Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Μελέτη περίπτωσης ενός προβλήματος δρομολόγησης οχημάτων με περιορισμένη χωρητικότητα (CVRP)** **Case Study of a Capacitated Vehicle Routing Problem (CVRP)**
Ονοματεπώνυμο Φοιτητή	**Κωνσταντίνος Σιδεριάδης**
Πατρώνυμο	**Θεόδωρος**
Αριθμός Μητρώου	**ΜΠΠΛ/ 13071**
Επιβλέπων	**Καθηγητής Τσιχριντζής Γ.**

Ημερομηνία Παράδοσης **Ιανουάριος 2016**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Γεώργιος Τσιχριντζής Καθηγητής	Μαρία Βίρβου Καθηγήτρια	Αποστόλου Καθηγητής

Περίληψη

Στα πλαίσια αυτής της μεταπτυχιακής διατριβής πραγματοποιήθηκε η μελέτη για τα προβλήματα δρομολόγησης στόλων οχημάτων και φορτηγών στον κλάδο της εφοδιαστικής αλυσίδας (Logistics). Ειδικότερα, η μελέτη στην αρχή περιγράφει τη σχέση εντός ενός δικτύου δρομολόγησης και του ρόλου του στην εφοδιαστική αλυσίδα. Στη συνέχεια αναφέρονται τα βασικά χαρακτηριστικά όλων των προβλημάτων προς αντιμετώπιση σε ένα δίκτυο δρομολόγησης. Αφού αναλυθούν οι βασικές κατηγορίες γίνεται η ανάλυση του βασικού προβλήματος που ερευνάται σε αυτή τη μελέτη περίπτωσης. Πιο συγκεκριμένα, γίνεται η υπόθεση οτι το πρόβλημα ανήκει στην κατηγορία των προβλημάτων δρομολόγησης οχημάτων με περιορισμένη χωρητικότητα (Capacitated Vehicle Routing Problem). Αναλαμβάνεται για επίλυση ένα πρόβλημα εφοδιασμού 100 πελατών, με κάθε φορτηγό να έχει χωρητικότητα 9 τόνους και ο μέγιστος χρόνος που μπορούν να εκτελούν διαδρομές είναι 9 ώρες. Η μελέτη ερευνά τον τρόπο με τον οποίο θα δομηθεί ο αλγόριθμος που θα αναλάβει να βελτιστοποιήσει το σύστημα "αποθήκης - πελατών". Πιο αναλυτικά, το πρόγραμμα χρησιμοποιεί τις βασικές αρχές του αλγορίθμου αποικίας μυρμηγκιών για την εύρεση της βέλτιστης διαδρομής και συνδυάζει τις βασικές αρχές μετα-ευρεστικών εργαλείων για την βελτιστοποίηση της βέλτιστης διαδρομής. Αφού περιγραφεί ο τρόπος με τον οποίο θα αναλάβει να αντιμετωπίσει το πρόβλημα ο υβριδικός αλγόριθμος, δίνεται και η μαθηματική παρουσίασή του. Τέλος, δίνεται η δέσμευση οτι συνεχίζοντας τις έρευνες στο επόμενο διάστημα, σε νέα έκδοση της εργασίας, θα έχει υλοποιηθεί ο αλγόριθμος και θα έχουν πραγματοποιηθεί και συγκρίσεις με ήδη υπάρχοντα εργαλεία.

Λέξεις κλειδιά: Εφοδιαστική αλυσίδα, πρόβλημα δρομολόγησης οχημάτων με περιορισμένη χωρητικότητα, αλγόριθμος βελτιστοποίησης αποικίας μυρμηγκιών, μεταευρεστικοί αλγόριθμοι.

Abstract

For the purpose of this master thesis was conducted a study for the vehicle routing problems in the Logistics department of a company. At first, the study describes the relationships inside a routing network and its role in the supply chain. Then are described the main features of all the problems to be addressed in a routing network. After analyzing the main categories of problems, follows the analysis of the basic problem investigated in this case study. More specifically, it is assumed that the problem belongs to the class of Capacitated Vehicle Routing Problems. This class has the distinction of a limited capacity for the vehicles. The problem taken to resolve has a number of 100 customers, with each truck having a capacity of 9 tons per truck and the maximum time that a truck can operate is 9 hours at most. The study investigates the way in which the algorithm that will optimize the system "warehouse - customer" is constructed. More specifically, the algorithm uses the basic principles of ant colony optimization for finding the optimal path and combines it with the basic principles of the best meta-heuristic tools to optimize it even more. Having described the way in which the hybrid algorithm will suit the problem and solve it, what follows next is the mathematical presentation of what is described. Finally, it is given a commitment that the investigations will continue in the near future and a new addition in the work will be released, where the algorithm will be implemented and some comparisons between existing tools will be carried out.

Keywords: Logistics, Capacitated VehicleRouting Problem, hybrid algorithm, ant colony optimization algorithm, meta-heuristics.

Περιεχόμενα

Εισαγωγή

Η εξέλιξη της έννοιας και των βασικών αρχών των logistics και της εφοδιαστικής αλυσίδας είναι άμεσα συνδεδεμένες με την εξέλιξη της βιομηχανίας και του εμπορίου. Την τελευταία δεκαετία παρατηρείται μια ραγδαία εξέλιξη στον τρόπο δραστηριοποίησης των επιχειρήσεων, μιας και το επιχειρηματικό περιβάλλον συνεχώς μεταβάλλεται. Η παγκοσμιοποίηση του εμπορίου μέσω του Διαδικτύου (Internet) σε συνδυασμό με τον ανταγωνισμό κάνει αναγκαία την ελαχιστοποίηση του κόστους παραγωγής και διάθεσης των προϊόντων, τη μείωση των αποθεμάτων, την ακριβέστερη πρόβλεψη της ζήτησης, καθώς επίσης και τη μείωση των χρόνων παράδοσης. Το αβίαστο, λοιπόν, συμπέρασμα των ανωτέρω είναι οτι τα Logistics χαράσσουν την πολιτική και τη στρατηγική μιας επιχείρησης σε πολύ μεγάλο βαθμό και σε αρκετές περιπτώσεις.

Τα Logistics αποτελούν εδώ και πάρα πολλά χρόνια πεδίο διεπιστημονικής έρευνας και ενδιαφέροντος για τις επιχειρήσεις. Το δίκτυο διαχείρισης της εφοδιαστικής αλυσίδας είναι υπεύθυνο για την κίνηση των υλικών (μέσα και έξω από την παραγωγική διαδικασία), την διαδικασία διαχείρισης των αποθεμάτων, τις πωλήσεις, την διανομή από την αποθήκη έως και τον τελικό καταναλωτή, όπως επίσης και για την ενημέρωση της επιχείρησης σχετικά με ελαττωματικά προϊόντα προκειμένου να επιδιορθωθούν και να επαναπωληθούν.

Οι αρμοδιότητες του τομέα των Logistics είναι αρκετές. Κάθε μια από αυτές χρειάζεται ειδική αντιμετώπιση στα θέματα που ανακύπτουν. Με την όσο το δυνατόν πιο άρτια αντιμετώπιση αυτών των ζητημάτων, επιτρέπεται σε μια επιχείρηση να επιτυγχάνει την μακροπρόθεσμη κερδοφορία και να εδραιώνεται στην αγορά. Μερικά παραδείγματα τέτοιων ζητημάτων αποτελούν προβλήματα εφοδιασμού στην παραγωγική διαδικασία ή προβλήματα διανομής των τελικών προϊόντων από τις εγκαταστάσεις στους τελικούς καταναλωτές.

Στην παρούσα μελέτη περίπτωσης ερευνάται ένα πρόβλημα διανομής, στο οποίο μια επιχείρηση θέλει να εξυπηρετήσει τους πελάτες της διανέμοντάς τους τις παραγγελίες τους. Με δεδομένο οτι οι εγκαταστάσεις της επιχείρησης είναι σταθερές και δεν θα μεταβληθούν μακροπρόθεσμα, το πρόβλημα της διανομής περιορίζεται στον αριθμό των πελατών που θα εξυπηρετηθούν και τον συνολικό αριθμό των χιλιομέτρων που θα διανυθούν. Σύμφωνα με τα πλάνα της επιχείρησης κάθε πελάτης της είναι συνδεδεμένος με μια συγκεκριμένη αποθήκη στην περιοχή που διαμένει, που αναλαμβάνει να τον εξυπηρετεί. Τα φορτηγά θα αναχωρήσουν από την αποθήκη για να εξυπηρετήσουν τις παραγγελίες, αλλά θα κινηθούν σε διαδρομές που περιλαμβάνουν περισσότερους του ενός πελάτες. Με αυτή τη λογική, στην μελέτη μας δεχόμαστε οτι στην αποθήκη που ερευνάται αντιστοιχούν 100 πελάτες. Επομένως, η σχέση που μένει να διερευνηθεί είναι μεταξύ του αριθμού των πελατών και του αριθμού των χιλιομέτρων που απέχει κάθε ένας από αυτούς από την αποθήκη.

Η σχέση μεταξύ των πελατών και των χιλιομέτρων είναι γραμμική και από την παρακάτω συνάρτηση:

$$f(x) = ax + by \quad (1)$$

, όπου α, b είναι οι σταθερές του συστήματος, με y συμβολίζεται ο αριθμός των πελατών και με x συμβολίζεται ο αριθμός των χιλιομέτρων. Για τους πελάτες ισχύει οτι η συνολική ποσότητα είναι ίση με το άθροισμα όλων των παραγγελιών και των 100 πελατών:

$$\mathrm{y} = \mathrm{y1} + \mathrm{y2} + \mathrm{y3} + \cdots + \mathrm{y100} \quad (2)$$

Ζητείται να ελαχιστοποιηθεί το σύστημα λαμβάνοντας ως αντικειμενική συνάρτηση την (1) και ως περιορισμούς τα παρακάτω δεδομένα:

- Κάθε φορτηγό έχει 10 δεξαμενές των 1000lt με διαφορετικά προϊόντα.
- Θα πρέπει να χρησιμοποιηθεί ο ελάχιστος αριθμός φορτηγών.
- Κάθε φορτηγό μπορεί να λειτουργεί το μέγιστο 9 ώρες την ημέρα.
- Τα φορτηγά είναι ίδια μεταξύ τους για να μην επηρρεάζουν τα αποτελέσματα ως εξωτερικοί παράγοντες και έχουν χωρητικότητα 23tn.

1. Γενικές πληροφορίες

1.1 Logistics - Εφοδιαστική Αλυσίδα

Η εξέλιξη της έννοιας και των βασικών αρχών των logistics και της εφοδιαστικής αλυσίδας είναι άμεσα συνδεδεμένες με την εξέλιξη της βιομηχανίας και του εμπορίου. Την τελευταία δεκαετία παρατηρείται μια ραγδαία εξέλιξη στον τρόπο δραστηριοποίησης των επιχειρήσεων, μιας και το επιχειρηματικό περιβάλλον συνεχώς μεταβάλλεται. Η παγκοσμιοποίηση του εμπορίου μέσω του Διαδικτύου (Internet) σε συνδυασμό με τον ανταγωνισμό κάνει αναγκαία την ελαχιστοποίηση του κόστους παραγωγής και διάθεσης των προϊόντων, τη μείωση των αποθεμάτων, την ακριβέστερη πρόβλεψη της ζήτησης, καθώς επίσης και τη μείωση των χρόνων παράδοσης. Το αβίαστο, λοιπόν, συμπέρασμα των ανωτέρω είναι οτι τα Logistics χαράσσουν την πολιτική και τη στρατηγική μιας επιχείρησης σε πολύ μεγάλο βαθμό και σε αρκετές περιπτώσεις.

Με τον όρο Logistics στην βιβλιογραφία αναφέρεται το σύνολο των ενεργειών ώστε να επιτυγχάνεται η μείωση του κόστους και η αύξηση της ποιότητας των αγαθών και των υπηρεσιών που θα οδηγήσουν μια επιχείρηση στην κερδοφορία. Τα Logistics αποτελούν εδώ και πάρα πολλά χρόνια πεδίο διεπιστημονικής έρευνας και ενδιαφέροντος για τις επιχειρήσεις. Ο τομέας των Logistics είναι υπεύθυνος για την κίνηση των υλικών (μέσα και έξω από την παραγωγική διαδικασία), την διαδικασία διαχείρισης των αποθεμάτων, τις πωλήσεις, την διανομή από την αποθήκη έως και τον τελικό καταναλωτή, όπως επίσης και για την ενημέρωση της επιχείρησης σχετικά με ελαττωματικά προϊόντα προκειμένου να επιδιορθωθούν και να επαναπωληθούν.

Εικόνα 1.1: Αναπαράσταση εφοδιαστικής αλυσίδας

Οι αρμοδιότητες του τομέα των Logistics είναι αρκετές. Ωστόσο, το βασικότερο πεδίο εφαρμογής των Logistics εδώ και πάρα πολλά χρόνια αποτελεί η εφοδιαστική αλυσίδα. Η εφοδιαστική αλυσίδα (supply chain) είναι το σύστημα του οποίου όλα τα συστατικά στοιχεία εμπλέκονται άμεσα ή έμμεσα στη μεταφορά προϊόντων, υπηρεσιών ή πληροφοριών ώστε να ικανοποιηθούν οι απαιτήσεις των πελατών.

Ο J. Aitner εξελίσσει τον ορισμό της εφοδιαστικής αλυσίδας αναφέροντας στο βιβλίο του Κ. Σιφνιώτη " Logistics Management: Θεωρία & Πράξη" ότι: «Η εφοδιαστική αλυσίδα αποτελεί ένα δίκτυο συνδεδεμένων και αλληλεξαρτημένων οργανώσεων, που λειτουργούν από κοινού σε ένα κλίμα συνεργασίας για να ελέγξουν, να διευθύνουν και να βελτιώσουν τη ροή υλικών και πληροφοριών από τους προμηθευτές στους τελικούς χρήστες».

Μια εφοδιαστική αλυσίδα αποτελείται από προμηθευτές, κατασκευαστές, πωλητές, πρώτες ύλες, κέντρα διανομών, εγκαταστάσεις αποθήκευσης, μεταφορείς και πελάτες. Όλα αυτά τα συστατικά στοιχεία της εφοδιαστικής αλυσίδας εκτελούν το καθένα διαφορετικές διαδικασίες αλλά αλληλεπιδρούν μεταξύ τους. Όσο πιο αρμονικά αλληλεπιδρούν μεταξύ τους, τόσο περισσότερο επιτρέπεται σε μια επιχείρηση να επιτυγχάνει συγκριτικό πλεονέκτημα και να εδραιώνεται στην αγορά. Έτσι ο M. Christofer, ένας από τους σημαντικότερους μελετητές στον τομέα της εφοδιαστικής αλυσίδας, υποστηρίζει πως ο ανταγωνισμός των εταιρειών θα γίνεται πλέον μέσω των εφοδιαστικών αλυσίδων τόσο προς την κατεύθυνση της μείωσης του κόστους, όσο και προς την κατεύθυνση του βέλτιστου συντονισμού των διεργασιών της επιχείρησης.

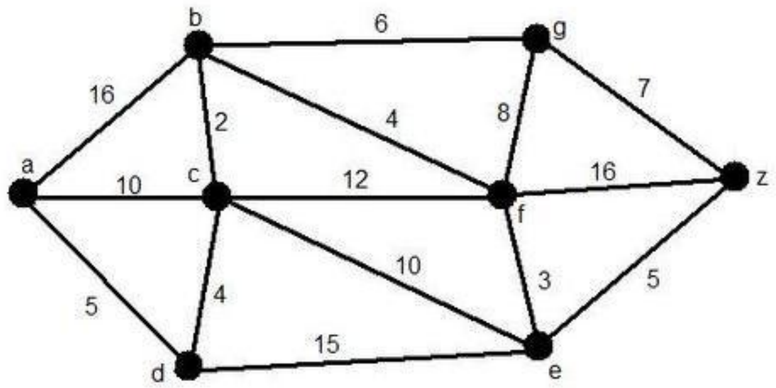
Η διαδικασία του σχεδιασμού, της οργάνωσης και του προγραμματισμού της αποτελεσματικής ροής των αγαθών καθώς και του ελέγχου και του συντονισμού όλων των εργασιών ανάλογα με τις απαιτήσεις των πελατών αποτελεί αντικείμενο του τομέα της διαχείρισης της εφοδιαστικής αλυσίδας (Logistics Management) μιας επιχείρησης. Η λειτουργία της εφοδιαστικής αλυσίδας επηρεάζει και επηρεάζεται τόσο από τα περισσότερα τμήματα της ίδιας επιχείρησης όσο και από τους πελάτες και τους προμηθευτές της. Οι παράγοντες που με πιο αποτελεσματική διαχείριση θα οδηγήσουν σε μεγαλύτερη κερδοφορία και ευελιξία ως προς τον ανταγωνισμό είναι:

- Αγορές
- Αποθέματα
- Μεταφορές
- Αποθήκευση
- Διανομή

Επομένως, συνοψίζοντας, σκοπός των logistics και της διαχείρισης της εφοδιαστικής αλυσίδας είναι να μεταφερθεί το σωστό προϊόν, στη σωστή ποσότητα, στο σωστό τόπο, στο σωστό χρόνο και με το μικρότερο δυνατό κόστος.

1.2 Η έννοια της Δρομολόγησης

Η δρομολόγηση συνιστά καθοριστικό παράγοντα για την αποδοτική ροή αγαθών. Ως γενική έννοια σχετίζεται με την ανεύρεση και τη σχεδίαση της βέλτιστης διαδρομής πάνω σε έναν γράφο, του οποίου οι ακμές (δυνατές μεταβάσεις μεταξύ των κόμβων) χαρακτηρίζονται από κόστη. Τα κόστη μπορούν να αντιπροσωπεύουν χιλιομετρικές αποστάσεις ή χρονικές καθυστερήσεις ανάλογα με τη φύση του προβλήματος που εξετάζεται. Ως βέλτιστη διαδρομή χαρακτηρίζεται ως η διαδρομή που οδηγεί από τον έναν κόμβο στον επόμενο με το κόστος να είναι το ελάχιστο.



Εικόνα 1.2: Δρομολόγηση σε ένα δίκτυο υπολογιστών

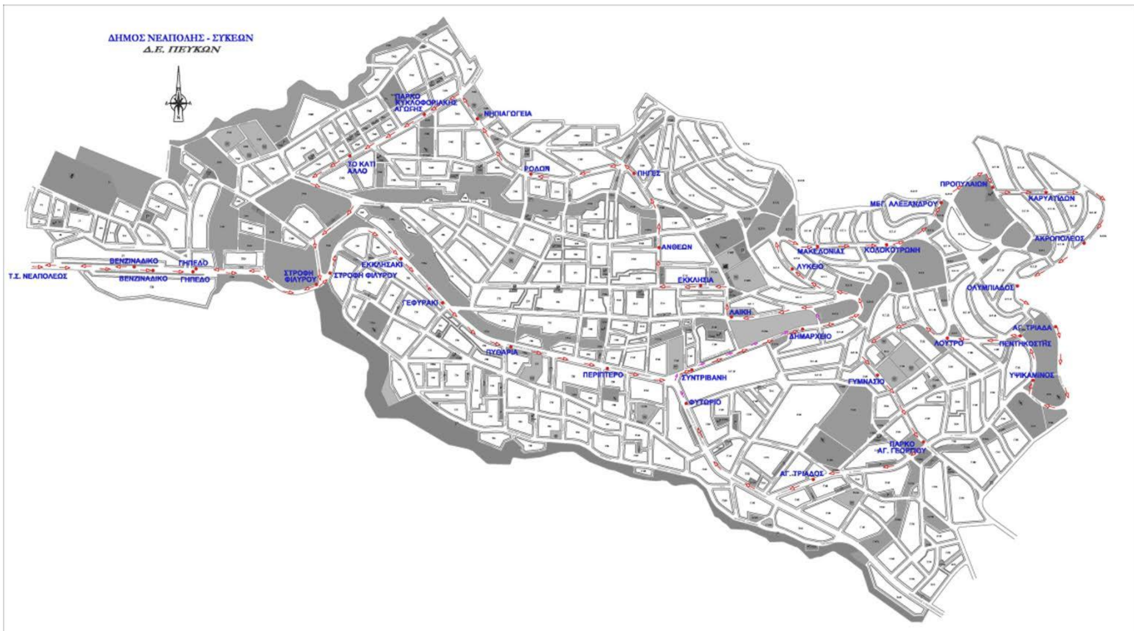
Η εύρεση, όμως, αυτών των βέλτιστων διαδρομών αντιμετωπίζει αρκετά προβλήματα. Τα προβλήματα βέλτιστης δρομολόγησης ανάγονται ουσιαστικά σε προβλήματα βελτιστοποίησης. Για την εύρεση της βέλτιστης λύσης χρησιμοποιείται η βιβλιοθήκη αλγορίθμων και τεχνικών του τομέα της Συνδυαστικής Βελτιστοποίησης (Combinatorial Optimization). Προβλήματα δρομολόγησης συναντώνται σήμερα σε πολλές περιοχές της έρευνας, όπως για παράδειγμα η Θεωρητική Πληροφορική ή η Επιχειρησιακή Έρευνα και ορισμένες εφαρμογές επάνω σε σύγχρονες τεχνολογίες είναι στα Δίκτυα Υπολογιστών (Computer networks), στο Παγκόσμιο Σύστημα Θεσιθεσίας (GPS), στη Διαχείριση Στόλου Οχημάτων (Fleet Management) και σε άλλες πολλές.

Η ιδέα της εύρεσης της βέλτιστης διαδρομής είναι κοινή για όλα τα προβλήματα και οι αρχές σχεδίασής της παρόμοιες. Ωστόσο, για τη μοντελοποίηση ενός προβλήματος, η βέλτιστη διαδρομή του ορίζεται διαφορετικά γιατί υπάρχουν διάφορα κριτήρια που πρέπει να ληφθούν υπόψη. Για παράδειγμα τέτοια κριτήρια μπορεί να αποτελούν διάφορες χρονικές καθυστερήσεις, χιλιομετρικές αποστάσεις, κυκλοφοριακή συμφόρηση και άλλα. Γι'αυτό το λόγο κάθε μοντελοποίηση προβλήματος πρέπει να είναι σωστά στοχευμένη ώστε να μπορεί να εξαχθεί ακριβής και εφικτή βέλτιστη λύση.

Στα πλαίσια της παρούσας μεταπτυχιακής διατριβής δίνεται ιδιαίτερη έμφαση στην εφαρμογή των αρχών και των κανόνων της δρομολόγησης με εφαρμογή σε ένα στόλο οχημάτων προκειμένου να επιλυθεί το πρόβλημα της εξυπηρέτησης πελατών από έναν αριθμό φορτηγών.

1.2.1 Η Δρομολόγηση Στόλου Οχημάτων

Η δρομολόγηση του στόλου οχημάτων αποτελεί εφαρμογή του γενικού προβλήματος της διανομής αγαθών και υπηρεσιών από στόλο οχημάτων για την εξυπηρέτηση πελατών. Στη συγκεκριμένη περίπτωση εφαρμογών, τα οχήματα ορισμένων επιχειρήσεων διακινούνται μεταξύ των πελατών της κάθε επιχείρησης, ξεκινώντας από έναν ή περισσότερους κεντρικούς σταθμούς μέσω ενός οδικού δικτύου. Στα οχήματα αυτά, αντιστοιχεί συγκεκριμένο πλήρωμα και εξαρτάται από τη χωρητικότητα πόσους πελάτες μπορούν να εξυπηρετήσουν.



Εικόνα 1.2.1: Παρουσίαση δρομολόγησης ενός στόλου φορτηγών στον δήμο Νεάπολης-Συκαιών

Με αυτή τη σκέψη, η έννοια της διακίνησης μπορεί να γενικευτεί. Αν γίνει η υπόθεση οτι και οι πελάτες μπορούν να μεταφέρουν αγαθά και υπηρεσίες μεταξύ τους, τότε ένα όχημα μπορεί να χρησιμοποιηθεί ταυτόχρονα ως μέσο διανομής και ως μέσο συλλογής αγαθών και υπηρεσιών. Με άλλα λόγια, είναι δυνατή η διανομή αγαθών από ΙΧ-μεταφορικό μέσο μιας επιχείρησης για λογαριασμό τρίτων. Ωστόσο, αυτό το φαινόμενο δεν συναντάται σχεδόν καθόλου σε απλές διανομές. Γι'αυτό και δίνεται ιδιαίτερη έμφαση στο πρόβλημα της δρομολόγησης τόσο σε επιχειρησιακό επίπεδο, όσο και σε επίπεδο στρατηγικού σχεδιασμού.

Το επιχειρησιακό επίπεδο αφορά ανάγκες που προκύπτουν σε δεδομένη χρονική στιγμή και πρέπει να αντιμετωπιστούν. Το επίπεδο του στρατηγικού σχεδιασμού αναλαμβάνει να προγραμματίσει το στόλο με βάση τα λειτουργικά προβλήματα που μπορεί να προκύψουν, όπως για παράδειγμα ελλείψεις προσωπικού, βαθμός ετοιμότητας οχημάτων, προκειμένου να ανταπεξέλθει με βέλτιστο τρόπο η επιχείρηση στα προβλήματα παραγγελιών που θα ζητηθούν.

Σε κάθε περίπτωση, η διακίνηση γίνεται μέσω ενός πραγματικού οδικού δικτύου στο οποίο η κυκλοφοριακή κίνηση μπορεί να επηρεάσει σημαντικά τις αποφάσεις που σχετίζονται με το πρόβλημα. Έτσι, η δρομολόγηση των οχημάτων από το τμήμα της επιχείρησης ανάγεται σε πρόβλημα μαθηματικού προγραμματισμού, στο οποίο αντικειμενική συνάρτηση είναι το κόστος και ως περιορισμοί εισέρχονται οι περιορισμοί του σχεδιασμού εξυπηρέτησης ενός πελάτη στο δίκτυο, λαμβάνοντας υπόψη όλες τις επιμέρους λεπτομέρειες που έχει αυτή η επιλογή εξυπηρέτησης.

Ο αντικειμενικός στόχος ενός προβλήματος δρομολόγησης είναι η ελαχιστοποίηση του λειτουργικού κόστους που προκύπτει από τη δρομολόγηση των οχημάτων και τον προγραμματισμό της εργασίας από το πλήρωμα. Η εξασφάλιση του αντικειμενικού στόχου έχει ως αποτέλεσμα:

- Τον προσδιορισμό του ακριβούς αριθμού των οχημάτων που έχουν αναλάβει τη μεταφορά και την καταγραφή της μεταφερόμενης ποσότητας.
- Τον προσδιορισμό των ελάχιστων οχημάτων που μπορούν να εξυπηρετήσουν το πλήθος των παραγγελιών.
- Τον προσδιορισμό των βέλτιστων διαδρομών των οχημάτων μέσα από το συγκεκριμένο οδικό δίκτυο.

Για να επιτευχθεί ο αντικειμενικός στόχος, χρειάζεται η δημιουργία ενός συστήματος βέλτισης δρομολόγησης που θα αναλαμβάνει τη δρομολόγηση του στόλου των οχημάτων των επιχειρήσεων με στόχο τη βέλτιστη οργάνωση από τη σκοπιά της ασφάλειας μιας μεταφοράς και του ελάχιστου κόστους. Επιχειρήσεις που ωφελούνται από ένα τέτοιο σύστημα είναι επιχειρήσεις του κυκλώματος των Logistics, εταιρείες διανομών, εταιρείες που ενδιαφέρονται για τον προγραμματισμό συναντήσεων πωλητών και εισπρακτόρων και γενικότερα επιχειρήσεων που δραστηριοποιούνται στις διανομές.

Επενδύοντας σήμερα σε υψηλή τεχνολογία βέλτιστης δρομολόγησης και αφουγκράζοντας τις απαιτήσεις που εμφανίζουν τα προβλήματα δρομολόγησης ανάλογα με την περίπτωση, οι διαχειριστές του στόλου έχουν πλέον τη δυνατότητα να χρησιμοποιούν συστήματα διαχείρισης του στόλου. Τα συστήματα αυτά μπορεί να είναι απλά συστήματα διαχείρισης που ελέγχουν τους πόρους, αξιοποιούν τα οχήματα στο μέγιστο δυνατό βαθμό, μετρούν την παραγωγικότητα και την αποδοτικότητα σε κάθε δρομολόγιο και απεικονίζουν γραφικά τις μεταβλητές του προβλήματος δρομολόγησης. Μπορούν, όμως, να είναι και συστήματα που βασίζονται σε καταγραφή των δεδομένων αυτόματα και σε πραγματικό χρόνο χρησιμοποιώντας εργαλεία τηλεματικής. Η τηλεματική προσφέρει μια ευρεία γκάμα εφαρμογών και δίνει τη δυνατότητα στο χρήστη να γνωρίζει την πραγματική θέση του οχήματος σε οδικό χάρτη. Από τη χρήση τέτοιων εργαλείων, μόνο κερδισμένη μπορεί να βγει η επιχείρηση μιας και έτσι θα εμφανιστούν αποτελέσματα σχετικά με την υψηλή απόδοση, γιατί θα εκτελούνται περισσότερα δρομολόγια με τον ίδιο αριθμό οχημάτων, που θα επηρεάσσουν τόσο την παραγωγικότητα και όσο και την πιο άμεση επιστροφή των αρχικών πόρων που επενδύθηκαν.

2. Προβλήματα Δρομολόγησης και Συνδυαστική Βελτιστοποίηση

Στο κεφάλαιο αυτό η μελέτη εμβαθύνει ερευνώντας τα βασικά χαρακτηριστικά των προβλημάτων δρομολόγησης από τη σκοπιά της Συνδυαστικής Βελτιστοποίησης (Combinatorial Optimization). Η Συνδυαστική Βελτιστοποίηση είναι ένα υποσύνολο της μαθηματικής βελτιστοποίησης. Σαν κλάδος ανήκει στον τομέα της Επιστήμης των Υπολογιστών. Αντικείμενό της είναι η εύρεση της βέλτιστης λύσης μέσα από ένα σύνολο πεπερασμένων λύσεων.

Τα περισσότερα θεμελιώδη προβλήματα της Συνδυαστικής Βελτιστοποίησης ανήκουν στην κλάση NP-Hard problems (Non-deterministic Polynomial Time Hard). Σε NP-Hard προβλήματα η επίλυση μέσα στο σύνολο των πεπερασμένων πιθανών διαδρομών είναι ανέφικτη. Έτσι χρησιμοποιούνται ειδικά διαμορφωμένοι αλγόριθμοι, οι οποίοι δεν έχουν ως στόχο τους να βρούν το απόλυτα βέλτιστο, αλλά το βέλτιστο μέσα σε λογικά πλαίσια δοσμένα από παραμέτρους που θα παράσχουν μια ικανοποιητική διαδρομή. Μερικές κατηγορίες αλγορίθμων που εφαρμόζονται με επιτυχία μέχρι στιγμής, μέσω της Συνδυαστικής Βελτιστοποίησης για την επίλυση προβλημάτων δρομολόγησης είναι:

- ευρεστικοί αλγόριθμοι
- αλγόριθμοι τοπικής αναζήτησης
- μετα-ευρεστικοί αλγόριθμοι

Τα προβλήματα δρομολόγησης ταξινομούνται με τρόπο ικανό ώστε ένα νέο πρόβλημα, ανάλογα με τα χαρακτηριστικά του, να μπορεί να κατηγοριοποιηθεί γρήγορα και σωστά και να αντιμετωπιστεί με τα κατάλληλα εργαλεία. Ένα πρόβλημα δρομολόγησης κατατάσσεται ανάλογα με διάφορα χαρακτηριστικά του, όπως για παράδειγμα αν η τοπολογία ενός δικτύου αλλάζει ανάλογα με το χρόνο, αν τα κόστη μεταβάλλονται, αν η διαδρομή είναι κλειστή ή ανοιχτή (μια διαδρομή είναι ανοιχτή ή κλειστή ανάλογα με το αν σχηματίζεται μονοπάτι ή κύκλος μεταξύ του αρχικού σημείου και του τελικού σημείου) και άλλα. Παρακάτω παραθέτονται μερικές κατηγορίες προβλημάτων δρομολόγησης που συναντώνται σε διάφορες περιοχές της επιστήμης και γίνεται μια σύντομη θεωρητική επισκόπηση και μια εισαγωγή στις αλγοριθμικές έννοιές τους.

2.1 Το πρόβλημα του Πλανόδιου Πωλητή (Travelling Salesman Problem - TSP)

2.1.1 Περιγραφή του προβλήματος του Πλανόδιου Πωλητή

Το πρόβλημα του περιοδεύοντος πωλητή (Travelling Salesman Problem - TSP) διατυπώθηκε για πρώτη φορά το 1930 από τον Ιρλανδό μαθηματικό William Rowan Hamilton και τον Βρετανό μαθηματικό Thomas Kirkman και αποτελεί βασικό αντικείμενο μελέτης και ανάλυσης για πολλά προβλήματα βελτιστοποίησης. Η απλότητα της διατύπωσης του προβλήματος είναι που το κάνει ένα από τα πιο εντόνως μελετημένα προβλήματα των Υπολογιστικών Μαθηματικών.

Με βάση την υπάρχουσα βιβλιογραφία, το πρόβλημα του περιοδεύοντος πωλητή ερευνά την πορεία που θα πρέπει να ακολουθήσει ένας πωλητής, ο οποίος επισκέπτεται διάφορες πόλεις με σκοπό να πουλήσει τα προϊόντα του. Γνωρίζοντας τις αποστάσεις μεταξύ των πόλεων ανα ζεύγη, θα πρέπει να επιλέξει τις κατάλληλες διαδρομές που θα ακολουθήσει ώστε ξεκινώντας από την πόλη του και επιστρέφοντας σε αυτή να έχει διανύσει τη συντομότερη απόσταση. Μέχρι σήμερα δεν έχει βρεθεί καμία γενική μέθοδος επίλυσής του.

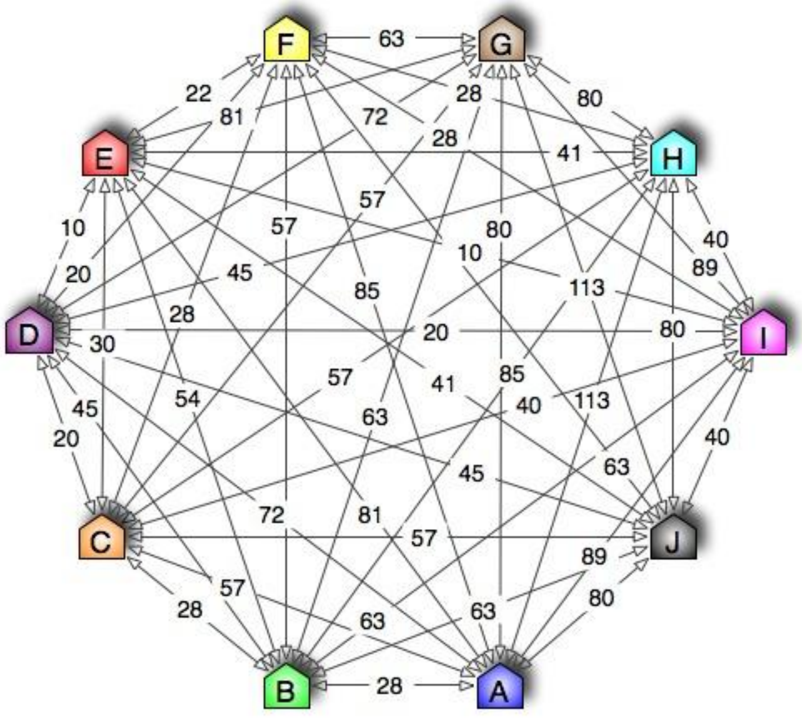
Τη δεκαετία του 1950 και του 1960, οι George Dantzig, Delbert Ray Fulkerson και Selmer Johnson στο συνέδριο RAND στην Santa Monica, εξέφρασαν το πρόβλημα ως ένα πρόβλημα ακέραιου γραμμικού προγραμματισμού, ενώ περίπου την ίδια εποχή, ο Hamilton δημιούργησε το Icosian Puzzle (το όνομα είναι από το αρχαίο ελληνικό είκοσι), που περιλαμβάνει την εύρεση ενός κύκλου από τις άκρες ενός δωδεκάεδρου, έτσι ώστε να διέρχεται από κάθε κόμβο μία μόνο φορά, να μην διέρχεται από κανέναν κόμβο δεύτερη φορά και το τελικό σημείο να είναι ίδιο με το αρχικό. Ο κύκλος αυτός ονομάστηκε κύκλος του Hamilton, ο οποίος είναι ένα μονοπάτι με την ιδιότητα αυτή. Οπότε ένα μονοπάτι του Hamilton είναι ένα μονοπάτι σε μη κατευθυνόμενο γράφημα το οποίο επισκέπτεται κάθε κόμβο ακριβώς μια φορά.

