίηση του συνολικού αριθμού των χρησιμοποιούμενων οχημάτων, καθώς και η μείωση στο ελάχιστο της συνολικά διανυθείσας απόστασης. Και στην περίπτωση του προβλήματος δρομολόγησης οχημάτων περιόδου, θα πρέπει να καλύπτονται οι τυπικοί περιορισμοί ενός προβλήματος VRP, επιπροσθέτως όμως θα πρέπει τα οχήματα να αναχωρούν και να επιστρέφουν στην αποθήκη την ίδια ημέρα, καθώς επίσης και το σύνολο των πελατών θα πρέπει να έχει δεχθεί επίσκεψη τουλάχιστον μία φορά εντός της ορισθείσας περιόδου των διανομών.

2.3.7 Προβλήματα Δρομολόγησης Οχημάτων με Διασπαρμένες Παραδόσεις (SplitDeliveryVehicleRoutingProblem – SDVRP)

Το Πρόβλημα της Δρομολόγησης Οχημάτων με διασπαρμένες παραδόσεις στην ουσία είναι πιο απλό από ένα κλασικό πρόβλημα VRP. Η διαφορά έγκειται στο γεγονός ότι είναι δύσκολο να επιτευχθεί βέλτιστη λύση. Στην ουσία κάθε πελάτης δύναται να εξυπηρετηθεί από περισσότερα οχήματα, εφόσον όμως πετυχαίνονται μειωμένα κόστη διανομής. Η λύση του εν λόγω προβλήματος SDVRP είναι κατάλληλη για τις περιπτώσεις εκείνες που η ζήτηση των προϊόντων από ένα πελάτη ξεπερνάει το όριο της χωρητικότητας του οχήματος.

Έτσι σε ένα πρόβλημα SDVRP πραγματοποιείται κατάτμηση της παραγγελίας του πελάτη σε μικρότερες αδιαίρετες παραγγελίες και στη συνέχεια υπολογίζονται οι διαδρομές με σκοπό την ελαχιστοποίηση του κόστους διανομών. Στόχος όπως και σε όλα τα προβλήματα δρομολόγησης οχημάτων, είναι η ελαχιστοποίηση του συνολικού αριθμού των χρησιμοποιούμενων οχημάτων, καθώς και η μείωση στο ελάχιστο της συνολικά διανυθείσας απόστασης. Και στην περίπτωση του προβλήματος δρομολόγησης οχημάτων με διασπασμένες παραδόσεις, θα πρέπει να καλύπτονται οι τυπικοί περιορισμοί ενός προβλήματος VRP, επιπροσθέτως όμως υπάρχει ευελιξία στην εξυπηρέτηση των πελατών από περισσότερα του ενός οχήματα.

2.3.8 Λοιπά Προβλήματα Δρομολόγησης (1)

Στη συνέχεια παρουσιάζονται συνοπτικά και άλλες παραλλαγές του Προβλήματος της Δρομολόγησης Οχημάτων.

- Στοχαστικό Πρόβλημα Δρομολόγησης Οχημάτων (StochasticVehicleRoutingProblem– SVRP):

Σε αυτή την κατηγορία, ένα ή και περισσότερα στοιχεία του προβλήματος, ήτοι ο αριθμός των πελατών καθώς και η ζήτηση τους, δεν αποτελούν σταθερά δεδομένα αλλά δυναμικά. Αυτό σημαίνει ότι τα εν λόγω στοιχεία δύναται να αλλάξουν ανά πάσα στιγμή και άρα δεν είναι γνωστά εκ των προτέρων.

- Χρόνο-εξαρτώμενη Δρομολόγηση Στόλου Οχημάτων
(TimeDependentVehicleRoutingProblem – TDVRP):
Στην περίπτωση αυτή λαμβάνεται υπόψη η κίνηση του χρησιμοποιούμενου δικτύου κάθε χρονική στιγμή. Κατά συνέπεια ο χρόνος που απαιτείται από έναν κόμβο σε κάποιον άλλο είναι μεταβλητός.
- Δρομολόγηση Ετερογενούς Στόλου Οχημάτων
(HeterogeneousFleetVehicleRoutingProblem – HFVRP):
Η συγκεκριμένη κατηγορία προβλήματος αντιμετωπίζει την ετερογένεια μεταξύ των οχημάτων που ανήκουν στον ίδιο στόλο. Η εν λόγω ετερογένεια μπορεί να οφείλεται στην διαφορετική χωρητικότητα των οχημάτων, στην διαφορετική τεχνολογία, καθώς και στον τρόπο κοστολόγησης.
- Δρομολόγηση Στόλου Οχημάτων με Εξισορρόπηση Διαδρομών
(VehicleRoutingProblemwithRouteBalancing – VRPRB):
Με σκοπό την ίση κατανομή του φόρτου εργασίας μεταξύ των οδηγών, το εν λόγω πρόβλημα καλείται να εξισορροπήσει τις διάφορες διαδρομές, ήτοι να μικρύνει μια μεγάλη διαδρομή ενώ παράλληλα θα αυξήσει τους κόμβους που θα επισκεφθεί μια μικρή διαδρομή.
- Δυναμική Δρομολόγηση Στόλου Οχημάτων
(DynamicVehicleRoutingProblem – DVRP):
Στη εν λόγω κατηγορία προβλημάτων, λαμβάνεται υπόψη η δυνατότητα που παρέχεται στον πελάτη να ακυρώνει ή να τοποθετεί νέες παραγγελίες κατά τη διάρκεια κίνησης του οχήματος.

3.1 Σπουδαιότητα μελέτης του προβλήματος δρομολόγησης οχημάτων VRP (VehicleRoutingProblems)

Ιδιαίτερα σημαντική η εφαρμογή του παρόντος προβλήματος για τον τομέα της εφοδιαστικής αλυσίδας, αφού έχει ως στόχο την εκτέλεση της διαδικασίας των παραδόσεων και των επιστροφών, με το ίδιο όχημα ανά πελάτη κάθε φορά, στον λιγότερο χρόνο και με τα λιγότερα οχήματα, κάτι που θα οδηγήσει στη μείωση του κόστους στην εκάστοτε επιχείρηση. Σε κάθε περίπτωση όμως, πρέπει να έχει υπολογιστεί ώστε η συνολική ποσότητα προϊόντων (προς παράδοση αλλά και επιστροφή) στο όχημα να μην υπερβαίνει την μεταφορική του ικανότητα. Η παρούσα εργασία παρουσιάζει επίλυση του προβλήματος δρομολόγησης οχημάτων με ταυτόχρονη παραλαβή και παράδοση στον πελάτη-καταναλωτή με την χρήση των αλγορίθμων επίλυσης όπου θα αναφερθούν στο επόμενο κεφάλαιο. Πλεονέκτημά μας η μείωση του χρόνου ανά όχημα, λόγω της ταυτόχρονης εξυπηρέτησής του πελάτη από το αντίστοιχο όχημα, με στόχο την ακόμα μεγαλύτερη μείωση του κόστους.

3.2 Πρόβλημα δρομολόγησης οχημάτων VRP (VehicleRoutingProblems)

Τα προβλήματα δρομολόγησης VRPPD (VehicleRoutingProblemwithPickupsandDeliveries) αναφέρονται στη μεταφορά αντικειμένων ή ανθρώπων από ένα συγκεκριμένο σημείο προς ένα άλλο (προέλευση-προορισμός). Ένα VRPPD είναι στατικό όταν όλα τα δεδομένα του προβλήματος είναι γνωστά πριν την κατασκευή του δρομολογίου, ενώ ονομάζεται δυναμικό όταν ορισμένα από τα δεδομένα εμφανίζονται ή μεταβάλλονται κατά τη διάρκεια της εφαρμογής των δρομολογίων. Επίσης, ένα VRPPD μπορεί να χαρακτηριστεί ως single-vehicle όταν υπάρχει διαθέσιμο μόνο ένα όχημα ενώ όταν χρησιμοποιούνται περισσότερα οχήματα χαρακτηρίζεται ως multi-vehicle. Τα VRPPD αναλύονται σε δύο κατηγορίες:

- Η πρώτη ασχολείται με περιπτώσεις όπου η σχέση του σημείου προέλευσης με τον προορισμό είναι ανύπαρκτη, δηλαδή κάθε προϊόν το οποίο περισυλλέγεται μπορεί να παραδοθεί σε διαφορετικούς από έναν προορισμούς. Οπότε κάθε απαίτηση παράδοσης σε κάποιο συγκεκριμένο σημείο μπορεί να εκπληρωθεί για οποιοδήποτε αγαθό.
- Η δεύτερη κατηγορία περιλαμβάνει το κλασικό PDP (PickupandDeliveryProblem) και το DARP (Dial-A-Ride Problem), τα οποία ασχολούνται με αιτήσεις μεταφοράς από ένα σημείο προέλευσης σε ένα σημείο παράδοσης. Σημειώνεται ότι τα δύο τελευταία αποτελούν ‘ζευγάρι’ δηλαδή ο προορισμός δέχεται μόνο ένα αγαθό από ένα συγκεκριμένο σημείο προέλευσης. Τα PDP αναφέρονται σε μεταφορές προϊόντων ενώ τα DARP σε μεταφορές ανθρώπων. Αυτή η διαφορά συνήθως εκφράζεται με όρους επιπρόσθετων περιορισμών οι οποίοι αναφέρονται σαφώς στην άνεση του πελάτη. Στο παρακάτω σχήμα παρουσιάζονται όλες οι κατηγορίες των VRPPD οι οποίες θα αναλυθούν περαιτέρω στις παραγράφους που ακολουθούν.

3.2.1 Πρόβλημα Δρομολόγησης Οχημάτων με Διανομή και Παραλαβή κατά τη Διάρκεια της Διαδρομής (VehicleRoutingProblemwithPickupandDelivery – VRPPD)

Το Πρόβλημα Δρομολόγησης Οχημάτων με διανομή και παραλαβή προϊόντων κατά τη διάρκεια της διαδρομής (VRPPD), αποτελεί ένα πιο σύνθετο πρόβλημα από το CVRP, το οποίο θα πρέπει να λαμβάνει υπόψη και την επιστροφή προϊόντων από τους πελάτες. Κατά συνέπεια, θα πρέπει να προβλεφθεί ικανοποιητικός χώρος στο όχημα, γεγονός που θα επιτρέπει την παραλαβή και τοποθέτηση των επιστρεφόμενων προϊόντων σε αυτό. Ο εν λόγω περιορισμός αποτελεί και τη δυσκολία του συγκεκριμένου προβλήματος, καθώς μπορεί να οδηγήσει σε εσφαλμένη διαχείριση της χωρητικότητας των οχημάτων, ενώ υπάρχει η πιθανότητα να απαιτούνται είτε περισσότερα οχήματα, είτε τα υπάρχοντα να πρέπει να εκτελέσουν περισσότερες διαδρομές.

Για την επίλυση ενός προβλήματος VRPPD, θα πρέπει να λαμβάνονται υπόψη τα παρακάτω:

- Αρχή και τέλος της διαδρομής είναι η αποθήκη. Κατ' επέκταση όλες οι παραγγελίες προς παράδοση έχουν ως αφετηρία την αποθήκη, ενώ οι επιστροφές των προϊόντων που παραλαμβάνονται κατά τη διαδρομή τερματίζουν στην αποθήκη.
- Δεν ανταλλάσσονται προϊόντα μεταξύ των πελατών που αποτελούν και τους κόμβους της διαδρομής.
- Το όχημα επισκέπτεται κάθε κόμβο μόνο μία φορά.

Συμπερασματικά, ο στόχος επίλυσης τέτοιου είδους προβλημάτων, είναι η εύρεση ενός συνόλου διαδρομών, όπου κάθε διαδρομή θα έχει ως αρχή και τέλος την αποθήκη, ενώ θα πρέπει να λαμβάνεται σοβαρά υπόψη η ύπαρξη διαθέσιμου χώρου για την τοποθέτηση των επιστρεφόμενων προϊόντων, εκτός από αυτών προς παράδοση. Απώτερος σκοπός των παραπάνω είναι η εύρεση του απαιτούμενου σε αριθμό στόλο οχημάτων, καθώς και η ελαχιστοποίηση του συνολικού μήκους της κάθε διαδρομής, γεγονός που συνεπάγεται με μείωση του κόστους.

Για οποιοδήποτε υποσύνολο $S \subseteq V$ θεωρούμε ότι $A(S, S') = \{(i, j) \in A : i \in S, j \notin S\}$.

$$\text{Έχουμε: } \min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (1)$$

υπό τους όρους:

$$\sum_{(i,j) \in A} x_{ij} = 1$$

Η αντικειμενική συνάρτηση (1) ελαχιστοποιεί το συνολικό κόστος δρομολόγησης.

3.2.2 Το μικτό πρόβλημα δρομολόγησης οχημάτων με δύο είδη πελατών κατά τη διάρκεια της διαδρομής (TheMixedVehicleRoutingProblemwithBackhauls (MVRPB))

Το μικτό πρόβλημα δρομολόγησης οχημάτων με δύο είδη πελατών κατά τη διάρκεια της διαδρομής προέρχεται από το VRPB χαλαρώνοντας κάποιους περιορισμούς. Δηλαδή, επιτρέπεται ο συνδυασμός των πελατών που απαιτούν την διανομή κάποιας ποσότητας προϊόντων (linehaulcustomers) με τους πελάτες που ο κάθε ένας εκ των οποίων απαιτεί μια ποσότητα του προϊόντος να περισυλλεχθεί από αυτόν (backhaulcustomers) μέσα σε μια διαδρομή και είμαστε ελεύθεροι να χρησιμοποιήσουμε όσα οχήματα θέλουμε. Ισχύουν οι περιορισμοί χωρητικότητας των οχημάτων. Ο έλεγχος για την χωρητικότητα των οχημάτων είναι ελαφρώς πιο περίπλοκος στο πρόβλημα MVRPB καθώς το φορτίο του οχήματος αυξομειώνεται κατά τη διάρκεια της διαδρομής. Επιπλέον, ορισμένα MVRPB έχουν επίσης ένα όριο διάρκειας διαδρομής, αυτό σημαίνει ότι οι διαδρομές θα πρέπει να ολοκληρωθούν εντός συγκεκριμένο χρονικού πλαισίου. Για τέτοιου είδους προβλήματα ο χρόνος ταξιδιού μεταξύ των πελατών και οι χρόνοι εξυπηρέτησης των πελατών δίνονται. Οι ευρετικοί αλγόριθμοι για αυτό το πρόβλημα παρουσιάζονται από τους Halse(5), Nagy και Salhi(6) (7) και Wade και Salhi(8). Ο στόχος είναι να βρεθεί το σύνολο των διαδρομών με το μικρότερο κόστος όπου:

- Κάθε πελάτης επισκέπτεται ακριβώς μία φορά,
- όλες οι διαδρομές αρχίζουν και τελειώνουν στην αποθήκη,
- το φορτίο του οχήματος σε οποιοδήποτε σημείο της διαδρομής δεν πρέπει να υπερβαίνει την χωρητικότητα του οχήματος.

Το μικτό πρόβλημα είναι πιο περίπλοκο από το κλασσικό VRPB λόγω του κυμαινόμενου φορτίου του οχήματος. Το μικτό VRPB δεν έχει λάβει τόσο μεγάλη προσοχή στη βιβλιογραφία όσο το αντίστοιχο του κλασσικό VRPB. Το VRPB είναι στενά συνδεδεμένο με το πρόβλημα του πλανόδιου πωλητή (TSP), βασίζουμε τη μέθοδο μας στο σύστημα AntColony γνωστό ως ο αλγόριθμος ACS που χρησιμοποιήθηκε επιτυχώς για την επίλυση του TSP, Dorigo και Gambardella

3.2.3 Πρόβλημα δρομολόγησης οχημάτων με χρονικά περιθώρια (VehicleRoutingProblemwithTimeWindows)

Το πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα μπορεί να καθοριστεί ως εξής: κάθε πελάτης i πρέπει να εξυπηρετηθεί μέσα σε μία χρονική περίοδο $[\alpha_i, \beta_i]$ η οποία ονομάζεται χρονικό παράθυρο. Τα επιπλέον δεδομένα που δίνονται για το πρόβλημα είναι η χρονική στιγμή κατά την οποία τα οχήματα φεύγουν από την αποθήκη, ο χρόνος ταξιδιού t_{ij} , για κάθε τόξο (i,j) και φυσικά ένας χρόνος εξυπηρέτησης για κάθε πελάτη. Η εξυπηρέτηση για κάθε πελάτη πρέπει να αρχίσει μέσα στο χρονικό παράθυρο που μπορεί ο πελάτης να εξυπηρετηθεί, και το όχημα πρέπει να παραμείνει στην τοποθεσία που βρίσκεται ο πελάτης για χρόνο s_i . Επιπλέον, αν κάποιο όχημα φτάσει σε κάποιον πελάτη νωρίτερα από τον προκαθορισμένο χρόνο, στις περισσότερες περιπτώσεις το όχημα επιτρέπεται να παραμείνει στην τοποθεσία του πελάτη μέχρι να ξεκινήσει το χρονικό παράθυρο.

Στις περισσότερες φορές οι πίνακες κόστους και ταξιδιού συμπίπτουν, και τα χρονικά παράθυρα καθορίζονται βάσει το γεγονός ότι όλα τα οχήματα φεύγουν από την αποθήκη τη χρονική στιγμή 0. Επιπλέον τα χρονικά παράθυρα απαιτούν ένα πλήρη προσανατολισμό της κάθε διαδρομής ακόμα και αν οι αρχικοί πίνακες είναι συμμετρικοί. Άρα, στις περισσότερες φορές το πρόβλημα προτυποποιείται σαν μη συμμετρικό πρόβλημα.

Υπάρχουν δύο ειδών χρονικά παράθυρα, τα χαλαρά κατά τα οποία αν ένα όχημα φτάσει σε κάποιο πελάτη κάποια χρονική στιγμή εκτός του χρονικού παραθύρου, μπορεί να ξεκινήσει την εξυπηρέτηση του εκείνη τη στιγμή και τα σκληρά χρονικά παράθυρα που δεν επιτρέπουν να φτάσει το όχημα στον πελάτη μετά από τον αργότερο χρόνο εξυπηρέτησης. Σε αυτές τις περιπτώσεις αν ένα όχημα φτάσει στον νωρίτερα από το χρονικό παράθυρο τότε θα περιμένει για να αρχίσει η εξυπηρέτηση.

Το πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα αποτελείται από την εύρεση K ακριβώς κύκλων με ελάχιστο κόστος τέτοιους ώστε :

- Κάθε κύκλος περνάει από την αποθήκη.
- Κάθε πελάτης επισκέπτεται από ένα μόνο κύκλο.
- Το άθροισμα της ζήτησης των κόμβων που επισκέπτονται από ένα κύκλο δεν ξεπερνάει την χωρητικότητα του οχήματος.

- Για κάθε πελάτη η εξυπηρέτηση πρέπει να ξεκινήσει και να ολοκληρωθεί μέσα στο χρονικό παράθυρο $[α_i, β_i]$ ενώ το όχημα θα παραμείνει στον χώρο του πελάτη για χρόνο s_i μέχρι να ξεφορτώσει.

