ΜΟΝΤΕΛΑ ΜΕΙΚΤΟΥ ΑΚΕΡΑΙΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΓΙΑ ΤΗΝ ΔΡΟΜΟΛΟΓΗΣΗ ΟΧΗΜΑΤΩΝ ΠΕΡΙΟΡΙΣΜΕΝΗΣ ΧΩΡΗΤΙΚΟΤΗΤΑΣ ΜΕ ΧΡΟΝΙΚΟΥΣ ΠΕΡΙΟΡΙΣΜΟΥΣ ΚΑΙ ΧΡΟΝΙΚΑ ΠΑΡΑΘΥΡΑ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ,

ΔΙΟΙΚΗΣΗ LOGISTICS

από

ΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΣΟΜΠΑΝΙΔΗΣ ΒΑΣΙΛΗΣ

ΤΜΗΜΑ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ &

ΤΕΧΝΟΛΟΓΙΑΣ, 2021

ΔΗΛΩΣΗ

Δηλώνω υπεύθυνα ότι:

«Η εργασία αυτή είναι πρωτότυπη και εκπονήθηκε αποκλειστικά και μόνο για την απόκτηση του συγκεκριμένου μεταπτυχιακού τίτλου. Τα πνευματικά δικαιώματα χρησιμοποίησης του μη πρωτότυπου υλικού ΜΔΕ ανήκουν στο μεταπτυχιακό φοιτητή και το επιβλέπον μέλος ΔΕΠ εις ολόκληρο, δηλαδή εκάτερος μπορεί να κάνει χρήση αυτών χωρίς τη συναίνεση άλλου. Τα πνευματικά δικαιώματα χρησιμοποίησης του πρωτότυπου μέρους ΜΔΕ ανήκουν στον μεταπτυχιακό φοιτητή και τον επιβλέποντα από κοινού, δηλαδή δεν μπορεί ο ένας από τους δύο να κάνει χρήση αυτού χωρίς τη συναίνεση του άλλου. Κατ' εξαίρεση, επιτρέπεται η δημοσίευση του πρωτότυπου μέρους της διπλωματικής εργασίας σε επιστημονικό περιοδικό ή πρακτικά συνεδρίου από τον ένα εκ των δύο, με την προϋπόθεση ότι αναφέρονται τα ονόματα και των δύο ως συν-συγγραφέων. Στην περίπτωση αυτή προηγείται γραπτή ενημέρωση του μη συμμετέχοντα στη συγγραφή του επιστημονικού άρθρου. Δεν επιτρέπεται η κατά οποιοδήποτε τρόπο δημοσιοποίηση υλικού το οποίο έχει δηλωθεί εγγράφως ως απόρρητο».

Ο μεταπτυχιακός φοιτητής

Τσομπανίδης Βασίλης

(Υπογραφή)

Ευχαριστίες

Ευχαριστώ την οικογένεια μου για την υπομονή και την ενθάρρυνση σε δύσκολες στιγμές, καθώς και τους φίλους μου Αλέξανδρο και Φώτη για την πολύτιμη βοήθεια τους.

Επίσης ευχαριστώ θερμά τον καθηγητή μου και επιβλέποντα της διπλωματικής εργασίας κ. Παύλο Ειρηνάκη για τις υποδείξεις αλλά και την άριστη συνεργασία κατά την διάρκεια εκπόνησης της εργασίας.

Περίληψη

Τα προβλήματα δρομολόγησης οχημάτων έχουν αποτελέσει αντικείμενο ενδελεχούς έρευνας από την επιστημονική κοινότητα τις τελευταίες δεκαετίες λόγω του θεωρητικού ενδιαφέροντος που παρουσιάζουν αλλά και των θετικών οικονομικών αποτελεσμάτων που αποφέρει η βέλτιστη λύση τους. Στη διεθνή βιβλιογραφία συναντάτε μια πληθώρα μοντέλων δρομολόγησης οχημάτων τα οποία εξειδικεύονται στην προσομοίωση προβλημάτων που συναντώνται στο περιβάλλον των μεταφορών. Η επίλυση των διαφόρων μοντέλων δρομολόγησης με χρήση πακέτων βελτιστοποίησης έχει αποδειχτεί ότι μπορεί να μειώσει το κόστος μεταφορών στα Logistics έως και 20% (Λάιος, 2010).

Στην παρούσα διπλωματική εργασία παρουσιάζονται κάποια από τα πιο κλασικά μοντέλα δρομολόγησης που συναντώνται στην διεθνή βιβλιογραφία. Στη συνέχεια επιχειρείται η δημιουργία δυο νέων μοντέλων δρομολόγησης τα οποία βασίζονται στα μοντέλα δρομολόγησης οχημάτων με χρονικά παράθυρα και περιορισμένη χωρητικότητα. Τα μοντέλα διαθέτουν αυξημένη πληροφορία αναφορικά με τις δυνατότητες των οχημάτων που απαρτίζουν τον στόλο αλλά και περιορισμούς που εξασφαλίζουν την ελαχιστοποίηση των οχημάτων και την τήρηση του εργασιακού οκταώρου.

Τέλος, τα αξιολογούνται με πειράματα τα οποία εξετάζουν την σωστή λειτουργία τους καθώς και το μέγεθος των προβλημάτων που είναι ικανά να λύσουν. Στα πλαίσια της εργασίας πραγματοποιείται επίσης σύγκριση της χρονικής αποτελεσματικότητας των δύο μοντέλων με τα λογισμικά cplex και gurobi.

Summary

Vehicle routing problems have been the focus of extensive research by the scientific community in recent decades driven by their economic importance and theoretical interest. A myriad of vehicle routing variations can be found in scientific literature emulating real-life distribution problems. The use of optimization techniques based on mathematical programming has been proven to reduce distribution cost up to 20% (Λάιος, 2010).

In this thesis a review of some of the most known vehicle routing problem variations is provided. In addition, we attempt to create two new vehicle routing models based on the vehicle routing models with time windows and the capacitated vehicle routing models we find in scientific literature. The new models receive additional information regarding the capabilities of the trucks that form the fleet. Furthermore, the models are enhanced with constraints that minimize truck usage and enforce a legal working schedule for the drivers.

Experiments are run on both models ensuring that they work as intended and exploring the maximum size of problems that the models are capable of solving. A comparison between the time efficiency of the cplex and gurobi optimizers is also made.

Περιεχόμενα

Ευρετήριο Πινάκων

Ευρετήριο Εικόνων

1. Εισαγωγή

Η παρούσα διπλωματική εργασία γίνεται στα πλαίσια φοίτησης στο μεταπτυχιακό πρόγραμμα του τμήματος Βιομηχανικής Διοίκησης και Τεχνολογίας του Πανεπιστημίου Πειραιώς με κατεύθυνση την *Διοίκηση Logistics.* Στόχος της εργασίας είναι αρχικά η μελέτη και κατανόηση της βιβλιογραφίας που αφορά στην δρομολόγηση οχημάτων. Ιστορικά το πρόβλημα της δρομολόγησης οχημάτων εκφράζεται για πρώτη φορά από τους (Dantzig & Ramser, 1959) οι οποίοι δημιουργούν ένα μοντέλο μαθηματικού προγραμματισμού καθώς και έναν αλγόριθμο επίλυσης αυτού. Η δρομολόγηση οχημάτων (Vehicle Routing) αποτελεί μία γενίκευση του προβλήματος του περιοδεύοντος πωλητή (Traveling Salesman Problem - TSP). Πρόκειται για δύο από τα πιο γνωστά προβλήματα βελτιστοποίησης αλλά και βασικά προβλήματα μεταφοράς που έχουν απασχολήσει την επιστημονική κοινότητα με πολλές παραλλαγές και πλούσια βιβλιογραφία. Στην κλασική μορφή του προβλήματος TSP ένας πωλητής καλείται να εξυπηρετήσει τις ανάγκες διαφόρων πελατών σε μία επίσκεψη και στην συνέχεια να επιστρέψει στο σημείο από το οποίο ξεκίνησε έχοντας διανύσει την μικρότερη δυνατή απόσταση. Στόχος του προβλήματος είναι η εύρεση της βέλτιστης διαδρομής με την προϋπόθεση ότι θα εξυπηρετηθούν όλοι οι πελάτες από μια φορά. Αντίστοιχα το κλασικό πρόβλημα δρομολόγησης οχημάτων στοχεύει :

- στον ορισμό των βέλτιστων διαδρομών που απαιτούνται για την μεταφορά των προϊόντων από μια κεντρική αποθήκη στους πελάτες με το μικρότερο δυνατό κόστος
- στην ελαχιστοποίηση του αριθμού οχημάτων που απαιτούνται για την κάλυψη των μεταφορικών αναγκών
- στην επίτευξη υψηλών επιπέδων ικανοποίησης των πελατών για θέματα που αφορούν χρόνους παράδοσης, μέγεθος παραγγελιών, κ.α.

Είναι λοιπόν προφανές ότι το πρόβλημα δρομολόγησης οχημάτων αποτελεί μια γενίκευση του προβλήματος του περιοδεύοντος πωλητή με την διαφορά ότι στο πρώτο χρησιμοποιείται ένας στόλος οχημάτων.

Για το πρόβλημα δρομολόγησης οχημάτων έχουν υπάρξει διάφορες παραλλαγές όπως τα προβλήματα δρομολόγησης με χρονοπαράθυρα (Vehicle Routing Problem with Time Windows - VRPTW), με περιορισμένη χωρητικότητα (Capacitated Vehicle Routing Problem - CVRP), με δυνατότητα backhaul (Vehicle Routing Problem with Backhauls – VRPB), με πολλαπλές αποθήκες (Vehicle Routing Problem with Multiple Depot – MDVRP), με διανομή και παραλαβή κατά τη διάρκεια της διαδρομής (Vehicle Routing Problem with Pickup and Delivery – VRPPD), κ.α., κάποιες από τις οποίες παρουσιάζονται αναλυτικά στο κεφάλαιο 3. Στην συγκεκριμένη εργασία δίνεται ιδιαίτερη έμφαση στα προβλήματα δρομολόγησης με χρονοπαράθυρα (VRPTW) και περιορισμένη χωρητικότητα (CVRP) με βάση τα οποία επιχειρείται η ανάπτυξη δύο μοντέλων δρομολόγησης. Τα μοντέλα που αναπτύσσονται δέχονται αυξημένη πληροφορία αναφορικά με τις δυνατότητες των οχημάτων που αποτελούν τον στόλο σε κάθε περίπτωση καθώς διαθέτουν τρισδιάστατη μεταβλητή κόστους – χρόνου. Επιπλέον διαθέτουν περιορισμούς που στοχεύουν στην τήρηση του εργασιακού οκταώρου για τους οδηγούς των οχημάτων, ελαχιστοποίηση των χρησιμοποιηθέντων οχημάτων κ.α.

Τα μοντέλα που αναπτύσσονται ανήκουν στην κατηγορία του μεικτού ακέραιου γραμμικού προγραμματισμού (Mixed integer Linear Programming – MILP). Μοντέλα MILP συναντώνται συχνά στην βιομηχανία ιδιαίτερα σε προβλήματα που αφορούν οργάνωση παραγωγής και χρονοπρογραμματισμό.

Στα προβλήματα αυτής της κατηγορίας κάποιες από τις μεταβλητές του μοντέλου περιορίζονται έτσι ώστε να παίρνουν μόνο ακέραιες τιμές ενώ άλλες αφήνονται ελεύθερες να πάρουν τιμές που ανήκουν στο R^+. Για την επίλυση των προβλημάτων MILP υπάρχουν διαθέσιμοι διάφοροι επιλυτές όπως ο cplex, Gurobi, glpk οι οποίοι στοχεύουν στην εξασφάλιση της βέλτιστης λύσης.

Η ανάπτυξη των μοντέλων καθώς και των δεδομένων με τα οποία τα τροφοδοτούμε πραγματοποιήθηκε σε περιβάλλον python και ο επιλυτής cplex χρησιμοποιήθηκε κατά βάση για την παροχή των αποτελεσμάτων. Στην εργασία γίνεται επίσης μια συνοπτική σύγκριση μεταξύ των λογισμικών cplex και Gurobi ως προς την ταχύτητα εξαγωγής των αποτελεσμάτων. Στα μοντέλα διενεργούνται πειράματα κατά τα οποία μεταβάλλεται ο αριθμός κόμβων και οχημάτων με στόχο αρχικά την εξασφάλιση της σωστής λειτουργίας τους και δευτερευόντως την εύρεση του μέγιστου αριθμού κόμβων και οχημάτων για τα οποία μπορούν να δώσουν αποτελέσματα μέσα σε ένα εύλογο χρονικό διάστημα.

Τα αποτελέσματα των πειραμάτων επαληθεύουν την βιβλιογραφία αναφορικά με την δυσκολία στο να λυθούν NP-hard προβλήματα βελτιστοποίησης με ακριβείς (exact) μεθόδους επίλυσης, χωρίς δηλαδή τη χρήση ευρετικών ή άλλων αλγορίθμων, που είναι ικανοί να δώσουν ποιοτικά καλές προσεγγιστικές λύσεις απαιτώντας σχετικά μικρό χρόνο επίλυσης. Στην κατηγορία των NP-hard προβλημάτων βελτιστοποίησης ανήκουν τα προβλήματα δρομολόγησης οχημάτων, το πρόβλημα του περιοδεύοντος πωλητή, το μονοπάτι Χάμιλτον κ.α. Πρόκειται για προβλήματα στα οποία είναι πολύ πιθανό να μην υπάρχει πολυωνυμικός αλγόριθμος που να είναι ικανός να οδηγήσει σε λύση (Sanjeev & Boaz, 2009). Για την εύρεση λύσης σε NP – hard προβλήματα στον MILP ακολουθούνται τεχνικές χαλάρωσης του προβλήματος και μετατροπής του σε πρόβλημα γραμμικού προγραμματισμού, στρογγυλοποίησης της λύσης στην πλησιέστερη ακέραια τιμή, τεχνικές κλάδου και φράγματος, κ.α. Τροχοπέδη στην επίλυση NP – hard προβλημάτων με χρήση MILP αποτελεί η αδυναμία επίλυσης μεγάλων προβλημάτων (μεγάλος αριθμός κόμβων) καθώς οι χρόνοι επίλυσης ανεβαίνουν εκθετικά (Garey & Johnson, 1990).