Παρότι το πρόβλημα του περιοδεύοντος πωλητή παρουσιάζει μεγάλη δυσκολία ως προς την υπολογισιμότητα και πολυπλοκότητά του, έχει ένα χαρακτήρα πρότυπο για πολλούς κλάδους των Μαθηματικών, των Υπολογιστών και της Επιχειρησιακής Έρευνας. Οι κύριες θεωρητικές συνιστώσες για τις σημερινές πιο επιτυχείς εφαρμογές σε δύσκολα συνδυαστικά προβλήματα βελτιστοποίησης είναι οι αλγόριθμοι Ευρεστικής Αναζήτησης και οι αλγόριθμοι Branch-and-Bound (B&B). Οι συνιστώσες αυτές αρχικά διατυπώθηκαν για το TSP και στη συνέχεια επεκτάθηκαν για την αναζήτηση λύσεων σε πιο περίπλοκα προβλήματα και με περισσότερα δεδομένα σαν είσοδο του προβλήματος.

Ο Richard Karp, απέδειξε το 1972, ότι το πρόβλημα με τον κύκλο του Hamilton ήταν ένα NP-Complete πρόβλημα (δηλαδή μπορεί να επαληθευτεί σε πολυωνυμικό χρόνο), από το οποίο προκύπτει για το πρόβλημα του περιοδεύοντος πωλητή ότι ανήκει στα NP-Hard προβλήματα λόγω της πολυπλοκότητας του. Αυτό σημαίνει ότι δεν υπάρχει αλγόριθμος που να επιλύει το πρόβλημα σε πολυώνυμο χρόνο, δηλαδή σε χρόνο που εξαρτάται πολυωνυμικά από το πλήθος των πόλεων που περιέχει σαν δεδομένο το πρόβλημα. Επομένως όταν αυξάνονται οι πόλεις, ο απαιτούμενος χρόνος μεγαλώνει εκθετικά, δηλαδή ένας υπολογιστής θα χρειαζόταν μερικά χρόνια υπολογισμών για να λύσει με ακρίβεια ένα πρόβλημα μερικών εκατοντάδων πόλεων σαν είσοδο.

2.1.2 Μαθηματική περιγραφή και αλγόριθμοι επίλυσης του προβλήματος TSP

Το πρόβλημα του περιοδεύοντος πωλητή (TSP) περιγράφεται μαθηματικά από έναν γράφο του οποίου οι κορυφές αναπαριστούν τις πόλεις και οι ακμές, που ενώνουν τις κορυφές, αναπαριστούν τις δυνατές μεταβάσεις. Στην ένωση μεταξύ των δυο πόλεων μεταξύ τους εμφανίζεται το κόστος της μετάβασης από τη μια στην άλλη. Όπως αναφέρθηκε και στην εισαγωγή του κειμένοιυ, μια διαδρομή σε ένα πρόβλημα TSP αναπαριστά γραφοθεωρητικά έναν κύκλο Hamilton και η αναζήτηση της βέλτιστης διαδρομής γίνεται στα πλαίσια της εξεύρεσης της διαδρομής του κύκλου Hamilton ελάχιστου μήκους. Συχνά το μοντέλο του προβλήματος ή του κύκλου Hamilton αναπαρίσταται από έναν γράφο.



Εικόνα 2.1.2: Ένα παράδειγμα TSP με τις ακμές να είναι τα χιλιόμετρα μετάβασης

Από τη φύση του και τη δομή του, το TSP δεν επιτρέπει ο πωλητής να επισκεφθεί πόλεις περισσότερες από μια φορά αλλά τόσο στη βιβλιογραφία όσο και σε αρκετές εφαρμογές δεν εισέρχεται αυτός ο περιορισμός. Το πρόβλημα μπορεί να είναι εφοδιασμένο με μια σχέση μεταξύ των κορυφών του γράφου. Ένα απλό παράδειγμα μιας τέτοιας περίπτωσης είναι να έχουμε σε ισχύ την τριγωνική ανισότητα για τις αποστάσεις μεταξύ τριών διαφορετικών κορυφών Α, Β και Γ. Πρακτικά αυτό μπορεί να σημαίνει ότι στο δίκτυο του γράφου δεν επιτρέπεται να υπάρχουν συντομεύσεις, δηλαδή ότι η απευθείας μετάβαση από την Α στη Β δεν πρέπει να έχει μεγαλύτερη απόσταση, από ότι να γίνει η μετάβαση διαμέσου της κορυφής Γ. Προφανώς όμως, σε πολλά προβλήματα δρομολόγησης, οι αποστάσεις δεν ικανοποιούν την τριγωνική ανισότητα. Για παράδειγμα, διαδρομές που καλύπτονται αεροπορικά μπορεί να είναι ταχύτερες, παρότι οι αποστάσεις είναι μεγαλύτερες.

Όπως έχει ήδη αναφερθεί, το πρόβλημα του περιοδεύοντος πωλητή ανήκει στην κατηγορία των NP - Hard προβλημάτων από άποψη υπολογισιμότητας και πολυπλοκότητας. Έχει αποδειχθεί πως αυτή η κατηγορία προβλημάτων είναι πλήρης ως πρός την κατηγορία προβλημάτων FP πολυπλοκότητας. Με τον όρο FP, υπονοούμε την κλάση όλων των function problems που επιλύονται από μια μηχανή Turing σε πολυωνυμικό χρόνο. Στη θεωρία της Υπολογιστικής Πολυπλοκότητας, ένα function problem είναι ένα πρόβλημα, για το οποίο μία και μόνο έξοδος αναμένεται για κάθε είσοδο. Αυτή η έξοδος είναι συνθετότερη από εκείνη ενός προβλήματος απόφασης (decision problem), δηλαδή δεν περιορίζεται σε δύο λογικές καταστάσεις τύπου ΝΑΙ ή ΟΧΙ. Στην περίπτωση που η πολυπλοκότητα είναι τύπου NP, υπάρχει μια μηχανή χρησμού (oracle machine) που οδηγεί το πρόβλημα στην επίλυσή του, μέσω λήψης κάποιας απόφασης.

Για να βρεθεί ακριβής λύση ενός προβλήματος TSP, μια πρακτική διαδικασία είναι να δοκιμαστούν όλες οι δυνατές διαδρομές (permutations) και να καταλήξουμε σε κάποια που θα ειναι αποδοτικότερη. Για να επιτευχθεί αυτό χρησιμοποιείται συνήθως η τεχνική της brute-force αναζήτησης. Η μέθοδος αυτή περιλαμβάνει την απαρίθμηση όλων των δυνατών υποψηφίων λύσεων και τον έλεγχο τους ως προς το αν ικανοποιούν τα ζητούμενα του προβλήματος. Ο τρόπος αυτός παρουσιάζει δυο βασικά πλεονεκτήματα. Είναι αρκετά απλός στην υλοποίησή του, και βρίσκει πάντα τη λύση, εφόσον αυτή υπάρχει. Το μοναδικό του μειονέκτημα έγκειται στην πολυπλοκότητά του. Είναι ανάλογη του αριθμού των πιθανών λύσεων, γεγονός που καθιστά την εφαρμογή του αλγορίθμου απαγορευτική για προβλήματα μεγάλου μεγέθους. Αυτό συμβαίνει επειδή η πολυπλοκότητα του είναι της τάξης του ($n!$), όπου n είναι ο αριθμός των πόλεων που πρέπει να επισκεφτεί ο πωλητής. Επομένως, λόγω της πολυπλοκότητας αυτής καθίσταται αναγκαίος ο σχεδιασμός αλγορίθμων για να προσεγγιστεί η βέλτιστη διαδρομή και να το πετυχαίνουν σε ικανοποιητικό χρόνο.

Οι αλγόριθμοι επεξεργάζονται το πρόβλημα από διαφορετική σκοπιά από οτι η brute - force αναζήτηση και η εύρεση ενός καλού ευρεστικού αλγόριθμου για το TSP αποτελεί δύσκολη διαδικασία. Έστω ένα TSP με n κόμβους. Κάθε διαδρομή που περνά από κάθε κόμβο ακριβώς μία φορά είναι μια υποψήφια λύση, και το κόστος της αποτελεί ένα άνω φράγμα (upper bound) για το ελάχιστο (βέλτιστο) κόστος. Τέτοιοι αλγόριθμοι ονομάζονται προσεγγιστικοί και παράγουν λύσεις χωρίς, όμως, καμία εγγύηση για το πόσο απέχουν οι λύσεις αυτές από τη βέλτιστη λύση του προβλήματος. Οι προσεγγιστικοί αλγόριθμοι, σύμφωνα με τη θεωρία της Υπολογιστικής Πολυπλοκότητας, μπορούν να εξασφαλίσουν λύση αν και μόνο αν για το πρόβλημα ισχύσει:

$$P = NP$$

Δυστυχώς, σε μελέτες της Υπολογιστικής Πολυπλοκότητας έχει αποδειχθεί οτι μάλλον πρόκειται για αδύνατη υπόθεση σε ένα πρόβλημα TSP.

Η δυσκολία στην επίλυση του προβλήματος του περιοδεύοντος πωλητή αποτελεί ταυτόχρονα και πρόκληση για την ανακάλυψη τεχνικών και μεθόδων, που θα οδηγήσουν στην παραπέρα συστηματική έρευνα και επίλυση προβλημάτων δρομολόγησης. Δεδομένης της απλότητας στην διατύπωση του, το TSP αποτελεί ίσως το πιο βασικό πρόβλημα δρομολόγησης, το οποίο συναντούμε έντονα στην πράξη. Μπορεί να ισχυριστεί κανείς, πως οι αλγόριθμοι και οι διάφορες τεχνικές που δίνουν έστω και προσεγγιστικά τη βέλτιστη διαδρομή για ένα πρόβλημα TSP, μπορούν να εφαρμοστούν με επιπλέον προεκτάσεις και για άλλα προβλήματα δρομολόγησης που συναντούμε στην περιοχή της Συνδυαστικής Βελτιστοποίησης (Combinatorial Optimization).

Κατά καιρούς προτάθηκαν διάφορες προσεγγίσεις, για την επίλυση του TSP. Σχεδόν κάθε νέα τεχνική που εφευρέθηκε για την αντιμετώπιση διαφόρων ζητημάτων βελτιστοποίησης, εφαρμόστηκε και στο TSP, στα πλαίσια της δοκιμής διαφόρων μεθόδων με τις οποίες καταπιάνονταν οι ερευνητές παντού στον κόσμο. Οπως αναφέρθηκε, τα πρώτα βήματα αφορούσαν τις κλασσικές τεχνικές, δηλαδή την ακριβή μέθοδο με κύριο εκφραστή την τεχνική του "brute force search" και τους προσεγγιστικούς αλγόριθμους. Οι τελευταίοι εντοπίζονται κυρίως μέσω των ευρεστικών τεχνικών, που περιλαμβάνουν γνωστές μεθόδους, όπως για παράδειγμα την "cutting-plane" μέθοδο. Ο τρόπος λειτουργίας της μεθόδου "cutting-planes" βασίζεται σε μια σειρά επαναλήψεων, που βελτιώνουν ένα εφικτό σύνολο (feasible set) ή μια αντικειμενική συνάρτηση (objective function) μέσω γραμμικών ανισοτήτων, που παρουσιάζονται δημιουργώντας ένα "κόψιμο". Η βέλτιστη λύση είναι αυτή με το μικρότερο "κόψιμο". Μια άλλη τεχνική που χρησιμοποιήθηκε ευρύτατα είναι αυτή του "branch and bound" αλγορίθμου. Ο αλγόριθμος αυτός περιλαμβάνει τη συστηματική απαρίθμηση όλων των υποψηφίων λύσεων. Εκεί όπου σημειώνονται αδόκιμες λύσεις, έχουμε τη μαζική απόρριψή τους χρησιμοποιώντας ένα άνω και ένα κάτω φράγμα της ποσότητας που προσπαθούμε να ελαχιστοποιήσουμε.

Μέθοδοι όπως αυτές του cutting-plane και του branch-and-bound. μπορούν να επιλύσουν μικρά προβλήματα βελτιστοποίησης, ενώ άλλες ευριστικές τεχνικές, όπως οι 2-opt, 3-opt, οι αλυσίδες Markov και η προσωμειωμένη ανόπτηση (simulated annealing) προτείνονται για μεγαλύτερα προβλήματα. Ωστόσο, πιο αποτελεσματικές τεχνικές επίλυσης προβλημάτων αποτελούν ορισμένοι αλγόριθμοι που βασίζονται στην άπληστη μέθοδο (greedy method), όπως αυτή του κοντινότερου «γείτονα» (nearest neighbor) και του συνδετικού δέντρου (spanning tree).

Μια από τις σημαντικότερες κατηγορίες αλγοριθμικών μεθόδων επίλυσης, αποτελούν οι Εξελικτικοί Αλγόριθμοι (Evolutionary Algorithms). Ένας αλγόριθμος τέτοιου είδους ακολουθεί τη λογική της επιβίωσης του ισχυρότερου (survival of the fittest), όπως αυτή παρουσιάζεται στον φυσικό κόσμο. Η φιλοσοφία στη χρήση ενός εξελικτικού αλγόριθμου, περιστρέφεται γύρω από έναν «πληθυσμό» λύσεων, με την ισχυρότερη λύση να επικρατεί και τις υπόλοιπες που είναι «αδύναμες» να απορρίπτονται. Δομικό στοιχείο των αλγορίθμων αυτής της κατηγορίας αποτελούν οι έννοιες του "ανασυνδυασμού (recombination)" και την "μεταλλαγής (mutation)" της λύσης. Ανασυνδυασμός συμβαίνει όταν γίνεται ανταλλαγή τμημάτων μεταξύ των λύσεων σε κάθε επανάληψη και οι συνδυασμένες λύσεις που προκύπτουν είναι ισχυρότερες από αυτές που τις δημιούργησαν. Μεταλλαγή της λύσης συμβαίνει οταν σε κάθε επανάληψη παρατηρείται να παραποιείται τμήμα του αλγορίθμου.

Μια από τις πιο πρόσφατες και πετυχημένες μεθόδους επίλυσης του προβλήματος του περιοδεύοντος πωλητή αποτελεί ο τομέας των Γενετικών Αλγορίθμων (Genetic Algorithms). Οι Γενετικοί Αλγόριθμοι, όντας υποκατηγορία των Εξελικτικών Αλγορίθμων, είναι εμπνευσμένοι από τη θεωρία της εξέλιξης των ειδών. Οι γενετικοί αλγόριθμοι εκτελούν μία αναζήτηση από το χώρο των υποψήφιων λύσεων, με στόχο την εύρεση αποδεκτών, σύμφωνα με κάποιο κριτήριο, λύσεων. Έχουν εφαρμοστεί επιτυχώς σε προβλήματα βελτιστοποίησης, όπως δρομολόγηση καλωδίων (wire routing), χρονοπρογραμματισμό (scheduling), παίγνια (game playing), προβλήματα εφοδιαστικής αλυσίδας (logistics) κ.τ.λ.

Η εισαγωγή των γενετικών αλγορίθμων έγινε το 1958 από τον Friedberg, ο οποίος επιχείρησε την αυτόματη παραγωγή σύνθετων προγραμμάτων FORTRAN με το συνδυασμό μικρότερων προγραμμάτων. Ωστόσο τα προγράμματα που προέκυπταν, τις περισσότερες φορές δεν ήταν εκτελέσιμα. Η διαδικασία αυτή είναι μία ειδική περίπτωση των γενετικών αλγορίθμων και αναφέρεται ως Γενετικός Προγραμματισμός (Genetic Programming). Ο Holland το 1975 έδωσε νέα ώθηση στο χώρο, χρησιμοποιώντας σειρές δυαδικών ψηφίων (bits) για να αναπαραστήσει λειτουργίες με τέτοιο τρόπο ώστε, κάθε συνδυασμός bit να είναι μία έγκυρη λειτουργία.

Αρχικά δημιουργείται με τυχαίο τρόπο ένα σύνολο Π από υποψήφιες λύσεις του προβλήματος, οι οποίες ως επί το πλείστον είναι μη αποδεκτές. Έστω Ν το πλήθος του συνόλου Π. Οι λύσεις αυτές βαθμολογούνται από μία συνάρτηση καταλληλότητας (fitness function). Η βαθμολόγησή τους συνίσταται στην αντιστοίχιση κάθε υποψήφιας λύσης σε έναν αριθμό, ο οποίος δηλώνει την εγγύτητα της υποψήφιας μη αποδεκτής λύσης ως προς κάποια αποδεκτή. Στη συνέχεια από τον αρχικό πληθυσμό σχηματίζονται Ν/2 ζευγάρια όχι απαραίτητα

μοναδικών γονέων, δίνοντας μεγαλύτερη προτεραιότητα στις πλέον κατάλληλες λύσεις. Κάθε ζευγάρι ζευγαρώνει (mates), δίνοντας δύο νέες λύσεις, τους απόγονους. Ο νέος πληθυσμός Π΄ αποτελείται από το σύνολο των απογόνων και συνήθως αποτελεί βελτίωση του προηγούμενου πληθυσμού. Σε παραλλαγές του αλγορίθμου, η ανανέωση του πληθυσμού μπορεί να μην είναι πλήρης αλλά ο νέος πληθυσμός Π΄ να αποτελείται από απογόνους και στοιχεία του αρχικού πληθυσμού Π. Η διαδικασία επαναλαμβάνεται για το νέο πληθυσμό Π΄, ενώ οι πιο συνηθισμένες συνθήκες τερματισμού είναι η εύρεση μίας τέλειας λύσης με βάση τη συνάρτηση καταλληλότητας ή η σύγκλιση όλων των λύσεων σε μία.
Η γενική μορφή ενός γενετικού αλγορίθμου είναι η εξής:

1. Δημιούργησε έναν αρχικό πληθυσμό Π, με Ν υποψήφιες λύσεις.
2. Υπολόγισε την καταλληλότητα κάθε λύσης.
3. Όσο δεν ισχύει κάποια συνθήκη τερματισμού:
    A. Επανέλαβε Ν/2 φορές τα ακόλουθα βήματα:
        i. Επέλεξε δύο λύσεις από τον πληθυσμό Π.
        ii. Συνδύασε τις δύο λύσεις για να βγάλεις δύο απογόνους.
        iii. Υπολόγισε την καταλληλότητα των δύο απογόνων.
    B. Δημιούργησε το νέο πληθυσμό Π΄ έχοντας υπόψη όλους τους νέους απογόνους που προέκυψαν από το βήμα 3Α και θέσε Π=Π΄.

Ο γενετικός αλγόριθμος εκτελεί μία αναζήτηση στο χώρο των υποψήφιων λύσεων με στόχο την εύρεση κάποιας λύσης που μεγιστοποιεί τη συνάρτηση καταλληλότητας. Η αναζήτηση αυτή είναι παράλληλη, καθώς σε κάθε υποψήφια λύση μπορεί να εκτελεστεί ξεχωριστή αναζήτηση. Η μέθοδος της αναζήτησης μπορεί να θεωρηθεί σαν αναρρίχηση λόφου (hill climbing), καθώς δεν γίνεται εξερεύνηση όλου του χώρου αναζήτησης αλλά γίνονται μικρές αλλαγές στις υποψήφιες λύσεις του πληθυσμού και επιλέγονται πάντα οι καλύτερες, βάσει της συνάρτησης καταλληλότητας. Η αναρρίχηση λόφου δεν κοιτάζει πιο πέρα από τους άμεσους γείτονες της τρέχουσας κατάστασης. Η αναζήτηση επικεντρώνεται στις περισσότερο κατάλληλες λύσεις, χωρίς όμως να αγνοούνται οι υπόλοιπες, καθώς υπάρχει πάντα ο κίνδυνος να παγιδευτεί η διαδικασία σε τοπικό μέγιστο(local maximum).

Ένας γενετικός αλγόριθμος για ένα συγκεκριμένο πρόβλημα περιλαμβάνει πέντε συστατικά:

- Ένα μηχανισμό για τη δημιουργία ενός αρχικού πληθυσμού πιθανών λύσεων (συνήθως δημιουργείται τυχαία).
- Έναν τρόπο αναπαράστασης των υποψήφιων λύσεων.
- Μία συνάρτηση καταλληλότητας για την αξιολόγηση των υποψήφιων λύσεων.
- Ένα μηχανισμό επιλογής γονέων.

- Ένα σύνολο γενετικών τελεστών για τη διαδικασία της αναπαραγωγής.

Ο μηχανισμός για τη δημιουργία ενός αρχικού πληθυσμού εξαρτάται από το δοσμένο πρόβλημα και είναι διαφορετικός.Η αναπαράσταση των υποψήφιων λύσεων γίνεται με μία συμβολοσειρά(string) ενός πεπερασμένου αλφάβητου. Συνήθως χρησιμοποιείται το δυαδικό αλφάβητο, οπότε οι συμβολοσειρές ονομάζονται και δυαδικές συμβολοσειρές. Στην βιολογία η συμβολοσειρά αναφέρεται και ως χρωμόσωμα, ενώ τα επιμέρους τμήματά της, που κωδικοποιούν κάποιο χαρακτηριστικό, ονομάζονται γονίδια.

Η συνάρτηση καταλληλότητας αποτελεί το κριτήριο για την αξιολόγηση των χρωμοσωμάτων, δηλαδή των υποψήφιων λύσεων. Η αξιολόγηση αυτή χρησιμοποιείται από τη συνθήκη τερματισμού ή από τη διαδικασία της πιθανοκρατικής επιλογής τους για να συμπεριληφθούν (ή όχι) στον πληθυσμό της επόμενης γενιάς. Η συνάρτηση αυτή δέχεται ως είσοδο ένα χρωμόσωμα και επιστρέφει έναν αριθμό που υποδηλώνει το βαθμό καταλληλότητάς του. Συνήθως το πεδίο τιμών αυτής της συνάρτησης είναι συνήθως το διάστημα των πραγματικών αριθμών από το 0 μέχρι το 1. Συνήθως η τιμή 1 δηλώνει ότι το χρωμόσωμα είναι τέλειο ενώ οι τιμές κάτω από το 1 δείχνουν το πόσο κοντά σε μία αποδεκτή λύση βρίσκεται.

Η διαδικασία επιλογής χρωμοσωμάτων-γονέων έχει να κάνει με την επιλογή ορισμένων χρωμοσωμάτων προς αναπαραγωγή με βάση την υψηλή τιμή που θα έχουν αυτά στη συνάρτηση καταλληλότητας, ενώ ενδέχεται να επιλεγούν προς αναπαραγωγή περισσότερες από μία φορές, ενώ αυτά που έχουν χαμηλή τιμή ενδέχεται να μην επιλεγούν καθόλου.

Η αναπαραγωγή είναι η διαδικασία δημιουργίας απογόνων. Σε αυτή εμπλέκονται ένα σύνολο από τελεστές με τους πιο χαρακτηριστικούς από αυτούς να είναι η μετάλλαξη (mutation) και η διασταύρωση (crossover). Ο τελεστής διασταύρωσης παράγει 2 απογόνους από 2 γονείς, αντιγράφοντας επιλεγμένα bit από κάθε γονέα με τέτοιο τρόπο ώστε το i-οστό bit του απογόνου να είναι το i-οστό bit ενός εκ των γονέων του. Το ποιος γονέας θα συνεισφέρει το κάθε bit αποφασίζεται βάση ενός μηχανισμού που ονομάζεται μάσκα διασταύρωσης. Ο συντελεστής μετάλλαξης αλλοιώνει ένα χρωμόσωμα του νέου πληθυσμού, μεταβάλλοντας τη τιμή κάποιου bit.

2.1.3 Παραλλαγές του προβλήματος TSP

Μέχρι σήμερα στη βιβλιογραφία, το πρόβλημα του περιοδεύοντος πωλητή έχει αντιμετωπιστεί σε χιλιάδες περιπτώσεις και σε πάρα πολλούς τομείς της καθημερινότητας. Μέσω αυτών των πρακτικών εφαρμογών, λοιπόν, εμφανίζονται πάρα πολλές μελέτες με διάφορες υποκατηγορίες ή μεταλλάξεις του προβλήματος του περιοδεύοντος πωλητή. Σε αυτό το υποκεφάλαιο θα αναφερθούν μερικές παραλλαγές που προκύπτουν προς επίλυση αναθεωρώντας μερικές μεταβλητές εισόδου στο αρχικό πρόβλημα.

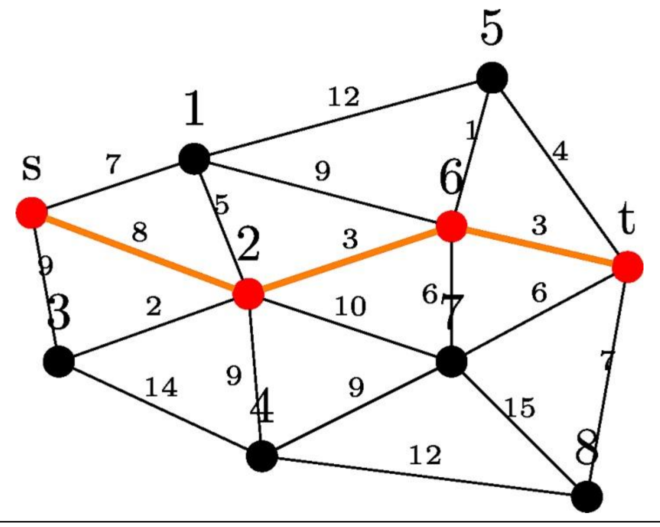
- MAX TSP: Αποτελεί την αντίθετη περίπτωση του TSP αφού σε αυτή την κατηγορία προβλημάτων σκοπός είναι η εύρεση μιας διαδρομής όπου το συνολικό κόστος των ακμών της να είναι το μέγιστο δυνατό. Μπορεί να αναχθεί στο αρχικό TSP αν αντικατασταθεί το κόστος κάθε πλευράς με τον αντίστροφο αριθμό. Αν υπάρχει ο περιορισμός στο πρόβλημα για μη-μηδενικά κόστη, μπορεί να προστεθεί μια σταθερά στα κόστη όλων των πλευρών, χωρίς να αλλάξει η βέλτιστη λύση του προβλήματος.
- Bottleneck TSP: Σε αυτήν την κατηγορία προβλημάτων σκοπός είναι να βρεθεί μια διαδρομή, της οποίας το μέγιστο κόστος των ακμών να είναι όσο το δυνατόν πιο μικρό. Το Bottleneck TSP μπορεί να αναχθεί στο κλασσικό TSP, αν θεωρήσουμε εκθετικώς μεγάλα κόστη στις ακμές του γράφου.
- Messenger Problem: Το πρόβλημα αυτό είναι γνωστό και σαν "the wandering salesman problem (το πρόβλημα του πλανόδιου πωλητή)". Μεταξύ δύο κόμβων u και v σε ένα γράφο G, αναζητούμε το μονοπάτι Hamilton ελάχιστου μήκους που οδηγεί διαδοχικά από την κορυφή u στη v. Το πρόβλημα μπορεί να αναχθεί στην περίπτωση του κλασσικού TSP εκλέγοντας μια τιμή κόστους ίση με -M για την ακμή (v,u), όπου M είναι ένας μεγάλος αριθμός. Αν δεν υπάρχουν προκαθορισμένοι κόμβοι, και θέλουμε να βρούμε το βέλτιστο μονοπάτι Hamilton επί του G μπορούμε πάλι να κάνουμε μια αναγωγή σε ένα ισοδύναμο TSP. Αυτό επιτυγχάνεται τροποποιώντας τον αρχικό γράφο G. Προσθέτουμε ένα νέο κόμβο, τον οποίο και ενώνουμε με όλους εκείνους τους κόμβους στους οποίους προσπίπτουν ακμές κόστους -M (αν ο γράφος είναι κατευθυνόμένος, κατασκευάζουμε δύο τόξα για κάθε ζεύγος κόμβων, ένα στην ευθεία και ένα στην αντίθετη κατεύθυνση). Από τη βέλτιστη διαδρομή για το TSP στον τροποποιημένο γράφο μπορεί να εξαχθεί και η βέλτιστη λύση για το αρχικό πρόβλημα.
- Clustered TSP: Στην περίπτωση αυτή το σύνολο των κόμβων του γράφου G διαμοιράζεται στα (υπο)σύνολα $V_1,V_2,...,V_k$ (ξένα μεταξύ τους). Το clustered TSP συνίσταται στην εύρεση της συμφερότερης διαδρομής στον G με τον περιορισμό, όμως, ότι ο πωλητής πρέπει να επισκεφθεί διαδοχικά κόμβους που ανήκουν στο ίδιο υποσύνολο. Το πρόβλημα μπορεί να αναχθεί στο κλασσικό TSP προσθέτοντας ένα μεγάλο κόστος M στα κόστη εκείνων των ακμών που προσπίπτουν σε κόμβους του ίδιου υποσυνόλου.
- m-salesmen TSP: Θεωρείται ότι m είναι το πλήθος των πωλητών που βρίσκονται στον κόμβο 1 ενός γράφου G. Ξεκινώντας από τον κόμβο 1, κάθε πωλητής επισκέπτεται ένα υποσύνολο X_i κορυφών του G ακριβώς μία φορά και επιστρέφει στον κόμβο 1. Θέλουμε να βρούμε μια διαμέριση X_1, X_2, ..., X_m του συνόλου V- {1}, όπου V είναι το σύνολο των κόμβων του γράφου, και τη διαδρομή καθενός από τους m πωλητές έτσι ώστε
  (i) $|X_i|>1$ για κάθε i,
  (ii) $U^m{}_{i=1} X_i = V - \{1\}$,
  (iii) $X_j \cap X_j = \emptyset$ για $i \neq j$ και

(iv) η συνολική διαδρομή που ακολουθούν οι m πωλητές είναι η ελάχιστη.
Αν ο γράφος G δεν είναι κατευθυνόμενος. το m-salesmen TSP μπορεί να αναχθεί σε κλασσικό TSP πάνω στο γράφο G' = (V', E') με m-I επιπρόσθετους κόμβους. Η βέλτιστη λύση για το TSP στον γράφο G΄ δίνει και τη βέλτιστη λύση για το m-salesmen TSP. Η περίπτωση όπου ο G είναι ένας κατευθυνόμενος γράφος μπορεί να αντιμετωπιστεί με αρκετά απλές τροποποιήσεις στη μέθοδο που περιγράφηκε παραπάνω.

2.2 Το πρόβλημα του Συντομότερου Μονοπατιού (Shortest Path Problem - SPP)

Ένα τεράστιο κομμάτι των προβλημάτων συνδυαστικής βελτιστοποίησης αποτελούν τα προβλήματα συντομότερης διαδρομής. Η επίλυσή τους πάντα καλύπτει τεράστιες ανάγκες σε διάφορες περιπτώσεις τεχνικών εφαρμογών. Ο ορισμός του προβλήματος μας λέει οτι δίνεται ένα γράφημα $G \equiv (V,E)$, με V το σύνολο των κορυφών και E το σύνολο των ακμών. Οι κόμβοι είναι αριθμημένοι 1, 2, ..., N . Για κάθε ακμή $(i, j) \in E$ υπάρχει η δυνατότητα αντιστοίχισης μίας συνάρτησης βάρους w_{ij} η οποία έχει διάφορες ερμηνείες (μήκος, χρόνος, κόστος μεταφοράς) ανάλογα με το πρόβλημα υπο εξέταση. Το μήκος μιας διαδρομής στο γράφημα $G \equiv (V,E)$ ισούται με το άθροισμα των ακμών της αντίστοιχης διαδρομής. Η διαδρομή αυτή χαρακτηρίζεται ως η συντομότερη αν έχει το μικρότερο μήκος από όλες τις διαδρομές με τους ίδιους κόμβους πηγής και προορισμού. Το πρόβλημα του συντομότερου μονοπατιού αφορά στον υπολογισμό της βέλτιστης διαδρομής μεταξύ επιλεγμένων ζευγών κόμβων.

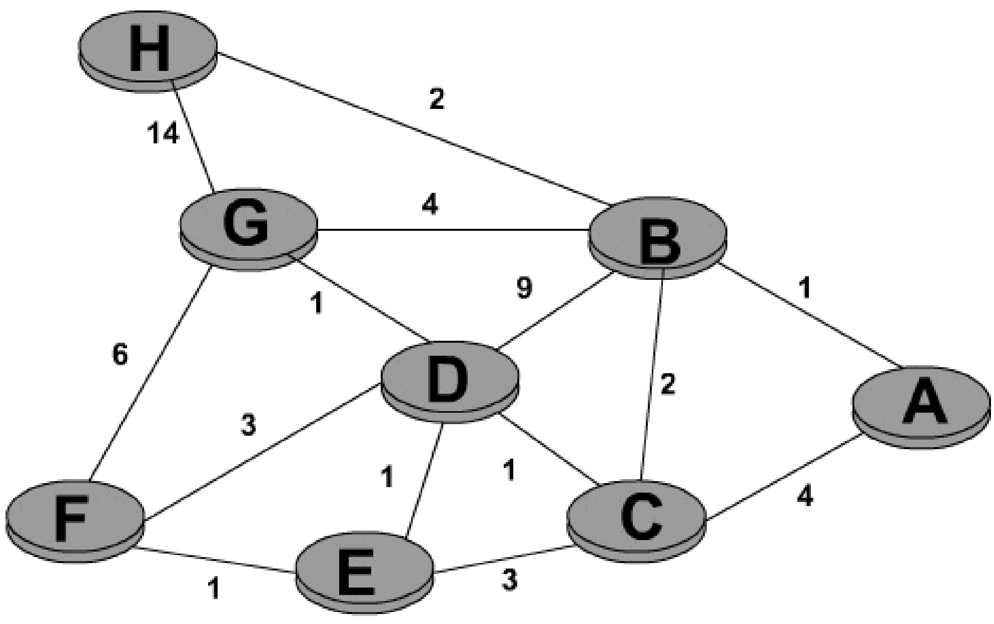
Το πεδίο εφαρμογής της συγκεκριμένης κατηγορίας προβλημάτων είναι ιδιαιτέρως ευρύ. Σε επόμενη υποενότητα θα προβάλλουμε ορισμένα αντιπροσωπευτικά παραδείγματα εφαρμογών του προβλήματος. Έπειτα, θα αναφέρουμε ορισμένους αλγορίθμους που επιλύουν το SPP. Οι περισσότεροι από τους αλγορίθμους αυτούς μπορούν να θεωρηθούν ως τεχνικές βελτιστοποίησης βασικού κόστους, εκτός της περίπτωσης του προβλήματος ελαχίστου κόστους ροής όπου εκεί εξετάζεται η έννοια του δυϊκού κόστους.



Εικόνα 2.2α: Δομή ενός προβλήματος συντομότερης διαδρομής. Με κόκκινο φαίνεται η βέλτιστη διαδρομή.

Το πρόβλημα συντομότερου μονοπατιού, όπως και το TSP, αποτελεί θεμελιώδες πρόβλημα βελτιστοποίησης. Και αυτό για δύο λόγους εξίσου σημαντικούς. Πρώτον, γιατί οι αλγόριθμοι που το επιλύουν χρησιμοποιούνται σε μια γενικότερη πληθώρα ζητημάτων βελτιστοποίησης. Και δεύτερον, γιατί το SSP δίνει τη δυνατότητα της παρουσίασης γραφημάτων στο περιβάλλον ενός απλού προβλήματος, γεγονός που παρέχει υποσχέσεις για πλούσια εφαρμογή των χαρακτηριστικών του για ακόμα συνθετότερα, συνδυαστικά προβλήματα της Θεωρίας Γραφημάτων.

Μια χαρακτηριστική εφαρμογή στην καθημερινή ζωή που απαιτεί την επίλυση προβλημάτων συντομότερης διαδρομής αποτελεί η δρομολόγηση πληροφοριών στα δίκτυα επικοινωνιών. Για να πραγματοποιηθεί η επικοινωνία μέσω δικτύων είτε για σύνδεση μέσω υπολογιστή στον Παγκόσμιο Ιστό (World Wide Web) είτε για επικοινωνία μεταξύ των ανθρώπων μέσω τηλεπικοινωνιών απαιτείται η χρήση ενός πεπερασμένου συνόλου δρομολογητών (κόμβων) που ανταλλάσουν μεταξύ τους πακέτα διαμέσω συνδέσεων καλωδίων (ακμές). Σε κάθε προσπάθεια σύνδεσης μεταξύ των δρομολογητών προκειμένου να μεταφερθούν τα πακέτα, υπολογίζεται αυτόματα ο συντομότερος δρόμος μεταξύ ενός ζεύγους δρομολογητών. Η δρομολόγηση των πακέτων γίνεται μέσω ειδικών αλγορίθμων που σε κάθε επανάληψή τους υπολογίζουν την συντομότερη διαδρομή που οδηγεί από τον κόμβο του πομπού στον κόμβο του δέκτη. Η συντομότερη διαδρομή είναι αυτή με τον μικρότερο αριθμό συνδέσεων.



Εικόνα 2.2β: Δομή προβλήματος συντομότερης διαδρομής εφαρμοσμένο σε δίκτυα υπολογιστών. Με το κυκλικό σχήμα παρουσιάζονται οι δρομολογητές.