Στη συνέχεια θα δούμε μια μορφοποίηση του προβλήματος. Το πρόβλημα δρομολόγησης με χρονικά παράθυρα αναφέρεται σε ένα γράφημα $G = (V, A)$, όπου η αποθήκη συμβολίζεται με τους κόμβους 0 και $n+1$. Όλες οι εφικτές διαδρομές αντιπροσωπεύουν μονοπάτια που ξεκινούν από το 0 και καταλήγουν στο $n+1$. Ένα χρονικό παράθυρο αντιστοιχίζεται και με την αποθήκη είτε αναφερόμαστε στον κόμβο 0 είτε στον κόμβο $n+1$, για παράδειγμα $\{α_0, β_0\} = \{α_{n+1}, β_{n+1}\} = \{E, L\}$ όπου τα E και L, είναι η ελάχιστη πιθανή αναχώρηση από την αποθήκη και η αργότερη δυνατή άφιξη. Επιπλέον, μηδενική ζήτηση και χρόνοι εξυπηρέτησης καθορίζονται για αυτούς τους δύο κόμβους, δηλαδή $d_0 = d_{n+1} = s_0 = s_{n+1} = 0$

Στο πρότυπο που ακολουθεί περιλαμβάνονται δύο ειδών μεταβλητές ροής:

$$x_{ijk} = \begin{cases} 1, & \text{εάν το όχημα επισκέπτεται τον πελάτη } j \text{ αμέσως μετά τον πελάτη } i \\ 0, & \text{αλλιώς} \end{cases} \quad (3.1)$$

και μία οι χρονικές μεταβλητές w_{ik} που καθορίζουν πότε θα ξεκινήσει η εξυπηρέτηση στον πελάτη i από το όχημα k .

Το βασικό πρόβλημα δρομολόγησης οχημάτων μπορεί να καθοριστεί ως εξής:

$$\min \sum_{i,j} c_{ij} \sum_k x_{ijk} \quad (3.2)$$

υπό

$$\sum_{k \in K} \sum_{j \in V} x_{ijk} = 1 \quad \forall i \in N \quad (3.3)$$

$$\sum_{i \in V - \{0\}} x_{0jk} = 1 \quad \forall j \in N, k \in K \quad (3.4)$$

$$\sum_{i \in V - \{j\}} x_{ijk} - \sum_{i \in V - \{j\}} x_{jik} = 0 \quad \forall j \in N, k \in K \quad (3.5)$$

$$\sum_{i \in V - \{n+1\}} x_{in+1k} = 1 \quad \forall k \in K \quad (3.6)$$

$$x_{ijk}(w_{ik} + s_i + t_{ij} - w_{jk}) \leq 0 \quad \forall i, j \in N, k \in K \quad (3.7)$$

$$\alpha_i \sum_{j \in V} x_{ijk} \leq w_{ik} \leq \beta_i \sum_{j \in V} x_{ijk} \quad \forall i \in N, k \in K \quad (3.8)$$

$$E \leq w_{ik} \leq L \quad \forall i \in (0, n+1), k \in K \quad (3.9)$$

$$\sum_{i \in N} d_i \sum_{j \in V} x_{ijk} \leq C \quad k \in K \quad (3.10)$$

$$x_{ijk} \geq 0 \quad k \in K, (i,j) \in A \quad (3.11)$$

$$x_{ijk} \in \{0,1\} \quad \forall k \in K, (i,j) \in A \quad (3.12)$$

Η αντικειμενική συνάρτηση εκφράζει το συνολικό κόστος. Οι περιορισμοί (3.3) περιορίζουν την εκχώρηση κάθε πελάτη σε ένα και μόνο όχημα. Οι περιορισμοί (3.4)-(3.6) χαρακτηρίζουν τη ροή στο μονοπάτι που ακολουθείται από το όχημα k . Οι περιορισμοί (3.7),(3.9),(3.10) εγγυώνται την εφικτότητα των διαδρομών με βάση τους χρονικούς περιορισμούς και τους περιορισμούς χωρητικότητας των οχημάτων. Τέλος, ο περιορισμός (3.8) εξαναγκάζει το w_{ik} για ένα δεδομένο k να γίνει ίσο με το 0 αν το όχημα δεν επισκέπτεται τους πελάτες i και j σε αυτή τη διαδρομή.

Μια πιο σύνθετη εφαρμογή είναι αν υποθέσουμε επίσης ότι σε κάθε πελάτη, το όχημα που τον επισκέπτεται εκτελεί όχι μόνο παράδοση εμπορεύματος αλλά και παραλαβή, μέσα στα πλαίσια της αντίστροφης εφοδιαστικής (reverselogistics). Ο σκοπός είναι να ορισθούν κυκλικές διαδρομές για τα οχήματα που να καλύπτουν όλους τους πελάτες με το ελάχιστο δυνατόν κόστος.

Υποθέτουμε ότι υπάρχει ένας στόλος K οχημάτων, όλα με την ίδια δυναμικότητα u και ένα σύνολο n πελατών που πρέπει να εξυπηρετηθούν από ένα κέντρο, το οποίο όπως και στο πρόβλημα του πλανόδιου πωλητή θα το διαιρέσουμε σε δύο κόμβους: 0 και $n+1$. Για κάθε πελάτη i θα υποθέσουμε ότι υπάρχει αίτημα για παράδοση d_i μονάδων και παραλαβή p_i μονάδων από το όχημα που τον επισκέπτεται και ότι η επίσκεψη του οχήματος θα πρέπει να πραγματοποιηθεί στο χρονικό διάστημα $[a_i, b_i]$.

Επειδή κάθε πελάτης εξυπηρετείται από ένα και μόνο όχημα που τον επισκέπτεται μία και μόνη φορά θα υποθέσουμε αναγκαστικά ότι $0 \leq d_i, p_i \leq u$, $\forall i$. Επίσης, είναι φυσιολογικό να δεχτούμε ότι ένα όχημα μπορεί να αφικνείται στον χώρο του πελάτη i πριν τη χρονική στιγμή a_i και ότι τότε αναμένει ως την έναρξη του χρονικού διαστήματος εξυπηρέτησης. Χωρίς απώλεια της γενικότητας θα υποθέσουμε ότι η αναμονή αυτή δεν κοστίζει. Φυσικά και δεν επιτρέπεται η άφιξη του οχήματος καθυστερημένα, δηλαδή μετά την χρονική στιγμή b_i . Τέλος, δεδομένου του μεγέθους του στόλου οχημάτων, η κάλυψη όλων των πελατών δύναται να γίνει χρησιμοποιώντας το πολύ K κυκλικές διαδρομές.

Ας είναι το c_{ij} το κόστος και t_{ij} ο χρόνος της απευθείας κίνησης ενός οχήματος από το $i=0,1,\dots,n,n+1$ στο $j=0,1,n,n+1$, $j \neq i$. Ορίζοντας τότε τις μεταβλητές

$$x_{ij}^k = \begin{cases} 1, & \text{εάν το όχημα καταξιδεύει απευθείας απο το } i \text{ στο } j \\ 0, & \text{αλλιώς, } \forall i, \forall j, i \neq j, k = 1, \dots, K \end{cases}$$

y_i^k = η ποσότητα από τις απομένουσες παραδόσεις που διεκπεραιώνονται από το όχημα k όταν αυτό εκκινεί από τον πελάτη i , $\forall i \in N$

z_i^k = η ποσότητα από τις απομένουσες παραλαβές που διεκπεραιώνονται από το όχημα k όταν αυτό εκκινεί από τον πελάτη i , $\forall i \in N$

t_i^k = η χρονική αρχή για την εξυπηρέτηση του πελάτη i , από το όχημα $k \forall i \in N$, $\forall k \in K$

Τότε το μαθηματικό πρότυπο διαμορφώνεται ως εξής:

$$\text{Min } \sum_{k=1}^K \sum_{i=0}^{n+1} \sum_{j=0}^{n+1} c_{ij} x_{ij}^k \quad (3.13)$$

Υπό

$$\sum_{k=1}^K \sum_{j=0}^{n+1} x_{ij}^k = 1, \quad i=1, \dots, n \quad (3.14)$$

$$\sum_{i=0}^{n+1} x_{il}^k - \sum_{j=0}^{n+1} x_{lj}^k = 0, \quad l=1, \dots, n, \quad \forall k \in K \quad (3.15)$$

$$\sum_{j=1}^n x_{0j}^k \leq 1, \quad k \in K \quad (3.16)$$

$$\sum_{i=1}^n x_{i,n+1}^k - \sum_{j=1}^n x_{0j}^k = 0, \quad \forall k \in K \quad (3.17)$$

$$y_i^k + z_i^k \leq u, \quad i=1, \dots, n, \quad \forall k \in K \quad (3.18)$$

$$y_{n+1}^k = 0, \quad \forall k \in K \quad (3.19)$$

$$y_0^k = \sum_{i=1}^n \sum_{j=1}^n d_i x_{ij}^k, \quad \forall k \in K \quad (3.20)$$

$$z_{n+1}^k = \sum_{i=1}^n \sum_{j=1}^n p_i x_{ij}^k, \quad \forall k \in K \quad (3.21)$$

$$z_0^k=0, \quad \forall k \quad (3.22)$$

$$x_{ij}^k(z_i^k + p_j - z_j^k)=0, \quad \forall i, \forall j, i \neq j, \forall k \quad (3.23)$$

$$x_{ij}^k(y_i^k - d_j - y_j^k)=0, \quad \forall i, \forall j, i \neq j, \forall k \quad (3.24)$$

$$x_{ij}^k(t_i^k + t_{ij} - t_j^k) \leq 0, \quad i, j \neq j, \forall k \quad (3.25)$$

$$a_i \leq t_i^k \leq b_i, \quad i, k \quad (3.26)$$

$$y_i^k \geq 0, \quad i, k \quad (3.27)$$

$$z_i^k \geq 0, \quad i, k \quad (3.28)$$

$$x_{ij}^k \in \{0,1\}, \quad \forall j \forall i \neq j, k \quad (3.29)$$

Η αντικειμενική συνάρτηση (3.13) ελαχιστοποιεί το ολικό κόστος των διαδρομών. Οι περιορισμοί (3.14) επιβάλλουν την εξυπηρέτηση κάθε πελάτη από ακριβώς ένα όχημα, ενώ οι (3.15) εγγυώνται ότι το όχημα που εισέρχεται και εξέρχεται από κάθε κόμβο είναι το ίδιο.

Οι περιορισμοί (3.16) και (3.17) επιβάλλουν τη χρήση κάθε οχήματος το πολύ μία φορά. Οι περιορισμοί (3.15), (3.16) και (3.17) αντιστοιχούν στην ισορροπία ροής και απαιτούν ότι κάθε όχημα εκκινεί από τον κόμβο αφετηρίας (κόμβος 0) το πολύ μία φορά, ότι εξέρχεται από έναν κόμβο 1 μόνο εάν έχει εισέλθει σε αυτόν και τέλος επιστρέφει στο αρχικό σημείο εκκίνησης (κόμβος n+1) το πολύ μία φορά. Οι περιορισμοί (3.18) εγγυώνται ότι το φορτίο ενός οχήματος, όταν αυτό εκκινεί από έναν πελάτη, δεν υπερβαίνει την δυναμικότητα του.

Οι περιορισμοί (3.19) και (3.21) εγγυώνται ότι όταν ένα όχημα εκκινεί από το κέντρο έχει όλο το αναγκαίο φορτίο για τις παραδόσεις στους πελάτες που εξυπηρετεί και το φορτίο παραλαβής του είναι μηδενικό. Οι περιορισμοί (3.20) και (3.21) εγγυώνται ότι όταν το όχημα επιστρέφει στην αφετηρία έχει επιτελέσει όλες τις διανομές και έχει παραλάβει όλες τις ποσότητες που του αντιστοιχούν. Οι μη-γραμμικοί περιορισμοί (3.22)-(3.23) επιβάλλουν την μείωση του φορτίου παράδοσης ενός οχήματος k με d_j μονάδες όταν αυτό έχει κινηθεί απευθείας από το i στο j και αντίστοιχα την αύξηση του φορτίου παραλαβής με p_j μονάδες, διότι η κίνηση αυτή επιφέρει την εξυπηρέτηση του πελάτη j.

Οι μη-γραμμικοί περιορισμοί (3.24) συνεπάγονται ότι εάν το όχημα k κινηθεί απευθείας από το i στο j τότε ο πελάτης j αρχίζει να εξυπηρετείται κάποια χρονική στιγμή που είναι ίση ή μεγαλύτερη της χρονικής στιγμής που άρχισε η εξυπηρέτηση του πελάτη i συν το χρόνο που χρειάστηκε το όχημα για το απευθείας ταξίδι από το i στο j . Οι περιορισμοί αυτοί επιτρέπουν το όχημα να παραμείνει σε αναμονή σε κάθε πελάτη μέχρι να έρθει η καθορισμένη από τον πελάτη στιγμή έναρξης του χρονικού διαστήματος εξυπηρέτησης. Οι περιορισμοί (3.26) επιβάλλουν την έναρξη εξυπηρέτησης να ανήκει χρονικά στα πλαίσια που έχει καθορίσει ο πελάτης. Τέλος οι περιορισμοί (3.27)-(3.29) είναι αυτοί που επιβάλλουν τη μη-αρνητικότητα και την δυαδικότητα των αντίστοιχων μεταβλητών.

3.2.4 Το μικτό πρόβλημα δρομολόγησης οχημάτων με δύο είδη πελατών κατά τη διάρκεια της διαδρομής και πολλές αποθήκες (TheMultipleDepotMixedVehicleRoutingProblemwithbackhauls -MDMVRPB)

Το μικτό πρόβλημα δρομολόγησης οχημάτων με δύο είδη πελατών κατά τη διάρκεια της διαδρομής και πολλές αποθήκες είναι μια γενίκευση του MVRPB. Στο MDVRPB χαλαρώνουμε τον περιορισμό των αποθηκών έτσι ώστε αντί να εξετάζουμε απλώς μια αποθήκη αντιμετωπίζουμε το πρόβλημα των πολλών αποθηκών. Σε κάθε αποθήκη υπάρχει διαθέσιμος περιορισμένος στόλος οχημάτων και κάθε όχημα πρέπει να ξεκινήσει και να τερματίσει τη διαδρομή του στην ίδια αποθήκη. Οι ευρετικοί αλγόριθμοι για το πρόβλημα προτείνονται από τους Nagy και Salhi

Κεφάλαιο 4ο

4.1 Αλγόριθμοι και το πρόβλημα δρομολόγησης οχημάτων

Η επίλυση ενός προβλήματος συνδυαστικής βελτιστοποίησης γίνεται ολοένα και δυσκολότερη όσο αυξάνεται το μέγεθος του προβλήματος και πολλές φορές το να προσπαθούμε να βρούμε την ολικά βέλτιστη λύση σε λογικό χρόνο είναι πρακτικά αδύνατο. Για να επιλύσουμε προβλήματα αυτής της μορφής συχνά καταφεύγουμε σε διαφορετικές τεχνικές που μας οδηγούν σε μία σχεδόν βέλτιστη, αλλά ικανοποιητική λύση. Μια λύση ενός ευρετικού αλγορίθμου γίνεται αποδεκτή εάν ικανοποιεί κάποια κριτήρια όπως η ποιότητα της λύσης, δηλαδή η απόκλιση της από την βέλτιστη, η ευκολία απόκτησης μίας λύσης, η λογική πάνω στην οποία στηρίζονται οι κανόνες του ευρετικού αλγορίθμου που χρησιμοποιήθηκαν για να οδηγηθούμε στη λύση. Σε αυτό το κεφάλαιο θα μιλήσουμε για διάφορες κατηγορίες ευρετικών αλγορίθμων. Η κάθε μια από αυτές έχει κάποια ιδιαίτερα χαρακτηριστικά. Οι κατηγορίες των αλγορίθμων αυτών είναι οι ακόλουθες:

- Αλγόριθμοι απληστίας (greedy algorithms)
- Προσεγγιστικοί αλγόριθμοι (approximation algorithms)
- Αλγόριθμοι τοπικής αναζήτησης (local search algorithms)

Συνοπτικά μπορούμε να πούμε ότι οι αλγόριθμοι απληστίας προσπαθούν να οδηγήσουν σε μία εφικτή λύση του προβλήματος, αλλά πολλές φορές χρειάζονται πάρα πολύ μεγάλο χρόνο γιατί είναι μυωπικοί αλγόριθμοι, δηλαδή βλέπουν μόνο μπροστά. Οι προσεγγιστικοί αλγόριθμοι προσπαθούν να λύσουν αυτό το πρόβλημα χρησιμοποιώντας επιπλέον πληροφορία. Τέλος, οι αλγόριθμοι τοπικής αναζήτησης προσπαθούν από μία αρχική εφικτή λύση να βελτιώσουν τη λύση με κάποια μέθοδο αναζήτησης.

Σύμφωνα με τον Bodin και τον Golden το 1981 οι στρατηγικές επίλυσης για τα προβλήματα δρομολόγησης οχημάτων (VRP) ταξινομούνται στις παρακάτω προσεγγίσεις:

1. Ομαδοποίηση πρώτα - δρομολόγηση έπειτα (Clusterfirst – routessecond)
2. Δρομολόγηση πρώτα – ομαδοποίηση έπειτα (Routefirst – clustersecond)
3. Εξοικονομήσεις / Καταχώρηση (Savings/Insertion)
4. Βελτίωση ή ανταλλαγή (Improvement or exchange)
5. Προσέγγιση μαθηματικού προγραμματισμού

(Mathematicalprogrammingapproach)

6. Αλληλεπιδρών βελτιστοποίηση (Interactive optimization)
7. Ακριβής διαδικασία (Exact procedure)

Στη συνέχεια παρουσιάζουμε πιο αναλυτικά ορισμένα χαρακτηριστικά των προσεγγίσεων αυτών

1. Ομαδοποίηση πρώτα – δρομολόγηση έπειτα: Ομαδοποιούμε πρώτα τους κόμβους και / ή τα τόξα ζήτησης και σχεδιάζουμε οικονομικές διαδρομές για κάθε ομάδα σαν δεύτερο στάδιο. Η διαμόρφωση στρατηγικής αυτού του αλγορίθμου αποσκοπεί στο να διαμορφώσει εκείνος τα σημεία της διαδρομής έτσι ώστε το καθένα από τα οχήματα να μην υπερβαίνει τα όρια της χωρητικότητάς του που έχουν τεθεί.
2. Δρομολόγηση πρώτα – ομαδοποίηση έπειτα: Μια μεγάλη (συνήθως μη εφικτή) διαδρομή ή κύκλος κατασκευάζεται που περιλαμβάνει όλες τις οντότητες ζήτησης. Έπειτα η μεγάλη διαδρομή διαμερίζεται σε έναν αριθμό από μικρότερες αλλά εφικτές διαδρομές. Η μέθοδος αυτή επιλύει το πρόβλημα του πλανόδιου πωλητή όπου όλα τα σημεία σχηματίζουν μία μεγάλη διαδρομή. Στη συνέχεια αυτή η διαδρομή χωρίζεται σε μικρότερες διαδρομές ώστε η κάθε μία να ικανοποιεί τους περιορισμούς του κάθε οχήματος. Οι εφαρμογές αυτής της προσέγγισης δίνονται από τους NewtonandThomas (1969) και τους Bodin και Berman (1979) για τη δρομολόγηση σχολικών λεωφορείων. Επέκταση της προσέγγισης της μεθόδου αυτής ως προς το VRPB δεν φαίνεται εφικτή λόγω παραβίασης των περιορισμών προτεραιότητας.

