



NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

**SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS**

BSc THESIS

Vehicle Routing Problem

Nikolaos E. Iosifidis

Supervisor: Ioannis Z. Emiris, Professor

ATHENS

December 2022



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Το πρόβλημα της δρομολόγησης οχημάτων

Νικόλαος Ε. Ιωσηφίδης

Επιβλέπων: Ιωάννης Ζ. Εμίρης, Καθηγητής

ΑΘΗΝΑ

Δεκέμβριος 2022

BSc THESIS

Vehicle Routing Problem

Nikolaos E. Iosifidis

S.N.: 1115201400051

SUPERVISOR: Ioannis Z. Emiris, Professor

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Το πρόβλημα της δρομολόγησης οχημάτων

Νικόλαος Ε. Ιωσηφίδης

A.M.: 1115201400051

ΕΠΙΒΛΕΠΩΝ: Ιωάννης Ζ. Εμίρης, Καθηγητής

ABSTRACT

New approaches for the vehicle routing problem are now in position to provide with quality solutions to harder instances in reasonable time. With that in mind, the number of real world problems that can be tackled has increased, which inevitably will lead to a larger number of computational methods used in routing problems. The purpose of this thesis is to review the literature on the vehicle routing problem but also investigate the problem features that determine the practical difficulty of instances.

SUBJECT AREA: Vehicle Routing Problem

KEYWORDS: Operations Research, Integer Programming, Metaheuristics, Machine Learning

ΠΕΡΙΛΗΨΗ

Τα τελευταία χρόνια έχει σημειωθεί σημαντική πρόοδος στην επίλυση του προβλήματος της δρομολόγησης οχημάτων και οι σύγχρονες προσεγγίσεις είναι σε θέση να παρέχουν καλά αποτελέσματα σε στιγμιότυπα μεγαλύτερης δυσκολίας σε αποδεκτό χρόνο. Αυτό σημαίνει ότι ο αριθμός των προβλημάτων του πραγματικού κόσμου που μπορούν να αντιμετωπιστούν έχει αυξηθεί σημαντικά, συνεπώς η χρήση υπολογιστικών τεχνικών σε προβλήματα δρομολόγησης ολοένα και θα αυξάνεται. Σκοπός αυτής της εργασίας είναι η επισκόπηση των μεθόδων επίλυσης του προβλήματος της δρομολόγησης οχημάτων, αλλά και των χαρακτηριστικών του προβλήματος τα οποία καθορίζουν την πρακτική δυσκολία των στιγμιότυπων.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Το πρόβλημα της δρομολόγησης οχημάτων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Επιχειρησιακή Έρευνα, Ακέραιος Προγραμματισμός, Μεθευρετικές Μέθοδοι, Μηχανική Μάθηση

Στην οικογένεια μου

ACKNOWLEDGEMENTS

Για τη διεκπεραίωση της παρούσας Πτυχιακής Εργασίας, θα ήθελα να ευχαριστήσω τον Καθηγητή Ιωάννη Ζ. Εμίρη για την ευκαιρία να δουλέψω σε ένα πολύ ενδιαφέρον πρόβλημα, τόσο για την θεωρητική του δυσκολία όσο και για τις πολυάριθμες πρακτικές εφαρμογές του.

CONTENTS

1. ΕΙΣΑΓΩΓΗ	12
1.1 Ορισμός του προβλήματος	12
1.2 Μερικές Παραλλαγές	14
1.3 Εφαρμογές	15
1.4 Μελέτη των χαρακτηριστικών του προβλήματος	16
1.5 Μέθοδοι Επίλυσης	19
2. ΑΚΕΡΑΙΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	20
2.1 Τα βασικά στοιχεία ενός BCP αλγορίθμου	20
2.2 Επίλυση του SPPRC	23
2.3 Ενίσχυση των φραγμάτων με την προσθήκη περιορισμών	24
2.4 Τεχνικές επιτάχυνσης	26
3. ΜΕΘΕΥΡΙΣΤΙΚΕΣ ΜΕΘΟΔΟΙ	29
3.1 Βασικές Γειτονιές	29
3.2 Μεθευριστικός Γενετικός Αλγόριθμος HGS	31
3.3 Προσαρμοστική Επαναλαμβανόμενη τοπική αναζήτηση AILS	34
4. ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ	36
5. ΣΥΝΟΨΗ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ	39
5.1 Αποτελέσματα του αλγορίθμου ακέραιου προγραμματισμού	39
5.2 Σύγκριση σύγχρονων Μεθευριστικών μεθόδων	40
5.3 Επιτάχυνση των Μεθευριστικών σε συνδιασμό με τη Μηχανική Μάθηση	45
ABBREVIATIONS - ACRONYMS	47
TABLE OF TERMINOLOGY	48
REFERENCES	50

LIST OF FIGURES

1.1	Στιγμιότυπο του προβλήματος μαζί με λύση	14
1.2	Όχημα διανομής καυσίμων με 6 διαμερίσματα	14
1.3	Απόδοση σε σχέση με τον αριθμό των πελατών	17
1.4	Απόδοση σε σχέση με την κατανομή της ζήτησης των πελατών	17
1.5	Απόδοση σε σχέση με τον αριθμό των πελατών ανά διαδρομή	18
2.1	Το πολύτοπο του γραμμικού προγράμματος προσεγγίζει το κυρτό περιβλημα των ακέραιων λύσεων. Το σημείο με το αστέρι δείχνει την αρχική λύση, η νέα λύση σημειώνεται με τετράγωνο. Με την προσθήκη κατάλληλων περιορισμών η νέα λύση του γραμμικού προγράμματος είναι ακέραια. .	22
3.1	Οι πελάτες i και j ανταλλάζουν θέση σε διαφορετικές διαδρομές	29
3.2	Ο πελάτης i εξυπηρετείται από άλλη διαδρομή	30
3.3	Οι ακμές $(i,i+1)$ και $(j,j+1)$ αντικαθίστανται από τις $(i,j+1)$ και $(j,i+1)$ αντίστοιχα	30
3.4	Οι πελάτες μετά τον i μεταφέρονται στην κάτω διαδρομή και επισκέπτονται μετά τον j ενώ οι πελάτες μετά τον j μεταφέρονται στην πάνω διαδρομή και επισκέπτονται μετά τον i	30
5.1	Η απόδοση του Generic BCP σε σχέση τους δύο άλλους καλύτερους αλγορίθμους	39
5.2	Αναλυτικά αποτελέσματα στα στιγμιότυπα που επιλύθηκαν βέλτιστα για πρώτη φορά	39
5.3	Η HGS παρέχει έως 40 φορές καλύτερη λύση	40
5.4	Σύγκριση για κάθε κατηγορία στιγμιοτύπων	41
5.5	Αποτελέσματα για στιγμιότυπα από 100 μέχρι 250 πελάτες	42
5.6	Αποτελέσματα για στιγμιότυπα από 250 μέχρι 500 πελάτες	43
5.7	Αποτελέσματα για στιγμιότυπα από 500 μέχρι 1000 πελάτες	44
5.8	Αποτελέσματα του μοντέλου επιλογής υποπροβλημάτων σε σχέση με άλλες μεθόδους	45
5.9	Σύγκριση της επιτάχυνσης με την χρήση του HGS και του LKH-3	45

PREFACE

Η εργασία αυτή γράφτηκε στο τμήμα Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών υπό την επίβλεψη του Καθηγητή Ιωάννη Εμίρη.

Στο κεφάλαιο 1 δίνεται ο ορισμός του προβλήματος, κάποιες εφαρμογές, μερικές από τις παραλλαγές του και μια μελέτη των χαρακτηριστικών του προβλήματος.

Στο κεφάλαιο 2 παρουσιάζεται η μέθοδος επίλυσης του προβλήματος με τη χρήση μοντελοποίησης Ακέραιου Προγραμματισμού.

Στο κεφάλαιο 3 παρουσιάζεται η προσεγγιστική μέθοδος επίλυσης του προβλήματος με τη χρήση μεθόδων τοπικής αναζήτησης αλλά και συνδιασμό αυτών με μεθευριστικούς αλγόριθμους.

Στο κεφάλαιο 4 παρουσιάζεται η προσέγγιση του προβλήματος με τη χρήση μοντέλων μηχανικής μάθησης και συνδιασμός τους με μεθόδους τοπικής αναζήτησης.

1. ΕΙΣΑΓΩΓΗ

Το πρόβλημα της δρομολόγησης οχημάτων είναι ένα από τα πιο μελετημένα προβλήματα συνδυαστικής βελτιστοποίησης, με πολυάριθμες πρακτικές εφαρμογές διανομής αγαθών και παροχής υπηρεσιών οι οποίες είναι ζωτικής σημασίας στις σύγχρονες οικονομίες. Σημαντική είναι επίσης και η θεωρητική του σημασία αφού ανήκει στην κλάση των \mathcal{NP} -hard προβλημάτων για τα οποία πιστεύεται ότι δεν υπάρχει αποδοτικός αλγόριθμος επίλυσης ενώ μία λύση σε ένα από αυτά μπορεί να μετατραπεί σε λύση για όλα τα προβλήματα της κλάσης \mathcal{NP} .

1.1 Ορισμός του προβλήματος

Στο πρόβλημα της δρομολόγησης οχημάτων με χωρικούς και χρονικούς περιορισμούς γνωστό στη βιβλιογραφία ως \mathcal{VRPTW} ζητείται να βρεθούν διαδρομές για δοσμένο στόλο οχημάτων οι οποίες να εξυπηρετούν ένα σύνολο πελατών χωρίς να παραβιάζεται η χωρητικότητα των οχημάτων ή το χρονικό διάστημα επίσκεψης που έχει ορίσει ο κάθε πελάτης. Κάθε πελάτης πρέπει να επισκέπτεται μόνο μία φορά, ενώ όλες οι διαδρομές αρχίζουν και τελειώνουν στην αποθήκη. Το κριτήριο βελτιστοποίησης είναι συνήθως η ελαχιστοποίηση της συνολικής απόστασης των διαδρομών.

Πιο συγκεκριμένα ορίζεται ως ένα κατευθυνόμενο γράφημα $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ στο οποίο οι κορυφές αναπαριστούν τους πελάτες ενώ υπάρχουν δύο ακόμη κορυφές για την αποθήκη. Το σύνολο των πελατών \mathcal{C} περιέχει τους πελάτες $1, 2, \dots, n$. Η αποθήκη από την οποία ξεκινάνε τα οχήματα ορίζεται ως η κορυφή 0 ενώ όταν ολοκληρώνουν τη διαδρομή τους τα οχήματα είναι η κορυφή $n + 1$. Όλες οι κορυφές $0, 1, \dots, n + 1$ ορίζουν το σύνολο \mathcal{N} . Το σύνολο των οχημάτων \mathcal{V} περιέχει τα οχήματα χωρητικότητας Q , ενώ σε άλλες παραλλαγές κάθε όχημα έχει την δικιά του χωρητικότητα. Για κάθε πελάτη $i > 0$ ορίζεται μια ποσότητα προς διανομή $d_i > 0$, ο χρόνος που απαιτείται για την εξυπηρέτηση του $s_i > 0$ και το χρονικό παράθυρο $[e_i, l_i]$ με $0 \leq e_i \leq l_i$ το οποίο ορίζει την ελάχιστη και μέγιστη ώρα στην οποία μπορεί να ξεκινήσει η εξυπηρέτηση του πελάτη. Για την αποθήκη η ζητούμενη ποσότητα είναι $d_0 = d_{n+1} = 0$, ο χρόνος εξυπηρέτησης $s_0 = s_{n+1} > 0$ είναι ο χρόνος φόρτωσης των οχημάτων στην αποθήκη, τέλος το χρονικό παράθυρο δείχνει την ελάχιστη ώρα έναρξης και μέγιστη ώρα επιστροφής των οχημάτων.

Όσο αφορά τις ακμές του γραφήματος ορίζεται για κάθε ζεύγος πελατών η ακμή (i, j) μαζί με το κόστος μετακίνησης και τον χρόνο μετακίνησης $c_{i,j}$ και $t_{i,j}$ αντίστοιχα. Μία εφικτή διαδρομή είναι ένα μονοπάτι $i_0 = 0, i_1, \dots, i_{m-1}, i_m = n + 1$ που περιέχει την σειρά επίσκεψης των πελατών το οποίο ξεκινάει και τελειώνει στην αποθήκη, περιέχει κάθε πελάτη μόνο μία φορά χωρίς να υπερβαίνει την χωρητικότητα των οχημάτων $\sum_{i=0}^m d_i \leq Q$ και ο χρόνος έναρξης εξυπηρέτησης t_i για κάθε πελάτη να είναι εντός του χρονικού παραθύρου $t_i \in [e_i, l_i]$ με $t_i = \max\{e_i, t_{i-1} + s_{i-1} + t_{i-1,i}\}$ ώστε να καλύπτεται το ενδεχόμενο το όχημα να φτάσει νωρίτερα και να περιμένει. Το κόστος c_r μιας διαδρομής r υπολογίζεται ως $c_r = \sum_{j=0}^m c_{i_j, i_{j+1}}$

Το \mathcal{VRPTW} μπορεί να μοντελοποιηθεί όπως στο [1] με τη χρήση δυαδικών μεταβλητών x_{ij} οι οποίες δείχνουν με τιμή 1 αν η ακμή (i, j) χρησιμοποιείται σε κάποια διαδρομή αλλιώς έχουν την τιμή 0.

$$\min \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} c_{i,j} x_{i,j} \quad \mathbf{s.t.}, \quad (1.1)$$

$$\sum_{i \in \mathcal{N}} x_{i,j} = 1 \quad \forall j \in \mathcal{C}, \quad (1.2)$$

$$\sum_{j \in \mathcal{N}} x_{i,j} = 1 \quad \forall i \in \mathcal{C}, \quad (1.3)$$

$$\sum_{j \in \mathcal{N}} x_{0,j} = K \quad (1.4)$$

$$\sum_{i \in \mathcal{N}} x_{i,n+1} = K \quad (1.5)$$

$$u_i - u_j + Qx_{i,j} \leq Q - d_i, \quad \forall i, j \in \mathcal{C}, i \neq j, \quad \mathbf{s.t.}, \quad d_i + d_j \leq Q \quad (1.6)$$

$$d_i \leq u_i \leq Q, \quad \forall i \in \mathcal{C}, \quad (1.7)$$

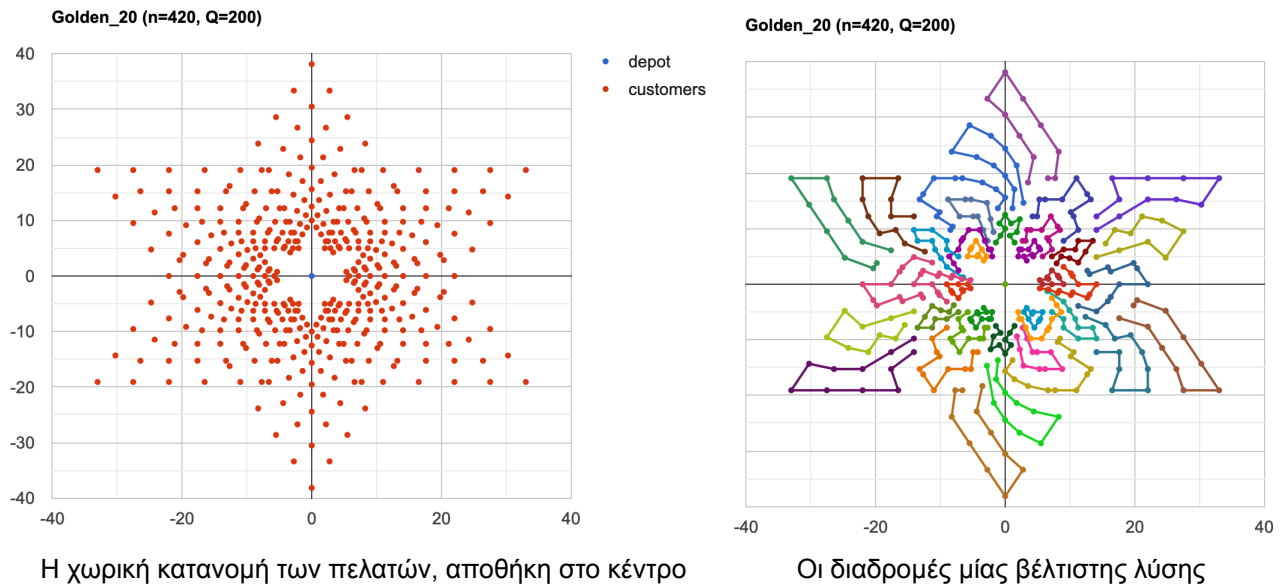
$$t_j \geq t_i + (s_i + t_{ij})x_{ij} - M_{ij}(1 - x_{ij}), \quad i = 0, \dots, n, \quad j = 1, \dots, n+1, \quad M_{ij} = \max\{l_i - e_i, 0\} \quad (1.8)$$

$$e_i \leq t_i \leq l_i \quad (1.9)$$

$$x_{i,j} \in \{0, 1\} \quad \forall i, j \in \mathcal{N} \quad (1.10)$$

Η αντικειμενική συνάρτηση 1.1 ελαχιστοποιεί τη συνολική διαδρομή των οχημάτων. Οι περιορισμοί 1.2 και 1.3 εξασφαλίζουν ότι κάθε πελάτης επισκέπτεται μόνο μία φορά, ενώ οι 1.4 και 1.5 ότι k οχήματα φεύγουν και επιστρέφουν από την αποθήκη αντίστοιχα. Οι 1.6 και 1.7 εφαρμόζουν τις απαιτήσεις σχετικά με τη χωρητικότητα των οχημάτων. Οι 1.8 και 1.9 εφαρμόζουν τα χρονικά παράθυρα των πελατών.

Στο σχήμα που ακολουθεί φαίνεται ένα στιγμιότυπο από το [2] με 420 πελάτες, αποθήκη στο κέντρο και οχήματα χωρητικότητας 200. Στην λύση του, οι 38 διαδρομές δεν φαίνεται ότι ξεκινάνε και καταλήγουν στην αποθήκη για να είναι το σχήμα πιο ευανάγνωστο.