Βασικό κίνητρο για την πραγματοποίηση της συγκεκριμένης διπλωματικής εργασίας αποτέλεσε η άμεση συσχέτιση μεταξύ της δρομολόγησης οχημάτων και του τομέα των Logistics. Οι επιχειρήσεις καθημερινά αναλώνονται στην εύρεση μεθόδων για την βελτιστοποίηση του κόστους που προκύπτει από την δρομολόγηση οχημάτων η οποία συνδυάζει το υψηλό κόστος μεταφορών, την ανάγκη για άψογη εξυπηρέτηση των πελάτων καθώς και την ύπαρξη πολλών αστάθμητων παραγόντων και περιορισμών. Είναι γνωστό ότι οι μεταφορές αποτελούν μεγάλο ποσοστό του κόστους που αφορά τα Logistics οπότε η βελτιστοποίηση αυτών μέσω ενός προγράμματος δρομολόγησης παρουσιάζει μεγάλο ενδιαφέρον κάτι που αντικατοπτρίζεται και από το μέγεθος της βιβλιογραφίας που συναντάει κανείς πάνω στην δρομολόγηση οχημάτων. Δευτερευόντως, κίνητρο αποτέλεσε η διάθεση για εξοικείωση με την ανάπτυξη μοντέλων και επίλυση αυτών στο περιβάλλον της python.

Αναφορικά με την δομή της διπλωματικής εργασίας, στο κεφάλαιο 2 γίνεται αναφορά στις βασικές έννοιες της Διοίκησης Αλυσίδας Εφοδιασμού, των Logistics και κατά επέκταση των βασικών κατηγοριών Εμπορευματικών Μεταφορών. Έπειτα στο κεφάλαιο 3 γίνεται παρουσίαση κάποιων εκ των βασικότερων μοντέλων δρομολόγησης πάνω στα οποία στηρίχτηκε η δημιουργία των μοντέλων της παρούσας εργασίας καθώς και η αντίστοιχη βιβλιογραφία. Στη συνέχεια ακολουθεί η ανάλυση των δύο μοντέλων δρομολόγησης CVRPt, VRPTWs που κατασκευάστηκαν στα πλαίσια της εργασίας και στηρίχθηκαν στην διεθνή βιβλιογραφία που αφορά προβλήματα δρομολόγησης με περιορισμένη χωρητικότητα και χρονικά παράθυρα (κεφάλαιο 4). Στο κεφάλαιο 5 γίνεται μια αναλυτική παρουσίαση των αποτελεσμάτων που προέκυψαν κατά την διενέργεια διαφόρων πειραμάτων πάνω στα προαναφερθέντα μοντέλα. Έπειτα στο κεφάλαιο 6 βρίσκονται διάφορα συμπεράσματα που προέκυψαν κατά την εκπόνηση της εργασίας γενικώς, καθώς επίσης διάφορες σκέψεις για μελλοντικά βήματα που μπορούν να εξελίξουν τους αλγορίθμους αλλά και να οδηγήσουν σε επίτευξη λύσεων για μεγαλύτερο αριθμό κόμβων και πελατών. Τέλος, στο παράρτημα βρίσκεται ο κώδικας των δύο μοντέλων αναλυτικά γραμμένος σε python με χρήση του εργαλείου μοντελοποίησης pyomo.

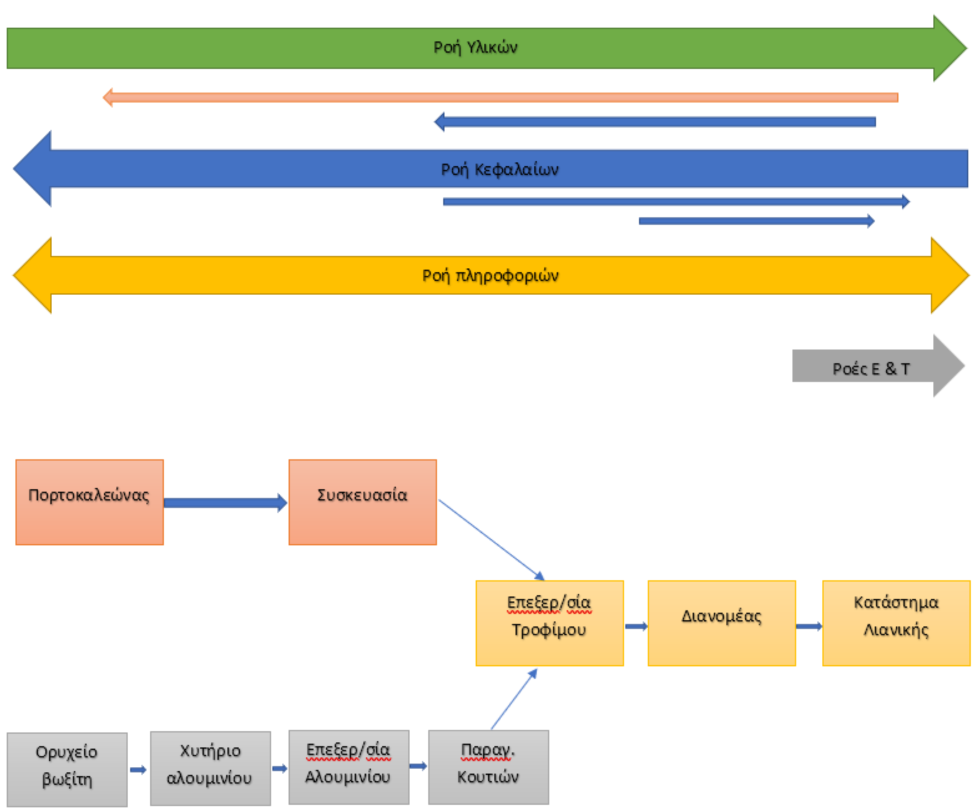
2. Αλυσίδα Εφοδιασμού

2.1 Διοίκηση της Αλυσίδας Εφοδιασμού

Στην σύγχρονη εποχή μας, σε μία άκρως ανταγωνιστική παγκόσμια αγορά σε συνδυασμό με την ολοένα αυξανόμενη χρήση του ηλεκτρονικού εμπορίου αλλά και την ανεπάρκεια πόρων, είναι επιτακτική η ανάγκη για σωστή διοίκηση της αλυσίδας εφοδιασμού των επιχειρήσεων αλλά και αυξημένη σύνδεση μεταξύ αυτής και της γενικότερης επιχειρηματικής στρατηγικής αυτών.

Ως Διοίκηση της Αλυσίδας Εφοδιασμού ορίζουμε την διοίκηση των διαδικασιών απόκτησης, μετατροπής και παράδοσης οι οποίες καθιστούν δυνατές και κατευθύνουν τις ροές προϊόντων και υπηρεσιών, καθώς επίσης και τις υποστηρικτικές διπλής κατεύθυνσης ροές πληροφοριών και κεφαλαίων κατά μήκος της αλυσίδας οδηγώντας τις από τις πηγές των βασικών εισροών προς τους τελικούς πελάτες, έχοντας στόχο την επίτευξη του καλύτερου δυνατού επιπέδου εξυπηρέτησης του πελάτη στο χαμηλότερο δυνατό κόστος.

Η αλυσίδα εφοδιασμού αποτελεί μια δυναμική έννοια η οποία περιλαμβάνει την σταθερή ροή πόρων μεταξύ των συμμετεχόντων (Εικόνα 1).



Εικόνα 1: Ροές στην αλυσίδα εφοδιασμού κατά την παραγωγή αναψυκτικών

Οι αλληλοσχετιζόμενες ροές της αλυσίδας εφοδιασμού είναι οι εξής :

- **Ροές Προϊόντων και υπηρεσιών**
- **Ροές Πληροφοριών**
- **Ροές κεφαλαίων**
- **Ροές Εμπειρίας και Τεχνολογίας**

Μιλώντας για ροές προϊόντων και υπηρεσιών αναφερόμαστε στα προϊόντα και τις υπηρεσίες των οποίων αυξάνεται η προστιθέμενη αξία όσο προχωρούν κατά μήκος της αλυσίδας. Συνήθως οι ροές είναι προς τα εμπρός αλλά παρατηρούνται και ροές προς τα πίσω σε περιπτώσεις επανεπεξεργασίας.

Οι ροές πληροφοριών δίνουν μια εικόνα της κατάστασης της αγοράς των προμηθευτών και γεγονότων που μπορεί να την επηρεάσουν αλλά κυρίως αναφέρονται στα επίπεδα ζήτησης από τους πελάτες .

Στις ροές κεφαλαίων συναντάμε πληρωμές για προϊόντα και υπηρεσίες και συνήθως είναι ροές που κατευθύνονται προς τα πίσω.

Τέλος, οι ροές εμπειρίας και τεχνολογίας αφορούν ζητήματα όπως ο σχεδιασμός των προϊόντων, το μάρκετινγκ, πληροφορική κ.α.

2.2 Διαδικασίες κατά Μήκος της Αλυσίδας Εφοδιασμού

Σε όλο το μήκος της εφοδιαστικής αλυσίδας συντελείται μια πληθώρα δραστηριοτήτων τις οποίες μπορούμε να αντιληφθούμε ακολουθώντας την κίνηση ενός προϊόντος μέσα στην επιχείρηση. Στην συνέχεια ακολουθεί μια σύντομη περιγραφή για καθεμία από τις λειτουργίες της εφοδιαστικής αλυσίδας.

2.2.1 Διαχείριση Κέντρων Διανομής και Αποθηκών

Η συγκεκριμένη λειτουργία αφορά τις εγκαταστάσεις που απαιτούνται για αποθήκευση πρώτων υλών, ετοίμων προϊόντων, ανταλλακτικών αλλά και σε ζητήματα που αφορούν στην λειτουργία των εγκαταστάσεων. Οι εγκαταστάσεις μπορεί να ανήκουν είτε στην επιχείρηση είτε σε τρίτους (outsourcing). Πρόκειται για μία λειτουργία με μεγάλο ενδιαφέρον λόγω της οικονομίας κλίμακας που μπορεί σε πολλές περιπτώσεις να επιτευχθεί.

2.2.2 Διαχείριση Υλικών

Αφορά στην διαχείριση των κινήσεων των πρώτων υλών αλλά και των ετοίμων προϊόντων μέσα στους χώρους της επιχείρησης. Στοχεύει στην ελαχιστοποίηση των εξόδων που συνοδεύουν τις μετακινήσεις των υλικών μέσω αλλαγής της διαρρύθμισης των χώρων στους οποίους κινούνται, μείωση αποστάσεων, ελαχιστοποίηση ζημιών – φθορών κ.α.

2.2.3 Διαχείριση Αποθεμάτων

Μία από τις πιο σημαντικές λειτουργίες της εφοδιαστικής αλυσίδας λόγω της άμεσης σύνδεσης της με την εύρυθμη λειτουργία της παραγωγής και της διακίνησης των προϊόντων. Στοχεύει στην μείωση του αποθέματος σε σημείο που δεν επηρεάζει αρνητικά την λειτουργία της της επιχείρησης λαμβάνοντας ταυτόχρονα υπ΄όψιν διάφορες παραμέτρους όπως η παλαίωση του αποθέματος, υψηλό κόστος αποθήκευσης, αυξομειώσεις στην ζήτηση των προϊόντων, κ.λ.π.

2.2.4 Τροφοδοσία

Πραγματοποιεί όλες τις αγορές προϊόντων που κρίνονται αναγκαίες για την αποδοτική λειτουργία όλων των τμημάτων της επιχείρησης.

2.2.5 Συσκευασία

Πρόκειται για μία διαδικασία που επηρεάζει άμεσα η έμμεσα τα διάφορα τμήματα που εργάζονται πάνω στη δημιουργία, την αποστολή ενός προϊόντος αλλά και την εμπορική επιτυχία αυτού. Πιο συγκεκριμένα η συσκευασία ενός προϊόντος επηρεάζει την κίνηση του μέσα στην επιχείρηση αλλά και εκτός αυτής σε διαδικασίες που αφορούν μεταφορά και αποθήκευση. Πρόσθετα ο σχεδιασμός της συσκευασίας επηρεάζει και επηρεάζεται από αποφάσεις που έχουν να κάνουν με θέματα marketing.

2.2.6 Μεταφορές

Χωρίζονται σε εσωτερικές και εξωτερικές μεταφορές. Πραγματεύονται ζητήματα που αφορούν στην επιλογή του είδους του μεταφορικού μέσου που θα χρησιμοποιηθεί δεδομένου του επιπέδου του κόστους, της ασφάλειας και του χρόνου που θέλουμε να επιτευχθεί. Στην περίπτωση των εσωτερικών μεταφορών όλα τα παραπάνω αφορούν πρώτες ύλες που πρέπει να μεταφερθούν από τους προμηθευτές προς την επιχείρηση ενώ στην περίπτωση των εξωτερικών μεταφορών αφορούν την μεταφορά του τελικού προϊόντος στον πελάτη.

2.2.7 Ανακύκλωση και διαχείριση απορριμμάτων

Πρόκειται για μια διαδικασία της αλυσίδας εφοδιασμού με ανάστροφη ροή από τις προηγούμενες η οποία ασχολείται με την διαχείριση ελλαττωματικών προϊόντων, λάθη παραδόσεων (παράδοση διαφορετικού προϊόντος από το απαιτούμενο, αριθμός παραδοθέντων κ.α.), περισυλλογή υλικών που μπορούν να επαναχρησιμοποιηθούν από την επιχείρηση (παλέτες, συσκευασίες) ή να ανακυκλωθούν και τέλος την ασφαλή απόρριψη των απορριμμάτων επικίνδυνων και μη.

2.2.8 Επικοινωνία

Η διαχείριση των πληροφοριών είναι μια πάρα πολύ σημαντική διαδικασία για όλες τις επιχειρήσεις η οποία περιπλέκεται καθημερινά όλο και περισσότερο. Ο βασικός όγκος των πληροφοριών που καλείται να διαχειριστεί το τμήμα ΔΕΑ μιας επιχείρησης περιλαμβάνει θέματα όπως διαθεσιμότητα υλικών, ζήτηση πελατών, επίπεδο αποθεμάτων, συντονισμός παραγωγής – χώρων αποθήκευσης, έτοιμα προϊόντα κ.α. Οι ανάγκες του σύγχρονου κόσμου για άμεση πληροφόρηση αλλά και η παγκοσμιοποίηση της αγοράς περιπλέκουν, όπως προαναφέρθηκε, όλο και περισσότερο την διαδικασία αυτή.