3. Εξοικονομήσεις / καταχώρηση: Στις διαδικασίες εναποθήκευσης ή καταχώρησης κατασκευάζεται μια λύση με τέτοιο τρόπο όπου σε κάθε βήμα της διαδικασίας ένας τρέχων σχηματισμός που πιθανώς να είναι μη εφικτός συγκρίνεται με έναν εναλλακτικό σχηματισμό που μπορεί επίσης να είναι μη εφικτός. Ο εναλλακτικός σχηματισμός είναι αυτός που αποδίδει τις μεγαλύτερες εξοικονομήσεις σε όρους κάποιας συνάρτησης κριτηρίου, όπως του ολικού κόστους, ή καταχωρεί με το ελάχιστο κόστος από μια οντότητα ζήτησης όχι στον τρέχων σχηματισμό μέσα στην υπάρχουσα διαδρομή ή στις υπάρχουσες διαδρομές. Η διαδικασία ολοκληρώνεται με έναν εφικτό σχηματισμό. Ίσως ο πιο γνωστός αλγόριθμος εξοικονόμησης/καταχώρησης αναπτύχθηκε από την Clarke και Wright (1964). Οι Deif και Bodin (1984) έχουν προτείνει την επέκταση αυτού του αλγορίθμου για το VRPB. Η διαδικασία τους βασίζεται σε δύο τροποποιήσεις του αλγορίθμου Clarke-Wright. Η πρώτη τροποποίηση προσθέτει τον περιορισμό ότι μόνο ένας σύνδεσμος από linehaul σε backhaul (ή αντίστροφα) επιτρέπεται σε κάθε διαδρομή και η δεύτερη τροποποίηση συμπεριέλαβε ποινή λόγω της μείωσης του μεγέθους της εξοικονόμησης για τη μετάβαση από linehaul σε backhaul έτσι ώστε να μην καθυστερεί τη σύνδεση.
4. Βελτίωση ή ανταλλαγή : Οι διαδικασίες βελτίωσης ή ανταλλαγής, ίσως η καλύτερη γνωστή μέθοδος είναι ο r-opt αλγόριθμος των LinandKernigham, πάντα διατηρούν την εφικτότητα τους και οδηγούν στην καλύτερη δυνατή λύση. Σε κάθε βήμα μια εφικτή λύση βελτιώνεται (ανταλλάσσοντας κάποιες ακμές) για να οδηγήσει σε μια άλλη εφικτή λύση με μειωμένο το συνολικό κόστος. Αυτή η διαδικασία ακολουθείται μέχρι να μην υπάρχουν μειώσεις στα επιπρόσθετα κόστη. Οι Χριστοφίδης και Έιλον (1972) ανέπτυξαν έναν αλγόριθμο VRP μετασχηματίζοντας το πρόβλημα σε ένα πρόβλημα πλανόδιου πωλητή, εφαρμόζοντας 2-opt και 3-opt. Οι τεχνικές βελτιστοποίησης βασίζονται στην ανάπτυξη ενός αλγορίθμου επίλυσης βασισμένου σε μία μαθηματική διατύπωση του προβλήματος. Ο σχεδιασμός βελτιστοποίησης της ευρετικής διαδικασίας περιλαμβάνει την ανάπτυξη ενός μαθηματικού μοντέλου του προβλήματος παρατηρώντας την εξειδικευμένη δομή του μοντέλου, χρησιμοποιώντας μια στρατηγική βελτιστοποίησης όπως η αποσύνθεση της δομής και ίσως να ενισχύσει την διαδικασία επιτρέποντας την είσοδο από το χρήστη.

5. Προσέγγιση μαθηματικού προγραμματισμού: Η προσέγγιση του μαθηματικού προγραμματισμού περιλαμβάνει αλγορίθμους που βασίζονται ευθέως σε μορφοποίηση μαθηματικού προγραμματισμού του θεμελιώδους προβλήματος δρομολόγησης.
6. Αλληλεπιδρών βελτιστοποίηση: Είναι μια γενικού σκοπού προσέγγιση στην οποία ένας υψηλός βαθμός αλληλεπιδρώντων ατόμων ενσωματώνονται στην διαδικασία επίλυσης του προβλήματος. Η ιδέα είναι ότι ο έμπειρος αποφασίζων μπορεί να έχει την ικανότητα της τοποθέτησης και αναθεώρησης παραμέτρων και να δίνει αντικειμενικές εκτιμήσεις βασισμένος στη γνώση και την διαίσθηση του μοντέλου της βελτιστοποίησης. Αυτό σχεδόν πάντα αυξάνει την πιθανότητα ότι το μοντέλο θα υλοποιηθεί και θα χρησιμοποιηθεί αποτελεσματικά.
7. Ακριβής διαδικασία: Οι ακριβείς διαδικασίες για την επίλυση προβλημάτων δρομολόγησης περιλαμβάνουν εξειδικευμένους αλγορίθμους branchandbound και cuttingplanes.

Ευρετικοί αλγόριθμοι

Οι ευρετικοί αλγόριθμοι διακρίνονται σε κατασκευαστικούς και σε βελτιωτικούς. Με τους κατασκευαστικούς αλγορίθμους δημιουργείται μια διαδρομή. Η διαδρομή που θα κατασκευαστεί είναι αποτέλεσμα της ελαχιστοποίησης ενός κριτηρίου σύμφωνα με το οποίο μπορεί να κάνει είτε ένα όχημα δύο διαδοχικά δρομολόγια είτε δύο οχήματα να κινηθούν παράλληλα. Μια γνωστή μέθοδος των κατασκευαστικών αλγορίθμων είναι αυτή του πλησιέστερου γείτονά.

Με τη μέθοδο αυτή μειώνεται η χρονική διάρκεια ολοκλήρωσης της διαδρομής και οι αποστάσεις των σημείων που ακολουθεί το ένα το άλλο καθώς επίσης μελετάται και πιο σημείο θα είναι της παραλαβής και πιο σημείο της παράδοσης. Για να μπορέσει μια λύση να είναι ικανοποιητική θα πρέπει ο χρόνος ολοκλήρωσης του αλγορίθμου καθώς επίσης και οι χωρητικότητες των οχημάτων να μην υπερβαίνουν τα όρια που έχουν τεθεί. Με την εφαρμογή της μεθόδου αυτής δημιουργούνται δρομολόγια στα οποία περιλαμβάνονται όλοι οι κόμβοι του προβλήματος.

Γνωστοί κατασκευαστικοί αλγόριθμοι είναι των ClarkandWright και ο αλγόριθμος που δημιουργήθηκε από τον Solomon σύμφωνα με τον οποίο η διαδρομή ξεκινά από το κεντρικό σημείο όλων των διαδρομών. Για να εισαχθούν νέα σημεία σε ένα δρομολόγιο θα πρέπει να εξυπηρετείται από ένα όχημα έναντι άλλων.

Στους βελτιωτικούς ευρετικούς αλγόριθμους ανήκει η κατηγορία των αλγορίθμων, οι οποίοι βασίζονται στην τοπική αναζήτηση από όπου ξεκινά μία εφικτή λύση και ακολουθεί μια μέθοδος αναζήτησης για την επόμενη λύση. Οι βελτιωτικοί αλγόριθμοι χαρακτηρίζονται για την ταχύτητα και την αποτελεσματικότητά τους.

Ο σχεδιασμός και η πραγματοποίηση ενός βελτιωτικού αλγορίθμου πραγματοποιείται με τα εξής βήματα:

- Προσδιορίζεται ο τρόπος που θα δημιουργηθεί και θα σχεδιαστεί για να εντοπιστεί μια λύση.
- Παρουσιάζονται οι μηχανισμοί οι οποίοι θα διαφοροποιήσουν τη λύση.
- Προσδιορίζεται το κριτήριο με το οποίο θα γίνει αποδεκτή μία λύση.
- Προσδιορίζεται το κριτήριο με το οποίο θα επιλεγεί η βέλτιστη λύση.

Βασικοί τελεστές ανταλλαγής χαρακτηρίζονται οι εξής τρεις:

Relocate: Η επανατοποθέτηση η οποία πραγματοποιείται όταν γίνεται μεταφορά ενός σημείου από ένα δρομολόγιο σε κάποιο άλλο χωρίς να πραγματοποιείται αλλαγή στον συντονισμό τους.

Exchange: Πρόκειται για την περίπτωση όπου δύο σημεία δύο διαδρομών ανταλλάσσονται χωρίς όμως να μεταβάλλεται η επίλυση του προβλήματος.

Cross: Πρόκειται για τη διασταύρωση ζευγών διαδοχικών σημείων χωρίς να αλλάζει κατεύθυνση το δρομολόγιο.

4.1.1a Το πρόβλημα δρομολόγησης οχημάτων – Ο αλγόριθμος των εξοικονομήσεων των Clarke&Wright (Thesavingsalgorithm)

Ο πρώτος κατασκευαστικός αλγόριθμος που έχει προταθεί για την επίλυση του προβλήματος δρομολόγησης οχημάτων είναι ο αλγόριθμος των εξοικονομήσεων των Clarke&Wright. Σε αυτή την μέθοδο αρχικά υπολογίζονται οι εξοικονομήσεις όλων των πελατών και στην συνέχεια δημιουργούνται διαδρομές βάση των καλύτερων εξοικονομήσεων. Η διαδικασία έχει δύο εκδοχές την παράλληλη και την ακολουθητική. Η διαδικασία έχει την παρακάτω μορφή:

Βήμα 1. Υπολογισμός των εξοικονομήσεων (savings) $s_{ij}=c_{1i}-c_{ij}+c_{j1}$ για όλα τα ζεύγη των πελατών i και j . Παρατηρούμε ότι s_{ij} είναι η εξοικονόμηση σε κόστος αν συνδέοντας τις κορυφές i και j παράγουμε την διαδρομή $(1,i,j,1)$ αντί να έχουμε τα i και j σε δύο διαδρομές $(1,i,1)$ και $(1,j,1)$.

Βήμα 2. Κατατάσσουμε τις εξοικονομήσεις σε φθίνουσα σειρά.

Βήμα 3. Ξεκινώντας από την κορυφή της λίστας, κάνουμε τα ακόλουθα :

Παράλληλη εκδοχή

Βήμα 4. Αν μετά τη δημιουργία ενός δεσμού, δημιουργείται μια εφικτή διαδρομή, ικανοποιώντας τους περιορισμούς του VRP, δεχόμαστε τον δεσμό στην λύση, αλλιώς την απορρίπτουμε.

Βήμα 5. Εξετάζουμε τον επόμενο δεσμό στην λίστα και επαναλαμβάνουμε το βήμα 4 έως ότου να μην μπορεί να επιλεγεί άλλος δεσμός.

Ακολουθητική εκδοχή

Βήμα 4. Βρίσκουμε τον πρώτο εφικτό δεσμό στην λίστα που μπορεί να χρησιμοποιηθεί για να προεκτείνει ένα από τα δύο άκρα της τρέχουσας διαδρομής.

Βήμα 5. Εάν η διαδρομή δεν μπορεί να επεκταθεί επιπλέον, ολοκληρώνουμε την διαδρομή. Επιλέγουμε τον πρώτο εφικτό δεσμό στην λίστα για να ξεκινήσουμε μια καινούρια διαδρομή.

Βήμα 6. Επαναλαμβάνουμε τα βήματα 4 και 5 μέχρι κανένας δεσμός να μην μπορεί να επιλεγεί.

Και στις δύο μορφές αυτής της διαδικασίας, πρέπει να ελέγχεται η εφικτότητα της μερικής λύσης σε κάθε βήμα, για να βεβαιωθούμε ότι δεν παραβιάζεται κανένας περιορισμός. Αλλιώς, είναι προφανές ότι καμία εφικτή λύση δεν θα έχει βρεθεί στο πρόβλημα. Παρατηρούμε επίσης, ότι στην αρχική λύση όταν όλοι οι πελάτες είναι σε διαφορετική διαδρομή η λύση δεν είναι εφικτή. Όμως, στο τέλος της διαδικασίας υπάρχει η πιθανότητα κάποιοι πελάτες να μείνουν χωρίς διαδρομή.

4.1.1β Το πρόβλημα δρομολόγησης οχημάτων – Μέθοδοι δύο φάσεων

Στις μεθόδους δύο φάσεων, η πρώτη αποτελείται από ομαδοποίηση των πελατών εκχωρώντας τους σε οχήματα, και στην δεύτερη φάση η δρομολόγηση αυτής της διαδρομής. Ο αλγόριθμος σαρώματος των Gillet&Miller. Τόσο η πρώτη όσο και η δεύτερη φάση αυτής της διαδικασίας είναι ακολουθητική. Υποθέτουμε ότι οι αποστάσεις μεταξύ των κορυφών είναι ευκλείδειες και ότι οι πελάτες τοποθετούνται βάση των πολικών συντεταγμένων (r_i, θ_i) με την αποθήκη να είναι στο $r_i=0$ και ένας τυχαίος πελάτης i^* και θ_{i^*} , Στη συνέχεια κατατάσσουμε τους πελάτες έτσι ώστε $\theta_2 \leq \dots \leq \theta_n$.

Φάση 1

Βήμα 1. Επέλεξε ένα αχρησιμοποίητο όχημα k .

Βήμα 2. Ξεκινώντας από τον πελάτη i , που δεν είναι στη διαδρομή, με το μικρότερο κόστος θ_i και συμπεριέλαβε τους γειτονικούς του πελάτες στην διαδρομή μέχρι να παραβιαστούν οι περιορισμοί χωρητικότητας του οχήματος.

Βήμα 3. Αν όλοι οι πελάτες έχουν εκχωρηθεί ή όλα τα οχήματα έχουν γεμίσει πήγαινε στη Φάση 2, αλλιώς γύρνα στο βήμα 1.

Φάση 2

Βήμα 4. Λύνουμε το TSP για κάθε σύνολο πελατών που έχουν εκχωρηθεί σε κάποιο όχημα και μορφοποιούμε τις τελικές διαδρομές.

4.1.1γ Η μέθοδος των δύο φάσεων των Fisher&Jaikumar.

Αυτή η μέθοδος είναι ίσως η πιο διάσημη ομαδοποίηση πρώτα – δρομολόγηση έπειτα μέθοδος. Η πρώτη φάση αυτής της ευρετικής διαδικασίας βρίσκει μια ανάθεση των πελατών στα οχήματα λύνοντας βέλτιστα ένα γενικευμένο πρόβλημα εκχώρησης (Generalized Assignment Problem - GAP) με αντικειμενική συνάρτηση να είναι μια προσέγγιση του κόστους διανομής. Στην δεύτερη φάση του αλγορίθμου επιλύεται ένα πρόβλημα πλανόδιου πωλητή για κάθε ένα όχημα.

Φάση 1

Βήμα 1. Διαλέγουμε m πελάτες για να είναι οι αρχικοί πελάτες των ομάδων και τοποθετούμε ένα όχημα για τον κάθε ένα από αυτούς.

Βήμα 2. Για κάθε πελάτη i και για κάθε ομάδα k , υπολογίζουμε ένα κόστος εισόδου d_{ik} σε σχέση με τον αρχικό πελάτη της κάθε ομάδας.

Βήμα 3. Επιλύουμε το γενικευμένο πρόβλημα εκχώρησης.

Φάση 2

Βήμα 4. Λύνουμε το πρόβλημα του πλανόδιου πωλητή (TSP) για κάθε σύνολο πελατών που έχουν εκχωρηθεί σε κάποιο όχημα και μορφοποιούμε τις τελικές διαδρομές.

4.1.1δ Η μέθοδος των δύο φάσεων των Mole&Jameson.

Ο αλγόριθμος των Mole&Jameson είναι μια ακολουθητική διαδικασία στην οποία για μια δεδομένη τιμή των δύο παραμέτρων λ και μ , τα ακόλουθα δύο κριτήρια χρησιμοποιούνται για την κατασκευή μιας διαδρομής :

$$e(i,l,j)=c_{il} + c_{lj} - \mu c_{ij}$$

$$\sigma(i,l,j)=\lambda c_{0l}-e(i,l,j)$$

Ο αλγόριθμος έχει ως εξής:

Βήμα 1. Για κάθε μη δρομολογημένο πελάτη x_i υπολογίζουμε σε ποιο σημείο της διαδρομής μπορεί να εισαχθεί ο πελάτης: $e(i_l,l,j_l)=\min[e(r,l,s)]$ για όλους τους γειτονικούς πελάτες $x_r, x_s \in R$, όπου x_{il} και x_{jl} είναι οι πελάτες για τους οποίους η εισαγωγή του πελάτη x_i έχει την καλύτερη δυνατή τιμή.

Βήμα 2. Ο καλύτερος πελάτης $x_{i_l^*}$ που θα εισαχθεί στην διαδρομή υπολογίζεται να είναι εκείνος για τον οποίο η ακόλουθη έκφραση μεγιστοποιείται:

$$\sigma(i_l^*,l^*,j_l^*)=\max [\sigma(i_l,l,j_l)]$$

για τον κάθε πελάτη x_i που δεν βρίσκεται είδη σε κάποια διαδρομή και που δεν παραβιάζει κάποιο περιορισμό.

Βήμα 3. Εισάγουμε $x_{i_l^*}$ στην διαδρομή R ανάμεσα στον x_{il^*} και στον x_{jl^*} .

Βήμα 4. Βελτιστοποιούμε την διαδρομή R χρησιμοποιώντας μεθόδους $r - opt$.

Βήμα 5. Επιστρέφουμε στο βήμα 1 για να ξεκινήσουμε μια καινούρια διαδρομή R εκτός αν όλοι οι πελάτες έχουν δρομολογηθεί ή κανένας από τους εναπομείναντες πελάτες δεν μπορεί να δρομολογηθεί.

4.1.1ε Η μέθοδος των δύο φάσεων των Christofides, Mignozzi&Toth

Η πρώτη φάση αυτού του ευρετικού αλγορίθμου αποτελείται από την εφαρμογή ενός αριθμού δοκιμών για την ομαδοποίηση χρησιμοποιώντας ένα κριτήριο εκχώρησης ελαχίστου κόστους.

Φάση 1

Βήμα 1.(Ακολουθητικές δοκιμές) Επιλέγουμε έναν πελάτη που δεν είναι στη διαδρομή για πελάτη σπόρο. Διαλέγουμε ένα όχημα για τον πελάτη.

Βήμα 2. Βάζουμε τους πελάτες που δεν είναι σε διαδρομή σε κάποια ομάδα, βάση κάποιου κόστους εκχώρησης που σχετίζεται με τον πελάτη της κάθε ομάδας, μέχρις ότου να παραβιαστεί ο περιορισμός χωρητικότητας του κάθε οχήματος. Αν όλοι οι πελάτες είναι σε ομάδες ή όλα τα οχήματα έχουν χρησιμοποιηθεί, πηγαίνουμε στο βήμα 3, αλλιώς επιστρέφουμε στο βήμα 1.

Βήμα 3. (Παράλληλες δοκιμές) Χρησιμοποιώντας όλους τους αρχικούς πελάτες για μια διαδρομή που βρέθηκαν στις ακολουθητικές δοκιμές, απελευθερώνουμε όλους τους πελάτες από τις ομάδες τους.

Βήμα 4. Για κάθε ελεύθερο πελάτη, υπολογίζουμε το κόστος εισόδου σε σχέση με τους αρχικούς πελάτες της κάθε ομάδας, σε μια εφικτή ομάδα. Ελέγχουμε όλες τις ομάδες και στη συνέχεια κρατάμε αυτή με το καλύτερο κόστος για τον πελάτη.

Βήμα 5. Εκχωρούμε τον πελάτη με το καλύτερο κόστος στην ομάδα που αντιστοιχεί.

Βήμα 6. Επαναλαμβάνουμε τα βήματα 4 και 5 για όλους τους ελεύθερους πελάτες.

Φάση 2

Βήμα 7. Και για τις δύο ομαδοποιήσεις λύνουμε ένα πρόβλημα πλανόδιου πωλητή (TSP) και κρατάμε την καλύτερη από τις δύο λύσεις.