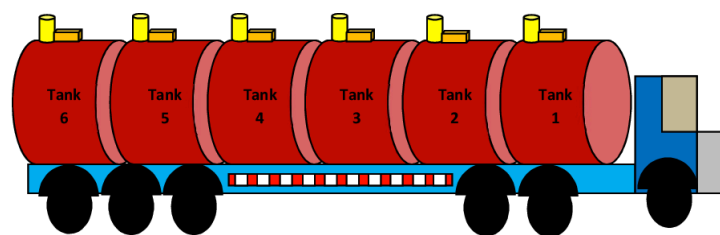


Σχήμα 1.1: Στιγμιότυπο του προβλήματος μαζί με λύση

1.2 Μερικές Παραλλαγές

Στην προηγούμενη ενότητα παρουσιάστηκε η βασική εκδοχή του προβλήματος όπως μελετάται στην βιβλιογραφία [3], όμως οι πρακτικές εφαρμογές του έχουν τις δικές τους απαιτήσεις όταν για παράδειγμα μοντελοποιούν την δρομολόγηση καυσίμων διαφορετικού είδους. Μία ταξινόμηση των βασικών παραλλαγών γίνεται στο [4]

Δρομολόγηση Καυσίμων Σε πολλές εφαρμογές η χωρητικότητα των οχημάτων δεν είναι ενιαία, αλλά αποτελείται από ένα σύνολο διαμερισμάτων. Στο σχήμα που ακολουθεί δίνεται παράδειγμα τέτοιου οχήματος από το [5].



Σχήμα 1.2: Όχημα διανομής καυσίμων με 6 διαμερίσματα

Σε αυτή την παραλλαγή κάθε πελάτης έχει ένα διάνυσμα ζήτησης του οποίου η κάθε συνιστώσα αντιστοιχεί σε τύπο καυσίμου. Συνήθως οι πελάτες δέχονται μία μικρή απόκλιση στην ποσότητα που παραλαμβάνουν η οποία προκύπτει από την χωρητικότητα των διαμερισμάτων και την προσπάθεια τα οχήματα να μεταφέρουν τη μέγιστη ποσότητα καυσίμων. Μια διαδρομή σε αυτό το πρόβλημα, πρέπει εκτός από την σειρά επίσκεψης των πελατών να ορίζει και την ποσότητα που θα περιέχει το κάθε διαμέρισμα. Επιπλέον περιορισμοί εφαρμόζονται στις διαδρομές, οι οποίοι σχετίζονται με την ασφάλεια των οχημάτων όπως για παράδειγμα ότι το διαμέρισμα 1 κοντά στον οδηγό θα πρέπει να είναι φορτωμένο κατά 50% τουλάχιστον σε όλο το μήκος της διαδρομής. Επίσης τα διάφορα μεγέθη των οχημάτων δεν καθιστούν εφικτή την πρόσβαση στο σύνολο των πελατών, έτσι ορίζεται για κάθε πελάτη ένα υποσύνολο των οχημάτων τα οποία μπορούν να τον εξυπηρετήσουν. Τέλος τα οχήματα δύναται να έχουν το δικό τους ωράριο λειτουργίας, το οποίο

σε συνδιασμό με τα χρονικά παράθυρα των πελατών περιορίζει την εφικτότητα των διαδρομών. Μία ανασκόπηση του προβλήματος γίνεται στο [5].

Παράδοση και Επιστροφή Μία άλλη παραλλαγή είναι το *VRPB* στο οποίο οι πελάτες χωρίζονται σε δύο σύνολα, το ένα περιέχει εκείνους οι οποίοι θέλουν να παραλάβουν μία ποσότητα από την αποθήκη και εκείνους που θέλουν να επιστρέψουν μια ποσότητα στην αποθήκη. Σημαντικός είναι επίσης ο περιορισμός ότι σε μία διαδρομή πρώτα ολοκληρώνονται όλες οι παραδόσεις και στη συνέχεια επισκέπτονται οι πελάτες με επιστροφές. Κάθε διαδρομή πρέπει να κάνει τουλάχιστον μία παράδοση. Μία ανασκόπηση του προβλήματος γίνεται στο [6].

Παραλαβή και Παράδοση Αρκετά όμοιο με το *VRPB* είναι το *VRPPD* στο οποίο η διανομή των αγαθών δεν γίνεται από την αποθήκη αλλά ο κάθε πελάτης έχει τη δυνατότητα να λάβει ή να στείλει μια ζητούμενη ποσότητα αγαθών. Κάθε αίτημα αποτελείται από μία τριάδα που περιλαμβάνει μία τοποθεσία παραλαβής, μια τοποθεσία παράδοσης και την ζητούμενη ποσότητα. Κάθε αίτημα πρέπει να εξυπηρετηθεί από μία μόνο διαδρομή, ενώ η τοποθεσία παραλαβής πρέπει να επισκεφεί πριν την τοποθεσία παράδοσης. Μία ανασκόπηση του προβλήματος γίνεται στο [7].

1.3 Εφαρμογές

Δρομολόγηση Ηλεκτρικών Οχημάτων Τα τελευταία χρόνια μεγάλος αριθμός επιστημονικών άρθρων έχουν δημοσιευθεί σχετικά με το *EVRP* καθώς η χρήση των ηλεκτρικών οχημάτων αναμένεται να αυξηθεί σημαντικά. Η σημαντική διαφορά αυτής τη παραλλαγής είναι ότι η φόρτιση των οχημάτων μπορεί να πραγματοποιηθεί μόνο σε συγκεκριμένα σημεία τους σταθμούς φόρτισης, ενώ κάθε σταθμός μπορεί να φορτίζει παραπάνω από ένα όχημα. Τα οχήματα έχουν την δυνατότητα να επισκεφτούν τους σταθμούς φόρτισης ενδιάμεσα από τους πελάτες. Οι πολιτικές επαναφόρτισης είναι δύο, τα οχήματα φεύγουν από το σταθμό με πλήρη μπαταρία ή φορτίζουν για κάποιο χρονικό διάστημα. Μία άλλη τακτική για εξοικονόμηση χρόνου είναι η αντικατάσταση της μπαταρίας με φορτισμένη. Μία ανασκόπηση του προβλήματος γίνεται στο [8] ενώ στο [9] αναλύεται η χρήση ηλεκτρικών οχημάτων σε μεγάλες αποστάσεις εντός της Γερμανίας.

Δρομολόγηση με τη χρήση drones Τα drones μπορούν να μεταφέρουν πιο γρήγορα από τα οχήματα, με χαμηλότερο κόστος ανά χιλιόμετρο όμως δεν μπορούν να διανύσουν μεγάλες αποστάσεις και να μεταφέρουν μεγάλο φορτίο. Στο [10] παρουσιάζεται η πραγματοποίηση διανομών με το συνδιασμό οχημάτων και drones. Ένα όχημα μπορεί να εξυπηρετήσει πελάτες, να μεταφέρει drones, να τα εκτοξεύσει αλλά και να τα συλλέξει από διαφορετικά σημεία. Τα drones πραγματοποιούν την παράδοση στα τελευταία χιλιόμετρα, ενώ στη συνέχεια μπορούν να παραλάβουν φορτίο από διαφορετικό όχημα. Τα πειράματα έδειξαν ότι με το συνδιασμό των δύο μέσων το συνολικό κόστος μεταφοράς μπορεί να μειωθεί 20%.

Δρομολόγηση αεροσκαφών Στο [11] χρησιμοποιείται η μοντελοποίηση του *VRPTW* για την δρομολόγηση αεροσκαφών. Οι μεταβολές στις καιρικές συνθήκες, η συμφόρηση στα αεροδρόμια, οι περιορισμοί εναέριας κυκλοφορίας, οι μηχανικές βλάβες πολλές φορές καθιστούν τον τρέχων σχεδιασμό μη εφικτό. Για αυτό είναι αναγκαία η εύρεση νέων διαδρομών σε σύντομο χρονικό διάστημα. Χρησιμοποιώντας πραγματικά δεδομένα από

αεροπορική εταιρία στην Κίνα τα πειράματα έδειξαν ότι μπορεί να επιτευχθεί αποτελεσματικός επανασχεδιασμός.

Δρομολόγηση πλοίων Στο [12] αναλύεται η δρομολόγηση του στόλου των πλοίων μίας Καναδικής εταιρίας, όπου τα πλοία εξυπηρετούν αιτήματα παραλαβής και παράδοσης. Στην μοντελοποίηση που χρησιμοποιείται περιλαμβάνονται περιορισμοί που αφορούν τον όγκο και το βάρος των πλοίων, τα λιμάνια που μπορεί να επισκεφθεί κάθε πλοίο, χρονικά παράθυρα, ενώ χρησιμοποιείται και μία εκτίμηση καθυστέρησης από το ιστορικό της εταιρίας για κάθε μήνα και κάθε περιοχή. Ο αντίκτυπος ήταν σημαντικός για την εταιρία καθώς επέτρεψε την ανάλυση διαφορετικών σεναρίων ενώ χρησιμοποιήθηκε και για την ανάλυση της κατανάλωσης καυσίμου και τον καθορισμό της ταχύτητας των πλοίων.

1.4 Μελέτη των χαρακτηριστικών του προβλήματος

Σύμφωνα με τους συγγραφείς του [13] η έρευνα στο πρόβλημα της δρομολόγησης οχημάτων παρεμποδίζεται από την έλλειψη καθιερωμένων στιγμιότυπων ικανών για τη σύγκριση των κορυφαίων αλγορίθμων. Για αυτό δημιούργησαν 100 νέα στιγμιότυπα τα οποία ξεχωρίζουν από τα προηγούμενα επειδή είναι πιο ανομοιογενή καθώς ποικίλουν όχι μόνο στον αριθμό των πελατών αλλά και σε άλλα χαρακτηριστικά όπως η θέση της αποθήκης, η θέση των πελατών, οι ζητούμενες ποσότητες για παράδοση και ο μέσος αριθμός πελατών ανά διαδρομή ο οποίος αντιπροσωπεύει το μέγεθος μιας διαδρομής και όχι η συνολική απόσταση της. Επίσης τα νέα στιγμιότυπα χρησιμοποιούνται για την καλύτερη σύγκριση ευριστικών τεχνικών αλλά και την απόδοση ενός ακριβή αλγορίθμου. Η μετρική που χρησιμοποιείται για την σύγκριση είναι το ποσοστό απόκλισης από την καλύτερη γνωστή λύση και υπολογίζεται ως

$$gap = 100 \times \frac{(z - z_{bks})}{z_{bks}} \quad (1.11)$$

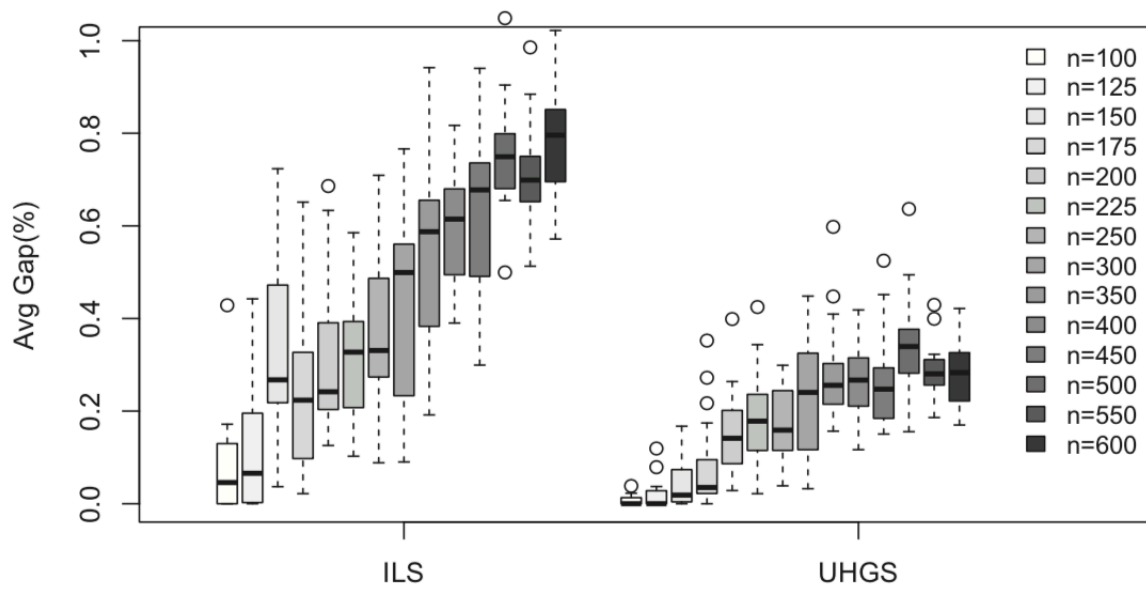
όπου z είναι το κόστος της λύσης του αλγορίθμου και z_{bks} το κόστος της καλύτερης γνωστής λύσης.

Σύγκριση Μεθευριστικών με βάση τα διάφορα χαρακτηριστικά Αρχικά ο αριθμός των πελατών είναι σημαντικός παράγοντας για την απόδοση, η UHGS λύνει βέλτιστα σχεδόν όλα τα προβλήματα με λιγότερους από 150 πελάτες ενώ σε μεγαλύτερα προβλήματα η μέγιστη απόκλιση ήταν 0,64% και 1,05% ενώ η μέση 0,19% και 0,52% για την UHGS και την ILS αντίστοιχα. Σε προβλήματα με 400-600 πελάτες η UGHS έχει μικρή διακύμανση στην απόκλιση με ποσοστά από 0,27% μέχρι 0,34% ενώ για την ILS η απόκλιση είναι περίπου από 0,6 % μέχρι 0,8%.

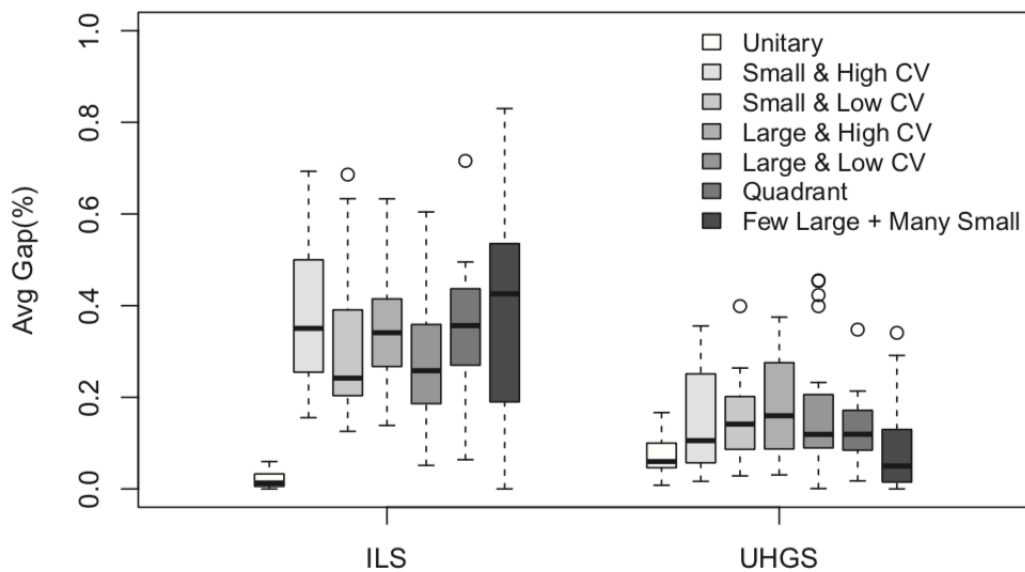
Οι δύο μέθοδοι ILS και UHGS έχουν διαφορετική συμπεριφορά σχετικά με το μέγεθος των διαδρομών με την ILS να είναι καλύτερη στις μικρές διαδρομές ενώ στις μεγάλες η UGHS έχει καλύτερη απόδοση. Όσο αφορά την θέση της αποθήκης οι δύο μέθοδοι έχουν παρόμοια συμπεριφορά, με τα προβλήματα που είχαν την αποθήκη σε γωνία να είναι πιο δύσκολα σε σχέση με εκείνα που την είχαν στο κέντρο αφού αυτό το χαρακτηριστικό δημιουργεί περισσότερες διαδρομές στην ίδια περιοχή. Τα προβλήματα όπου η χωρική κατανομή των πελατών ήταν σε συστάδες ήταν πιο εύκολα, γεγονός που οφείλεται στην μειωμένη επικάλυψη διαδρομών σε διαφορετικές περιοχές.

Όσο αφορά την ζήτηση των πελατών και για τις δύο μεθόδους ο υψηλός συντελεστής μεταβλητότητας καθιστά σημαντικά πιο δύσκολη την επίλυση, η ανάθεση των πελα-

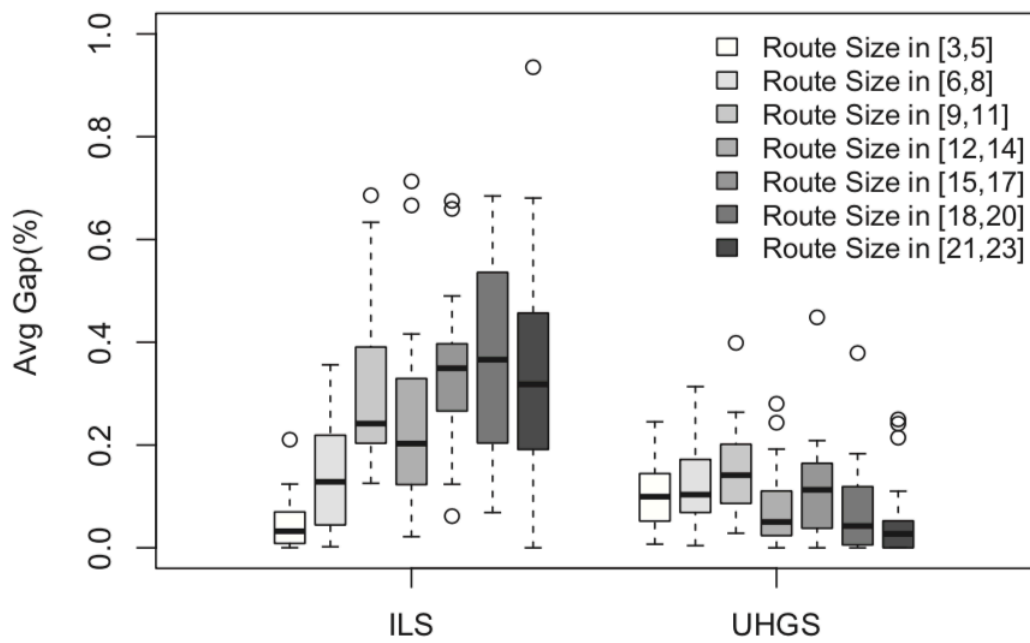
τών σε διαδρομές είναι πολύ δύσκολη όταν οι ζητούμενες ποσότητες είναι διαφορετικής κλίμακας. Τέλος διαφορετική επίδοση έχουν οι δύο αλγόριθμοι όταν η ζήτηση δημιουργεί μερικές διαδρομές σημαντικά μεγαλύτερες από τις υπόλοιπες ή όταν περιέχουν πολλές μικρές ποσότητες και λίγες μεγάλες με την απόδοση της ILS να επιδεινώνεται. Στη συνέχεια παρουσιάζονται σχήματα τα οποία φανερώνουν την διαφορετική συμπεριφορά των αλγορίθμων σε σχέση με τις παραμέτρους του προβλήματος. Σύγχρονες υλοποιήσεις της UGHS και της ILS παρουσιάζονται στις ενότητες 3.2 και 3.3 αντίστοιχα.



