ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ Μ/Υ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΝΑΥΤΙΛΙΑΣ ΚΑΙ ΒΙΟΜΗΧΑΝΙΑΣ
ΤΜΗΜΑ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ
ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Το Πρόβλημα του Περιοδεύοντος Πωλητή: Ανάλυση Ερευνητικού Πεδίου, Αλγόριθμοι Επίλυσης και Επιχειρησιακές Εφαρμογές

ΑΛΕΞΙΟΣ ΤΡΥΦΩΝΑΣ ΧΑΡΑΛΑΜΠΟΥΣ

Επιβλέπων Καθηγητής:

Ματσόπουλος Γ., Καθηγητής ΕΜΠ

Αθήνα, Οκτώβριος 2019

Ευχαριστίες

Ολοκληρώνοντας τη διπλωματική μου εργασία δε θα μπορούσα να μην αναφερθώ και να μην ευχαριστήσω τους ανθρώπους που με βοήθησαν και στήριξαν την προσπάθειά μου.

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Ματσόπουλο Γεώργιο για την καθοδήγησή του, την ενθάρρυνση και την υποστήριξή του καθ' όλη τη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας και για την ευκαιρία που μου έδωσε να διερευνήσω ένα τόσο ενδιαφέρον θέμα όπως το Πρόβλημα του Περιοδεύοντος Πωλητή.

Ιδιαίτερες ευχαριστίες οφείλω στη συνάδελφό μου Τουρνάκη Ελένη για τις υποδείξεις της και τις διορθώσεις της πάνω στο κείμενο.

Τέλος, θα ήθελα να ευχαριστήσω θερμά την οικογένειά μου που με έχει στηρίξει αμέριστα σε ό,τι και αν έχω κάνει έως τώρα. Ένα μεγάλο ευχαριστώ οφείλω, ακόμη, στους φίλους μου, τους συναδέλφους μου και τους συμφοιτητές μου, που έχω την τύχη να με συντροφεύουν εντός και εκτός σπουδών τα τελευταία χρόνια.

Αλέξιος Τρύφωνας Χαραλάμπους
Οκτώβριος 2019

ΠΕΡΙΛΗΨΗ

Αντικείμενο της παρούσας εργασίας είναι η διευρένηση του Προβλήματος του Περιοδεύοντος Πωλητή και των προεκτάσεών του. Καταρχάς, παρουσιάζεται η διατύπωση του προβλήματος καθώς και οι κυριότερες παραλλαγές του και γίνεται αναφορά στα κύρια χαρακτηριστικά κάθε μίας εξ αυτών. Στη συνέχεια, γίνεται αναφορά στους κυριότερους αλγορίθμους (ακριβείς και ευρετικούς) που έχουν χρησιμοποιηθεί κατά καιρούς για την επίλυση προβλημάτων ολικής βελτιστοποίησης. Ακόμη, επιχειρείται η αναγνώριση των κριτηρίων με βάση τα οποία οι αλγόριθμοι αυτοί συγκρίνονται στη βιβλιογραφία καθώς και η συγκριτική τους αξιολόγηση. Τέλος, παρουσιάζεται πλήθος επιχειρησιακών εφαρμογών του Προβλήματος του Περιοδεύοντος Πωλητή και των παραλλαγών του από τη διεθνή βιβλιογραφία, που καταδεικνύουν την πρακτική του αξία σε αρκετά προβλήματα της καθημερινότητας.

Λέξεις κλειδιά: Πρόβλημα Περιοδεύοντος Πωλητή, TSP, Εφαρμογές TSP, Ακριβείς Αλγόριθμοι, Ευριστικοί Αλγόριθμοι

ABSTRACT

The purpose of this thesis is to investigate the Travelling Salesman Problem and its extensions. First of all, the problem itself as well as its main variations are presented and the main characteristics of each one of them are discussed. Then, the main algorithms (exact and heuristic) that have been used to solve total optimization problems are analysed. It is, also, attempted to identify the criteria on the basis of which these algorithms are compared in the literature and their benchmarking. Finally, a number of operational applications of the Travelling Salesman Problem and its variations, mentioned in the international literature, are presented, demonstrating its practical value in many everyday problems.

Keywords: Travelling Salesman Problem, TSP, TSP Applications, Exact Algorithms, Heuristic Algorithms

Πίνακας περιεχομένων

1. Εισαγωγή

Στο κεφάλαιο της εισαγωγής αναφέρονται λίγα λόγια για το Πρόβλημα του Περιοδεύοντος Πωλητή που αναλύεται στην παρούσα εργασία ενώ, στη συνέχεια, ακολουθεί το αντικείμενο, ο στόχος και η διάρθρωση αυτής.

1.1. Γενικά

Το Πρόβλημα του Περιοδεύοντος Πωλητή είναι ένα από τα βασικότερα προβλήματα στη θεωρία βελτιστοποίησης και ως εκ τούτου έχει μελετηθεί εις βάθος. Παρότι όμως έχεις μελετηθεί διεξοδικά από ερευνητές για δεκαετίες, το ενδιαφέρον για αυτό παραμένει αμείωτο. Ένας πωλητής ξεκινάει από μία αρχική πόλη με σκοπό να επισκεφτεί από ακριβώς μια φορά κάθε πόλη μιας δοθείσας λίστας και στη συνέχεια να επιστρέψει στην αρχική πόλη, επιλέγοντας εύλογα τη σειρά με την οποία θα επισκεφτεί τις πόλεις, ώστε η συνολική απόσταση που θα διανύσει να είναι η μικρότερη δυνατή. Το πρόβλημα εύρεσης της διαδρομής ονομάζεται «Το Πρόβλημα του Περιοδεύοντος Πωλητή» και στην αγγλική βιβλιογραφία είναι γνωστό ως “Travelling Salesman Problem” (TSP). Το πρόβλημα αυτό μοιάζει, αρχικά, εύκολο καθώς για ένα δοθέν σύνολο ν πόλεων υπάρχουν πεπερασμένες πιθανές διαδρομές. Υπάρχει, όμως, μια σημαντική δυσκολία. Από ένα πλήθος πόλεων και έπειτα, ο αριθμός των διαφορετικών, εφικτών λύσεων είναι υπερβολικά μεγάλος ώστε να τις ελέγξουμε όλες μία προς μία. Για ν πόλεις υπάρχουν (ν-1)! / 2 πιθανές διαδρομές που αποτελούν λύσεις του προβλήματος. Για ένα συμμετρικό πρόβλημα 33 πόλεων έχουμε λοιπόν περίπου $131 \cdot 10^{33}$ πιθανές διαδρομές-λύσεις του προβλήματος.

Το TSP είναι χαρακτηριστική περίπτωση των προβλημάτων «συνδυαστικής βελτιστοποίησης» (combinatorial optimization). Προσπαθούμε να ελαχιστοποιήσουμε τη συνολικά διανυόμενη απόσταση μίας διαδρομής που θα προκύψει μέσα από ένα πεπερασμένο πλήθος όλων των πιθανών διαδρομών του προβλήματος. Η σπουδαιότητα του προβλήματος αυτού, πέρα από την πληθώρα των εφαρμογών του, έγκειται και στα πολλά επίπεδα μοντελοποίησης. Πιο συγκεκριμένα, χρησιμοποιούνται τόσο ακριβείς αλγόριθμοι, δηλαδή αλγόριθμοι που δίνουν ως λύση το ολικό βέλτιστο (global optimum), όσο και αλγόριθμοι προσεγγιστικής επίλυσης μέσω ανάπτυξης ευρετικών μηχανισμών (heuristics) που μπορούν να χρησιμοποιηθούν και σε άλλες μαθηματικές εφαρμογές πέραν των προβλημάτων συνδυαστικής βελτιστοποίησης.

Μαθηματικά, το πρόβλημα εκφράζεται ως εξής: θεωρούμε G = (V,A) μια γραφική παράσταση με V ένα σύνολο n κόμβων. Θεωρούμε A το σύνολο των τόξων, ή ακμών που συνδέουν τους κόμβους μεταξύ τους, και C_{ij} τον πίνακα κόστους που σχετίζεται με τα τόξα. Το TSP συνίσταται στην εύρεση της κλειστής διαδρομής με το ελάχιστο κόστος, για την οποία κάθε κόμβος επισκέπτεται ακριβώς μία φορά. Αποτελεί ένα πρόβλημα Μικτού Ακέραιου Γραμμικού Προγραμματισμού (Mixed Integer Linear Programming – MILP). Οι μαθηματικές σχέσεις δηλαδή που το περιγράφουν είναι γραμμικές, ενώ οι μεταβλητές που περιέχονται είναι συνδυασμός συνεχών και ακέραιων. Ένας μεγάλος αριθμός ακριβών αλγορίθμων έχει προταθεί για την επίλυση του Προβλήματος του Περιοδεύοντος Πωλητή. Οι αλγόριθμοι μπορούν να γίνουν καλύτερα αντιληπτοί και να εξηγηθούν στα πλαίσια του Ακέραιου Γραμμικού Προγραμματισμού [1]. Κάποιοι από τους ακριβείς αλγορίθμους παρουσιάζονται στο κεφάλαιο 3.1 ενώ στο κεφάλαιο 3.2 παρουσιάζονται οι κυριότεροι ευρετικοί αλγόριθμοι που έχουν αναπτυχθεί για την επίλυση του TSP.

1.2. Αντικείμενο και Στόχος της Εργασίας

Αντικείμενο και στόχος της παρούσας εργασίας είναι να μελετηθεί το Πρόβλημα του Περιοδεύοντος Πωλητή. Πιο συγκεκριμένα, γίνεται ανάλυση του ερευνητικού πεδίου όπου, ύστερα από επισκόπηση της σχετικής διεθνούς βιβλιογραφίας παρουσιάζεται το Πρόβλημα του Περιοδεύοντος Πωλητή αλλά και συναφή προβλήματα που συνδέονται με αυτό. Στη συνέχεια, γίνεται μια αναφορά στους κυριότερους ακριβείς και ευρετικούς αλγορίθμους που έχουν χρησιμοποιηθεί για την επίλυση του Προβλήματος καθώς και στα κριτήρια που λαμβάνονται υπόψη όταν αξιολογούνται αυτοί οι αλγόριθμοι, ενώ επιχειρείται και μία σύγκριση ορισμένων από αυτούς. Τέλος, γίνεται αναφορά στις επιχειρησιακές εφαρμογές του προβλήματος με παραδείγματα όπου η αντιμετώπισή τους απαιτεί την κατανόηση και την επίλυση του Προβλήματος του Περιοδεύοντος Πωλητή ή κάποιας εκ των παραλλαγών του.

1.3. Διάρθρωση Εργασίας

Η ανάπτυξη του θέματος της παρούσας εργασίας περιλαμβάνει, εκτός από το κεφάλαιο της εισαγωγής, πέντε ακόμα κεφάλαια. Στο δεύτερο κεφάλαιο γίνεται η διατύπωση του προβλήματος που μελετάται και παρουσιάζονται διάφορες παραλλαγές αυτού. Στο τρίτο κεφάλαιο γίνεται αναφορά στα είδη αλγορίθμων που χρησιμοποιούνται για να βρεθεί η λύση του Προβλήματος του Περιοδεύοντος Πωλητή ενώ επιχειρείται και μία σύγκριση ορισμένων από αυτούς. Στο τέταρτο κεφάλαιο παρουσιάζονται επιχειρηματικές εφαρμαγές του Προβλήματος του Περιοδεύοντος Πωλητή και των παραλλαγών του, ενώ στο πέμπτο κεφάλαιο γίνεται μια σύνοψη των εξαχθέντων συμπερασμάτων της παρούσας εργασίας. Το έκτο και τελευταίο κεφάλαιο περιλαμβάνει τις εισηγήσεις για περαιτέρω έρευνα.

2. Διατύπωση του Προβλήματος

Σε αυτό το κεφάλαιο αρχικά διατυπώνεται το βασικό Πρόβλημα του Περιοδεύοντος Πωλητή ενώ στη συνέχεια αναφέρονται οι κυριότερες παραλλαγές του.

2.1. Το Πρόβλημα του Περιοδεύοντος Πωλητή

Το Πρόβλημα του Περιοδεύοντος Πωλητή (Travelling Salesman Problem – TSP) είναι ένα δύσκολο πρόβλημα συνδυαστικής βελτιστοποίησης που μελετάται κυρίως στην επιχειρησιακή έρευνα και τη θεωρητική πληροφορική. Στο Πρόβλημα του Περιοδεύοντος Πωλητή δίνεται ένα σύνολο από κόμβους (συχνά πόλεις) για τους οποίους είναι γνωστές οι μεταξύ τους αποστάσεις. Σε μια πιο γενική περίπτωση, αντί για αποστάσεις χρησιμοποιούνται βάρη που ενώνουν τον κάθε κόμβο με τον επόμενο. Τα βάρη αυτά μπορεί να είναι πέρα από την απόσταση μεταξύ δύο κόμβων μετρημένη σε μονάδες μήκους ή σε μονάδες χρόνου, το κόστος μεταφοράς για να μεταβεί κάποιος από τον έναν κόμβο στον άλλον ή οτιδήποτε άλλο. Έστω ότι μιλάμε για αποστάσεις, τα ίδια ισχύουν σε κάθε περίπτωση. Στόχος του προβλήματος είναι να βρεθεί η συντομότερη διαδρομή που πρέπει να ακολουθηθεί ώστε ο πωλητής να περάσει μία μόνο φορά από κάθε πόλη μιας δοθείσας λίστας πόλεων και να επιστρέψει πάλι στην πόλη από την οποία ξεκίνησε. Το πρόβλημα της εύρεσης της συντομότερης διαδρομής που πρέπει να διανυθεί από έναν πωλητή ώστε να επισκεφθεί όλες τις πόλεις περνώντας από την κάθε μια, διατυπώθηκε αρχικά το 1930 και αποτελεί ένα από τα πιο μελετημένα προβλήματα της θεωρίας της βελτιστοποίησης. Αποτελεί έναν κορμό προβλήματος πάνω στο οποίο χτίζονται λύσεις για διάφορα άλλα προβλήματα. Χρησιμοποιείται επίσης σαν συγκριτικό μέτρο επιδόσεων για πολλές μεθόδους βελτιστοποίησης. Ακόμα και αν το πρόβλημα είναι υπολογιστικά δύσκολο, έχουν αναπτυχθεί μέχρι σήμερα πολλές ευρετικές μέθοδοι μέσω των οποίων αντιμετωπίζεται [2].

Οι απαρχές του TSP δεν είναι σαφείς. Ένα εγχειρίδιο για πλανόδιους πωλητές το 1832 αναφέρει το πρόβλημα και περιλαμβάνει παραδείγματα περιηγήσεων μέσω της Γερμανίας και της Ελβετίας, αλλά δεν περιέχει καμία μαθηματική προσέγγιση. Τον $19^{ο}$ αιώνα το TSP ορίστηκε από τον Ιρλανδό μαθηματικό William Rowan Hamilton και το Βρετανό μαθηματικό Thomas Kirkman. Ο Hamilton δημιούργησε το Icosian Puzzle (το όνομα είναι από το αρχαίο ελληνικό είκοσι, ενώ αναφέρεται στη δεθνή βιβλιογραφία και ως Icosian game), που περιλαμβάνει την εύρεση ενός κύκλου από τις άκρες ενός δωδεκάεδρου, έτσι ώστε κάθε κόμβος να επισκέπτεται μία μόνο φορά, κανένας κόμβος να μην επισκέπτεται 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.

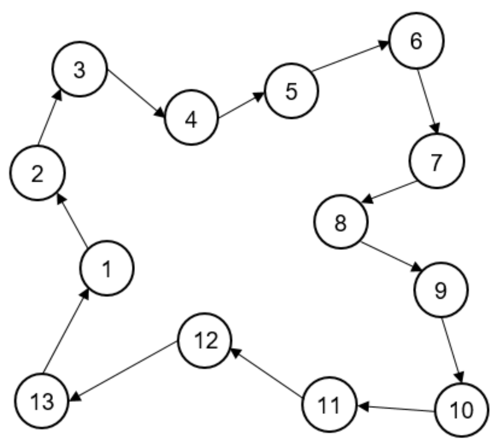
Η γενική μορφή του TSP φαίνεται να έχει μελετηθεί για πρώτη φορά κατά την δεκαετία του 1930 στη Βιέννη και στο Χάρβαρντ κυρίως από τον Karl Menger ο οποίος καθόρισε και το πρόβλημα. Ο Merrill Meeks Flood φαίνεται να είναι ο πρώτος που έδωσε, την ίδια περίοδο, μαθηματική υπόσταση στο πρόβλημα στην προσπάθειά του να λύσει το πρόβλημα δρομολόγησης ενός σχολικού λεωφορείου. Ο Hasslier Whitney στο Πανεπιστήμιο του Princeton εισήγαγε το όνομα Travelling Salesman Problem (TSP) αμέσως μετά. Στην δεκαετία του 1950 και 1960, έγινε όλο και πιο δημοφιλές στους επιστημονικούς κύκλους της Ευρώπης και της Αμερικής, ειδικότερα αφότου σε ένα συνέδριο στην Santa Monica προσφέρθηκαν βραβεία σε όσους κατάφερναν να συνεισφέρουν στην επίλυση του προβλήματος [3].

Οι κύριες συνιστώσες για τις σημερινές πιο επιτυχείς προσεγγίσεις για δύσκολα συνδυαστικά προβλήματα βελτιστοποίησης είναι οι αλγόριθμοι Ευρετικής Αναζήτησης, ο Γραμμικός Προγραμματισμός και οι αλγόριθμοι Branch and Bound (B&B), που αρχικά διατυπώθηκαν για το TSP αλλά συνήθιζαν να επιλύουν περιπτώσεις πιο πρακτικών προβλημάτων. Στις επόμενες δεκαετίες το πρόβλημα μελετήθηκε περαιτέρω από πολλούς ερευνητές διαφόρων κλάδων των Μαθηματικών, των Υπολογιστών, της Χημείας, της Φυσικής και της Επιχειρησιακής Έρευνας. Ο Richard Karp, έδειξε το 1972, ότι το πρόβλημα με τον κύκλο του Hamilton ήταν ένα NP-πλήρες πρόβλημα (δηλαδή μπορεί να επαληθευτεί σε πολυωνυμικό χρόνο), από το οποίο προκύπτει για το TSP ότι ανήκει στα NP-δύσκολα προβλήματα λόγω της πολυπλοκότητας του. Αυτό παρέχει μια μαθηματική εξήγηση για την προφανή υπολογιστική δυσκολία εύρεσης βέλτιστων περιηγήσεων. Αργότερα, αναπτύχθηκαν νέες αλγοριθμικές τεχνικές και εφαρμόστηκαν στο TSP για να αποδείξουν την αποτελεσματικότητά τους. Παραδείγματα τέτοιων τεχνικών, πέρα από τον αλγόριθμο B&B είναι η μέθοδος χαλάρωσης του Lagrange, ο αλγόριθμος και η ευριστίκη συνάρτηση των Lin-Kernighan, η Προσομοιωμένη Ανόπτηση και το πεδίο των συνδυαστικών πολύεδρων για δύσκολα συνδυαστικά προβλήματα βελτιστοποίησης.

Μεγάλη πρόοδος επιτεύχθηκε στα τέλη των δεκαετιών 1970 και 1980, όταν οι Grotschel, Padberg, Rinaldi και άλλοι, κατάφεραν να επιλύσουν ακριβώς περιπτώσεις με έως και 2.392 πόλεις, χρησιμοποιώντας τις μεθόδους Cutting Plane και B&B. Κατά την δεκαετία του 1990, οι Applegate, Bixby, Chvatal και Cook ανάπτυξαν το πρόγραμμα “Concorde TSP Solver” που έχει χρησιμοποιηθεί σε πολλές πρόσφατες καταγεγραμμένες λύσεις ενώ είναι διαθέσιμο δωρεάν για ακαδημαϊκούς σκοπούς. Ο Gerhard Reinelt το 1991 δημοσίευσε το TSPLIB, το οποίο είναι μια συλλογή από συγκριτικές αξιολογήσεις περιπτώσεων διαφορετικής δυσκολίας και έχει χρησιμοποιηθεί επίσης από πολλούς ερευνητές για σύγκριση των αποτελεσμάτων. Το 2006, ο Cook και οι συνεργάτες του, υπολόγισαν μία περίπτωση βέλτιστης διαδρομής διαμέσου 85.900 πόλεων, η οποία δόθηκε από ένα πρόβλημα διάταξης ενός μικροτσίπ και αποτελεί επί του παρόντος τη μεγαλύτερη λυμένη περίπτωση. Για πολλές άλλες περιπτώσεις, με εκατομμύρια πόλεις, οι λύσεις που μπορούν να βρεθούν εγγυώνται μόνο σε ποσοστό 1% να είναι μια βέλτιστη διαδρομή.

Ένας από τους λόγους που έκαναν το TSP ένα τόσο δημοφιλές πρόβλημα ήταν και η στενή του σχέση με εξέχοντα θέματα των συνδυαστικών προβλημάτων που προέκυπταν από την, τότε, νέα μέθοδο του γραμμικού προγραμματισμού, ιδίως του προβλήματος εκχώρησης και, πιο γενικά, του προβλήματος μεταφοράς. Το TSP ήταν σαν όλα αυτά τα προβλήματα, αλλά προφανώς πιο δύσκολο στο να λυθεί, και έτσι η πρόκληση έγινε ακόμα πιο ενδιαφέρουσα. Και, φυσικά, το Πρόβλημα του Περιοδεύοντος Πωλητή έγινε τόσο δημοφιλές επειδή έχει ένα όνομα που υπενθύμιζε και υπενθυμίζει σε όλους άλλα πράγματα. Ο περιπλανώμενος πωλητής ως προσωπικότητα ήταν μια από τις κλασικές προσωπικότητες στην αμερικάνικη ιστορία. Αδιαμφισβήτητο επομένως είναι το γεγονός ότι η μεγάλη απήχηση που λαμβάνει το TSP οφείλεται, εν μέρει, και στο όνομα που του αποδόθηκε [4].

Στην Εικόνα 1 που ακολουθεί φαίνεται το παράδειγμα επίλυση ενός Προβλήματος Περιοδεύοντος Πωλητή σε ένα γράφο με 13 κόμβους.



Εικόνα 1: Παράδειγμα επίλυσης Προβλήματος Περιοδεύοντος Πωλητή 13 κόμβων

Όπως προαναφέρθηκε, το Πρόβλημα του Περιοδεύοντος Πωλητή, υπό μία άποψη, αποτελεί ένα εύκολο πρόβλημα καθώς για ένα δοσμένο σύνολο ν πόλεων υπάρχει πεπερασμένο πλήθος πιθανών διαδρομών, μία εκ των οποίων είναι και η ολικά βέλτιστη. Η δυσκολία του προβλήματος έγκειται στο γεγονός ότι από ένα πλήθος πόλεων και πάνω, ο αριθμός όλων των πιθανών διαδρομών είναι αρκετά μεγάλος για να τις εξετάσουμε όλες μία προς μία. Πιο συγκεκριμένα, έστω ότι αναζητάμε τη λύση ενός προβλήματος TSP για ν πόλεις. Ξεκινώντας από την πόλη που ορίζεται πρώτη στη διαδρομή (αρχική πόλη) έχουμε ν-1 επιλογές για το ποιά θα είναι η δεύτερη πόλη της διαδρομής μας, ν-2 επιλογές για την τρίτη πόλη και ούτω κάθε εξής. Κατά συνέπεια, υπάρχουν συνολικά (ν-1)·(ν-2)·(ν-3)·…·3·2·1 = (ν-1)! πιθανές διαδρομές που αποτελούν όλες λύσεις του προβλήματος. Στο TSP αναζητείται η ελάχιστη από όλες αυτές τις λύσεις. Εφόσον το πρόβλημα είναι συμμετρικό, δηλαδή οι αποστάσεις μεταξύ των πόλεων *i* και *j* ισούνται με τις αποστάσεις μεταξύ των *j* και *i* για κάθε ζεύγος πόλεων *i, j* του προβλήματος, οι πιθανές λύσεις είναι (ν-1)! / 2 [1].

Για να κατανοήσουμε το μέγεθος του προβλήματος, εξετάζουμε την παρακάτω περίπτωση. Έστω ότι έχουμε μια λίστα με 30 πόλεις (δηλαδή ν = 30). Σε αυτή την περίπτωση, το TSP έχει 30! / 2 = $1{,}3253 \cdot 10^{32}$ πιθανές λύσεις. Αυτή τη στιγμή, ο πιιο γρήγορος υπερυπολογιστής στον κόσμο καλείται Summit, κατασκευάστηκε από την IBM για λογαριασμό της αμερικανικής κυβέρνησης, κόστισε 325 εκατομμύρια δολλάρια, χρησιμοποιείται από το Υπουργείο Ενέργειας των ΗΠΑ και τρέχει με ταχύτητα 200 Petaflops (δηλαδή υπολογισμών κινητής υποδιαστολής το δευτερόλεπτο). Αυτή η ταχύτητα αντιστοιχεί σε $200 \cdot 10^{15}$ πράξεις το δευτερόλεπτο. Υποθέτουμε λοιπόν ότι είμαστε σε θέση να χρησιμοποιήσουμε αυτόν τον υπολογιστή για να εξετάσουμε μία προς μία όλες τις πιθανές λύσεις του TSP των 30 πόλεων και πως με μία μόνο πράξη ο υπολογιστής εξετάζει μία διαδρομή. Τότε, αυτός ο υπορυπολογιστής θα χρειαζόταν $663 \cdot 10^{12}$ δευτερόλεπτα για να βρει την ολικά βέλτιστη λύση του προβλήματος ή πιο απλά κάτι παραπάνω από περίπου 21.000 χρόνια ώστε να εξετάσει όλες τις πιθανές λύσεις και να βρει τη βέλτιστη διαδρομή. Όπως γίνεται εύκολα αντιληπτό από έναν αριθμό πόλεων και πάνω είναι πρακτικά αδύνατο να εξεταστούν όλες οι πιθανές λύσεις μία προς μία.

Το Πρόβλημα του Περιοδεύοντος Πωλητή, στη γενική του μορφή μπορεί να περιγραφεί με το παρακάτω μαθηματικό μοντέλο [5]:

$$min \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} d_{ij} x_{ij} \tag{1}$$

$$\sum_{\substack{i=1 \\ i \neq j}}^{n} x_{ij} = 1 \text{ για } j = 1 \dots n \qquad \sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij} = 1 \text{ για } i = 1 \dots n \tag{2}$$

$$y_i - y_j + n x_{ij} \leq n - 1 \text{ για } 2 \leq i \neq j \leq n \tag{3}$$

$$x_{ij} \in \{0,1\} \text{ για } i, j = 1 \dots n \qquad y_i \in \mathbb{N}_0 \text{ για } 2 = 1 \dots n \tag{4}$$

Όπου:

n ο αριθμός των κόμβων του προβλήματος,

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

x_{ij} μεταβλητή που λαμβάνει την τιμή 1 όταν η σύνδεση μεταξύ των κόμβων i και j ανήκει στο Χαμιλτονιανό κύκλο.

Οι ισότητες (2) προβλέπουν ότι κάθε κόμβος μπορεί να εισαχθεί και να αφαιρεθεί μόνο μία φορά. Ο περιορισμός (3) προβλέπει ότι ο Χαμιλτονιανός κύκλος είναι μόνο μία διαδρομή και όχι ένα άθροισμα μικρότερων κύκλων [5].

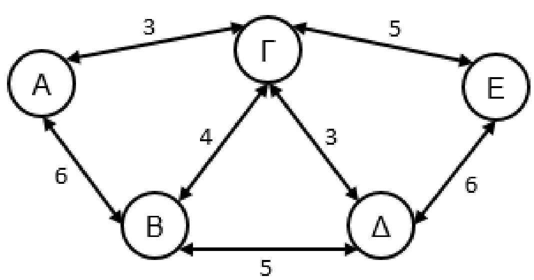
2.2. Παραλλαγές του Κλασικού Προβλήματος

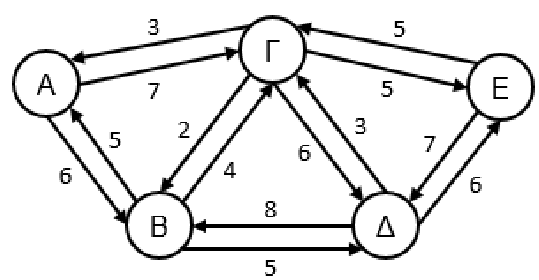
Πέρα από το κλασικό Πρόβλημα του Περιοδεύοντος Πωλητή που παρουσιάστηκε παραπάνω, υπάρχει στη διεθνή βιβλιογραφία και πλήθος άλλων προβλημάτων που αποτελούν παραλλαγή του κλασικού. Μερικές εξ' αυτών παρουσιάζονται στα υποκεφάλαια που ακολουθούν.

2.2.1. Συμμετρικό και Ασύμμετρο Πρόβλημα Περιοδεύοντος Πωλητή

Όπως έχει αναφερθεί στον ορισμό του προβλήματος, αρχικά ορίζεται μεταξύ όλων των κόμβων του προβλήματος από μία τιμή βάρους. Στο συμμετρικό Πρόβλημα του Περιοδεύοντος Πωλητή (Symmetric Travelling Salesman Problem - STSP) το κόστος μετάβασης από τον κόμβο (i) στον κόμβο (j) είναι το ίδιο με αυτό της μετάβασης από τον τον κόμβο (j) στον κόμβο (i) και σε αυτή την περίπτωση έχει ήδη αναφερθεί ότι το πλήθος όλων των πιθανών λύσεων υπολογίζεται από τη σχέση (ν-1)! / 2 όπου ν το πλήθος των πόλεων του προβλήματος. Αντίθετα, ορίζεται και το Ασύμμετρο Πρόβλημα του Περιοδεύοντος Πωλητή (Asymmetric Travelling Salesman Problem – ATSP) όπου το κόστος μετάβασης από τον κόμβο (i) στον κόμβο (j) δεν είναι το ίδιο με αυτό της μετάβασης από τον κόμβο (j) στον κόμβο (i) αλλά διαφέρει. Στο ATSP το πλήθος όλων των πιθανών λύσεων υπολογίζεται από τη σχέση (ν-1)!, υπάρχουν δηλαδή, για τον ίδιο αριθμό πόλεων ν, διπλάσιοι εφικτοί συνδυασμοί που μπορούν να μελετηθούν [6].

Στην Εικόνα 2 που ακολουθεί φαίνεται ένας γράφος με συμμετρικά βάρη μεταξύ των κόμβων του (στα αριστερά) και ένας γράφος με ασύμμετρα βάρη μεταξύ των κόμβων του (στα δεξιά).





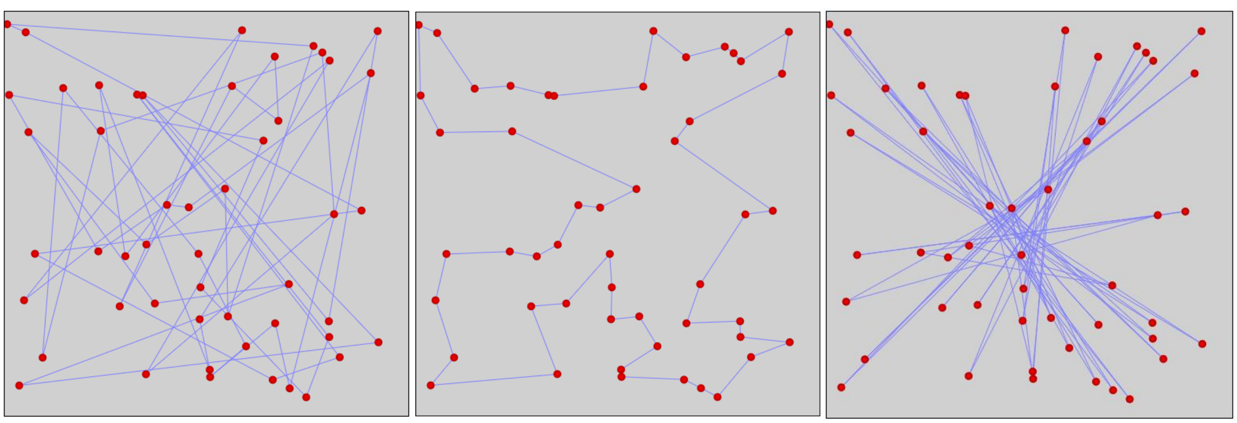
Εικόνα 2: Γράφος με συμμετρικά βάρη (αριστερά) και με ασύμμετρα βάρη (δεξιά)

2.2.2. Πρόβλημα του Περιοδεύοντος Πωλητή Μέγιστης Διασποράς

Στο Πρόβλημα του Περιοδεύοντος Πωλητή συνήθως αναζητείται η συντομότερη διαδρομή με την οποία ο πωλητής θα περάσει από όλες τις πόλεις, όμως οι περισσότερες μέθοδοι επίλυσης που προτείνονται δουλεύουν εξίσου αποτελεσματικά και στην περίπτωση που αναζητείται η μεγιστοποίηση της απόστασης που θα καλύψει ο πωλητής.

Το πρόβλημα αυτό ονομάζεται Πρόβλημα του Περιοδεύοντος Πωλητή Μέγιστης Διασποράς (Maximum Scatter TSP). Σε αυτή την περίπτωση η αντικειμενική συνάρτηση αντιστρέφεται και γίνεται αρνητική με αποτέλεσμα να αναζητείται η ελάχιστη τιμή που αυτή μπορεί να πάρει δηλαδή η μέγιστη δυνατή διαδρομή [7]. Υπάρχουν διάφορες εργασίες που εξερευνούν το πρόβλημα της μέγιστης διαδρομής. Σε μία από αυτές αναφέρεται ότι η πιο φυσική περίπτωση αυτής της κατηγορίας προβλημάτων (όπου αναζητείται η μεγιστοποίηση της απόστασης) προκύπτει για τις ευθύγραμμες αποστάσεις στο επίπεδο R^2 όπου η σφαίρα της μονάδας είναι το τετράγωνο, δηλαδή, σε γεωμετρικά προβλήματα όπου ο στόχος είναι να βρεθεί μια διαδρομή που μεγιστοποιεί το συντομότερο άκρο [8].

Στην Εικόνα 3, που ακολουθεί, φαίνεται αρχικά ένα Πρόβλημα Περιοδεύοντος Πωλητή με 50 κόμβους. Ύστερα φαίνεται η επίλυση αυτού του προβλήματος με την εύρεση της ελάχιστης διαδρομής που συνδέει αυτούς τους 50 κόμβους. Τέλος φαίνεται πώς μοιάζει μια λύση του ίδιου προβλήματος αν το ζητούμενο είναι η μεγιστοποίηση της διαδρομής που συνδέει αυτούς τους 50 κόμβους.