4.1.2 Αλγόριθμοι τοπικής αναζήτησης (Local search algorithms)

Η τοπική αναζήτηση βασίζεται στην αρχαιότερη μέθοδο βελτιστοποίησης, την μέθοδο δοκιμής και σφάλματος. Ένας γενικός αλγόριθμος είναι ο εξής:

Έχουμε ένα πρόβλημα βελτιστοποίησης (F, c) όπου F είναι ένα εφικτό σύνολο και c είναι το κόστος, επιλέγουμε τη γειτονία

$$N:F \rightarrow 2^F$$

Στην οποία εφαρμόζεται αναζήτηση για να βρεθεί καλύτερη λύση από την αρχική αν υπάρχει κάποιο σημείο $S \in N$ που να έχει μικρότερο κόστος από το κόστος που έχουμε αυτή τη στιγμή. Για όσο χρονικό διάστημα βρίσκεται μία καλύτερη λύση, αντικαθιστούμε την υπάρχουσα λύση με την καινούρια και συνεχίζουμε την αναζήτηση μέχρι το σημείο που θα βρούμε κάποιο τοπικό ελάχιστο και η λύση που έχουμε δεν θα βελτιώνεται παραπάνω. Στη συνέχεια περιγράφεται ένας πολύ γενικός αλγόριθμος τοπικής αναζήτησης. Στον αλγόριθμο ξεκινάμε από μια εφικτή αρχική λύση $t \in F$ και χρησιμοποιούμε την υπορουτίνα `improve` (i) για να ψάξουμε για μια καλύτερη λύση στη γειτονιά της αρχικής λύσης. Όσο βρίσκουμε μια βελτιωμένη λύση, την εφαρμόζουμε και επαναλαμβάνουμε την διαδικασία αναζήτησης από τη νέα λύση. Όταν φτάσουμε στο τοπικό ελάχιστο σταματάμε.

Διαδικασία `local_search`:

- ✧ `Begin`
- ✧ `t` μια αρχική λύση του προβλήματος
- ✧ `Dowhile` βρίσκεται μια βελτιωμένη λύση (`improve(t)`)
- ✧ `t=improve(t)`
- ✧ `end`

Ένα πολύ σημαντικό χαρακτηριστικό της τοπικής αναζήτησης είναι το γεγονός ότι μπορεί να εκτελείται από πολλά διαφορετικά αρχικά σημεία και να επιλέγεται το καλύτερο σαν βέλτιστη λύση. Σε αυτές τις περιπτώσεις μια άλλη απόφαση που πρέπει να πάρουμε είναι πόσα θα είναι τα αρχικά σημεία που θα

πρέπει να επιλέξουμε. Το επόμενο πρόβλημα που πρέπει να επιλυθεί για να είμαστε σε θέση να πούμε ότι η τοπική αναζήτηση που εφαρμόζουμε οδηγεί σε ικανοποιητικά αποτελέσματα είναι το ότι θα πρέπει να γίνει σωστή επιλογή της γειτονίας που θα πραγματοποιηθεί η αναζήτηση.

Τέλος το σημαντικότερο ίσως στοιχείο για την επιτυχία της διαδικασίας είναι η επιλογή της μεθόδου που θα χρησιμοποιηθεί για την αναζήτηση. Στην βιβλιογραφία του προβλήματος υπάρχουν αρκετές μέθοδοι που έχουν εφαρμοστεί για την επίλυση του προβλήματος, οι σημαντικότερες εκ των οποίων είναι η μέθοδος 2-opt, η μέθοδος 3-opt, η περιορισμένη αναζήτηση και ο αλγόριθμος των Lin- Kernigham.

Τα σημαντικότερα προβλήματα που πρέπει να αντιμετωπιστούν κατά την διάρκεια της φάσης της σχεδίασης ενός αλγορίθμου τοπικής αναζήτησης είναι:

- Το πρώτο, όπως είπαμε και παραπάνω, είναι η επιλογή της γειτονίας, και αυτό σχετίζεται με την ανησυχία που υπάρχει για το αν είναι ή όχι εφικτή η λύση.
- Το δεύτερο πολύ σημαντικό στοιχείο που πρέπει να αντιμετωπίσουμε είναι η ποιότητα της αρχικής λύσης. Είναι ένα πολύ σημαντικό στοιχείο, γιατί όσο καλύτερη είναι η αρχική λύση τόσο περισσότερες πιθανότητες υπάρχουν να οδηγηθούμε ευκολότερα και γρηγορότερα σε βελτίωση της λύσης με την χρήση της τοπικής αναζήτησης.
- Το τρίτο και σημαντικότερο στοιχείο για την επιτυχία του αλγορίθμου είναι η μέθοδος που θα χρησιμοποιηθεί για να βελτιώσει την αρχική λύση. Οι περισσότερες μέθοδοι που έχουν αναπτυχθεί στην βιβλιογραφία παρουσιάζουν σημαντικά προβλήματα όσον αφορά την ποιότητα της λύσης που επιστρέφουν και αυτό γιατί αν κάποια στιγμή πέσουν σε τοπικό ελάχιστο είναι πολύ δύσκολο να ξεκολλήσουν από εκεί. Από τις μεθόδους που έχουν αναπτυχθεί και που αποφεύγουν εύκολα την παγίδα του κολλήματος σε τοπικά ελάχιστα είναι η μέθοδος Lin-Kernigham.

4.1.2α Το πρόβλημα δρομολόγησης οχημάτων και οι αλγόριθμοι τοπικής αναζήτησης

Στο πρόβλημα δρομολόγησης οχημάτων υπάρχουν δύο είδη αποφάσεων, οι αποφάσεις δρομολόγησης των οχημάτων και οι αποφάσεις ανάθεσης πελατών στα οχήματα. Οι αλγόριθμοι τοπικής αναζήτησης χωρίζονται και αυτοί σε αλγόριθμους τοπικής αναζήτησης δύο φάσεων. Στους αλγόριθμους που προσπαθούν να βελτιώσουν τη δρομολόγηση των οχημάτων σε μία διαδρομή, οι οποίοι είναι στην ουσία ίδιοι με τους αλγορίθμους που εφαρμόζονται στο πρόβλημα του πλανόδιου πωλητή και στους αλγορίθμους που προσπαθούν να βελτιώσουν την ανάθεση των πελατών σε οχήματα και οι οποίοι εφαρμόζονται μεταξύ 2 ή και περισσότερων διαδρομών.

Ο Waters(21) πρότεινε έναν αριθμό από διαδικασίες ανταλλαγής που μπορούν να εφαρμοστούν μεταξύ δύο ή περισσότερων διαδρομών. Πρότεινε τέσσερις διαφορετικές διαδικασίες τοπικής αναζήτησης :

1. Μια απλή διαγραφή ενός πελάτη από μια διαδρομή και επανατοποθέτηση του σε μία άλλη διαδρομή με καλύτερο κόστος. Η μέθοδος αυτή ονομάζεται 1-0 επανατοποθέτηση (1-0 relocate).
2. Μια ταυτόχρονη ανταλλαγή 2 πελατών που βρίσκονται σε διαφορετικές διαδρομές. Η μέθοδος αυτή ονομάζεται 1-1 ανταλλαγή (1-1 exchange).
3. Μια συνδυασμένη ανταλλαγή όπου ένας πελάτης που ανήκει σε ένα κύκλο ανταλλάσσεται με δύο γειτονικούς πελάτες που ανήκουν σε ένα άλλο κύκλο. Η μέθοδος αυτή ονομάζεται (1-2 exchange).
4. Μια συνδυασμένη ανταλλαγή όπου δύο γειτονικοί πελάτες που ανήκουν σε ένα κύκλο ανταλλάσσονται με δύο γειτονικούς πελάτες που ανήκουν σε ένα άλλο κύκλο. Η μέθοδος αυτή ονομάζεται (2-2 exchange).

4.1.3 Διαδικασία άπληστης Τυχοποιημένης Προσαρμοστικής Αναζήτησης – Μεθευρετικοί Αλγόριθμοι

Οι μεθευρετικοί αλγόριθμοι είναι μέθοδοι επίλυσης που συνδυάζουν διαδικασίες τοπικής αναζήτησης και υψηλότερου επιπέδου στρατηγικές για να δημιουργήσουν μια διαδικασία που είναι ικανή να ξεφύγει από κάποιο τοπικό ελάχιστο. Τα τελευταία χρόνια οι περισσότεροι αλγόριθμοι που έχουν αναπτυχθεί για την επίλυση των προβλημάτων συνδυαστικής βελτιστοποίησης ανήκουν σε αυτή τη κατηγορία.

Κάποιες από αυτές τις μεθόδους είναι η Προσομοιωμένη Ανόπτηση (SimulatedAnnealing), η Περιορισμένη Αναζήτηση (TabuSearch), οι Γενετικοί και Εξελικτικοί Αλγόριθμοι (GeneticandEvolutionaryalgorithms), τα Νευρωνικά Δίκτυα (NeuralNets), ο Αλγόριθμος Βελτιστοποίησης Αποικίας των Μυρμηγκιών (AntColonyOptimization), ο Αλγόριθμος Διασκορπισμένης Αναζήτησης (ScatterSearch), η Διαδικασία Άπληστης Τυχοποιημένης Προσαρμοστικής Αναζήτησης (GreedyRandomizedAdaptiveSearchProcedure), ο Αλγόριθμος της Διαφορικής Εξέλιξης (DifferentialEvolution) και ο Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων (ParticleSwarmOptimization).

Σε αυτές τις μεθόδους εξερευνάται το πεδίο λύσης με σκοπό να βρεθεί μια καλύτερη λύση. Τόσο η προσομοιωμένη ανόπτηση όσο και η περιορισμένη αναζήτηση ξεκινάνε από μια αρχική λύση x_1 και προχωράνε σε κάθε επανάληψη t από τη λύση x_1 στη λύση x_{t+1} στη γειτονιά $N(x_t)$ του x_t μέχρι ένα κριτήριο σταματήματος να ικανοποιηθεί. Οι γενετικοί αλγόριθμοι εξετάζουν σε κάθε βήμα ένα πληθυσμό από λύσεις. Κάθε πληθυσμός παράγεται από τον προηγούμενο του, συνδυάζοντας τα καλύτερα στοιχεία του και διαγράφοντας τα χειρότερα του. Η διαδικασία που χρησιμοποιείται για την δημιουργία των απογόνων, ονομάζεται διασταύρωση (crossover). Τα νευρωνικά δίκτυα περικλείουν έναν ευρετικό μάθησης που προσαρμόζει βαθμιαία ένα σύνολο από βάρη μέχρι να φτάσουμε σε μια αποδεκτή λύση.

Οι μεθευρετικοί αλγόριθμοι συνήθως χρησιμοποιούν πιο παραδοσιακούς ευρετικούς αλγορίθμους σαν υποδιαδικασίες τους. Πολλές φορές μπορούν να επιτραπούν σε κάποιο μεθευρετικό αλγόριθμο, και βήματα που οδηγούν σε μη εφικτή ενδιάμεση λύση σε κάποιο βήμα του αλγορίθμου. Ο λόγος που επιτρέπεται αυτό είναι για να αποφευχθεί κάποιο τοπικό ελάχιστο και η ολική λύση που θα εξαχθεί από τον αλγόριθμο να είναι καλύτερη. Ένα επιπλέον χαρακτηριστικό

αυτών αλγορίθμων είναι ότι προσομοιάζουν μια διαδικασία που συνήθως έχει εφαρμογή στη φύση. Τα στοιχεία που χρησιμοποιούν αυτοί οι αλγόριθμοι και μεταφορικά τα παρατηρούμε στη φύση είναι τα εξής :

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

Τα χαρακτηριστικά των μεθευρετικών είναι τα εξής:

- ❖ Μοντελοποιούν ένα φαινόμενο που υπάρχει στη φύση
- ❖ Μπορούν να μεταφερθούν εύκολα σε παράλληλη μορφή
- ❖ Είναι προσαρμοστικοί αλγόριθμοι

4.1.3.α Μεθευρετικοί Αλγόριθμοι και το πρόβλημα δρομολόγησης οχημάτων

Μια από τις πρώτες προσπάθειες να εφαρμοστεί ο αλγόριθμος της περιορισμένης αναζήτησης στο πρόβλημα δρομολόγησης οχημάτων οφείλεται στο Willard(22). Εδώ η αρχική λύση μετατρέπεται σε ένα πρόβλημα πλανόδιου πωλητή και οι γειτονικές λύσεις καθορίζονται σαν όλες τις εφικτές λύσεις που μπορούν να δημιουργηθούν από την τρέχουσα λύση χρησιμοποιώντας 2- opt και 3-opt μεθόδους. Η επόμενη κίνηση καθορίζεται από την καλύτερη μη απαγορευμένη κίνηση. Ο Osman(23) όπως και στην περίπτωση της προσομοιωμένης ανόπτησης καθορίζει τις γειτονικές χρησιμοποιώντας τον μηχανισμό των λ-ανταλλαγών. Η μέθοδος περιλαμβάνει 2-opt κινήσεις, επανατοποθετήσεις κόμβων σε διαφορετικές διαδρομές και ανταλλαγές κόμβων μεταξύ δύο διαφορετικών διαδρομών. Στη μια

εκδοχή του αλγόριθμου, που ονομάζεται βέλτιστα επιτρεπτή (best-admissible) , ολόκληρη η γειτονιά εξερευνάται και η καλύτερη επιτρεπτή λύση επιλέγεται. Στην δεύτερη εκδοχή που ονομάζεται πρώτη βέλτιστα επιτρεπτή (first-best-admissible), η πρώτη επιτρεπτή κίνηση, αν υπάρχει, επιλέγεται αλλιώς συνεχίζεται ο αλγόριθμος με την ήδη υπάρχουσα λύση.

Ο Αλγόριθμος TABUROUTE περιλαμβάνει αρκετά πολύπλοκα και καινοτόμα χαρακτηριστικά. Τα οποία είναι τα εξής:

- Η δομή της γειτονιάς καθορίζεται από όλες τις λύσεις στις οποίες μπορεί να οδηγηθούμε από την τρέχουσα λύση διαγράφοντας ένα κόμβο από τη διαδρομή του και εισάγοντας τον σε μια άλλη διαδρομή που περιέχεται ένα από τους p γείτονες του. Αυτό μπορεί να οδηγήσει στην διαγραφή μιας υπάρχουσας ή στην δημιουργία μιας καινούριας.
- Η διαδικασία αναζήτησης εξετάζει λύσεις που μπορεί να μην είναι εφικτές σε σχέση με τους περιορισμούς χωρητικότητας ή τους περιορισμούς που συσχετίζονται με το μήκος της διαδρομής. Πιο συγκεκριμένα, η αντικειμενική συνάρτηση περιέχει δύο όρους, που ο ένας μετράει την υπερχωρητικότητα, και ο άλλος την υπερδιάρκεια. Ο κάθε ένας από αυτούς έχει κάποιο βάρος: κάθε 10 επαναλήψεις, τα βάρη είτε διαιρούνται με το 2 αν και οι δέκα προηγούμενες επαναλήψεις οδηγούσαν σε εφικτή λύση είτε πολλαπλασιάζεται με το 2 αν και οι δέκα προηγούμενες επαναλήψεις οδηγούν σε μη εφικτές λύσεις. Με αυτό τον τρόπο αποφεύγεται η περίπτωση ο αλγόριθμος να βρει και να κολλήσει σε κάποιο τοπικό ελάχιστο.
- Ο αλγόριθμος Taburoute δεν χρησιμοποιεί στην ουσία την λίστα περιορισμένων κινήσεων, αλλά τυχαιοποιημένες περιορισμένες ετικέτες.
- Ο αλγόριθμος Taburoute χρησιμοποιεί μια στρατηγική διάχυσης. Αυτό πραγματοποιείται βάζοντας μια ποινή στους κόμβους που έχουν μετακινηθεί πάρα πολλές φορές κατά την διάρκεια των επαναλήψεων με στόχο να αυξηθεί η πιθανότητα να μετακινηθούν οι κόμβοι που δεν έχουν μετακινηθεί πολλές φορές κατά την διάρκεια των επαναλήψεων.

Στον αλγόριθμο Xu και του Kelly χρησιμοποιείται μια πιο περίπλοκη δομή για την εύρεση των γειτονιών. Η διαδικασία που ακολουθείται έχει τρία βασικά χαρακτηριστικά:

- Ανταλλαγές κόμβων ανάμεσα σε δύο διαδρομές.
- Ολική επανατοποθέτηση κάποιων κόμβων σε διαφορετικές διαδρομές.
- Τοπική αναζήτηση σε κάθε διαδρομή.

Η ολική επανατοποθέτηση επιτυγχάνεται με την επίλυση ενός μοντέλου ροής σε δίκτυα. Η βελτίωση μέσα στις ίδιες τις διαδρομές επιτυγχάνεται με την εφαρμογή της διαδικασίας 3-opt. Ο αλγόριθμος έχει πάρα πολλές παραμέτρους οι οποίες δυναμικά αναπροσαρμόζονται κατά την διάρκεια της αναζήτησης. Οι καλύτερες λύσεις που βρίσκονται κρατούνται στην μνήμη και περιοδικά χρησιμοποιούνται για να ξανα-αρχικοποιηθεί η αναζήτηση με καινούριες παραμέτρους.

Το κύριο χαρακτηριστικό του αλγορίθμου των Rego και Roucarol είναι η χρήση των αλυσίδων μετακίνησης για να μετακινηθούμε από τη μία λύση στην άλλη. Μια μετακίνηση αποτελείται από την μετακίνηση μιας κορυφής σε μια θέση που καταλαμβάνεται από μια άλλη κορυφή, και έτσι δημιουργείται μια αλυσίδα αντίδρασης 1 επιπέδων. Για μία δοσμένη διαδρομή, ορίζουμε σαν v_{i-1} τον προηγούμενο κόμβο του v_i και σαν v_{i+1} τον επόμενο.

Μια από τις πιο ενδιαφέρουσες βελτιώσεις στο χώρο της περιορισμένης αναζήτησης είναι ο αλγόριθμος της προσαρμοστικής μνήμης (AdaptiveMemory) των Rochat και Traillard. Η προσαρμοστική μνήμη δεν χρησιμοποιείται μόνο στον αλγόριθμο περιορισμένης αναζήτησης αλλά και σε άλλους μεθευρετικούς αλγορίθμους. Μια προσαρμοστική μνήμη είναι η συλλογή καλών λύσεων που δυναμικά ενημερώνονται κατά τη διάρκεια της διαδικασίας αναζήτησης. Περιοδικά κάποια στοιχεία εξάγονται από την ομάδα και συνδυάζονται διαφορετικά με τέτοιο τρόπο ώστε να οδηγήσουν σε νέες, πιθανώς, καλύτερες λύσεις. Στο πρόβλημα δρομολόγησης οχημάτων, οι διαδρομές των οχημάτων επιλέγονται από πάρα πολλές λύσεις και θα χρησιμοποιηθούν σαν αρχικό σημείο. Η διαδικασία δίνει μεγαλύτερα βάρη σε εκείνες τις διαδρομές που ανήκουν στις καλύτερες λύσεις. Η γειτονιά αναζήτησης που χρησιμοποιείται συνδυάζει τις μερικές διαδρομές με μία διαδικασία εισαγωγής κόμβων για να μπορέσει να

ολοκληρώσει τις λύσεις. Στην συνέχεια παρουσιάζεται το βασικό σχήμα της διαδικασίας της Προσαρμοστικής Αναζήτησης.