Σχήμα 1.3: Απόδοση σε σχέση με τον αριθμό των πελατών



Σχήμα 1.4: Απόδοση σε σχέση με την κατανομή της ζήτησης των πελατών



Σχήμα 1.5: Απόδοση σε σχέση με τον αριθμό των πελατών ανά διαδρομή

Ανάλυση σε ακριβή μέθοδο Τα πειράματα έγιναν σε προβλήματα μέχρι 250 πελάτες καθώς εκτιμάται ότι για περισσότερους πελάτες θα χρειαζόντουσαν πολλές ημέρες αφού η αύξηση του χρόνου ήταν σχεδόν εκθετική για αριθμό πελατών στο διάστημα 100-250.

Η θέση της αποθήκης είχε φανερό αντίκτυπο στον χρόνο εκτέλεσης με τα προβλήματα που είχαν την αποθήκη σε γωνία να έχουν διάμεσο χρόνο 37 φορές μεγαλύτερο σε σχέση με εκείνα που είχαν αποθήκη στο κέντρο. Η θέση των πελατών δεν είχε σημαντική επιρροή απλά τα προβλήματα με πελάτες σε συστάδες ήταν λίγο πιο εύκολα.

Η ζήτηση των πελατών είχε σημαντικό αντίκτυπο στον χρόνο εκτέλεσης. Τα στιγμιότυπα με μεγάλες ποσότητες ήταν δυσκολότερα σε σχέση με εκείνα που είχαν μικρότερες. Τα προβλήματα με μεγάλο συντελεστή μεταβλητότητας ήταν πιο εύκολα σε σχέση με τα αντίστοιχα που είχαν μικρό. Όταν η ζήτηση αποτελούνταν από πολλές μικρές ποσότητες και λίγες μεγάλες ο διάμεσος χρόνος ήταν 12 μέρες.

Τα δυσκολότερα προβλήματα ήταν εκείνα στα οποία η ζήτηση δημιουργεί μερικές διαδρομές σημαντικά μεγαλύτερες από τις υπόλοιπες. Μόνο 8 μπόρεσαν να λυθούν σε λιγότερο από 9 μέρες, ενώ τα υπόλοιπα 12 διακόπηκαν μετά από 13 μέρες εκτέλεσης. Τέλος επιβεβαιώθηκε η σημαντική επίδραση του μέσου μεγέθους των διαδρομών στον χρόνο εκτέλεσης.

1.5 Μέθοδοι Επίλυσης

Ανάλογα με τις απαιτήσεις κάθε εφαρμογής χρησιμοποιείται διαφορετική προσέγγιση επίλυσης. Οι δύο βασικοί παράγοντες με βάση τους οποίους επιλέγεται η κατάλληλη μέθοδος είναι ο χρόνος επίλυσης και η βελτιστότητα της λύσης. Επειδή δεν πιστεύεται ότι το πρόβλημα μπορεί να λυθεί και γρήγορα και βέλτιστα επιλέγεται ποιος παράγοντας είναι πιο σημαντικός για την κάθε εφαρμογή. Όταν είναι διαθέσιμος αρκετός υπολογιστικός χρόνος και το κόστος της λύσης χρειάζεται να είναι το βέλτιστο χρησιμοποιείται ο ακέραιος προγραμματισμός. Ανάλογα με τη δυσκολία του στιγμιότυπου χρειάζονται από ώρες έως και μέρες για να βρεθεί η βέλτιστη λύση. Επίσης ο ακέραιος προγραμματισμός χρησιμοποιείται για την απόδειξη της βελτιστότητας μίας λύσης και την σύγκριση των άλλων μεθόδων σε σχέση με την απόκλιση τους από τη βέλτιστη λύση. Σε εφαρμογές όπου τα αποτελέσματα χρειάζεται να βρεθούν γρήγορα και χαλαρώνεται η απαίτηση για βελτιστότητα χρησιμοποιούνται οι μεθευριστικές μέθοδοι, οι οποίες παρέχουν αρκετά καλή λύση σε μερικά λεπτά. Τέλος η μηχανική μάθηση σε συνδυασμό με τις μεθευριστικές μεθόδους μπορεί να παρέχει ακόμη γρηγορότερα τα αποτελέσματα.

2. ΑΚΕΡΑΙΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Τα τελευταία χρόνια έχει σημειωθεί σημαντική πρόοδος όσο αφορά τις μεθόδους επίλυσης με τη χρήση ακέραιου προγραμματισμού οι οποίες είναι πλέον σε θέση να επιλύσουν προβλήματα μέχρι 360 πελάτες για το $CVRP$, 200 πελάτες για το $VRPTW$ ενώ για τα περισσότερα προβλήματα με περίπου 100 πελάτες για τα οποία χρειαζόντουσαν αρκετές ώρες πλέον χρειάζονται μόνο μερικά λεπτά [14]. Με αυτά τα ενθαρρυντικά αποτελέσματα ο αριθμός των πραγματικών προβλημάτων που επιλύονται βέλτιστα σε εύλογο χρόνο αναμένεται να αυξηθεί.

2.1 Τα βασικά στοιχεία ενός BCP αλγορίθμου

Μία διαφορετική προσέγγιση μοντελοποίησης

Παρόλο που η μοντελοποίηση 1.1- 1.10 γνωστή και ως compact έχει το πλεονέκτημα ότι ο αριθμός των μεταβλητών του προβλήματος αυξάνει πολυωνυμικά σε σχέση με το μέγεθος της εισόδου, η γραμμική χαλάρωση δεν προσφέρει ικανοποιητικά φράγματα. Για αυτό πολλές φορές χρησιμοποιείται η extended η οποία προσφέρει καλύτερη προσέγγιση. Η ανάλυση που ακολουθεί βασίζεται στο [3].

$$\min \sum_{r \in \Omega} c_r \lambda_r \quad s.t. \quad (2.1)$$

$$\sum_{r \in \Omega} a_i^r \lambda_r = 1, \quad \forall i \in \mathcal{C} \quad (2.2)$$

$$\lambda_r \in \{0, 1\}, \quad \forall r \in \Omega \quad (2.3)$$

Η δυαδική παράμετρος a_i^r δείχνει αν ο πελάτης $i \in \mathcal{C}$ επισκέπτεται από την διαδρομή $r \in \Omega$. Για κάθε διαδρομή η δυαδική μεταβλητή λ_r έχει τιμή 1 όταν η διαδρομή r επιλέγεται στη λύση αλλιώς είναι 0. Η αντικειμενική συνάρτηση 2.1 ελαχιστοποιεί το κόστος των διαδρομών, ο περιορισμός 2.2 εξασφαλίζει ότι κάθε πελάτης επισκέπτεται μία φορά. Το μεγάλο πλεονέκτημα αυτής της μοντελοποίησης είναι ότι μερικοί περιορισμοί όπως για παράδειγμα τα χρονικά παράθυρα δεν εκφράζονται άμεσα αλλά εμπεριέχονται έμμεσα στον ορισμό του συνόλου Ω . Παρόλο που αυτή η μοντελοποίηση προσφέρει πιο σφικτά φράγματα περιέχει τεράστιο αριθμό μεταβλητών οι οποίες δεν είναι εφικτό να διαχειριστούν ταυτόχρονα η αλγοριθμική τεχνική BCP η οποία περιλαμβάνει διακλάδωση, δημιουργία μεταβλητών και περιορισμών είναι η κατάλληλη για να ξεπεράσει αυτό το μειονέκτημα.

Επίλυση με BCP

Ένας BCP αλγόριθμος είναι ένας BB αλγόριθμος όπου τα φράγματα υπολογίζονται με τη μέθοδο δημιουργίας στηλών (CG). Η μέθοδος δημιουργίας στηλών επιλύει δύο προβλήματα. Το περιορισμένο κύριο πρόβλημα το οποίο είναι η γραμμική χαλάρωση του μοντέλου 2.1-2.3 επιλύεται εξετάζοντας ένα σχετικά μικρό υποσύνολο $\lambda_r, r \in \Omega' \subseteq \Omega$ των εφικτών διαδρομών.

$$\min \sum_{r \in \Omega} c_r \lambda_r \quad s.t. \quad (2.4)$$

$$\sum_{r \in \Omega} a_i^r \lambda_r = 1, \quad \forall i \in \mathcal{C} \quad (2.5)$$

$$\lambda_r \geq 0, \quad \forall r \in \Omega' \quad (2.6)$$

Το δεύτερο πρόβλημα, γνωστό ως pricing problem επιλύεται ώστε να βρεθούν μεταβλητές που βελτιώνουν το κόστος και να προστεθούν στο κύριο πρόβλημα. Αν δεν βρεθούν τέτοιες διαδρομές σημαίνει ότι η τρέχουσα λύση είναι βέλτιστη και μπορεί να επεκταθεί σε πλήρη λύση θέτοντας $\lambda_r = 0, \forall r \in \Omega \setminus \Omega'$

Algorithm 1 Αλγόριθμος δημιουργίας στηλών

- 1: Επίλυση του περιορισμένου κύριου προβλήματος
 - 2: Επίλυση του pricing problem για την αναζήτηση μεταβλητής η οποία βελτιώνει την τρέχουσα λύση
 - 3: **if** Βρέθηκε μεταβλητή **then**
 - 4: Προσθήκη στο πρόβλημα και μετάβαση στο βήμα 1
 - 5: **else**
 - 6: Η λύση είναι βέλτιστη, Τερματισμός.
-

Σημαντική έννοια για το κριτήριο βελτίωσης είναι αυτή του μειωμένου κόστους. Γενικά το μειωμένο κόστος μίας μεταβλητής δείχνει την μεταβολή της αντικειμενικής συνάρτησης αν αυξηθεί κατά μία μονάδα το κάτω φράγμα της μεταβλητής. Πιο συγκεκριμένα δείχνει την μεταβολή στο κόστος αν η μεταβλητή λ_r πάρει την τιμή 1 δηλαδή να περιλαμβάνεται στη λύση η διαδρομή r . Με κάθε λύση του περιορισμένου κύριου προβλήματος, επιστρέφεται και η συμπληρωματική δυϊκή λύση $\pi_i, i \in \mathcal{C}$ η οποία συνδέεται με τον περιορισμό 2.5. Το μειωμένο κόστος \bar{c}_r της μεταβλητής λ_r είναι :

$$\bar{c}_r = c_r - \sum_{i \in \mathcal{C}} a_i^r \pi_i \quad (2.7)$$

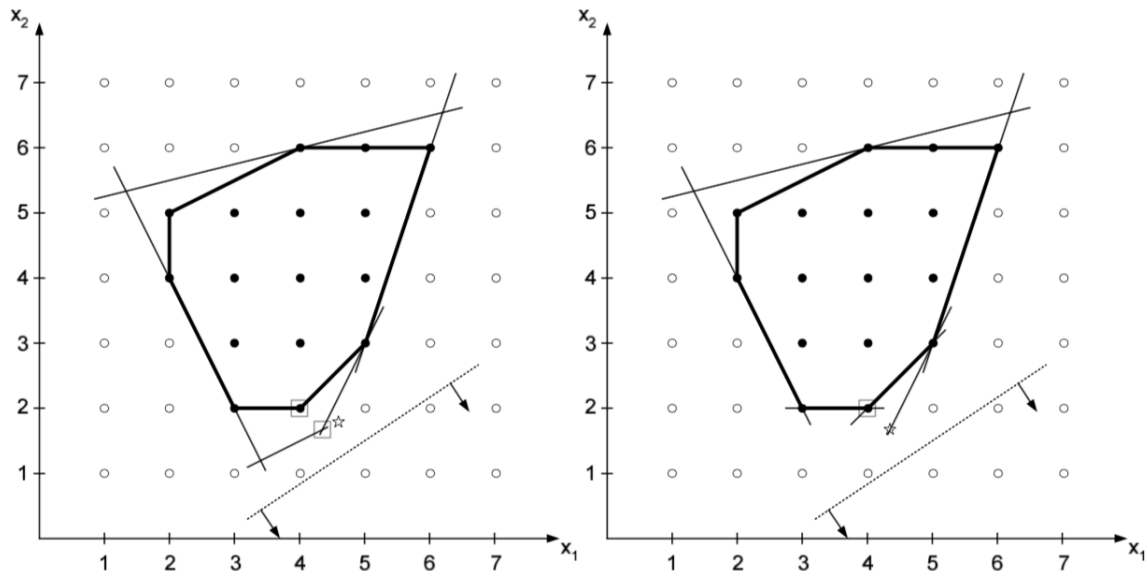
Το pricing problem μοντελοποιείται σαν πρόβλημα συντομότερου μονοπατιού με περιορισμούς πόρων (SPPRC). Οι πόροι είναι ο χρόνος και η χωρητικότητα των οχημάτων και καθορίζουν την εφικτότητα των διαδρομών. Επίσης επιτρέπεται η δημιουργία διαδρομών με πολλαπλές επισκέψεις στον ίδιο πελάτη ενώ το νόημα της παραμέτρου a_i^r δείχνει τις φορές που ο πελάτης επισκέπτεται από τη διαδρομή r . Αυτή η χαλάρωση είναι αναγκαία αφού η εύρεση διαδρομών χωρίς πολλαπλές επισκέψεις είναι \mathcal{NP} -hard. Ο περιορισμός 2.2 εξασφαλίζει ότι οι ακέραιες λύσεις θα περιέχουν μόνο μία επίσκεψη για κάθε πελάτη. Επίσης κατά την επίλυση για κάθε ακμή (i, j) υπολογίζεται το μειωμένο κόστος $\bar{c}_{ij} = c_{ij} - \sum_{i \in \mathcal{C}} a_i^r \pi_i$.

Αυτό γίνεται για να διασφαλιστεί ότι το κόστος ενός μονοπατιού στο γράφημα αντιστοιχεί στο μειωμένο κόστος της διαδρομής $r \in \Omega$.

$$\bar{c}_r = c_r - \sum_{i \in \mathcal{C}} a_i^r \pi_i = \sum_{(i,j) \in \mathcal{A}} c_{ij} b_{ij}^r - \sum_{i \in \mathcal{C}} a_i^r \pi_i = \sum_{(i,j) \in \mathcal{A}} \bar{c}_{ij} b_{ij}^r \quad (2.8)$$

Με την παράμετρο b_{ij}^r να δείχνει πόσες φορές η διαδρομή r διέσχισε την ακμή (i, j) .

Για την δημιουργία αποδοτικού αλγορίθμου είναι αναγκαία η προσθήκη επιπλέον περιορισμών οι οποίοι αποκόπτουν την τρέχουσα βέλτιστη λύση του γραμμικού προγράμματος και επιτυγχάνεται καλύτερη προσέγγιση του χώρου των ακεραίων λύσεων και βελτιωμένο φράγμα. Το επόμενο σχήμα δείχνει αυτή την τεχνική, η εικόνα είναι από [15].



Σχήμα 2.1: Το πολύτοπο του γραμμικού προγράμματος προσεγγίζει το κυρτό περίβλημα των ακεραίων λύσεων. Το σημείο με το αστέρι δείχνει την αρχική λύση, η νέα λύση σημειώνεται με τετράγωνο. Με την προσθήκη κατάλληλων περιορισμών η νέα λύση του γραμμικού προγράμματος είναι ακεραία.

Πολλές φορές τέτοιοι περιορισμοί ή άλλοι που προκύπτουν από αποφάσεις διακλάδωσης ορίζονται με τη χρήση των μεταβλητών x_{ij} . Αυτές οι μεταβλητές μπορούν να υπολογιστούν με τη χρήση των μεταβλητών λ_r σύμφωνα με τον ακόλουθο τύπο.

$$x_{ij} = \sum_{r \in \Omega} b_{ij}^r \lambda_r \tag{2.9}$$

Η προσθήκη περιορισμών της μορφής

$$\sum_{(i,j) \in A} \beta_{ij} x_{ij} \leq \beta_0 \tag{2.10}$$

ονομάζονται *robust* επειδή δεν αυξάνουν την δυσκολία του pricing problem. Επειδή μπορούν να εκφραστούν μέσω των μεταβλητών x_{ij} οι δυϊκές μεταβλητές τους μπορούν να ενσωματωθούν στο τροποποιημένο κόστος της ακμής (i, j) όπως στην 2.8. Αντίθετα περιορισμοί της μορφής

$$x_{ij} = \sum_{r \in \Omega} b_r \lambda_r \leq \beta_0 \tag{2.11}$$

ονομάζονται nonrobust καθώς οι δυϊκές μεταβλητές τους δεν μπορούν να εκφραστούν ως γραμμική συνάρτηση των μεταβλητών x_{ij} και άρα δεν μπορούν να ενσωματωθούν στο τροποποιημένο κόστος της ακμής (i, j) . Έτσι αυξάνεται η δυσκολία του pricing problem με την προσθήκη περιορισμών αυτής της μορφής, καθώς απαιτείται η δημιουργία επιπλέον πόρων κατά της επίλυση του SPPRC. Παρά την αύξηση της πολυπλοκότητας του προβλήματος περιορισμοί της μορφής 2.10 έχουν φανεί καθοριστικοί στην βελτίωση των BPC αλγορίθμων.

