ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ

ΤΜΗΜΑ ΔΙΟΙΚΗΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΔΗΜΗΤΡΑΣ ΤΥΠΑ

ΠΡΟΒΛΗΜΑ ΦΟΡΤΩΣΗΣ ΦΟΡΤΗΓΩΝ ΟΧΗΜΑΤΩΝ ΓΙΑ ΤΗ ΔΙΑΝΟΜΗ ΠΡΟΪΟΝΤΩΝ ΜΕ ΠΕΡΙΟΡΙΣΜΟΥΣ ΕΥΣΤΑΘΕΙΑΣ

Επιβλέπων : Κωνσταντίνος Ν. Ανδρουτσόπουλος – Επίκουρος Καθηγητής

Υποβληθείσα ως μέρος των απαιτήσεων για την απόκτηση
Μεταπτυχιακού Διπλώματος (MSc) στη Διοικητική Επιστήμη και Τεχνολογία

Αθήνα, Ιανουάριος 2022

Η σελίδα αυτή είναι σκόπιμα λευκή.



ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ

ΤΜΗΜΑ ΔΙΟΙΚΗΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΔΗΜΗΤΡΑΣ ΤΥΠΑ

ΠΡΟΒΛΗΜΑ ΦΟΡΤΩΣΗΣ ΦΟΡΤΗΓΩΝ ΟΧΗΜΑΤΩΝ ΓΙΑ ΤΗ ΔΙΑΝΟΜΗ ΠΡΟΪΟΝΤΩΝ ΜΕ ΠΕΡΙΟΡΙΣΜΟΥΣ ΕΥΣΤΑΘΕΙΑΣ

Επιβλέπων : Κωνσταντίνος Ν. Ανδρουτσόπουλος – Επίκουρος Καθηγητής

Υποβληθείσα ως μέρος των απαιτήσεων για την απόκτηση
Μεταπτυχιακού Διπλώματος (MSc) στη Διοικητική Επιστήμη και Τεχνολογία

Αθήνα, Ιανουάριος 2022



Βεβαίωση εκπόνησης Διπλωματικής εργασίας

«Δηλώνω υπεύθυνα ότι η συγκεκριμένη μεταπτυχιακή εργασία για τη λήψη του μεταπτυχιακού τίτλου σπουδών του ΠΜΣ στη Διοικητική Επιστήμη και Τεχνολογία του Τμήματος Διοικητικής Επιστήμης και Τεχνολογίας του Οικονομικού Πανεπιστημίου Αθηνών έχει συγγραφεί από εμένα προσωπικά και δεν έχει υποβληθεί ούτε έχει εγκριθεί στο πλαίσιο κάποιου άλλου μεταπτυχιακού ή προπτυχιακού τίτλου σπουδών στην Ελλάδα ή το εξωτερικό. Η εργασία αυτή έχοντας εκπονηθεί από εμένα, αντιπροσωπεύει τις προσωπικές μου απόψεις επί του θέματος. Οι πηγές στις οποίες ανέτρεξα για την εκπόνηση της συγκεκριμένης διπλωματικής αναφέρονται στο σύνολό τους, δίνοντας πλήρεις αναφορές στους συγγραφείς, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο».

(Υπογραφή)

31/01/2022

ΤΥΠΑ ΔΗΜΗΤΡΑ

Φοιτήτρια MSc στη Διοικητική Επιστήμη και Τεχνολογία



Περίληψη

Σκοπό της παρούσας Διπλωματικής εργασίας αποτελεί η ανάπτυξη μοντέλων για την επίλυση του προβλήματος φόρτωσης ενός βυτιοφόρου οχήματος διανομής καυσίμων με πολλαπλές στάσεις, λαμβάνοντας υπόψιν περιορισμούς ευστάθειας για κάθε φάση της διαδρομής. Για την επίτευξη του σκοπού αυτού πραγματοποιήθηκε η μοντελοποίηση και η ενσωμάτωση των περιορισμών ευστάθειας τριαξονικού βυτιοφόρου οχήματος, δηλαδή της μη υπέρβασης των μέγιστων επιτρεπόμενων ορίων βάρους κατ' άξονα και της ομοιόμορφης κατανομής του βάρους του φορτίου, σε μοντέλα μεικτού ακέραιου γραμμικού προγραμματισμού για τη φόρτωση οχήματος με ή χωρίς μετρητές στάθμης. Τα μοντέλα φόρτωσης που αναπτύχθηκαν εφαρμόστηκαν σε ένα σύνολο προβλημάτων, που αφορούν τη διανομή καυσίμων σε πέντε πρατήρια καυσίμων και η επίλυση πραγματοποιήθηκε μέσω του solver Cplex στη Java. Τα αποτελέσματα των πειραμάτων οδήγησαν στο συμπέρασμα ότι προκειμένου να εξασφαλιστεί η ευστάθεια του βυτιοφόρου οχήματος και κατ' επέκταση η οδική του ασφάλεια, μειώνεται η ποσότητα φόρτωσης, αυξάνονται οι συνολικές αποκλίσεις επί της αρχικής ζήτησης και μειώνεται το ποσοστό αξιοποίησης της χωρητικότητας του οχήματος.

Λέξεις Κλειδιά: μοντέλα φόρτωσης, πρόβλημα φόρτωσης, βυτιοφόρο όχημα, περιορισμοί ευστάθειας, όρια βάρους κατ' άξονα, ομοιόμορφη κατανομή, διανομή καυσίμων, μεικτός ακέραιος γραμμικός προγραμματισμός.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Η σελίδα αυτή είναι σκόπιμα λευκή.



Abstract

The purpose of this Diploma Thesis is to develop models to solve the loading problem of a multi-stop fuel tanker, taking into consideration load balance constraints during the entire delivery route. For this purpose, the modeling and integration of the load balance constraints of a three-axle tanker, i.e., the non-exceedance of the maximum permissible axle weight limits and the even distribution of the weight of the load, were carried out in Mixed Integer Linear Programming (MILP) models for the loading problem of a truck equipped or not with flow meters. The loading models that were developed, were applied to a set of problems, involving the fuel distribution at five service stations and the solution was developed using Cplex solver in Java. The computational results of the experiments demonstrated that in order to ensure the load balance of the tanker vehicle and consequently its road safety, the loading quantity is reduced, the total deviations of the quantity assigned to the vehicle compartments from the total demand are increased and the utilization rate of the vehicle capacity is reduced.

Keywords: loading models, loading problem, fuel tanker, load balance constraints, axle weight limits, even distribution, fuel distribution, Mixed Integer Linear Programming.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Η σελίδα αυτή είναι σκόπιμα λευκή.



Πίνακας Περιεχομένων

1 Εισαγωγή 1

1.1 Φόρτωση φορτηγών οχημάτων για τη διανομή προϊόντων 1

1.1.1 Επικίνδυνα φορτία- Συμφωνία ADR 2

1.1.2 Ατυχήματα κατά τη μεταφορά επικίνδυνων φορτίων 3

1.1.3 Χερσαία μεταφορά επικίνδυνων φορτίων μέσω βυτιοφόρων οχημάτων 4

1.1.4 Οδική συμπεριφορά βυτιοφόρων οχημάτων 5

1.1.5 Ευστάθεια βυτιοφόρου οχήματος- Βαθμός πλήρωσης 5

1.1.6 Φορτοεκφόρτωση βυτιοφόρων οχημάτων 6

1.1.7 Μέγιστα επιτρεπόμενα βάρη έμφορτου οχήματος και βάρη ανά άξονα 7

1.1.8 Φόρτωση βυτιοφόρου οχήματος 8

1.2 Αντικείμενο διπλωματικής 9

1.2.1 Συνεισφορά 9

1.3 Οργάνωση κειμένου 9

2 Σχετικές εργασίες 10

2.1 Πρόβλημα φόρτωσης container 10

2.1.1 Ορισμός προβλήματος 10

2.1.2 Περιορισμοί προβλήματος 11

2.1.3 Αλγόριθμοι επίλυσης 15

2.2 Φόρτωση φορτηγών οχημάτων για τη μεταφορά container με περιορισμούς ευστάθειας 16

2.2.1 Φόρτωση ενός container 16

2.2.2 Φόρτωση παλετών σε φορτηγά οχήματα 19

2.3 Φόρτωση βυτιοφόρων οχημάτων-Πρόβλημα φόρτωσης και δρομολόγησης βυτιοφόρων οχημάτων για τη διανομή καυσίμων 22

3 Θεωρητικό υπόβαθρο 25

3.1 Εφοδιαστική αλυσίδα υγρών καυσίμων 25

3.1.1 Διαδικασία παραλαβής καυσίμων από τα διυλιστήρια 25

3.1.2 Διαδικασία φόρτωσης των βυτιοφόρων οχημάτων από τις εγκαταστάσεις των εταιριών 25

3.1.3 Διαδικασία παράδοσης καυσίμων από τα βυτιοφόρα οχήματα προς τα πρατήρια υγρών καυσίμων 26

3.2 Περιγραφή του προβλήματος φόρτωσης βυτιοφόρου οχήματος 27

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

3.3 Μαθηματικά μοντέλα φόρτωσης βυτιοφόρου οχήματος 28
3.3.1 Ανάπτυξη του μοντέλου χωρίς μετρητές στάθμης (M1) *29*
3.3.2 Ανάπτυξη του μοντέλου με μετρητές στάθμης (M2) *32*
4 Μοντελοποίηση των περιορισμών ευστάθειας 35
4.1 Καθορισμός των παραμέτρων του βυτιοφόρου οχήματος 35
4.2 Ανάπτυξη περιορισμών ευστάθειας 39
5 Αξιολόγηση 43
5.1 Οργάνωση πειραμάτων 43
5.1.1 Σύνολο Δεδομένων *43*
5.1.2 Εύρεση διαδρομής *45*
5.1.3 Ανάπτυξη σεναρίων *46*
5.1.4 Πλατφόρμες και προγραμματιστικά εργαλεία *47*
5.2 Αποτελέσματα 48
5.2.1 Μοντέλο χωρίς μετρητές στάθμης *48*
5.2.2 Μοντέλο με μετρητές στάθμης *49*
5.2.3 Συγκεντρωτικά αποτελέσματα *51*
5.3 Σύνοψη συμπερασμάτων αξιολόγησης 59
6 Επίλογος 60
6.1 Σύνοψη και συμπεράσματα 60
6.2 Μελλοντικές επεκτάσεις 61
7 Βιβλιογραφία 62
Παράρτημα 66

ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ

ΠΙΝΑΚΑΣ 1:ΚΑΤΑΤΑΞΗ ΤΩΝ ΕΠΙΚΙΝΔΥΝΩΝ ΕΜΠΟΡΕΥΜΑΤΩΝ ΣΥΜΦΩΝΑ ΜΕ ΤΗ ΣΥΜΦΩΝΙΑ ADR 3
ΠΙΝΑΚΑΣ 2: ΑΤΥΧΗΜΑΤΑ ΚΑΤΑ ΤΗ ΜΕΤΑΦΟΡΑ ΕΠΙΚΙΝΔΥΝΩΝ ΟΥΣΙΩΝ 4
ΠΙΝΑΚΑΣ 3: ΓΕΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΒΥΤΙΟΦΟΡΟΥ ΟΧΗΜΑΤΟΣ 36
ΠΙΝΑΚΑΣ 4: ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΩΝ ΔΙΑΜΕΡΙΣΜΑΤΩΝ ΤΟΥ ΒΥΤΙΟΦΟΡΟΥ ΟΧΗΜΑΤΟΣ 36
ΠΙΝΑΚΑΣ 5: ΜΕΓΙΣΤΑ ΒΑΡΗ ΑΝΑ ΑΞΟΝΑ Η ΟΜΑΔΑ ΑΞΟΝΩΝ ΓΙΑ ΔΙΕΘΝΕΙΣ ΜΕΤΑΦΟΡΕΣ 38
ΠΙΝΑΚΑΣ 6: ΠΛΗΘΟΣ ΠΑΡΑΓΓΕΛΙΩΝ ΑΝΑ ΠΟΣΟΤΗΤΑ ΠΑΡΑΓΓΕΛΘΕΝΤΩΝ ΠΡΟΪΟΝΤΩΝ 44
ΠΙΝΑΚΑΣ 7:ΔΕΔΟΜΕΝΑ ΓΙΑ ΤΙΣ ΕΓΚΑΤΑΣΤΑΣΕΙΣ 44

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΠΙΝΑΚΑΣ 8: ΔΕΔΟΜΕΝΑ ΓΙΑ ΤΙΣ ΑΠΟΣΤΟΛΕΣ-ΠΡΑΤΗΡΙΑ (EXCEL 1) 44
ΠΙΝΑΚΑΣ 9: ΔΕΔΟΜΕΝΑ ΓΙΑ ΤΙΣ ΠΑΡΑΓΓΕΛΙΕΣ (EXCEL 1) 44
ΠΙΝΑΚΑΣ 10: ΔΕΔΟΜΕΝΑ ΓΙΑ ΤΑ ΠΑΡΑΓΓΕΛΘΕΝΤΑ ΠΡΟΪΟΝΤΑ (EXCEL 1) 45
ΠΙΝΑΚΑΣ 11: ΔΕΔΟΜΕΝΑ ΓΙΑ ΤΙΣ ΑΠΟΣΤΟΛΕΣ-ΠΡΑΤΗΡΙΑ (EXCEL 2) 45
ΠΙΝΑΚΑΣ 12: ΔΕΔΟΜΕΝΑ ΓΙΑ ΤΙΣ ΠΑΡΑΓΓΕΛΙΕΣ (EXCEL 2) 45
ΠΙΝΑΚΑΣ 13: ΔΕΔΟΜΕΝΑ ΓΙΑ ΤΑ ΠΑΡΑΓΓΕΛΘΕΝΤΑ ΠΡΟΪΟΝΤΑ (EXCEL 2) 45
ΠΙΝΑΚΑΣ 14: ΜΟΝΤΕΛΟ ΧΩΡΙΣ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ – ΣΕΝΑΡΙΟ Α 48
ΠΙΝΑΚΑΣ 15: ΜΟΝΤΕΛΟ ΧΩΡΙΣ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ – ΣΕΝΑΡΙΟ Β 48
ΠΙΝΑΚΑΣ 16: ΜΟΝΤΕΛΟ ΜΕ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ – ΣΕΝΑΡΙΟ Α 49
ΠΙΝΑΚΑΣ 17: ΜΟΝΤΕΛΟ ΜΕ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ – ΣΕΝΑΡΙΟ Β 50
ΠΙΝΑΚΑΣ 18: ΜΟΝΤΕΛΟ ΜΕ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ – ΣΕΝΑΡΙΟ C 50
ΠΙΝΑΚΑΣ 19: ΣΥΓΚΕΝΤΡΩΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΟΝΤΕΛΟΥ ΧΩΡΙΣ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ-ΣΕΝΑΡΙΟ Α 52
ΠΙΝΑΚΑΣ 20: ΣΥΓΚΕΝΤΡΩΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΟΝΤΕΛΟΥ ΧΩΡΙΣ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ-ΣΕΝΑΡΙΟ Β 53
ΠΙΝΑΚΑΣ 21: ΣΥΓΚΕΝΤΡΩΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΟΝΤΕΛΟΥ ΧΩΡΙΣ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ-ΣΥΓΚΡΙΣΗ ΣΕΝΑΡΙΩΝ 54
ΠΙΝΑΚΑΣ 22: ΣΥΓΚΕΝΤΡΩΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΟΝΤΕΛΟΥ ΜΕ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ-ΣΕΝΑΡΙΟ Α 55
ΠΙΝΑΚΑΣ 23: ΣΥΓΚΕΝΤΡΩΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΟΝΤΕΛΟΥ ΜΕ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ-ΣΕΝΑΡΙΟ Β 56
ΠΙΝΑΚΑΣ 24: ΣΥΓΚΕΝΤΡΩΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΟΝΤΕΛΟΥ ΜΕ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ-ΣΕΝΑΡΙΟ C 57
ΠΙΝΑΚΑΣ 25: ΣΥΓΚΕΝΤΡΩΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΟΝΤΕΛΟΥ ΜΕ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ-ΣΥΓΚΡΙΣΗ ΣΕΝΑΡΙΩΝ 58

ΠΙΝΑΚΑΣ ΔΙΑΓΡΑΜΜΑΤΩΝ

ΔΙΑΓΡΑΜΜΑ 1: ΑΝΤΙΚΕΙΜΕΝΙΚΗ ΣΥΝΑΡΤΗΣΗ - ΠΟΣΟΣΤΟ ΑΠΟΚΛΙΣΗΣ ΤΟΥ ΚΕΝΤΡΟΥ ΒΑΡΟΥΣ ΤΟΥ ΦΟΡΤΙΟΥ ΑΠΟ ΤΟ ΓΕΩΜΕΤΡΙΚΟ ΚΕΝΤΡΟ ΒΑΡΟΥΣ ΤΟΥ ΟΧΗΜΑΤΟΣ – ΣΕΝΑΡΙΟ C 50

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

ΕΙΚΟΝΑ 1: ΤΡΙΑΞΟΝΙΚΟ ΒΥΤΙΟΦΟΡΟ ΟΧΗΜΑ 39

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

1

Εισαγωγή

Η φόρτωση των φορτηγών οχημάτων είναι αποτελεσματική όταν γίνεται με γνώμονα τη βέλτιστη αξιοποίηση της χωρητικότητας τους καθώς και τη διασφάλιση της ασφαλούς μεταφοράς των εμπορευμάτων. Πιο συγκεκριμένα, στην περίπτωση της μεταφοράς των επικίνδυνων εμπορευμάτων μέσω βυτιοφόρων οχημάτων, η οδική ασφάλεια καθίσταται καίριας σημασίας για την αποφυγή ατυχημάτων τα οποία έχουν δυσμενείς επιπτώσεις για τον ανθρώπινο παράγοντα, το κοινωνικό και το φυσικό περιβάλλον. Η εξασφάλιση της ευστάθειας κατά τη φόρτωση φορτηγών οχημάτων για λόγους ασφαλείας αποτελεί εξαιρετική πρόκληση για τις μεταφορικές υπηρεσίες. Για τα βυτιοφόρα οχήματα, η ευστάθεια επιτυγχάνεται μέσω του κανόνα ταλάντωσης του υγρού καυσίμου και εμπειρικών κανόνων. Καθίσταται, συνεπώς αναγκαίο οι περιορισμοί της ευστάθειας, δηλαδή η μη υπέρβαση των μέγιστων επιτρεπόμενων ορίων βάρους κατ' άξονα και η ομοιόμορφη κατανομή του βάρους του φορτίου, να ενταχθούν και με αναλυτικό τρόπο στη φάση του σχεδιασμού της φόρτωσης ενός βυτιοφόρου οχήματος και να τηρούνται για κάθε φάση της διαδρομής του, ώστε να ελαχιστοποιείται η πιθανότητα ανατροπής του οχήματος ή η δημιουργία ατυχήματος λόγω κακού χειρισμού. Επομένως, η παρούσα διπλωματική εργασία στοχεύει στη μοντελοποίηση των περιορισμών ευστάθειας και την ενσωμάτωσή τους σε μοντέλα φόρτωσης καθώς και στη διερεύνηση trade-offs ανάμεσα στη βέλτιστη φόρτωση και την ικανοποίηση των περιορισμών αυτών.

1.1 Φόρτωση φορτηγών οχημάτων για τη διανομή προϊόντων

Η αποτελεσματική φόρτωση των φορτηγών οχημάτων για την καθημερινή διανομή προϊόντων είναι καίριας σημασίας για την εύρυθμη λειτουργία και την οικονομικότερη εξέλιξη της εφοδιαστικής αλυσίδας. Η βέλτιστη αξιοποίηση του διαθέσιμου χώρου των φορτηγών αποτελεί ένα πολύ σημαντικό

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

πεδίο έρευνας καθώς η παγκόσμια οικονομία και οι μεταφορές γίνονται ολοένα και πιο ανταγωνιστικές. Η εύρεση αποτελεσματικών και εύχρηστων μεθόδων υπολογισμού της βέλτιστης φόρτωσης καθίσταται επιβεβλημένη, καθώς συμβάλει στη μείωση του κόστους και του χρόνου μεταφοράς, την αύξηση των κερδών, τη βελτίωση του επιπέδου εξυπηρέτησης των πελάτων και την προστασία του περιβάλλοντος (μείωση κατανάλωσης καυσίμων και εκπομπών CO2). Συνεπώς, η επίλυση του προβλήματος φόρτωσης ενός φορτηγού οχήματος είναι μείζονος σημασίας για τον τομέα των μεταφορών και των logistics και αντιμετωπίζεται από την πλειοψηφία των κέντρων διανομής και των αποθηκών, για αυτό και έχει μελετηθεί διεξοδικά και με πολλές προσεγγίσεις από τη βιβλιογραφία.

1.1.1 Επικίνδυνα φορτία- Συμφωνία ADR

Μία ειδική κατηγορία εμπορευμάτων τα οποία μεταφέρονται μέσω φορτηγών οχημάτων είναι τα επικίνδυνα φορτία, που αποτελούνται από εμπορεύματα, είδη και υλικά, η μεταφορά των οποίων ενέχει κινδύνους για το περιβάλλον, τη δημόσια τάξη και την ασφάλεια. Για τα φορτία αυτά θεσπίστηκε το 1957 η Ευρωπαϊκή Συμφωνία για την Διεθνή Οδική Μεταφορά Επικίνδυνων Εμπορευμάτων ADR (Accord Dangereux Routier). Η συμφωνία ADR επικυρώθηκε από την Ελλάδα στις 27/05/1987 με τον νόμο 1741/1987 και από τότε αποτελεί νόμο του ελληνικού κράτους και ρυθμίζει τις οδικές μεταφορές επικίνδυνων εμπορευμάτων. Η Συμφωνία ADR επικαιροποιείται κάθε δύο χρόνια και η ισχύουσα έκδοση είναι η ADR 2017, ενώ η χώρα μας προσαρμόστηκε στην νέα αυτή έκδοση τον Αύγουστο του 2017 (ΦΕΚ/Β'/2915/24-8-2017). Σκοπός της συμφωνίας είναι η διασφάλιση της ασφαλούς μεταφοράς των επικίνδυνων εμπορευμάτων και εάν οι μεταφορές πληρούν τις απαιτήσεις της ADR, τότε απαλλάσσονται από την τήρηση της εθνικής νομοθεσίας των συμβαλλόμενων κρατών.

Η συμφωνία ADR εμπεριέχει την αναλυτική ταξινόμηση των επικίνδυνων φορτίων σε κλάσεις και υποδιαιρέσεις αυτών, όπως διαφαίνεται στον Πίνακα 1.

Κλάσεις	Επικίνδυνα Εμπορεύματα	Ενδεικτικά Παραδείγματα
Κλάση 1	Εκρηκτικές ύλες	
Κλάση 1.1	Εκρηκτικές ύλες και είδη	βαβμακοπυρίτιδα, φυσίγγια πεπιεσμένης μελανής πυρίτιδας
Κλάση 1.2	Είδη γεμισμένα με εκρηκτικές ύλες	εκρηκτικά φυσίγγια, πυροσωλήνες κρουσιφλεγή καψύλλια
Κλάση 1.3	Αναφλεκτήρες, πυροτεχνήματα & παρόμοια Εμπορεύματα	πυρεία ασφαλείας, σωληνωτοί αναφλεκτήρες, πυροτεχνικά είδη, καπνογόνες ύλες
Κλάση 2	Αέρια πεπιεσμένα, υγροποιημένα ή διαλυόμενα υπό Πίεση	άζωτο, αργό, οξυγόνο, υδρογόνο, μεθάνιο
Κλάση 3	Εύφλεκτα υγρά	βενζίνη, πετρέλαια, ορυκτά έλαια, προϊόντα συμπυκνώσεως φυσικού αερίου
Κλάση 4	Εύφλεκτα στερεά ή ουσίες υποκείμενες σε αυτόματη καύση	
Κλάση 4.1	Εύφλεκτα στερεά	ξυλάνθρακας, θείον, καουτσούκ, λιγνίτης, γαιάνθρακας, ναφθαλίνη
Κλάση 4.2	Εύφλεκτα στερεά ή ύλες υποκείμενες σε αυτόματη Καύση	λευκός και κίτρινος φώσφορος, ενώσεις φωσφόρου με διάφορα μέταλλα
Κλάση 4.3	Εύφλεκτα στερεά ή ουσίες που ελκύουν εύφλεκτα αέρια σε επαφή με το νερό	αλκαλομέταλλα (π.χ. κάλιο, νάτριο, ασβέστιο), πυριτιούχο ασβέστιο σε σκόνη
Κλάση 5	Οξειδωτικές ουσίες και οργανικά υπεροξείδια	
Κλάση 5.1	Οξειδωτικές Ύλες	υδατικά διαλύματα υπεροξειδίου του υδρογόνου, υπερχλωρικό οξύ
Κλάση 5.2	Οργανικά Υπεροξείδια	υπεροξείδιο διτριτογενούς βουτυλίου
Κλάση 6	Δηλητηριώδεις (τοξικές) και μολυσματικές ύλες	
Κλάση 6.1	Τοξικές Ύλες	υδροκυανικό οξύ, νιτρίλια, νιτρογενείς οργανικές ενώσεις
Κλάση 6.2	Απεχθείς ύλες και ύλες που μπορούν να προκαλέσουν μόλυνση	ύλες ζώων
Κλάση 7	Ραδιενεργές ύλες	
Κλάση 8	Διαβρωτικές ύλες	θειικό οξύ, νιτρικά οξέα, διαλύματα υδροχλωρικού οξέος
Κλάση 9	Διάφορες επικίνδυνες ύλες και είδη	άσβεστος PCB"S (κλοφέν), συσκευές που περιέχουν PCB"S (μετασχηματιστές)

Πίνακας 1:Κατάταξη των επικίνδυνων εμπορευμάτων σύμφωνα με τη Συμφωνία ADR

1.1.2 Ατυχήματα κατά τη μεταφορά επικίνδυνων φορτίων

Καθώς η μεταφορά των επικίνδυνων εμπορευμάτων αποτελεί μια από τις πιο σημαντικές οικονομικές δραστηριότητες στις βιομηχανικά ανεπτυγμένες χώρες, η ασφαλής μεταφορά τους καθίσταται καίριας σημασίας προκειμένου να αποφευχθούν ατυχήματα, τα οποία έχουν δυσμενείς επιπτώσεις για τον ανθρώπινο παράγοντα (τραυματισμοί/θάνατοι), το κοινωνικό και το φυσικό περιβάλλον, με περιβαλλοντικές καταστροφές και υλικές ζημιές υψηλού κόστους. Παράλληλα, εκτός από τις προαναφερθείσες δυσμενείς επιπτώσεις, η ύπαρξη ενός ατυχήματος οδηγεί σε διάσπαση της ροής της εφοδιαστικής αλυσίδας και κατ' επέκταση σε καθυστερήσεις στη διανομή των φορτίων. Αν και ο αριθμός των ατυχημάτων αυτών είναι σχετικά μικρός, συγκρινόμενος με ατυχήματα στα οποία εμπλέκονται άλλοι τύποι οχημάτων, υπάρχει μία έντονη ευαισθησία του κοινού σχετικά με τους κινδύνους που συνεπάγεται η μεταφορά επικίνδυνων υλών.