Βήμα 1. Δημιουργία μιας προσαρμοστικής μνήμης με τις καλύτερες λύσεις.

Βήμα2. Συνδυασμός κάποιων στοιχείων των λύσεων με στόχο την δημιουργία καινούριων λύσεων.

Βήμα 3. Οι μερικές λύσεις ολοκληρώνονται με μια διαδικασία εισαγωγής κόμβων.

Βήμα 4. Εφαρμογή της διαδικασίας της Περιορισμένης Αναζήτησης σε κάθε διαδρομή.

4.2 Αλγόριθμοι επίλυσης

Στο παρόν υποκεφάλαιο έχουμε αναφερθεί και περιγράψει με τον καλύτερο δυνατό τρόπο τους αλγορίθμους που χρησιμοποιήσαμε στην εργασία μας για την επίλυση του προβλήματος οχημάτων με ταυτόχρονη παραλαβή και παράδοση.

4.2.1 Αλγόριθμος Περιορισμένης Αναζήτησης (TabuSearch – TS) (4)

Ο μεθευρετικός αλγόριθμος της περιορισμένης αναζήτησης, είναι ίσως ο πιο γνωστός μεθευρετικός αλγόριθμος και αρχικά παρουσιάστηκε από τον Glover. Η περιορισμένη αναζήτηση χρησιμοποιεί ένα ευρετικό αλγόριθμο για να μετακινηθεί από την μια λύση στην άλλη. Όμως, όπως και στους άλλους μεθευρετικούς αλγόριθμους, υπάρχει η πιθανότητα η λύση να παγιδευτεί σε τοπικό ελάχιστο. Η σημαντική διαφορά της περιορισμένης αναζήτησης από τη στρατηγική της προσομοιωμένης ανόπτησης είναι ότι για να ξεφύγει η λύση από το τοπικό ελάχιστο χρησιμοποιείται μια συγκεκριμένη στρατηγική για την επιλογή της επόμενης λύσης και δεν γίνεται τυχαία επιλογή επόμενης λύσης. Η στρατηγική αυτή χρησιμοποιεί μνήμη (αντί για πιθανότητα). Όταν λέμε μνήμη εννοούμε

μνήμη από τις προηγούμενες κινήσεις που έχουν πραγματοποιηθεί. Για να αποφευχθούν οι συνεχείς επιστροφές σε προηγούμενες λύσεις οι τελευταίες κινήσεις καταγράφονται σε μία λίστα, η οποία ονομάζεται λίστα περιορισμένων κινήσεων (tabelist) και οι συγκεκριμένες κινήσεις απαγορεύεται να επιστρέψουν στην λύση για ένα συγκεκριμένο αριθμό επαναλήψεων.

Μία σύντομη περιγραφή του αλγόριθμου της περιορισμένης αναζήτησης είναι η ακόλουθη: Αρχικά, μία αρχική λύση κατασκευάζεται αξιοποιώντας τα ιδιαίτερα χαρακτηριστικά του κάθε προβλήματος. Στη συνέχεια η τιμή της συνάρτησης κόστους για τις υποψήφιες λύσεις στη γειτονιά της αρχικής λύσης υπολογίζεται και γίνεται μια προσπάθεια να κινηθούμε στη γειτονιά που δίνει την καλύτερη λύση (ή τουλάχιστον την λιγότερο κακή). Συνήθως, είναι η καλύτερη τιμή της συνάρτησης κόστους που καθορίζει την ποιότητα της λύσης. Για να γλιτώσουμε περιττούς υπολογισμούς του κόστους υπολογίζεται μια τιμή κίνησης αντί να υπολογίζεται το ολικό κόστος. Έτσι, αντί της συνάρτησης κόστους $c(s)$ χρησιμοποιείται μια συνάρτηση $dc(s)$ για να υπολογιστούν οι αλλαγές που συσχετίζονται με την κίνηση. Για να αποφευχθεί η παγίδευση σε τοπικό ελάχιστο, χρησιμοποιούνται πληροφορίες από τις k τελευταίες επαναλήψεις. Το σύνολο των κινήσεων που καθορίζονται από αυτές τις πληροφορίες μορφοποιούν την λίστα περιορισμένης αναζήτησης (tabulist). Αυτή η διαδικασία ονομάζεται μνήμη μικρής περιόδου (shorttermmemory) και σύμφωνα με αυτή ο αλγόριθμος "θυμάται" τις τελευταίες κινήσεις που έκανε.

Αυτές οι απαγορευμένες κατευθύνσεις αναζήτησης ονομάζονται περιορισμένες κινήσεις (tabumoves) ενώ σαν k ορίζεται το μέγεθος της λίστας περιορισμένης αναζήτησης. Το k είναι μια τιμή που είτε προσδιορίζεται από την αρχή, είτε είναι μια μεταβλητή που εξαρτάται από τις ιδιαίτερες συνθήκες του κάθε προβλήματος. Είναι επίσης πιθανή, η τυχαία επιλογή του k από ένα δοθέν διάστημα όπως επίσης και να αλλάζει δυναμικά κατά τη διάρκεια των επαναλήψεων. Το μέγεθος της λίστας είναι ένας πολύ σημαντικός παράγοντας επιτυχίας της μεθόδου. Σε κάθε επανάληψη προστίθεται στη λίστα η τελευταία κίνηση που έγινε και αν έχει συμπληρωθεί το μέγεθος της λίστας τότε η παλιότερη λύση αφαιρείται από τη λίστα. Η ύπαρξη αυτής της λίστας βοηθάει στην αποφυγή της λήψης των ίδιων λύσεων (cycling) ανάμεσα στις λύσεις που προκύπτουν σε διάφορες επαναλήψεις.

Συμπληρωματικά, περιορισμοί που βασίζονται στη συχνότητα (frequency) μπορούν να χρησιμοποιηθούν. Σε αυτή την προσέγγιση, μετράται ο αριθμός των φορών που μια συγκεκριμένη κίνηση εμφανίζεται κατά τη διάρκεια της

αναζήτησης και έτσι με δάση τους περιορισμούς της συχνότητας εμφάνισης μιας κίνησης γίνεται μια προσπάθεια να εμποδιστεί η συνεχιζόμενη επανάληψη της ίδιας κίνησης. Αυτό σημαίνει ότι μια μνήμη μακράς διαρκείας (longtermmemory) ενθαρρύνει τη διάχυση (diversification) της αναζήτησης μέσα σε ένα συγκεκριμένο σύνολο από πιθανώς καλές λύσεις. Για τη σωστή και ολοκληρωμένη εκμετάλλευση αυτών των περιορισμών μπορούν να χρησιμοποιηθούν κινήσεις δάση των ιδιοτήτων (moveattributes). Μια ιδιότητα μιας κίνησης από μια λύση στην επόμενη, ή γενικότερα μιας δοκιμαστικής κίνησης από την τρέχουσα λύση στην υποψήφια λύση, μπορεί να περιλαμβάνει οποιαδήποτε πλευρά που αλλάζει σαν επακόλουθο αυτής της κίνησης. Μερικές από τις ιδιότητες που μπορούν να επηρεαστούν και να αλλάξει η τιμή τους από μια κίνηση είναι οι ακόλουθες :

- Η αλλαγή μιας άλλης μεταβλητής x_k από 1 σε 0.
- Η συνδυασμένη αλλαγή των δύο προηγούμενων μεταβλητών αν εξετασθούν μαζί.
- Η αλλαγή του κόστους της λύσης από $f(S)$ σε $f(S')$.
- Μια ιδιότητα μπορεί να είναι η αλλαγή μιας μεταβλητής x_j από 0 σε 1.

Έτσι, όπως μπορεί να γίνει κατανοητό, μια κίνηση μπορεί στη συνέχεια να οδηγήσει σε αλλαγή πολλαπλών ιδιοτήτων. Για παράδειγμα, όπως φαίνεται και από την προηγούμενη λίστα μια αλλαγή στην τιμή δύο μόνο μεταβλητών μπορεί να αλλάξει και τις τέσσερις ιδιότητες. Ο λόγος που χρησιμοποιούνται αυτές οι κινήσεις βάση των ιδιοτήτων είναι για να επιβληθούν περιορισμοί (taburestrictions), που να εμποδίζουν να επιλεγθούν κινήσεις που να αντιστρέψουν τις αλλαγές που αντιπροσωπεύουν αυτές οι ιδιότητες.

Πιο συγκεκριμένα, αν από μία κίνηση $m(s)$ προκύπτει μια λύση S με κάποιες συγκεκριμένες ιδιότητες που αποθηκεύονται σε μια λίστα, και από την κίνηση $m(s')$ αντιστρέφονται οι ιδιότητες και προκύπτει η λύση S' , τότε αυτή η λύση απορρίπτεται. Ο περιορισμός δεν είναι ανάγκη να είναι μονάχα στις κινήσεις αλλά μπορεί να εξαρτάται και από την επανάληψη, για παράδειγμα σε διαφορετικές επαναλήψεις, διαφορετικές γειτονιές για την αναζήτηση πρέπει να είναι διαθέσιμες. Οι γειτονιές υποτίθεται ότι είναι συμμετρικές, $s \in N(s')$ ~~$s' \in N(s)$~~ $N(s)$. Μερικές φορές είναι ευνοϊκό να αγνοήσουμε τους περιορισμούς. Έστω, για παράδειγμα, η κατάσταση όπου μια κίνηση έχει ως αποτέλεσμα μια καλύτερη συνάρτηση κόστους. Σε αυτή την περίπτωση είναι προτιμητέο να κάνουμε την κίνηση ούτως ή άλλως. Η αγνόηση αυτή των περιορισμών ονομάζεται κριτήριο απενεργοποίησης των περιορισμών (aspirationcriterion). Όταν οι περιορισμοί

αγνοηθούν για μια καλύτερη τιμή αυτή τότε ονομάζεται κριτήριο απενεργοποίησης της καλύτερης λύσης (bestsolutionaspirationcriterion). Εάν το αποτέλεσμα όλων των κινήσεων είναι μια κατώτερη λύση και η καλύτερη από αυτές είναι απαγορευμένη, πάντοτε εφαρμόζεται η πρώτη μη απαγορευμένη λύση. Αυτά είναι τα πιο συνηθισμένα χαρακτηριστικά της περιορισμένης αναζήτησης. Πολλές φορές όμως για την αποδοτικότερη λειτουργία του αλγορίθμου, χρησιμοποιούνται κάποιες επιπλέον στρατηγικές.

Η πιο σημαντική από αυτές είναι η στρατηγική που χρησιμοποιείται με στόχο κάποιες μεταβλητές που βρίσκονται σταθερά μετά από κάποιες κινήσεις μέσα στη λύση ή που εμφανίζονται πολύ συχνά μέσα στη λύση να παραμείνουν εκεί (intensificationstrategies). Η μνήμη αυτή ονομάζεται μνήμη μεσαίας περιόδου (mediumtermmemory). Αυτή η στρατηγική έχει την απλή συλλογιστική ότι η βέλτιστη λύση θα πρέπει να βρίσκεται κοντά στο πιο υποσχόμενο σημείο. Αυτό σημαίνει ότι αν κάποιες μεταβλητές βρίσκονται είτε συνέχεια μέσα στη λύση είτε βρίσκονται πάρα πολύ συχνά μέσα στη λύση, το πιο πιθανό είναι η βέλτιστη λύση να βρίσκεται στη γειτονιά αυτών των μεταβλητών. Άρα διατηρούνται μέσα στη λύση αυτές οι μεταβλητές και γίνεται μια αναζήτηση στη γειτονιά τους για τη βέλτιστη λύση. Ο πιο απλός τρόπος για να χρησιμοποιηθούν οι στρατηγικές εντατικοποίησης της αναζήτησης είναι αποθηκεύοντας πολύ καλές (εκλεκτές) λύσεις κατά τη διάρκεια της αναζήτησης και χρησιμοποιώντας στοιχεία από αυτές τις λύσεις. Έτσι, όταν ενεργοποιηθεί η στρατηγική εντατικοποίησης (intensificationstrategies), ο πιο συνήθης τρόπος για να κινηθούμε είναι ο ακόλουθος :

Έχοντας αποθηκεύσει τις καλύτερες λύσεις κατά τη διάρκεια των επαναλήψεων επαναρχίζουμε τον αλγόριθμο και συνδέουμε τις εκλεκτές λύσεις με τη δέλτιστη τωρινή λύση. Ένας τρόπος για να πραγματοποιηθεί αυτό είναι με τη χρήση της μεθόδου επανασύνδεσης διαδρομών (PathRelinking). Ένας τρόπος για να δημιουργηθεί η μνήμη μεσαίας περιόδου είναι με τη χρήση της μνήμης πρόσφατων κινήσεων (recencymemory) η οποία έχει ως εξής :

Αρχικά, κάποιοι παράγοντες που συσχετίζονται με τη λύση αποθηκεύονται στη μνήμη. Στη συνέχεια, κάθε φορά που επιλέγεται μία καινούρια λύση τα στοιχεία που αποτελούν τη λύση καταγράφονται στη μνήμη πρόσφατων κινήσεων. Όταν ενεργοποιηθεί η μνήμη τα στοιχεία αυτά χρησιμοποιούνται μαζί με την τρέχουσα δέλτιστη λύση με στόχο να βελτιωθεί η λύση. Για παράδειγμα, αν λυθεί το πρόβλημα του πλανόδιου πωλητή η μνήμη πρόσφατων καταστάσεων θα μπορούσε να περιέχει όλα τα τόξα του προβλήματος. Έτσι, αρχικά έστω ότι αρχίζουν με τη τιμή 0. Όταν, όμως, επιλεγεί μία λύση τα αντίστοιχα τόξα στο διάνυσμα αυξάνουν

την τιμή τους κατά 1. Έτσι, όταν ενεργοποιηθούν οι συνθήκες για να κάνουμε τη στρατηγική εντατικοποίησης, τα στοιχεία (τόξα) που έχουν το μεγαλύτερο αριθμό σημαίνει ότι ήταν τα πιο ελπιδοφόρα ως τώρα και προσπαθούμε να τα εισάγουμε στη δέλτιστη λύση μήπως και δώσουν καλύτερα αποτελέσματα.

Ένα από τα μεγαλύτερα προβλήματα όλων των μεθόδων τοπικής αναζήτησης είναι η ίδια η φύση τους, δηλαδή όπως λέει και το όνομα τους η μέθοδος αναζητά να βρει το δέλτιστο τοπικά γύρω από κάποιες υποσχόμενες για εξερεύνηση περιοχές. Έτσι, πάρα πολλές περιοχές του χώρου αναζήτησης μένουν ουσιαστικά ανεξερεύνητες γεγονός που έχει ως αποτέλεσμα να βρίσκονται κάποιες καλές λύσεις αλλά να μένουν αναξιοποίητοι κάποιοι χώροι αναζήτησης που μπορούσαν να οδηγήσουν σε πολύ καλύτερες λύσεις. Η στρατηγική της διάχυσης (diversification) είναι ένας μηχανισμός που προσπαθεί να οδηγήσει την αναζήτηση σε ανεξερεύνητες περιοχές. Συνήθως, βασίζεται σε κάποιου είδους μνήμη μακράς διάρκειας (longterm memory), στην οποία καταγράφονται ο συνολικός αριθμός των επαναλήψεων όπως επίσης και διάφοροι παράγοντες των περιοχών που έχουν εξερευνηθεί και των μεταδλητών που βρίσκονται τώρα στην τρέχουσα λύση ή που έχουν βρεθεί κατά τη διάρκεια των επαναλήψεων.

Η πιο σημαντική στρατηγική διαφοροποίησης περιλαμβάνει την αναγκαστική χρησιμοποίηση διαφόρων παραγόντων που έχουν χρησιμοποιηθεί ελάχιστα ως τώρα στην τρέχουσα (ή στην καλύτερη) λύση και η επανεκκίνηση της διαδικασίας από αυτή τη λύση. Η μνήμη μακρίας διάρκειας αναπαρίσταται συνήθως με τη μνήμη συχνοτήτων (frequency memory). Η μνήμη συχνοτήτων λειτουργεί όπως η μνήμη πρόσφατων καταστάσεων αλλά έχοντας ακριβώς την αντίθετη λειτουργία. Δηλαδή, ενώ στη μνήμη πρόσφατων καταστάσεων χρησιμοποιούνται τα πιο σημαντικά στοιχεία της μνήμης, στη μνήμη συχνοτήτων χρησιμοποιούνται τα λιγότερο σημαντικά στοιχεία της μνήμης με στόχο η αναζήτηση να ξεκινήσει από τελείως διαφορετικό σημείο. Έτσι, στο πρόβλημα του πλανόδιου πωλητή όταν ενεργοποιηθούν οι συνθήκες της μνήμης συχνοτήτων τότε επιλέγονται να μπουν στη λύση τα τόξα που έχουν χρησιμοποιηθεί λιγότερο στις προηγούμενες επαναλήψεις. Η στρατηγική διάχυσης της λύσης μπορεί να εφαρμοστεί με αρκετούς διαφορετικούς τρόπους. Για παράδειγμα, ένας τρόπος θα μπορούσε να ήταν να χρησιμοποιηθούν είτε στην τρέχουσα είτε στην καλύτερη λύση τα λιγότερο χρησιμοποιημένα στοιχεία και μετά να συνεχίσουμε την αναζήτηση από αυτό το σημείο. Ένας άλλος τρόπος είναι να χρησιμοποιηθεί κάποια συνάρτηση τιμωρίας που να οδηγεί προς τη λειτουργία της διάχυσης των λύσεων.

Η περιορισμένη αναζήτηση δεν συγκλίνει με φυσικό τρόπο, έτσι κάποια κριτήρια τερματισμού θα πρέπει να καθοριστούν. Για παράδειγμα, ο αλγόριθμος μπορεί να τερματίσει όταν κάποιος προκαθορισμένος αριθμός επαναλήψεων έχει περάσει ή όταν ένας προκαθορισμένος αριθμός επαναλήψεων ολοκληρωθεί χωρίς η λύση να βελτιωθεί. Ο ψευδοκώδικας του αλγορίθμου παρουσιάζεται στη συνέχεια:

Αλγόριθμος Περιορισμένη Αναζήτηση

Αρχικοποίηση Κατασκευή μιας αρχικής λύσης S_0

Υπολογισμός της συνάρτησης κόστους της λύσης $S \square = S_0$!

Αρχικοποίηση της βέλτιστης λύσης $f(S \square) = f(S_0)$

Κύρια Φάση:

Dowhile κάποιο κριτήριο σταματήματος δεν έχει ικανοποιηθεί

Υπολογισμός μιας γειτονικής λύσης S :

iff(S) < $f(S \square)$

then $S \square = S$ $f \square = f(S)$

endif

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

Κάλεσε κάθε k_1 επαναλήψεις τη στρατηγική εντατικοποίησης

Iff($S_{intensification}$) < $f(S \square)$

then $S \square = S_{intensification}$ $f \square = f(S_{intensification})$

endif

Κάλεσε κάθε k_2 επαναλήψεις τη στρατηγική διαφοροποίησης

Iff($S_{diversification}$) < $f(S \square)$

then $S \square = S_{diversification}$ $f \square = f(S_{diversification})$

endifEnddo

Επέστρεψε τη βέλτιστη λύση $S \square$.