2.2 Επίλυση του SPPRC

Το SPPRC επιλύεται αποτελεσματικά με τη χρήση του αλγορίθμου 2, ενώ ένα μονοπάτι p είναι μια μερική διαδρομή και αναπαριστάται από μία πλειάδα $L = (v, \bar{c}, q, t)$ όπου $v \in \mathcal{C}$ είναι ο τελευταίος πελάτης στο μονοπάτι, \bar{c} είναι το μειωμένο κόστος του μονοπατιού, q είναι η συνολική ποσότητα φόρτου του μονοπατιού, τέλος t είναι η ελάχιστη ώρα που μπορεί να ξεκινήσει η εξυπηρέτηση του v αν χρησιμοποιηθεί το p ενώ αποθηκεύεται και ένας δείκτης προς το προηγούμενο μονοπάτι ώστε να δημιουργούνται πλήρη μονοπάτια στο τέλος του αλγορίθμου. Ο αλγόριθμος ξεκινάει με το $L_0 = (0, 0, 0, 0)$ και οι συνιστώσες $v(L), \bar{c}(L), q(L), t(L)$ περιγράφουν ένα μονοπάτι που καταλήγει στον πελάτη $v(L) = i$. Η επέκταση ενός μονοπατιού κατά μήκος της ακμής (i, j) δημιουργεί το L' και γίνεται με τις ακόλουθες σχέσεις.

$$v(L') = j \quad (2.12)$$

$$\bar{c}(L') = \bar{c}(L) + \bar{c}_{ij} \quad (2.13)$$

$$q(L') = q(L) + d_j \quad (2.14)$$

$$t(L') = \max\{t(L) + t_{ij} + s_i, e_j\} \quad (2.15)$$

Στη συνέχεια ελέγχεται η εφικτότητα του νέου μονοπατιού. Το μονοπάτι $p(L')$ είναι εφικτό αν $q(L') \in [0, Q]$ και $t(L') \in [e_j, l_j]$. Η απόδοση του αλγορίθμου εξαρτάται σε μεγάλο βαθμό από την ικανότητα αφαίρεσης περιττών μονοπατιών, μέσω κανόνων κυριαρχίας οι οποίοι διασφαλίζουν ότι μια μερική διαδρομή μπορεί να αντικατασταθεί από μία άλλη που μπορεί να μειώσει το κόστος. Έστω δύο μονοπάτια L_1 και L_2 που καταλήγουν στον ίδιο πελάτη, δηλαδή $v(L_1) = v(L_2)$ τότε το L_1 κυριαρχεί το L_2 αν

$$c(\bar{L}_1) \leq c(\bar{L}_2) \quad (2.16)$$

$$q(L_1) \leq q(L_2) \quad (2.17)$$

$$t(L_1) \leq t(L_2) \quad (2.18)$$

Παρόλου που επιτρέπεται η επίσκεψη ενός πελάτη πάνω από μία φορά στην ίδια διαδρομή για λόγους αποδοτικότητας, αυτό γίνεται με ελεγχόμενο τρόπο. Οι πιο σύγχρονοι BCP αλγόριθμοι χρησιμοποιούν την χαλάρωση ng -μονοπατιού. Για κάθε πελάτη i , ορίζεται μία γειτονία γνωστή ως σύνολο ng . Με τη χρήση της παραμέτρου $\Delta \geq 0$ η γειτονία $N_i \subseteq \mathcal{C}$ περιέχει τους Δ κοντινότερους γείτονες του πελάτη i συμπεριλαμβανομένου και τον ίδιο. Διάφορα κριτήρια μπορούν να χρησιμοποιηθούν για τον ορισμό της εγγύτητας των πελατών όπως η απόσταση ή ο χρόνος. Ένα ng -μονοπάτι περιέχει έναν κύκλο που ξεκινάει και τελειώνει στον πελάτη j αν και μόνο αν στον κύκλο περιέχεται πελάτης i για τον οποίο ο j δεν εμπεριέχεται στους κοντινότερους γείτονες του i , $j \notin N_i$. Σε ένα μονοπάτι $p(L) = (0, i_1, \dots, i_k)$ οι γειτονίες των πελατών του μονοπατιού χρησιμοποιούνται για την δημιουργία του συνόλου $\Pi(L) \subseteq \mathcal{C}$ το οποίο καθορίζει σε ποιους πελάτες μπορεί το μονοπάτι να επεκταθεί χωρίς να παραβιάζονται οι περιορισμοί του ng -μονοπατιού. Έστω $V(L) = \{i_1, \dots, i_l\}$ το σύνολο των πελατών του μονοπατιού $p(L)$, το σύνολο $\Pi(L)$ ορίζεται ως

$$\Pi(L) = \{i_u \in V(L) : i_u \in \bigcap_{s=u}^k N_i\} \quad (2.19)$$

Όταν μια κορυφή j μπαίνει σε ένα σύνολο $\Pi(L)$ μπορεί να φύγει από αυτό όταν το $p(L)$ επεκταθεί με αποτέλεσμα η j να ξαναεπισκεφθεί.

Πλέον στο $ng - SPPRC$ ένα μονοπάτι L μπορεί να επεκταθεί προς την ακμή (i, j) αν $j \notin \Pi(L)$. Όταν δημιουργείται νέο μονοπάτι L' , το νέο σύνολο $\Pi(L')$ υπολογίζεται ως

$$\Pi(L') = (\Pi(L) \cap N_j) \cup \{j\} \quad (2.20)$$

Ενώ στους κανόνες κυριαρχίας πρέπει να προστεθεί η σχέση

$$\Pi(L_1) \subseteq \Pi(L_2) \quad (2.21)$$

Τέλος σημαντική είναι η τιμή της παραμέτρου Δ καθώς μεγαλύτερες τιμές της αυξάνουν την δυσκολία επειδή το $ng - SPPRC$ προσεγγίζει καλύτερα το $SPPRC$ χωρίς πολλαπλές επισκέψεις πελατών αλλά προσφέρει καλύτερα φράγματα. Η τιμή $\Delta = 8$, αποτελεί συνήθως έναν καλό συμβιβασμό ενώ για πιο δύσκολα στιγμιότυπα επιλέγονται τιμές $\Delta \in \{10, 20\}$.

2.3 Ενίσχυση των φραγμάτων με την προσθήκη περιορισμών

Σημαντικό ρόλο στην απόδοση των BPC αλγορίθμων έχει η προσθήκη περιορισμών της μορφής

$$\sum_{r \in \Omega} \lceil \gamma \sum_{i \in C'} a_i^r \rceil \lambda_r \leq \lceil \gamma |C'| \rceil \quad (2.22)$$

όπου $C' \subseteq \mathcal{C}$. Ένας τέτοιος περιορισμός προκύπτει από την στρογγυλοποίηση ενός υποσυνόλου των περιορισμών 2.5 και ονομάζονται $SRCs$. Ο συντελεστής $\gamma = \frac{1}{k}$, $k \in \{1, \dots, |C'|\}$ είναι ίδιος για όλους. Μία διαδρομή με μεταβλητή λ_r έχει συντελεστή διάφορο του 0 στον 2.22 αν $\sum_{i \in C'} a_i^r \geq k$. Δηλαδή αν η διαδρομή r επισκέπτεται

τουλάχιστον k πελάτες στο σύνολο C' . Το μειωμένο κόστος μίας διαδρομής r μετά την προσθήκη ενός περιορισμού s της μορφής 2.22 είναι

$$\bar{c}_r = \sum_{(i,j) \in \mathcal{A}} b_{ij}^r \bar{c}_{ij} - \sigma_s v_s^r \quad (2.23)$$

Όπου $v_s^r = \lfloor \sum_{i \in C'} a_i^r \rfloor$ και C' το υποσύνολο των πελατών για τους οποίους ορίζεται ο περιορισμός s . $\sigma_s \leq 0$ είναι η δυϊκή μεταβλητή που αντιστοιχεί στον περιορισμό s . Το τελευταίο μέρος του μειωμένου κόστους μπορεί να θεωρηθεί ως μία ποινή σ_s για κάθε k επισκέψεις στους πελάτες που ορίζουν το σύνολο C' , αυξάνοντας το μειωμένο κόστος της διαδρομής. Για τον χειρισμό αυτής τη ποινής ο αλγόριθμος επίλυσης του *SPPRC* πρέπει να τροποποιηθεί με τις παρακάτω αλλαγές.

Έστω Θ το σύνολο όλων των περιορισμών s της μορφής 2.22. Για κάθε $s \in \Theta$ μία νέα συνιστώσα $v_s(L)$ προστίθεται στον ορισμό των μονοπατιών L η οποία μετράει το πλήθος των επισκέψεων ($\text{mod } k$) του μονοπατιού σε πελάτες που ανήκουν στο αντίστοιχο σύνολο C_s που ορίζει τον περιορισμό. Όταν ένα μονοπάτι L επεκτείνεται προς την ακμή (i, j) το νέο μονοπάτι L' έχει $v_s(L') = v_s(L)$ αν $j \notin C_s$ αλλιώς $v_s(L') = v_s(L) + 1(\text{mod } k)$. Επίσης όταν $j \in C_s$ και $v_s(L') = 0$ αφαιρείται σ_s από το $\bar{c}(L')$

Επίσης πρέπει να προστεθεί στους κανόνες κυριαρχίας η σχέση

$$\bar{c}(L_1) \leq \bar{c}(L_2) + \sum_{s \in \Theta_{1,2}} \sigma_s \quad (2.24)$$

στη θέση της 2.16, όπου $\Theta_{1,2} \subseteq \Theta$ είναι το υποσύνολο των SRCs για τα οποία $v_s(L_1) > v_s(L_2)$. Ο δεύτερος όρος της δεξιάς μεριάς είναι ένα άνω φράγμα στο όφελος που μπορεί να έχει η διαδρομή L_2 αποφεύγοντας τις ποινές που έχει υποστεί η διαδρομή L_1 στα SRCs για τα οποία έχει περισσότερες επισκέψεις σε πελάτες που τα ορίζουν.

Μία σημαντική βελτίωση των *SRCs* είναι τα Limited-Memory Rank-1 Cuts (*R1Cs*), τα οποία έχουν στόχο να μειώσουν την επίδραση τους στην επίλυση του *SPPRC*. Η επίδραση αυτή αφορά την εξέταση ενός μεγαλύτερου αριθμού μερικών διαδρομών προς επέκταση καθώς η αφαίρεση μερικών διαδρομών, μέσω κανόνων κυριαρχίας είναι πιο αδύναμη μετά την προσθήκη της σχέσης 2.24. Για αυτό τα Limited-Memory *R1Cs* ορίζονται με τέτοιο τρόπο ώστε για περισσότερες μερικές διαδρομές L , και *R1Cs* s να ισχύει η σχέση $v_s(L) = 0$. Ένα Limited-Memory *R1C* ορίζεται μέσω ενός συνόλου C' , ενός συνόλου μνήμης ($C' \subseteq M \subseteq C$) και γράφεται ως

$$\sum_{r \in \Omega} \alpha(C', M, p, r) \lambda_r \leq \lfloor \sum_{i \in C'} p_i \rfloor \quad (2.25)$$

Όπου ο συντελεστής α υπολογίζεται ως συνάρτηση των (C', M, γ, r) και κάθε πελάτη στο σύνολο C' , ο οποίος έχει τον δικό του συντελεστή p_i . Η πιο σημαντική διαφορά με τα *SRCs* είναι ότι κάθε φορά που επισκέπτεται ένας πελάτης $j \notin M_s$ ο οποίος δεν ανήκει στο σύνολο M_s η τιμή του $v_s(L)$ επαναφέρεται σε 0 δηλαδή όλες οι προηγούμενες επισκέψεις σε πελάτες στο C' ξεχνιούνται. Μια λεπτομερής ανάλυση για τον καθορισμό των συντελεστών των *R1Cs* γίνεται στο [16]. Μία ακόμη βελτίωση που προτάθηκε στο [17] και είχε σημαντικό ρόλο στην επίλυση δύσκολων στιγμιοτύπων είναι ο ορισμός της μνήμης ως σύνολο ακμών, με $\alpha(C', AM, p, r)$ και οι προηγούμενες επισκέψεις σε πελάτες στο C' ξεχνιούνται όταν επισκέπτεται μια ακμή $(i, j) \notin AM$.

2.4 Τεχνικές επιτάχυνσης

Σε αυτή την ενότητα παρουσιάζονται μερικές τεχνικές που χρησιμοποιούνται στο [14].

Οργάνωση σε buckets και Pricing διπλής κατεύθυνσης Η χρήση των κανόνων κυριαρχίας για την αφαίρεση μονοπατιών πρέπει να γίνει οργανωμένα καθώς δεν είναι αποδοτικό να γίνει η σύγκριση όλων των ζευγαριών. Ειδάλως, αν δεν πραγματοποιηθούν αρκετοί έλεγχοι ο αριθμός των πιθανών επεκτάσεων μπορεί να γίνει πολύ μεγάλος. Γι' αυτό τα μονοπάτια που έχουν τον ίδιο τελευταίο πελάτη και η συνολική ποσότητα όπως και ο χρόνος του μονοπατιού βρίσκονται στο ίδιο εύρος καταλήγουν στο ίδιο bucket. Αυτή η οργάνωση επιτρέπει τον συχνό έλεγχο κυριαρχίας κυρίως για μονοπάτια στο ίδιο bucket μειώνοντας τον χρόνο που καταναλώνεται σε αυτήν την διαδικασία.

Στο [18] προτάθηκε η χρήση του bucket graph. Πρόκειται για ένα κατευθυνόμενο γράφο στον οποίο οι κορυφές του γραφήματος είναι τα buckets και οι ακμές συνδέουν τα buckets μεταξύ τους ώστε να δείχνουν τις πιθανές επεκτάσεις των μονοπατιών. Αυτήν τη δομή έχει τα εξής οφέλη:

- Η σειρά επεξεργασίας των buckets γίνεται πιο αποδοτική.
- Η σύγκριση μέσω κανόνων κυριαρχίας για μονοπάτια που ανήκουν σε διαφορετικά buckets γίνεται πιο αποδοτικά.
- Ακμές του bucket graph μπορούν να αφαιρεθούν όταν οδηγούν σε επέκταση η οποία δεν βελτιώνει την τρέχουσα βέλτιστη λύση μειώνοντας τις πιθανές επεκτάσεις.

Ο αλγόριθμος επίλυσης του SPPRC μπορεί να υλοποιηθεί και ανάποδα. Σε αυτή την περίπτωση τα μονοπάτια έχουν την μορφή $P = (i, \dots, 0)$. Παρόλο που οι δύο εκδοχές έχουν το ίδιο υπολογιστικό κόστος, έχει παρατηρηθεί στο [19] ότι στην μπροστινή εκδοχή το μεγαλύτερο μέρος των υπολογισμών αφορά buckets με τιμές q κοντά στο Q . Αυτό συμβαίνει γιατί σε ένα bucket με εύρος πιο κοντά στο Q καταλήγουν περισσότερα μονοπάτια. Αντίστοιχα με την ανάποδη εκδοχή η υπολογιστική προσπάθεια παρουσιάζεται κυρίως σε buckets με εύρος σε μικρές τιμές q . Γι' αυτό χρησιμοποιείται συνδιασμός των δύο εκδοχών με τη χρήση της μεταβλητής q^* η οποία μεταβάλλεται δυναμικά. Τα buckets με εύρος μέχρι q^* χρησιμοποιούν τη μπροστινή εκδοχή ενώ για τα υπόλοιπα χρησιμοποιείται η ανάποδη εκδοχή. Στη τελευταία φάση εκτελείται ένα βήμα συνένωσης για την δημιουργία ολοκληρωμένων μονοπατιών. Τέλος ορίζεται διαφορετικός bucket graph για την προσινή και την ανάποδη εκδοχή.

Ευρετικό pricing Κατά την επίλυση του αλγορίθμου 1 αρκεί να βρεθεί μια μεταβλητή με αρνητικό μειωμένο κόστος χωρίς να χρειάζεται να βρεθεί αυτή με τη μεγαλύτερη βελτίωση. Έτσι προκειμένου να επιταχυνθεί η σύγκλιση του αλγορίθμου χρησιμοποιείται pricing τριών σταδίων. Στο πρώτο και ελαφρύτερο αποθηκεύεται μόνο ένα μονοπάτι σε κάθε bucket, αυτό με το μεγαλύτερο μειωμένο κόστος. Στο δεύτερο δεν χρησιμοποιούνται οι διαστάσεις που αφορούν την επέκταση των ng μονοπατιών και τα $rank-1$ cuts κατά τους ελέγχους κυριαρχίας των μονοπατιών αλλά μόνο το μειωμένο κόστος και οι διαστάσεις που αφορούν τον χρόνο και την ποσότητα. Στο τρίτο στάδιο εκτελείται η πλήρης εκδοχή.

Σταθεροποίηση των δυϊκών μεταβλητών Έχει παρατηρηθεί ότι οι τιμές των δυϊκών μεταβλητών εμφανίζουν τις εξής προβληματικές συμπεριφορές: Έστω π^t το διάνυσμα που περιέχει όλες τις δυϊκές μεταβλητές.

- Η τιμή του π αλλάζει με απρόβλεπτο και μη μονότονο τρόπο με αποτέλεσμα $\|\pi^{t+1} - \pi^*\| > \|\pi^t - \pi^*\|$.
- Προς το τέλος του αλγορίθμου οι περιορισμοί κόβουν ένα πολύ μικρό ποσοστό του δυϊκού χώρου καθιστώντας πολύ αργή την σύγκλιση.
- Η ύπαρξη πολλών βέλτιστων λύσεων μπορεί να οδηγήσει συνεχόμενη μετάβαση μεταξύ βέλτιστων λύσεων χωρίς να βελτιώνεται η αντικειμενική συνάρτηση.

Προκειμένου να αντιμετωπιστούν αυτές οι δυσκολίες και να επιτευχθεί γρήγορη σύγκλιση εφαρμόζεται η μέθοδος ομαλοποίησης η οποία λύνει τον 1 με την χρήση του

$$\tilde{\pi} = \alpha \hat{\pi} + (1 - \alpha)\pi \quad (2.26)$$

Όπου $\alpha \in (0, 1]$ και το $\hat{\pi}$ περιέχει πληροφορία για τις δυϊκές τιμές από προηγούμενες επαναλήψεις και μπορεί είναι η καλύτερη λύση μέχρι στιγμής ή σταθμισμένο άθροισμα προηγούμενων λύσεων. Με την χρήση του νέου διανύσματος μπορεί να προκύψει το πρόβλημα της μη εύρεσης μεταβλητών με αρνητικό μειωμένο κόστος ως προς το π . Τότε η παράμετρος α μεταβάλλεται αυτόματα μετά από λίγες επαναλήψεις ώστε $\tilde{\pi} = \pi$.

Μία εκτενής μελέτη της σταθεροποίησης σε αλγορίθμους δημιουργίας στηλών γίνεται στο [20].

Απαρίθμηση διαδρομών Ένας τρόπος για να επιταχυνθεί η σύγκλιση είναι η απαρίθμηση όλων των διαδρομών p οι οποίες μπορούν να βελτιώσουν την τρέχουσα βέλτιστη ακέραια λύση. Αυτό συμβαίνει όταν έχουν μειωμένο κόστος

$$\bar{c}_p < UB - LB \quad (2.27)$$

όπου UB είναι το κόστος της καλύτερης ακέραιας λύσης που έχει βρεθεί και LB το κόστος της τρέχουσας γραμμικής χαλάρωσης.

Για την απαρίθμηση των διαδρομών χρησιμοποιείται μια τροποποιημένη εκδοχή του αλγορίθμου 2, η οποία εξετάζει επεκτάσεις μόνο εφόσον το νέο μειωμένο κόστος είναι μικρότερο από $UB - LB$. Αν κατά την διάρκεια της απαρίθμησης προκύψει μεγάλος αριθμός διαδρομών ή ξεπεραστεί ένα χρονικό όριο τότε η διαδικασία ματαιώνεται. Αν το πλήθος των διαδρομών που προέκυψαν δεν είναι πολύ μεγάλο τότε δίνεται για επίλυση σε ένας αλγόριθμο μεικτού ακεραίου προγραμματισμού. Αν όμως το πλήθος είναι αρκετά μεγάλο για τέτοιο αλγόριθμο οι διαδρομές που προέκυψαν μπορούν να χρησιμοποιηθούν με άλλο τρόπο. Η αναζήτηση για διαδρομή με αρνητικό μειωμένο κόστος γίνεται με εξέταση των διαδρομών που προέκυψαν από την απαρίθμηση και όχι μέσω επεκτάσεων και συγκρίσεων μέσω κανόνων κυριαρχίας. Έτσι μπορούν να προστεθούν περισσότεροι non-robust περιορισμοί χωρίς να επιρεάζεται η δυσκολία του προβλήματος σημαντικά.