2.3 Logistics

Η ταύτιση της έννοιας των logistics με αυτή της ΔΕΑ αποτελεί συχνό φαινόμενο αλλά στην πραγματικότητα τα logistics αποτελούν κομμάτι της ΔΕΑ. Ο παρακάτω ορισμός των logistics ξεκαθαρίζει τα πράγματα :

Τα logistics σύμφωνα με το βιβλίο Διοίκηση Εφοδιασμού του κ.Λάιου (2010) αποτελούν το τμήμα της διαδικασίας της αλυσίδας εφοδιασμού το οποίο σχεδιάζει, υλοποιεί και ελέγχει την αποτελεσματική και αποδοτική ροή και αποθήκευση υλικών, υπηρεσιών και σχετικών πληροφοριών από το σημείο προέλευσης προς το σημείο κατανάλωσης, με στόχο να ικανοποιηθούν οι απαιτήσεις των πελατών.

Στόχος των logistics είναι η διασφάλιση :

- Σωστού τύπου υλικών
- Σωστής ποσότητας υλικών
- Άριστης κατάστασης κατά την παράδοση
- Παράδοσης σε σωστή τοποθεσία και πελάτη
- Σωστών χρόνων παράδοσης
- Σωστού κόστους
- Παροχής υπηρεσιών εξυπηρέτησης (π.χ. τεχνική υποστήριξη, συντήρηση) που απαιτεί ο αγοραστής
- Παροχής σωστής και συνεχούς πληροφόρησης (π.χ. ενημέρωση για το στάδιο υλοποίησης της παραγγελίας, για αποτελέσματα επιθεωρήσεων και δοκιμών, για θέματα υγιεινής και ασφάλειας) που χρειάζεται ο πελάτης κατά την διάρκεια και στο τέλος της διαδικασίας logistics

Γενικώς μπορούμε να πούμε ότι η ΔΕΑ αναφέρεται στην διοίκηση των διαδικασιών για τις οποίες ευθύνονται όλα τα μέλη της αλυσίδας εφοδιασμού σε αντίθεση με τα logistics που ασχολούνται κυρίως με την διοίκηση των πόρων (υλικά, κεφάλαια, πληροφορίες κ.α.). Παρόλα αυτά η στρατηγική logistics μιας επιχείρησης πρέπει να βρίσκεται ευθυγραμμισμένη με την γενικότερη στρατηγική της αλυσίδας εφοδιασμού.

Πρωταρχικός στόχος του τμήματος logistics μια επιχείρησης είναι η ελαχιστοποίηση του κόστους αυτών. Το κόστος των logistics είναι ιδιαίτερα υψηλό (μπορεί να αγγίξει το 30% του κόστους ενός προϊόντος) και περιλαμβάνει το κόστος μεταφοράς, αποθήκευσης, συσκευασίας και διανομής. Επίσης το τμήμα των logistics είναι επιφορτισμένο με την επίτευξη γρήγορων παραγγελιών – παραδόσεων, υψηλής διαθεσιμότητας προϊόντων και ικανοποίηση ιδιαίτερων αναγκών.

2.4 Μεταφορές

2.4.1 Οδικές Μεταφορές

Οι οδικές μεταφορές αποτελούν το πιο διαδεδομένο μέσο εμπορευματικών μεταφορών στην εφοδιαστική αλυσίδα. Μέσω των οδικών μεταφορών διακινείται περίπου το 75% του συνόλου των εμπορευματικών μεταφορών δια ξηράς και κοντά στο 90% της αξίας του συνόλου των εμπορευματικών μεταφορών. Αποτελούν στρατηγικής σημασίας τομέα για την παγκόσμια οικονομία αντιπροσωπεύοντας το 3,4% του ευρωπαϊκού ΑΕΠ παρέχοντας απασχόληση περισσότερους από 8 εκατομμύρια ανθρώπους ανά την Ευρώπη.

2.4.2 Αστικές Εμπορευματικές Μεταφορές

Σκοπός των Αστικών Εμπορευματικών Μεταφορών είναι η αναβάθμιση της κινητικότητας εντός των πόλεων αλλά και της περιβαλλοντικής ασφάλειας. Στηρίζονται σε συστήματα διανομής των εμπορευμάτων στα κέντρα των πόλεων, αλλά και συγκέντρωσης επιστρεφόμενων προϊόντων και αποβλήτων.

2.4.3 Αεροπορικές Εμπορευματικές Μεταφορές

Οι αεροπορικές εμπορευματικές μεταφορές διακρίνονται σε δύο μεγάλες κατηγορίες:

- Τα εμπορεύματα και πάσης φύσεως φορτία
- Το ταχυδρομείο.

Ο διαχωρισμός αυτός έχει να κάνει περισσότερο με εμπορικά ζητήματα αλλά και με θέματα διαχείρισης των φορτίων. Η μεταφορά του ταχυδρομείου μπορεί να γίνει εύκολα σε μια επιβατική πτήση, ενώ άλλα φορτία τα οποία μπορεί να είναι ογκώδη συχνά πρέπει να μεταφερθούν με ειδικά αεροσκάφη, τα οποία απαιτούν ιδιαίτερες υποδομές και επίγειο εξοπλισμό.

2.4.4 Rail – Road Συνδυασμένη Μεταφορά

Πρόκειται για μία μορφή συνδυαστικής μεταφοράς, όπου το μεγαλύτερο μέρος της γίνεται με σιδηρόδρομο και κάθε αρχικό ή/και τελικό σκέλος της διαδρομής που εκτελείται σε οδικό άξονα, είναι όσο πιο σύντομο γίνεται.

Οι σιδηροδρομικές μεταφορές χαρακτηρίζονται συχνά ως «η βιομηχανία του μέλλοντος» και γίνονται συνεχείς προσπάθειες προκειμένου το τρένο να γίνει ένα πιο ελκυστικό, ασφαλές, ανταγωνιστικό, αξιόπιστο και φιλικό προς το περιβάλλον μέσο μεταφοράς, συμβάλλοντας στην αναβάθμιση της ενιαίας αγοράς. Ο σιδηρόδρομος είναι, πιθανόν, η μόνη τεχνολογία που κατά τη διάρκεια της εξελικτικής της πορείας γνώρισε αρχικά περίοδο μεγάλης ακμής, έπειτα πέρασε μια περίοδο μεγάλης αμφισβήτησης και τα τελευταία χρόνια κατόρθωσε όχι μόνο να ανακάμψει, αλλά να αποτελεί, για πολλές χώρες, τεχνολογία αιχμής .

2.4.5 Εμπορευματικά Κέντρα

Ως εμπορευματικό κέντρο ορίζεται μια συγκεκριμένη περιοχή όπου εκτελούνται από διάφορους φορείς, όλες οι δραστηριότητες σχετικά με τη μεταφορά, τα logistics και τη διανομή εμπορευμάτων για εθνική αλλά και διεθνή διαμετακόμιση. Υποστηρίζεται η συνδυασμένη μεταφορά με τη συνύπαρξη διαφόρων μέσων μεταφοράς. Τα εμπορευματικά κέντρα αποτελούν σημεία διασύνδεσης σε ένα δίκτυο logistics, παίζοντας το ρόλο της διεπαφής μεταξύ τοπικών και μεγαλύτερης απόστασης μεταφορών, γεγονός που επιτυγχάνει αυτόματα εξοικονομήσεις σε όλη την εφοδιαστική αλυσίδα.

3. Το πρόβλημα της Δρομολόγησης Οχημάτων

3.1 Δρομολόγηση Οχημάτων

Το πρόβλημα της δρομολόγησης οχημάτων εκφράστηκε για πρώτη φορά το 1959 από τους Dantzig και Ramser (1959) ως γενικευμένη μορφή του προβλήματος του πλανόδιου πωλητή (TSP) προσπαθώντας να δώσουν λύση στο πρόβλημα της διανομής πετρελαίου μέσω οχημάτων από έναν κεντρικό κόμβο σε διάφορα πρατήρια καυσίμων. Έπειτα οι Clarke και Wright (1964) δημιούργησαν την κλασική γραμμική μορφή του προβλήματος που συναντάτε συχνά στον κλάδο των logistics και αφορά στο πώς μπορούμε να ικανοποιήσουμε την ζήτηση πελατών που βρίσκονται διασκορπισμένοι γύρω από μία αποθήκη χρησιμοποιώντας έναν στόλο οχημάτων διαφόρων αποθηκευτικών δυνατοτήτων. Η υλοποίηση των Clarke και Wright έγινε γνώστη ως Vehicle Routing Problem ή αλλιώς VRP και αποτελεί ένα πολύ γνωστό πρόβλημα συνδυαστικής βελτιστοποίησης το οποίο έχει μελετηθεί εις βάθος από την επιστημονική κοινότητα.

Σήμερα τα μοντέλα δρομολόγησης οχημάτων διαφοροποιούνται σημαντικά σε σχέση με την πρώτη υλοποίηση τους το 1964 καθώς καλούνται να προσομοιάσουν τις πραγματικές - σύγχρονες ανάγκες της αγοράς. Έτσι λοιπόν συναντάμε μοντέλα τα οποία περιλαμβάνουν παραλαβές / παραδόσεις εντός συγκεκριμένων χρονικών παραθύρων, προβλέψεις για καθυστερήσεις λόγω κυκλοφοριακής συμφόρησης αλλά και δυναμικών αλλαγών στη ζήτηση (Πίνακας 1). Αυτό έχει ως αποτέλεσμα την αύξηση της πολυπλοκότητας των προβλημάτων τα οποία ήδη ανήκουν στην οικογένεια των μη πολυωνυμικών δύσκολων προβλημάτων (NP-hard) (Lenstra & Rinnooy Kan, 1981) . Η λύση των προβλημάτων με χρήση ακριβών αλγορίθμων είναι εφικτή μόνο για πολύ μικρά παραδείγματα (μικρός αριθμός οχημάτων και κόμβων) τα οποία σε πολλές περιπτώσεις δεν αντικατοπτρίζουν την πραγματικότητα, αντιθέτως η χρήση ευρετικών και μετα-ευρετικών αλγορίθμων παρουσιάζεται ως βέλτιστη επιλογή.

Ονομασία	Περιγραφή
Capacitated Vehicle Routing Problem (CVRP)	Σχεδιασμός βέλτιστων διαδρομών οχημάτων συγκεκριμένης χωρητικότητας
Vehicle Routing Problem with Time Windows (VRPTW)	Σχεδιασμός βέλτιστων διαδρομών οχημάτων, κατά τον οποίο ο κάθε πελάτης πρέπει να εξυπηρετηθεί εντός συγκεκριμένου χρονοπαράθυρου
Vehicle Routing Problem with Pickups and Deliveries (VRPPD)	Σχεδιασμός βέλτιστων διαδρομών οχημάτων, όπου σε κάθε θέση εξυπηρέτησης πραγματοποιείται ταυτόχρονη παράδοση ή/και παραλαβή προϊόντων στον κάθε πελάτη
Vehicle Routing Problem with Backhauls (VRPWB)	Σχεδιασμός βέλτιστων διαδρομών οχημάτων με παράδοση και παραλαβή προϊόντων από τους πελάτες

Πίνακας 1: Επεκτάσεις Προβλήματος Βέλτιστης Δρομολόγησης Οχημάτων

Η δρομολόγηση οχημάτων βρίσκει πολλές εφαρμογές στην καθημερινότητα μας όπως στην διανομή αγαθών, στην διανομή αλληλογραφίας, στην αποκομιδή των απορριμμάτων, καθαρισμό δρόμων κ.α. με κυριότερες τις δύο πρώτες. Ιδιαίτερο ενδιαφέρον πέραν αυτού της επιστημονικής κοινότητας υπάρχει και από την αγορά καθώς έχει αποδειχτεί (Maffioli, 2003) ότι η χρήση μηχανογραφημένων μεθόδων σε διαδικασίες διανομής μπορεί να οδηγήσει σε μείωση έως 20% του κόστους που τις αφορά.

Από το 1959 μέχρι σήμερα έχουν δημοσιευτεί χιλιάδες έρευνες που πραγματεύονται το πρόβλημα της δρομολόγησης των οχημάτων παρουσιάζοντας μία πληθώρα διαφορετικών μοντέλων. Στην συνέχεια παρουσιάζεται το κλασικό μοντέλο δρομολόγησης οχημάτων καθώς και μερικές από τις πιο γνωστές κατηγορίες μοντέλων πάνω στην δρομολόγηση οχημάτων.

3.2 Κλασικό πρόβλημα δρομολόγησης οχημάτων (VRP)

Στο κλασικό πρόβλημα δρομολόγησης οι πελάτες , οι χρόνοι εξυπηρέτησης καθενός από αυτούς καθώς και η απόσταση αναμεταξύ τους αποτελούν γνωστά μεγέθη (Madsen, et al., 1995). Σκοπός του προβλήματος είναι η δημιουργία οικονομικών διαδρομών για κάθε όχημα με ταυτόχρονη ελαχιστοποίηση του αριθμού των οχημάτων που απαιτούνται. Το κλασικό πρόβλημα δρομολόγησης μπορεί να οριστεί ως (Laporte, 1992) :

Ορίζουμε ως $G = (V,A)$ ένα γράφημα όπου $V = \{1 \dots n\}$ αποτελεί το σύνολο των κόμβων που αντιπροσωπεύουν πόλεις-σταθμούς με το 1 να αντιπροσωπεύει την αφετηρία και A το σύνολο των τόξων (Εικόνα 2). Για κάθε σύνδεση (i,j) όπου $i \neq j$ ορίζουμε έναν μην αρνητικό πίνακα απόστασης $D = (d_{ij})$. Ο πίνακας D μπορεί να αντιπροσωπεύει είτε το κόστος μετακίνησης από το i στο j, είτε το χρόνο. Επίσης ορίζουμε ως m το σύνολο των διαθέσιμων οχημάτων στην αφετηρία όπου $m_L < m < m_U$. Θεωρούμε το m ως σταθερό όταν $m_L = m_U$ και μεταβλητό όταν $m_L = 1$ και $m_U = n - 1$. Συνήθως στις περιπτώσεις όπου το m είναι μεταβλητό συσχετίζουμε την χρήση ενός οχήματος με κάποιο κόστος f. Στόχος του μοντέλου είναι ο σχεδιασμός ενός συνόλου «οικονομικών» διαδρομών έτσι ώστε :

- Κάθε πόλη – κόμβος στο V\ {1} δέχεται επίσκεψη μία φορά και από ένα όχημα
- Όλα τα οχήματα ξεκινούν και καταλήγουν στην αφετηρία
- Να ικανοποιούνται κάποιοι επιπλέον περιορισμοί

Ακολουθεί το μαθηματικό μοντέλο για το VRP που συναντάμε στο έργο του Laporte (1992) :

Έστω x_{ij} ακέραια μεταβλητή $\in \{0, 1\}$, $\forall \{i, j\} : \in E\backslash\{\{0, j\} : j \in V\}$ με τιμές $\{0, 1, 2\}$, $\forall \{0, j\} \in E$. $j \in V$. Το $x_{0j} = 2$ στην περίπτωση που διαδρομή στην οποία συμπεριλαμβάνεται ο πελάτης j επιλέγεται ως λύση.