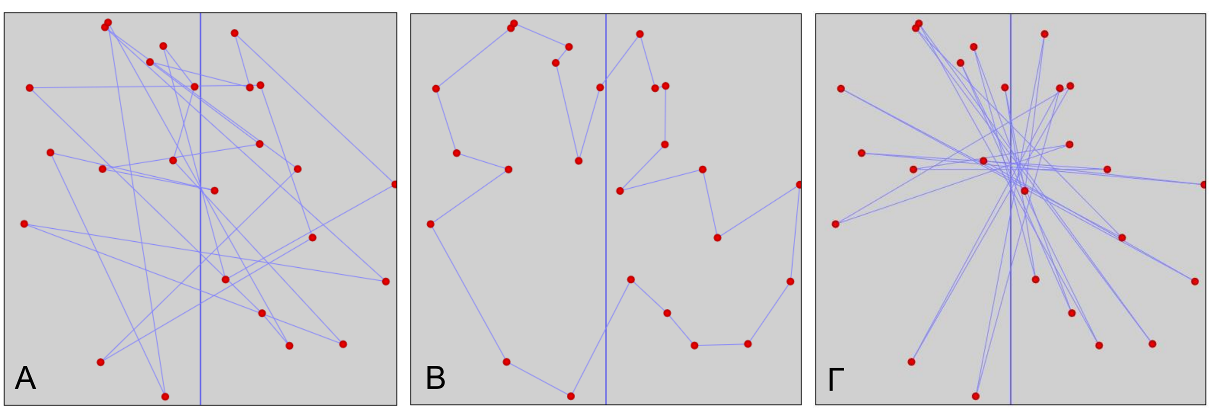


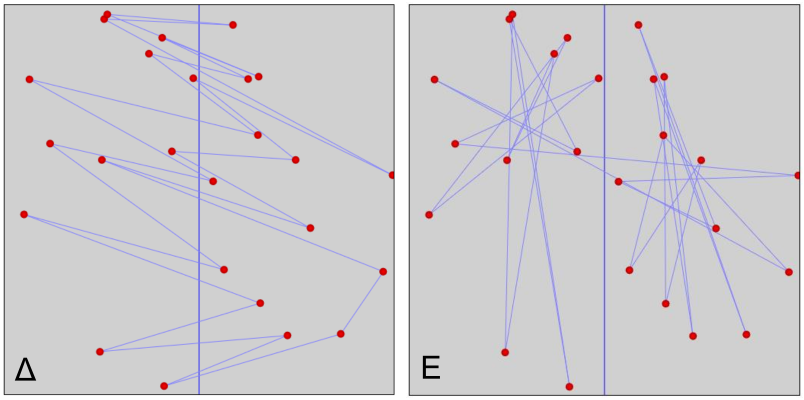
Εικόνα 3: (α) TSP 50 κόμβων, (β) Επίλυση Ελάχιστης Διαδρομής, (γ) Επίλυση Μέγιστης Διασποράς [7]

2.2.3. Πρόβλημα του Περιοδεύοντος Πωλητή Με Ποτάμι

Σε μια άλλη παραλλαγή του προβλήματος μπορεί να θεωρηθεί ότι υπάρχει ένα ποτάμι που διέρχεται από το μέσον του γράφου. Σε αυτή την περίπτωση προστίθεται μία επιπλέον παράμετρος στη συνάρτηση κόστους η οποία μπορεί να είναι είτε θετική, οπότε και υπάρχει αύξηση του κόστους της διαδρομής κάθε φορά που ο πωλητής διασχίζει το ποτάμι, είτε αρνητική οπότε και προτιμώνται λύσεις όπου το ποτάμι διασχίζεται συχνά. Στην πρώτη περίπτωση η λύση που θα προκύψει θα είναι μία διαδρομή όπου το ποτάμι διασχίζεται μόλις δύο φορές και εκατέρωθεν αυτού, στην κάθε όχθη, παρουσιάζεται από μία σχεδόν βέλτιστη διαδρομή. Στη δεύτερη περίπτωση, η λύση που θα προκύψει θα διασχίζει το ποτάμι όσο το δυνατόν πιο συχνά καθότι θα προτιμώνται λύσεις που εμπεριέχουν τη διάσχιση του ποταμού εφόσον αυτό μειώνει τη συνάρτηση κόστους. Για αυτή τη δεύτερη περίπτωση αντί για πωλητή μπορούμε να φανταστούμε έναν λαθρέμπορο που προτιμάει να διασχίζει συχνά το ποτάμι για να πουλάει τα κλοπιμαία του σε γειτονικές πόλεις που χωρίζονται από ένα ποταμό [7].

Στην Εικόνα 4 που ακολουθεί φαίνονται διάφορες επιλύσεις ενός TSP 25 κόμβων όπου ζητείται η εύρεση είτε της ελάχιστης είτε της μέγιστης διαδρομής όταν η διάσχιση ενός ποταμού είτε βελτιώνει είτε χειροτερεύει την αντικειμενική συνάρτηση του προβλήματος. Για να είναι πιο εμφανής ο αντίκτυπος της διάσχισης του ποταμού στην κάθε περίπτωση επιλέχθηκε η κάθε διάσχιση να μεταβάλλει αρκετά την αντικειμενική συνάρτηση.





Εικόνα 4: Από πάνω αριστερά προς κάτω δεξιά: (Α) TSP 25 κόμβων με ποτάμι στη μέση,
(Β) Επίλυση Ελάχιστης Διαδρομής με αύξηση κόστους όταν διασχίζεται το ποτάμι,
(Γ) Επίλυση Μέγιστης Διαδρομής με αύξηση κόστους όταν διασχίζεται το ποτάμι,
(Δ) Επίλυση Ελάχιστης Διαδρομής με μείωση κόστους όταν διασχίζεται το ποτάμι,
(Ε) Επίλυση Μέγιστης Διαδρομής με μείωση κόστους όταν διασχίζεται το ποτάμι [7]

Από τα διάφορα παραδείγματα που φαίνονται στην Εικόνα 4 γίνεται αντιληπτό ότι όταν η διάσχιση του ποταμού οδηγεί σε καλύτερη λύση (μείωση κόστους όταν ζητάμε το ολικό ελάχιστο – παράδειγμα Δ – ή αύξηση κόστους όταν αναζητούμε το ολικό μέγιστο – παράδειγμα Γ) τότε επιλέγονται ως βέλτιστες λύσεις διαδρομές που διασχίζουν το ποτάμι όσο συχνότερα γίνεται. Αντιθέτως, όταν η διάσχιση του ποταμού οδηγεί σε χειρότερη αντικειμενική συνάρτηση (αύξηση κόστους όταν ζητάμε το ολικό ελάχιστο – παράδειγμα Β – ή μείωση κόστους όταν αναζητούμε το ολικό μέγιστο – παράδειγμα Ε) τότε επιλέγονται ως βέλτιστες λύσεις διαδρομές που διασχίζουν το ποτάμι μόλις δύο φορές.

2.2.4. Πρόβλημα Περιοδεύοντος Πωλητή με Διαδοχικές Παραγγελίες

Αυτό είναι ένα ασύμμετρο πρόβλημα όπου υπάρχουν επιπροσθέτως και περιορισμοί προτεραιότητας, δηλαδή κάθε τέτοιος περιορισμός απαιτεί ότι ένας κόμβος (i) θα πρέπει να επισκεφθεί υποχρεωτικά πριν από έναν άλλον κόμβο (j) [9].

Τα προβλήματα αλληλουχίας όπως το Πρόβλημα του Περιοδεύοντος Πωλητή με Διαδοχικές Παραγγελίες (Sequential Ordering Problem – SOP) είναι από τα πλέον ευρέως μελετημένα προβλήματα στην έρευνα των επιχειρήσεων. Διάφορες παραλλαγές των προβλημάτων αλληλουχίας περιλαμβάνουν τον προγραμματισμό ενός μηχανήματος, το Πρόβλημα του Περιοδεύοντος Πωλητή με Χρονικούς Περιορισμούς και τον προγραμματισμό μηχανημάτων με περιορισμούς προτεραιότητας. Προβλήματα αλληλουχίας είναι εκείνα στα οποία πρέπει να καθοριστεί η καλύτερη σειρά εκτέλεσης μιας σειράς εργασιών, η οποία σε πολλές περιπτώσεις οδηγεί σε ένα πρόβλημα NP-δύσκολο [10]. Τα προβλήματα αλληλουχίας κυριαρχούν σε περιπτώσεις κατασκευής προϊόντων και σε εφαρμογές δρομολόγησης, συμπεριλαμβανομένων των μονάδων παραγωγής όπου οι εργασίες θα πρέπει να εκτελούνται μία-μία τη φορά σε μια γραμμή συναρμολόγησης και στις υπηρεσίες ταχυδρομείου όπου προγραμματίζεται η παράδοση των πακέτων με οχήματα. Άλλα προβλήματα βιομηχανιών που περιλαμβάνουν πολλαπλές εγκαταστάσεις μπορεί επίσης να αντιμετωπίζονται ως προβλήματα αλληλουχίας σε ορισμένα σενάρια όπου για παράδειγμα ένα μηχάνημα αποτελεί τον πιο αδύναμο κρίκο μιας μονάδας παραγωγής [11]. Οι υπάρχουσες μέθοδοι για προβλήματα αλληλουχίας ακολουθούν είτε μία ευρετική μέθοδο που απευθύνεται σε μια συγκεκριμένη κατηγορία προβλημάτων, είτε χρησιμοποιούν μια γενική μεθοδολογία επίλυσης όπως Ακέραιο Προγραμματισμό (Integer Programming) ή Προγραμματισμό Περιορισμών (Constraint Programming). Λαμβάνοντας υπόψη την πρακτική σημασία των προβλημάτων αλληλουχίας και τον υπολογιστικό φόρτο που απαιτούν για την επίλυσή τους, η κατανόηση του πώς αυτά τα προβλήματα μπορούν να επιλυθούν πιο αποτελεσματικά αποτελεί ακόμα και σήμερα ένα ενεργό πεδίο έρευνας.

Οι A.A. Cire και W.-J. van Hoeve, στην εργασία τους [12] προτείνουν μια νέα προσέγγιση για την επίλυση προβλημάτων αλληλουχίας βασισμένη σε διαγράμματα αποφάσεων πολλαπλών τιμών (Multivalued Decision Diagrams – MDD). Τα διαγράμματα αυτά είναι γραφικές παραστάσεις των συναρτήσεων Boolean, που εισήχθησαν αρχικά για εφαρμογές στον σχεδιασμό κυκλωμάτων από τον Lee το 1959 [13], και έχουν μελετηθεί και εφαρμόζονται ευρέως στην επιστήμη των υπολογιστών. Έχουν χρησιμοποιηθεί πρόσφατα για να αντιπροσωπεύσουν το εφικτό σύνολο διακριτών προβλημάτων βελτιστοποίησης, όπως καταδεικνύεται από τις εργασίες των Becker, Behle, Eisenbrand & Wimmer [14] και των Bergman, Cire, Hoeve & Hooker [15]. Αυτό γίνεται με την αναπαράσταση των περιορισμών ενός προβλήματος ως μια Boolean συνάρτηση f(x) που δείχνει εάν μια λύση x είναι εφικτή. Παρόλα αυτά, τέτοια διαγράμματα μπορούν να αυξηθούν εκθετικά, κάτι που καθιστά γενικά δύσκολη την πρακτική επίλυση προβλημάτων χρησιμοποιώντας τα. Για να ξεπεραστεί αυτό το πρόβλημα, οι Andersen, Hadzic, Hooker & Tiedemann [16] εισαγάγανε την έννοια των χαλαρών διαγραμμάτων MDD, τα οποία είναι διαγράμματα περιορισμένου μεγέθους που αντιπροσωπεύουν μια υπέρ-προσέγγιση της ομάδας των εφικτών λύσεων του προβλήματος.

Οι Cire και Hoeve, στην εργασία τους υποστηρίζουν ότι τέτοια MDD μπορεί να είναι ιδιαίτερα χρήσιμα ως απλοποίηση του συνόλου των εφικτών λύσεων προβλημάτων αλληλουχίας. Ειδικότερα, ένα χαλαρό MDD μπορεί να είναι ενσωματωμένο σε μια πλήρη διαδικασία αναζήτησης, όπως αυτή της μεθόδου Branch and Bound στον Ακέραιο Προγραμματισμό ή να αποτελεί υποβοήθημα στην αναζήτηση του Προγραμματισμού Περιορισμών [16], [17]. Εστιάζουν, ακόμα, σε μια ευρεία κατηγορία προβλημάτων αλληλουχίας, όπου η σειρά των εργασιών που απαιτούνται να γίνουν πρέπει να προγραμματιστούν σε ένα μόνο μηχάνημα και υπόκεινται σε περιορισμούς προτεραιότητας και άλλους χρονικούς περιορισμούς όπου μπορεί να δίνονται και συγκεκριμένοι χρόνοι εκτέλεσης των εργασιών. Γενικεύουν, λοιπόν, μια σειρά προβλημάτων προγραμματισμού ενός μηχανήματος και άλλες παραλλαγές του Προβλήματος του Περιοδεύοντος Πωλητή. Η χαλάρωση που παρέχεται από το MDD, ωστόσο, είναι κατάλληλη για οποιοδήποτε πρόβλημα όπου η λύση καθορίζεται από μια μεταβολή ενός ορισμένου αριθμού εργασιών και δεν εξαρτάται άμεσα από συγκεκριμένους περιορισμούς ή από την αντικειμενική συνάρτηση.

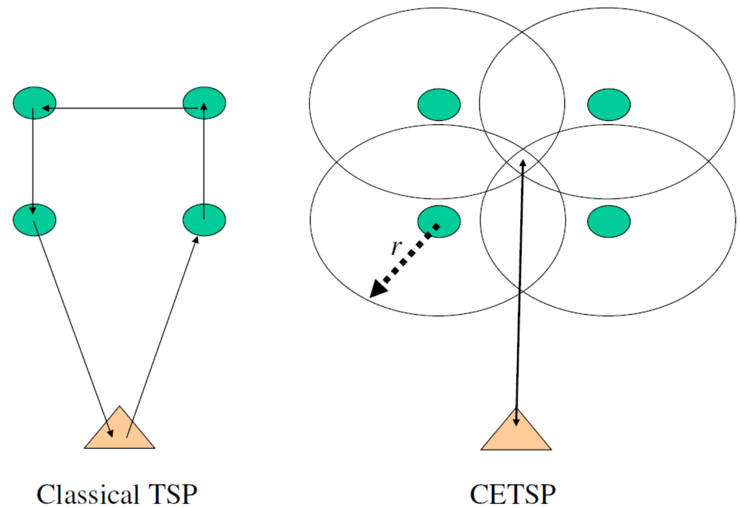
Προτείνουν επίσης μια σειρά τεχνικών για την ενίσχυση των διαγραμμάτων MDD, οι οποίες λαμβάνουν υπόψη τους περιορισμούς προτεραιότητας και χρονικών περιθωρίων. Παρουσιάζουν στην εργασία τους ότι αυτές οι γενικές τεχνικές μπορούν να χρησιμοποιηθούν για την εξαγωγή ενός αλγορίθμου πολυωνυμικού χρόνου για μια συγκεκριμένη παραλλαγή του TSP που εισήχθη από τον Balas το 1999 [18] δείχνοντας ότι το σχετικό διάγραμμα MDD έχει πολυωνυμικό μέγεθος. Για να καταδείξουν τη χρήση χαλαρών MDD στην πράξη, εφαρμόζουν τις τεχνικές τους στην χρονοδρομολόγηση [19]. Αυτή παίζει κεντρικό ρόλο ως γενική μεθοδολογία σε πολύπλοκα και μεγάλης κλίμακας προβλήματα προγραμματισμού. Παραδείγματα εμπορικών εφαρμογών που εφαρμόζουν αυτή τη μεθοδολογία περιλαμβάνουν τον προγραμματισμό του ναυπηγείου της

Σιγκαπούρης, την οργάνωση των πυλών αφίξεων στον αερολιμένα του Χονγκ Κονγκ [20], τον σχεδιασμό του συστήματος των αγωγών πετρελαίου στη Βραζιλία [21] και την οργάνωση του συστήματος υγειονομικής περίθαλψης στο σπίτι [22]. Δείχνουν ότι, χρησιμοποιώντας τις τεχνικές των MDD που περιγράφονται στην εργασία τους, μπορούν να βελτιώσουν την απόδοση των πιο σύγχρονων συστημάτων χρονοπρογραμματισμών, βασισμένων σε περιορισμούς κατά τάξη μεγέθους, σε προβλήματα μίας μηχανής χωρίς να χάσουν τη γενικότητα της μεθόδου. Επιπρόσθετα, μπόρεσαν να κλείσουν τρία παραδείγματα του TSPLIB που παρέμεναν ανοιχτά, σχετικά με το πρόβλημα αλληλουχίας [23].

2.2.5. «Αρκετά Κοντά» Πρόβλημα του Περιοδεύοντος Πωλητή

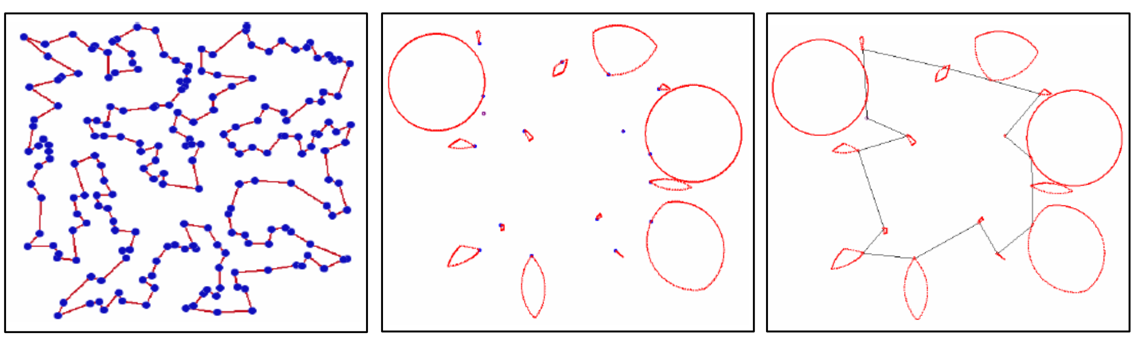
Στην παραλλαγή «Αρκετά Κοντά» Πρόβλημα του Περιοδεύοντος Πωλητή (Close Enough Travelling Salesman Problem – CETSP) ο πωλητής δε χρειάζεται να επισκεφτεί συγκεκριμένα τον κάθε κόμβο αλλά αρκεί να βρεθεί από τον καθένα σε μια απόσταση μικρότερη από μια δοσμένη τιμή r. Σημειώνεται ότι η ακτίνα r δε χρειάζεται να είναι η ίδια για τον κάθε κόμβο.

Στην Εικόνα 5 που ακολουθεί φαίνεται πώς το κλασικό TSP μπορεί να μετασχηματιστεί σε CETSP αν μας δοθεί μία περιοχή ακτίνας r γύρω από τον κάθε κόμβο του προβλήματος, εντός της οποίας αρκεί να βρεθεί ο πωλητής ώστε να είναι σε θέση να προσεγγίσει τους πελάτες του. Στην περίπτωση της εικόνας, αρκεί ο πωλητής να βρεθεί στην περιοχή τομής των τεσσάρων περιοχών ακτίνας r ώστε να θεωρηθεί ότι, με μία στάση, έχει επισκεφτεί και τους 4 πελάτες.



Εικόνα 5: Επίλυση κλασικού TSP (αριστερά) και πώς απλοποιείται η λύση στο CETSP (δεξιά) [24]

Στην Εικόνα 6 που ακολουθεί φαίνεται μια γενίκευση της προηγούμενης εικόνας όπου σε ένα TSP 200 κόμβων, αντί να βρούμε τη βέλτιστη λύση του (στα αριστερά της εικόνας), υπολογίζουμε μία ακτίνα r_i γύρω από τον κάθε κόμβο του προβλήματος και έτσι οδηγούμαστε σε ένα πρόβλημα όπου υπάρχουν 16 περιοχές (στη μέση της εικόνας) εντός των οποίων αν βρεθεί ο πωλητής θα έχει προσεγγίσει όλους τους υποψήφιους πελάτες του. Επιλύοντας αυτό το πρόβλημα οδηγούμαστε σε μία λύση όπως αυτή που φαίνεται στα δεξιά της εικόνας.



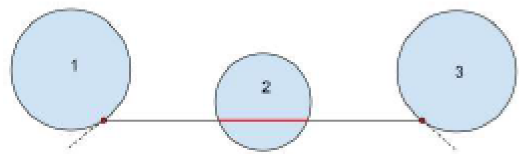
Εικόνα 6: Από αριστερά προς τα δεξιά: (α) Βέλτιση λύση ενός TSP 200 κόμβων (συνολική απόσταση 1074.4 χλμ), (β) Ορισμός προβλήματος CETSP, (γ) Τελική λύση του CETSP (συνολική απόσταση 312.3 χλμ) [24]

Το πρόβλημα CETSP είναι μια γενίκευση του TSP και μπορεί να επιλυθεί, σύμφωνα με τον Mennell [25], ως ιδιαίτερη περίπτωση τριών άλλων προβλημάτων που σχετίζονται με το TSP: Το Πρόβλημα του Περιοδεύοντος Πωλητή με Περιοχές (Travelling Salesman Problem with Neighborhoods - TSPN) [26], το Πρόβλημα του Γενικευμένου

Περιοδεύοντος Πωλητή (Generalized Travelling Salesman Problem – GTSP) [27] και το Πρόβλημα Κάλυψης Περιήγησης (Covering Tour Problem – CTP) [28]. Όπως ήδη αναφέρθηκε, στο CETSP, αντί ο πωλητής να επισκεφτεί τον κάθε πελάτη/κόμβο ξεχωριστά αρκεί να επισκεφθεί μια συγκεκριμένη περιοχή που περιέχει έναν ή περισσότερους πελάτες/κόμβους. Στην εργασία τους οι Pereira Coutinho, Subramanian, Quirino do Nascimento & Alves Pessoa [29] υποθέτουν ότι οι περιοχές κάλυψης που σχηματίζονται γύρω από τους κόμβους του προβλήματος είναι κύκλοι. Αυτή είναι άλλωστε η πιο κλασική υπόθεση στην βιβλιογραφία του CETSP. Επομένως η κάθε περιοχή $i \in V = \{1,..., n\}$ θεωρείται ότι καλύπτεται και ο πωλητής περνάει από το δίσκο D_i με ακτίνα r_i εαν η διαδρομή που θα ακολουθήσει ο πωλητής περιέχει την κορυφή i ή τουλάχιστον αγγίζει τα όρια αυτού του δίσκου. Σε ένα τρισδιάστατο χώρο οι περιοχές θεωρούνται σφαίρες και με την ίδια έννοια οι κόμβοι θεωρείται ότι είναι καλυμμένοι αν ο πωλητής περνάει ή τουλάχιστον αγγίζει τα όρια των αντίστοιχων σφαιρών κάλυψης.

Το CETSP μπορεί να περιγραφεί μαθηματικά ως εξής: Έχουμε αρχικά ένα σύνολο κόμβων $V = \{0,. . . , n\}$ σε ένα δισδιάστατο χώρο και τις συντεταγμένες τους $(\bar{x}_i,\ \bar{y}_i)$, i = 0,. . . , n. Κάθε κόμβος i καλύπτεται από ένα κύκλο D_i με ακτίνα r_i. Υποθέτουμε ότι $(\bar{x}_i,\ \bar{y}_i) \neq (\bar{x}_j,\ \bar{y}_j)$, $\forall$ i, j $\in$ V, i $\neq$ j, δηλαδή δεν υπάρχει αλληλεπικάλυψη μεταξύ των κόμβων. Το πρόβλημα έγκειται στον καθορισμό της τιμής των συντεταγμένων των σημείων πρόσκρουσης $(x_i, y_i) \in R^2$, i = 0,. . . , n, δηλαδή των αντιπροσωπευτικών σημείων για κάθε περιοχή [25], και μια ακολουθία $S = \{k_0, k_1,. . . , k_n\}$ τέτοια ώστε η περιήγηση πάνω από αυτές τις συντεταγμένες να σχηματίζει έναν Χαμιλτονιανό κύκλο ελάχιστου μήκους και $(x_i, y_i) \in D_i$, $\forall$ i $\in$ V.

Στο παράδειγμα της επόμενης εικόνας (Εικόνα 7), η διαδρομή "χτυπά" τον κύκλο που σχετίζεται με τους κόμβους 1 και 3 σε ένα μόνο σημείο, ενώ για τον κόμβο 2 η περιήγηση "χτυπά" τον συσχετισμένο κύκλο της σε άπειρο αριθμό σημείων. Εδώ ορίζουμε ότι ο κόμβος i = 0 αντιπροσωπεύει την αποθήκη και υποθέτουμε ότι D_0 είναι ένας κύκλος με $r_0 = 0$. Αυτός ο ορισμός μπορεί εύκολα να επεκταθεί στον τρισδιάστατο χώρο χρησιμοποιώντας τρεις συντεταγμένες, δηλαδή $(\bar{x}_i, \bar{y}_i, \bar{z}_i)$, αντί για δύο [29].



Εικόνα 7: Παραδείγματα σημείων πρόσκρουσης [29]

2.2.6. Πρόβλημα του Περιοδεύοντος Πωλητή με Χρονικούς Περιορισμούς

Όπως και στο κλασικό Πρόβλημα του Περιοδεύοντος Πωλητή (TSP), το Πρόβλημα του Περιοδεύοντος Πωλητή με Χρονικούς Περιορισμούς (Travelling Salesman Problem with Time Windows – TSPTW) απαιτεί τον προσδιορισμό ενός Χαμιλτονιανού κύκλου μέσα από ένα γράφημα κόμβων, αλλά προσθέτει την απαίτηση ότι κάθε κόμβος πρέπει να επισκεφθεί μέσα σε ένα προκαθορισμένο χρονικό περιθώριο. Για τις εφαρμογές δρομολόγησης, το TSPTW αντιπροσωπεύει το πρόβλημα της εξεύρεσης αποτελεσματικής διαδρομής, ξεκινώντας και τελειώνοντας σε μια συγκεκριμένη αποθήκη, η οποία επισκέπτεται ένα σύνολο πελατών, τον καθένα στο προκαθορισμένο χρονικό παράθυρό του. Στο περιβάλλον προγραμματισμού ενός μηχανήματος, το TSPTW μπορεί να χρησιμοποιηθεί για να μοντελοποιήσει το πρόβλημα συνεχόμενων εργασιών σε ένα μόνο μηχάνημα, όπου ο χρόνος διεκπεραίωσης κάθε εργασίας εξαρτάται από την προηγούμενη εργασία και κάθε εργασία έχει ένα χρόνο εντός του οποίου πρέπει να αρχίσει και έναν χρόνο εντός του οποίου πρέπει να ολοκληρωθεί. Στο πλαίσιο δρομολόγησης, ο στόχος του TSPTW είναι να ελαχιστοποιήσει το χρόνο του συνολικού ταξιδιού. Στο πλαίσιο του προγραμματισμού ενός μηχανήματος, ο στόχος του TSPTW είναι να ελαχιστοποιήσει το νεκρό χρόνο λειτουργίας του μηχανήματος. Και οι δύο εφαρμογές του TSPTW έχει αποδειχθεί ότι είναι NP-δύσκολες, και ακόμη και η εύρεση εφικτών λύσεων είναι ένα NP-πλήρες πρόβλημα [30]. Για ευκολία διαχωρισμού των δύο προβλημάτων γίνεται αναφορά στο TSPTW στα πλαίσια δρομολόγησης οχήματος ως TSPTW-TT (Travel Time) και στο TSPTW για τον προγραμματισμό ενός μηχανήματος ως TSPTW-M (Makespan) [31].

2.2.7. Πρόβλημα Πολλαπλών Περιοδεύοντων Πωλητών

Το Πρόβλημα Πολλαπλών Περιοδεύοντων Πωλητών (multiple Travelling Salesman Problem – mTSP) αποτελεί μια γενίκευση του κλασικού TSP και ταυτόχρονα μια απλοποίηση του Προβλήματος Δρομολόγησης Οχήματος (Vehicle Routing Planning – VRP), όπου περισσότεροι του ενός πωλητές (η «χωρητικότητα» του καθενός θεωρείται άπειρη) επιτρέπονται ανά πάσα στιγμή. Δίνεται, αρχικά, μια λίστα πόλεων και μια αρχική αποθήκη από όπου ξεκινάνε και καταλήγουν όλοι οι πωλητές. Στόχος του προβλήματος είναι να προσδιοριστεί μια διαδρομή για τον κάθε πωλητή, τέτοια ώστε η συνολική διαδρομή που θα καλύψουν όλοι οι πωλητές να ελαχιστοποιηθεί και κάθε πόλη να την επισκεφτεί ακριβώς μια φορά, ένας μόνο πωλητής. Έχοντας έναν μόνο πωλητή (m=1) τότε επί της ουσίας επιστρέφουμε στο κλασικό TSP.

Το Πρόβλημα των Πολλαπλών Περιοδεύοντων Πωλητών μπορεί να περιγραφεί μαθηματικά με τις ακόλουθες εξισώσεις [32]:

Θεωρούμε ένα γράφημα G = (V, A), όπου V είναι το σύνολο n κόμβων, και A είναι το σύνολο των ακμών. Συνδεδεμένο με κάθε ακμή (i, j) ∈ A είναι ένα κόστος (ή μία απόσταση) c_{ij}. Υποθέτουμε ότι η αποθήκη είναι ο κόμβος 1 και ότι υπάρχουν m εμπορικοί αντιπρόσωποι στην αποθήκη. Ορίζουμε μια δυαδική μεταβλητή x_{ij} για κάθε ακμή (i, j) ∈ A τέτοια ώστε το x_{ij} να παίρνει την τιμή 1 εάν η ακμή (i, j) περιλαμβάνεται σε μια διαδρομή ενός πωλητή ενώ το x_{ij} θα παίρνει την τιμή 0 σε διαφορετική περίπτωση. Για τους περιορισμούς απόρριψης μιας υποδιαδρομής, ορίζουμε μια ακέραια μεταβλητή u_i για να δηλώσουμε τη θέση του κόμβου i σε μια περιήγηση και ορίζουμε μια τιμή p ως τον μέγιστο αριθμό κόμβων που μπορεί να επισκεφθεί κάποιος πωλητής.

Αντικειμενική συνάρτηση:

Ελαχιστοποίηση της εξίσωσης:

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

Περιορισμοί:

Εξασφάλιση ότι ακριβώς m πωλητές αναχωρούν από τον κομβο 1:

$$\sum_{j \in V:(1,j) \in A} x_{1j} = m,$$

Εξασφάλιση ότι ακριβώς m πωλητές επιστρέφουν στον κομβο 1:

$$\sum_{j \in V:(j,1) \in A} x_{j1} = m,$$

Εξασφάλιση ότι μια συγκεκριμένη διαδρομή διέρχεται από κάθε κόμβο:

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

Εξασφάλιση ότι ακριβώς μία διαδρομή εξέρχεται από τον κάθε κόμβο:

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

Απόρριψη μιας υποδιαδρομής όταν ισχύει:

$$ui - uj + p \cdot xij \leq p - 1, \; \forall \, 2 \leq i \neq j \leq n$$

Στη διεθνή βιβλιογραφία υπάρχουν διάφορες εναλλακτικές μαθηματικές διατυπώσεις του προβλήματος όπως για παράδειγμα των Laport & Nobert [33] και των Christofides, Mingozzi & Toth [34]. Για την επίλυση του προβλήματος γραμμικού προγραμματισμού που παρουσιάστηκε παραπάνω μπορεί να χρησιμοποιηθεί ένας από τους NEOS Server Solvers στην κατηγορία του Μεικτού Ακέραιου Γραμμικού Προγραμματισμού [32].

2.2.8. Πρόβλημα του Περιοδεύοντος Πωλητή με Σημεία Συμφόρησης

Το Πρόβλημα του Περιοδεύοντος Πωλητή με Σημεία Συμφόρησης αναφέρεται στη διεθνή βιβλιογραφία ως Bottleneck Travelling Salesman Problem (Bottleneck TSP). Σε αυτό το πρόβλημα, αντί να αναζητά κανείς το Χαμιλτονιανό κύκλο με την ελάχιστη συνολική απόσταση, ψάχνει η μακρύτερη ακμή της διαδρομής να γίνει όσο συντομότερη γίνεται. Το Πρόβλημα του Περιοδεύοντος Πωλητή με Σημεία Συμφόρησης μπορεί να επιλυθεί ως μία ακολουθία προβλημάτων TSP. Μπορεί κανείς να παρατηρήσει ότι στην αντικειμενική συνάρτηση αυτού του προβλήματος οι ακριβείς τιμές των αποστάσεων δεν παρουσιάζουν ενδιαφέρον αλλά μόνο η σχετική σειρά τους έχει σημασία. Επομένως μπορεί να υποτεθεί ότι υπάρχουν το πολύ ½n(n - 1) διαφορετικές ολοκληρωμένες αποστάσεις και ότι η μεγαλύτερη από αυτές δεν είναι μεγαλύτερη από ½n(n - 1). Σε αυτή την περίπτωση λύνεται το πρόβλημα του παρακάτω τύπου για κάποια παράμετρο b: Είναι το γράφημα, που αποτελείται από όλες τις ακμές με βάρος το πολύ b, Χαμιλτονιανό; Αυτό είναι ακριβώς το πρόβλημα που συζητήθηκε παραπάνω.

Εκτελώντας μια δυαδική αναζήτηση στην παράμετρο b (ξεκινώντας, π.χ., με b = ¼n (n - 1)) μπορεί να προσδιοριστεί το μικρότερο τέτοιο b που οδηγεί σε μια απάντηση "ναι" επιλύοντας το πολύ O(log n) προβλήματα TSP. Τα υπολογιστικά αποτελέσματα για το TSP με Σημεία Συμφόρησης αναφέρονται στην εργασία των [35]. Έχει δειχθεί ότι μια ποικιλία σχετικών προβλημάτων μπορεί να μετατραπεί σε TSP. Ωστόσο, κάθε τέτοιος μετασχηματισμός πρέπει να εξεταστεί με κάποια προσοχή, πριν προσπαθήσει κανείς να τον χρησιμοποιήσει για πρακτική επίλυση προβλημάτων.

Για παράδειγμα, οι υπολογισμοί της συντομότερης διαδρομής που είναι απαραίτητοι για την επεξεργασία ενός γραφικού TSP ως TSP απαιτούν χρόνο $O(n^3)$ που μπορεί να μην είναι αποδεκτός στην πράξη. Πολλοί μετασχηματισμοί απαιτούν την εισαγωγή ενός μεγάλου αριθμού M. Αυτό μπορεί να οδηγήσει σε αριθμητικά προβλήματα ή μπορεί ακόμη και να αποτρέψει την εύρεση εφικτών λύσεων με τη χρήση ευρετικών αλγορίθμων.

Ειδικότερα, για τις προσεγγίσεις που βασίζονται σε μεθόδους γραμμικού προγραμματισμού, η χρήση του "μεγάλου M" δεν ενδείκνυται. Σε αυτό το πρόβλημα είναι προτιμότερο να χρησιμοποιηθούν τεχνικές μεταβλητού καθορισμού (variable fixing techniques) ώστε να προωθηθούν, ως μέρος τη λύσης, οι ακμές με κόστος -M και να αποφευχθούν όσες λύσεις εμπεριέχουν το κόστος M. Επιπλέον, γενικά, οι μετασχηματισμοί που περιγράφηκαν παραπάνω μπορούν να παράγουν περιπτώσεις TSP που είναι δύσκολο να λυθούν τόσο από τους ευρετικούς όσο και από τους ακριβείς αλγόριθμους [36].

2.2.9. Πρόβλημα Περιοδεύοντος Πωλητή με Παραλαβή και Παράδοση Πολλαπλών Εμπορευμάτων

Πολλές πρακτικές εφαρμογές στις μεταφορές περιλαμβάνουν τη βελτιστοποίηση προβλημάτων δρομολόγησης και παράδοσης εμπορευμάτων. Στο Πρόβλημα Περιοδεύοντος Πωλητή με Παραλαβή και Παράδοση Πολλαπλών Εμπορευμάτων (multi-commodity Pickup-and-Delivery Travelling Salesman Problem – m-PDTSP) δίνεται ενας συγκεκριμένος αριθμός πόλεων και το κόστος μεταφοράς από τη μία πόλη στην άλλη ενώ το δίκτυο των πόλεων δε θεωρείται ότι είναι συμμετρικό. Μία από τις πόλεις θεωρείται η αποθήκη ενώ οι υπόλοιπες προσδιορίζονται ως πελάτες. Κάθε ένας από αυτούς τους πελάτες χρειάζεται συγκεκριμένη ποσότητα από ένα εμπόρευμα. Κάθε μονάδα εμπορεύματος που ανακτάται από τον ένα πελάτη μπορεί να διατεθεί σε οποιονδήποτε άλλο πελάτη το χρειάζεται. Θεωρείται ότι το κάθε όχημα έχει συγκεκριμένη χωρητικότητα και πρέπει να αρχίσει και να τελειώσει τη διαδρομή του στην αποθήκη. Η διαδρομή που θα υπολογιστεί πρέπει να είναι Χαμιλτονιανή και να περνάει από όλες τις πόλεις του προβλήματος. Το πρόβλημα Παραλαβής και Παράδοσης Πολλαπλών Εμπορευμάτων αναζητά τη διαδρομή εκείνη που πρέπει να ακολουθήσει το όχημα ώστε οι παραλαβές και οι παραδόσεις όλων των ποσοτήτων των προϊόντων να να μην ξεπερνούν τη χωρητικότητα του οχήματος και να ελαχιστοποιούν το συνολικό κόστος διαδρομής. Αφού η κάθε πόλη επισκέπτεται μόνο μία φορά, κάθε μονάδα προϊόντος που φορτώνεται στο όχημα μεταφοράς μένει σε αυτό έως ότου παραδοθεί στον προορισμό της.