(5)

4.2.2 Αλγόριθμος εμπνευσμένος από την φύση για ad-hoc δίκτυα και clouds(AntColonyOptimization - ACO) (6)

Ο ACO είναι ένας μεθευριστικός αλγόριθμος γενικού σκοπού, ο οποίος μπορεί να χρησιμοποιηθεί για την επίλυση NP-complete προβλημάτων και είχε πολλές εξελίξεις. Ο αλγόριθμος ACO αποτελεί ιδανική λύση για δίκτυα που χαρακτηρίζονται από αυτονομία, λόγω της αυτοργάνωσης που παρουσιάζει . Ο ACO αποτέλεσε πηγή έμπνευσης για την δημιουργία πρωτοκόλλων δρομολόγησης σε adhoc δίκτυα ενώ αποτελεί ενδιαφέρον κομμάτι έρευνας για τα Clouds.

Οι αλγόριθμοι που προέρχονται από τη μελέτη της συμπεριφοράς των αποικιών μυρμηγκιών, έχουν ως στόχο την επίλυση προβλημάτων συνδυαστικής βελτιστοποίησης και όχι την εξομοίωση της λειτουργίας των αποικιών των μυρμηγκιών και ως εκ τούτου παρουσιάζουν ορισμένες σημαντικές διαφορές από τα χαρακτηριστικά που παρουσιάζουν τα ανάλογα συστήματα στη φύση: τα τεχνητά μυρμήγκια έχουν κάποια διαθέσιμη μνήμη και χρησιμοποιούν σε κάποιο βαθμό την όρασή τους . Η κάθε λύση για το TSP με χρήση αυτού του αλγορίθμου σχηματίζεται με τη διαδοχική μετάβαση των τεχνητών μυρμηγκιών-πρακτόρων από τη μια πόλη στην άλλη με κάποια πιθανότητα . Το «ταξίδι» ολοκληρώνεται όταν το τεχνητό μυρμήγκι επιστρέψει στην αρχική πόλη. Τότε βαθμολογείται η λύση που έχει επιτύχει με κριτήριο το συνολικό μήκος της διαδρομής και προστίθεται η ανάλογη φερορμόνη στο μονοπάτι που ακολούθησε. Η παραπάνω διαδικασία επαναλαμβάνεται για κάθε άτομο της αποικίας μέχρι να συμπληρωθεί ο ζητούμενος αριθμός επαναλήψεων.

Αρχικά τα μυρμήγκια τοποθετούνται είτε τυχαία στις πόλεις είτε το καθένα σε μια διαφορετική πόλη ως αφετηρία της διαδρομής τους. Η πόλη αυτή καταγράφεται ως η αφετηρία στη λίστα της μνήμης κάθε μυρμηγκιού έτσι ώστε να αποκλειστεί από τις επόμενες επιλογές του. Η επιλογή της επόμενης πόλης την οποία θα επισκεφθεί κάθε μυρμήγκι ορίζεται από μια πιθανότητα μετάβασης η οποία εξαρτάται από την «ορατότητα» του κάθε μυρμηγκιού (ορατότητα ορίζεται ως το αντίστροφο της απόστασης μεταξύ δύο πόλεων και άρα εισάγει μεγάλη πιθανότητα στον κανόνα μετάβασης όσον αφορά τις κοντινές πόλεις) και στην ποσότητα της φερορμόνης που έχει εναποτεθεί στην κάθε διαδρομή εκφράζοντας την εμπειρία της υπόλοιπης αποικίας. Αφού όλα τα μυρμήγκια ολοκληρώσουν τα “ταξίδια” τους σε κάθε επανάληψη, τότε προστίθεται φερορμόνη σε κάθε μονοπάτι ανάλογα με την επίδοση του κάθε μυρμηγκιού. Σημαντικό στοιχείο στην όλη διαδικασία είναι και η εισαγωγή ενός παράγοντα σχετικού με την εξάτμιση της

φερορμόνης. Κατά αυτόν τον τρόπο αποφεύγεται η ενίσχυση των αρχικών τυχαίων διακυμάνσεων.

Τα βήματα του αλγορίθμου ACO είναι τα ακόλουθα:

- Βήμα 1ο:

Θέτουμε τη φερορμόνη στην αρχική της τιμή (είτε σε μια πολύ μικρή τιμή είτε σε μια τυχαία τιμή) σε όλα τα μονοπάτια που συνδέουν τις πόλεις μεταξύ τους. Υπολογίζουμε όλες τις αποστάσεις των πόλεων. Επιλέγουμε (τυχαία) την πόλη-αφετηρία της διαδρομής κάθε μυρμηγκιού και την τοποθετούμε στην λίστα της μνήμης του.

- Βήμα 2ο:

Για κάθε μυρμήγκι, επιλέγουμε την επόμενη πόλη που θα επισκεφτεί με βάση τον τυχαίο αναλογικό κανόνα μετάβασης μέχρι να ολοκληρώσει τη διαδρομή του αφού έχει περάσει από κάθε πόλη μια μόνο φορά και τελικά να επιστρέψει στην αφετηρία του.

- Βήμα 3ο:

Καταγράφουμε την καλύτερη διαδρομή που βρέθηκε.

- Βήμα 4ο:

Ανανεώνουμε τη φερορμόνη στα μονοπάτια που επισκέφτηκαν τα μυρμήγκια με βάση τον κανόνα ανανέωσης της φερορμόνης.

- Βήμα 5ο:

Επαναλαμβάνουμε τη διαδικασία από το βήμα 2 έως ότου ολοκληρωθεί ένας συγκεκριμένος αριθμός επαναλήψεων ή επιτευχθεί ένα κριτήριο σύγκλισης.

Ο ACO είναι μια στοχαστική αναζήτηση λύσης σε ένα συνδυαστικό πρόβλημα με πυρήνα το μοντέλο απόδοσης φερορμόνης. Το πρώτο βήμα όμως της εφαρμογής του αλγορίθμου είναι η μοντελοποίηση του προβλήματος. Το μοντέλο $P(S, \Omega, f)$ ενός συνδυαστικού προβλήματος αποτελείται από :

- Ένα χώρο αναζήτησης S ορισμένο από ένα πεπερασμένο αριθμό διακριτών μεταβλητών απόφασης
- Ένα σύνολο περιορισμών Ω
- Μία συνάρτηση $f : S \rightarrow R_0^+$ της οποίας στόχος είναι να βρεθεί το ελάχιστο.

Ο χώρος αναζήτησης ορίζεται ως εξής:

Με δεδομένο ένα σύνολο διακριτών μεταβλητών

$X_i, i = 1, \dots, n$ με τιμές v_i^j

που ανήκουν σε ένα πίνακα τιμών της μορφής.

$D^i = \{v_i^j, \dots, v_i^{D^i}\}$

Η απόδοση των τιμών v_i^j στο X_i συμβολίζεται με $X_i \leftarrow v_i^j$. Μια λύση s είναι εφικτή όταν όλες οι αποδόσεις τιμών στις μεταβλητές ικανοποιούν τους περιορισμούς Ω . Μια λύση s^* λέγεται ολικό ελάχιστο όταν για κάθε s που ανήκει στο S ισχύει $f(s^*) \leq f(s)$. Το σύνολο των ολικών ελάχιστων συμβολίζεται με S^* . Στόχος του ACO είναι η εύρεση ενός από τα ολικά ελάχιστα. Η απόδοση τιμής σε μια μεταβλητή συμβολίζεται με το c_{ij} το οποίο ονομάζεται τμήμα λύσης. Στη συνέχεια στο κάθε τμήμα λύσης (c_{ij}) αποδίδεται μια τιμή επιπέδου φερορμόνης (συμβολισμένη με τ_{ij}). Η τιμή αυτή ανανεώνεται από τον ACO και καθορίζει την πιθανότητα επιλογής ενός τμήματος λύσης.

Στον ACO τα τεχνητά μυρμήγκια κατασκευάζουν μια λύση διασχίζοντας το γράφημα του συνδυαστικού προβλήματος, $G_c(V, E)$, ο οποίος αποτελείται από μια συλλογή κορυφών V και μια συλλογή ακμών E . Τα τμήματα λύσης c αντιστοιχίζονται είτε στις κορυφές είτε στις ακμές του γράφου. Τα τεχνητά μυρμήγκια διασχίζουν τον γράφο από κορυφή σε κορυφή μέσω των ακμών δημιουργώντας σταδιακά μια λύση επιλέγοντας τα τμήματα λύσης c που αντιστοιχούν στις ακμές και κορυφές που διέσχισαν. Επιπλέον, εναποθέτουν σε κάθε c μια ποσότητα φερορμόνης η οποία μπορεί να εξαρτάται από διάφορους παράγοντες ανάλογα με την εκάστοτε υλοποίηση του αλγορίθμου. Επόμενα μυρμήγκια χρησιμοποιώντας τα επίπεδα φερορμόνης οδηγούνται στο γράφημα προς πιο ελπιδοφόρες περιοχές λύσης.

4.2.3 ΟΑλγόριθμός Ant Colony System (ACS) (7)

Ο αλγόριθμος αυτός εισήχθη από τον Dorigo και αποτελεί βελτίωση του AS. Μάλιστα, τα αποτελέσματα του ACS είναι σαφώς ανώτερα του AS για το TSP. Οι βασικές διαφορές των δύο αλγορίθμων μπορούν να συνοψιστούν στα εξής:

1. Κανόνας μετάβασης (transitionrule):

Τώρα πλέον, ένα μυρμήγκι k που βρίσκεται στην πόλη i επιλέγει την επόμενη πόλη j που θα επισκεφθεί με τον εξής κανόνα:

$$j = \{ \tau_{iu}(t) [n_{iu}(t)]^\beta \}, \text{ αν } q \leq q_0$$

$$j = J, \quad \text{αν } q > q_0$$

2. Ανανέωση της φερομόνης (pheromoneupdate):

Η εναπόθεση φερομόνης γίνεται μόνο από το μυρμήγκι που μέχρι αυτήν τη χρονική στιγμή έχει κάνει την καλύτερη διαδρομή (αντί από όλα τα μυρμήγκια όπως στον AS) σε μια προσπάθεια να κατευθυνθεί η εξερεύνηση του χώρου προς την περιοχή του ολικού βέλτιστου.

Συνεπώς, η σχέση που διέπει την εναπόθεση της φερομόνης μετασχηματίζεται στην εξής (κανόνας εναπόθεσης φερομόνης στη βέλτιστη διαδρομή) :

$$\tau_{ij}(t) \leftarrow (1-\rho) \tau_{ij} + \Delta \tau_{ij}(t)$$

$$\Delta \tau_{ij}(t) = \begin{cases} \frac{\rho}{L_{best}(t)}, \\ 0, \end{cases}$$

με :

- $L_{best}(t)$ το μήκος της βέλτιστης διαδρομής στην επανάληψη t .
- $T_{best}(t)$ η σειρά των πόλεων της βέλτιστης διαδρομής στην επανάληψη t

3. Τοπική ανανέωση της φερομόνης (localpheromoneupdate):

Για να αποφευχθεί ο εγκλωβισμός των μυρμηγκιών σε μια και μόνο διαδρομή, κάτι που θα έκανε τη χρήση των m μυρμηγκιών ατελέσφορη, πρέπει κάθε μυρμήγκι k που επισκέπτεται την πόλη j από την πόλη i , να αφαιρεί ένα ποσό φερομόνης από το μονοπάτι που συνδέει αυτές τις δύο πόλεις. Με αυτόν τον τρόπο, η εξερεύνηση για νέες λύσεις κατευθύνεται μακριά από την καλύτερη που έχει υπολογισθεί έως αυτή τη στιγμή. Η σχέση που ισχύει σε αυτή την περίπτωση είναι:

$$\tau_{ij}(i+1) \leftarrow (1-\rho) \tau_{ij} + \rho \tau_0$$

όπου τ_0 η ποσότητα της φερομόνης με την οποία γίνεται η αρχικοποίηση.

Έχει βρεθεί πειραματικά ότι: $\tau_0 = (n \cdot L_{nn})^{-1}$ με n τον αριθμό των πόλεων και L_{nn} το μήκος μιας διαδρομής που δημιουργείται με τον αλγόριθμο NearestNeighbor (άπληστος αλγόριθμος λύσης του TSP ο οποίος επιλέγει ως επόμενο σταθμό την πιο κοντινή πόλη).

4. Χρήση λίστας υποψηφίων πόλεων (candidatelist):

Στα προβλήματα με μεγάλο αριθμό πόλεων – και ειδικά σε προβλήματα ATSP – είναι χρονοβόρο να εξετάζονται όλες οι πόλεις. Για το λόγο αυτό η λίστα αυτή περιέχει ένα αριθμό από τις πιο «επιθυμητές», δηλαδή κοντινές, πόλεις ταξινομημένες από την κοντινότερη προς την πιο μακρινή. Μόνο όταν οι πόλεις που βρίσκονται στη λίστα των υποψηφίων πόλεων ανήκουν και στην Tabulist είναι δυνατόν τα μυρμήγκια να κατευθυνθούν σε άλλες πόλεις.

5. Άλλες διαφοροποιήσεις:

- ανανέωση της φερομόνης από τα δύο μυρμήγκια με τις καλύτερες διαδρομές
- αφαίρεση φερομόνης από τις κακές διαδρομές
- χρήση αλγορίθμων για τοπική αναζήτηση λύσεων, πχ. 2-opt, 3-opt, Lin-Kernighan.

4.2.4 Ο Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων (ParticleSwarmOptimization(PSO)) (8)

Ο Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων (ParticleSwarmOptimization (PSO)) έχει ως σκοπό την εύρεση βέλτιστης λύσης, δηλαδή τη βελτιστοποίηση μιας αντικειμενικής συνάρτησης, διεξάγοντας μια στοχαστική αναζήτηση βασιζόμενη σε έναν πληθυσμό. Ο πληθυσμός αποτελείται από άτομα(particles), το κάθε ένα από τα οποία αποτελεί μια πιθανή λύση(candidate solution). Αρχικά ο πληθυσμός αρχικοποιείται κατά τυχαίο τρόπο, αλλά έτσι ώστε να ισχύουν οι περιορισμοί ισότητας και ανισότητας στους οποίους υπόκειται το πρόβλημα, και τα άτομα κινούνται ελεύθερα εντός του πολυδιάστατου χώρου αναζήτησης με κάποια ταχύτητα(velocity) η οποία μαθηματικά έχει την ίδια ακριβώς μορφή με τα άτομα.

Κατά τη διάρκεια των επαναλήψεων τα άτομα διορθώνουν τόσο την ταχύτητά τους όσο και τη θέση τους(position) βασιζόμενα στη βέλτιστη λύση που έχει επιτύχει το κάθε ένα από αυτά (pbest) αλλά και στη βέλτιστη λύση που έχει επιτύχει ο συνολικός πληθυσμός (gbest). Ο κύριος κορμός της PSO είναι δύο σχέσεις ταχύτητας και θέσης του κάθε ατόμου οι οποίες ανανεώνονται συνεχώς όπως ειπώθηκε παραπάνω. Συγκεκριμένα, η θέση και η ταχύτητα του κάθε ατόμου αναπαρίστανται από ένα διάνυσμα d διαστάσεων, όπου d_0 αριθμός των παραμέτρων που πρόκειται να βελτιστοποιηθούν. Δηλαδή, η θέση και η ταχύτητα του i -οστού ατόμου έχουν την εξής μορφή: $X_i=[X_{i,1}, X_{i,2}, \dots, X_{i,d}]^T$ και $V_i=[V_{i,1}, V_{i,2}, \dots, V_{i,d}]^T$ αντίστοιχα. Η βέλτιστη λύση του i -οστού ατόμου (pbest) συμβολίζεται με $p_i=[p_{i,1}, p_{i,2}, \dots, p_{i,d}]^T$ και η βέλτιστη λύση του συνολικού πληθυσμού(gbest) συμβολίζεται με $p_g=[p_{g,1}, p_{g,2}, \dots, p_{g,d}]^T$.

Η σχέση σύμφωνα με την οποία ανανεώνεται η ταχύτητα του κάθε ατόμου υπολογίζεται ως εξής:

$$V_{i,j}(k+1)=w \cdot V_{i,j}(k)+c_1 \cdot r_1 \cdot (p_{i,j}-X_{i,j}(k))+c_2 \cdot r_2 \cdot (p_{g,j}-X_{i,j}(k)) \quad j=1, 2, \dots, d$$

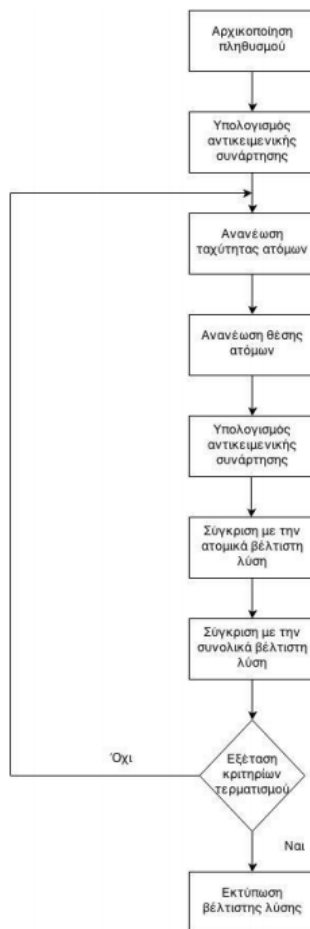
Οι c_1 και c_2 είναι δύο θετικές σταθερές που ονομάζονται συντελεστές βαρύτητας, ενώ οι r_1, r_2 είναι δύο τυχαίες μεταβλητές που παίρνουν τιμή στο διάστημα $[0,1]$ προσδίδοντας στον αλγόριθμο στοχαστικό χαρακτήρα. Η παράμετρος w ονομάζεται παράγοντας αδράνειας(inertia factor) και ελέγχει πως επηρεάζουν οι προηγούμενες ταχύτητες τις καινούριες. Όσον αφορά τα πρώτα στάδια της αναζήτησης είναι προτιμότερο να χρησιμοποιούνται μεγάλες τιμές του w ενώ όταν πρόκειται για τις τελευταίες επαναλήψεις όπου ο αλγόριθμος έχει ήδη

προσεγγίσει την περιοχή της βέλτιστης λύσης, είναι αποδοτικότερο να χρησιμοποιούνται μικρές τιμές του w έτσι ώστε να υπάρχει καλύτερη και πιο λεπτομερής τοπική αναζήτηση. Έτσι το w παρουσιάζεται να μεταβάλλεται γραμμικά σύμφωνα με την παρακάτω σχέση:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{\text{iteration}} \cdot G_{\max}$$

όπου iterations ο αριθμός της τρέχουσας επανάληψης, G_{\max} οι συνολικές επαναλήψεις και w_{\max} και w_{\min} η μέγιστη και η ελάχιστη αντίστοιχα τιμή που έχουμε ορίσει. Μετά την ανανέωση της ταχύτητας, η νέα θέση του κάθε ατόμου για την επόμενη γενιά ανανεώνεται σύμφωνα με την παρακάτω σχέση: $X_{i,j}(k+1) = X_{i,j}(k) + V_{i,j}(k+1)$, $j=1, 2, \dots, d$

Αφού γίνει ο υπολογισμός της νέας θέσης και ταχύτητας, ανανεώνονται τα $pbest$ και $gbest$ στην περίπτωση που το άτομο ή ο πληθυσμός έχει επιτύχει κάποια καλύτερη λύση. Μετά από αυτό το βήμα εξετάζεται αν ικανοποιούνται οι συνθήκες τερματισμού. Αν αυτές ικανοποιούνται ο αλγόριθμος τερματίζεται.