Σύμφωνα με τη θεωρία των δικτύων, το μήκος μιας σύνδεσης εξαρτάται από την ικανότητα που έχει το μέσο να μεταδίδει πακέτα και από το φορτίο διακίνησής της. Η βέλτιστη διαδρομή θα πρέπει να περιλαμβάνει σχετικά λίγες και ασύμβατες συνδέσεις. Οι πολύπλοκοι αλγόριθμοι πορείας επιτρέπουν συνεχείς μεταβολές στο μήκος κάθε σύνδεσης, προκειμένου να υπάρχει συνεχώς αποτελεσματικότερη συμφόρηση στο δίκτυο. Υπό αυτό το πρίσμα, μία συντομότερη διαδρομή μπορεί να προσαρμόζεται σε κάποια προσωρινή υπερφόρτωση και να ανακατευθύνει την κίνηση των πακέτων μακριά από σημεία συμφόρησης. Γι'αυτό και ο αλγόριθμος του συντομότερου μονοπατιού εφαρμόζεται συνεχώς σε αυτόν τον τομέα, γιατί επιλύει το πρόβλημα της συντομότερης διαδρομής με μεταβαλλόμενα μήκη κατά την πορεία μιας σύνδεσης.

Ένα από τα χαρακτηριστικά σχεδιασμού των αλγορίθμων που αντιμετωπίζουν το πρόβλημα του συντομότερου μονοπατιού όταν εφαρμόζεται σε δίκτυα είναι πως κάνει χρήση κατανεμημένης και ασύγχρονης επικοινωνίας. Πιο αναλυτικά, κάθε κόμβος του δικτύου επικοινωνίας "παρακολουθεί" την κυκλοφορία των άμεσα συνδεδεμένων κόμβων του ("γείτονες κόμβοι") και υπολογίζει τις συντομότερες διαδρομές προς τον πομπό και διαβιβάζει τα αποτελέσματα στους άλλους κόμβους, οι οποίοι προσαρμόζουν, με τη σειρά τους, τους δικούς τους υπολογισμούς. Η διαδικασία αυτή βασίζεται σε κανονικούς, τυπικούς, συνήθεις αλγόριθμους συντομότερης διαδρομής, αλλά πραγματοποιείται ασύγχρονα και με παλαιότερες πληροφορίες εξαιτίας της καθυστέρησης του χρόνου που απαιτείται για την επικοινωνία μεταξύ των κόμβων.

2.2.1 Αλγόριθμοι επίλυσης του προβλήματος του Συντομότερου Μονοπατιού

Κατά καιρούς έχουν προταθεί διάφοροι αλγόριθμοι για την επίλυση προβλημάτων συντομότερου μονοπατιού , με αρκετούς από αυτούς να είναι πολύ καινοτόμοι και σημαντικοί για τον τομέα τους. Επειδή είναι αδύνατον να εξεταστεί κάθε αλγόριθμος ξεχωριστά, θα γίνει μια προσπάθεια κατηγοριοποίησής τους με βάση τις βασικές μεθόδους που χρησιμοποιούνται μέχρι σήμερα.

Οι αλγόριθμοι επίλυσης και υπολογισμού των προβλημάτων συντομότερου μονοπατιού είναι πάρα πολλοί γιατί και το ίδιο το πρόβλημα εκφράζεται με διαφορετικό τρόπο και διαφορετικά χαρακτηριστικά κάθε φορά. Έτσι, ο κύριος διαχωρισμός που μπορεί να γίνει είναι μεταξύ του αριθμού των κόμβων που θα είναι πομποί και του αριθμού των κόμβων που θα είναι δέκτες. Έτσι δημιουργούνται δυο ρέυματα, το πρώτο περιλαμβάνει προβλήματα με μια αφετηρία και έναν προορισμό και το δεύτερο με πολλαπλές αφετηρίες και πολλαπλούς προορισμούς.

Η κατηγορία των προβλημάτων συντομότερης διαδρομής με μια αφετηρία και έναν προορισμό αποτελεί τη συνηθέστερη παρουσίαση προβλημάτων καθημερινά. Τα βασικά ρεύματα αλγορίθμων που χρησιμοποιούνται για την αντιμετώπιση τέτοιων προβλημάτων είναι με τη χρήση ενός αλγορίθμου που θα έχει κεντρικό στόχο τη χρησιμοποπίηση ετικετών για τη δρομολόγηση συγκεκριμένων πακέτων μεταξύ των κόμβων "γειτόνων". Κύριοι εκφραστές αυτών των αλγορίθμων είναι οι αλγόριθμοι του Dijkstra, του Bellman - Ford, του SLF & LLL, του Dial, η μέθοδος του δυαδικού σωρού και η μέθοδος του "κατωφλίου".

Στην κατηγορία των προβλημάτων συντομότερης διαδρομής με περισσότερους του ενός κόμβους - αφετηρίες και περισσότερους του ενός κόμβους - προορισμούς γίνεται χρήση διαφορετικών αλγορίθμων. Κύριοι εκφραστές αλγορίθμων αντιμετώπισης τέτοιων προβλημάτων αποτελούν οι αλγόριθμοι Floyd-Warsall και ο αλγόριθμος του Dantig. Ωστόσο προβλήματα τέτοιας κατηγορίας συνήθως απαιτούν διαφορετική αντιμετώπιση. Γι'αυτό το λόγο και γίνεται χρήση διαφορετικών αλγορίθμων, πιο τεχνικών ή πρακτικών, όπως για παράδειγμα ευρεστικούς αλγόριθμους ή αλγόριθμους νοημοσύνης σμήνους ή ακόμα και αλγορίθμους βασισμένους σε κάποιας μορφής μηχανική μάθηση.

Στα πλαίσια αυτής της θεωρητικής προσέγγισης του προβλήματος του συντομότερου μονοπατιού θα πραγματοποιηθεί μια σύντομη παρουσίαση του αλγορίθμου Dijkstra, του αλγορίθμου Bellman-Ford και του αλγορίθμου Floyd-Warsall.

2.2.2 Ο αλγόριθμος του Dijkstra

Ο αλγόριθμος του Dijkstra επινοήθηκε από τον Ολλανδό επιστήμονα E.Dijkstra το 1959. Είναι ένας αλγόριθμος αναζήτησης που επιλεύει το πρόβλημα του συντομότερου μονπατιού με είσοδο έναν κόμβο-αφετηρία και έξοδο έναν κόμβο-προορισμό. Ο αλγόριθμος επιλύει το πρόβλημα πάνω σε ένα γράφημα με αρνητικά βάρη επί των ακμών-αποστάσεων. Το αποτέλεσμα που παράγει είναι ένα δένδρο ελάχιστης απόστασης (shortest path tree) μεταξύ όλων των πιθανών διαδρομών που ξεκινούν από τον κόμβο-πομπό και καταλήγουν στον κόμβο-δέκτη. Είναι επαναληπτικός και σε κάθε επανάληψη, αφού έχουν γίνει γνωστές οι αποστάσεις μεταξύ των κόμβων, επιλέγει τον κόμβο που μικραίνει τη συνολική απόσταση. Ο αλγόριθμος του Dijkstra χρησιμοποιείται τακτικά σε ζητήματα δρομολόγησης με χαρακτηριστικότερο παράδειγμα τη χρήση σε δίκτυα υπολογιστών και τηλεπικοινωνιών.

Για να περιγραφεί ο αλγόριθμος χρησιμοποιείται ένα παράδειγμα. Έστω οτι πρέπει να βρεθεί το συντομότερο μονοπάτι μεταξύ δυο υπολογιστών μέσα σε ένα δίκτυο από δρομολογητές. Σύμφωνα με τη θεωρία του αλγορίθμου από το βιβλίο των Kurose-Ross "Computer Networking", η εύρεση του δρόμου από την αφετηρία στον προορισμό είναι η εξής: «σε κάθε επανάληψη θεωρούμε ένα σύνολο μη επιλεγμένων διασταυρώσεων μεταξύ δρομολογητών, οι οποίες συνδέονται απευθείας με μια ήδη επιλεγμένη διασταύρωση. Από το σύνολο αυτό βρίσκουμε την πλησιέστερη διασταύρωση προς τον προορισμό επιλέγοντας το δρόμο εκείνης της διασταύρωσης και εισάγουμε τον κόμβο στο δένδρο ελάχιστης διαδρομής. Σε κάθε βήμα σημειώνουμε μόνο μια διασταύρωση από δρομολογητές. Μόλις φτάσουμε στον προορισμό, ακολουθούμε τα βέλη προς την αντίστροφη κατεύθυνση. Θα υπάρχει μονο ένα μονοπάτι με την ολοκλήρωση της διαδικασίας, το οποίο θα είναι και το συντομότερο».

```
Initialization:
  N' = {u}
  for all nodes v
    if v is a neighbor of u
      then D(v) = c(u,v)
    else D(v) = ∞

Loop
  find w not in N' such that D(w) is a minimum
  add w to N'
  update D(v) for each neighbor v of w and not in N':
       D(v) = min( D(v), D(w) + c(w,v) )
  /* new cost to v is either old cost to v or known
   least path cost to w plus cost from w to v */
until N'= N
```

Εικόνα 2.2.2α: Κώδικας του αλγορίθμου Dijkstra πάνω στο πρόβλημα ελάχιστης διαδρομής για δίκτυα υπολογιστών.

Τα βήματα του αλγορίθμου είναι τα εξής:

1. Αναθέτουμε σε κάθε κόμβο μια τιμή απόστασης. Συγκεκριμένα, αναθέτουμε το μηδέν στον κόμβο-πηγή και το άπειρο σε κάθε άλλο κόμβο.
2. Σημειώνουμε όλους τους κόμβους εκτός του κόμβου-πηγή ως μη επισκεπόμενους. Το δε κόμβο-πηγή τον σημειώνουμε ως τρέχον κόμβο.
3. Για τον τρέχον κόμβο, ερευνούμε για τους μη επισκεπτόμενους γειτονικούς του κόμβους. Για τους κόμβους αυτούς υπολογίζουμε τη δοκιμαστική απόσταση (tentative distance) από τον κόμβο πηγής. Αν, για παράδειγμα, έχουμε τον τρέχοντα κόμβο Α με απόσταση 6 και μια ακμή κόστους 2 ενώνει τον κόμβο Α με τον κόμβο Β, η απόσταση του Β διαμέσου του Α θα είναι 6+2=8. Αν η απόσταση αυτή είναι μικρότερη από την απόσταση που καταγράφηκε σε προηγούμενο βήμα (άπειρο στην αρχή, μηδέν για τον κόμβο-πηγή), την αντικαθιστούμε.
4. Όταν ολοκληρωθεί η εξέταση των γειτονικών κόμβων του τρέχοντα κόμβου, σημειώνουμε τον κόμβο αυτό σαν επισκεπτόμενο (visited). Ένας

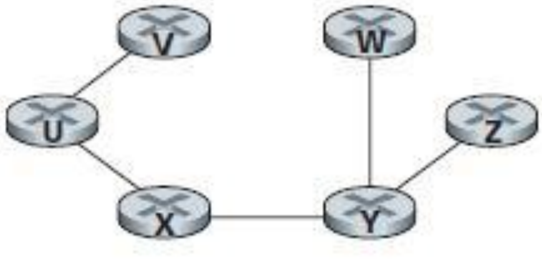
επισκεπτόμενος κόμβος δεν πρόκειται να εξεταστεί ξανά για την απόστασή του. Η απόστασή του είναι τελική και ελάχιστη δυνατή.

5. Άν όλοι οι κόμβοι είναι επισκεπτόμενοι, ο αλγόριθμος ολοκληρώθηκε. Διαφορετικά, θέτουμε εκείνον τον μη-επισκεπτόμενο κόμβο με τη μικρότερη απόσταση από την πηγή ως τρέχοντα κόμβο και συνεχίζουμε τη διαδικασία από το βήμα 3.

Στην παρακάτω εικόνα παρουσιάζεται ένα παράδειγμα από το βιβλίο των Kurose-Ross "Computer Networking" στο οποίο εμφανίζονται 6 δρομολογητές και ερευνάται το δένδρο ελάχιστης διαδρομής που ενώνει και τους 6 κόμβους, με αρχή τον κόμβο u. Πιο συγκεκριμένα, παρουσιάζεται ο πίνακας εύρεσης του δένδρου βασισμένος στα βήματα του αλγορίθμου παραπάνω και πιο κάτω παρουσιάζεται η εικόνα με το τελικό αποτέλεσμα του μετά την εφαρμογή του αλγορίθμου.

step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

Εικόνα 2.2.2β: Πίνακας εφαρμογή του αλγορίθμου Dijkstra σε πρόβλημα δικτύων υπολογιστών με 6 δρομολογητές.



Destination	Link
v	(u, v)
w	(u, x)
x	(u, x)
y	(u, x)
z	(u, x)

Εικόνα 2.2.2γ: Τελικό αποτέλεσμα του αλγορίθμου Dijkstra για τη σύνδεση 6 δρομολογητών.

Αυτό που μένει να ερευνηθεί αφού έχει βρεθεί το ελάχιστο μονοπάτι είναι το πόσος χρόνος και πόσοι πόροι υπολογιστικά χρειάστηκαν για να βρεθεί το τελικό αποτέλεσμα. Η πολυπλοκότητα του αλγορίθμου Dijkstra υπολογίζεται στο σκεπτικό οτι από τους συνολικούς κόμβους, αν αφαιρεθεί ο κόμβος-πηγή, πόσοι υπολογισμοί χρειάζονται για να βρεθεί η βέλτιστη λύση. Συγκεντρωτικά, ο συνολικός αριθμός των κόμβων που θα πρέπει να αναζητηθούν για να βρεθεί η βέλτιστη λύση είναι:

$$n(n+1)/2$$

και επομένως, η χειρότερη περίπτωση υπολογισμού κόμβων είναι ίση με:

$$n^2$$

Ωστόσο, σύμφωνα με τη βιβλιογραφία των αλγορίθμων, είναι σημαντικό να τονιστεί πως ο αλγόριθμος Dijkstra θα μπορούσε να έχει λιγότερη πολυπλοκότητα αν είχε λογαριθμικούς υπολογισμούς και οχι γραμμικούς.

Όπως έχει αναφερθεί στην εισαγωγή του προβλήματος του συντομότερου μονοπατιού, αλλάζοντας τα χαρακτηριστικά του προβλήματος αλλάζει και όλο το πλάνο επίλυσής του. Γι'αυτό και υπάρχουν πάρα πολλοί αλγόριθμοι επίλυσης τέτοιων προβλημάτων. Το ίδιο μπορεί να ειπωθεί οτι συμβαίνει και με τον αλγόριθμο του Dijkstra. Από τη στιγμή που αλλάζουν χαρακτηριστικά του προβλήματος προς επίλυση, επεκτείνεται και η χρησιμότητα του αλγορίθμου του Dijkstra. Για παράδειγμα, ορισμένες φορές είναι επιθυμητή η εύρεση λύσεων που είναι λιγότερο βέλτιστες από τη βέλτιστη δυνατή (less than mathematically optimal). Προκειμένου να επάρξει ένας κατάλογος λιγότερο βέλτιστων λύσεων, υπολογίζεται σε πρώτη φάση η βέλτιστη λύση του προβλήματος. Αφαιρείται μια ακμή που περιέχεται στο βέλτιστο μονοπάτι και υπολογίζεται εκ νέου η βέλτιστη λύση του υπογράφου που προκύπτει. Κατά παρόμοιο τρόπο, αφαιρείται μια ακμή του βέλτιστου μονοπατιού που βρήκαμε από τον υπογράφο και επαναλαμβάνεται η διαδικασία για τον επόμενο υπογράφο. Έτσι, προκύπτει ένα σύνολο βέλτιστων λύσεων και φυσικά το βέλτιστο μονοπάτι του αρχικού γράφου αποτελεί τη λύση υψηλότερης ιεραρχίας του καταλόγου των λύσεων που έχουν βρεθεί.

Τέλος, αξίζει να αναφερθεί οτι ο αλγόριθμος του Dijkstra χρησιμοποιεί την άπληστη (greedy) μέθοδο παρόμοια με αυτή του αλγορίθμου του Prim, που εφαρμόζεται όταν θέλουμε να βρούμε το ελάχιστο ζευγνύον δέντρο (Minimum Spanning Tree) σε ένα γράφημα. Το ελάχιστο ζευγνύον δέντρο σε ένα γράφημα είναι το συνεκτικό ακυκλικό υπογράφημα που διέρχεται από όλες τις κορυφές του αρχικού γραφήματος και το κόστος του είναι το ελάχιστο δυνατό. Σύμφωνα με τον αλγόριθμο του Prim, ταξινομούνται (sorting) τα κόστη όλων των ακμών του γραφήματος κατά αύξουσα σειρά και έπειτα προσθέτονται ακμές βάση αυτής της ταξινόμησης ξεκινώντας από το γράφημα, που περιέχει όλες τις κορυφές του γραφήματος χωρίς καμία όμως ακμή. Χρειάζεται προσοχή να μη σχηματιστούν κύκλοι (αφού το δέντρο είναι ακυκλικό γράφημα), απορρίπτοντας τυχόν ακμές της ταξινόμησης που παραβιάζουν αυτή την αρχή. Ο αλγόριθμος σταματά την προσθήκη ακμών μόλις έχουν τοποθετηθεί $(m = n-1)$ ακμές, αφού σύμφωνα με γνωστό θεώρημα, για κάθε δέντρο T(n,m) ισχύει $(n = m + 1)$. Το τελικό δέντρο από αυτή την άπληστη διαδικασία είναι το ελάχιστο δυνατό.

2.2.3 Ο αλγόριθμος των Bellman - Ford

Ο αλγόριθμος των Bellman - Ford υπολογίζει τα συνεχόμενα μονοπάτια μιας πηγής σε γραφήματα που φέρουν βάρη επί των ακμών. Για γράφους με μη αρνητικά βάρη, ο αλγόριθμος του Dijkstra είναι πολύ πιο γρήγορος. Έτσι ο αλγόριθμος των Bellman- Ford αντιμετωπίζει κυρίως γραφήματα με αρνητικά βάρη επί των ακμών.

Εάν κάποιος γράφος περιέχει έναν "αρνητικό κύκλο", δηλαδή έναν κύκλο του οποίου το άθροισμα των βαρών' είναι αρνητικό, τότε μπορούν' να κατασκευαστούν μονοπάτια αυθαίρετα μικρού συνολικού βάρους, οπότε δεν μπορεί να υπάρξει συντομότερη διαδρομή. Ο Αλγόριθμος των Bellman-Ford μπορεί να ανιχνεύσει κύκλους αρνητικού βάρους και δεν πρόκειται να κατασκευάσει μονοπατι στο οποίο επαναλαμβάνεται κάποια κορυφή.

Σύμφωνα με το βιβλίο των Kurose και Ross "Computer Networking" υπάρχει μια σχέση μεταξύ των στοιχείων κόστους των διαδρομών ελαχίστου κόστους. Έστω $d_x(y)$, το κόστος της διαδρομής ελαχίστου κόστους από τον x μέχρι τον y. Τα στοιχεία ελάχιστου κόστους σχετίζονται μεταξύ τους με την φημισμένη εξίσωση Bellman - Ford:

$$d_x(y) = min_v \{c(x,v) + d_v(v)\}$$

, όπου το min_v στην εξίσωση λαμβάνεται για τους γειτονικούς κόμβους του x. Η εξίσωση του Bellman -Ford είναι διαισθητική. Πράγματι, αφού γίνει η επιλογή του v μετά τον x, αν ληφθεί κατόπιν η διαδρομή ελαχίστου κόστους από τον v στον y, το κόστος διαδρομής θα είναι $c(x,v)+d_v(y)$. Η εξίσωση αυτή έχει μεγάλη πρακτική σημασία.

Η βασική ιδέα είναι η εξής: Κάθε κόμβος x αρχίζει με $D_x(y)$, μια εκτίμηση του κόστους της διαδρομής ελάχιστου κόστους από τον εαυτό του προς τον κόμβο y, για όλους τους κόμβους. Έστω οτι $D_x = \{D_x(y)$:y μέσα στο N$\}$ το διάνυσμα απόστασης του κόμβου x, που είναι το διάνυσμα των εκτιμήσεων του κόστους από τον x προς όλους τους άλλους κόμβους μέσα στο N. Με τον αλγόριθμο του Bellman - Ford κάθε κόμβος διατηρεί τις εξής πληροφορίες:

- Για κάθε γείτονα, αποθηκεύει το κόστος από τον x προς τον κάθε απευθείας συνδεδεμένο γείτονα.
- Το διάνυσμα απόστασης του x, δηλαδή ένα διάνυσμα που περιέχει την εκτίμηση του κόστους του x προς όλους τους προορισμούς μέσα στο N, όπου N είναι το πλήθος όλων των κόμβων.
- Τα διανύσματα απόστασης για κάθε έναν από τους γείτονές του.

Ο Αλγόριθμος των Bellman-Ford μοιάζει πολύ με τον αλγόριθμο του Dijkstra, αλλά αντί της άπληστης μεθόδου, απλά χαλαρώνει όλες τις ακμές, κάνοντάς το αυτό (n-1) φορές, όπου n=|V(G)|. Οι επαναλήψεις του αλγορίθμου επιτρέπουν την ανίχνευση των ελάχιστων αποστάσεων πάνω στο γράφημα, αφού υπό την απουσία μη αρνητικών κύκλων το συντομότερο μονοπάτι θα επισκέπτεται κάθε κόμβο του το πολύ μία φορά. Σε αντίθεση με την άπληστη μέθοδο, η οποία εξαρτάται σε συγκεκριμένες θεωρήσεις που πηγάζουν από πς θετικές τιμές των βαρών, η προσέγγιση των Bellman-Ford αποτελεί τη γενική περίπτωση αντιμετώπισης του προβλήματος συντομότερου μονοπατιού.

```
Initialization:
   for all destinations y in N:
      Dx(y) = c(x,y)   /* if y is not a neighbor then c(x,y) = ∞ */
   for each neighbor w
      Dw(y) = ? for all destinations y in N
   for each neighbor w
      send distance vector Dx = [Dx(y): y in N] to w

loop
   wait (until I see a link cost change to some neighbor w or
         until I receive a distance vector from some neighbor w)

   for each y in N:
      Dx(y) = minv{c(x,v) + Dv(y)}

   if Dx(y) changed for any destination y
      send distance vector Dx = [Dx(y): y in N] to all neighbors

forever
```

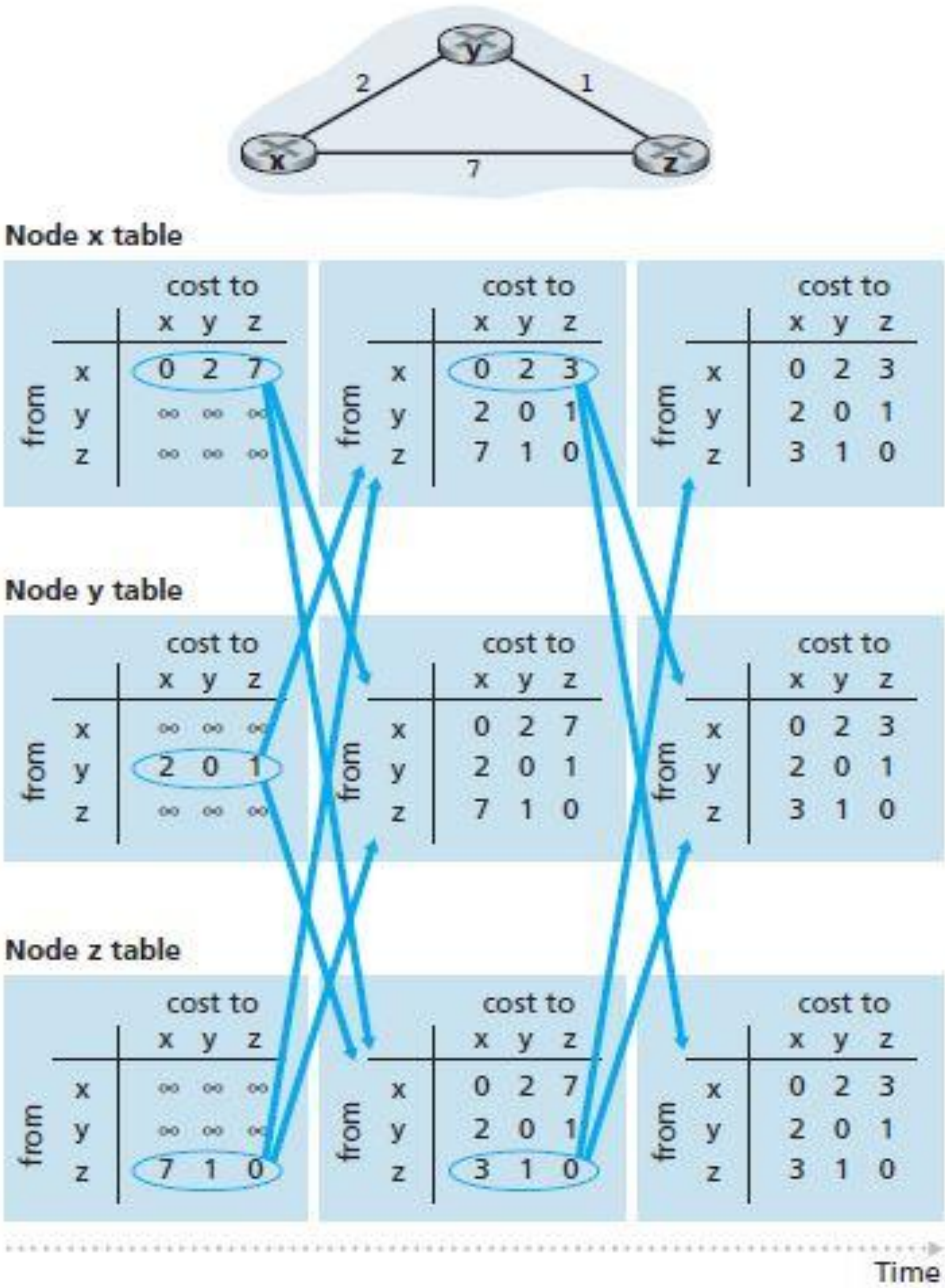
Εικόνα 2.2.3α: Κώδικας περιγραφής του αλγορίθμου Bellman-Ford πάνω στο πρόβλημα συντομότερης διαδρομής σε δίκτυα υπολογιστών.

Στην παρακάτω εικόνα, απεικονίζεται η λειτουργία του αλγορίθμου Bellman-Ford για το απλό δίκτυο τριών κόμβων που φαίνεται στην κορυφή της εικόνας. Η λειτουργία του αλγορίθμου επιδεικνύεται με έναν σύγχρονο τρόπο, όπου όλοι οι κόμβοι δέχονται ταυτόχρονα διανύσματα απόστασης από τους γείτονές τους. Όταν αυτά τα διανύσματα αλλάζουν υπολογίζουν τα νέα διανύσματα και πληροφορούν τους γείτονές τους οτι τα διανύσματα έχουν αλλάξει. Στην αριστερή στήλη της εικόνας εμφανίζονται τρείς αρχικοί πίνακες δρομολόγησης για κάθε έναν από τους τρείς κόμβους. Μέσα σε έναν πίνακα δρομολόγησης, ο πίνακας κάθε κόμβου περιλαμβάνει το δικό του διάνυσμα απόστασης και το διάνυσμα απόστασης κάθε ενός από τους απευθείας συνδεδεμένους γείτονές του. Μετά την αρχικοποίηση, κάθε κόμβος στέλνει το διάνυσμα απόστασής του σε κάθε έναν από τους δυο γείτονές του. Αυτό επιδεικνύεται και από τα βέλη στην εικόνα.

Έτσι, η δεύτερη στήλη εμφανίζει, για κάθε κόμβο, το νέο διάνυσμα απόστασης του κόμβου μαζί με τα διανύσματα απόστασης που μόλις έλαβε από τους γείτονές του. Αφού οι κόμβοι υπολογίσουν εκ νέου τα διανύσματα απόστασης, στέλνουν ξανά τα ενημερωμένα διανύσματα απόστασης στους γείτονές τους (αν υπήρξε κάποια αλλαγή). Αυτό επιδεικνύεται από τα βέλη από τη δεύτερη στήλη προς την τρίτη. Μετά τη λήψη των ενημερώσεων, οι κόμβοι υπολογίζουν εκ νέου τα διανύσματα απόστασής τους και ενημερώνουν τους δικούς τους πίνακες δρομολόγησης όπως φαίνεται στην τρίτη στήλη.

Η διεργασία λήψης ενημερωμένων διανυσμάτων από τους γείτονες, ο υπολογισμός των πινάκων δρομολόγησης εκ νέου και η πληροφόρηση των γειτόνων για το αλλαγμένο κόστος της διαδρομής ελαχίστου κόστους προς έναν προορισμό συνεχίζονται μέχρι να μην σταλεί κάνενα άλλο μήνυμα ενημέρωσης. Σ'αυτό το σημείο, εφόσον δεν στέλνονται μηνύματα ενημέρωσης, δεν θα γίνει κανένας υπολογισμός πίνακα δρομολόγησης και ο αλγόριθμος θα εισέλθει σε κατάσταση ηρεμίας. Αυτό σημαίνει οτι όλοι οι κόμβοι θα πέσουν σε κατάσταση αναμονής όπως φαίνεται και από τις γραμμές 10-11 από τον ψευδοκώδικα του αλγορίθμου (εικόνα 2.2.3α).

Ολοκληρώνοντας, σε αντίθεση με τον αλγόριθμο του Dijkstra, ο αλγόριθμος των Bellman - Ford μπορεί να χρησιμοποιηθεί και σε γραφήματα που φέρουν και αρνητικά κόστη επί των ακμών τους, αρκεί βέβαια να μην υπάρχουν κυκλικές διαδρομές αρνητικού συνολικού κόστους που να είναι προσπελάσιμες από τον κόμβο-πηγή. Η παρουσία τέτοιων κύκλων σημαίνει οτι δεν υπάρχει το βέλτιστο μονοπάτι που αναζητείται, αφού κάθε φορά που διατρέχονται οι ακμές του κύκλου, το κόστος μειώνεται.



Εικόνα 2.2.3β: Παράδειγμα χρήσης του αλγορίθμου Bellman - Ford σε δρομολόγηση πακέτων σε ένα δίκτυο υπολογιστών.

2.2.4 Ο αλγόριθμος Floyd - Warshall

Στην Επιστήμη των Υπολογιστών, ο αλγόριθμος των Floyd-Warshall (αλλιώς WFI αλγόριθμος ή Roy-Floyd) είναι μια αναλυτική μέθοδος εύρεσης των βέλτιστων μονοπατιών πάνω σε γράφους που φέρουν κόστη επί των ακμών τους (θετικά και/ή αρνητικά). Μια εκτέλεση του αλγορίθμου θα βρει τα συνολικά μήκη (total lengths) των βέλτιστων μονοπατιών για όλα τα δυνατά ζεύγη διακεκριμένων κορυφών του γράφου, χωρίς ωστόσο να μας επιστρέφει την ακολουθία των κόμβων που απαρτίζουν τα μονοπάτια αυτά. Ο αλγόριθμος βασίζεται στον Δυναμικό Προγραμματισμό. Δημοσιεύτηκε με την τρέχουσα μορφή του από τον Robert Floyd το1962. Ωστόσο, είναι παρόμοιος με αλγορίθμους που είχαν παρουσιαστεί από τον Bernard Roy το 1959 και από τον Stephen Warshall το 1962.

Ο Αλγόριθμος των Floyd-Warshall συγκρίνει όλα τα δυνατά μονοπάτια πάνω σε ένα γράφημα μεταξύ όλων των δυνατών ζευγών διακεκριμένων κορυφών. Μπορεί να λειτουργήσει με μόνο $\Theta(n^3)$ συγκρίσεις ($n=|V|$). Κάτι τέτοιο είναι αξιοσημείωτο δεδομένου ότι το μέγιστο πλήθος ακμών είναι ίσο με $n(n-1)/2$, και ο αλγόριθμος εξετάζει όλα τα δυνατά ζεύγη κορυφών. Ο αλγόριθμος σταδιακά βελτιώνει μια εκτίμηση του συντομοτέρου μονοπατιού μεταξύ δύο κορυφών, έως ότου η τελική εκτίμηση να είναι η βέλτιστη.

Έστω ένας γράφος G και V το σύνολο των κορυφών του, καθεμία αριθμούμενη από 1 έως N. Έστω η συνάρτηση shortestPath(i. j, k) η οποία επιστρέφει το συντομότερο μονοπάτι ανάμεσα στους κόμβους i και j χρησιμοποιώντας μόνο τις κορυφές 1,2,...,k σαν ενδιάμεσα σημεία κατά την εξέταση του μονοπατιού. Με δεδομένη αυτή τη συνάρτηση, σκοπός είναι η εύρεση του συντομοτέρου μονοπατιού μεταξύ των κόμβων i και j χρησιμοποιώντας μόνο τους κόμβους 1,2,...,k+1.

Υπάρχουν δύο υποψήφιες λύσεις: είτε το πραγματικό βέλτιστο μονοπάτι που χρησιμοποιεί ενδιάμεσους κόμβους από το σύνολλο {1,2,...,k}, είτε υπάρχει κάποιο μονοπάτι που οδηγεί από τον κόμβο i στον k+1 και έπειτα από τον k+1 στον j, το οποίο είναι καλύτερο. Γνωρίζουμε ότι το καλύτερο μονοπάτι που οδηγεί από τον κόμβο i στον j και χρησιμοποιεί μόνο τους κόμβους 1,2,...,k καθορίζεται από τη συνάρτηση shortestPath(i, j, k) και είναι φανερό πως αν υπήρχε ένα μονοπάτι από τον i στον k+1 και έπειτα στον j, τότε το μήκος του μονοπατιού αυτού θα ήταν η αληλουχία των συντομότερων μονοπατιών από τον i στον k-1 (χρησιμοποιώντας τις κορυφές {1,2,...,k}) και από τον k+1 στον j (επίσης χρησιμοποιώντας τις κορυφές {1,2,...,k}).

Επομένως, μπορεί να οριστεί η συνάρτηση shortestPath(i, j, k) αναδρομικά ως εξής:

ShortestPath(i, j, k)
= min{ShortestPath(i, j, k -1), ShortestPath(i, k, k-1)
+ Shortest Path (k, j, k-1)}

ShortestPath(i, j, 0) = edgeCost(i, j)

Η αναδρομική αυτή διαδικασία είναι το επίκεντρο του αλγορίθμου Floyd-Warshall. Ο αλγόριθμος δουλεύει υπολογίζοντας τη συνάρτηση shortestPath(i, j, k) για όλα τα ζεύγη (i, j) και k=1,2,... Η διαδικασία αυτή συνεχίζεται έως ότου k=n οπότε έχουμε βρει τα συντομότερα μονοπάτια για κάθε ζεύγος (i, j) χρησιμοποιώντας όλους τους ενδιάμεσους κόμβους.

Παρακάτω παρατίθεται ο αλγόριθμος των Floyd - Warshall:

```
let dist be a |V| × |V| array of minimum distances initialized to ∞ (infinity)
for each vertex v
   dist[v][v] ← 0
for each edge (u,v)
   dist[u][v] ← w(u,v)  // the weight of the edge (u,v)
for k from 1 to |V|
   for i from 1 to |V|
      for j from 1 to |V|
         if dist[i][j] > dist[i][k] + dist[k][j]
             dist[i][j] ← dist[i][k] + dist[k][j]
         end if
```

Εικόνα2.2.4α: Κώδικας με τον αλγόριθμο των Floyd - Warshall

Ο αλγόριθμος Floyd-Warshall υπολογίζει μόνο τα μήκη των βέλτιστων διαδρομών μεταξύ όλων των δυνατών ζευγών διακεκριμένων κόμβων. Με απλές τροποποιήσεις, είναι δυνατό να δημιουργηθεί μια μέθοδος για την κατασκευή της βέλτιστης διαδρομής μεταξύ δύο κόμβων. Η αποθήκευση όλων των μονοπατιών είναι μια διαδικασία αρκετά δαπανηρή για τη διαθέσιμη μνήμη του προγράμματος. Για κάθε κόμβο, αρκεί να αποθηκεύουμε τις πληροφορίες σχετικά με τον κόμβο που πρέπει να διατρέξουμε, εάν θέλουμε να καταλήξουμε σε κάποιον δεδομένο κόμβο του δικτύου.

Επομένως, οι πληροφορίες για την εύρεση όλων των βέλτιστων διαδρομών μπορούν να αποθηκευτούν σε μία NxN μήτρα next[i][j]. Το στοιχείο next[i][j] αντιπροσωπεύει τον κόμβο που πρέπει κανείς να διατρέξει εφόσον προτίθεται να κινηθεί πάνω στη συντομότερη διαδρομή από τον κόμβο i στον κόμβο j. Η εφαρμογή ενός τέτοιου προγραμματιστικού σχήματος γίνεται ασήμαντη, όταν μια νέα συντομότερη διαδρομή βρίσκεται μεταξύ δύο κόμβων. Τότε η μήτρα που περιέχει τα μονοπάτια ενημερώνεται. Η επόμενη μήτρα ενημερώνεται μαζί με τη μήτρα διαδρομής έτσι ώστε, κατά την ολοκλήρωση της ενημέρωσης, οι δύο μήτρες είναι πλήρεις και ακριβείς και όσες καταχωρήσεις έχουν την τιμή άπειρο στην μήτρα διαδρομής θα έχουν την τιμή null στη μήτρα next. Το μονοπάτι τότε από τον iστον j θα αποτελείται από τα υπομονοπάτια: (1) από τον i στον next[i][j] και (2) από τον next[i][j] στον j. Αυτά τα δύο συντομότερα μονοπάτια καθορίζονται αναδρομικά. Αυτός ο τροποποιημένος αλγόριθμος λειτουργεί με τον ίδιο χρόνο εκτέλεσης και με την ίδια πολυπλοκότητα χώρου μνήμης με τον μη τροποποιημένο αλγόριθμο.