Το αρχικό φορτίο οποιουδήποτε προϊόντος που υπάρχει στο όχημα όταν αυτό ξεκινάει από την αποθήκη είναι άγνωστο και πρέπει να προσδιοριστεί μέσω της βελτιστοποίησης του προβλήματος. Η παραλαγή του m-PDTSP, όπου το αρχικό φορτίο του οχήματος όταν αυτό φεύγει από την αποθήκη είναι γνωστό, μπορεί επίσης να επιλυθεί με την προσέγγιση των Hernandez-Perez & Salazar-Gonzalez με μερικές προσαρμογές.

Άλλο πρόβλημα στενά συνδεδεμένο με το m-PDTSP το οποίο περιλαμβάνει πολλές πηγές και πολλούς προορισμούς με διάφορα προϊόντα είναι το μη προληπτικό ικανοποιημένο πρόβλημα ανταλλαγής (Non-Preemptive Capacitated Swapping Problem - NCSP). Στο NCSP υπάρχει μια αποθήκη και ένα σύνολο πελατών. Κάθε πελάτης μπορεί να παρέχει ένα είδος προϊόντος ή / και να ζητάει ένα άλλο, διαφορετικό προϊόν. Κάθε μονάδα προϊόντος θεωρείται ότι έχει κάποιο βάρος. Όταν ένας πελάτης προμηθεύει και ζητάει μια μονάδα από το ίδιο προϊόν τότε ο πελάτης αυτός ονομάζεται πελάτης μεταφόρτωσης. Τα προϊόντα που μεταφέρονται δεν μπορούν να χωριστούν σε μικρότερα και δεν μπορούν να ξεφορτωθούν σε έναν ενδιάμεσο πελάτη. Ωστόσο, μπορεί να χρησιμοποιηθεί ένας πελάτης μεταφόρτωσης ως ενδιάμεσος πελάτης για το προϊόν που προμήθευσε και ζήτησε αυτός ο πελάτης. Το πρόβλημα συνίσταται στην εξεύρεση της διαδρομής εκείνης ελάχιστου κόστους για την οποία ικανοποιούνται όλοι οι πελάτες. Διαφορές αυτού του προβλήματος σε σχέση με το m-PDTSP είναι: (1) μια εφικτή λύση μπορεί να μην είναι μία Χαμιλτονιανή διαδρομή, (2) ο κάθε πελάτης μπορεί να προμηθεύσει και / ή να ζητήσει μόνο μία μονάδα ενός προϊόντος, (3) υπάρχουν πελάτες μεταφόρτωσης και (4) κάθε εμπόρευμα έχει βάρος. Οι Bordenave, Gendreau & Laporte [37] περιγράψανε έναν αλγόριθμο Branch and Bound για να λύσουνε τη συγκεκριμένη περίπτωση του NCSP όπου η χωρητικότητα του οχήματος είναι ίση με μία μονάδα και όλα τα βάρη των στοιχείων είναι επίσης μία μονάδα [38].

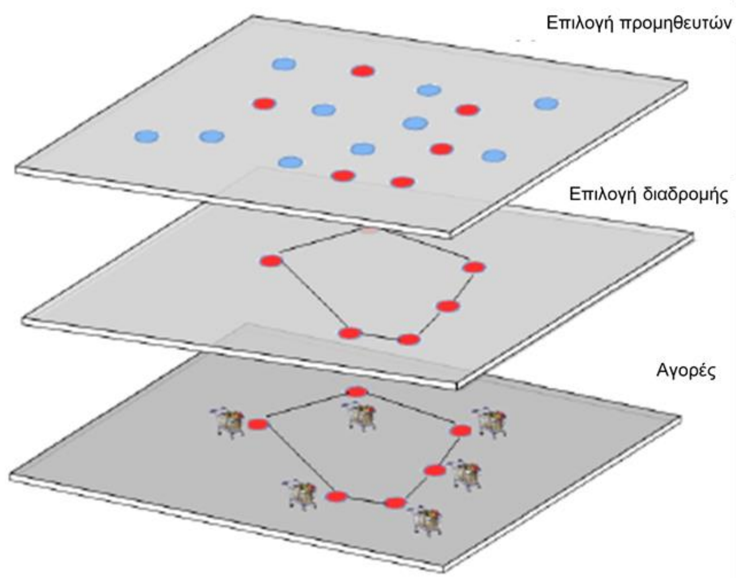
Οι Gouveia & Ruthmair [39], στην εργασία τους εξετάσανε το πρόβλημα ένα-προς-ένα πολλαπλά εμπορεύματα και παραλαβή εμπορευμάτων (m-PDTSP). Δείξανε ότι αυτό το m-PDTSP είναι ισοδύναμο με το 1-PDTSP (μια διαφορετική παραλλαγή των προβλημάτων παραλαβής και παράδοσης όπου θεωρείται ότι υπάρχει μόνο ένα προϊόν) αλλά με πρόσθετους περιορισμούς προτεραιότητας που ορίζονται από τα ζεύγη πηγής-προορισμού κάθε προϊόντος και έχουν εκμεταλλευτεί αυτή τη σχέση για να παρέχουν μοντέλα για το m-PDTSP που κατασκευάζονται συνδυάζοντας δύο διαφορετικά στοιχεία μοντελοποίησης: ένα μοντέλο ροής με περιορισμούς χωρητικότητας και ένα μοντέλο με σχέσεις προτεραιότητας. Όσον αφορά τη συνιστώσα της σχέσης προτεραιότητας, εισήγαγαν επίσης νέες ανισότητες με βάση τις ακολουθίες και τις λογικές συνέπειες των σχέσεων προτεραιότητας που είναι σε θέση να ενισχύσουν σημαντικά τα όρια LP, ειδικά για περιπτώσεις με μεγάλο αριθμό περιορισμών προτεραιότητας. Για το στοιχείο περιορισμού της χωρητικότητας, παρουσιάσανε επίσης εναλλακτικούς τρόπους για να μοντελοποιηθούν οι περιορισμοί χωρητικότητας με βάση γραφήματα διαφορετικών επιπέδων που εξαρτώνται από το φορτίο και τα οποία είναι επωφελή για μικρές χωρητικότητες οχημάτων όσον αφορά τα όρια LP.

Πολλές παραλλαγές ενός αλγόριθμου Branch and Bound αναπτύχθηκαν με βάση τα παρουσιαζόμενα μοντέλα. Αυτές οι προσεγγίσεις συνδυάστηκαν με αρκετές μεθόδους προεπεξεργασίας, ευρετικές και διαχωριστικές ρουτίνες για τις ανισότητες του Προβλήματος Διαδοχικής Παραγγελίας (SOP). Ειδικά για οχήματα περιορισμένης χωρητικότητας με μεγάλο αριθμό μεταφερόμενων προϊόντων, οι Gouveia & Ruthmair είναι σε θέση στην εργασία τους να ξεπεράσουν τις προσεγγίσεις των Hernandez-Perez & Salazar-Gonzalez [38]. Επιπλέον, εξετάσανε το m-PDTSP άπειρης χωρητικότητας το οποίο είναι ισοδύναμο με το TSP με Περιορισμούς Προτεραιότητας (ή Πρόβλημα με Διαδοχικές Παραγγελίες όπως αναφέρθηκε παραπάνω). Στην εργασία τους, που αποτελεί μια προσαρμοσμένη παραλλαγή του αλγορίθμου Branch & Bound που είχαν αναπτύξει παλιότερα οι ίδιοι, είναι σε θέση να βρουν τη βέλτιστη λύση αρκετών, τότε, ανοιχτών περιπτώσεων από το TSPLIB και το SOPLIB [39].

2.2.10. Πρόβλημα του Περιοδεύοντος Αγοραστή

Το Πρόβλημα του Περιοδεύοντος Αγοραστή (Travelling Purchaser Problem - TPP) είναι ένα πρόβλημα δρομολόγησης και προμηθειών ενός οχήματος και μπορεί να περιεγραφεί ως εξής: Θεωρούμε μια αποθήκη 0, ένα σύνολο K προϊόντων/αντικειμένων προς αγορά και ένα σύνολο M γεωγραφικά διασκορπισμένων προμηθευτών/αγορών για να διαλέξουμε. Μια διακριτή ζήτηση d_k καθορίζεται για κάθε προϊόν $k \in K$, το οποίο με τη σειρά του μπορεί να αγοραστεί σε ένα $M_k \subseteq M$ σε τιμή $p_{ik} > 0$, όπου $i \in M_k$. Επιπλέον, η διαθεσιμότητα του προϊόντος $q_{ik} > 0$ είναι επίσης γνωστή για κάθε προϊόν $k \in K$ και κάθε προμηθευτή $i \in M_k$. Σημειώνεται επίσης ότι, για να διασφαλιστεί η ύπαρξη ενός εφικτού σχεδίου αγοράς σε σχέση με τη ζήτηση του κάθε προϊόντος, πρέπει να πληρείται η συνθήκη $\sum_{i \epsilon M_k} q_{ik} \geq d_k, \forall \mathrm{k} \in \mathrm{K}$. Το πρόβλημα προσδιορίζεται από ένα πλήρες κατευθυνόμενο γράφημα G = (V, A) όπου $V \coloneqq M \cup \{0\}$ είναι οι κόμβοι του προβλήματος και $A \coloneqq \{(i, j) : i, j \,\epsilon\, V, i \neq j\}$ είναι οι ακμές του προβλήματος. Ένα κόστος μετακίνησης c_{ij} συνδέεται με κάθε ακμή $(i, j) \,\epsilon\, A$. Το TPP αναζητά μια απλή περιήγηση στο G ξεκινώντας και τελειώνοντας στην αποθήκη, επισκέπτοντας ένα υποσύνολο προμηθευτών και αποφασίζοντας πόσα κομμάτια να αγοραστούν από το κάθε προϊόν από τον κάθε προμηθευτή έτσι ώστε να ικανοποιηθεί η ζήτηση με τα ελάχιστα δυνατά κόστη ταξιδιού και αγοραπωλησιών.

Το μεγάλο ενδιαφέρον γύρω από το Πρόβλημα του Περιοδεύοντος Αγοραστή οφείλεται πιθανότατα στο γεγονός ότι συνδυάζει αριστοτεχνικά την επιλογή προμηθευτή, την κατασκευή δρομολογίου που πρέπει να ακολουθηθεί και τον προγραμματισμό αγοράς των προϊόντων. Η επόμενη εικόνα (Εικόνα 8) δείχνει τα τρία συστατικά του προβλήματος σε μία στρωματική απεικόνιση. Είναι σαφές ότι η βέλτιστη επίλυση κάθε υποπροβλήματος ξεχωριστά δεν εγγυάται την επίτευξη της βέλτιστης λύσης για το TPP.



Εικόνα 8: Κύρια συστατικά του TPP σε στρωματική απεικόνιση [40]

Μια κρίσιμη πτυχή που διαφοροποιεί το TPP από το μεγαλύτερο μέρος των προβλημάτων δρομολόγησης είναι ότι δεν θεωρεί απαραίτητο να περάσει ο αγοραστής από όλους τους προμηθευτές (πρώτο στρώμα στο παραπάνω σχήμα). Αυτό το χαρακτηριστικό συνδέει το TPP με τα λεγόμενα «Προβλήματα Δρομολόγησης με Κέρδος» [41]. Δεδομένου ότι ο κύριος στόχος του αγοραστή είναι να ικανοποιήσει την ζήτηση των προϊόντων, η επιλογή της επίσκεψης ή μη του κάθε προμηθευτή εξαρτάται γενικά από την εξισσορόπηση μεταξύ του πρόσθετου κόστους μετακίνησης του αγοραστή στον προμηθευτή και της πιθανής αποταμίευσης που επιτυγχάνεται κατά την αγορά προϊόντων σε χαμηλότερες τιμές. Το TPP έχει ένα διττό χαρακτήρα, συνδυάζοντας γραμμικά σε μια ενιαία αντικειμενική συνάρτηση την ελαχιστοποίηση τόσο του κόστους ταξιδιού όσο και του κόστους των αγορών (δεύτερο και τρίτο στρώμα στην παραπάνω εικόνα). Αυτό καθιστά το πρόβλημα της επιλογής βέλτιστων προμηθευτών πιο περίπλοκο. Αφενός, η βελτιστοποίηση του κόστους ταξιδιού ωθεί τον αγοραστή να επιλέξει μόνο προμηθευτές που είναι απολύτως απαραίτητοι για να ικανοποιήσουν τη ζήτηση προϊόντων. Από την άλλη πλευρά, η ελαχιστοποίηση του κόστους αγοράς ωθεί να επιλέξει ένα πιο βολικό και δυνητικά μεγαλύτερο σύνολο προμηθευτών [40].

Παρατηρείται ότι η επιλογή προμηθευτών στο TPP πρέπει να προορίζεται μόνο σε λειτουργικό επίπεδο, ανάλογα με την καθημερινή ζήτηση των προϊόντων, τις τιμές και τις διαθεσιμότητες. Οι στρατηγικές αποφάσεις για την επιλογή των καλύτερων προμηθευτών βάσει ποιοτικών κριτηρίων (π.χ. ποιότητα υπηρεσιών και αξιοπιστία) αφορούν ένα άλλο καλά μελετημένο πεδίο έρευνας [42].

Στη γενική του μορφή το Ασύμμετρο Πρόβλημα του Αγοραστή μπορεί να μοντελοποιηθεί μαθηματικά όπως παρακάτω:

Έστω $\mathrm{y}_i, \mathrm{i} \in \mathrm{M}$, είναι μια δυαδική μεταβλητή που παίρνει τιμή 1 αν ο προμηθευτής i είναι επιλεγμένος και 0 σε διαφορετική περίπτωση. Έστω $x_{ij}, (\mathrm{i}, \mathrm{j}) \in \mathrm{A}$, είναι μια δυαδική μεταβλητή που παίρνει την τιμή 1 αν η ακμή (i, j) διασχίζεται και 0 σε διαφορετική περίπτωση. Θεωρείται ότι $z_{ik}, \mathrm{k} \in \mathrm{K}, \mathrm{i} \in M_k$, είναι μια μεταβλητή που αντιπροσωπεύει τον αριθμό μονάδων του προϊόντος k που αγοράζεται στον προμηθευτή i. Επιπλέον, για κάθε υποσύνολο κόμβων V', προσδιορίζεται $\delta^{+}(\mathrm{V'}) := \{(\mathrm{i}, \mathrm{j}) \in \mathrm{A} : \mathrm{i} \in \mathrm{V'}, \mathrm{j} \notin \mathrm{V'}\}$ και $\delta^{-}(\mathrm{V'}) := \{(\mathrm{i}, \mathrm{j}) \in \mathrm{A} : \mathrm{i} \notin \mathrm{V'}, \mathrm{j} \in \mathrm{V'}\}$. Σε αυτή την περίπτωση το TPP μπορεί να περιγραφεί με τις παρακάτω εξισώσεις:

$$(ATPP) \quad \min \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{k\in K}\sum_{i\in M_k} p_{ik}z_{ik} \tag{1}$$

Περιορισμοί:

$$\sum_{i\in M_k} z_{ik} = d_k \qquad k\in K \tag{2}$$

$$z_{ik} \le q_{ik}y_i \qquad k\in K, i\in M_k \tag{3}$$

$$\sum_{(i,j)\in\delta^+(\{h\})} x_{ij} = \sum_{(i,j)\in\delta^-(\{h\})} x_{ij} = y_h \qquad h\in M \tag{4}$$

$$\sum_{(i,j)\in\delta^-(M')} x_{ij} \ge y_h \qquad M'\subseteq M, h\in M' \tag{5}$$

$$x_{ij}\in\{0,1\} \qquad (i,j)\in A \tag{6}$$

$$y_i\in\{0,1\} \qquad i\in M \tag{7}$$

$$z_{ik}\ge 0 \qquad k\in K, i\in M_k. \tag{8}$$

Η αντικειμενική συνάρτηση (1) στοχεύει ταυτόχρονα στην ελαχιστοποίηση του κόστους ταξιδιού και αγοράς. Οι εξισώσεις (2) εξασφαλίζουν ότι κάθε ζήτηση προϊόντος ικανοποιείται ακριβώς. Οι περιορισμοί (3) επιβάλλουν ότι κάθε προμηθευτής πρέπει να επισκεφθεί ώστε να αγοραστεί ένα προϊόν από αυτόν και η ποσότητα που αγοράζεται δεν πρέπει να υπερβαίνει την αντίστοιχη διαθεσιμότητα. Οι περιορισμοί (4) και (5) διέπουν την εφικτότητα του επισκέπτη. Οι εξισώσεις (4) επιβάλλουν ότι, για κάθε προμηθευτή που επισκέπτεται, ακριβώς μία ακμή πρέπει να εισέλθει και να εξέλθει από τον σχετικό κόμβο. Οι ανισότητες (5) είναι περιορισμοί συνδεσιμότητας που εμποδίζουν τη δημιουργία υπο-διαδρομών που δεν περιλαμβάνουν την αποθήκη επιβάλλοντας ότι τουλάχιστον μία ακμή πρέπει να εισέρχεται σε κάθε υποσύνολο προμηθευτών M' στο οποίο επισκέπτεται τουλάχιστον ένας προμηθευτής h. Τέλος, οι περιορισμοί (6) μέχρι (8) επιβάλλουν δυαδικές και μη αρνητικές συνθήκες στις μεταβλητές. Δεν απαιτούνται συνθήκες ακεραιότητας για τις μεταβλητές z, ακόμη και αν αντιπροσωπεύουν πραγματικά τον αριθμό μονάδων που αγοράζονται από κάθε προϊόν από κάθε προμηθευτή. Εάν όλα τα δεδομένα εισόδου είναι ακέραια, τότε υπάρχει πάντα μια βέλτιστη λύση όπου όλες οι μεταβλητές z έχουν ακέραιες τιμές.

Μια αρχική προεπεξεργασία μπορεί να εφαρμοστεί για την ενίσχυση του μοντέλου των εξισώσεων (1)-(8). Για το σκοπό αυτό, καθορίζεται $M^* := \{0\} \cup \{\, \mathrm{i} \in \mathrm{M} :\ \exists\, \mathrm{k} \in \mathrm{K}$ τέτοιο ώστε $\sum_{j \in M_{k \setminus \{i\}}} q_{jk} < \mathrm{d}_k \,\}$ ως το σύνολο κόμβων που πρέπει απαραίτητα να είναι μέρος οποιασδήποτε εφικτής λύσης TPP και $K^* := \{\, \mathrm{k} \in \mathrm{K} :\ \sum_{i \in M_k} q_{ik} = \mathrm{d}_k \,\}$ ως το καθορισμένο προϊόν για το οποίο μπορούν να καθοριστούν προαποφασισμένες επιλογές προμηθευτών και σχεδίου αγοράς. Έτσι, οι περιορισμοί (7) μπορούν να αντικατασταθούν από $y_i = 1$ όταν $i \in M^*$, και οι περιορισμοί (2) από $z_{ik} = q_{ik}$ όταν $k \in K^*, i \in M_k$.

2.2.11. Πρόβλημα του Αγροτικού Ταχυδρόμου

Το Πρόβλημα του Αγροτικού Ταχυδρόμου (Rural Postman Problem – RPP), συνίσταται στον καθορισμό μιας διαδρομής ελαχίστου κόστους που να διέρχεται από ένα σύνολο καθορισμένων ακμών ενός γράφου αλλά με την ιδιαιτερότητα ότι αυτό το σύνολο ακμών που πρέπει να διασχιστεί τουλάχιστον μία φορά αποτελεί υποσύνολο όλων των ακμών του γράφου του προβλήματος.

Η κύρια διαφορά του RPP με το TSP είναι ότι ενώ το TSP ψάχνει τη βέλτιστη διαδρομή που περνάει από έναν αριθμό κόμβων και καταλήγει στον κόμβο από όπου ξεκίνησε, το RPP αναζητάει μια διαδρομή που να ξεκινάει από ένα κόμβο αλλά να περνάει από ένα δοσμένο πλήθος ακμών του γράφου του προβλήματος πριν καταλήξει στον κόμβο από τον οποίο ξεκίνησε [43].

Στο Πρόβλημα του Αγροτικού Ταχυδρόμου δίνεται ένας γράφος G = (V, E) με τα βάρη των ακμών c(i, j) και ένα σύνολο F, υποσύνολο του E. Το πρόβλημα του αγροτικού ταχυδρόμου συνίσταται στην εύρεση της συντομότερης διαδρομής, που περιέχει όλες τις ακμές του συνόλου F, στον υπογράφο του G, προερχόμενο από κάποιο υποσύνολο του V. Η εύρεση μιας τέτοιας διαδρομής ονομάζεται Πρόβλημα του Αγροτικού Ταχυδρόμου στο G. Όσο για τη γραφική επίλυση αυτού του TSP πρέπει να υποτεθεί ότι υπάρχουν μόνο μη αρνητικά βάρη στις ακμές ώστε να αποφευχθεί να παγιδευτεί το πρόβλημα σε μη πεπερασμένο αριθμό λύσεων.

Γενικά, το πρόβλημα είναι NP-δύσκολο, αφού το TSP μπορεί εύκολα να μετασχηματιστεί σε αυτό. Αρχικά, προστίθεται ένας αρκετά μεγάλος αριθμός σε όλα τα βάρη των ακμών ώστε να εξασφαλιστεί ότι παραμένει η τριγωνική ανισότητα. Στη συνέχεια διαιρείται κάθε κόμβος i σε δύο κόμβους i και i’. Για κάθε ακμή (i, j) δημιουργούνται τα άκρα (i’, j) και (i, j’) με βάρη c (i’, j) = c (i, j’) = c (i, j) και οι ακμές που συνδέουν τους κόμβους i με i’ και j με j’ λαμβάνουν μηδενικά βάρη. Το F αποτελείται από όλες τις ακμές (i, i').

Δεν αξίζει να μετατραπεί ένας βέλτιστος Χαμιλτονιανός κύκλος στο G’ σε μια βέλτιστη διαδρομή αγροτικού ταχυδρόμου στο G. Μπορεί εύκολα να γενικευτεί αυτή η μετατροπή στην περίπτωση όπου μόνο οι ακμές, αλλά και ορισμένοι κόμβοι πρέπει να βρίσκονται στην διαδρομή που θα ακολουθηθεί. Τέτοιοι κόμβοι προστίθενται απλά στην προκύπτουσα διαδρομή του TSP και όλα τα νέα άκρα λαμβάνουν ως βάρη τα αντίστοιχα συντομότερα μήκη διαδρομής [36].

Αν το σύνολο F παράγει ένα συνδεδεμένο υπογράφο του G, τότε έχουμε την ειδική περίπτωση του Προβλήματος του Κινέζου Ταχυδρόμου (Chinese Postman Problem – CPP) που μπορεί να επιλυθεί σε πολυωνυμικό χρόνο χρησιμοποιώντας τεχνικές αντιστοίχισης [44]. Το CPP έχει γίνει γνωστό με αυτό το όνομα επειδή ήταν το πρώτο που συζητήθηκε από έναν Κινέζο μαθηματικό, τον K. Mei-Ko. Το Πρόβλημα του Κινέζου Ταχυδρόμου έχει διάφορα πεδία εφαρμογής, για παράδειγμα, στην περιπολία δρόμων από έναν αστυνομικό, στον καθορισμό της διαδρομής των οδοκαθαριστών στους δρόμους και στην δρομολόγηση των οχημάτων συλλογής των οικιακών απορριμμάτων, στην παράδοση καυσίμων στα νοικοκυριά, στον ψεκασμό δρόμων με άμμο ή αλάτι κατά τους χειμερινούς μήνες και σε άλλα [45].

3. Αλγόριθμοι Επίλυσης

Μέχρι σήμερα, όλοι οι αλγόριθμοι που χρησιμοποιούνται για την επίλυση του TSP εφρμόζονται και σε πολλά άλλα πεδία εφαρμογών. Αυτοί οι αλγόριθμοι χωρίζονται κυρίως σε δύο κατηγορίες: στους ακριβείς αλγόριθμους και στους ευρετικούς αλγορίθμους [46]. Η βασική διαφορά μεταξύ των ακριβών και των ευρετικών αλγορίθμων είναι το γεγονός ότι οι ευρετικοί αλγόριθμοι δε δίνουν τη βέλτιστη λύση (αλλά δεν είναι και αδύνατο να καταλήξουν τελικά σε αυτή), αλλά δίνουν μια προσέγγιση που είναι αρκετά κοντά (άλλοτε λιγότερο και άλλοτε περισσότερο) στη βέλτιστη λύση [47].

Πριν αναφερθούμε στους διάφορους τύπους αλγορίθμων που χρησιμοποιούνται για την επίλυση του Προβλήματος του Περιοδεύοντος Πωλητή αξίζει να αναφερθούμε στον ορισμό των NP-δύσκολων προβλημάτων. Στη Θεωρία Υπολογιστικής Πολυπλοκότητας, μια κατηγορία πολυπλοκότητας είναι ένα σύνολο από προβλήματα που σχετίζονται με βάση την πολυπλοκότητά τους. Η κατηγορία NP είναι ένα σύνολο προβλημάτων απόφασης των οποίων οι λύσεις μπορούν να προσδιοριστούν από μια μη-ντετερμινιστική μηχανή Turing σε πολυωνυμικό χρόνο. Η μηχανή Turing είναι μια θεωρητική μηχανή που χρησιμοποιείται σε πειράματα σκέψης για να εξετάσει τις δυνατότητες και τους περιορισμούς των υπολογιστών. Η μη-ντετερμινιστική μηχανή Turing έχει ένα σύνολο κανόνων το οποίο ορίζει περισσότερες από μία ενέργειες για μια συγκεκριμένη κατάσταση. Το πρόβλημα με τον κύκλο του Hamilton είναι ένα NP-πλήρες πρόβλημα (δηλαδή μπορεί να επαληθευτεί σε πολυωνυμικό χρόνο), από το οποίο προκύπτει για το TSP ότι ανήκει στα NP-δύσκολα προβλήματα λόγω της πολυπλοκότητάς του. Αυτό παρέχει μια μαθηματική εξήγηση για την προφανή υπολογιστική δυσκολία εύρεσης βέλτιστων διαδρομών. Αρχικά αναπτύχθηκαν νέες αλγοριθμικές τεχνικές κα εφαρμόστηκαν στο TSP για να αποδείξουν την αποτελεσματικότητά τους. Παραδείγματα τέτοιων τεχνικών είναι η μέθοδος Branch & Bound, η μέθοδος χαλάρωσης του Lagrange, ο αλγόριθμος και η ευριστίκη συνάρτηση των Lin-Kernighan, η προσομοιωμένη ανόπτηση και το πεδίο των συνδυαστικών πολυέδρων για δύσκολα συνδυαστικά προβλήματα βελτιστοποίησης [4]. Ορισμένες από αυτές τις μεθόδους αναλύονται στη συνέχεια της παρούσας εργασίας.

Όποιος αναγνώστης επιθυμεί να εμβαθύνει περισσότερο στη θεωρία των P, NP και NP-πλήρων προβλημάτων μπορεί να ανατρέξει στο βιβλίο των Garey & Johnson με τίτλο «Computers and Intractability: A Guide to the Theory of NP Completeness» [48].

3.1. Ακριβείς Αλγόριθμοι

Είναι πιθανό για κάποια προβλήματα να μην υπάρχει ακριβής αλγόριθμος που να μπορεί να τα επιλύσει. Οι ακριβείς αλγόριθμοι είναι συνήθως πολύ απαιτητικοί (τόσο από άποψη χρόνου όσο και από άποψη υπολογιστικής ισχύος) και δεν μπορούν να λύσουν κάποια προβλήματα σε πολυωνυμικό χρόνο. Ένα είδος ακριβών αλγορίθμων είναι οι εξαντλητικοί αλγόριθμοι. Στο Πρόβλημα του Περιοδεύοντος Πωλητή, πιο συγκεκριμένα, εξαντλητικός αλγόριθμος σημαίνει να εξεταστούν όλες οι πιθανές λύσεις του προβλήματος ώστε να βρεθεί ποιά είναι η βέλτιστη, ωστόσο, όπως έχει ήδη αναφερθεί και στην εισαγωγή της εργασίας, από ένα μικρό αριθμό κόμβων και πάνω το να χρησιμοποιηθεί ένας εξαντλητικός αλγόριθμος είναι πρακτικά αδύνατον [47].

Ως ακριβείς αλγόριθμοι θεωρούνται από τη διεθνή βιβλιογραφία και, επί της ουσίας, προσεγγιστικοί αλγόριθμοι που όμως οδηγούν αρκετά συχνά στο ολικό βέλτιστο του προβλήματος που εξετάζουν. Οι πιο γνωστοί τέτοιοι ακριβείς αλγόριθμοι που έχουν χρησιμοποιηθεί έως σήμερα για την επίλυση του TSP είναι ο αλγόριθμος Branch and Bound, ο αλγόριθμος Cutting Plane και ο αλγόριθμος Lin-Kernighan που παρουσιάζονται στη συνέχεια. Πάνω σε αυτούς τους αλγόριθμους στηρίζονται σχεδόν όλες οι μεταγενέστερες προσπάθειες που έχουν γίνει ώστε να βρεθεί το ολικό βέλτιστο διαφόρων Προβλημάτων Συνδυαστικής Βελτιστοποίησης.

- **Branch and Bound**

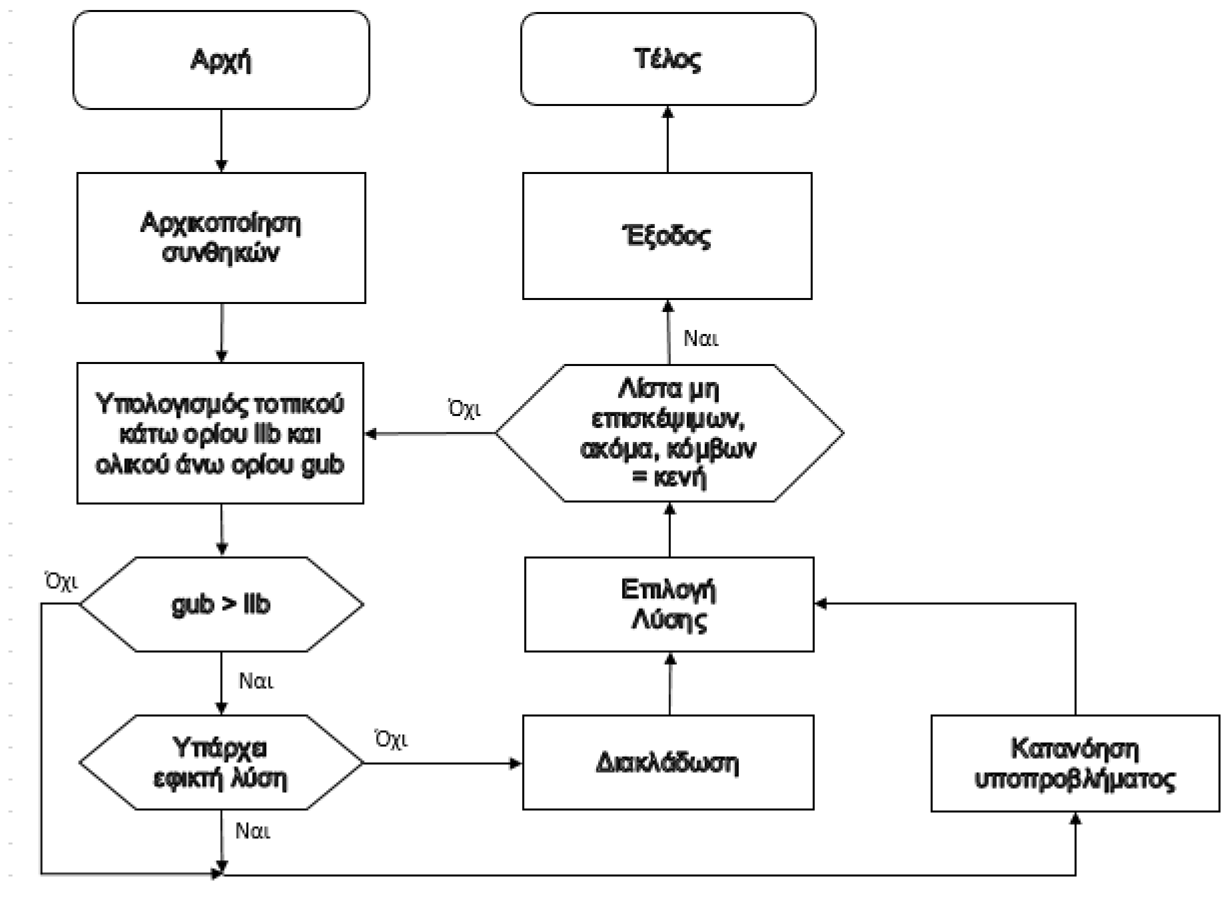
Ο αλγόριθμος Branch & Bound (B&B) θα μπορούσε να μεταφραστεί στα ελληνικά ως Αλγόριθμος Διακλάδωσης και Οριοθέτησης ή Επέκτασης και Οριοθέτησης όμως επιλέγεται στην παρούσα εργασία να χρησιμοποιείται, κατά κύριο λόγο, ο πρωτότυπος αγγλικός όρος. Είναι ένας γενικός αλγόριθμος για εύρεση βέλτιστων λύσεων για διάφορα προβλήματα βελτιστοποίησης, ειδικά για διακριτά ή συνδυαστικά προβλήματα. Αποτελείται από μια συστηματική απογραφή των υποψήφιων λύσεων, όπου μεγάλα υποσύνολα από αχρείαστα δεδομένα απορρίπτονται μαζικά, χρησιμοποιώντας άνω και κάτω εκτιμώμενα όρια της ποσότητας που βελτιστοποιείται [4]. Η μέθοδος προτάθηκε από τους A. H. Land και A. G. Doig το 1960 και στηρίζεται στην χρήση άνω ή κάτω ορίων για τον περιορισμό του χώρου λύσεων [49].

Ο αλγόριθμος Branch and Bound ανήκει στην κατηγορία των ακριβών αλγορίθμων και χρησιμοποιείται ευρέως για την επίλυση του TSP. Έχουν αναπτυχθεί επίσης διάφοροι αλγόριθμοι Branch and Cut που αποτελούν γενίκευση του αλγορίθμου Branch and Bound. Είναι αρκετά περίπλοκο για τους Branch and Bound αλγορίθμους το πώς μπορεί να απλοποιηθεί το Πρόβλημα του Περιοδεύοντος Πωλητή [46].

Οι περισσότερες δημοσιευμένες μέθοδοι που μπορούν να υπολογίσουν το ολικό βέλτιστο ενός προβλήματος βελτιστοποίησης είναι παραλλαγές της αρχής Branch and Bound. Ένας αλγόριθμος Branch and Bound διατηρεί μια λίστα υπο-προβλημάτων του αρχικού προβλήματος του οποίου η ένωση των εφικτών λύσεων περιέχει όλες τις εφικτές λύσεις του αρχικού προβλήματος. Αυτή η λίστα αρχικοποιείται με το ίδιο το αρχικό πρόβλημα.

Το διάγραμμα ροής του επόμενου σχήματος (Εικόνα 9) δίνει τη βασική δομή ελέγχου ενός αλγόριθμου Branch and Bound για ένα συνδυαστικό πρόβλημα βελτιστοποίησης του οποίου στόχος είναι η ελαχιστοποίηση της αντικειμενικής συνάρτησης.