Η Ευρωπαϊκή Κοινότητα και η Ελληνική Νομοθεσία θα πρέπει να διασφαλίζουν, με τη λήψη αυξημένων μέτρων ασφαλείας, ότι αυτά τα εμπορεύματα μεταφέρονται με τη μεγαλύτερη δυνατή ασφάλεια, το μικρότερο δυνατό κόστος, ελαχιστοποιώντας του κινδύνους για τους ανθρώπους και το περιβάλλον.

Οι επικίνδυνες ουσίες που εμπλέκονται στην πλειονότητα των ατυχημάτων είναι τα εύφλεκτα υγρά, οι διαβρωτικές ουσίες, οι οξειδωτικές ουσίες και τα αέρια, ενώ οι συνήθεις τύποι των ατυχημάτων

σχετίζονται με την πρόκληση έκρηξης, φωτιάς, τραυματισμού, θανάτου, μόλυνση, εισροή σε υγρό στοιχείο, διαρροή καυσίμου, εξάπλωση ραδιενέργειας και διασπορά αέριου νέφους.

Στον Πίνακα 2 αναφέρονται ενδεικτικά ατυχήματα κατά τη μεταφορά επικίνδυνων ουσιών με βυτιοφόρα οχήματα.

Χρονολογία	Τόπος	Ατύχημα	Συνέπεια
25/10/1994	Ασπρόπυργος	Βυτιοφόρο όχημα μεταφοράς κηροζίνης ανετράπη.	Θαλάσσια ρύπανση μέσω αγωγών ομβρίων υδάτων.
20/3/1995	23^{ο} χλμ. Θες/νίκης-Κιλκίς	Βυτιοφόρο όχημα μεταφοράς βενζίνης ανετράπη και πήρε φωτιά.	Θάνατος του οδηγού.
3/7/1995	Ανατολικό Θες/νίκης	Βυτιοφόρο όχημα μεταφοράς θειικού οξέος ανετράπη.	Διαρροή.
22/6/2002	Tivissa, Catalonia	Οδηγός οχήματος (19 τόνοι φυσικό αέριο) έχασε τον έλεγχο εξαιτίας υψηλής ταχύτητας, με αποτέλεσμα να αναποδογυρίσει το όχημα. Πραγματοποιήθηκαν δύο συνεχόμενες εκρήξεις και φωτιά.	Θάνατος οδηγού, τραυματισμός δύο ανθρώπων λόγω εγκαυμάτων και περιβαλλοντικές επιπτώσεις.
27/8/2012	Kerala,Ινδία	Βυτιοφόρο όχημα (16 τόνους υγραέριο) έπεσε στο μεσαίο διαχωριστικό του δρόμου στην προσπάθεια προσπέρασης άλλου οχήματος. Έγινε διαρροή υγραερίου, πρόκληση φωτιάς και τρείς συνεχόμενες εκρήξεις.	21 τραυματίες, 40 θάνατοι, συνέπειες στη χλωρίδα, την πανίδα και σε κτήρια της κατοικημένης περιοχής.
21/7/2014	Lice, Τουρκία	Βυτιοφόρο όχημα μεταφοράς υγραερίου εκτράπηκε από την πορεία του και ανετράπη. Προκλήθηκε διαρροή υγραερίου και έκρηξη.	1 ακαριαίος θάνατος, 34 θάνατοι επιβατών εντός τριμήνου και 34 τραυματισμών λόγω εκτεταμένων εγκαυμάτων.
30/8/2021	Λεωφόρος Δραπετσώνας	Βυτιοφόρο όχημα μεταφοράς καυσίμων ανετράπη μετά από πρόσκρουση στο διαχωριστικό κυκλίδωμα.	Τραυματισμός του οδηγού και διαρροή καυσίμου στο οδόστρωμα.
9/11/2021	Εθνική Οδός Αθηνών-Κορίνθου	Βυτιοφόρο όχημα μεταφοράς κηροζίνης εκτράπηκε και ανετράπη.	Διαρροή στο οδόστρωμα του εύφλεκτου υλικού.

Πίνακας 2: Ατυχήματα κατά τη μεταφορά επικίνδυνων ουσιών

1.1.3 Χερσαία μεταφορά επικίνδυνων φορτίων μέσω βυτιοφόρων οχημάτων

Τα επικίνδυνα φορτία μπορούν να μεταφέρονται με δεξαμενές-βυτία, μόνο αν ο τρόπος αυτός της μεταφοράς επιτρέπεται για τα συγκεκριμένα φορτία από τις αντίστοιχες διατάξεις της Συμφωνίας ADR και υπό την πρόσθετη προϋπόθεση ότι πληρούνται οι ειδικές απαιτήσεις που προβλέπονται για τις μεταφορές αυτές. Συνεπώς, οι επικίνδυνες ύλες μπορούν να μεταφέρονται με δεξαμενές μέσω βυτιοφόρων οχημάτων, τα οποία είναι εξοπλισμένα με ενσωματωμένα βυτία και διακρίνονται σε απλά βυτιοφόρα, οδικούς συρμούς αποτελούμενους από ρυμουλκό βυτιοφόρο και ρυμουλκούμενο βυτιοφόρο και επικαθήμενα βυτιοφόρα. Ανάλογα με τις ύλες που μεταφέρουν, τα οχήματα διακρίνονται σε βυτιοφόρα με δεξαμενές για μεταφορά αεριών, με δεξαμενές για μεταφορά πετρελαιοειδών και διαφόρων χημικών υλών και με βυτία για μεταφορά στερεών υλών, σε μορφή σκόνης ή κόκκων.

Ειδικότερα, αναφορικά με τα βυτιοφόρα οχήματα μεταφοράς υγρών καυσίμων (πχ βενζίνη, προϊόντα πετρελαίου-αργού, θερμάνσεως, ντίζελ), αυτά έχουν συγκεκριμένη μέγιστη ικανότητα φόρτωσης και αποτελούνται από βυτία πολλαπλών προϊόντων. Τα βυτία αυτά έχουν πολλαπλά διαμερίσματα (compartments), τα οποία δεν επικοινωνούν μεταξύ τους, έχοντας τη δυνατότητα να μεταφέρουν διαφορετικούς τύπους καυσίμου, χωρίς να υπάρχει κίνδυνος αναμίξεώς τους, ακόμα και κατά τη διαδικασία της εκφόρτωσης.

Συνήθως, τα βυτιοφόρα οχήματα διαθέτουν μετρητή στάθμης (flow meter), ο οποίος είναι απαραίτητος για να ελέγχεται η ύπαρξη του αναγκαίου κενού χώρου εντός του βυτίου, που θα καλύπτει τη θερμική διαστολή του περιεχομένου του και την υπέρβαση του μέγιστου επιτρεπόμενου βάρους του βυτίου.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Στην περίπτωση που ένα βυτιοφόρο όχημα δεν είναι εξοπλισμένο με μετρητές στάθμης, το φορτίο που υπάρχει μέσα σε ένα διαμέρισμα πρέπει να εκκενώνεται πλήρως κατά την παράδοσή του στο πρατήριο. Συνεπώς, σε κάθε διαμέρισμά του οχήματος μπορεί να ανατεθεί η ποσότητα το πολύ ενός προϊόντος για ένα πελάτη και δεν υπάρχει η δυνατότητα να διαχωριστεί σε διαφορετικούς πελάτες. Αντίθετα, στην περίπτωση που ένα βυτιοφόρο όχημα διαθέτει μετρητές στάθμης μπορούν να ανατεθούν πολλαπλές ποσότητες ενός προϊόντος που ανήκουν σε διαφορετικούς πελάτες στο ίδιο διαμέρισμα.

1.1.4 Οδική συμπεριφορά βυτιοφόρων οχημάτων

Στις παρακάτω υποενότητες αναφέρονται οι απαιτήσεις και πρακτικές που αφορούν την οδική συμπεριφορά, την ευστάθεια και τη φορτοεκφόρτωση των βυτιοφόρων οχημάτων σύμφωνα με το Μέρος Β' της Συμφωνίας ADR, το οποίο αναφέρεται στην Οδική Μεταφορά με Βυτιοφόρα Οχήματα.

Η οδική συμπεριφορά των βυτιοφόρων οχημάτων είναι διαφορετική από τη συμπεριφορά άλλων κατηγοριών οχημάτων, καθώς επιβαρύνεται από τη σχετική κίνηση του υγρού μέσα στη δεξαμενή κατά την εκκίνηση, την επιτάχυνση, την επιβράδυνση ή την αλλαγή πορείας (στροφές) του οχήματος.

Η σχετική κίνηση του υγρού επηρεάζει τη δυναμική συμπεριφορά του οχήματος και μπορεί να το οδηγήσει σε αστάθεια, δημιουργώντας τα ακόλουθα δυσμενή αποτελέσματα: αυξάνεται η απόσταση πέδησης και δημιουργείται ταλάντωση του οχήματος, επειδή το μεταφερόμενο υγρό, μετά την ολοκλήρωση του φρεναρίσματος και την πρόσκρουσή του στο εμπρόσθιο τοίχωμα του βυτίου, γυρίζει προς τα πίσω και έτσι αρχίζει μια παλινδρομική κίνηση του υγρού μέσα στο βυτίο. Σε περίπτωση επιτάχυνσης παρατηρείται το ίδιο φαινόμενο.

Ο πιο συχνός κίνδυνος κατά την κίνηση ενός βυτιοφόρου οχήματος είναι η ανατροπή. Πιο συγκεκριμένα, σε περίπτωση αλλαγής της κατεύθυνσης του οχήματος, δηλαδή στις στροφές, το υγρό κινείται προς την έξω πλευρά της στροφής, πιέζοντας το αντίστοιχο πλευρικό τοίχωμα της δεξαμενής. Στις περιπτώσεις αυτές, η φυγόκεντρος δύναμη επενεργεί πάνω στο όχημα, με αποτέλεσμα να μετατοπίζεται το κέντρο βάρους του βυτιοφόρου οχήματος προς την εξωτερική πλευρά της στροφής και, αν η ταχύτητα του οχήματος είναι μεγάλη ή η στροφή απότομη, παρουσιάζεται κίνδυνος ανατροπής. Ταυτόχρονα η φυγόκεντρος επιδρά και στο περιεχόμενο του βυτίου μετατοπίζοντας και αυτό προς το έξω μέρος της στροφής και αναπτύσσεται κυματισμός του υγρού που μπορεί να οδηγήσει σε επικίνδυνη ταλάντωση του οχήματος.

1.1.5 Ευστάθεια βυτιοφόρου οχήματος- Βαθμός πλήρωσης

Όπως αναφέρθηκε η ευστάθεια ενός βυτιοφόρου οχήματος επηρεάζεται από τις κινήσεις του υγρού μέσα στο βυτίο. Κατά την κίνηση ενός βυτιοφόρου οχήματος, η δύναμη που ασκείται από το υγρό

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

φορτίο στο όχημα αυξάνεται αναλογικά με τη μάζα του φορτίου. Για να αποφεύγονται οι μεγάλες μετακινήσεις υγρού μέσα στο βυτίο, τα βυτία χωρίζονται σε διαμερίσματα με ενδιάμεσα τοιχώματα. Όταν το βυτίο είναι (σχεδόν) γεμάτο, το υγρό δεν έχει χώρο να μετακινηθεί, συνεπώς η δύναμη που ασκεί είναι περιορισμένη.

Επιπλέον, όταν το βυτίο είναι (σχεδόν) άδειο, η μάζα του υγρού είναι πολύ μικρή για να αναπτύξει μεγάλες δυνάμεις ικανές να ανατρέψουν το όχημα. Όταν το βυτίο είναι μισογεμάτο, τότε η μάζα του υγρού είναι υπολογίσιμη και υπάρχει ελεύθερος χώρος για να κινηθεί. Σε αυτήν την περίπτωση αναπτύσσονται οι ισχυρότερες δυνάμεις αυξάνοντας τον κίνδυνο ανατροπής. Η πιο κρίσιμη ποσότητα είναι όταν πληρούται το 79% περίπου του χώρου του βυτίου (Συμφωνία ADR 2011).

Τα βυτιοφόρα οχήματα μεταφοράς υγρών καυσίμων διαθέτουν έναν μέγιστο και έναν ελάχιστο βαθμό πληρώσεως, ο οποίος πρέπει να τηρείται. Ο μέγιστος βαθμός εξαρτάται από το μεταφερόμενο προϊόν και κυμαίνεται από 90-97% του συνολικού όγκου του διαμερίσματος με το αντίστοιχο ελάχιστο περιθώριο του βυτίου να ποικίλλει από 3-10%. Βάσει της συμφωνίας ADR, ο βαθμός πλήρωσης διαμερίσματος δεξαμενής χωρητικότητας μεγαλύτερης των 7.500 λίτρων πρέπει να είναι μεγαλύτερος από το 80% ή μικρότερος από το 20% του συνολικού όγκου του διαμερίσματος. Ο κανόνας αυτός καλείται κανόνας της ταλάντωσης (διακύμανσης) του υγρού καυσίμου της αξιοποίησης του 20%/80% της χωρητικότητας ενός διαμερίσματος- fuel oscillation rule. Εάν ένα διαμέρισμα έχει χωρητικότητα μέχρι 7.500 λίτρα, τότε δεν υπάρχει ελάχιστος βαθμός πληρώσεως. Ωστόσο, υπάρχει ο κίνδυνος να συντονιστούν πολλά μικρά διαμερίσματα (με όγκο μικρότερο από 7.500 λίτρα) και συνεπώς υπάρχει η σύσταση ο κανόνας της ταλάντωσης του υγρού καυσίμου να εφαρμόζεται σε όλα τα διαμερίσματα ανεξαρτήτως όγκου.

1.1.6 Φορτοεκφόρτωση βυτιοφόρων οχημάτων

Κατά τη φόρτωση ενός βυτιοφόρου οχήματος πρέπει να δοθεί ιδιαίτερη προσοχή στις εξής περιπτώσεις: 1.όταν το κέντρο βάρους του οχήματος είναι πολύ πίσω, υπάρχει κίνδυνος εκτροπής του πίσω μέρους του στις στροφές, 2.όταν είναι φορτωμένα μόνο τα πίσω διαμερίσματα, η φόρτιση του εμπρόσθιου άξονα του οχήματος είναι ελλιπής, δυσχεραίνοντας την οδική του συμπεριφορά ενώ, 3. όταν είναι φορτωμένα μόνο τα μπροστινά διαμερίσματα, η κινητήρια δύναμη δεν μεταφέρεται ικανοποιητικά στον πίσω άξονα.

Κατά την εκφόρτωση ενός βυτιοφόρου οχήματος που είναι χωρισμένο σε στεγανά διαμερίσματα, τα οποία δεν εκφορτώνονται συγχρόνως, πρέπει ο κινητήριος και ο διευθυντήριος άξονας να παραμένουν με αρκετό φορτίο. Όταν είναι γεμάτο μόνο το πίσω διαμέρισμα, αυξάνεται ο κίνδυνος ανατροπής σε στροφή, επειδή οι πλευρικές (αδρανειακές) δυνάμεις είναι μεγαλύτερες στο πιο βαρύ σημείο του οχήματος. Συνεπώς, πρέπει να ακολουθούνται κανόνες για τη διαδοχή εκφόρτωσης κατά τις τμηματικές

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

παραδόσεις: πρώτα πραγματοποιείται η εκφόρτωση του μεσαίου διαμερίσματος (για να είναι επαρκής η φόρτιση στο διευθυντήριο και στον κινητήριο άξονα), έπειτα η εκφόρτωση του πίσω διαμερίσματος (για να αποφευχθεί η ανατροπή) και τέλος η εκφόρτωση του μπροστινού διαμερίσματος. Εν γένει, πρέπει να τηρείται η σωστή σειρά φόρτωσης και εκφόρτωσης των πολυδιαμερισματικών βυτίων και η σωστή σειρά εκφόρτωσης σε περίπτωση τμηματικών παραδόσεων.

Επίσης για την ασφαλέστερη κίνηση ενός βυτιοφόρου οχήματος, πρέπει να εφαρμοσθούν οι ακόλουθοι γενικοί κανόνες, σύμφωνα με τη συμφωνία ADR:

- ✓ Τα βυτία δεν πρέπει να είναι μισογεμάτα. Ο ελάχιστος βαθμός πλήρωσης διαμερίσματος δεξαμενής χωρητικότητας μεγαλύτερης των 7.500 λίτρων πρέπει να είναι μεγαλύτερος από το 80% ή μικρότερος από το 20% του συνολικού όγκου του διαμερίσματος.
- ✓ Έλεγχος του συνολικού βάρους του οχήματος χωρίς σε καμία περίπτωση να υπερβαίνεται.
- ✓ Η κατανομή του βάρους να γίνεται με τέτοιο τρόπο ώστε αφενός το φορτίο να είναι ισομερώς κατανεμημένο και αφετέρου να μην υπερβαίνει το μέγιστο επιτρεπόμενο μικτό βάρος ανά άξονα.
- ✓ Το κέντρο βάρους του φορτίου να είναι ορθά κατανεμημένο και πλησιέστερα προς το μέσον του διαμήκη και του εγκάρσιου άξονα του οχήματος καθώς και το δυνατόν χαμηλότερα.
- ✓ Οδήγηση του οχήματος με μικρή ταχύτητα και ιδιαίτερα στις στροφές.
- ✓ Αποφυγή των αποτόμων χειρισμών (απότομο φρενάρισμα ή απότομη εκκίνηση - επιτάχυνση).
- ✓ Φρενάρισμα πριν από τις στροφές και όχι επάνω σε αυτές.
- ✓ Τα επικαθήμενα και τα ρυμουλκούμενα βυτία ανατρέπονται πιο εύκολα από τα αυτοκινούμενα βυτιοφόρα.
- ✓ Στα ημιρυμουλκούμενα πρέπει να ληφθεί υπόψη ότι ο κίνδυνος ανατροπής σε στροφή είναι μεγαλύτερος, όταν είναι γεμάτα μόνο τα εμπρός διαμερίσματα, διότι το κέντρο βάρους βρίσκεται πολύ κοντά στη γραμμή ανατροπής.
- ✓ Τήρησης της σωστής σειράς φόρτωσης και εκφόρτωσης των πολυδιαμερισματικών βυτίων.
- ✓ Τήρηση της σωστής σειράς εκφόρτωσης σε περίπτωση τμηματικών παραδόσεων.

1.1.7 Μέγιστα επιτρεπόμενα βάρη έμφορτου οχήματος και βάρη ανά άξονα

Σύμφωνα με το νομοθετικό πλαίσιο, τόσο για τις εθνικές όσο και για τις διεθνείς μεταφορές, ορίζονται μέγιστα επιτρεπόμενα βάρη έμφορτου οχήματος και μέγιστα επιτρεπόμενα βάρη ανά άξονα. Το βάρος του έμφορτου οχήματος φέρεται από τους άξονες. Οι τροχοί μεταφέρουν αυτό το βάρος στο οδόστρωμα μέσω της επιφάνειας επαφής των ελαστικών στο έδαφος. Τα φορτηγά οχήματα τα οποία παραβιάζουν το μέγιστο επιτρεπόμενο βάρος ή το μέγιστο επιτρεπόμενο βάρος κατ' άξονα αποτελούν σημαντική απειλή για την οδική ασφάλεια και μπορεί να προκαλέσουν βλάβες στην επιφάνεια του οδοστρώματος,

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ενώ παράλληλα οι μεταφορείς υπόκεινται σε πρόστιμα σε περίπτωση παραβίασης των επιτρεπόμενων ορίων.

Σε ορισμένους αυτοκινητοδρόμους υπάρχουν εγκατεστημένα συστήματα ζύγισης των βαρέων οχημάτων εν κινήσει, τα οποία μετρούν μέσω αισθητήρων το βάρος του κάθε άξονα και το συνολικό βάρος του οχήματος και το συγκρίνουν απ' ευθείας με τα επιτρεπτά όρια. Για τα βυτιοφόρα φορτηγά μεταφοράς υγρών καυσίμων, προκειμένου να διεξαχθεί ο έλεγχος για την μη υπέρβαση του μέγιστου συνολικού επιτρεπόμενου μικτού βάρους, το βάρος του φορτίου υπολογίζεται ως το γινόμενο των λίτρων που αναγράφονται στα φορτωτικά έγγραφα (δελτίο αποστολής/φορτωτική) με το ειδικό βάρος του μεταφερόμενου είδους (ενδεικτικά για το πετρέλαιο 0.80 και για τη βενζίνη 0.75).

1.1.8 Φόρτωση βυτιοφόρου οχήματος

Κατά τη διαδικασία βέλτιστης φόρτωσης ενός βυτιοφόρου οχήματος, το οποίο εκτελεί διανομές καυσίμων με πολλαπλές στάσεις πρέπει να τηρούνται κάποιοι βασικοί περιορισμοί όπως η μη υπέρβαση της χωρητικότητας ενός διαμερίσματος, η μη υπέρβαση του μέγιστου επιτρεπόμενου βάρους του οχήματος, η ανάθεση ενός μόνο τύπου καυσίμου σε ένα διαμέρισμα και η τήρηση του κανόνα ταλάντωσης (διακύμανσης) υγρού καυσίμου της αξιοποίησης του 20%/80% της χωρητικότητας κάθε διαμερίσματος του φορτηγού.

Λαμβάνοντας υπόψιν την επισκόπηση που πραγματοποιήθηκε αναφορικά με τη νομοθεσία, την οδική συμπεριφορά και την ευστάθεια του βυτιοφόρου οχήματος, προκύπτει ότι υπάρχουν κάποιοι περιορισμοί οι οποίοι λαμβάνονται υπόψιν με πρακτικούς τρόπους, αλλά πρέπει να ενταχθούν και με αναλυτικό τρόπο στη φάση του σχεδιασμού της φόρτωσης ενός βυτιοφόρου οχήματος, ώστε να ελαχιστοποιείται η πιθανότητα ανατροπής του οχήματος ή η δημιουργία ατυχήματος λόγω κακού χειρισμού του οχήματος.

Η εξασφάλιση της ευστάθειας κατά τη φόρτωση φορτηγών οχημάτων για λόγους ασφαλείας αποτελεί εξαιρετική πρόκληση για τις μεταφορικές υπηρεσίες. Η ευστάθεια/ισορροπία (load balance) του φορτίου αναφέρεται στην κατανομή του βάρους του φορτίου ανάμεσα στους άξονες. Το φορτηγό όχημα απαιτεί ομοιόμορφη κατανομή του βάρους του φορτίου ανάμεσα στους άξονές του ώστε να διασφαλίζεται η μεταφορά των εμπορευμάτων υπό ασφαλείς συνθήκες. Η κατανομή αυτή επηρεάζεται από το βάρος του φορτίου και τη θέση του κέντρου βάρους του.

Πιο συγκεκριμένα, ο κανόνας που ορίζει την ευστάθεια είναι η μη υπέρβαση των μέγιστων επιτρεπόμενων ορίων βάρους κατ' άξονα του οχήματος, τα οποία ορίζονται από το νομοθετικό πλαίσιο. Παράλληλα, ένας επιπλέον κανόνας που δημιουργεί επιπρόσθετο επίπεδο οδικής ασφάλειας, ώστε να αποφευχθεί η ανατροπή του οχήματος είναι η ομοιόμορφη κατανομή του βάρους του φορτίου ώστε το

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

κέντρο βάρους του να βρίσκεται όσο το δυνατόν πλησιέστερα στο γεωμετρικό κέντρο βάρους του οχήματος. Σκοπός της σωστής κατανομής του φορτίου είναι η επίτευξη της μέγιστης δυνατής ευστάθειας κατά την κίνηση του οχήματος (φρενάρισμα, επιτάχυνση, στροφές ή αλλαγή πορείας) και της αποφυγής μετακίνησης του φορτίου κατά τη μεταφορά. Για τη σωστή κατανομή του φορτίου πρέπει να λαμβάνεται, σε κάθε περίπτωση, υπόψιν η φόρτιση των αξόνων του οχήματος. Οι παραπάνω κανόνες πρέπει να τηρούνται σε κάθε φάση της διαδρομής και όχι μόνο κατά την αρχική φόρτωση.

1.2 Αντικείμενο διπλωματικής

Αντικείμενο της παρούσας διπλωματικής εργασίας αποτελεί η ανάπτυξη μοντέλων για την επίλυση του προβλήματος φόρτωσης ενός βυτιοφόρου οχήματος διανομής καυσίμων με πολλαπλές στάσεις (με ή χωρίς μετρητές στάθμης) λαμβάνοντας υπόψιν περιορισμούς ευστάθειας για κάθε φάση της διαδρομής, το οποίο δεν έχει μελετηθεί στη βιβλιογραφία, καθώς και η σύγκριση των αποτελεσμάτων των μοντέλων αυτών με τα μοντέλα που δε λαμβάνουν υπόψιν τους περιορισμούς ευστάθειας.

1.2.1 Συνεισφορά

Η συνεισφορά της Διπλωματικής Εργασίας συνοψίζεται ως εξής:

1. Μοντελοποίηση και ενσωμάτωση των περιορισμών ευστάθειας σε μοντέλα μεικτού ακέραιου γραμμικού προγραμματισμού για τη φόρτωση βυτιοφόρου οχήματος με ή χωρίς μετρητές στάθμης.
2. Διερεύνηση trade-offs αναφορικά με την ποσότητα φόρτωσης, τις αποκλίσεις μεταξύ της παραγγελθείσας και της παραδοθείσας ποσότητας και του ποσοστού αξιοποίησης της χωρητικότητας του βυτιοφόρου οχήματος, μεταξύ των μοντέλων που λαμβάνουν υπόψιν τους περιορισμούς ευστάθειας και των μοντέλων που δεν τους λαμβάνουν.

1.3 Οργάνωση κειμένου

Εργασίες σχετικές με το αντικείμενο της διπλωματικής παρουσιάζονται στο Κεφάλαιο 2. Το Κεφάλαιο 3 αναφέρεται στο θεωρητικό υπόβαθρο, την περιγραφή του προβλήματος φόρτωσης βυτιοφόρου οχήματος και τα μαθηματικά μοντέλα φόρτωσης. Στο Κεφάλαιο 4 πραγματοποιείται η μοντελοποίηση των περιορισμών ευστάθειας. Στο κεφάλαιο 5 παρουσιάζεται η ανάλυση των πειραμάτων, αναφορικά με το σύνολο των δεδομένων, την ανάπτυξη των σεναρίων και τα προγραμματιστικά εργαλεία που χρησιμοποιούνται, ενώ αναλύονται τα συνολικά αποτελέσματα. Στο κεφάλαιο 6 πραγματοποιείται η σύνοψη των συμπερασμάτων και παρουσιάζονται μελλοντικές επεκτάσεις της Διπλωματικής εργασίας.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

2

Σχετικές εργασίες

Το πρόβλημα φόρτωσης φορτηγών οχημάτων έχει μελετηθεί διεξοδικά και με πολλές προσεγγίσεις από τη διεθνή βιβλιογραφία. Αρχικά, στο κεφάλαιο αυτό παρουσιάζεται η λεπτομερής περιγραφή του προβλήματος φόρτωσης container, υποκατηγορία του οποίου αποτελεί το πρόβλημα φόρτωσης φορτηγών οχημάτων. Στη συνέχεια, περιγράφονται εργασίες αναφορικά με τη φόρτωση φορτηγών οχημάτων για μεταφορά container (φόρτωση ενός container, φόρτωση παλετών σε φορτηγά οχήματα), που λαμβάνουν υπόψιν περιορισμούς ευστάθειας, η ανάπτυξη των οποίων αποτελεί αντικείμενο της παρούσας Διπλωματικής Εργασίας. Τέλος, αναφέρονται εργασίες σχετικές με τη φόρτωση βυτιοφόρων οχημάτων, οι οποίες όμως δε λαμβάνουν υπόψιν περιορισμούς ευστάθειας. Συνεπώς, περιγράφονται διαθέσιμα μοντέλα και μεθοδολογίες, τα οποία είναι χρήσιμα για την υλοποίηση της Διπλωματικής και ταυτόχρονα τονίζεται το καινοτόμο στοιχείο της, που είναι η ένταξη των περιορισμών ευστάθειας σε κάθε φάση της διαδρομής ενός βυτιοφόρου οχήματος.

