



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

**«Συγκριτική Μελέτη Αλγορίθμων για την επίλυση
προβλήματος πλανόδιου πωλητή με χρονικά παράθυρα»**

Υπό
ΚΑΡΡΑΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

Διπλωματική Εργασία

Υπεβλήθη για την εκπλήρωση μέρους των απαιτήσεων για
την απόκτηση του Διπλώματος Μηχανολόγου Μηχανικού

Βόλος, 2022

[ΚΕΝΗ ΣΕΛΙΔΑ]

© 2022 Κωνσταντίνος Καρράς

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανολόγων Μηχανικών της Πολυτεχνικής Σχολής του Πανεπιστημίου Θεσσαλίας δεν υποδηλώνει αποδοχή των απόψεων του συγγραφέα (Ν. 5343/32 αρ. 202 παρ. 2).

Εγκρίθηκε από τα Μέλη της Τριμελούς Εξεταστικής Επιτροπής:

Πρώτος Εξεταστής (Επιβλέπων)	Σαχαρίδης Γεώργιος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας
Δεύτερος Εξεταστής	Παντελής Δημήτριος, Δεύτερος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας
Τρίτος Εξεταστής	Λυμπερόπουλος Γεώργιος Αναπληρωτής Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας

Συγκριτική Μελέτη Αλγορίθμων για την επίλυση προβλήματος πλανόδιου πωλητή με χρονικά παράθυρα

ΚΑΡΡΑΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας, 2022

Επιβλέπων Καθηγητής: Σαχαρίδης Γεώργιος

Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας ήταν η συγκριτική μελέτη αλγορίθμων για την επίλυση προβλήματος πλανόδιου πωλητή με χρονικά παράθυρα. Αρχικά στο πρώτο κεφάλαιο γίνεται ιστορική αναδρομή του προβλήματος του περιοδεύοντος πωλητή και αναφοράς στους τρόπους επίλυσης. Ακολούθως στο επόμενο κεφάλαιο ορίζεται το πώς μεταβάλλεται το ανωτέρω πρόβλημα με την εισαγωγή της έννοιας των χρονικών παραθύρων σε αυτό. Το μαθηματικό μοντέλο του προβλήματος αναπτύσσεται τόσο για το συμμετρικό όσο και το ασύμμετρο πρόβλημα. Στο επόμενο κεφάλαιο επιλέγονται τέσσερις μέθοδοι επίλυσης του προβλήματος από την βιβλιογραφία και αναπτύσσεται το μαθηματικό μοντέλο που τις διέπει καθώς επίσης και τα αποτελέσματα τις εκάστης μεθόδου. Αυτές οι μέθοδοι είναι οι εξής: Επίλυση του προβλήματος του περιοδεύοντος πωλητή με χρονικά παράθυρα α) με χρήση λαγκραντζιανής χαλάρωσης β) με χρήση χρονικά διευρυμένων δικτύων γ) με χρήση υβριδικών δικτύων δεικτών με χρονικά χαρακτηριστικά και τέλος δ) με χρήση δυναμικής διακριτοποίησης. Τέλος στο τελευταίο κεφάλαιο παρουσιάζονται τα συμπεράσματα της εργασίας.

Λέξεις-κλειδιά: περιοδεύοντος πωλητή, χρονικά παράθυρα, αλγόριθμος, TSP-TW

A Comparative Study of Algorithms for Solving the Traveling Salesman Problem with Time Windows

KARRAS KONSTANTINOS

Department of Mechanical Engineering, University of Thessaly, 2022

Supervisor: DR Saxaridis. Georgios

Abstract

The purpose of this thesis was the comparative study of algorithms for solving the traveling salesman problem with time windows. Initially, in the first chapter, there is a historical review of the problem and a reference to the ways of solving it. Then, in the next chapter, it is defined how the above problem is changed by introducing the concept of time windows into it. The mathematical model of the problem is developed for both the symmetric and the asymmetric problem. In the next chapter four methods for solving the problem are selected from the literature and the mathematical model that governs them is developed as well as the results of each method. These methods are as follows: Solving the traveling salesman problem with time windows a) using Lagrangian relaxation b) using time-expanded networks c) using hybrid pointer networks with time features and finally d) using dynamic discretization. Finally, the conclusions of the work are presented in the last chapter.

Keywords: traveling salesman problem, time windows, algorithm, TSP-TW

Περιεχόμενα

<i>Περίληψη</i>	4
<i>Abstract</i>	5
<i>Κεφάλαιο 1^ο</i>	8
1.1 Εισαγωγή	8
1.2 Ιστορική αναδρομή	8
1.3 Εφαρμογή TSP και συνδέσεις με άλλα προβλήματα.....	9
1.3.1 . Διάτρηση πλακών τυπωμένου κυκλώματος	9
1.3.2 Γενική επισκευή κινητήρων αεριοστροβίλων	10
1.3.3 Κρυσταλλογραφία ακτίνων Χ	10
1.3.4 Καλωδίωση υπολογιστή	11
1.3.5 Το πρόβλημα της επιλογής παραγγελιών στις αποθήκες	11
1.3.6 Δρομολόγηση οχημάτων	12
1.3.7 Σχεδίαση μάσκας στην παραγωγή PCB	12
1.4 Σύγκριση Αλγορίθμων για την Επίλυση Προβλήματος Περιοδευόντος Πωλητή.....	13
1.4.1 Μέθοδος πλησιέστερου γειτονικού κόμβου	14
1.4.2 Γενετικός αλγόριθμος.....	16
1.4.3 Άπληστος Ευρετικός Αλγόριθμος	17
1.5 Πρόβλημα TSP με μεγαλύτερο αριθμό κόμβων.....	18
1.6 Πρόβλημα περιοδευόντος πωλητή με χρονικά παράθυρα.....	22
1.7 Σκοπός εργασίας.....	22
<i>Κεφάλαιο 2^ο</i>	24
2.1 Πρόβλημα περιοδευόντος πωλητή με χρονικά παράθυρα.....	24
2.2 Ανασκόπηση μελετών για προβλήματα TSP-TW	27
2.3 Ανασκόπηση μελετών για προβλήματα VRP-TW	31
2.4 Συμμετρικό πρόβλημα TSPTW	32
2.5 Ασύμμετρο πρόβλημα TSPTW	33

<i>Κεφάλαιο 3^ο</i>	37
3.1 Μεθοδολογίες επίλυσης του προβλήματος περιοδεύοντος πωλητή με χρονικά παράθυρα	37
3.1.1 Επίλυση του προβλήματος του περιοδεύοντος πωλητή με χρονικά παράθυρα με χρήση λαγκρατζιανής χαλάρωσης.....	37
3.1.2 Επίλυση του προβλήματος του περιοδεύοντος πωλητή με χρονικά παράθυρα με χρήση του χρονικά διευρυμένων δικτύων.....	46
3.1.3 Επίλυση του προβλήματος του περιοδεύοντος πωλητή με χρονικά παράθυρα με χρήση υβριδικών δικτύων δεικτών με χρονικά χαρακτηριστικά.	50
3.1.4 Επίλυση του προβλήματος του περιοδεύοντος πωλητή με χρονικά παράθυρα με χρήση Δυναμικής Διακριτοποίησης	54
<i>Κεφάλαιο 4^ο</i>	61
4.1 Συμπεράσματα	61
<i>ΒΙΒΛΙΟΓΡΑΦΙΑ</i>	65

Κεφάλαιο 1^ο

1.1 Εισαγωγή

Δεδομένου ενός συνόλου πόλεων και λαμβάνοντας υπόψη το κόστος μετακίνησης μεταξύ κάθε ζεύγους από αυτές, το πρόβλημα του περιοδεύοντος πωλητή, ή TSP για συντομία, χρησιμοποιείται για να υπολογιστεί ο φθηνότερος τρόπος ώστε κάποιος να επισκεφθεί το σύνολο των πόλεων και να επιστρέψει στο σημείο εκκίνησης.

Αυτή το πρόβλημα το οποίο αρχικά φαίνεται σχετικά απλό είναι στην πραγματικότητα ένα από τα πιο έντονα μελετημένα προβλήματα στα υπολογιστικά μαθηματικά. Έχει εμπνεύσει μελέτες από μαθηματικούς, επιστήμονες υπολογιστών, χημικούς, φυσικούς, ψυχολόγους και μια σειρά από μη επαγγελματίες ερευνητές. Οι εκπαιδευτικοί χρησιμοποιούν το TSP για να εισάγουν διακριτά μαθηματικά στο δημοτικό, γυμνάσιο και στο λύκειο καθώς και σε πανεπιστήμια και επαγγελματικά σχολεία. Το TSP έχει δει εφαρμογές στους τομείς της εφοδιαστικής, της γενετικής, της κατασκευής, των τηλεπικοινωνιών και της νευροεπιστήμης.

Η απήχηση του TSP το οδήγησε στο να αποτελεί ένα από τα λίγα σύγχρονα προβλήματα στα μαθηματικά το οποίο ταυτόχρονα έγινε μέρος της λαϊκής κουλτούρας. Το έξυπνο όνομά του σίγουρα έπαιξε κάποιο ρόλο, αλλά ο πρωταρχικός λόγος για το μεγάλο ενδιαφέρον είναι το γεγονός ότι αυτό το εύκολα κατανοητό μοντέλο εξακολουθεί να διαφεύγει από μια γενική λύση. Η απλότητα του TSP, σε συνδυασμό με τη φαινομενική δυσκολία του, το καθιστά ιδανική πλατφόρμα για την ανάπτυξη ιδεών και τεχνικών για την επίλυση σε υπολογιστικά προβλήματα γενικά.

1.2 Ιστορική αναδρομή

Η προέλευση του ονόματος «πρόβλημα περιοδεύοντος πωλητή» αποτελεί μυστήριο. Δεν φαίνεται να υπάρχει καμία έγκυρη τεκμηρίωση που να υποδεικνύει τον

δημιουργό του ονόματος και δεν έχουμε καλές εικασίες για το πότε τέθηκε σε χρήση για πρώτη φορά. Ένας από τους πρώτους ερευνητές του TSP με τη μεγαλύτερη επιρροή ήταν ο Merrill Flood του Πανεπιστημίου του Πρίνστον και της RAND Corporation. Εκτός από μικρές παραλλαγές στην ορθογραφία και τη στίξη, στα μέσα της δεκαετίας του 1950 το όνομα TSP ήταν σε ευρεία χρήση. Η πρώτη αναφορά που περιέχει τον όρο φαίνεται να είναι η αναφορά του 1949 της Julia Robinson, «On the Hamiltonian game (a traveling salesman problem)» [483], αλλά φαίνεται ξεκάθαρο από τη γραφή ότι δεν εισήγαγε το όνομα. Το μόνο που μπορούμε να συμπεράνουμε είναι ότι κάποια στιγμή κατά τη διάρκεια της δεκαετίας του 1930 ή του 1940, πιθανότατα στο Πρίνστον, το TSP πήρε το όνομά του και οι μαθηματικοί άρχισαν να μελετούν σοβαρά το πρόβλημα.

Αν και δεν μπορούμε να προσδιορίσουμε τον δημιουργό του ονόματος TSP, είναι εύκολο να υποστηρίξουμε ότι είναι ένα κατάλληλο αναγνωριστικό για το πρόβλημα της εύρεσης της συντομότερης διαδρομής μέσω πόλεων σε μια δεδομένη περιοχή.

1.3 Εφαρμογή TSP και συνδέσεις με άλλα προβλήματα

1.3.1 . Διάτρηση πλακών τυπωμένου κυκλώματος

Μια άμεση εφαρμογή του TSP είναι στο πρόβλημα διάτρησης των πλακών τυπωμένου κυκλώματος (PCB). Για να συνδεθεί ένας αγωγός σε ένα στρώμα με έναν αγωγό σε ένα άλλο στρώμα ή για να τοποθετηθούν οι ακίδες των ολοκληρωμένων κυκλωμάτων, πρέπει να διανοιχθούν αρχικά οπές στην πλακέτα. Οι οπές μπορεί να είναι διαφορετικών μεγεθών. Για να δημιουργηθούν δύο οπές διαφορετικής διαμέτρου διαδοχικά, η κεφαλή του μηχανήματος πρέπει να μετακινηθεί σε μια εργαλειοθήκη και να αλλάξει τον εξοπλισμό διάτρησης. Αυτό είναι αρκετά χρονοβόρο. Επομένως, είναι σαφές ότι πρέπει να επιλεγθεί κάποια διάμετρος, να ανοιχθούν όλες τις τρύπες της ίδιας διαμέτρου, να αλλαχθεί το τρυπάνι, να ανοιχθούν οι τρύπες της επόμενης διαμέτρου κ.λπ. Έτσι, αυτό το πρόβλημα διάτρησης μπορεί να θεωρηθεί ως μια σειρά από TSP, ένα για κάθε διάμετρο κάθε οπής, όπου οι «πόλεις» είναι η αρχική θέση και το σύνολο όλων των οπών που

μπορούν να διανοιχθούν με ένα συγκεκριμένο τρυπάνι. Η «απόσταση» μεταξύ δύο πόλων δίνεται από το χρόνο που χρειάζεται για να μετακινηθεί η κεφαλή γεώτρησης από τη μια θέση στην άλλη. Ο στόχος είναι να ελαχιστοποιηθεί ο χρόνος ταξιδιού για την κεφαλή του μηχανήματος.

1.3.2 Γενική επισκευή κινητήρων αεριοστροβίλων

Οι Plante et al., (1987) ανέφεραν αυτήν την εφαρμογή και συμβαίνει όταν οι κινητήρες αεριοστροβίλων αεροσκαφών πρέπει να επισκευαστούν. Για να διασφαλιστεί η ομοιόμορφη ροή αερίου μέσω των στροβίλων, υπάρχουν συγκροτήματα πτερυγίων οδηγού ακροφυσίου που βρίσκονται σε κάθε στάδιο του στροβίλου. Ένα τέτοιο συγκρότημα αποτελείται βασικά από έναν αριθμό οδηγών πτερυγίων ακροφυσίων στερεωμένα γύρω από την περιφέρειά του. Όλα αυτά τα πτερύγια έχουν μεμονωμένα χαρακτηριστικά και η σωστή τοποθέτηση των πτερυγίων μπορεί να οδηγήσει σε σημαντικά οφέλη (μείωση κραδασμών, αύξηση της ομοιομορφίας ροής, μείωση κατανάλωσης καυσίμου). Το πρόβλημα της τοποθέτησης των πτερυγίων με τον καλύτερο δυνατό τρόπο μπορεί να μοντελοποιηθεί ως TSP με ειδική αντικειμενική συνάρτηση.

1.3.3 Κρυσταλλογραφία ακτίνων X

Η ανάλυση της δομής των κρυστάλλων (Bland & Shallcross, 1989) [1] είναι μια σημαντική εφαρμογή του TSP. Εδώ χρησιμοποιείται ένα περιθλασίμετρο ακτίνων X για τη λήψη πληροφοριών σχετικά με τη δομή του κρυσταλλικού υλικού. Για το σκοπό αυτό ένας ανιχνευτής μετρά την ένταση των ανακλάσεων ακτίνων X του κρυστάλλου από διάφορες θέσεις. Ενώ η ίδια η μέτρηση μπορεί να πραγματοποιηθεί αρκετά γρήγορα, υπάρχει σημαντικό κόστος στον χρόνο τοποθέτησης, καθώς πρέπει να πραγματοποιηθούν έως και εκατοντάδες χιλιάδες θέσεις για ορισμένα πειράματα. Η τοποθέτηση περιλαμβάνει κίνηση τεσσάρων κινητήρων. Ο χρόνος που απαιτείται για τη μετάβαση από τη μια θέση στην άλλη μπορεί να υπολογιστεί με μεγάλη ακρίβεια. Το αποτέλεσμα του πειράματος δεν εξαρτάται από τη σειρά με την οποία γίνονται οι μετρήσεις στις διάφορες θέσεις. Ωστόσο, ο συνολικός χρόνος που απαιτείται για το πείραμα εξαρτάται από τη σειρά. Επομένως, το πρόβλημα

συνίσταται στην εύρεση μιας ακολουθίας που ελαχιστοποιεί τον συνολικό χρόνο τοποθέτησης. Αυτό οδηγεί σε πρόβλημα περιοδεύοντος πωλητή.

1.3.4 Καλωδίωση υπολογιστή

Οι Lenstra & Rinnooy Kan, (1974) [2] ανέφεραν μια ειδική περίπτωση σύνδεσης εξαρτημάτων σε μια πλακέτα υπολογιστή. Οι μονάδες βρίσκονται σε μια πλακέτα υπολογιστή και ένα δεδομένο υποσύνολο ακίδων πρέπει να συνδεθεί. Σε αντίθεση με τη συνηθισμένη περίπτωση όπου είναι επιθυμητή μια σύνδεση δέντρου Steiner, εδώ η απαίτηση είναι να μην συνδέονται περισσότερα από δύο καλώδια σε κάθε πείρο. Ως εκ τούτου, έχουμε το πρόβλημα να βρούμε μια συντομότερη διαδρομή με απροσδιόριστα σημεία έναρξης και τερματισμού. Παρόμοια κατάσταση συμβαίνει για τη λεγόμενη καλωδίωση testbus. Για να δοκιμάσετε την κατασκευασμένη πλακέτα, πρέπει να πραγματοποιήσετε μια σύνδεση που εισέρχεται στην πλακέτα σε κάποιο καθορισμένο σημείο, διατρέχει όλες τις μονάδες και τερματίζει σε κάποιο καθορισμένο σημείο. Για κάθε ενότητα έχουμε επίσης ένα καθορισμένο σημείο εισόδου και εξόδου για αυτήν τη δοκιμαστική καλωδίωση. Αυτό το πρόβλημα ισοδυναμεί επίσης με την επίλυση ενός προβλήματος Χαμιλτονιανής διαδρομής με τη διαφορά ότι οι αποστάσεις δεν είναι συμμετρικές και ότι το σημείο έναρξης και λήξης καθορίζεται.

1.3.5 Το πρόβλημα της επιλογής παραγγελιών στις αποθήκες

Αυτό το πρόβλημα σχετίζεται με το χειρισμό υλικών σε μια αποθήκη (Ratliff & Rosenthal, 1983) [3]. Ας υποθέσουμε ότι σε μια αποθήκη φτάνει μια παραγγελία για ένα συγκεκριμένο υποσύνολο των ειδών που είναι αποθηκευμένα στην αποθήκη. Κάποιο όχημα πρέπει να συλλέξει όλα τα αντικείμενα αυτής της παραγγελίας για να τα στείλει στον πελάτη. Η σχέση με το TSP φαίνεται αμέσως. Οι θέσεις αποθήκευσης των στοιχείων αντιστοιχούν στους κόμβους του γραφήματος. Η απόσταση μεταξύ δύο κόμβων δίνεται από το χρόνο που απαιτείται για τη μετακίνηση του οχήματος από τη μια θέση στην άλλη. Το πρόβλημα της εύρεσης μιας συντομότερης διαδρομής για το όχημα με ελάχιστο χρόνο παραλαβής μπορεί

πλέον να λυθεί ως TSP. Σε ειδικές περιπτώσεις αυτό το πρόβλημα μπορεί να λυθεί εύκολα, βλ. (van Dal, 1992) για εκτενή συζήτηση και για αναφορές.

1.3.6 Δρομολόγηση οχημάτων

Ας υποθέσουμε ότι σε μια πόλη n ταχυδρομικές θυρίδες πρέπει να αδειάζονται κάθε μέρα μέσα σε ένα συγκεκριμένο χρονικό διάστημα, ας πούμε στις 14:00. Το πρόβλημα είναι να βρεθεί ο ελάχιστος αριθμός φορτηγών για να γίνει αυτό και ο συντομότερος χρόνος για να γίνουν οι συλλογές χρησιμοποιώντας αυτόν τον αριθμό φορτηγών. Ως άλλο παράδειγμα, ας υποθέσουμε ότι n πελάτες απαιτούν ορισμένες ποσότητες ορισμένων εμπορευμάτων και ένας προμηθευτής πρέπει να ικανοποιήσει όλες τις απαιτήσεις με ένα στόλο φορτηγών. Το πρόβλημα είναι να βρεθεί μια ανάθεση πελατών στα φορτηγά και ένα χρονοδιάγραμμα παράδοσης για κάθε φορτηγό, ώστε να μην ξεπεραστεί η χωρητικότητα κάθε φορτηγού και να ελαχιστοποιηθεί η συνολική απόσταση διαδρομής. Πολλές παραλλαγές αυτών των δύο προβλημάτων, όπου οι περιορισμοί χρόνου και χωρητικότητας συνδυάζονται, είναι κοινές σε πολλές εφαρμογές του πραγματικού κόσμου. Αυτό το πρόβλημα είναι επιλύσιμο ως TSP εάν δεν υπάρχουν περιορισμοί χρόνου και χωρητικότητας και εάν ο αριθμός των φορτηγών είναι σταθερός (ας πούμε m). Σε αυτή την περίπτωση λαμβάνουμε ένα πρόβλημα m – πωλητών. Ωστόσο, μπορεί κανείς να εφαρμόσει μεθόδους για το TSP για να βρει καλές εφικτές λύσεις για αυτό το πρόβλημα (βλ. Lenstra & Rinnooy Kan, 1974).