Παρακάτω παρατίθεται ο κώδικας του τροποποιημένου αλγορίθμου:

```
procedure FloydWarshallWithPathReconstruction ()
  for k := 1 to n
    for i := 1 to n
      for j := 1 to n
        if path[i][k] + path[k][j] < path[i][j] then
          path[i][j] := path[i][k]+path[k][j];
          next[i][j] := k;

procedure GetPath (i,j)
  if path[i][j] equals infinity then
    return "no path";
  int intermediate := next[i][j];
  if intermediate equals 'null' then
    return " ";   /* there is an edge from i to j, with no vertices between */
  else
    return GetPath(i,intermediate) + intermediate + GetPath(intermediate,j);
```

Εικόνα 2.2.4β: Κώδικας τροποποιημένου αλγορίθμου Floyd-Warshall.

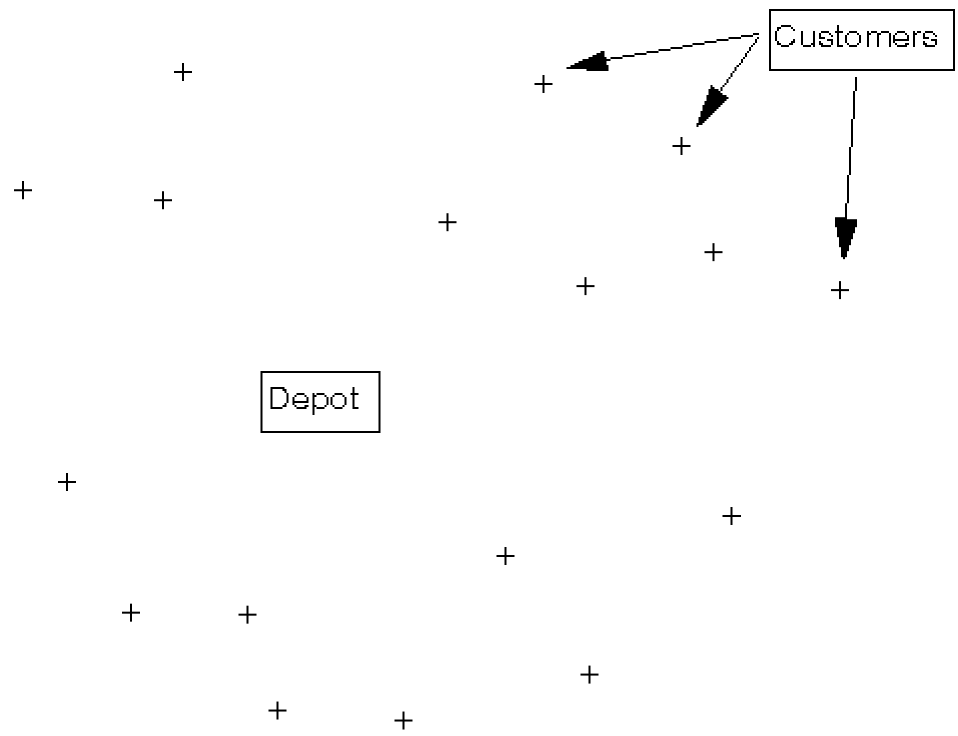
2.3 Το πρόβλημα δρομολόγησης οχημάτων (Vehicle Routing Problem - VRP)

Τα προβλήματα δρομολόγησης οχημάτων βρίσκονται στο επίκεντρο της διαχείρισης των διανομών. Είναι από τα πιο μελετημένα προβλήματα συνδυαστικής βελτιστοποίησης (combinatorial optimization), τόσο λόγω της πρακτικής τους σημασίας όσο και των σημαντικών δυσκολιών τους. Αποτελούν, συνήθως, ειδικές περιπτώσεις προβλημάτων Γραμμικού Προγραμματισμού (Linear Programming) στα οποία, όμως, οι μεταβλητές λαμβάνουν επιτρεπτές τιμές από ένα σύνολο ακέραιων αροθμών. Η βασική παρατήρηση που μπορεί να γίνει για τον Ακέραιο Προγραμματισμό (Integer Programming) έγκειται στο οτι πολλά από τα προβλήματά του παρουσιάζουν πολυπλοκότητα τύπου NP, δηλαδή δεν έχει βρεθεί αλγόριθμος για αυτά τα προβλήματα που να τα επιλύει σε πολυωνυμικό χρόνο.

Ένα VRP πραγματεύεται τον προσδιορισμό των βέλτιστων διαδρομών που θα ακολουθήσει ένας στόλος οχημάτων, εκκινώντας από τις εγκαταστάσεις για να εξυπηρετηθεί ένα σύνολο πελατών. Αντιμετωπίζεται καθημερινά από χιλιάδες επιχειρήσεις και οργανισμούς που ασχολούνται με την παράδοση και παραλαβή των αγαθών ή των ανθρώπων. Επειδή οι συνθήκες διαφέρουν από τη μια θέση στην άλλη, οι στόχοι και οι περιορισμοί που συναντώνται στην πράξη είναι εξαιρετικά μεταβλητοί.

Για παράδειγμα, η υπηρεσία μπορεί να περιλαμβάνει και παραδόσεις και παραλαβές, το φορτίο κατά μήκος κάθε διαδρομής δεν πρέπει να υπερβαίνει την καθορισμένη μεταφορική ικανότητα των οχημάτων, το συνολικό μήκος της κάθε διαδρομής δεν πρέπει να είναι μεγαλύτερο από ένα προκαθορισμένο όριο, η εξυπηρέτηση των πελατών πρέπει να γίνεται μέσα σε δεδομένα χρονικά περιθώρια, ο στόλος μπορεί να περιέχει ετερογενή οχήματα, μπορεί να υπάρχει προτεραιότητα στην εξυπηρέτηση των πελατών, οι απαιτήσεις των πελατών μπορεί να μην είναι πλήρως γνωστές εκ των προτέρων, η εξυπηρέτηση του πελάτη μπορεί να διαιρεθεί σε διαφορετικά οχήματα, καθώς και χαρακτηριστικά προβλήματα, που οι απαιτήσεις ή οι χρόνοι ταξιδίου μπορεί να διαφέρουν δυναμικά.

Η θεωρητική έρευνα και οι πρακτικές εφαρμογές στο πεδίο της δρομολόγησης οχημάτων ξεκίνησαν το 1959 με το «πρόβλημα αποστολής φορτηγών (truck dispatching problem)», το οποίο παρουσιάστηκε από τους Dantzig και Ramser. Ο ορισμός του προβλήματος ανέφερε: "Να βρεθεί η βέλτιστη δρομολόγηση ενός στόλου φορτηγών διανομής καυσίμου, μεταξύ ενός τερματικού στοθμού ανεφοδιασμού και ενός μεγάλου αριθμού σταθμών εξυπηρέτησης, οι οποίοι τροφοδοτούνται με καύσιμο από τον σταθμό ανεφοδιασμού". Χρησιμοποιώντας μια μέθοδο βασισμένη στις αρχές του Γραμμικού Προγραμματισμού (Linear Programming), οι Dantzig και Ramser κατέληξαν σε μια προσεγγιστικά βέλτιστη λύση του προβλήματος που περιελάμβανε τέσσερις διαδρομές για ένα πρόβλημα με δώδεκα σταθμούς εξυπηρέτησης.



Εικόνα 2.3: Χαρακτηριστικός σχεδιασμός για τη προβολή των πελατών και της αποθήκης για ένα πρόβλημα δρομολόγησης οχημάτων.

Μερικά χρόνια αργότερα και πιο συγκεκριμένα το 1964, οι Clarke και Wright πρότειναν έναν ευρεστικό αλγόριθμο βασισμένο στην άπληστη μέθοδο, ο οποίος πέτυχε καλύτερες προσεγγίσεις από αυτές των Dantzig και Ramser. Ακολουθώντας τις παραπάνω μελέτες προτάθηκαν εκατοντάδες νέοι αλγόριθμοι για την ακριβή και προσεγγιστική επίλυση διάφορων περιπτώσεων VRP. Σήμερα, υπάρχουν διαθέσιμα στην αγορά αρκετά πακέτα λογισμικού που χρησιμοποιούνται για την επίλυση πραγματικών προβλημάτων διαχείρισης στόλων οχημάτων.

Το VRP που ερευνάται φέρνει στο προσκήνιο την ιδέα της διανομής αγαθών ανάμεσα σε ένα σύνολο αποθηκών (depots) και ένα σύνολο γεωγραφικά διασκορπισμένων πελατών (customers). Μπορούμε να θεωρήσουμε τις θέσεις των πελατών σαν κόμβους εξυπηρέτησης πάνω σε ένα (οδικό) δίκτυο, στο οποίο σημειώνουμε προφανώς και όλες τις δυνατές μεταβάσεις (ακμές) που ενώνουν τους κόμβους εξυπηρέτησης μεταξύ τους αλλά και με τους σταθμούς ανεφοδιασμού. Τα μοντέλα και οι αλγόριθμοι που προτείνονται για τη λύση του VRP μπορούν να χρησιμοποιηθούν επαρκώς και στην περίπτωση γενικότερων ζητημάτων πάνω στα συστήματα μεταφοράς.

2.3.1 Περιγραφή του προβλήματος της δρομολόγησης οχημάτων

Όπως είδαμε στο κεφάλαιο 2.1, ένα θεμελιώδες πρόβλημα δρομολόγησης αποτελεί το πρόβλημα του πλανόδιου πωλητή (TSP). Υπενθυμίζεται σε αυτό το σημείο ότι, στο πρόβλημα του πλανόδιου πωλητη, ένας πλανόδιος πωλητής πρέπει να επισκεφτεί ένα σύνολο πόλεων και να επιστρέψει στην πόλη από την οποία ξεκίνησε, ελαχιστοποιώντας την απόσταση που συνολικά διανύει. Προφανώς, ο πωλητής πρέπει να επισκεφτεί κάθε πόλη μία και μόνο φορά, διαφορετικά η λύση δεν είναι βέλτιστη. Το VRP αποτελεί μια επέκταση του TSP, με την έννοια ότι στο VRP πρέπει να βρεθούν m στο πλήθος βέλτιστες διαδρομές οχημάτων, στις οποίες κάθε όχημα θα ξεκινάει από έναν κόμβο-αποθήκη (depot), θα διατρέχει ένα υποσύνολο κόμβων-πελατών (customers) σε καθορισμένη σειρά και θα επιστρέφει στον αρχικό κόμβο-αποθήκη.

Ο αντικειμενικός σκοπός του προβλήματος δρομολόγησης οχημάτων είναι η ελαχιστοποίηση του κόστους για κάθε μία από τις m διαδρομές (overall distribution costs). Στις πραγματικές εφαρμογές δρομολόγησης, όμως, προκύπτουν περιορισμοί που καθιστούν το πρόβλημα πολυπλοκότερο. Ανάλογα με τους περιορισμούς που μπορεί να εφαρμόζονται στο βασικό πρόβλημα της δρομολόγησης ενός στόλου οχημάτων κατατάσσονται τα διάφορα προβλήματα.

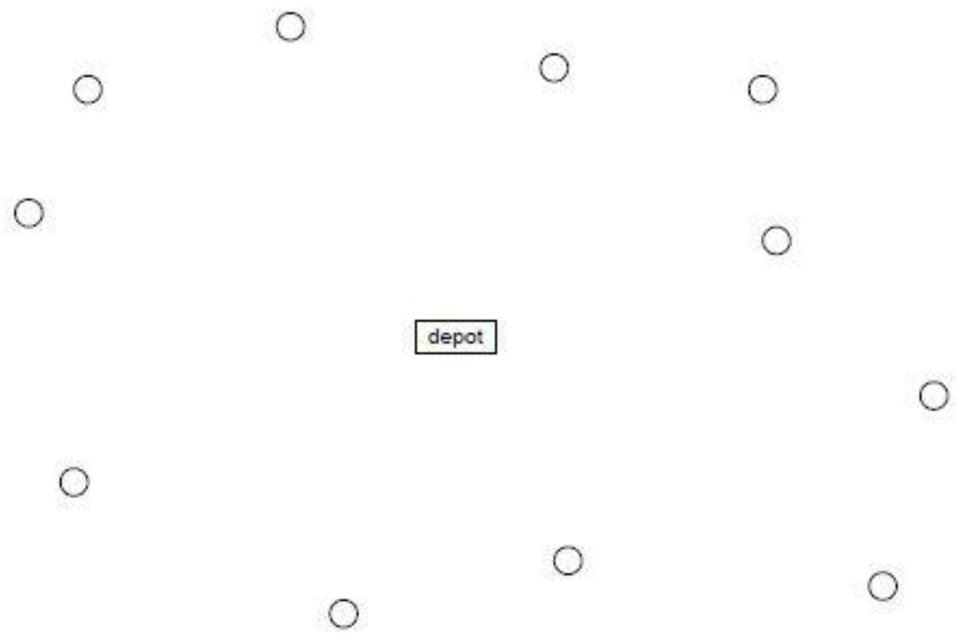
Χαρακτηριστικό παράδειγμα αποτελεί το πρόβλημα της δρομολόγησης ενός στόλου οχημάτων με περιορισμό στην μεταφερόμενη (προσφερόμενη) ποσότητα. Η ποσότητα αυτή δεν πρέπει σε καμία περίπτωση να ξεπεράσει τη χωρητικότητα των φορτηγών που εκτελούν τη μεταφορά. Τέτοια προβλήματα ονομάζονται "Capacitated VRPs". Μια άλλη κατηγορία έχει να κάνει με το είδος των φορτηγών που μεταφέρουν τα προϊόντα από την αποθήκη στους πελάτες. Ο περιορισμός εντοπίζεται στο είδος των φορτηγών. Με άλλα λόγια, τα φορτηγά δεν είναι ομογενή. Άλλη χωρητικότητα μπορεί να έχει το φορτηγό Α και άλλη χωρητικότητα να έχει το φορτηγό Β. Τέτοια προβλήματα ονομάζονται "Heterogeneous VRPs" και σπάνια εγείρονται προς επίλυση, μιας και οι επιχειρήσεις αναθέτουν την επιλογή φορτηγών με διαγωνισμό. Το αποτέλεσμα είναι να εξασφαλίζεται οτι τα φορτηγά θα έχουν κοινά χαρακτηριστικά. Σύμφωνα με τους Vigo, Cordeau και Laporte στη μελέτη τους "Vehicle Routing" αναφέρουν οτι το κλασσικό πρόβλημα είναι ένα πολύ γενικό πρόβλημα και πως χρειάζεται να λαμβάνονται υπόψη πάρα πολλοί περιορισμοί, συμπεριλαμβανομένης και της τεχνολογίας, όσον αφορά την σωστή επιλογή του προβλήματος που καλείται ο τομέας των Logistics μιας επιχείρησης να λύσει.

Στον παρακάτω συγκεντρωτικό πίνακα παρουσιάζονται μερικές γνωστές παραλλαγές του βασικού προβλήματος δρομολόγησης οχημάτων, με βάση τη βιβλιογραφία που έχει αναπτυχθεί, και ο περιορισμός στον οποίο στηρίζονται:

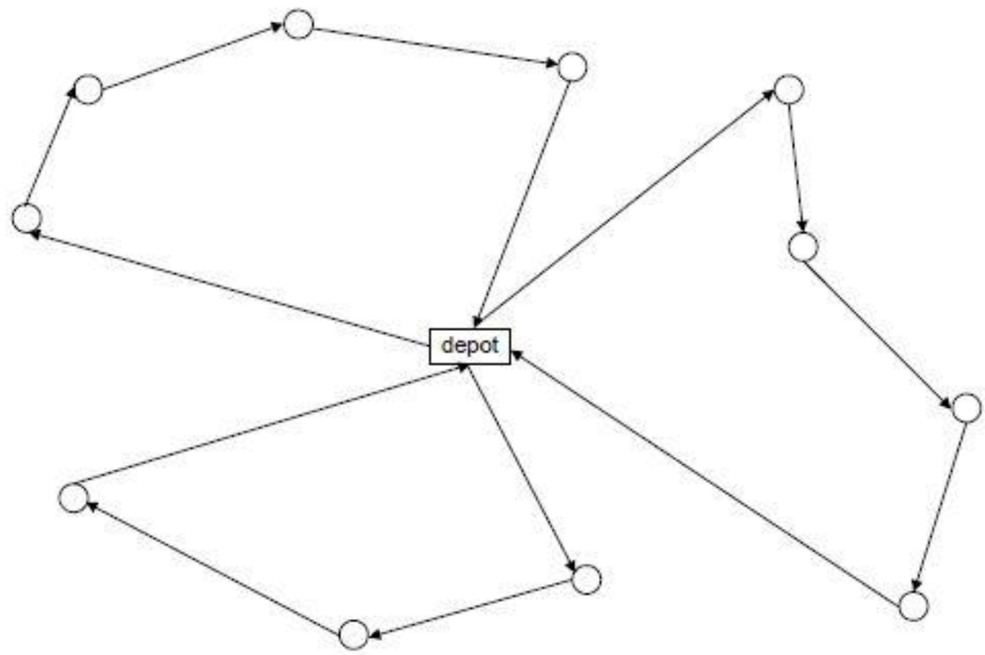
Κατηγορία Προβλήματος	Περιορισμός
Μέγεθος του στόλου	Ένα ή περισσότερα οχήματα για τις μεταφορές
Εγκαταστάσεις (Depot / Facilities)	Μια ή περισσότερες κεντρικές αποθήκες
Χωρητικότητα των οχημάτων (Capacity)	Ίδια ή διαφορετική για όλα τα οχήματα
Χρονικό παράθυρο συναλλαγών (Time Window)	Με ή χωρίς χρονικό παράθυρο για τις διανομές των προϊόντων στους πελάτες
Είδος των εργασιών	Διανομή μόνο από την αποθήκη προς τους πελάτες ή/και διανομή από τους πελάτες σε άλλους πελάτες εντός του δικτύου διανομών
Αντικειμενικοί στόχοι	Ελαχιστοποίηση συνολικού κόστους
	Ελαχιστοποίηση αριθμού οχημάτων
	Ελαχιστοποίηση χρόνου εξυπηρέτησης πελατών

Πίνακας 1: Παραλλαγές προβλημάτων δρομολόγησης οχημάτων και οι περιορισμοί που τα συνθέτουν

Οπως στο πρόβλημα του πλανόδιου πωλητή, έτσι και στο πρόβλημα δρομολόγησης στόλου οχημάτων μπορεί να χρησιμοποιηθεί ένα γράφημα για την αναπαράστασή του. Στην εικόνα παρακάτω (εικόνα 2.3.1α) παρουσιάζεται μια μορφή του προβλήματος δρομολόγησης με τους κόμβους εισόδου. Στο γράφημα που εμφανίζονται μόνο οι κόμβοι εισόδου γίνεται φανερό ποιοί είναι οι προορισμοί και σε ποιό σημείο μέσα σε ένα οδικό δίκτυο. Γνωρίζοντας το γράφημα με τους κόμβους εισόδου και επεξεργάζοντάς το κατάλληλα, προκύπτει ως αποτέλεσμα το γράφημα εξόδου. Σε ένα γράφημα εξόδου παρουσιάζονται, με βάση τους κόμβους εισόδου, οι οικονομικότερες διαδρομές που θα ακολουθήσουν τα οχήματα εξυπηρέτησης της ζήτησης. Στην παρακάτω εικόνα (εικόνα 2.3.1β) παρουσιάζεται η απάντηση στο γράφημα εισόδου της εικόνας 2.3.1α, επιλέγοντας το σύστημα να δρομολογήσει 3 οχήματα με τις βέλτιστες διαδρομές.



Εικόνα 2.3.1α: Διάγραμμα κόμβων εισόδου σε ένα σύστημα για επεξεργασία βέλτισης δρομολόγησης



Εικόνα 2.3.1β: Αποτέλεσμα εικόνας 2.3.1α εμφανίζοντας τις βέλτιστες διαδρομές και το πλήθος των οχημάτων εξυπηρέτησης.

Ένα πρόβλημα δρομολόγησης οχημάτων αποτελεί ένα συνδυαστικό πρόβλημα με πεδίο ορισμού το σύνολο των κορυφών και το σύνολο των ακμών ενός πλήρους γραφήματος, $G \equiv (V,E)$. Για το πρόβλημα αυτό ορίζονται τα κάτωθι:

- V = {v_0, v_1, ..., v_n}, ένα σύνολο κορυφών στο οποίο με V_0 ορίζεται η αποθήκη και $V' = V \backslash \{v_0\}$ το σύνολο των υπόλοιπων n πόλων.
- A = { (vi,vj)/ vi,vj ∈ V ; i≠j} , δηλαδή A είναι ένα υποσύνολο ακμών της μήτρας ακμών VxV
- C είναι ένας πίνακας από μη αρνητικά κόστη ή αποστάσεις μεταξύ των πελατών Vi kai Vj
- d είναι ένα διάνυσμα που περιέχει τις ποσότητες ζήτησης των πελατών.
- Ri είναι η διαδρομή του i-οστού οχήματος
- m είναι το πλήθος των οχημάτων. Οποιαδήποτε διαδρομή αντιστοιχεί σε ένα και μόνο όχημα.

Μια δυνατή λύση του VRP αποτελείται από μια διαμέριση $\{R_k\}_{k=1}^{m}$, μια μεταλλαγή (permutation) σ_i που να δηλώνει τη διαδοχή στην εξυπηρέτηση των πελατών πάνω στη διαδρομή i.

Το κόστος μιας συγκεκριμένης διαδρομής Ri = {v0, v1, ..., vm+1}, όπου $v_0 \equiv$ vm+1 (depot), δίνεται ως C(R_i) = $\sum_{i=0}^{m} c_{i,i+1} + \sum_{i=0}^{m} \delta i$. Η διαδρομή R_i είναι μέλος μιας υποψήφιας λύσης, αν το όχημα i σταματά ακριβώς μια φορά σε κάθε κόμβο εξυπηρέτησης και η συνολική διάρκεια της διαδρομής δεν ξεπερνά ένα προκαθορισμένο κατώφλι D, δηλαδή C(R_i)≤D . Τέλος, το κόστος της λύσης S του προβλήματος είναι ίσο με $f_{VRP}(S) = \sum_{i=1}^{m} C(Ri)$.

2.3.2 Παραλλαγές του προβλήματος της δρομολόγησης οχημάτων

Όπως έχει ήδη αναφερθεί, στη βιβλιογραφία του προβλήματος δρομολόγησης οχημάτων έχουν εμφανιστεί πάρα πολλές μελέτες. Αυτό συμβαίνει γιατί κάθε αλλαγή ενός χαρακτηριστικού του προβλήματος, δημιουργεί μια νέα παραλλαγή του αρχικού προβλήματος. Κάθε παραλλαγή αντιμετωπίζεται, όπως και το αρχικό πρόβλημα από έναν αλγόριθμο βελτιστοποίησης. Παρακάτω παρουσιάζονται ορισμένες παραλλαγές του προβλήματος δρομολόγησης οχημάτων και τα βασικά χαρακτηριστικά τους.

- <u>Capacitated VRP (CVRP)</u>: Το CVRP είναι μια περίπτωση προβλήματος δρομολόγησης στόλου οχημάτων με τα εξής χαρακτηριστικά: οι απαιτήσεις είναι ντετερμινιστικές, γνωστές εκ των προτέρων και δεν μπορούν να χωριστούν σε τμήματα, τα οχήματα είναι πανομοιότυπα και βασίζονται σε μια ενιαία κεντρική αποθήκη, επιβάλλεται μόνο ο περιορισμός για τη χωρητικότητα των οχημάτων, και ο στόχος είναι να ελαχιστοποιηθεί το συνολικό κόστος (δηλαδή, ο αριθμός των δρομολογίων ή / και του μήκους ή ο χρόνος ταξιδιού τους) που απαιτείται για να εξυπηρετηθούν όλοι οι πελάτες. Το CVRP μοιάζει με το βασικό VRP, το οποίο περιγράφηκε προηγουμένως, με τον επιπλέον περιορισμό ότι όλα τα οχήματα έχουν την ίδια χωρητικότητα και εξυπηρετούν την παράδοση ενός είδους εμπορεύματος. Τα CVRP έχουν μελετηθεί εκτενώς από τις αρχές της δεκαετίας του εξήντα και τα τελευταία χρόνια παρουσιάστηκαν πολλές νέες ευρεστικές και ακριβείς προσεγγίσεις. Τα μεγαλύτερα προβλήματα που μπορούν να επιλυθούν από τους πιο ακριβείς και αποτελεσματικούς αλγόριθμους που έχουν προταθεί μέχρι στιγμής έχουν όριο περίπου 50 πελάτες, ενώ μεγαλύτερες απαιτήσεις μπορούν να λυθούν μόνο σε ειδικές περιπτώσεις. Έτσι, περιπτώσεις με εκατοντάδες πελάτες, όπως αυτές που προκύπτουν σε πρακτικές εφαρμογές, μπορούν να αντιμετωπιστούν μόνο με ευρεστικές μεθόδους.

- Multiple Depots VRP (MDVRP): Μια επιχείρηση μπορεί να διαθέτει περισσότερες από μια αποθήκες, από τις οποίες εξυπηρετεί την πελατειακή ζήτηση που της παρουσιάζεται. Αν οι κόμβοι των πελατών διαμερίζονται σε υποσύνολα (clusters), καθένα από τα οποία περιέχουν ως επίκεντρό τους τις αποθήκες της εταιρείας, τότε το πρόβλημα της διανομής μπορεί να αναχθεί σε ένα σύνολο ανεξάρτητων μεταξύ τους VRPs. Αν όμως, οι πελάτες και οι αποθήκες διαπλέκονται μεταξύ τους, τότε πρέπει να επιλύσουμε ένα MDVRP. Ένα τέτοιο πρόβλημα απαιτεί την ανάθεση των πελατών στις διαθέσιμες αποθήκες. Σε κάθε αποθήκη υπάρχει ένας στόλος οχημάτων. Κάθε όχημα αναχωρεί από μια αποθήκη, εξυπηρετεί τους πελάτες που έχουν ανατεθεί στη συγκεκριμένη αποθήκη, και επιστρέφει στο σημείο από το οποίο ξεκίνησε. Συγκεντρωτικά, τα χαρακτηριστικά ενός MDVRP προβλήματος συνοψίζονται στα εξής:
    - Ο αντικειμενικός σκοπός του MDVRP είναι η ελαχιστοποίηση του αριθμού των οχημάτων του στόλου και του κόστους των χρόνων μετάβασης από κόμβο σε κόμβο. Η συνολική ζήτηση των εμπορευμάτων κάθε διαδρομής εξυπηρετείται από συγκεκριμένη αποθήκη, που γνωστοποιείται αμέσως μετά τη διαδικασία της ανάθεσης (assignment).
    - Ένα σύνολο διαδρομών στο MDVRP συνιστά μια υποψήφια λύση, αν κάθε διαδρομή ικανοποιεί τις προϋποθέσεις του βασικού προβλήματος δρομολόγησης οχημάτων και επιπλέον ξεκινά και καταλήγει στην ίδια αποθήκη.
    - Το σύνολο των κόμβων στο MDVRP γράφεται σαν $V=\{v_1,...,v_n\} \cup V_o$ όπου $V_0=\{v_{01},...,v_{0n}\}$ είναι το σύνολο των κόμβων των αποθηκών. Μια διαδρομή i ορίζεται ως $R_i=\{d,v_1.....vm,d\}$, όπου $d \in V_0$. Το κόστος της διαδρομής υπολογίζεται όπως και στην περίπτωση του VRP.

- Περιοδικά VRP (Periodic VRP): Αυτή η κατηγορία προβλημάτων δρομολόγησης αποτελεί ειδική περίπτωση που δε μελετάται πολύ συχνά. Στόχος της παραμένει ο κεντρικός στόχος του βασικού προβλήματος, δηλαδή η ελαχιστοποίηση του κόστους μεταφοράς, του χρόνου μεταφοράς και των οχημάτων που θα πραγματοποιήσουν την μεταφορά. Στην περίπτωση ενός PVRP, όμως, πρέπει να αποφασιστεί το σύνολο των διαδρομών εκείνων που θα επιτρέπουν στα οχήματα να μην επιστρέφουν την ίδια ημέρα στις εγκαταστάσεις της επιχείρησης, αλλά θα λαμβάνεται υπόψη οτι θα πραγματοποιείται ταξίδι που θα διαρκεί Μ ημέρες. Στο ταξίδι αυτό θα πρέπει κάθε πελάτης να έχει εξυπηρετηθεί τουλάχιστον μια φορά.

  Το PVRP μπορεί να αντιμετωπιστεί σαν ένα πρρόβλημα στο οποίο ζητείται ένα σύνολο διαδρομών για κάθε ημέρα έτσι ώστε οι συνολικοί περιορισμοί να ικανοποιούνται και να πετυχαίνουμε το ελάχιστο κόστος διαδρομών. Η επιχείρηση λαμβάνει ως δεδομένο την ημερήσια ζήτηση φορτίου για κάθε πελάτη που θα εξυπηρετήσουν τα φορτηγά. Η εξυπηρέτηση σε κάθε κόμβο πρέπει να γίνει εξ'ολοκλήρου σε μία επίσκεψη και από ένα μόνο όχημα. Αν η περίοδος Μ=1, το PVRP μεταπίπτει σε περίπτωση κλασσικού VRP. Κάθε κόμβος στο PVRP, πρέπει να προσπελαστεί k φορές, όπου $1 \leq k \leq M$. Στο κλασσικό PVRP, η ημερήσια ζήτηση για κάθε πελάτη είναι σταθερή.

  Στη θεωρία της συνδυαστικής βελτιστοποίησης αναφέρεται οτι ένα περιοδικό πρόβλημα δρομολόγησης οχημάτων μπορεί να μελετηθεί και από τη σκοπιά ενός προβλήματος βελτιστοποίησης πολλών επιπέδων. Στο πρώτο επίπεδο, θα αντιμετωπιστεί η εύρεση ενός συνόλου δυνατών συνδυασμών για την εξυπηρέτηση κάθε πελάτη. Στο δεύτερο επίπεδο, με δεδομένους τους συνδυασμούς, επιλέγετεαι ένας από αυτούς για κάθε πελάτη, προσέχοντας να ικανοποιούνται οι ημερήσιοι περιορισμοί. Έτσι, γίνεται μια διαλογή των πελατών που θα εξυπηρετούνται ανά μέρα για μια συγκεκριμένη χρονική περίοδο. Στο τρίτο και τελευταίο επίπεδο απομονώνεται το VRP κάθε ημέρας και επιλύεται με όλα τα εργαλεία που θα επιλυόταν ένα κλασσικό VRP πρόβλημα.

- Split Delivery VRP (SDVRP): Και αυτή η κατηγορία αποτελεί μια ειδική περίπτωση ενός προβλήματος δρομολόγησης ενός στόλου οχημάτων. Σε αυτή την κατηγορία προβλημάτων, ζητείται η ελαχιστοποίηση του κόστους, του χρόνου μεταφοράς και του αριθμού των οχημάτων που θα την χρησιμοποιήσουν, ωστόσο χαλαρώνει μια από τις βασικές υπόθεσεις του κλασσικού προβλήματος. Επιτρέπεται σε περισσότερα από ένα φορτηγά να επισκέπτονται τον ίδιο κόμβο για μεταφορά προϊόντων. Σημειώνεται οτι αυτή η κατηγορία προβλημάτων αντιμετωπίζεται μόνο όταν η εξυπηρέτηση των πελατών από περισσότερα από ένα οχήματα επιφέρει μείωση του συνολικού κόστους. Η εύρεση λύσης σε αυτή την κατηγορία προβλημάτων είναι μια αρκετά πολύπλοκη διαδικασία από την αντίστοιχη του κλασσικού VRP και στη βιβλιογραφία δεν υπάρχουν ακόμα αρκετές αναλυτικές μελέτες περιπτώσεων που να μπορούν να ληφθούν υπόψη. Χαρακτηριστικό του οτι ακόμα αντιμετωπίζονται τέτοια προβλήματα.

- VRP with Backhauls (VRPB): Αυτή η κατηγορία προβλημάτων αποτελεί προέκταση της κατηγορίας προβλημάτων δρομολόγησης οχημάτων με περιορισμό τη χωρητικότητα του κάθε φορτηγού. Προτού αναλυθούν τα βασικά χαρακτηριστικά της κατηγορίας αυτής, θεωρείται οτι κάθε πελάτης i χαρακτηρίζεται από δύο ανεξάρτητες μεταβλητές, την d_i και την p_i. Η d_i αναπαριστά την ζήτηση παραλαβής φορτίου από τον πελάτη και η p_i αναπαριστά την ποσότητα φορτίου που επιστρέφει στις εγκαταστάσεις της επιχείρησης. Για κάθε πελάτη i υπάρχει μια αποθήκη αποστολής φορτίου O_i και μια αποθήκη παραλαβής φορτίου D_i. Υποτίθεται ότι σε κάθε κόμβο - πελάτη, η παράδοση φορτίου από το όχημα συμβαίνει προτού ο πελάτης φορτώσει στο φορτηγό την ποσότητα του φορτίου που θα επιστραφεί.

  Η κατηγορία αυτή είναι διαφορετική στον τρόπο αντιμετώπισής της σε σύγκριση με το βασικό πρόβλημα και τις παραλλαγές του. Σύμφωνα με τη μελέτη των Laporte και Vigo, προκειμένου να μελετηθεί πλήρως αυτή η κατηγορία, πρέπει να διαμοιραστεί το σύνολο των πελατών σε δυο υποσύνολα. Το πρώτο υποσύνολο, έστω L, περιέχει έναν αριθμό από n πελάτες «εμπρόσθιας εξυπηρέτησης (linehaul customers)», καθένας από τους οποίους ζητάει να του παραδοθεί μια συγκεκριμένη ποσότητα. Το δεύτερο υποσύνολο, έστω B, περιέχει έναν αριθμό από m πελάτες «οπίσθειας εξυπηρέτησης (backhaul customers)», από τους οποίους κάποια φορτηγά θα πρέπει να τους επισκεφτούν για να παραλάβουν κάποια ποσότητα εισερχόμενου φορτίου. Για τα υποσύνολα των πελατών ισχύει οτι: $L = \{1, 2, \dots, n\}$ και $B = \{n+1, n+2, \dots, n+m\}$.

  Στόχος ενός τέτοιου προβλήματος είναι η εύρεση ενός συνόλου, K στο πλήθος, απλών κλειστών διαδρομών ελάχιστου κόστους για το οποίο θα ισχύουν οι εξής περιορισμοί:

  i. Κάθε κύκλος θα πρέπει να διέρχεται από τον κόμβο ανεφοδιασμού (depot) μία φορά,
  ii. Κάθε κόμβος - πελάτη θα πρέπει να ανήκει αποκλειστικά και μόνο σε μια κλειστή διαδρομή,
  iii. Η συνολική ζήτηση τόσο των bakhaul όσο και των linehaul πελατών, στα πλαίσια μιας κλειστής διαδρομής, δεν θα ξεπερνάει τη χωρητικότητα C του οχήματος, και
  iv. Σε κάθε κλειστή διαδρομή, οι linehaul πελάτες έχουν προτεραιότητα έναντι των backhaul πελατών.

Σύμφωνα με τη μελέτη των Toth και Vigo "The Vehicle Routing Problem", σε μια τέτοια κατηγορία προβλημάτων είναι σημαντικό να ερευνηθεί πρώτα σε βάθος κάθε διαδρομή αν είναι ικανή να εισέλθει στο σύνολο και να κοστολογηθεί. Για να θεωρηθεί μια διαδρομή δυνατή, θα πρέπει να ικανοποιήσει τρία κριτήρια που εξαρτώνται άμεσα με τη χωρητικότητα των οχημάτων που θα εκτελέσουν τις μεταφορές.

Το πρώτο κριτήριο ελέγχει τη συνολική ποσότητα των εμπορευμάτων να μην ξεπερνάει τη χωρητικότητα C του οχήματος. Για μια διαδρομή θα πρέπει να ισχύει οτι: C_d $(v_k) \leq$ C και C_d $(v_{k+1}) >$ C. Με C_d (v_k) περιγράφεται η συνολική ποσότητα εμπορευμάτων που παραδόθηκε σε όλους τους πελάτες κατα μήκος μιας διαδρομής που ξεκινά από τον κόμβο - αποθήκη (v_0) και καταλήγει στον κόμβο v_k .

Το δεύτερο κριτήριο εξασφαλίζει οτι το όχημα θα έχει αρκετό χώρο για να παραλάβει όλα τα φορτία των πελατών πάνω σε μια διαδρομή. Σε αυτό το κριτήριο θα πρέπει να ισχύει οτι C_n $(v_k) \leq$ C και C_n $(v_{k+1}) >$ C. Με C_n (v_k) περιγράφεται η συνολική ποσότητα εμπορευμάτων που παραλαμβάνεται από όλους τους πελάτες κατα μήκος μιας διαδρομής από τον κόμβο - αποθήκη (v_0) και καταλήγει στον κόμβο v_k.