2.1 Πρόβλημα φόρτωσης container

2.1.1 Ορισμός προβλήματος

Η αποτελεσματική φόρτωση κιβωτίων σε containers (εμπορευματοκιβώτια) και η βέλτιστη αξιοποίηση του διαθέσιμου χώρου του container αποτελούν ένα πολύ σημαντικό πεδίο έρευνας. Το πρόβλημα φόρτωσης container (container loading problem) είναι ένα πρόβλημα συνδυαστικής βελτιστοποίησης με σημαντικό αντίκτυπο στην οικονομία, το περιβάλλον και την ασφάλεια και ανήκει στην κατηγορία των τρισδιάστατων προβλημάτων «κοπής και πακετοποίησης» -«cutting and packing».

Αποτελεί ένα πρόβλημα αντιστοίχισης, στο οποίο «μικρά» τρισδιάστατα ορθογώνια αντικείμενα (κιβώτια) πρέπει να αντιστοιχηθούν σε «μεγάλα» τρισδιάστατα ορθογώνια αντικείμενα (containers), έτσι ώστε να βελτιστοποιηθεί μία αντικειμενική συνάρτηση, λαμβάνοντας υπόψιν δύο βασικούς περιορισμούς που αφορούν την εφικτότητα του προβλήματος: 1) τα μικρά αντικείμενα να τοποθετούνται ολόκληρα μέσα στο container και 2) τα μικρά αντικείμενα να μην αλληλεπικαλύπτονται μεταξύ τους. Ως "μεγάλο" τρισδιάστατο αντικείμενο μπορεί να οριστεί ένα container, ο χώρος φόρτωσης ενός φορτηγού ή μία παλέτα. Τα "μικρά" αντικείμενα στη βιβλιογραφία καλούνται «κουτιά».

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Συνεπώς, στο πρόβλημα αυτό ορθογώνια κουτιά φορτώνονται σε ένα όχημα ή μία παλέτα ή ένα container, έως ότου να μην είναι εφικτή η φόρτωση κάποιου άλλου κουτιού. Καθίσταται, συνεπώς, προφανές ότι το πρόβλημα φόρτωσης φορτηγών οχημάτων αποτελεί μία υποκατηγορία του προβλήματος φόρτωσης container (Wäscher et al., 2007).

Το είδος του προβλήματος μπορεί να χωριστεί σε τρεις βασικές κατηγορίες ανάλογα με τον αριθμό των διαφορετικών τύπων κιβωτίων προς φόρτωση: 1. έντονα ετερογενές, όπου υπάρχουν πολλά είδη κιβωτίων με διαφορετικές διαστάσεις, 2. ελαφρώς ετερογενές, με λίγα διαφορετικά είδη κιβωτίων και 3. ομοιογενές, όπου όλα τα κιβώτια έχουν ίδιο σχήμα και διαστάσεις.

Σύμφωνα με τους Wäscher et al., 2007, το πρόβλημα φόρτωσης container μπορεί να διακριθεί σε δύο κατηγορίες: μεγιστοποίηση εξόδου, όπου ο αριθμός των containers είναι περιορισμένος και ως αντικειμενική συνάρτηση ορίζεται η μεγιστοποίηση του αριθμού των κιβωτίων που μπορούν να φορτωθούν στα διαθέσιμα containers (output maximization), και ελαχιστοποίηση εισόδου, όπου ο αριθμός των containers είναι επαρκής για τη φόρτωση όλων των κιβωτίων και ως αντικειμενική συνάρτηση ορίζεται η ελαχιστοποίηση του αριθμού των containers (input minimization).

2.1.2 Περιορισμοί προβλήματος

Το πρόβλημα φόρτωσης container, εκτός από τη μεγιστοποίηση της αξιοποίησης του χώρου, πρέπει να ικανοποιεί και πρακτικούς περιορισμούς. Σύμφωνα με τους Bortfeldt & Wäscher, 2013, γίνεται διάκριση μεταξύ των περιορισμών που σχετίζονται με τα μεγάλα αντικείμενα (container-related constraints), με τα μικρά αντικείμενα, είτε με ένα μοναδικό αντικείμενο (item related constraints,) είτε με ένα σύνολο αντικειμένων (cargo-related constraints), με την τοποθέτηση των μικρών αντικειμένων μέσα στα containers (positioning constraints) και τέλος, με το αποτέλεσμα της διαδικασίας συσκευασίας, δηλ. με τη φόρτωση (load-related constraints).

2.1.2.1 Περιορισμοί που σχετίζονται με τα containers (Container-related constraints)

2.1.2.1.1 Όριο βάρους του container (Container weight limits)

Ο περιορισμός του ορίου βάρους του container αναφέρεται στο μέγιστο επιτρεπόμενο βάρος το οποίο μπορεί να φορτωθεί στο container και στα όρια του μέγιστου επιτρεπόμενου βάρους που μπορεί να υποστηρίξει ο κάθε άξονας. Στην περίπτωση βαρέων αντικειμένων, τα όρια βάρους αποτελούν πιο αυστηρό περιορισμό από τον περιορισμό του όγκου που τίθεται από τις διαστάσεις του container. Σύμφωνα με τους Bortfeldt & Wäscher, 2013, το όριο βάρους του container λαμβάνεται υπόψιν στο 14% των ερευνών (πχ Gehring & Bortfeldt, 1997, Liu & Hsiao, 1997, Terno et al., 2000, Iori & Martello, 2010, Chan et al., 2006, Egeblad et al.,2010 και Wang et al., 2010).

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Στα μαθηματικά μοντέλα, ο περιορισμός του ορίου βάρους θεωρείται σαν περιορισμός της μορφής «linear knapsack», ο οποίος εξασφαλίζει ότι το συνολικό βάρος του φορτίου δεν θα υπερβαίνει το όριο βάρους του container. Σε πρόσφατες έρευνες (Lim et al., 2013, Pollaris et al., 2014, Alonso et al., 2017, Alonso et al., 2016,), εκτός από το όριο του συνολικού μεικτού βάρους φόρτωσης σε ένα φορτηγό όχημα, λαμβάνεται υπόψιν στους αλγορίθμους και το μέγιστο βάρος που μπορούν να αντέξουν οι άξονες του οχήματος. Η υπέρβαση του βάρους κατ' άξονα αποτελεί σημαντική απειλή για την οδική ασφάλεια και μπορεί να προκαλέσει βλάβες στην επιφάνεια του οδοστρώματος, καθώς οι τροχοί μεταφέρουν αυτό το βάρος στο οδόστρωμα μέσω της επιφάνειας επαφής των ελαστικών με το έδαφος, ενώ παράλληλα οι μεταφορείς υπόκεινται σε πρόστιμα σε περίπτωση υπέρβασης. Συνεπώς, η συμπερίληψη των περιορισμών αυτών είναι μείζονος σημασίας.

2.1.2.1.2 Κατανομή βάρους μέσα στο container (Weight distribution/Load balance constraints)

Οι περιορισμοί που αφορούν την κατανομή του βάρους, απαιτούν το βάρος του φορτίου να κατανέμεται όσο το δυνατόν πιο ομοιόμορφα στο πάτωμα του container, για να αποφεύγονται οι μετακινήσεις του φορτίου κατά τη μεταφορά, να ισορροπεί το φορτίο ανάμεσα στους άξονες του φορτηγού και να μην υπερβαίνεται το μέγιστο επιτρεπόμενο μικτό βάρος ανά άξονα. Η μη ισορροπία των φορτίων μπορεί να οδηγήσει σε ανομοιόμορφες κατανομές του βάρους στους άξονες ενός φορτηγού οχήματος, γεγονός το οποίο έχει δυσμενείς επιπτώσεις στην σταθερότητα κατά την κίνηση, την επιτάχυνση, την επιβράδυνση και την αλλαγή κατεύθυνσης του οχήματος.

Όπως επισημαίνεται από τους Bortfeldt & Wäscher, 2013, στις περισσότερες έρευνες ο περιορισμός της σωστής κατανομής του βάρους του φορτίου, θεωρείται ως «χαλαρός» περιορισμός, δηλαδή το κέντρο βάρους του φορτίου πρέπει να βρίσκεται στο γεωμετρικό κέντρο βάρους του container (Gehring et al., 1990, Davies & Bischoff, 1999, Techanitisawad & Tangwiwatwong, 2004, Balakirsky et al., 2010) ή να μην υπερβαίνει μία συγκεκριμένη απόσταση από αυτό (Bortfeldt & Gehring, 1997, Liu et al., 2007, Sciomachen & Tanfani, 2007, Liu et al., 2011). Μια διαφορετική προσέγγιση αναφέρεται στον καθορισμό μίας εφικτής περιοχής: το κέντρο βάρους του φορτίου να τοποθετείται στο διάστημα $[\frac{1}{4}, \frac{3}{4}]$ του μήκους, $[\frac{1}{4}, \frac{3}{4}]$ του πλάτους και $[0, \frac{1}{2}]$ του ύψους του container (Baldi et al., 2012), ή με απόκλιση της τάξης του 5% από το μήκος και το πλάτος (Moon & Nguyen, 2014).

Η αντιμετώπιση του προβλήματος της κατανομής βάρους είναι μείζονος σημασίας από την άποψη της αποφυγής μετακίνησης και βλάβης του φορτίου, της εγγύησης της ασφάλειας του προσωπικού, που φορτώνει/εκφορτώνει τα αντικείμενα, της σωστής μεταχείρισης του φορτίου και της οδικής ασφάλειας του οχήματος.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

2.1.2.2 Περιορισμοί που σχετίζονται με το αντικείμενο (Item-related constraints)

2.1.2.2.1 Προτεραιότητες φόρτωσης (Loading priorities)

Στην πράξη, η φόρτωση κάποιων αντικειμένων είναι πιο σημαντική από τη φόρτωση άλλων, λόγω προθεσμιών παράδοσης ή απαιτήσεων που σχετίζονται με τη διάρκεια ζωής των προϊόντων. Τα αντικείμενα διακρίνονται σε κλάσεις διαφορετικής προτεραιότητας. Η προτεραιότητα μπορεί να είναι απόλυτη (τα αντικείμενα φορτώνονται αποκλειστικά βάσει της προτεραιότητάς τους) ή σχετική (συγκριτική αξία τοποθέτησης ενός αντικειμένου σε ένα container αντί για ένα άλλο, χωρίς να πραγματοποιούνται αποκλεισμοί κατά τη φόρτωση).

2.1.2.2.2 Περιορισμοί προσανατολισμού (Orientation constraints)

Υπάρχουν έξι προσανατολισμοί σύμφωνα με τους οποίους ένα κουτί μπορεί να τοποθετηθεί ορθογώνια εντός ενός container. Στην πράξη, ο αποδεκτός αριθμός κατακόρυφου και οριζόντιου προσανατολισμού περιορίζεται σε μία ή δύο διαστάσεις (πχ τα μακριά αλλά χαμηλά και στενά κουτιά δεν πρέπει να τοποθετούνται πάνω στη μικρότερη επιφάνεια τους). Χαρακτηριστικά, αναφέρεται η ένδειξη που υπάρχει στα κουτιά και προσδιορίζει το πάνω μέρος τους (κατακόρυφος προσανατολισμός) και ο προσδιορισμός του μπροστά μέρους των κουτιών στην περίπτωση συσκευασίας φορτίου από δύο εισόδους (οριζόντιος προσανατολισμός).

2.1.2.2.3 Περιορισμοί στοιβασίας (Stacking/ Load-bearing constraints)

Οι περιορισμοί αυτοί καθορίζουν πως τοποθετούνται τα κουτιά το ένα πάνω στο άλλο λόγω περιορισμένης φέρουσας αντοχής τους. Το μέγιστο βάρος/πίεση που μπορεί να αντέξει ένα κουτί εξαρτάται από τον τρόπο και το υλικό κατασκευής του, το περιεχόμενο και τις συνθήκες (υγρασία, διάρκεια φόρτωσης). Μία προσέγγιση αφορά την τοποθέτηση κουτιών με μεγαλύτερη πυκνότητα κάτω από κουτιά με μικρότερη.

2.1.2.3 Περιορισμοί που σχετίζονται με το φορτίο (Cargo-related constraints)

2.1.2.3.1 Περιορισμοί πλήρους αποστολής (Complete-shipment constraints)

Ομάδες αντικειμένων ενός φορτίου μπορεί να συγκροτούν λειτουργικές οντότητες (πχ εξαρτήματα για τη συναρμολόγηση ενός μηχανισμού) αλλά για διαχειριστικούς λόγους να πρέπει να μεταχειριστούν ως απλές οντότητες. Στην περίπτωση αυτή πρέπει να διασφαλιστεί ότι εάν τουλάχιστον ένα αντικείμενο μίας τέτοιας ομάδας φορτωθεί σε μία αποστολή τότε πρέπει και τα υπόλοιπα αντικείμενα να φορτωθούν είτε στο ίδιο είτε σε διαφορετικό container.

2.1.2.3.2 Περιορισμοί κατανομής (Allocation constraints)

Από τη μία πλευρά, απαιτείται συγκεκριμένα αντικείμενα να τοποθετηθούν στο ίδιο container (connectivity constraints), λόγω ίδιου προορισμού ή επιθυμίας του πελάτη να τα λάβει σαν ενιαία

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

αποστολή. Από την άλλη πλευρά απαιτείται συγκεκριμένες κατηγορίες αντικειμένων (πχ τρόφιμα και αρώματα) να μη φορτώνονται στο ίδιο container (separation constraints).

2.1.2.4 Περιορισμοί τοποθέτησης (Positioning constraints)

Οι περιορισμοί τοποθέτησης διακρίνονται σε απόλυτους, σχετικούς και πολλαπλών προορισμών. Οι απόλυτοι περιορισμοί απαιτούν συγκεκριμένα αντικείμενα να τοποθετηθούν ή όχι σε συγκεκριμένες θέσεις/περιοχές εντός του container λόγω μεγέθους, βάρους ή περιεχομένου (πχ η είσοδος αποτελεί την πιο δύσκολη περιοχή φόρτωσης, λόγω περιορισμένης ευελιξίας του οχήματος, και δεν πρέπει να τοποθετούνται σε αυτήν ογκώδη αντικείμενα).

Οι σχετικοί περιορισμοί απαιτούν συγκεκριμένα αντικείμενα είτε να τοποθετούνται κοντά (ομαδοποίηση αντικειμένων) επειδή προορίζονται για τον ίδιο πελάτη, είτε να τοποθετούνται σε απόσταση λόγω αρνητικών αλληλεπιδράσεων (πχ τρόφιμα και χημικές ουσίες). Τέλος, αναφορικά με τους πολλαπλούς περιορισμούς, τα αντικείμενα τα οποία προορίζονται για διαφορετικούς πελάτες πρέπει να ταξινομούνται με τέτοιον τρόπο ώστε να διευκολύνουν τη σειρά επίσκεψης και να αποφεύγονται επιπλέον διαδικασίες εκφόρτωσης και επαναφόρτωσης κατά τη διάρκεια της μεταφοράς.

2.1.2.5 Περιορισμοί που σχετίζονται με τη φόρτωση (Load-related constraints)

2.1.2.5.1 Περιορισμοί ευστάθειας (Stability Constraints)

Τα ασταθή φορτία μπορεί να οδηγήσουν σε φθορές των αντικειμένων ή και σε τραυματισμούς του προσωπικού κατά τη διάρκεια φόρτωσης/εκφόρτωσης τους. Από πρακτική σκοπιά, προκειμένου να εμποδιστεί η μετακίνηση του φορτίου χρησιμοποιούνται ιμάντες, αερόσακοι, αφρολέξ και άλλα βοηθητικά μέσα. Επιπλέον, γίνεται διάκριση ανάμεσα στην κατακόρυφη και την οριζόντια ευστάθεια.

Η κατακόρυφη/στατική ευστάθεια εξασφαλίζει την μη πτώση των αντικειμένων (ικανότητα αντίστασης στη βαρύτητα) κατά την ακινησία του οχήματος. Διασφαλίζεται μέσω πλήρους ή μερικής υποστήριξης της βάσης του αντικειμένου από το πάτωμα του container/την επιφάνεια άλλου αντικειμένου ή της απαίτησης να υποστηρίζεται το κέντρο βάρους ενός αντικείμενου από μία επιφάνεια (στατική μηχανική ισορροπία).

Η οριζόντια/δυναμική ευστάθεια ενός αντικειμένου διαβεβαιώνει ότι αυτό δεν θα μετακινηθεί σημαντικά κατά τη μετακίνηση του οχήματος και διασφαλίζεται μέσω της πλευρικής επαφής του με τον τοίχο του container ή με άλλο αντικείμενο. Σύμφωνα με τους Bischoff & Ratcliff, 1995a, πρέπει τουλάχιστον οι τρείς (από τις τέσσερεις) πλευρικές επιφάνειες ενός αντικειμένου να είναι σε επαφή με μία άλλη επιφάνεια.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

2.1.2.5.2 Περιορισμοί πολυπλοκότητας (Complexity Constraints)

Οι περιορισμοί πολυπλοκότητας αντανακλούν περιορισμούς ανθρώπινου δυναμικού και τεχνολογικού εξοπλισμού. Η υλοποίηση σύνθετων μοτίβων φόρτωσης μπορεί να είναι χρονοβόρα και μη κατανοητή από το προσωπικό φόρτωσης, ενώ από την άλλη οι αυτοματοποιημένες τεχνολογίες φόρτωσης ενδεχομένως να μην είναι κατάλληλες και να απαιτούν επιπρόσθετη εργασία υψηλού κόστους. Ενδεικτικά, αναφέρεται ότι ο πιο συχνός περιορισμός πολυπλοκότητας είναι η κοπή μέσω γκιλοτίνας (guillotine cutting), δηλαδή κοπές παράλληλες στις πλευρές του container.

2.1.3 Αλγόριθμοι επίλυσης

Το πρόβλημα φόρτωσης container είναι ένα τυπικό NP-hard πρόβλημα, είναι ιδιαίτερα πολύπλοκο στην επίλυση του και απαιτεί μεγάλη υπολογιστική ισχύ και χρόνο επίλυσης. Για την επίλυσή του χρησιμοποιούνται ακριβείς, ευρετικοί και μετευρετικοί αλγόριθμοι.

Ένας ακριβής αλγόριθμος στοχεύει στον προσδιορισμό βέλτιστων λύσεων αλλά είναι υπολογιστικά δαπανηρός και όχι πρακτικός όταν πρόκειται για πραγματικά προβλήματα. Ευρέως γνωστοί ακριβείς αλγόριθμοι είναι ο γραμμικός, ο ακέραιος προγραμματισμός και η μέθοδος branch and bound. Ο γραμμικός προγραμματισμός επιτρέπει την έκφραση των σχέσεων μεταξύ των εμπλεκόμενων οντοτήτων μέσω μεταβλητών και την έκφραση των κανόνων του προβλήματος μέσω περιορισμών, ενώ μέσω της αντικειμενικής συνάρτησης τίθεται ως στόχος η εύρεση της βέλτιστης ή στις περιπτώσεις που αυτό δεν είναι εφικτό, μιας πολύ καλής λύσης. Η τεχνική αναζήτησης branch and bound είναι μία αποτελεσματική προσέγγιση για την επίλυση προβλημάτων γραμμικού προγραμματισμού και βασίζεται στην αναδρομική διαίρεση δύσκολων προβλημάτων σε συνεχώς μικρότερα υποπροβλήματα.

Οι Martello et al., 2000, παρουσίασαν έναν ακριβή αλγόριθμο με τη χρήση του περιορισμού της περιστροφής και ως μέγιστο αριθμό κιβωτίων προς φόρτωση τα 20 κιβώτια. Ο Hifi, 2004, παρουσίασε ένα ακριβή αλγόριθμο, χωρίς χρήση περιορισμών με δυνατότητα υπολογισμού για έως 50 κιβώτια προς φόρτωση, αλλά τα αποτελέσματα δεν ήταν ικανοποιητικά.

Ένας ευρετικός (heuristic) αλγόριθμος μπορεί να οδηγήσει σε μία λύση του προβλήματος κοντά στη βέλτιστη σε σύντομο υπολογιστικό χρόνο και παρέχει την απαραίτητη ευελιξία για την προσαρμογή διαφορετικών αναγκών. Ο κατασκευαστικός αλγόριθμος παράγει τη λύση προσθέτοντας βαθμιαία μέρη της λύσης στην αρχικά κενή μερική λύση. Είναι εν γένει ο γρηγορότερος προσεγγιστικός αλγόριθμος και οι πιο διαδεδομένες προσεγγίσεις για την επίλυση του προβλήματος φόρτωσης container είναι η τεχνική block building, η wall building και η layer building.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Η τεχνική block building τοποθετεί στο χώρο κυβοειδή blocks τα οποία αποτελούνται από κουτιά του ιδίου συνήθως μεγέθους και προσανατολισμού (Fanslau & Bortfeldt, 2010, Parreño et al., 2010, Eley, 2002, Bortfeldt et al., 2003). Η τεχνική wall building, η οποία προτάθηκε από τους George & Robinshon, 1980, βασίστηκε στην άποψη της πλήρωσης του container χρησιμοποιώντας κάθετα επίπεδα, τα οποία αποκαλούνται «τοίχοι». Κάθε επίπεδο είναι ένα τμήμα του μήκους (ή του πλάτους) του container, με ύψος και πλάτος (ή μήκος) που ταυτίζεται με το ύψος και πλάτος (ή μήκος) του container αντίστοιχα. Διαφορετικές μορφές της wall building προσέγγισης έχουν χρησιμοποιηθεί από τους Liu & Chen, 1981, Bischoff & Marriott, 1990, Hemminki, 1994 και Moura & Oliveira, 2005.

Η τεχνική layer building αφορά τη δημιουργία επιπέδων (layers), δηλαδή αντικειμένων ίδιου μεγέθους και προσανατολισμού τοποθετημένα σε οριζόντιες διατάξεις έτσι ώστε να σχηματίζουν ένα επίπεδο με επιφάνεια ίση με την επιφάνεια του χώρου. Ο χώρος συμπληρώνεται από τα οριζόντια επίπεδα από κάτω προς τα πάνω με όσο το δυνατόν λιγότερο κενό χώρο (Portmann, 1990, Loh & Nee, 1992, Bischoff et al., 1995).

Οι μετευρετικοί αλγόριθμοι διακρίνονται στους αλγορίθμους τοπικής αναζήτησης (local search techniques) και στους εξελικτικούς αλγορίθμους (evolutionary algorithms). Οι μέθοδοι τοπικής αναζήτησης επαναλαμβάνουν την αντικατάσταση μιας τρέχουσας λύσης από μία νέα λύση μέχρι να ικανοποιηθούν οι συνθήκες για τον τερματισμό της διαδικασίας. Μία τέτοια μέθοδος τοπικής αναζήτησης είναι η tabu search (Bortfeldt & Gehring, 1997), η οποία αποθηκεύει και χρησιμοποιεί προηγούμενες πληροφορίες έτσι ώστε να αποφευχθεί η ανακύκλωση των λύσεων και η παγίδευση στο τοπικό βέλτιστο. Οι εξελικτικοί αλγόριθμοι αποτελούν τεχνικές επίλυσης προβλημάτων βάσει πληθυσμών. Μία από αυτές τις τεχνικές χρησιμοποιεί τους γενετικούς αλγορίθμους (Bortfeldt & Gehring, 1997, 2001).

2.2 Φόρτωση φορτηγών οχημάτων για τη μεταφορά container με περιορισμούς ευστάθειας

2.2.1 Φόρτωση ενός container

Οι Davies & Bischoff, 1999 ανέπτυξαν μία ευρετική μέθοδο για την επίλυση του προβλήματος φόρτωσης container, με σκοπό το υψηλό ποσοστό αξιοποίησης του χώρου του, λαμβάνοντας υπόψιν τον περιορισμό της ομοιόμορφης κατανομής του βάρους του φορτίου. Η μέθοδος αυτή χωρίζει το container σε κατακόρυφους τοίχους, διαφορετικού μήκους με πλάτος και ύψος όσο του container και δημιουργεί blocks κουτιών, τα οποία συνδυάζουν μία ή περισσότερες στρώσεις. Τα κουτιά επιλέγονται με βάση τα εξής κριτήρια: μεγιστοποίηση του όγκου μίας ομοιογενής στήλης (στήλη με κουτιά του ίδιου τύπου), ελαχιστοποίηση του μήκους τους και μεγιστοποίηση του όγκου του κάθε κουτιού. Μέσω

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

της αλλαγής της διαμορφωμένης διάταξης (περιστροφή, ανταλλαγή, αντανάκλαση) των blocks επιτυγχάνεται αλλαγή της θέσης του κέντρο βάρους του φορτίου κατά μήκος του container. Η μέθοδος εφαρμόστηκε σε 1600 test cases και προέκυψαν τα εξής συμπεράσματα: όσο πιο ετερογενές είναι το φορτίο τόσο πιο ομοιόμορφη είναι η κατανομή του βάρους, όταν υπάρχει μικρό πλήθος κουτιών δημιουργούνται συστάδες (clusters) πολύ βαριών ή πολύ ελαφριών κουτιών γεγονός το οποίο οδηγεί σε ανομοιόμορφη κατανομή του βάρους, αλλά όσο το πλήθος των κουτιών αυξάνεται, η κατανομή γίνεται όλο και πιο ομοιόμορφη. Γενικά, η μέθοδος παρήγαγε ικανοποιητικά αποτελέσματα τόσο ως προς την αξιοποίηση του όγκου του container, όσο και ως προς τη θέση του κέντρου βάρους.

Παράλληλα, και οι Liu et al., 2011 μελέτησαν το πρόβλημα φόρτωσης ενός container με ετερογενείς τύπους κουτιών. Αναπτύχθηκε μία ευρετική μέθοδος με την προσέγγιση «tabu search». Λήφθηκαν υπόψιν πρακτικοί περιορισμοί, όπως ο προσανατολισμός, η στατική ευστάθεια, το όριο βάρους και η κατανομή του βάρους. Κατά τη διαδικασία φόρτωσης των κουτιών, ακολουθήθηκαν ευρετικές στρατηγικές βάσει του προσανατολισμού, της θέσης και της δημιουργίας ορθογώνιων blocks. Για τη διαχείριση των εναπομείναντων χώρων, πραγματοποιήθηκαν τεχνικές συγχώνευσης των χώρων που έχουν το ίδιο ύψος, με σκοπό τη μέγιστη αξιοποίηση του όγκου.