1.3.7 Σχεδίαση μάσκας στην παραγωγή PCB

Για την παραγωγή κάθε στρώσης πλακέτας τυπωμένου κυκλώματος, καθώς και για στρώματα ολοκληρωμένων συσκευών ημιαγωγών, πρέπει να παραχθεί μια φωτογραφική μάσκα. Στην περίπτωσή μας για πλακέτες τυπωμένων κυκλωμάτων αυτό γίνεται με μηχανική διάταξη σχεδίασης. Το plotter μετακινεί έναν φακό πάνω από μια φωτοευαίσθητη επικαλυμμένη γυάλινη πλάκα. Το κλείστρο μπορεί να ανοίξει ή να κλείσει για να εκτεθούν συγκεκριμένα μέρη της πλάκας. Υπάρχουν

διάφορα διαθέσιμα ανοίγματα για να είναι δυνατή η δημιουργία διαφορετικών δομών στην πλακέτα. Πρέπει να ληφθούν υπόψη δύο τύποι δομών. Μια γραμμή εκτίθεται στην πλάκα μετακινώντας το κλειστό κλείστρο σε ένα τελικό σημείο της γραμμής, στη συνέχεια ανοίγοντας το κλείστρο και μετακινώντας το στο άλλο τελικό σημείο της γραμμής. Τότε το κλείστρο κλείνει. Μια δομή τύπου σημείου δημιουργείται μετακινώντας (με το κατάλληλο διάφραγμα) στη θέση αυτού του σημείου, ανοίγοντας το κλείστρο μόνο για να απελευθερώσει ένα σύντομο φλας και μετά κλείνοντάς το ξανά. Η ακριβής μοντελοποίηση του προβλήματος ελέγχου του plotter οδηγεί σε ένα πρόβλημα πιο περίπλοκο από το TSP και επίσης πιο περίπλοκο από το πρόβλημα του αγροτικού ταχυδρόμου. Μια εφαρμογή πραγματικού κόσμου στο πραγματικό περιβάλλον παραγωγής αναφέρεται στο (Grötschel et al., 1991).

1.4 Σύγκριση Αλγορίθμων για την Επίλυση Προβλήματος Περιοδευόντος Πωλητή

Το πρόβλημα του περιοδευόντος πωλητή (Travel Salesman Problem (TSP)) διατυπώθηκε για πρώτη φορά το 1930 και θεωρείται ένα δύσκολο μη ντετερμινιστικό πολυωνυμικό πρόβλημα (Non-Deterministic Polynomial-time hard (NP-hard)). Για την επίλυσή του υπάρχουν και ακριβείς και κατά προσέγγιση λύσεις. Οι ακριβείς λύσεις του TSP συνεπάγονται δοκιμή όλων των συνδυασμών μετάθεσης με πολυπλοκότητα $O(n!)$. Επομένως, ακόμη και με μικρό αριθμό πόλεων ($n=10$) οι συνδυασμοί λειτουργίας είναι 3628800, κάτι που δεν είναι πρακτικό. Η κατά προσέγγιση λύση, από την άλλη πλευρά, χρησιμοποιεί ευρετικούς αλγόριθμους για να βρει μια προσέγγιση στη βέλτιστη λύση σε μικρότερο αριθμό βημάτων και ως εκ τούτου μειωμένη πολυπλοκότητα. Η ευρετική λύση μπορεί να μην είναι η βέλτιστη (συντομότερη διαδρομή και ελάχιστο κόστος), αλλά εξακολουθεί να θεωρείται πολύτιμη καθώς απαιτεί μικρότερο χρόνο σε σύγκριση με τη βέλτιστη. Οι ευρετικοί αλγόριθμοι μπορούν επίσης να χρησιμοποιηθούν σε συνδυασμό με άλλους αλγόριθμους βελτιστοποίησης για τη βελτιστοποίηση της λύσης.

Υπάρχουν πολλοί αλγόριθμοι που χρησιμοποιούνται στον δυναμικό προγραμματισμό για την επίλυση του TSP. Δεδομένου ότι η εύρεση της βέλτιστης λύσης μπορεί να μην είναι εφικτή (μεγάλος αριθμός πόλεων), το κάτω όριο Held-Karp χρησιμοποιείται για την αξιολόγηση της απόδοσης ενός δεδομένου αλγορίθμου. Δηλαδή, το όριο Held-Karp καθορίζει πόσο κοντά μια δεδομένη λύση (χρησιμοποιώντας έναν δεδομένο αλγόριθμο) είναι στη βέλτιστη λύση. Το όριο Held-Karp είναι η χαλαρή λύση στον γραμμικό προγραμματισμό του TSP. Συνήθως, το κάτω όριο του Held-Karp είναι περίπου 0,8% κάτω από τη βέλτιστη λύση.

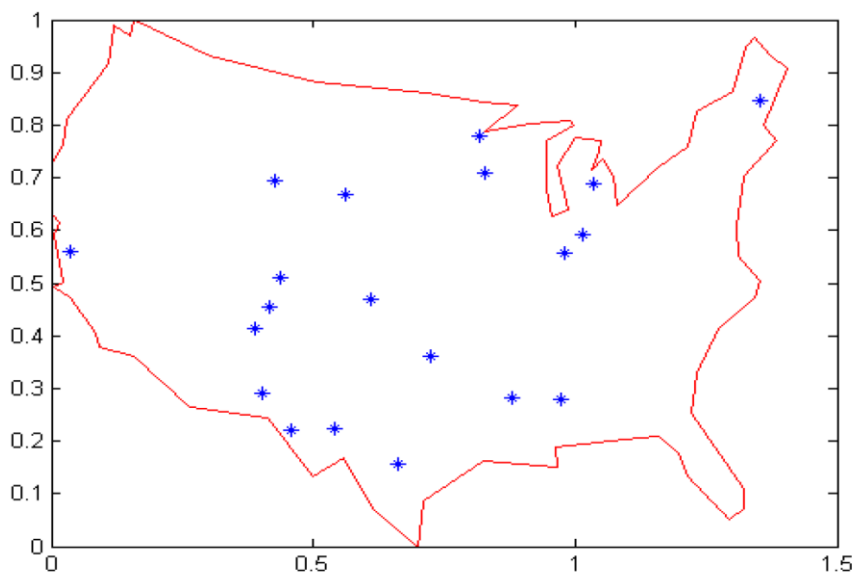
1.4.1 Μέθοδος πλησιέστερου γειτονικού κόμβου

Είναι ο απλούστερος ευρετικός αλγόριθμος που χρησιμοποιείται για την επίλυση του TSP. Ο αλγόριθμος μπορεί να συνοψιστεί ως εξής:

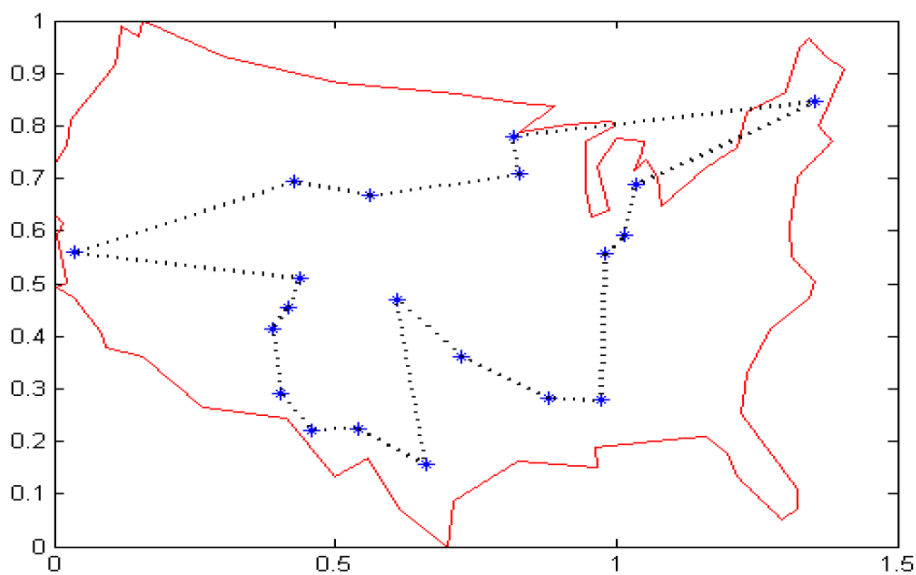
- 1) Επιλέξτε μια τυχαία πόλη n και θεώρησε πως είναι η πόλη εκκίνησης n_0 .
- 2) Βρείτε την πλησιέστερη πόλη χωρίς επίσκεψη και πηγαίνετε εκεί
- 3) Επιστημάνετε την τρέχουσα πόλη ως επίσκεψη
- 4) Υπάρχουν πόλεις που δεν έχουν επισκεφτεί; Εάν ναι, μεταβείτε στο (2)
- 5) Επιστροφή στην πόλη εκκίνησης

Για την αξιολόγηση της λύσης αλγορίθμου πλησιέστερου γειτονικού κόμβου έστω ότι εξετάζεται ο χάρτης των Ηνωμένων Πολιτειών. Η έκτασή του είναι 9,9 εκατομμύρια χιλιόμετρα. Η ορθογώνια περιοχή έχει πλάτος 4313 km και μήκος 2545 km. Οι πόλεις κατανέμονται τυχαία σε αυτήν την περιοχή. Το πρώτο παράδειγμα TSP φαίνεται στο Σχ. 1.1. Υπάρχουν 20 πόλεις (κόμβοι) που κατανέμονται τυχαία σε περιοχή 4313 επί 2545 Km. Ο στόχος είναι να υπολογίσετε τη βέλτιστη διαδρομή για να επισκεφθείτε κάθε πόλη μία φορά και να επιστρέψετε στο σημείο εκκίνησης. Αξίζει να αναφέρουμε ότι αυτό το παράδειγμα είναι ένα συμμετρικό TSP, καθώς το κόστος ταξιδιού από την πόλη A στην πόλη B είναι ακριβώς το ίδιο με αυτό από το B στην A. Το ασύμμετρο TSP λαμβάνει υπόψη την ανισότητα του κόστους ταξιδιού και από τις δύο πόλεις,. Η βέλτιστη λύση φαίνεται στο Σχ. 1.2 με συνολικό συνδυασμένο μήκος 4616 Km.

Αυτή η λύση έχει τη μεγαλύτερη πολυπλοκότητα και τον μεγαλύτερο αριθμό επαναλήψεων, ωστόσο βρίσκει τη συντομότερη διαδρομή.



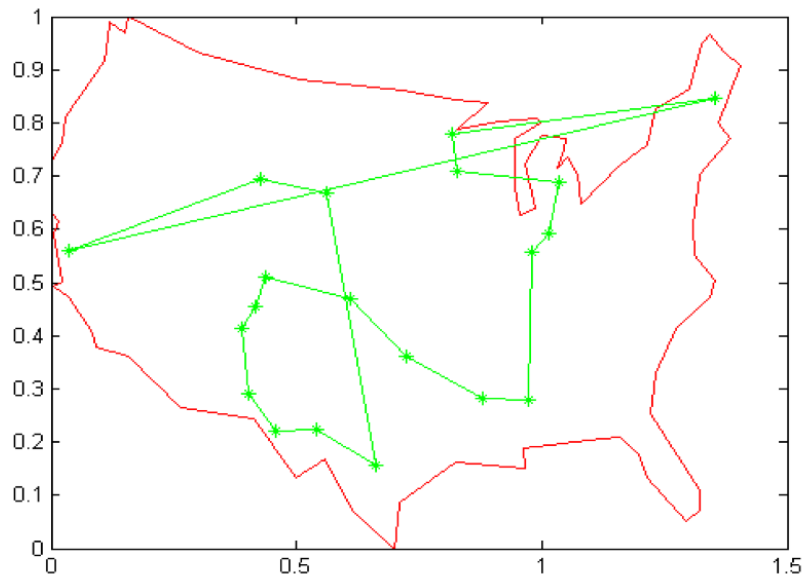
Εικ. 1.1 Παράδειγμα TSP 20 πόλεων [5]



Εικ. 1.2. Παράδειγμα TSP 20 πόλεων: Βέλτιστη λύση [5]

Επιλύοντας το ίδιο παράδειγμα με τον αλγόριθμο του πλησιέστερου γειτονικού κόμβου, λαμβάνουμε τη διαδρομή που φαίνεται στο Σχ. 1.3. Η λύση έχει μεγαλύτερο συνδυασμένο μήκος (15800 Km) αλλά βρίσκει λύση σε επαναλήψεις $O(N^2 \log_2(N))$, όπου N είναι ο αριθμός των πόλεων που πρέπει να επισκεφτείτε. Ο πλησιέστερος

γειτονικός κόμβος διατηρεί την λύση εντός του 25% του κατώτερου ορίου Held-Karp.

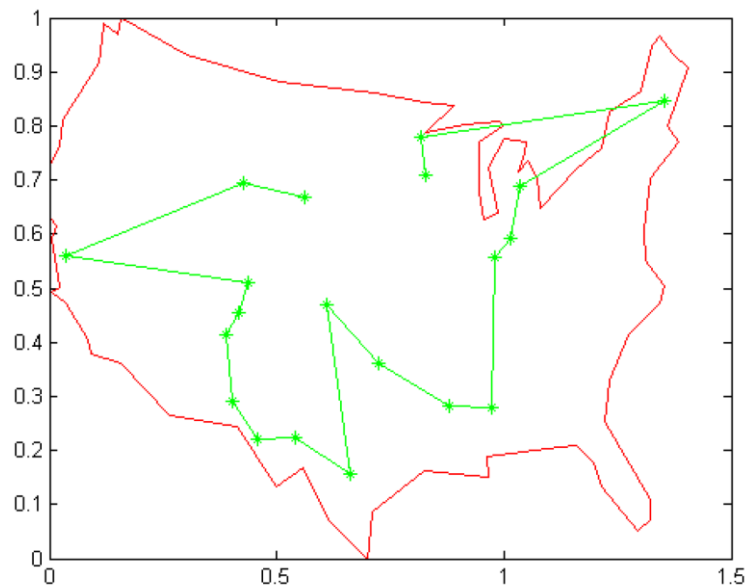


Εικ. 3. Παράδειγμα TSP 20 Πόλεων: Πλησιέστερος γειτονικός κόμβος [5]

1.4.2 Γενετικός αλγόριθμος

Αυτός ο αλγόριθμος είναι πολύ γνωστός για την επίλυση πολύπλοκων προβλημάτων, όπου η βέλτιστη λύση είναι δύσκολο να βρεθεί. Μοιάζει με τη διαδικασία της φυσικής επιλογής. Ο αλγόριθμος υπολογίζει τη συνάρτηση καταλληλότητας για κάθε μέλος του πληθυσμού. Στη συνέχεια δημιουργεί νέα άτομα του πληθυσμού. Χρησιμοποιεί μετάλλαξη για να προσθέσει τυχαιοποίηση στη διαδικασία, παρόμοια με αυτή του φυσικού γονιδιώματος. Τέλος, επιλέγει το άτομο (λύση) με την υψηλότερη λειτουργία φυσικής κατάστασης. Η εφαρμογή γενετικού αλγορίθμου στο TSP απαιτεί ορισμένους περιορισμούς. Για παράδειγμα, σε κάθε διαδρομή, κάθε πόλη δεν πρέπει να επαναλαμβάνεται, διαφορετικά εμφανίζονται βρόχοι. Επιπλέον, στον αλγόριθμο λαμβάνονται υπόψη μόνο έγκυρες διαδρομές. Για παράδειγμα, μια πόλη που βρίσκεται στα δυτικά δεν μπορεί να θεωρηθεί στο τμήμα της διαδρομής που βρίσκεται στην Άπω Ανατολή. Η λύση TSP για 20 πόλεις που χρησιμοποιούν Γενετικό Αλγόριθμο φαίνεται στο Σχ. 1.4. Με μεγαλύτερο αριθμό επαναλήψεων από την μέθοδο του πλησιέστερου γειτονικού

κόμβου, ο Γενετικός Αλγόριθμος μπορεί να βρει μια συντομότερη διαδρομή (11900 km).

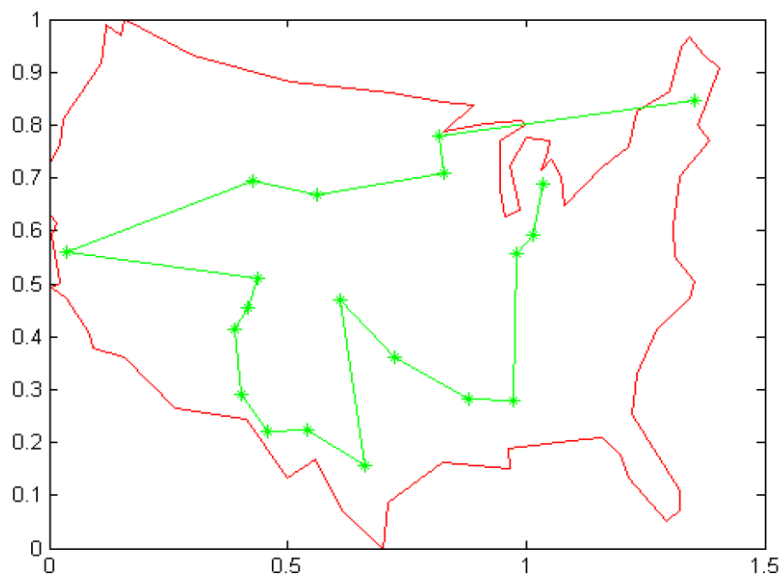


Εικ. 4. Παράδειγμα TSP 20 Πόλεων: Γενετικός Αλγόριθμος [5]

1.4.3 Άπληστος Ευρετικός Αλγόριθμος

Αυτός ο αλγόριθμος ανήκει στην κατηγορία των ευρετικών αλγορίθμων, που αναζητά το τοπικό βέλτιστο και βελτιστοποιεί την τοπική βέλτιστη λύση για να βρει το συνολικό βέλτιστο. Ξεκινά ταξινομώντας όλες τις άκρες και στη συνέχεια επιλέγει την άκρη με το ελάχιστο κόστος. Συνεχίζει να επιλέγει τις καλύτερες επόμενες επιλογές, δεδομένου ότι δεν σχηματίζονται βρόχοι. Η υπολογιστική πολυπλοκότητα του άπληστου αλγορίθμου είναι $O(N^2 \log_2(N))$ και δεν υπάρχει καμία εγγύηση ότι θα βρεθεί μια συνολική βέλτιστη λύση. Από την άλλη πλευρά, ο άπληστος αλγόριθμος τερματίζεται σε έναν εύλογο αριθμό βημάτων και διατηρεί τη λύση εντός 15 - 20 % του ορίου Held-Karp. Λαμβάνεται υπόψη το ίδιο παράδειγμα TSP και η περιήγηση υπολογίζεται χρησιμοποιώντας τον συγκεκριμένο αλγόριθμο όπως φαίνεται στο Σχ. 1.5 με συνδυασμένο συνολικό μήκος 12900 Km. Αν και αυτή η λύση έχει οριακά μειωμένη συνολική απόσταση σε σύγκριση με τον αλγόριθμο του πλησιέστερου γειτονικού κόμβου, έχει ακόμη μεγαλύτερη πολυπλοκότητα και χρόνο εκτέλεσης.

Αξίζει να σημειωθεί ότι όλοι οι αλγόριθμοι συνέκλιναν, λόγω του μικρού αριθμού πόλεων και ως εκ τούτου του περιορισμένου αριθμού πιθανοτήτων διαδρομής. Επομένως, όλα αυτά συγκλίνουν σε μια έγκυρη λύση και με τον όρο «έγκυρο» εννοούμε όχι πολύ μεγάλη διαδρομή.



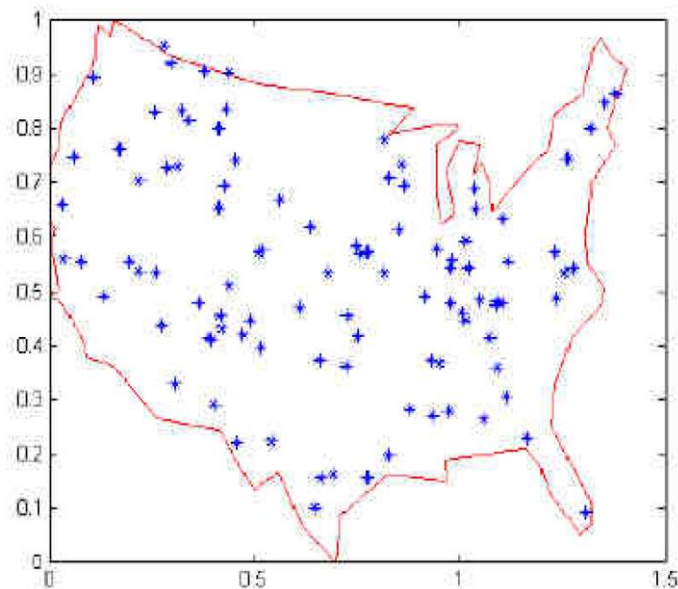
Εικ. 1.5. Παράδειγμα TSP 20 Πόλεων: Άπληστος Ευρετικός Αλγόριθμος [5]

1.5 Πρόβλημα TSP με μεγαλύτερο αριθμό κόμβων

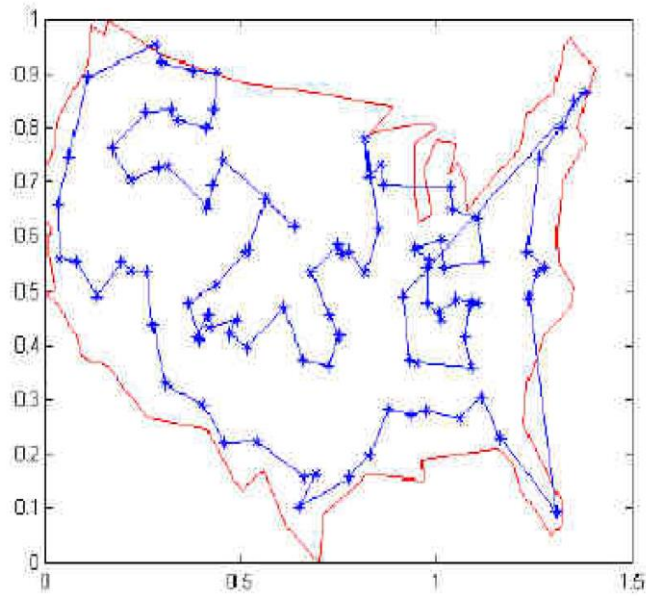
Για την προσέγγιση ρεαλιστικών μοντέλων, εξετάζεται μεγαλύτερος αριθμός πόλεων και συγκρίνονται ο αντίστοιχος χρόνος εκτέλεσης και τα μήκη διαδρομής. Για παράδειγμα, λαμβάνονται υπόψη 100 τυχαία καταναμημένες πόλεις εντός των συνόρων των ΗΠΑ, όπως φαίνεται στο Σχ. 1.6. Τα αποτελέσματα με χρήση των τριών αλγορίθμων και των βέλτιστων υπολογισμένων διαδρομών φαίνονται στο Σχ. 1.7, Σχ. 1.8 και Σχ. 1.9, αντίστοιχα. Ο Πίνακας 1.1 απεικονίζει την ομοιότητα μεταξύ γενετικών και άπληστων ευρετικών αλγορίθμων όσον αφορά το βέλτιστο μήκος διαδρομής.