Διακλάδωση Η επιλογή των μεταβλητών διακλάδωσης πραγματοποιείται σε κάθε κόμβο του δέντρου χρησιμοποιώντας τρία στάδια. Η κεντρική ιδέα είναι η αφιέρωση περισσότερου χρόνου στην εκτίμηση των μεταβλητών διακλάδωσης στα αρχικά επίπεδα του branch-and-bound δέντρου όπου οι επιλογές έχουν μεγαλύτερο αντίκτυπο στον συνολικό χρόνο εκτέλεσης ενώ σε επόμενα επίπεδα δίνεται λιγότερος χρόνος χρησιμοποιώντας το ιστορικό των προηγούμενων εκτιμήσεων. Οι φάσεις εκτίμησης είναι οι εξής:

Φάση 1 Οι μισοί υποψήφιοι διαλέγονται από το ιστορικό εφόσον υπάρχει, οι υπόλοιποι επιλέγονται με βάση την απόσταση στον κοντινότερο ακέραιο, ενώ όταν επιλέγονται οι ακμές του γραφήματος λαμβάνεται επίσης υπόψιν η απόσταση από την κοντινότερη αποθήκη.

Φάση 2 Εκτιμούνται οι υποψήφιοι από την προηγούμενη φάση επιλύοντας την γραμμική χαλάρωση για κάθε παιδί του κόμβου χωρίς την φάση δημιουργίας στηλών. Επιλέγονται οι μεταβλητές, με τη μεγαλύτερη τιμή $\Delta LB_1 \times \Delta LB_2$ όπου ΔLB_i είναι η βελτίωση στο κάτω γράμμα η οποία προκύπτει για τον κόμβο παιδί με $i = 1, 2$. Η τεχνική αυτή είναι γνωστή ως κανόνας του γινομένου.

Φάση 3 Εκτιμούνται οι υποψήφιοι από την προηγούμενη φάση, επιλύοντας την γραμμική χαλάρωση με ευρυστική δημιουργία στηλών αλλά χωρίς την προσθήκη επιπλέον περιορισμών. Ο καλύτερος υποψήφιος επιλέγεται με τον κανόνα του γινομένου.

Τέλος παρουσιάζεται ο αλγόριθμος επίλυσης του *SPPRC*, όπου για κάθε Im-SRC s οι πελάτες που το ορίζουν είναι το σύνολο $C(s)$, οι συντελεστές είναι $p(s)$ και το σύνολο μνήμης $M(s)$, ενώ η διάσταση S αποθηκεύει πληροφορία σχετικά με την ποινή στο μειωμένο κόστος που προκύπτει από τα n_s Im-SRCs.

Algorithm 2 Αλγόριθμος επίλυσης του SPPRC

```

1: for all buckets  $b \in Buckets$  do ▷ Διέτρεξε όλα τα buckets
2:   for all μονοπάτι  $L_1 = (\bar{c}_1, i, q, t, \Pi_1, S_1) \in b$  do ▷ Για κάθε διαδρομή σε ένα bucket
3:     for all ακμή  $(i, j)$  do
4:       if  $j \notin \Pi_1$  then ▷ Έλεγχος εφικτότητας επέκτασης μέσω της ακμής  $(i, j)$ 
5:          $\bar{c}_2 = \bar{c}_1 + \bar{c}_{ij}$ ,  $S_2 = S_1$ 
6:         if  $L_2 = (\bar{c}_2, j, q + d_j, t + t_{ij} + s_i, \Pi_1, S_2)$  είναι εφικτή then
7:           for  $s = 1$  to  $n_s$  do
8:             if  $j \notin M(s)$  then
9:                $S_2[s] = 0$ .
10:            else if  $j \in C(s)$  then
11:               $S_2[s] = S_2[s] + p(s)$ 
12:              if  $S_2[s] \geq 1$  then
13:                 $\bar{c}_2 = \bar{c}_2 - \sigma_s$ ,  $S_2[s] = S_2[s] - 1$ 
14:                 $L_2 = (\bar{c}_2, j, q + d_j, t + t_{ij} + s_i, \Pi_1 \cap NG(j) \cap j, S_2, pointertoL_1)$ 
15:                 $insertLabel = true$ 
16:            for all μονοπάτι  $L \in b$  do
17:              if  $L_2$  κυριαρχεί το  $L$  then
18:                αφαίρεσε το  $L$ 
19:              else if  $L$  κυριαρχεί το  $L_2$  then
20:                 $insertLabel = false$ , break
21:            if  $insertLabel$  then
22:              εισαγωγή του νέου μονοπατιού στο αντίστοιχο bucket

```

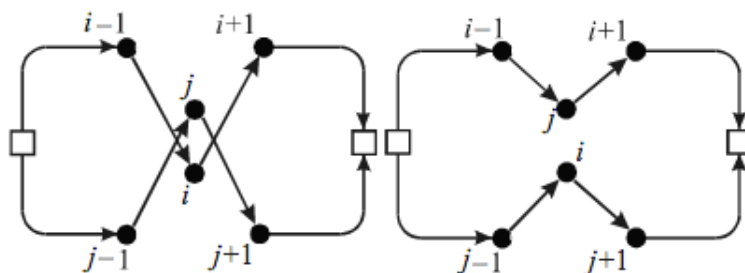
3. ΜΕΘΕΥΡΙΣΤΙΚΕΣ ΜΕΘΟΔΟΙ

Σε μερικές εφαρμογές του $VRPTW$ είναι αναγκαία η εύρεση μιας λύσης σε σύντομο χρονικό διάστημα χαλαρώνοντας την απαίτηση να είναι βέλτιστη. Αυτές οι εφαρμογές επιλύονται με μεθόδους τοπικής αναζήτησης, οι οποίες βασίζονται στην επαναληπτική βελτίωση μιας λύσης και τερματίζονται όταν ξεπεραστεί ένα χρονικό όριο ή όταν δεν υπάρχει περαιτέρω βελτίωση για έναν αριθμό επαναλήψεων. Ξεκινώντας από μία αρχική λύση, πραγματοποιούν αλλαγές στη θέση των πελατών εντός της διαδρομής που τους επισκέπτεται ενώ άλλες φορές πελάτες από διαφορετικές διαδρομές ανταλλάζουν διαδρομές. Για κάθε διαδρομή p αντιστοιχίζεται μία γειτονία $N(p)$ η οποία περιέχει τις εφικτές διαδρομές που προκύπτουν από τις αλλαγές. Σε αυτή τη γειτονιά γίνεται αναζήτηση για καλύτερη λύση, εάν βρεθεί μία τέτοια διαδρομή p' τότε η αναζήτηση συνεχίζεται στη γειτονία $N(p')$.

3.1 Βασικές Γειτονιές

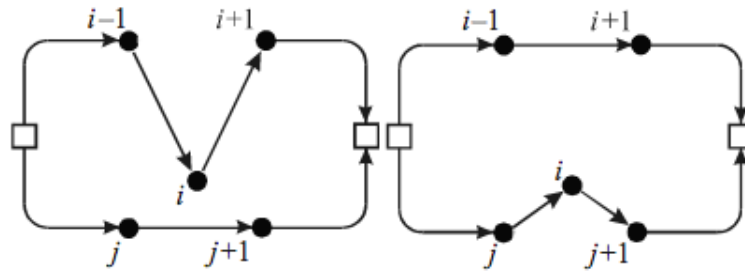
Οι περισσότερες μέθοδοι επαναληπτικής βελτίωσης είναι αλγόριθμοι ανταλλαγής ακμών. Ξεκινώντας από μία διαδρομή η γειτονιά της ορίζεται ως οι διαδρομές που προκύπτουν αν αντικατασταθούν k ακμές με ένα σύνολο από k άλλες. Μία διαδρομή η οποία δεν μπορεί να βελτιωθεί με k αλλαγές ονομάζεται k -optimal. Σε αυτή την ενότητα παρουσιάζονται μερικές από τις πιο δημοφιλείς γειτονιές που χρησιμοποιούνται σύμφωνα με το [21] από το οποίο προέρχονται τα σχήματα των αλλαγών. Στα σχήματα το λευκό τετράγωνο είναι η αποθήκη και οι μαύρες κορυφές οι πελάτες.

Swap Η γειτονία $SWAP$ γνωστή και ως Exchange ανταλλάζει την θέση δύο πελατών, ο ένας παίρνει τη θέση του άλλου και χρησιμοποιείται για βελτίωση τόσο εντός μίας διαδρομής αλλά και μεταξύ διαφορετικών διαδρομών.



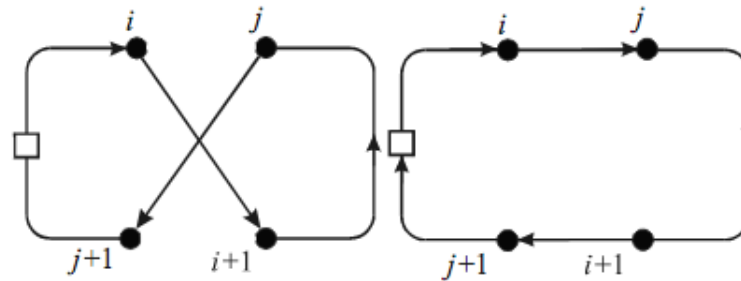
Σχήμα 3.1: Οι πελάτες i και j ανταλλάζουν θέση σε διαφορετικές διαδρομές

Relocate Η γειτονία $RELOCATE$ αλλάζει την διαδρομή για έναν μόνο πελάτη, χρησιμοποιείται για βελτίωση μεταξύ διαφορετικών διαδρομών.



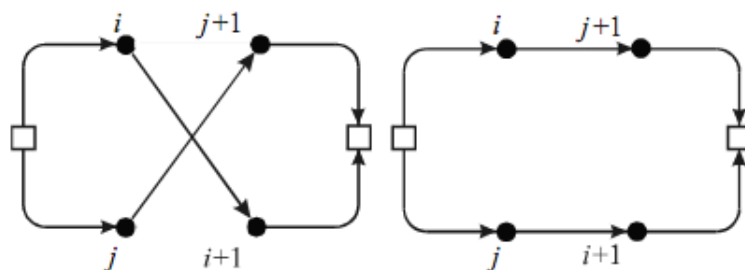
Σχήμα 3.2: Ο πελάτης i εξυπηρετείται από άλλη διαδρομή

2-OPT Η γειτονία 2-OPT αλλάζει δύο ακμές μίας διαδρομής με δύο άλλες. Η βασική ιδέα είναι η επανατοποθέτηση των πελατών ώστε να μην υπάρχουν διασταυρώσεις εντός της διαδρομής.



Σχήμα 3.3: Οι ακμές $(i, i+1)$ και $(j, j+1)$ αντικαθίστανται από τις $(i, j+1)$ και $(j, i+1)$ αντίστοιχα

2-OPT* Η γειτονία 2-OPT* αλλάζει μέρη δύο διαδρομών, οι τελευταίοι πελάτες μίας διαδρομής μεταφέρονται μετά τους πρώτους της άλλης διαδρομής έτσι ώστε να διατηρείται ο προσανατολισμός των διαδρομών



Σχήμα 3.4: Οι πελάτες μετά τον i μεταφέρονται στην κάτω διαδρομή και επισκέπτονται μετά τον j ενώ οι πελάτες μετά τον j μεταφέρονται στην πάνω διαδρομή και επισκέπτονται μετά τον i

3.2 Μεθευριστικός Γενετικός Αλγόριθμος HGS

Το πρόβλημα με την τοπική αναζήτηση είναι ότι όταν δεν μπορεί να βρεθεί καλύτερη λύση εντός της γειτονίας η αναζήτηση εγκλωβίζεται σε ένα τοπικό ελάχιστο. Για να ξεπεραστεί αυτή η δυσκολία η τοπική αναζήτηση συνδιάζεται με μεθόδους που βοηθάνε να συνεχιστεί η αναζήτηση και τελικά να βρεθεί η βέλτιστη λύση ή μία με μικρή απόκλιση από αυτή. Στο [22] παρουσιάζεται μία τέτοια υλοποίηση για το $CVRP$, η οποία στηρίζεται στις εξής ιδέες.

- Συνδιασμός της τοπικής αναζήτησης με την διαφοροποίηση της αναζήτησης στο χώρο των λύσεων.
- Ελεγχόμενη εξερεύνηση μη εφικτών λύσεων, όπου η πλεονάζουσα χωρητικότητα τιμωρείται με γραμμικό τρόπο. Η αναζήτηση σε περιοχές κοντά στα όρια της εφικτότητας μπορεί να οδηγήσει σε βέλτιστες ή σχεδόν βέλτιστες λύσεις.
- Διατήρηση ενός συνόλου ποιοτικών αλλά και διαφοροποιημένων λύσεων ώστε να αντισταθμιστεί η έλλειψη ποικιλίας που προκύπτει από την αναζήτηση σε γειτονίες.

Για την διαφοροποίηση των λύσεων χρησιμοποιούνται ιδέες από τους γενετικούς αλγορίθμους, όπως η διασταύρωση (ζευγάρωμα) λύσεων. Σε αυτό το πλαίσιο οι υποψήφιες λύσεις προς βελτίωση ονομάζονται πληθυσμός.

Περιληπτικά ο αλγόριθμος μετά την αρχικοποίηση του πληθυσμού, παράγει επαναληπτικά νέα λύση ακολουθώντας τα παρακάτω βήματα.

1. Επιλέγονται δύο λύσεις ως γονείς
2. Ζευγάρωμα των δύο λύσεων ώστε να παραχθεί νέα λύση
3. Βελτίωση της λύσης με τοπική αναζήτηση
4. Εισαγωγή του αποτελέσματος στον πληθυσμό

Επιλογή γονιών Η καταλληλότητα μίας λύσης για επιλογή βασίζεται τόσο στο κόστος της λύσης $c(s)$ όσο και σε έναν παράγοντα διαφοροποίησης $div(s)$, ο οποίος υπολογίζεται μετρώντας το ποσοστό των κοινών ακμών με τις $n^{closest}$ πιο παρόμοιες λύσεις στον πληθυσμό \mathcal{P} .

$$f(s) = c(s) + (1 - \frac{n^{elite}}{\|\mathcal{P}\|})div(s) \quad (3.1)$$

Η ποιότητα της λύσης έχει λίγο μεγαλύτερο συντελεστή ώστε να διασφαλιστεί η διατήρηση των n^{elite} καλύτερων λύσεων κατά την αναζήτηση.

Δημιουργία νέων λύσεων Σε αυτό το στάδιο χρησιμοποιείται αναπαράσταση των διαδρομών ως μία τεράστια διαδρομή, χωρίς επισκέψεις στην αποθήκη, αγνοώντας τους περιορισμούς χωρητικότητας. Η νέα λύση κληρονομεί ένα τυχαίο τμήμα του πρώτου γονέα, συμπληρώνοντας τους υπόλοιπους σύμφωνα με την σειρά επίσκεψης του δεύτερου, μετά από το τμήμα που κληρονομήθηκε. Για παράδειγμα δύο γονείς $P_1 = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$, $P_2(1, 3, 5, 7, 9, 2, 4, 6, 8, 10)$ και το τμήμα $(5, 6, 7, 8)$ του πρώτου γονέα επιλέγεται για κληρονομήση. Σε αυτό το βήμα η νέα λύση θα είναι $NS = (, , , 5, 6, 7, 8, ,)$. Οι υπόλοιποι πελάτες συμπληρώνονται μετά το τμήμα που κληρονομήθηκε ξεκινώντας από τους δύο τελευταίους πελάτες του P_2 δίνοντας την λύση

$NS = (3, 9, 2, 4, 5, 6, 7, 8, 10, 1)$. Η νέα λύση μετατρέπεται με τη χρήση του γραμμικού αλγορίθμου *SPLIT* που παρουσιάστηκε στο [23] σε ολοκληρωμένη λύση του *CVRP* με την εισαγωγή επισκέψεων στην αποθήκη.

Αναζήτηση Γειτονιών Στη συνέχεια η νέα λύση οδηγείται στο στάδιο της τοπικής αναζήτησης. Χρησιμοποιούνται οι γειτονίες *SWAP*, *2-OPT*, *2-OPT** και *RELOCATE* με την τελευταία να γενικεύεται σε ακολουθίες δύο συνεχόμενων πελατών. Οι αλλαγές περιορίζονται σε ακμές (i, j) που περιλαμβάνουν κοντινούς πελάτες έτσι ώστε ο j να ανήκει στους Γ κοντινότερους πελάτες του i . Η επιλογή των ακμών γίνεται με τυχαία σειρά και δοκιμάζονται όλες οι γειτονίες μέχρι να βρεθεί ένα τοπικό ελάχιστο.

Επίσης χρησιμοποιείται μία νέα γειτονία, η *SWAP** στην οποία δύο πελάτες από δύο διαδρομές μπορούν να βρεθούν σε οποιαδήποτε θέση της άλλης διαδρομής. Οι πιθανές αλλαγές είναι $\mathcal{O}(n^3)$ και για την μείωση της πολυπλοκότητας σε $\mathcal{O}(n^2)$ οι δύο πελάτες είτε ανταλλάζουν θέση είτε η εισαγωγή γίνεται σε μία από τις τρεις καλύτερες θέσεις που έχουν υπολογιστεί πριν την αφαίρεση. Τέλος επειδή η τετραγωνική πολυπλοκότητα μεγαλώνει σημαντικά τον χρόνο εκτέλεσης για μεγάλα στιγμιότυπα, η νέα γειτονία χρησιμοποιείται μόνο σε διαδρομές οι οποίες έχουν κοινό πολικό τμήμα.

Μία λύση μπορεί να παραμένει μη εφικτή μετά την τοπική αναζήτηση, τότε με 50% πιθανότητα η αναζήτηση επαναλαμβάνεται με 10 φορές μεγαλύτερη ποινή προσπαθώντας να ανακτηθεί μία εφικτή λύση.

Διαχείριση πληθυσμού Χρησιμοποιούνται δύο ξεχωριστοί πληθυσμοί για εφικτές και μη εφικτές λύσεις. Κάθε ένας περιέχει ένα αριθμό από λύσεις μεταξύ μ και $\mu + \lambda$. Αρχικά δημιουργούνται 4μ τυχαίες λύσεις οι οποίες μετά την τοπική αναζήτηση ταξινομούνται ανάλογα με την εφικτότητά τους. Όταν ένας πληθυσμός φτάσει τις $\mu + \lambda$ λύσεις επαναληπτικά αφαιρούνται λ από αυτές αφαιρώντας πρώτα ίδιες λύσεις αν υπάρχουν αλλιώς αφαιρούνται οι λιγότερο κατάλληλες σύμφωνα με την εξίσωση 3.1. Τέλος η ποινή για την μη εφικτότητα των λύσεων προσαρμόζεται κατά τις επαναλήψεις έτσι ώστε να εξασφαλιστεί ένα ποσοστό εφικτών λύσεων στο τέλος του αλγορίθμου.