Το μαθηματικό μοντέλο διαμορφώνεται ως εξής :

$$min \sum_{i \neq j} d_{ij} * x_{ij} \tag{1}$$

Περιορισμοί :

$$\sum_j x_{ij} = 1, \forall\, i \ \in V \tag{2}$$

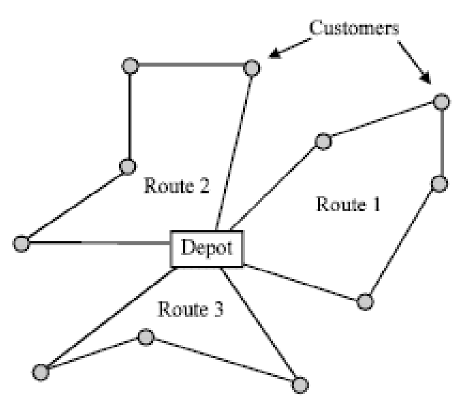
$$\sum_i x_{ij} = 1, \forall\, j \ \in V \tag{3}$$

$$\sum_i x_{ij} \ \geq |S| - \nu(S), \{S : S \ \subseteq V\{1\}, |S| \geq 2\} \tag{4}$$

$$X_{ij} \in \{0, 1\}, \forall \{I, j\} \in E\, ;\, i \neq j \tag{5}$$

Πρόκειται για ένα τροποποιημένο μοντέλο δρομολόγησης οχημάτων καθώς όπως παρατηρούμε στο (1), αποτρέπεται η χρήση των τιμών όπου το i = j. Στα (2) και (3) εξασφαλίζεται η επίσκεψη κάθε κόμβου από ένα όχημα. Τέλος, στο (4) έχουμε την απαγόρευση δημιουργίας υποδιαδρομών.

Όπως προαναφέρθηκε η δρομολόγηση οχημάτων ανήκει στην κατηγορία των NP-hard προβλημάτων και έτσι πολλοί ερευνητές όπως οι Chiang και Russell (1996), Braysy et al. (2004) οδηγήθηκαν στην χρήση ευρετικών αλγορίθμων. Πέρα από τους ευρετικούς έχει προταθεί και ένα εύρος άλλων αλγορίθμων όπως οι αλγόριθμοι εύρεσης άριστης λύσης από τους Laporte και Osman (1995), Tabu Search που συναντάτε στο έργο των Taillard (1993) και Rochat και Taillard (1995), προσομοιωμένης ανόπτησης και γενετικών αλγορίθμων στο έργο των Baker και Ayechew (2003), κ.α.



Εικόνα 2: Σχηματική Απεικόνιση VRP

3.3 Προβλήματα δρομολόγησης περιορισμένης χωρητικότητας (CVRP)

Το CVRP αποτελεί μια επέκταση του προβλήματος δρομολόγησης οχημάτων στο οποίο τα οχήματα έχουν όλα την ίδια περιορισμένη χωρητικότητα. Σκοπός των CVRP μοντέλων είναι η ελαχιστοποίηση του χρόνου μεταφοράς ή/και της διαδρομής των οχημάτων χωρίς να ξεπερνιέται το όριο χωρητικότητας τους. Η προφανής πρακτική χρησιμότητα των CVRP μοντέλων ήταν αυτή που έδωσε ώθηση στην εξέλιξη των ευρετικών αλγορίθμων (Baldacci, et al., 2007). Παρόλα αυτά επίλυση του CVRP επιτυγχάνεται και μέσω αλγορίθμων εύρεσης άριστης λύσης όπως βλέπουμε στο έργο του Baldacci et al. (2004) αλλά και branch and cut αλγορίθμων Lysgaard et al. (2004). Στην συνέχεια παρουσιάζεται ένα μοντέλο CVRP (Fukasawa, et al., 2004)

Έστω H = (N, A), d, q και Q ορίζουν μια περίπτωση CVRP όπου ο κόμβος 0 αντιπροσωπεύει την αφετηρία και οι υπόλοιποι τους πελάτες.

Το μαθηματικό μοντέλο διαμορφώνεται ώς εξής :

$$min \sum_{e=(u,v)\in A} d(e)x_e \tag{6}$$

Περιορισμοί:

$$\sum_{e\in\delta(\{u\})} x_e = 2, \forall\, u \in N\{0\} \tag{7}$$

$$\sum_{e\in\delta(\{0\})} x_e \geq 2 * k^* \tag{8}$$

$$\sum_{e\in\delta(S)} x_e \geq 2 * k(S), \forall\, S \in N\{0\} \tag{9}$$

$$X_e \leq 1, \forall\, e \in A \setminus \delta(\{0\}) \tag{10}$$

$$\sum_{l=1}^{p} q_l^e \lambda_l - x_e = 0, \forall\, e \,\forall\, A \tag{11}$$

$$x_e \in \{0, 1, 2\}, \forall\, e \in A \tag{12}$$

$$l_l \geq 0 \;\forall\, l \in \{1, \dots, p\} \tag{13}$$

Όπου :

X_e αφορά το πόσες φορές η κορυφή e χρησιμοποιήθηκε από κάποιο όχημα. Η μεταβλητή μπορεί να πάρει την τιμή 2 αν το e βρίσκεται κοντά στην αφετηρία αντιπροσωπεύοντας μια διαδρομή για έναν πελάτη. Οι μεταβλητές l_l αφορούν έγκυρες διαδρομές και σχετίζονται με όλες τις q-διαδρομές που ικανοποιούν τον περιορισμό χωρητικότητας (11). Μία q-διαδρομή ξεκινά από την αφετηρία επισκέπτεται ένα σύνολο πελατών με την ζήτηση να είναι μικρότερη η ίση με το Q και επιστρέφει εκεί που ξεκίνησε. Ένας πελάτης μπορεί να εμφανιστεί παραπάνω από μια φορά μέσα σε μία q-διαδρομή. Κάθε μεταβλητή l_l σχετίζεται με μία από τις πιθανές q-διαδρομές.

Ο περιορισμός (7) επιβάλει ότι μόνο ένα όχημα θα επισκεφθεί κάθε πελάτη και ο (8) ότι τουλάχιστον K^* φεύγουν από και επιστρέφουν στην αφετηρία. Οι περιορισμοί στο (9) χρησιμοποιούν το $k(S) = [\sum_{u\in S} q(u)/Q]$ ως κάτω όριο αναφορικά με τον ελάχιστο αριθμό οχημάτων που απαιτούνται για να εξυπηρετήσουν τους πελάτες στο σύνολο $S \subseteq N$. Οι περιορισμοί στο (10) υποχρεώνουν το χ να είναι ένας γραμμικός συνδυασμός από q διαδρομές.

3.4 Πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα (VRPTW)

Το VRPTW αποτελεί άλλη μια προέκταση του κλασικού προβλήματος δρομολόγησης στο οποίο επιδιώκεται ο σχεδιασμός βέλτιστων διαδρομών οχημάτων όπου ο πελάτης πρέπει να εξυπηρετηθεί εντός συγκεκριμένου χρονοπαράθυρου. Στα VRPTW προβλήματα τα οχήματα δεν επιτρέπεται να επισκεφθούν έναν πελάτη εκτός της τελευταίας καθορισμένης στιγμής ενώ αν προσεγγίσουν νωρίτερα θα έχουμε ως αποτέλεσμα αδράνεια χρόνου. Επίσης πολλές φορές παρατηρείται η σύνδεση ενός χρόνου συντήρησης που συνδέεται με την εξυπηρέτηση του πελάτη (Ellabib, et al., 2002). Ακολουθεί ένα μαθηματικό μοντέλο δρομολόγησης με χρονικά περιθώρια όπου x_{kij} μεταβλητή που αφορά την κίνηση ή όχι του k οχήματος από το i στο j, s_{ki} μεταβλητή που αφορά την χρονική στιγμή έναρξης της εξυπηρέτησης του κόμβου i από το όχημα k, t_{ij} χρόνος μετάβασης από το i στο j και a_i, b_i το χρονικό παράθυρο εξυπηρέτησης του κόμβου i.

$$\min \sum_{i \in N} \sum_{j \in N} C_{ij} * x_{kij} \tag{13}$$

$$\sum_{k \in V} \sum_{j \in N} x_{kij} = 1 \quad \forall\, i \in C \tag{14}$$

$$\sum_{i \in C} d_i * \sum_{j \in N} x_{kij} \leq q \quad \forall\, k \in V \tag{15}$$

$$\sum_{j \in N} x_{k0j} = 1, \quad \forall k \in V \tag{16}$$

$$\sum_{i \in N} x_{kih} - \sum_{j \in N} x_{khj} = 0, \quad \forall h \in C, k \in V \tag{17}$$

$$\sum_{i \in N} x_{k,i,n+1} = 1, \quad \forall k \in V \tag{18}$$

$$s_{ki} + t_{ij} + s_{kj} \leq M(1 - X_{kij}) \quad \forall\, i, j \in N, k \in V \tag{19}$$

$$\alpha_i \leq s_{ik} \leq b_i, \forall\, i \in N, \quad \forall\, k \in V \tag{20}$$

$$x_{kij} \in \{0,1\}, \forall\, i,j \in N, \quad \forall\, k \in V \tag{21}$$

Στον περιορισμό (14) ορίζεται ότι κάθε πελάτης επισκέπτεται μία φορά και στον (15) ότι δεν πρέπει να ξεπερνιέται η χωρητικότητα του κάθε οχήματος. Στους περιορισμούς (16), (17), εξασφαλίζεται ότι κάθε όχημα ξεκινάει από την αφετηρία και επιστρέφει στον τερματισμό, ενώ στον (18) το κάθε όχημα υποχρεώνεται να αναχωρήσει από τον κόμβο που έχει επισκεφθεί. Στην συνέχεια απαγορεύεται μέσω του περιορισμού (19) όχημα k να καταφθάσει νωρίτερα από $s_{ki} + t_{ij}$ ταξιδεύοντας από το i στο j. Τέλος στο (20) επιδιώκουμε την τήρηση των χρονικών παραθύρων και στο (21) γίνεται η δήλωση των μεταβλητών.

Όπως είναι προφανές το VRPTW αποτελεί έναν συνδυασμό δρομολόγησης με ταυτόχρονη τήρηση χρονοδιαγράμματος που βρίσκει πολλές εφαρμογές στην καθημερινότητα. Έχει βρεθεί στο επίκεντρο πολλών ερευνών και έχουν προταθεί τόσο ευρετικοί αλγόριθμοι για την επίλυση του (Campbell & Savelsbergh, 2004), (Solomon, 1987), (Bramel & Simchi-Levi, 1996), όσο αλγόριθμοι εύρεσης άριστης λύσης Cordeau et al. (2001).

4. Μοντελοποίηση προβλημάτων

4.1 Μοντέλο δρομολόγησης περιορισμένης χωρητικότητας με χρονικούς περιορισμούς (CVRPt)

Το πρώτο μοντέλο που κατασκευάζουμε βασίζεται στα μοντέλα που συναντάμε στο έργο των Kuklarni και Bhave (1985) με κάποιες τροποποιήσεις. Πιο συγκεκριμένα χρησιμοποιούμε έναν τρισδιάστατο πίνακα κόστους προκειμένου το κόστος να μην είναι κοινό για όλα τα οχήματα που κάνουν την ίδια διαδρομή αλλά να εξατομικεύεται ανάλογα με το όχημα. Επίσης προσθέτουμε έναν περιορισμό που ελέγχει ότι τα οχήματα δεν λειτουργούν για χρόνο που υπερβαίνει τις οκτώ ώρες.

Αρχικά ορίζουμε ως $N = \{n_1, n_2, \ldots, n_n\}$ το σύνολο των κορυφών και $V = \{v_1.v_2, \ldots v_v\}$ το σύνολο των οχημάτων, $A=\{ (n_i,n_j) \mid n_i,n_j \in N;\ i \neq j \}$ το σύνολο των τόξων και $C = (C_{kij})$ έναν πίνακα κόστους μετάβασης του οχήματος k από το i στο j ορισμένο στο *A*. Στόχος του μοντέλου είναι η δρομολόγηση των οχημάτων από την αφετηρία (n_1), με ταυτόχρονη ελαχιστοποίηση του κόστους, σε όλους τους υπόλοιπους κόμβους *N*. Στο μοντέλο συμβολίζουμε με *N'* το σύνολο των κόμβων «πελατών» ($N' = N - \{n_1\}$). Επίσης ορίζουμε ώς Q_k την χωρητικότητα των οχημάτων, w_i το μέγεθος της παραγγελίας των πελατών, T_{ij} τον χρόνο που χρειάζεται το όχημα για να μεταβεί από το i στο j, T_{ej} τον χρόνο εξυπηρέτησης του κάθε πελάτη και ως $H1$ σταθερά που αντιπροσωπεύει το εργασιακό οκτάωρο σε δευτερόλεπτα. Η μεταβλητή u_i είναι μια ακέραια βοηθητική μεταβλητή. Στη συνέχεια παρουσιάζεται το μοντέλο δρομολόγησης.

$$x_{kij} = \begin{cases} 1, & \text{αν το όχημα } k \text{ μεταβαίνει απο το } i \text{ στο } j \\ 0, & \text{οπουδήποτε αλλού} \end{cases}$$

$$\min \sum_i \sum_j \sum_k C_{kij} * x_{kij} \tag{22}$$

$$\sum_{j\in N} \sum_k x_{kij} = 1 \quad \forall\, i \in N' \tag{23}$$

$$\sum_{i\in N'} x_{ki1} \leq 1 \quad \forall\, k \tag{24}$$

$$\sum_{j\in N'} \sum_k x_{k1j} \geq 1 \tag{25}$$

$$\sum_{i\in N} x_{kih} - \sum_{j\in N} x_{khj} = 0 \quad \forall\, h \in N, k \tag{26}$$

$$u_i - u_j + n * \sum_k x_{kij} \leq n - 1 \quad \forall\, i,j \in N' \tag{27}$$

$$\sum_{i\in N'} \sum_{j\in N} w_i * x_{kij} \leq Q_k \quad \forall\, k \tag{28}$$

$$\sum_{i\in N} \sum_{j\in N} x_{kij} * (T_{ij} + T_{ej}) \leq H1 \tag{29}$$

$$X_{kij} \in \{0,1\} \quad \forall\, k, i, j \tag{30}$$

$$u_i \in Z^+ \quad \forall\, i \in N' \tag{31}$$

Αρχικά (22) συναντάμε την αντικειμενική συνάρτηση του μοντέλου, η οποία επιδιώκει την ελαχιστοποίηση του κόστους όπως προαναφέρθηκε. Στο (23) βρίσκεται ο πρώτος περιορισμός του μοντέλου που αποτρέπει την επίσκεψη κάποιου κόμβου – πελάτη από παραπάνω από ένα οχήματα. Ο επόμενος περιορισμός (24) δεν επιτρέπει στα οχήματα να φύγουν από την αφετηρία παραπάνω από μία φορά και στο (25) βρίσκεται ο περιορισμός που εξαναγκάζει τουλάχιστον ένα όχημα να ξεκινήσει προκειμένου να εξυπηρετηθούν οι κόμβοι πελάτες. Ο περιορισμός (26) εξαναγκάζει τα οχήματα να φύγουν από τον κόμβο τον οποίο επισκέφτηκαν (flow conservation constraint) και ο (27) αποτρέπει την δημιουργία ανεπιθύμητων δρομολογίων. Στην συνέχεια ο περιορισμός (28) ελέγχει ότι δεν υπερβαίνεται η αποθηκευτική δυνατότητα των οχημάτων και ο (29) ότι δεν ξεπερνάτε το οκτάωρο. Τέλος στο (30) και (31) βρίσκονται τα ορίσματα των μεταβλητών που χρησιμοποιήθηκαν.