Σε κάθε σημαντική επανάληψη ο αλγόριθμος επιλέγει ένα τρέχον υποπρόβλημα από τη λίστα των υποπροβλημάτων και προσπαθεί να το κατανοήσει με έναν από τους ακόλουθους τρόπους: καθορίζει ένα κατώτερο όριο για την τιμή της βέλτιστης λύσης του τρέχοντος υποπροβλήματος, η οποία έχει τιμή το πολύ όσο αυτή της βέλτιστης εφικτής λύσης που έχει βρεθεί μέχρι εκείνη τη στιγμή, ή αποδεικνύεται ότι το υποπρόβλημα δεν περιέχει καμία εφικτή λύση ή το τρέχον υποπρόβλημα επιλύεται και βρίσκεται η ολική βέλτιστη λύση αυτού. Εάν το τρέχον υποσύστημα δεν μπορεί να κατανοηθεί σύμφωνα με ένα από αυτά τα κριτήρια, τότε χωρίζεται σε νέα υποπροβλήματα, των οποίων η ένωση των εφικτών λύσεών τους περιέχει όλες τις εφικτές λύσεις του τρέχοντος προβλήματος. Αυτά τα πρόσφατα δημιουργημένα προβλήματα προστίθενται στη λίστα των υποπρογραμμάτων που αναφέρθηκε αρχικά. Αυτή η διαδικασία επανάληψης εκτελείται έως ότου η λίστα με τα υποπροβλήματα να είναι κενή.



Εικόνα 9: Διάγραμμα ροής του αλγορίθμου Branch and Bound [36]

Κάθε φορά που βρίσκεται μια εφικτή λύση μέσω της παραπάνω διαδικασίας, η τιμή αυτής αποτελεί ένα ολικό άνω όριο για την τιμή της βέλτιστης λύσης. Αντιστοίχως, η ελάχιστη τιμή των τοπικών κατώτατων ορίων σε οποιοδήποτε σημείο του υπολογισμού είναι ένα ολικό κάτω όριο για την ολικά βέλτιστη τιμή της αντικειμενικής συνάρτησης. Για το TSP, αυτό σημαίνει ότι κατά την εκτέλεση ενός αλγορίθμου Branch and Bound παράγεται μια ακολουθία εφικτών λύσεων μειούμενων μηκών και μια ακολουθία κατώτερων ορίων αυξανόμενων τιμών. Ο αλγόριθμος ολοκληρώνεται με την εύρεση της βέλτιστης λύσης μόλις το κατώτερο όριο συμπίπτει με το μήκος του βραχύτερου Χαμιλτονιανού κύκλου που έχει βρεθεί. Αν ληφθεί υπόψη ότι η πρακτική επίλυση προβλημάτων συνίσταται στην παραγωγή μιας λύσης με προδιαγεγραμμένη ποιότητα p% (δηλαδή απέχει p% από το βραχύτερο Χαμιλτονιανό κύκλο), τότε ένας αλγόριθμος Branch and Bound μπορεί να επιτύχει αυτόν τον στόχο εάν σταματήσει μόλις βρει λύση T του μήκους c(T) και ένα κατώτερο όριο l τέτοιο ώστε $(c(T)-l)/l < (p/100)$. Μια γενική έρευνα πάνω στην εφαρμογή της μέθοδου Branch and Bound στο TSP δόθηκε από τους Balas & Toth το 1985. Ένα από τα σημαντικότερα στοιχεία ενός τέτοιου αλγορίθμου είναι η τεχνική της κατώτερης οριοθέτησης. Τα κατώτερα όρια υπολογίζονται συνήθως με την επίλυση μιας κατάλληλης απλοποίησης του TSP [36].

- **Αλγόριθμος Cutting Plane**

Μία από τις πρώτες, επιτυχείς προσπάθειες επίλυσης ενός TSP με πολλές πόλεις αναφέρεται στην εργασία των Dantzig, Fulkerson & Johnson to 1954 [50] όπου επιλύθηκε ένα TSP 48 πόλεων. Εκείνη η εργασία είναι ένας από τους ακρογωνιαίους λίθους στους οποίους βασίζεται μεγάλο μέρος της χρήσης ευρετικών μεθόδων, Γραμμικού Προγραμματισμού και μεθόδων διαχωρισμού (όπως η Cutting Plane) για την επίλυση συνδυαστικών προβλημάτων βελτιστοποίησης. Αρχικά, ο συστηματικός τρόπος χρήσης της μεθόδου Cutting Plane σε ακέραιο προγραμματισμό, η οποία είχε τεθεί σε σταθερή βάση με το έργο του Gomory [1958, 1960, 1963], δεν ήταν επιτυχής στην πράξη. Από τις αρχές της δεκαετίας του 1980 όμως, αναπτύχθηκε περισσότερη διορατικότητα πάνω στη δομή του Προβλήματος του Περιοδεύοντος Πωλητή και βελτιωμένοι αλγόριθμοι με βάση τη μέθοδο Cutting Plane αναπτύχθηκαν σταδιακά.

Σημαντικό ρόλο στην εξέλιξη της μεθόδου έπαιξαν και δύο ακόμα εργασίες των Padberg & Hong [51] και των Crowder & Padberg [52] το 1980. Στην πρώτη εργασία χρησιμοποιείται μια πρωτογενής προσέγγιση της μεθόδου cutting plane για την επίτευξη καλών ορίων στην ποιότητα των λύσεων που προκύπτουν με τον ακόλουθο τρόπο: Ένας αρχικός Χαμιλτονιανός κύκλος καθορίζεται από τον ευρετικό αλγόριθμο Lin-Kernighan. Στη συνέχεια, επιλέγεται μια μεταβλητή περιστροφής (pivoting variable) με κριτήριο την πιο απότομη ακμή. Εάν η παραγόμενη βασική λύση μετά τη χρήση της μεταβλητής περιστροφής είναι επίσης ενός Χαμιλτονιανός κύκλος τότε αυτός ο κύκλος επιλέγεται και ο αλγόριθμος συνεχίζει με το νέο Χαμιλτονιανό κύκλο. Διαφορετικά, εντοπίζεται μια παραβιαζόμενη ανισότητα που όμως ικανοποιείται με την ισότητα από την τρέχουσα λύση αλλά παραβιάζεται από την παραγόμενη λύση. Αν βρεθεί μια τέτοια ανισότητα, προσαρτάται στο τρέχον Γραμμικό Προγραμματισμό, δημιουργείται ένας εκφυλισμένος πίνακας πάνω στην επιλεγμένη μεταβλητή περιστροφής και επιλέγεται η επόμενη μεταβλητή περιστροφής. Διαφορετικά, το τρέχον (τελικό) γραμμικό πρόγραμμα επιλύεται με βέλτιστο τρόπο προκειμένου να επιτευχθεί το χαμηλότερο δυνατό όριο μήκους του μικρότερου Χαμιλτονιανού κύκλου. Από 74 προβλήματα που εξετάστηκαν με αριθμό πόλεων που κυμαίνεται από 15 έως 318 πόλεις, τα 54 από αυτά τα προβλήματα θα μπορούσαν να οδηγηθούν στο ολικό βέλτιστο ακολουθώντας αυτή τη διαδικασία.

Στη δεύτερη εργασία χρησιμοποιήθηκε το πακέτο προγραμματισμού MPSX-MIP ακέραιων αριθμών της IBM για την εύρεση του διανύσματος μιας βέλτιστης λύσης ως εξής: Το MIP εφαρμόζεται στο τελικό LP για να βρεθεί μια βέλτιστη ολοκληρωμένη λύση. Αν αυτή η λύση είναι ένας Χαμιλτονιανός κύκλος, αυτός ο κύκλος επιστρέφεται ως η βέλτιστη λύση. Διαφορετικά η λύση είναι αναγκαστικά μια συλλογή από δευτερεύουσες μετακινήσεις και οι αντίστοιχες ανισότητες απομάκρυνσης των υποδιαδρομών προσαρτώνται στο ακέραιο πρόγραμμα και η διαδικασία επαναλαμβάνεται. Έτσι, για πρώτη φορά ήταν διαθέσιμο ένα πλήρως αυτόματο υπολογιστικό πρόγραμμα που δεν απαιτούσε ανθρώπινη αλληλεπίδραση για την επίλυση Προβλημάτων Περιοδεύοντων Πωλητών με χρήση ευρετικών μεθόδων, γραμμικού προγραμματισμού, διαχωρισμού και απαρίθμησης στο πνεύμα της εργασίας των Dantzig, Fulkerson και Johnson. Χρησιμοποιώντας τον κώδικα που ανέπτυξαν, οι συγγραφείς ήταν σε θέση να επιλύσουν και τα 74 προβλήματα του δείγματος βρίσκοντας το ολικό βέλτιστο. Το παράδειγμα με τις 318 πόλεις λύθηκε σε λιγότερο από μία ώρα χρόνου CPU σε έναν υπολογιστή IBM 370/168 υπό το λειτουργικό σύστημα MVS [36].

Η μαθηματική διατύπωση του αλγορίθμου Cutting Plane, όπως αναπτύχθηκε από τους Dantzig, Fulkerson & Johnson [50] φαίνεται στη συνέχεια [53]:

$$\sum_{i,j} c_{ij} x_{ij}, \qquad (1)$$

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2 \quad (k = 1, \dots, n), \qquad (2)$$

$$\sum_{\substack{i \in S \\ j \in S}} x_{ij} \leqq |S| - 1 \qquad (3)$$

Για όλα τα σύνολα S που είναι κατάλληλα υποσύνολα του συνολικού αριθμού κορυφών και:

$$x_{ij} = \emptyset, 1 \qquad (4)$$

Όπου είναι προφανές ότι μόνο μεταβλητές x_{ij} με $i < j$ λαμβάνονται υπόψιν.

Σκοπός του προβλήματος είναι η ελαχιστοποίηση της εξίσωσης (1). Ο περιορισμός (2) ονομάζεται και περιορισμός των κορυφών ενώ ο περιορισμός (3) ονομάζεται και περιορισμός του βρόχου. Η τελευταία εξίσωση μπλοκάρει τυχόν υποδιαδρομές με μήκος μικρότερο από n, περιορίζοντας το S ώστε να είναι ένα σωστό υποσύνολο των n πόλεων. Με το |S| δηλώνεται ο αριθμός στοιχείων του S. Με την παρουσία του 1 ως ανώτερου ορίου στις μεταβλητές, οι βρόχοι μήκους 2 και n-2 αποκλείονται αναγκαστικά και το S πρέπει να είναι τέτοιο ώστε $3 \leq |S| \leq n-3$.

Όπως στον αλγόριθμο Branch and Bound που παρουσιάστηκε παραπάνω, το αρχικό σύνολο περιορισμών είναι ένα υποσύνολο του συνόλου περιορισμών που περιγράφει πλήρως το πρόβλημα (εκτός από τις συνθήκες πλήρους ολοκλήρωσης). Μετά από έναν αρχικό πειραματισμό έχει προκύψει ότι είναι καλύτερο να χρησιμοποιείται ο περιορισμός (2) ως μέρος του αρχικού συνόλου περιορισμών. Το σχετικό LP επιλύεται επανειλημμένα, αλλά αντί για διακλάδωση σε μια κλασματική μεταβλητή για να επιτευχθεί μια ακέραια λύση, τα επίπεδα κοπής (Cutting Plane) χρησιμοποιούνται για να οδηγήσουν μια κλασματική λύση σε μία ακέραια. Όταν βρεθεί μια ακέραια λύση, δοκιμάζεται η εφικτότητά της (δηλαδή γίνεται μια δοκιμή αν αντιπροσωπεύει μια πλήρη διαδρομή). Η δοκιμή αυτή εκτελείται με την ίδια τεχνική επισήμανσης όπως στο Branch and Bound. Εάν η λύση είναι εφικτή είναι επίσης βέλτιστη και η διαδικασία ολοκληρώνεται. Διαφορετικά δημιουργούνται οι παραβιαζόμενοι περιορισμοί, προστίθενται στην τρέχουσα δέσμη περιορισμών και επαναλαμβάνεται η διαδικασία. Αυτή η έκδοση του αλγόριθμου κοπής επιπέδων ονομάζεται «ευθείς αλγόριθμος». Μια διαφορετική εκδοχή λαμβάνεται εάν αλλάξει η σειρά εισαγωγής των επιπέδων κοπής και οι παραλειπόμενοι περιορισμοί. Η δεύτερη εκδοχή ονομάζεται «αντίστροφος αλγόριθμος» [53].

- **Αλγόριθμος Lin-Kernighan**

Η ευρετική μέθοδος Lin-Kernighan [54] θεωρείται γενικά ότι είναι μία από τις πιο αποτελεσματικές μεθόδους για την εύρεση βέλτιστων ή σχεδόν βέλτιστων λύσεων για το Συμμετρικό Πρόβλημα του Περιοδεύοντος Πωλητή. Ωστόσο, ο σχεδιασμός και η υλοποίηση ενός αλγορίθμου με βάση τη μέθοδο των Lin-Kernighan είναι αρκετά πολύπλοκη διαδικασία. Υπάρχουν πολλές αποφάσεις σχεδιασμού και υλοποίησης που μπορούν να παρθούν και οι περισσότερες από αυτές μπορούν να έχουν μεγάλη σημασία στην τελική απόδοση της όποιας υλοποίησης.

Ο Αλγόριθμος Lin-Kernighan έχει σαν βάση του τον αλγόριθμο 2-opt που είναι μια ειδική περίπτωση του αλγόριθμου λ-opt [55], όπου, σε κάθε επανάληψη, λ συνδέσεις της τρέχουσας διαδρομής αντικαθίστανται από λ άλλες συνδέσεις με τέτοιο τρόπο ώστε να επιτυγχάνεται μια μικρότερη διαδρομή. Με άλλα λόγια, σε κάθε βήμα προκύπτει μια μικρότερη διαδρομή μέσω της διαγραφής λ συνδέσεων και συνδέοντας τις νέες διαδρομές που προκύπτουν, πιθανώς, αντιστρέφοντας μία ή περισσότερες από αυτές τις συνδέσεις. Ο αλγόριθμος λ-opt βασίζεται στο ότι μία διαδρομή αποκαλείται λ-βέλτιστη (ή απλά λ-opt) αν είναι αδύνατο να επιτευχθεί μια συντομότερη διαδρομή αντικαθιστώντας οποιεσδήποτε λ συνδέσεις της με οποιαδήποτε άλλη ομάδα λ συνδέσεων.

Γενικά, όσο μεγαλύτερη είναι η τιμή του λ, τόσο πιο πιθανό είναι ότι η τελική διαδρομή είναι η βέλτιστη. Για αρκετά μεγάλο λ φαίνεται, τουλάχιστον διαισθητικά, ότι η λ-opt διαδρομή πρέπει να είναι η βέλτιστη. Δυστυχώς, ο αριθμός των πράξεων για τη δοκιμή όλων των λ-ανταλλαγών αυξάνεται ραγδαία καθώς ο αριθμός των πόλεων αυξάνεται. Σε μια απλή υλοποίηση, ο έλεγχος της λ-ανταλλαγής έχει μια πολυπλοκότητα χρόνου του $O(n^2)$. Επιπλέον, δεν υπάρχει μη τετριμμένο ανώτατο όριο του αριθμού των λ-ανταλλαγών. Ως αποτέλεσμα, οι τιμές λ = 2 και λ = 3 είναι οι συχνότερα χρησιμοποιούμενες αν και υπάρχουν μελέτες όπου χρησιμοποιήθηκαν οι τιμές λ = 4 και λ = 5. Ωστόσο, είναι μειονέκτημα ότι πρέπει να προσδιοριστεί εκ των προτέρων η τιμή του λ. Είναι δύσκολο να γνωρίζουμε τι λ πρέπει να χρησιμοποιηθεί ώστε να επιτευχθεί ο καλύτερος συμβιβασμός μεταξύ χρόνου εκτέλεσης και ποιότητας λύσης.

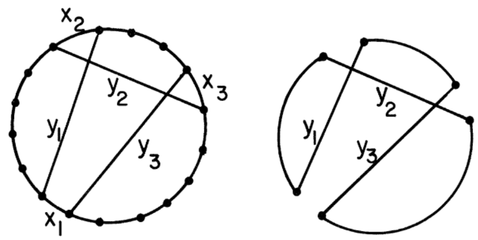
Οι Lin & Kernighan αφαίρεσαν αυτό το μειονέκτημα εισάγοντας έναν ισχυρό μεταβλητό αλγόριθμο λ-opt. Σε κάθε βήμα επανάληψης ο αλγόριθμος εξετάζει, για τις αύξουσες τιμές του λ, αν η ανταλλαγή λ συνδέσεων μπορεί να οδηγήσει σε μικρότερη διαδρομή. Δεδομένου ότι εξετάζεται η ανταλλαγή λ συνδέσεων, πραγματοποιείται μια σειρά δοκιμών για να καθοριστεί εάν θα πρέπει να ληφθούν υπόψη οι ανταλλαγές συνδέσεων λ + 1. Αυτό συνεχίζεται μέχρι να ικανοποιηθούν κάποιες συνθήκες ολοκλήρωσης του αλγορίθμου. Σε κάθε βήμα ο αλγόριθμος θεωρεί ένα αυξανόμενο σύνολο πιθανών ανταλλαγών (ξεκινώντας από λ = 2). Αυτές οι ανταλλαγές επιλέγονται κατά τέτοιο τρόπο ώστε να είναι εφικτή η ύπαρξη διαδρομής σε οποιοδήποτε στάδιο της διαδικασίας. Εάν η εξερεύνηση κατορθώσει να βρει μια νέα συντομότερη διαδρομή, τότε η πραγματική διαδρομή αντικαθίσταται με τη νέα.

Με μια εφικτή διαδρομή, ο αλγόριθμος εκτελεί εκ νέου ανταλλαγές που μειώνουν το μήκος της τρέχουσας διαδρομής, μέχρι να επιτευχθεί μια διαδρομή για την οποία καμία ανταλλαγή δεν αποφέρει βελτίωση του συνολικού της μήκους. Αυτή η διαδικασία μπορεί να επαναληφθεί πολλές φορές από αρχικές περιηγήσεις που παράγονται με κάποιο τυχαίο τρόπο. Ο αλγόριθμος περιγράφεται παρακάτω λεπτομερέστερα.

Έστω Τ είναι η τρέχουσα διαδρομή. Σε κάθε βήμα επανάληψης ο αλγόριθμος προσπαθεί να βρει δύο ομάδες συνδέσεων, $X = [x_1, \ldots, x_r]$ και $Y = [y_1, \ldots, y_r]$, τέτοιες ώστε, εάν οι συνδέσεις του Χ διαγραφούν από το Τ και αντικατασταθούν από τις συνδέσεις του

Y, το αποτέλεσμα να είναι μια καλύτερη διαδρομή. Αυτή η ανταλλαγή συνδέσεων ονομάζεται κίνηση r-opt. Η Εικόνα 10 παρακάτω απεικονίζει μια κίνηση 3-opt. Τα δύο σύνολα X και Y είναι κατασκευασμένα στοιχείο-στοιχείο. Αρχικά τα X και τα Y είναι κενά. Στο βήμα i ένα ζεύγος δεσμών, x_i και y_i, προστίθενται στα X και Y αντίστοιχα. Προκειμένου να επιτευχθεί ένας επαρκώς αποτελεσματικός αλγόριθμος, μόνο οι συνδέσεις που πληρούν τα ακόλουθα κριτήρια μπορούν να εισέλθουν στα X και Y:

1) Το κριτήριο διαδοχικής ανταλλαγής: τα x_i και y_i πρέπει να μοιράζονται ένα τελικό σημείο, και το ίδιο πρέπει να ισχύει και για τα y_i και x_{i+1}. Αν ως t_1 δηλώνεται ένα από τα δύο τελικά σημεία του x_1, έχουμε γενικά: $x_i = (t_{2i-1}, t_{2i})$, $y_i = (t_{2i}, t_{2i+1})$ και $x_{i+1} = (t_{2i+1}, t_{2i+2})$ για $i \geq 1$.
2) Το κριτήριο σκοπιμότητας: Απαιτείται να επιλέξουμε το $x_i = (t_{2i-1}, t_{2i})$ έτσι ώστε, αν το t_{2i} είναι ενωμένο με το t_1, τότε η ένωση που θα προκύψει να είναι μια διαδρομή. Αυτό το κριτήριο σκοπιμότητας χρησιμοποιείται για $i \geq 3$ και εγγυάται ότι είναι δυνατή η ολοκλήρωση μιας διαδρομής. Αυτό το κριτήριο συμπεριλήφθηκε στον αλγόριθμο τόσο για τη μείωση του χρόνου λειτουργίας όσο και για την απλοποίηση της κωδικοποίησης.
3) Το κριτήριο θετικού κέρδους: Απαιτείται ότι το y_i επιλέγεται πάντα έτσι ώστε το κέρδος G_i από το προτεινόμενο σύνολο ανταλλαγών να είναι θετικό. Έστω ότι το $g_i = c(x_i) – c(y_i)$ είναι το κέρδος από την ανταλλαγή του x_i με το y_i. Τότε G_i είναι το άθροισμα $g_1 + g_2 + .. + g_i$. Το κριτήριο αυτό παίζει σημαντικό ρόλο στην αποτελεσματικότητα του αλγορίθμου. Η απαίτηση ότι κάθε μερικό άθροισμα, G_i, πρέπει να είναι θετικό, φαίνεται αρχικά να είναι πολύ περιοριστική όμως, δεν ισχύει όπως φαίνεται και από το κόλουθο παράδειγμα: Εάν μια ακολουθία αριθμών έχει ένα θετικό άθροισμα, υπάρχει μια κυκλική μεταβολή αυτών των αριθμών έτσι ώστε κάθε μερικό άθροισμα να είναι θετικό. Η απόδειξη είναι απλή και μπορεί να βρεθεί στην εργασία των Lin & Kernighan [54].
4) Το κριτήριο της διαζευκτικότητας: Τέλος, είναι και πάλι επιθυμητό τα σύνολα X και Y να είναι διακριτά. Αυτό απλοποιεί την κωδικοποίηση, μειώνει το χρόνο επίλυσης και δίνει στον αλγόριθμο ένα κριτήριο πραγματικής διακοπής.



Εικόνα 10: Μια κίνηση 3-opt. Αριστερά η διαδρομή T και δεξιά η διαδρομή T' [54]

Πέρα από τα αρχικά 4 κριτήρια, για να περιοριστεί η αναζήτηση ακόμα περισσότερο, οι Lin και Kernighan βελτίωσαν τον αλγόριθμο εισάγοντας τους ακόλουθους κανόνες:

5) Η αναζήτηση ενός συνδέσμου $y_i = (t_{2i}, t_{2i+1})$ για την είσοδό του στη διαδρομή, περιορίζεται στους πέντε πλησιέστερους γείτονες του t_{2i}.
6) Για i≥4, κανένας σύνδεσμος, x_i, της διαδρομής δεν πρέπει να αφαιρεθεί αν είναι ένας κοινός σύνδεσμος ενός μικρού αριθμού (2-5) αποδεκτών λύσεων.
7) Η αναζήτηση βελτιώσεων διακόπτεται αν η τρέχουσα διαδρομή είναι ίδια με μια προηγούμενη αποδεκτή λύση.

Οι κανόνες 5 και 6 είναι ευρετικοί κανόνες. Βασίζονται στις προσδοκίες σχετικά με το ποιοί νέοι σύνδεσμοι πιθανόν να ανήκουν σε μια βέλτιστη διαδρομή. Εξοικονομούν χρόνο εκτέλεσης, αλλά μερικές φορές σε βάρος της μη επίτευξης της καλύτερης εφικτής λύσης. Ο κανόνας 7 εξοικονομεί επίσης χρόνο επίλυσης, αλλά δεν επηρεάζει την ποιότητα των λύσεων που βρίσκονται. Εάν μια διαδρομή είναι η ίδια με μια προηγούμενη λύση, δεν έχει νόημα να προσπαθήσουμε να την βελτιώσουμε περαιτέρω. Επομένως, ο χρόνος που απαιτείται για να επιβεβαιωθεί ότι δεν είναι δυνατές παραπάνω βελτιώσεις προστίθεται στο χρόνο ολοκλήρωσης και αποθηκεύεται.

Εκτός από αυτές τις βελτιώσεις, των οποίων ο σκοπός είναι κυρίως ο περιορισμός της αναζήτησης, οι Lin και Kernighan πρόσθεσαν κάποιες βελτιώσεις, σκοπός των οποίων είναι κυρίως να κατευθύνουν την αναζήτηση. Όταν ο αλγόριθμος έχει επιλογή εναλλακτικών λύσεων, οι ευρετικοί κανόνες χρησιμοποιούνται για να δώσουν προτεραιότητα σε αυτές τις εναλλακτικές λύσεις. Σε περιπτώσεις που πρέπει να επιλεγεί μόνο μία από τις εναλλακτικές λύσεις, επιλέγεται εκείνη με την υψηλότερη προτεραιότητα. Σε περιπτώσεις όπου πρέπει να δοκιμαστούν αρκετές εναλλακτικές λύσεις, οι εναλλακτικές λύσεις δοκιμάζονται κατά φθίνουσα τάξη προτεραιότητας. Πιο συγκεκριμένα, χρησιμοποιούνται οι ακόλουθοι κανόνες:

8) Όταν πρέπει να επιλεγεί η σύνδεση $y_i(i \geq 2)$, για κάθε πιθανή επιλογή δίνεται η προτεραιότητα $c(x_{i+1}) - c(y_i)$.
9) Εάν υπάρχουν δύο εναλλακτικές λύσεις για το x_4, επιλέγεται το σημείο όπου το $c(x_4)$ είναι το υψηλότερο.

Ο κανόνας 8 είναι ένας ευρετικός κανόνας για την ταξινόμηση των συνδέσεων που πρέπει να προστεθούν στο Y. Η προτεραιότητα για το y_i είναι το μήκος της επόμενης (μοναδικής) σύνδεσης που θα σπάσει, x_{i+1}, αν το y_i εμπεριέχεται στη διαδρομή, μείον το μήκος του y_i. Με τη μεγιστοποίηση της ποσότητας $c(x_{i+1})$-$c(y_i)$, ο αλγόριθμος στοχεύει στη διάσπαση μιας μακριάς σύνδεσης και στην αντικατάστασή της με μία συντομότερη. Ο κανόνας 9 αναφέρεται στην ειδική κατάσταση όπου υπάρχουν δύο επιλογές για το x_4. Ο κανόνας προτιμά τον μακρύτερο σύνδεσμο στην περίπτωση αυτή.

Ως τελευταία βελτίωση, οι Lin και Kernighan συμπεριέλαβαν μια περιορισμένη άμυνα έναντι των καταστάσεων όπου μόνο οι μη διαδοχικές ανταλλαγές μπορούν να οδηγήσουν σε μια καλύτερη λύση. Μετά την εύρεση ενός τοπικού βέλτιστου, ο αλγόριθμος ελέγχει, μεταξύ των συνδέσεων που επιτρέπεται να σπάσει, αν είναι δυνατόν να γίνει μια περαιτέρω βελτίωση με μια μη διαδοχική 4-opt αλλαγή [56].

Ο Helsgaun στην εργασία του [56] περιέγραψε την εφαρμογή μιας, τότε, νέας τροποποιημένης έκδοσης του Lin-Kernighan αλγορίθμου για την οποία τα υπολογιστικά πειράματα έδειξαν ότι είναι εξαιρετικά αποτελεσματική. Ο νέος αλγόριθμος διέφερε σε πολλές λεπτομέρειες από τον αρχικό που περιγράφηκε παραπάνω. Η πιο αξιοσημείωτη διαφορά βρίσκεται στη στρατηγική αναζήτησης. Ο νέος αλγόριθμος χρησιμοποιεί μεγαλύτερα (και πιο σύνθετα) βήματα αναζήτησης απ' ότι ο αρχικός. Επίσης νέα είναι η χρήση της ανάλυσης ευαισθησίας για να κατευθύνει και να περιορίσει την αναζήτηση. Σαν αποτέλεσμα, ο νέος αλγόριθμος είναι πολύ πιο αποτελεσματικός και καθιστά δυνατή την αντιμετώπιση προβλημάτων μεγάλης κλίμακας σε λογικό χρόνο επίλυσης. Για ένα τυπικό πρόβλημα 100 πόλεων η βέλτιστη λύση βρίσκεται σε λιγότερο από ένα δευτερόλεπτο ενώ για ένα τυπικό πρόβλημα 1.000 πόλεων, η βέλτιστη λύση μπορεί να βρεθεί σε λιγότερο από ένα λεπτό.

Παρόλο που ο αλγόριθμος, θεωρητικά, βρίσκει λύσεις κατά προσέγγιση, ολικά βέλτιστες λύσεις εμφανίζονται με εντυπωσιακά υψηλή συχνότητα. Έχει δημιουργήσει ολικά βέλτιστες λύσεις για πολλά λυμένα προβλήματα, συμπεριλαμβανομένου ενός προβλήματος 13.509 πόλεων. Επιπλέον, ο αλγόριθμος βελτίωσε τις πιο γνωστές λύσεις για μια σειρά μεγάλης κλίμακας προβλημάτων με άγνωστο το ολικό βέλτιστο, μεταξύ των οποίων το TSP των 85.900 πόλεων [56].

Ο Helsgaun συνέχισε να εργάζεται τα επόμενα χρόνια πάνω στον αλγόριθμο Lin-Kernighan. Σε μία από τις τελευταίες του εργασίες [57], μαζί με τους Tinos και Whitley, παρουσίασαν μια επανεξέταση μιας εξέλιξης του αλγορίθμου Lin-Kernighan που έχει πάρει το όνομα Lin-Kernighan-Helsgaun (LKH).

- **Concorde TSP Solver**

Ολοκληρώνοντας το υποκεφάλαιο των ακριβών αλγορίθμων αξίζει να αναφερθούμε στο Concorde TSP Solver. Το Concorde είναι ένας κώδικας υπολογιστή που χρησιμοποιείται για την επίλυση του Συμμετρικού Προβλήματος του Περιοδεύοντος Πωλητή καθώς και ορισμένων άλλων σχετικών προβλημάτων βελτιστοποίησης δικτύου. Ο κώδικας είναι γραμμένος στη γλώσσα προγραμματισμού ANSI C και είναι ελεύθερα διαθέσιμος για ακαδημαϊκή έρευνα.

Ο TSP Solver της Concorde έχει χρησιμοποιηθεί για να προσδιοριστούν τα ολικά βέλτιστα και των 110 περιπτώσεων που εμπεριέχονται στην TSPLIB, εκ των οποίων η πολυπλοκότερη περίπτωση έχει 85.900 πόλεις. Η βιβλιοθήκη της Concorde στην οποία έχουν πρόσβαση οι χρήστες περιλαμβάνει πάνω από 700 λειτουργίες που επιτρέπουν στους χρήστες να δημιουργούν εξειδικευμένους κώδικες για προβλήματα που μοιάζουν με το TSP. Όλες οι λειτουργίες της Concorde είναι ασφαλείς από ιούς για προγραμματισμό σε παράλληλα περιβάλλοντα κοινής μνήμης. Ο κύριος TSP Solver περιλαμβάνει κώδικα που μπορεί να εκτελεστεί μέσω δικτύων υπολογιστών UNIX.

Πλέον, το Concorde υποστηρίζει και το γραμμικό προγραμματιστή QSopt. Λειτουργικές εκδόσεις του Concorde με QSopt είναι διαθέσιμες σε περιβάλλον Linux και Solaris. Επιπλέον, ο Hans Mittelmann δημιούργησε έναν διακομιστή NEOS για το Concorde, επιτρέποντας στους χρήστες να επιλύουν περιπτώσεις TSP στο διαδίκτυο. Τέλος, ο Pavel Striz έχει δημιουργήσει πακέτα για τη δημιουργία εικόνων LaTeX χρησιμοποιώντας τα αρχεία λύσεων που έδωσε το Concorde [58].

3.2. Ευρετικοί Αλγόριθμοι

Όπως έχει ήδη αναφερθεί, οι ευρετικοί αλγόριθμοι δεν βρίσκουν τη βέλτιστη λύση στα προβλήματα που εξετάζουν (χωρίς να είναι και αδύνατο να καταλήξουν τελικά σε αυτή), αλλά δίνουν μια προσέγγιση που είναι αρκετά κοντά (άλλοτε λιγότερο και άλλοτε περισσότερο) στη βέλτιστη λύση.

Στη διεθνή βιβλιογραφία υπάρχει πλήθος ευρετικών αλγορίθμων που χρησιμοποιούνται για την επίλυση του Προβλήματος του Περιοδεύοντος Πωλητή. Μεταξύ άλλων, οι ευρετικοί αλγόριθμοι περιλαμβάνουν τη μέθοδο Simulated Annealing (SA), την Αναζήτηση Ταμπού (Tabu Search – TS), τη Βελτιστοποίηση Αποικίας Μυρμηγκιών (Ant Colony Optimization – ACO), τον Γενετικό Αλγόριθμο (Genetic Algorithm – GA), τη Βελτιστοποίηση Σμήνους Σωματιδίων (Particle Swarm Optimization – PSO) και άλλους. Σε διάφορες περιπτώσεις, αυτοί οι ευρετικοί αλγόριθμοι συνδυάζονται με αλγόριθμους τοπικής αναζήτησης για τη βελτίωση της αποτελεσματικότητάς τους και προκύπτουν έτσι αλγόριθμοι όπως οι 2-opt, 3-opt [55] και Lin & Kernighan ([54], [46]).

Όπως γίνεται κατανοητό, υπάρχουν πολλοί αλγόριθμοι που χρησιμοποιούνται στο δυναμικό προγραμματισμό για την επίλυση του TSP. Δεδομένου ότι η βέλτιστη λύση μπορεί να μην είναι εφικτή (για μεγάλο αριθμό πόλεων), το κατώτερο όριο Held-Karp χρησιμοποιείται για την αξιολόγηση της απόδοσης ενός συγκεκριμένου αλγορίθμου. Δηλαδή, το όριο Held-Karp καθορίζει πόσο κοντά είναι μια δεδομένη λύση (χρησιμοποιώντας έναν δεδομένο αλγόριθμο) στη βέλτιστη λύση. Συνήθως, το κατώτερο όριο Held-Karp είναι περίπου 0,8% χαμηλότερο από το συνολικό μήκος της βέλτιστης διαδρομής που βρίσκουν οι ευρετικοί αλγόριθμοι [59].

Αξίζει σε αυτό το σημείο να γίνει μια αναφορά και στον όρο μεθευρετικοί αλγόριθμοι παρότι, στη διεθνή βιβλιογραφία χρησιμοποιείται πολύ συχνά αντ’ αυτού ο όρος ευρετικοί αλγόριθμοι κάτι που επιλέγεται να ακολουθηθεί και στην παρούσα εργασία χωρίς να γίνεται διάκριση μεταξύ των δύο όρων.

Οι ευρετικοί αλγόριθμοι είναι τεχνικές που εξαρτώνται από το πρόβλημα που προσπαθούν να επιλύσουν. Ως εκ τούτου, συνήθως προσαρμόζονται σε αυτό και προσπαθούν να εκμεταλλευτούν πλήρως τις ιδιαιτερότητές του. Ωστόσο, επειδή είναι συχνά υπερβολικά άπληστοι (greedy), συνήθως παγιδεύονται σε ένα τοπικό βέλτιστο και έτσι αποτυγχάνουν, γενικά, να βρουν την ολικά βέλτιστη λύση του προβλήματος που εξετάζουν.

Οι μεθευρετικοί ή μετα-ευρετικοί αλγόριθμοι από την άλλη πλευρά, είναι τεχνικές ανεξάρτητες από το πρόβλημα στο οποίο χρησιμοποιούνται. Ως εκ τούτου, δεν επωφελούνται από οποιαδήποτε ιδιαιτερότητα αυτού ενώ, σε γενικές γραμμές, δεν είναι άπληστοι. Στην πραγματικότητα, μπορούν ακόμη και να δεχτούν μια προσωρινή επιδείνωση της βέλτισης λύσης που έχουν βρει σε κάποιο σημείο (για παράδειγμα στη μέθοδο της Προσομοιωμένης Ανόπτησης / Simulated Annealing που περιγράφεται παρακάτω), κάτι που τους επιτρέπει να διερευνήσουν πιο διεξοδικά όλο το χώρο των εφικτών λύσεων και έτσι να βρουν, σε επόμενη επανάληψη, μια λύση ακόμα καλύτερη από την προηγούμενη που απέρριψαν (και που μπορεί ακόμα και να συμπίπτει με το ολικό βέλτιστο που αναζητούν). Ας σημειωθεί ακόμη ότι παρόλο που ένας μεθευρετικός αλγόριθμος είναι τεχνική ανεξάρτητη από το πρόβλημα που εξετάζει, είναι απαραίτητο να γίνει κάποια τελειοποίηση των εγγενών παραμέτρων του, προκειμένου να προσαρμοσθεί σε αυτό.