Διάγραμμα 1: Διάγραμμα ροής PSO

Ο Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων (Particle Swarm Optimization (PSO)) είναι ένας πολύ γνωστός αλγόριθμος ολικής βελτιστοποίησης που προτάθηκε από τους Kennedy και Eberhart [82, 83, 65] για να προσομοιώσει την κοινωνική συμπεριφορά κάποιων οργανισμών όπως το πέταγμα σε μορφή σμήνους των πουλιών και την ομαδική κίνηση των ψαριών. Αρχικά η ιδέα ήταν να προσομοιωθεί γραφικά η κίνηση που κάνει ένα σμήνος από πουλιά, να βρεθούν και να αναλυθούν οι κανόνες που οδηγούν τα πουλιά να κινούνται ως ένα σμήνος και πως ξαφνικά αλλάζουν κατεύθυνση χωρίς να καταστρέφεται ο σχηματισμός που είχαν αρχικά. Ο Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων χρησιμοποιεί τη φυσική κίνηση των ατόμων του πληθυσμού στο σμήνος και έχει ένα πολύ ευέλικτο και καλά ισορροπημένο μηχανισμό που προσαρμόζεται στις ολικές και τοπικές ικανότητες εξερεύνησης των ατόμων του σμήνους. Οι αλλαγές ενός σωματιδίου μέσα στο σμήνος επηρεάζονται από την εμπειρία και τις γνώσεις των γειτονικών του σωματιδίων, και ως εκ τούτου ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων μπορεί να θεωρηθεί ως ένας συμβιωτικός συνεργατικός αλγόριθμος. Η συγκεκριμένη μέθοδος έχει αποδειχτεί ότι είναι πάρα πολύ χρήσιμη σε ένα αριθμό από προβλήματα, λόγω της εύκολης υλοποίησής της και των πολύ καλών αποτελεσμάτων που έχει δώσει στα προβλήματα που έχει χρησιμοποιηθεί, ιδιαίτερα δε σε προβλήματα που έχουν συνεχείς μεταβλητές. Τα τελευταία χρόνια η συγκεκριμένη μέθοδος έχει εφαρμοστεί και σε προβλήματα συνδυαστικής βελτιστοποίησης. Σε σύγκριση με τους γενετικούς αλγορίθμους ο Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων έχει μερικά ενδιαφέροντα χαρακτηριστικά. Αρχικά έχει μνήμη πράγμα που σημαίνει ότι η γνώση από προηγούμενες καλές λύσεις κληρονομείται στις επόμενες γενιές και δεν χάνεται από επανάληψη σε επανάληψη όπως συμβαίνει στους γενετικούς. Επίσης, παρατηρείται μια πολύ μεγάλη συνεργασία ανάμεσα στα σωματίδια του σμήνους αφού τα μέλη της ομάδας συνεργάζονται μεταξύ τους στην κατασκευή των λύσεων.

Πολλά προβλήματα βελτιστοποίησης περιλαμβάνουν έναν αριθμό περιορισμών που η λύση πρέπει να ικανοποιεί. Έτσι, ο σκοπός των βελτιστοποιήσεων με περιορισμούς είναι να ψάξει για εφικτές λύσεις με καλύτερες αντικειμενικές τιμές. Ένα πρόβλημα βελτιστοποίησης με περιορισμούς μπορεί να περιγραφεί ως εξής:

Ζητείται η εύρεση του x έτσι ώστε να ελαχιστοποιηθεί η $f(x)$ και να υπόκειται στους ακόλουθους περιορισμούς:

$$g_i(x) \leq 0, i = 1, 2, \dots, n$$

$$h_j(x) \leq 0, i= 1, 2, \dots, n$$

όπου,

$$x=[x_1, x_2, \dots, x_d]^T$$

συμβολίζει το διάνυσμα της λύσης, n είναι ο αριθμός των ανισοτικών περιορισμών και p ο αριθμός των ισοτικών περιορισμών. Πρακτικά, ένας ισοτικός περιορισμός $h_j(x)=0$ μπορεί να αντικατασταθεί από δύο μη ανισοτικούς περιορισμούς $h_j(x) \leq \delta$ και $h_j(x) \geq -\delta$ (δ είναι μια πολύ μικρή ποσότητα). Επομένως όλοι οι περιορισμοί μπορούν να μετασχηματιστούν σε $N=2n+p$ ανισοτικούς περιορισμούς. Καλούμε το x εφικτή (feasible) λύση αν ικανοποιεί όλους τους περιορισμούς. Παραδοσιακοί μαθηματικοί μέθοδοι προγραμματισμού όπως η μέθοδος του πολλαπλασιαστή Lagrange συνήθως απαιτούν πληροφορίες της παραγώγου της αντικειμενικής συνάρτησης. Παράλληλα, η λύση που προκύπτει συνήθως τείνει να είναι τοπικό (local) βέλτιστο.

Τα τελευταία χρόνια, διάφοροι εξελικτικοί αλγόριθμοι (Evolutionary algorithms-EAs) έχουν προσελκύσει ιδιαίτερο ενδιαφέρον για μια σειρά από προβλήματα βελτιστοποίησης χάρη στα πλεονεκτήματά τους υπέρ των παλαιότερων μεθόδων. Οι εξελικτικοί αλγόριθμοι δεν απαιτούν η αντικειμενική συνάρτηση να είναι παραγωγίσιμη-ούτε καν συνεχής- και λόγω του ότι εξερευνούν όλο το εύρος της περιοχής που βρίσκεται η λύση παρουσιάζουν ολικό (global) βέλτιστο. Το 1995 αναπτύχθηκε μια στοχαστική μέθοδος βελτιστοποίησης από τους Kennedy και Eberhart, η μέθοδος των ατόμων σμήνους (Particle Swarm Optimization PSO). Η ανάπτυξη της PSO βασίζεται στην παρατήρηση της συμπεριφοράς ατόμων σμήνους, όπως για παράδειγμα ένα σμήνος πουλιών ή ψαριών. Στην αναζήτησή τους για τροφή από το ένα μέρος στο άλλο, πάντα υπάρχει κάποιο μέλος του σμήνους που έχει μεγαλύτερη ικανότητα στο να εντοπίσει την τροφή.

Καθώς υπάρχει αλληλεπίδραση και μετάδοση πληροφορίας μεταξύ των ατόμων του σμήνους, το σμήνος τελικά θα μετακινηθεί προς την περιοχή όπου υπάρχει τροφή. Στη μέθοδο της PSO υπάρχει ένας αριθμός ατόμων (particles) τα οποία τοποθετούνται στο χώρο αναζήτησης ενός προβλήματος ή μιας συνάρτησης και υπολογίζεται η τιμή της αντικειμενικής συνάρτησης στην τρέχουσα θέση. Κάθε σωματίδιο καθορίζει στη συνέχεια την κίνησή του στο χώρο αναζήτησης βασιζόμενο τόσο στη δική του εμπειρία (τρέχουσα και καλύτερη θέση) όσο και στην εμπειρία των υπόλοιπων ατόμων του πληθυσμού. Η επόμενη επανάληψη λαμβάνει χώρα αφού έχουν κινηθεί όλα τα σωματίδια.

Τελικά, το σμήνος ως σύνολο θα προσεγγίσει τη βέλτιστη τιμή της συνάρτησης. Χάρη στην γρήγορη σύγκλιση και στην εύκολη εφαρμογή της η PSO έχει κερδίσει ιδιαίτερο ενδιαφέρον και έχει εφαρμοστεί επιτυχημένα σε πολλούς τομείς κυρίως για προβλήματα βελτιστοποίησης συνεχών μεταβλητών χωρίς περιορισμούς. Όσον αφορά τα προβλήματα βελτιστοποίησης με περιορισμούς έχουν προταθεί διάφορες υβριδικές μέθοδοι. Εμείς θα ασχοληθούμε με μια υβριδική PSO (Hybrid PSO) η οποία χρησιμοποιεί έναν κανόνα που βασίζεται στο αν η λύση υπακούει στους περιορισμούς (feasibility-based rule).

Πλεονεκτήματα PSO

Ορισμένα από τα πλεονεκτήματα της PSO συνοψίζονται:

- Η PSO βασίζεται στη νοημοσύνη. Επίσης, μπορεί να εφαρμοστεί τόσο στην επιστημονική έρευνα όσο και για βιομηχανική χρήση.

- Δεν περιλαμβάνει μεταλλάξεις και διασταυρώσεις σωματιδίων.

- Η αναζήτηση εξαρτάται από την ταχύτητα των ατόμων.

Κατά τη διάρκεια της παραγωγής

των γενεών, μόνο το άτομο με τη βέλτιστη λύση μπορεί να μεταδώσει

πληροφορίες στα υπόλοιπα άτομα και η ταχύτητα αναζήτησης είναι πολύ μεγάλη.

- Ο υπολογισμός της PSO είναι πολύ απλός. Σε σύγκριση με άλλους αναπτυσσόμενες μεθόδους, η PSO έχει τη μεγαλύτερη ικανότητα βελτιστοποίησης και μπορεί να ολοκληρωθεί εύκολα.

- Η ποιότητα της λύσης δεν εξαρτάται από τον αρχικό πληθυσμό αλλά ξεκινώντας από οποιοδήποτε σημείο στο χώρο, η μέθοδος εξασφαλίζει τη σύγκλιση στη βέλτιστη λύση.

- Χρησιμοποιεί πιθανοκρατικούς και όχι ντετερμινιστικούς κανόνες μετάβασης

και έτσι είναι ένα είδος στοχαστικού αλγορίθμου βελτιστοποίησης ο οποίος μπορεί να ψάχνει λύση σε μία πολύπλοκη περιοχή. Αυτό προσδίδει στην PSO

μεγαλύτερη ευελιξία από τις συμβατικές μεθόδους.

4.2.5 Ο Αλγόριθμος Deathloff(9)

Το πρόβλημα VRPSPD έχει διαφορετικές εκδόσεις με μικρές παραλλαγές. Εδώ ορίζεται ως η επιλογή διαδρομών για περιορισμένο αριθμό οχημάτων που ξεκινούν και καταλήγουν σε μία αποθήκη που εξυπηρετεί μια ομάδα γεωγραφικά διάσπαρτων πελατών. Μαθηματικά, το πρόβλημα μπορεί να οριστεί σε ένα πλήρες γράφημα $G = (V_0, E)$ με $V_0 = \{0, 1, 2, \dots, n\}$ ως σύνολο κόμβων και E ως το σύνολο τόξου που ορίζεται μεταξύ κάθε ζεύγους κόμβων, $E = \{i, j \in V_0, i \neq j\}$.

Ο κόμβος 0 είναι η αποθήκη που λειτουργεί τόσο ως κέντρο διανομής όσο και ως κέντρο συλλογής για ένα στόλο οχημάτων ($K = \{1, 2, \dots, m\}$), το καθένα με χωρητικότητα Q_k . Κάθε πελάτης έχει μια σταθερή ζήτηση παράδοσης D_i και ζήτηση pickup P_i . Ο στόχος είναι να ελαχιστοποιηθεί η συνολική μεταφορά. Ο στόχος είναι να ελαχιστοποιηθεί το συνολικό κόστος μεταφοράς ενώ ικανοποιούνται τα ακόλουθα.

c_d : ζήτηση παράδοσης ανά πελάτη k , $k \in K$

c_t : το κόστος ταξιδιού ανά απόσταση μονάδας

d_{ij} : απόσταση μεταξύ δύο πελατών i και j , $i \in V, j \in V; i \neq j$

P_i : ποσότητα παραλαβής του πελάτη i , $i \in V$

D_i : ζήτηση παράδοσης του πελάτη i , $i \in V$

σ : μια παράμετρος που υποδηλώνει την αντιστάθμιση μεταξύ του κόστους αποστολής και του κόστους ταξιδιού, $\alpha \in [0, 1]$

M : ένας αυθαίρετα μεγάλος αριθμός.

Μεταβλητές απόφασης

x_{ijk} : μεταβλητή ένδειξη της κατάστασης κίνησης ενός οχήματος, $k \in K$, $x_{ijk} \in \{0, 1\}$.
 $x_{ijk} = 1$ αν πηγαίνει απευθείας από τον κόμβο $i \in V_0$ σε κόμβο $j \in V_0$; Αλλιώς $x_{ijk} = 0$

L_{0k} : φορτίο του οχήματος $k \in K$ όταν φεύγετε από την αποθήκη

L_j : φορτίο του οχήματος μετά την εξυπηρέτηση του πελάτη

Η μαθηματική σύνθεση του μικτού-ακέραιου προγραμματισμού του VRPSPD βασίζεται στο μοντέλο που προτάθηκαν από τον Dethloff (2001) και μπορούν να δηλωθούν ως εξής:

$$\text{Minimize } Z = \sigma \cdot \sum_{J \in V_0} \sum_{k \in K} c_d \cdot x_{0jk} + (1-\sigma) \cdot \sum_{i \in V_0} \sum_{J \in V_0} \sum_{k \in K} c_t \cdot d_{ijk} \cdot x_{ijk}$$

$$\text{περιορισμός} \quad \sum_{J \in V_0} \sum_{k \in K} x_{ijkd} = 1 \quad \forall j \in V$$

4.2.6 Ο Αλγόριθμος AMM(10)

Ο αυτόνομος αλγόριθμος εντοπισμού στόχων AMM (AMM) είναι ο απλούστερος τρόπος υλοποίησης όλων των αλγορίθμων που θα συζητηθούν εδώ. Αυτή η μέθοδος εκτελεί ένα φίλτρο Kalman για κάθε μοντέλο στο σετ μοντέλων και αφού ενημερωθεί ανεξάρτητα από κάθε μοντέλο, όλες οι ενημερώσεις αναμειγνύονται για μια συνολική εκτίμηση. Η συνολική έξοδος του αλγορίθμου παρακολούθησης είναι το άθροισμα των εξόδων όλων των μεμονωμένων φίλτρων που σταθμίζονται από τις πιθανότητες των μοντέλων τους.

Οι πιθανότητες του μοντέλου υπολογίζονται χρησιμοποιώντας την πιθανότητα του μοντέλου likelihood, $L_k^{(i)}$. Η πιθανότητα μοντέλου για το μοντέλο (i) υπολογίζεται χρησιμοποιώντας το σφάλμα πρόβλεψης μέτρησης, $\tilde{z}_k^{(i)}$ και η συνδιακύμανση σφάλματος πρόβλεψης μέτρησης $S_k^{(i)}$. Βασικά, ο $L_k^{(i)}$ μετρά το προβλεπόμενο σφάλμα πρόβλεψης μέτρησης από το μοντέλο (i) που είναι αληθές κατά την ώρα k δεδομένης της χρονικής στιγμής k-1

$$\text{Likelihood: } L_k^{(i)} \triangleq p[\tilde{z}_k^{(i)} | m_{(i)}^k, z^{k-1}] = \frac{e^{[-\frac{1}{2}(\tilde{z}_k^{(i)})(S_k^{(i)})^{-1}\tilde{z}_k^{(i)}]}}{|2\pi S_k^{(i)}|^{1/2}}$$

Η πιθανότητα λειτουργίας κανονικοποιείται έτσι ώστε :

$$\sum_{i \in M} \mu_k^{(i)} = 1$$

Αυτό θα πρέπει να έχει νόημα διαισθητικά επειδή το άθροισμα όλων των πιθανοτήτων πρέπει να είναι ίσο με την ενότητα. Η συνολική έξοδος του αλγόριθμου AMM συνενώνει όλες τις ενημερωμένες καταστάσεις από το σύνολο μοντέλων χρησιμοποιώντας τις πιθανότητες λειτουργίας και το θεώρημα συνολικής πιθανότητας. Ένας κύκλος του αλγορίθμου AMM έχει ως εξής:

1. Model-conditioned filtering (for $i=1,2,\dots,M$):

$$\left(\hat{\mathbf{x}}_{k-1|k-1}^{(i)}, \mathbf{P}_{k-1|k-1}^{(i)}\right) \rightarrow \left(\hat{\mathbf{x}}_{k|k}^{(i)}, \mathbf{P}_{k|k}^{(i)}, \tilde{\mathbf{z}}_k^{(i)} \mathbf{S}_k^{(i)}\right)$$

2. Mode probability update (for $i=1,2,\dots,M$):

$$\left(\mu_{k-1}^{(i)}, \tilde{\mathbf{z}}_k^{(i)}, \mathbf{S}_k^{(i)}\right) \rightarrow \left(\mu_k^{(i)}, \mathbf{L}_k^{(i)}\right)$$

3. Estimate fusion:

$$\left(\hat{\mathbf{x}}_{k|k}^{(i)}, \mathbf{P}_{k|k}^{(i)}, \mu_k^{(i)}\right) \rightarrow \left(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}\right)$$

4.2.7 Ο αλγόριθμος πλησιέστερου γείτονα (Largeneighborhoodsearch(LNS)) (11)

Πρόσφατες έρευνες σχετικά με τις τοπικές μεθόδους αναζήτησης υποδεικνύουν ότι οι μεγαλύτερες γειτονιές (largerneighborhoods) μπορεί να χρειαστούν για να λυθούν μερικά δύσκολα προβλήματα βελτιστοποίησης. Λόγω του μεγέθους των γειτονιών χρησιμοποιούνται διάφορα ευρετικά αναζήτησης στη γειτονιά ώστε να διατηρηθεί ο χρόνος πολυπλοκότητας σε λογικά επίπεδα. Αυτό σημαίνει ότι η απόδοση ενός τοπικού αλγορίθμου αναζήτησης περιορίζεται από την ποιότητα του ευρετικού που αναζητά στη γειτονιά. Για να αντιμετωπιστεί αυτή η δυσκολία, Οι Ropke και Pisinger πρότειναν τη χρήση πολλών ευρετικών για την αναζήτηση στη γειτονιά, όπου η συχνότητα χρήσης κάθε ευρετικού βασίζεται σε μερικά εμπειρικά στοιχεία από την έρευνα. Μια εκτεταμένη έκδοση αυτής της ευρετικής χρησιμοποιείται για τη λύση του δικού μας παραδείγματος.

Ο ευρετικός αλγόριθμος LNS (LargeNeighborhoodSearch) που προτάθηκε από τον Shaw(32).Ο ευρετικός μας επανειλημμένα περνάει από τα παρακάτω βήματα:

1. Επιλέγει ένα αλγόριθμο διαγραφής κόμβων R και ένα αλγόριθμο εισαγωγής κόμβων I

2. Διαγράφει ένα αριθμό q από κόμβους από τις διαδρομές χρησιμοποιώντας τον ευρετικό αλγόριθμο R
3. Εισάγει τους ελεύθερους (αυτούς που απομάκρυνε πριν) κόμβους σε κάποια από τις υπάρχουσες διαδρομές χρησιμοποιώντας τον ευρετικό αλγόριθμο I .
4. Υπολογίζει την αντικειμενική συνάρτηση της καινούριας λύσης
5. Εάν η τιμή της αντικειμενικής συνάρτησης βελτιώνεται τότε αποδέχεται την καινούρια λύση. Αλλιώς αποδέχεται την νέα λύση με μία πιθανότητα που εξαρτάται από την αύξηση της τιμής της αντικειμενικής συνάρτησης.