Το τρίτο κριτήριο που ερευνάται εξασφαλίζει οτι η χωρητικότητα του οχήματος δεν θα παραβιαστεί σε καμία περίπτωση. Μια τέτοια παραβίαση εξαρτάται άμεσα από την σειρά εξυπηρέτησης των πελατών. Δηλαδή, αν γίνει η υπόθεση οτι το όχημα έχει ένα αρχικό φορτίο L(1) $\leq$ C, τότε μετά την προσπέλαση του κόμβου v_k, το συνολικό φορτίο θα είναι L(v_k) = C_n (v_k) + L(1) - C_d (v_k). Η εξίσωση αυτή δείχνει πως το τρέχον φορτίο μπορεί να ξεπεράσει την χωρητικότητα του οχήματος, που έχει δωθεί σαν όριο στα δεδομένα του προβλήματος. Σε μια τέτοια περίπτωση, η διαδρομή χαρακτηρίζεται αυτόματα μη εφικτή, διότι το όχημα δεν μπορεί να εξυπηρετήσει τον επόμενο κόμβο (κ+1).

Όπως αναφέρθηκε παραπάνω στους περιορισμούς, υπάρχει μια προτεραιότητα μεταξύ linehaul και backhaul πελατών. Ο περιορισμός της προτεραιότητας είναι από τα πιο σημαντικά στοιχεία σε αυτή την κατηγορία προβλημάτων. Γι'αυτό και σε πολλές περιπτώσεις της βιβλιογραφίας τονίζεται οτι αν δεν εξυπηρετηθούν πρώτα όλοι οι linehaul πελάτες, το κάθε όχημα δεν έχει κανένα δικαίωμα να αρχίσει να εξυπηρετεί backhaul πελάτες. Ταυτόχρονα, πρέπει να φροντίζεται οτι τα περιθώρια χωρητικότητας κάθε οχήματος δεν θα παραβιαστούν από κανένα πελάτη, ανεξάρτητα αν είναι linehaul ή backhaul.

- <u>VRP with Pick-up and Delivery (VRPPD)</u>: Αποτελεί μια από τις πιο ιδιάζουσες περιπτώσεις προβλημάτων δρομολόγησης οχημάτων. Τα βασικά της χαρακτηριστικά είναι παρόμοια με τα χαρακτηριστικά που έχουν αναφερθεί στην κατηγορία των προβλημάτων που έχουν ενεργοποιημένη την «οπίσθεια εξυπηρέτηση». Η κύρια διαφορά έγκειται στο χαρακτηριστικό πως σε αυτή την κατηγορία προβλημάτων, τα οχήματα είναι ανεξάρτητα κατεύθυνσης και οι περιορισμοί εισάγονται μόνο στην ποσότητα φορτίου που μπορεί να μεταφέρει κάθε όχημα. Και σε αυτή την περίπτωση ισχύει η εξίσωση που διατυπώθηκε στην κατηγορία των "VRP with Backhauls", L(v_k) = C_n (v_k) + L(1) - C_d (v_k). Η εξίσωση αυτή δείχνει πως το τρέχον φορτίο δεν μπορεί να ξεπεράσει την χωρητικότητα του οχήματος, που έχει δωθεί σαν όριο στα δεδομένα του προβλήματος. Σε περίπτωση που δεν ισχύσει αυτή η εξίσωση, η διαδρομή δεν ακολουθείται.

- <u>VRP with Time Windows (VRPTW)</u>: Σε αυτή την κατηγορία προβλημάτων, μερικά σημεία ζήτησης απαιτούν να εξυπηρετηθούν από κάποιο όχημα μέσα σε συγκεκριμένο χρονικό διάστημα που ονομάζεται χρονικό παράθυρο. Η βιβλιογραφία αυτής της κατηγορίας προβλημάτων είναι εκτενής με κύριους εκφραστές της τους Cordeau, (2002), Lau (2003), Braysy (2005).Το κυριότερο παράδειγμα αυτής της κατηγορίας είναι η ανάπτυξη του κλάδου των διανομών φαγητού από παραγγελίες πελατών. Στόχος αυτής της κατηγορίας προβλημάτων αποτελεί η ελαχιστοποίηση του κόστους μετάβασης, του αριθμού των οχημάτων που εκτελούν τις μεταφορές, του συνολικού χρόνου μετάβασης και του χρόνου αναμονής προκειμένου να εξυπηρετηθούν όλοι οι πελάτες εντός ενός προκαθορισμένου χρονικού παραθύρου.

  Οι περιορισμοί που εισέρχονται σε αυτή την παραλλαγή του κλασσικού VRP είναι οι εξής:

  i. Μια λύση είναι αδύνατη αν κάποιος πελάτης εξυπηρετείται έξω από τα όρια του χρονοπαραθύρου του.
  ii. Η άφιξη ενός οχήματος πριν το χρονοπαράθυρο σε κάποιον πελάτη, αυξάνει την καθυστέρηση της διαδρομής.
  iii. Κάθε διαδρομή θα πρέπει να ξεκινά και να τερματίζει εντός του χρονικού παραθύρου της αποθήκης (depot).
  iv. Στην περίπτωση χαλαρών χρονικών παραθύρων, μια καθυστερημένη άφιξη δεν επηρεάζει μια βέλτιστη λύση αλλά θα πρέπει, όμως, να ληφθεί υπόψη στον προσδιορισμό της αντικειμενικής συνάρτησης.

  Σύμφωνα με τον Solomon (1995), για μια υποψήφια λύση το κόστος θα είναι: $F_{VRPTW}(S) = \sum_{i=1}^{m}(C_{VRPTW}(Ri) + M)$, όπου M είναι μια σταθερά μεγάλης τιμής. Αυτή η σταθερά προστίθεται γιατί η ελαχιστοποίηση του μεγέθους του στόλου των οχημάτων αποτελεί πρωταρχικό στόχο της κατηγορίας αυτών των προβλημάτων. Γίνεται η υπόθεση οτι αρχικά όλα τα οχήματα αναχωρούν από τις εγκαταστάσεις μετά τον ανεφοδιασμό στη μικρότερη δυνατή χρονική στιγμή του χρονικού παραθύρου. Μια λύση είναι εφικτή αν όλες οι διαδρομές που προτείνονται από αυτή τη λύση είναι εφικτές και εξυπηρετούν κάθε πελάτη με ακριβώς ένα φορτηγό. Αφού έχει βρεθεί η λύση, ρυθμίζεται η χρονική στιγμή αναχώρησης των φορτηγών και εξαλείφεται οποιαδήποτε καθυστέρηση μπορεί να έχει προκύψει, ώστε να είναι πιο ορθό το αποτέλεσμα.

2.3.3 Τεχνικές επίλυσης για τα προβλήματα της δρομολόγησης οχημάτων

Σε αυτό το υποκεφάλαιο γίνεται μια περιγραφή τεχνικών επίλυσης προβλημάτων δρομολόγησης οχημάτων. Οι τεχνικές που έχουν εφαρμοστεί είναι πάρα πολλές και που έχουν δημοσιευτεί ακόμα περισσότερες. Θα γίνει μια προσπάθεια να περιγραφούν οι πιο δημοφιλείς από αυτές. Το πρόβλημα δρομολόγησης οχημάτων (VRP), όπως έχει διατυπωθεί και στην εισαγωγή, ανήκει στην κατηγορία των NP - Hard προβλημάτων. Τα NP - "δύσκολα (Hard)" προβλήματα αποτελούν αντικείμενο μελέτης της Θεωρίας της Πολυπλοκότητας και της Θεωρίας Υπολογισμού. Τέτοιου είδος προβλήματα αποτελούν αυτά που είναι τουλάχιστον τόσο δύσκολα όσο τα NP προβλήματα. Η διαφορά τους έγκειται στο χαρακτηριστικό του πολυωνυμικού χρόνου. Δηλαδή, όλα τα NP - "δύσκολα" προβλήματα μπορούν να αναχθούν σε πολυωνυμικό χρόνο στο υπερσύνολο των NP προβλημάτων. Σύμφωνα με τη Θεωρία Υπολογισμού, τα NP - "δύσκολα" προβλήματα δεν πρέπει να έχουν λύσεις που επιβεβαιώνονται σε πολυωνυμικό χρόνο.

Η πολυπλοκότητα των προβλημάτων δρομολόγησης οχημάτων περιορίζει το πεδίο εύρεσης τεχνικών επίλυσης. Επειδή το πλήθος των κόμβων υπολογισμού είναι σχετικά μεγάλο, κανένας αλγόριθμος ακρίβειας δεν μπορεί να εγγυηθεί την εύρεση της βέλτιστης λύσης. Συνήθως χρησιμοποιούνται διάφορες ευρεστικές και μετα-ευρεστικές μέθοδοι για την προσέγγιση σε μια βέλτιστη λύση. Από όλες τις τεχνικές επίλυσης προβλημάτων, στα πλαίσια της παρούσας μελέτης περίπτωσης θα εξεταστούν οι πιο δημοφιλείς για την αντιμετώπιση προβλημάτων δρομολόγησης ενός στόλου οχημάτων με μόνο περιορισμό την χωρητικότητα των οχημάτων που εκτελούν τις μεταφορές.

Με δεδομένο τον περιορισμό της χωρητικότητας των οχημάτων, ένα πρόβλημα δρομολόγησης οχημάτων (VRP) από κλασσικό μετασχηματίζεται σε "Capacitated VRP (CVRP)", όπως έχει ήδη αναφερθεί σε προηγούμενη παράγραφο. Οι περισσότερες πληροφορίες σχετικά με τις μεθόδους που χρησιμοποιούνται για την επίλυση CVRP προβλημάτων βρίσκονται στα συγγράμματα των Laporte και Nobert (1987) και Toth και Vigo (2002). Σε σύγγραμμα και μελέτες του Laporte αναφέρεται οτι ένα πρόβλημα δρομολόγησης οχημάτων (VRP) μπορεί να αντιμετωπιστεί με μεθόδους ακριβείας. Ωστόσο, με την εξέλιξη της τεχνολογίας, σε μελέτη του Toth και του Vigo το 2002 αναφέρεται οτι μπορούν να χρησιμοποιηθούν ευρεστικές και μετα-ευρεστικές μέθοδοι για την εύρεση της βέλτιστης λύσης.

2.3.3.1 Μέθοδοι Ακριβείας (exact methods)

Μια μέθοδος ακριβείας (exact method) προβλέπει τη συστηματική απαρίθμηση κάθε πιθανής λύσης μέχρις ότου βρεθεί η βέλτιστη δυνατή. Μερικές γνωστές μέθοδοι ακριβείας είναι η μέθοδος "branch and bound" και η μέθοδος "branch and cut".

Η "branch and bound" μέθοδος αναπτύχθηκε στα τέλη της δεκαετίας του 1980 και δημιουργήθηκε με στόχο να δώσει λύση σε πολύπλοκα προβλήματα. Στα πλαίσια αυτής της εργασίας θα ερευνηθεί από τη σκοπιά των προβλημάτων δρομολόγησης. Όπως έχει περιγραφεί σε προηγούμενη παράγραφο της εργασίας, η κατηγοριοποίηση των προβλημάτων δρομολόγησης αλλάζει ανάλογα με τους περιορισμούς και τα χαρακτηριστικά που θέτονται σε κάθε περίπτωση. Ένας "branch and bound" αλγόριθμος αποτελεί την πρώτη προσπάθεια προσέγγισης μιας βέλτιστης λύσης για κάθε ένα από αυτά τα προβλήματα.

Το βασικό χαρακτηριστικό των "branch and bound" αλγορίθμων, που συνιστά και την παραδοχή με την οποία προσεγγίζεται μια βέλτιστη λύση, είναι η χαλάρωση που επιτρέπει στους περιορισμούς του κάθε προβλήματος. Η χαλάρωση αυτή αποτελεί την γνωστή “χαλάρωση κατά Lagrange”. Σύμφωνα με τη μελέτη του Fisher (1985), μια "χαλάρωση" κατά Lagrange είναι μια τεχνική για την προσεγγιστική επίλυση πολύπλοκων και δύσκολων προβλημάτων. Μια "χαλάρωση" πραγματοποιείται σε προβλήματα που μπορούν να χωριστούν οι περιορισμοί σε δυο κατηγορίες, σε αυτούς που επιλύονται εύκολα και σε αυτούς που επειδή δεν μπορούν να λυθούν κάνουν το πρόβλημα πολύπλοκο. Για να επιτευχθεί η “χαλάρωση”, παρακάμπτωνται από το πρόβλημα οι περιορισμοί που δεν μπορούν να επιλυθούν και εισέρχονται στην αντικειμενική συνάρτηση με έναν ειδικό πολλαπλασιαστή (βάρος). Κάθε βάρος αντιπροσωπεύει έναν βαθμό ποινής που εκχωρείται στην τελική λύση που δεν επιλύει ακριβώς τον αντίστοιχο περιορισμό.

Η "branch and bound" μέθοδος είναι μια διαδικασία που χρησιμοποιεί τη στρατηγική του "διαίρει και βασίλευε" για την προσέγγιση μιας βέλτιστης λύσης. Διαχωρίζει, δηλαδή τον χώρο των λύσεων σε υποσύνολα και έπειτα εφαρμόζει τη διαδικασία της βελτιστοποίησης για κάθε ένα από αυτά τα υποσύνολα χωριστά. Έστω οτι ο χώρος των εφικτών λύσεων ενός προβλήματος είναι S. Γνωρίζοντας οτι το πρόβλημα που ερευνάμε είναι NP-Hard πολυπλοκότητας, επιλέγεται για επίλυση η μέθοδος του "branch and bound". Αφού οριστεί το σύνολο των εφικτών λύσεων S, αρχικά εφαρμόζεται η θεωρία του Fisher και πραγματοποιείται μια χαλάρωση στους περιορισμούς του προβλήματος. Με δεδομένη την χαλάρωση των περιορισμών γίνονται δεκτές λύσεις που δεν ανήκουν στο σύνολο των εφικτών λύσεων. Σε αυτό το σημείο υπάρχουν δυο βήματα. Στο πρώτο βήμα, ελέγχεται αν η λύση που επιτυγχάνεται με την χαλάρωση είναι ίση με κάποια από τις λύσεις που ανήκουν στο σύνολο των εφικτών λύσεων. Τότε, αυτή η λύση αποθηκεύεται ως η βέλτιστη λύση του προβλήματος και συνεχίζει ο αλγόριθμος στο δεύτερο βήμα του.

Στο δεύτερο βήμα, εφαρμόζονται με σειρά τα στάδια του αλγορίθμου "branch and bound", προκειμένου να βρεθεί μια λύση που θα προσεγγίζει μια από τις λύσεις του συνόλου S. Λειτουργώντας τον αλγόριθμο, καθορίζονται τα υποσύνολα $\{S_1, S_2, \ldots, S_n\}$ του S. Κάθε ένα από τα υποσύνολα αυτά καλείται υποπρόβλημα ή παιδί του S. Προσθέτουμε τα παιδιά του S στη λίστα των υποψηφίων υποπροβλημάτων (αυτών δηλαδή που αναμένουν εξέταση). Η διαδικασία αυτή λέγεται branching. Για να συνεχίσουμε τον αλγόριθμο, επιλέγουμε ένα από τα υποψήφια υποπροβλήματα και το εξετάζουμε. Σύμφωνα με τον αλγόριθμο, υπάρχουν τρια πιθανά αποτελέσματα:

- Αν βρεθεί μια εφικτή λύση καλύτερη από το πρώτο βήμα, αντικαθιστούμε τη λύση του πρώτου βήματος με αυτή και προχωράει ο αλγόριθμος παρακάτω.
- Αν δεν βρεθεί λύση στο υποπρόβλημα, τότε απορρίπτεται.
- Στην περίπτωση που δεν πρέπει να απορριφθεί το πρόβλημα, επιβάλλεται να ξαναχρησιμοποιείται η τεχνική του branching και να προστεθούν τα παιδιά αυτού του υποπροβλήματος στη λίστα των υπολοίπων υποπροβλημάτων προς εξέταση. Η διαδικασία συνεχίζεται μέχρις ότου να αδειάσει η λίστα των υποψήφιων υποπροβλημάτων. Τότε θα έχει επιτευχθεί η βέλτιστη λύση του προβλήματος.

Μια επέκταση του αλγορίθμου "branch and bound" στα πλαίσια της αντιμετώπισης ενός προβλήματος δρομολόγησης οχημάτων με περιορισμένη χωρητικότητα παρουσίαστηκε το 1981 από τον Christofides et al. Η επέκταση αυτη, που περιγράφτηκε το 1989 στη μελέτη των Christofides & Mingozzi, παρουσίαζε την εξέλιξη των χαλαρώσεων κατά Lagrange και την διεύρυνση του μοντέλου με την κατασκευή m-βαθμού ελάχιστου ζευγνύοντος δένδρου για την εύρεση της βέλτιστης λύσης. Με τον αλγόριθμο αυτό δόθηκε για πρώτη φορά η δυνατότητα να επιλυθούν προβλήματα με μέγιστο τους 30 κόμβους και με τυχαίες Ευκλείδειες αποστάσεις μεταξύ τους.

Το 1994, με τη βοήθεια της τεχνολογίας, αναθεωρήθηκε ο αλγόριθμος του Christofides από τον Fischetti. Στη μελέτη των Fischetti & Toth παρουσιάστηκε μια υλοποίηση "branch and bound" που έλυνε πρόβλημα δρομολόγησης οχημάτων με περιορισμό την χωρητικότητα των οχημάτων χρησιμοποιώντας στην τεχνική τους διπλό χαμηλό όριο στη “χαλάρωση κατά Lagrange” προκειμένου να εξεταστούν τα ασύμμετρα κόστη. Ο αλγόριθμος κατάφερε να επιλύσει με επιτυχία πρόβλημα 300 κόμβων με τυχαίες Ευκλείδειες αποστάσεις. Στον ίδιο αλγόριθμο βασίστηκε ο Fisher (1994) προκειμένου να κατηγοριοποιήσει για πρώτη φορά προβλήματα δρομολόγησης οχημάτων με περιορισμό χρονικές καθυστερήσεις και προβλήματα δρομολόγησης οχημάτων διαφορετικής χωρητικότητας (heterogeneous). Η τεχνική "branch and bound" που δημιουργεί ελάχιστα ζεύγνυα δένδρα, χρησιμοποιείται μέχρι και σήμερα για την επίλυση προβλημάτων δρομολόγησης οχημάτων ανεξάρτητα από περιορισμούς αλλά όχι συχνά λόγω του υπολογιστικού κόστους που έχει.

Μια άλλη τεχνική επίλυσης προβλημάτων είναι η μέθοδος "branch and cut". Σύμφωνα με τον Laporte, αρκετοί ερευνητές αναφέρουν οτι αποτελεί εδώ και δεκαετίες την καλύτερη μέθοδο επίλυσης προβλημάτων CVRP. Έχει αναπτυχθεί αρκετή έρευνα γύρω από αυτή την τεχνική με αφορμή την επιτυχία που έτυχαν οι θεωρίες πολύεδρων της συνδυαστικής. Η χρήση της μεθόδου "branch and cut" για την επίλυση προβλημάτων δρομολόγησης οχημάτων με περιορισμό τη χωρητικότητα έχει τις ρίζες της στον αλγόριθμο του Laporte et al (1985).

Ο αλγόριθμος του Laporte περιγράφεται από τις παρακάτω εξισώσεις όπως δημοσιεύονται στη μελέτη του "Vehicle Routing":

$$\text{(CVRP1)} \quad \text{minimize} \quad \sum_{e \in E} c_e x_e \tag{1}$$

subject to

$$\sum_{e \in \delta(i)} x_e = 2, \quad i \in V \setminus \{0\}, \tag{2}$$

$$\sum_{e \in \delta(0)} x_e = 2m, \tag{3}$$

$$\sum_{e \in \delta(S)} x_e \geqslant 2r(S), \quad S \subseteq V \setminus \{0\}, S \neq \varnothing, \tag{4}$$

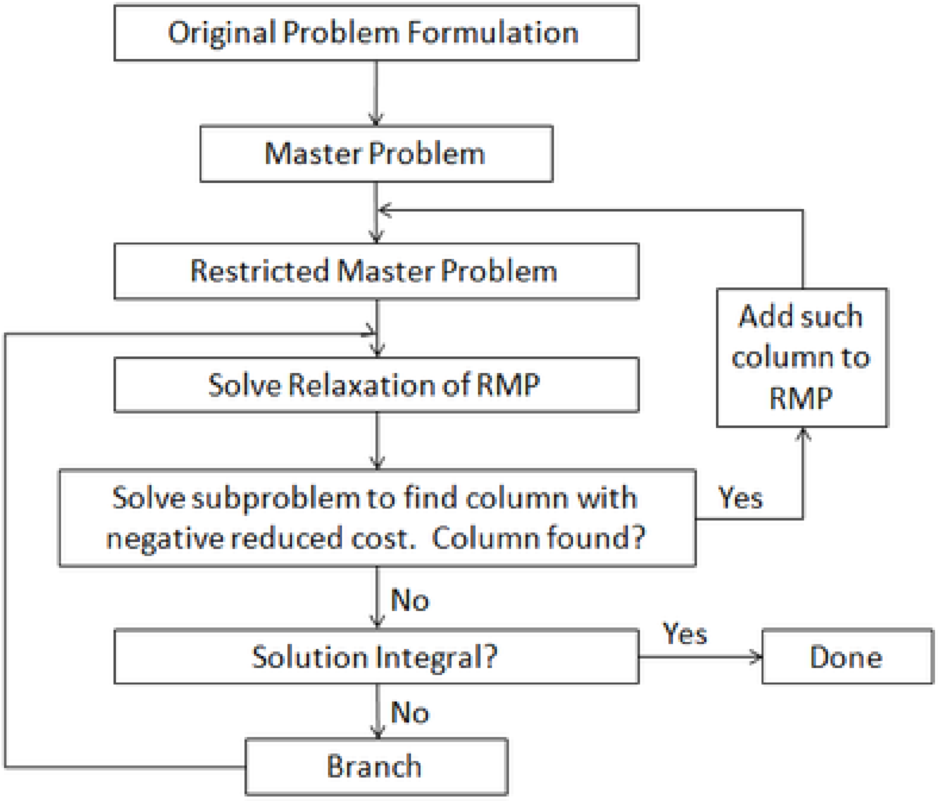
$$x_e \in \{0, 1\}, \quad e \notin \delta(0), \tag{5}$$

$$x_e \in \{0, 1, 2\}, \quad e \in \delta(0). \tag{6}$$

Εικόνα 2.3.3.1α : Μαθηματικό μοντέλο του Laporte για την περιγραφή των CVRP

Ο αλγόριθμος του Laporte ήταν σε θέση να λύσει προβλήματα μέχρι 60 πελατών που ενώνονται μεταξύ τους με ευκλείδειες και μη αποστάσεις και μέχρι 3 οχήματα. Το πετυχαίνει χρησιμοποιώντας γραμμικό προγραμματισμό με δεδομένο το υπόδειγμα ενός CVRP απαλλαγμένο από τους περιορισμούς της χωρητικότητάς του. Αυτό το υπόδειγμα το χρησιμοποιεί για να επιλύσει ένα κλασσικό μοντέλο VRP με περιορισμούς την χωρητικότητα των οχημάτων και την μέγιστη απόσταση που μπορούν να διανύσουν τα οχήματα.

Βασισμένοι πάνω σε αυτή τη θεωρία, οι Cornuéjols και Harche το 1993 παρουσίασαν την πρώτη μελέτη που έκανε χρήση των πολύεδρων για την επίλυση προβλημάτων. Σε αυτή τη μελέτη αποδείχτηκε οτι ένα CVRP μπορεί να παρουσιαστεί με γραφικό τρόπο δημιουργώντας ένα Graphical VRP στο οποίο τα οχήματα θα μπορούσαν να επισκεφτούν τους πελάτες τους περισσότερες από μια φορές. Αυτό αποτελούσε και το πρώτο πλήρως διαμορφωμένο πολύεδρο με ανεπτυγμένες όλες τις ιδιότητές του, όπως τον βαθμό του και τους περιορισμούς χωρητικότητάς του. Την μελέτη των Cornuéjols και Harche επέκτειναν οι Augerat et al. το 1995 παρουσιάζοντας την πρώτη πλήρως ανεπτυγμένη τεχνική επίλυσης ενός CVRP με την μέθοδο "branch and cut". Οι Augerat et al. βασίστηκαν πολύ στις νέες ευρεστικές θεωρίες και κατάφεραν να αποδείξουν οτι η θεωρία τους δουλεύει επιλύοντας πρόβλημα CVRP 134 πελατών.



Εικόνα 2.3.3.1β: Περιγραφή βημάτων αλγοριθμου "branch and cut" σε μορφή ψευδογλώσσας. (Christofides et al. (1981))

Μετά από αρκετά χρόνια, η μελέτη των Lysgaard et al. το 2004, ενίσχυσε περισσότερο την μέθοδο "branch and cut". Αυτό έγινε δυνατό με την εισαγωγή του αλγορίθμου των περικοπών Gomory (Gomory cuts). Ο αλγόριθμος των "Gomory cuts" βοηθάει στην επίλυση προβλημάτων γραμμικού προγραμματισμού ακεραίων με συνεχείς χαλαρώσεις τύπου Lagrange και στην παραγωγή ανισοτήτων, μέσω της βέλτιστης λύσης, που θα ανατροφοδοτούνται στο αρχικό πρόβλημα πριν τις χαλαρώσεις και θα επαναβελτιστοποιείται. Η διαδικασία παραγωγής ανισοτήτων για κάθε χαλάρωση που θα δημιουργείται επαναλαμβάνεται όσες φορές απαιτείται προκειμένου να βρεθεί λύση, η οποία θα είναι ακέραια και βέλτιστη για το αρχικό πρόβλημα. Η προσέγγιση αυτή των

Lysgaard et al. ήταν ικανή να λύσει μέσα σε μέτριους υπολογιστικούς χρόνους προβλήματα που είχαν ήδη επιλυθεί.

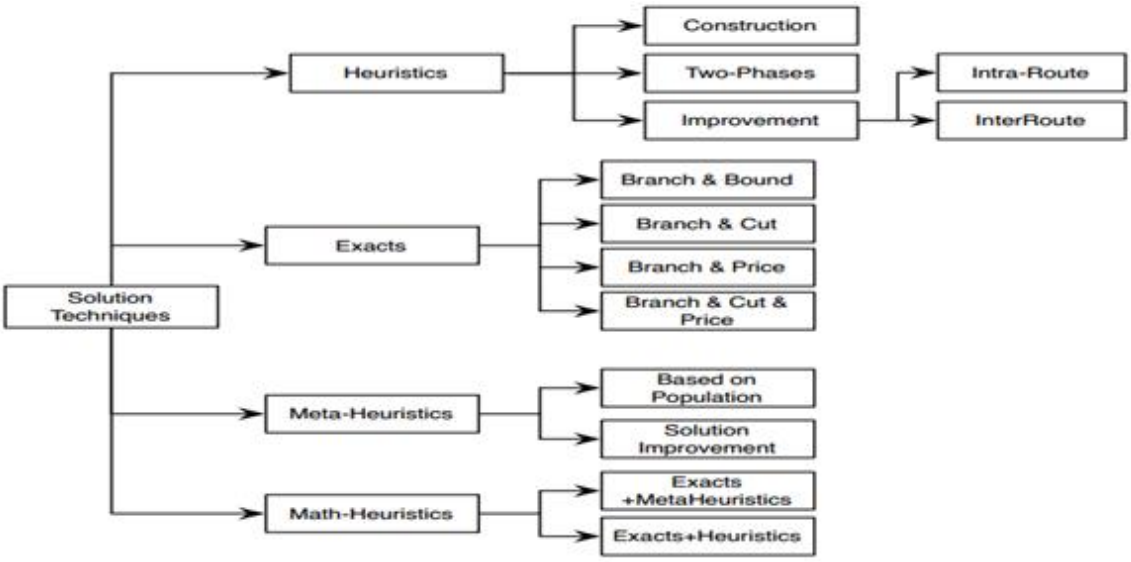
Τέλος, οι Fukasawa et al. (2006) έχουν προτείνει μια επιτυχημένη παραλλαγή του αλγορίθμου "branch and cut" με την είσοδο στο υπόδειγμα έναν περιορισμό που περιγράφει την τιμολόγηση. Η νέα αυτή παραλλαγή κάνει χρήση του μοντέλου και του αλγορίθμου των Christofides et al. (1981). Αυτή η μέθοδος παράγει αυστηρότερα όρια από άλλους αλγορίθμους "branch and cut" και είναι ικανή για την επίλυση αρκετών, προηγουμένως άλυτων περιπτώσεων, με μέγιστο όριο τους 75 πελάτες.

Άλλοι αλγόριθμοι της μεθόδου "branch and cut" περιγράφονται στο Achuthan et al. (1996, 2003) και Blasum και Hochstättler (2000). Επίσης, πρέπει να αναφερθεί οτι μια πολυεδρική δομή μιας ειδικής περίπτωσης προβλημάτων CVRP, όπου όλοι οι πελάτες έχουν ζήτηση ίση με τη μονάδα, μελετήθηκε από τους Campos et al. (1991) και από τους Araque et al. (1990).

2.3.3.2 Ευρεστικές μέθοδοι

Έχει προταθεί ένας εντυπωσιακός αριθμός ευρεστικών αλγορίθμων για την επίλυση προβλημάτων δρομολόγησης οχημάτων. Στην αρχή ξεκίνησαν ως απλοί αλγόριθμοι κατασκευής διαδρομών, ενώ πλέον πιο πρόσφατα έχουν κατασκευαστεί και αλγόριθμοι για μια νέα κατηγορία μετα-ευρεστικών τεχνικών. Για κάθε έναν από τους ευρεστικούς αλγορίθμους που δοκιμάζονται για την επίλυση των VRP, λαμβάνεται ως δεδομένο οτι το πρόβλημα είναι NP-Hard. Αυτό αναφέρεται γιατί για κάθε διαφορετική μέθοδο, είναι σημαντικό να λαμβάνονται υπόψη όλοι οι περιορισμοί ώστε να προσαρμόζεται ο αλγόριθμος, ανεξάρτητα από το αν θα επηρεαστεί η απόδοσή του.

Στα πλαίσια της παρούσας εργασίας χρησιμοποιώντας την ταξινόμηση των Laporte και Sermet, θα περιγραφούν οι κλασσικές ευρεστικές μέθοδοι της κατασκευής διαδρομής, των δυο φάσεων και μεθόδων βελτίωσης μιας διαδρομής.



Εικόνα 2.3.3.2: Κατηγοριοποίηση τεχνικών επίλυσης ενός VRP

Ευρεστικές μέθοδοι κατασκευής διαδρομών

Οι ευρεστικές μέθοδοι κατασκευής διαδρομών αποτελούν από τις πρώτες ευρεστικές μεθόδους για την επίλυση προβλημάτων CVRP και εξακολουθούν να αποτελούν τον πυρήνα για πάρα πολλές εφαρμογές λογισμικού που αναλαμβάνουν απαιτήσεις δρομολόγησης. Αυτοί οι αλγόριθμοι, συνήθως, ξεκινούν από μια άδεια κατάσταση λειτουργίας και επαναληπτικά κατασκευάζει δρόμους προσθέτοντας σε κάθε επανάληψη του αλγορίθμου έναν ή περισσότερους πελάτες μέχρι να δρομολογηθούν όλοι. Οι αλγόριθμοι για την κατασκευή δρόμων διαχωρίζονται σε αλγόριθμους κατασκευής διαδοχικών δρόμων και σε αλγόριθμους κατασκευής παράλληλων δρόμων, ανάλογα με τις απαιτήσεις δρομολόγησης για την ικανοποίηση κάποιου πελάτη. Το βασικό χαρακτηριστικό των αλγορίθμων κατασκευής διαδοχικών δρόμων είναι η επέκταση μιας μόνο διαδρομής κάθε φορά. Αντίθετα, οι αλγόριθμοι κατασκευής παράλληλων διαδρομών είναι σχεδιασμένοι ώστε να εξετάζουν και να επεκτείνουν περισσότερες από μια διαδρομές κάθε φορά.

Οι αλγόριθμοι αυτής της ευρεστικής τεχνικής χαρακτηρίζονται, από την δημιουργία τους, από τρία βασικά κριτήρια. Ένα κριτήριο εκκίνησης, ένα κριτήριο επιλογής που καθορίζει ποιοί πελάτες θα εισέρχονται στην τρέχουσα επανάληψη και ένα κριτήριο εισαγωγής προκειμένου να μπορεί να αποφασίζεται που θα εντοπίζονται οι επιλεγμένοι πελάτες στις τρέχουσες οδούς.

Ο πρώτος και ο πιο δημοφιλής ευρεστικός αλγόριθμος αυτής της κατηγορίας προτάθηκε από τους Clarke και Wright (1964) και βασίζεται στην έννοια της "αποταμίευσης". Σύμφωνα με τους συγγραφείς, η έννοια της "αποταμίευσης" περιλαμβάνει μια εκτίμηση της ελαχιστοποίησης τους κόστους που προκύπτει από την εξυπηρέτηση n πελατών διαδοχικά στην ίδια διαδρομή και όχι σε ξεχωριστές διαδρομές. Για παράδειγμα, αν με i συμβολιστεί ο τελευταίος πελάτης μιας διαδρομής και με j συμβολιστεί ο πρώτος πελάτης της επόμενης επιλογής διαδρομής, η σχετική εξοικονόμηση (ή αποταμίευση) ορίζεται ως $S_{ij} = C_{i0} + C_{0j} - C_{ij}$. Αν η σχετική εξοικονόμηση είναι θετική, τότε τα οχήματα μπορούν να εξυπηρετήσουν τους i και j διαδοχικά σε μια διαδρομή και η επιχείρηση θα έχει καλύτερη κερδοφορία.

Ο αλγόριθμος των Clarke και Wright ερευνά και καταχωρεί όλους τους πελάτες σε ζεύγη και ταξινομεί τις σχετικές αποταμιεύσεις που προκύπτουν σε μη-αυξανόμενη σειρά. Η εκκίνηση του αλγορίθμου γίνεται με μια λύση στην οποία κάθε πελάτης εμφανίζεται ξεχωριστά σε μια διαδρομή. Στη συνέχεια, εξετάζεται το ζεύγος των πελατών ως λίστα και όπου είναι εφικτό, οι δύο διαδρομές συγχωνεύονται. Μια συγχώνευση διαδρομής γίνεται αποδεκτή μόνο αν το αποτέλεσμα της σχετικής αποταμίευσης (S_{ij}) είναι μη αρνητικό. Ωστόσο, αν ο αριθμός που πρόκειται να ελαχιστοποιηθεί στο πρόβλημα αφορά τον αριθμό των οχημάτων που πρόκειται να χρησιμοποιηθούν, τότε γίνεται δεκτή και μια συγχώνευση αρνητικής σχετικής αποταμίευσης.

Έχουν προταθεί και έχουν εφαρμοστεί διάφορες βελτιώσεις της προσέγγισης αυτής. Οι Golden και al. (1977), Nelson et al. (1985), Desrochers και Verhoog (1989), Altinkemer και Gavish (1991), Wark και Holt (1994) είναι μερικές από τις πιο δημοφιλείς προσεγγίσεις που έχουν δημοσιεύτεί πάνω στον αλγόριθμο της "αποταμίευσης". Τα αποτελέσματα που ελήφθησαν με αυτούς τους αλγόριθμους είναι γενικά καλύτερα από εκείνα αυτών που πρωτοδημοσιεύτηκαν, ωστόσο οι αλγόριθμοι που βασίζονται σε αντιστοιχία για την εύρεση των βέλτιστων μονοπατιών απαιτούν υπερβολικά αρκετό υπολογιστικό χρόνο.

Μια άλλη κλασική μέθοδος ευρεστικής κατασκευής μονοπατιών είναι ο αλγόριθμος διαδοχικής εισαγωγής μονοπατιών των Mole και Jameson (1976). Ο αλγόριθμος χρησιμοποιεί ως κριτήριο επιλογής και εισαγωγής την αξιολόγηση της επιπλέον απόστασης που προκύπτει από την εισαγωγή ενός πελάτη k που δεν έχει δρομολογηθεί ακόμα, μεταξύ δύο διαδοχικών πελάτών i και j της τρέχουσας διαδρομής, δηλαδή $a(i, k, j) = C_{ik} + C_{jk} - \lambda C_{ij}$, όπου λ είναι μια παράμετρος που ελέγχεται από κάποιον χρήστη. Στη μελέτη των Mole και Jameson αναφέρεται οτι έχουν ληφθεί υπόψη και διάφορες παραλλαγές αυτού του κριτηρίου, που επεξεργάζονται και άλλους παράγοντες, όπως η απόσταση του πελάτη από την αποθήκη. Μετά από κάθε εισαγωγή πελάτη, η τρέχουσα διαδρομή αναεώνεται με τη χρήση μιας διαδικασίας 3-opt.