Η αντικειμενική συνάρτηση περιελάμβανε το σταθμισμένο άθροισμα τριών μαθηματικών εκφράσεων: το ποσοστό αξιοποίησης του όγκου, το πηλίκο του συνολικού βάρους του φορτίου προς το όριο βάρους και το ποσοστό απόκλισης του κέντρου βάρους του φορτίου από το γεωμετρικό κέντρο βάρους του container. Ως απόσταση του κέντρου βάρους του φορτίου από το ιδανικό κέντρο βάρους του container θεωρήθηκε η μέγιστη από τις αποστάσεις στους τρείς άξονες. Η μέθοδος εφαρμόστηκε σε πραγματικά δεδομένα μέσω C++ και τα αποτελέσματα έδειξαν την αποτελεσματικότητά της και ότι μπορεί να επιτευχθεί ομοιόμορφη κατανομή του βάρους του φορτίου με μικρή μείωση του ποσοστού αξιοποίησης του χώρου (trade-off).

Η έρευνα των Lim et al., 2013 συγκροτήθηκε με σκοπό τη μελέτη του προβλήματος φόρτωσης ενός container (single container loading problem) με περιορισμούς που σχετίζονται με το συνολικό και το κατ' άξονα βάρος του φορτηγού οχήματος, όπως αυτοί προβλέπονται από τον κώδικα κυκλοφορίας της Καλιφόρνιας. Για την επίλυση του προβλήματος αναπτύχθηκε μία ευρετική μέθοδος, η οποία περιλαμβάνει έναν GRASP αλγόριθμο (Greedy Randomized Adaptive Search Procedure- διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης) με την τεχνική wall building, που επιλύει, επαναληπτικά, μοντέλα ακέραιου-γραμμικού προγραμματισμού. Αρχικά, εφαρμόστηκε ένας wall building αλγόριθμος (βασισμένος στον GRASP των Moura & Oliviera, 2005), ο οποίος λαμβάνει υπόψιν το ποσοστό αξιοποίησης και μη του container, την πυκνότητα του κάθε κιβωτίου και την αποφυγή της εμπλοκής των κιβωτίων.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Στη συνέχεια, χρησιμοποιήθηκε μία ολιστική προσέγγιση για να ληφθούν υπόψιν οι απαιτήσεις βάρους. Σημειώνεται ότι έγινε η υπόθεση ομαδοποίησης των αξόνων του φορτηγού σε δύο σημεία περιστροφής. Στην περίπτωση παραβίασης του μέγιστου επιτρεπόμενου συνολικού καθαρού βάρους του φορτίου, εφαρμόζεται ένα μοντέλο ακέραιου γραμμικού προγραμματισμού (IP3), ώστε να αφαιρεθεί ο απαραίτητος αριθμός κιβωτίων, ελαχιστοποιώντας το συνολικό όγκο τους. Στην περίπτωση παραβίασης του μέγιστου επιτρεπόμενου βάρους κατ' άξονα αρχικά εφαρμόζεται ένα wall-based μοντέλο μεικτού ακέραιου προγραμματισμού (IP2) για να βελτιωθεί η κατανομή των δυνάμεων εντός του container, εναλλάσσοντας τις θέσεις των τοίχων, με στόχο την ελαχιστοποίηση της απόστασης ανάμεσα στο πραγματικό και το ιδανικό κέντρο βάρους του φορτίου.

Έπειτα, εφαρμόζεται ένα μοντέλο ακέραιου προγραμματισμού (IP4), ώστε να αφαιρεθεί ο απαραίτητος αριθμός κιβωτίων, ικανοποιώντας τους περιορισμούς που αφορούν τις δυνάμεις που εφαρμόζονται στους άξονες του φορτηγού. Τέλος, εφαρμόζεται ξανά το μοντέλο IP2, για να βελτιωθεί η ισορροπία του container και η κατανομή των δυνάμεων, και προκύπτει η τελική διάταξη του φορτίου. Ο αλγόριθμος εφαρμόστηκε σε πραγματικά δεδομένα της εταιρείας Johnson Electric σε φορτηγό με τέσσερεις άξονες και container με μήκος 20 πόδια και επιλύθηκε με Java. Προέκυψε ότι η προτεινόμενη μέθοδος ήταν αποτελεσματική και ότι για να ικανοποιηθούν οι περιορισμοί που σχετίζονται με το βάρος για κάθε container πρέπει, ως αντιστάθμισμα (trade-off), να αυξηθεί ο αριθμός των απαιτούμενων containers για τον ίδιο αριθμό προϊόντων.

Οι Liu et al., 2017 ανέπτυξαν ένα μοντέλο ακέραιου προγραμματισμού για την επίλυση του ίδιου προβλήματος που μελετήθηκε από τους Lim et al., 2013. Ως αντικειμενική συνάρτηση τέθηκε η μεγιστοποίηση του συνολικού όγκου των κιβωτίων που τοποθετούνται σε ένα container. Οι περιορισμοί αφορούσαν τον προσανατολισμό των κιβωτίων, κάθε κιβώτιο να τοποθετείται ακριβώς μέσα στο container, δύο κιβώτια να μην υπερκαλύπτονται, να μην υπερβαίνεται το συνολικό όριο βάρους και το όριο βάρους ανά άξονα του φορτηγού οχήματος. Χρησιμοποιήθηκαν βοηθητικές μεταβλητές για να μετατραπούν οι μη γραμμικοί περιορισμοί που αφορούσαν τους περιορισμούς του βάρους κατ' άξονα σε γραμμικούς. Το μοντέλο εφαρμόστηκε σε μία μελέτη περίπτωσης που αφορούσε 30 κιβώτια για έπιπλα και επιλύθηκε μέσω του solver CPLEX, αποδεικνύοντας την αποτελεσματικότητά του.

Οι Ramos et al., 2018 ανέπτυξαν μια νέα μεθοδολογία για το πρόβλημα φόρτωσης container στις οδικές μεταφορές, η οποία λαμβάνει υπόψιν τρείς πρακτικούς περιορισμούς: ισορροπία φορτίου, όρια βάρους και στατική ευστάθεια. Αναφορικά με την ισορροπία του φορτίου, χρησιμοποιήθηκαν ειδικά διαγράμματα κατανομής φορτίου (load distribution diagram-LDD), τα οποία προσδιορίζουν το μέγιστο αποδεκτό φορτίο σε συνάρτηση με την επιμήκη και την εγκάρσια θέση του κέντρου βάρους του, ορίζοντας έτσι και μία εφικτή περιοχή της θέσης του κέντρου βάρους, σύμφωνα με τα τεχνικά

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

χαρακτηριστικά του φορτηγού οχήματος. Αναπτύχθηκε ένας multi-population biased random-key γενετικός αλγόριθμος (BRKGA) με συνάρτηση καταλληλότητας η οποία λαμβάνει υπόψιν τον χρησιμοποιούμενο όγκο του container, τη στατική ευστάθεια και την ισορροπία του φορτίου.

Τα υπολογιστικά πειράματα χωρίστηκαν σε δύο στάδια. Στο πρώτο στάδιο, αξιολογήθηκαν οι λύσεις που έχουν προκύψει από άλλους αλγορίθμους ως προς τη θέση του κέντρου βάρους του φορτίου σύμφωνα με τα διαγράμματα LDD, αποδεικνύοντας ότι δεν παράγουν απαραίτητα λύσεις που λαμβάνουν υπόψιν την ισορροπία του φορτίου. Στο δεύτερο στάδιο, ο αλγόριθμος εφαρμόστηκε σε ένα σύνολο δεδομένων (θεωρώντας φορτηγό όχημα με δύο άξονες) και υιοθετήθηκαν δύο διαφορετικές στρατηγικές αναφορικά με τη στατική ευστάθεια του φορτίου: (1) πλήρης υποστήριξη βάσης και (2) στατική μηχανική ισορροπία. Τα αποτελέσματα έδειξαν ότι ο αλγόριθμος είναι αποτελεσματικός και ότι μπορούν να παραχθούν λύσεις που ικανοποιούν και τους τρείς περιορισμούς ταυτόχρονα χωρίς να διακυβεύεται ο όγκος του container που χρησιμοποιείται.

2.2.2 Φόρτωση παλετών σε φορτηγά οχήματα

Οι Alonso et al., 2016 μελέτησαν το συνδυαστικό πρόβλημα τοποθέτησης προϊόντων (σε στρώσεις) σε παλέτες και τη φόρτωση των παλετών σε φορτηγά οχήματα, λαμβάνοντας υπόψιν περιορισμούς που σχετίζονται με το μέγιστο συνολικό βάρος του φορτίου, το μέγιστο επιτρεπτό βάρος κατ' άξονα και την κατανομή του φορτίου. Αναπτύχθηκε ένας GRASP αλγόριθμος, ο οποίος περιελάμβανε στρατηγική τυχαίας επιλογής για την κατασκευαστική φάση και κάποιες βελτιωτικές κινήσεις. Στη φάση της τοποθέτησης των στρώσεων στις παλέτες λήφθηκαν υπόψιν περιορισμοί προσανατολισμού, υποστήριξης, προτεραιότητας και stackability, ενώ στη φάση της φόρτωσης των φορτηγών οχημάτων (δύο αξόνων) λήφθηκαν υπόψιν, πέρα από τους περιορισμούς του βάρους, περιορισμοί προτεραιότητας, ευστάθειας και stackability. Ως αντικειμενική συνάρτηση τέθηκε η ελαχιστοποίηση του συνολικού αριθμού των φορτηγών και ως δευτερεύουσα αντικειμενική η ελαχιστοποίηση του φορτίου του τελευταίου φορτηγού, ώστε να υπάρχει επαρκής χώρος για νέες παραγγελίες.

Ο κατασκευαστικός αλγόριθμος παράγει τη λύση σε δύο βήματα: 1. επιλογή της θέσης στο φορτηγό για την τοποθέτηση της επόμενης παλέτας και 2. τοποθέτηση στρώσεων στην παλέτα αυτή. Προκειμένου να ληφθούν υπόψιν οι περιορισμοί του κέντρου βάρους και του βάρους κατ' άξονα, το φορτηγό χωρίστηκε σε δύο τμήματα (μπροστινό-πίσω) και η φόρτωση των παλετών γινόταν εναλλάξ για κάθε τμήμα ξεκινώντας από το κέντρο με σκοπό τη διατήρηση του κέντρου βάρους του φορτίου κοντά στο γεωμετρικό κέντρο βάρος του φορτηγού. Ο αλγόριθμος εφαρμόστηκε σε πραγματικά δεδομένα της εταιρίας ORTEC μέσω C++ και τα αποτελέσματα συγκρίθηκαν με απλά κατώτερα όρια, καταδεικνύοντας ότι στην πλειοψηφία των περιπτώσεων οι λύσεις του αλγορίθμου είναι βέλτιστες.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Στη συνέχεια, οι Alonso et al., 2017 ανέπτυξαν και μοντέλα ακέραιου-γραμμικού προγραμματισμού για την επίλυση του προβλήματος φόρτωσης πολλαπλών containers. Σκοπός του προβλήματος ήταν η ελαχιστοποίηση του πλήθους των containers που χρησιμοποιούνται για τη μεταφορά προϊόντων που έχουν τοποθετηθεί σε στρώσεις (layers) πάνω σε παλέτες. Αρχικά, αναπτύχθηκαν βασικά μοντέλα αναφορικά με τις διαστάσεις του φορτηγού και τη μη υπέρβαση του μέγιστου συνολικού και του κατ' άξονα βάρους του (μέσω ισορροπίας ροπών). Στη συνέχεια, λήφθηκαν υπόψιν και άλλοι παράγοντες όπως η χρήση μίας ή δύο παλετών ανά θέση, καθώς και η απόκλιση του κέντρου βάρους του φορτίου από το γεωμετρικό κέντρο βάρους του φορτηγού και η ελαχιστοποίηση του αριθμού των παλετών. Τα μοντέλα εφαρμόστηκαν σε πραγματικά δεδομένα (111 περιπτώσεις), επιλύθηκαν με CPLEX 12.51 και τα αποτελέσματα έδειξαν ότι στην πλειοψηφία των περιπτώσεων η βέλτιστη λύση εντοπίζεται σε μικρό υπολογιστικό χρόνο.

Και οι Costa et al. 2016 μελέτησαν το πρόβλημα φόρτωσης παλετών σε ένα φορτηγό όχημα μίας Πορτογαλικής εταιρείας λαμβάνοντας υπόψιν πρακτικούς περιορισμούς (ευστάθεια φορτίου, προσανατολισμός κιβωτίων, όριο βάρους φορτηγού και ομοιόμορφη κατανομή βάρους). Για τον σκοπό αυτό προτάθηκε μία κατασκευαστική ευρετική μέθοδος, η οποία βασίζεται στη διάταξη των κιβωτίων σε στρώσεις (layer-building) και περιλαμβάνει στρατηγική επιλογής γωνιών για την τοποθέτησή τους. Ως αντικειμενική συνάρτηση τέθηκε η μεγιστοποίηση του αριθμού των κιβωτίων μέσα στο φορτηγό, ενώ για την αξιολόγηση της ποιότητας της λύσης αναφορικά με την κατανομή του βάρους, χρησιμοποιήθηκε ως μετρική η σταθμισμένη απόσταση του κέντρου βάρους του φορτίου από το γεωμετρικό κέντρο βάρους του φορτηγού.

Οι γωνίες επιλέχθηκαν διαδοχικά μέσω star σχήματος, με τα πιο βαριά κιβώτια να τοποθετούνται μακριά το ένα από το άλλο και με κατεύθυνση από τις πλευρές προς το κέντρο του container, εξασφαλίζοντας επαρκή διασπορά των βαρέων φορτίων στο φορτηγό. Μετά το «χτίσιμο» των layers ακολούθησε η διαδικασία φόρτωσης των φορτηγών, κατατάσσοντας τα layers σε φθίνουσα σειρά ποσοστού πληρότητας και γεμίζοντας διαδοχικά το κάθε φορτηγό. Στην περίπτωση φορτηγών με χαμηλό συνολικό ποσοστό πληρότητας, πραγματοποιήθηκε μία τοπική διαδικασία και συνδυασμός κιβωτίων διαφορετικών layers με σκοπό τη μείωση του συνολικού αριθμού των φορτηγών. Η μέθοδος εφαρμόστηκε σε πραγματικά δεδομένα της εταιρίας μέσω Visual C++ και αξιολογήθηκε ως αποτελεσματική αναφορικά με τα ποσοστά πληρότητας και την κατανομή του βάρους των φορτίων.

Οι Novas et al., 2020 μελέτησαν τη φόρτωση παλετών σε φορτηγά οχήματα μέσω της ανάπτυξης μοντέλων «Γενικευμένου Διαζευκτικού Προγραμματισμού» για μία εταιρία εμφιάλωσης αναψυκτικών. Αναπτύχθηκαν δύο μοντέλα, το πρώτο με σκοπό τη βελτίωση της διαδικασίας φόρτωσης και το δεύτερο με σκοπό να εξασφαλιστεί η ισορροπία του φορτίου και η μη υπέρβαση του βάρους κατ' άξονα. Γίνεται η υπόθεση ότι το φορτηγό χωρίζεται σε δύο πλευρές κατά πλάτος και στην κάθε πλευρά μπορεί να

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

τοποθετηθεί μόνο μία παλέτα στο ίδιο μήκος. Οι περιορισμοί του πρώτου μοντέλου ήταν οι εξής: μη υπέρβαση του ορίου συνολικού βάρους του φορτίου, μη υπέρβαση του συνολικού μήκους του φορτηγού, τα μη ευσταθή προϊόντα να μην υπερβαίνουν ένα ποσοστό, η ποσότητα κάθε προϊόντος να ικανοποιεί κάποια ανώτερα και κατώτερα όρια και η διαφορά ανάμεσα στα βάρη που ανατίθενται στις δύο πλευρές του φορτηγού να μην υπερβαίνει μία συγκεκριμένη τιμή. Αναφορικά με το δεύτερο μοντέλο, ως τύπος οχήματος ορίστηκε ένα αρθρωτό φορτηγό με τέσσερεις άξονες και οι περιορισμοί ήταν οι εξής: μη υπέρβαση του ορίου βάρους ανά άξονα, ισορροπία δυνάμεων και ροπών στον τράκτορα και το τρέιλερ, μη δημιουργία τεχνητών κενών και τέλος, ικανοποίηση της εξίσωσης συμβιβαστού των μετακινήσεων.

Αναφορικά με την αντικειμενική συνάρτηση τέθηκαν τρείς διαφορετικοί δείκτες λειτουργικής απόδοσης: μεγιστοποίηση του φορτίου, ελαχιστοποίηση της εργασίας στην αποθήκη και τέλος, ικανοποίηση της ζήτησης. Τα δύο μοντέλα εφαρμόστηκαν σε πραγματικά δεδομένα της εταιρίας και σε διαφορετικά σενάρια μέσω του solver Cplex 12.6.3.0 και τα αποτελέσματα έδειξαν ότι το δεύτερο μοντέλο με αντικειμενική συνάρτηση τη μεγιστοποίηση του φορτίου είναι το καλύτερο, δεδομένου ότι το πρόστιμο παραβίασης των ορίων βάρους στους άξονες υπερβαίνει κατά πολύ το κόστος του μικρότερου φορτίου το οποίο προκύπτει συγκριτικά με το πρώτο μοντέλο.

Τέλος, οι Pollaris et al., 2014 μελέτησαν το πρόβλημα δρομολόγησης ομογενούς στόλου φορτηγών οχημάτων συνδυαστικά με τη φόρτωση ομογενών παλετών και λαμβάνοντας υπόψιν περιορισμούς βάρους κατ' άξονα για κάθε φάση της διαδρομής. Για το σκοπό αυτό αναπτύχθηκε ένα μοντέλο μεικτού ακέραιου γραμμικού προγραμματισμού. Θεωρήθηκε αρθρωτό φορτηγό με πέντε άξονες και ορίστηκαν τα μέγιστα επιτρεπόμενα όρια συνολικού βάρους, βάρους αξόνων του trailer (κεντρικός άξονας) και βάρους συζευκτήρα. Οι παλέτες τοποθετήθηκαν σε δύο οριζόντιες σειρές, χωρίς να στοιβάζονται κατακόρυφα και χωρίς κενά, ενώ η φόρτωση έγινε βάσει ακολουθίας. Ορίστηκε το κέντρο βάρους του φορτίου δυναμικά κατά την άφιξη σε κάθε πελάτη βάσει του πλήθους των παλετών που υπήρχαν και εκείνων που φορτώνονταν στο όχημα.

Ως αντικειμενική συνάρτηση τέθηκε η ελαχιστοποίηση του κόστους της διαδρομής και κάποιοι από τους περιορισμούς ήταν οι εξής: μη υπέρβαση του συνολικού μέγιστου επιτρεπτού ορίου βάρους του οχήματος και μη υπέρβαση ανώτερων και κατώτερων ορίων επιτρεπτού βάρους στον συζευκτήρα και στους άξονες του trailer κατά τη διάρκεια του ταξιδιού. Το μοντέλο εφαρμόστηκε μέσω Cplex 12.5 σε τυχαία παραγόμενα δεδομένα και προέκυψε ότι η ενσωμάτωση των περιορισμών βάρους κατ' άξονα δεν οδηγεί απαραίτητα σε σημαντική αύξηση του τελικού κόστους και ότι το ποσοστό της αύξησης εξαρτάται από το πλήθος και το βάρος των παλετών ανά πελάτη και τη μέγιστη παραγγελία.

2.3 Φόρτωση βυτιοφόρων οχημάτων-Πρόβλημα φόρτωσης και δρομολόγησης βυτιοφόρων οχημάτων για τη διανομή καυσίμων

Οι Brown et al., 1987 μελέτησαν το σύστημα της εταιρείας Mobil Oil Corporation για τη διανομή προϊόντων πετρελαίου σε πελάτες (computer assisted dispatch-CAD), το οποίο βασίζεται σε μοντέλα ακέραιου προγραμματισμού. Τα βυτιοφόρα οχήματα της εταιρείας έχουν διαφορετικές χωρητικότητες και αποτελούνται από διαμερίσματα διαφορετικού πλήθους και χωρητικότητας. Η κάθε παραγγελία μπορεί να αποτελείται από αρκετά προϊόντα και να είναι πλήρους ή μερικού φορτίου. Η αντιστοίχιση των παραγγελιών στα οχήματα απαιτεί διόρθωση των ποσοτήτων, ώστε να προσαρμόζονται στα διαμερίσματα. Η φόρτωση των οχημάτων πρέπει γίνεται με ασφαλή τρόπο (σε μερικώς φορτωμένα οχήματα το φορτίο πρέπει να βρίσκεται στα μπροστινά διαμερίσματα). Οι στόχοι της διαδικασίας αποστολής είναι η ελαχιστοποίηση του κόστους διανομής και η φόρτωση έως το μέγιστο επιτρεπόμενο βάρος των οχημάτων.

Το CAD ακολουθεί την εξής διαδικασία: αρχικά πραγματοποιείται μία προκαταρκτική αντιστοίχιση παραγγελιών σε οχήματα, με βάση την εγγύτητα των πελατών και κατά πόσο προσαρμόζονται οι παραγγελίες σαν φορτία στα οχήματα. Στη συνέχεια, με βάση την αρχική αντιστοίχιση, πραγματοποιούνται ανταλλαγές φορτίων στα φορτηγά ώστε να βελτιωθεί η ποιότητα της λύσης (μείωση του συνολικού κόστους). Καθορίζεται η βέλτιστη διαδρομή του κάθε οχήματος για την παράδοση των παραγγελιών (TSP) και τέλος, γίνεται η βέλτιστη αντιστοίχιση των προϊόντων στα διαμερίσματα των οχημάτων μέσω ενός μαθηματικού μοντέλου. Το μοντέλο αυτό ενσωματώνει τις «κυρώσεις» που προκύπτουν από την προσαρμογή της ποσότητας της παραγγελίας κάθε προϊόντος (αύξηση/μείωση). Η αντικειμενική συνάρτηση ελαχιστοποιεί τις κυρώσεις λόγω αλλαγών στις ποσότητες της παραγγελίας, υπό τους εξής περιορισμούς: αντιστοίχιση ενός μόνο προϊόντος ανά διαμέρισμα, οι προσαρμογές να βρίσκονται εντός των αποδεκτών ορίων, μη υπέρβαση του όγκου και του μέγιστου επιτρεπόμενου βάρους του διαμερίσματος.

Επιπλέον, και οι Avella et al., 2004, μελέτησαν την περίπτωση μίας εταιρείας διανομής καυσίμων σε αντλίες, μέσω στόλου βυτιοφόρων οχημάτων, καθένα από τα οποία αποτελείται από δεξαμενές διαφορετικής χωρητικότητας, με σκοπό το ελάχιστο κόστος (σε χρόνο) ταξιδιού. Το πρόβλημα αυτό θεωρείται συνδυασμός προβλήματος συσκευασίας και δρομολόγησης. Η εταιρία υιοθετεί την εξής πολιτική για την απλοποίηση του προβλήματος: κάθε δεξαμενή ενός οχήματος είναι ή εντελώς γεμάτη ή εντελώς άδεια και όχι μερικώς γεμισμένη και όλο το καύσιμο μίας δεξαμενής πρέπει να παραδίδεται σε μία και μόνο αντλία. Επιπλέον, δεν πρέπει να υπερβαίνεται η χωρητικότητα ενός οχήματος. Αρχικά, αναπτύχθηκε μία γρήγορη ευρετική μέθοδος για την επίλυση του προβλήματος συσκευασίας (packing) της μορφής «best fit decreasing" με ελέγχους «savings» για την παραγωγή των διαδρομών. Πιο συγκεκριμένα, κατατάσσονται σε φθίνουσα σειρά οι συνολικές παραγγελίες και οι χωρητικότητες των

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

οχημάτων, κάθε παραγγελία ανατίθεται σε ένα όχημα, λαμβάνοντας υπόψιν περιορισμούς χωρητικότητας, χρόνου εργασίας του οδηγού και κόστους διαδρομής (saving) και η διαδικασία τερματίζεται όταν έχουν αντιστοιχηθεί όλες οι παραγγελίες.

Επιπλέον, για την επίλυση του ίδιου προβλήματος αναπτύχθηκε μία ακριβής μέθοδος, κατά την οποία το πρόβλημα αντιμετωπίζεται σαν «set partitioning» με συμπληρωματικούς περιορισμούς και αποτελείται από τα εξής ταυτόχρονα στάδια: δημιουργία πίνακα πελατών-διαδρομών (στήλες) και έλεγχοι προεπεξεργασίας για κατάργηση στηλών, ώστε να μειωθεί το μέγεθος του πίνακα. Οι έλεγχοι προεπεξεργασίας αφορούν τη χωρητικότητα του οχήματος, των δεξαμενών και του χρόνου ταξιδιού. Οι συμπληρωματικοί περιορισμοί αφορούν τη μοναδικότητα επίσκεψης σε κάθε πελάτη και τη μη υπέρβαση του χρόνου εργασίας του οδηγού. Για την επίλυση του προβλήματος χρησιμοποιείται η μέθοδος «branch and price», η οποία χρησιμοποιεί ως αρχικό υποσύνολο διαδρομών, εκείνες που προκύπτουν από την προηγούμενη ευρετική μέθοδο και βρίσκει τη βέλτιστη λύση προσθέτοντας μόνο ορισμένες μεταβλητές-στήλες. Και οι δύο μέθοδοι εφαρμόστηκαν στα πραγματικά δεδομένα της εταιρίας και προέκυψε ότι είναι αποτελεσματικές, με την ακριβή μέθοδο να οδηγεί σε καλύτερα αποτελέσματα.

Παράλληλα, οι Cornillier et al., 2008, μελέτησαν το πρόβλημα ανεφοδιασμού πρατηρίων καυσίμων (petrol station replenishment problem-PSRP) μέσω ετερογενούς στόλου βυτιοφόρων οχημάτων με διαμερίσματα. Αναπτύχθηκε ένας ακριβής αλγόριθμος ο οποίος διακρίνει το πρόβλημα σε δύο υποπροβλήματα: πρόβλημα φόρτωσης βυτιοφόρων οχημάτων και πρόβλημα δρομολόγησης. Για την επίλυση του προβλήματος φόρτωσης χρησιμοποιήθηκε ένας αλγόριθμος αντιστοίχισης και μοντέλα ακέραιου-γραμμικού προγραμματισμού. Ως αντικειμενική συνάρτηση τέθηκε η μεγιστοποίηση της συνολικής ποσότητας που παραδίδεται στις δεξαμενές των πρατηρίων από ένα όχημα με τους εξής περιορισμούς: η ποσότητα φόρτωσης δεν πρέπει να υπερβαίνει τη χωρητικότητα του εκάστοτε διαμερίσματος και το πολύ μία ζητούμενη ποσότητα μπορεί να ανατεθεί σε ένα διαμέρισμα, λόγω απουσίας μετρητών στάθμης.