Πέρα από τους μεθευρετικούς αλγορίθμους υπάρχουν και οι λεγόμενοι υπερ-ευρετικοί, οι οποίοι ξεπερνούν τα όρια των μεθευρετικών. Η ιδιαιτερότητα των υπερ-ευρετικών είναι ότι ο χώρος αναζήτησης τους δεν είναι ο συνηθισμένος χώρος των λύσεων του προβλήματος, αλλά είναι μάλλον ο χώρος των ευρετικών ή των μεθευρετικών. Οι υπερ-ευρετικοί θα μπορούσαν να θεωρηθούν και ως "ευρετικοί για την αναζήτηση ευρετικών". Υπάρχει επίσης μια ελαφρώς διαφορετική κατηγορία που ορίζεται ως "ευρετικοί για την παραγωγή ευρετικών" [60].

Περαιτέρω ενασχόληση και αναφορά στους μεθευρετικούς και στους υπερ-ευρετικού αλγορίθμους ξεφεύγει από τα όρια και το σκοπό αυτής της εργασίας. Όποιος αναγνώστης ενδιαφέρεται μπορεί να διαβάσει και την εργασία του Sorensen K. [61] όπου αναφέρεται ότι οι περισσότεροι μεθευρετικοί αλγόριθμοι αποτελούν επί της ουσίας μια αναδιατύπωση ορισμένων βασικών τεχνικών που επανέρχονται παραλλαγμένοι και με χρήση νέας ορολογίας.

Στη συνέχεια αναφέρονται λίγα λόγια για τους γνωστότερους ευρετικούς αλγορίθμους που απαντώνται στη διεθνή βιβλιογραφία.

- **Πλησιέστερος Γείτονας**

Ο αλγόριθμος Πλησιέστερου Γείτονα είναι γνωστός στη διεθνή βιβλιογραφία με τον όρο Nearest Neighbor (NN). Είναι ο απλούστερος ευρετικός αλγόριθμος που χρησιμοποιείται για την επίλυση του TSP. Ο αλγόριθμος μπορεί να συνοψιστεί ως εξής [59]:

1) Επιλέξτε μια τυχαία πόλη n και ορίστε την ως την αρχική πόλη n_0,
2) Βρείτε την πλησιέστερη πόλη που δεν έχετε ακόμα επισκεφθεί και πηγαίνετε εκεί,
3) Σημειώστε ότι επισκεφτήκατε αυτή την πόλη,
4) Υπάρχουν πόλεις που δεν έχετε επισκεφθεί; Εάν ναι, μεταβείτε στο βήμα (2),
5) Επιστρέψτε στην αρχική πόλη n_0

Αυτός ο αλγόριθμος, σε γενικές γραμμές, επιτυγχάνει οι λύσεις που βρίσκει να απέχουν από το κατώτερο όριο Held-Karp όχι περισσότερο από 25% [62].

- **Γενετικός Αλγόριθμος**

Ο Γενετικός Αλγόριθμος (Genetic Algorithm – GA) είναι γνωστός για την επίλυση σύνθετων προβλημάτων, όπου η βέλτιστη λύση είναι δύσκολο να βρεθεί. Μοιάζει με τη διαδικασία φυσικής επιλογής, απ' όπου πήρε και το όνομά του. Ο αλγόριθμος υπολογίζει μια συνάρτηση καταλληλότητας για κάθε κόμβο του αρχικού δικτύου. Στη συνέχεια, δημιουργεί νέους κόμβους. Χρησιμοποιεί τη διαδικασία της μετάλλαξης για να προσθέσει τυχαιότητα στη διαδικασία, παρόμοια με αυτή του φυσικού γονιδιώματος. Τέλος, επιλέγει τη λύση εκείνη με την υψηλότερη συνάρτηση καταλληλότητας. Η εφαρμογή του Γενετικού Αλγορίθμου στο TSP έχει όμως και ορισμένους περιορισμούς. Για παράδειγμα, σε κάθε διαδρομή, κάθε πόλη δεν πρέπει να επαναλαμβάνεται, διαφορετικά εμφανίζονται βρόχοι (loops). Επιπλέον, κατά την εκτέλεση του αλγορίθμου εξετάζονται μόνο έγκυρες διαδρομές. Για παράδειγμα, μια πόλη που βρίσκεται στα Δυτικά του γράφου των πόλεων δεν μπορεί να θεωρηθεί ως πιθανή συνέχεια μιας διαδρομής που βρίσκεται στα Ανατολικά του γράφου [59].

Οι Naveen Kumar, Karambir & Rajiv Kumar πραγματοποίησαν μια βιβλιογραφική έρευνα για την επίλυση του TSP χρησιμοποιώντας το Γενετικό Αλγόριθμο [63]. Αναφέρουν αρκετούς γενετικούς τελεστές που χρησιμοποιήθηκαν για την επίλυση του TSP χρησιμοποιώντας το Γενετικό Αλγόριθμο, όπως διασταύρωση (crossover), μετάλλαξη (mutation) ή συνδυασμό αυτών. Μετά την ανάλυση των αποτελεσμάτων της έρευνας, οι ερευνητές παρατήρησαν ότι στο μέλλον μπορούν να εισαχθούν και νέοι τελεστές για να αυξήσουν την απόδοση του Γενετικού Αλγορίθμου για την επίλυση του προβλήματος TSP [64].

- **Άπληστος Αλγόριθμος**

Ο Άπληστος Αλγόριθμος (Greedy Heuristic Algorithm – GH) ανήκει στην κατηγορία ευρετικών αλγορίθμων και αναζητά τοπικά βέλτιστα, βελτιστοποιώντας περαιτέρω μία τοπικά βέλτιστη λύση αναζητώντας το ολικό βέλτιστο του προβλήματος. Ξεκινά με την ταξινόμηση όλων των ακμών του προβλήματος και στη συνέχεια επιλέγει αυτή με το ελάχιστο κόστος. Συνεχίζει επιλέγοντας την καλύτερη επόμενη επιλογή, δεδομένου ότι δε σχηματίζονται βρόχοι κατά τη διαδικασία αυτή. Η υπολογιστική πολυπλοκότητα του Άπληστου Αλγορίθμου υπολογίζεται από τη σχέση $N^2 *\log_2(N)$ χωρίς να υπάρχει εγγύηση ότι θα βρεθεί μια βέλτιστη λύση. Από την άλλη πλευρά, ο Άπληστος Αλγόριθμος ολοκληρώνεται έπειτα από ένα εύλογο αριθμό βημάτων και διατηρεί τη βέλτιστη λύση μέσα στο 15-20% του κατώτερου ορίου Held-Karp [59].

- **Ευρετικοί Αλγόριθμοι Εισαγωγής**

Μια άλλη διαισθητικά ελκυστική προσέγγιση είναι να ξεκινήσει κάποιος με τη σχεδίαση διαδρομών που επισκέπτονται μόνο μικρά υποσύνολα των κόμβων του προβλήματος και στη συνέχεια να επεκτείνονται αυτοί οι κύκλοι εισάγοντας και τους υπόλοιπους κόμβους. Χρησιμοποιώντας αυτή την αρχή, κατασκευάζεται ένας κύκλος που περιέχει όλο και περισσότερους κόμβους του προβλήματος έως ότου εισαχθούν όλοι οι κόμβοι και βρεθεί ένας Χαμιλτονιανός κύκλος [36].

Οι Ευρετικοί Αλγόριθμοι Εισαγωγής (Insertion Heuristics) είναι αρκετά απλοί στη χρήση τους και υπάρχουν πολλές παραλλαγές από τις οποίες μπορεί να επιλεγεί η κατάλληλη για κάποιο πρόβλημα. Το βασικό στοιχείο αυτού του αλγορίθμου είναι να ξεκινήσει ο πωλητής πραγματοποιώντας μια διαδρομή που διέρχεται από ένα υποσύνολο όλων των πόλεων του προβλήματος (sub-tour) και στη συνέχεια, μέσω μιας ευρετικής διαδικασίας, να εισαχθούν στη διαδρομή και οι υπόλοιπες πόλεις του προβλήματος. Η αρχική διαδρομή που ακολουθείται είναι συχνά ένα τρίγωνο. Μπορεί επίσης κανείς σαν αρχική διαδρομή να επιλέξει τη σύνδεση δύο μόνο πόλεων. Η πολυπλοκότητα αυτού του τύπου ευρετικής προσέγγισης δίνεται ως $O(n^2)$. Τα βήματα αυτής της ευριστικής μεθόδου είναι [65]:

1. Επιλέξτε τη μικρότερη ακμή του γράφου του προβλήματος και θεωρήστε την ως την αρχική σας διαδρομή,
2. Επιλέξτε την πόλη που δεν εμπεριέχεται στην υφιστάμενη διαδρομή και που έχει τη μικρότερη απόσταση από οποιαδήποτε από τις πόλεις που ανήκουν σε αυτή,
3. Βρείτε μια ακμή στην υφιστάμενη διαδρομή τέτοια ώστε το κόστος εισαγωγής της επιλεγμένης πόλης στη διαδρομή να είναι το ελάχιστο,
4. Επαναλάβετε το βήμα 2 έως ότου δεν παραμείνουν άλλες πόλεις εκτός της υφιστάμενης διαδρομής.

Φυσικά, υπάρχουν πολλές δυνατότητες για την εφαρμογή ενός τέτοιου συστήματος εισαγωγής όπως αυτό που περιγράφηκε παραπάνω. Η κύρια διαφορά μεταξύ τους είναι η επιλογή του κανόνα επιλογής στο βήμα 2. Αντί για την ακμή με το μικρότερο μήκος, στο βήμα 1, ως αρχική διαδρομή μπορεί να επιλεγεί ένας κύκλος με τρεις κόμβους ή, σε εκφυλισμένες περιπτώσεις, ένας βρόχος ή ένα άκρο. Ο επιλεγμένος κόμβος που εισάγεται, εισάγεται συνήθως στον κύκλο, στο σημείο που προκαλεί τη μικρότερη αύξηση του μήκους του κύκλου [36].

- **Ευρετικός Αλγόριθμος Χριστοφίδη**

Οι περισσότεροι ευρετικοί αλγόριθμοι μπορούν να εγγυηθούν μόνο μια εφικτή λύση ή μια λύση αρκετά κοντά στη βέλτιστη. Ο Χριστοφίδης επέκτεινε μία από αυτές τις ευρετικές προσεγγίσεις αναπτύσσοντας έναν αλγόριθμο που είναι γνωστός ως Ευρετικός Αλγόριθμος Χριστοφίδη (Christofide Heuristic). Η πολυπλοκότητα αυτής της προσέγγισης είναι $O(n^3)$. Τα βήματά της δίνονται παρακάτω [65]:

1. Δημιουργήστε ένα Ελάχιστα Εκτεταμένο Δένδρο (Minimum Spanning Tree – MST) από το σύνολο όλων των πόλεων του προβλήματος.
2. Δημιουργήστε ένα Ελάχιστου Βάρους Ταίριασμα (Minimum-Weight Matching – MWM) από το σύνολο των κόμβων με μονό βαθμό. Προσθέστε το MST μαζί με το MWM.

3. Δημιουργήστε έναν κύκλο Euler από το συνδυασμένο γράφο και διασχίστε τον χρησιμοποιώντας συντομεύσεις για να αποφύγετε τους κόμβους που έχετε ήδη επισκεφθεί.

Δοκιμές έχουν δείξει ότι ο αλγόριθμος του Χριστοφίδη τείνει να βρίσκει μια λύση περίπου 10% πάνω από το κάτω όριο Held-Karp. Χρησιμοποιεί τεχνικές K-opt, ως επί το πλήστον 2-opt και 3-opt. Στην πρώτη περίπτωση, στο βήμα 3, επιλέγονται 2 ακμές του γράφου (που συνήθως διασταυρώνονται) για να γίνει ανταλλαγή ενώ στη δεύτερη περίπτωση επιλέγονται 3 ακμές που εναλλάσσονται ώστε να προκύψει μια μικρότερη διαδρομή [66].

- **Αναζήτηση Ταμπού**

Η Αναζήτηση Ταμπού (Tabu Search – TS) που προτάθηκε αρχικά από το Glover είναι πολύ αποτελεσματική για την επίλυση προβλημάτων NP-δύσκολων [67]. Ως τεχνική τοπικής αναζήτησης, η TS μετακινείται από μια τρέχουσα λύση σε μια καλύτερη λύση στη γειτονιά της τρέχουσας διερευνώντας το χώρο των λύσεων ύστερα από κάθε επανάληψη. Η κύρια αρχή της μεθόδου TS είναι η αποδοχή γειτονικών λύσεων που επιδεινώνουν την τρέχουσα τιμή της αντικειμενικής συνάρτησης προκειμένου να ξεφύγουν από την παγίδευση σε κάποιο τοπικό βέλτιστο. Η μέθοδος TS δημιουργεί μια λίστα (λίστα Tabu) με τις λύσεις που εξέτασε πρόσφατα ή με τις εφαρμοζόμενες κινήσεις που πραγματοποίησε πριν από συγκεκριμένο αριθμό επαναλήψεων (Tabu Tenure). Αυτός ο κατάλογος Tabu έχει δύο κύριους σκοπούς: Πρώτον, να αποτρέψει την επιστροφή στις πιο πρόσφατες διαδρομές που έχουν ελεγχθεί, προκειμένου να αποφύγει την επαναληψιμότητα και δεύτερον, να μεταφέρει την αναζήτηση προς περιοχές του χώρου των λύσεων που δεν έχουν εξερευνηθεί ακόμη και έχουν υψηλή πιθανότητα να περιέχουν καλές λύσεις. Όταν η αναζήτηση επιχειρεί να μετακινηθεί προς μια λύση που ανήκει σε μία από τις προαναφερθείσες λίστες τότε η μετάβαση αυτή απαγορεύεται, εκτός και αν με αυτή την κίνηση βελτιωθεί η καλύτερη λύση που έχει βρεθεί μέχρι εκείνο το σημείο κατά την αναζήτηση (κριτήριο αναρρόφησης – aspiration criterion).

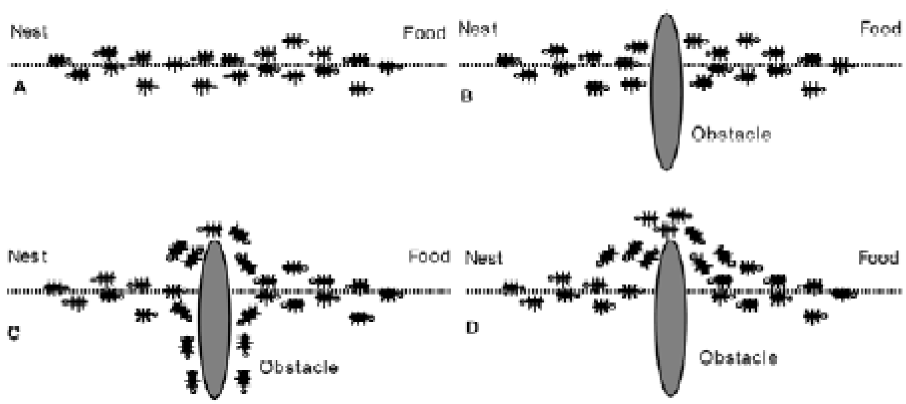
Η επιτυχής εφαρμογή της μεθόδου TS απαιτεί μια ισχυρή τεχνική για την εντατικοποίηση και τη διαφοροποίηση της αναζήτησης. Η εντατικοποίηση είναι μια περιεκτική εξερεύνηση κάποιας περιοχής του χώρου των πιθανών λύσεων, συνήθως στη γειτονιά μιας καλής λύσης ενώ η διαφοροποίηση οδηγεί την αναζήτηση σε αρκετά υποσχόμενες περιοχές του χώρου των εφικτών λύσεων που δεν έχουν εξερευνηθεί ακόμη [68], [69].

- **Βελτιστοποίηση Αποικίας Μυρμηγκιών**

Μια από τις σημαντικότερες ευριστικές προσεγγίσεις που δημιούργησε ο Dorigo τα τελευταία χρόνια (1992) και έχει αποδείξει την αξία της στην επίλυση του Προβλήματος του Περιοδεύοντος Πωλητή και σε άλλα προβλήματα είναι η Βελτιστοποίηση Αποικίας Μυρμηγκιών (Ant Colony Optimization – ACO) [70] που χρησιμοποιείται παρόμοια με τον Γενετικό Αλγόριθμο (GA) και τη Βελτιστοποίηση Σμήνους Σωματιδίων (PSO) για την επίλυση των συνδυαστικών προβλημάτων βελτιστοποίησης. Αυτή η πληθυσμιακή προσέγγιση έχει εφαρμοστεί επιτυχώς σε αρκετά NP-δύσκολα συνδυαστικά προβλήματα βελτιστοποίησης όπως η Δρομολόγηση Οχήματος [71], τα δίκτυα επικοινωνιών [72], το Πρόβλημα Διαδοχικής Παραγγελίας [73], το Πρόβλημα Τετραγωνικής Αντιστοίχισης [74] και άλλα.

Ο Αλγόριθμος ACO, είναι εμπνευσμένος από τη φύση αφού προσομοιώνει τη φυσική διαδικασία που ακολουθούν τα μυρμήγκια για την εξεύρεση τροφής και την εφαρμόζει για την επίλυση συνδυαστικών προβλημάτων βελτιστοποίησης για τα οποία δεν έχει βρεθεί ακόμη ένας αποτελεσματικός αλγόριθμος. Μελέτες σε πραγματικά μυρμήγκια έχουν δείξει ότι παρότι τα μυρμήγκια δεν έχουν την αίσθηση της όρασης μπορούν να βρουν τη συντομότερη διαδρομή από μια πηγή τροφής προς τη φωλιά τους. Κάποια εξατμιζόμενη ουσία, που ονομάζεται φερομόνη, εκκρίνεται από τα μυρμήγκια όταν μετακινούνται από το ένα μέρος στο άλλο και τα βοηθάει ώστε να βρουν τη συντομότερη διαδρομή. Τα μυρμήγκια εκκρίνουν αυτή τη χημική ουσία αρχικά για να καθοδηγούν τα άλλα μυρμήγκια, τα οποία πρόκειται να βγουν από τη φωλιά αργότερα, και επιπλέον, για την αναγνώριση της διαδρομής επιστροφής στη φωλιά τους. Ως εκ τούτου, η διαδρομή που ακολουθούν τα μυρμήγκια χαρακτηρίζεται από αυτή τη χημική ουσία, έτσι ώστε μετά από λίγο καιρό, ταυτόχρονα, περισσότερα μυρμήγκια να περάσουν από αυτή τη συντομότερη διαδρομή και να παραμείνουν περισσότερες φερομόνες σε αυτή τη βραχύτερη διαδρομή. Επιπλέον, το ένστικτο των μυρμηγκιών με μεγαλύτερη πιθανότητα επιλέγει τη διαδρομή, η οποία έχει περισσότερες φερομόνες από άλλες [75].

Η Εικόνα 11 που ακολουθεί δείχνει πως τα μυρμήγκια καταφέρνουν να βρίσκουν τη συντομότερη διαδρομή μεταξύ της φωλιάς τους και μιας πηγής τροφής χρησιμοποιώντας τις φερομόνες. Όταν ένα εμπόδιο εμφανίζεται μεταξύ της φωλιάς τους και της τροφής έχουν δύο επιλογές ώστε να παρακάμψουν το εμπόδιο. Και οι δύο επιλογές έχουν την ίδια πιθανότητα να επιλεγούν. Σταδιακά όμως η εναπόθεση των φερομονών γίνεται με μεγαλύτερο ρυθμό στη συντομότερη διαδρομή, κάτι που οδηγεί στο να την ακολουθούν όλο και περισσότερα μυρμήγκια ώσπου στο τέλος, όλα τα μυρμήγκια ακολουθούνε μόνο τη συντομότερη διαδρομή.



Εικόνα 11: Από πάνω αριστερά προς κάτω δεξιά: (A) Μυρμήγκια ακολουθούν μια διαδρομή μεταξύ της φωλιάς τους και μιας πηγής τροφής, (B) Ένα εμπόδιο εμφανίζεται στη διαδρομή. Τα μυρμήγκια μπορούν (με την ίδια πιθανότητα) είτε να στρίψουν αριστερά, είτε δεξιά, ώστε να το αποφύγουν, (Γ) Η φερομόνη εναπόθειται πιο γρήγορα στη συντομότερη διαδρομή, (Δ) Στο τέλος, όλα τα μυρμήγκια επιλέγουν τη συντομότερη διαδρομή [75].

Στη συνέχεια, παρουσιάζεται η μαθηματική διατύπωση του ACO αλγορίθμου:

Στην αρχή του αλγορίθμου m στον αριθμό μυρμήγκια τοποθετούνται σε τυχαία επιλεγμένους κόμβους του γράφου. Σε κάθε βήμα του αλγορίθμου, τα μυρμήγκια μετακινούνται σε έναν νέο κόμβο με πιθανότητα που δίνεται από τον τυχαίο αναλογικό κανόνα που ορίζεται ως [76]:

$$p_{ij} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{l \in N_k} (\tau_{il})^{\alpha}(\eta_{il})^{\beta}}$$

Όπου:

τ_{ij} η ποσότητα φερομόνης στην ακμή μεταξύ των κόμβων i και j,

η_{ij} η ελκυστικότητα της ακμής από τον κόμβο i στον j υπολογισμένη από τη σχέση: 1/μήκος ακμής,

α,β παράμετροι της μεθόδου που χρησιμοποιούνται για την εκτίμηση της σχετικής επίδρασης της φερομόνης και της ελκυστικότητας της ακμής,

N_k το σύνολο των κόμβων που το k-οστό μυρμήγκι δεν έχει ακόμα επισκεφτεί.

Μετά την επίσκεψη όλων των κόμβων (δηλαδή όταν το σύνολο των μη επισκέψιμων ακόμα κόμβων είναι κενό) τα μυρμήγκια επιστρέφουν στους αρχικούς τους κόμβους και η ποσότητα φερομόνης σε κάθε ακμή ενημερώνεται σύμφωνα με τον ακόλουθο τύπο [76]:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

Όπου:

ρ η παράμετρος εξάτμισης της φερομόνης,

$\Delta\tau_{ij}^{k}$ υπολογίζεται από τη σχέση:

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k} & \text{αν το k-οστό μυρμήγκι χρησιμοποιεί την ακμή (i,j) στο δικό του Χαμιλτονιανό κύκλο,} \\ 0 & \text{σε κάθε διαφορετική περίπτωση.} \end{cases}$$

Όπου:

Q μια σταθερά επιλεγμένη πριν από την εκτέλεση του αλγορίθμου,

L_k το μήκος του Χαμιλτονιανού κύκλου του k-οστού μυρμηγκιού.

Η μελέτη των μυρμηγκιών δείχνει ότι η απλή νοημοσύνη ενός σμήνους, που χρησιμοποιείται και από τα μυρμήγκια για την εξεύρεση τροφής, οδηγεί στην επίλυση των δύσκολων συνδυαστικών προβλημάτων και στην επίτευξη μιας λύσης, η οποία βρίσκεται αρκετά κοντά στη βέλτιστη. Ο αλγόριθμος ACO έχει αναπτυχθεί πολύ τα τελευταία χρόνια. Η τροποποίηση της μεθόδου ενημέρωσης των τοπικών και ολικών φερομονών και η κατανομή των μυρμηγκιών στους κόμβους είναι μερικά από τα παραδείγματα εξέλιξης του αλγορίθμου [75].

- **Βελτιστοποίηση Σμήνους Σωματιδίων**

Η Βελτιστοποίηση Σμήνους Σωματιδίων (Particle Swarm Optimization – PSO) πρωτοεμφανίστηκε από τους Kennedy & Eberhart το 1995 [77] και είναι από τους ευρέως χρησιμοποιούμενους εξελικτικούς αλγορίθμους, αλλά γενικά διαφέρει από αυτούς. Είναι εμπνευσμένος από την αναδυόμενη κίνηση ενός κοπαδιού πτηνών που αναζητούν τροφή και τη συνεταιριστική και ανταγωνιστική κοινωνική συμπεριφορά των ζώων, η οποία έχει τα χαρακτηριστικά της νοημοσύνης του σμήνους, του εγγενούς παραλληλισμού, της επαναληπτικής μορφής ενώ είναι απλός από άποψη υπολογιστικού χρόνου [78]. Το PSO είναι αποτελεσματικό στην εξεύρεση καλών λύσεων για προβλήματα βελτιστοποίησης [79] καθιστώντας το ένα ακόμα ισχυρό εργαλείο στην επίλυση τέτοιων προβλημάτων πέρα από άλλους εξελικτικούς αλγορίθμους, όπως για παράδειγμα οι Γενετικοί Αλγορίθμοι (GA) [80]. Το PSO μπορεί να παρομοιαστεί αρχικά με έναν πληθυσμό σωματιδίων τυχαία τοποθετημένο σε έναν n-διάστατο χώρο αναζήτησης. Κάθε σωματίδιο του πληθυσμού έχει δύο παραμέτρους που το περιγράφουν, για παράδειγμα την παράμετρο ταχύτητας και την παράμετρο θέσης. Ο αλγόριθμος PSO είναι αναδρομικός, παρακινώντας τη συμπεριφορά κοινωνικής αναζήτησης ανάμεσα στα σωματίδια στον χώρο αναζήτησης, όπου κάθε σωματίδιο αντιπροσωπεύει ένα σημείο. Σε σύγκριση με άλλους Εξελικτικούς Αλγόριθμους, όπως οι GA, ο PSO έχει καλύτερες επιδόσεις αναζήτησης με ταχύτερα και πιο σταθερά ποσοστά σύγκλισης.

Η διατήρηση της ισορροπίας μεταξύ της αναζήτησης ολικών και τοπικών μεγίστων κατά τη διάρκεια όλων των διαδικασιών που ακολουθεί ένας αλγόριθμος βελτιστοποίησης είναι κρίσιμη για την επιτυχία αυτού του αλγορίθμου [81]. Όλοι οι Εξελικτικοί Αλγόριθμοι χρησιμοποιούν διάφορες μεθόδους για την επίτευξη αυτού του στόχου. Για να επιτευχθεί ισορροπία μεταξύ των δύο αναζητήσεων, οι Shi & Eberhart πρότειναν ένα PSO με βάση το βάρος αδράνειας στο οποίο η ταχύτητα κάθε σωματιδίου ενημερώνεται σύμφωνα με μια σταθερή εξίσωση [82]. Μια υψηλότερη τιμή του βάρους αδράνειας συνεπάγεται μεγαλύτερες αυξομειώσεις της ταχύτητας, πράγμα που σημαίνει ότι τα σωματίδια έχουν περισσότερες πιθανότητες να εξερευνήσουν νέες περιοχές αναζήτησης. Ωστόσο, μικρότερο βάρος αδράνειας σημαίνει μικρότερη μεταβολή της ταχύτητας και βραδύτερη ενημέρωση για το σωματίδιο με αποτέλεσμα να κινδυνεύει να παγιδευτεί σε τοπικές περιοχές αναζήτησης και σε τοπικά βέλτιστα [83].

Λόγω των καλών αποτελεσμάτων που δίνει ο αλγόριθμος, έχει εφαρμοστεί σε πολλούς τομείς όπως η αναγνώριση συστημάτων [84], τα νευρωνικά δίκτυα [85] και ο έλεγχος συστημάτων ([86] και [87]) [78]. Πλήθος άλλων πηγών με εφαρμογές του αλγορίθμου στους τομείς των ηλεκτρονικών υπολογιστών, των αυτοματοποιημένων συστημάτων ελέγχου, της θεωρίας επικοινωνίας, της επιχειρησιακής έρευνας, των μηχανολόγων μηχανικών, των πολιτικών μηχανικών, των καυσίμων και της ενέργειας, της ιατρικής μηχανικής, των χημικών μηχανικών και της μηχανικής της βιολογίας μπορεί να βρει όποιος ενδιαφέρεται στην εργασία των Zhang, Wang & Ji [88].

- **Προσομοιωμένη Ανόπτηση**

Η μέθοδος της Προσομοιωμένης Ανόπτησης (γνωστή στη διεθνή βιβλιογραφία ως Simulated Annealing – SA) είναι μια στοχαστική τεχνική για την προσέγγιση του ολικού βέλτιστου μιας συνάρτησης. Πιο συγκεκριμένα, πρόκειται για μια μετα-ευρετική τεχνική για την προσέγγιση της ολικά βέλτιστης λύσης όταν οι πιθανοί συνδυασμοί είναι πάρα πολλοί, αποφεύγοντας να εγκλωβιστούμε σε τοπικά βέλτιστα. Η μέθοδος Simulated Annealing είναι συχνά προτιμότερη από άλλες αντίστοιχες μεθόδους βελτιστοποίησης σε περιπτώσεις όπου είναι πιο σημαντικό να βρούμε μια λύση, σε συγκεκριμένο χρόνο, κοντά στην ολική βέλτιστη από το να βρούμε με ακρίβεια μια τοπική βέλτιστη λύση. Αυτή η μέθοδος για την επίλυση μαθηματικών προβλημάτων διατυπώθηκε για πρώτη φορά από τους Khachaturyan, Semenovskaya & Vainshtein [89] το 1979 και ανεξάρτητα από τους Kirkpatrick, Gelatt & Vecchi [90] το 1983.

Το όνομα και η έμπνευση της μεθόδου προέρχονται από μία τεχνική της μεταλλουργίας που περιλαμβάνει θέρμανση και ελεγχόμενη ψύξη ενός μετάλλου για την αύξηση του μεγέθους των κρυστάλλων του και τη μείωση των ελαττωμάτων τους. Και τα δύο αυτά χαρακτηριστικά ενός μετάλλου εξαρτώνται από τη θερμοκρασία στην οποία αυτό βρίσκεται. Τα άτομα μέσα στο υλικό έχουν υψηλές ενέργειες σε υψηλές θερμοκρασίες και επομένως έχουν υψηλότερη ελευθερία. Ενώ η θερμοκρασία μειώνεται, μειώνεται παράλληλα και η ενέργεια των ατόμων. Σκοπός της διαδικασίας σταδιακής ψύξης είναι η δημιουργία κρυστάλλων ομαλής δομής. Αυτό επιτυγχάνεται στην κατάσταση όπου τα άτομα έχουν ελάχιστη ενέργεια. Εάν, όμως, η ψύξη γίνει πολύ γρήγορα (διαδικασία γνωστή και ως rapid quenching), παρατηρούνται εκτεταμένες ανωμαλίες στην κρυσταλλική δομή των ατόμων. Το σύστημα, έτσι, δε φτάνει στην κατάσταση ελάχιστης ενέργειας και καταλήγει σε μία πολυκρυσταλλική δομή, η οποία έχει ενέργεια υψηλότερη από την ελάχιστη.

Η μέθοδος της Προσομοιωμένης Ανόπτησης προσομοιάζει την αργή ψύξη ως την αργή μείωση της πιθανότητας αποδοχής μιας χειρότερης λύσης, καθώς διερευνά το χώρο των λύσεων. Η αποδοχή χειρότερης λύσης είναι μια θεμελιώδης ιδιότητα της μετα-ευρετικής ανάλυσης επειδή επιτρέπει μια πιο εκτεταμένη έρευνα για τη βέλτιστη λύση. Ο στόχος είναι να φέρουμε το σύστημα, για παράδειγμα, μια εξίσωση ή ένα σύστημα εξισώσεων, από μια αυθαίρετη αρχική κατάσταση σε μια κατάσταση με την ελάχιστη δυνατή ενέργεια δηλαδή πλησίον της βέλτιστης λύσης.

Σε κάθε βήμα η Simulated Annealing βρίσκει μια γειτονική λύση της τρέχουσας κατάστασης και βάσει πιθανοτήτων αποφασίζει αν θα κάνει αυτή τη λύση αποδεκτή και, επομένως, θα την ορίσει ως τη νέα τρέχουσα κατάσταση ή όχι. Αν η νέα λύση είναι πιο κοντά στη βέλτιστη, τότε ασφαλώς και επιλέγεται ενώ διαφορετικά, η νέα και χειρότερη μπορεί να γίνει αποδεκτή βάσει μιας πιθανότητας που υπολογίζεται στα πλαίσια της μεθόδου. Η διαδικασία επαναλαμβάνεται πολλές φορές μειώνοντας σταδιακά την πιθανότητα να γίνει αποδεκτή μια χειρότερη λύση με το σύστημα, τελικά, να οδηγείται σε λύση η οποία, αν δεν είναι η βέλτιστη, τότε την προσεγγίζει ικανοποιητικά. Η επαναληπτική διαδικασία υπολογισμών ολοκληρώνεται όταν ολοκληρώσουμε ένα πλήθος κύκλων υπολογισμού (μειώνοντας σε κάθε κύκλο την αρχική πιθανότητα να γίνει αποδεκτή μια χειρότερη λύση) ή όταν σε ένα κύκλο υπολογισμού δεν προκύψει καμία νέα λύση, είτε καλύτερη είτε χειρότερη που να γίνεται αποδεκτή [2].

Σε κάθε δεδομένη στιγμή, η πιθανότητα p να επιλεγεί μια διαδρομή χειρότερη από αυτή που έχει βρεθεί μέχρι εκείνη τη στιγμή δίνεται από την παρακάτω σχέση:

$$p(x,x',T) = \begin{cases} e^{\left(-\left(f(x')-f(x)\right)\right)/T} & \text{όταν } f(x') \geq f(x) \\ 1 & \text{όταν } f(x') < f(x) \end{cases}$$

Όπου:

f(x) η αντικειμενική συνάρτηση της λύσης x

T η παράμετρος «θερμοκρασία» του αλγορίθμου

Κάθε επανάληψη του αλγορίθμου ξεκινάει με μία θερμοκρασία T που παίρνει υψηλή τιμή και σταδιακά μειώνεται με κάθε επόμενη επανάληψη του αλγορίθμου. Η Προσομοιωμένη Ανόπτηση μπορεί να επεκταθεί με τη σταδιακή θέρμανση δηλαδή την αύξηση της θερμοκρασίας. Ο αλγόριθμος σταματά όταν πληρείται κάποιο κριτήριο διακοπής (χρόνος, αριθμός επαναλήψεων χωρίς βελτίωση βέλτιστης λύσης κλπ) [90].

Λίστα με διάφορες άλλες ευρετικές μεθόδους αναζήτησης που χρησιμοποιούνται για την επίλυση του Προβλήματος του Περιοδεύοντος Πωλητή μπορούν να βρεθούν στη διεθνή βιβλιογραφία. Ενδεικτικά, χωρίς να επεκταθούμε περαιτέρω θα αναφέρουμε μερικές από αυτές με τον αγγλικό τους όρο: Great Deluge Algorithms, Simulated Tunneling, Scatter Search, Space Filling Curves κ.α. [91].

3.3. Κριτήρια και Σύγκριση

Υπάρχουν αρκετές εργασίες στη διεθνή βιβλιογραφία που προσπαθούν να συγκρίνουν τους διάφορους αλγορίθμους που χρησιμοποιούνται για την επίλυση του Προβλήματος του Περιοδεύοντος Πωλητή και άλλων συναφών προβλημάτων βελτιστοποίησης. Για το σκοπό της σύγκρισης χρησιμοποιούνται τα παρακάτω κριτήρια:

- **Χρόνος επίλυσης**

Ένα από τα κύρια κριτήρια που χρησιμοποιούνται για να αξιολογήσουν έναν αλγόριθμο είναι ο χρόνος που χρειάζεται για να επιλύσει ένα TSP. Όπως έχει ήδη αναφερθεί οι ακριβείς αλγόριθμοι βρίσκουν το ολικό βέλτιστο αλλά για να το πετύχουν αυτό χρειάζονται πάρα πολύ χρόνο. Αντίθετα, οι ευρετικοί αλγόριθμοι βρίσκουν λύσεις κοντά στη βέλτιστη μέσα σε μερικά δευτερόλεπτα. Έτσι, συχνά, ο χρόνος που χρειάζεται ένας αλγόριθμος για να επιλύσει ένα συγκεκριμένο TSP καθορίζει και την αξία του έναντι άλλων αντίστοιχων αλγορίθμων.