Η διαδρομή λύσης του άπληστου ευρετικού αλγορίθμου είναι μικρότερη κατά 2168 km. Βρίσκει επίσης την ταχύτερη λύση στο πρόβλημα (0,18 δευτερόλεπτα). Από την άλλη πλευρά, του πλησιέστερου γειτονικού κόμβου προσεγγίζει σχετικά μεγαλύτερη διαδρομή (26.664 km), αλλά καταναλώνει λιγότερες επαναλήψεις από τους υπόλοιπους αλγόριθμους. Αυτό οφείλεται στην επαναληπτική φύση του αλγορίθμου. Ο γενετικός αλγόριθμος δεν εγγυάται ότι θα βρει το συντομότερο μονοπάτι, αν και το προσεγγίζει. Σε αυτό το παράδειγμα, πάλι και οι τρεις αλγόριθμοι συγκλίνουν.

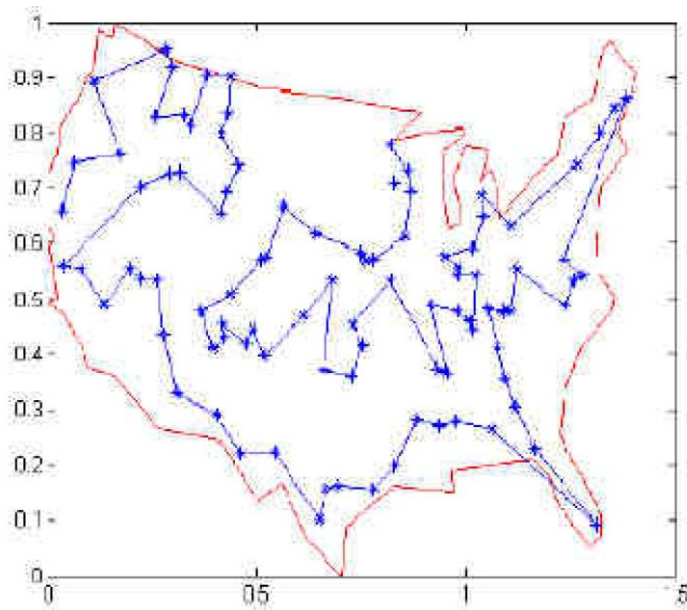
Για να διαπιστωθεί η διαφορά μεταξύ των αποτελεσμάτων των αλγορίθμων πιο ξεκάθαρα, στην μελέτη [5] εξετάστηκε ένα δεύτερο παράδειγμα 1000 τυχαία κατανομημένων πόλεων. Μετά την προσομοίωση, προκύπτει ο Πίνακας 1.2 Ο άπληστος ευρετικός αλγόριθμος είναι και πάλι ο νικητής του συντομότερη διαδρομή μήκους 72801 km. Η διαδρομή λύσης του πλησιέστερου γειτονικού κόμβου είναι μεγαλύτερη κατά 11.137 km, αλλά έχει μικρότερο χρόνο υπολογισμού. Από την άλλη πλευρά, ο γενετικός αλγόριθμος δεν έχει καμία εγγύηση για την εύρεση της βέλτιστης λύσης και ως εκ τούτου η διαδρομή του είναι η μεγαλύτερη (282866). Ως εκ τούτου, ο γενετικός αλγόριθμος δεν συγκλίνει.



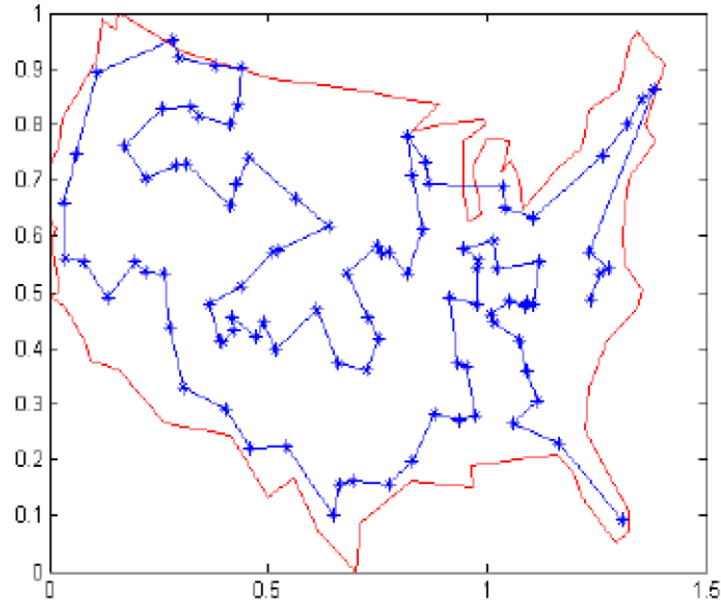
Εικ. 1.6. 100 τυχαία κατανομημένες πόλεις. [5]



Εικ. 1.7. Λύση πλησιέστερου γειτονικού κόμβου σε πρόβλημα 100 πόλεων [5]



Εικ. 1.8. Λύση γενετικού αλγορίθμου για το πρόβλημα 100 πόλεων. [5]



Εικ. 9. Λύση ευρετικού αλγορίθμου για το πρόβλημα 100 πόλεων. [5]

Πίνακας 1.1 Σύγκριση αλγορίθμων TSP 100 πόλεων [5]

Αλγόριθμος	Βέλτιστο μήκος διαδρομής (χλμ)	Χρόνος που έχει παρέλθει μέχρι ολοκλήρωση (δευτ.)	Επαναλήψεις
Πλησιέστερου γειτονικού κόμβου	26664	2.5	100
Γενετικός	25479	45	10000
Άπληστος Γενετικός	23311	0.07	18

Πίνακας 1.2 Σύγκριση αλγορίθμων TSP 1000 πόλεων[5]

Αλγόριθμος	Βέλτιστο μήκος διαδρομής (χλμ)	Χρόνος που έχει παρέλθει μέχρι ολοκλήρωση (δευτ.)	Επαναλήψεις
Πλησιέστερου γειτονικού κόμβου	83938	95.5	1000
Γενετικός	282866	468	10000
Άπληστος Γενετικός	72801	127	151

1.6 Πρόβλημα περιοδεύοντος πωλητή με χρονικά παράθυρα

Το πρόβλημα του περιοδεύοντος πωλητή (TSP) είναι ένα κλασικό πρόβλημα συνδυαστικής βελτιστοποίησης. Έχει μελετηθεί από ερευνητές που εργάζονται σε διάφορους τομείς, συμπεριλαμβανομένων των μαθηματικών, της επιστήμης των υπολογιστών και της επιχειρησιακής έρευνας. Στο TSP, ένας πωλητής φεύγει από την πόλη του, πρέπει να επισκεφτεί μια σειρά από άλλες πόλεις και μετά να επιστρέψει στο σπίτι. Ο στόχος είναι να ελαχιστοποιηθεί το συνολικό κόστος μετακίνησης μεταξύ πόλεων, δεδομένου του κόστους για ταξίδια μεταξύ κάθε ζεύγους πόλεων.

Σε ένα πλαίσιο logistics, ο πωλητής αντικαθίσταται από ένα όχημα, η πόλη καταγωγής από μια αποθήκη και οι πόλεις από πελάτες. Η εισαγωγή του χρόνου ταξιδιού μεταξύ των ζευγών τοποθεσιών πελατών και ενός χρονικού παραθύρου για κάθε πελάτη, το οποίο απαιτεί από το όχημα να επισκεφτεί τον πελάτη κατά τη διάρκεια μιας προκαθορισμένης χρονικής περιόδου, δίνει το Πρόβλημα του περιοδεύοντος πωλητή με παράθυρα χρόνου (Traveling Salesman Problem with Time Windows - TSPTW). Το TSPTW είναι πιο σχετικό με τα logistics, καθώς οι πελάτες συχνά ζητούν, ή αναφέρονται, ένα παράθυρο χρόνου παράδοσης.

1.7 Σκοπός εργασίας

Σκοπός της παρούσας διπλωματικής εργασίας είναι η συγκριτική μελέτη αλγορίθμων για την επίλυση προβλήματος πλανόδιου πωλητή με χρονικά παράθυρα. Αρχικά στο πρώτο κεφάλαιο γίνεται ιστορική αναδρομή του προβλήματος του περιοδεύοντος πωλητή και αναφοράς στους τρόπους επίλυσης. Ακολούθως στο επόμενο κεφάλαιο ορίζεται το πώς μεταβάλλεται το ανωτέρω πρόβλημα με την εισαγωγή της έννοιας των χρονικών παραθύρων σε αυτό. Το μαθηματικό μοντέλο του προβλήματος αναπτύσσεται τόσο για το συμμετρικό όσο και το ασύμμετρο πρόβλημα. Στο επόμενο κεφάλαιο επιλέγονται τέσσερις μέθοδοι επίλυσης του

προβλήματος από την βιβλιογραφία και αναπτύσσεται το μαθηματικό μοντέλο που τις διέπει καθώς επίσης και τα αποτελέσματα τις εκάστης μεθόδου. Αυτές οι μέθοδοι είναι οι εξής: Επίλυση του προβλήματος του περιοδεύοντος πωλητή με χρονικά παράθυρα α) με χρήση λαγκραντζιανής χαλάρωσης β) με χρήση χρονικά διευρυμένων δικτύων γ) με χρήση υβριδικών δικτύων δεικτών με χρονικά χαρακτηριστικά και τέλος δ) με χρήση δυναμικής διακριτοποίησης.

Κεφάλαιο 2^ο

2.1 Πρόβλημα περιοδεύοντος πωλητή με χρονικά παράθυρα

Το πρόβλημα του περιοδεύοντος πωλητή είναι το πρόβλημα εύρεσης της συντομότερης διαδρομής για ένα άτομο το οποίο πρέπει να ταξιδέψει από την αρχή σε n αριθμό πόλεων. Ο πωλητής θα πρέπει να επισκεφθεί μόνο μία φορά την κάθε πόλη και στη συνέχεια να επιστρέψει στην αρχή. Το πρόβλημα αυτό είναι ένα πρόβλημα γραμμικού προγραμματισμού το οποίο είναι εύκολο να δηλωθεί αλλά δύσκολο να λυθεί. Η πολυπλοκότητα του προβλήματος είναι $O(n!)$ για την εύρεση της περιήγησης για n αριθμό πόλεων. Σημαίνει ότι όσο αυξάνεται ο αριθμός των πόλεων, το πρόβλημα γίνεται όλο και πιο δύσκολο να λυθεί.

Η εισαγωγή του χρόνου ταξιδιού μεταξύ των ζευγών τοποθεσιών πελατών και ενός χρονικού παραθύρου για κάθε πελάτη, μεταβάλλει το αρχικό μας πρόβλημα δίνοντας πλέον το Πρόβλημα του περιοδεύοντος πωλητή με παράθυρα χρόνου (Traveling Salesman Problem with Time Windows - TSPTW). Το TSPTW είναι ακόμη πιο σχετικό με τα logistics, καθώς οι πελάτες συχνά ζητούν, ή αναφέρονται σε ένα συγκεκριμένο παράθυρο χρόνου παράδοσης. Αντιμετωπίζεται καθημερινά από χιλιάδες εταιρείες και οργανισμούς που ασχολούνται με την παράδοση και τη συλλογή αγαθών ή ανθρώπων. Επειδή οι συνθήκες ποικίλλουν από τη μία περίπτωση στην άλλη, οι στόχοι και οι περιορισμοί που συναντώνται στην πράξη είναι μεταβλητοί με μεγάλο εύρος. Οι περισσότερες αλγοριθμικές έρευνες και ανάπτυξη λογισμικού σε αυτόν τον τομέα επικεντρώνονται σε περιορισμένο αριθμό πρωτοτύπων προβλημάτων. Χτίζοντας αρκετή ευελιξία στα συστήματα βελτιστοποίησης μπορεί κανείς να τα προσαρμόσει σε διάφορα πρακτικά πλαίσια.

Στο Πρόβλημα περιοδεύοντος πωλητή με τα παράθυρα χρόνου (TSPTW), ο ταξιδιώτης πρέπει να επισκεφτεί κάθε πελάτη εντός ενός χρονικού διαστήματος, που ορίζεται ως το χρονικό παράθυρο που υποδεικνύει το νωρίτερο και το αργότερο χρονικό διάστημα στο οποίο αυτός ο πελάτης θα επιτρέψει την έναρξη της διαδρομής.

Το TSPTW μπορεί να προκύψει σε παραδείγματα logistic διανομής, όπως τραπεζικές παραδόσεις, ταχυδρομικές παραδόσεις, συλλογή βιομηχανικών απορριμμάτων, δρομολόγηση σχολικών λεωφορείων, μεταφορά ευπαθών αγαθών, προγραμματισμός καταστημάτων εργασίας, στρατιωτικές επιχειρήσεις κ.λπ., όπου δεν ισχύουν περιορισμοί χωρητικότητας.

Το TSPTW δεν έχει λάβει τόση προσοχή όσο το TSP ή το Πρόβλημα Δρομολόγησης Οχημάτων (VRP). Μια σύντομη βιβλιογραφική ανασκόπηση του προβλήματος TSPTW μπορεί να βρεθεί σε πρόσφατες εργασίες των Lopez-Ibanez και Blum (2010) [6] και Kara et al. (2013) [7]. Επίσης, ο Τσιτσικλής (1992) [9] μελέτησε ειδικές περιπτώσεις του TSPTW.

Η πρώτη προσπάθεια να βρεθεί μια βέλτιστη λύση στο TSPTW εμφανίστηκε σε άρθρο των Christofides et al. (1981) [9]. Ανέπτυξαν έναν αλγόριθμο διακλάδωσης και δέσμευσης χρησιμοποιώντας δυναμικό προγραμματισμό και λύνοντας προβλήματα που περιείχαν έως και 50 κόμβους μέχρι τη βέλτιστη. Οι Dumas et al. (1995) [10] επέκτειναν τις προηγούμενες προσεγγίσεις δυναμικού προγραμματισμού και έλυσαν περιπτώσεις έως και 200 κόμβους.

Ο Savelsbergh (1985) [11] έχει αποδείξει ότι ακόμη και η εύρεση μιας εφικτής λύσης για το TSPTW είναι πρόβλημα κλάσης NP-Complete. Ο περιορισμός του ακριβούς αλγορίθμου για την εύρεση μιας βέλτιστης λύσης και η πολυπλοκότητα του προβλήματος οδηγούν τους ερευνητές στην ανάπτυξη αποτελεσματικών ευρετικών. Πιο πρόσφατα, ορισμένες μετα-ευρετικές προσεγγίσεις Οι Lopez-Ibanez και Blum (2010) [6] έχουν χρησιμοποιήσει διαμορφώσεις προγραμματισμού ακέραιων αριθμών για να αποκτήσουν λύσεις υψηλής ποιότητας.

Ο στόχος του TSPTW μπορεί να είναι:

- (1) η ελαχιστοποίηση της συνολικής διάρκειας της περιήγησης (χρόνος ολοκλήρωσης της περιήγησης)
- (2) η ελαχιστοποίηση του αθροίσματος του χρόνου διέλευσης τόξων (ή του συνολικού κόστους που προέκυψε κατά τη διαδρομή των τόξων)

- (3) η ελαχιστοποίηση του αθροίσματος χρόνου διέλευσης τόξων (κόστος ταξιδιού) συν χρόνο αναμονής (κόστος αναμονής) λόγω της αναμονής μπροστά από τους κόμβους.

Η περίπτωση όπου κάποιος προσπαθεί να ελαχιστοποιήσει τη διάρκεια της περιόδου ονομάζεται επίσης «Makespan Problem with Time Windows». Όταν το μοναδιαίο κόστος ταξιδιού και το μοναδιαίο κόστος αναμονής φαίνονται ίσα, τότε μπορεί κανείς να λύσει το πρόβλημα απλώς ελαχιστοποιώντας τη διάρκεια της περιόδου. Ο Savelsbergh (1992) [11] όρισε για πρώτη φορά την παραλλαγή του TSPTW που ονομάζεται Πρόβλημα δρομολόγησης οχημάτων με το Time Windows ελαχιστοποιώντας τη διάρκεια της περιόδου. Δεν πρότεινε ένα μαθηματικό μοντέλο, αλλά μάλλον μια μέθοδο βελτίωσης ανταλλαγής ακμών για την επίλυση του προβλήματος. Υπάρχει μόνο μία διατύπωση για το συμμετρικό TSPTW που προτάθηκε από τον Baker (1983) [12] που ασχολείται με τη διάρκεια της περιόδου, αλλά είναι ένα μη γραμμικό μοντέλο προγραμματισμού. Ο Baker (1983) ανέπτυξε μια διαδικασία διακλάδωσης και δέσμευσης και έλυσε περιπτώσεις έως και 50 κόμβους με βέλτιστο τρόπο..

Ειδική περίπτωση του προβλήματος που εμφανίστηκε πρόσφατα στη βιβλιογραφία μπορεί να χρησιμοποιηθεί για την ελαχιστοποίηση της διάρκειας της διαδοχής (Kara et al., 2013) [7]. Η αξιοσημείωτη βελτίωση στις τεχνολογίες υλικού και λογισμικού και η ευρεία διαθεσιμότητα του εμπορικού λογισμικού βελτιστοποίησης θα μας επιτρέψουν να λύσουμε εύκολα μαθηματικά μοντέλα φιλικά προς τον χρήστη στο εγγύς μέλλον.

Επιπλέον, το αυξανόμενο ενδιαφέρον για το ηλεκτρονικό εμπόριο και τις διαδικτυακές εγκαταστάσεις θα απαιτήσει την επίλυση προβλημάτων δρομολόγησης πιο εύκολα και γρήγορα με τη χρήση οποιουδήποτε βελτιστοποιητή. Ο κύριος στόχος αυτής της εργασίας είναι να παρουσιάσει νέες συνθέσεις για την ελαχιστοποίηση της διάρκειας περιήγησης του συμμετρικού και ασύμμετρου TSPTW που μπορεί να χρησιμοποιηθεί για την επίλυση μικρού και μεσαίου μεγέθους TSPTW απευθείας με τη χρήση οποιουδήποτε βελτιστοποιητή.

2.2 Ανασκόπηση μελετών για προβλήματα TSP-TW

Οι Chi-Bin Cheng et al. [13] χρησιμοποίησαν την τεχνική βελτιστοποίησης «αποικίας μυρμηγκιών» για να λύσουν το TSPTW. Οι αποστάσεις μεταξύ των κόμβων των περιπτώσεων αναφοράς θεωρήθηκαν ως ο χρόνος ώστε να ληφθούν υπόψη τα χρονικά παράθυρα. Οι συγγραφείς έλαβαν τα αριθμητικά αποτελέσματα τόσο του ACS-TSPTW όσο και του ACS-Time. Αυτό το πρόβλημα έχει μια σειρά από σημαντικές πρακτικές εφαρμογές, συμπεριλαμβανομένου του προγραμματισμού και της δρομολόγησης. Το πρόβλημα θεωρείται κλάσης NP-complete και ως εκ τούτου οι παραδοσιακοί αλγόριθμοι βελτιστοποίησης είναι αναποτελεσματικοί όταν εφαρμόζονται για την επίλυση προβλημάτων TSPTW μεγαλύτερης κλίμακας. Κατά συνέπεια, η ανάπτυξη αλγορίθμων προσέγγισης έχει λάβει μεγάλη προσοχή τα τελευταία χρόνια. Η βελτιστοποίηση αποικίας μυρμηγκιών (Ant colony optimization algorithms - ACO), εμπνευσμένη από τη συμπεριφορά αναζήτησης τροφής πραγματικών μυρμηγκιών, είναι ένας από τους πιο ελκυστικούς αλγόριθμους προσέγγισης. Κατά συνέπεια, αυτή η μελέτη αναπτύσσει έναν τροποποιημένο αλγόριθμο μυρμηγκιών, που ονομάζεται ACS-TSPTW, βασισμένος στην τεχνική ACO για την επίλυση του TSPTW. Δύο τοπικά ευρετικά είναι ενσωματωμένα στον αλγόριθμο ACS-TSPTW για τη διαχείριση των περιορισμών του χρονικού παραθύρου του προβλήματος. Τα αριθμητικά αποτελέσματα που ελήφθησαν για μια σειρά περιπτώσεων προβλημάτων αναφοράς επιβεβαιώνουν ότι η απόδοση του ACS-TSPTW είναι συγκρίσιμη με εκείνη του ACS-Time, ενός υπάρχοντος σχήματος ACO για την επίλυση του προβλήματος TSPTW.