Ο ακριβής αλγόριθμος αποτελείται από τα ακόλουθα βήματα. Στο πρώτο βήμα αναγνωρίζονται οι ανέφικτες περιπτώσεις, δηλ. όταν το άθροισμα των απαιτούμενων ποσοτήτων παράδοσης για όλες τις δεξαμενές υπερβαίνει τη συνολική χωρητικότητα των διαμερισμάτων του οχήματος. Στο δεύτερο βήμα εφαρμόζεται μία ευρετική μέθοδος αντιστοίχισης της ζήτησης στα διαμερίσματα, και σε περίπτωση που η ζητούμενη ποσότητα υπερβαίνει τη χωρητικότητα ενός διαμερίσματος χωρίζεται σε δύο επιμέρους. Εάν μέσω αυτής της διαδικασίας κάποιες ζητούμενες ποσότητες δεν μπορούν να ανατεθούν σε κάποιο διαμέρισμα, τότε το πρόβλημα δεν μπορεί να επιλυθεί μέσω της ευρετικής μεθόδου και εφαρμόζεται το έκτο βήμα, αλλιώς η λύση θεωρείται εφικτή και οι διαχωρισμοί που έχουν γίνει για τις ζητούμενες ποσότητες των δεξαμενών διατηρούνται για το επόμενο βήμα. Στο τρίτο βήμα εφαρμόζεται ένας

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

αλγόριθμος αντιστοίχισης με σκοπό την ελαχιστοποίηση της συνολικής αχρησιμοποίητης χωρητικότητας και στο τέταρτο βήμα ανατίθενται οι υπολειπόμενες ζητούμενες ποσότητες σε αχρησιμοποίητα διαμερίσματα. Στο πέμπτο βήμα πραγματοποιείται έλεγχος για το εάν η λύση είναι βέλτιστη, όπου και τερματίζεται ο αλγόριθμος, αλλιώς εφαρμόζεται ο solver του ακέραιου γραμμικού προγραμματισμού. Ο αλγόριθμος εφαρμόστηκε σε τυχαία παραγόμενα δεδομένα και σε μία πραγματική μελέτη περίπτωσης, επιλύθηκε μέσω Objective-C (και GLPK 4.2 για τον ακέραιο γραμμικό προγραμματισμό) και προέκυψε ότι οδηγεί σε βέλτιστες λύσεις.

Τέλος, οι Coelho & Laporte., 2015, μελέτησαν το πρόβλημα διανομής προϊόντων πετρελαίου με ομοιογενή βυτιοφόρα οχήματα πολλαπλών διαμερισμάτων. Αναφορικά με τα οχήματα διακρίνονται οι εξής περιπτώσεις: ένα βυτιοφόρο όχημα μπορεί να είναι ή όχι εξοπλισμένο με μετρητή στάθμης, άρα το περιεχόμενο ενός διαμερίσματος του μπορεί να χωριστεί σε παραγγελίες για διαφορετικούς πελάτες (split compartments) ή να αδειάζει πλήρως σε κάθε παράδοση (unsplit compartments) αντίστοιχα. Αναφορικά με τους πελάτες, ένας πελάτης μπορεί να επιτρέπει (split tanks) ή όχι (unsplit tanks) σε διαφορετικά οχήματα να γεμίζουν την ίδια δεξαμενή σε μία δεδομένη περίοδο. Από το συνδυασμό των παραπάνω περιπτώσεων προέκυψαν τέσσερα υποπροβλήματα, τα οποία διαμορφώθηκαν ως μοντέλα μεικτού ακέραιου γραμμικού προγραμματισμού. Ως αντικειμενική συνάρτηση τέθηκε η ελαχιστοποίηση του συνολικού κόστους διαδρομής και διατήρησης αποθεμάτων και κάποιοι από τους περιορισμούς ήταν οι εξής: μη υπέρβαση της χωρητικότητας κάθε διαμερίσματος, αντιστοίχιση ενός μόνο προϊόντος σε κάθε διαμέρισμα, το επίπεδο αποθέματος να μην υπερβαίνει τη χωρητικότητα της δεξαμενής του πελάτη. Για την επίλυση των μοντέλων χρησιμοποιήθηκε η μέθοδος «brunch and cut» από την οποία προέκυψαν βέλτιστες λύσεις.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

3

Θεωρητικό υπόβαθρο

3.1 Εφοδιαστική αλυσίδα υγρών καυσίμων

3.1.1 Διαδικασία παραλαβής καυσίμων από τα διυλιστήρια

Η εφοδιαστική αλυσίδα των υγρών καυσίμων αποτελεί ένα πολύπλοκο σύστημα με πολλούς εμπλεκόμενους φορείς. Στην Ελλάδα, οι εταιρίες διανομής καυσίμων προμηθεύονται τα προϊόντα τους κυρίως από τα τέσσερα ελληνικά διυλιστήρια (Ασπροπύργου, Ελευσίνας, Κορίνθου και Θεσσαλονίκης). Οι εταιρίες εμπορίας διαθέτουν εγκαταστάσεις αποθήκευσης είτε κοντά στα διυλιστήρια, είτε σε απομακρυσμένες περιοχές (σε νησιά ή εν γένει παραθαλάσσιες, απαραίτητες δεδομένης της ιδιομορφίας της γεωγραφίας της Ελλάδας). Επομένως, τα προϊόντα μεταφέρονται από τις δεξαμενές των διυλιστηρίων στις δεξαμενές αποθήκευσης στις εγκαταστάσεις των εταιριών διανομής και μεταπώλησης. Η μεταφορά γίνεται είτε μέσω συστήματος αγωγών απευθείας από τα διυλιστήρια για τις κοντινές εγκαταστάσεις, είτε μέσω δεξαμενόπλοιων, είτε σπανιότερα μέσω βυτιοφόρων. Οι κύριοι λόγοι κατασκευής αυτών των εγκαταστάσεων αποθήκευσης είναι η αυτονομία των εταιριών εμπορίας, η δυνατότητα προϊοντικής διαφοροποίησης και η βελτιωμένη εξυπηρέτηση του δικτύου πελατών.

Οι εγκαταστάσεις αποθήκευσης διαθέτουν 4-30 μεταλλικές δεξαμενές, με χωρητικότητα από 300-10.000 m^3 στις οποίες αποθηκεύονται τα προϊόντα. Η συνολική χωρητικότητα των εγκαταστάσεων συνήθως κυμαίνεται από 5.000-50.000 m^3. Οι εγκαταστάσεις διαθέτουν 1-12 νησίδες με γεμιστήρα τροφοδοσίας των βυτιοφόρων, μέσω των οποίων ανεφοδιάζονται για να μεταφέρουν τα προϊόντα στα πρατήρια υγρών καυσίμων.

3.1.2 Διαδικασία φόρτωσης των βυτιοφόρων οχημάτων από τις εγκαταστάσεις των εταιριών

Μετά την είσοδο του στο χώρο της εγκατάστασης, το βυτιοφόρο όχημα οδηγείται στην πλάστιγγα όπου ζυγίζεται και καταγράφεται το απόβαρο του (σε κιλά). Στη συνέχεια, οδηγείται στους σταθμούς φόρτωσης (loading stations-γεμιστήρια), όπου ισχύουν αυστηρές διατάξεις ασφαλείας για την αποφυγή

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ατυχήματος (απαγορεύεται η ύπαρξη «γυμνής φλόγας», γείωση βυτιοφόρου, ύπαρξη πυροσβεστικών μέσων, διακοπή φόρτωσης σε ηλεκτρικά φορτισμένη ατμόσφαιρα κ.τ.λ.). Μετά το άνοιγμα της αντλίας, πραγματοποιείται η πλήρωση των διαμερισμάτων του βυτιοφόρου με τα επιθυμητά προϊόντα στην κατάλληλη ποσότητα. Η φόρτωση γίνεται είτε από τα στόμια που βρίσκονται στο άνω μέρος των δεξαμενών του βυτιοφόρου (top loading) αφού πρώτα ελεγχθούν ότι είναι κλειστές οι κάτω βάνες του βυτιοφόρου, είτε από τις κάτω βάνες του βυτιοφόρου (bottom loading). Πιο συγκεκριμένα, τα συστήματα φόρτωσης διαφέρουν, από απλά συστήματα ενός βραχίονα με φόρτωση από την οροφή του βυτιοφόρου οχήματος σε πιο εξελιγμένα συστήματα, με αντλίες που ελέγχονται με αυτοματισμούς, διαθέτουν ανάκτηση ατμών, προστασία από υπερπλήρωση και μπορούν να φορτώνουν ταυτόχρονα 4 ή 5 προϊόντα.

Στη συνέχεια, το βυτιοφόρο οδηγείται εκ νέου στην πλάστιγγα, όπου προκύπτει το καθαρό βάρος του (μεικτό μείον απόβαρο) και μετατρέπεται σε λίτρα, ώστε να γίνει η χρέωση του προϊόντος που παρέλαβε το βυτιοφόρο. Τα βυτιοφόρα λαμβάνουν σφραγισμένα δείγματα των προϊόντων που βρίσκονται στα διαμερίσματά τους για αγορανομικούς λόγους και στη συνέχεια τα διαμερίσματα σφραγίζονται (καπάκια και βάνες).

3.1.3 Διαδικασία παράδοσης καυσίμων από τα βυτιοφόρα οχήματα προς τα πρατήρια υγρών καυσίμων

Στη συνέχεια, από τις εγκαταστάσεις αποθήκευσης της κάθε εταιρίας διανομής τα προϊόντα κατευθύνονται, μέσω βυτιοφόρων οχημάτων, σε διαφορετική διαδρομή ανάλογα με τον πελάτη. Στην Ελλάδα, η χωρητικότητα είναι συνήθως 30.000-33.000 lt για τα επικαθήμενα ή συρόμενα οχήματα και σπανιότερα από 10.000 έως 18.000 lt, κυρίως για τα βυτιοφόρα ιδιωτικής χρήσης. Το βυτίο μεταφοράς του οχήματος είναι χωρισμένο σε 4- 11 διαμερίσματα χωρητικότητας από 2.000-5.000 lt. Με αυτό τον τρόπο δίνεται η δυνατότητα στα οχήματα να μεταφέρουν κάθε φορά περισσότερα από ένα προϊόντα, αλλά και να παραδίδουν τμηματικά μικρότερα μέρη του φορτίου, μειώνοντας τον κίνδυνο για την ευστάθεια του οχήματος.

Κατά τη διαδικασία παράδοσης καυσίμων από τα βυτιοφόρα προς τα πρατήρια λαμβάνονται διάφορα μέτρα ασφαλείας (π.χ. το προσωρινό κλείσιμο του πρατηρίου για τους πελάτες, η αποφυγή ύπαρξης «γυμνής φλόγας», η γείωση του βυτιοφόρου). Ο πρατηριούχος ελέγχει ότι οι βάνες και τα καπάκια είναι σφραγισμένα, ελέγχει δια της ογκομετρικής βέργας αν όντως η ποσότητα που περιέχεται στο διαμέρισμα είναι αυτή που αναγράφεται στο δελτίο αποστολής (η ογκομετρική βέργα είναι μία λεπτή διαβαθμισμένη ράβδος που αναφέρεται σε κάθε διαμέρισμα) και ξεκινάει η διαδικασία παράδοσης.

Ξεσφραγίζονται οι βάνες των επιθυμητών διαμερισμάτων και τα αντίστοιχα καπάκια (για να μην δημιουργηθεί υποπίεση μέσα στα διαμερίσματα) και ο οδηγός του βυτιοφόρου ενώνει με σωλήνα τις κάτω βάνες των διαμερισμάτων με τις βάνες των δεξαμενών του πρατηρίου, τις ανοίγει και ξεκινάει η παράδοση. Τέλος, ο πρατηριούχος μετράει την ποσότητα των προϊόντων με βέργες μέσα στις δεξαμενές του.

Συνοψίζοντας, από τις δεξαμενές των εγκαταστάσεων αποθήκευσης των εταιριών εμπορίας τα βυτιοφόρα των εταιριών διανομής (Ιδιόκτητα ή Δημόσιας Χρήσης) φορτώνουν τα καύσιμα και τα παραδίδουν στα πρατήρια υγρών καυσίμων, όπου διαμέσου ελαστικών εύκαμπτων σωληνώσεων που βρίσκονται πάνω στο όχημα, τα καύσιμα μεταφέρονται στις υπόγειες συνήθως δεξαμενές αποθήκευσης. Τέλος, τα πρατήρια πωλούν μέσω των αντλιών τους τα καύσιμα στους καταναλωτές.

3.2 Περιγραφή του προβλήματος φόρτωσης βυτιοφόρου οχήματος

Μία εταιρεία η οποία διαθέτει διαφορετικά προϊόντα (τύπους καυσίμων) πραγματοποιεί διανομές καυσίμων σε πρατήρια υγρών καυσίμων-πελάτες. Η εταιρεία διαθέτει μία αποθήκη και έναν στόλο βυτιοφόρων οχημάτων για τον εφοδιασμό των πρατηρίων. Κάθε βυτιοφόρο όχημα αποτελείται από διαμερίσματα με διαφορετικές χωρητικότητες. Η εταιρεία έχει ένα σύνολο πελατών οι οποίοι αποστέλλουν παραγγελίες. Κάθε πελάτης πραγματοποιεί μία παραγγελία, η οποία συγκροτείται από τα παραγγελθέντα προϊόντα, δηλαδή τις ποσότητες ανά προϊόν της εταιρείας διανομής καύσιμων. Η εταιρεία έχει τη δυνατότητα να προσαρμόσει την ποσότητα που αναθέτει στα διαμερίσματα ενός οχήματος ως προς την ποσότητα του παραγγελθέντος προϊόντος που ζητείται από τον πελάτη τηρώντας δεδομένα ποσοστά αποδεκτής απόκλισης.

Κάθε βυτιοφόρο όχημα του στόλου της εταιρίας ακολουθεί μία δεδομένη διαδρομή επίσκεψης σε πελάτες, η οποία έχει ως αφετηρία και προορισμό την αποθήκη, και έχει προκύψει είτε από τον αποστολέα (dispatcher) είτε μέσω αλγορίθμων δρομολόγησης. Δημιουργείται, συνεπώς, η ανάγκη να αναπτυχθεί ένα μοντέλο, το οποίο θα επιλύει το πρόβλημα φόρτωσης, δηλαδή θα εξετάζει πως μπορεί να πραγματοποιηθεί η βέλτιστη δυνατή φόρτωση σε ένα βυτιοφόρο όχημα, ώστε το δρομολόγιο να είναι παράλληλα εφικτό ως προς κάποιους πρακτικούς περιορισμούς.

Η λύση του προβλήματος φόρτωσης καθορίζει το μέρος της ποσότητας (σε όγκο) του προϊόντος μίας παραγγελίας ενός πελάτη το οποίο ανατίθεται σε ένα διαμέρισμα του βυτιοφόρου οχήματος. Στόχος είναι η ελαχιστοποίηση της συνολικής απόκλισης ανάμεσα στην ποσότητα των προϊόντων που ανατίθενται στα διαμερίσματα του οχήματος και στη συνολική ποσότητα των παραγγελθέντων προϊόντων που ζητούνται από τους πελάτες και του συνολικού εναπομείναντα χώρου των διαμερισμάτων που χρησιμοποιήθηκαν. Το πρόβλημα υπόκειται στους εξής περιορισμούς:

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

- ✓ Μη υπέρβαση της χωρητικότητας σε όγκο ενός διαμερίσματος.
- ✓ Μη υπέρβαση του μέγιστου επιτρεπόμενου βάρους του οχήματος.
- ✓ Ανάθεση ενός μόνο τύπου καυσίμου σε ένα διαμέρισμα.
- ✓ Τήρηση του κανόνα ταλάντωσης (διακύμανσης) υγρού καυσίμου της αξιοποίησης του 20%/80% της χωρητικότητας κάθε διαμερίσματος του φορτηγού.
- ✓ Τήρηση των ορίων αποδεκτής μείωσης/αύξησης επί της ποσότητας παραγγελίας.

Για το πρόβλημα της φόρτωσης του βυτιοφόρου οχήματος διακρίνονται δύο περιπτώσεις. Στην πρώτη περίπτωση το βυτιοφόρο όχημα δεν είναι εξοπλισμένο με μετρητές στάθμης, δηλαδή σε κάθε διαμέρισμα μπορεί να ανατεθεί το πολύ μέχρι ένα παραγγελθέν προϊόν. Στη δεύτερη περίπτωση, το βυτιοφόρο όχημα είναι εξοπλισμένο με μετρητές στάθμης, δηλαδή πολλαπλά παραγγελθέντα προϊόντα του ίδιου καυσίμου μπορούν να ανατεθούν σε οποιοδήποτε διαμέρισμα.

3.3 Μαθηματικά μοντέλα φόρτωσης βυτιοφόρου οχήματος

Για την επίλυση του προβλήματος φόρτωσης ενός βυτιοφόρου οχήματος αναπτύσσονται δύο μοντέλα μεικτού ακέραιου γραμμικού προγραμματισμού (Mixed Integer Linear Programming-MILP) βάσει των Androutsopoulos & Karouti, 2021. Στο πρώτο μοντέλο (M1) γίνεται η υπόθεση ότι τα διαμερίσματα του οχήματος δεν είναι εξοπλισμένα με μετρητές στάθμης, ενώ στο δεύτερο μοντέλο (M2) γίνεται η υπόθεση ότι τα διαμερίσματα του οχήματος διαθέτουν μετρητές στάθμης.

Για τη δημιουργία των δύο μοντέλων μεικτού ακέραιου προγραμματισμού ορίζονται τα παρακάτω σύνολα, δείκτες και παράμετροι:

<u>Σύνολα (Sets) & Δείκτες (Indices)</u>

- c: δείκτης για κάθε διαμέρισμα (compartment)
- K_v: το σύνολο των διαμερισμάτων του οχήματος-v
- R: Διαδρομή
- i: παραγγελία (του πελάτη) i ∈ R
- i_0: αφετηρία (αποθήκη)
- i_{n+1}: προορισμός (αποθήκη)
- Π_i: το σύνολο των παραγγελθέντων προϊόντων της παραγγελίας i
- r: προϊόν της παραγγελίας i, $r \in \Pi_i$
- Π: το σύνολο όλων των παραγγελθέντων προϊόντων ($\cup i\ \Pi_i$)
- P: το σύνολο όλων των τύπων καυσίμου
- Π_p: το σύνολο των παραγγελθέντων προϊόντων που εμπεριέχουν το καύσιμο $p \in P$

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

- $p_r \in P$: το καύσιμο που εμπεριέχεται στο παραγγελθέν προϊόν r
- A_R: το σύνολο των συνδέσμων μεταξύ των πελατών της διαδρομής R

Παράμετροι (Parameters)

- $B > 0$: μέγιστο επιτρεπτό καθαρό βάρος του οχήματος v
- $Q_c > 0, c \in K_v$: μέγιστη χωρητικότητα (σε όγκο) του διαμερίσματος c
- $\beta_r^+ \in [0,1), r \in \Pi$: ποσοστό αποδεκτής αύξησης στην ποσότητα του προϊόντος r
- $\beta_r^- \in [0,1), r \in \Pi$: ποσοστό αποδεκτής μείωσης στην ποσότητα του προϊόντος r
- $\pi_r, r \in \Pi$: συνολική ζητούμενη ποσότητα (σε όγκο) του προϊόντος r
- $\rho_{p_r}, p_r \in P, r \in \Pi$: ειδικό βάρος του καυσίμου που εμπεριέχεται στο παραγγελθέν προϊόν r

3.3.1 Ανάπτυξη του μοντέλου χωρίς μετρητές στάθμης (M1)

Λόγω της απουσίας των μετρητών στάθμης, σε κάθε διαμέρισμα μπορεί να ανατεθεί το πολύ ένα παραγγελθέν προϊόν. Το προτεινόμενο μαθηματικό μοντέλο (M1) περιλαμβάνει τις ακόλουθες μεταβλητές απόφασης (decision variables):

- $z_{rc} \in \{0,1\}, r \in \Pi, c \in K_v$: λαμβάνει την τιμή 1, εάν το προϊόν r ανατίθεται στο διαμέρισμα c και 0 διαφορετικά.
- $q_{rc} \geq 0, r \in \Pi, c \in K_v$: εκφράζει το μέρος της ποσότητας (σε όγκο) του προϊόντος r, το οποίο ανατίθεται στο διαμέρισμα c. Σημείωση: εάν η συνολική ποσότητα (σε όγκο) του προϊόντος r είναι αρκετά μεγάλη τότε ενδεχομένως να χρειαστούν περισσότερα του ενός διαμερίσματα για να φορτωθεί το προϊόν r στο φορτηγό.
- $w_{ijc} \geq 0, (i,j) \in A_R, c \in K_v$: εκφράζει το συνολικό βάρος του φορτίου στο διαμέρισμα c, κατά τη διάσχιση του συνδέσμου (i, j) της διαδρομής R από το φορτηγό.
- $e \geq 0$: εκφράζει το συνολικό εναπομείναντα χώρο των διαμερισμάτων που έχουν χρησιμοποιηθεί (ο χώρος των διαμερισμάτων, στα οποία δεν έχει ανατεθεί κάποιο προϊόν δε συμπεριλαμβάνεται).
- $u_r, r \in \Pi$: εκφράζει την απόκλιση (σε απόλυτη τιμή) ανάμεσα στην ποσότητα του παραγγελθέντος προϊόντος που ζητείται από τον πελάτη και την αντίστοιχη ποσότητα του προϊόντος που ανατίθεται στα διαμερίσματα του οχήματος.
- $\varphi_{1c}, \varphi_{2c} \in \{0,1\}, c \in K_v$: βοηθητικές μεταβλητές οι οποίες χρησιμοποιούνται για λόγους μοντελοποίησης (δηλαδή για την ενσωμάτωση των διαζευκτικών ανισοτήτων που εκφράζουν τους περιορισμούς της «ταλάντωσης» του καυσίμου).

Το μαθηματικό μοντέλο (M1) εκφράζεται μέσω των σχέσεων (1)-(16). Η σχέση (1) εκφράζει τη λεξικογραφική ελαχιστοποίηση των δύο αντικειμενικών συναρτήσεων υπό εξέταση: τη συνολική

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

απόκλιση ανάμεσα στην ποσότητα που ανατίθεται στα διαμερίσματα του οχήματος και στη συνολική ποσότητα που ζητείται (πρωτεύουσα αντικειμενική συνάρτηση) και το συνολικό εναπομείναντα χώρο των διαμερισμάτων που χρησιμοποιήθηκαν. Στην περίπτωση της λεξικογραφικής ελαχιστοποίησης, αρχικά εξετάζεται η πρωτεύουσα αντικειμενική συνάρτηση και προκύπτει η λύση που βελτιστοποιεί το πρώτο κριτήριο και σε περίπτωση που προκύψουν παραπάνω από μία λύσεις, η επιλογή πραγματοποιείται βάσει του δεύτερου κριτήριου.

<u>Αντικειμενική Συνάρτηση</u>:

$$Minlex(\sum_{r\in\Pi} u_r\,, e) \qquad (1)$$

Στην πράξη, η λεξικογραφική ελαχιστοποίηση αποτυπώνεται ως εξής: $\sum_{r\in\Pi} 1000 * u_r + 0{,}01 * e$.

Το μαθηματικό μοντέλο (M1) περιλαμβάνει τις ακόλουθες κατηγορίες περιορισμών:

<u>Βασικοί περιορισμοί φόρτωσης</u>:

$$\sum_{r\in\Pi} z_{rc} \leq 1, \forall\, c \in K_v \qquad (2)$$

$$\sum_{r\in\Pi} q_{rc} \leq \sum_{r\in\Pi} Q_c * z_{rc}, \forall\, c \in K_v \qquad (3)$$

$$\sum_{r\in\Pi} \sum_{c\in K_v} q_{rc} * \rho_r \leq B \qquad (4)$$

$$Q_c * z_{rc} \geq q_{rc}, \forall\, c \in K_v, r \in \Pi \qquad (5)$$

$$q_{rc} \geq z_{rc}\,, \forall\, c \in K_v, r \in \Pi \qquad (6)$$

$$\sum_{c\in K_v} \sum_{r\in\Pi} (z_{rc} * Q_c - q_{rc}) = e \qquad (7)$$

Ο περιορισμός (2) δηλώνει ότι σε κάθε διαμέρισμα μπορεί να ανατεθεί το πολύ μέχρι ένα παραγγελθέν προϊόν. Ο περιορισμός (3) καταδεικνύει ότι η ποσότητα σε όγκο ενός παραγγελθέντος προϊόντος, το οποίο ανατίθεται σε ένα διαμέρισμα c δεν πρέπει να υπερβαίνει τη χωρητικότητα του (σε όγκο). Ο περιορισμός (4) δηλώνει ότι το βάρος της συνολικής ποσότητας (σε όγκο), η οποία ανατίθεται σε ένα όχημα, δεν πρέπει να υπερβαίνει το μέγιστο επιτρεπόμενο καθαρό βάρος του οχήματος.

Οι περιορισμοί (5) και (6) προσδιορίζουν τη σχέση ανάμεσα στις μεταβλητές q_{rc} και z_{rc}. Εάν μη μηδενική ποσότητα ενός παραγγελθέντος προϊόντος r ανατίθεται σε ένα διαμέρισμα c (δηλαδή $q_{rc} > 0$), τότε ο περιορισμός (5) επιβάλει στη μεταβλητή z_{rc} να λάβει την τιμή 1. Από την άλλη πλευρά, εάν καμία ποσότητα του παραγγελθέντος προϊόντος r δεν ανατίθεται σε ένα διαμέρισμα c (δηλαδή $q_{rc} = 0$), τότε ο περιορισμός (6) επιβάλει στη μεταβλητή z_{rc} να λάβει την τιμή 0. Ο περιορισμός (7) προσδιορίζει το συνολικό εναπομείναντα χώρο (e), ο οποίος προέρχεται από τα χρησιμοποιούμενα διαμερίσματα.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Περιορισμοί προσαρμογής της φόρτωσης:

$$\sum_{c\in K_v} q_{rc} - \pi_r \leq u_r, \forall\, r \in\ \Pi \quad (8)$$

$$\pi_r - \sum_{c\in K_v} q_{rc} \leq u_r, \forall\, r \in\ \Pi \quad (9)$$

$$(1-\beta_r^-) * \pi_r \leq \sum_{c\in K_v} q_{rc} \leq (1+\beta_r^+) * \pi_r, \forall\, r \in\ \Pi \quad (10)$$

Οι περιορισμοί (8) και (9) προσδιορίζουν την απόκλιση u_r ανάμεσα στην ποσότητα του παραγγελθέντος προϊόντος ενός πελάτη που ανατίθεται και που ζητείται. Επιπλέον, ο περιορισμός (10) καταδεικνύει ότι η ποσότητα ενός παραγγελθέντος προϊόντος που ανατίθεται στα διαμερίσματα ενός φορτηγού μπορεί να μειωθεί έως $\beta_r^- * 100\%$ ή να αυξηθεί έως $\beta_r^+ * 100\%$ επί της ποσότητας παραγγελίας.