Συνολικά ο αλγόριθμος μπορεί να περιγραφεί από τον παρακάτω ψευδοκώδικα.

Algorithm 3 Αλγόριθμος HGS

- 1: Αρχικοποίηση του πληθυσμού με τυχαίες λύσεις βελτιωμένες από τοπική αναζήτηση
 - 2: **while** Ο αριθμός των επαναλήψεων χωρίς βελτίωση $< It$ και χρόνος $< T_{max}$ **do**
 - 3: Επέλεξε δύο λύσεις ως γονείς P_1 και P_2
 - 4: Δημιουργία νέα λύσης NS
 - 5: Βελτίωση της NS με τοπική αναζήτηση
 - 6: Εισαγωγή της NS στον αντίστοιχο υποπληθυσμό
 - 7: **if** NS μη εφικτή **then**
 - 8: Με 50% πιθανότητα επανάληψη της τοπικής αναζήτησης με μεγαλύτερη ποινή και εισαγωγή της NS στον αντίστοιχο υποπληθυσμό
 - 9: **if** Κάποιος υποπληθυσμός έφτασε το μέγιστο μέγεθος **then** Αφαίρεσε λ από αυτούς
 - 10: Τροποίηση της ποινής για μη εφικτότητα
 - 11: Επέστρεψε την καλύτερη εφικτή λύση
-

3.3 Προσαρμοστική Επαναλαμβανόμενη τοπική αναζήτηση AILS

Η βασική ιδέα της ILS για να ξεπεράσει το τοπικό ελάχιστο είναι η διατάραξη της λύσης μετακινώντας κάποιους πελάτες σε άλλες θέσεις ώστε να δημιουργηθεί νέα λύση και στη συνέχεια να βελτιωθεί με τη χρήση της τοπικής αναζήτησης. Στο [24] χρησιμοποιείται αλγόριθμος δύο σταδίων όπου κάθε ένα χρησιμοποιεί διατάραξη και τοπική αναζήτηση αλλά η διαφορά τους είναι ότι χρησιμοποιούν διαφορετική λύση αναφοράς στην έναρξη τους. Το πρώτο στάδιο ανανεώνει την λύση αναφοράς με βάση ένα κριτήριο αποδοχής το οποίο ορίζει ένα κάτω φράγμα στην ποιότητα της λύσης ενώ στο δεύτερο στάδιο επιλέγεται τυχαία μία λύση από το σύνολο \mathcal{E} που περιέχει τις καλύτερες λύσεις που έχουν βρεθεί. Για να προστεθεί μία λύση στο σύνολο \mathcal{E} θα πρέπει να έχει κόστος μικρότερο από την χειρότερη λύση $s_e \in \mathcal{E}$ ενώ όταν φτάσει το μέγιστο μέγεθος αφαιρείται η s_e μετά την προσθήκη νέας λύσης. Το πρώτο στάδιο έχει μεγαλύτερη εξερεύνηση του χώρου των λύσεων ενώ το δεύτερο επικεντρώνεται στον χώρο γύρω από τις καλύτερες λύσεις. Ο αλγόριθμος επιλέγει σε κάθε επανάληψη με τυχαίο τρόπο ποιο στάδιο θα χρησιμοποιήσει. Τέλος σημαντικοί παράμετροι του αλγορίθμου όπως το κριτήριο αποδοχής της λύσης αναφοράς του πρώτου σταδίου και ο βαθμός διατάραξης της ο οποίος είναι κοινός για τα δύο στάδια μεταβάλλονται δυναμικά κατά την εκτέλεση του.

Διατάραξη των λύσεων Προκειμένου να δημιουργηθεί νέα λύση επιλέγεται τυχαία μία διαδρομή και ένα σημείο της, αφαιρούνται από αυτήν ω συνεχόμενοι πελάτες. Αν η διαδρομή μείνει κενή επιλέγεται νέο σημείο σε άλλη διαδρομή. Προκειμένου να εξασφαλιστεί διαφορετική λύση διασφαλίζεται ότι ένας πελάτης v_i με γείτονες v_{i-1} και v_{i+1} θα έχει διαφορετικούς γείτονες στη νέα λύση. Επίσης η εισαγωγή γίνεται σε θέση στην οποία ο v_i είναι γειτονικός με τον κοντινότερο γείτονά του. Η τιμή της παραμέτρου ω ανανεώνεται κάθε γ επαναλήψεις με βάση την απόσταση $d(s, s^r)$ μεταξύ της τρέχουσας λύσης s και της λύσης αναφοράς s^r . E^s είναι το σύνολο ακμών της λύσης s και η απόσταση ορίζεται με τη χρήση της συμμετρικής διαφοράς ως

$$d(s, s^r) = |E^s \Delta E^{s^r}|. \quad (3.2)$$

Με $\omega = \min\{n, \max\{1, \omega \frac{d_\beta}{d_k}\}\}$, όπου d_β είναι καλύτερη απόσταση από την s^r και d_k ο μέσος όρος των αποστάσεων $d(s, s^r)$ από την τελευταία ανανέωση. Αν η $d(s, s^r)$ είναι μεγαλύτερη από d_β τότε η ω μειώνεται ενώ αν είναι μεγαλύτερη αυξάνεται.

Κριτήριο αποδοχής νέας λύσης αναφοράς στο στάδιο 1 Προκειμένου μία λύση s να αντικαταστήσει την λύση αναφοράς s^r θα πρέπει να έχει κόστος λύσης $c(s)$ μικρότερο από b , $b = f_b + \eta(f_a - f_b)$ με $\eta \in [0, 1]$, όπου f_b είναι η καλύτερη λύση και f_a είναι ο μέσος όρος του κόστους που βρέθηκε στις γ τελευταίες επαναλήψεις. Μέσω της παραμέτρου η επιτυγχάνεται πιο χαλαρό κριτήριο στις πρώτες επαναλήψεις ενώ γίνεται πιο περιοριστικό όσο εξελίσσεται ο αλγόριθμος. Αρχικά $\eta = 1$ ενώ μετά από κάθε επανάληψη η τιμή της μειώνεται $\eta = \eta\alpha$, όπου $\alpha \in [0, 1]$ με την παράμετρο α να μεταβάλλεται κάθε γ επαναλήψεις παίρνοντας την τιμή $\epsilon \frac{t_a t_f}{it}$, όπου t_a είναι ο χρόνος που εκτελείται ο αλγόριθμος, t_f είναι ο μέγιστος χρόνος εκτέλεσης, it είναι ο αριθμός των επαναλήψεων μέχρι στιγμής και $\epsilon = 0.01$.

Algorithm 4 Αλγόριθμος AILS

- 1: Αρχικοποίηση με τυχαίες λύσεις βελτιωμένες από τοπική αναζήτηση
 - 2: **while** Ο αριθμός των επαναλήψεων χωρίς βελτίωση $< It$ και χρόνος $< T_{max}$ **do**
 - 3: Επέλεξε λύση αναφοράς s^r με τυχαίο τρόπο μεταξύ των δύο σταδίων S_1 και S_2
 - 4: Δημιουργία νέα λύσης s με τη διατάραξη της s^r
 - 5: Βελτίωση της s με τοπική αναζήτηση
 - 6: **if** $c(s) \leq c(s^*)$ **then**
 - 7: $s^* = s$
 - 8: Ανανέωση βαθμού διατάραξης
 - 9: Ανανέωση του συνόλου \mathcal{E} με τις καλύτερες λύσεις
 - 10: Ανανέωση κριτηρίου αποδοχής σταδίου 1
 - 11: Επέστρεψε την καλύτερη λύση s^*
-

4. ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Τα τελευταία χρόνια σημαντικό μέρος τη έρευνας για το $CVRP$ έχει στραφεί σε μεθόδους μηχανικής μάθησης. Σημαντική αρχή ήταν το άρθρο [25] στο οποίο χρησιμοποιείται ενισχυτική μάθηση πετυχαίνοντας καλά αποτελέσματα με πολύ μικρό χρόνο εκτέλεσης μετά την εκπαίδευση του μοντέλου όμως η προσέγγιση αυτή καταφέρνει να επιλύσει στιγμιότυπα μέχρι 100 πελάτες.

Προκειμένου να αντιμετωπιστούν πολύ μεγαλύτερα προβλήματα με περισσότερους από 1000 πελάτες, στο [26] προτείνεται μία μέθοδος εποπτευόμενης μάθησης η οποία συνδυάζεται με μεθευριστικούς αλγορίθμους. Η βασική ιδέα σε αυτή την προσέγγιση είναι η διάσπαση του μεγάλου προβλήματος σε μικρότερα υποπροβλήματα με τη χρήση της μηχανικής μάθησης και επίλυση τους με μεθόδους τοπικής αναζήτησης.

Δομή του αλγορίθμου

1. Κάθε επανάληψη ξεκινάει με μία τρέχουσα λύση
2. Κάθε διαδρομή αντιπροσωπεύεται από τον αριθμητικό μέσο των πελατών της
3. Για κάθε διαδρομή ορίζεται ως υποπρόβλημα οι k εγγύτεροι γείτονες της μέσω των αντιπροσώπων κάθε διαδρομής
4. Το μοντέλο μάθησης επιλέγει το υποπρόβλημα με την μεγαλύτερη εκτιμώμενη άμεση βελτίωση
5. Οι πελάτες που ορίζουν το υποπρόβλημα δίνονται στον αλγόριθμο τοπικής αναζήτησης ως είσοδος για τη δημιουργία νέας λύσης και όχι οι διαδρομές που τους εξυπηρετούσαν
6. Ανανέωσης της τρέχουσας λύσης και επανάληψη των βημάτων 2-6

Μείωση του χώρου αναζήτησης Ο αριθμός των διαδρομών είναι $\mathcal{O}(N)$ που οδηγεί σε εκθετικό χώρο αναζήτησης $\mathcal{O}(2^N)$ ο οποίος είναι δύσκολος να αντιμετωπισθεί. Προκειμένου να περιορισθεί, χρησιμοποιείται η έννοια της τοπικότητας, οι πελάτες επηρεάζονται περισσότερο από κοντινούς σε σχέση με πιο μακρινούς. Ο ορισμός του υποπροβλήματος ώστε να περιέχει μία διαδρομή r και k άλλες των οποίων οι αντιπρόσωποι έχουν την μικρότερη Ευκλείδεια απόσταση από τον αντιπρόσωπο της r , επιτρέπει την μείωση του χώρου αναζήτησης από εκθετικό σε πολυωνυμικό καθώς το μοντέλο επιλέγει από αριθμό υποπροβλημάτων ίσο με τον αριθμό των διαδρομών.

Κριτήριο επιλογής υποπροβλημάτων Ένα υποπρόβλημα S με λύση X_S και με νέα λύση X'_S μετά την τοπική αναζήτηση έχει άμεση βελτίωση

$$\delta(S) = c(X_S) - c(X'_S) \quad (4.1)$$

Το γεγονός ότι δεν δίνεται η X_S για βελτίωση μπορεί να οδηγήσει σε αρκετά βήματα με μη θετική βελτίωση $\delta(S) \leq 0$, κατά την εκπαίδευση του μοντέλου, όμως βοηθάει την τοπική αναζήτηση να ξεπεράσει το τοπικό ελάχιστο. Για την αποφυγή

επιδείνωσης του κόστους όταν $\delta(S) \leq 0$, διατηρείται η X_S και όχι η X'_S ενώ στη συνέχεια επιλέγεται διαφορετικό υποπρόβλημα. Μία δεύτερη προσέγγιση είναι η επιλογή του υποπροβλήματος με βάση την συνολική βελτίωση μετά από πολλαπλές επαναλήψεις χρησιμοποιώντας το άθροισμα με $\gamma = 0.25$

$$\sum_{t=1}^T \gamma^t \delta_t \quad (4.2)$$

Αρχιτεκτονική του μοντέλου Η συνάρτηση f_θ του μοντέλου ορίζεται με τη χρήση της αρχιτεκτονικής του μετασχηματιστή που προτάθηκε στο [27], το οποίο μαθαίνει να εστιάζει την προσοχή του σε μέρη της εισόδου. Η είσοδος είναι τα υποπροβλήματα, όπου κάθε ένα αναπαρίστανται ως το σύνολο των μη ταξινομημένων πελατών που βρίσκονται στις διαδρομές του υποπροβλήματος. Τα χαρακτηριστικά κάθε πελάτη είναι η ζήτηση d και η γεωγραφική θέση (x, y) . Η έξοδος του μοντέλου $f_\theta(S)$ είναι η εκτίμηση του κόστους της νέας λύσης $c(X'_S)$ για όλα τα υποπροβλήματα και επιλέγεται το

$$\arg \max_S c(X_S) - f_\theta(S) \quad (4.3)$$

Δημιουργία δεδομένων εκπαίδευσης Η μείωση του χώρου αναζήτησης επιτρέπει την απαρίθμηση όλων των δυνατών υποπροβλημάτων και την εύρεση του $c(X'_S)$ για κάθε ένα. Σε ένα στιγμιότυπο με 2000 πελάτες μία λύση αποτελείται από περίπου 200 διαδρομές και άρα 200 υποπροβλήματα. Παρόλο που αυτή η στρατηγική είναι εφικτή για την δημιουργία δεδομένων εκπαίδευσης είναι αργή για να χρησιμοποιηθεί σε κάθε νέο στιγμιότυπο και γ'αυτό χρησιμοποιείται η εκτίμηση του μοντέλου.

Αντικειμενική συνάρτηση Για την μεταβολή των παραμέτρων του μοντέλου χρησιμοποιείται η απώλεια Huber καθώς εμπειρικά φάνηκε να είναι πιο σταθερή από τη συνάρτηση του μέσου τετραγωνικού σφάλματος και ορίζεται ως

$$L(\theta; S) = \begin{cases} \frac{1}{2}(f_\theta(S) - c(X'_S))^2 & \text{αν } |f_\theta(S) - c(X'_S)| \leq 1 \\ |f_\theta(S) - c(X'_S)| - \frac{1}{2} & \text{αλλιώς} \end{cases} \quad (4.4)$$

Απόδοση και Γενίκευση σε άλλες κατανομές Για την επιβεβαίωση της ικανότητας του μοντέλου να αναγνωρίζει τα υποπροβλήματα με προοπτική βελτίωσης χρησιμοποιείται ως τιμή αναφοράς ο χρόνος επίλυσης με τυχαία επιλογή των υποπροβλημάτων με τη χρήση της ομοιόμορφης κατανομής. Για προβλήματα με 2000 πελάτες χρειάστηκαν περίπου 3 λεπτά δίνοντας επιτάχυνση 15 φορές για το μοντέλο ενώ η τυχαία επιλογή έδωσε επιτάχυνση 8 φορές.

Έγιναν δύο δοκιμές γενίκευσης η πρώτη σε στιγμιότυπα με 3000 πελάτες και η δεύτερη σε προβλήματα με 2000 πελάτες από πραγματικά δεδομένα περιοχών του Βελγίου. Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκαν δεδομένα με ομοιόμορφη κατανομή των πελατών και με συσταδοποιημένη. Στα προβλήματα με 3000 πελάτες η απόδοση ήταν παρόμοια για τα δύο σύνολα εκπαίδευσης. Στα πραγματικά δεδομένα η εικόνα ήταν διαφορετική με την συσταδοποιημένη εκπαίδευση να έχει καλή ικανότητα γενίκευσης ενώ όταν το μοντέλο εκπαιδεύτηκε με ομοιόμορφα δεδομένα είχε μικρότερη επιτάχυνση από ότι η τυχαία επιλογή υποπροβλημάτων.

Απόδοση σε παραλλαγές Το μοντέλο χρησιμοποιήθηκε και για την επίλυση του *VRPTW* με επιτάχυνση 8 φορές έναντι 6 της τυχαίας επιλογής. Στην επίλυση του *VRPPD* η επιτάχυνση ήταν 31 φορές ενώ η τυχαία επιλογή είχε επιτάχυνση 20 φορές. Η αυστηρή διάταξη των πελατών που καθορίζεται από τα χρονικά παράθυρα αύξησε σημαντικά την δυσκολία επιλογής του μοντέλου.

5. ΣΥΝΟΨΗ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ

5.1 Αποτελέσματα του αλγορίθμου ακέραιου προγραμματισμού

Ξεκινώντας από τα αποτελέσματα του Generic BCP [14] στο οποίο εξετάστηκαν στιγμιότυπα μέχρι 400 πελάτες για την σύγκριση των αποτελεσμάτων. Στην πρώτη στήλη αναφέρεται το πρόβλημα που επιλύεται, στη δεύτερη το σύνολο από το οποίο προέρχονται τα στιγμιότυπα, στην τρίτη ο συνολικός αριθμός των στιγμιότυπων, στην τέταρτη το χρονικό όριο επίλυσης των στιγμιότυπων. Η σύγκριση γίνεται με βάση τον αριθμό των στιγμιότυπων που επιλύθηκαν και τον μέσο χρόνο που χρειάστηκε για κάθε στιγμιότυπο. Τα καλύτερα αποτελέσματα φαίνονται με έντονη γραφή.

Problem	Data set	#	T.L.	Gen. BCP	Best Publ.	2nd Best
CVRP	E-M [25, 26]	12	10h	12 (61s)	12 (49s) [63]	10 (432s) [28]
	X [82]	58	60h	36 (147m)	34 (209m) [82]	—
VRPTW	Solomon Hardest [81]	14	1h	14 (5m)	13 (17m) [62]	9 (39m) [10]
	Hombberger 200 [40]	60	30h	56 (21m)	50 (70m) [62]	7 (-) [50]

Σχήμα 5.1: Η απόδοση του Generic BCP σε σχέση τους δύο άλλους καλύτερους αλγορίθμους

Επίσης ο αλγόριθμος χρησιμοποιήθηκε για 6 στιγμιότυπα τα οποία δεν είχαν λυθεί βέλτιστα, σε 4 από αυτά βρέθηκε νέα καλύτερη λύση η οποία είναι υπογραμμισμένη, ενώ στα άλλα δύο επιβεβαιώθηκε η βελτιστότητα. Στο σχήμα φαίνεται το στιγμιότυπο, η προηγούμενη καλύτερη γνωστή λύση, το κάτω φράγμα στην ρίζα του δέντρου αναζήτησης, ο αριθμός των κόμβων του δέντρου, ο συνολικός χρόνος και τέλος η βέλτιστη λύση.