Οι Imdat Kara et al. [7] εξηγήσαν ότι κάθε πόλη (κόμβοι, πελάτες) πρέπει να επισκέπτεται μέσα σε ένα χρονικό παράθυρο που ορίζεται από την παλαιότερη και την πιο πρόσφατη ώρα. Στο TSPTW, ο ταξιδιώτης πρέπει να περιμένει σε μια πόλη εάν φτάσει νωρίς. Συνεπώς, οι χρόνοι αναμονής επηρεάζουν άμεσα τη διάρκεια μιας περιόδου. Θα ήταν χρήσιμο να αναπτυχθεί ένα νέο μοντέλο επιλύσιμο από οποιονδήποτε βελτιστοποιητή άμεσα.

Οι Jeffrey W. Ohlmann et al. [14] χρησιμοποίησαν την προσέγγιση συμπίεσμνης ανόπτησης για να λύσει το πρόβλημα του περιοδεύοντος πωλητή με τα χρονικά παράθυρα (TSPTW) ως περιορισμό. Αντί να χρησιμοποιηθεί η παραδοσιακή προσομοιωμένη ανόπτηση, η μέθοδος που χρησιμοποιήθηκε για την επίλυση του TSPTW ήταν η προσέγγιση μεταβλητής ποινής της συμπίεσμνης ανόπτησης. Η σύγκριση με το πρόβλημα αναφοράς έγινε σε κάθε βήμα. Για τις περισσότερες περιπτώσεις, οι λύσεις που ελήφθησαν ήταν σχεδόν βέλτιστες ακόμη και για προβλήματα με περιορισμούς.

Ο Jing-Quan Li [15] παρουσιάζει έναν αμφίδρομο αλγόριθμο διόρθωσης περιορισμένων πόρων για το πρόβλημα του περιοδεύοντος πωλητή με τα χρονικά παράθυρα, στα οποία ο στόχος είναι να ελαχιστοποιηθούν οι χρόνοι ταξιδιού. Οι επεκτάσεις ετικετών και η κυριαρχία τους ξεκινούν ταυτόχρονα τόσο προς τις εμπρός όσο και προς τα πίσω: την κατεύθυνση προς τα εμπρός από την αρχική αποθήκη και την προς τα πίσω κατεύθυνση από την αποθήκη τερματισμού. Η προκύπτουσα διαδικασία επέκτασης ετικέτας σαρώνει πολύ μικρότερο χώρο από ό,τι στον δυναμικό προγραμματισμό μιας κατεύθυνσης, μειώνοντας σημαντικά τον αριθμό των ετικετών που δεν κυριαρχούν. Οι ετικέτες τόσο για την κατεύθυνση προς τα εμπρός όσο και για την κατεύθυνση προς τα πίσω ενώνονται τελικά για να σχηματίσουν μια πλήρη διαδρομή εάν πληρούνται όλες οι σχετικές προϋποθέσεις σκοπιμότητας.

Ο I. Τσιτσικλής [8] θεώρησε ένα πλήρες κατευθυνόμενο γράφημα στο οποίο κάθε τόξο έχει ένα δεδομένο μήκος. Υπάρχει ένα σύνολο εργασιών, κάθε εργασία που βρίσκεται σε κάποιον κόμβο του γραφήματος, με σχετικό χρόνο επεξεργασίας h_i και των οποίων η εκτέλεση πρέπει να ξεκινήσει μέσα σε ένα προκαθορισμένο χρονικό παράθυρο $[r_i, d_i]$. Υπάρχει ένας ενιαίος διακομιστής που μπορεί να κινηθεί στα τόξα του γραφήματος, με ταχύτητα μονάδας, και που πρέπει να εκτελέσει όλες τις εργασίες εντός των αντίστοιχων χρονικών παραθύρων τους.

Σε αυτήν την μελέτη λαμβάνονται υπόψη τα ακόλουθα δύο προβλήματα:

- (α) ελαχιστοποίηση του χρόνου κατά τον οποίο εκτελούνται όλες οι εργασίες (πρόβλημα περιοδεύοντος πωλητή) και

(β) ελαχιστοποίηση του αθροίσματος των χρόνων αναμονής των εργασιών (πρόβλημα ταξιδιωτικού επισκευαστή).

Η εστίαση είναι στις ακόλουθες δύο ειδικές περιπτώσεις:

(α) Οι εργασίες βρίσκονται σε μια γραμμή και

(β) ο αριθμός των κόμβων του γραφήματος οριοθετείται από κάποια ακέραια σταθερά B .

Επιπλέον, εξετάζουν λεπτομερώς τις ειδικές περιπτώσεις όπου (α) όλοι οι χρόνοι επεξεργασίας είναι 0, (β) όλοι οι χρόνοι έκδοσης τ_i είναι 0 και (γ) όλες οι προθεσμίες d_i είναι άπειρες.

Για πολλούς από τους συνδυασμούς προβλημάτων που προκύπτουν, διευθετούν την πολυπλοκότητά τους είτε καθιερώνοντας NP-πληρότητα είτε παρουσιάζοντας πολυωνυμικούς (ή ψευδοπολυωνυμικούς) αλγόριθμους χρόνου. Τέλος, οι συγγραφείς εξάγουν αλγόριθμους για την περίπτωση όπου, για οποιαδήποτε χρονική στιγμή t , ο αριθμός των εργασιών που μπορούν να εκτελεστούν εκείνη τη στιγμή είναι περιορισμένος.

Οι Kjetil Fagerholt et al. [16] πρότειναν έναν αλγόριθμο κατανομής και προτεραιότητας για το TSP μαζί με τους περιορισμούς χρονικού παραθύρου. Η εφαρμογή του αλγορίθμου χρησιμοποιήθηκε στον προγραμματισμό δρομολογίων πλοίων. Το TSP-ATWPC εμφανίζεται ως υποπρόβλημα της βέλτιστης αλληλουχίας ενός δεδομένου συνόλου επισκέψεων λιμένων σε ένα πραγματικό πρόβλημα προγραμματισμού πλοίων χύδην φορτίου, το οποίο είναι ένα συνδυασμένο πρόβλημα παραλαβής και παράδοσης πολλών πλοίων με χρονικά παράθυρα και πρόβλημα πολλαπλής κατανομής. Κάθε πλοίο του στόλου είναι εξοπλισμένο με ένα ευέλικτο αμπάρι που μπορεί να χωριστεί σε πολλά μικρότερα αμπάρια με δεδομένο αριθμό τρόπων, επιτρέποντας έτσι τη μεταφορά πολλών προϊόντων ταυτόχρονα από το ίδιο πλοίο. Οι περιορισμοί κατανομής του TSP-ATWPC διασφαλίζουν ότι ο διαχωρισμός του εύκαμπτου αμπάριου του πλοίου και η κατανομή των φορτίων στα μικρότερα αμπάρια είναι εφικτές καθ' όλη τη διάρκεια της σειράς επισκέψεων. Το

TSP-ATWPC επιλύεται ως πρόβλημα της συντομότερης διαδρομής σε ένα γράφημα του οποίου οι κόμβοι είναι οι καταστάσεις που αντιπροσωπεύουν το σύνολο των κόμβων στη διαδρομή, τον τελευταίο κόμβο που επισκέφθηκε και την συσσωρευμένη κατανομή φορτίου. Τα τόξα του γραφήματος αντιπροσωπεύουν μεταβάσεις από τη μια κατάσταση στην άλλη. Ο αλγόριθμος είναι ένας εμπρόσθιος αλγόριθμος δυναμικού προγραμματισμού. Ένας αριθμός δοκιμών κυριαρχίας και εξάλειψης εισάγονται για τη μείωση του χώρου κατάστασης. Τα υπολογιστικά αποτελέσματα δείχνουν ότι ο προτεινόμενος αλγόριθμος για τα έργα TSP-ATWPC και οι βέλτιστες λύσεις λαμβάνονται στο πραγματικό πρόβλημα προγραμματισμού πλοίων.

Οι Manuel Lopez-Ibanez et al. [6] εξηγούν τη συνδυαστική βελτιστοποίηση. Δεν είναι σπάνιο να βρεθούν προβλήματα των οποίων η μαθηματική δομή είναι σχεδόν ίδια, διαφέροντας μόνο σε κάποια πτυχή που σχετίζεται με την κινητήρια εφαρμογή. Για παράδειγμα, πολλά προβλήματα στον προγραμματισμό μηχανών και στη δρομολόγηση οχημάτων έχουν ισοδύναμες διατυπώσεις και διαφέρουν μόνο ως προς τον στόχο βελτιστοποίησης ή τους συγκεκριμένους περιορισμούς. Επιπλέον, ενώ ορισμένα προβλήματα τυγχάνουν μεγάλης προσοχής από την ερευνητική κοινότητα, άλλα παρόμοια προβλήματα δεν λαμβάνουν σχεδόν καθόλου προσοχή. Δεδομένων δύο στενά συνδεδεμένων προβλημάτων, είναι διαισθητικό ότι μπορεί να είναι αποτελεσματικό να προσαρμόζονται αλγόριθμοι τελευταίας τεχνολογίας - που αρχικά εισήχθησαν για την καλά μελετημένη παραλλαγή προβλήματος - στην παραλλαγή του προβλήματος που έχει μελετηθεί λιγότερο. Σε αυτό το έγγραφο οι συγγραφείς παρέχουν ένα παράδειγμα που βασίζεται στο πρόβλημα του περιοδεύοντος πωλητή με χρονικά παράθυρα που υποστηρίζει την ανωτέρω σκέψη. Σε αυτό το πλαίσιο, η καλά μελετημένη παραλλαγή του προβλήματος ελαχιστοποιεί τον χρόνο ταξιδιού, ενώ η παραλλαγή του λιγότερο μελετημένου προβλήματος ελαχιστοποιεί το εύρος της αλληλουχίας εργασιών.

2.3 Ανασκόπηση μελετών για προβλήματα VRP-TW

Οι Martin Desrochers et al. [17] εξήγησαν το πρόβλημα δρομολόγησης οχημάτων με παράθυρα χρόνου (VRPTW). Είναι μια γενίκευση του προβλήματος της δρομολόγησης οχημάτων όπου η εξυπηρέτηση ενός πελάτη μπορεί να ξεκινήσει εντός του χρονικού παραθύρου που ορίζεται από τις παλαιότερες και τις πιο πρόσφατες φορές που ο πελάτης θα επιτρέψει την έναρξη της υπηρεσίας. Οι συγγραφείς παρουσιάζουν την ανάπτυξη ενός νέου αλγορίθμου βελτιστοποίησης για τη λύση του. Η μέθοδος της χαλάρωσης της σύνθεσης του συνόλου του VRPTW επιλύεται με τη δημιουργία στηλών. Οι εφικτές στήλες προστίθενται όπως απαιτείται λύνοντας ένα πρόβλημα συντομότερης διαδρομής με χρονικά παράθυρα και περιορισμούς χωρητικότητας χρησιμοποιώντας δυναμικό προγραμματισμό. Η λύση LP που λαμβάνεται παρέχει γενικά ένα εξαιρετικό κατώτερο όριο που χρησιμοποιείται σε έναν αλγόριθμο διακλάδωσης και δέσμευσης για την επίλυση της διατύπωσης κατάτμησης συνόλου ακεραίων. Τα αποτελέσματα υποδεικνύουν ότι αυτός ο αλγόριθμος αποδείχθηκε επιτυχής σε μια ποικιλία προβλημάτων δοκιμής VRPTW συγκριτικής αξιολόγησης πρακτικού μεγέθους. Ο αλγόριθμος ήταν ικανός να λύνει βέλτιστα προβλήματα 100 πελατών.

Οι Olli Bräysy et al. [18] παρουσιάζουν μια έρευνα για το πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα (VRPTW). Το πρόβλημα που μελέτησαν αφορούσε τον σχεδιασμό διαδρομών χαμηλού κόστους από μια αποθήκη σε ένα σύνολο γεωγραφικά διάσπαρτων σημείων. Οι διαδρομές πρέπει να σχεδιάζονται με τέτοιο τρόπο ώστε κάθε σημείο να επισκέπτεται μόνο μία φορά ακριβώς ένα όχημα εντός δεδομένου χρονικού διαστήματος, όλες οι διαδρομές να ξεκινούν και να τελειώνουν στο ίδιο σημείο και οι συνολικές απαιτήσεις όλων των σημείων σε μια συγκεκριμένη διαδρομή δεν πρέπει να υπερβαίνουν το χωρητικότητα του οχήματος. Εξετάζονται τόσο οι παραδοσιακές ευρετικές μέθοδοι κατασκευής διαδρομής όσο και οι πρόσφατοι αλγόριθμοι τοπικής αναζήτησης. Περιγράφονται τα βασικά χαρακτηριστικά κάθε μεθόδου και παρουσιάζονται. Επιπλέον, αναλύεται πώς θα πρέπει να αξιολογούνται οι ευρετικές μέθοδοι και προτείνεται η χρήση της έννοιας της βελτιστοποίησης Pareto στη σύγκριση διαφορετικών ευρετικών προσεγγίσεων.

2.4 Συμμετρικό πρόβλημα TSPTW

Για το συμμετρικό πρόβλημα TSPTW ορίζουμε τα εξής:

Σύνολα δεικτών: Έστω $G = (V, A)$ ένα πλήρες γράφημα όπου, $V = \{0, 1, 2, \dots, n\}$ σύνολο κόμβων (κορυφές, πόλεις, πελάτες κ.λπ.), 0 είναι η αποθήκη και $A = \{(i, j) \mid i \neq j; i, j \in V\}$ είναι το σύνολο των μη κατευθυνόμενων τόξων.

Οι παράμετροι είναι οι εξής:

t_{ij} : Ο χρόνος ταξιδιού από τον κόμβο i στον κόμβο j .

a_i : Η ελάχιστη ώρα επίσκεψης (έναρξη εξυπηρέτησης) του κόμβου i .

b_i : Η αργότερη ώρα επίσκεψης του κόμβου i .

$[a_i, b_i]$: Χρονικό παράθυρο του κόμβου i .

Για απλότητα, εάν απαιτείται χρόνος εξυπηρέτησης σε έναν κόμβο i , αυτός ο χρόνος θα περιλαμβάνεται στον χρόνο ταξιδιού t_{ij} . Το TSPTW ονομάζεται συμμετρικό εάν $t_{ij} = t_{ji}$ και ασύμμετρο εάν $t_{ij} \neq t_{ji}$.

Οι μεταβλητές απόφασης του προβλήματος είναι;

t_i : Ο χρόνος που έχει περάσει μέχρι την επίσκεψη στον κόμβο i .

x_{ij} : Η βοηθητική δυαδική μεταβλητή, δηλαδή, $x_{ij} \in \{0,1\}$.

Ο Baker (1983) [12] παρουσίασε το πρώτο μαθηματικό μοντέλο για συμμετρικό TSPTW όπου η συνολική διάρκεια περιήγησης ελαχιστοποιείται. Το μοντέλο του Baker είναι η μόνη σύνθεση για απλή ελαχιστοποίηση του χρόνου ολοκλήρωσης. Με τις παραπάνω σημειώσεις, το μοντέλο του Baker μοντελοποιείται ως εξής:

$$\text{Minimize } t_{n+1} - t_0$$

Με

- $t_i - t_0 \geq t_{0i}$ για $i=1,2,\dots,n$
- $|t_i - t_j| \geq t_{ij}$ για $i=1,2,\dots,n, 1 \leq j < i$
- $t_{n+1} - t_i \geq t_{i0}$ για $i=1,2,\dots,n$
- $t_i \geq a_i$ για $i=1,2,\dots,n$
- $t_i \leq b_i$ για $i=1,2,\dots,n$
- $t_i \geq 0$ για $i=1,2,\dots,n+1$

Αυτό το μοντέλο περιέχει $n+1$ μεταβλητές απόφασης και περιορισμούς $O(n^2)$ και υποθέτει έναν συμμετρικό και πλήρη πίνακα απόστασης για τον οποίο ισχύει η ανισότητα του τριγώνου. Πρέπει επίσης να υποθέσουμε ότι ο χρόνος που δαπανάται στο τόξο είναι ένας βαθμωτός μετασχηματισμός της διανυθείσας απόστασης, επομένως ο χρόνος και η απόσταση μπορούν να χρησιμοποιηθούν εναλλακτικά. Η εγκυρότητα της παραπάνω διατύπωσης έχει αποδειχθεί με την απόδειξη ενός θεωρήματος που δόθηκε στον Baker (1983). Το μοντέλο είναι μη γραμμικό στη φύση του, δηλαδή δεν μπορεί να χρησιμοποιηθεί απευθείας από έναν επιλύτη γραμμικού προγραμματισμού. Ο Baker (1983) ανέπτυξε μια διαδικασία διακλάδωσης και δέσμευσης και έλυσε τυχαία δημιουργούμενα στιγμιότυπα με έως και 50 κόμβους μέχρι τη βέλτιστη.

2.5 Ασύμμετρο πρόβλημα TSPTW

Οι Aschuer et al. (2001) [19] μελέτησαν λεπτομερώς το ασύμμετρο TSPTW και παρουσίασαν τρεις συνθέσεις για την ελαχιστοποίηση του χρόνου που δαπανάται στα τόξα.

Σε αυτή την ενότητα, εξετάζουμε την περίπτωση όπου $t_{ij} \neq t_{ji}$, δηλ., το ασύμμετρο TSPTW. Με τους συμβολισμούς που ορίστηκαν στην προηγούμενη ενότητα, θεωρούμε ότι ορίζεται μια νέα απόφαση, οι μεταβλητές x_{ij} είναι ίσες με 1 εάν ο ταξιδιώτης πηγαίνει από τον κόμβο i στον κόμβο j , διαφορετικά 0.

Κάτωθι παρουσιάζεται μια διατύπωση ακέрайου προγραμματισμού για αυτήν την περίπτωση.

Αρχικά αναφέρονται κάποιες προτάσεις και παρουσιάζονται έγκυρες ανισότητες για τη διατύπωση και στη συνέχεια παρουσιάζεται το μαθηματικό μοντέλο.

Όταν ο ταξιδιώτης πηγαίνει από την αποθήκη σε έναν κόμβο i , τότε πρέπει να ισχύει ότι $t_i = \max\{a_i, t_{0i}\}$. Για να γίνει αυτό, λαμβάνοντας υπόψη τους πρώτους χρόνους εξυπηρέτησης κάθε κόμβου, δηλώνουμε την ακόλουθη πρόταση.

Οι ακόλουθες ανισότητες αρχικοποιούν τον χρόνο άφιξης του πρώτου κόμβου μιας περιήγησης και τον χρόνο που απαιτείται για να ταξιδέψετε από την αρχή σε αυτόν τον κόμβο.

- $t_i - t_{0i}x_{ij} \geq 0$ για $i=1,2,\dots,n$
- $t_i \geq a_i$ για $i=1,2,\dots,n$

Αν ο ταξιδιώτης πάει από την αποθήκη στον κόμβο i τότε $x_{0i} = 1$, άρα, λαμβάνουμε $t_i = \max\{a_i, t_{0i}\}$. Αν ο κόμβος i είναι ένας ενδιάμεσος κόμβος, τότε $x_{0i} = 0$.

Ο ορισμός των x_{ij} και t_i υπονοεί ότι, αν $x_{ij} = 1$, τότε πρέπει να ισχύει $t_j \geq t_i + t_{ij}$. Για να γίνει αυτό, δίνουμε την ακόλουθη πρόταση.

Η υπό όρους πρόταση «αν $x_{ij} = 1$ τότε $t_j \geq t_i + t_{ij}$ » μπορεί να γραμμικοποιηθεί ως:

$$t_i - t_j + (b_i - a_j + t_{ji})x_{ji} \leq b_i - a_j \quad i \neq j; i, j = 1, 2, \dots, n$$

Οι ανισότητες του προβλήματος εγγυώνται ότι οι χρόνοι άφιξης στον πελάτη σε μια περιήγηση αποτελούν μια βηματικά αυξανόμενη συνάρτηση, πράγμα που σημαίνει

ότι αυτές οι ανισότητες απαγορεύουν τη δημιουργία οποιασδήποτε μη επιτρεπτής υποπεριήγησης, δηλ. αποτελούν τους περιορισμούς εξάλειψης της υποπεριήγησης της σύνθεσής μας.