Μια πιο γενική και πιο αποτελεσματική ευρετική μέθοδο εισαγωγής κόμβων σε ένα μονοπάτι προτείνουν οι Christofides et al. (1979). Ο αλγόριθμος αυτός αποτελείται από δυο στάδια εισαγωγής. Στο πρώτο βήμα, ένας αλγόριθμος χρησιμοποιείται για τη διαδοχική εισαγωγή ενός συνόλου εφικτών διαδρομών. Το δεύτερο βήμα είναι μια προσέγγιση για παράλληλη εισαγωγή πελατών σε ένα μονοπάτι. Για κάθε διαδρομή που καθορίζεται στο πρώτο βήμα, επιλέγονται ένας πελάτης και μια σειρά από διαδρομές που ξεκινούν ή περιέχουν αυτούς τους πελάτες. Οι υπόλοιποι μη δρομολογημένοι πελάτες στη συνέχεια εισάγονται με τη χρήση ενός "κριτηρίου λύπης (regret criterion)", στο οποίο η διαφορά μεταξύ του πρώτου και του δεύτερου καλύτερου κόστους εισαγωγής λαμβάνεται υπόψη. Η μερική διαδρομή βελτιώνεται μέσω μιας διαδικασίας 3-opt. Ο τελικός αλγόριθμος που προκύπτει από τα παραπάνω βήματα είναι ανώτερος από εκείνον των Mole και Jameson και αντιπροσωπεύει έναν καλό συμβιβασμό μεταξύ της αποτελεσματικότητας και αποδοτικότητας.

Ευρεστικές μέθοδοι δυο - φάσεων (two - phase heuristics)

Οι ευρεστικές μέθοδοι των δυο φάσεων βασίζονται στην αποδόμηση ενός προβλήματος δρομολόγησης οχημάτων VRP, δημιουργώντας δυο βασικά υποπροβλήματα:

- ένα πρόβλημα ομαδοποίησης (clustering): αναλαμβάνει να επιλύσει την κατάτμηση ένος τμήματος των πελατών σε υποσύνολα, το καθένα από τα οποία θα αντιστοιχεί σε μια διαδρομή, και
- ένα πρόβλημα δρομολόγησης (routing): αναλαμβάνει να επιλύσει το πρόβλημα της αλληλουχίας των πελατών σε κάθε διαδρομή.

Σε μια μέθοδο "cluster-first-routing-second", οι πελάτες δέχονται πρώτα μια ομαδοποίηση σε ομάδες και στη συνέχεια οι διαδρομές προσδιορίζονται με κατάλληλη ταξινόμηση των πελατών της κάθε ομάδας. Έχουν προταθεί και υλοποιηθεί αρκετές διαφορετικές τεχνικές για τη φάση της ομαδοποίησης, ενώ για τη φάση της δρομολόγησης αρκεί μια απλή επίλυση ενός προβλήματος περιοδεύοντος πωλητή (TSP).

Ο αλγόριθμος "σκούπα", των Wren (1971), Wren & Holliday (1972) και Gillett & Miller (1974), αναφέρεται συχνά ως η πρώτη προσέγγιση μιας "cluster-first-routing-second" μεθόδου. Ο αλγόριθμος εφαρμόζεται σε πάρα πολλές περιπτώσεις VRP προβλημάτων. Εκκινά με έναν πελάτη επιλέγοντάς τον αυθαίρετα και τυχαία από τη λίστα και στη συνέχεια εκχωρεί διαδοχικά τους υπόλοιπους πελάτες σε κάθε όχημα θεωρώντας τους κατά πολική γωνία με αύξουσα τάξη σε σχέση με την τοποθεσία της αποθήκης και του αρχικού πελάτη. Μόλις ο πελάτης σε μια επανάληψη του αλγορίθμου δεν μπορεί να εκχωρηθεί με επιτυχία στο όχημα που εξετάζεται, μια νέα διαδρομή αρχικοποιείται με αυτόν. Μόλις όλοι οι πελάτες έχουν ανατεθεί σε οχήματα, κάθε διαδρομή ορίζεται ξεχωριστά από την επίλυση ενός TSP.

Μια άλλη ευρεστική μέθοδος δυο φάσεων είναι η "truncated branch-and-bound method", που εισήγαγαν οι Christofides et al. (1979). Σε αυτή τη μέθοδο, το σύνολο των διαδρομών προσδιορίζεται μέσω της προσέγγισης ενός "branch-and-bound" αλγορίθμου, ο οποίος χρησιμοποιεί μια στρατηγική όπου κάθε κλαδί μεταφράζεται σε διαδρομή για φορτηγό. Το δέντρο απόφασης είναι τόσο πολυεπίπεδο, όσος είναι ο αριθμός των διαθέσιμων οχημάτων που μελετώνται στο πρόβλημα. Σε κάθε επίπεδο του δέντρου απόφασης, ένας κόμβος αντιστοιχεί σε μια μερική λύση που αποτελείται από πλήρεις διαδρομές. Οι κόμβοι-απόγονοι αντιστοιχούν σε όλες τις δυνατές διαδρομές, όπου συμπεριλαμβάνεται και το υποσύνολο των πελατών που δεν έχουν δρομολογηθεί ακόμα. Ο χρόνος εκτέλεσης του αλγορίθμου μπορεί να ελεγχθεί, εισάγοντας τον περιορισμό οτι θα δημιουργείται σε κάθε επίπεδο ακριβώς ένα δρομολόγιο.

Το 1995 περιγράφηκε από τους Bramel & Simchi-Levi μια διαφορετική μέθοδος δυο φάσεων, η οποία λειτουργεί με έναν σταθερό αριθμό οχημάτων, έστω m. Ο αλγόριθμος καθορίζει "σπόρους" διαδρομής (route seeds) μέσω της επίλυσης ενός προβλήματος δρομολόγησης οχημάτων με σταθερή χωρητικότητα (CVRP), όπου m στο πλήθος πελάτες επιλέγονται με την ελαχιστοποίηση της συνολικής απόστασης μεταξύ κάθε πελάτη και του πλησιέστερου "σπόρου", και επιβάλλοντας ότι η συνολική ζήτηση για κάθε σπόρο δε θα ξεπερνάει έναν αριθμό Q. Από τη στιγμή που θα έχουν καθοριστεί οι "σπόροι" και οι διαδρομές για κάθε πελάτη ξεχωριστά, οι υπόλοιποι πελάτες εισέρχονται στην τρέχουσα διαδρομή ελαχιστοποιώντας το κόστος εισαγωγής.

Έχουν προταθεί και αναλυθεί διάφοροι τρόποι προσέγγισης για την ελαχιστοποίηση του κόστους εισαγωγής. Πρέπει να σημειωθεί οτι μια διαφορά του αλγορίθμου "σκούπα" που έχει ήδη αναφερθεί παραπάνω από τον συγκεκριμένο αλγόριθμο αποτελεί η απουσία άμεσου ελέγχου του αριθμού των δρομολογίων στην τελική λύση από την πλευρά του αλγορίθμου "σκούπα". Η απόδοση των αλγορίθμων των Christofides et al. και Bramel & Simchi-Levi είναι γενικά συγκρίσιμη με την απόδοση των αλγορίθμων κατασκευής διαδρομής από την άποψη της αποτελεσματικότητας. Η προσέγγιση με βάση την τοποθεσία του Bramel και Simchi-Levi παράγει καλύτερες λύσεις, αλλά απαιτεί πολύ μεγαλύτερη υπολογιστική ισχύ.

Μια διαφορετική οικογένεια μεθόδων δύο φάσεων αποτελούν οι "petal algorithms". Αυτοί δημιουργούν σύνολα εφικτών διαδρομών, που ονομάζονται πέταλα, και επιλέγουν το τελικό υποσύνολο μέσω της επίλυσης ενός μοντέλου εύρεσης τοποθεσίας. Οι Foster & Ryan (1976) και Ryan et al. (1993) έχουν προτείνει ευρεστικούς κανόνες για τον προσδιορισμό του συνόλου των διαδρομών που θα επιλεγεί, ενώ οι Renaud et al. (1996) έχουν περιγράψει μια επέκταση που χρησιμοποιεί μια μέθοδο που αποτελείται από δύο ενσωματωμένες ή τεμνόμενες γραμμές, που ονομάζεται 2-πέταλα (2-petals). Η συνολική απόδοση αυτών των αλγορίθμων είναι γενικά ανώτερη από εκείνη του αλγορίθμου "σκούπα".

Κλείνοντας την αναφορά στις ευρεστικές μεθόδους δυο φάσεων, αξίζει να αναφερθεί η εφαρμογή αλγορίθμων "route-first-cluster-second". Είναι μέθοδοι αντίστροφης διαδικασίας από όσους έχουν ήδη προαναφερθεί. Σε αυτή τη μέθοδο, δημιουργείται ένα πρόβλημα πλανόδιου πωλητή TSP για κάθε πελάτη ξεχωριστά μέχρι και τον τελευταίο και στη συνέχεια αντιμετωπίζεται το πρόβλημα της εύρεσης των βέλτιστων διαδρομών. Υπάρχουν αρκετά παραδείγματα στη βιβλιογραφία, όπως των Beasley (1983) και Bertsimas & Simchi-Levi (1996), ωστόσο η απόδοσή του είναι πολύ πιο φτωχή σε σχέση με τους αλγορίθμους της κατηγορίας " cluster-first-routing-second ".

Ευρεστικές μέθοδοι βελτίωσης διαδρομών (route improvement heuristics)

Οι αλγόριθμοι τοπικής αναζήτησης χρησιμοποιούνται συχνά για να βελτιώσουν λύσεις που παράγονται από άλλου ευρεστικούς αλγόριθμους. Ξεκινώντας από μια συγκεκριμένη λύση, μια μέθοδος τοπικής αναζήτησης εφαρμόζει απλές τροποποιήσεις, όπως ανταλλαγές ή κινήσεις πελατών στο δίκτυο, προκειμένου να επιτύχει λύσεις γειτνίασης με όσο καλύτερο κόστος είναι εφικτό. Εάν βρεθεί μια λύση που βελτιώνει, τότε γίνεται η τρέχουσα λύση και επαναλαμβάνεται η διαδικασία, αλλιώς ταυτοποιείται και καταχωρείται ένα τοπικό ελάχιστο.

Υπάρχει μια μεγάλη ποικιλία από διαθέσιμες γειτονιές. Αυτές μπορούν να υποδιαιρεθούν σε intra-route γειτονιές, εάν λειτουργούν σε μια ενιαία διαδρομή κάθε φορά, ή σε inter-route γειτονιές, αν χρησιμοποιούν περισσότερες από μία διαδρομές ταυτόχρονα. Ο πιο κοινός τύπος είναι η ευρεστική μέθοδος γειτονιάς λ-opt του Lin (1965) για το πρόβλημα του πλανόδιου πωλητή, όπου λ ακμές απομακρύνονται από την τρέχουσα λύση του αλγορίθμου και αντικαθιστώνται από λ άλλες. Ο υπολογιστικός χρόνος που απαιτείται για την εξέταση όλων των γειτόνων μιας λύσης είναι ανάλογος με n^{λ}. Με αυτόν τον τρόπο, μόνο τα λ = 2 ή λ= 3 χρησιμοποιούνται στην πράξη. Ως εναλλακτική λύση, μπορεί κανείς να χρησιμοποιήσει περιορισμένες γειτονιές που χαρακτηρίζονται από υποσύνολα των κινήσεων που σχετίζονται με μεγαλύτερες τιμές της μεταβλητής λ, όπως οι ανταλλαγές του Or (Or exchanges), που δημιουργήθηκαν από τον Or το 1976 ή μέθοδος των Renaud et al. (1996) για γειτονιές με ανταλλαγές 4-opt, στην οποία θεωρείται ότι μόνο ένα υποσύνολο όλων των πιθανών ανταλλαγών 4-opt δίνει τη λύση. Πιο πολύπλοκες inter-route γειτονιές αναλύθηκαν από τους Thompson & Psaraftis (1993), Van Breedam (1994), και Kindervater & Savelsbergh (1997).

2.3.3.3 Μετα-ευρεστικές μέθοδοι

Έχουν εφαρμοστεί πάρα πολλές μεταευρεστικές μέθοδοι (metaheuristics) στο VRP. Με σεβασμό στις κλασσικές ευρεστικές θεωρίες, τα metaheuristics εκτελούν μια πιο λεπτομερή αναζήτηση για την εύρεση της λύσης και είναι λιγότερο πιθανό να καταλήξουν με κάποιο τοπικό βέλτιστο ως απάντηση. Οι μέθοδοι των metaheuristics μπορούν να χωριστούν σε τρεις κατηγορίες:

- τοπικής αναζήτησης, που περιλαμβάνουν τους αλγορίθμους της προσομοιωμένης ανόπτησης (simulated annealing), της ντετερμινιστικής ανόπτησης (deterministic annealing), και της απαγορευμένης αναζήτησης (tabu search),
- αναζήτησης μέσω του πληθυσμού, που περιλαμβάνουν τους γενετικούς αλγόριθμους (genetic algorithms) καθώς και τους αλγορίθμους προσαρμοστικών διαδικασιών της μνήμης, και
- μηχανισμούς μάθησης, που περιλαμβάνουν τα νευρωνικά δίκτυα καθώς και αλγορίθμους που βασίζονται σε νοημοσύνη σμήνους.

Οι καλύτερες ευρεστικές θεωρίες (heuristics) συχνά συνδυάζουν ιδέες που δανείστηκαν από διαφορετικές αρχές μεταευρεστικών αλγορίθμων (metaheuristics). Πρόσφατες έρευνες για metaheuristics για την επίλυση VRP προβλημάτων, μπορούν να βρεθούν στις μελέτες των Gendreau et al. (2002), Cordeau & Laporte (2004) και Cordeau et al. (2005).

Οι αλγόριθμοι τοπικής αναζήτησης εξερευνούν το πεδίο λύσεων πραγματοποιώντας επαναληπτικά μεταβάσεις από μια λύση x_t σε μια λύση x_{t-1}, η οποία βρίσκεται στη γειτονιά του x_t, μέχρι να εφαρμοστεί κάποιο κριτήριο ακινητοποίησης. Αν με f(x) συμβολίζεται το κόστος της λύσης x, τότε το κόστος f(x_{t-1}) δεν είναι απαραιτήτως μικρότερη από f(x_t). Ως αποτέλεσμα, οι μηχανισμοί θα πρέπει να εφαρμοστούν για να αποφευχθεί ο βρόχος. Στην περίπτωση μιας προσομοιωμένης ανόπτησης (simulated annealing), μια λύση x έχει επιλεχθεί τυχαία από τη γειτονιά του x_t. Αν f(x) $\leq$ f(x_t), τότε x_{t-1} := x. Σε κάθε άλλη περίπτωση ισχύει:

$$x_{t-1} := \begin{cases} x & with\, probability\, p_t \\ x_t & with\, probability\, 1 - p_t \end{cases}$$

όπου p_t είναι μια φθίνουσα συνάρτηση του t και του f(x)-f(x_t). Αυτή η πιθανότητα είναι συχνά ίση με:

$$p_t = \exp\left(-\frac{f(x) - f(x_t)}{\theta_t}\right)$$

, όπου θ_t είναι η θερμοκρασία στην επανάληψη t, συχνά ορίσμένη ως μη-αυξανόμενη συνάρτηση (nonincreasing function) του t. Παρόμοια είναι και η ντετερμινιστική ανόπτηση (deterministic annealing) (Dueck, 1990, 1993). Υπάρχουν δύο κύριες εκδοχές αυτού του αλγορίθμου: σε έναν αλγόριθμο σχεδιασμένο να υπάρχει ένα κατώφλι αποδοχής, το x_{t-1} := x εάν f(x) < f(x_t)+θ_1, όπου θ_1 είναι μια παράμετρος ελεγχόμενη από τον χρήστη. Ανατρέχοντας όλες τις εγγραφές, μια εγγραφή είναι η βέλτιστη γνωστή λύση x*, με x_{t-1} := x αν f(x_{t-1})<θ_2 f(x*), όπου θ_2 είναι επίσης μια παράμετρος ελεγχόμενη από τον χρήστη. Σε μια "αναζήτηση με απαγόρευση (Tabu search)", προκειμένου να αποφευχθεί η επανάληψη, μια λύση, η όποια διαθέτει κάποια δεδομένα χαρακτηριστικά του x_{t-1}, δηλώνεται ως "απαγορευμένη" να χρησιμοποιηθεί για έναν αριθμό επαναλήψεων. Στην επανάληψη t, η αναζήτηση κινείται προς την καλύτερη μη-απαγορευμένη λύση x στη γειτονιά του x_t.

Αυτοί οι αλγόριθμοι τοπικής αναζήτησης είναι σπάνια υλοποιημένοι στην βασική τους έκδοση και η επιτυχία τους εξαρτάται από την προσεκτική εφαρμογή διαφόρων μηχανισμών απαραίτητων για τη λειτουργία τους. Ο κανόνας που χρησιμοποιείται για να καθορίζονται οι γειτονιές είναι κρίσιμος για τους περισσότερους ευρεστικούς αλγόριθμους τοπικής αναζήτησης. Σε μια προσομοιωμένη ανόπτηση (simulated annealing) έχουν προταθεί αρκετοί κανόνες για τον καθορισμό του θ_t (Osman, 1993). Οι αλγόριθμοι "απαγορευμένης αναζήτησης" (Tabu search) βασίζονται σε διάφορες στρατηγικές για την εφαρμογή διαφοροποίησης αναζητήσεων (που παίζει το ρόλο της μακροπρόθεσμης μνήμης) και εντατικοποίησης αναζητήσεων, η οποία επιτείνει την αναζήτηση σε μια υποσχόμενη περιοχή.

Οι αλγόριθμοι αναζήτησης μέσω του πληθυσμού λειτουργούν σε πάρα πολλές γενεές πληθυσμού λύσεων. Στη γενετική έρευνα είναι σύνηθες να επαναλαμβάνεται η ακόλουθη λειτουργία k φορές: γίνεται εξόρυξη δύο γονεϊκών διαλυμάτων από τον πληθυσμό λύσεων και δημιουργούνται δύο απόγονοι ως λύσεις μέσω μιας λειτουργίας "crossover" και εφαρμόζεται μια λειτουργία μετάλλαξης σε κάθε απόγονο. Στη συνέχεια αφαιρούνται τα 2k χειρότερα στοιχεία από τον πληθυσμό και γίνεται αντικατάστασή τους με τους 2k απογόνους.

Υπάρχουν πάρα πολλοί δημοσιεύμενοι κανόνες crossover στη βιβλιογραφία. Για τα προβλήματα γενετικής αλληλουχίας είναι διαθέσιμες για μελέτη οι δημοσιεύσεις των Bean (1994), Potvin (1996), Drezner (2003) και Prins (2004). Σε διαδικασίες προσαρμογής της μνήμης, ένας απόγονος δημιουργείται από την εξόρυξη και ανασυγκρότηση στοιχείων πολλών γονέων. Στην αρχική εκδοχή που προτείνεται από τους Rochat και Taillard (1995) σχετικά με τα προβλήματα δρομολόγησης οχημάτων (VRP), γίνεται εξόρυξη των μη επικαλυπτόμενων διαδρομών που προέρχονται από διάφορους γονείς προκειμένου να δημιουργηθεί μια μερική λύση. Αυτή η λύση, στη συνέχεια, σταδιακά ολοκληρώνεται και βελτιστοποιείται μέσα από την μέθοδο "απαγορευμένων αναζητήσεων".

Τα νευρωνικά δίκτυα είναι μοντέλα που αποτελούνται από πλούσιες διασυνδεδεμένες μονάδες από σταθμισμένες συνδέσεις, όπως οι νευρώνες στον εγκέφαλο. Κατασκευάζουν σταδιακά μια λύση χρησιμοποιώντας έναν μηχανισμό ανάδρασης που τροποποιεί τα βάρη των συνδέσμων ώστε μια παρατηρούμενη έξοδος να ταιριάζει όσο πιο πιστά είναι δυνατόν τα κριτήρια περιγραφής που έχουν γίνει είσοδος. Στον τομέα της επίλυσης προβλημάτων δρομολόγησης οχημάτων, τα μοντέλα νευρωνικών δικτύων "ελαστικό δίχτυ (elastic net)" και "χάρτης αυτοοργάνωσης (self-organizing map)" αποτελούν παραμορφωμένα πρότυπα που προσαρμόζονται στο περίγραμμα των κορυφών ενός δικτύου δρομολόγησης οχημάτων για να εξάγει μια εφικτή λύση. Ένα παράδειγμα παρέχεται από την μελέτη του Ghaziri (1993). Οι αλγόριθμοι αποικίας μυρμηγκιών (ant colony algorithms) (Dorigo et al., 1999) χρησιμοποιούν, επίσης, έναν μηχανισμό μάθησης. Προέρχονται από μια αναλογία με την αποικία των μυρμηγκιών που ένα μυρμήγκι καθορίζει κάποια ποσότητα φερομόνης στα ίχνη του κατά την αναζήτηση τροφής. Με τον καιρό όλο και περισσότερη φερομόνη εναποτίθεται στα πιο πολυσύχναστα μονοπάτια. Κατά τη διαδικασία κατασκευής μιας λύσης VRP, σε μια κίνηση στο δίκτυο μπορεί να ανατεθεί μια πιο μεγάλη πιθανότητα να επιλεγεί ως βέλτιστη διαδρομή αν έχει οδηγήσει το γενικό πρόβλημα σε καλύτερη λύση από όλες τις προηγούμενες επαναλήψεις.

Το 2005 οι Cordeau et al. δημοσίευσαν μια συγκριτική μελέτη από τους 14 πιο πρόσφατους μέχρι τότε ευρεστικούς αλγορίθμους πάνω στην αντιμετώπιση προβλημάτων VRP, όπως παρουσιάζονται στα 14 παραδείγματα VRP των Christofides et al. (1979), με πλήθος κόμβων από 50 έως 199, και των 20 μεγαλύτερων παραδειγμάτων των Li et al. (2005) με πλήθος κόμβων από 200 έως 480. Οι περισσότεροι μετα-ευρεστικοί αλγόριθμοι που χρησιμοποιήθηκαν στη συγκριτική μελέτη των Cordeau et al. συνεχώς έδιναν λύσεις των οποίων η ποιότητα ήταν αρκετά πιο υψηλή από αυτή των κλασσικών ευρεστικών μθόδων.

Στα παραδείγματα των Christofides et al. (1979), τις πιο βέλτιστες λύσεις έδωσαν οι Taillard (1993), Rachat & Taillard (1995) και οι Mester & Braysy (2005). Ωστόσο, κάνοντας σύγκριση το ίδιο σετ παραδειγμάτων με κριτήριο την ταχύτητα και τον υπολογιστικό χρόνο που χρειάζεται, οι αλγόριθμοι των Mester & Braysy (2005), Tarantilis & Kiranoudis (2002) και του Prins (2004) είναι οι πιο αποτελεσματικοί. Σε αυτό το σημείο θα πρέπει να σημειωθεί οτι οι τρεις παραπάνω μέθοδοι αποτελούν έναν συνδυασμό από αλγορίθμους αναζήτησης μέσω πληθυσμού και αλγορίθμων τοπικής αναζήτησης.Με αυτόν τον τρόπο επιτυγχάνεται μια πιο ευρεία εξερεύνηση του χώρου λύσεων.

Κλείνοντας, όπως σημειώνεται από τους Cordeau et al. (2002) οι ευρεστικοί αλγόριθμοι (heuristics) δεν θα πρέπει να κρίνονται με γνώμονα την ταχύτητά τους και την αποτελεσματικότητά τους. Τα κριτήρια της απλότητας και την εύκολης προσαρμογής στο πρόβλημα είναι εξίσου σημαντικά. Υπο αυτό το πρίσμα, ο αλγόριθμος των Li et al. (2005) που ερευνά εγγραφή προς εγγραφή το πρόβλημα, εμφανίζει μοναδικό ενδιαφέρον. Αυτός ο αλγόριθμος παρουσιάζει μια απλή δομή και μια απαράμιλλη ικανότητα να παράγει ποιοτικές λύσεις. Σε συνδυασμό με τον αλγόριθμο των Rachat & Taillard (1995) που είναι δομημένος με γνώμονα την καλύτερη προσαρμογή στα χαρακτηριστικά του προβλήματος, αποτελούν χρήσιμες ιδέες που μπορούν με μεγάλη άνεση να εφαρμοστούν σε οποιοδήποτε πρόβλημα δρομολόγησης οχημάτων, ανεξάρτητα από τους περιορισμούς του.

2.3.3.4 Νοημοσύνη Σμήνους (Swarm Intelligence)

Η Νοημοσύνη Σμήνους (Swarm Intelligence) είναι ένας όρος της τεχνητής νοημοσύνης ο οποίος υποδηλώνει αυτοοργανούμενα αποκεντρωμένα πολυπρακτορικά συστήματα με συλλογική νοημοσύνη. Την ιδέα αυτή εμπνεύστηκαν οι ειδικοί της ρομποτικής από ένα βιολογικό μηχανισμό πολύ διαδεδομένο στη φύση. Σμήνη πουλιών, κοπάδια ψαριών, αγέλες ή κοπάδια ζώων μετακινούνται συντονισμένα σαν ένας μοναδικός υπερογανισμός. Ακόμα πιο εκπληκτική είναι η περίπτωση μερικών εντόμων όπως τα μυρμήγκια, οι σφήκες, οι μέλισσες και, κυρίως, οι τερμίτες, που οργανώνονται και δημιουργούν περίτεχνες φωλιές ή εκδηλώνουν συμπεριφορές που ξεπερνούν τις ικανότητες των μεμονωμένων ατόμων, τα οποία αγνοούν τις εργασίες στις οποίες συμμετέχουν.

Αυτά τα συστήματα αποτελούνται από ένα πληθυσμό απλών παραγόντων, που αλληλεπιδρούν μεταξύ τους και με το περιβάλλον. Οι παράγοντες ακολουθούν πολύ απλούς κανόνες (όπως και στα κυτταρικά αυτόματα) κι ενώ δεν υπάρχει κεντρικός έλεγχος, που να κατευθύνει τη συμπεριφορά τους, οι αλληλεπιδράσεις μεταξύ τους (τοπικές και τυχαίες) οδηγούν στην εμφάνιση ευφυούς συλλογικής συμπεριφοράς (hive mind), άγνωστη στα μεμονωμένα άτομα. Από εδώ προκύπτουν αλγόριθμοι που μπορούν να επιλύσουν προβλήματα, που απλοποιούνται στην εύρεση των βέλτιστων διαδρομών σε διαγράμματα (Graph theory).

Εφαρμογές υπάρχουν σε διάφορους τομείς, από την τεχνητή νοημοσύνη, τηλεπικοινωνίες, μεταφορές, δίκτυα, μέχρι τον προγραμματισμό, υπολογιστικά συστήματα, αναδίπλωση πρωτεϊνών, γενετική, ακόμα και το γνωστό για τις απαιτήσεις σε υπολογιστική ισχύ «πρόβλημα τού περιοδεύοντος πωλητή».

Μοντέλο προσαρμοζόμενης κουλτούρας

Στα πλαίσια της ανάλυσης της ΝΣ θα εξεταστεί αρχικά το μοντέλο της προσαρμοζόμενης κουλτούρας. Το συγκεκριμένο μοντέλο δείχνει τι μπορεί να συμβεί όταν αλληλεπιδρούν πολλοί πράκτορες τόσο μεταξύ τους, όσο και με το περιβάλλον στο οποίο βρίσκονται. Αξίζει να σημειωθεί ότι τα άτομα στο μοντέλο κουλτούρας δεν προσπαθούν να λύσουν το πρόβλημα. Ακολουθούν τα απλά βήματα του αλγόριθμου τα οποία δεν αναφέρουν κάτι σχετικά με την ύπαρξη ενός προβλήματος ή με τον τρόπο επίλυσης του. Παρόλα αυτά διαμέσου αμοιβαίας κοινωνικής αλληλεπίδρασης κάθε άτομο βελτιώνει τη «καταλληλότητά» του (ο όρος είναι λιγότερο κατάλληλος εδώ από ότι στους εξελικτικούς και γενετικούς αλγόριθμους), και η επίδοση του πληθυσμού συνολικά βελτιώνεται. Η διαδικασία που υποδεικνύεται μέσω του αλγορίθμου προσαρμοζόμενης κουλτούρας δεν είναι η καλύτερη δυνατή, αλλά αποτελεί μία καλή εισαγωγή σε κάποιους κοινωνικούς αλγόριθμους όπως είναι ο αλγόριθμος ΝΣ.

Ο αλγόριθμος ΝΣ αρχικά θα εξεταστεί με όρους κοινωνικής και γνωσιακής συμπεριφοράς, παρότι χρησιμοποιείται ευρύτερα ως μέθοδος επίλυσης προβλημάτων στη μηχανική και την επιστήμη υπολογιστών. Ο λόγος είναι ότι μέσα από τις περιγραφόμενες διαδικασίες δίνεται η δυνατότητα βαθύτερης κατανόησης του αλγόριθμου. Θεμελιώδης έννοια στη ΝΣ είναι η δυαδική κωδικοποίηση προβλημάτων (αυτού του είδους η κωδικοποίηση υπάρχει και στους γενετικούς αλγόριθμους).

Μια πολύ απλή κοινωνικογνωσιακή θεωρία υποστηρίζει τα χαρακτηριστικά του μοντέλου προσαρμοζόμενης κουλτούρας και τις κατηγορίες αλγορίθμων νοημοσύνης σμήνους. Αρχικά γίνεται η υπόθεσή ότι οι διαδικασίες του μοντέλου της προσαρμοζόμενης κουλτούρας συνοψίζονται σε καθολικές συμπεριφορές των ατόμων, οι οποίες μπορούν να συγκεντρωθούν και να κατηγοριοποιηθούν στις παρακάτω τρεις αρχές:

- Εκτίμηση: Η τάση για εκτίμηση ερεθισμάτων ως προς την αξιολόγησή τους ως θετικά ή αρνητικά, ελκυστικά ή απωθητικά είναι κατά πάσα πιθανότητα το πλέον διαδεδομένο συμπεριφορικό χαρακτηριστικό των ζώντων οργανισμών.

  Η μάθηση δεν λαμβάνει χώρα εκτός και αν ο οργανισμός μπορεί να διακρίνει χαρακτηριστικά από το περιβάλλον τα οποία έλκουν και χαρακτηριστικά τα οποία απωθούν, μπορεί να ξεχωρίσει το καλό από το κακό. Από αυτή την πλευρά η μάθηση θα μπορούσε να οριστεί ως αλλαγή που επιτρέπει στον οργανισμό να βελτιώσει τη μέση εκτίμηση του περιβάλλοντός του.

- Σύγκριση: Η θεωρία κοινωνικής σύγκρισης του Festinger περιέγραψε κάποιους από τους τρόπους που οι άνθρωποι χρησιμοποιούν ως πρότυπο μέτρησης του εαυτού τους και πως η σύγκριση με άλλους μπορεί να λειτουργήσει σαν είδος κινήτρου για μάθηση και αλλαγή. Με αντίστοιχο τρόπο και τα άτομα στο μοντέλο της προσαρμοζόμενης κουλτούρας συγκρίνουν τον εαυτό τους με τους γείτονες τους με βάση ένα κρίσιμο μέτρο και μιμούνται μόνο τους γείτονες αυτούς που είναι ανώτεροι από αυτούς. Τα πρότυπα για κοινωνική συμπεριφορά καθορίζονται από τη σύγκριση με άλλους.

- Μίμηση: Θεωρητικά η μίμηση συναντάται παντού στη φύση καθώς είναι ένας εξαιρετικά αποτελεσματικός τρόπος για να μάθουμε να κάνουμε πράγματα. Παρόλα αυτά όπως τόνισε ο Lorenz πολύ λίγα ζώα είναι ικανά για πραγματική μίμηση. Στην πραγματικότητα βεβαιώνει ότι μόνο οι άνθρωποι και κάποια πτηνά είναι ικανά. Κάποιες ελαφρές διαφοροποιήσεις κοινωνικής μάθησης συναντώνται μεταξύ άλλων ειδών αλλά κανένα εξ'αυτών δεν συγκρίνεται με τη δυνατότητά μας να μιμούμαστε ο ένας τον άλλο. Η ανθρώπινη μίμηση αποτελείται από την οπτική που δημιουργούμε για το άλλο άτομο. Δεν μιλάμε απλά για μίμηση συμπεριφοράς αλλά για διαπίστωση του σκοπού της και εκτέλεση συμπεριφοράς όταν κρίνεται απαραίτητο.

Οι τρεις αρχές της εκτίμησης, σύγκρισης και μίμησης μπορούν να συνδυαστούν ακόμα και σε απλουστευμένα κοινωνικά όντα σε προγράμματα υπολογιστών, επιτρέποντάς τους να προσαρμοστούν σε περίπλοκα περιβάλλοντα, λύνοντας εξαιρετικά δύσκολα προβλήματα.

Μοντέλο δυαδικής απόφασης

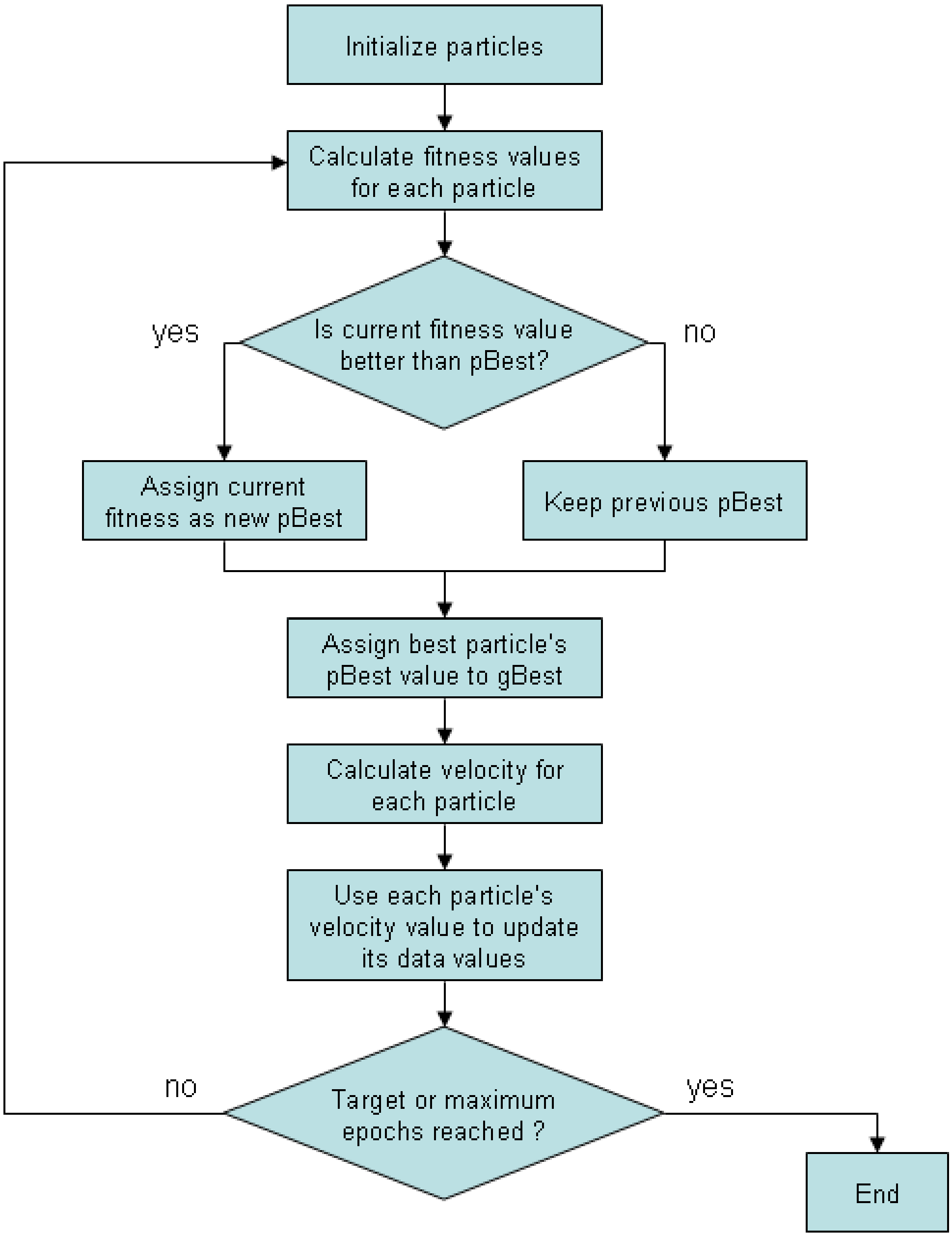
Σε αυτό το μοντέλο γίνεται η παραδοχή ότι ένα άτομο έχει μόνο ένα πράγμα στο μυαλό του και ένα σύνολο αποφάσεων που μπορεί να λάβει. Οι αποφάσεις που μπορεί να λάβει είναι: ΝΑΙ / ΟΧΙ ή ΑΛΗΘΕΣ / ΨΕΥΔΕΣ, δηλαδή δυαδικές αποφάσεις όπου είναι πολύ δύσκολο να διακριθεί ποιες αποφάσεις πρέπει να ληφθούν. Για κάθε απόφαση, αυτό το απλοϊκό άτομο μπορεί να είναι στη μία κατάσταση ή την άλλη, δηλαδή είτε σε κατάσταση ΝΑΙ που αναπαρίσταται με 1, είτε σε κατάσταση ΟΧΙ που θα αναπαρίσταται με 0. Περιτριγυρίζεται από άλλα άτομα με παρόμοια συλλογιστική τα οποία προσπαθούν να αποφασίσουν και να κάνουν τις καλύτερες επιλογές.