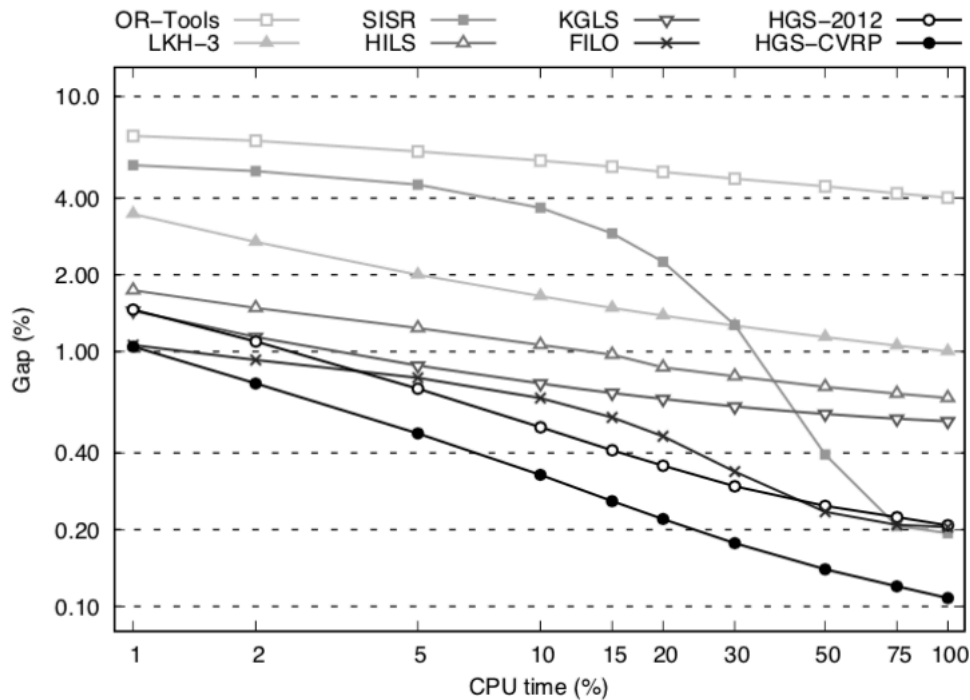
Instance	Prev. BKS	Root LB	Nodes	Total Time	OPT
X-n284-k15	20226	20168	940	11.0 days	<u>20215</u>
X-n322-k28	29834	29731	1197	5.6 days	<u>29834</u>
X-n344-k43	42056	41939	2791	11.6 days	<u>42050</u>
X-n393-k38	38260	38194	1331	5.8 days	<u>38260</u>
X-n469-k138	221909	221585	8964	15.2 days	<u>221824</u>
X-n548-k50	86701	86650	337	2.0 days	<u>86700</u>

Σχήμα 5.2: Αναλυτικά αποτελέσματα στα στιγμιότυπα που επιλύθηκαν βέλτιστα για πρώτη φορά

Με βάση τα αποτελέσματα ο Generic BCP επιλύει βέλτιστα στιγμιότυπα του *VRPTW* μέχρι 100 πελάτες με μέσο χρόνο 5 λεπτά ενώ για 200 πελάτες χρειάζεται περίπου 20 λεπτά. Για μεγαλύτερα στιγμιότυπα του *CVRP* ο μέσος χρόνος ήταν περίπου 2,5 ώρες. Συνεπώς είναι ο κατάλληλος για στιγμιότυπα με σχετικά μικρό αριθμό πελατών και για εφαρμογές στις οποίες είναι διαθέσιμος αρκετός υπολογιστικός χρόνος ή χρειάζεται να βρεθεί η βέλτιστη λύση.

5.2 Σύγκριση σύγχρονων Μεθευριστικών μεθόδων

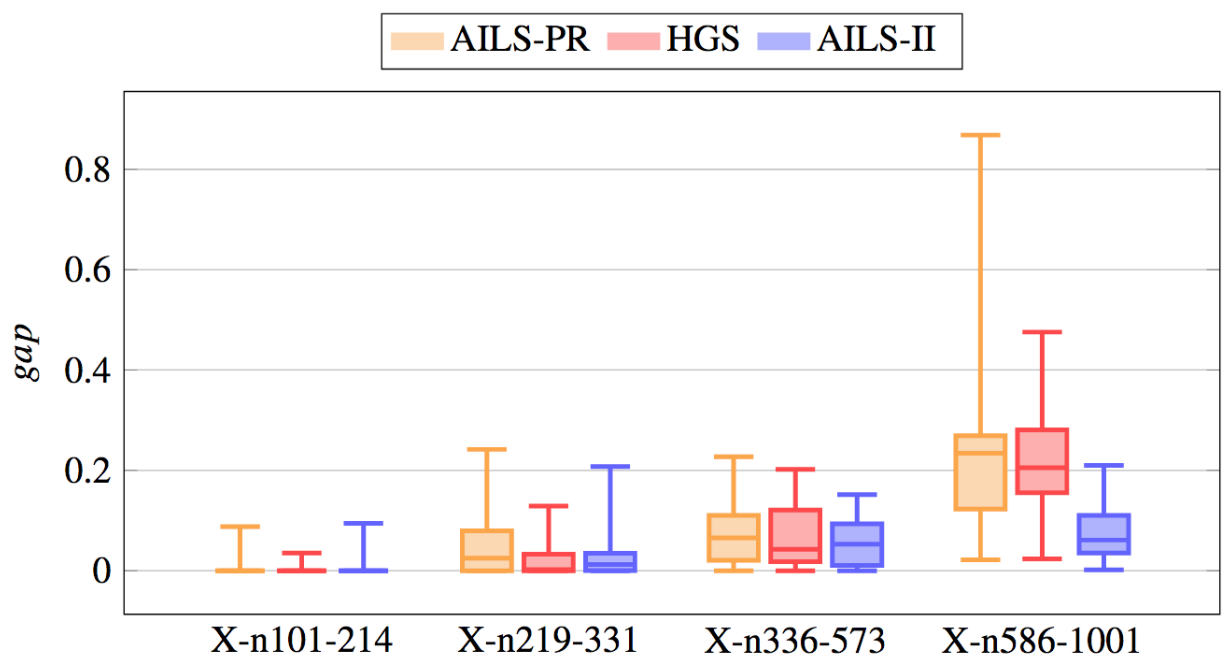
Στο [22] γίνεται σύγκριση της HGS με άλλες μεθευριστικές προσεγγίσεις σε στιγμιότυπα με έως 1000 πελάτες. Στο σχήμα που ακολουθεί φαίνεται η απόκλιση της κάθε μεθόδου σε σχέση με τον συνολικό χρόνο εκτέλεσης.



Σχήμα 5.3: Η HGS παρέχει έως 40 φορές καλύτερη λύση

Στο [24] γίνεται η σύγκριση μεταξύ της HGS και της AILS, τα στιγμιότυπα αρχικά χωρίζονται σε 4 κατηγορίες με βάση τον αριθμό των πελατών. Τα πολύ μικρά από 100 μέχρι 200 πελάτες, τα μικρά από 200 έως 300, τα μεσαία από 300 έως 500 και τα μεγάλα από 500 μέχρι 1000. Στο σχήμα που ακολουθεί φαίνεται η απόκλιση για κάθε κατηγορία και για κάθε μέθοδο με την HGS να έχει τα κόκκινα αποτελέσματα και την AILS να έχει τα μπλέ.

Με βάση τα αποτελέσματα φαίνεται ότι η HGS έχει καλύτερη απόδοση για στιγμιότυπα με μέγεθος μέχρι 500 πελάτες ενώ η AILS είναι καλύτερη για μεγαλύτερα στιγμιότυπα. Τέλος παρουσιάζονται τα αποτελέσματα για κάθε στιγμιότυπο αναλυτικά, στην πρώτη στήλη αναφέρεται το στιγμιότυπο, στην δεύτερη η καλύτερη γνωστή λύση η οποία έχει αστερίσκο όταν είναι βέλτιστη, ενώ για κάθε αλγόριθμο παρουσιάζονται τα αποτελέσματα για 50 εκτελέσεις η μέση λύση, η απόκλιση της σε παρένθεση με βάση την 1.11, η καλύτερη λύση που βρέθηκε και τέλος μέσος χρόνος εύρεσης της καλύτερης λύσης σε κάθε εκτέλεση.



Σχήμα 5.4: Σύγκριση για κάθε κατηγορία στιγμιοτύπων

Instance	BKS	AILS-PR (Máximo and Nascimento, 2021)			HGS (Vidal, 2022)			AILS-II		
		Avg (gap)	Best	T (min)	Avg (gap)	Best	T (min)	Avg (gap)	Best	T (min)
X-n101-k25	27591*	27591.00 (0.0000)	27591	0.12	27591.00 (0.0000)	27591	0.02	27591.00 (0.0000)	27591	0.45
X-n106-k14	26362*	26362.00 (0.0000)	26362	1.81	26371.32 (0.0354)	26362	7.90	26362.00 (0.0000)	26362	0.96
X-n110-k13	14971*	14971.00 (0.0000)	14971	0.02	14971.00 (0.0000)	14971	0.01	14971.00 (0.0000)	14971	0.07
X-n115-k10	12747*	12747.00 (0.0000)	12747	0.04	12747.00 (0.0000)	12747	0.02	12747.00 (0.0000)	12747	0.32
X-n120-k6	13332*	13332.00 (0.0000)	13332	0.19	13332.00 (0.0000)	13332	0.13	13332.00 (0.0000)	13332	0.35
X-n125-k30	55539*	55539.00 (0.0000)	55539	0.73	55539.00 (0.0000)	55539	0.35	55539.00 (0.0000)	55539	2.81
X-n129-k18	28940*	28940.00 (0.0000)	28940	3.13	28940.00 (0.0000)	28940	0.13	28940.00 (0.0000)	28940	0.55
X-n134-k13	10916*	10916.00 (0.0000)	10916	0.36	10916.00 (0.0000)	10916	0.43	10916.00 (0.0000)	10916	0.54
X-n139-k10	13590*	13590.00 (0.0000)	13590	0.20	13590.00 (0.0000)	13590	0.02	13590.00 (0.0000)	13590	0.58
X-n143-k7	15700*	15700.00 (0.0000)	15700	2.12	15700.00 (0.0000)	15700	0.17	15700.00 (0.0000)	15700	0.87
X-n148-k46	43448*	43448.00 (0.0000)	43448	0.11	43448.00 (0.0000)	43448	0.08	43448.00 (0.0000)	43448	0.50
X-n153-k22	21220*	21220.30 (0.0014)	21220	7.91	21224.70 (0.0221)	21220	1.41	21220.90 (0.0042)	21220	9.77
X-n157-k13	16876*	16876.00 (0.0000)	16876	0.12	16876.00 (0.0000)	16876	0.08	16876.00 (0.0000)	16876	0.71
X-n162-k11	14138*	14138.00 (0.0000)	14138	0.42	14138.00 (0.0000)	14138	0.04	14138.00 (0.0000)	14138	0.54
X-n167-k10	20557*	20557.00 (0.0000)	20557	0.43	20557.00 (0.0000)	20557	0.29	20557.00 (0.0000)	20557	1.25
X-n172-k51	45607*	45607.00 (0.0000)	45607	0.36	45607.00 (0.0000)	45607	0.09	45607.00 (0.0000)	45607	4.34
X-n176-k26	47812*	47812.00 (0.0000)	47812	0.44	47812.00 (0.0000)	47812	0.34	47812.00 (0.0000)	47812	0.87
X-n181-k23	25569*	25569.12 (0.0005)	25569	6.92	25569.00 (0.0000)	25569	0.85	25569.00 (0.0000)	25569	1.49
X-n186-k15	24145*	24145.04 (0.0002)	24145	7.63	24145.00 (0.0000)	24145	0.31	24145.00 (0.0000)	24145	3.56
X-n190-k8	16980*	16980.00 (0.0000)	16980	7.03	16981.04 (0.0061)	16980	14.72	16980.00 (0.0000)	16980	6.14
X-n195-k51	44225*	44225.00 (0.0000)	44225	0.71	44225.00 (0.0000)	44225	2.68	44231.68 (0.0151)	44225	17.43
X-n200-k36	58578*	58578.00 (0.0000)	58578	2.51	58578.00 (0.0000)	58578	5.69	58578.00 (0.0000)	58578	6.69
X-n204-k19	19565*	19565.00 (0.0000)	19565	5.86	19565.00 (0.0000)	19565	1.61	19565.00 (0.0000)	19565	2.20
X-n209-k16	30656*	30656.00 (0.0000)	30656	2.59	30656.00 (0.0000)	30656	5.07	30656.06 (0.0002)	30656	8.27
X-n214-k11	10856*	10865.52 (0.0877)	10856	13.38	10859.64 (0.0336)	10856	15.78	10866.24 (0.0943)	10856	12.81
X-n219-k73	117595*	117595.00 (0.0000)	117595	0.27	117595.00 (0.0000)	117595	5.34	117595.00 (0.0000)	117595	0.36
X-n223-k34	40437*	40437.00 (0.0000)	40437	12.16	40437.00 (0.0000)	40437	3.89	40437.16 (0.0004)	40437	9.49
X-n228-k23	25742*	25742.76 (0.0030)	25742	7.28	25742.26 (0.0010)	25742	12.41	25742.00 (0.0000)	25742	5.58
X-n233-k16	19230*	19230.60 (0.0031)	19230	11.89	19230.00 (0.0000)	19230	2.90	19230.32 (0.0017)	19230	7.38
X-n237-k14	27042*	27042.88 (0.0033)	27042	8.14	27042.00 (0.0000)	27042	2.53	27043.04 (0.0038)	27042	5.50
X-n242-k48	82751*	82818.38 (0.0814)	82764	15.19	82778.45 (0.0332)	82751	16.43	82759.36 (0.0101)	82751	20.46
X-n247-k50	37274*	37274.00 (0.0000)	37274	0.93	37274.00 (0.0000)	37274	14.80	37274.00 (0.0000)	37274	4.35
X-n251-k28	38684*	38715.06 (0.0803)	38684	24.74	38684.79 (0.0020)	38684	14.59	38695.04 (0.0285)	38684	16.74
X-n256-k16	18839	18879.28 (0.2138)	18864	1.32	18839.00 (0.0000)	18839	6.97	18878.12 (0.2077)	18839	2.65

Σχήμα 5.5: Αποτελέσματα για σιμιότυπα από 100 μέχρι 250 πελάτες

Instance	BKS	AILS-PR (Máximo and Nascimento, 2021)			HGS (Vidal, 2022)			AILS-II		
		Avg (gap)	Best	T (min)	Avg (gap)	Best	T (min)	Avg (gap)	Best	T (min)
X-n261-k13	26558*	26559.40 (0.0053)	26558	20.04	26558.05 (0.0002)	26558	16.15	26558.96 (0.0036)	26558	17.49
X-n266-k58	75478*	75551.94 (0.0980)	75529	17.92	75575.22 (0.1288)	75478	21.10	75495.04 (0.0226)	75478	17.04
X-n270-k35	35291*	35299.74 (0.0248)	35291	14.88	35302.67 (0.0331)	35291	5.32	35303.20 (0.0346)	35291	15.68
X-n275-k28	21245*	21245.00 (0.0000)	21245	2.99	21245.00 (0.0000)	21245	0.58	21245.00 (0.0000)	21245	1.92
X-n280-k17	33503	33529.64 (0.0795)	33503	20.48	33515.24 (0.0365)	33503	23.34	33532.30 (0.0875)	33503	24.96
X-n284-k15	20215*	20252.02 (0.1831)	20225	25.27	20237.56 (0.1116)	20225	27.86	20236.86 (0.1081)	20215	28.16
X-n289-k60	95151*	95203.30 (0.0550)	95151	27.53	95244.55 (0.0983)	95185	24.22	95196.34 (0.0477)	95151	30.74
X-n294-k50	47161	47174.56 (0.0288)	47167	20.97	47172.48 (0.0244)	47161	26.91	47176.28 (0.0324)	47161	28.09
X-n298-k31	34231*	34231.00 (0.0000)	34231	4.93	34231.00 (0.0000)	34231	8.64	34235.86 (0.0142)	34231	11.18
X-n303-k21	21736	21749.56 (0.0624)	21746	18.44	21740.97 (0.0229)	21738	24.25	21746.92 (0.0502)	21736	28.32
X-n308-k13	25859	25921.52 (0.2418)	25859	22.79	25863.84 (0.0187)	25861	23.81	25873.44 (0.0558)	25861	23.58
X-n313-k71	94043	94084.56 (0.0442)	94046	31.13	94082.84 (0.0424)	94045	28.20	94062.72 (0.0210)	94044	33.03
X-n317-k53	78355*	78355.00 (0.0000)	78355	6.58	78355.83 (0.0011)	78355	14.79	78355.00 (0.0000)	78355	3.67
X-n322-k28	29834*	29845.74 (0.0394)	29834	24.01	29842.10 (0.0272)	29834	20.96	29837.70 (0.0124)	29834	18.58
X-n327-k20	27532	27559.54 (0.1000)	27532	23.39	27541.55 (0.0347)	27532	25.11	27567.28 (0.1281)	27532	27.12
X-n331-k15	31102*	31103.24 (0.0040)	31103	16.76	31102.97 (0.0031)	31102	19.98	31103.00 (0.0032)	31103	12.44
X-n336-k84	139111	139202.40 (0.0657)	139111	33.24	139260.93 (0.1078)	139174	35.82	139204.78 (0.0674)	139130	37.12
X-n344-k43	42050	42065.34 (0.0365)	42050	26.81	42067.18 (0.0409)	42050	26.06	42089.32 (0.0935)	42055	27.17
X-n351-k40	25896	25936.98 (0.1582)	25917	37.70	25942.56 (0.1798)	25925	36.86	25935.22 (0.1515)	25900	39.69
X-n359-k29	51505	51540.96 (0.0698)	51505	33.63	51601.33 (0.1870)	51512	36.95	51524.56 (0.0380)	51505	39.74
X-n367-k17	22814	22814.00 (0.0000)	22814	6.80	22814.00 (0.0000)	22814	3.07	22814.88 (0.0039)	22814	26.08
X-n376-k94	147713*	147713.00 (0.0000)	147713	4.54	147713.40 (0.0003)	147713	20.72	147713.00 (0.0000)	147713	2.75
X-n384-k52	65928	66022.52 (0.1434)	65941	41.44	66016.48 (0.1342)	65984	34.09	65974.30 (0.0702)	65938	43.71
X-n393-k38	38260*	38262.50 (0.0065)	38260	24.40	38260.00 (0.0000)	38260	7.44	38268.86 (0.0232)	38260	31.96
X-n401-k29	66154	66202.66 (0.0736)	66169	39.97	66215.33 (0.0927)	66179	29.67	66188.96 (0.0528)	66162	50.37
X-n411-k19	19712	19756.76 (0.2271)	19723	33.66	19716.70 (0.0238)	19712	36.02	19731.02 (0.0965)	19717	53.38
X-n420-k130	107798*	107837.44 (0.0366)	107818	37.92	107817.35 (0.0179)	107798	37.51	107809.56 (0.0107)	107798	39.53
X-n429-k61	65449	65484.60 (0.0544)	65453	33.49	65477.05 (0.0429)	65449	32.18	65495.94 (0.0717)	65457	48.42
X-n439-k37	36391*	36408.14 (0.0471)	36391	24.21	36395.09 (0.0112)	36391	15.96	36400.10 (0.0250)	36395	18.04
X-n449-k29	55233	55290.32 (0.1038)	55236	42.46	55334.05 (0.1829)	55296	40.67	55288.26 (0.1000)	55240	57.30
X-n459-k26	24139	24160.02 (0.0871)	24140	41.15	24152.71 (0.0568)	24139	30.59	24163.66 (0.1022)	24139	42.67
X-n469-k138	221824*	221958.92 (0.0608)	221841	50.16	222091.90 (0.1208)	221992	36.13	221943.28 (0.0538)	221842	63.41
X-n480-k70	89449	89467.74 (0.0210)	89449	52.73	89490.80 (0.0467)	89458	44.60	89456.70 (0.0086)	89449	39.71
X-n491-k59	66483	66556.18 (0.1101)	66487	59.37	66603.26 (0.1809)	66516	48.59	66545.86 (0.0946)	66489	61.62