4.2 Μοντέλο δρομολόγησης με χρονικά παράθυρα και χρονικούς περιορισμούς (VRPTWs)

Το μοντέλο που ακολουθεί αποτελεί εξέλιξη του προηγούμενου όπου πλέον δεν αρκούμαστε το όχημα να κάνει το πιο «οικονομικό δρομολόγιο», αλλά να το πράξει τηρώντας κάποια χρονικά περιθώρια που ορίζουμε. Όπως θα δούμε παρακάτω ο κάθε κόμβος διαθέτει ένα χρονικό περιθώριο $[a_i,b_i]$ εντός του οποίου δέχεται να εξυπηρετηθεί. Ξεκινάμε ορίζοντας ως $N = \{n_1,n_2,\ldots,n_{n+2}\}$ το σύνολο των κορυφών, $V = \{v_1.v_2,\ldots v_v\}$ το σύνολο των οχημάτων, $A=\{ (n_i,n_j)|n_i,n_j \in N;\ i\neq j \}$ το σύνολο των τόξων και $C = (C_{kij})$ έναν πίνακα χρόνου μετάβασης του κάθε οχήματος k από το i στο j ορισμένο στο *A*. Συμβολίζουμε με N' το σύνολο των κόμβων-πελατών ($N'= N – (\{n_1\} + \{n_{n+2}\})$ και με V το σύνολο των οχημάτων. Ως t_{ik} ορίζεται ο χρόνος εξυπηρέτησης του κάθε οχήματος στον κάθε κόμβο, ενώ a_i και b_i αποτελούν το άνω και κάτω όριο του παράθυρου εξυπηρέτησης του κάθε κόμβου. Τέλος, ορίζουμε ως W_i την ζήτηση των κόμβων-πελατών και Q_k την χωρητικότητα των οχημάτων. Ακολουθεί το μοντέλο δρομολόγησης :

$$\min \quad \sum_k \sum_i \sum_j C_{kij} * x_{kij} \tag{32}$$

$$\sum_{k\in V} \sum_{j\in N} x_{kij} = 1 \quad \forall\, i\ \in N' \tag{33}$$

$$\sum_{j\in N} x_{k1j} = 1, \ \ \forall k\ \in V \tag{34}$$

$$\sum_{i\in N} x_{ki(n+2)} = 1, \ \ \forall k\ \in V \tag{35}$$

$$\sum_{i\in N} x_{kih} - \sum_{j\in N} x_{khj} = 0, \ \ \forall h \in N', k \in V \tag{36}$$

$$s_{ki} \geq a_i, \ \ \forall\ i \in N, k \in V \tag{37}$$

$$s_{ki} \leq b_i, \ \ \forall\ i \in N, k \in V \tag{38}$$

$$s_{ki} + t_{kj} + C_{kij} - s_{kj} \leq M(1 – X_{kij}) \ \ \forall\ i, j\in N, k\in V \tag{39}$$

$$\sum_{i\in N'} Wi * \sum_{j\in N} x_{kij} \leq Q_k \tag{40}$$

$$\sum_{k\in V} \sum_{j\in N} x_{k1j} \ \leq |V| \tag{41}$$

$$S_{ki} \leq 28800\ \forall\ k,i \tag{42}$$

$X_{kij} \in \{0,1\} \quad \forall\, k, i, j \qquad (43)$

$S_{ki} \in N^{+} \quad \forall\, k, i \qquad (44)$

Ξεκινάμε την ανάλυση της δομής του μοντέλου από το (32) όπου βρίσκουμε την αντικειμενική συνάρτηση με στόχο την ελαχιστοποίηση του συνολικού χρόνου. Στην συνέχεια ακολουθεί ο περιορισμός (33) σύμφωνα με τον οποίο κάθε κόμβος-πελάτης πρέπει να εξυπηρετηθεί από ένα όχημα. Οι περιορισμοί (34),(35) απαιτούν τα οχήματα να ξεκινήσουν από την αφετηρία(n=1) και να καταλήξουν στον τερματισμό (n+2). Ακολουθεί στο (36) ο περιορισμός που αναγκάζει τα οχήματα να αναχωρούν από τον κόμβο που επισκέφθηκαν. Με τους περιορισμούς (37), (38) ελέγχουμε ότι η εξυπηρέτηση των κόμβων από τα οχήματα γίνεται εντός του χρονικού περιθωρίου που δέχονται να εξυπηρετηθούν ($[a_i, b_i]$). Έπειτα στον περιορισμό (39) βεβαιωνόμαστε ότι κανένα όχημα που ταξιδεύει από τον κόμβο i στον j δεν θα φτάσει νωρίτερα από την χρονική στιγμή $s_{ki} + t_{ικ}+ C_{kij}$. Ο περιορισμός (40) αφορά στον έλεγχο της σχέσης χωρητικότητας των οχημάτων και παραγγελιών των κόμβων-πελατών ενώ στο (41) ο περιορισμός που επιχειρεί την ελαχιστοποίηση των οχημάτων που χρησιμοποιούνται. Στην συνέχεια ακολουθεί ο περιορισμός που επιβάλει ότι ο χρόνος που μπορούν να λειτουργήσουν τα οχήματα δεν θα ξεπερνάει το οκτάωρο(42). Κλείνοντας έχουμε την δήλωση των μεταβλητών του μοντέλου (43),(44).

5. Υπολογιστική εμπειρία

5.1 Υπολογιστικό Περιβάλλον

Ο υπολογιστής που χρησιμοποιήθηκε για τα πειράματα διαθέτει επεξεργαστή Intel Core i5 6500 με 16 GB μνήμης RAM και λογισμικό Windows 10. Η δημιουργία των μοντέλων έγινε στην 3.7 έκδοση της python[1] με χρήση του εργαλείου μοντελοποίησης pyomo[2] στην έκδοση 5.7.2.

Για την επίλυση των μοντέλων χρησιμοποιήθηκε κυρίως το λογισμικό επίλυσης cplex στην έκδοση 12.10 και δευτερευόντως το λογισμικό επίλυσης gurobi έκδοση 9.1. Το λογισμικό gurobi χρησιμοποιήθηκε σε περιπτώσεις όπου εμφανίζονται υψηλοί χρόνοι επίλυσης προκειμένου να επιχειρηθεί μια σύγκριση μεταξυ των δύο λογισμικών πάνω στους χρόνους επίλυσης που καταφέρνουν να επιτύχουν.

Η python είναι μια διερμηνευόμενη (interpreted), γενικού σκοπού (general purpose) γλώσσα προγραμματισμού υψηλού επιπέδου. Ανήκει στις γλώσσες προστακτικού προγραμματισμού (Imperative programming) και υποστηρίζει τόσο το διαδικαστικό (procedural programming) όσο και το αντικειμενοστραφές (object-oriented programming) προγραμματιστικό υπόδειγμα (programming paradigm). Είναι δυναμική γλώσσα προγραμματισμού (dynamically typed) και υποστηρίζει συλλογή απορριμμάτων (garbage collection ή GC). Δημιουργήθηκε από τον Ολλανδό Γκίντο βαν Ρόσσουμ (Guido van Rossum) στο ερευνητικό κέντρο Centrum Wiskunde & Informatica (CWI) το 1989 και κυκλοφόρησε για πρώτη φορά το 1991. Ο κύριος στόχος της είναι η αναγνωσιμότητα του κώδικά της και η ευκολία χρήσης της. Το συντακτικό της επιτρέπει στους προγραμματιστές να εκφράσουν έννοιες σε λιγότερες γραμμές κώδικα από ότι θα ήταν δυνατόν σε άλλες γλώσσες.

Το Pyomo είναι ένα πακέτο λογισμικού ανοιχτού κώδικα βασισμένο στην python το οποίο διαθέτει μια πληθώρα δυνατοτήτων για την δημιουργία , την λύση και την ανάλυση μοντέλων βελτιστοποίησης. Η βασική δυνατότητα που προσφέρει το εργαλείο pyomo είναι η μοντελοποίηση δομημένων εφαρμογών βελτιστοποίησης. Μπορεί να χρησιμοποιηθεί για τον ορισμό γενικών συμβολικών προβλημάτων, την δημιουργία συγκεκριμένων περιπτώσεων προβλημάτων και την λύση αυτών με χρήση διαφόρων λογισμικών ανοιχτού ή μη κώδικα. Το pyomo υποστηρίζει την ανάπτυξη και επίλυση ενός μεγάλου εύρους προβλημάτων που ανήκουν στις κατηγορίες του γραμμικού προγραμματισμού, τετραγωνικού προγραμματισμού, μη γραμμικού προγραμματισμού, μεικτού ακέραιου γραμμικού προγραμματισμού κ.α.

Τα αποτελέσματα που παρουσιάζονται στις παραγράφους 5.4 έως 5.6 αποτελούν το μέσο όρο που προκύπτει από την εκτέλεση πέντε επαναλήψεων για κάθε συνδυασμό κόμβων – οχημάτων που παρουσιάζεται. Στόχος των πειραμάτων αρχικά είναι η εξασφάλιση της σωστής λειτουργίας των μοντέλων και δευτερευόντως η εύρεση του μέγιστου συνδυασμού κόμβων – οχημάτων για τον οποίο μπορούν να λυθούν στην τελειότητα μέσα σε ένα εύλογο χρονικό διάστημα (μικρότερο των 2 ωρών). Επίσης μελετήθηκε η συμπεριφορά της αντικειμενικής συνάρτησης σε σχέση με τον αριθμό των οχημάτων.

[1] https://www.python.org/

[2] https://pyomo.readthedocs.io/en/stable/

5.2 Λογισμικό cplex

Το λογισμικό cplex χρησιμοποιείται για την επίλυση προγραμμάτων ακέραιου προγραμματισμού, γραμμικού προγραμματισμού, τετραγωνικού προγραμματισμού, μεικτού ακέραιου γραμμικού προγραμματισμού, κ.α.. Πήρε το όνομα του από τον γνωστό αλγόριθμο simplex που χρησιμοποιείται για την επίλυση προβλημάτων γραμμικού προγραμματισμού και δημιουργήθηκε από τον George Dantzig το 1947 ενώ σήμερα συνεχίζει να αναπτύσσεται από την IBM. Είναι συμβατό με τις γλώσσες προγραμματισμού C++, C#, Java, Python, καθώς και τα Microsoft Excel και Matlab. Επίσης το λογισμικό cplex είναι προσβάσιμο μέσω των ανεξάρτητων συστημάτων μοντελοποίησης AIMMS, AMPL, OptimJ και TOMLAB.

5.3 Λογισμικό Gurobi

Αντίστοιχα με το cplex και το λογισμικό Gurobi χρησιμοποιείται για την επίλυση προγραμμάτων ακέραιου, γραμμικού και τετραγωνικού προγραμματισμού καθώς και μεικτού ακέραιου γραμμικού προγραμματισμού, μεικτού ακέραιου τετραγωνικού προγραμματισμού, κ.α.. Δημιουργήθηκε το 2008 και πήρε το όνομα του από τα αρχικά των δημιουργών του Zonghao Gu, Edward Rothberg, Robert Bixby. Το Gurobi είναι προσβάσιμο από τις γλώσσες προγραμματισμού C, C++, .NET, Python, Matlab, R, αλλά και μέσω τον γλωσσών μοντελοποίησης AIMS, AMPL, GAMS και MPL.

5.4 Υπολογιστικά Αποτελέσματα Μοντέλου CVRPt

Στον Πίνακα 2 παρουσιάζονται τα αποτελέσματα που προέκυψαν κατά την λύση του μοντέλου από το λογισμικό cplex για αυξανόμενο αριθμό κόμβων αναζητώντας το μικρότερο δυνατό αριθμό οχημάτων που μπορεί να τους εξυπηρετήσει σε κάθε περίπτωση. Τα δεδομένα με τα οποία τροφοδοτούμε το μοντέλο παράγονται τυχαία στην αρχή του προγράμματος σύμφωνα με την ομοιόμορφη κατανομή.