Σύμφωνα με τις επεξηγήσεις και τους παραγόμενους περιορισμούς που δίνονται παραπάνω και προσθέτοντας επίσης την ιδιότητα της μη αρνητικότητας και τους δυαδικούς περιορισμούς, δηλώνουμε νέα ακέραια φόρμουλα προγραμματισμού για ασύμμετρο TSPTW ως εξής:

Minimize t_{n+1}

Και

$$\begin{array}{ll}
 t_i - t_{0i}x_{0i} \geq 0 & \text{για } i=1,2,\dots,n \\
 t_i \geq a_i & \text{για } i=1,2,\dots,n \\
 t_i - t_j + (b_i - a_j + t_{ji})x_{ji} \leq b_i - a_j & \text{για } i \neq j; i, j = 1,2..n \\
 \sum_{i=0}^n x_{ij} = 1 & \text{για } i=1,2,\dots,n \\
 \sum_{j=0}^n x_{ij} = 1 & \text{για } i=1,2,\dots,n \\
 t_i \leq b_i & \text{για } i=1,2,\dots,n \\
 t_i - t_{i0} \leq t_{n+1} & \text{για } i=1,2,\dots,n \\
 x_{ij} \in \{0,1\} \forall e \in A
 \end{array}$$

Σε αυτή τη διατύπωση, έχουμε $2n^2 + 6n + 2$ περιορισμούς, $n^2 - n$ δυαδικές μεταβλητές και $n+1$ συνεχείς μεταβλητές. Έτσι η προτεινόμενη σύνθεση έχει $O(n^2)$ δυαδικές μεταβλητές και $O(n^2)$ περιορισμούς. Η αντικειμενική συνάρτηση ελαχιστοποιεί τη διάρκεια της περιήγησης. Οι περιορισμοί που δίνονται αρχικοποιούν τα t_i 's και συνιστούν μια αυξανόμενη συνάρτηση βήματος των t_i 's, οπότε τότε δεν θα σχηματιστεί μη επιτρεπτή περιήγηση. Οι δυο ακόλουθοι περιορισμοί είναι οι περιορισμοί ανάθεσης και οι επόμενοι δυο είναι οι περιορισμοί του χρονικού παραθύρου. Ο επόμενος περιορισμός εγγυάται ότι η διάρκεια της περιήγησης είναι μικρότερη ή ίση με τον χρόνο επίσκεψης συν τον χρόνο που απαιτείται για την επιστροφή στην αποθήκη για όλους τους κόμβους. Η μη αρνητικότητα και οι δυαδικοί περιορισμοί δίνονται από τους τελικούς περιορισμούς.

Αυτή η διατύπωση μπορεί εύκολα να ξαναγραφτεί για την περίπτωση που έχουμε πολλούς ταξιδιώτες.

Εάν υπάρχουν πρόσθετοι περιορισμοί για τη σειρά της επίσκεψης στον κόμβο, όπως αποδεικνύεται για τη συμμετρική περίπτωση, μπορούν εύκολα να εισαχθούν ως νέοι περιορισμοί και σε αυτό το μοντέλο.

Κεφάλαιο 3^ο

3.1 Μεθοδολογίες επίλυσης του προβλήματος περιοδεύοντος πωλητή με χρονικά παράθυρα

3.1.1 Επίλυση του προβλήματος του περιοδεύοντος πωλητή με χρονικά παράθυρα με χρήση λαγκραντζιανής χαλάρωσης

Η μελέτη των Rosa Herrero, Juan José Ramos, and Daniel Guimarans [23] παρουσιάζει μια μεταερευτική μεθοδολογία βασισμένη στην τεχνική της Λαγκραντζιανής Χαλάρωσης, που εφαρμόζεται στο Πρόβλημα του Περιοδεύοντος Πωλητή με περιορισμό χρονικών παραθύρων. Η παρουσιαζόμενη προσέγγιση συνδυάζει τον αλγόριθμο Subgradient Optimization με μια ευρετική για να ληφθεί μια εφικτή αρχική λύση από μια διπλή λύση. Επιπλέον, έχει εισαχθεί μια παράμετρος για τη βελτίωση της σύγκλισης αλγορίθμων. Το κύριο πλεονέκτημα βασίζεται στην επαναληπτική εξέλιξη τόσο των άνω όσο και των κατώτερων ορίων στο βέλτιστο κόστος, παρέχοντας μια εφικτή λύση σε λογικό αριθμό επαναλήψεων με ένα στενό εύρος μεταξύ του αρχικού και του βέλτιστου κόστους.

Η Χαλάρωση Lagrangean (LR) είναι μια πολύ γνωστή μέθοδος για την επίλυση προβλημάτων συνδυαστικής βελτιστοποίησης μεγάλης κλίμακας. Λειτουργεί μεταφέροντας περιορισμούς που είναι δύσκολο να ικανοποιηθούν στην αντικειμενική συνάρτηση που συσχετίζει μια ποινή σε περίπτωση που δεν ικανοποιηθούν. Μια εξαιρετική εισαγωγή στο όλο θέμα της LR μπορεί να βρεθεί στο (Fisher, 1981).

Ο όρος «μεταερευτικό» αναφέρεται σε μια κύρια στρατηγική που καθοδηγεί και τροποποιεί με ευρετική μεθοδολογία για να παράγει λύσεις πέρα από αυτές που παράγονται κανονικά σε μια αναζήτηση τοπικής βελτιστοποίησης» (Glover και Laguna, 1997 [22]). Σε σύγκριση με την κλασική ευρετική, η μεταερευτική του Lagrange παρέχει τόσο ένα ανώτερο όσο και ένα κατώτερο όριο (UB και LB), επομένως έναν μεταγενέστερο ποιοτικό έλεγχο της λύσης που λαμβάνεται.

Επιπλέον, μειώνει τον χώρο αναζήτησης, καθώς μπορούν να χρησιμοποιηθούν διπλές πληροφορίες για την περικοπή μεταβλητών απόφασης.

Η προτεινόμενη μέθοδος στο [27] που βασίζεται στην μέθοδο LR χρησιμοποιεί τον αλγόριθμο βελτιστοποίησης Subgradient σε συνδυασμό με μια ευρετική. Με στόχο τη βελτίωση της σύγκλισης του αλγορίθμου στη βέλτιστη λύση, εισάγεται μια ευρετική για να ληφθεί μια εφικτή λύση από τη διπλή μεταβλητή. Πράγματι, αυτή η μέθοδος προσπαθεί να βελτιώσει το άνω όριο με τις τιμές αυτών των εφικτών λύσεων, έτσι επιτυγχάνεται καλύτερη σύγκλιση.

Το συμμετρικό πρόβλημα TSP μπορεί να οριστεί σε ένα πλήρες μη κατευθυνόμενο γράφημα $G=(I,E)$ που συνδέει το σύνολο πελατών $I=\{1,2,,\}$ μέσω ενός συνόλου μη κατευθυνόμενων ακμών $E=\{(i,j)/ i,j \in I\}$. Η άκρη $e=(i,j)$ στο E έχει συσχετίσει ένα κόστος ταξιδιού, που υποτίθεται ότι είναι η διαδρομή με το χαμηλότερο κόστος που συνδέει τον κόμβο i με τον κόμβο j .

Η επίλυση του TSP συνίσταται στον καθορισμό μιας διαδρομής της οποίας το συνολικό κόστος ταξιδιού ελαχιστοποιείται έτσι ώστε κάθε πελάτης να επισκέπτεται ακριβώς μία φορά και η διαδρομή να ξεκινά και να τελειώνει στην αρχή ($i=1$).

Η κλασική διατύπωση απαιτεί τον ορισμό της δυαδικής μεταβλητής x_e για να υποδηλώσει ότι η ακμή $e=(i,j) \in E$ χρησιμοποιείται στη διαδρομή. Αυτό είναι $x_e=1$ εάν ο πελάτης j επισκέπτεται αμέσως μετά το i . αλλιώς $x_e=0$. Έτσι, το TSP μπορεί να διατυπωθεί μαθηματικά ως εξής:

$$\min \sum_{e \in E} c_e x_e$$

και

$$\sum_{e \in \delta(i)} x_e = 2,$$

$$\forall i \in I$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1,$$

$$\forall S \subset I, |S| \leq \frac{1}{2}|I|$$

Όπου $\delta(i) = \{e \in E : \exists j \in I, e = (i, j) \text{ ή } (j, i)\}$ αντιπροσωπεύει το σύνολο των τόξων των οποίων ο αρχικός ή ο τελικός κόμβος είναι i . και $E(S) = \{e = (i, j) \in E : i, j \in S\}$ αντιπροσωπεύει το σύνολο τόξων των οποίων οι κόμβοι βρίσκονται στο υποσύνολο S των κορυφών.

Ο επόμενος περιορισμός δηλώνει ότι κάθε κόμβος $i \in I$ πρέπει να επισκέπτεται μία φορά, δηλαδή, κάθε πελάτης πρέπει να έχει δύο προσπίπτουσες ακμές.

Ο τελευταίος περιορισμός εξάλειψης υποπεριήγησης δηλώνει ότι η διαδρομή πρέπει να είναι κύκλος Hamiltonian, επομένως δεν μπορεί να έχει υποκύκλο.

Η μέθοδος LR εκμεταλλεύεται τη δομή του προβλήματος, επομένως μειώνει σημαντικά την πολυπλοκότητα του προβλήματος. Ωστόσο, είναι συχνά ένα σημαντικό ζήτημα να βρούμε τους βέλτιστους λαγκρατζιανούς πολλαπλασιαστές. Ο πιο συχνά χρησιμοποιούμενος αλγόριθμος είναι ο Subgradient Optimization (SO). Η κύρια δυσκολία αυτού του αλγορίθμου έγκειται στην επιλογή ενός σωστού μεγέθους βήματος λ_k προκειμένου να διασφαλιστεί η σύγκλιση του αλγορίθμου (Reinelt, 1994 [25]).

Προκειμένου να αντιμετωπιστεί αυτός ο περιορισμός, η προτεινόμενη μέθοδος συνδυάζει τον αλγόριθμο SO με έναν ευρετικό για να ληφθεί μια εφικτή λύση από μια διπλή λύση. Μπορεί να πάρει ένα καλύτερο άνω όριο, επομένως βελτιώνει τη σύγκλιση στη βέλτιστη λύση ξεκινώντας από ένα αρχικό UB που λαμβάνεται με ένα ευρετικό αλγόριθμο πλησιέστερο γειτονικού κόμβου. Παρά το ότι η βέλτιστη λύση δεν μπορεί να επιτευχθεί πάντα, η προτεινόμενη μέθοδος είναι σε θέση να παρέχει μια εφικτή λύση με ένα στενό εύρος μεταξύ του αρχικού και του βέλτιστου κόστους σε λογικό αριθμό επαναλήψεων.

Η προτεινόμενη από την μελέτη μέθοδος χαλάρωσης, μικραίνει το σύνολο περιορισμών που απαιτεί να εξυπηρετούνται όλοι οι πελάτες σταθμίζοντάς τους με ένα πολλαπλασιαστικό διάνυσμα u , αφού όλοι οι υποκύκλοι μπορούν να αποφευχθούν κατασκευάζοντας τη λύση x ως ένα δένδρο. Στην πραγματικότητα, μια

εφικτή λύση του TSP είναι ένα δέντρο που έχει δύο προσπίπτουσες ακμές σε κάθε κόμβο (Held και Karp, 1971 [24]). Το πλεονέκτημα είναι ότι η εύρεση ενός ελάχιστου δέντρου είναι σχετικά εύκολη.

Το δυαδικό πρόβλημα που προκύπτει είναι το εξής:

$$\max_{u \in R} L(u)$$

Όπου

$$L(u) = \sum_{e \in E} c_e x_e + \sum_{i \in I} u_i (2 - \sum_{e \in \delta(i)} x_e),$$

Και καθορίζεται η υποβαθμίδα ως:

$$\gamma_i^k = 2 - \sum_{e \in \delta(i)} x_e$$

Η προτεινόμενη μέθοδος που βασίζεται στην LR μέθοδο μπορεί να θεωρηθεί ως προδιαγραφή του μεταερευτικού Lagrangean που παρουσιάστηκε από τους Boschetti και Maniezzo (2009) [20]. Ένας ευρετικός αλγόριθμος λαμβάνει μια εφικτή λύση από τη διπλή μεταβλητή, έτσι προσπαθεί να βελτιώσει το UB και επιτυγχάνεται καλύτερη σύγκλιση. Τελικά, αυτή η εφικτή λύση μπορεί να παρασχεθεί ως η καλύτερη λύση εάν σταματήσει η μέθοδος. Το κριτήριο διακοπής βασίζεται στον μέγιστο αριθμό επαναλήψεων και επίσης σε μια εξαίρεση κινητής υποδιαστολής. Ο προτεινόμενος αλγόριθμος μεθόδου όπως παρουσιάστηκε στην μελέτη [27] φαίνεται στον Πίνακα 3.1.

Πίνακας 3.1: Ο προτεινόμενος αλγόριθμος μεθόδου

<p>Έναρξη</p> <ol style="list-style-type: none"> 1 Έναρξη παραμέτρων $u^0 = 0$; $\delta_0=2$; $\rho=0.95$; $a_L=1/3$ 2. αποκτήστε ένα UB εφαρμόζοντας το Ευρετικό Πλησιέστερου Γείτονα 3. Έναρξη $\bar{L} = L(u^0) + a_L(UB - L(u^0))$ <p>Επανάληψη k</p> <ol style="list-style-type: none"> 4. Λύστε τη λαγκραντζιανή συνάρτηση $L(u^k)$ 5. Ελέγξτε το υποκείμενο $\gamma_i^k = 2 - \sum_{e \in \delta(i)} x_e$ <p>Αν $\ \gamma^k\ ^2 = 0$ τότε έχει βρεθεί η βέλτιστη λύση $\rightarrow EXIT$</p> <p>Αν $\ \gamma^k\ ^2 < \xi$ εφαρμόστε μια ευρετική για να βελτιώσετε το UB</p> 6. Ελέγξτε την παράμετρο \bar{L} 7. Υπολογίστε το μέγεθος του βήματος λ_k 8. Ενημερώστε τον πολλαπλασιαστή $u^{k+1} = u^k + \lambda_k \gamma^k$ 9. $k \leftarrow k+1$

Η προτεινόμενη ευρετική για τη βελτίωση του UB εφαρμόζεται όταν το 1-δέντρο είναι σχεδόν ένα μονοπάτι Hamiltonian. Δηλαδή, εάν η υποβάθμιση ικανοποιεί τη εξής σχέση (βήμα 8).

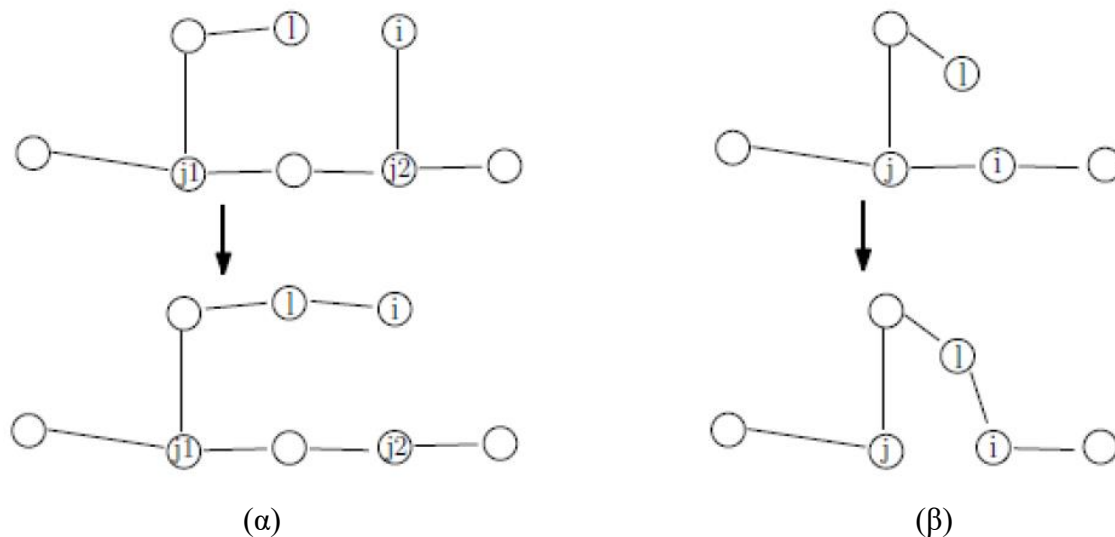
$$\|\gamma^k\|^2 < \xi$$

Καθώς κάθε λύση είναι 1-δέντρο, αυτό το κριτήριο σημαίνει ότι η λύση έχει λίγες κορυφές χωρίς δύο προσπίπτουσες ακμές. Αυτή η ευρετική αντικαθιστά μια ακμή $e=(i,j)$ όπου το j έχει μερικές επιπλέον ακμές για μια ακμή $e=(i,1)$ όπου η κορυφή 1 έχει μία μόνο ακμή ελαχιστοποιώντας το κόστος της ανταλλαγής.

Στην παρουσιαζόμενη προσέγγιση, έχουν οριστεί δύο διαφορετικές κινήσεις (Εικόνα 3.1):

(α) απεριόριστη κίνηση και

(β) η περιορισμένη κίνηση



Εικόνα 3.1: Κινήσεις του ευρετικού (α) απεριόριστης κίνησης και (β) περιορισμένης κίνησης [27]

Μια καλή εκτίμηση της παραμέτρου ξ θα αποφύγει την αύξηση του χρόνου υπολογισμού. Πρώτον, η τιμή του μπορεί να είναι μεγάλη, για παράδειγμα η τιμή της πρώτης επανάληψης

$$\xi = \|\gamma^1\|^2$$

αλλά θα πρέπει να ενημερώνεται όποτε βρεθεί μια καλύτερη εφικτή λύση σύμφωνα με το εξής:

$$\xi = \|\gamma^k\|^2$$

Εάν αυτή η παράμετρος δεν ενημερωθεί σωστά, η ευρετική γίνεται χρονοβόρα. Τελικά, η ευρετική θα μπορούσε να βρει τη βέλτιστη λύση χωρίς να την ανιχνεύσει, οπότε η μέθοδος θα συνέχιζε να επαναλαμβάνεται μέχρι το $LB=UB$.

Όπως αναφέρθηκε, η σύγκλιση του αλγορίθμου επηρεάζεται κρίσιμα από το μέγεθος βήματος λ_k . Αυτή η τιμή βασίζεται είτε στο LB είτε στο UB, τα οποία είναι συνήθως άγνωστα ή δεν εκτιμώνται σωστά. Επομένως, η σύγκλιση μπορεί να μην είναι εξασφαλισμένη για όλες τις περιπτώσεις. Για να ξεπεραστούν αυτοί οι περιορισμοί γίνεται χρήση μιας παραμέτρου \bar{L} τέτοια ώστε:

$$LB \leq \bar{L} \leq UB$$

Εξ ορισμού, αυτή η παράμετρος αντιστοιχεί σε μια καλύτερη εκτίμηση της βέλτιστης τιμής L^* από αυτές που λαμβάνονται για τα LB και UB. Ο υπολογισμός του μεγέθους του βήματος γίνεται:

$$\lambda_k = \delta_k \frac{|\bar{L} - L(u^k)|}{\|\gamma_k\|^2}$$

Η σύγκλιση προκύπτει και εξασφαλίζεται εάν ο ακόλουθος όρος τείνει στο μηδέν.

$$\bar{L} - L(u^k)$$

Με τη σειρά του, η απόδοση σύγκλισης μπορεί να βελτιωθεί όσο η νέα παράμετρος \bar{L} πλησιάζει πιο κοντά στην (άγνωστη) βέλτιστη λύση.

Η κύρια ιδέα είναι πολύ απλή: καθώς ο αλγόριθμος συγκλίνει στη λύση, είναι γνωστά νέα καλύτερα κατώτερα όρια και μπορούν να ληφθούν νέες καλύτερες εκτιμήσεις άνω ορίων χρησιμοποιώντας την ευρετική που έχει σχεδιαστεί για να ληφθούν εφικτές λύσεις. Επομένως, η παράμετρος \bar{L} ενημερώνεται σύμφωνα με τις ακόλουθες συνθήκες:

Αρχικοποιείται ως εξής:

$$\bar{L} = L(u^0) + \alpha_L(UB - L(u^0))$$

Όπου

$$0 < \alpha_L < 1$$

Στην συνέχεια αν,

$$L(u^k) > \bar{L}$$

Τότε η μεταβλητή μεταβάλλεται και ανανεώνεται με τον εξής τύπο:

$$\bar{L} = L(u^k) + \alpha_L(UB - L(u^k))$$

Αν

$$\bar{L} > UB$$

τότε

$$\bar{L} = UB$$

Τέλος, η παράμετρος δ_k αρχικοποιείται στην τιμή 2 και ενημερώνεται όπως προτείνουν οι Zamani και Lau (2010) [26]. Εάν το κάτω όριο δεν βελτιωθεί, το δ_k μειώνεται, χρησιμοποιώντας τον εξής τύπο:

$$\delta_{k+1} = \delta_k \rho$$

Όπου το ρ παίρνει τιμές ως εξής:

$$0 < \rho < 1$$

Από την άλλη πλευρά, εάν το κάτω όριο είναι βελτιωμένο, τότε η τιμή του αυξάνεται σύμφωνα με τον ακόλουθο τύπο:

$$\delta_{k+1} = \delta_k \frac{3-\rho}{2}$$

Με την προϋπόθεση ότι

$$0 \leq \delta_k \leq 2$$

Άρα συγκεντρωτικά στην μελέτη [27] παρουσιάστηκε μια μεταευρετική μεθοδολογία βασισμένη στην τεχνική λαγκραντζιανή χαλάρωση. Αυτό το σχήμα έχει χρησιμοποιηθεί για την αντιμετώπιση του προβλήματος του ταξιδιώτη πωλητή.