Καταγραφή του φορτίου των διαμερισμάτων (ανά τμήμα της διαδρομής):

$$w_{i_o i_1 c} = \sum_{r\in\Pi} q_{rc} * \rho_{r,}\, \forall\, c \in\ K_v \quad (11)$$

$$w_{i_{n_R} i_{n+1} c} = 0, \forall\, c \in\ K_v \quad (12)$$

$$w_{i_{k-1} i_k c} - \sum_{r\in\Pi_{i_k}} q_{rc} * \rho_r = w_{i_k i_{k+1} c}, \forall i_k \in R, c \in\ K_v \quad (13)$$

Οι περιορισμοί (11)-(13) εντοπίζουν το βάρος που έχει κάθε διαμέρισμα σε όλους τους συνδέσμους και ενημερώνονται κάθε φορά που γίνεται μία εκφόρτωση. Ο περιορισμός (11) προσδιορίζει το βάρος της ποσότητας του καυσίμου που ανατίθεται στο διαμέρισμα c (όπου ρ_r είναι το ειδικό βάρος του καυσίμου που εμπεριέχεται στο προϊόν r) πριν την έναρξη των παραδόσεων (δηλαδή στο σύνδεσμο (i_0, i_1)), ενώ ο περιορισμός (12) υποδηλώνει ότι δεν πρέπει να υπάρχει κανένα φορτίο στα διαμερίσματα όταν το όχημα επιστρέφει στην αποθήκη (όπου το n_R δείχνει τον τελευταίο πελάτη που επισκέπτεται το όχημα στη διαδρομή). Ο περιορισμός (13) δηλώνει ότι το βάρος του περιεχομένου οποιουδήποτε διαμερίσματος c, όταν το φορτηγό φεύγει από τον πελάτη i_k ισούται με το βάρος του περιεχομένου πριν την άφιξη στον πελάτη i_k μειωμένο κατά το βάρος της ποσότητας (εάν υπάρχει) στο διαμέρισμα c που παραδόθηκε στον πελάτη i_k.

Περιορισμοί ποσότητας «ταλάντωσης»-κατανομής καυσίμου:

$$\sum_{r\in\Pi} q_{rc} \geq 0{,}8 * Q_c * \varphi_{1c}, \forall\, c \in\ K_v \quad (14)$$

$$\sum_{r\in\Pi} q_{rc} + (\varphi_{2c} - 1) * Q_c \leq 0{,}2 * Q_c, \forall\, c \in\ K_v \quad (15)$$

$$\varphi_{1c} + \varphi_{2c} = 1, \forall\, c \in\ K_v \quad (16)$$

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Ο κανόνας της ταλάντωσης (διακύμανσης) υγρού καυσίμου της αξιοποίησης του 20%/80% της χωρητικότητας οποιουδήποτε διαμερίσματος του φορτηγού εκφράζεται μέσω των διαζευκτικών ανισοτήτων (14) και (15). Ο περιορισμός (14), όταν είναι ενεργός (δηλαδή το φ_{1c}=1) υποδηλώνει ότι η ποσότητα που ανατίθεται σε οποιοδήποτε διαμέρισμα c πρέπει να καλύπτει (τουλάχιστον) το 80% της χωρητικότητας (σε όγκο) του διαμερίσματος. Από την άλλη πλευρά, ο περιορισμός (15), όταν είναι ενεργός (δηλαδή το φ_{2c}=1) υποδηλώνει ότι η ποσότητα που ανατίθεται σε οποιοδήποτε διαμέρισμα c δεν πρέπει να καλύπτει περισσότερο από το 20% της χωρητικότητας (σε όγκο) του διαμερίσματος. Ο περιορισμός (16) δηλώνει ότι μόνο ένας εκ των περιορισμών (14) και (15) πρέπει να είναι ενεργός για κάθε διαμέρισμα του οχήματος.

3.3.2 Ανάπτυξη του μοντέλου με μετρητές στάθμης (M2)

Στο συγκεκριμένο μαθηματικό μοντέλο (M2) έχει γίνει η υπόθεση ότι τα διαμερίσματα του οχήματος είναι εξοπλισμένα με μετρητές στάθμης, συνεπώς πολλαπλά παραγγελθέντα προϊόντα του ίδιου καυσίμου μπορούν να ανατεθούν σε οποιοδήποτε διαμέρισμα. Στο μοντέλο αυτό προστίθεται το σύνολο $\Pi_p, p \in P$, το οποίο αποτελεί το σύνολο των παραγγελθέντων προϊόντων ενός συγκεκριμένου καυσίμου τύπου p. Επιπλέον, προστίθεται η μεταβλητή απόφασης $y_{pc} \in \{0,1\}, p \in P, c \in K_v$, η οποία λαμβάνει την τιμή 1 εάν το καύσιμο τύπου $p \in P$ ανατίθεται στο διαμέρισμα $c \in K_v$ και 0 αλλιώς.

Στην περίπτωση απουσίας μετρητών στάθμης, κάθε παράδοση περιλαμβάνει την εκφόρτωση ολόκληρου του περιεχομένου των αντίστοιχων διαμερισμάτων. Συνεπώς, εάν ο περιορισμός της «ταλάντωσης» του καυσίμου ικανοποιείται όταν το φορτηγό αναχωρεί από την αποθήκη, τότε θα ικανοποιείται για οποιοδήποτε τμήμα της διαδρομής. Όμως, στην περίπτωση της παρουσίας των μετρητών στάθμης και της ανάθεσης πολλαπλών ποσοτήτων ενός προϊόντος στο ίδιο διαμέρισμα, ο περιορισμός της ταλάντωσης καυσίμου πρέπει να ελέγχεται σε όλα τα τμήματα της διαδρομής (εκτός από το τελευταίο τμήμα από τον τελευταίο πελάτη στην αποθήκη). Άρα, οι μεταβλητές φ_{1c} και φ_{2c} οι οποίες χρησιμοποιήθηκαν στο μοντέλο (M1) αντικαθίστανται από τις μεταβλητές φ^1_{ijc} και φ^2_{ijc}, $(i,j) \in A_R, c \in K_v$, οι οποίες είναι βοηθητικές μεταβλητές για τους διαζευκτικούς περιορισμούς που εκφράζουν την ταλάντωση του καυσίμου σε κάθε τμήμα (i,j) της διαδρομής παράδοσης του φορτηγού.

Το προτεινόμενο μαθηματικό μοντέλο (M2) περιλαμβάνει την αντικειμενική συνάρτηση (1) και τις ακόλουθες κατηγορίες περιορισμών:

<u>Βασικοί περιορισμοί φόρτωσης</u>:

Ο περιορισμός (4) για το μέγιστο επιτρεπόμενο καθαρό βάρος του οχήματος και οι περιορισμοί (5), (6) για τη σχέση των μεταβλητών q_{rc} και z_{rc} παραμένουν οι ίδιοι και προστίθενται οι ακόλουθοι περιορισμοί:

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

$$\sum_{p\in P} y_{pc} \leq 1, \forall\, c \in K_v \quad (17)$$

$$|\Pi_P| * y_{pc} \geq \sum_{r\in\Pi_p} z_{rc}\,, \forall\, c \in K_v, p \in P \quad (18)$$

$$y_{pc} \leq \sum_{r\in\Pi_p} z_{rc}\,, \forall\, c \in K_v, p \in P \quad (19)$$

$$\sum_{r\in\Pi_p} q_{rc} \leq Q_c * y_{pc}\,, \forall\, c \in K_v, p \in P \quad (20)$$

$$\sum_{p\in P}\sum_{c\in K_v}\sum_{r\in\Pi_p}\left(y_{pc} * Q_c - q_{rc}\right) = e \quad (21)$$

Ο περιορισμός (17) δηλώνει ότι σε κάθε διαμέρισμα μπορεί να ανατεθεί μέχρι ένα καύσιμο. Ο περιορισμός (18) υποδηλώνει ότι εάν ένα καύσιμο τύπου p δεν ανατίθεται σε ένα διαμέρισμα (δηλαδή $y_{pc} = 0$), τότε κανένα παραγγελθέν προϊόν αυτού του τύπου καυσίμου δεν μπορεί να ανατεθεί στο διαμέρισμα c. Εάν κανένα παραγγελθέν προϊόν ενός τύπου καυσίμου δεν ανατίθεται σε ένα διαμέρισμα c, τότε ο περιορισμός (19) επιβάλει στη μεταβλητή y_{pc} να λάβει την τιμή 0. Ο περιορισμός (20) δείχνει ότι η συνολική ποσότητα διαφορετικών παραγγελθέντων προϊόντων του ίδιου καυσίμου που ανατίθενται σε ένα διαμέρισμα c δεν πρέπει να υπερβαίνουν τη χωρητικότητα (σε όγκο) του διαμερίσματος c. Τέλος, ο περιορισμός (21) αντικαθιστά τον περιορισμό (7) του μοντέλου (M1) και προσδιορίζει τον συνολικό εναπομείναντα χώρο των διαμερισμάτων στα οποία ανατίθεται ένας τύπος καυσίμου.

<u>Περιορισμοί προσαρμογής της φόρτωσης</u>:

Ίδιοι με τους περιορισμούς (8)-(10) του μοντέλου (M1).

<u>Καταγραφή του φορτίου των διαμερισμάτων (ανά τμήμα της διαδρομής)</u>:

Ίδιοι με τους περιορισμούς (11)-(13) του μοντέλου (M1).

<u>Περιορισμοί ποσότητας «ταλάντωσης»-κατανομής καυσίμου</u>:

Οι περιορισμοί (14)-(16) του μοντέλου (M1) αντικαθίστανται από τους ακόλουθους περιορισμούς:

$$w_{ijc} \geq 0{,}8 * Q_c * \left(\sum_{p\in P} y_{pc} * \rho_{p_r}\right) - \left(1 - \varphi^1_{ijc}\right) * Q_C, \forall\, (i,j) \in A_R, c \in K_v \quad (22)$$

$$w_{ijc} \leq 0{,}2 * Q_c * \left(\sum_{p\in P} y_{pc} * \rho_{p_r}\right) + \left(1 - \varphi^2_{ijc}\right) * Q_C, \forall\, (i,j) \in A_R, c \in K_v \quad (23)$$

$$\varphi^1_{ijc} + \varphi^2_{ijc} = 1, \forall\, (i,j) \in A_R, c \in K_v \quad (24)$$

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Οι περιορισμοί (22)-(24) δηλώνουν ότι σε κάθε διαμέρισμα, κατά τη διάρκεια ολόκληρης της διαδρομής παράδοσης του φορτηγού, πρέπει να γεμίζεται είτε λιγότερο από το 20% είτε περισσότερο από το 80% από τη χωρητικότητά του (σε όγκο). Εάν ο περιορισμός (22) είναι ενεργός (δηλαδή φ_{ijc}^{1}=1) τότε το βάρος του φορτίου μέσα στο διαμέρισμα c είναι τουλάχιστον το 80% της χωρητικότητας σε βάρος του διαμερίσματος (η οποία υπολογίζεται πολλαπλασιάζοντας τη χωρητικότητα σε όγκο του διαμερίσματος με το ειδικό βάρος του φορτωμένου καυσίμου). Παρομοίως, όταν ο περιορισμός (23) είναι ενεργός (δηλαδή φ_{ijc}^{2}=1) τότε επιβάλλει στο βάρος του φορτίου μέσα στο διαμέρισμα c να είναι ίσο ή λιγότερο με το 20% της χωρητικότητας σε βάρος του διαμερίσματος.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

4

Μοντελοποίηση των περιορισμών ευστάθειας

4.1 Καθορισμός των παραμέτρων του βυτιοφόρου οχήματος

Η συμπερίληψη των κανόνων ευστάθειας στα μοντέλα φόρτωσης είναι μείζονος σημασίας για την εξασφάλιση της ασφαλούς φόρτωσης των φορτηγών οχημάτων και εν γένει της οδικής ασφάλειας. Στα μαθηματικά μοντέλα που παρουσιάστηκαν στο κεφάλαιο 3 πρέπει να προστεθούν και οι περιορισμοί οι οποίοι αφορούν την εξασφάλιση της ευστάθειας του βυτιοφόρου οχήματος, το οποίο αποτελεί και το καινοτόμο στοιχείο της ανάλυσης της παρούσας διπλωματικής εργασίας. Μέχρι στιγμής, η εξασφάλιση της ευστάθειας γίνεται εμπειρικά μέσω του κανόνα ταλάντωσης υγρού καυσίμου για κάθε διαμέρισμα σε κάθε φάση της διαδρομής. Αξίζει να σημειωθεί ότι υπάρχει περίπτωση κατά την αρχική φόρτωση του βυτιοφόρου οχήματος, πριν την έναρξη του δρομολογίου, να ικανοποιούνται τα μέγιστα επιτρεπτά όρια βάρους κατ' άξονα αλλά μετά την εκφόρτωση κάποιου διαμερίσματος να αλλάξει η κατανομή του βάρους του φορτίου και να παραβιαστεί το όριο βάρους σε έναν άξονα.

Συνεπώς, πραγματοποιείται μία αναλυτική εκτίμηση του συνολικού βάρους του φορτίου σε κάθε διαμέρισμα για κάθε φάση του δρομολογίου, ώστε να διασφαλίζεται ότι δεν θα παραβιάζονται τα μέγιστα επιτρεπόμενα όρια βάρους των αξόνων και ότι το κέντρο βάρους του φορτίου θα βρίσκεται όσο το δυνατόν πλησιέστερα στο γεωμετρικό κέντρο βάρους του βυτιοφόρου οχήματος σε οποιοδήποτε τμήμα της διαδρομής.

Για την ένταξη, συνεπώς, στα παραπάνω μοντέλα των περιορισμών ευστάθειας, οι οποίοι είναι κοινοί και για τα δύο μοντέλα, επιλέγεται ως τύπος βυτιοφόρου οχήματος ένα τριαξονικό όχημα με κινητήρα, το οποίο αποτελείται από επτά διαμερίσματα με διαφορετικές διαστάσεις και χωρητικότητες (σε όγκο). Αναφορικά με τους άξονες, ο πρώτος καλείται διευθυντήριος (steering) και είναι αυτός που ορίζει την αλλαγή κατεύθυνσης του οχήματος. Οι πίσω άξονες αποτελούν ζεύγος αξόνων (δίδυμοι άξονες) με απόσταση 1,3 m, είναι οι κινητήριοι άξονες (driving) και είναι αυτοί που παρέχουν την κινητήρια δύναμη, συνδέονται με τον κινητήρα μέσω του διαφορικού, του άξονα μετάδοσης κίνησης και του

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

κιβωτίου ταχυτήτων και ωθούν το όχημα προς τα εμπρός ή προς τα πίσω πραγματοποιώντας την κίνηση. Το φορτίο επιμερίζεται στον μπροστινό άξονα και τους πίσω δίδυμους άξονες.

Τα χαρακτηριστικά του οχήματος απεικονίζονται στους παρακάτω πίνακες.

3 AXLE TANKER	
VEHICLE_GROSS_WEIGHT (kg)-BG	26000
VEHICLE_TARE (kg)-BT	9800
VEHICLE MAX NET (kg)-B	16000
VEHICLE COMPARTMENTS	7
L12 (m)	5,35
L23 (m)	1,3
AXLE1_POS (m)	1,245
AXLE2_POS (m)	6,595
AXLE3_POS (m)	7,895

Πίνακας 3: Γενικά χαρακτηριστικά βυτιοφόρου οχήματος

Compartment	Capacity (lt)	Dist_axle1 (m)	Dist_axle2 (m)	Dist_axle3 (m)	Start_pos (m)	End_pos (m)
1	3200	1,712	3,638	4,938	2,31	3,60
2	2450	2,854	2,496	3,796	3,60	4,59
3	3100	3,977	1,373	2,673	4,59	5,85
4	3450	5,301	0,049	1,349	5,85	7,24
5	2550	6,514	1,164	0,136	7,24	8,27
6	3350	7,707	2,357	1,057	8,27	9,63
7	3500	9,092	3,742	2,442	9,63	11,05

Πίνακας 4: Χαρακτηριστικά των διαμερισμάτων του βυτιοφόρου οχήματος

Το μήκος του βυτιοφόρου οχήματος (L_{TRUCK}) είναι 11,05m. Οι παράμετροι L_{12} και L_{13} ορίζουν την απόσταση μεταξύ του πρώτου-δεύτερου και του δεύτερου-τρίτου άξονα του οχήματος αντίστοιχα ($L_{12} = 5{,}35m$, $L_{23} = 1{,}3m$), ενώ οι παράμετροι Axle1_pos, Axle2_pos, Axle3_pos ορίζουν τη θέση του κάθε άξονα αντίστοιχα με σημείο αναφοράς τη μπροστινή ακμή του οχήματος (Axle1_pos = $1{,}245m$, Axle2_pos = $6{,}595m$ και Axle3_pos = $7{,}895m$).

Οι παράμετροι Dist_axle1, Dist_axle2 και Dist_axle3 ορίζουν την απόσταση του κέντρου κάθε διαμερίσματος (το οποίο ορίζεται στο μέσον του) από τον κάθε άξονα αντίστοιχα, ενώ οι παράμετροι Start_pos και End_pos ορίζουν την αρχή και το τέλος του διαμερίσματος με σημείο αναφοράς τη μπροστινή ακμή του οχήματος.

Γενικά, ως απόβαρο ορίζεται το βάρος του οχήματος χωρίς πλήρωμα ή φορτίο αλλά με την αποθήκη γεμάτη καύσιμα μέχρι και 90%, τα συνήθως φερόμενα εργαλεία και τον εφεδρικό τροχό. Το μικτό βάρος είναι το εκάστοτε πραγματικό βάρος του οχήματος μαζί με το φορτίο, το καθαρό βάρος είναι το

βάρος του φορτίου του οχήματος και το μέγιστο επιτρεπτό βάρος είναι το μέγιστο βάρος φορτωμένου οχήματος. Συνεπώς, η παράμετρος B, ορίζει το μέγιστο επιτρεπτό καθαρό βάρος του οχήματος ($B = 16tn$), η παράμετρος B^T ορίζει το βάρος του άδειου φορτηγού-απόβαρο ($B^T = 9{,}8tn$), και η παράμετρος B^G ορίζει το μέγιστο επιτρεπτό (μικτό) βάρος του έμφορτου οχήματος ($B^G = 26tn$).

Για απλοποίηση των υπολογισμών και προκειμένου να γίνει η ανάλυση των ροπών των δυνάμεων που ασκούνται στο βυτιοφόρο όχημα ορίζονται δύο θεωρητικοί κεντρικοί άξονες. Στην περίπτωση του μπροστινού άξονα, ο θεωρητικός άξονας ταυτίζεται με την κεντρική γραμμή του άξονα - άξονας (1), ενώ στην περίπτωση των πίσω δίδυμων αξόνων ο θεωρητικός άξονας προκύπτει ως το μέσον της απόστασης των κεντρικών γραμμών των δύο πίσω αξόνων - άξονας (4). Ορίζεται η απόσταση (L_{14}) μεταξύ του μπροστινού άξονα και του κέντρου των δίδυμων αξόνων, η οποία προκύπτει ως εξής: $L_{14} = L_{12} + \frac{L_{23}}{2} = 6m.$

Ταυτόχρονα, ορίζονται οι παράμετροι B^1 και B^4, οι οποίες καθορίζουν το βάρος του άδειου φορτηγού που φέρει ο μπροστινός άξονας και οι πίσω δίδυμοι άξονες αντίστοιχα. Τα βάρη B^1 και B^4 προκύπτουν μέσω των εξισώσεων ισορροπίας των ροπών για το βυτιοφόρο όχημα χωρίς φορτίο ως προς τα δύο σημεία ισορροπίας, τα οποία ταυτίζονται με τους θεωρητικούς κεντρικούς άξονες.

$$\Sigma M^1 = 0 \Rightarrow B^4 * L_{14} = B^T * \left(\frac{L_{TRUCK}}{2} - Axle1_pos\right) \Rightarrow B^4 * 6 = 9{,}8 * \left(\frac{11{,}05}{2} - 1{,}245\right) \Rightarrow B^4 = 6{,}991tn \quad (25)$$

$$\Sigma M^4 = 0 \Rightarrow B^1 * L_{14} = B^T * \left(Axle1_pos + L_{14} - \frac{L_{TRUCK}}{2}\right) \Rightarrow B^1 * 6 = 9{,}8 * \left(1{,}245 + 6 - \frac{11{,}05}{2}\right) \Rightarrow B^1 = 2{,}809tn \quad (26)$$

Πιο αναλυτικά, με σημείο αναφοράς τον άξονα (1), οι δυνάμεις που επενεργούν είναι η αντίδραση του βάρους στον άξονα (4) με απόσταση L_{14}, η οποία προσδίδει αρνητική ροπή και το βάρος του άδειου φορτηγού με απόσταση τη θέση του γεωμετρικού κέντρου βάρους μειωμένη κατά τη θέση του άξονα (1) με σημείο αναφοράς τη μπροστινή ακμή του οχήματος, η οποία προσδίδει θετική ροπή. Με σημείο αναφοράς τον άξονα (4), οι δυνάμεις που ενεργούν είναι η αντίδραση του βάρους στον άξονα (1) με απόσταση L_{14}, η οποία προσδίδει θετική ροπή και το βάρος του άδειου οχήματος με απόσταση τη θέση του άξονα (4) ($Axle1_pos + L_{14}$) μειωμένη κατά τη θέση του γεωμετρικού κέντρου βάρους του οχήματος, η οποία προσδίδει αρνητική ροπή.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

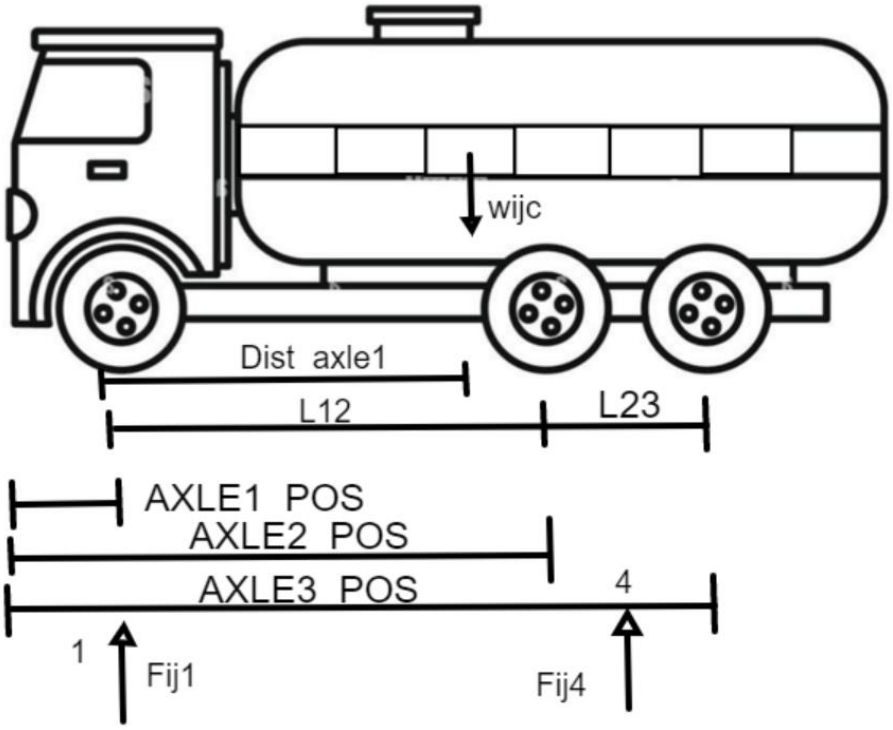
Παράλληλα, ορίζονται οι παράμετροι W_{max1} και W_{max2} , οι οποίες εκφράζουν το μέγιστο επιτρεπτό (καθαρό) βάρος φορτίου στον μπροστινό άξονα και στους πίσω δίδυμους άξονες αντίστοιχα. Οι παράμετροι αυτές προκύπτουν με βάσει τον κανονισμό για τα μέγιστα επιτρεπόμενα μεικτά βάρη ανά άξονα ή ομάδα αξόνων έμφορτου οχήματος για διεθνείς μεταφορές, όπως απεικονίζονται στον Πίνακα 5.

Αριθμός αξόνων	Κατηγορία αξόνων	Βάρος (kg)
Οχήματα με κινητήρα		
Απλός άξονας	Μη κινητήριος	10.000
	Κινητήριος άξονας	11.500
Δίδυμοι άξονες οχημάτων με κινητήρα	Με απόσταση α1,2 < 1 m.	11.500
	Με απόσταση 1 m ≤ α1,2 < 1,3 m.	16.000
	Με απόσταση 1,3 m ≤ α1,2 < 1,8 m.	18.000
Ρυμουλκούμενα και ημιρυμουλκούμενα		
Σύστημα δύο αξόνων	Με απόσταση α1,2 < 1 m.	11.000
	Με απόσταση 1 m ≤ α1,2 < 1,3 m.	16.000
	Με απόσταση 1,3 m ≤ α1,2 < 1,8 m.	18.000
	Με απόσταση α1,2 ≥ 1,8 m.	20.000
Σύστημα τριών αξόνων	Με απόσταση (α1,2, α2,3) ≤ 1,3 m.	21.000
	Με απόσταση 1,3 m < (α1,2, α2,3) ≤ 1,4 m.	24.000

Πίνακας 5: Μέγιστα βάρη ανά άξονα ή ομάδα αξόνων για διεθνείς μεταφορές

Συνεπώς, αναφορικά με τον μπροστινό άξονα, ο οποίος είναι διευθυντήριος και μη κινητήριος, το μέγιστο επιτρεπόμενο (καθαρό) βάρος του φορτίου-W_{max1} είναι $10.000 - B^1 = 7.191kg$. Αναφορικά με τους πίσω δίδυμους άξονες, οι οποίοι έχουν απόσταση 1,3 m το μέγιστο επιτρεπτό καθαρό βάρος του φορτίου- W_{max2} είναι $18.000 - B^4 = 11.009kg$.

Παρακάτω παρατίθεται ένα σχεδιάγραμμα του οχήματος για καλύτερη απεικόνιση των διαφόρων παραμέτρων:



Εικόνα 1: Τριαξονικό βυτιοφόρο όχημα

4.2 Ανάπτυξη περιορισμών ευστάθειας

Στην υποενότητα αυτή αναπτύσσονται οι περιορισμοί ευστάθειας, δηλαδή οι περιορισμοί αναφορικά με τα βάρη κατ' άξονα και την ομοιόμορφη κατανομή του βάρους του φορτίου. Για τη δημιουργία των περιορισμών ευστάθειας ορίζονται οι παρακάτω παράμετροι και μεταβλητές απόφασης:

Παράμετροι (Parameters)

- $x_c > 0$: απόσταση του κέντρου βάρους του διαμερίσματος c (το οποίο βρίσκεται στο μέσον του) από την μπροστινή πλευρά του βυτιοφόρου οχήματος ($x_c = \text{Axle1_pos} + \text{Dist_axle_1}$).
- $x_t > 0$ απόσταση του γεωμετρικού κέντρου βάρους του οχήματος στον άξονα x από την μπροστινή πλευρά του βυτιοφόρου οχήματος ($x_t = \frac{L_{TRUCK}}{2} = 5{,}525m$).
- $B^T > 0$: βάρος του άδειου φορτηγού.
- $B^1 > 0$: βάρος του άδειου φορτηγού που φέρει ο μπροστινός άξονας.
- $B^4 > 0$: βάρος του άδειου φορτηγού που φέρουν οι πίσω δίδυμοι άξονες.
- $W_{max1} > 0$: μέγιστο επιτρεπτό (καθαρό) βάρος φορτίου στον μπροστινό άξονα (διευθυντήριο).
- $W_{max2} > 0$: μέγιστο επιτρεπτό (καθαρό) βάρος φορτίου στους πίσω δίδυμους άξονες.
- $\text{Axle1_pos} > 0$: θέση του μπροστινού άξονα με σημείο αναφοράς τη μπροστινή ακμή του οχήματος.
- $\text{Dist_axle1} > 0$: απόσταση του κέντρου βάρους κάθε διαμερίσματος από τον μπροστινό άξονα.