- **Επαναλήψεις μεθόδου**

Ένα άλλο κριτήριο που χρησιμοποιείται συχνά κατά τη συγκριση των διαφόρων αλγορίθμων μεταξύ τους είναι ο αριθμός των επαναλήψεων που απαιτούνται να εκτελεστούν έως ότου προκύψει το τελικό αποτέλεσμα που δίνει ο αλγόριθμος. Υπάρχουν αλγόριθμοι που μπορούν να καταλήξουν σε μία τελική λύση μέσα σε μερικές μόνο επαναλήψεις και άλλοι που χρειάζονται μεγαλύτερο αριθμό επαναλήψεων.

- **Πολυπλοκότητα προβλήματος**

Για να συγκριθούν δύο ή περισσότεροι αλγόριθμοι μεταξύ τους χρησιμοποιούνται ορισμένα TSP, που συνήθως εμπεριέχονται στο TSPLIB, και στα οποία σημαντικό ρόλο παίζει η πολυπλοκότητά τους. Όπως είναι εύκολα αντιληπτό, για ένα TSP με διπλάσιο αριθμό κόμβων από ένα άλλο, ο αλγόριθμος που εξετάζεται θα χρειάζεται και περισσότερο χρόνο έως ότου καταλήξει σε μία λύση

- **Ακρίβεια επίλυσης**

Ένα άλλο κριτήριο που χρησιμοποιείται συχνά όταν πρόκειται να αξιολογηθούν δύο αλγόριθμοι είναι η ακρίβεια της λύσης που δίνουν. Εφόσον, συνήθως, χρησιμοποιούνται TSP που εμπεριέχονται στην TSPLIB, οι ερευνητές είναι σε θέση να γνωρίζουν εκ των προτέρων ποια είναι η βέλτιστη λύση και επομένως μπορούν να προσδιορίσουν με ακρίβεια πόσο απέχει από τη βέλτιστη η λύση που έδωσε ο αλγόριθμος.

Στη συνέχεια ακολουθούν αρισμένες εργασίες όπου οι ερευνητές προσπαθούν να συγκρίνουν δύο ή περισσότερους ευρετικούς αλγορίθμους με βάση κάποια από τα κριτήρια που αναλύθηκαν παραπάνω.

Στην εργασία των Abdulmarim & Alshammari [59] εξετάζονται και συγκρίνονται μεταξύ τους οι αλγόριθμοι του Πλησιέστερου Γείτονα, ο Γενετικός Αλγόριθμος και ο Άπληστος Αλγόριθμος. Κάθε αλγόριθμος προσομοιώνεται χρησιμοποιώντας το MATLAB και η κάθε λύση που βρίσκεται συγκρίνεται με την ολική βέλτιστη. Χρησιμοποιούνται τρία σενάρια: 20, 100 και 1.000 πόλεων εντός των ΗΠΑ. Κατά τη σύγκριση των αλγορίθμων συμπεραίνεται ότι, αν και ο Άπληστος Αλγόριθμος χρησιμοποιεί τις λιγότερες επαναλήψεις για να λύσει το TSP, το αποτέλεσμά του είναι το πλησιέστερο στη βέλτιστη λύση. Το αποτέλεσμα αυτό προκύπτει ύστερα από έναν λογικό αριθμό επαναλήψεων. Από την άλλη πλευρά, ο Γενετικός Αλγόριθμος δεν καταφέρνει να εντοπίσει τη συντομότερη διαδρομή αλλά εξακολουθεί να παρέχει μια εναλλακτική λύση αν και με συνολικό μήκος διαδρομής μεγαλύτερο από το ολικό ελάχιστο. Αυτό δεν είναι ένα μη αναμενόμενο αποτέλεσμα, δεδομένου ότι η γενετική διαδικασία χρησιμοποιεί μεταλλάξεις μεταξύ των πόλεων για να βρει την καλύτερη διαδρομή, αλλά αυτές οι μεταλλάξεις είναι τυχαίες και ως εκ τούτου δεν παρέχουν καμία εγγύηση ότι η διαδρομή που βρίσκεται στο τέλος είναι η βέλτιστη δυνατή. Επομένως, συμπεραίνεται ότι, για μεγάλο αριθμό κόμβων, ο γενετικός αλγόριθμος δεν συγκλίνει σε μια έγκυρη λύση [59].

Συγκεντρωτικά, τα αποτελέσμα αυτής της έρευνας όπου συγκρίνονται ο Αλγόριθμος του Πλησιέστερου Γείτονα, ο Γενετικός Αλγόριθμος και ο Άπληστος Αλγόριθμος φαίνονται στον πίνακα που ακολουθεί (Πίνακας 1).

Πίνακας 1: Σύγκριση των αλγορίθμων του Πλησιέστερου Γείτονα, του Γενετικού Αλγόριθμου και του Άπληστου Αλγόριθμου [59]

TSP 100 πόλεων			
Αλγόριθμος	Μήκος λύσης (χλμ)	Χρόνος Επίλυσης (sec)	Επαναλήψεις αλγορίθμου
Πλησιέστερος Γείτονας	26664	2.5	100
Γενετικός Αλγόριθμος	25479	45	10000
Άπληστος Αλγόριθμος	23311	0.07	18
TSP 1.000 πόλεων			
Αλγόριθμος	Μήκος λύσης (χλμ)	Χρόνος Επίλυσης (sec)	Επαναλήψεις αλγορίθμου
Πλησιέστερος Γείτονας	83938	95.5	100
Γενετικός Αλγόριθμος	282866	468	10000
Άπληστος Αλγόριθμος	72801	127	151

Στην εργασία των Arora, Agarwal &Tanwar [92] επιχειρείται η σύγκριση μεταξύ του Γενετικού Αλγορίθμου και του Αλγορίθμου του Πλησιέστερου Γείτονα. Για να εξεταστεί η αποτελεσματικότητα των δύο αυτών αλγορίθμων χρησιμοποιήθηκαν τρεις διαφορετικοί γράφοι 25, 50 και 100 πόλεων αντίστοιχα. Στα αποτελέσματα της έρευνάς τους, ως κριτήρια αξιολόγησης χρησιμοποιήθηκε ο χρόνος που απαιτήθηκε ώστε ο κάθε αλγόριθμος να καταλήξει σε μία λύση καθώς και το μήκος αυτής της λύσης. Οι ερευνητές επισημαίνουν ότι ως προς το μήκος της διανυόμενης απόστασης της τελικής λύσης των αλγορίθμων, ο Γενετικός Αλγόριθμος δίνει καλύτερα αποτελέσματα από τον Αλγόριθμο του Πλησιέστερου Γείτονα. Από την άλλη μεριά ο Αλγόριθμος του Πλησιέστερου Γείτονα καταλήγει σε μία λύση ταχύτερα απ' ότι ο Γενετικός Αλγόριθμος.

Τα αποτελέσματα της εργασίας τους συνοψίζονται στον πίνακα που ακολουθεί (Πίνακας 2) όπου συγκρίνονται ο χρόνος επίλυσης και το μήκος της βέλτιστης λύσης που δίνει ο Αλγόριθμος του Πλησιέστερου Γείτονα και ο Γενετικός Αλγόριθμος για τους τρεις γράφους των 25, 50 και 100 πόλεων αντίστοιχα.

Πίνακας 2: Σύγκριση Αλγορίθμου Πλησιέστερου Γείτονα και Γενετικού Αλγορίθμου [92]

Αλγόριθμος	Αριθμός Πόλεων	Μήκος Λύσης (χλμ)	Χρόνος Επίλυσης (sec)
Πλησιέστερος Γείτονας	25	42.2683	0.8631
	50	77.2339	6.6354
	100	86.4397	2.9486
Γενετικός Αλγόριθμος	25	41.3736	12.5727
	50	62.3458	14.579
	100	82.9532	21.1832

Ο Ondřej Míča, στην εργασία του [93], εξετάζει τέσσερις κλασικές μετα-ευρετικές μεθόδους: Tabu Search (TS), Simulated Annealing (SA), Genetic Algorithm (GA) και Ant Colony Optimization (ACO) και συγκρίνει τους αλγορίθμους χρησιμοποιώντας τις διαφορές μεταξύ της βέλτιστης λύσης που δίνει η κάθε μέθοδος και του ολικού βέλτιστου του προβλήματος. Τα αποτελέσματα που υπολογίζονται σε διάφορες τυπικές περιπτώσεις του Προβλήματος του Περιοδεύοντος Πωλητή δείχνουν την αποτελεσματικότητα των διαφόρων μεθόδων.

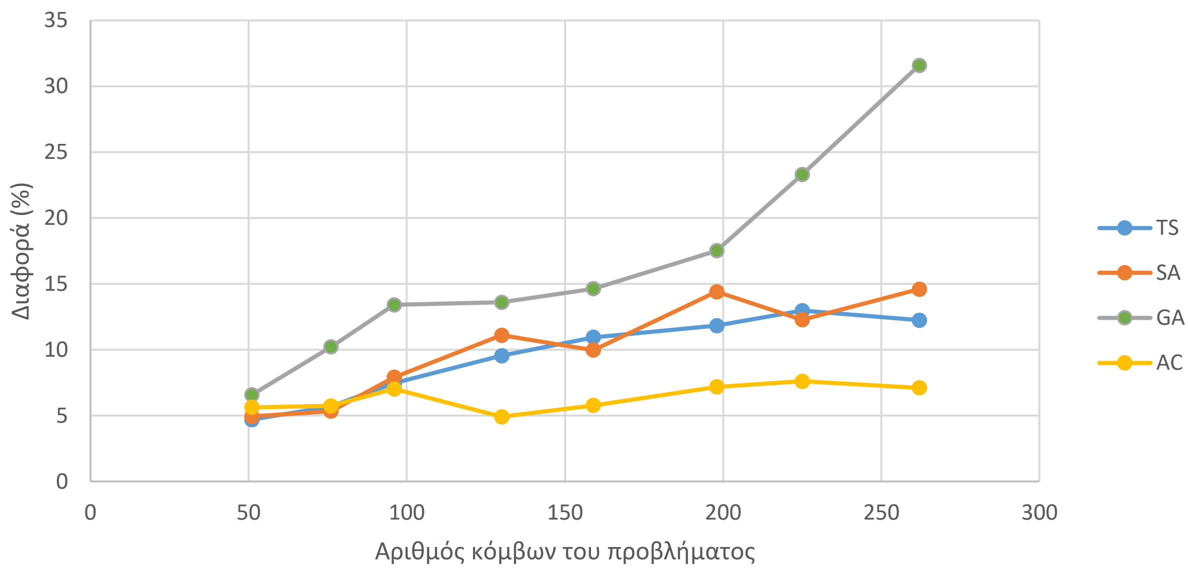
Οι τέσσερις μεταευρετικές μέθοδοι ενσωματώνονται στην γλώσσα Java. Για τη σύγκριση και την αξιολόγηση των αποτελεσμάτων χρησιμοποιήθηκε η TSPLIB η οποία είναι μια βιβλιοθήκη δειγμάτων για το Πρόβλημα του Περιοδεύοντος Πωλητή χρησιμοποιώντας διάφορες πηγές και διάφορους τύπους του προβλήματος. Αυτή η βιβλιοθήκη περιείχε τότε 112 περιπτώσεις συμμετρικών TSP με τις βέλτιστες λύσεις τους. Για την εργασία του, ο συγγραφέας, επέλεξε να εξετάσει 8 διαφορετικές περιπτώσεις TSP με διαφορετικό αριθμό κόμβων η κάθε περίπτωση (51, 76, 96, 130, 198, 225, 262 κόμβους αντίστοιχα).

Για την κάθε περίπτωση του TSP που εξετάστηκε πραγματοποιήθηκαν 500 κύκλοι για κάθε μία εκ των περιγραφόμενων μεθόδων και κάθε κύκλος περιορίστηκε στις 500 επαναλήψεις. Έτσι, μαζί πραγματοποιήθηκαν 16 000 κύκλοι επίλυσης του TSP για έλεγχο των μεθόδων (4 μέθοδοι Χ 500 κύκλοι Χ 8 διαφορετικές περιπτώσεις TSP). Για κάθε δοκιμασμένο συνδυασμό λήφθηκε η καλύτερη λύση και, στη συνέχεια, αυτή συγκρίθηκε με τη βέλτιστη λύση που δίνεται από την TSPLIB και προσδιορίστηκε η διαφορά (ως %) μεταξύ του καλύτερου αποτελέσματος που βρήκε ο αλγόριθμος και του βέλτιστου αποτελέσματος που έδωσε το TSPLIB.

Τα αποτελέσματα που προέκυψαν φαίνονται στον επόμενο πίνακα (Πίνακας 3) και στο διάγραμμα που ακολουθεί (Εικόνα 12).

Πίνακας 3: Διαφορές μεταξύ καλύτερης λύσης αλγορίθμων και βέλτιστης λύσης που έδωσε το TSPLIB (%) [93]

Αριθμός κόμβων	Αναζήτηση Tabu	Προσομοιωμένη Ανόπτηση	Γενετικός Αλγόριθμος	Βελτιστοποίηση Αποικίας Μυρμηγκιών
51	4.69	4.93	6.57	5.63
76	5.70	5.34	10.21	5.74
96	7.46	7.91	13.40	7.02
130	9.54	11.10	13.60	4.91
159	10.94	9.97	14.63	5.77
198	11.82	14.40	17.51	7.18
225	12.97	12.28	23.29	7.61
262	12.24	14.59	31.58	7.11



Εικόνα 12: Διαφορές μεταξύ καλύτερης λύσης που βρήκε ο αλγόριθμος και βέλτιστης λύσης που έδωσε το TSPLIB

Στα αποτελέσματα της εργασίας φαίνεται ότι για την περίπτωση του TSP με τους λιγότερους κόμβους, όλες οι χρησιμοποιούμενες μέθοδοι δίνουν λύση με παρόμοιες διαφορές. Αλλά, αυξάνοντας τον αριθμό των κόμβων του προβλήματος, ο Γενετικός Αλγόριθμος καθίσταται όλο και πιο ανακριβής. Η Tabu Search και η μέθοδος Simulated Annealing δίνουν λύσεις με πολύ παρόμοιες διαφορές. Το καλύτερο αποτέλεσμα σε όλες σχεδόν τις περιπτώσεις δίνει ο αλγόριθμος Βελτιστοποίησης Αποικίας Μυρμηγκιών.

Έτσι, το καλύτερο αποτέλεσμα της επίλυσης του Προβλήματος του Περιοδεύοντος Πωλητή από όλες τις συγκρινόμενες μεθόδους δίνει η Ant Colony Optimization. Το μειονέκτημα αυτής της μεθόδου είναι η μεγαλύτερη δυσκολία που εμπεριέχει η εφαρμογή αυτού του αλγορίθμου. Είναι επίσης σημαντικό να αναφερθεί ότι όλες οι μέθοδοι είναι πολύ ευαίσθητες στις ρυθμίσεις παραμέτρων. Έτσι με διαφορετική ρύθμιση μπορούν να δώσουν καλύτερα ή πολύ χειρότερα αποτελέσματα [93].

Οι Dostal & Kratochvil στην εργασία τους [94] συγκρίνουν μεταξύ τους 10 διαφορετικούς αλγορίθμους που χρησιμοποιούνται για την επίλυση του Προβλήματος του Περιοδεύοντος Πωλητή. Για το σκοπό αυτό χρησιμοποίησαν ένα γράφο 10 πόλεων και εξέτασαν το χρόνο που χρειάστηκε κάθε αλγόριθμος για να καταλήξει σε μία λύση, το συνολικό μήκος αυτής της λύσης, εαν αυτή η λύση είναι το ολικό βέλτιστο καθώς και τον αριθμό των προσπαθειών που απαιτήθηκαν από τον κάθε αλγόριθμο ώστε να καταλήξει στην τελική του λύση. Οι αλγόριθμοι που εξετάστηκαν καθώς και τα αποτελέσματα που προέκυψαν φαίνονται στον πίνακα που ακολουθεί (Πίνακας 4).

Πίνακας 4: Σύγκριση 10 μεθόδων επίλυσης ενός TSP 10 κόμβων [94]

Αλγόριθμος	Χρόνος	Μήκος λύσης	Προσπάθειες	Ολικό Βέλτιστο
Exhaustive	426.214	2627	1	Ναι
Back Tracking	21.549	2627	1	Ναι
Random Search	0.019	3438	20	Όχι
Greedy	0.020	2627	1	Ναι
Hill Climbing	0.005	2627	10	Ναι
Simulated Annealing	0.016	2914	20	Όχι
Tabu Search	0.385	2627	2	Ναι
Ant Colony Optimization	1.092	2627	1	Ναι
Genetic	1.337	2658	20	Όχι
Particle Swarms Optimization	1.753	2627	3	Ναι

4. Επιχειρησιακές Εφαρμογές

Στη διεθνή βιβλιογραφία υπάρχει πλήθος παραδειγμάτων από προβλήματα που αντιμετωπίζουν οι βιομηχανίες ή συνδέονται με προβλήματα της καθημερινότητας και μπορούν να επιλυθούν βάσει κάποιας παραλλαγής του Προβλήματος του Περιοδεύοντος Πωλητή. Με αυτό τον τρόπο επιτυγχάνεται εξοικονόμηση χρημάτων, χρόνου και άλλων πόρων. Στη συνέχεια αναφέρονται ορισμένα από αυτά τα παραδείγματα, χωρισμένα σε υποκεφάλαια αναλόγως με ποια παραλλαγή του βασικού TSP συνδέονται.

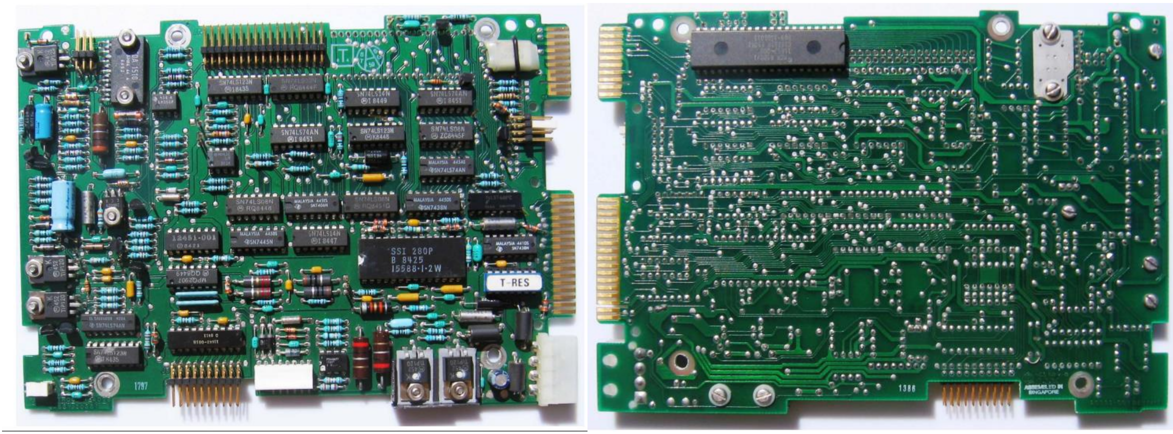
4.1. Εφαρμογές του TSP και Διασύνδεση με άλλα Προβλήματα

Στη συνέχεια αυτού του κεφαλαίου ακολουθούν παραδείγματα εφαρμογών του Προβλήματος του Περιοδεύοντος Πωλητή στη γενική μορφή του.

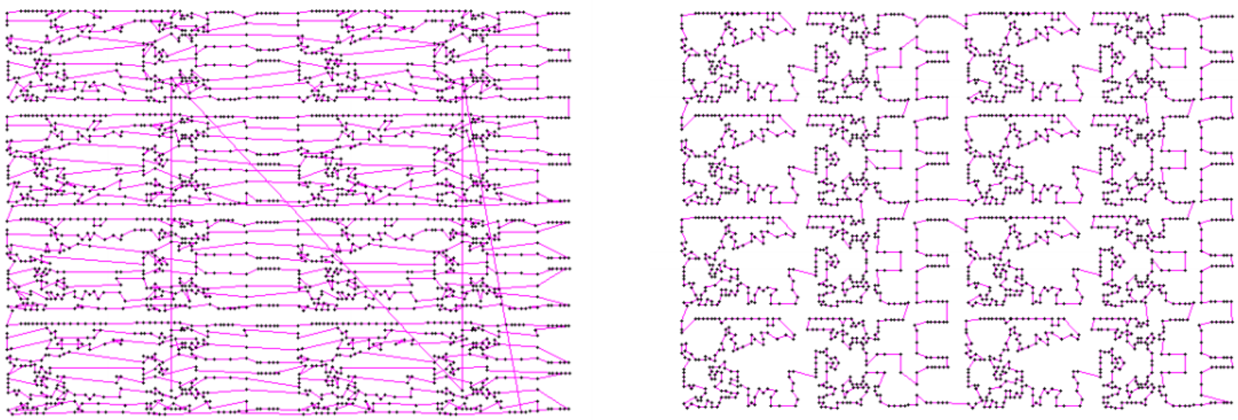
- **Διάτρηση Πλακετών Τυπωμένων Κυκλωμάτων**

Μια άμεση εφαρμογή του TSP είναι το πρόβλημα διάτρησης πλακετών τυπωμένων κυκλωμάτων (Printed Circuit Boards - PCBs) [95]. Για να συνδεθεί ένας αγωγό ενός στρώματος με έναν αγωγό ενός άλλου στρώματος ή για να τοποθετηθούν οι ακίδες των ολοκληρωμένων κυκλωμάτων, πρέπει να διανοιχθούν οπές πάνω στην πλακέτα. Οι οπές αυτές μπορεί να έχουν διαφορετικά μεγέθη. Για να διανοιχθούν δύο οπές διαφορετικών διαμέτρων διαδοχικά, όμως, η κεφαλή του μηχανήματος θα πρέπει να μετακινηθεί σε ένα κιβώτιο εργαλείων και να αλλαχθεί ο εξοπλισμός διάτρησης. Αυτό είναι πολύ χρονοβόρο. Έτσι είναι προφανές ότι πρέπει να επιλεχθεί κάποια διάμετρος, να διανοιχθούν όλες οι οπές της διαμέτρου αυτής και στη συνέχεια να αλλαχθεί η κεφαλή του τρυπανιού, να διανοιχθούν οι οπές της επόμενης διαμέτρου κλπ. Συνεπώς, αυτό το πρόβλημα διάνοιξης οπών μπορεί να θεωρηθεί ως μια σειρά TSPs, ενός για κάθε διάμετρο οπών, όπου οι «πόλεις» είναι η αρχική θέση και το σύνολο όλων των οπών που μπορούν να διανοιχθούν με την ίδια κεφαλή του τρυπανιού. Η «απόσταση» μεταξύ δύο πόλεων δίνεται από το χρόνο που χρειάζεται για να μετακινηθεί η κεφαλή διάτρησης από τη μια θέση στην άλλη. Ο στόχος είναι να ελαχιστοποιηθεί ο χρόνος διαδρομής για την κεφαλή της μηχανής.

Στην Εικόνα 13 που ακολουθεί φαίνεται το μπροστά και το πίσω μέρος μιας πλακέτας τυπωμένων κυκλωμάτων (στα αριστερά και στα δεξιά της εικόνας αντίστοιχα) ενώ στην Εικόνα 14 φαίνεται σε ένα παράδειγμα διάτρησης 2103 οπών σε μία πλακέτα τυπωμένων κυκλωμάτων η λύση που προτείνει η βιομηχανία (στα αριστερά) και πως βελτιστοποιείται η λύση του προβλήματος με αναγωγή του σε TSP (στα δεξιά).



Εικόνα 13: Το μπροστά (αριστερά) και το πίσω (δεξιά) μέρος μιας πλακέτας τυπωμένων κυκλωμάτων (PCB) [91]



Εικόνα 14: Διάτρηση 2103 οπών σε μία πλακέτα τυπωμένων κυκλωμάτων. Η λύση της βιομηχανίας φαίνεται στα αριστερά ενώ η απλοποίηση της λύσης με αναγωγή του προβλήματος σε TSP φαίνεται στα δεξιά [91]

- **Αναθεώρηση Κινητήρων Αεριοστρόβιλων Αεροσκαφών**

Αυτή η εφαρμογή παρουσιάστηκε από τους Plante, Lowe & Chandrasekaran [96] το 1987 και εμφανίζεται όταν οι κινητήρες αεριοστρόβιλων των αεροσκαφών πρέπει να αναθεωρηθούν. Για να εγγυηθούμε μια ομοιόμορφη ροή αερίου μέσω των στροβίλων υπάρχουν τα επονομαζόμενα στόμια-οδηγοί συναρμολογούμενων πτερυγίων που βρίσκονται σε διάφορα τμήματα του στροβίλου. Μια τέτοια συναρμολόγηση βασικά αποτελείται από ένα αριθμό στομίων κατευθυντήριων πτερυγίων που είναι τοποθετημένα στην περιφέρειά του. Όλα αυτά τα πτερύγια έχουν ξεχωριστά χαρακτηριστικά και η σωστή τοποθέτησή τους μπορεί να οδηγήσει σε σημαντικά οφέλη (μείωση των κραδασμών, αύξηση της ομοιομορφίας της ροής, μείωση της κατανάλωσης καυσίμων). Το πρόβλημα της τοποθέτησης των πτερυγίων με τον καλύτερο πιθανό τρόπο μπορεί να μοντελοποιηθεί ως ένα TSP με μια ειδική αντικειμενική συνάρτηση.

- **Κρυσταλλογραφία με Ακτίνες Χ**

Η ανάλυση της δομής των κρυστάλλων [97] είναι μία σημαντική εφαρμογή του TSP. Σε αυτήν την περίπτωση, χρησιμοποιείται ένα περιθλασίμετρο ακτινών Χ (X-ray diffractometer) για την ανάκτηση πληροφοριών σχετικών με τη δομή του κρυσταλλικού υλικού. Για το σκοπό αυτό, ένας ανιχνευτής μετρά την ένταση των αντανακλάσεων των ακτινών Χ του κρυστάλλου από διάφορες θέσεις. Παρόλο που η μέτρηση αυτή καθαυτή μπορεί να διενεργηθεί πολύ γρήγορα, υπάρχει μια σημαντική επιβάρυνση στο χρόνο τοποθέτησης εφόσον για μερικά πειράματα πρέπει να εξετασθούν εκατοντάδες χιλιάδες θέσεις. Σε αυτήν την περίπτωση, κάθε τοποθέτηση του περιθλασίμετρου περιλαμβάνει τη μετακίνηση τεσσάρων κινητήρων. Ο χρόνος που χρειάζεται για τη μετακίνηση από μία θέση σε μία άλλη μπορεί να υπολογιστεί με μεγάλη ακρίβεια. Το αποτέλεσμα του πειράματος δεν εξαρτάται από τη σειρά των θέσεων στις οποίες λαμβάνονται οι μετρήσεις. Παρόλα αυτά, ο συνολικός χρόνος που απαιτείται για το πείραμα εξαρτάται από αυτήν τη σειρά. Επομένως, το πρόβλημα εναπόκειται στην εύρεση της σειράς των διαδοχικών θέσεων του περιθλασίμετρου που ελαχιστοποιεί το συνολικό χρόνο του πειράματος. Αυτό οδηγεί σε ένα πρόβλημα TSP.

- **Καλωδίωση Υπολογιστή**

Οι Lenstra & Rinnooy [98] ανέφεραν μια ειδική περίπτωση σύνδεσης εξαρτημάτων σε μια πλακέτα υπολογιστή. Οι λειτουργικές μονάδες βρίσκονται σε μια πλακέτα υπολογιστή και ένα δεδομένο υποσύνολο ακίδων πρέπει να συνδεθεί. Σε αντίθεση με τη συνήθη περίπτωση όπου μια σύνδεση τύπου δέντρου Στάινερ είναι επιθυμητή, εδώ η απαίτηση είναι ότι το πολύ δύο καλώδια ενώνονται με κάθε ακίδα. Ως εκ τούτου, προκύπτει το πρόβλημα της εύρεσης της συντομότερης Χαμιλτονιανής διαδρομής με απροσδιόριστα σημεία εκκίνησης και τερματισμού.

Μια παρόμοια κατάσταση προκύπτει για την λεγόμενη καλωδίωση testbus. Για να ελεγχθεί μία κατασκευασμένη πλακέτα, πρέπει να υλοποιηθεί μια σύνδεση που εισέρχεται στην πλακέτα σε κάποιο συγκεκριμένο σημείο, διατρέχει όλες τις λειτουργικές μονάδες και τερματίζει σε κάποιο συγκεκριμένο σημείο. Για κάθε λειτουργική μονάδα έχουμε, επίσης, ένα καθορισμένο σημείο εισόδου και εξόδου για αυτή τη δοκιμαστική καλωδίωση. Αυτό το πρόβλημα λοιπόν, ισοδυναμεί επίσης με την επίλυση ενός προβλήματος Χαμιλτονιανής διαδρομής με τη διαφορά ότι οι αποστάσεις δεν είναι συμμετρικές και ότι καθορίζονται το σημείο έναρξης και τερματισμού.

- **Το Πρόβλημα της Συλλογής Παραγγελιών σε Αποθήκες**

Αυτό το πρόβλημα σχετίζεται με τη διαχείριση ειδών σε μια αποθήκη [99]. Ας υποθέσουμε ότι σε μία αποθήκη φτάνει μια παραγγελία για ένα συγκεκριμένο υποσύνολο των ειδών που είναι αποθηκευμένα σε αυτήν. Κάποιο όχημα πρέπει να συλλέξει όλα τα είδη αυτής της παραγγελίας για να τα αποστείλει στον πελάτη. Η σχέση με το TSP διαφαίνεται αμέσως: Οι θέσεις αποθήκευσης των ειδών αντιστοιχούν στους κόμβους του γράφου. Η απόσταση μεταξύ δύο κόμβων δίνεται από το χρόνο που απαιτείται για τη μετακίνηση του οχήματος από τη μία θέση στην άλλη. Το πρόβλημα της εύρεσης της συντομότερης διαδρομής για το όχημα με τον ελάχιστο χρόνο συλλογής μπορεί να λυθεί ως TSP.

4.2. Εφαρμογές του Maximum Scatter TSP

Ακολουθούν δύο εφαρμογές του Προβλήματος του Περιοδεύοντος Πωλητή Μέγιστης Διασποράς από τους τομείς της παραγωγής προϊόντος και της ιατρικής.

- **Διαδικασίες Παραγωγή με Απαίτηση Διαχωρισμού Διαδοχικών Λειτουργιών**

Το Πρόβλημα του Περιοδεύοντος Πωλητή Μέγιστης Διασποράς TSP εμφανίζεται σε ορισμένες διαδικασίες παραγωγής όπου είναι σημαντικό να υπάρχει ουσιαστικός διαχωρισμός (απόσταση) μεταξύ διαδοχικών (ή σχεδόν συνεχόμενων) λειτουργιών σε ένα αντικείμενο εργασίας. Για παράδειγμα, αυτό το πρόβλημα έχει εμφανιστεί στην Boeing, όπου υπάρχει αλληλουχία των εργασιών καρφώματος (πριτσίνωσης) κατά τη στερέωση των μεταλλικών φύλλων μεταξύ τους [100]. Προκειμένου να αποφευχθεί η μη ομοιόμορφη παραμόρφωση των φύλλων λαμαρίνας που είναι ενωμένα, είναι σημαντικό να ακολουθηθεί η διαδικασία πριτσινώματος έτσι ώστε η απόσταση μεταξύ ενός πριτσινιού και του επόμενου να είναι μεγάλη επομένως οι εργασίες πριτσίνωσης πρέπει να διασκορπιστούν χρονικά. Σε λειτουργίες που περιλαμβάνουν τη θέρμανση του τεμαχίου, μπορεί να είναι σημαντικό όχι μόνο να διαχωρίζεται κάθε σημείο από το αμέσως προηγούμενο και το αμέσως επόμενό του, αλλά και από τους n-γείτονές του, ώστε να υπάρχει στο ενδιάμεσο ο απαιτούμενος χρόνος ψύξης κάθε λειτουργίας [101].

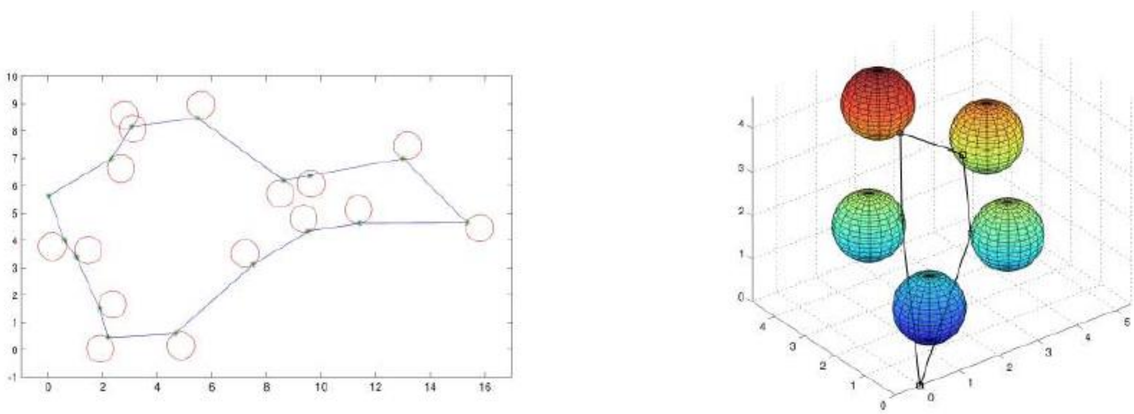
- **Δυναμικός Χωρικός Ανασυγκροτητής**

Το TSP Μέγιστη Διασποράς προκύπτει επίσης σε ορισμένες εφαρμογές ιατρικής απεικόνισης, όπως έχει μελετηθεί από την εταιρεία Penavic. Όταν απεικονίζονται φυσιολογικές λειτουργίες χρησιμοποιώντας έναν Δυναμικό Χωρικό Ανασυγκροτητή (Dynamic Spatial Reconstructor - DSR), οι πηγές ακτινοβολίας τοποθετούνται κατά μήκος του άνω μισού ενός κυκλικού δακτυλίου, με τους αισθητήρες τοποθετημένους ακριβώς απέναντι, στο κάτω μισό του δακτυλίου. Η «ακολουθία ενεργοποίησης» καθορίζει τη σειρά με την οποία ενεργοποιούνται οι πηγές και οι αντίστοιχοί τους αισθητήρες, συνήθως σε περιοδικό ρυθμό. Οι αισθητήρες συλλέγουν στοιχεία έντασης της ενέργειας που διέρχεται από τον ασθενή που τοποθετείται στο κέντρο του δακτυλίου. Όταν ενεργοποιείται η πηγή i, κάποια ενέργεια διαχέεται, οπότε είναι σημαντικό να μην ενεργοποιηθούν οι αισθητήρες κοντινών πηγών (π.χ. i + 1, i - 1, i + 2, ...) αμέσως μετά την ενεργοποίηση της πηγής i. Αυτό το χαρακτηριστικό των DSR ώθησε την εταιρεία να μελετήσει την ακολουθία ενεργοποίησης των πηγών και των αντίστοιχών τους αισθητήρων για ορισμένες συγκεκριμένες γεωμετρίες μηχανημάτων DSR και οδήγησε και στη μελέτη των Arkin, Chiang, Mitchell, Skiena & Yang σχετικά με ορισμένες περιπτώσεις του προβλήματος της μέγιστης διασποράς [101].