Η προτεινόμενη μέθοδος που βασίζεται χρησιμοποιεί τον αλγόριθμο SO σε συνδυασμό με μια ευρετική. Με στόχο τη βελτίωση της σύγκλισης του αλγορίθμου στη βέλτιστη λύση, εισάγεται η ευρετική προκειμένου να ληφθεί μια εφικτή λύση από τη διπλή μεταβλητή. Από τη μία πλευρά, η μέθοδος παρέχει τόσο UB όσο και LB, επομένως έναν μεταγενέστερο ποιοτικό έλεγχο της λύσης που προκύπτει. Από την άλλη πλευρά, μειώνει τον χώρο αναζήτησης, καθώς μπορούν να χρησιμοποιηθούν διπλές πληροφορίες για την περικοπή μεταβλητών απόφασης. Τέλος, παρά το ότι η βέλτιστη δεν μπορεί να επιτευχθεί πάντα, η προτεινόμενη μέθοδος είναι σε θέση να παρέχει μια εφικτή λύση με ένα στενό χάσμα μεταξύ του αρχικού και του βέλτιστου κόστους σε λογικό αριθμό επαναλήψεων.

Πρέπει να σημειωθεί ότι υπάρχουν αρκετά σημεία για μελλοντική έρευνα. Πρώτον, οι παράμετροι ρ και αL πρέπει να ρυθμιστούν με διαδικασίες μικρορύθμισης. Δεύτερον, δεδομένου ότι η παράμετρος \bar{L} είναι μια καλή εκτίμηση, η ευρετική για να ληφθεί μια εφικτή λύση από μια διπλή πρέπει να βελτιωθεί προκειμένου να μειωθεί ο χρόνος υπολογισμού της. Τέλος, ο παρουσιαζόμενος αλγόριθμος έχει συμπεριληφθεί σε ένα πλαίσιο αναζήτησης μεταβλητού γειτονικού κόμβου για την αντιμετώπιση του προβλήματος δρομολόγησης χωρητικότητας οχημάτων, δείχνοντας πολύ καλά αποτελέσματα τόσο ως προς την ποιότητα της λύσης όσο και ως προς την υπολογιστική απόδοση (Guimarans et al 2010).

3.1.2 Επίλυση του προβλήματος του περιοδεύοντος πωλητή με χρονικά παράθυρα με χρήση χρονικά διευρυμένων δικτύων

Πρόσφατα, για ένα πρόβλημα σχεδιασμού δικτύου υπηρεσιών οι Boland et al. (2015) [28] εισάγουν μια νέα προσέγγιση, στην οποία αξιοποιείται η ισχύς μιας διατύπωσης διευρυμένου ακέραιου γραμμικού προγραμματισμού (LP), χωρίς τα αρνητικά στοιχεία ενός υπερβολικά μεγάλου μοντέλου.

Η προσέγγιση λειτουργεί με ένα προσεκτικά σχεδιασμένο μερικώς εκτεταμένο δίκτυο. Η τιμή μιας διατύπωσης LP σε αυτό το δίκτυο αποδίδει ένα διπλό όριο. Από τη λύση, μπορούν να εξαχθούν νέα σημεία χώρου και χρόνου που, όταν προστεθούν, διασφαλίζουν ότι θα προκύψει μια νέα λύση και εγγυώνται τη σύγκλιση στη βέλτιστη λύση, καθώς η διαδικασία επαναλαμβάνεται.

Αυτή η δυναμική προσέγγιση για τη δημιουργία ενός επαρκούς μερικώς διευρυμένου χρονικά δικτύου καταδεικνύεται από τους Boland et al. (2015) [28] να έχει μεγάλη επιτυχία. Δημιουργεί μόνο ένα μικρό ποσοστό του πλήρους χρονικά διευρυμένου δικτύου και μπορεί να βρει αποδεδειγμένες, βέλτιστες, λύσεις σε περιπτώσεις που είναι πολύ μεγάλες για να επιχειρηθεί ακόμη και με άλλες συνθήσεις.

Στην μελέτη [28] γίνεται προσπάθεια επέκτασης αυτής τη δυναμικής προσέγγισης στο πρόβλημα TSPTW.

Οι Wang και Regan (2002) [29] δίνουν δύο μοντέλα LP που βασίζονται στη διακριτοποίηση των χρονικών παραθύρων. Ένα που παράγει ένα άνω όριο και ένα που αντίστοιχα παράγει ένα κάτω όριο. Προτείνουν την επαναληπτική επαναδιαμέριση των χρονικών παραθύρων. Αποδεικνύεται ότι τα κάτω όρια συγκλίνουν (Wang and Regan, 2009)[30].

Ωστόσο, το μέγεθος των μοντέλων LP αυξάνεται πολύ γρήγορα. Πιο πρόσφατα, οι Dash et al. (2012) [31] προτείνουν μια μέθοδο που χρησιμοποιεί ένα μοντέλο LP που βασίζεται στη διακριτοποίηση του χρόνου. Τα αποτελέσματα είναι εξαιρετικά καλά.

Υποθέτοντας ακέραιο χρονικό παράθυρο και δεδομένα χρόνου ταξιδιού, το TSPTW μπορεί φυσικά να διατυπωθεί ως το πρόβλημα της εύρεσης μιας διαδρομής ελάχιστου κόστους μέσω ενός (πλήρους) διευρυμένου χρονικά δικτύου.

Χωρίς να επηρεάζεται η γενικότητα της εφαρμογής της μεθόδου, μπορούμε να ορίσουμε $e_0 = \min_{n \in N}(e_n - \tau_{0n})$ και $l_0 = \max_{n \in N}(l_n - \tau_{0n})$. Το χρονικά διευρυμένο δίκτυο έχει κόμβους (n, t) εκπροσωπώντας την πόλη $n \in N$ στην χρονική στιγμή $t \in \{e_0 + \tau_{0n}, \dots, l_n\}$, τους κόμβους $(0, e_0)$ και $(0, \min_{n \in N}(l_n + \tau_{n0}), \dots, (0, l_0)$ και τα τόξα $((n, t), (n, t + 1))$ για $n \in N$ και $t = e_0 + \tau_{0n}, \dots, l_n - 1$, $((0, t), (0, t + 1))$ για $t = \min_{n \in N}(l_n + \tau_{n0}), \dots, l_0 - 1$, $((0, e_0), (n, e_0 + \tau_{0n}))$ για $n \in N$, $((n_1, t), (n_2, t + \tau_{n_1, n_2}))$ για όλα τα $n_1, n_2 \in N$ και όλα τα $t = e_{n_1}, \dots, l_{n_1}$ με $t + \tau_{n_1, n_2} \leq l_{n_2}$, και $((n, l_n), (0, l_n + \tau_{n0}))$ για $n \in N$.

Έστω x_a για τόξο a στο δίκτυο με επέκταση του χρόνου να είναι μια δυαδική μεταβλητή που υποδεικνύει εάν χρησιμοποιείται το τόξο ($x_a = 1$) ή όχι ($x_a = 0$).

Το ακόλουθο LP επιλύει το TSPTW:

$$\begin{aligned} \min \quad & \sum_a c_a x_a \\ & \sum_{a \in \delta^+((0, e_0))} x_a = 1, \\ & \sum_{a \in \delta^-((n, t))} x_a - \sum_{a \in \delta^+((n, t))} x_a = 0 \end{aligned}$$

Για

$$(n, t) \notin \{(0, e_0), (0, l_0)\},$$

$$\sum_{t=e_n}^{l_n} \sum_{a \in \delta^+_{\neq}((n, t))} x_a = 1$$

Για

$$n \in \mathbb{N}$$

όπου τα δ^- και δ^+ δηλώνουν, ως συνήθως, το σύνολο τόξων που εισέρχονται και εξέρχονται από έναν κόμβο στο διευρυμένο κατά τον χρόνο δίκτυο, αντίστοιχα, και το $\delta/$ δηλώνει το σύνολο τόξων που εξέρχονται από ένα κόμβο στο διευρυμένο κατά τον χρόνο δίκτυο σε ένα σε διαφορετική πόλη.

Το πλήρες χρονικά διευρυμένο δίκτυο μπορεί να είναι εξαιρετικά μεγάλο, επομένως μπορεί να είναι μη πρακτικό (ακόμα και αδύνατο) να επιλυθεί. Ως εκ τούτου, δημιουργούνται δίκτυα μερικώς διευρυμένα χρονικά.

Ένα μερικώς διευρυμένο δίκτυο $DT = (NT, WT \cup AT)$ προκύπτει από μια συλλογή χρονικών σημείων $T = \{T_i\}_{i \in \mathbb{N}}$ με $T_i = \{t_i, \dots, t_i\}$, όπου $t_i \leq ln$ για όλα τα $i \in \mathbb{N}_0$. Το σύνολο NT έχει έναν κόμβο (i, t) για κάθε $i \in \mathbb{N}_0$ και $t \in T_i$. Το σύνολο WT περιέχει τα τόξα $((i, t_i), (i, t_i))$, για όλα τα $i \in \mathbb{N}_0$ και $k = 1, \dots, n_i - 1$, που διαμορφώνουν τη δυνατότητα αναμονής στην πόλη i , η οποία μπορεί να είναι απαραίτητη εάν η άφιξη στο i συμβεί πριν από την έναρξη του παραθύρου στο i .

Το σύνολο AT περιέχει τόξα της μορφής $((i, t), (j, t^-))$, όπου $(i, j) \in A$, $t \in T_i$, και $t^- \in T_j$, που μοντελοποιούν τη δυνατότητα να ταξιδέψετε από την πόλη i τη στιγμή t για να φτάσετε στην πόλη j τη στιγμή t^- .

Σημειώστε ότι ένα τόξο $((i, t), (j, t^-))$ δεν χρειάζεται να ικανοποιήσει το $t^- - t = t_{ij}$. Στην πραγματικότητα, η ευελιξία εισαγωγής τόξων $((i, t), (j, t^-))$ με χρόνο ταξιδιού μικρότερο από τον πραγματικό χρόνο ταξιδιού, t_{ij} , του τόξου (i, j) , είναι ένα ουσιαστικό χαρακτηριστικό των μερικώς διευρυμένων χρονικά δικτύων και παρέχει έναν μηχανισμό ελέγχου του μεγέθους τους. Υποδηλώνουμε το πρόβλημα της εύρεσης μιας διαδρομής ελάχιστου κόστους από $(0, e_0)$ που επισκέπτεται τον κόμβο (i, t) για κάποιο χρόνο $t \in T_i$, για κάθε $i \in \mathbb{N}_0$, τερματίζοντας στο $i = 0$, από τον αλγόριθμο TSPTW (DT).

Ο αλγόριθμος Solve-TSPTW βελτιώνει (επεκτείνει) επαναληπτικά το NT μέχρι να αποδειχθεί ότι η λύση του TSPTW(DT) δίνει μια βέλτιστη λύση για το TSPTW. Σε κάθε επανάληψη, τα τόξα στο DT κατασκευάζονται προσεκτικά για να εγγυηθούν ότι

- (i) κάθε εφικτή περιήγηση έχει μια αντίστοιχη λύση σε DT, επομένως το TSPTW(DT) είναι μια χαλάρωση και
- (ii) Η τιμή του TSPTW(DT) αποδίδει το καλύτερο πιθανό κάτω όριο αυτού του τύπου.

Ο αλγόριθμος ξεκινά με τη δημιουργία ενός αρχικού DT με βάση το $NT = i \in \mathbb{N}_0 \{(i, e_0 + \tau_{0i}), (i, e_i)\}$. Επειτα το TSPTW(DT) επιλύεται με χρήση γραμμικού προγραμματισμού. Εάν η λύση που προκύπτει δεν είναι μια εφικτή περιήγηση με πραγματικούς χρόνους ταξιδιού, και περιέχει ένα τόξο $((i, t), (j, t'))$, για παράδειγμα, το οποίο είναι πολύ μικρό δηλαδή έχει $t' < t + \tau_{ij}$ επιμηκύνουμε αυτό το τόξο, προσθέτοντας ένα νέο χρονικό σημείο $t' = t + \tau_{ij} = T_j$. Στη συνέχεια η διαδικασία επαναλαμβάνεται. Δεδομένου ότι ο αριθμός των πιθανών χρονικών σημείων είναι πεπερασμένος, μπορούμε να είμαστε σίγουροι για το τι πρόκειται να ακολουθήσει.

Στην παρούσα μελέτη [28] δοκιμάστηκαν 8 παραλλαγές του αλγορίθμου Solve-TSPTW στις 135 περιπτώσεις του Dumas (Dumas et al., 1995) [10].

Σχεδόν όλες οι περιπτώσεις θα μπορούσαν να επιλυθούν με λίγες μόνο επαναλήψεις του αλγορίθμου. Ακόμη και η παραλλαγή αλγορίθμου με τη χειρότερη απόδοση έλυσε σχεδόν το 75% των περιπτώσεων, ενώ η καλύτερη έλυσε λίγο λιγότερο από το 95%, με λιγότερες από 10,2 επαναλήψεις, κατά μέσο όρο. Δεδομένου ότι αυτά τα αποτελέσματα προέκυψαν με σχετικά απλοϊκές διατυπώσεις γραμμικού προγραμματισμού και χωρίς το όφελος μιας υλοποίησης διακλαδώσεων και διάτμησης; η προσέγγιση φαίνεται να είναι πολλά υποσχόμενη.

3.1.3 Επίλυση του προβλήματος του περιοδεύοντος πωλητή με χρονικά παράθυρα με χρήση υβριδικών δικτύων δεικτών με χρονικά χαρακτηριστικά.

Το TSPTW έχει μια ποικιλία τρόπων επίλυσης, που κυμαίνονται από τεχνικές ακριβούς μαθηματικού προγραμματισμού για προβλήματα μικρού μεγέθους έως ευρετικές προσεγγίσεις. Τεχνικές αθέρατου και δυναμικού προγραμματισμού έχουν χρησιμοποιηθεί σε ακριβείς προσεγγίσεις για την επίλυση του TSPTW.

Όπως αναφέρθηκε και ανωτέρω οι Χριστοφίδης κ.ά. και ο Baker [9, 12] ανέπτυξαν αλγόριθμους διακλάδωσης και σύνδεσης για προβλήματα με έως και 50 κόμβους που χρειάζονται μετρίως στενά χρονικά παράθυρα. Dumas, Y., et al. [10] χρησιμοποιούν τεχνικές μείωσης χώρου κατάστασης για να επεκτείνουν προηγούμενες προσεγγίσεις δυναμικού προγραμματισμού, επιτρέποντάς τους να λύνουν προβλήματα με έως και 200 πελάτες.

Σε μια εναλλακτική προσέγγιση, οι Pesant et al. [32] Χρησιμοποίησαν προγραμματισμό περιορισμών για να αναπτύξουν μια ακριβή μέθοδο και μια ευρετική για το TSPTW.

Οι Qiang Ma et al. [33] χρησιμοποίησαν RL (reinforcement learning - RL) για να εκπαιδεύσουν ένα pointer δίκτυο με είσοδο γραφήματος [34] για να αντιμετωπιστεί αυτό το πρόβλημα. Ωστόσο, η προσέγγισή τους εξακολουθεί να είναι μη ικανή να καθορίσει τη βέλτιστη λύση για ένα ευρύ φάσμα μεγεθών προβλημάτων, καθώς η μελέτη τους καλύπτει μόνο έως και 20 σημεία.

Στην μελέτη [35] αναπτύχθηκε ένα μοντέλο deep learning που εκπαιδεύτηκε χρησιμοποιώντας reinforcement learning για την επίλυση του προβλήματος TWTSP. Επαναπροσδιορίστηκαν από την αρχή η κατάσταση του συστήματος, τα σχήματα ανταμοιβής και την πολιτική κάλυψης για το TWTSP με βάση την αρχιτεκτονική που περιγράφεται στο [36]. Ακολουθήθηκε η ίδια προσέγγιση όπως στο [36], ωστόσο το παρουσιάζεται πρόβλημα όταν χρειάζεται να προσομοιωθούν τα δεδομένα για εκπαίδευση.

Δεδομένου ότι έχουμε να κάνουμε με ένα περιορισμένο πρόβλημα, δεν έχουμε καμία εγγύηση ότι τα δεδομένα που δημιουργούνται έχουν μια εφικτή λύση. Οι Qiang Ma [33] αντιμετώπισαν αυτό το ζήτημα βρίσκοντας πρώτα μια χωρίς περιορισμούς διαδρομή TSP χρησιμοποιώντας την τοπική αναζήτηση με τον αλγόριθμο 2-opt και στη συνέχεια δημιουργώντας το χρονικό παράθυρο για τα λυμένα δεδομένα, αλλά υπάρχουν δύο ζητήματα εδώ:

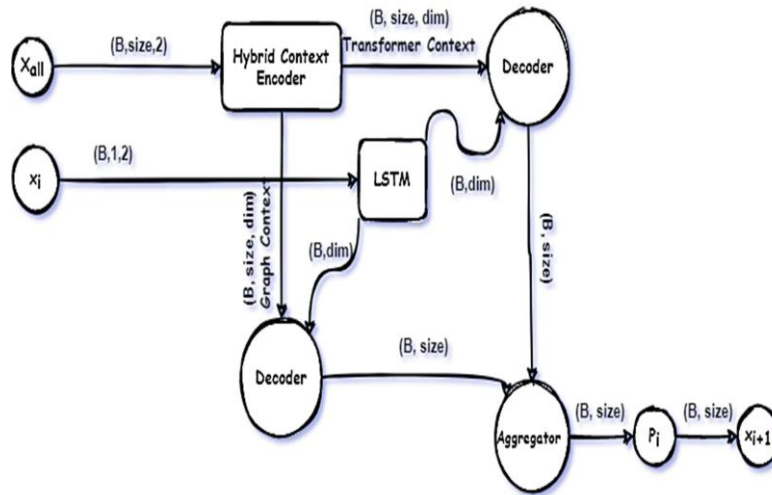
1. Η μέθοδος 2-opt δεν είναι ακριβής αλγόριθμος, επομένως τα δεδομένα που δημιουργούνται δεν θα επιλυθούν βέλτιστα.
2. Αυτή η μέθοδος δεν εκμεταλλεύεται την παραλληλοποίηση της GPU, επομένως η εκπαίδευση του μοντέλου θα είναι εξαιρετικά αργή.

Στην μελέτη [35] αντιμετωπίστηκαν αυτά τα ζητήματα αλλάζοντας τον τρόπο παραγωγής των δεδομένων. Αντί να χρησιμοποιήσουμε ένα μοντέλο 2 επιλογών για την επίλυση του χωρίς περιορισμούς TSP, χρησιμοποιήσαμε το δίκτυο υβριδικών δεικτών HPN [36] καθώς δίνει μια πιο κοντά στη βέλτιστη λύση για δεδομένα που κυμαίνονται από 20 έως 100 σημεία. Η χρήση ενός HPN που είναι προεκπαιδευμένο για την επίλυση μη TSP (ανεφ περιορισμών) θα μας επιτρέψει να εκμεταλλευτούμε τη GPU, επομένως η εκπαίδευση δεν είναι αργή. Συμπληρώνοντας αυτά τα κενά και τροποποιώντας το HPN ώστε να μπορεί να κωδικοποιεί χρονικά χαρακτηριστικά, επιλύονται έως και 100 σημεία στο TWTSP, ενώ στην μελέτη [33] επιλύονται μόνο 20 σημεία. Επομένως, πρέπει να τονίσουμε ότι το μοντέλο που χρησιμοποιείται για τη δημιουργία δεδομένων δεν είναι το ίδιο με το μοντέλο που χρησιμοποιείται για την εκπαίδευση του TWTSP.

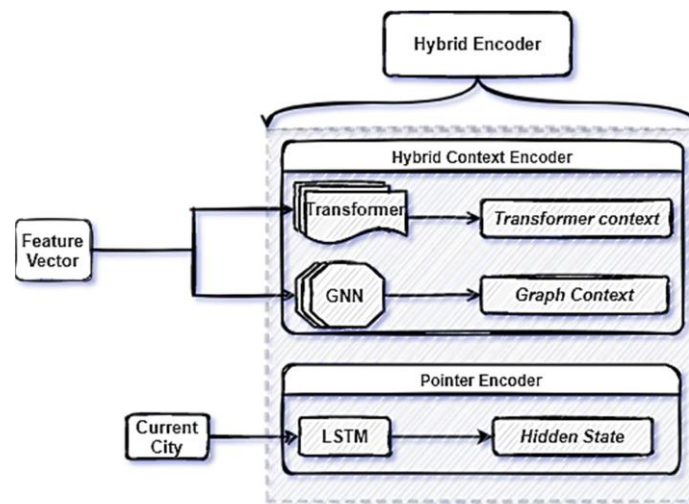
Όπως περιεγράφηκε στο [36], το προτεινόμενο μοντέλο είναι μια παραλλαγή των pointer δικτύων [34]. Σε σύγκριση με το GPN, το προτεινόμενο μοντέλο έχει δύο επιπλέον στοιχεία: έναν υβριδικό κωδικοποιητή περιβάλλοντος και έναν αποκωδικοποιητή πολλαπλής προσοχής.

Ο υβριδικός κωδικοποιητής αποτελείται κυρίως από τον κωδικοποιητή του μετασηματιστή και τον κωδικοποιητή γραφήματος. Αυτοί οι κωδικοποιητές κωδικοποιούν το διάγραμμα χαρακτηριστικών πριν το περάσουν στο επίπεδο

αποκωδικοποιητή προσοχής. Επομένως, υπάρχουν δύο αποκωδικοποιητές στο στάδιο της αποκωδικοποίησης, ένας για το περιβάλλον του μετασχηματιστή και ένας για το περιβάλλον του γραφήματος. Τέλος, τα διανύσματα προσοχής εξόδου από τους δύο αποκωδικοποιητές συγκεντρώνονται πριν από την επιλογή του επόμενου κόμβου στην περιήγηση. Το προτεινόμενο HPN φαίνεται στην εικόνα 3.2.