Δύο σημαντικά είδη πληροφορίας είναι διαθέσιμα σε αυτά τα άτομα του υποδείγματος που εξετάζεται. Το πρώτο είναι η ίδια τους η εμπειρία, δηλαδή έχουν κάνει επιλογές και ξέρουν ποια είναι ως τώρα η καλύτερη και έχουν εικόνα για το πόσο καλή ήταν. Το δέυτερο είδος είναι η γνώση του τρόπου με τον οποίο τα άλλα άτομα γύρω τους έχουν ανταπεξέλθει. Στην πραγματικότητα είναι τόσο απλά που το μόνο πράγμα που ξέρουν είναι ποιες επιλογές οι γείτονές τους έχουν βρει ως τώρα. Αυτοί οι δύο τύποι πληροφορίας αντιστοιχούν σε αυτό που οι Boyd και Richerson περιέγραψαν ως αυτόνομη μάθηση και μετάδοση κουλτούρας. Η πιθανότητα το άτομο να διαλέξει ΝΑΙ για κάθε μία από τις αποφάσεις που θα κληθεί να λάβει είναι σε συνάρτηση του πόσο επιτυχημένη είναι η επιλογή ΝΑΙ για αυτό σε σχέση με τα προηγούμενα ΟΧΙ. Η απόφαση επηρεάζεται επίσης από την κοινωνική επιρροή, αν και ο τρόπος με τον οποίο εμφανίζεται στους ανθρώπους δεν είναι ξεκάθαρος. Η θεωρία κοινωνικής αλληλεπίδρασης δηλώνει ότι οι δυαδικές αποφάσεις του ατόμου τείνουν να συμφωνούν με την άποψη της πλειοψηφίας, ισοζυγισμένης από την ισχύ και την εγγύτητα. Με άλλα λόγια τα άτομα τείνουν να επηρεάζονται από τη καλύτερη δυνατή απόφαση οποιουδήποτε ατόμου με τον οποίο συνδέονται.

Οι δύο πιο γνωστές κατηγορίες αλγόριθμων νοημοσύνης σμήνους είναι οι αλγόριθμοι Particle Swarm Optimization (PSO) και οι αλγόριθμοι Ant Colony Optimization (ACO). Και οι δύο στηρίζονται στο γεγονός ότι ένα πρόβλημα οδηγείται στη βελτιστοποίηση μέσα από την κοινωνική συμπεριφορά.

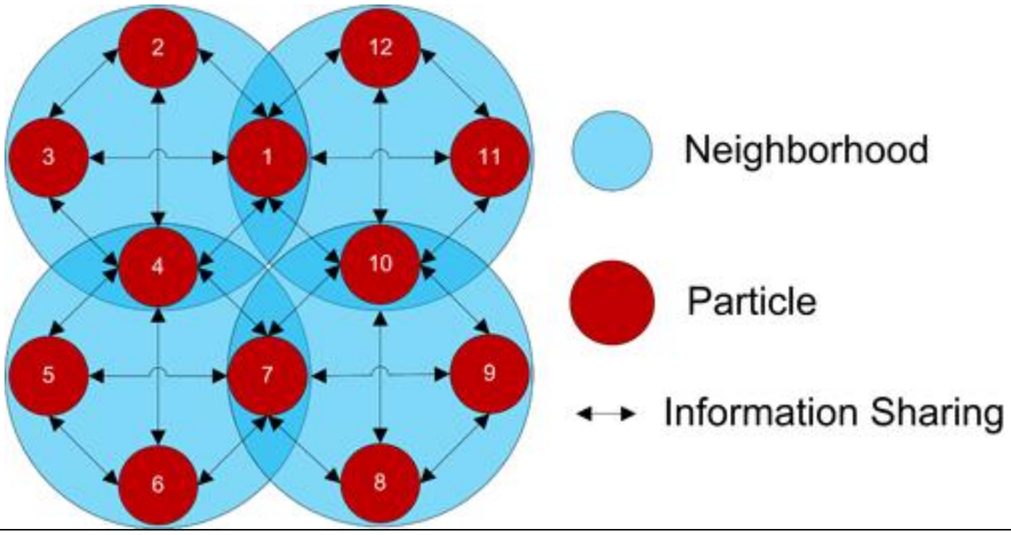
- Αλγόριθμοι Particle Swarm Optimization (PSO)

Ο αλγόριθμος PSO είναι ένας αλγόριθμος βελτιστοποίησης για την επίλυση προβλημάτων στα οποία η βέλτιστη λύση μπορεί να αναπαρασταθεί σε έναν πολυδιαστατικό χώρο. Η αρχική ιδέα για σμήνη από σωμάτια ήταν των Kennedy και Eberhart με τις πρώτες εξομοιώσεις να γίνονται το 1995. Ο PSO αποτελείται από ένα σύνολο σωματίων (παρότι τα μέλη του πληθυσμού δεν διαθέτουν μάζα και όγκο), τα οποία τοποθετούνται μέσα σε ένα χώρο που ορίζεται από μία συνάρτηση και κάθε ένα από αυτά εκτιμά την αντικειμενική συνάρτηση στην τρέχουσα θέση κάθε επανάληψης. Κάθε οντότητα στη συνέχεια καθορίζει την κίνησή του μέσα στο χώρο αναζήτησης των λύσεων συνδυάζοντας κάποιο μέρος του ιστορικού της παρούσας και της τοποθεσίας - στόχου με την βέλτιστη καταλληλότητα στη βάση κάποιων τυχαίων μεταθέσεων. Η επόμενη επανάληψη λαμβάνει χώρα αφού όλες οι οντότητες έχουν κινηθεί. Με βάση την παραπάνω διαδικασία το σμήνος μέσα από τη συλλογική εμπειρία θα καταλήξει σε βέλτιστο της συνάρτησης καταλληλότητας.



Εικόνα 2.3.3.4α: Διάγραμμα ροής του αλγορίθμου PSO

Τα σωμάτια σε κάθε στιγμή εκτέλεσης του αλγορίθμου χαρακτηρίζονται από 3 διανύσματα. Το ένα καθορίζει την ταχύτητα, δεύτερο την τρέχουσα θέση και το άλλο την προηγούμενη θέση. Κάθε σωματίδιο μπορεί να επηρεάζεται είτε από την εμπειρία όλου του σμήνους, είτε λαμβάνοντας υπόψη του την εμπειρία ενός υποσυνόλου το οποίο περιλαμβάνει όλα τα σωμάτια με τα οποία γειτνιάζει. Στην πρώτη περίπτωση όπου συνδέονται νοητά όλα τα μέλη του πληθυσμού τότε η προσέγγιση ονομάζεται g-best και έχει ως αποτέλεσμα την σύγκλιση του συνόλου των σωμάτιων προς τη λύση. Στη δεύτερη περίπτωση όπου έχουμε τη δημιουργία μιας γειτονιάς από ένα σωμάτιο και τους k πλησιέστερους γείτονες στον πληθυσμό, η προσέγγιση ονομάζεται l-best και έχει ως αποτέλεσμα τα σωμάτια να έλκονται τοπικά προς τη βέλτιστη λύση.

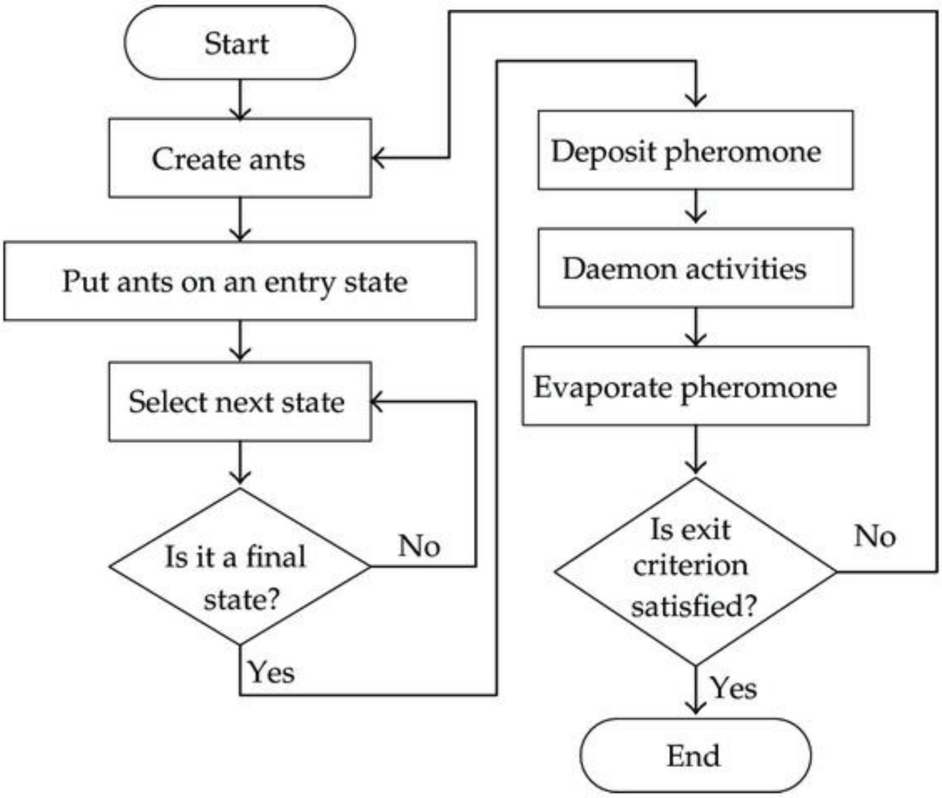


Εικόνα 2.3.3.4β: Αναπαράσταση l-best γειτονιάς

Το μεγαλύτερο πλεονέκτημα αυτού του αλγόριθμου είναι ότι ο μεγάλος αριθμός των μελών που απαρτίζουν το σμήνος σωματιδίων καθιστούν την τεχνική εντυπωσιακά ανθεκτική στο πρόβλημα των τοπικών ελαχίστων.

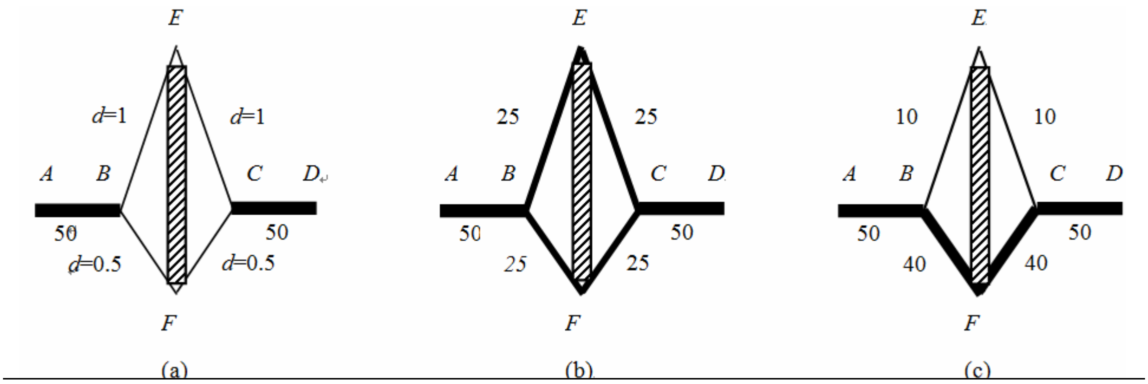
- <u>Αλγόριθμοι Ant Colony Optimization (ACO)</u>

Ο αλγόριθμος Ant Colony Optimization (ACO) είναι εμπνευσμένος από την συμπεριφορά που εμφανίζουν οι αποικίες των μυρμηγκιών σχετικά με την εύρεση τροφής. Τεχνητά μυρμήγκια - πράκτορες προσομοίωσης- εντοπίζουν βέλτιστες λύσεις, κινούμενα μέσα σε παραμετροποιημένο χώρο, ο οποίος περιλαμβάνει όλες τις πιθανές λύσεις. Ουσιαστικά ό αλγόριθμος αντιγράφει τη μεθοδολογία των μυρμηγκιών για να βρουν το βέλτιστο μονοπάτι ανάμεσα στη τροφή και την αποικία.



Εικόνα 2.3.3.4γ: Διάγραμμα ροής αλγορίθμου αποικίας μυρμηγκιών

Η βασική αρχή που στηρίζεται η συγκεκριμένη υλοποίηση έγκειται στο γεγονός ότι τα μυρμήγκια εκκρίνουν μία χημική ουσία, τη φερομόνη, στο έδαφος και με αυτόν τον τρόπο ουσιαστικά δείχνουν τον δρόμο στα υπόλοιπα μέλη της αποικίας. Τα υπόλοιπα μυρμήγκια έχουν την τάση να ακολουθούν τις διαδρομές με τη μεγαλύτερη ποσότητα φερομόνης. Στο παρακάτω σχήμα περιγράφεται η διαδικασία με την οποία τα μυρμήγκια εκμεταλλεύονται την έκκριση φερομόνης για τον καθορισμό της συντομότερης διαδρομής από τη φωλιά προς τη πηγή τροφής και πίσω.



Εικόνα 2.3.3.4δ: Παράδειγμα κίνησης μυρμηγκιών και επιλογής διαδρομής

Στο παραπάνω σχήμα το σημείο A είναι η φωλιά, το σημείο E είναι μία πηγή τροφής, το Σημείο B και το σημείο C είναι ένα διχαλωτό μονοπάτι, η γραμμή EF είναι ένα εμπόδιο και d είναι η απόσταση δύο κόμβων. Εξαιτίας του εμποδίου EF, τα μυρμήγκια που βγαίνουν από την φωλιά A ή επιστρέφουν από την πηγή τροφής D μπορούν να τα καταφέρουν μόνο μέσω τον κόμβων E ή F. Υποθέτοντας ότι όλα τα μυρμήγκια θα πάνε και θα γυρίσουν μεταξύ του A και D με ταχύτητα μίας μονάδας χρόνου και υπάρχουν 50 μυρμήγκια ανά μονάδα χρόνου από το A στο D στο D και A αντίστοιχα που θα πρέπει να αποφασίσουν που θα στρίψουν τυχαία. Σε αυτή τη περίπτωση 25 από τα 50 που ανήκουν στο κάθε τμήμα της διαδρομής θα πάνε προς τα αριστερά και 25 προς τα δεξιά όπως φαίνεται στο (b). Σε μία μονάδα χρόνου θα έχουν περάσει από το συντομότερο μονοπάτι BFC 50 μυρμήγκια καθώς το d=0,5 και από τη μεγαλύτερη διαδρομή BEC μόλις 25 μυρμήγκια. Ως εκ τούτου η ποσότητα της φερομόνης στο μονοπάτι BFC θα είναι διπλάσια από την ποσότητα στο BEC.

Τώρα ας πούμε ότι τώρα πάλι 50 μυρμήγκια ξεκινάνε από το σημείο D και φτάνουν στο C και 50 μυρμήγκια ξεκινούν από τη φωλία A και φτάνουν στο B. Άρα θα έχουμε 50 μυρμήγκια στον κόμβο C με κατεύθυνση στον A και 50 στον κόμβο B με κατεύθυνση στον κόμβο D. Τώρα όμως με δεδομένη την μεγαλύτερη ποσότητα φερομόνης τα περισσότερα μυρμήγκια από τον κάθε κόμβο θα πάνε από τη συντομότερη διαδρομή και μόλις 10 θα πάνε από τη μεγαλύτερη διαδρομή. Όσο κυλάει ο χρόνος και η διαδικασία επαναλαμβάνεται τόσο ποιο γρήγορα ανανεώνεται η φερομόνη στην σύντομη διαδρομή με αποτέλεσμα όλο και περισσότερα μυρμήγκια να επιλέγουν αυτή τη διαδρομή.

Προγραμματιστικά αυτή η διαδικασία επιτυγχάνεται με τη βοήθεια και ενός θετικού βρόγχου ανάδρασης όπου η πιθανότητα με την οποία ένα μυρμήγκι επιλέγει ένα μονοπάτι αυξάνεται με τον αριθμό των μυρμηγκιών που επέλεξαν πριν την ίδια διαδρομή. Στο τέλος όλα τα μυρμήγκια θα επιλέξουν την συντομότερη διαδρομή.

3. Αλγόριθμος αποικίας μυρμηγκιών για την επίλυση προβλημάτων δρομολόγησης

3.1 Ανάλυση του προβλήματος της μελέτης περίπτωσης

3.1.1 Περιγραφή του προβλήματος περιορισμένης χωρητικότητας δρομολόγησης οχημάτων (CVRP)

Το πρόβλημα προς επίλυση ανήκει στην κατηγορία των προβλημάτων δρομολόγησης οχημάτων με περιορισμένη χωρητικότητα (Capacitated VRP). Το ζητούμενο του προβλήματος είναι η εύρεση των διαδρομών του οχήματος, που ελαχιστοποιούν τον χρόνο παράδοσης, ώστε να προμηθευτούν οι πελάτες την ζητούμενή τους ποσότητα. Σύμφωνα με τα δεδομένα του προβλήματος, τα οχήματα είναι ομογενή και έχουν συγκεκριμένη χωρητικότητα, η οποία δεν καλύπτει τις απαιτήσεις όλων των πελατών σε μία ενιαία διαδρομή. Αν ένα όχημα κάλυπτε τις απαιτήσεις όλων των πελατών σε μια ενιαία διαδρομή, τότε θα γινόταν λόγος για απλοποίηση του προβλήματος σε "πρόβλημα πλανόδιου πωλητή (TSP)", το οποίο λύνεται σχετικά πιο εύκολα. Αντίθετα, γίνεται λόγος για VRP, που όπως περιγράφηκε στο κεφάλαιο 2 αποτελεί γενίκευση ενός TSP, και στόχος του είναι η εύρεση πολλαπλών διαδρομών από και προς την αποθήκη, που καθεμία τους είναι ένα πρόβλημα TSP. Η καθορισμένη χωρητικότητα των οχημάτων δεν πρέπει να υπερβαίνει τους 23 τόνους.

Εκτός από την καθορισμένη χωρητικότητα των οχημάτων που εκτελούν τις μεταφορές των προϊόντων στους πελάτες, γνωστά είναι ο αριθμός και οι απαιτήσεις των πελατών. Οι απαιτήσεις των πελατών εκφράζονται από το χαρακτηριστικό πως για κάθε διαδρομή δεν πρέπει οι παραγγελίες να υπερβαίνουν την χωρητικότητα των οχημάτων. Αντίστοιχα, από τα δεδομένα του προβλήματος προκύπτει οτι υπάρχουν 100 πελάτες προς εξυπηρέτηση που χαρακτηρίζονται από την παρακάτω εξίσωση:

$$y = y1 + y2 + y3 + \cdots + y100$$

Εκτός από την καθορισμένη χωρητικότητα του φορτηγού, τον αριθμό και τις γνωστές απαιτήσεις των πελατών, σημειώνεται οτι στα δεδομένα του προβλήματος εισέρχονται τα βασικά χαρακτηριστικά ενός VRP προβλήματος που είναι τα εξής:

- κάθε πελάτη τον επισκέπτεται ακριβώς ένα όχημα και μόνο μια φορά,
- όλα τα οχήματα ακολουθούν διαδρομές που αρχίζουν και ολοκληρώνονται στην αποθήκη, και
- ο αριθμός των φορτηγών που εξυπηρετούν τους πελάτες θα πρέπει να είναι ο ελάχιστος δυνατός.

Το τελευταίο δεδομένο του προβλήματος είναι μια τιμή η οποία αντιπροσωπεύει την μέγιστη χρονικά ικανότητα του οχήματος για συνεχή παράδοση σε πελάτες, χωρίς να χρειάζεται η επιστροφή του στην αποθήκη. Οι φυσικοί περιορισμοί, τους οποίους μπορεί να αντιπροσωπεύει, είναι η χωρητικότητα του οχήματος σε καύσιμα, η μέγιστη επιτρεπόμενη συνεχής εργασία του οδηγού κ.ά. Στα πλαίσια της μελέτης περίπτωσης που εξετάζεται γίνεται η υπόθεση οτι αυτή η τιμή αντιστοιχεί στον περιορισμό του οχήματος σε καύσιμα. Τα καύσιμα κάθε οχήματος είναι αρκετά για 9 ώρες χωρίς την επιστροφή του στην αποθήκη.

Αφού έχουν παρουσιαστεί ολα τα δεδομένα του προβλήματος, πρέπει να δομηθεί και η κατάλληλη αντικειμενική συνάρτηση. Η αντικειμενική συνάρτηση είναι άμεσα εξαρτώμενη από την ευκλείδεια απόσταση όλων των κόμβων μεταξύ τους. Πιο συγκεκριμένα, γίνεται η υπόθεση ότι ο χρόνος μετακίνησης του οχήματος μεταξύ δυο σημείων είναι ανάλογος της ευκλείδειας απόστασης, και πλέον χρησιμοποιούμε αυτήν την απόσταση στην αντικειμενική συνάρτηση του προβλήματος.

Όμως, στην αντικειμενική συνάρτηση θα έπρεπε να ληφθεί υπόψη και μια ακόμα παράμετρος. Η παράμετρος αυτή έχει τη βάση της στον άξονα του χρόνου και περιγράφει το χρονικό διάστημα που χρειάζεται ένα φορτηγό για να κάνει την εκφόρτωση των προϊόντων του σε έναν πελάτη. Στα πλαίσια της παρούσας μελέτης γίνεται η υπόθεση οτι αυτός ο χρόνος εκφόρτωσης είναι ίδιος και προκαθορισμένος για όλους τους πελάτες και γι'αυτό δεν εισέρχεται με τη μορφή μεταβλητής στην αντικειμενική συνάρτηση. Ωστόσο, στο στάδιο της υλοποίησης του αλγορίθμου, εκχωρείται μια εξωτερική μεταβλητή που δέχεται τιμές για αυτό το χρονικό διάστημα, γιατί οι συγκρίσεις των διαδρομών από τον αλγόριθμο γίνονται ανά κυκλική διαδρομή (μια διαδρομή από και προς την αποθήκη) και κάθε κυκλική διαδρομή αποτελείται από διαφορετικό αριθμό κόμβων(πελατών).

3.1.2 Μαθηματική μοντελοποίηση του CVRP

Το πρόβλημα μπορεί να αναπαρασταθεί σαν ένα γράφημα $G \equiv (V, A)$, όπου:

- $V = \{0,1, ..., 101\}$ είναι το σύνολο των κόμβων - πελατών, ενώ με το 0 συμβολίζεται η κεντρική εγκατάσταση (αποθήκη)
- $A = \{(i,j): i,j \in N, i \neq j\}$ είναι το σύνολο των ακμών που συνδέουν τους κόμβους μεταξύ τους.

Σε κάθε ακμή $(i,j) \in N$ αντιστοιχεί ένα μη αρνητικό κόστος διάσχισης d_{ij} από τον κόμβο i στον κόμβο j. Αυτό το κόστος αφορά τον αριθμό των χιλιομέτρων που απέχουν οι πελάτες μεταξύ τους. Όπως έχει προαναφερθεί, κάθε φορτηγό έχει χωρητικότητα Q_k = 23 τόνους. Η χωρητικότητα είναι σταθερή και τα οχήματα είναι ομογενή. Συγκεντρωτικά, τα δεδομένα είναι μαζεμένα ως εξής:

- V = {0,1, ..., 101}
- A = {(i,j):i,j ∈ N, i ≠ j}
- q_i συμβολίζεται η ζήτηση των προϊόντων για κάθε πελάτη
- d_{ij} συμβολίζεται το μη αρνητικό κόστος μεταξύ δυο πελατών
- K = {K_1, K_2, ..., K_m} συμβολίζεται το πλήθος των φορτηγών
- Q_k = 23 τόνους συμβολίζεται η χωρητικότητα κάθε φορτηγού

Σε αυτό το σημείο, και πριν την παρουσίαση της αντικειμενικής συνάρτησης και των περιορισμών του προβλήματος, πρέπει να σημειωθεί οτι για την εύρεση της σειράς με την οποία τα οχήματα θα επισκεφθούν τους πελάτες δηλώνονται δυο μεταβλητές απόφασης.

$$x_{ij}^k = \begin{cases} 1 \text{ , αν το φορτηγό επισκέπτεται τον πελάτη j μετά τον i} \\ \qquad 0 \text{ , για κάθε άλλη περίπτωση} \end{cases}$$

$$y_i^k = \begin{cases} 1 \text{ , αν ο πελάτης i εξυπηρετείται από το φορτηγό k} \\ \qquad 0 \text{ , αν δεν εξυπηρετείται} \end{cases}$$

Η αντικειμενική συνάρτηση του προβλήματος είναι της μορφής:

$$\min \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij}^k \qquad (1)$$

με τους εξής περιορισμούς:

$$\sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1, \forall\, i \in V \qquad (2)$$

$$\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{ij}^k = 0, \forall\, i \in V, k \in K \qquad (3)$$

$$\sum_{j \in Vv} x_{0,j}^k = 1, \forall\, k \in K \qquad (4)$$

$$x_{ij}^k = 1 \therefore y_i - q_i = y_j, \forall\, i,j \in V, k \in K \qquad (5)$$

$$y_0 = Q, \qquad 0 \leq y_i, \forall\, i \in V \qquad (6)$$

$$x_{ij}^{k} \in \{0,1\} \ \forall \ i,j \in V, k \in K \qquad (7)$$

$$Αριθμός\ οχημάτων\ (X) = \sum_{k\in K}\sum_{i\in V} x_{0,j}^{k} \qquad (8)$$

Όπως έχει ήδη αναφερθεί, οι περισσότεροι από τους περιορισμούς αποτελούν χαρακτηριστικά της δομής ενός προβλήματος δρομολόγησης οχημάτων. Ο τρόπος που επιλέγεται για την επίλυσή του είναι με τον αλγόριθμο αποικίας μυρμηγκιών.

3.2 Μια προσέγγιση επίλυσης του CVRP με τον αλγόριθμο αποικίας μυρμηγκιών (ant colony algorithm)

Σύμφωνα με τα όσα έχουν περιγραφεί στο κεφάλαιο 2 με τις τεχνικές επίλυσης των προβλημάτων δρομολόγησης οχημάτων με περιορισμένη χωρητικότητα, επιλέγεται σε αυτή τη μελέτη περίπτωσης να χρησιμοποιηθεί μια μελέτη περίπτωσης με βασικά χαρακτηριστικά στον αλγόριθμο αποικίας μυρμηγκιών. Προηγούμενη βιβλιογραφία έχει υπάρξει για παρόμοιες επιλύσεις στο βιβλίο του κ. Φούντα "Μαθηματικά Αποφάσεων". Στην παρούσα μελέτη, ωστόσο, ο αλγόριθμος γίνεται υβριδικός προσθέτοντας τη λογική των αλγορίθμων "αποταμίευσης (savings)" και τοπικής αναζήτησης. Ο αλγόριθμος της "αποταμίευσης" χρησιμοποιείται ως μια φρέσκια προσέγγιση, προκειμένου να βοηθήσει στην αναζήτηση και τον υπολογισμό μεταξύ δυο πελατών για το αν μπορούν να συνδυαστούν σε μια ενιαία διαδρομή ή πρέπει να χωριστούν σε δυο ξεχωριστές. Αντίστοιχα, το κομμάτι του αλγορίθμου τοπικής αναζήτησης δίνει μια διαφορετική προσέγγιση στον τρόπο που γίνεται η τελική ανάπτξη των διαδρομών, λίγο πριν γίνει η διαδικασία ανανέωσης των φερομόνων. Έτσι ο τελικός αλγόριθμος που χρησιμοποιείται είναι πιο ενισχυμένος από τον απλό αλγόριθμο αποικίας μυρμηγκιών. Τέλος, η προσέγγιση αυτή κάνει αναφορά σε χαρακτηριστικά και θεωρίες που είχαν δημοσιευτεί το 1997 από τους Dorigo & Gambardella.

3.2.1 Η ευρεστική μέθοδος της αποταμίευσης (savings heuristic)

Όπως έχει ήδη αναφερθεί από το κεφάλαιο 2, οι πρώτοι που εισήγαγαν τη θεωρία της μεθόδου των αποταμιεύσεων ήταν οι Clark & Wright το 1964 και η τελευταία ανανέωση των θεωριών έγινε το 1988 στη μελέτη του Paessens "The savings algorithm for the vehicle routing problem". Ο αλγόριθμος και οι αρχές του ήταν τόσο καινοτόμα που όλα τα λογισμικά που έχουν ως στόχο την επίλυση των προβλημάτων δρομολόγησης οχημάτων, ακόμα και σήμερα, τα χρησιμοποιούν ως βάση για την κατασκευή τους. Ο στόχος αυτού του αλγορίθμου και ο λόγος δημιουργίας του βασίζεται στην ερώτηση του αν συμφέρει περισσότερο να βρίσκονται ανα δυο οι πελάτες στην ίδια διαδρομή, ή να βρίσκονται σε διαφορετικές διαδρομές.

Αυτό που μελετάται είναι η τιμή της μεταβλητής γij , η οποία υπολογίζεται από τον τύπο:

$$\gamma_{ij} = d_{i0} + d_{0j} - g.d_{ij} + f.\left|d_{i0} - d_{0j}\right|$$

,όπου g,f είναι δυο ευρεστικές παράμετροι που υπολογίζονται από τον αλγόριθμο,

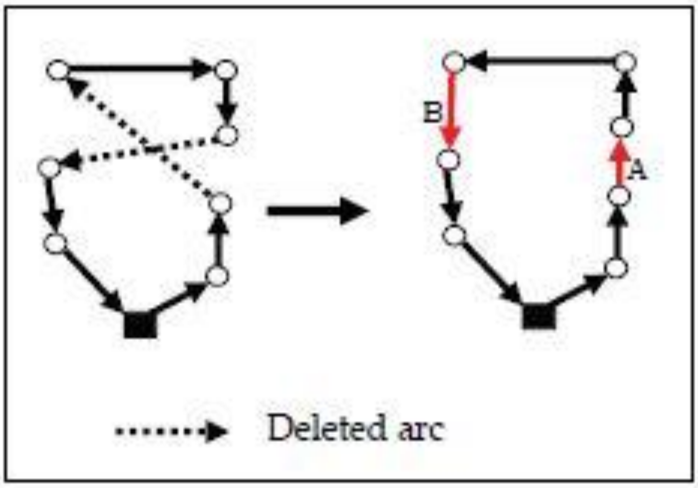
d_{ij} είναι η απόσταση μεταξύ των κόμβων i και j

και

d_{i0}, d_{0j} είναι οι αποστάσεις κάθε κόμβου από την αποθήκη.

3.2.2 Η ευρεστική μέθοδος του 2-opt (2-opt heuristic)

Οι αρχές της μεθόδου 2-opt εφαρμόζονται άμεσα ανάμεσα σε δυο κόμβους της ίδιας διαδρομής, που όμως η απόσταση μεταξύ τους δεν είναι η μέγιστη. Αυτό καμιά φορά μπορεί να συμβεί με τον αλγόριθμο αποικίας μυρμηγκιών. Έτσι, αυτή η μέθοδος διαγράφει τους δυο αυτούς κόμβους και τους αντικαθιστά με δυο εναλλακτικούς, οι οποίοι θα βελτιώνουν το συνολικό άθροισμα των ελάχιστων χιλιομέτρων. Για παράδειγμα, όπως παρουσιάζεται στην εικόνα παρακάτω, οι δυο διαδρομές που δημιουργούν ανωμαλία στο γράφημα της επίλυσης διαγράφτηκαν και αντικαταστάθηκαν από δυο άλλους που ομαλοποιούν την κυκλοφορία των οχημάτων και το συνολικό τους άθροισμα είναι κατα πολύ μικρότερο. Σε μερικούς κόμβους που η 2-opt μέθοδος δεν μπορεί να λάβει υπόψη την κατεύθυνση του δικτύου, ο χρήστης μπορεί να μεταβάλλει την κατεύθυνση του γραφήματος δημιουργώντας νέα διαδρομή.



Εικόνα 3.2.2: Χρησιμοποίηση της μεθόδου 2-opt για την βελτιστοποίηση της διαδρομής

3.2.3 Περιγραφή του αλγορίθμου για την επίλυση του προβλήματος

Στους αλγορίθμους αποικίας μυρμηγκιών κάθε μυρμήγκι αντιστοιχεί σε ένα όχημα. Έχει μεγάλη αξία να περιγραφεί ο τρόπος με τον οποίο το μυρμήγκι επιλέγει τον επόμενο κόμβο. Αυτή η επιλογή συνδέεται με την ποσότητα της φερομόνης που εκλύεται από το μυρμήγκι κι εναποτίθεται σε κάθε διαδρομή που περνάει το μυρμήγκι και φανερώνει το κόστος της διαδρομής. Στην επίλυση αυτής της μελέτης περίπτωσης που χρησιμοποιείται αυτός ο υβριδικός αλγόριθμος, για την κατασκευή της διαδρομής ερευνώνται και τα κριτήρια του αλγορίθμου "αποταμιεύσεων".

Αρχικά στον αλγόριθμο αποικίας μυρμηγκιών, m αριθμός από μυρμήγκια τοποθετείται σε όλους τους κόμβους του γραφήματος και σε κάθε ακμή μευαξύ των κόμβων τοποθετείται μια ποσότητα φερομόνης. Σύμφωνα με τις υποθέσεις του αλγορίθμου αποικίας μυρμηγκιών θα πρέπει να ισχύουν τα εξής:

- Κάθε μυρμήγκι θα ξεκινάει από τον κόμβο – αποθήκη (0) και θα μπορεί να επιστρέφει στην αποθήκη όσες φορές επιθυμεί ή είναι απαραίτητο,
- Κάθε μυρμήγκι πρέπει να επισκεφθεί ακριβώς μια φορά όλους τους πελάτες που του αντιστοιχούν, και
- Κάθε πελάτης πρέπει να έχει ικανοποιημένες τις απαιτήσεις του από ακριβώς ένα μυρμήγκι.

Στην περίπτωση που ένα μυρμήγκι επιστρέφει στην κεντρική αποθήκη πρέπει να ξεκινήσει όλη τη διαδικασία από την αρχή. Δηλαδή, να φορτώσει τόσο εμπόρευμα, όσο του επιτρέπει η περιορισμένη χωρητικότητά του και να σχεδιάσει από την αρχή τη νέα διαδρομή που θα ακολουθήσει. Αυτή η διαδικασία επαναλαμβάνεται μέχρις ότου να εξαντληθεί ο αριθμός των πελατών προς εξυπηρέτηση. Μόλις εξαντληθεί ο αριθμός των προς εξυπηρέτηση πελατών, τότε θα έχει βρεθεί η βέλτιστη λύση του προβλήματος δρομολόγησης οχημάτων που ερευνάται.

Κατά τη διαδικασία δημιουργίας της βέλτιστης διαδρομής, το μυρμήγκι μεταβάλλει την ποσότητα της φερομόνης στις διαδρομές που περνάει χρησιμοποιώντας έναν παράγοντα ανανέωσης της φερομόνης τοπικά. Στο τέλος του αλγορίθμου, όταν θα έχει βρεθεί η βέλτιστη λύση, η βέλτιστη αυτή διαδρομή θα έχει την περισσότερη ποσότητα φερομόνης από κάθε άλλη πιθανή διαδρομή του δικτύου, η οποία θα υπολογίζεται με βάση έναν “global” παράγοντα ανανέωσης.

Οι παράγοντες υπολογισμού και αναγνώρισης της ποσότητας της φερομόνης που τοποθετείται και ανανεώνεται τοπικά και ολικά στο δίκτυο διαδρομών, υπολογίζεται στις παρακάτω εξισώσεις. Γίνεται η υπόθεση οτι k μυρμήγκια που έχουν τοποθετηθεί σε έναν κόμβο i, επιλέγουν να επισκεφθούν τον επόμενο κόμβο της διαδρομής j, τον οποίο επιλέγουν μέσα από τον πιθανοτικό κανόνα P_k (i,j) που δίνεται από τους παρακάτω υπολογισμούς.

$$j = \begin{cases} \arg\max\limits_{u \in F_k(i)} \left\{ (\tau_{iu})^{\alpha} . (\eta_{iu})^{\beta} . (\gamma_{iu})^{\lambda} \right\} & \text{if } q \le q_0 \\ J & \text{if } q \succ q_0 \end{cases}$$

Το J της περιπτώσεως $q>q_0$ υπολογίζεται από την εξίσωση της διανομής που παρουσιάζεται παρακάτω:

$$p_{ij} = \begin{cases} \dfrac{(\tau_{ij})^{\alpha} . (\eta_{ij})^{\beta} . \ (\gamma_{ij})^{\lambda}}{\sum\limits_{u \in F_k(i)} (\tau_{iu})^{\alpha} . (\eta_{iu})^{\beta} . \ (\gamma_{iu})^{\lambda}} & \text{if } u \in F_k(i) \\ 0 & \text{otherwise} \end{cases}$$

Για τις παραπάνω εξισώσεις ισχύει οτι:

- Το F_k(i) περιγράφει τη λίστα των κόμβων – πελατών που το μυρμήγκι k που βρίσκεται στον κόμβο i δεν έχει επισκεφθεί ακόμα,
- το q είναι μια τυχαία μεταβλητή που ακολουθεί μια ομοιόμορφη κατανομή [0,1],
- το q_0 είναι μια παράμετρος μεταξύ 0 και 1 ($0 \le q_0 \le 1$) που δηλώνει την σχετικότητα μεταξύ της εκμετάλλευσης και της εξερεύνησης των κόμβων. Προτού κάποιο μυρμήγκι επισκεφθεί τον επόμενό του κόμβο – πελάτη, η μεταβλητή q παίρνει μια τυχαία τιμή. Εάν η τιμή της μεταβλητής q είναι

μιρκότερη της μεταβλητής q_0, τότε ενισχύεται στο μυαλό του μυρμηγκιού η ιδέα της χρησιμοποιήσης του συγκεκριμένου κόμβου αλλιώς αν η τιμή q είναι μεγαλύτερη της μεταβλητής q_0 τότε η αναζήτηση του απόμενου κόμβου – πελάτη συνεχίζεται,

- η μεταβλητή τij περιγράφει την ποσότητα της φερομόνης που υπάρχει στην ακμή που συνδέει τους κόμβους i,j μεταξύ τους,
- η μεταβλητή ηij περιγράφει την ευρεστική μέθοδο της ορατότητας του μυρμηγκιού που είναι το αντίστροφο της απόστασης μεταξύ των κόμβων i,j, και
- η εξίσωση $\gamma_{ij} = d_{i0} + d_{0j} - g.d_{ij} + f.\left|d_{i0} - d_{0j}\right|$ περιγράφει την ευρεστική μέθοδο των "αποταμιεύσεων" , όπου f και g αποτελούνδυο παραμέτρους.
- Τέλος, τα α,β,λ είναι τρεις παράμετροι που δηλώνουν την σχετική σημαντικότητα της φερομόνης, της απόστασης και της μεθόδου "αποταμίευσης" αντίστοιχα.