- Start_pos και End_pos > 0: θέση αρχής και τέλους του κάθε διαμερίσματος με σημείο αναφοράς τη μπροστινή ακμή του οχήματος.
- $L_{14} > 0$: απόσταση μεταξύ του μπροστινού άξονα και του κέντρου των δίδυμων αξόνων.
- $\pi \in [0,1)$: ποσοστό αποδεκτής απόκλισης της επιμήκους απόστασης του κέντρου βάρους του φορτίου από το γεωμετρικό κέντρο βάρους του φορτηγού οχήματος.

Μεταβλητές απόφασης (decision variables)

$F_{ij}^1 \geq 0, (i,j) \in A_R$: εκφράζει το συνολικό βάρος του φορτίου στον μπροστινό άξονα στο σύνδεσμο (i,j).

$F_{ij}^4 \geq 0, (i,j) \in A_R$: εκφράζει το συνολικό βάρος του φορτίου στο κέντρο των πίσω δίδυμων αξόνων στο σύνδεσμο (i,j).

Εξισώσεις ισορροπίας:

$$\frac{1}{L_{14}} * \sum_{c \in K_v} (w_{ijc} * \text{Dist_axle1}) = F_{ij}^4, \forall\, (i,j) \in A_R \qquad (27)$$

$$\frac{1}{L_{14}} * \sum_{c \in K_v} (w_{ijc} * ((\text{Axle1_pos} + L_{14}) - (\text{Start_pos} + \text{End_pos})/2)) = F_{ij}^1, \forall\, (i,j) \in A_R \qquad (28)$$

Οι σχέσεις (27) και (28) προκύπτουν από τις εξισώσεις ισορροπίας ως προς τον άξονα (1) και τον άξονα (4) αντίστοιχα. Πιο συγκεκριμένα, με σημείο αναφοράς τον άξονα (1), οι δυνάμεις που επενεργούν είναι η αντίδραση στον άξονα (4) με απόσταση ίση με L_{14}, η οποία προσδίδει αρνητική ροπή και το βάρος του φορτίου κάθε διαμερίσματος με απόσταση Dist_axle1, η οποία προσδίδει θετική ροπή.

Με σημείο αναφοράς τον άξονα (4), οι δυνάμεις που επενεργούν είναι η αντίδραση στον άξονα (1) με απόσταση ίση με L_{14}, η οποία προσδίδει θετική ροπή και το βάρος του φορτίου κάθε διαμερίσματος με απόσταση ίση με την απόσταση του άξονα (4) από την μπροστινή πλευρά του βυτιοφόρου οχήματος ($\text{Axle1_pos} + L_{14} = 1{,}245 + 6 = 7{,}245m$) μειωμένη κατά τη θέση του κέντρου βάρους του κάθε διαμερίσματος. Τα διαμερίσματα των οποίων το κέντρο βάρους βρίσκεται αριστερά από τον άξονα (4) προσδίδουν αρνητική ροπή, ενώ τα διαμερίσματα των οποίων το κέντρο βάρους βρίσκεται δεξιά, θετική ροπή. Η διαφορά στο πρόσημο των δύο ροπών εξασφαλίζεται από το πρόσημο της τιμής της απόστασης, η οποία είναι αρνητική στην περίπτωση των διαμερισμάτων που βρίσκονται δεξιά από τον άξονα (4).

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Περιορισμοί ευστάθειας:

$$F_{ij}^4 \leq W_{max2}, \forall\ (i,j) \in A_R \quad (29)$$

Ο περιορισμός (29) δηλώνει ότι το συνολικό βάρος του φορτίου στο κέντρο των πίσω δίδυμων αξόνων δεν πρέπει να υπερβαίνει το αντίστοιχο μέγιστο επιτρεπτό (καθαρό) βάρος σε οποιοδήποτε σύνδεσμο.

$$F_{ij}^1 \leq W_{max1}, \forall\ (i,j) \in A_R \quad (30)$$

Ο περιορισμός (30) δηλώνει ότι το συνολικό βάρος του φορτίου στον μπροστινό άξονα δεν πρέπει να υπερβαίνει το αντίστοιχο μέγιστο επιτρεπτό (καθαρό) βάρος σε οποιοδήποτε σύνδεσμο.

$$F_{ij}^4 \geq -B^4, \forall\ (i,j) \in A_R \quad (31)$$

Ο περιορισμός (31) δηλώνει ότι το συνολικό βάρος του φορτίου στο κέντρο των πίσω δίδυμων αξόνων πρέπει να είναι τουλάχιστον ίσο με το αρνητικό βάρος του άδειου φορτηγού που φέρουν οι πίσω δίδυμοι άξονες σε οποιοδήποτε σύνδεσμο, ώστε να αποφευχθεί αρνητικό αξονικό βάρος στους πίσω άξονες.

$$F_{ij}^1 \geq -B^1, \forall\ (i,j) \in A_R \quad (32)$$

Ο περιορισμός (32) δηλώνει ότι το συνολικό βάρος του φορτίου στον μπροστινό άξονα πρέπει να είναι τουλάχιστον ίσο με το αρνητικό βάρος του άδειου φορτηγού που φέρει ο μπροστινός άξονας σε οποιοδήποτε σύνδεσμο, ώστε να αποφευχθεί αρνητικό αξονικό βάρος στον μπροστινό άξονα.

$$F_{ij}^4 + B^4 \geq 0{,}25 * (\sum_{c \in K_v} w_{ijc} + B^T), \forall\ (i,j) \in A_R \quad (33)$$

Ο περιορισμός (33) δηλώνει ότι το βάρος που φέρει ο κινητήριος άξονας ή οι κινητήριοι άξονες οποιουδήποτε τύπου οχήματος πρέπει να είναι τουλάχιστον ίσο με το 25% του συνολικού βάρους του έμφορτου οχήματος σύμφωνα με τη νομοθεσία.

$$\frac{\sum_{c \in K_v}(w_{ijc} * x_c)}{\sum_{c \in K_v} w_{ijc}} - x_t \leq \pi, \forall\ (i,j) \in A_R \quad (34)$$

$$x_t - \frac{\sum_{c \in K_v}(w_{ijc} * x_c)}{\sum_{c \in K_v} w_{ijc}} \leq \pi, \forall\ (i,j) \in A_R \quad (35)$$

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Οι περιορισμοί (34) και (35) δηλώνουν ότι το κέντρο βάρους του φορτίου πρέπει να βρίσκεται όσο το δυνατόν πλησιέστερα στο γεωμετρικό κέντρο βάρος του φορτηγού οχήματος κατά τον άξονα x, ώστε η κατανομή του βάρους να είναι ομοιόμορφη. Σύμφωνα με τη βιβλιογραφία, το αποδεκτό μέγιστο ποσοστό της απόκλισης της επιμήκους απόστασης μπορεί να είναι 1% (Bortfeldt & Gehring, 2001), 5% (Moon & Nguyen, 2014), 6% (Davies & Bischoff, 1999), 10% (Gehring & Bortfeldt, 1997) ή 50% (Liu et al., 2011).

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

5

Αξιολόγηση

5.1 Οργάνωση πειραμάτων

Τα μοντέλα φόρτωσης μεικτού ακέραιου γραμμικού προγραμματισμού που αναπτύχθηκαν στα προηγούμενα κεφάλαια μαζί με τους περιορισμούς ευστάθειας θα εφαρμοστούν σε ένα σύνολο δεδομένων ώστε να αναπτυχθούν διάφορα πειράματα και να αξιολογηθούν τα αποτελέσματά τους.

5.1.1 Σύνολο Δεδομένων

Το σύνολο των δεδομένων το οποίο χρησιμοποιήθηκε και στο οποίο εφαρμόστηκαν τα μοντέλα φόρτωσης, αποτελείται από 34 αρχεία τα οποία είναι σε μορφή excel. Οι θέσεις των πρατηρίων καυσίμων που χρησιμοποιήθηκαν σαν δεδομένα προέρχονται από έναν διανομέα καυσίμων, ενώ τα υπόλοιπα δεδομένα παρήχθησαν (generate) με τυχαίο τρόπο.

Κάθε αρχείο περιλαμβάνει, σε κάθε φύλο, τις εξής πληροφορίες:

- ✓ Αποστολές-πρατήρια (ship-to): κάθε πρατήριο έχει μοναδικό κωδικό (*ship-to id*) και συγκεκριμένες συντεταγμένες (γεωγραφικό πλάτος-*ship-to latitude* και γεωγραφικό μήκος-*ship to longitude*).
- ✓ Παραγγελίες (orders): κάθε παραγγελία αντιστοιχεί σε ένα πρατήριο και ο μοναδικός κωδικός της παραγγελίας (*order id*) ταυτίζεται με το μοναδικό κωδικό του πρατηρίου (*ship-to id*).
- ✓ Παραγγελθέντα προϊόντα (order items): κάθε παραγγελία αποτελείται από τουλάχιστον ένα προϊόν, το οποίο αντιστοιχεί σε έναν τύπο καυσίμου (*order item type*), έχει συγκεκριμένη ποσότητα παραγγελίας (*order item quantity*) σε λίτρα (*order item unit of measure*) και ειδικό βάρος (*order item specific weight*).
- ✓ Εγκαταστάσεις (premises): η αποθήκη, η οποία είναι κοινή για όλα τα αρχεία και αποτελεί το σημείο αφετηρίας και προορισμού για την κάθε διαδρομή του βυτιοφόρου οχήματος. Η αποθήκη έχει συγκεκριμένο κωδικό (*premise id*) και γεωγραφικές συντεταγμένες (γεωγραφικό πλάτος-*premise latitude* και γεωγραφικό μήκος-*premise longitude*).

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Κάθε επιμέρους αρχείο excel αντιστοιχεί στη διανομή καυσίμων που πραγματοποιεί το βυτιοφόρο όχημα σε συνολικά πέντε πρατήρια. Συνεπώς, συγκροτούνται πέντε παραγγελίες, μία για κάθε πρατήριο, κάθε μία από τις οποίες στο συγκεκριμένο σύνολο δεδομένων αποτελείται από ένα ή δύο παραγγελθέντα προϊόντα. Συνολικά, προκύπτει το εξής πλήθος παραγγελιών ανά ποσότητα παραγγελθέντων προϊόντων σε κάθε παραγγελία:

ORDER ITEMS	ORDERS
5 ORDER ITEMS	6
6 ORDER ITEMS	15
7 ORDER ITEMS	12
TOTAL	**33**

Πίνακας 6: Πλήθος παραγγελιών ανά ποσότητα παραγγελθέντων προϊόντων

Ενδεικτικά, παρουσιάζονται τα δεδομένα για τις εγκαταστάσεις και κάθε φύλλο δύο αρχείων excel, εκ των οποίων το ένα αποτελείται από πέντε (excel 1) και το άλλο από έξι (excel 2) παραγγελθέντα προϊόντα.

PREMISE_ID	PREMISE_LAT	PREMISE_LONG
00A4	38,03677	23,61218

Πίνακας 7:Δεδομένα για τις εγκαταστάσεις

SHIPTO_ID	SHIPTO_LAT	SHIPTO_LONG
505789	37,94802	23,68963
505867	38,01347	23,751
505215	38,043	23,70108
18014906	38,01139	23,83794
505596	37,85842	23,7649

Πίνακας 8: Δεδομένα για τις αποστολές-πρατήρια (excel 1)

ORDER_ID	SHIPTO_ID
505789	505789
505867	505867
505215	505215
18014906	18014906
505596	505596

Πίνακας 9: Δεδομένα για τις παραγγελίες (excel 1)

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ORDER_ID	ORDER_ITEM_TYPE	ORDER_ITEM_QUANTITY	ORDER_ITEM_UNIT_OF_MEASURE	ORDER_ITEM_SPECIFIC_WEIGHT
505789	10	3247	LT	0,68
505867	10	2816	LT	0,68
505215	10	2386	LT	0,68
18014906	10	5556	LT	0,68
505596	10	6324	LT	0,68

Πίνακας 10: Δεδομένα για τα παραγγελθέντα προϊόντα (excel 1)

SHIPTO_ID	SHIPTO_LAT	SHIPTO_LONG
18015040	37,93025	23,7098
507702	37,98871	23,71844
18011583	38,08931	23,80933
505686	37,91968	23,74405
505842	38,0177	23,68993

Πίνακας 11: Δεδομένα για τις αποστολές-πρατήρια (excel 2)

ORDER_ID	SHIPTO_ID
18015040	18015040
507702	507702
18011583	18011583
505686	505686
505842	505842

Πίνακας 12: Δεδομένα για τις παραγγελίες (excel 2)

ORDER_ID	ORDER_ITEM_TYPE	ORDER_ITEM_QUANTITY	ORDER_ITEM_UNIT_OF_MEASURE	ORDER_ITEM_SPECIFIC_WEIGHT
18015040	10	2901	LT	0,68
507702	10	5569	LT	0,68
18011583	10	3466	LT	0,68
18011583	20	3340	LT	0,68
505686	10	3168	LT	0,68
505842	10	2406	LT	0,68

Πίνακας 13: Δεδομένα για τα παραγγελθέντα προϊόντα (excel 2)

Αναφορικά με τα παραγγελθέντα προϊόντα (Πίνακας 10,13) το ειδικό βάρος των καυσίμων που διανέμονται λαμβάνει για όλα τα αρχεία δεδομένων την τιμή 0,68kg/lt, η οποία αντιστοιχεί σε καύσιμο βενζίνη, ενώ παράλληλα υπάρχουν δύο τύποι βενζίνης, οι οποίοι χαρακτηρίζονται ως «10» και «20».

5.1.2 Εύρεση διαδρομής

Κάθε αρχείο περιλαμβάνει τα πρατήρια που θα εφοδιάσει με καύσιμα το βυτιοφόρο όχημα σε μία διαδρομή, η οποία έχει ως αφετηρία και προορισμό την αποθήκη. Προκειμένου να βρεθεί η βέλτιστη σειρά επίσκεψης των πρατηρίων επιλύεται το πρόβλημα του περιοδεύοντος πωλητή (Traveling Salesman Problem) με στόχο, δεδομένων των αποστάσεων, να βρεθεί η διαδρομή ώστε:

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

- ✓ Η συνολική απόσταση που διανύει το όχημα να ελαχιστοποιείται.
- ✓ Το όχημα να επιστρέφει στην αποθήκη από όπου ξεκίνησε (κλειστή διαδρομή).
- ✓ Το όχημα να επισκέπτεται μία μόνο φορά το κάθε πρατήριο.

Το πρόβλημα επιλύεται βάσει του ευρεστικού αλγορίθμου του πλησιέστερου γείτονα (Nearest Neighbor), κατά τον οποίο με αφετηρία την αποθήκη, τοποθετείται επαναληπτικά, μετά το τελευταίο πρατήριο, το πρατήριο που θα οδηγήσει στη μικρότερη αύξηση της διανυόμενης απόστασης. Ο ευρεστικός αλγόριθμος του πλησιέστερου γείτονα έχει ως αποτέλεσμα μία καλή λύση και όχι απαραίτητα τη βέλτιστη. Η απόσταση (σε km) μεταξύ δύο σημείων με γεωγραφικές συντεταγμένες (φ_1, λ_1) και (φ_2, λ_2) αντίστοιχα υπολογίζεται μέσω των ακόλουθων τύπων:

$$a = sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_1 * \cos\varphi_2 * sin^2\left(\frac{\Delta\lambda}{2}\right) \quad (36)$$

$$c = 2 * atan2(\sqrt{a}, \sqrt{1-a}) \quad (37)$$

$$d = R * c \quad (38)$$

όπου τα φ_1, φ_2 αναπαριστούν τα γεωγραφικά πλάτη και τα λ_1, λ_2 τα γεωγραφικά μήκη αντίστοιχα, ενώ το R αναπαριστά την ακτίνα της γης σε χιλιόμετρα, η οποία ισούται με 6.371 km. Η συνάρτηση $atan2$ επιστρέφει την εφαπτομένη του πηλίκου των ορισμάτων της, ως αριθμητική τιμή μεταξύ των ακτινίων π και -π.

Μέσω, λοιπόν, της μεθόδου του Nearest Neighbor εντοπίζεται μία λύση αναφορικά με τη διαδρομή του βυτιοφόρου οχήματος και ως εκ τούτου τη σειρά εκφόρτωσης των παραγγελθέντων προϊόντων σε κάθε στάση-πρατήριο, η οποία είναι καθοριστική για τους περιορισμούς που αφορούν την ταλάντωση καυσίμου και την ευστάθεια. Στη συνέχεια, ελέγχεται με δεδομένη τη σειρά επίσκεψης στους πελάτες που προκύπτει, εάν τα μοντέλα φόρτωσης έχουν εφικτή λύση, ικανοποιώντας τους περιορισμούς που απαιτούνται. Στην περίπτωση που η εφαρμογή των μοντέλων δεν οδηγεί σε εφικτή λύση, τότε εξετάζεται μία εναλλακτική διαδρομή. Μετά την εφαρμογή των μοντέλων, προέκυψε ότι οι λύσεις είναι εφικτές για τα δρομολόγια που προκύπτουν από την εφαρμογή του Nearest Neighbor σε όλα τα αρχεία.

5.1.3 Ανάπτυξη σεναρίων

Για την ανάλυση των πειραμάτων γίνεται διάκριση σε τρία σενάρια τα οποία εφαρμόζονται τόσο στο μοντέλο φόρτωσης με τους μετρητές στάθμης, όσο και στο μοντέλο χωρίς τους μετρητές στάθμης. Τα σενάρια είναι τα εξής:

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

- ✓ **Σενάριο Α**: στο σενάριο αυτό περιλαμβάνονται οι περιορισμοί των μοντέλων Μ1 και Μ2, χωρίς τους περιορισμούς που αφορούν την ευστάθεια και τη θέση του κέντρου βάρους του φορτίου (περιορισμοί 29-35).
- ✓ **Σενάριο Β**: στο σενάριο αυτό περιλαμβάνονται οι περιορισμοί των μοντέλων Μ1 και Μ2 και οι περιορισμοί που αφορούν την ευστάθεια (περιορισμοί 29-33).
- ✓ **Σενάριο C**: στο σενάριο αυτό περιλαμβάνονται οι περιορισμοί των μοντέλων Μ1 και Μ2 καθώς και οι περιορισμοί που αφορούν την ευστάθεια και τη θέση του κέντρου βάρους του φορτίου (περιορισμοί 29-35).

Όλα τα σενάρια θα εφαρμοστούν για συγκεκριμένες τιμές των μεταβλητών β_r^+ και β_r^-, οι οποίες είναι οι εξής: 0, 0,025, 0,05, 0,075, 0,1 ,0,125, 0,15, 0,175 και 0,2 και κάθε φορά θα επιλέγεται η μικρότερη δυνατή. Αν κάποιο πρόβλημα δεν λύνεται για καμία από τις παραπάνω τιμές, δηλαδή μέχρι την τιμή 0,2, τότε θεωρείται αδύνατο (infeasible), καθώς ποσοστά αύξησης/μείωσης στις ποσότητες των προϊόντων μεγαλύτερα από 20% δεν είναι αποδεκτά στην πράξη.

5.1.4 Πλατφόρμες και προγραμματιστικά εργαλεία

Η φόρτωση των δεδομένων και η ανάπτυξη των μοντέλων πραγματοποιήθηκε μέσω της γλώσσας προγραμματισμού Java η οποία αναπτύχθηκε στο λογισμικό Eclipse IDE for Java Developers-2021. Η επίλυση των μοντέλων φόρτωσης πραγματοποιήθηκε μέσω του solver CPLEX. Το IBM ILOG CPLEX Optimization Studio 20.1.0 (ή πιο σύντομα Cplex) αποτελεί ένα λογισμικό πακέτο βελτιστοποίησης και το όνομα του προκύπτει από τη μέθοδο SimPLEX εφαρμοσμένη στη γλώσσα προγραμματισμού C. Αποτελεί ένα εργαλείο για την επίλυση προβλημάτων γραμμικού προγραμματισμού (linear programming-LP), μεικτού ακέραιου προγραμματισμού (mixed integer programming-MIP), χρησιμοποιώντας τη μέθοδο simplex, τετραγωνικού προγραμματισμού (quadratic programming-QP) και εν γένει για την αποδοτική αντιμετώπιση πολύπλοκων προβλημάτων βελτιστοποίησης. Αποτελείται από την τεχνολογία Concert, η οποία είναι βιβλιοθήκη κλάσης και προσφέρει μία Διεπαφή Προγραμματισμού Εφαρμογών-API (Application Programming Interface) που χρησιμοποιείται για να ενσωματώσει το βελτιστοποιητή CPLEX στη Java. Οι κλάσεις της τεχνολογίας Concert για τη Cplex εφαρμόζονται μέσω της Διεπαφής «Java Native»-JNI (Java Native Interface). Οι κλάσεις και οι μέθοδοι της Java που αναπτύχθηκαν παρουσιάζονται αναλυτικά στο παράρτημα.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

5.2 Αποτελέσματα

Ενδεικτικά, παρουσιάζονται και αναλύονται τα αποτελέσματα από την εφαρμογή των προαναφερθέντων σεναρίων (A,B,C) για το dataset το οποίο παρουσιάστηκε στους Πίνακες 11, 12 και 13, για το οποίο προέκυψε η εξής σειρά επίσκεψης των πρατηρίων: {5,2,1,4,3}.

5.2.1 Μοντέλο χωρίς μετρητές στάθμης

Σενάριο A: β_r^+ και $\beta_r^- = 0$

		ΣΕΝΑΡΙΟ A	Vehicle Compartments								
		Demand	1	2	3	4	5	6	7	Assigned Quantity	Deviation
Order Items	6	2406		2406						2406	0
	2	5569			3100		2469			5569	0
	1	2901				2901				2901	0
	5	3168	3168							3168	0
	3	3466							3466	3466	0
	4	3340						3340		3340	0
		Compartment load	3168	2406	3100	2901	2469	3340	3466	Total Assigned Quantity	20850
		Compartment Capacity	3200	2450	3100	3450	2550	3350	3500	Objective Function Value	7,5
		Compartment Remaining Space	32	44	0	549	81	10	34		
		Compartment Utilization (%)	99,00	98,20	100,00	84,09	96,82	99,70	99,03		
		Total order deviation	0								
		Total remaining space of used compartments	750								
		Percentage of extra total assigned quantity (A vs B) (%)	13,1								
		Percentage of total order deviation out of total demand (B) (%)	11,60								

Πίνακας 14: Μοντέλο χωρίς μετρητές στάθμης – Σενάριο A

Σενάριο B: β_r^+και $\beta_r^- = 0{,}125$

		ΣΕΝΑΡΙΟ B	Vehicle Compartments										
		Demand	1	2	3	4	5	6	7	Assigned Quantity	Deviation	Percentage of order deviation out of demand	Order Flexibility
Order Items	6	2406					2105			2105	301	0,125	301
	2	5569		2193				2680		4873	696	0,125	696
	1	2901			2538					2538	363	0,125	363
	5	3168							2800	2800	368	0,116	396
	3	3466	3193							3193	273	0,079	433
	4	3340				2923				2923	417	0,125	418
		Compartment load	3193	2193	2538	2923	2105	2680	2800	Total Assigned Quantity	18432		
		Compartment Capacity	3200	2450	3100	3450	2550	3350	3500	Objective Function Value	2417969,96		
		Compartment Remaining Space	7	257	562	527	445	670	700				
		Compartment Utilization (%)	99,78	89,51	81,87	84,72	82,55	80,00	80,00				
		Total order deviation	2418										
		Total remaining space of used compartments	3168										

Πίνακας 15: Μοντέλο χωρίς μετρητές στάθμης – Σενάριο B

Σενάριο C: Το μοντέλο είναι αδύνατο για τιμές παραμέτρων έως $\beta_r^+ / \beta_r^- = 0{,}20$.

Από τα αποτελέσματα προκύπτει η συνολική ζητούμενη ποσότητα (σε λίτρα) του κάθε παραγγελθέντος προϊόντος (Demand), η κατανομή του κάθε προϊόντος στο κάθε διαμέρισμα, η συνολική ποσότητα του κάθε προϊόντος που ανατίθεται στο όχημα (Assigned Quantity), η απόκλιση ανάμεσα στην ποσότητα του παραγγελθέντος προϊόντος που ζητείται από τον πελάτη και την αντίστοιχη ποσότητα που ανατίθεται στα διαμερίσματα του οχήματος (Deviation), το ποσοστό της απόκλισης αυτής ως προς τη ζήτηση (Percentage of order deviation out of demand) και την αποδεκτή απόκλιση της ποσότητας φόρτωσης ως προς τη ζήτηση (Order Flexibility).

Παράλληλα, παρουσιάζεται η συνολική ποσότητα που ανατίθεται σε κάθε διαμέρισμα (compartment load), ο εναπομείναν χώρος του κάθε διαμερίσματος (Compartment Remaining Space), το ποσοστό αξιοποίησης του κάθε διαμερίσματος (Compartment Utilization), η συνολική απόκλιση ανάμεσα στις ποσότητες όλων των παραγγελθέντων προϊόντων που ανατίθενται ως προς τη συνολική ζήτηση (Total order deviation) και ο συνολικός εναπομείναν χώρος όλων των χρησιμοποιούμενων διαμερισμάτων (Total remaining space of used compartments). Τέλος, παρουσιάζεται η συνολική ποσότητα φόρτωσης στο όχημα (Total Assigned Quantity) και η τιμή της αντικειμενικής συνάρτησης (Objective Function Value).

Πραγματοποιώντας σύγκριση των σεναρίων του μοντέλου χωρίς τους μετρητές στάθμης προκύπτουν τα εξής αποτελέσματα:

- ✓ Οι αποκλίσεις ανάμεσα στη συνολική ποσότητα φόρτωσης και τη συνολική ζήτηση είναι μηδέν στο σενάριο Α, άρα όλη η ζήτηση ικανοποιείται όταν δεν συμπεριλαμβάνονται οι περιορισμοί που αφορούν την ευστάθεια και τη θέση του κέντρου βάρους του φορτίου.
- ✓ Η συνολική ποσότητα φόρτωσης στο σενάριο Α είναι κατά 13,1% περισσότερη από τη συνολική ποσότητα φόρτωσης στο σενάριο Β (Percentage of extra total assigned quantity (A vs B)).
- ✓ Οι συνολικές αποκλίσεις ανάμεσα στη συνολική ποσότητα φόρτωσης και τη συνολική ζήτηση αποτελούν το 11,6% της συνολικής ζήτησης στο σενάριο Β (Percentage of total order deviation out of total demand (B).