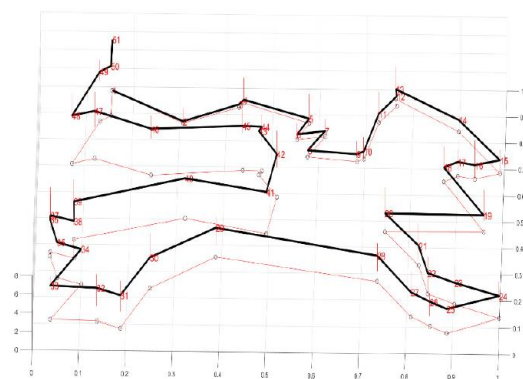
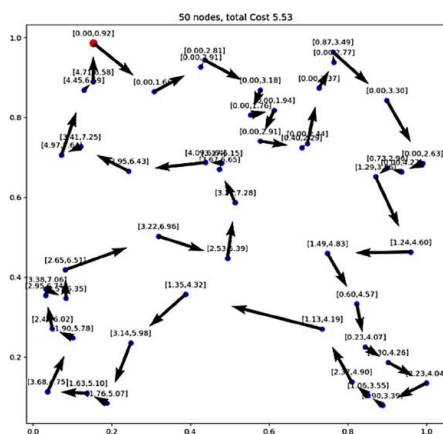
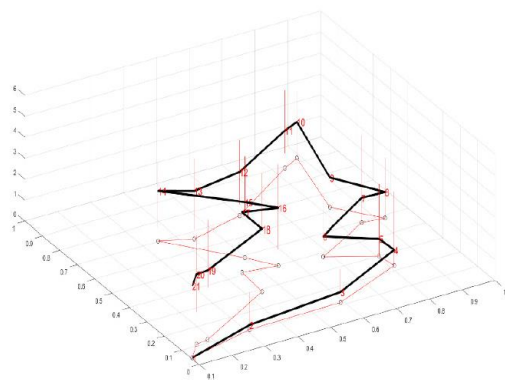
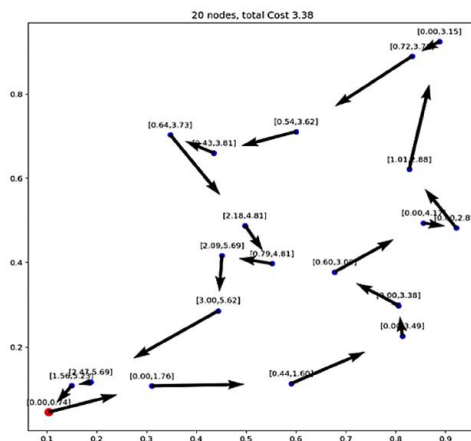
Εικόνα 3.2. Αρχιτεκτονική HPN.[35]

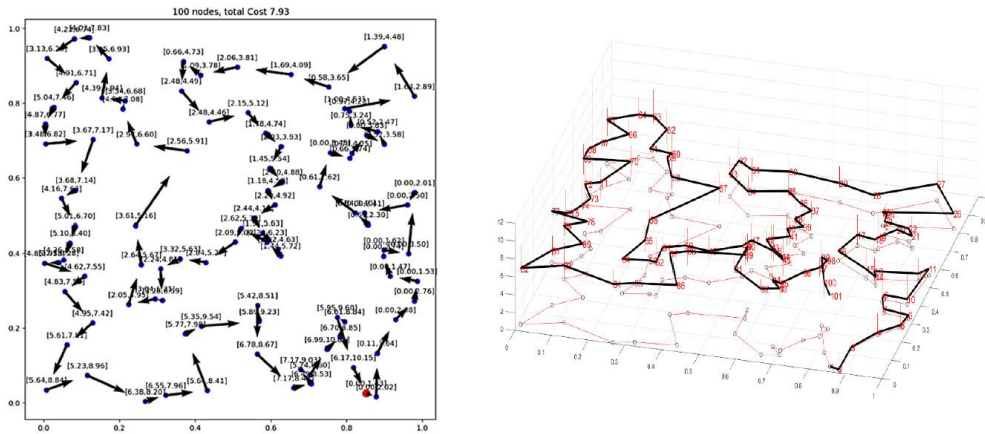


Εικόνα 3.3. Υβριδικός κωδικοποιητής.[35]

Άρα συγκεντρωτικά σε αυτήν την μελέτη [35], αναπτύχθηκε ένα μοντέλο βασισμένο σε τεχνολογία reinforcement learning για να επιλυθεί το πρόβλημα του περιοδεύοντος πωλητή με τα χρονικά παράθυρα. Το μοντέλο που αναπτύχθηκε βασίζεται στη χρήση διαφορετικών κωδικοποιητών και αποκωδικοποιητών για να είναι σε θέση να μοντελοποιήσει καλύτερα την κοινή κατανομή μεταξύ των χαρακτηριστικών του προβλήματος και η λύση.

Συγκρίθηκε το προτεινόμενο μοντέλο με το GPN και αποδείχθηκε ότι το μοντέλο της μελέτης αποδίδει καλύτερες λύσεις και μπορεί να λύσει μεγαλύτερου μεγέθους περιπτώσεις. Το μοντέλο εξακολουθεί να μην είναι ικανό να γενικεύσει δύο πολύ μεγάλες περιπτώσεις προβλημάτων, π.χ., το TWTSP500 και TWTSP1000.





Σχήμα 3.4. Δείγματα περιηγήσεων TWTSP20, TWTSP50 και TWTSP100 που επιλύθηκαν από την HPN.[35]

3.1.4 Επίλυση του προβλήματος του περιοδεύοντος πωλητή με χρονικά παράθυρα με χρήση Δυναμικής Διακριτοποίησης

Στην μελέτη των Duc Minh Vu [37] παρουσιάζεται ένας αλγόριθμος για την επίλυση του TSPTW ο οποίος έχει δύο διαφορετικούς στόχους. Ο πρώτος στόχος είναι να επιστρέψει το όχημα στην αποθήκη το συντομότερο δυνατό μετά την έναρξη του ορίζοντα σχεδιασμού. Αυτός ο στόχος αναφέρεται συχνά ως στόχος makespan (χρονικό διάστημα που μεσολαβεί από την έναρξη των εργασιών μέχρι το τέλος). Έχει λάβει πρόσφατη προσοχή στη βιβλιογραφία. βλέπε, για παράδειγμα, Arigliano et al. (2015) [38] και Montero et al. (2017) [39].

Ο δεύτερος στόχος είναι το όχημα να περνά όσο το δυνατόν λιγότερο χρόνο μακριά από την αποθήκη, που αναφέρεται ως στόχος διάρκειας.

Ο αλγόριθμος που προτείνεται στην μελέτη [37] βασίζεται στο πλαίσιο Dynamic Discretization Discovery (DDD) που εισήχθη από τους Boland et al. (2017a) [40], και εφαρμόστηκε στο Πρόβλημα Σχεδιασμού Δικτύου Υπηρεσιών. Αν και το

πλαίσιο έχει αποδειχθεί αποτελεσματικό, δεν έχει ακόμη προσαρμοστεί σε προβλήματα με τους χρόνους ταξιδιού που εξαρτώνται από το χρόνο.

Είναι γεγονός σήμερα ότι πολλές εταιρείες να προσφέρουν όλο και πιο αυστηρές υποσχέσεις χρόνου παράδοση και είναι σε θέση να χειρίζονται (αποτελεσματικά) χρόνους ταξιδιού που ποικίλλουν κατά τη διάρκεια της ημέρας. Έτσι ο προγραμματισμός των διαδρομών παράδοσης γίνεται κρίσιμος.

Η αστικοποίηση, με τα Ηνωμένα Έθνη να προβλέπουν ότι έως το 2050 τα δύο τρίτα του παγκόσμιου πληθυσμού θα ζουν σε πόλεις (ΟΗΕ 2014) [41], απλώς επιδεινώνει το πρόβλημα, καθώς οδηγεί σε αύξηση των παραδόσεων (είτε σε ιδιώτες είτε σε πωλητές λιανικής) σε πυκνά κατοικημένες περιοχές όπου η κυκλοφοριακή συμφόρηση και οι διακυμάνσεις του χρόνου ταξιδιού είναι συχνές. Αυτές οι τάσεις προτρέπουν την έρευνα για προβλήματα προγραμματισμού μεταφορών που εξηγούν ρητά τους χρόνους ταξιδιού που εξαρτώνται από το χρόνο (Figliozzi, Gendreau et al. 2015) [42,43].

Η κύρια συνεισφορά της έρευνας που παρουσιάζεται σε αυτή την εργασία [37] είναι μια νέα μέθοδος λύσης για το TD-TSPTW, η οποία ξεπερνά τις πιο γνωστές μεθόδους σε δύο σύνολα περιπτώσεων αναφοράς. Φαίνεται περαιτέρω ότι η μέθοδος μπορεί εύκολα να τροποποιηθεί για να επιλύσει αποτελεσματικά αυτές τις περιπτώσεις για έναν στόχο διάρκειας.

Μια δευτερεύουσα συνεισφορά είναι ότι η έρευνα δείχνει ότι το πλαίσιο DDD μπορεί να χρησιμοποιηθεί για την ανάπτυξη ενός αλγορίθμου που μπορεί να χειριστεί (λίγα) προβλήματα με χρόνους ταξιδιού που εξαρτώνται από το χρόνο, γεγονός που υποδηλώνει ότι το πλαίσιο DDD μπορεί να χρησιμοποιηθεί για την ανάπτυξη αποτελεσματικών αλγορίθμων για άλλους σχεδιασμούς προβλημάτων μεταφορών στα οποία οι χρόνοι ταξιδιού εξαρτώνται από το χρόνο. Ενώ η υπολογιστική απόδοση των υπάρχουσών μεθόδων για το TD-TSPTW είναι εξαιρετικά ευαίσθητη στη συχνότητα με την οποία αλλάζουν οι χρόνοι διαδρομής, τα αποτελέσματά μας δείχνουν ότι η νέα μέθοδος λύσης δεν είναι. Αυτά τα αποτελέσματα υπογραμμίζουν ένα βασικό πλεονέκτημα του πλαισίου DDD, το οποίο βασίζεται σε διαμορφώσεις

προγραμματισμού ακέραιων αριθμών που ορίζονται σε (μερικώς) διευρυμένα δίκτυα χρόνου: η χρονική εξάρτηση των χρόνων ταξιδιού μπορεί να ενσωματωθεί άμεσα και φυσικά σε ένα διευρυμένο δίκτυο χρόνου.

Στην [37] γίνεται μελέτη μια περίπτωσης στην οποία ένα όχημα αναχωρεί από μια αρχική τοποθεσία, που ονομάζεται αποθήκη, επισκέπτεται κάθε τοποθεσία σε ένα γνωστό σύνολο τοποθεσιών ακριβώς μία φορά και κατά τη διάρκεια ενός προκαθορισμένου χρονικού παραθύρου και στη συνέχεια επιστρέφει στην αποθήκη. Το πρόβλημα του προσδιορισμού ενός συνόλου τέτοιων κινήσεων, γνωστό ως περιήγηση, αναφέρεται ως Πρόβλημα Περιοδευόντος Πωλητή με Παράθυρα Χρόνου (TSPTW). Λαμβάνετε υπόψη καταστάσεις στις οποίες ο χρόνος κατά τον οποίο συμβαίνει το ταξίδι επηρεάζει το χρόνο ταξιδιού, με αποτέλεσμα ένα πρόβλημα που ονομάζεται Πρόβλημα εξαρτώμενου από τον χρόνο ταξιδιώτη πωλητή με τα παράθυρα χρόνου (TD-TSPTW).

Όπως γίνεται συνήθως η μελέτη υποθέτει ότι οι χρόνοι ταξιδιού συμμορφώνονται με την ιδιότητα First-In, First-Out (FIFO), που σημαίνει ότι μεταξύ δύο τοποθεσιών, μια μεταγενέστερη αναχώρηση δεν μπορεί να οδηγήσει σε νωρίτερα άφιξη και ότι ο χρόνος ταξιδιού είναι κατά τμήματα γραμμική εξίσωση της ώρας αναχώρησης. Αρχικά επιδιώκουμε να ελαχιστοποιήσουμε έναν στόχο makespan, ο οποίος επιδιώκει μια περιήγηση που μπορεί να ολοκληρωθεί, ώστε να επιστρέψουμε στην αποθήκη όσο το δυνατόν νωρίτερα μετά την έναρξη του ορίζοντα σχεδιασμού. Κάτω από αυτόν τον στόχο και με την ιδιότητα FIFO, υπάρχει μια βέλτιστη λύση στην οποία η μόνη αναμονή του οχήματος, εάν υπάρχει, είναι όταν φτάνει σε μια τοποθεσία πριν ανοίξει το χρονικό παράθυρο της τοποθεσίας.

Σύμφωνα με αυτόν τον στόχο, ο χρόνος αναχώρησης από την αποθήκη είναι ευέλικτος και η διάρκεια της περιήγησης μπορεί να ελαχιστοποιηθεί με την αναχώρηση από το την αποθήκη κάποια στιγμή μετά την έναρξη του ορίζοντα σχεδιασμού.

Το TD-TSPTW ορίζεται ως εξής. Έστω (N, A) υποδηλώνουν ένα κατευθυνόμενο γράφημα, όπου ο κόμβος έχει οριστεί $N = \{0, 1, 2, \dots, n\}$ και περιλαμβάνει την αποθήκη (κόμβος 0) καθώς και το σύνολο των τοποθεσιών (κόμβοι $1, \dots, n$) που πρέπει να γίνει επίσκεψη.

Συνδεδεμένη με κάθε τοποθεσία $i = 1, \dots, n$ είναι ένα χρονικό παράθυρο, $[e_i, l_i]$, κατά την οποία πρέπει να επισκεφθείτε την τοποθεσία. Σημειώστε ότι το όχημα μπορεί να φτάσει στην τοποθεσία $i \in N \setminus \{0\}$ πριν από την ώρα ανοίγματος του παραθύρου, το οποίο σημαίνει, οπότε πρέπει να περιμένει μέχρι να ανοίξει το παράθυρο χρόνου. Υπάρχει επίσης ένα χρονικό παράθυρο, $[e_0, l_0]$, που σχετίζεται με την αποθήκη, το οποίο υποδεικνύει ότι η περιήγηση μπορεί να αναχωρήσει από την αποθήκη όχι νωρίτερα από το e_0 και πρέπει να επιστρέψει στην αποθήκη το αργότερο στις l_0 .

Το σετ $A \subseteq N \times N$ αποτελείται από τόξα που αντιπροσωπεύουν τη διαδρομή μεταξύ τοποθεσιών. Συνδεδεμένοι με κάθε τόξο τα $(i, j) \in A$ είναι μια τμηματικά γραμμική συνάρτηση χρόνου ταξιδιού, $t_{ij}(t)$, που δίνει το χρόνο διαδρομής κατά μήκος του τόξου (i, j) αν η διάβαση του (i, j) αρχίζει την ώρα t . Κάθε συνάρτηση χρόνου ταξιδιού έχει έναν πεπερασμένο αριθμό σημείων διακοπής, (γραμμικά κομμάτια) και ικανοποιεί την ιδιότητα FIFO, δηλαδή, για όλα $(i, j) \in A$ και για χρόνους t, t_i με $t \leq t_i$, πρέπει $t + t_{ij}(t) \leq t_i + t_{ij}(t_i)$. Αυτό είναι ισοδύναμο με την απαίτηση όλα τα γραμμικά κομμάτια να έχουν κλίση τουλάχιστον -1 . Έτσι, τυπικά, μια εφικτή λύση στο TD-TSPTW είναι μια ακολουθία ζευγών κόμβου-χρόνου, $((i_0, t_0), (i_1, t_1), \dots, (i_n, t_n), (i_{n+1}, t_{n+1}))$, που ικανοποιεί $i_0 = i_{n+1} = 0$, $\{i_1, \dots, i_n\} = \{1, \dots, n\}$, $t_0 \geq e_0$, $t_{k+1} \geq \max\{e_{k+1}, t_k + t_{ik, ik+1}(t_k)\}$ για όλα τα $k = 0, \dots, n$, και $t_{n+1} \leq l_0$. Ο στόχος makespan ελαχιστοποιεί το t_{n+1} καθώς ο στόχος διάρκειας ελαχιστοποιείται $t_{n+1} - t_0$.

Διατυπώνεται το TD-TSPTW ως ένα ακέραιο πρόγραμμα που ορίζεται σε ένα διευρυμένο κατά το χρόνο δίκτυο, $D = (N, A)$ με πεπερασμένο σύνολο κόμβων N και πεπερασμένο σύνολο τόξων A . Το ότι υπάρχει ένα τέτοιο ακέραιο πρόγραμμα βασίζεται στην παρατήρηση ότι υπάρχει ένας πεπερασμένος αριθμός εφικτών περιηγήσεων, ότι κάθε συνάρτηση χρόνου διαδρομής τόξου έχει έναν πεπερασμένο

αριθμό σημείων διακοπής και ότι υπάρχει μια βέλτιστη περιήγηση που αναχωρεί από τουλάχιστον μία τοποθεσία σε ένα σημείο διακοπής, πράγμα που συνεπάγεται ότι υπάρχει πεπερασμένος χρόνος διευρυμένος δίκτυο που περιέχει μια βέλτιστη λύση.

Επομένως, έστω το N περιέχει κόμβους (i, t) για $i \in N$ και $t \in T_i$, το (πεπερασμένο) σύνολο χρονικών σημείων στην τοποθεσία i συμπεριλαμβανομένου e_i και l_i . Επιπλέον, έστω το A περιέχει τόξα της μορφής $((i, t), (j, t_j))$ με $i = j$, $(i, j) \in A$, $t \in T_i$, $t_j \in T_j$ και $t_j = \max\{e_j, t + t_{ij}(t)\}$ (το τόξο ενσωματώνει τον απαραίτητο χρόνο αναμονής) και $t_j \leq l_j$ (το όχημα δεν μπορεί να φτάσει αργά).

Για να διατυπωθεί το ακέραιο πρόγραμμα, για κάθε τόξο $((i, t), (j, t_j)) \in A$ ορίζεται η δυαδική μεταβλητή $x_{((i,t),(j,t_j))}$ για να υποδείξει εάν το όχημα κινείται ή όχι κατά μήκος αυτού του τόξου.

Καθώς βελτιστοποιούνται δύο διαφορετικοί στόχοι σε αυτό το άρθρο, παρουσιάζεται η ακόλουθη διατύπωση του TD-TSPTW που επιδιώκει να βελτιστοποιήσει μια γενική αντικειμενική συνάρτηση.

$$z = \text{minimize} \quad \sum_{((i,t),(j,t')) \in A} c_{ij}(t) x_{((i,t),(j,t'))}$$

και

$$\begin{aligned} \sum_{((i,t),(j,t')) \in A: i \neq j} x_{((i,t),(j,t'))} &= 1, \quad \forall j \in N, \\ \sum_{((i,t),(j,t')) \in A} x_{((i,t),(j,t'))} - \sum_{((j,\tilde{t}),(i,t)) \in A} x_{((j,\tilde{t}),(i,t))} &= 0, \quad \forall (i, t) \in \mathcal{N}, i \neq 0 \\ x_{((i,t),(j,t'))} &\in \{0, 1\}, \quad \forall ((i, t), (j, t')) \in A. \end{aligned}$$

Οι περιορισμοί (1) διασφαλίζουν ότι το όχημα φτάνει σε κάθε θέση ακριβώς μία φορά κατά τη διάρκεια του χρονικού του παραθύρου. Οι περιορισμοί (2) διασφαλίζουν στη συνέχεια ότι το όχημα αναχωρεί από κάθε κόμβο στον οποίο φτάνει (εκτός από την περίπτωση που επιστρέφει στην αποθήκη). Τέλος, οι

περιορισμοί (3) ορίζουν τους τομείς των μεταβλητών απόφασης. Ονομάζουμε αυτή τη διατύπωση ως TD-TSPTW(D) για να επισημάνουμε τη χρήση ενός δικτύου διευρυμένου χρόνου.

Για να αντιπροσωπεύσουμε τον στόχο makespan, τέθηκε ότι $c_{ij}(t) = 0$ για όλα τα $((i, t), (j, t)) \in A, j \neq 0$ και $c_{i0}(t) = t + \tau_{i0}(t)$ για όλα $((i, t), (0, t)) \in A$ για να υποδειχθεί ότι τα έξοδα πραγματοποιούνται μόνο κατά την επιστροφή στην αποθήκη. Σημειώστε ότι το $c_{ij}(t)$ είναι, σε αυτή την περίπτωση, τόσο μη αρνητικό όσο και μη φθίνον στο t . Για την αντιπροσώπευση του στόχου διάρκειας, η συνάρτηση κόστους κάλυψης τροποποιείται με το εξής: $c_{0j}(t) := -t$ for all $((0, t), (j, t)) \in A$, που είναι μη θετικό και μη αυξανόμενο. Σημειώστε ότι ο αντίκτυπος των χρονικών παραθύρων και των χρόνων ταξιδιού που εξαρτώνται από το χρόνο στην εφικτή περιοχή του TD-TSPTW είναι ενσωματωμένοι στο δίκτυο διευρυμένου χρόνου, D . Το ότι το όχημα πρέπει να αναχωρήσει από μια τοποθεσία εντός του χρονικού παραθύρου του αντιμετωπίζεται διασφαλίζοντας ότι το A δεν περιέχει τόξα που φτάνουν σε έναν κόμβο έξω από το χρονικό παράθυρο της τοποθεσίας του. Οι χρόνοι ταξιδιού που εξαρτώνται από το χρόνο είναι ενσωματωμένοι στα τόξα, $((i, t), (j, t)) \in A$, μέσω της επιλογής του t .