Κόμβοι	Οχήματα (Min)	Αντικειμενική Συνάρτηση (μέση τιμή)	Ταχύτητα λύσης (μέση τιμή) (s)
10	2	48438	7.58
11	3	55781.05	95.74
12	3	58426.51	325.45
13	5	77155.34	8952.84

Πίνακας 2: Αποτελέσματα μοντέλου CVRPt για αυξανόμενο αριθμό κόμβων με στόχο την ελαχιστοποίηση των οχημάτων

Τα αποτελέσματα που παρουσιάζονται στον Πίνακα 2 αποτελούν την μέση τιμή των αποτελεσμάτων για πέντε επαναλήψεις του προγράμματος. Παρατηρούμε ότι για αριθμό κόμβων ίσο με δέκα το μοντέλο μας καταφέρνει να δώσει λύση με χρήση μόνο δύο οχημάτων με την ταχύτητα λύσης να βρίσκεται σε χαμηλά επίπεδα (7.5 δευτερόλεπτα). Έπειτα για αριθμό κόμβων έντεκα έως δώδεκα γίνεται χρήση τριών οχημάτων με τους χρόνους επίλυσης να βρίσκονται στα ενενήντα πέντε και τριακόσια είκοσι πέντε δευτερόλεπτα αντίστοιχα. Στην συνέχεια για αριθμό κόμβων ίσο με δεκατρία το μοντέλο χρησιμοποιεί πέντε οχήματα για την παροχή λύσης με τον χρόνο επίλυσης να εκτοξεύεται στα οκτώ χιλιάδες εννιακόσια πενήντα δύο δευτερόλεπτα. Δυστυχώς δεν καταφέραμε να αποσπάσουμε αποτελέσματα για αριθμό κόμβων μεγαλύτερο από δεκατρία μέσα σε εύλογα χρονικά πλαίσια. Η συμπεριφορά του προγράμματος, αναφορικά με τους χρόνους επίλυσης όσο αυξάνεται ο αριθμός των κόμβων, συμβαδίζει με αυτά που γνωρίζουμε από την βιβλιογραφία για την επίλυση NP-hard προβλημάτων.

Στον Πίνακα 3 καταγράφονται τα αποτελέσματα του μοντέλου CVRPt για σταθερό αριθμό κόμβων αλλά αυξανόμενο αριθμό οχημάτων.

Κόμβοι	Διαθέσιμα Οχήματα	Χρησιμοποιηθέντα οχήματα	Ταχύτητα λύσης (μέση τιμή) (s)
10	5	3	7,66
10	7	3	10,18
10	9	3	19,12
10	11	3	21,78
10	13	3	34,18

Πίνακας 3: *Αποτελέσματα CVRPt για σταθερό αριθμό κόμβων ίσο με 10 και αυξανόμενο αριθμό οχημάτων*

Μελετώντας τα αποτελεσμάτα που παρουσιάζονται στον Πίνακα 3 παρατηρούμε ότι η αύξηση του αριθμού των οχημάτων επηρεάζει την ταχύτητα λύσης του μοντέλου. Επίσης το μοντέλο μας κάνει επιτυχημένα την ελαχιστοποίηση των χρησιμοποιηθέντων οχημάτων καθώς παρατηρούμε ότι παρόλο που ο αριθμός των διαθέσιμων οχημάτων αυξάνεται τα οχήματα που χρησιμοποιούνται παραμένουν σταθερά.

Τα αποτελέσματα του Πίνακα 4 που ακολουθεί για αριθμό κόμβων ίσο με έντεκα και αυξανόμενο αριθμό οχημάτων παρουσιάζουν διαφορές σε σχέση με αυτά που αναλύσαμε προηγουμένως. Αρχικά παρατηρούμε μεγαλύτερα «άλματα» στην ταχύτητα λύσης του μοντέλου όσο αυξάνεται ο αριθμός των διαθέσιμων οχημάτων σε σχέση με τα αποτελέσματα του Πίνακα 3. Πρόσθετα βλέπουμε ότι σπάει το φράγμα των τριών οχημάτων για αριθμό διαθέσιμων οχημάτων μεγαλύτερο από επτά. Η χρήση τεσσάρων οχημάτων αρχικά προβληματίζει διότι παρατηρούμε ότι το μοντέλο έχει καταφέρει προηγουμένως να δώσει λύση χρησιμοποιώντας μόνο τρία οχήματα για αριθμό διαθέσιμων οχημάτων ίσο με πέντε.

Παρόμοια είκονα έχουμε και στα αποτελέσματα του Πίνακα 5 όπου εκεί παρατηρούμε διακύμανσεις στον αριθμό των οχημάτων που χρησιμοποεί το μοντέλο όσο αυξάνεται ο αριθμός των διαθέσιμων οχημάτων.

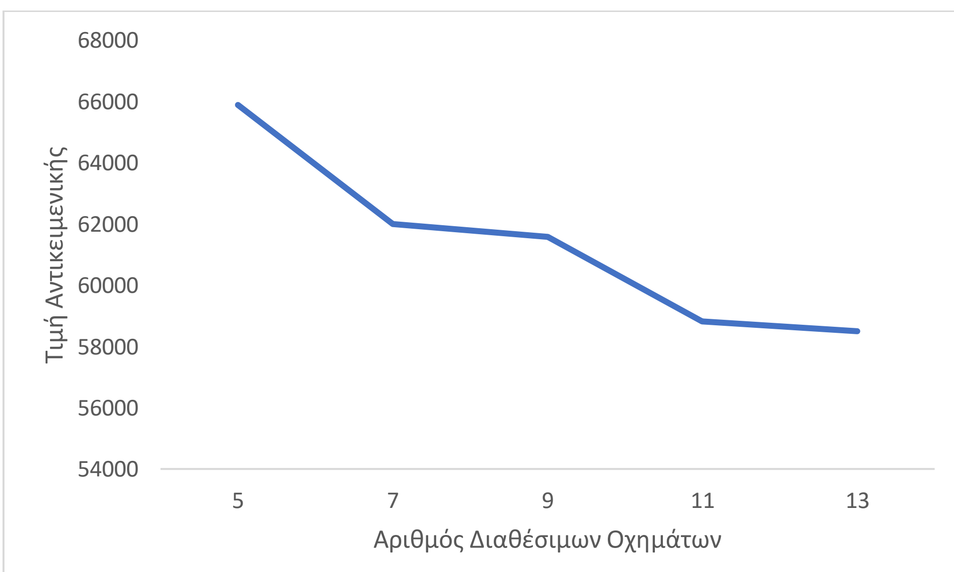
Κόμβοι	Διαθέσιμα Οχήματα	Χρησιμοποιηθέντα οχήματα	Ταχύτητα λύσης (μέση τιμή) (s)
11	5	3	10,63
11	7	4	40,86
11	9	4	68,41
11	11	4	138,37
11	13	4	441,18

Πίνακας 4: Αποτελέσματα CVRPt για σταθερό αριθμό κόμβων ίσο με 11 και αυξανόμενο αριθμό οχημάτων

Κόμβοι	Διαθέσιμα Οχήματα	Χρησιμοποιηθέντα οχήματα	Ταχύτητα λύσης (μέση τιμή) (s)
12	5	4	71,22
12	7	3	169,98
12	9	4	372,99
12	11	3	271,32
12	13	3	774,05

Πίνακας 5: Αποτελέσματα CVRPt για σταθερό αριθμό κόμβων ίσο με 12 και αυξανόμενο αριθμό οχημάτων

Για να κατανοήσουμε γιατί εμφανίζονται διακυμάνσεις στον αριθμό των οχημάτων που χρησιμοποιεί το μοντέλο μελετάμε το διάγραμμα της Εικόνας 3 στο οποίο παρουσιάζεται η σχέση μεταξύ της τιμής της αντικειμενικής συνάρτησης του μοντέλου και του αριθμού οχημάτων. Τα δεδομένα που χρησιμοποιούνται στο διάγραμμα αφορούν την περίπτωση που παρουσιάζεται στον Πίνακα 4.



Εικόνα 3: Σχέση Τιμής Αντικειμενικής Συνάρτησης και Αριθμού Διαθέσιμων Οχημάτων του Πίνακα 4

Παρατηρούμε λοιπόν ότι η τιμή της αντικειμενικής συνάρτησης του μοντέλου εμφανίζει πτωτική τάση όσο ο αριθμός των οχημάτων που μπορεί να χρησιμοποιήσει το λογισμικό για την λύση του προβλήματος αυξάνεται. Έτσι λοιπόν, αντιλαμβανόμαστε ότι τα νέα οχήματα που προστίθενται στον στόλο των διαθέσιμων οχημάτων σε κάθε βήμα, διαθέτουν χαμηλότερες τιμές κόστους (C_{kij}) δίνοντας την δυνατότητα στο λογισμικό να δημιουργήσει πιο οικονομικές διαδρομές που αποτελεί και τον στόχο του μοντέλου δρομολόγησης μας.

Αντίστοιχη λογική με αυτή που διατυπώθηκε παραπάνω διέπει τα αποτελέσματα που παρουσιάζονται στον Πίνακα 6 που ακολουθεί και αφορούν πείραμα με σταθερό αριθμό κόμβων ίσο με δεκατρία και αυξανόμενο αριθμό διαθέσιμων οχημάτων. Για τους συνδυασμούς κόμβων - οχημάτων του Πίνακα 6, ιδιαίτερα στην τελευταία περίπτωση, βλέπουμε ότι οι χρόνοι λύσης ξεπερνούν τα εβδομήντα τρία λεπτά όταν για αριθμό κόμβων ίσο με δέκα (Πίνακας 3) είχαμε χρόνους λύσης περίπου ίσους με τριάντα τέσσερα δευτερόλεπτα.

Κόμβοι	Διαθέσιμα Οχήματα	Χρησιμοποιηθέντα οχήματα	Ταχύτητα λύσης (μέση τιμή) (s)
13	5	4	615,3
13	7	4	2135,63
13	9	5	2856,5
13	11	3	2538,98
13	13	4	4430,56

Πίνακας 6: Αποτελέσματα CVRPt για σταθερό αριθμό κόμβων ίσο με 13 και αυξανόμενο αριθμό οχημάτων

Παρατηρώντας τα αποτελέσματα στους παραπάνω πίνακες μπορούμε με ασφάλεια να καταλήξουμε στο συμπέρασμα ότι η αύξηση του μεγέθους του πειράματος (αριθμός κόμβων – αριθμός οχημάτων) οδηγεί σε εκθετική αύξηση των χρόνων που απαιτούνται για την λύση του μοντέλου.

5.5 Υπολογιστικά Αποτελέσματα VRPTWs

Στον Πίνακα 7 που ακολουθεί παρουσιάζονται τα αποτελέσματα που προέκυψαν κατά την λύση του μοντέλου από το λογισμικό cplex για αυξανόμενο αριθμό κόμβων με στόχο την εύρεση των γρηγορότερων διαδρομών και την ελαχιστοποίηση των οχημάτων. Το μοντέλο καταφέρνει να δώσει λύση για αριθμό κόμβων (πελατών) έως δώδεκα για μέγεθος στόλου οχημάτων ίσο με δέκα (Πίνακας 7). Στο συγκεκριμένο μοντέλο τα δεδομένα που παράγουμε τυχαία και έπειτα χρησιμοποιούμε παίζουν μεγάλο ρόλο στον αριθμό των κόμβων που τελικώς καταφέρνει να λύσει το μοντέλο. Πιο συγκεκριμένα τα δεδομένα που αφορούν τον πίνακα κόστους – χρόνου C_{kij} ,τα χρονικά παράθυρα στα οποία μπορεί να επισκεφθεί ένα όχημα τον κόμβο και τον χρόνο εξυπηρέτησης του οχήματος στον κάθε κόμβο παράγονται τυχαία σύμφωνα με την ομοιόμορφη κατανομή μεταξύ μιας μέγιστης και ελάχιστης τιμής. Έτσι λοιπόν παρατηρούμε ότι το μοντέλο δεν «δυσκολεύεται» να δώσει λύση καθώς οι χρόνοι επίλυσης είναι πολύ χαμηλοί αλλά στην συγκεκριμένη περίπτωση είναι οι χρονικοί περιορισμοί στην εξυπηρέτηση των κόμβων από τα οχήματα που κρατούν τον αριθμό των κόμβων χαμηλά.

Κόμβοι	Οχήματα (Min)	Αντικειμενική (μέση τιμή) (s)	Ταχύτητα λύσης (μέση τιμή) (s)
10	6	28729.38	0.18
11	6	32315.45	0.11
12	10	32339.06	0.218

Πίνακας 7: Αποτελέσματα μοντέλου VRPTWs για αυξανόμενο αριθμό κόμβων και ελαχιστοποίηση αριθμού οχημάτων

Στον Πίνακα 8 παρουσιάζονται τα αποτελέσματα της επίλυσης του μοντέλου, με αλλαγή της μήτρας (seed) της συνάρτησης rand της python με την οποία παράγονται τυχαία τα δεδομένα και αύξηση του αριθμού οχημάτων.

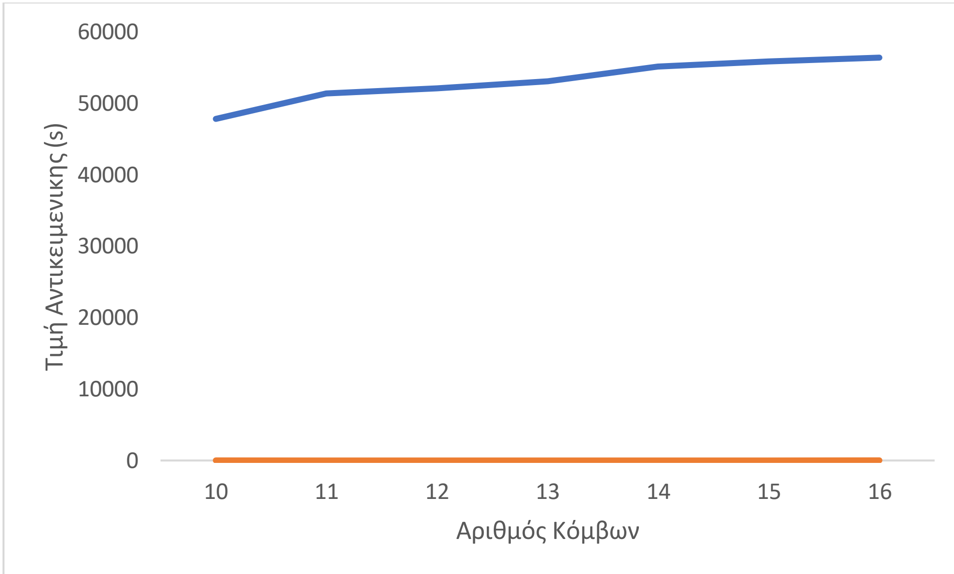
Κόμβοι	Διαθέσιμα Οχήματα	Χρησιμοποιηθέντ α οχήματα	Ταχύτητα λύσης (μέση τιμή) (s)
10	20	6	0,05
11	20	7	0,16
12	20	10	0,03
13	20	10	0,2
14	20	12	0,06
15	20	13	0,08
16	20	15	0,11

Πίνακας 8: Αποτελέσματα μοντέλου VRPTWs για seed της συνάρτησης rand ισο με 0

Μετά τις δύο αυτές αλλαγές παρατηρούμε ότι το πρόγραμμα καταφέρνει να δώσει λύση για αριθμό κόμβων ίσο με δεκαέξι. Ο λόγος για τον οποίον αυτές οι δύο αλλαγές οδηγούν σε αύξηση του αριθμού των κόμβων που μπορεί να επιλύσει το πρόγραμμα είναι:

α) η αλλαγή της μήτρας (seed) σύμφωνα με την οποία παράγονται τυχαία τα δεδομένα για τα χρονικά παράθυρα, τον πίνακα κόστους και τους χρόνους εξυπηρέτησης οδηγεί στην δημιουργία δεδομένων που επιτρέπουν εξυπηρέτηση περισσότερων κόμβων

β) η αύξηση του αριθμού οχημάτων αυξάνει τις πιθανότητες να υπάρξει όχημα με τιμή κόστους - χρόνου ικανή να εξυπηρετήσει περισσότερους κόμβους.