4.2.8 Ο Αλγόριθμος TG(12)

Ο αλγόριθμος TG (DRIESSENS, RAMON, BLOCKEEL, 2001) είναι ένας συνδυασμός του αλγορίθμου TILDE (BLOCKEEL, RAEDT, 1998), που κατασκευάζει μια πρώτη τάξη ταξινόμησης και παλινδρόμησης και του αλγορίθμου G (CHAPMAN- KAELEBLING, 1991), που χρησιμοποιεί μια σειρά στατιστικών τιμών που σχετίζονται με την απόδοση κάθε πιθανής επέκτασης για κάθε φύλλο του δέντρου για να το φτιάξει σταδιακά. Ο αλγόριθμος TG χρησιμοποιεί μια γλώσσα σχεσιακής αναπαράστασης για να περιγράψει τα παραδείγματα και τις δοκιμές που μπορούν να χρησιμοποιηθούν στο δέντρο παλινδρόμησης. Το RRL στέλνει όλα τα παραγόμενα παραδείγματα σε ένα επεισόδιο στο TG, το οποίο, ένα προς ένα, ελέγχει τα παραδείγματα μέσω του δέντρου μέχρι να φτάσει σε ένα φύλλο. Η τιμή Q του φύλλου που έχει επιτευχθεί ενημερώνεται με τη νέα τιμή (η τιμή Q που διατηρείται σε κάθε φύλλο είναι ο μέσος όρος των τιμών Q των παραδειγμάτων που το φτάνουν). Τα στατιστικά στοιχεία για κάθε φύλλο αποτελούνται από τον αριθμό των παραδειγμάτων στα οποία επιτυγχάνεται ή αποτυγχάνεται κάθε πιθανή δοκιμή καθώς και το άθροισμα των τιμών Q και το άθροισμα των τετραγωνικών τιμών Q για κάθε ένα από τα δύο υποσύνολα που δημιουργούνται από τη δοκιμή.

Algorithm 2.3 TG

```
//inputs: examples provided by the RRL
//output: TG tree
starts by creating a tree with a single leaf and empty statistics
for each training example that becomes available do
  sort the example down the tree using the tests of the internal nodes until it reaches
  a leaf
  updates the statistics in the leaf according to the new example
  if the statistics in the leaf indicate that a new split is needed then
    generate an internal node using the indicated test
    grow two new leaves with empty statistics
  end if
end for
```

Εάν η δοκιμή είναι σημαντική με μεγάλη εμπιστοσύνη, τότε το φύλλο χωρίζεται σε δύο νέα φύλλα και κάθε ένα από αυτά λαμβάνει μια τιμή Q με βάση τα στατιστικά στοιχεία που λαμβάνονται από τη δοκιμή που χρησιμοποιήθηκε για τη διάσπαση του γονικού κόμβου. Αργότερα, αυτή η τιμή ενημερώνεται καθώς τα νέα παραδείγματα ταξινομούνται στο φύλλο. Ο αλγόριθμος TG λαμβάνει ως εισροές τα παραδείγματα που παρέχονται από το RRL και παρέχει ως έξοδο ένα δέντρο με τις αποθηκευμένες γνώσεις. Για παράδειγμα, πάρτε ένα νεοσύστατο φύλλο, με μηδενικά στατιστικά στοιχεία. Σε κάθε επανάληψη από το σύστημα, παράγονται και παραδίδονται νέα παραδείγματα στο δέντρο. Κάθε δείγμα ταξινομείται μέσω του δέντρου μέχρι να φτάσει σε ένα φύλλο όπου αποθηκεύεται και η τιμή Q του φύλλου ενημερώνεται με την τιμή Q του παραδείγματος (η τιμή Q του φύλλου είναι η μέση τιμή των τιμών Q από τα δείγματα που έχουν αποθηκευτεί στο φύλλο). Αφού αποθηκευτεί ένα ελάχιστο μέγεθος δείγματος (MSS) σε ένα φύλλο, ο αλγόριθμος TG εφαρμόζει όλες τις πιθανές δοκιμές σε αυτό το σύνολο παραδειγμάτων και επαληθεύει ποια προσφέρει τον καλύτερο διαχωρισμό των παραδειγμάτων σε δύο υποσύνολα. Αυτός που προσφέρει την καλύτερη διάσπαση των παραδειγμάτων στο φύλλο επιλέγεται ως δοκιμή φύλλου και από αυτό δημιουργούνται δύο νέα φύλλα με μηδενικά στατιστικά στοιχεία.

Οι δοκιμές μπορούν να είναι οι εξής:

- equal (X, Y) το μπλοκ X είναι ίσο με το μπλοκ Y με την έννοια ότι το μπλοκ X είναι το μπλοκ Y
- on (X, Y) X είναι το μπλοκ απευθείας στο μπλοκ Y
- above (X, Y) το μπλοκ X είναι στη στοίβα των μπλοκ που βρίσκονται στο μπλοκ Y
- clear (X) το μπλοκ X είναι ελεύθερο

Μόλις χτιστεί μετά το στάδιο εκμάθησης, το δέντρο στη συνέχεια χρησιμοποιείται ως βασική γνώση για να βοηθήσει τον πράκτορα να επιλέξει μελλοντικές ενέργειες. Όντας αυξητικός, ο αλγόριθμος TG δεν χρειάζεται να ξαναχτίσει το

δέντρο σε κάθε επεισόδιο και δεν χρειάζεται να κρατήσει όλα τα παραγόμενα παραδείγματα κατά τη διάρκεια των διαφόρων επεισοδίων.

4.2.9 Ο Αλγόριθμος GTS(13)

Για να λύσουμε το πρόβλημα VRPSPD, αναπτύξαμε μια αλγοριθμική μεθοδολογία (που δηλώνεται GTS) συνδυάζοντας τις στρατηγικές αναζήτησης από δύο γνωστές μεθευριστικές μεθόδους, δηλαδή την αναζήτηση tabu (TS) και την τοπική αναζήτηση (GLS), οι οποίες και οι δύο έχουν αποδειχθεί ότι είναι πολύ αποτελεσματικές για το βασικό VRP και τις παραλλαγές του. (Gendreau, Hertz, & Laporte, 1994, Nanry&Barnes, 2000; Zhong&Cole, 2005; Mester&Bräysy, 2007; Taran-tilis κ.ά., 2007)

Ο αλγόριθμος GTS αποσκοπεί στην επίτευξη ικανοποιητικής απόδοσης αλληλεπίδρασης της εντατικοποίησης και της διαφοροποίησης της πραγματοποιήσιμης αναζήτησης. Επιπλέον, για να εξασφαλιστεί η ευρωστία της μεθόδου μας, διατηρήθηκε ο αριθμός των παραμέτρων αναζήτησης στο ελάχιστο. Ο προτεινόμενος αλγόριθμος ξεκινάει από μία παραγόμενη αρχική εφικτή λύση μέσω της εξοικονόμησης κόστους η οποία στη συνέχεια βελτιώνεται από την υβριδική μεθοδολογία TS-GLS.

Η αρχική λύση VRPSPD παράγεται από την ευρετική δομή με βάση την εξοικονόμηση κόστους (Paessens, 1988). Αυτή η μέθοδος ξεκινά δημιουργώντας μια νέα διαδρομή, η οποία είναι κενή από πελάτες και η κεντρική αποθήκη κενή από προϊόν. Στη συνέχεια υπολογίζεται το κόστος εισαγωγής για κάθε πελάτη (ακόμη μη επιλεγμένο) – που είναι εφικτή η εισαγωγή του στη διαδρομή. Η μέτρηση κόστους για κάθε εισαγόμενο πελάτη ορίζεται ως v_k , μεταξύ των κορυφών v_i και v_j ως $cost_{ijk} = c_{ik} + c_{kj} \cdot g \cdot c_{ij} + f_j c_{ik} c_{kj}$, όπου τα g και f είναι κατανεμημένα στοχαστικά στα διαστήματα $[0, 3]$ και $[0, 1]$, αντίστοιχα. Τα εύρη τιμών των παραμέτρων f και g καθορίστηκαν από τον Paessens (1988), και οδηγούν την κατασκευή των αρχικών λύσεων. Για κάθε πελάτη i , καθορίζεται το ελάχιστο κόστος εισαγωγής του στη διαδρομή του που δηλώνεται $mcost_i$.

Η διαδικασία περιλαμβάνει την εισαγωγή του πελάτη i , στη θέση mr_i για τη διαδρομή mr_i . Θέτουμε m_c τον πελάτη με τη χαμηλότερη τιμή $mcost_i$. Για κάθε

πελάτη j , με $mr_j = mrmc$ και $mp_j = mpmc$, το μέτρο του κόστους εισαγωγής $mcost_0$ j αξιολογείται ως $mcost_0 j \frac{1}{4} k \delta c_j 0 \leq c_0 j \leq mcost_j$, όπου k είναι ομοιόμορφα κατανομημένη στο διάστημα $[0, 1]$, ενώ $c_j 0$ και $c_0 j$ υποδηλώνουν το κόστος για τη μετάβαση από v_j στη κεντρική αποθήκη και από την κεντρική αποθήκη έως v_j , αντίστοιχα. Θέτουμε ins τον πελάτη για τον οποίο το $mcost_0$ είναι ελάχιστο. Ο πελάτης ins τοποθετείται στη θέση $mpins$. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να είναι κάθε πελάτης σε κάποια διαδρομή. Τη στιγμή που ένας πελάτης εκχωρηθεί σε μια κενή διαδρομή, δημιουργείται μια νέα διαδρομή η οποία είναι διαθέσιμη, χωρίς κανένα όριο που επιβάλλεται στο μέγεθος της παραγόμενης διαδρομής.

ΚΕΦΑΛΑΙΟ 5

Αποτελέσματα

Στον Πίνακα 5.1 γίνεται η παρουσίαση των αποτελεσμάτων που έδωσαν οι αλγόριθμοι που μελετούνται στα δεδομένα που χρησιμοποιούνται στο παράδειγμα SCA3-0 έως SCA8-9.

Στον πίνακα 5.2 φαίνονται οι διαφορές των παραδειγμάτων από το κάθε ελάχιστο.

| Instance | Dethloff (2001). | T&G | GTS | TS sol. Value | LNS | ACO | AMM | h_PSO | ACS |
|----------|------------------|-----------|-----------|---------------|-----------|-----------|-----------|-----------|-----------|
| SCA3-0 | 689,0000 | 640,5500 | 636,0600 | 583,7700 | 636,1000 | 635,6200 | 635,6200 | 635,6200 | 635,6200 |
| SCA3-1 | 765,6000 | 697,8400 | 697,8400 | 655,6300 | 697,8000 | 697,8400 | 697,8400 | 697,8400 | 697,8400 |
| SCA3-2 | 742,8000 | 659,3400 | 659,3400 | 627,1200 | 659,3000 | 659,3400 | 659,3400 | 659,3400 | 659,3400 |
| SCA3-3 | 737,2000 | 680,0400 | 680,0400 | 633,5600 | 680,6000 | 680,0400 | 680,0400 | 680,0400 | 680,0400 |
| SCA3-4 | 747,1000 | 690,5000 | 690,5000 | 642,8900 | 690,5000 | 690,5000 | 690,5000 | 690,5000 | 690,5000 |
| SCA3-5 | 784,4000 | 659,9000 | 659,9000 | 603,0600 | 659,9000 | 659,9000 | 659,9000 | 659,9000 | 659,9000 |
| SCA3-6 | 720,4000 | 653,8100 | 651,0900 | 607,5300 | 651,1000 | 651,0900 | 651,0900 | 651,0900 | 651,0900 |
| SCA3-7 | 707,9000 | 659,1700 | 659,1700 | 616,4000 | 666,1000 | 659,1700 | 659,1700 | 659,1700 | 659,1700 |
| SCA3-8 | 807,2000 | 719,4700 | 719,4700 | 668,0400 | 719,5000 | 719,4700 | 719,4700 | 719,4800 | 719,4700 |
| SCA3-9 | 764,1000 | 681,0000 | 681,0000 | 619,0300 | 681,0000 | 681,0000 | 681,0000 | 681,0000 | 681,0000 |
| SCA8-0 | 1132,9000 | 981,4700 | 961,5000 | 877,5500 | 975,1000 | 961,5000 | 961,5000 | 961,5000 | 961,5000 |
| SCA8-1 | 1150,9000 | 1077,4400 | 1050,2000 | 954,2900 | 1052,4000 | 1049,6500 | 1049,6500 | 1049,6500 | 1050,3800 |
| SCA8-2 | 1100,8000 | 1050,9800 | 1039,6400 | 950,7400 | 1044,5000 | 1042,6900 | 1039,6400 | 1039,6400 | 1044,4800 |
| SCA8-3 | 1115,6000 | 983,3400 | 983,3400 | 905,2900 | 999,1000 | 983,3400 | 983,3400 | 983,3400 | 983,3400 |
| SCA8-4 | 1235,4000 | 1073,4600 | 1065,4900 | 972,6200 | 1065,5000 | 1065,4900 | 1065,4900 | 1065,4900 | 1065,4900 |
| SCA8-5 | 1231,6000 | 1047,2400 | 1027,0800 | 940,6000 | 1027,1000 | 1027,0800 | 1027,0800 | 1027,0800 | 1027,0800 |
| SCA8-6 | 1062,5000 | 995,5900 | 971,8200 | 885,3400 | 977,0000 | 971,8200 | 971,8200 | 971,8200 | 971,8200 |
| SCA8-7 | 1217,4000 | 1068,5600 | 1052,1700 | 955,8600 | 1061,0000 | 1052,1700 | 1051,2800 | 1051,2800 | 1063,1500 |
| SCA8-8 | 1231,6000 | 1080,5800 | 1071,1800 | 986,5200 | 1071,2000 | 1071,1800 | 1071,1800 | 1071,1800 | 1071,1800 |
| SCA8-9 | 1185,6000 | 1084,8000 | 1060,5000 | 978,9000 | 1060,5000 | 1060,5000 | 1060,5000 | 1060,5000 | 1061,2300 |

Πίνακας 5.1

Το ελάχιστο κόστος / η ελάχιστη τιμή που καταγράφηκε για όλα τα υποπαραδείγματα του παραδείγματος SCA3-0 έως SCA8-9 παρατηρείτε από τον αλγόριθμο Περιορισμένη Αναζήτηση (TabuSearch(TS)).

Ο τύπος που χρησιμοποιήθηκε για την καταγραφή των διαφορών ανάμεσα στα αποτελέσματα των αλγορίθμων είναι για τον κάθε αλγόριθμο : $((\text{τιμή-βέλτιστη λύση})/\text{βέλτιστη λύση}) * 100\%$.

| ΔΙΑΦΟΡΕΣ ΑΠΌ ΤΟ ΜΙΝ | | | | | | | | |
|---------------------|--------|--------|---------------|--------|--------|--------|--------|--------|
| Dethloff (2001). | T&G | GTS | TS sol. Value | LNS | ACO | AMM | h_PSO | ACS |
| 0,1803 | 0,0973 | 0,0896 | 0,0000 | 0,0896 | 0,0888 | 0,0888 | 0,0888 | 0,0888 |
| 0,1677 | 0,0644 | 0,0644 | 0,0000 | 0,0643 | 0,0644 | 0,0644 | 0,0644 | 0,0644 |
| 0,1845 | 0,0514 | 0,0514 | 0,0000 | 0,0513 | 0,0514 | 0,0514 | 0,0514 | 0,0514 |
| 0,1636 | 0,0734 | 0,0734 | 0,0000 | 0,0742 | 0,0734 | 0,0734 | 0,0734 | 0,0734 |
| 0,1621 | 0,0741 | 0,0741 | 0,0000 | 0,0741 | 0,0741 | 0,0741 | 0,0741 | 0,0741 |
| 0,3007 | 0,0943 | 0,0943 | 0,0000 | 0,0943 | 0,0943 | 0,0943 | 0,0943 | 0,0943 |
| 0,1858 | 0,0762 | 0,0717 | 0,0000 | 0,0717 | 0,0717 | 0,0717 | 0,0717 | 0,0717 |
| 0,1484 | 0,0694 | 0,0694 | 0,0000 | 0,0806 | 0,0694 | 0,0694 | 0,0694 | 0,0694 |
| 0,2083 | 0,0770 | 0,0770 | 0,0000 | 0,0770 | 0,0770 | 0,0770 | 0,0770 | 0,0770 |
| 0,2344 | 0,1001 | 0,1001 | 0,0000 | 0,1001 | 0,1001 | 0,1001 | 0,1001 | 0,1001 |
| 0,2910 | 0,1184 | 0,0957 | 0,0000 | 0,1112 | 0,0957 | 0,0957 | 0,0957 | 0,0957 |
| 0,2060 | 0,1290 | 0,1005 | 0,0000 | 0,1028 | 0,0999 | 0,0999 | 0,0999 | 0,1007 |
| 0,1578 | 0,1054 | 0,0935 | 0,0000 | 0,0986 | 0,0967 | 0,0935 | 0,0935 | 0,0986 |
| 0,2323 | 0,0862 | 0,0862 | 0,0000 | 0,1036 | 0,0862 | 0,0862 | 0,0862 | 0,0862 |
| 0,2702 | 0,1037 | 0,0955 | 0,0000 | 0,0955 | 0,0955 | 0,0955 | 0,0955 | 0,0955 |
| 0,3094 | 0,1134 | 0,0919 | 0,0000 | 0,0920 | 0,0919 | 0,0919 | 0,0919 | 0,0919 |
| 0,2001 | 0,1245 | 0,0977 | 0,0000 | 0,1035 | 0,0977 | 0,0977 | 0,0977 | 0,0977 |
| 0,2736 | 0,1179 | 0,1008 | 0,0000 | 0,1100 | 0,1008 | 0,0998 | 0,0998 | 0,1122 |
| 0,2484 | 0,0953 | 0,0858 | 0,0000 | 0,0858 | 0,0858 | 0,0858 | 0,0858 | 0,0858 |
| 0,2112 | 0,1082 | 0,0834 | 0,0000 | 0,0834 | 0,0834 | 0,0834 | 0,0834 | 0,0841 |

Πίνακας 5.2

Βιβλιογραφία

1. **ΙΩΑΝΝΗΣ, ΜΑΡΙΝΑΚΗΣ ΚΑΙ ΑΘΑΝΑΣΙΟΣ, ΜΥΓΔΑΛΑΣ.ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΕΦΟΔΙΑΣΤΙΚΗΣ ΑΙΥΣΙΔΑΣ.** 2008.
2. **Bodin L.D., Golden B.L.,Assad A.A and Ball M.O.** Routing and Scheduling of Vehicles and Crews,Computers and Operations Research. 1983.
3. **N, Christofides.** Vehicle Routing.The travelling sales-man problem. 1985.
4. **I.H., Osman.** Metastrategy Simulated Annealing and Tabu Search Algorithms for Combinatorial Optimization Problems. Anna'sofOperationalResearch. 1993.
5. **Ευάγγελος, Καλπουρτζής.** Επίλυση του Προβλήματος Δρομολόγησης Οχημάτων Παραλαβής και Επίδοσης με Αλγόριθμο Προσομοιωμένης ανόπτησης. 2010.
6. **M.Βάια, Ζυγούρα.**ACO:Αλγόριθμος εμπνευσμένος από την φύση για ad-hoc δίκτυα και clouds. 2013.
7. **Yuvraj Gajpal, Prakash Abad.** An ant colony system(ACS)for vehicle rooting problem with simultaneous delivery and pickup. 2009.
8. **Emmanuel E, Zachariadis,Christos D. Tarantilis,Chris T.Kiranoudis.** An adeptive memory methodology for vehicle routing problem with simultaneous pick up and deliveries. 2010.
9. **A.Subramanian, L.M.A. Drummond,C.Bentes,L.S.Ochi,R.Farias.** A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery.
10. **Pitre, Ryan.** ACoparison of Multiple-Model Tracking Algorithms. 2004.
11. **Stefan Ropke, David Pisinger.** A unified heuristic for a large class of vehicle,Diku-Department of computer Science, University of Copenhagen.
12. **Millan.** Relation Reinforcement Learning with the Algorithm TG. 2010.

13. **Emmanuel E. Zachariadis, Christos D. Tarantilis.** A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick up service. 2009.