Ένα ακόμα στοιχείο που εισάγεται ως καινοτομία στον υβριδικό αλγόριθμο που περιγράφεται αποτελεί η επιλογή της μεθόδου προσομοιωμένης ανόπτησης (simulated annealing) κατά τον υπολογισμό της ανανεωσης της φερομόνης. Κατα τη διάρκεια που ένα μυρμήγκι διαμορφώνει τη βέλτιστη λύση που θα ακολουθεί, τα επίπεδα φερομόνης σε κάθε ακμή που έχει ήδη επιλεγεί ανανεώνονται τοπικά με βάση τον παρακάτω κανόνα:

$$\tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old} + \rho\Delta\tau_{ij}$$

, όπου η μεταβλητή ρ εκφράζει τον παράγοντα εξάτμισης της φερομόνης και παίρνει τιμές μεταξύ 0 και 1 και η μεταβλητή Δτij = τ0 εκφράζει την ποσότητα της αρχικής φερομόνης που εκλύεται από το μυρμήγκι.

Όταν το μυρμήγκι έχει καταλήξει στις διαδρομές που θα ακολουθήσει, αυτόματα ενεργοποιείται η χρήση της μεθόδου 2-opt για την βελτίωση αυτών των διαδρομών. Αυτή η ευρεστική μέθοδος είναι ρυθμισμένη να λειτουργεί με την ολοκλήρωση και της τελευταίας επανάληψης, προτού εφαρμοστεί ο ολικός κανόνας ανανέωσης της φερομόνης των διαδρομών.

Στον αλγόριθμο αποικίας μυρμηγκιών, στα μυρμήγκια επιτρέπεται να εναποθέσουν μια ποσότητα φερομόνης σε μια διαδρομή μικρού μήκους, προκειμένου να μπορέσουν όταν θα την αναζητήσουν να την προσθέσουν στην βέλτιστη διαδρομή που θα ακολουθούν. Αντίθετα, στον υβριδικό αλγόριθμο αποικίας μυρμηγκιών που εξετάζεται, κάθε μυρμήγκι καταλήγει σε μια διαδρομή S' στην οποία για να εκλυθεί φερομόνη και να χαρακτηριστεί ως βέλτιστη θα πρέπει να ισχύουν οι παρακάτω περιορισμοί (El Hassani, L. Bouhafs και A. Koukam):

$$\Delta S (= Cost(S') - Cost(S)) < 0$$

, όπου με S δηλώνεται η τρέχουσα βέλτιστη διαδρομή στο δίκτυο, με Cost(S) περιγράφεται το κόστος της τρέχουσας διαδρομής S και με Cost(S') περιγράφεται το κόστος της συνολικής διαδρομής αν προστεθεί και το κόστος της εξεταζόμενης διαδρομής. Παράλληλα με τον ανωτέρω κανόνα ερευνάται και:

$$\exp(-\Delta S / T) \succ \mu$$

, όπου με μ δηλώνεται μια τυχαία μεταβλητή που ακολουθεί μια ομοιόμορφη κατανομή [0,1], με T δηλώνεται μια παράμετρος που παρακολουθεί τη θερμοκρασία κατά τη διάρκεια της προσομειωμένης ανόπτησης και με exp(-ΔS/T) περιγράφεται το κριτήριο της προσομειωμένης ανόπτησης. Με την χρησιμοποιήση όλων των ανωτέρω κριτηρίων, γίνεται σαφές οτι δεν αφήνεται κανένα περιθώριο στο σύστημα να επιλέξει μια διαδρομή που δεν θα βελτιστοποιεί στον μέγιστο δυνατό βαθμό την διαδρομή.

Τέλος, εάν ένα μυρμήγκι έχει καταφέρει και έχει μια αποδεκτή βέλτιστη διαδρομή, τότε κατά τη διάρκεια διάσχισης της βέλτιστης αυτής διαδρομής, το μυρμήγκι εκλύει ποσότητα φερομόνης αυξημένη κατά 1/L*, όπου L* είναι το πλήθος των ακμών που αποτελούν τη βέλτιστη διαδρομή.

3.2.4 Λειτουργία του υβριδικού αλγορίθμου αποικίας μυρμηγκιών

Η λειτουργία του βασικού αλγορίθμου αποικίας μυρμηγκιών, όπως μελετάται στον τομέα της Νοημοσύνης Σμήνους, χωρίζεται σε τρεις φάσεις:

- Τη φάση της δημιουργίας διαδρομής, που σε αυτή τη φάση εντοπίζονται όλες οι δυνατές μεταβάσεις από τον κόμβο - αποθήκη σε κάποιον κόμβο - προορισμό,
- Τη φάση συντήρησης της διαδρομής, που σε αυτή τη φάση ισχυροποείται η διαδρομή που επιλέχθηκε και που συνδέει τους κόμβους μεταξύ τους, και
- Τη φάση χειρισμού της αποτυχίας μιας διαδρομής, που αναλαμβάνει να παράγει από την αρχή νέες διαδρομές, στην περίπτωση που κάποια διαδρομή από την πηγή στον προορισμό καταστραφεί ή τα μυρμήγκια δεν μπορούν πια να περάσουν από εκεί.

Η φάση της ανακάλυψης μιας διαδρομής είναι υπεύθυνη για την παραγωγή όλων των δυνατών διαδρομών μεταξύ του κόμβου - αφετηρίας και του κόμβου - προορισμού. Χρησιμοποιεί πακέτα ελέγχου που ονομάζονται

Forward Ant (FA) και Backward Ant (BA) για την ανακάλυψη μιας διαδρομής. Ένα FA εγκαθιστά στη διαδρομή το κομμάτι φερομόνης προς τον κόμβο - πηγή και ένα BA εγκαθιστά στη διαδρομή το κομμάτι φερομόνης προς τον προορισμό. Ένα FA εκκινάει το ταξίδι του από τον αποστολέα και περνάει από τους ενδιάμεσους κόμβους μέχρι να φτάσει στον προορισμό. Ένας κόμβος που λαμβάνει ένα FA για πρώτη φορά δημιουργεί μια εγγραφή στον πίνακα δρομολόγησης του. Η εγγραφή του στο μητρώο περιέχει τη διεύθυνση προορισμού, το επόμενο hop και την αξία φερομόνης. Ο κόμβος ερμηνεύει τη διεύθυνση αφετηρίας του FA ως διεύθυνση προορισμού, τη διεύθυνση του προηγούμενου κόμβου ως το επόμενο hop και υπολογίζει την τιμή φερομόνης, ανάλογα με τον αριθμό των hops που χρειάζεται το FA για να φτάσει στον τρέχον κόμβο. Στη συνέχεια, ο κόμβος προωθεί το FA στους γείτονές του. Τα πακέτα ενός FA έχουν έναν μοναδικό αύξοντα αριθμό σειράς. Οποιοδήποτε διπλότυπο FA ανιχνεύεται μέσω του αύξοντα αριθμού του. Μόλις ανιχνευθούν οι διπλές εγγραφές μυρμηγκιών στον κόμβο, αναλαμβάνει ο κόμβος να τα διώξει. Όταν το FA φτάσει στον προορισμό, εξάγονται οι πληροφορίες του και καταστρέφεται. Αυτόματα δημιουργείται ένα BA με τον ίδιο αύξοντα αριθμό και αποστέλλεται προς την πηγή. Το BA διατηρεί τους πόρους κατά μήκος των κόμβων στην πορεία προς την πηγή. Το BA εγκαθιστά μια διαδρομή προς τον κόμβο προορισμού. Μόλις η πηγή λαμβάνει το BA από τον κόμβο προορισμού, δημιουργείται η διαδρομή και τα δεδομένα μπορούν να αποσταλούν κατά μήκος της διαδρομής.

Ο ψευδοκώδικας της φάσης δημιουργίας διαδρομής παρατίθεται παρακάτω:

<u>Βήμα 1</u>: Ο κόμβος - πηγή αναμένει για κάποιο Backward Ant που μπορεί να καταφθάνει. Άν δεν ληφθεί κάποιο BA και λήξει μια συγκεκριμένη χρονική περίοδος (timeout period), τότε παράγεται ένα FA με έναν νέο αύξον αριθμό σειράς και αποστέλλεται στους γειτονικούς κόμβους. Στην περίπτωση, όμως, που παραληφθεί κάποιο BA, σημαίνει οτι η διαδρομή έχει ολοκληρωθεί και στέλνονται τα πακέτα αμέσως.

<u>Βήμα 2</u>: Σε οποιονδήποτε κόμβο, όταν λαμβάνεται ένα FA ισχύει ο παρακάτω κώδικας:

Εάν (current node = destination node) {

- Ρύθμισε τον τύπο των πακέτων ελέγχου του BA με τον ίδιο αριθμό σειράς με το FA
- Αποθήκευσε τους πόρους στον τρέχοντα κόμβο
- Στείλε το BA στον κόμβο από τον οποίο ελήφθη το FA

}αλλιώς {

- hop count = hop count +1
- $\varphi_{i,j} = \varphi_{i,j} + \Delta$ //ανανέωση τιμών φερομόνης
- Αποστολή του FA στον επόμενο πλησιέστερο κόμβο της λίστας

}

Βήμα 3: Σε οποιονδήποτε κόμβο, όταν λαμβάνεται ένα BA ισχύει ο παρακάτω κώδικας:

Εάν (currernt node is not source node){

- Αποθήκευσε τους πόρους στον τρέχοντα κόμβο
- Αποστολή του BA στον κόμβο από τον οποίο ελήφθη το FA

}αλλιώς{ //δηλαδή εάν το BA φτάσει στον κόμβο πηγη

- Λήψη όλων των τιμών της φερομόνης από όλες τις διαδρομές στη λίστα γειτόνων
- Υπολογισμός των πιθανοτήτων όλων των κόμβων στην λίστα γειτόνων
- Αποστολή πακέτων μέσω των διαδρομών με την μεγαλύτερη τιμή πιθανότητας

}

Η φάση της συντήρησης της διαδρομής είναι υπεύθυνη για τη διατήρηση της διαδρομής που παράγεται κατά τη διαδικασία της φάσης δημιουργίας που προηγήθηκε. Η φάση αυτή βοηθά ουσιαστικά στη διατήρηση της διαδρομής που έχει ήδη καθοριστεί κατά τη φάση της δημιουργίας. Δεδομένου ότι η τοπολογία των κόμβων στο δίκτυο αλλάζει, θα πρέπει να ανανεώνεται η διαδρομή μεταξύ των κόμβων. Μόλις η διαδρομή μεταξύ πηγής και προορισμού έχει συσταθεί, εναπόκειται στα πακέτα δεδομένων να διατηρήσουν τη διαδρομή ζωντανή και ενεργοποιημένη. Όταν ένας κόμβος Vi προωθεί το πακέτο δεδομένων στον κόμβο Vj, για να φτάσει στον προορισμό Vd, αυτό αυξάνει την τιμή φερομόνης κατά μήκος της διαδρομής Vj και Vd με μία τιμή Δ, ενισχύοντας έτσι τη διαδρομή. Μια επιβεβαίωση αποστέλλεται σε όλα τα πακέτα που λαμβάνονται. Εάν η επιβεβαίωση δεν έχει ληφθεί μέσα στο χρονικό όριο τότε μεταδίδεται το μήνυμα σφάλματος της διαδρομής στον προηγούμενο κόμβο.

Σε συνέχεια του ψευδοκώδικα της φάσης δημιουργίας διαδρομής παρατίθεται παρακάτω ο ψευδοκώδικας της φάσης συντήρησης της διαδρομής από τα πακέτα:

Βήμα 1: Σε οποιονδήποτε κόμβο, όταν λαμβάνεται ένα πακέτο δεδομένων εκτελούνται αυτόματα οι παρακάτω διαδικασίες:

Εάν (current node = destination node) {

- Γίνεται εξαγωγή των δεδομένων
- Αποστέλλεται πακέτο επιβεβαίωσης στον προηγούμενο κόμβο

} else {

- Ανάκτηση όλων των τιμών των φερομόνων από όλες τις διαδρομές χρησιμοποιώντας τη λίστα γειτνίασης
- Υπολογισμός πιθανοτήτων για όλους τους κόμβους στη λίστα

- Αποστολή πακέτου στη διαδρομή που έχει τη μεγαλύτερη πιθανότητα
- Αύξηση κατά ένα απίπεδο των επιπέδων φερομόνης της διαδρομής με τη μεγαλύτερη πιθανότητα.

}

Βήμα 2: Σε κάθε επανάληψη του αλγορίθμου για πέρασμα στον επόμενο κόμβο, ελαττώνεται η τιμή της φερομόνης στις διαδρομές κατά μια σταθερά a. Εάν το επίπεδο της φερομόνης για οποιαδήποτε διαδρομή γίνει 0, τότε το σύστημα επιστρέφει υποχρεωτικά στη φάση της δημιουργίας διαδρομής.

Βήμα 3: Εάν το πακέτο επιβεβαίωσης δεν έχει ληφθεί από τον τρέχον κόμβο πρίν λήξει η χρονική περίοδος, αποστέλλεται μήνυμα σφάλματος της διαδρομής στον προηγούμενο κόμβο.

Βήμα 4: Ανανέωση της διαδρομής μετά το πέρας του χρονικού ορίου.

Η τρίτη και τελευταία φάση του αλγορίθμου αποικίας μυρμηγκιών είναι υπεύθυνη για τη δημιουργία εναλλακτικών διαδρομών σε περίπτωση που η υφιστάμενη οδός αποτύχει. Η κινητικότητα των κόμβων στο ad hoc δίκτυο μπορεί να προκαλέσει ορισμένες διαδρομές να αποτύχουν. Κάθε πακέτο σχετίζεται με ένα πακέτο επιβεβαίωσης. Ως εκ τούτου, εάν ένας κόμβος δεν λάβει επιβεβαίωση, αυτό δείχνει ότι η διαδρομή έχει κάποιο σφάλμα. Με την ανίχνευση μιας αποτυχίας στη διαδρομή, ο κόμβος στέλνει ένα μήνυμα σφάλματος στον προηγούμενο κόμβο και απενεργοποιεί αυτό το μονοπάτι, ορίζοντας την τιμή της φερομόνης στο μηδέν. Ο προηγούμενος κόμβος στη συνέχεια προσπαθεί να βρει μια εναλλακτική διαδρομή για τον προορισμό. Εάν υπάρχει η εναλλακτική διαδρομή, το πακέτο προωθείται προς εκείνη την πορεία αλλιώς ο κόμβος ενημερώνει τους γείτονες να αναμεταδώσουν το πακέτο προς την πηγή. Αυτό συνεχίζεται μέχρι να φτάσει το πακέτο στην πηγή. Μόλις φτάσει στην πηγή, η πηγή αρχίζει μια νέα φάση δημιουργίας διαδρομής. Ο αλγόριθμος αποικίας μυρμηγκιών παρέχει πολλαπλές διαδρομές. Εάν η βέλτιστη διαδρομή αποτύχει, αυτόματα ο αλγόριθμος οδηγεί το σύστημα σε επιλογή της επόμενης καλύτερης διαδρομής. Η επόμενη καλύτερη διαδρομή θα είναι η πορεία που έχει τις διαδρομές με την επόμενη υψηλότερη τιμή φερομόνης (δεύτερη καλύτερη διαδρομή). Ως εκ τούτου, ο αλγόριθμος αποικίας μυρμηγκιών δεν σταματάει σε περίπτωση αποτυχίας της βέλτιστης διαδρομής. Αυτό βοηθά στην εξισορρόπηση φορτίου. Δηλαδή, εάν η βέλτιστη διαδρομή είναι υπερφορτωμένη, τα πακέτα δεδομένων μπορούν να ακολουθήσουν τις επόμενες καλύτερες διαδρομές.

Σε συνέχεια του ψευδοκώδικα του αλγορίθμου αποικίας μυρμηγκιών παρακάτω παρατίθεται ο ψευδοκώδικας της τελευταίας φάσης του βασικού αλγορίθμου:

Εάν υπάρχει εναλλακτική διαδρομή ήδη, γίνεται αποστολή των πακέτων από αυτή τη διαδρομή

αλλιώς {

- ρύθμιση επιπέδων φερομόνων στο 0 για όλο τον πίνακα δρομολόγησης και
- αποστολή πακέτου σφάλματος διαδρομής στον προηγούμενο κόμβο

}

Εάν το πακέτο σφάλματος φτάσει στον κόμβο - πηγή, επανεκκινά τη φάση δημιουργίας διαδρομής.

Ολοκληρώνοντας τη μελέτη του βασικού αλγορίθμου αποικίας μυρμηγκιών και προτού επεκτείνουμε τον αλγόριθμο δημιουργώντας το υβρίδιό του, πρέπει να σημειωθούν δυο βασικές παρατηρήσεις/προβλήματα που λαμβάνονται υπόψη στην υλοποίηση του αλγορίθμου.

Το πρώτο πρόβλημα αφορά την απώλεια μυρμηγκιών από το σύστημα, ανεξάρτητα από τη φάση στην οποία θα χαθούν. Αυτό το φαινόμενο μπορεί να συμβεί είτε κατά τη διαδικασία που το μυργμήγκι εκκινά από τον κόμβο - πηγή για να φτάσει στον προορισμό, είτε κατά τη διαδικασία που το μυρμήγκι επιστρέφει από τον τελικό προορισμό στην αφετηρία. Οι τρόποι αντιμετώπισης χωρίζονται σε δυο κατηγορίες:

- Αν υπάρχει απώλεια FA, τότε ενεργοποιείται ένας προκαθορισμένος χρόνος περίπου 6 δευτερολέπτων στον κόμβο - πηγή μέσα στον οποίο αν δεν επιστρέψει το μυρμήγκι BA, τότε ένα νέο μυρμήγκι FA με διαφορετικό αριθμό ξεκινάει το ταξίδι του.
- Αν υπάρχει απώλεια BA μετά την αποστολή του από τον κόμβο - πηγή, ενεργοποιείται ο ίδιος μηχανισμός χρονικής περιόδου με την προηγούμενη περίπτωση για να πραγματοποιηθεί χειρισμός της απώλειας μυρμηγκιού. Προκειμένου να γίνει πιο δυναμικό το σύστημα και πιο αποδοτικό, ξαναδημιουργείται ένα BA με τον ίδιο αύξοντα σειριακό αριθμό και ξεκινάει από τον κόμβο που είχε παρατηρηθεί τελευταία φορά, μειώνοντας τις καθυστερήσεις με την επανεκκίνηση όλου του συστήματος.

Μια άλλη κατηγορία προβλημάτων που μπορεί να εμφανιστούν στην υλοποίηση του βασικού αλγορίθμου αφορά το ποσό της μνήμης που πρέπει να δεσμευτεί εξαρχής για την αποθήκευση των πακέτων κατά τη διαδικασία της δημιουργίας διαδρομής. Μερικές φορές είναι αναγκαίο τα πακέτα που μεταφέρονται να αποθηκεύονται σε ενδιάμεσους κόμβους προκειμένου να αποφευχθεί πιθανή απώλειά τους κατά τη διάδοση. Εάν τα πακέτα χαθούν, μπορούν κάλλιστα από οποιονδήποτε προηγούμενο κόμβο έχουν αποθηκευτεί να επανασταλούν. Στην περίπτωση της απώλειας ολόκληρου μυρμηγκιού και όχι πακέτου, η τεχνική της αποθήκευσης βοηθάει να ξαναπαραχθούν μυρμήγκια για την εύρεση της διαδρομής.

Αφού έχει περιγραφεί ο ψευδοκώδικας πάνω στον οποίο θα βασιστεί η υλοποίηση του βασικού αλγορίθμου, επεκτείνεται ο αλγόριθμος αποικίας μυρμηγκιών με την προσθήκη μετα-ευρεστικών τεχνικών προσομοιωμένης ανόπτησης (simulated annealing). Η προσομοιωμένη ανόπτηση είναι μια γενική πιθανολογική μετα-ευρεστική τεχνική για την παγκόσμια βέλτιστη λύση σε ένα μεγάλο χώρο αναζήτησης. Χρησιμοποιείται συχνά όταν ο χώρος αναζήτησης των λύσεων είναι διακριτός. Η προσομοιωμένη ανόπτηση είναι μια από τις πιο ευέλικτες διαθέσιμες τεχνικές για την επίλυση συνδυαστικών προβλημάτων. Το κύριο πλεονέκτημά της είναι ότι μπορεί να εφαρμοστεί σε μεγάλου μεγέθους προβλήματα.

Η προσομοιωμένη ανόπτηση περιγράφει την ορισμένη πιθανότητα να αποφευχθεί ο εγκλωβισμός σε ένα τοπικό βέλτιστο, επιτρέποντας περιστασιακές αλλαγές που αυξάνουν την ποικιλομορφία των χαρακτηριστικών του σμήνους. Κατά τη διαδικασία της αναζήτησης, η προσομοιωμένη ανόπτηση δεν δέχεται μόνο καλύτερες, αλλά και χειρότερες γειτονικές λύσεις με μια ορισμένη πιθανότητα. Ο μηχανισμός αυτός μπορεί να θεωρηθεί ως μια δοκιμή για τη διερεύνηση νέων χώρων για νέες βέλτιστες λύσεις. Η πιθανότητα αποδοχής μιας χειρότερης λύσης είναι μεγαλύτερη σε υψηλότερες αρχικές θερμοκρασίες μέτρησης. Καθώς η θερμοκρασία μειώνεται, η πιθανότητα της αποδοχής χειρότερης λύσης σταδιακά προσεγγίζει το μηδέν. Πιο συγκεκριμένα, ξεκινώντας από μια αρχική κατάσταση, το σύστημα διαταράσσεται τυχαία σε μια νέα κατάσταση στη γειτονιά του πρωτότυπου. Έτσι, αυτό που πρέπει να γίνει είναι να υπολογιστεί η τιμή της νέας fitness function. Επιστρέφοντας σε αυτά που σχολιάστηκαν στην ενότητα 3.2.3, επιλέγεται εκείνη η νέα κατάσταση, της οποίας η πιθανότητα θα είναι

$$\min \{1, \exp(-\Delta S/T)\}$$

Αυτό που πετυχαίνει ο τελικός υβριδικός αλγόριθμος είναι να συνδυάσει όλα τα πλεονεκτήματα τόσο του βασικού αλγορίθμου αποικίας μυρμηγκιών όσο και της μετα-ευρεστικής τεχνικής προσομοιωμένης ανόπτησης. Όλα αυτά σε συνδυασμό βέβαια με τη χρήση των μετ-ευρεστικών τεχνικών της "αποταμίευσης" και της 2-opt για την εύρεση της βέλτιστης λύσης. Αφού βρεθεί η βέλτιστη λύση με τον αλγόριθμο αποικίας μυρμηγκιών, αυτό που ελέγχει ο υβριδικός αλγόριθμος είναι αν η λύση αποτελεί το βέλτστο των βέλτιστων λύσεων του συστήματος (min - min). Για να εξεταστεί ως προς min - min, κάθε διαδρομή αποτελεί μια διακριτή τυχαία μεταβλητή στην μήτρα φερομόνων (pheromone matrix). Για να ενεργοποιηθεί η αναζήτηση ως προς min-min, απαιτείται να έχει ήδη βρεθεί μια βέλτιστη λύση, χρησιμοποιώντας τεχνικές τοπικής αναζήτησης.

Στις παρακάτω γραμμές παρατίθεται ο αλγόριθμος της προσομοιωμένης ανόπτησης σε ψευδοκώδικα και μια υλοποίησή του σε C ανεξάρτητα από το πλήθος των κόμβων:

```
procedure SIMULATED_ANNEALING;
begin
    INITIALIZE (i_start, c_0, L_0);
    k := 0;
    i := i_start;
    repeat
        for l := 1 to L_k do
        begin
            GENERATE (j from S_i);
            if f(j) ≤ f(i) then i := j
            else
            if exp((f(i) − f(j)) / c_k) > random[0, 1) then i := j
        end;
        k := k + 1;
        CALCULATE_LENGTH (L_k);
        CALCULATE_CONTROL (c_k);
    until stopcriterion
end;
```

```python
from random import random

def anneal(sol):
    old_cost = cost(sol)
    T = 1.0
    T_min = 0.00001
    alpha = 0.9
    while T > T_min:
        i = 1
        while i <= 100:
            new_sol = neighbor(sol)
            new_cost = cost(new_sol)
            ap = acceptance_probability(old_cost, new_cost, T)
            if ap > random():
                sol = new_sol
                old_cost = new_cost
            i += 1
        T = T*alpha
    return sol, cost
```

4. Επίλογος - Μελλοντική εργασία

Τα προβλήματα δρομολόγησης οχημάτων με περιοσμένη χωρητικότητα είναι από τα πιο σημαντικά προβλήματα στον τομέα της διανομής και της εφοδιαστικής αλυσίδας. Από τη στιγμή που οι διαδρομές μπορούν να αποτελούνται από οποιονδήποτε συνδυασμό πελατών, τα προβλήματα αυτά κατατάσσονται στην κατηγορία των NP-Hard. Με αυτή τη μελέτη περίπτωσης προτείνεται η επίλυση της κατηγορίας των προβλημάτων με έναν υβριδικό αλγόριθμο αποικίας μυρμηγκιών συνδυασμένος με τις αρχές αρκετών μετα-ευρεστικών τεχνικών.

Σε αυτή τη μελέτη περίπτωσης έχει περιγραφεί όλο το βασικό υπόβαθρο της έρευνας για την επίλυση προβλημάτων δρομολόγησης οχημάτων με περιορισμένη χωρητικότητα. Παρουσιάστηκε εκτενή βιβλιογραφική ανάλυση, περιγράφηκε το πρόβλημα που ζητείται να επιλυθεί και τα χαρακτηριστικά του, μαθηματικοποιήθηκε με τις κατάλληλες συναρτήσεις ώστε να προκύψει η αντικειμενική του συνάρτηση, οι περιορισμοί του αλλά και η fitness function που θα λειτουργήσει για την εύρεση της βέλτιστης λύσης.

Σε συνέχεια της έρευνας αυτής, θα παρουσιαστούν τα αποτελέσματα από την επίλυση του προβλήματος καθώς και συγκριτικοί πίνακες με τις διαφορές που μπορεί να έχει αυτός ο αλγόριθμος, σε σχέση με άλλους που έχουν δημοσιευτεί στη βιβλιογραφία.

Σημειώνεται οτι όλες οι υλοποιήσεις θα πραγματοποιηθούν σε γλώσσα C# στο προγραμματιστικό περιβάλλον του Visual Studio της Microsoft αυξάνοντας σταδιακά το πλήθος των πελατών προς εξυπηρέτηση, με στόχο να φανεί η αποτελεσματικότητα του αλγορίθμου και να καταγραφεί η εξέλιξη των χρόνων υπολογισμού.

Βιβλιογραφία - Πηγές

1. N. Christofides, (1985) Vehicle routing, in: *E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys (Eds.), The Traveling Salesman Problem*, Wiley, Chichester, pp. 431–448.
2. N. Christofides, S. Eilon, (1969) An algorithm for the vehicle dispatching problem, *Oper. Res. Quart. 20* 309–318.
3. N. Christofides, A. Mingozzi, (1989) Vehicle routing: practical and algorithmic aspects, in: *C.F.H. van Rijn (Ed.), Logistics: Where Ends Have to Meet*, Pergamon Press, Oxford, pp. 30–48.
4. N. Christofides, A. Mingozzi, P. Toth, (1979) The vehicle routing problem, in: *N. Christofides, A. Mingozzi,P. Toth, C. Sandi (Eds.), Combinatorial Optimization*, Wiley, Chichester, pp. 315–338.
5. N. Christofides, A. Mingozzi, P. Toth, (1981) Exact algorithms for the vehicle routing problem based on the spanning tree and shortest path relaxations, *Math. Programming 20* pp. 255–282.
6. G. Cornuejols, F. Harche, (1993) Polyhedral study of the capacitated vehicle routing problem, Math*. Programming 60* pp. 21–52.
7. M. Fischetti, P. Toth, D. Vigo, (1994) A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs, *Oper. Res. 42* , pp. 846–859.
8. B.L. Golden, A.A. Assad, (1988) Vehicle Routing: Methods and Studies, North-Holland, Amsterdam.
9. B.L. Golden, E.A. Wasil, J.P. Kelly, I. Chao, (1998) The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results, in: *T.G. Crainic, G. Laporte (Eds.), Fleet Management and Logistic*, Kluwer Academic Publisher, Boston (MA), pp. 33–56.
10. R.V. Kulkarni, P.V. Bhave, (1985) Integer programming formulations of vehicle routing problems, *European J. Oper. Res. 20*, pp. 58–67.
11. G. Laporte, (1992) The vehicle routing problem: An overview of exact and approximate algorithms, *European J. Oper. Res. 59,* pp. 345–358.
12. G. Laporte, (1997) Vehicle routing, in: *M. Dell'Amico, F. Maffoli, S. Martello (Eds.), Annotated Bibliographies in Combinatorial Optimization*, Wiley, Chichester
13. G. Laporte, Y. Nobert, M. Desrochers, (1985) Optimal routing under capacity and distance restrictions, *Oper. Res. vol.33* , pp. 1050–1073.
14. T.L. Magnanti, (1981) Combinatorial optimization and vehicle fleet planning: Perspectives and prospects, *Networks vol.11* ,pp.179–214
15. S. Martello, P. Toth, (1990) Knapsack Problems: Algorithms and Computer Implementations, Wiley, Chichester

16. P. Toth, D. Vigo, (1997) An exact algorithm for the vehicle routing problem with backhauls, Transport. Sci. 31, 372–385.
17. P. Toth, D. Vigo, (1998) Exact solution of the vehicle routing problem, in: *T.G. Crainic, G. Laporte (Eds.), Fleet Management and Logistic*, Kluwer Academic Publisher, Boston (MA), pp. 1–31.
18. P. Toth, D. Vigo, (1998) The granular tabu search (and its application to the vehicle routing problem). *Technical Report OR/98/9, D.E.I.S.* - Universita di Bologna
19. D. Vigo, (1999) VRPLIB: a vehicle routing problem library, *Technical Report OR/99/9, D.E.I.S*., Universita di Bologna
20. Eilon, S., Watson-Gandy, C.D.T., and Christofides N., (1971) Distribution Management: Mathematical Modelling and Practical Analysis, *Griffin*, London, 1971
21. Fisher, M., Jaikumar R., (1981) A generalized assignment heuristic for vehicle routing, *Networks, vol. 11*, 109-124
22. Paessens, H. (1988) The savings algorithm for the vehicle routing problem, *European Journal of Operational Research vol. 34, 336-344*
23. Lin S., (1965) Computer solutions for the traveling salesman problem, *Bell System Technical Journal, vol 44*, 2245-2269.
24. Augerat, P., Belenguer, J.M., Benavent, E., Corberán, A., Naddef, D., Rinaldi, G. (1995). Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical Report RR 949-M, Université Joseph Fourier, Grenoble.
25. Augerat, P., Belenguer, J.M., Benavent, E., Corberán, A., Naddef, D. (1999). Separating capacity inequalities in the CVRP using tabu search. *European Journal of Operational Research* 106, 546–557.
26. Bastian, C., Rinnooy Kan, A.H.G. (1992). The stochastic vehicle routing problem revisited. *European Journal of Operational Research* 56, 407–412.
27. Battiti, R., Tecchiolli, G. (1994). The reactive tabu search. *ORSA Journal on Computing* 6, 126–140.
28. Beasley, J.E. (1983). Route-first cluster-second methods for vehicle routing. *Omega* 11, 403–408.
29. Beraldi, P., Ghiani, G., Laporte, G., Musmanno, G. (2005). Efficient neighbourhood search for the probabilistic pickup and delivery travelling salesman problem. *Networks* 46, 195–198.
30. Berger, J., Barkaoui, M., Bräysy, O. (2003). A route-directed hybrid genetic approach for the vehicle routing problem with time windows. *INFOR* 41, 179–194.
31. Bräysy, O. (2003). A reactive variable neighborhood search for the vehicle routing problem with time windows. *INFORMS Journal on Computing* 15, 347–368.

32. Bräysy, O., Gendreau,M. (2005a). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science* 39, 104–118.
33. Bräysy, O., Gendreau,M. (2005b). Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation Science* 39, 119–139.
34. Cordeau, J.-F., Laporte, G. (2001). A tabu search algorithm for the site dependent vehicle routing problem with time windows. *INFOR* 39, 292–298.
35. Cordeau, J.-F., Laporte, G. (2004). Tabu search heuristics for the vehicle routing problem. In: Rego, C., Alidaee, B. (Eds.), *Metaheuristic Optimization via Memory and Evolution: Tabu Search and Scatter Search*. Kluwer Academic, Boston, pp. 145–163.
36. Cordeau, J.-F., Gendreau,M., Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* 30, 105–119.
37. Cordeau, J.-F., Laporte, G., Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society* 52, 928–936.
38. Cordeau, J.-F., Desaulniers, G., Desrosiers, J., Solomon, M.M., Soumis, F. (2002a). VRP with Time Windows. In: Toth, P., Vigo, D. (Eds.), *The Vehicle Routing Problem*. *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, Philadelphia, pp. 157–193.
39. Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y., Semet, F. (2002b). A guide to vehicle routing heuristics. *Journal of the Operational Research Society* 53, 512–522.
40. Cordeau, J.-F., Laporte, G.,Mercier, A. (2004). An improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows. *Journal of the Operational Research Society* 55, 542–546.
41. Cordeau, J.-F., Gendreau,M., Hertz, A., Laporte,G., Sormany, J.-S. (2005). New heuristics for the vehicle routing problem. In: Langevin, A., Riopel, D. (Eds.), *Logistics Systems: Design and Optimization*. Springer-Verlag, New York, pp. 279–297.
42. Cordone, R., Wolfler Calvo, R. (2001). A heuristic for the vehicle routing problem with time windows. *Journal of Heuristics* 7, 107–129.
43. Dantzig, G.B., Ramser, J.M. (1959). The truck dispatching problem. *Management Science* 6, 81–91.
44. Dantzig, G.B., Wolfe, P. (1960). Decomposition principle for linear programming. *Operations Research* 8, 101–111.
45. Lysgaard, J., Letchford, A., Eglese, R.W. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming* 100, 423–445.

46. Or, I. (1976). Traveling salesman-type combinatorial problems and their relation to the logistics of blood banking. PhD thesis, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL.
47. Potvin, J.-Y. (1996). Genetic algorithms for the traveling salesman problem. *Annals of Operations Research* 63, 339–370.
48. Potvin, J.-Y., Bengio, S. (1996). The vehicle routing problem with time windows – Part II: Genetic search. *INFORMS Journal on Computing* 8, 165–172.
49. Potvin, J.-Y., Rousseau, J.-M. (1993). A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research* 66, 331–340.
50. Potvin, J.-Y., Rousseau, J.-M. (1995). An exchange heuristic for routing problems with time windows. *Journal of the Operational Research Society* 46, 1433–1446.
51. Potvin, J.-Y., Kervahut, T., Garcia, B.L., Rousseau, J.-M. (1996). The vehicle routing problem with time windows – Part I: Tabu search. *INFORMS Journal on Computing* 8, 158–164.
52. Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research* 31, 1985–2002.
53. Wren, A. (1971). *Computers in Transport Planning and Operation*. Ian Allan, London.
54. Wren, A., Holliday, A. (1972). Computer scheduling of vehicles from one or more depots to a number of delivery points. *Operational Research Quarterly* 23, 333–344.
55. Xu, J., Kelly, J.P. (1996). A network flow-based tabu search heuristic for the vehicle routing problem. *Transportation Science* 30, 379–393.
56. Φούντας Ε. , Βλάχος Α.Γ. (2207) Μαθηματικά Αποφάσεων, Πανεπιστήμιο Πειραιώς, Πειραιάς