Σχήμα 5.6: Αποτελέσματα για στιμιότυπα από 250 μέχρι 500 πελάτες

Instance	BKS	AILS-PR (Máximo and Nascimento, 2021)			HGS (Vidal, 2022)			AILS-II		
		Avg (gap)	Best	T (min)	Avg (gap)	Best	T (min)	Avg (gap)	Best	T (min)
X-n502-k39	69226	69234.96 (0.0129)	69226	42.59	69238.95 (0.0187)	69228	35.63	69231.64 (0.0081)	69226	52.39
X-n513-k21	24201	24229.80 (0.1190)	24201	44.99	24201.00 (0.0000)	24201	8.88	24229.02 (0.1158)	24201	32.70
X-n524-k153	154593*	154596.36 (0.0022)	154593	25.86	154650.40 (0.0371)	154605	39.76	154601.74 (0.0057)	154593	67.18
X-n536-k96	94846	94972.96 (0.1339)	94906	69.21	95037.56 (0.2020)	94977	53.29	94896.02 (0.0527)	94830	65.29
X-n548-k50	86700*	86801.28 (0.1168)	86704	53.01	86744.53 (0.0514)	86706	51.76	86719.22 (0.0222)	86700	47.30
X-n561-k42	42717	42754.86 (0.0886)	42719	62.08	42733.76 (0.0392)	42719	45.53	42751.34 (0.0804)	42728	72.24
X-n573-k30	50673	50763.96 (0.1795)	50701	68.02	50767.38 (0.1862)	50744	58.32	50730.34 (0.1132)	50680	81.46
X-n586-k159	190316	190397.26 (0.0427)	190336	67.26	190519.19 (0.1068)	190427	53.50	190379.70 (0.0335)	190316	72.57
X-n599-k92	108451	108586.38 (0.1248)	108455	69.63	108619.38 (0.1553)	108563	71.66	108553.66 (0.0947)	108472	77.41
X-n613-k62	59535	59583.44 (0.0814)	59544	75.86	59668.40 (0.2241)	59582	46.95	59569.56 (0.0580)	59536	74.25
X-n627-k43	62164	62326.34 (0.2611)	62221	74.96	62324.47 (0.2581)	62264	71.23	62205.46 (0.0667)	62168	89.10
X-n641-k35	63682	63831.40 (0.2346)	63752	77.06	63836.07 (0.2419)	63751	58.86	63752.26 (0.1103)	63696	79.87
X-n655-k131	106780*	106803.30 (0.0218)	106780	68.71	106805.07 (0.0235)	106780	49.13	106781.94 (0.0018)	106780	45.80
X-n670-k130	146332	146905.48 (0.3919)	146445	73.07	146742.92 (0.2808)	146606	63.03	146639.28 (0.2100)	146423	98.48
X-n685-k75	68205	68298.62 (0.1373)	68246	70.04	68345.00 (0.2053)	68310	70.69	68281.58 (0.1123)	68227	97.30
X-n701-k44	81923	82143.14 (0.2687)	81954	82.05	82158.31 (0.2872)	82038	65.68	81981.40 (0.0713)	81923	101.54
X-n716-k35	43373	43474.50 (0.2340)	43380	88.43	43492.31 (0.2751)	43467	53.27	43397.96 (0.0575)	43369	99.92
X-n733-k159	136187	136354.38 (0.1229)	136257	87.13	136393.08 (0.1513)	136315	42.73	136270.20 (0.0611)	136190	101.67
X-n749-k98	77269	77496.82 (0.2948)	77382	94.80	77616.83 (0.4502)	77513	64.42	77387.72 (0.1536)	77321	111.73
X-n766-k71	114417	114688.16 (0.2370)	114493	99.56	114712.67 (0.2584)	114647	79.62	114457.30 (0.0352)	114425	121.56
X-n783-k48	72386	72715.06 (0.4546)	72509	111.15	72730.25 (0.4756)	72580	83.51	72462.08 (0.1051)	72382	110.22
X-n801-k40	73305	73626.22 (0.4382)	73435	79.97	73444.50 (0.1903)	73374	79.55	73356.02 (0.0696)	73306	100.75
X-n819-k171	158121	158349.54 (0.1445)	158208	118.80	158432.64 (0.1971)	158332	58.30	158215.00 (0.0594)	158143	123.45
X-n837-k142	193737	194059.30 (0.1664)	193906	91.09	194133.36 (0.2046)	193902	77.74	193807.54 (0.0364)	193735	118.31
X-n856-k95	88965*	89060.46 (0.1073)	88967	93.89	89018.27 (0.0599)	88983	70.09	89016.98 (0.0584)	88965	88.97
X-n876-k59	99299	99539.58 (0.2423)	99419	125.68	99624.45 (0.3278)	99484	111.50	99413.80 (0.1156)	99338	133.27
X-n895-k37	53860	54058.18 (0.3680)	53932	95.17	54097.80 (0.4415)	54053	100.79	53968.12 (0.2007)	53854	121.42
X-n916-k207	329179	329562.74 (0.1166)	329351	116.67	329782.40 (0.1833)	329653	115.97	329277.62 (0.0300)	329180	134.71
X-n936-k151	132715	133072.02 (0.2690)	132883	125.49	133326.80 (0.4610)	133198	88.83	132933.16 (0.1644)	132803	146.13
X-n957-k87	85465	85647.98 (0.2141)	85514	126.17	85526.10 (0.0715)	85512	89.54	85489.56 (0.0287)	85464	121.11
X-n979-k58	118976	120009.00 (0.8682)	119244	147.63	119195.11 (0.1842)	119130	101.92	118992.68 (0.0140)	118961	151.30
X-n1001-k43	72355	72737.48 (0.5286)	72558	124.45	72654.67 (0.4142)	72601	115.18	72469.70 (0.1585)	72382	155.29
Average		(0.0977)		38.19	(0.0881)		31.79	(0.0456)		42.96

Σχήμα 5.7: Αποτελέσματα για σιμιότυπα από 500 μέχρι 1000 πελάτες

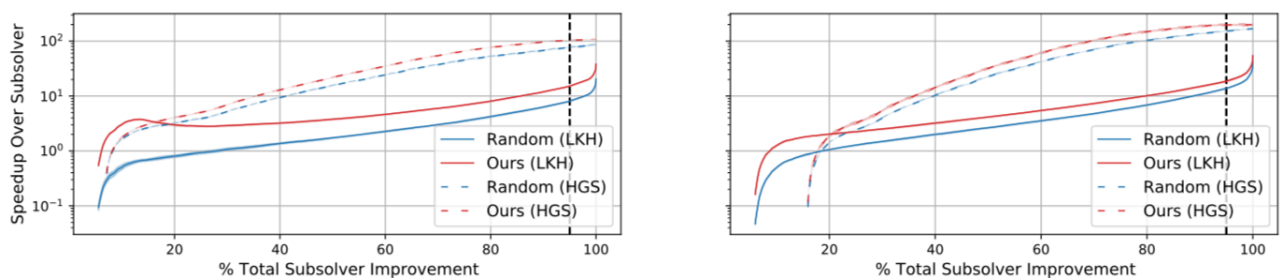
5.3 Επιτάχυνση των Μεθουριστικών σε συνδιασμό με τη Μηχανική Μάθηση

Τέλος στο [26] παρουσιάζονται πολύ καλά αποτελέσματα με την επίλυση στιγμιοτύπων με 2000 πελάτες σε μόλις 3 λεπτά! Για την επιβεβαίωση της ικανότητας του μοντέλου να επιλέγει υποπροβλήματα χρησιμοποιούνται 3 επιλογές υποπροβλημάτων ως βάση αναφοράς, πρώτη η τυχαία επιλογή, δεύτερη η επιλογή υποπροβλημάτων ως τη διαδρομή της οποίας οι πελάτες έχουν επιλεγεί το λιγότερο σε προηγούμενα βήματα, τρίτη η επιλογή του υποπροβλήματος το οποίο έχει τη μέγιστη απόσταση από το προηγούμενο υποπρόβλημα που επιλέχθηκε. Στο σχήμα που ακολουθεί παρουσιάζονται τα αποτελέσματα με την χρήση του LKH-3 ως μέθοδο τοπικής αναζήτησης, για τον οποίο παρουσιάζονται αποτελέσματα μέχρι τη εύρεση λύσης με απόκλιση 5% LKH-3(95%) ή την εκτέλεση για 30.000 επαναλήψεις LKH-3(30k), ενώ συγκρίνεται και με άλλες μεθόδους μηχανικής μάθησης και τοπικής αναζήτησης ενώ σε παρένθεση φαίνεται η επιτάχυνση σε σχέση με τον LKH-3(95%).

Method	N = 500		N = 1000		N = 2000	
	Cost	Time	Cost	Time	Cost	Time
LKH-3 (95%)	62.00	4.4min	120.02	18min	234.89	52min
LKH-3 (30k)	61.87	30min	119.88	77min	234.65	149min
OR Tools	65.59	15min	126.52	15min	244.65	15min
AM sampling	69.08	4.70s	151.01	17.40s	356.69	32.29s
AM greedy	68.58	25ms	142.84	56ms	307.86	147ms
NeuRewriter	73.60	58s	136.29	2.3min	257.61	8.1min
Random	61.99	71s (3.8x)	120.02	3.2min (5.5x)	234.88	6.4min (8.0x)
Count-based	61.99	59s (4.5x)	120.02	2.1min (8.2x)	234.88	5.3min (10x)
Max Min	61.99	59s (4.5x)	120.02	2.5min (7.0x)	234.89	5.2min (10x)
Ours (Short)	61.99	38s (7.0x)	119.87	1.5min (12x)	234.89	3.4min (15x)
Ours (Long)	61.70	76s	119.55	3.0min	233.86	6.8min

Σχήμα 5.8: Αποτελέσματα του μοντέλου επιλογής υποπροβλημάτων σε σχέση με άλλες μεθόδους

Επίσης έγινε και η χρήση του αλγορίθμου HGS ως μέθοδο για την βελτίωση των λύσεων και φάνηκε να υπάρχει σημαντική επιτάχυνση σε σχέση με τη χρήση του LKH-3, καθώς ο HGS φαίνεται να είναι σχεδιασμένος για την επίλυση στιγμιοτύπων με περίπου 500 πελάτες. Στο σχήμα φαίνεται η επιτάχυνση σε σχέση με τη σύγκλιση των αλγορίθμων τοπικής αναζήτησης, αριστερά για στιγμιότυπα με 2000 πελάτες και δεξιά με 3000.



Σχήμα 5.9: Σύγκριση της επιτάχυνσης με την χρήση του HGS και του LKH-3

Τέλος όλες οι σύγχρονες τεχνικές επίλυσης ενσωματώνουν με τον δικό τους τρόπο την έννοια της τοπικότητας, δηλαδή ότι οι κοντινότεροι πελάτες έχουν μεγαλύτερο αντίκτυπο σε σχέση με πιο μακρινούς προκειμένου να εξερευνηθούν πιο αποτελεσματικά τον εκθετικό χώρο αναζήτησης του προβλήματος.

ABBREVIATIONS - ACRONYMS

VRPTW	Vehicle Routing Problem with Time Windows
VRPPD	Vehicle Routing Problem with Pick-up and Delivery
VRPB	Vehicle Routing Problem with Backhauls
EVRP	Electric Vehicle Routing Problem
ILS	Iterated Local Search
UHGS	Unified Hybrid Genetic Search
CVRP	Capacitated Vehicle Routing Problem
BCP	Branch-Cut-and-Price
BB	Branch and Bound
CG	Column Generation
SPPRC	Sortest Path Problem with Resource Constraints
SRC	Subset Row Cuts
opt	optimal
AILS	Adaptive Iterated Local Search

TABLE OF TERMINOLOGY

compact	συμπαγής
extended	εκτεταμένη
pricing problem	πρόβλημα τιμολόγησης
robust	ανθεκτικός
nonrobust	μη ανθεκτικός
limited memory	περιορισμένη μνήμη
bucket	Μέρος της διαμέρισης των δεδομένων
graph	γράφος
swap	ανταλλαγή
relocate	μεταφέρω
optimal	βέλτιστος
exchange	ανταλλαγή
closest	εγγύτεροι

REFERENCES

- [1] P. Munari, T. Dollevoet, and R. Spliet, "A generalized formulation for vehicle routing problems [Online]," Sep 2017. Available: <https://arxiv.org/abs/1606.01935>.
- [2] I. Lima, "(2022) CVRPLIB - plotted instances," [Online]. Available: <http://vrp.galgos.inf.puc-rio.br/index.php/en/plotted-instances>.
- [3] G. D. Luciano Costa, Claudio Contardo, "Exact Branch-Price-and-Cut Algorithms for Vehicle Routing," *Transportation Science*, vol. 53, no. 4, pp. 946–985, Jun 2019.
- [4] R. Lahyani, M. Khemakhem, and F. Semet, "Rich vehicle routing problems: From a taxonomy to a definition," *European Journal of Operational Research*, vol. 241, no. 1, pp. 1–14, Feb 2015.
- [5] W. Chowmali and S. Sukto, "A novel two-phase approach for solving the multi-compartment vehicle routing problem with a heterogeneous fleet of vehicles: a case study on fuel delivery," *Decision Science Letters*, vol. 9, no. 1, pp. 77–90, Jan 2020.
- [6] Çağrı Koç and G. Laporte, "Vehicle routing with backhauls: Review and research perspectives," *Computers Operations Research*, vol. 91, pp. 79–91, Mar 2018.
- [7] P. Ballesteros Silva and A. Escobar Zuluaga, "Review of state of the art vehicle routing problem with pickup and delivery (VRPPD)," *Ingeniería y Desarrollo*, vol. 34, pp. 463–482, Aug. 2016.
- [8] I. Kucukoglu, R. Dewil, and D. Cattrysse, "The electric vehicle routing problem and its variations: A literature review," *Computers Industrial Engineering*, vol. 161, p. 107650, Nov 2021.
- [9] C. Hecht, K. Victor, S. Zurmühlen, and D. U. Sauer, "Electric vehicle route planning using real-world charging infrastructure in Germany," *eTransportation*, vol. 10, p. 100143, Nov 2021.
- [10] Z. Wang and J.-B. Sheu, "Vehicle routing problem with drones," *Transportation Research Part B: Methodological*, vol. 122, pp. 350–364, Apr 2019.
- [11] M. Le, J. Gao, and C. Zhan, "Solving the airline recovery problem based on vehicle routing problem with time window modeling and genetic algorithm," in *2013 Ninth International Conference on Natural Computation (ICNC)*, pp. 822–828, Jul 2013.
- [12] L.-P. Trottier and J.-F. Cordeau, "Solving the vessel routing and scheduling problem at a Canadian maritime transportation company," *INFOR: Information Systems and Operational Research*, vol. 57, no. 2, pp. 260–285, Feb 2019.
- [13] E. Uchoa, D. Pecin, A. Pessoa, M. Poggi, T. Vidal, and A. Subramanian, "New benchmark instances for the Capacitated Vehicle Routing Problem," *European Journal of Operational Research*, vol. 257, no. 3, pp. 845–858, Mar 2017.
- [14] A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck, "A generic exact solver for vehicle routing and related problems," *Mathematical Programming*, vol. 183, no. 1, pp. 483–523, Jun 2020.
- [15] S. Røpke, "Heuristic and exact algorithms for vehicle routing problems". Ph.D. dissertation, Dept. of Computer Sci., Univ. of Copenhagen, Copenhagen, Denmark, 2006.
- [16] D. Pecin, A. Pessoa, M. Poggi, E. Uchoa, and H. Santos, "Limited memory Rank-1 Cuts for Vehicle Routing Problems," *Operations Research Letters*, vol. 45, no. 3, pp. 206–209, May 2017.
- [17] D. Pecin, C. Contardo, G. Desaulniers, and E. Uchoa, "New Enhancements for the Exact Solution of the Vehicle Routing Problem with Time Windows," *INFORMS Journal on Computing*, vol. 29, no. 3, pp. 489–502, Jun 2017.
- [18] R. Sadykov, E. Uchoa, and A. Pessoa, "A Bucket Graph-Based Labeling Algorithm with Application to Vehicle Routing," *Transportation Science*, vol. 55, no. 1, pp. 4–28, Oct 2020.
- [19] P. Diego, P. A. P. Marcus, and U. Eduardo, "Improved branch-cut-and-price for capacitated vehicle routing," *Mathematical Programming Computation*, vol. 9, no. 1, pp. 61–100, Mar 2017.
- [20] A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck, "Automation and combination of linear-programming based stabilization techniques in column generation," *INFORMS Journal on Computing*, vol. 30, no. 2, pp. 339–360, May 2018.

- [21] O. Bräysy and M. Gendreau, “Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms,” *Transportation Science*, vol. 39, no. 1, pp. 104–118, Feb 2015.
- [22] T. Vidal, “Hybrid Genetic Search for the CVRP: Open-Source Implementation and SWAP* Neighborhood,” *Computers Operations Research*, vol. 140, p. 105643, Apr 2022.
- [23] T. Vidal, “Technical note: Split algorithm in $O(n)$ for the capacitated vehicle routing problem,” *Computers Operations Research*, vol. 69, pp. 40–47, May 2016.
- [24] V. R. Máximo, J.-F. Cordeau, and M. C. V. Nascimento, “AILS-II: An Adaptive Iterated Local Search Heuristic for the Large-scale Capacitated Vehicle Routing Problem [Online],” May 2022 Available: <https://doi.org/10.48550/arxiv.2205.12082>.
- [25] W. Kool, H. van Hoof, and M. Welling, “Attention, Learn to Solve Routing Problems!,” in *International Conference on Learning Representations*, May 2019.
- [26] S. Li, Z. Yan, and C. Wu, “Learning to delegate for large-scale vehicle routing,” in *Advances in Neural Information Processing Systems*, vol. 34, pp. 26198–26211, Dec 2021.
- [27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is All you Need,” vol. 30, pp. 5998–6008, Dec 2017.