Άρα συγκεντρωτικά στην μελέτη [39], παρουσιάστηκε ένας αλγόριθμος, DDD-TD-TSPTW, που βασίζεται στο πλαίσιο Dynamic Discretization Discovery (DDD) που μπορεί να λύσει περιπτώσεις του TD-TSPTW όταν επιθυμούμε βελτίωση δύο στόχων:

(1) Στόχος makespan και

(2) στόχος διάρκειας.

Αυτός ο αλγόριθμος διαφέρει από τις υπάρχουσες εφαρμογές του πλαισίου DDD στο ότι μπορεί να φιλοξενήσει χρόνους ταξιδιού που εξαρτώνται από το χρόνο. Επιπλέον, οι τεχνικές που διέπουν αυτόν τον αλγόριθμο μπορούν να χρησιμοποιηθούν για την εφαρμογή DDD σε άλλα προβλήματα προγραμματισμού μεταφοράς στα οποία οι χρόνοι ταξιδιού εξαρτώνται από το χρόνο. Τα αποτελέσματα της υπολογιστικής

μελέτης σε περιπτώσεις αναφοράς υποδεικνύουν ότι ο αλγόριθμος που έχει καλύτερη απόδοση από τον αλγόριθμο με την καλύτερη απόδοση στη βιβλιογραφία. Επίσης σε αντίθεση με τις υπάρχουσες μεθόδους, ο αλγόριθμος είναι ισχυρός σε σχέση με όλες τις παραμέτρους των περιπτώσεων, ιδιαίτερα τον βαθμό μεταβλητότητας του χρόνου ταξιδιού.

Στην περίπτωση του στόχου διάρκειας, η αναλογία του αριθμού των κόμβων στον τελικό κόμβο με επέκταση χρόνου προς τον αριθμό των κόμβων στο αρχικό διευρυμένο χρονικά δίκτυο είναι κατά προσέγγιση η ίδια με την αναλογία για τον στόχο makespan,.

Η επιτυχία του DDD-TD-TSPTW οφείλεται, εν μέρει, στο γεγονός ότι ο συνδυασμός ενός αντικειμενικού στόχου και των χρόνων ταξιδιού που ικανοποιούν την ιδιοκτησία FIFO, συνεπάγεται ότι το κόστος ταξιδιού δεν μειώνεται σε t , πράγμα που σημαίνει ότι το $c_{ij}(t)$ δεν είναι ποτέ υποτιμητικό, αλλά πάντα ακριβές. Για άλλους στόχους, π.χ. ελαχιστοποίηση του συνολικού χρόνου ταξιδιού, αυτό δεν ισχύει πλέον και απαιτούνται πρόσθετες τεχνικές για να διασφαλιστεί η αποδεκτή υπολογιστική απόδοση.

Κεφάλαιο 4^ο

4.1 Συμπεράσματα

Στην παρούσα διπλωματική εργασία έγινε συγκριτική μελέτη αλγορίθμων οι οποίοι χρησιμοποιούνται για την επίλυση του προβλήματος πλανόδιου πωλητή με χρονικά παράθυρα.

Αρχικά η κλασική μορφή του προβλήματος αναλύθηκε στο πρώτο κεφάλαιο και ακολούθως εισήχθη ο περιορισμός των χρονικών παραθύρων. Το πρόβλημα πλέον με αλλάζει μορφή άρα και τρόπο επίλυσης.

Η βασική δομή του προβλήματος με χρονικά παράθυρα αναλύεται στο δεύτερο κεφάλαιο και διατυπώνεται η μαθηματική έκφραση τόσο του συμμετρικού αλλά και του ασύμμετρου προβλήματος του περιοδεύοντος πωλητή με χρονικά παράθυρα.

Στην συνέχεια ακολουθεί βιβλιογραφική ανασκόπηση κατά την οποία επιλέχθηκαν τρεις διαφορετικές μελέτες (άρα και τρεις διαφορετικοί τρόποι επίλυσης του προβλήματος με χρονικά παράθυρα) που έχουν δημοσιοποιηθεί στην βιβλιογραφία αναλύονται. Οι μελέτες που αναλύθηκαν είναι οι εξής:

- Επίλυση του προβλήματος του περιοδεύοντος πωλητή με χρονικά παράθυρα με χρήση λαγκραντζιανής χαλάρωσης
- Επίλυση του προβλήματος του περιοδεύοντος πωλητή με χρονικά παράθυρα με χρήση του χρονικά διευρυμένων δικτύων
- Επίλυση του προβλήματος του περιοδεύοντος πωλητή με χρονικά παράθυρα με χρήση υβριδικών δικτύων δεικτών με χρονικά χαρακτηριστικά.
- Επίλυση του προβλήματος του περιοδεύοντος πωλητή με χρονικά παράθυρα με χρήση Δυναμικής Διακριτοποίησης

Στην πρώτη μελέτη [27] παρουσιάστηκε μια μεταερευτική μεθοδολογία βασισμένη στην τεχνική λαγκραντζιανή χαλάρωση. Αυτό το σχήμα έχει χρησιμοποιηθεί για την αντιμετώπιση του προβλήματος του ταξιδιώτη πωλητή.

Η προτεινόμενη μέθοδος που βασίζεται χρησιμοποιεί τον αλγόριθμο SO σε συνδυασμό με μια ευρετική. Με στόχο τη βελτίωση της σύγκλισης του αλγορίθμου στη βέλτιστη λύση, εισάγεται η ευρετική προκειμένου να ληφθεί μια εφικτή λύση από τη διπλή μεταβλητή.

Από τη μία πλευρά, η μέθοδος παρέχει τόσο άνω όριο όσο και κάτω όριο επομένως έναν μεταγενέστερο ποιοτικό έλεγχο της λύσης που προκύπτει. Από την άλλη πλευρά, μειώνει τον χώρο αναζήτησης, καθώς μπορούν να χρησιμοποιηθούν διπλές πληροφορίες για την περικοπή μεταβλητών απόφασης. Τέλος, παρά το ότι η βέλτιστη δεν μπορεί να επιτευχθεί πάντα, η προτεινόμενη μέθοδος είναι σε θέση να παρέχει μια εφικτή λύση με ένα στενό χάσμα μεταξύ του αρχικού και του βέλτιστου κόστους σε λογικό αριθμό επαναλήψεων. Πρέπει να σημειωθεί ότι υπάρχουν αρκετά σημεία για μελλοντική έρευνα. Πρώτον, οι παράμετροι που ορίστηκαν στην μελέτη ρ και αL πρέπει να ρυθμιστούν με διαδικασίες μικρορύθμισης. Δεύτερον, δεδομένου ότι η παράμετρος L^- είναι μια καλή εκτίμηση, η ευρετική για να ληφθεί μια εφικτή λύση από μια διπλή πρέπει να βελτιωθεί προκειμένου να μειωθεί ο χρόνος υπολογισμού της. Τέλος, ο παρουσιαζόμενος αλγόριθμος έχει συμπεριληφθεί σε ένα πλαίσιο αναζήτησης μεταβλητού γειτονικού κόμβου για την αντιμετώπιση του προβλήματος δρομολόγησης χωρητικότητας οχημάτων, δείχνοντας πολύ καλά αποτελέσματα τόσο ως προς την ποιότητα της λύσης όσο και ως προς την υπολογιστική απόδοση.

Στην δεύτερη μελέτη [35], αναπτύχθηκε ένα μοντέλο βασισμένο σε τεχνολογία reinforcement learning για να επιλυθεί το πρόβλημα του περιοδεύοντος πωλητή με τα χρονικά παράθυρα. Το μοντέλο που αναπτύχθηκε βασίζεται στη χρήση διαφορετικών κωδικοποιητών και αποκωδικοποιητών για να είναι σε θέση να μοντελοποιήσει καλύτερα την κοινή κατανομή μεταξύ των χαρακτηριστικών του προβλήματος και η λύση.

Συγκρίθηκε το προτεινόμενο μοντέλο με το GPN και αποδείχθηκε ότι το μοντέλο της μελέτης αποδίδει καλύτερες λύσεις και μπορεί να λύσει μεγαλύτερου μεγέθους περιπτώσεις. Το μοντέλο εξακολουθεί να μην είναι ικανό να γενικεύσει δύο πολύ μεγάλες περιπτώσεις προβλημάτων, π.χ., το TWTSP500 και TWTSP1000.

Στην επόμενη μελέτη [28] δοκιμάστηκαν οκτώ παραλλαγές του αλγορίθμου Solve-TSPTW στις 135 περιπτώσεις του Dumas (1995) [10]. Σχεδόν όλες οι περιπτώσεις θα μπορούσαν να επιλυθούν με λίγες μόνο επαναλήψεις του αλγορίθμου. Ακόμη και η παραλλαγή αλγορίθμου με τη χειρότερη απόδοση έλυσε σχεδόν το 75% των περιπτώσεων, ενώ η καλύτερη έλυσε λίγο λιγότερο από το 95%, με λιγότερες από 10,2 επαναλήψεις, κατά μέσο όρο. Δεδομένου ότι αυτά τα αποτελέσματα προέκυψαν με σχετικά απλοϊκές διατυπώσεις γραμμικού προγραμματισμού και χωρίς το όφελος μιας υλοποίησης διακλαδώσεων και διάτμησης; η προσέγγιση φαίνεται να είναι πολλά υποσχόμενη.

Πίνακας 4.1 Αποτελέσματα μελέτης [28]

Αλγόριθμος	Αριθμός που επιλύθηκε	Μέσος χρόνος ολοκλήρωσης (s)	Μέσος αριθμός επαναλήψεων
BASIC	110	2066	18.42
BASIC+PRE	116	1307	18.19
BASIC+PRE+ST	128	794	10.14
BASIC+PRE+ST+PTH	125	779	9.70
F+PRE	114	1009	4.96
F+PRE+PTH	119	709	4.97
TDF+PRE	105	1073	4.50
TDF+PRE+PTH	101	891	4.46

Τέλος στην τελευταία μελέτη [39], παρουσιάστηκε ένας αλγόριθμος, DDD-TD-TSPTW, που βασίζεται στο πλαίσιο Dynamic Discretization Discovery (DDD) που μπορεί να λύσει περιπτώσεις του TD-TSPTW όταν επιθυμούμε βελτίωση δύο στόχων:

- Στόχος makespan

➤ στόχος διάρκειας.

Αυτός ο αλγόριθμος διαφέρει από τις υπάρχουσες εφαρμογές του πλαισίου DDD στο ότι μπορεί να φιλοξενήσει χρόνους ταξιδιού που εξαρτώνται από το χρόνο. Επιπλέον, οι τεχνικές που διέπουν αυτόν τον αλγόριθμο μπορούν να χρησιμοποιηθούν για την εφαρμογή DDD σε άλλα προβλήματα προγραμματισμού μεταφοράς στα οποία οι χρόνοι ταξιδιού εξαρτώνται από το χρόνο. Τα αποτελέσματα της υπολογιστικής μελέτης σε περιπτώσεις αναφοράς υποδεικνύουν ότι ο αλγόριθμος που έχει καλύτερη απόδοση από τον αλγόριθμο με την καλύτερη απόδοση στη βιβλιογραφία. Επίσης σε αντίθεση με τις υπάρχουσες μεθόδους, ο αλγόριθμος είναι ισχυρός σε σχέση με όλες τις παραμέτρους των περιπτώσεων, ιδιαίτερα τον βαθμό μεταβλητότητας του χρόνου ταξιδιού.

Στην περίπτωση του στόχου διάρκειας, η αναλογία του αριθμού των κόμβων στον τελικό κόμβο με επέκταση χρόνου προς τον αριθμό των κόμβων στο αρχικό διευρυμένο χρονικά δίκτυο είναι κατά προσέγγιση η ίδια με την αναλογία για τον στόχο makespan,.

Η επιτυχία του DDD-TD-TSPTW οφείλεται, εν μέρει, στο γεγονός ότι ο συνδυασμός ενός αντικειμενικού στόχου και των χρόνων ταξιδιού που ικανοποιούν την ιδιοκτησία FIFO, συνεπάγεται ότι το κόστος ταξιδιού δεν μειώνεται σε t , πράγμα που σημαίνει ότι το κόστος υποεκτιμάται. Το $c_{ij}(t)$ δεν είναι ποτέ υποτιμητικό, αλλά πάντα ακριβές. Για άλλους στόχους, π.χ. ελαχιστοποίηση του συνολικού χρόνου ταξιδιού, αυτό δεν ισχύει πλέον και απαιτούνται πρόσθετες τεχνικές για να διασφαλιστεί η αποδεκτή υπολογιστική απόδοση.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1]. R.G. Bland et al.\ Large traveling salesman problems arising experiments in X-ray crystallography: A preliminary report on computation Operations Research Letters (1989)
- [2]. J. K. Lenstra and A. H. G. Rinnooy Kan (1974) Some Simple Applications of the Travelling Salesman Problem
- [3]. H. DONALD Ratliff Arnon Rosenthal (1983) Order-Picking in a Rectangular Warehouse: A Solvable Case of the Traveling Salesman Problem
- [4]. Martin Grötschel (1991) Optimal control of plotting and drilling machines: A case study Technische Universität Berlin
- [5]. Haider A Abdulkarim, Ibrahim F Alshammari (2015) Comparison of Algorithms for Solving Traveling Salesman Problem International Journal of Engineering and Advanced Technology (IJEAT)
- [6]. Manuel Lopez-Ibanez, Christian Blum, Jeffrey W. Ohlmann, Barrett W. Thomas, —The Travelling Salesman Problem with Time Windows: Adapting Algorithms from Travel-time to Makespan Optimization, Appl. Soft Comput. J. 2013
- [7]. Imdat Kara, Ozge Nimet Koc, Fulya Altıparmak & Berna Dengiz, —New integer linear programming formulation for the traveling salesman problem with time windows: minimizing tour duration with waiting times, Optimization: A Journal of Mathematical Programming and Operations Research, 2013
- [8]. John N. Tsitsiklis, —Special Cases of Traveling Salesman and Repairman Problems with Time Windows, NETWORKS, Vol. 22, 1992, pp. 263-282

- [9]. Christofides, N.; Mingozzi, A.; Toth, P. State-space relaxation procedures for the computation of bounds to routing problems. *Networks* 1981, 11, 145–164. [CrossRef]
- [10]. Dumas, Y.; Desrosiers, J.; Gelinas, E.; Solomon, M.M. An Optimal Algorithm for the Traveling Salesman Problem with Time Windows. *Oper. Res.* 1995, 43, 367–371. [CrossRef]
- [11]. Savelsbergh, M. (1985), Local search in routing problems with time windows, *Annals of Operations Research* 4, 285-305.
- [12]. Baker, E.K. Technical Note—An Exact Algorithm for the Time-Constrained Traveling Salesman Problem. *Oper. Res.* 1983, 31, 938–945. [CrossRef]
- [13]. [2] Chi-Bin Cheng, Chun-Pin Mao, —A modified ant colony system for solving the travelling salesman problem with time windows, *Mathematical and Computer Modelling* 46 (2007) pp. 1225–1235, © 2007 Elsevier Ltd
- [14]. [7] Jeffrey W. Ohlmann, Barrett W. Thomas, —Compressed-Annealing Heuristic for the Traveling Salesman Problem with Time Windows, *INFORMS Journal on Computing* Vol. 19 No. 1, pp. 80–90, 2007, ISSN: 1091-9856, eISSN: 1526-5528-07-1901-0080
- [15]. [8] Jing-Quan Li, —A Bi-directional Resource-bounded Dynamic Programming Approach for the Traveling Salesman Problem with Time Windows, Submitted manuscript
- [16]. [10] Kjetil Fagerholt, Marielle Christiansen, —A travelling salesman problem with allocation, time window and precedence constraints-an application to ship scheduling, *International Transactions in Operation Research* 7 (2000), pp. 231-244
- [17]. [12] Martin Desrochers; Jacques Desrosiers; Marius Solomon, —A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows, *Operations Research*, Vol. 40, No. 2. (Mar. - Apr., 1992), pp. 342-354
- [18]. [14] Olli Bräysy, Michel Gendreau, —Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms, *Transportation Science*, Vol. 39, No. 1, February 2005, pp. 104–118
- [19]. Norbert Ascheuer (2001) Solving the Asymmetric Travelling Salesman Problem with Time Windows by branch-and-cut
- [20].

- [21]. Boschetti, M. and Maniezzo, V. (2009), Benders decomposition, lagrangean relaxation and metaheuristic design, *Journal of Heuristics* 15, 283{312.
- [22]. Fisher, M. (1981), The lagrangean relaxation method for solving integer programming problems, *Management Science* 27, 1-18.
- [23]. Glover, F. and Laguna, M. (1997), *Tabu Search*, Kluwer Academic Publishers: London
- [24]. Guimarans, D., Herrero, R., Riera, D., Juan, A. and Ramos, J. (2010), Combining constraint programming, lagrangean relaxation and probabilistic algorithms to solve the vehicle routing problem, in *Proceedings of 17th RCRA workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion*, Bologna, Italy.
- [25]. Held, M. and Karp, R. (1971), The travelling salesman problem and minimum spanning trees: Part II, *Mathematical Programming* 1, 6-25.
- [26]. Reinelt, G. (1994), *The traveling salesman: computational solutions for TSP applications*, *Lecture Notes in Computer Science* 840.
- [27]. Zamani, R. and Lau, S. (2010), Embedding learning capability in lagrangean relaxation: An application to the travelling salesman problem, *European Journal of Operational Research* 201(1), 82-88.
- [28]. Rosa Herrero, Juan José Ramos, and Daniel Guimarans Lagrangean Metaheuristic for the Travelling Salesman Problem Departament de Telecomunicació i Enginyeria de Sistemes Universitat Autònoma de Barcelona Spain,
- [29]. Michel Gendreau, Gianpaolo Ghiani, and Emanuela Guerriero. Time-dependent routing problems: A review. *Computers & Operation Research*, 64:189–197, 2015
- [30]. Natasha Boland H. Milton Stewart School of Industrial and Systems Engineering Georgia Institute of Technology, USAN. Solving the Traveling Salesman Problem with Time Windows Using Time-Expanded Networks Boland, M. Hewitt, L. Marshall and M.W.P. Savelsbergh. The continuous time service network design problem. Working paper, 2015-01-4729, *Optimization Online*, 2015.
- [31]. X. Wang and A.C. Regan. Local truckload pickup and delivery with hard time window constraints. *Transportation Research B*, 36:97{112, 2002.

- [32]. X.Wang and A.C. Regan. On the convergence of a new time window discretization method for the traveling salesman problem with time window constraints. *Computers and Industrial Engineering*, 56:161{164, 2009
- [33]. S. Dash, O. Gunluk, A. Lodi, and A. Tramontani. A time bucket formulation for the travelling salesman problem with time windows. *INFORMS Journal on Computing*, 24: 132{147, 2012.
- [34]. Pesant, G.; Gendreau, M.; Potvin, J.-Y.; Rousseau, J.-M. An Exact Constraint Logic Programming Algorithm for the Traveling Salesman Problem with Time Windows. *Transp. Sci.* 1998, 32, 12–29. [CrossRef]
- [35]. Ma, Q.; Ge, S.; He, D.; Thaker, D.; Drori, I. Combinatorial optimization by graph pointer networks and hierarchical reinforcement learning. *arXiv* 2019, arXiv:1911.04936.
- [36]. Vinyals, O.; Fortunato, M.; Jaitly, N. Pointer networks. *arXiv* 2015, arXiv:1506.03134.
- [37]. Majed G. Alharbi , Ahmed Stohy , Mohammed Elhenawy , Mahmoud Masoud and Hamiden Abd El-Wahed Khalifa (2021) Solving Traveling Salesman Problem with Time Windows Using Hybrid Pointer Networks with Time Features
- [38]. Stohy, A.; Abdelhakam, H.-T.; Ali, S.; Elhenawy, M.; Hassan, A.A.; Masoud, M.; Glaser, S.; Rakotonirainy, A. Hybrid Pointer Networks for Traveling Salesman Problems Optimization. *arXiv* 2021, arXiv:2110.03104.
- [39]. Duc Minh Vu, Mike Hewitt¹ and Natasha Boland, Martin Savelsbergh (2018) Solving Time Dependent Traveling Salesman Problems with Time Windows Department of Information Systems and Supply Chain Management, Quinlan
- [40]. Anna Arigliano, Gianpaolo Ghiani, Antonio Grieco, and Emanuela Guerriero. Time dependent traveling salesman problem with time windows: Properties and an exact algorithm. Technical report, Optimization Online, 2015.
- [41]. Agustín Montero, Isabel M´endez-Díaz, and Juan Jos´e Miranda-Bront. An integer programming approach for the time-dependent traveling salesman problem with time windows. *Computers & Operations Research*, 88:280–289, 2017.

- [42]. Natasha Boland, Mike Hewitt, Luke Marshall, and Martin Savelsbergh. The continuous-time service network design problem. *Operations Research*, 65(5):1303–1321, 2017a.
- [43]. UN. Our urbanizing world. Technical report, United Nations Department of Economic and Social Affairs, 2014.
- [44]. Miguel Andres Figliozzi. The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics. *Transportation Research Part E: Logistics and Transportation Review*, (3):616 – 636. ISSN 1366-5545.