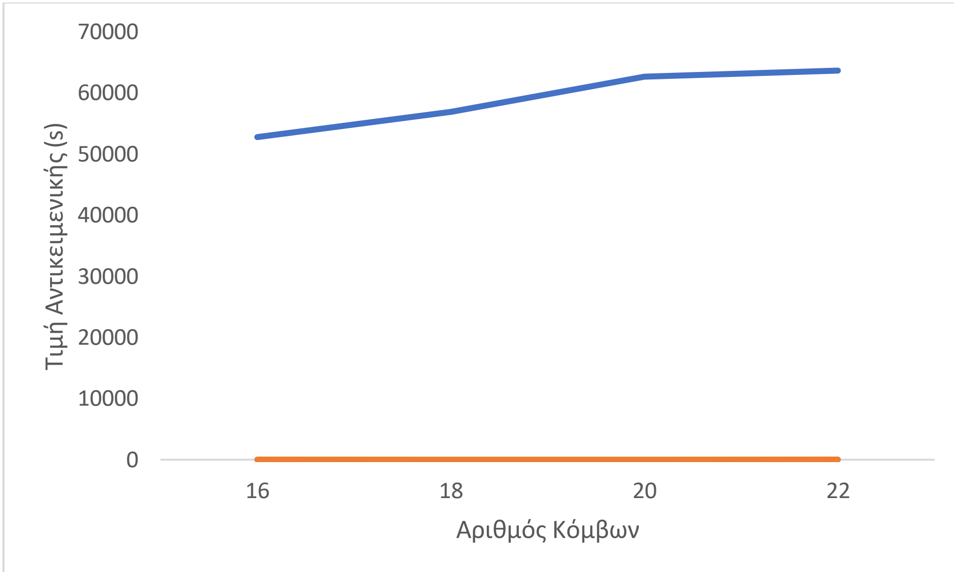
Εικόνα 4: Σχέση αντικειμενικής συνάρτησης και αριθμού κόμβων για σταθερό αριθμό οχημάτων ίσο με 20

Στην Εικόνα 4 παρουσιάζονται οι αντίστοιχες τιμές της αντικειμενικής συνάρτησης για τον Πίνακα 8. Παρατηρούμε ότι ο αυξανόμενος αριθμός κόμβων οδηγεί αντίστοιχα σε μεγαλύτερες τιμές της αντικειμενικής συνάρτησης.

Έπειτα στον Πίνακα 9 παραθέτουμε τα αποτελέσματα για τιμή μήτρας (seed) ίση με μηδέν και αριθμό οχημάτων ίσο με είκοσι που αποτελεί το μεγαλύτερο σε μέγεθος πείραμα (αριθμός κόμβων – αριθμός διαθέσιμων οχημάτων) που καταφέραμε να λύσουμε με το συγκεκριμένο μοντέλο.

Κόμβοι	Διαθέσιμα Οχήματα	Χρησιμοποιηθέντα οχήματα	Ταχύτητα λύσης (μέση τιμή) (s)
16	20	14	0,13
18	20	16	0,16
20	20	18	0,2
22	20	20	0,24

Πίνακας 9: Αποτελέσματα μοντέλου VRPTWs για seed της συνάρτησης rand ίσο με 2



Εικόνα 5: Σχέση αντικειμενικής συνάρτησης και αριθμού κόμβων για σταθερό αριθμό οχημάτων ίσο με 20

Τέλος, στην Εικόνα 5 παραθέτουμε την σχέση μεταξύ τιμής της αντικειμενικής συνάρτησης και του αριθμού κόμβων για τα δεδομένα του Πίνακα 9, η οποία εμφανίζεται να έχει ανοδική πορεία όσο αυξάνεται ο αριθμός των κόμβων που χρησιμοποιούνται από το μοντέλο.

5.6 Σύγκριση Λογισμικών

Σε αυτή την παράγραφο πραγματοποιούμε μια σύγκριση μεταξύ των λογισμικών gurobi και cplex αναφορικά με τους χρόνους στους οποίους καταφέρνουν να επιλύσουν το μοντέλο CVRPt που παρουσιάστηκε παραπάνω για τα δεδομένα που χρησιμοποιήθηκαν στην παράγραφο 5.4.

Στον Πίνακα 10 και στον Πίνακα 11 που ακολουθούν παρουσιάζονται συγκριτικά οι χρόνοι που χρειάστηκαν τα δύο λογισμικά για την επίλυση του μοντέλου CVRPt για αριθμό κόμβων ίσο με δώδεκα και δεκατρία αντίστοιχα καθώς και αυξανόμενο αριθμό οχημάτων. Η επιλογή των δύο παραδειγμάτων από το προηγούμενο κεφάλαιο έγινε λόγω των μεγάλων χρόνων επίλυσης που παρουσίασε το λογισμικό cplex κατά την επίλυση τους.

Κόμβοι	Διαθέσιμα Οχήματα	Ταχύτητα λύσης gurobi (s)	Ταχύτητα λύσης cplex (s)
12	5	81,97	71,22
12	7	193,4	169,98
12	9	317,125	372,92
12	11	231,02	271,32
12	13	433,25	774,05

Πίνακας 10: Σύγκριση ταχύτητας επίλυσης με τα λογισμικά cplex, gurobi για αριθμό κόμβων ίσο με 12

Κόμβοι	Διαθέσιμα Οχήματα	Ταχύτητα λύσης gurobi (s)	Ταχύτητα λύσης cplex (s)
13	5	501,6	615,3
13	7	1049,85	2135,63
13	9	1163,6	2856,5
13	11	1831,16	2538,98
13	13	3320,1	4430,56

Πίνακας 11:Σύγκριση ταχύτητας επίλυσης με τα λογισμικά cplex, gurobi για αριθμό κόμβων ίσο με 13

Όπως είναι προφανές από τα αποτελέσματα των δύο πινάκων το λογισμικό gurobi καταφέρνει γρηγορότερες λύσεις στην πλειοψηφία των περιπτώσεων και ιδιαίτερα για μεγαλύτερους αριθμούς διαθέσιμων οχημάτων όπου η διαδικασία της λύσης αρχίζει να γίνεται ιδιαιτέρως χρονοβόρα.

6. Συμπεράσματα και Μελλοντική Έρευνα

Όπως πρωτοαναφέρθηκε, οι στόχοι που τέθηκαν αρχικά κατά την συγγραφή της διπλωματικής εργασίας ήταν η κατανόηση της βιβλιογραφίας αναφορικά με την δρομολόγηση οχημάτων, η δημιουργία ενός μοντέλου εμπνευσμένου από τα ήδη υπάρχοντα στην βιβλιογραφία και σαν φυσικό επακόλουθο όλων αυτών η εξοικείωση με την δημιουργία προγραμμάτων σε python. Τελικώς, στα πλαίσια της εργασίας αναπτύχθηκαν δύο μοντέλα δρομολόγησης τα οποία ανήκουν στις κατηγορίες των προβλημάτων δρομολόγησης με περιορισμένη χωρητικότητα και χρονικούς περιορισμούς αντίστοιχα. Τα δεδομένα με τα οποία τροφοδοτήσαμε τα μοντέλα μας όπως χρόνοι μετακίνησης, κόστος μετακίνησης, χρονικά παράθυρα κ.α., δημιουργήθηκαν με τυχαίο τρόπο σύμφωνα με την ομοιόμορφη κατανομή προσομοιάζοντας ρεαλιστικές συνθήκες.

Στην πρώτη περίπτωση το μοντέλο CVRPt που ανήκει στην κατηγορία των προβλημάτων δρομολόγησης περιορισμένης χωρητικότητας εμπλουτίστηκε με μία τρισδιάστατη μεταβλητή κόστους για αυξημένη πληροφορία αναφορικά με τις δυνατότητες των οχημάτων καθώς και με περιορισμό που εξασφαλίζει την τήρηση του εργασιακού οκταώρου. Πρόσθετα, το μοντέλο VRPTWs έχει εμπλουτιστεί σε σχέση με τα αντίστοιχα μοντέλα της βιβλιογραφίας με μία τρισδιάστατη μεταβλητή χρόνου C_{kij} που περιλαμβάνει διαφορετικές τιμές για την κίνηση από το i στο j του κάθε οχήματος αλλά και με περιορισμούς που οδηγούν στην ελαχιστοποίηση των χρησιμοποιηθέντων οχημάτων και τήρησης του εργασιακού οκταώρου.

Τα πειράματα που πραγματοποιήσαμε στα δύο μοντέλα αποδεικνύουν ότι τα μοντέλα λειτουργούν σωστά καθώς καταφέρνουν σε κάθε περίπτωση να επιτύχουν τον στόχο τους, δηλαδή τη βελτιστοποίηση της δρομολόγησης των οχημάτων υπό προϋποθέσεις. Η χρήση τρισδιάστατης μεταβλητής κόστους - χρόνου για τα οχήματα (C_{kij}) αντί δισδιάστατης που χρησιμοποιείται συνήθως, καταφέρνει να αυξήσει το επίπεδο παραμετροποίησης που μπορεί να επιτευχθεί καθώς το κόστος ή ο χρόνος δεν είναι όμοιος για όλα τα οχήματα που πραγματοποιούν την ίδια διαδρομή αλλά διαφορετικός για κάθε όχημα. Με αυτόν τον τρόπο καταφέρνουμε να προσομοιάσουμε όχι μόνο τις διαφορετικές επιδόσεις των οχημάτων που συναντάμε στην πραγματικότητα αλλά και τις διαφορετικές επιδόσεις των οδηγών. Επίσης οι περιορισμοί που αφορούν το εργασιακό οκτάωρο αλλά και την μείωση των οχημάτων που απαιτούνται αυξάνουν την ρεαλιστικότητα των μοντέλων.

Η επίλυση του μοντέλου CVRPt με το λογισμικό cplex κατάφερε να δώσει σωστά αποτελέσματα σε εύλογο χρόνο για μικρό όμως αριθμό κόμβων καθώς ο χρόνος επίλυσης παρατηρήσαμε ότι αυξήθηκε εκθετικά σε συνδυασμό με την αύξηση των κόμβων. Επίσης παρατηρήθηκε ότι ο αριθμός των οχημάτων συμβάλει εξίσου με τον αριθμό των κόμβων στο επίπεδο που κυμαίνεται ο χρόνος επίλυσης του προγράμματος. Οι υψηλοί χρόνοι επίλυσης που συναντάμε αυξάνοντας των αριθμό κόμβων - οχημάτων επαληθεύονται και από την βιβλιογραφία καθώς τα προβλήματα δρομολόγησης ανήκουν στην κατηγορία των Np-hard προβλημάτων συνδυαστικής βελτιστοποίησης.

Αναφορικά με το μοντέλο VRPTWs παρατηρήσαμε ότι η ύπαρξη χρονικών περιορισμών στην εξυπηρέτηση του κάθε κόμβου και τον περιορισμό του εργασιακού οκταώρου σε συνδυασμό με την τυχαία δημιουργία όλων των δεδομένων οδήγησε σε πολλές περιπτώσεις σε αδυναμία εύρεσης λύσης. Τέλος, στην σύγκριση μεταξύ των λογισμικών cplex και gurobi παρατηρήσαμε ότι ο δεύτερος πετυχαίνει αισθητά ταχύτερους χρόνους επίλυσης και για αυτό προτείνετε η χρήση του για επίλυση παρόμοιων προβλημάτων.

Όπως έχουμε προαναφέρει τα μοντέλα δρομολόγησης οχημάτων ανήκουν στην κατηγορία των NP-hard προβλημάτων συνδυαστικής βελτιστοποίησης τα οποία παρουσιάζουν εκθετική αύξηση των χρόνων επίλυσης όσο αυξάνουμε το μέγεθος τους (αριθμό κόμβων – οχημάτων). Εύλογα λοιπόν θα παρουσίαζε ιδιαίτερο ενδιαφέρον η χρήση ενός ευρετικού αλγορίθμου για τα μοντέλα που παρουσιάζονται στην παρούσα εργασία προκειμένου να αποσπάσουμε λύσεις σε λογικούς χρόνους για μεγάλους αριθμούς κόμβων και οχημάτων.

Πρόσθετα, θα μπορούσαμε να χρησιμοποιήσουμε κάποια από τα έτοιμα πακέτα δεδομένων που υπάρχουν στην βιβλιογραφία και αφορά τα προβλήματα δρομολόγησης προκειμένου να πραγματοποιήσουμε συγκρίσεις μεταξύ των μοντέλων της εργασίας και άλλων που εμφανίζονται στην βιβλιογραφία. Επίσης θα μπορούσε να γίνει χρήση πραγματικών δεδομένων που αφορούν τους χρόνους μετακίνησης ανάμεσα στους πελάτες, ωράρια καταστημάτων, χρόνους εξυπηρέτησης που θα αποσπούσαμε από κάποια επιχείρηση στην αγορά.

Τέλος, θα άξιζε να μελετηθεί η τροποποίηση του μοντέλου VRPTWs έτσι ώστε να μην περιέχει απλά έναν περιορισμό ελέγχου τήρησης του εργασιακού οκταώρου για τα οχήματα, αλλά να δίνεται η δυνατότητα λειτουργίας των οχημάτων σε βάρδιες αυξάνοντας έτσι τον ρεαλισμό του μοντέλου.