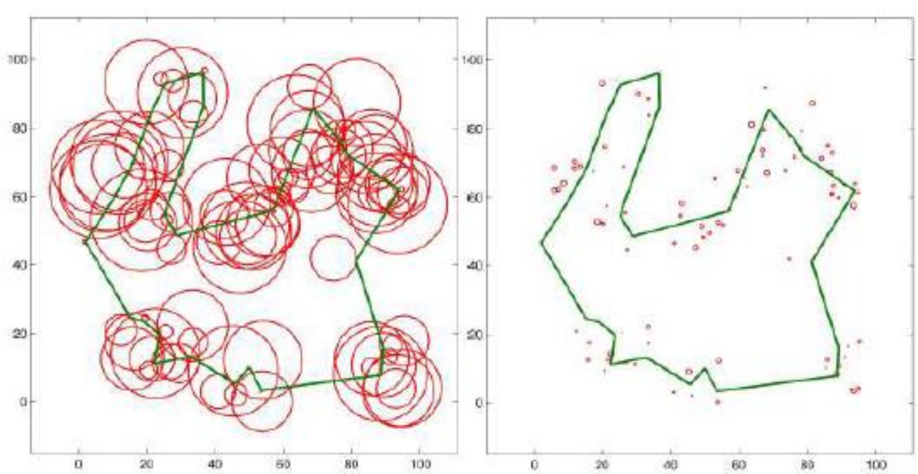
4.3. Εφαρμογές του CETSP

Γενική εφαρμογή αυτού του προβλήματος είναι η περίπτωση του πωλητή που θέλει να επισκεφτεί υποψήφιους πελάτες οι οποίοι είναι διατεθειμένοι να μετακινηθούν μέχρι μια απόσταση r από τη βάση τους για να τον συναντήσουν. Πιο συγκεκριμένες εφαρμογές αυτού του προβλήματος είναι η αναγνώριση του σχεδιασμού της διαδρομής ενός αεροσκάφους, η παρακολούθηση των δρομολογίων των πλοίων, η ανίχνευση πυρκαγιάς με εναέρια μέσα και η παρακολούθηση ρομπότ με αισθητήρες ασύρματου δικτύου [24].

Στην επόμενη εικόνα (Εικόνα 15) φαίνεται γραφικά μια επίλυση ενός προβλήματος CETSP σταθερών ακτινών r_i σε δύο διαστάσεις (στα αριστερά) και σε τρεις διαστάσεις (στα δεξιά) ενώ παρακάτω (Εικόνα 16) φαίνεται η επίλυση ενός CETSP 101 κόμβων (με πράσινη συνεχόμενη γραμμή) και ακτίνες r_i διαφορετικού μήκους.



Εικόνα 15: Παράδειγμα επίλυσης CETSP σε 2 διαστάσεις (αριστερά) και σε 3 διαστάσεις (δεξιά) [29]



Εικόνα 16: Επίλυση ενός CETSP με 101 κόμβους [29]

Παρακάτω αναφέρονται δύο εφαρμογές του CETSP:

- **Αναγνώριση Ετικετών Ραδιοσυχνοτήτων**

Η επίλυση του CETSP έχει χρησιμότητα και σε άλλα προβλήματα του πραγματικού κόσμου. Για παράδειγμα, χρησιμοποιώντας τις ετικέτες αναγνώρισης ραδιοσυχνοτήτων (Radio Frequency Identification – RFID) που είναι συνδεδεμένες με φυσικούς μετρητές, μπορεί κανείς να κωδικοποιήσει τον αριθμό αναγνώρισης του μετρητή και τις τρέχουσες μετρήσεις του σε ψηφιακά σήματα. Με αυτόν τον τρόπο, ένα φορτηγό που είναι εφοδιασμένο με σύστημα αυτόματης ανάγνωσης μετρητών (Automatic Meter Reading – AMR) μπορεί να συλλέξει και να μεταφέρει δεδομένα από κάποια απόσταση. Ως εκ τούτου, στο πλαίσιο του συστήματος AMR, ο αναγνώστης μετρητών δεν απαιτείται να επισκέπτεται προσωπικά τον κάθε πελάτη, αλλά μόνο να φτάνει σε μία συγκεκριμένη ακτίνα από τον κάθε ένα [102], [103].

- **Μη Επανδρωμένα Αεροσκάφη**

Άλλη εφαρμογή είναι τα μη επανδρωμένα αεροσκάφη (Unmanned Aerial Vehicles – UAV) που έχουν χρησιμοποιηθεί ευρέως για στρατιωτικές και πολιτικές αποστολές, ειδικά όταν η παρουσία ενός ανθρώπινου πληρώματος στο αεροσκάφος γίνεται πολύ επικίνδυνη. Ως παραδείγματα χρήσης των UAV μπορούμε να αναφέρουμε την εναέρια αναγνώριση εδάφους, την ανίχνευση πυρκαγιών σε δάση με εναέρια μέσα, την παρακολούθηση πλοίων, την παράδοση υλικών (τροφή, πυρομαχικά κ.α.) σε συγκεκριμένους στόχους, την παρακολούθηση μιας γεωγραφικής περιοχής και την επιτήρηση διαφόρων αγωγών. Αν ένα μη επανδρωμένο αεροσκάφος είναι εφοδιασμένο με αισθητήρες τότε μπορεί να λειτουργήσει επιτυχώς από μια ορισμένη απόσταση χωρίς να μεταβεί ακριβώς πάνω από τον επιθυμητό στόχο. Αντίστοιχες περιπτώσεις έχουμε όταν το UAV χρειάζεται απλά να ρίξει το φορτίο του όσο το δυνατόν πλησιέστερα σε ένα συγκεκριμένο στόχου, όπως για παράδειγμα σε περιπτώσεις ειδικών στρατιωτικών επιχειρήσεων [29].

4.4. Εφαρμογές του mTSP

Οι κύριες εφαρμογές του Προβλήματος Πολλαπλών Περιοδεύοντων Πωλητών προκύπτουν αρκετά συχνά στην πραγματικότητα, καθώς αφορούν στη διαχείριση πολλαπλών πωλητών. Τέτοιες καταστάσεις προκύπτουν κυρίως σε προβλήματα δρομολόγησης και προγραμματισμού. Στη συνέχεια, παρουσιάζονται ορισμένες εφαρμογές που έχουν αναφερθεί στη διεθνή βιβλιογραφία.

- **Πρόβλημα Προγραμματισμού Τυπογραφείων**

Μια από τις μείζονες και πρωταρχικές εφαρμογές του mTSP προκύπτει στον προγραμματισμό ενός τυπογραφείου με πολλές και διαφορετικές εκδόσεις. Υπάρχουν 5 ζεύγη από κυλίνδρους μεταξύ των οποίων κυλά το χαρτί και τυπώνονται ταυτόχρονα και οι δύο πλευρές του χαρτιού. Υπάρχουν 3 διαφορετικά είδη εντύπων, το 4σέλιδο, το 6σέλιδο και το 8σέλιδο, τα οποία χρησιμοποιούνται σαν πρότυπα για να τυπωθούν οι εκδόσεις. Το πρόβλημα προγραμματισμού συνίσταται στην επιλογή ποιου εντύπου θα χρησιμοποιηθεί για κάθε εκτύπωση και πόσο αυτή θα διαρκεί. Το κόστος αλλαγής της πλάκας εκτύπωσης αντιστοιχεί στο κόστος μεταξύ των πόλεων στο πρόβλημα mTSP.

- **Πρόβλημα Δρομολόγησης Σχολικών Λεωφορείων**

Το 1972 οι Angel, Caudle & Noonan [104] διερεύνησαν το πρόβλημα αυτό ως μια παραλλαγή του mTSP με κάποιους πλευρικούς περιορισμούς. Ο σκοπός του προγραμματισμού αυτού είναι να πετύχει ένα μοτίβο επιβίβασης και αποβίβασης έτσι ώστε ο αριθμός των διαδρομών να ελαχιστοποιείται, η συνολική απόσταση διαδρομών όλων των λεωφορείων να διατηρείται στο ελάχιστο δυνατό, κανένα λεωφορείο να μην υπερφορτώνεται, καθώς και ο χρόνος που απαιτείται για να πραγματοποιηθεί οποιαδήποτε διαδρομή να μην ξεπερνά τα μέγιστα επιτρεπόμενα χρονικά όρια.

- **Πρόβλημα Προγραμματισμού Προσωπικού**

Μια εφαρμογή σχετική με καταθέσεις που μεταφέρονται μεταξύ διαφορετικών υποκαταστημάτων τραπεζών αναφέρθηκε το 1973 από τους Svestka & Huckefeldt [105].

Σε αυτό το πρόβλημα, οι καταθέσεις χρειάζεται να περισυλλεχθούν από τα υποκαταστήματα και να επιστραφούν στο κεντρικό κατάστημα από μια ομάδα εργαζομένων. Το πρόβλημα έγκειται στον προσδιορισμό των διαδρομών με το ελάχιστο συνολικό κόστος. Δύο παρόμοιες εφαρμογές παρουσιάζονται από τους Lenstra & Rinnooy [106] και Zhang, Gruver & Smith [107].

- **Πρόβλημα Προγραμματισμού Συνεντεύξεων**

Οι Gilbert & Hofstra το 1992 [108] βρήκαν αυτή την εφαρμογή του mTSP που αφορά στον προγραμματισμό συνεντεύξεων μεταξύ περιοδεύοντων μεσιτών και προμηθευτών της τουριστικής βιομηχανίας διαφορετικής διάρκειας. Κάθε μεσίτης αντιστοιχεί σε ένα πωλητή του mTSP ο οποίος πρέπει να επισκεφθεί ένα συγκεκριμένο σύνολο προμηθευτών, οι οποίοι αναπαρίστανται ως ένα σύνολο πόλεων.

- **Πρόβλημα Προγραμματισμού σε Μονάδα Θερμής Ελάσεως**

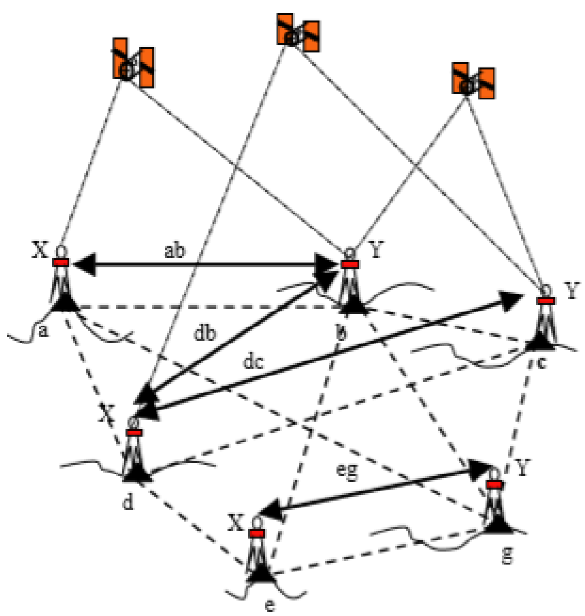
Στη βιομηχανία Σιδήρου και Χάλυβα, οι παραγγελίες στις μονάδες θερμής ελάσεως προγραμματίζονται με τέτοιο τρόπο ώστε το συνολικό κόστος παραγωγής να μπορεί να ελαχιστοποιηθεί. Σε μια πρόσφατη μελέτη [109], οι παραγγελίες αντιμετωπίζονται ως πόλεις και το κόστος μετάβασης από μια παραγγελία σε μια άλλη ως η απόσταση μεταξύ δύο πόλεων. Η λύση του μοντέλου οδηγεί σε ένα πλήρες πρόγραμμα για τη μονάδα θερμής ελάσεως.

- **Πρόβλημα Σχεδιασμού Αποστολών**

Αυτό το πρόβλημα συνίσταται στον καθορισμό μίας βέλτιστης διαδρομής για κάθε συμμετέχοντα ώστε να επιτύχει τους στόχους της αποστολής του στον ελάχιστο δυνατό χρόνο. Ο σχεδιαστής της κάθε αποστολής χρησιμοποιεί μια παραλλαγή του mTSP στην οποία υπάρχουν n συμμετέχοντας, m στόχοι που πρέπει να επιτευχθούν από κάποιους από αυτούς και μια πόλη ως βάση στην οποία όλοι οι συμμετέχοντες τελικά θα επιστρέψουν. Η εφαρμογή του mTSP στο σχεδιασμό αποστολών έχει αναφερθεί από τους Brummit & Stentz [110], Brummit & Stentz [111] και τους Yu, Jinhai, Guochang, Rubo & Haiyan [112] στις εργασίες τους. Αντίστοιχα, παρόμοια προβλήματα δρομολόγησης που προκύπτουν κατά το σχεδιασμό αποστολών μη-επανδρωμένων εναέριων οχημάτων μπορούν, επίσης, να μοντελοποιηθούν ως mTSP όπως έχει διερευνηθεί από τους Ryan, Bailey, Moore & Carlton [113].

- **Σχεδιασμός του Παγκόσμιου Δορυφορικού Συστήματος Πλοήγησης Τοπογραφικών Δικτύων**

Μια πολύ πρόσφατη και ενδιαφέρουσα εφαρμογή του mTSP, προκύπτει στο σχεδιασμό του Παγκόσμιου Δορυφορικού Συστήματος Πλοήγησης (Global Navigation Satellite System – GNSS) [114]. Το GNSS είναι ένα διαστημικό δορυφορικό σύστημα το οποίο παρέχει κάλυψη για όλες τις τοποθεσίες παγκοσμίως και είναι αρκετά σημαντικό σε πρακτικές εφαρμογές, όπως η έγκαιρη προειδοποίηση και διαχείριση φυσικών καταστροφών, η παρακολούθηση του περιβάλλοντος και της γεωργίας κλπ. Ο στόχος του είναι να καθορίζει τη γεωγραφική θέση άγνωστων σημείων επί της γης ή πάνω από αυτήν με χρήση δορυφορικού εξοπλισμού. Αυτά τα σημεία, επί των οποίων τοποθετούνται δέκτες, συντονίζονται από μια σειρά από συνεδρίες παρατήρησης. Όταν υπάρχουν πολλαπλοί δέκτες, ή πολλαπλές περίοδοι λειτουργίας, το πρόβλημα της εύρεσης της καλύτερης σειράς συνεδριών για τους δέκτες μπορεί να διαμορφωθεί ως ένα mTSP.



Εικόνα 17: Παρατήρηση συνεδριών με χρήση δεκτών GPS

- **Συνδέσεις με άλλα προβλήματα**

Τα προαναφερθέντα προβλήματα μπορούν να μοντελοποιηθούν ως ένα mTSP. Εκτός αυτών το mTSP μπορεί επίσης να συνδεθεί με άλλα προβλήματα. Ένα τέτοιο παράδειγμα είναι η εξισορρόπηση του φόρτου εργασίας μεταξύ των πωλητών που περιγράφεται από τον Okonjo-Adigwe [115]. Σε αυτό, παρουσιάζεται ένα μοντέλο βασιζόμενο σε mTSP για να λύσει ένα πρόβλημα προγραμματισμού φόρτου εργασίας με κάποιους πρόσθετους περιορισμούς όπως άνω και κάτω φράγματα στους χρόνους δρομολογίου και υπολογισμός του συνολικού βάρους (φόρτου) κάθε πωλητή. Ένα ακόμα παράδειγμα αφορά τις υπηρεσίες ασφαλείας κατά τη διάρκεια της νύχτας ([116], [117]). Αυτό το πρόβλημα συνίσταται στην εκχώρηση καθηκόντων σε έναν αριθμό φυλάκων, οι οποίοι θα εκτελέσουν επιθεωρήσεις σε ένα δοθέν σύνολο τοποθεσιών με περιορισμούς όπως η ικανότητα κάθε φύλακα, καθώς και οι χρονικές περίοδοι κάθε επιθεώρησης.

Το mTSP προκύπτει επίσης ως ένα υποπρόβλημα ενός γερανού αποβάθρας στο λειτουργικό σχεδιασμό πλοίου. Το mTSP με χρονικές περιόδους μπορεί να χρησιμοποιηθεί στην μοντελοποίηση προβλημάτων μεταφοράς εμπορευμάτων. Επίσης μια ενδιαφέρουσα εφαρμογή, βασίζεται στο πρόβλημα συντονισμού κίνησης πολλαπλών αντικειμένων. Ένα τέτοιο πρόβλημα προκύπτει στην συναρμολόγηση ηλεκτρονικών κυκλωμάτων, στη διαχείριση μνήμης των κατανεμημένων συστημάτων, καθώς και στο συντονισμό κινητών ρομπότ σε ένα δομημένο χώρο, όπως μια αποθήκη. Το πρόβλημα ορίζεται από ένα ορθογώνιο πλέγμα το οποίο διαμοιράζεται σε ένα αριθμό τετραγώνων. Τα τετράγωνα μπορεί είτε να περιέχουν ένα αντικείμενο είτε να είναι κενά. Στη συνέχεια η βέλτιστη κίνηση των αντικειμένων σε ένα πλέγμα με κενά διαστήματα μπορεί να βρεθεί χρησιμοποιώντας ένα μοντέλο mTSP.

4.5. Εφαρμογές του Bottleneck TSP

Ακουλουθεί στη συνέχεια μία εφαρμογή του προβλήματος προγραμματισμού ενός μηχανήματος όπου αναζητώντας τη διεργασία η οποία είναι υπεύθυνη για τη μη βέλτιστη λειτουργία της μηχανής που εξετάζουμε οδηγούμαστε στην επίλυση ενός Προβλήματος Περιοδεύοντος Πωλητή με Σημεία Συμφόρησης.

- **Προγραμματισμός με Χρόνους Επεξεργασίας που Εξαρτώνται από τη Σειρά με την οποία Εκτελούνται**

Έστω ότι δίνονται n στον αριθμό εργασίες που πρέπει να εκτελεστούν σε κάποια μηχανή. Ο χρόνος που απαιτείται για την επεξεργασία της εργασίας j είναι t_{ij}, αν i είναι η εργασία που εκτελείται αμέσως πριν την j ενώ αν η j είναι η πρώτη εργασία που εκτελείται τότε ο χρόνος επεξεργασίας της είναι t_{0j}. Η δυσκολία του προβλήματος είναι να βρεθεί μια ακολουθία εκτέλεσης των εργασιών τέτοια ώστε ο συνολικός χρόνος επεξεργασίας και ολοκλήρωσης όλων των εργασιών n να είναι όσο γίνεται μικρότερος. Είναι φανερό ότι αυτό το πρόβλημα μπορεί να μοντελοποιηθεί ως ένα πρόβλημα ελαχίστης (κατευθυνόμενης) Χαμιλτονιανής διαδρομής.

Αν υποτεθεί ότι η μηχανή που εξετάζεται είναι στην ουσία μια γραμμή συναρμολόγησης τότε οι εργασίες n αντιστοιχούν σε λειτουργίες οι οποίες πρέπει να εκτελεστούν για κάποιο προϊόν στους σταθμούς εργασίας της γραμμής παραγωγής. Σε μια τέτοια περίπτωση το κύριο ενδιαφέρον βρίσκεται στην εξισορρόπηση των αναγκών της γραμμής συναρμολόγησης. Επομένως, αντί του ελάχιστου χρόνου εκτέλεσης όλων των εργασιών που αφορούν ένα προϊόν, είναι σημαντικό να είναι γνωστός και ο μεμονωμένα μεγαλύτερος χρόνος επεξεργασίας που απαιτείται σε ένα μόνο σταθμό εργασίας. Για τις ανάγκες αυτής της απαίτησης κατάλληλο είναι το Πρόβλημα του Περιοδεύοντος Πωλητή με Σημεία Συμφόρησης που έχει ήδη αναφερθεί.

4.6. Εφαρμογές του m-PDTSP

Ακολουθούν δύο από τις κυριότερες εφαρμογές του Προβλήματος του Περιοδεύοντος Πωλητή με Παραλαβή και Παράδοση Πολλαπλών Εμπορευμάτων.

- **Πρόβλημα Επανατοποθέτησης του Προϊόντος**

Η κυριότερη εφαρμογή του m-PDTSP περιλαμβάνει, επιπρόσθετα του TSP, και τις περιπτώσεις επανατοποθέτησης του αποθέματος. Υποθέτουμε ότι έχουμε ένα δίκτυο λιανοπωλητών γεωγραφικά διασκορπισμένο σε μία περιοχή. Συχνά, λόγω της φύσης των αιτημάτων, ορισμένοι έμποροι λιανικής πώλησης έχουν πλεόνασμα της απογραφής ορισμένων προϊόντων ενώ άλλοι έχουν τόσο ισχυρές πωλήσεις ώστε να χρειάζονται επιπλέον αποθέματα. Σε πολλές περιπτώσεις η επιχείρηση μπορεί να αποφασίσει να μεταφέρει μονάδες προϊόντος από τους λιανοπωλητές που είχαν λιγότερες από τις αναμενόμενες μέσες πωλήσεις σε εκείνους που είχαν ζήτηση για το προϊόν μεγαλύτερη του αναμενόμενου. Ο προσδιορισμός του φθηνότερου τρόπου εκτέλεσης της μεταφοράς των προϊόντων (με την προϋπόθεση φυσικά ότι πρέπει ο κάθε πωλητής λιανικής να επισκεφθεί ακριβώς μία φορά) είναι ο στόχος του m-PDTSP [38].

- **Σύστημα Ενοικίασης Ποδηλάτων**

Μια άλλη εφαρμογή του m-PDTSP εμφανίζεται στο πλαίσιο ανάπτυξης ενός συστήματος ενοικίασης ποδηλάτων με αυτοεξυπηρέτηση, όπου κάθε βράδυ ένα όχημα συγκεκριμένης χωρητικότητας πρέπει να επισκεφθεί τους σταθμούς ποδηλάτων μιας πόλης για να συλλέξει ή να παραδώσει ποδήλατα ώστε να αποκατασταθεί η αρχική κατανομή ποδηλάτων στο σύστημα. Οι Chemla, Meunier & Wolfler Calvo [118] όπως και οι Raviv, Tzur & Forma [119], μεταξύ άλλων, στις εργασίες τους προσέγγισαν την περίπτωση όπου όλα τα ποδήλατα είναι πανομοιότυπα με ένα πρόβλημα που σχετίζεται με το 1-PDTSP. Όταν υπάρχουν διαφορετικοί τύποι ποδηλάτων (για παράδειγμα, με και χωρίς παιδικό κάθισμα) το πρόβλημα σχετίζεται με το m-PDTSP [38].

4.7. Προβλήματα με κοινά χαρακτηριστικά με το TSP

Κάποιες φορές το TSP εμφανίζεται ως ένα υποπρόβλημα σε πιο σύνθετες συνδυαστικές διαδικασίες βελτιστοποίησης που έχουν σχεδιαστεί ώστε να αντιμετωπίσουν διάφορα προβλήματα των βιομηχανιών παραγωγής προϊόντων.

Πιο κάτω δίνονται τρία παραδείγματα που δεν μπορούν να μετασχηματιστούν σε TSP, αλλά μοιράζονται κάποια κοινά χαρακτηριστικά με αυτό ή το TSP εμφανίζεται σε αυτά ως υποπρόβλημα.

- **Μάσκα Αποτύπωσης κατά την Παραγωγή Πλακετών Τυπωμένων Κυκλωμάτων**

Για την παραγωγή κάθε στρώματος μίας πλακέτας τυπωμένων κυκλωμάτων, καθώς και για τα στρώματα συσκευών ενσωματωμένων ημιαγωγών, πρέπει να παράγεται μια φωτογραφική μάσκα. Στην περίπτωση των πλακετών τυπωμένων κυκλωμάτων, αυτό γίνεται από μια συσκευή αποτύπωσης. Η συσκευή αυτή μετακινεί ένα φακό πάνω από μια φωτοευαίσθητη γυάλινη πλάκα. Το κλείστρο μπορεί να ανοίξει ή να κλείσει για να εκθέσει συγκεκριμένα μέρη της πλάκας. Υπάρχουν διαφορετικά διαφράγματα που είναι διαθέσιμα για να μπορούν να δημιουργήσουν διαφορετικές δομές πάνω στην πλακέτα. Πρέπει να ληφθούν υπόψη δύο είδη δομών. Μια γραμμή εκτίθεται στην πλάκα μετακινώντας το κλειστό κλείστρο σε ένα τελικό σημείο της γραμμής και έπειτα ανοίγοντας το κλείστρο και μετακινώντας το στο άλλο άκρο της γραμμής. Έπειτα το κλείστρο κλείνει. Η δομή τύπου σημείου δημιουργείται μετακινώντας το κλείστρο (με το κατάλληλο διάφραγμα) στη θέση του σημείου αυτού και ανοίγοντας το κλείστρο μόνο για ένα σύντομο χρονικό διάστημα και στη συνέχεια κλείνοντας το ξανά. Η ακριβής μοντελοποίηση του προβλήματος της συσκευής αποτύπωσης οδηγεί σε ένα πρόβλημα πιο περίπλοκο από το TSP και επίσης πιο περίπλοκο από το Πρόβλημα του Αγροτικού Ταχυδρόμου. Μια εφαρμογή σε πραγματικό περιβάλλον παραγωγής αναφέρεται στην εργασία των Grötschel, Jünger & Reinelt [120].

- **Το Πρόβλημα της Δρομολόγησης Οχημάτων**

Παρότι αυτό το πρόβλημα δε μπορεί να μετασχηματιστεί άμεσα σε ένα TSP, εντούτοις το TSP εμφανίζεται ως υποπρόβλημα κατά την επίλυση του Προβλήματος Δρομολόγησης Οχημάτων και καθώς αποτελεί ένα από τα πλέον σημαντικά προβλήματα βελτιστοποίησης που αντιμετωπίζουν διάφορες εταιρείες που διαχειρίζονται οχήματα γίνεται εκτενής αναφορά σε αυτό σε επόμενο υποκεφάλαιο αυτής της εργασίας.

- **Έλεγχος Κινήσεων Ρομπότ**

Για να κατασκευαστεί ένα κομμάτι ενός προϊόντος, συνήθως, ένα ρομπότ πρέπει να εκτελέσει μια σειρά από διαφορετικές εργασίες σε αυτό (άνοιγμα οπών διαφόρων διαμέτρων, κοπή σχισμών, λείανση επιφανειών κλπ). Στόχος του προβλήματος, είναι να καθοριστεί μια ακολουθία των απαραίτητων εργασιών που να οδηγεί στο συντομότερο συνολικό χρόνο επεξεργασίας του κομματιού του προϊόντος. Η δυσκολία που προκύπτει από αυτή την εφαρμογή έγκειται στο ότι υπάρχουν περιορισμοί προτεραιότητας οι οποίοι που πρέπει να τηρηθούν. Έτσι εδώ έχουμε το πρόβλημα εύρεσης της συντομότερης Χαμιλτονιανής διαδρομής, όπου οι αποστάσεις αντιστοιχούν στους χρόνους που απαιτούνται για την τοποθέτηση και τις πιθανές αλλαγές των διαφόρων εργαλείων, που ικανοποιεί ορισμένες σχέσεις προτεραιότητας μεταξύ των δραστηριοτήτων.

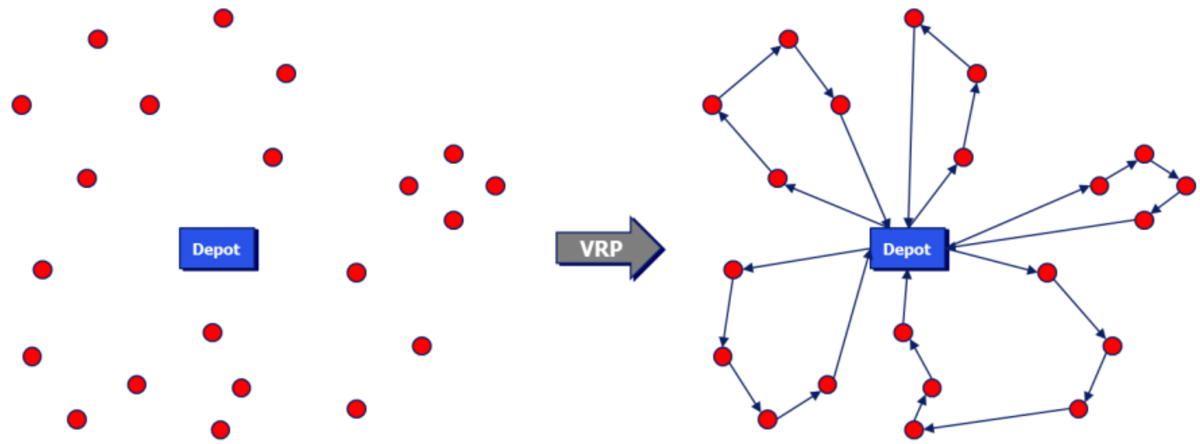
4.8. Εφαρμογές του VRP

Στο Πρόβλημα Δρομολόγησης Οχημάτων (Vehicle Routing Problem – VRP), πέρα από τους N κόμβους και τις αποστάσεις μεταξύ αυτών υπάρχει και μία αποθήκη, ενώ δίνεται και η απόσταση μεταξύ της αποθήκης και καθενός εκ των κόμβων. Επιπλέον, ο κάθε κόμβος έχει μια δοσμένη ζήτηση σε κάποιο προϊόν που πρέπει να ικανοποιηθεί από την αποθήκη. Για τον σκοπό αυτό διατίθονται φορτηγά (χωρίς να καθορίζεται πάντα ο συνολικός αριθμός τους) με ταυτόσημες χωρητικότητες. Το ζητούμενο εδώ είναι να βρεθούν οι διαδρομές εκείνες των φορτηγών, ελάχιστου ολικού μήκους, που ικανοποιούν τις απαιτήσεις όλων των κόμβων χωρίς να παραβιάζεται ο περιορισμός της χωρητικότητας των φορτηγών. Σε κάθε ανεξάρτητη διαδρομή το φορτηγό επισκέπτεται ένα υποσύνολο των κόμβων ενώ αρχίζει και τερματίζει τη διαδρομή του στην αποθήκη [9].

Αυτή η κατηγορία προβλημάτων παρουσιάστηκε για πρώτη φορά από τους Dantzing & Ramser το 1959, ενώ σήμερα αποτελεί ένα από τα πιο σημαντικά και εφαρμόσιμα προβλήματα διανομής της εφοδιαστικής αλυσίδας. Οι Dantzing & Ramser δημιούργησαν την πρώτη αλγοριθμική προσέγγιση επίλυσης τέτοιων προβλημάτων και την εφάρμοσαν για την διανομή γκαζολίνης σε έναν αριθμό σταθμών τροφοδοσίας. Στη συνέχεια, το 1964, οι Clarke & Wright πρότειναν έναν ευρετικό αλγόριθμο εξοικονόμησης, ο οποίος παρουσιάζει σημαντικές βελτιώσεις σε σχέση με αυτόν των Dantzing & Ramser. Το Πρόβλημα της Δρομολόγησης Οχημάτων αφορά στην ουσία ένα πρόβλημα διανομής, το οποίο όταν λυθεί θα πρέπει να επιφέρει όφελος για μια επιχείρηση μέσα από την εξοικονόμηση κόστους. Με σκοπό την επίλυση τέτοιου είδους προβλημάτων, θα πρέπει αρχικά να υπολογιστούν οι βέλτιστες διαδρομές, τις οποίες θα εκτελέσει ένας συγκεκριμένος αριθμός οχημάτων, με γνώμονα πάντα να καλύπτεται όλη η ζήτηση των πελατών της εταιρείας. Στην πραγματικότητα υπάρχουν αρκετές περιπτώσεις, που θέτουν διαφορετικούς περιορισμούς ως προς τη διανομή των προϊόντων για την επίλυση του εν λόγω προβλήματος [121].

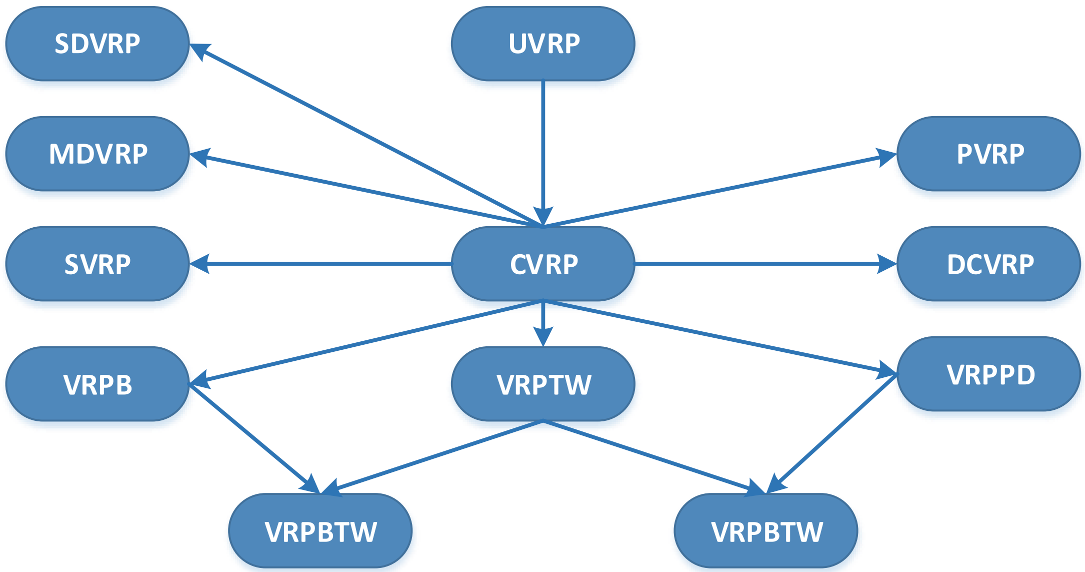
Αξίζει να αναφερθεί ότι το VRP στη γενική του μορφή δεν έχει χρονικούς περιορισμούς ενώ δεν υπάρχουν και περιορισμοί ως προς την σειρά με την οποία θα εξυπηρετηθούν οι πελάτες. Στόχος του προβλήματος είναι το σύνολο όλων των διαδρομών που θα πραγματοποιήσουν όλα τα οχήματα να δίνει το ελάχιστο δυνατό κόστος. Αυτό θα επιτευχθεί μέσα από τον υπολογισμό του συνόλου των διαδρομών εκείνων που ελαχιστοποιούν τη διανυθείσα απόσταση ή το χρόνο παράδοσης των προϊόντων, αναλόγως τον στόχο της εταιρείας διανομής. Επιπλέον, όταν ο συνολικός αριθμός των διαθέσιμων οχημάτων δεν είναι εκ των προτέρων γνωστός, μείωση του κόστους του προβλήματος επέρχεται και μέσω της ελαχιστοποίησης αυτού του αριθμού και κατ’ επέκταση του λειτουργικού κόστους των οχημάτων (οδηγοί, κόστη συντήρησης και επισκευές). Ακόμα, η εξισορρόπηση των διαδρομών των οχημάτων που θα προκύψουν με βάση τις ώρες που απαιτούνται για να διανυθούν όλες οι διαδρομές, καθώς και η ελαχιστοποίηση του φορτίου της σε κάθε διαδρομή, όπως και η ελαχιστοποίηση των ενδεχόμενων κυρώσεων που μπορεί να προκύψουν από την μερική εξυπηρέτηση των πελατών είναι στους στόχους της επίλυσης του VRP [122].

Στην εικόνα που ακολουθεί (Εικόνα 18) φαίνεται ένα παράδειγμα επίλυσης ενός απλού Προβλήματος Δρομολόγησης Οχημάτων μίας αποθήκης και 21 πελατών που πρέπει να εξυπηρετηθούν.



Εικόνα 18: Παράδειγμα επίλυσης ενός VRP [123]

Στη συνέχεια, στην Εικόνα 19 φαίνονται διάφορες παραλλαγές του Προβλήματος Δρομολόγησης Οχημάτων και πώς αυτές συνδέονται με το βασικό πρόβλημα ενώ στον πίνακα που ακολουθεί (Πίνακας 5) εξηγούνται οι συντομογραφίες των διαφόρων παραλλαγών και παρουσιάζονται για κάθε μία από αυτές οι βασικοί της περιορισμοί.