Βιβλιογραφία

Baker, B. M. & Ayechew, M. A., 2003. A genetic algorithm for the vehicle routing problem. *Computers & Operations Research,* pp. 787-800.

Baldacci, R., Christofides, N. & Mingozzi, A., 2007. n exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming,* pp. 351-385.

Baldacci, R., Hadjiconstantinou, E. & Mingozzi, A., 2004. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations Research,* Issue 5, pp. 723-738.

Bramel, J. & Simchi-Levi, D., 1996. Probabilistic Analyses and Practical Algorithms for the Vehicle Routing Problem with Time Windows. *Operations Research,* pp. 423-528.

Braysy, O., Gendreau, M. & Dullaert, W., 2004. Evolutionary Algorithms for the Vehicle Routing Problem with TimeWindows. *Journal of Heuristics,* pp. 587-611.

Campbell, A. M. & Savelsbergh, M., 2004. Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation Science,* Issue 3, pp. 369-378.

Chiang, W.-C. & Russell, R. A., 1996. Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research,* pp. 3-27.

Clarke, G. & Wright, J. W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research,* 12(4), pp. 568-582.

Cordeau, J.-F., Laporte, G. & Mercier, A., 2001. A unified tabu search heuristic for vehicle routing problems with time windows. *The Journal of the Operational Research Society,* Issue 8, pp. 928-936.

Dantzig, G. B. & Ramser, J. H., 1959. The truck dispatching problem. *Management Science,* Issue 1, pp. 80-92.

Ellabib, I., Basir, O. A. & Calamai, P., 2002. An Experimental Study of a Simple Ant Colony System for the Vehicle Routing Problem with Time Windows. *LNCS,* pp. 53-64.

Fukasawa, R. και συν., 2004. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming,* pp. 491-511.

Garey, R. M. & Johnson, S. D., 1990. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* New York: W. H. Freeman & Co.

Kulkarni, R. V. & Bhave, P. R., 1985. Integer programming formulations of vehicle routing problems. *European Journal of Operational Research,* Issue 1, pp. 58-67.

Laporte, G., 1992. The Vehicle Routing Problem: An overview of exact and approximate algorithms. *European Journal of Operational Research,* pp. 345-358.

Laporte, G. & Osman, I. H., 1995. Routing problems: A bibliography. *Annals of Operations Research,* pp. 227-262.

Lenstra, J. K. & Rinnooy Kan, A. H., 1981. Complexity of vehicle routing and scheduling problems. *Networks,* Issue 2, pp. 221-227.

Lysgaard, J., Letchford, A. N. & Eglese, R. W., 2004. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming,* pp. 423-445.

Madsen, O. B. G., Ravn, H. F. & Rygaard, J. M., 1995. A heuristic algorithm for the dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research,* pp. 193-208.

Maffioli, F., 2003. The vehicle routing problem: A book review. *4OR,* Issue 2, pp. 149-153.

Rochat, Y. & Taillard, R. E., 1995. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics,* pp. 147-167.

Sanjeev , A. & Boaz, B., 2009. *Computational Complexity: A Modern Approach.* New York: Cambridge University Press.

Solomon, M. M., 1987. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research,* pp. 254-265.

Taillard, R. E., 1993. Parallel iterative search methods for vehicle routing problems. *Networks,* pp. 661-673.

Λάιος, Λ., 2010. *Διοίκηση Εφοδιασμού.* Αθήνα: Humantec MEPE.

Παράρτημα A

Κώδικας CVRPt

```
#Imports
import random
import numpy as np
rnd = np.random
rnd.seed(1)
n = 5 # pelates (sinolikoi komvoi = pelates + depot( n+1))
k = 7  #arithmos fortigwn
C = np.random.uniform(low=3600, high=3*3600, size=(k,n+1,n+1)) #dimiourgia pinaka kostous (xronou k oximatos na metavei apo i--> j)
# diamorfwsi keliwn este (i,j)= (j,i)
for kk in range(k):
    for ii in range(n+1):
        for jj in range(n+1):
            if jj > ii:
                C[kk,jj,ii]=C[kk,ii,jj]
Wi = np.random.randint(low=5,high=50,size=(n+1)) #paraggelia pelatwn
l = [i for i in range(1,k+1)]
Qk =[100 for i in l]#apothikeutiki dinatotita 100
# xronos metavasis apo i se j
tij = np.random.uniform(low=300, high=3600, size=(n+2,n+2))
for ii in range(n+2):
    for jj in range(n+2):
        if jj > ii:
            tij[jj,ii]=tij[ii,jj]
tekj = np.random.uniform(low=300, high=1500, size=(n+1)) # xronos eksipiretisis pelati
```

```
H1 = 30600 # tirisi 8orou (8,5 wres se second)
#import pyomo
from __future__ import division
import pyomo.environ as pyo
model = pyo.ConcreteModel()
# DIMIOURGIA DEIKTWN GIA PARAMETROUS KAI METAVLITES
model.I = pyo.RangeSet(1,n+1) #deiktis gia to i pou perilamvanei to depot
model.In = pyo.RangeSet(2,n+1)#deiktis gia to i pou den perilamvanei to depot
model.K = pyo.RangeSet(1,k) #deiktis gia fortiga
#DIMIOURGIA PARAMETRWN KAI ANATHESI TIMWN
model.TIJ = pyo.Param(model.I,model.I,initialize=lambda model,i,j: tij[i-1][j-1])
model.TEKJ = pyo.Param(model.I, initialize=lambda model,i: tekj[i-1])
model.Ckij = pyo.Param(model.K,model.I,model.I,initialize=lambda model, k, i, j: C[k-1][i-1][j-1])
model.WI = pyo.Param(model.In,initialize = lambda model,i: Wi[i-1])
model.QK = pyo.Param(model.K,initialize = lambda model,k: Qk[k-1])
#DIMIOURGIA METAVLITWN
model.Xkij = pyo.Var(model.K,model.I,model.I,within = pyo.Binary)
model.u = pyo.Var(model.In,within = pyo.NonNegativeIntegers)
#DIMIOURGIA MONTELOU
#ANTIKEIMENIKI SINARTISI
def obj_func(model):
  return sum(model.Ckij[k,i,j] * model.Xkij[k,i,j] for i in model.I for j in model.I if i!=j for k in model.K)
model.objective = pyo.Objective(rule=obj_func,sense=pyo.minimize)
#PERIORISMOI
def rule_const1(model,I):
  return sum(model.Xkij[k,I,j] for j in model.I for k in model.K) == 1
model.const1 = pyo.Constraint(model.In,rule = rule_const1)
def rule_const2(model,K):
  return sum(model.Xkij[K,i,1] for i in model.In) <= 1
```

```
model.const2 = pyo.Constraint(model.K,rule = rule_const2)
def rule_const3(model):
   return sum(model.Xkij[k,1,j] for j in model.In for k in model.K) >= 1
model.const3 = pyo.Constraint(rule = rule_const3)
def rule_const4(model,H,K):
   return (sum(model.Xkij[K,i,H] for i in model.I) - sum(model.Xkij[K,H,j] for j in model.I) == 0)
model.const4 = pyo.Constraint(model.I,model.K,rule = rule_const4)
def rule_const5(model,I,J):
    return(model.u[I] - model.u[J] + n*sum(model.Xkij[k,I,J] for k in model.K)<= n-1)
model.const5 = pyo.Constraint(model.In,model.In,rule = rule_const5)
def rule_const6(model,K):
   return sum(model.WI[i] * model.Xkij[K,i,j] for i in model.In for j in model.I) <= model.QK[K]
model.const6 = pyo.Constraint(model.K, rule = rule_const6)
def rule_const7(model,K):
   return sum(model.Xkij[K,i,j]*(model.TIJ[i,j]+model.TEKJ[j])for i in model.I for j in model.I) <=H1
model.const7 = pyo.Constraint(model.K, rule = rule_const7)
#model.pprint()
#LYSH
solver = pyo.SolverFactory('cplex')
result = solver.solve(model,tee = False)
#EKTIPOSI APOTELESMATWN
print(result)
```

Κώδικας VRPTWs

```
n = 13#pelates anaferomaste se pelates oxi komvous, oi komvoi tha einai n+2 opou 1 afetiria kai n+2 termatismos)

l = 20#fortiga

#IMPORTS

import random

import numpy as np

rnd = np.random

rnd.seed(2)

# DIMIOURGIA DEIKTI POU ANTIPROSWPEUEI TA FORTIGA GIA TIN DIMIOURGIA DEDOMENWN POU TA AFOROUN

k = [i for i in range(1,l+1)]

Qk = [500 for i in k]#apothikeutiki dinatotita 500

 # xronos eksipiretisis se kathe komvo i gia to kathe oxima k

tij = np.random.uniform(low=300, high=1300, size=(n+2,n+2))

for ii in range(n+2):

   for jj in range(n+2):

       if jj > ii:

          tij[jj,ii]=tij[ii,jj]

ai=np.random.uniform(low=0,high=3600,size=(n+2)) #anw orio gia arxi eksipiretisis

ai[0]=0

bi=np.zeros(n+2)

for d in range (n+2):

   bi[d]=np.random.uniform(low=(ai[d]+600),high=3600*8)#katw orio gia arxi eksipiretisis

bi[n+1]=3600*8

C4= np.random.uniform(low=1200, high=3600, size=(l,n+2,n+2)) #dimiourgia pinaka kostous

for kk in range(l):

   for ii in range(n+2):

      for jj in range(n+2):
```

```
        if jj > ii:
            C4[kk,jj,ii]=C4[kk,ii,jj]
H1 = 30600
M1 = 999999999 # MEGALOS CONSTANT
XW =np.zeros(n+1)
XW = np.random.randint(low=0,high=50,size=(n+1))
#IMPORT PYOMO
from __future__ import division
import pyomo.environ as pyo
model = pyo.ConcreteModel()
# DIMIOURGIA DEIKTWN (INDICES) GIA TO MONTELO
model.In = pyo.RangeSet(2,n+1)
model.Is = pyo.RangeSet(1,n+2)
model.K = pyo.RangeSet(1,l)
# DIMIOURGIA PARAMETRWN KAI ARXIKOPOIISI TOUS ME TA DEDOMENA POU KATASKEUASAME STIN ARXI
model.CKIJ = pyo.Param(model.K,model.Is,model.Is,initialize=lambda model, k, i, j: C4[k-1][i-1][j-1])
model.QK = pyo.Param(model.K,initialize=lambda model,k:Qk[k-1])
model.TIJ = pyo.Param(model.Is,model.Is,initialize=lambda model,i,j: tij[i-1][j-1])
model.AI = pyo.Param(model.Is,initialize =lambda model,i: ai[i-1])
model.BI = pyo.Param(model.Is,initialize = lambda model,i: bi[i-1])
model.XW = pyo.Param(model.In,initialize = lambda model,i: XW[i-1])
#DIMIOURGIA METAVLITWN
model.XKIJ = pyo.Var(model.K,model.Is,model.Is,within =pyo.Binary)
model.SKI = pyo.Var(model.K,model.Is,domain=pyo.NonNegativeReals,bounds=(0,H1))
#ANTIKEIMENIKI
def obj_func(model):
  return sum(model.CKIJ[k,i,j] * model.XKIJ[k,i,j] for k in model.K for i in model.Is for j in model.Is if i!=j)
model.objective = pyo.Objective(rule=obj_func,sense=pyo.minimize)
```

```
#PERIORISMOI

#periorismos pou afora to poses fores tha episkefthei ena oxima enan pelati (provlima=anagastiko xalaroma)

def rule_const1(model,In):

   return sum(model.XKIJ[k,In,j] for k in model.K for j in model.Is if In!=j)== 1

model.const1 = pyo.Constraint(model.In,rule = rule_const1)

#periorismos pou anagazei ta oximata na feugoun apo tin afetiria

def rule_const2(model,K):

   return sum(model.XKIJ[K,1,j]for j in model.Is if j!=1) == 1

model.const2 = pyo.Constraint(model.K,rule = rule_const2)

#periorismos pou anagazei ta oximata na kataligoun ston termatismo

def rule_const3(model,K):

   return sum(model.XKIJ[K,i,n+2] for i in model.Is if i!=n+2) == 1

model.const3 = pyo.Constraint(model.K,rule = rule_const3)

#periorismos pou anagazei ena oxima pou episkeptetai enan komvo na feugei apo auton kai na pigaine se allon

def rule_const4(model,K,H):

   return (sum(model.XKIJ[K,i,H] for i in model.Is if i!= H) - sum(model.XKIJ[K,H,j] for j in model.Is if j!=H) ) == 0

model.const4 = pyo.Constraint(model.K,model.In,rule = rule_const4)

#periorismoi 5 kai 6 --> elegxoun oti ta fortiga ftanoun ston pelati entos tou parathirou eksipiretisis

def rule_const5(model,K,I):

   return model.SKI[K,I] >= model.AI[I]

model.const5 = pyo.Constraint(model.K,model.Is,rule = rule_const5)

def rule_const6(model,K,I):

   return (model.SKI[K,I] <= model.BI[I])

model.const6 = pyo.Constraint(model.K,model.Is,rule = rule_const6)

#periorismos pou elegxei an i paraggelia tou pelati ksepernaei to orio pou mporei na xwresei sto fortigo

def rule_const7(model,K):
```

```
    return (sum(model.XW[h] for h in model.In)*sum(model.XKIJ[K,i,j] for i in model.In for j in model.Is if i!=j) <= model.QK[K])

model.const7 = pyo.Constraint(model.K,rule = rule_const7)

#periorismos pou apotrepei na ksekinisei i eksipiretisi enos pelati prin ftasei to oxima ekei

def rule_const8(model,K,I,J):

    if I!=J:

        return model.SKI[K,I] + model.TIJ[I,J]+ model.CKIJ[K,I,J]- model.SKI[K,J] <= M1 * ((1-model.XKIJ[K,I,J]))

    else:

        return pyo.Constraint.Skip

model.const8 = pyo.Constraint(model.K,model.Is,model.Is,rule = rule_const8)

def rule_const10(model):

    return (sum(model.XKIJ[k,1,j] for k in model.K for j in model.Is if j!=1 )<= l)

model.const10 = pyo.Constraint(rule=rule_const10)

#model.pprint()

#LYSH

solver = pyo.SolverFactory('cplex')

result = solver.solve(model,tee = False)

#EKTIPOSI APOTELESMATWN

print(result)
```