Εικόνα 19: Ταξινόμηση προβλημάτων VRP [124]

Πίνακας 5: Παρουσίαση παραλλαγών VRP και περιορισμών κάθε μίας εξ αυτών

Συντομογραφία	Αγγλικός Όρος	Ελληνικός Όρος	Περιορισμός
CVRP	Capacitated Vehicle Routing Problem	Πρόβλημα Δρομολόγησης Οχημάτων με Συγκεκριμένο Όριο Χωρητικότητας	Η χωρητικότητα του κάθε οχήματος είναι σταθερή και γνωστή από την αρχή του προβλήματος.
VRPTW	Vehicle Routing Problem with Time Windows	Πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα	Η εξυπηρέτηση του κάθε πελάτη θα πρέπει να λάβει χώρα εντός συγκεκριμένου χρονικού διαστήματος.

Συντομογραφία	Αγγλικός Όρος	Ελληνικός Όρος	Περιορισμός
VRPPD	Vehicle Routing Problem with Pickup and Delivery	Πρόβλημα Δρομολόγησης Οχημάτων με Διανομή και Παραλαβή κατά τη Διάρκεια της Διαδρομής	Δεν ανταλλάσσονται προϊόντα μεταξύ των πελατών/κόμβων της διαδρομής. Επομένως, όλες οι παραγγελίες προς παράδοση έχουν ως αφετηρία την αποθήκη, ενώ όλες οι επιστροφές των προϊόντων που παραλαμβάνονται κατά τη διαδρομή τερματίζουν στην αποθήκη.
VRPB	Vehicle Routing Problem with Backhauls	Πρόβλημα Δρομολόγησης Οχημάτων με Backhauls	Κάθε όχημα, σε κάθε διαδρομή, πραγματοποιεί πρώτα το σύνολο των παραδόσεων που έχει να κάνει, ύστερα επισκέπτεται τους πελάτες από τους οποίους έχει να παραλάβει επιστρεφόμενα προϊόντα και τέλος επιστρέφει στην αποθήκη ώστε να αποφύγει τη μη συμφέρουσα ανακατάταξη των εμπορευμάτων σε κάθε επισκεπτόμενο κόμβο-πελάτη.
MDVRP	Vehicle Routing Problem with Multiple Depot	Πρόβλημα Δρομολόγησης Οχημάτων Πολλαπλών Αποθηκών	Για τη λύση του προβλήματος υπάρχουν οι εξής δύο τρόποι: 1. Κάθε αποθήκη διαθέτει το δικό της αριθμό οχημάτων, καθώς και πελατών που εξυπηρετεί, οπότε στην ουσία δημιουργούνται πολλαπλά, ανεξάρτητα VRP προβλήματα. 2. Κάθε όχημα έχει ως αφετηρία μία αποθήκη, τερματισμό μια άλλη, ενώ δύναται να διέρχεται από μία τρίτη αποθήκη για ανεφοδιασμό.
PVRP	Periodic Vehicle Routing Problem	Πρόβλημα Δρομολόγησης Οχημάτων Περιόδου	Το διάστημα του προγραμματισμού των διαδρομών επεκτείνεται από τη 1 σε M ημέρες.
SDVRP	Split Delivery Vehicle Routing Problem	Πρόβλημα Δρομολόγησης Οχημάτων με Διασπαρμένες Παραδόσεις	Κάθε πελάτης δύναται να εξυπηρετηθεί από περισσότερα του ενός οχήματα. Η λύση του εν λόγω προβλήματος είναι κατάλληλη για τις περιπτώσεις εκείνες όπου η ζήτηση των

Συντομογραφία	Αγγλικός Όρος	Ελληνικός Όρος	Περιορισμός
			προϊόντων από ένα πελάτη ξεπερνάει τη χωρητικότητα του οχήματος.
SVRP	Stochastic Vehicle Routing Problem	Στοχαστικό Πρόβλημα Δρομολόγησης Οχημάτων	Ένα ή και περισσότερα στοιχεία του προβλήματος, όπως ο αριθμός των πελατών καθώς και η ζήτησή τους, δεν αποτελούν σταθερά δεδομένα, αλλά δυναμικά επομένως μπορεί να αλλάξουν ανά πάσα στιγμή και άρα δεν είναι γνωστά εκ των προτέρων.
TDVRP	Time Dependent Vehicle Routing Problem	Χρόνο-εξαρτώμενη Δρομολόγηση Στόλου Οχημάτων	Λαμβάνεται υπόψη η κίνηση του χρησιμοποιούμενου δικτύου κάθε χρονική στιγμή με αποτέλεσμα ο χρόνος που απαιτείται από έναν κόμβο σε κάποιον άλλο να είναι μεταβλητός.
HFVRP	Heterogeneous Fleet Vehicle Routing Problem	Δρομολόγηση Ετερογενούς Στόλου Οχημάτων	Ετερογενής στόλος οχημάτων που μπορεί να οφείλεται στη διαφορετική χωρητικότητά τους, στην τεχνολογία που χρησιμοποιούν, καθώς και στον τρόπο κοστολόγησής τους.
VRPRB	Vehicle Routing Problem with Route Balancing	Δρομολόγηση Στόλου Οχημάτων με Εξισορρόπηση Διαδρομών	Εξισορρόπηση των διαδρομών, δηλαδή μείωση μιας μεγάλης διαδρομής με παράλληλη αύξηση των κόμβων που θα επισκεφθεί μια μικρή διαδρομή, με σκοπό την ίση κατανομή του φόρτου εργασίας μεταξύ των οδηγών των οχημάτων.
DVRP	Dynamic Vehicle Routing Problem	Δυναμική Δρομολόγηση Στόλου Οχημάτων	Λαμβάνεται υπόψη η δυνατότητα που παρέχεται στον πελάτη να ακυρώνει ή να τοποθετεί νέες παραγγελίες κατά τη διάρκεια κίνησης του οχήματος.

5. Συμπεράσματα

Το Πρόβλημα του Περιοδεύοντος Πωλητή είναι ένα από τα σημαντικότερα προβλήματα της θεωρίας βελτιστοποίησης και από το 1930, όταν και πρωτοδιατυπώθηκε, μέχρι και σήμερα, το ενδιαφέρον των ερευνητών για αυτό παραμένει αμείωτο. Η απλότητα της διατύπωσης του προβλήματος σε συνδυασμό με τη δυσκολία εξέτασης όλων των πιθανών του λύσεων το έχει κάνει ένα από τα πιο μελετημένα προβλήματα των τελευταίων δεκαετιών.

Στην πιο απλή διατύπωση του προβλήματος, ένας πωλητής ξεκινάει από μία αρχική πόλη με σκοπό να επισκεφτεί από ακριβώς μία φορά κάθε πόλη μιας δοθείσας λίστας και στη συνέχεια να επιστρέψει στην αρχική πόλη, επιλέγοντας όμως τη σειρά με την οποία θα επισκεφτεί τις πόλεις, έτσι ώστε η συνολική απόσταση που θα διανύσει να είναι η μικρότερη δυνατή.

Με το πέρασμα των χρόνων και τις ολοένα και αυξανόμενες ανάγκες της βιομηχανίας για βελτιστοποίηση της παραγωγικής τους διαδικασίας, το αρχικό πρόβλημα άρχισε σταδιακά να αποκτάει διάφορες παραλλαγές, κάθε μία εκ των οποίων ανταποκρίνεται καλύτερα στις διαφορετικές ανάγκες των σύγχρονων βιομηχανιών. Έτσι εμφανίστηκαν παραλλαγές όπως το TSP με Διαδοχικές Παραγγελίες, το «Αρκετά Κοντά» TSP, το TSP με Χρονικούς Περιορισμούς, το TSP με Πολλαπλούς Πωλητές, το TSP με Σημεία Συμφόρησης και πολλά άλλα. Κάθε ένα από αυτά απαντάει σε διαφορετικές ανάγκες των βιομηχανιών αλλά κοινός στόχος όλων αυτών είναι η βελτιστοποίηση μιας αρχικής δοθείσας αντικειμενικής συνάρτησης.

Μέχρι σήμερα πλήθος ερευνητών έχουν αναπτύξει πολλές μεθόδους που προσπαθούν να βρούνε το ολικό βέλτιστο στο TSP στον ταχύτερο δυνατό χρόνο και ανεξαρτήτως μεγέθους του προβλήματος. Στους πιο σημαντικούς ακριβείς αλγορίθμους που έχουν αναπτυχθεί συγκαταλέγονται ο Branch and Bound, ο Cutting Plane και ο Lin-Kernighan ενώ έχει δημιουργηθεί και το TSPLIB που είναι μια βιβλιοθήκη δειγμάτων για το TSP και ο TSP Solver της Concorde που διατίθεται δωρεάν και βοηθάει τους σύγχρονους ερευνητές να επιλύσουν προβλήματα συνδυαστικής βελτιστοποίησης όπως το TSP.

Όπου οι ακριβείς αλγόριθμοι δε μπορούν να δώσουν λύση σε εύλογο χρονικό διάστημα έχουν αναπτυχθεί διάφοροι ευρετικοί αλγόριθμοι όπως ο Γενετικός Αλγόριθμος, ο Αλγόριθμος Χριστοφίδη, η Αναζήτηση Ταμπού, η Βελτιστοποίηση Αποικίας Μυρμηγκιών, η Βελτιστοποίηση Σμήνους Σωματιδίων, η Προσομοιωμένη Ανόπτηση και πολλοί άλλοι. Συχνά οι ευρετικοί αλγόριθμοι συνδυάζονται με αλγόριθμους τοπικής αναζήτησης για τη βελτίωση της αποτελεσματικότητάς τους και προκύπτουν έτσι αλγόριθμοι όπως οι 2-opt και οι 3-opt.

Πολλές εργασίες μέχρι σήμερα έχουν επιχειρήσει να αξιολογήσουν και να συγκρίνουν μεταξύ τους τούς διάφορους αλγορίθμους όμως το ποιά μέθοδος είναι η καλύτερη συχνά εξαρτάται από την παραλλαγή του προβλήματος, το πλήθος των πόλεων του δικτύου που εξετάζουν καθώς και τη ρύθμιση των αρχικών παραμέτρων, όπου αυτές υπάρχουν.

Τέλος, υπάρχουν αναρίθμητες επιχειρησιακές εφαρμογές του Προβλήματος του Περιοδεύοντος Πωλητή και των παραλλαγών του σχεδόν σε όλους τους τομείς της βιομηχανίας καθώς και σε φαινομενικά διαφορετικούς κλάδους όπως σε αυτούς της ιατρικής, των πολιτικών μηχανικών, της βιολογίας, της χημείας, της δρομολόγησης οχημάτων και άλλων.

6. Εισηγήσεις για Περαιτέρω Έρευνα

Επειδή το Πρόβλημα του Περιοδεύοντος Πωλητή έχει μελετηθεί διεξοδικά θα είχε ενδιαφέρον μια μελέτη της εξέλιξης της μαθηματικής του διατύπωσης, πώς αυτή έχει μεταβληθεί τα τελυταία χρόνια, κυρίως ως προς τις αρχικές του παραμέτρους.

Ακόμα, θα έχει πολύ ενδιαφέρον μια σε βάθος μελέτη των παραλλαγών του Προβλήματος του Περιοδεύοντος Πωλητή, ποιά κοινά χαρακτηριστικά μοιράζονται, ποιές είναι οι διαφορές τους ως προς τη μαθηματική τους διατύπωση και πώς με το πέρασμα των χρόνων και με την εξέλιξη των επιχειρησιακών εφαρμογών οι παραλλαγές αυτές εξελίχθηκαν και έγιναν σταδιακά πιο πολύπλοκες.

Σαν πρόταση για περαιτέρω έρευνα αναφέρουμε ακόμα την εκτενέστερη και πληρέστερη σύγκριση των διαφόρων αλγορίθμων μεταξύ τους. Υπάρχουν αρκετές εργασίες που ασχολούνται με το θέμα αυτό όμως ελάχιστες εξετάζουν ταυτόχρονα μεγάλο πλήθος αλγορίθμων. Επιπλέον, ποιά ειναι τα δυνατά και τα αδύνατα σημεία του κάθε αλγορίθμου σε σχέση με τους υπόλοιπους καθώς και το σε ποιές περιπτώσεις κάθε αλγόριθμος δουλεύει καλύτερα, δεν είναι, ακόμα, πάντα σαφές.

Τέλος, αν και στην παρούσα εργασία αναφέρονται πάρα πολλές επιχειρησιακές εφαρμογές, το κεφάλαιο 4 θα μπορούσε να ανανεωθεί με πιο πρόσφατα και πιο πολύπλοκα παραδείγματα της καθημερινότητας αλλά και προκλήσεων που αντιμετωπίζουν οι σύγχρονες επιχειρήσεις.

Βιβλιογραφία

[1] Ι. Βαρελλας-Ουζουνόπουλος, Μελέτη νέου αλγόριθμου για την επίλυση του «προβλήματος του περιοδεύοντος πωλητή» και η προσέγγιση του πολυκριτηριακού προβλήματος, Αθήνα: Εθνικό Μετσόβιο Πολυτεχνείο, 2014.

[2] Α. Τ. Χαραλάμπους, Βελτιστοποίηση διαδρομής ποδηλατοδρόμου με χρήση Visual Basic for Applications. Εφαρμογή στη δημοτική κοινότητα Νέου Ψυχικού, Αθήνα: Εθνικό Μετσόβιο Πολυτεχνείο, 2016.

[3] Wikipedia, "Travelling salesman problem".

[4] Ν. Στυλιανού, Προσεγγίζοντας το Πρόβλημα του Πλανόδιου Πωλητή, Πάτρα: Πανεπιστήμιο Πατρών, 2013.

[5] C. E. Miller, Tucker A. W. and R. A. Zemlin, "Integer Programming Formulation of Traveling Salesman Problems," *Journal of the ACM,* vol. 7, no. 4, pp. 326-329, 1960.

[6] G. Gutin, A. Yeo and A. Zverovitch, "Exponential neighborhoods and domination analysis for the TSP," *ResearchGate,* p. 38, 2001.

[7] J. Walker, "Simulated Annealing The Travelling Salesman Problem," June 2018. [Online]. Available: https://www.fourmilab.ch/documents/travelling/anneal/.

[8] S. P. Fekete, "Simplicity and Hardness of the Maximum Traveling Salesman Problem under Geometric Distances," in *10th annual ACM-SIAM symposium on Discrete algorithms*, Baltimore, Maryland, 1999.

[9] G. Reinelt, "TSPLIB," 2013. [Online]. Available: https://wwwproxy.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.

[10] M. R. Garey and D. S. Johnson, Computers and Intractability - A Guide to the Theory of NP-Completeness, Freeman, 1979.

[11] M. Pinedo, Scheduling: Theory, Algorithms and Systems, Third ed. Prentice Hall, 2008.

[12] A. A. Cire and W.-J. v. Hoeve, "MDD propagation for disjunctive scheduling," in *Twenty-Second International Conference on Automated Planning and Scheduling*, 2012.

[13] C. Y. Lee, "Representation of switching circuits by binary-decision programs," *Bell Systems Technical Journal,* vol. 38, pp. 985-999, 1959.

[14] B. Becker, M. Behle, F. Eisenbrand and R. Wimmer, "BDDs in a branch and cut framework," in *Proceedings of the 4th International Workshop on Effcient and Experimental Algorithms*, 2005.

[15] D. Bergman, A. A. Cire, W.-J. v. Hoeve and J. N. Hooker, "Variable ordering for the application of bdds to the maximum independent set problem," in *Proceedings of the 9th international conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Berlin, Heidelberg, 2012.

[16] H. R. Andersen, T. Hadzic, J. N. Hooker and P. Tiedemann, "A constraint store based on multivalued decision diagrams," in *Proceedings of the 13th international conference on Principles and practice of constraint programming*, Berlin, Heidelberg, 2007.

[17] S. Hoda, W. Jan van Hoeve and J. N. Hooker, "A systematic approach to MDD-based constraint programming," in *16th international conference on Principles and practice of constraint programming*, Berlin, Heidelberg, 2010.

[18] E. Balas, "New classes of effciently solvable generalized traveling salesman problems," *Annals of Operations Research,* vol. 86, pp. 529-558, 1999.

[19] P. Baptiste, C. L. Pape and W. Nuijten, Constraint-Based Scheduling: Applying Constraint Programming to Scheduling Problems, Kluwer: International Series in Operations Research and Management Science, 2001.

[20] G. Freuder and M. Wallace, "Constraint technology and the commercial world," *Intelligent Systems and their Applications,* vol. 15, no. 1, pp. 20-23, 2000.

[21] T. Lopes, A. A. Cire, C. d. Souza and A. Moura, "A hybrid model for a multiproduct pipeline planning and scheduling problem," *Constraints,* vol. 15, pp. 151-189, 2010.

[22] A. Rendl, M. Prandtstetter, G. Hiermann, J. Puchinger and G. Raidl, "Hybrid heuristics for multimodal homecare scheduling," in *Proceedings of the 9th international conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Berlin, Heidelberg, 2012.

[23] A. A. Cire and W.-J. van Hoeve, "Multivalued Decision Diagrams for Sequencing Problems," 2013.

[24] W. Mennell, B. Golden and E. Wasil, "Solving the Close Enough Traveling Salesman Problem," in *9th INFORMS Telecommunications Conference*, Maryland, 2008.

[25] W. Mennell, Heuristics for solving three routing problems: Close-Enough Traveling Salesman Problem Close-Enough Vehicle Routing Problem, Sequence-Dependent Team Orienteering Problem, Meryland: University of Meryland, 2009.

[26] E. M. Arkin and R. Hassin, "Approximation Algorithms For The Geometric Covering Salesman Problem," *Discrete Applied Mathematics,* vol. 55, no. 3, pp. 197-218, December 1994.

[27] J. Silberholz and B. Golden, "The Generalized Traveling Salesman Problem: A New Genetic Algorithm Approach," *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies,* vol. 37, pp. 165-181, 2007.

[28] M. Gendreau, G. Laporte and F. Semet, "The Covering Tour Problem," *Operations Research,* vol. 45, no. 4, pp. 568-576, 1997.

[29] W. Pereira Coutinho, A. Subramanian, R. Quirino do Nascimento and A. Alves Pessoa, "A Branch-and-Bound Algorithm for the Close-Enough Traveling Salesman Problem," 2013.

[30] M. W. P. Savelsbergh, "Local search in routing problems with time windows," *Annals of Operations Research,* vol. 4, no. 1, pp. 285-305, 1985.

[31] M. Lopez-Ibanez, C. Blum, J. W. Ohlmann and B. W. Thomas, "The Travelling Salesman Problem with Time Windows: Adapting Algorithms from Travel-time to Makespan Optimization," Iridia, Bruxelles, 2013.

[32] "Multiple Traveling Salesman Problem (mTSP)," Neos guide, [Online]. Available: https://neos-guide.org/content/multiple-traveling-salesman-problem-mtsp. [Accessed 21 Septembre 2019].

[33] G. Laporte and Y. Nobert, "A cutting planes algorithm for the m-salesmen problem," *Journal of the Operational Research Society,* vol. 31, no. 11, pp. 1017-1023, 1980.

[34] N. Christofides, A. Mingozzi and P. Toth, "Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations," *Mathematical Programming,* vol. 20, no. 1, pp. 255-282, 1981.

[35] G. Carpaneto, S. Martello and P. Toth, "An algorithm for the bottleneck traveling salesman problem," *Operational Research,* vol. 32, pp. 380-389, 1984.

[36] M. Junger, G. Reinelt and G. Rinaldi, "Chapter 4: The traveling salesman problem," in *Network Models*, Elsevier Science B. V., 1995, pp. 225-330.

[37] C. Bordenave, M. Gendreau and G. Laporte, "A branch-and-cut algorithm for the nonpreemptive swapping problem," *Naval Research Logistics,* vol. 56, no. 5, pp. 478-486, 2009.

[38] H. Hernandez-Perez and J.-J. Salazar-Gonzalez, "Heuristics procedures to solve the multi-commodity Pickup-and-Delivery Traveling Salesman Problem," 2009.

[39] L. Gouveia and M. Ruthmair, "Load-Dependent and Precedence-Based Models for Pickup and Delivery Problems," *Preprint submitted to Computers & Operations Research,* p. 28, 2015.

[40] D. Menerba, R. Mansini and J. Riera-Ledesma, "The Traveling Purchaser Problem and its Variants," *European Journal of Operational Research,* 2016.

[41] D. Feillet, P. Dejax and M. Gendreau, "Traveling Salesman Problems with Profits," *Transport. Sci.,* vol. 39, no. 2, pp. 188-205, 2005.

[42] Z. Degraeve, E. Labro and F. Roodhooft, "An evaluation of vendor selection models from a total cost of ownership prospective," *European Journal of Operational Research,* vol. 125, no. 1, pp. 35-48, 2000.

[43] A. M. Rodrigues and J. Soeiro Ferreira, "Solving the Rural Postman Problem by Memetic Algorithms," in *4th Metaheuristics International Conference*, Porto, Portugal, 2001.

[44] J. Edmonds and E. Johnson, "Matching, Euler tours and the Chinese postman," *Math. Program,* vol. 5, pp. 88-124, 1973.

[45] R. K. Ahuja, T. L. Magnati, J. B. Orlin and M. R. Reddy, "Chapter 1 Applications of network optimization," in *Network Models*, Elsevier B. V., 1995, pp. 1-83.

[46] L. Dongmei, W. Xiangbin and W. Dong, "Exact Heuristic Algorithm for Traveling Salesman Problem," *9th International Conference for Young Computer Scientists,* pp. 9-13, 2008.

[47] J. Štencek, Traveling salesman problem, Jyväskylä: JAMK University of Applied Sciences, 2013.

[48] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman Publications, 1979.

[49] Έ. Ρόκου and Κ. Κηρυττόπουλος, "Το Πρόβλημα του Περιοδεύοντος Πωλητή - The Travelling Salesman Problem," Αθήνα, 2012.

[50] G. B. Dantzig, D. R. Fulkerson and S. M. Johnson, "Solution of a large scale travelling salesman problem," *Operations Research,* vol. 2, pp. 393-410, 1954.

[51] M. W. Padberg and S. Hong, "On the symmetric traveling salesman problem: a computational study," *Mathematical Programming Study,* vol. 12, pp. 78-107, 1980.

[52] H. Crowder and M. W. Padberg, "Solving Large-Scale Symmetric Travelling Salesman Problems to Optimality," *Management Science,* vol. 26, no. 5, pp. 495-509, 1980.

[53] P. Miliotis, "Using Cutting Planes to solve the Symmetric Travelling Salesman Problem," *Mathematical Programming,* vol. 15, pp. 177-188, 1978.

[54] S. Lin and B. W. Kernighan, "An Effective Heuristic Algorithm for the Traveling-Salesman Problem," *Operations Research,* vol. 21, no. 2, pp. 498-516, 1973.

[55] S. Lin, "Computer solutions of the traveling salesman problem," *Bell System Technical Journal,* vol. 44, pp. 2245-2269, 1965.

[56] K. Helsgaun, "An effective implementation of the Lin-Kernighan traveling salesman heuristic," *European Journal of Operational Research,* vol. 126, pp. 106-130, 2000.

[57] R. Tinós, K. Helsgaun and D. Whitley, "Efficient Recombination in the Lin-Kernighan-Helsgaun Traveling Salesman Heuristic," in *International Conference on Parallel Problem Solving from Nature*, Coimbra Portugal, 2018.

[58] W. Cook, "Concorde TSP Solver," [Online]. Available: http://www.math.uwaterloo.ca/tsp/concorde/index.html. [Accessed 21 Septembre 2019].

[59] H. A. Abdulkarim and I. F. Alshammari, "Comparison of Algorithms for Solving Traveling Salesman Problem," *International Journal of Engineering and Advanced Technology (IJEAT),* vol. 4, no. 6, pp. 76-79, 2015.

[60] H. E. Lehtihet, "What are the differences between heuristics and metaheuristics?," 7 Ιούλιος 2013. [Online]. Available: https://www.researchgate.net/post/What_are_the_differences_between_heuristics_and_metaheuristics.

[61] K. Sorensen, "Metaheuristics - the metaphor exposed," *International Transactions in Operational Research,* pp. 3-18, 2012.

[62] D. S. Johnson and L. A. McGeoch, The Traveling Salesman Problem: A Case Study in Local Optimization, 1995.

[63] N. Kumar, Karambir and R. Kumar, "A genetic algorithm approach to study travelling salesman problem," *Journal of Global Research in Computer Science,* vol. 3, no. 3, pp. 33-38, 2012.

[64] A. Khushboo, A. Samiksha and T. Rohit, "Solving TSP using Genetic Algorithm and Nearest Neighbour Algorithm and their Comparison," *International Journal of Scientific & Engineering Research,* vol. 7, no. 1, pp. 1014-1018, 2016.

[65] R. Matai, S. P. Singh and M. L. Mittal, "Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution Approaches," 2010.

[66] M. Khalil, J.-P. Li, Y. Wang and A. Khan, "Algorithm to solve Travel Salesman Problem efficently," in *13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Chengdu, 2016.

[67] F. Glover, "Future Paths for Integer Programming and Links to Artificial Intelligence," *Computers and Operations Research,* vol. 13, pp. 533-549, 1986.

[68] C. D. Tarantilis and C. T. Kiranoudis, "A flexible adaptive memory-based algorithm for real-life transportation operations: two case studies from dairy and construction sector," *European Journal of Operational Research,* pp. 806-822, 2007.

[69] M. Ahmadvand, M. Yousefikhoshbakht and N. M. Darani, "Solving the Traveling Salesman Problem by an Efficient Hybrid Metaheuristic Algorithm," *Journal of Advances in Computer Research,* vol. 3, no. 3, pp. 75-84, 2012.

[70] M. Dorigo, E. Bonabeau and G. Theraulaz, "Ant algorithms and stigmergy," *Future Generation Computer Systems,* vol. 16, pp. 851-871, 2000.

[71] B. Bullnheimer, R. F. Hartl and C. Strauss, "An improved ant system algorithm for the vehicle routing problem," *Annals of Operations Research,* vol. 89, pp. 319-328, 1999.

[72] G. Di Caro and M. Dorigo, "AntNet: distributed stigmergetic control for communications networks," *Journal Of Artificial Intelligence Research,* vol. 9, pp. 317-365, 1998.

[73] L. M. Gambardella and M. Dorigo, "HAS-SOP: hybrid ant system for the sequential ordering problem," Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, Lugano, Switzerland, 1997.

[74] V. Maniezzo, "Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem," *INFORMS Journal on Computing,* vol. 11, no. 4, pp. 329-431, 1999.

[75] M. Yousefikhoshbakht and M. Sedighpour, "An Optimization Algorithm for the Capacitated Vehicle Routing Problem Based on Ant Colony System," *Australian Journal of Basic and Applied Sciences,* vol. 5, no. 12, pp. 2729-2737, 2011.

[76] M. Dorigo, L. M. Gambardella and M. P. Vecchi, "Ant colonies for the travelling salesman problem," *Biosystems,* vol. 43, no. 2, pp. 73-81, 1997.

[77] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *IEEE: International Conference on Neural Networks*, Perth, Australia, 1995.

[78] H. T. Liang and F. H. Kang, "Adaptive mutation particle swarm algorithm with dynamic nonlinear changed inertia weight," *Optik,* pp. 8036-8042, 2016.

[79] K. E. Parsopoulos, V. P. Plagianakos, G. D. Magoulas and M. N. Vrahatis, "Objective function “stretching” to alleviate convergence to local minima," *Nonlinear Analysis: Theory, Methods & Applications,* vol. 47, no. 5, pp. 3419-3424, 2001.

[80] R. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *7th Annual Conference on Evolutionary Programming*, San Diego, USA, 1998.

[81] Y. Shi and R. Eberhart, "Fuzzy adaptive particle swarm optimization," in *Congress on Evolutionary Computation*, Seoul, Korea, 2001.

[82] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *IEEE International Conference on Evolutionary Computation*, Anchorage, USA, 1998.

[83] L. Zhang, Y. Tang, C. Hua and X. Guan, "A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques," *Applied Soft Computing,* vol. 28, pp. 138-149, 2015.

[84] H. H. Chang, L. S. Lin, N. M. Chen and W. J. Lee, "Particle-swarm-optimization-based nonintrusive demand monitoring and load identification in smart meters," *IEEE Transactions on Industry Applications,* vol. 49, no. 5, pp. 2229-2236, 2013.

[85] Z. A. Bashir and M. E. El-Hawary, "Applying Wavelets to Short-Term Load Forecasting Using PSO-Based Neural Networks," *IEEE Transactions on Power Systems,* vol. 24, no. 1, pp. 20-27, 2009.

[86] W. Si, H. Ogai, K. Hirai and T. Li, "An Improved Control Algorithm for Lighting Systems by Using PSO," *IEEJ Transactions on Electronics, Information and Systems,* vol. 133, no. 8, pp. 1501-1508, 2013.

[87] S. Milner, C. Davis, H. Zhang and J. Llorca, "Nature-Inspired Self-Organization, Control, and Optimization in Heterogeneous Wireless Networks," *IEEE Transactions on Mobile Computing,* vol. 11, no. 7, pp. 1207-1222, 2012.

[88] Y. Zhang, S. Wang and G. Ji, "A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications," *Mathematical Problems in Engineering,* 2015.

[89] A. G. Khachaturyan, S. V. Semenovskaya and B. K. Vainshtein, "Statistical-Thermodynamic Approach to Determination of Structure Amplitude Phases," *Soviet physics, crystallography,* vol. 24, no. 5, pp. 519-524, 1979.

[90] S. Kirkpatrick, D. C. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing," *Science,* vol. 220, no. 4598, pp. 671-680, 1983.

[91] M. Grötschel , The Travelling Salesman Problem and its Applications, Berlin, 2009.

[92] K. Arora, S. Agarwal and R. Tanwar, "Solving TSP using Genetic Algorithm and Nearest Neighbour Algorithm and their Comparison," *International Journal of Scientific & Engineering Research,* vol. 7, no. 1, pp. 1014-1018, 2016.

[93] O. Míča, "Comparison of metaheuristic methods by solving travelling salesman Problem," *The International Scientific Conference INPROFORUM,* 2015.

[94] P. Dostál and Kratochvíl, "The Comparison of Methods Solving the Travel Salesman Problem," 2010.

[95] M. Grötschel, M. Jünger and G. Reinelt, "Optimal Control of Plotting and Drilling Machines: A Case Study," *Mathematical Methods of Operations Research,* vol. 35, no. 1, pp. 61-84, 1991.

[96] R. Plante, T. Lowe and R. Chandrasekaran, "The Product Matrix Traveling Salesman Problem: An Application and Solution Heuristics.," *Operations Research,* vol. 35, pp. 772-783, 1987.

[97] R. Bland and D. Shallcross, "Large traveling salesman problem arising from experiments in X-ray crystallography: a preliminary report on computation," *Operations Research Letters,* vol. 8, no. 3, pp. 125-128, 1989.

[98] J. Lenstra and A. Rinnooy Kan, Some Simple Applications of the Travelling Salesman Problem, Amsterdam: Stichting Mathematisch Centrum, 1974.

[99] H. Ratliff and A. Rosenthal, "Order-Picking in a Rectangular Warehouse: A Solvable Case for the Travelling Salesman Problem," *Operations Research,* vol. 31, pp. 507-521, 1983.

[100] F. Scholz, "Coordination hole tolerance stacking," Boeing Computer Services, 1993.

[101] E. M. Arkin, Y.-J. Chiang, J. S. B. Mitchell, S. S. Skiena and T.-C. Yang, "On the Maximum Scatter TSP," in *8th annual ACM-SIAM Symposium on Discrete Algorithms*, New Orleans, Louisiana, USA, 1997.

[102] D. J. Gulczynski, J. W. Heath and J. J. Price, "The Close Enough Traveling Salesman Problem: A Discussion of Several Heuristics," *Perspectives in Operations Research,* vol. 36, pp. 271-283, 2006.

[103] J. Dong, N. Yang and M. Chen, "Heuristic Approaches for a TSP Variant: The Automatic Meter Reading Shortest Tour Problem," *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies,* vol. 37, pp. 145-163, 2007.

[104] R. Angel, W. Caudle and R. &. W. A. Noonan, "Computer assisted school bus scheduling," *Management Science,* vol. 18, p. 279–88, 1972.

[105] J. Svestka and V. Huckfeldt, "Computational experience with an m-salesman traveling salesman algorithm," *Management Science,* vol. 19, no. 7, p. 790–799, 1973.

[106] J. Lenstra and A. Rinnooy Kan, "Some simple applications of the traveling salesman problem," *Operational Research Quarterly,* vol. 26, p. 717–733, 1975.

[107] T. Zhang, W. Gruver and M. Smith, "Team scheduling by genetic search," in *Proceedings of the second international conference on intelligent processing and manufacturing of materials*, 1999.

[108] K. Gilbert and R. Hofstra, "A new multiperiod multiple traveling salesman problem with heuristic and application to a scheduling problem," *Decision Sciences,* vol. 23, p. 250–259, 1992.

[109] L. Tang, J. Liu, A. Rong and Z. Yang, "A multiple traveling salesman problem model for hot rolling scheduling in Shangai Baoshan Iron & Steel Complex," *European Journal of Operational Research,* vol. 124, p. 267–282, 2000.

[110] B. Brummit and A. Stentz, "Dynamic mission planning for multiple mobile robots," *Proceedings of the IEEE international conference on robotics and automation,* 1996.

[111] B. Brummit and A. Stentz, "GRAMMPS: a generalized mission planner for multiple mobile robots," *Proceedings of the IEEE international conference on robotics and automation,* 1998.

[112] Z. Yu, L. Jinhai, G. Guochang, Z. Rubo and Y. Haiyan, "An implementation of evolutionary computation for path planning of cooperative mobile robots," 2002.

[113] J. Ryan, T. Bailey, J. Moore and W. Carlton, "Reactive Tabu search in unmanned aerial reconnaissance simulations," in *1998 Winter Simulation Conference*, Washington, DC, USA, 1998.

[114] H. Saleh and R. Chelouah, "The design of the global navigation satellite system surveying networks using genetic algorithms," *Engineering Applications of Artificial Intelligence,* vol. 17, p. 111–122, 2004.

[115] C. Okonjo-Adigwe, "An effective method of balancing the workload amongst salesmen," *Omega,* vol. 16, no. 2, p. 159–163, 1988.

[116] R. Calvo and R. Cordone, "A heuristic approach to the overnight security service problem," *Computers and Operations Research,* vol. 30, p. 1269–1287, 2003.

[117] K. Kim and Y. Park, "A crane scheduling method for port container terminals," *European Journal of Operational Research,* vol. 156, p. 752–768, 2004.

[118] D. Chemla, F. Meunier and R. Wolfler Calvo, "Bike hiring system: solving the rebalancing problem in the static case," *Discrete Optimization,* vol. 10, no. 2, pp. 120-146, 2013.

[119] T. Raviv, M. Tzur and I. A. Forma, "Static repositioning in a bike-sharing system: models and solution approaches," *EURO Journal on Transportation and Logistics,* vol. 2, no. 3, pp. 187-229, 2013.

[120] M. Grötschel, M. Jünger and G. Reinelt, "Optimal Control of Plotting and Drilling Machines: A Case Study.," *Mathematical Methods of Operations Research,* vol. 35, no. 1, pp. 61-84, 1991.

[121] Κ. Κετσάτη, Το Πρόβλημα της Δρομολόγησης Οχημάτων - Διαχείριση Διανομών σε Ελληνική Εταιρία, Θεσσαλονίκη: Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, 2017.

[122] B. Ombuki - Berman and F. Hanshar, "Using Genetic Algorithms for Multi-depot Vehicle Routing," *Bio-inspired Algorithms for the Vehicle Routing Problem Studies in Computational Intelligence,* vol. 161, pp. 77-99, 2009.

[123] G. Ninikas, Solving the dynamic Vehicle Routing Problem with mixed backhauls through re-optimization, Chios, Grece: University of the Aegean, 2014.

[124] M. Jafari-Eskandari, A. R. Aliahmadi and G. H. H. Khaleghi, "A robust optimisation approach for the milk run problem with time windows with inventory uncertainty: an auto industry supply chain case study," *International Journal of Rapid Manufacturing,* vol. 1, no. 3, pp. 334-347, 2010.