5.2.2 Μοντέλο με μετρητές στάθμης

Σενάριο Α: β_r^+ και $\beta_r^- = 0$

		ΣΕΝΑΡΙΟ Α	Vehicle Compartments								
		Demand	1	2	3	4	5	6	7	Assigned Quantity	Deviation
Order Items	6	2406			2405	1				2406	0
	2	5569			1		2550	3018		5569	0
	1	2901				2901				2901	0
	5	3168	100	2449	619					3168	0
	3	3466	3100	1		365				3466	0
	4	3340							3340	3340	0
		Compartment load	3200	2450	3025	3267	2550	3018	3340	Total Assigned Quantity	20850
		Compartment Capacity	3200	2450	3100	3450	2550	3350	3500	Objective Function Value	731,5
		Compartment Remaining Space	0	0	75	183	0	332	160		
		Compartment Utilization (%)	100,00	100,00	97,58	94,70	100,00	90,09	95,43		
		Total order deviation	0								
		Total remaining space of used compartments	750								
		Percentage of extra total assigned quantity (A vs B) (%)	10,2								
		Percentage of extra total assigned quantity (A vs C) (%)	21,5								
		Percentage of less total assigned quantity (C vs B) (%)	-9,3								
		Percentage of total order deviation out of total demand (B) (%)	9,26								
		Percentage of total order deviation out of total demand (C) (%)	17,70								

Πίνακας 16: Μοντέλο με μετρητές στάθμης – Σενάριο Α

Σενάριο Β: β_r^+και $\beta_r^- = 0{,}10$

		ΣΕΝΑΡΙΟ Β	Vehicle Compartments										
		Demand	1	2	3	4	5	6	7	Assigned Quantity	Deviation	Percentage of order deviation out of demand	Order Flexibility
Order Items	6	2406	640		318	508			700	2166	240	0,10	241
	2	5569	2111			2941	1			5053	516	0,09	557
	1	2901							2611	2611	290	0,10	290
	5	3168			302		2549			2851	317	0,10	317
	3	3466	449		2480	1			189	3119	347	0,10	347
	4	3340		2450				670		3120	220	0,07	334
		Compartment load	3200	2450	3100	3450	2550	670	3500	Total Assigned Quantity	18920		
		Compartment Capacity	3200	2450	3100	3450	2550	3350	3500	Objective Function Value	1930658,8		
		Compartment Remaining Space	0	0	0	0	0	2680	0				
		Compartment Utilization (%)	100,00	100,00	100,00	100,00	100,00	20,00	100,00				
		Total order deviation	1930										
		Total remaining space of used compartments	2680										

Πίνακας 17: Μοντέλο με μετρητές στάθμης – Σενάριο Β

Σενάριο C: β_r^+ και $\beta_r^- = 0{,}20$ και $\pi = 0{,}30$.

Για το σενάριο C, παρουσιάζεται το παρακάτω διάγραμμα, στο οποίο απεικονίζεται η σχέση μεταξύ της αντικειμενική συνάρτησης και του ποσοστού απόκλισης του κέντρου βάρους του φορτίου από το γεωμετρικό κέντρο βάρους του φορτηγού οχήματος. Καθίσταται εμφανές ότι όσο αυξάνεται το ποσοστό, δηλαδή όσο πιο χαλαρός γίνεται ο περιορισμός της ομοιόμορφης κατανομής του βάρους, τόσο μειώνεται η αντικειμενική συνάρτηση, δηλ. βελτιώνεται η λύση (αυξάνεται η ποσότητα φόρτωσης). Η τιμή της αντικειμενικής συνάρτησης σταθεροποιείται σχετικά όταν το ποσοστό απόκλισης είναι 60%, το οποίο όμως είναι αρκετά μεγάλο και γι' αυτό επιλέγεται η ελάχιστη τιμή (π=30%), ώστε η κατανομή να είναι όσο το δυνατόν πιο ομοιόμορφη.



Διάγραμμα 1: Αντικειμενική συνάρτηση - ποσοστό απόκλισης του κέντρου βάρους του φορτίου από το γεωμετρικό κέντρο βάρους του οχήματος – Σενάριο C

		ΣΕΝΑΡΙΟ C	Vehicle Compartments										
		Demand	1	2	3	4	5	6	7	Assigned Quantity	Deviation	Percentage of order deviation out of demand	Order Flexibility
Order Items	6	2406	178				1769			1947	459	0,19	481
	2	5569	2382					2073		4455	1114	0,20	1114
	1	2901		1960			257	104		2321	580	0,20	580
	5	3168	598	490		690	253	503		2534	634	0,20	634
	3	3466	42			2760				2802	664	0,19	693
	4	3340			3100					3100	240	0,07	668
		Compartment load	3200	2450	3100	3450	2279	2680	0	Total Assigned Quantity	17159		
		Compartment Capacity	3200	2450	3100	3450	2550	3350	3500	Objective Function Value	3691503,928		
		Compartment Remaining Space	0	0	0	0	271	670	3500				
		Compartment Utilization (%)	100,00	100,00	100,00	100,00	89,37	80,00	0,00				
		Total order deviation	3691										
		Total remaining space of used compartments	941										

Πίνακας 18: Μοντέλο με μετρητές στάθμης – Σενάριο C

Πραγματοποιώντας σύγκριση των σεναρίων του μοντέλου με τους μετρητές στάθμης προκύπτουν τα εξής αποτελέσματα:

- ✓ Οι αποκλίσεις ανάμεσα στη συνολική ποσότητα φόρτωσης και τη συνολική ζήτηση είναι μηδέν στο σενάριο Α, άρα όλη η ζήτηση ικανοποιείται όταν δεν συμπεριλαμβάνονται οι περιορισμοί που αφορούν την ευστάθεια και τη θέση του κέντρου βάρους του φορτίου.
- ✓ Η συνολική ποσότητα φόρτωσης στο σενάριο Α είναι κατά 10,2% περισσότερη από τη συνολική ποσότητα φόρτωσης στο σενάριο Β.
- ✓ Η συνολική ποσότητα φόρτωσης στο σενάριο Α είναι κατά 21,5% περισσότερη από τη συνολική ποσότητα φόρτωσης στο σενάριο C.
- ✓ Η συνολική ποσότητα φόρτωσης στο σενάριο C είναι κατά 9,3% λιγότερη από τη συνολική ποσότητα φόρτωσης στο σενάριο Β (Percentage of less total assigned quantity (C vs B)).
- ✓ Οι συνολικές αποκλίσεις ανάμεσα στη συνολική ποσότητα φόρτωσης και τη συνολική ζήτηση αποτελούν το 9,26% της συνολικής ζήτησης στο σενάριο Β.
- ✓ Οι συνολικές αποκλίσεις ανάμεσα στη συνολική ποσότητα φόρτωσης και τη συνολική ζήτηση αποτελούν το 17,70% της συνολικής ζήτησης στο σενάριο C.

5.2.3 Συγκεντρωτικά αποτελέσματα

Τα παραπάνω σενάρια εφαρμόστηκαν σε όλα τα προβλήματα, τα οποία είναι συνολικά 34. Παρουσιάζονται οι συγκεντρωτικοί πίνακες για τα δύο μοντέλα αντίστοιχα, από τους οποίους προκύπτουν για κάθε αρχείο, η συνολική ποσότητα φόρτωσης (λίτρα), οι συνολικές αποκλίσεις (λίτρα), το ποσοστό αξιοποίησης της χωρητικότητας του οχήματος και το ποσοστό των συνολικών αποκλίσεων επί της ζήτησης για κάθε σενάριο, καθώς και η διαφορά (σε ποσοστό) της ποσότητας φόρτωσης από τη σύγκριση των σεναρίων.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΜΟΝΤΕΛΟ ΧΩΡΙΣ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ				
ΣΕΝΑΡΙΟ Α				
Α/Α	**Συνολική ποσότητα φόρτωσης (lt)**	**Συνολικές αποκλίσεις (lt)**	**Ποσοστό αξιοποίησης της χωρητικότητας του οχήματος**	**Ποσοστό συνολικών αποκλίσεων επί της ζήτησης (Α)**
1	20850	0	96,53%	0,00%
2	20551	0	95,14%	0,00%
3	20598	0	95,36%	0,00%
4	20552	0	95,15%	0,00%
5	20247	0	93,74%	0,00%
6	20405	0	94,47%	0,00%
7	20719	0	95,92%	0,00%
8	20683	0	95,75%	0,00%
9	20156	0	93,31%	0,00%
10	20414	0	94,51%	0,00%
11	20573	0	95,25%	0,00%
12	20498	0	94,90%	0,00%
13	20592	0	95,33%	0,00%
14	20880	0	96,67%	0,00%
15	20479	0	94,81%	0,00%
16	20642	0	95,56%	0,00%
17	20740	0	96,02%	0,00%
18	20220	0	93,61%	0,00%
19	20367	0	94,29%	0,00%
20	20829	0	96,43%	0,00%
21	20881	0	96,67%	0,00%
22	20290	0	93,94%	0,00%
23	20553	0	95,15%	0,00%
24	20329	0	94,12%	0,00%
25	20986	0	97,16%	0,00%
26	20537	0	95,08%	0,00%
27	20405	0	94,47%	0,00%
28	20971	0	97,09%	0,00%
29	20612	0	95,43%	0,00%
30	20385	0	94,38%	0,00%
31	20641	0	95,56%	0,00%
32	20869	0	96,62%	0,00%
33	20446	0	94,66%	0,00%
34	20709	0	95,88%	0,00%
		Μέσος Όρος	**95,26%**	0,00%
		Τυπική απόκλιση	**0,997%**	0,00%

Πίνακας 19: Συγκεντρωτικά αποτελέσματα μοντέλου χωρίς μετρητές στάθμης- Σενάριο Α

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΜΟΝΤΕΛΟ ΧΩΡΙΣ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ				
ΣΕΝΑΡΙΟ Β				
A/A	**Συνολική ποσότητα φόρτωσης (lt)**	**Συνολικές αποκλίσεις (lt)**	**Ποσοστό αξιοποίησης της χωρητικότητας του οχήματος**	**Ποσοστό συνολικών αποκλίσεων επί της ζήτησης (Β)**
1	18432	2418	85,33%	11,60%
2	18604	1947	86,13%	9,47%
3	18526	2072	85,77%	10,06%
4	18495	2057	85,63%	10,01%
5	18518	1729	85,73%	8,54%
6	18556	1849	85,91%	9,06%
7	18512	2207	85,70%	10,65%
8	18560	2123	85,93%	10,26%
9	18572	1584	85,98%	7,86%
10	18585	1829	86,04%	8,96%
11	18604	1969	86,13%	9,57%
12	18604	1894	86,13%	9,24%
13	18473	2119	85,52%	10,29%
14	18431	2449	85,33%	11,73%
15	18604	1875	86,13%	9,16%
16	18552	2090	85,89%	10,12%
17	18442	2298	85,38%	11,08%
18	18604	1616	86,13%	7,99%
19	18920	1447	87,59%	7,10%
20	18524	2305	85,76%	11,07%
21	18389	2492	85,13%	11,93%
22	18566	1724	85,95%	8,50%
23	18490	2063	85,60%	10,04%
24	18574	1755	85,99%	8,63%
25	18539	2447	85,83%	11,66%
26	18560	1977	85,93%	9,63%
27	18538	1867	85,82%	9,15%
28	18577	2394	86,00%	11,42%
29	18513	2099	85,71%	10,18%
30	18541	1844	85,84%	9,05%
31	18376	2265	85,07%	10,97%
32	18546	2323	85,86%	11,13%
33	18536	3064	85,81%	9,34%
34	18518	2191	85,73%	10,58%
		Μέσος Όρος	**85,84%**	**9,88%**
		Τυπική απόκλιση	**0,410%**	**1,192%**

Πίνακας 20: Συγκεντρωτικά αποτελέσματα μοντέλου χωρίς μετρητές στάθμης- Σενάριο Β

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΜΟΝΤΕΛΟ ΧΩΡΙΣ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ	
ΣΥΓΚΡΙΣΗ ΣΕΝΑΡΙΩΝ	
Α/Α	**Ποσοστό επιπλέον ποσότητας φόρτωσης (A vs B)**
1	13,10%
2	10,50%
3	11,20%
4	11,10%
5	9,30%
6	10,00%
7	11,90%
8	11,40%
9	8,50%
10	9,80%
11	10,60%
12	10,20%
13	11,50%
14	13,30%
15	10,10%
16	11,30%
17	12,50%
18	8,70%
19	7,60%
20	12,40%
21	13,60%
22	9,30%
23	11,20%
24	9,40%
25	13,20%
26	10,70%
27	10,10%
28	12,90%
29	11,30%
30	9,90%
31	12,30%
32	12,50%
33	10,30%
34	11,80%
Μέσος Όρος	**10,99%**
Τυπική απόκλιση	**1,471%**

Πίνακας 21: Συγκεντρωτικά αποτελέσματα μοντέλου χωρίς μετρητές στάθμης- Σύγκριση σεναρίων

Μοντέλο χωρίς μετρητές στάθμης (M1):

- ✓ Οι αποκλίσεις ανάμεσα στη συνολική ποσότητα φόρτωσης και τη συνολική ζήτηση είναι μηδέν στο σενάριο Α, άρα όλη η ζήτηση ικανοποιείται όταν δεν συμπεριλαμβάνονται οι περιορισμοί που αφορούν την ευστάθεια και τη θέση του κέντρου βάρους του φορτίου.
- ✓ Στο σενάριο Α η χωρητικότητα του βυτιοφόρου οχήματος αξιοποιείται σε ποσοστό κατά μέσο όρο 95,26% με τυπική απόκλιση 0,997%.
- ✓ Στο σενάριο Β η χωρητικότητα του βυτιοφόρου οχήματος αξιοποιείται σε ποσοστό κατά μέσο όρο 85,84% με τυπική απόκλιση 0,410%.

- ✓ Οι συνολικές αποκλίσεις ανάμεσα στη συνολική ποσότητα φόρτωσης και τη συνολική ζήτηση αποτελούν κατά μέσο όρο το 9,88% της συνολικής ζήτησης στο σενάριο Β με τυπική απόκλιση 1,192%.
- ✓ Η συνολική ποσότητα φόρτωσης στο σενάριο Α είναι κατά μέσο όρο 10,99% περισσότερη από τη συνολική ποσότητα φόρτωσης στο σενάριο Β με τυπική απόκλιση 1,471%.

ΜΟΝΤΕΛΟ ΜΕ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ				
ΣΕΝΑΡΙΟ Α				
Α/Α	**Συνολική ποσότητα φόρτωσης (lt)**	**Συνολικές αποκλίσεις (lt)**	**Ποσοστό αξιοποίησης της χωρητικότητας του οχήματος**	**Ποσοστό συνολικών αποκλίσεων επί της ζήτησης (Α)**
1	20850	0	96,53%	0,00%
2	20551	0	95,14%	0,00%
3	20598	0	95,36%	0,00%
4	20552	0	95,15%	0,00%
5	20247	0	93,74%	0,00%
6	20405	0	94,47%	0,00%
7	20719	0	95,92%	0,00%
8	20683	0	95,75%	0,00%
9	20156	0	93,31%	0,00%
10	20414	0	94,51%	0,00%
11	20573	0	95,25%	0,00%
12	20498	0	94,90%	0,00%
13	20592	0	95,33%	0,00%
14	20880	0	96,67%	0,00%
15	20479	0	94,81%	0,00%
16	20642	0	95,56%	0,00%
17	20740	0	96,02%	0,00%
18	20220	0	93,61%	0,00%
19	20367	0	94,29%	0,00%
20	20829	0	96,43%	0,00%
21	20881	0	96,67%	0,00%
22	20920	0	96,85%	0,00%
23	20553	0	95,15%	0,00%
24	20329	0	94,12%	0,00%
25	20986	0	97,16%	0,00%
26	20537	0	95,08%	0,00%
27	20405	0	94,47%	0,00%
28	20971	0	97,09%	0,00%
29	20612	0	95,43%	0,00%
30	20385	0	94,38%	0,00%
31	20641	0	95,56%	0,00%
32	20869	0	96,62%	0,00%
33	20446	0	94,66%	0,00%
34	20709	0	95,88%	0,00%
		Μέσος Όρος	**95,35%**	0,00%
		Τυπική απόκλιση	**1,004%**	0,00%

Πίνακας 22: Συγκεντρωτικά αποτελέσματα μοντέλου με μετρητές στάθμης- Σενάριο Α

ΜΟΝΤΕΛΟ ΜΕ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ				
ΣΕΝΑΡΙΟ Β				
A/A	**Συνολική ποσότητα φόρτωσης (lt)**	**Συνολικές αποκλίσεις (lt)**	**Ποσοστό αξιοποίησης της χωρητικότητας του οχήματος**	**Ποσοστό συνολικών αποκλίσεων επί της ζήτησης (Β)**
1	18920	1930	87,59%	9,26%
2	18920	1631	87,59%	7,94%
3	18852	1746	87,28%	8,48%
4	18919	1633	87,59%	7,59%
5	18920	1327	87,59%	6,55%
6	18920	1485	87,59%	7,28%
7	18912	1807	87,56%	8,72%
8	18920	1763	87,59%	8,52%
9	18805	1351	87,06%	6,70%
10	18921	1493	87,60%	7,31%
11	18788	1785	86,98%	8,68%
12	18920	1578	87,59%	7,70%
13	18788	1804	86,98%	8,76%
14	18920	1960	87,59%	9,39%
15	18919	1560	87,59%	7,62%
16	18789	1853	86,99%	8,98%
17	18873	2727	87,38%	9,00%
18	18789	1431	86,99%	7,08%
19	18920	1447	87,59%	7,10%
20	18768	2061	86,89%	9,89%
21	18921	1960	87,60%	9,39%
22	18921	2679	87,60%	6,75%
23	18920	1633	87,59%	7,95%
24	18919	1410	87,59%	6,94%
25	18595	2391	86,09%	11,39%
26	18920	1617	87,59%	7,87%
27	18919	1486	87,59%	7,28%
28	18921	2050	87,60%	9,78%
29	18919	1693	87,59%	8,21%
30	18920	1465	87,59%	7,19%
31	18740	1901	86,76%	9,21%
32	18921	1948	87,60%	9,33%
33	18920	1526	87,59%	7,46%
34	18920	1789	87,59%	8,64%
			87,40%	**8,23%**
			0,347%	**1,092%**

Πίνακας 23: Συγκεντρωτικά αποτελέσματα μοντέλου με μετρητές στάθμης- Σενάριο Β

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΜΟΝΤΕΛΟ ΜΕ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ				
ΣΕΝΑΡΙΟ C				
A/A	**Συνολική ποσότητα φόρτωσης (lt)**	**Συνολικές αποκλίσεις (lt)**	**Ποσοστό αξιοποίησης της χωρητικότητας του οχήματος**	**Ποσοστό συνολικών αποκλίσεων επί της ζήτησης (C)**
1	17159	3691	79,44%	17,70%
2	17159	3392	79,44%	16,51%
3	17160	3438	79,44%	16,69%
4	17159	3393	79,44%	16,51%
5	17158	3089	79,44%	15,26%
6	17159	3246	79,44%	15,91%
7	17158	3561	79,44%	17,19%
8	17159	941	79,44%	17,04%
9	17159	2997	79,44%	14,87%
10	17159	3255	79,44%	15,94%
11	17159	3414	79,44%	16,59%
12	17160	3338	79,44%	16,28%
13	17159	3433	79,44%	16,67%
14	17159	3721	79,44%	17,82%
15	17157	3322	79,43%	16,22%
16	17159	3483	79,44%	16,87%
17	17159	3581	79,44%	17,27%
18	17160	3060	79,44%	15,13%
19	17160	3207	79,44%	15,75%
20	17159	3670	79,44%	17,62%
21	17159	3722	79,44%	17,82%
22	17159	3131	79,44%	15,43%
23	17159	3394	79,44%	16,51%
24	17159	3170	79,44%	15,59%
25	17446	3540	80,77%	16,87%
26	17472	3065	80,89%	14,92%
27	17159	3246	79,44%	15,91%
28	17159	3812	79,44%	18,18%
29	17159	3453	79,44%	16,75%
30	17159	3226	79,44%	15,83%
31	17159	3482	79,44%	16,87%
32	16853	4016	78,02%	19,24%
33	17159	3287	79,44%	16,08%
34	17159	3550	79,44%	17,14%
		Μέσος Όρος	**79,48%**	**16,56%**
		Τυπική απόκλιση	0,414%	**0,964%**

Πίνακας 24: Συγκεντρωτικά αποτελέσματα μοντέλου με μετρητές στάθμης- Σενάριο C

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΜΟΝΤΕΛΟ ΜΕ ΜΕΤΡΗΤΕΣ ΣΤΑΘΜΗΣ			
ΣΥΓΚΡΙΣΗ ΣΕΝΑΡΙΩΝ			
Α/Α	**Ποσοστό επιπλέον ποσότητας φόρτωσης (A vs B)**	**Ποσοστό επιπλέον ποσότητας φόρτωσης (A vs C)**	**Ποσοστό λιγότερης ποσότητας φόρτωσης (C vs B)**
1	10,20%	21,50%	-9,30%
2	8,60%	19,80%	-9,30%
3	9,30%	20,00%	-9,00%
4	8,60%	19,80%	-9,30%
5	7,00%	18,00%	-9,30%
6	7,80%	18,90%	-9,30%
7	9,60%	20,80%	-9,30%
8	9,30%	20,50%	-9,30%
9	7,20%	17,50%	-8,80%
10	7,90%	19,00%	-9,30%
11	9,50%	19,90%	-8,70%
12	8,30%	19,50%	-9,30%
13	9,60%	20,00%	-8,70%
14	10,40%	21,70%	-9,30%
15	8,20%	19,40%	-9,30%
16	9,90%	20,30%	-8,70%
17	9,90%	20,90%	-9,10%
18	7,60%	17,80%	-8,70%
19	7,60%	18,70%	-9,30%
20	11,00%	21,40%	-8,60%
21	10,40%	21,70%	-9,30%
22	7,20%	18,20%	-9,30%
23	8,60%	19,80%	-9,30%
24	7,50%	18,50%	-9,30%
25	12,90%	20,30%	-6,20%
26	8,50%	17,50%	-7,70%
27	7,90%	18,90%	-9,30%
28	10,80%	22,20%	-9,30%
29	8,90%	20,10%	-9,30%
30	7,70%	18,80%	-9,30%
31	10,10%	20,30%	-8,40%
32	10,30%	23,80%	-10,90%
33	8,10%	19,20%	-9,30%
34	9,50%	20,70%	-9,30%
Μέσος Όρος	**9,00%**	**19,86%**	**-9,06%**
Τυπική απόκλιση	**1,313%**	**1,392%**	**0,684%**

Πίνακας 25: Συγκεντρωτικά αποτελέσματα μοντέλου με μετρητές στάθμης- Σύγκριση σεναρίων

Μοντέλο με μετρητές στάθμης (M2):

- ✓ Οι αποκλίσεις ανάμεσα στη συνολική ποσότητα φόρτωσης και τη συνολική ζήτηση είναι μηδέν στο σενάριο Α, άρα όλη η ζήτηση ικανοποιείται όταν δεν συμπεριλαμβάνονται οι περιορισμοί που αφορούν την ευστάθεια και τη θέση του κέντρου βάρους του φορτίου.
- ✓ Στο σενάριο Α η χωρητικότητα του βυτιοφόρου οχήματος αξιοποιείται σε ποσοστό κατά μέσο όρο 95,35% με τυπική απόκλιση 1,004%.
- ✓ Στο σενάριο Β η χωρητικότητα του βυτιοφόρου οχήματος αξιοποιείται σε ποσοστό κατά μέσο όρο 87,40% με τυπική απόκλιση 0,347%.
- ✓ Στο σενάριο C η χωρητικότητα του βυτιοφόρου οχήματος αξιοποιείται σε ποσοστό κατά μέσο όρο 79,48% με τυπική απόκλιση 0,414%.
- ✓ Οι συνολικές αποκλίσεις ανάμεσα στη συνολική ποσότητα φόρτωσης και τη συνολική ζήτηση αποτελούν κατά μέσο όρο το 8,23% της συνολικής ζήτησης στο σενάριο Β με τυπική απόκλιση 1,092%.
- ✓ Οι συνολικές αποκλίσεις ανάμεσα στη συνολική ποσότητα φόρτωσης και τη συνολική ζήτηση αποτελούν κατά μέσο όρο το 16,56% της συνολικής ζήτησης στο σενάριο C με τυπική απόκλιση 0,964%.

- ✓ Η συνολική ποσότητα φόρτωσης στο σενάριο Α είναι κατά μέσο όρο 9,00% περισσότερη από τη συνολική ποσότητα φόρτωσης στο σενάριο Β με τυπική απόκλιση 1,313%.
- ✓ Η συνολική ποσότητα φόρτωσης στο σενάριο Α είναι κατά μέσο όρο 19,86% περισσότερη από τη συνολική ποσότητα φόρτωσης στο σενάριο C με τυπική απόκλιση 1,392%.
- ✓ Η συνολική ποσότητα φόρτωσης στο σενάριο C είναι κατά μέσο όρο 9,06% λιγότερη από τη συνολική ποσότητα φόρτωσης στο σενάριο Β με τυπική απόκλιση 0,684%.

5.3 Σύνοψη συμπερασμάτων αξιολόγησης

1. Όταν το βυτιοφόρο όχημα δε διαθέτει μετρητές στάθμης, προκειμένου να εξασφαλιστεί η ευστάθεια αναφορικά με τη μη υπέρβαση του βάρους των αξόνων, μειώνεται η συνολική ποσότητα φόρτωσης και το ποσοστό αξιοποίησης της χωρητικότητας του οχήματος και επακόλουθα αυξάνονται οι αποκλίσεις ανάμεσα στην τελική ποσότητα φόρτωσης και τη ζήτηση. Επιπλέον, όταν ένα βυτιοφόρο όχημα δε διαθέτει μετρητές στάθμης, είναι αδύνατο να εξασφαλιστεί ο περιορισμός που αφορά τη θέση του κέντρου βάρους του φορτίου, δεδομένου του ορίου του 20% αναφορικά με την αποδεκτή απόκλιση στην ποσότητα του προϊόντος.
2. Όταν το βυτιοφόρο όχημα διαθέτει μετρητές στάθμης, προκειμένου να εξασφαλιστεί η ευστάθεια αναφορικά με τη μη υπέρβαση του βάρους των αξόνων και με την εξασφάλιση της ομοιόμορφης κατανομής του βάρους του φορτίου, μειώνεται η συνολική ποσότητα φόρτωσης και το ποσοστό αξιοποίησης της χωρητικότητας του οχήματος και αυξάνονται οι αποκλίσεις ανάμεσα στην τελική ποσότητα φόρτωσης και τη ζήτηση. Παράλληλα, ο περιορισμός της θέσης του κέντρου βάρους του φορτίου μειώνει περαιτέρω τη συνολική ποσότητα φόρτωσης και το ποσοστό αξιοποίησης της χωρητικότητας του οχήματος συγκριτικά με το εάν υπήρχαν μόνο οι περιορισμοί ευστάθειας, καθιστώντας τον περιορισμό αυτό αρκετά αυστηρό.
3. Πραγματοποιώντας διαμοντελική σύγκριση στην περίπτωση του σεναρίου Β, προκύπτει ότι στο μοντέλο με τους μετρητές στάθμης, η ποσότητα φόρτωσης είναι μεγαλύτερη κατά 1,81% κατά μέσο όρο με τυπική απόκλιση 0,59% και τα ποσοστά αποδεκτής αύξησης και μείωσης στην ποσότητα του προϊόντος (β_r^+, β_r^-) είναι μικρότερα. Παράλληλα, στην περίπτωση της ύπαρξης μετρητών στάθμης, είναι εφικτό να συμπεριληφθεί και ο περιορισμός της θέσης του κέντρου βάρους του φορτίου. Συνεπώς, η ύπαρξη μετρητών στάθμης βελτιώνει τόσο το επίπεδο εξυπηρέτησης των πελατών όσο και την κατανομή του βάρους του φορτίου, καθιστώντας τη πιο ομοιόμορφη και εξασφαλίζοντας επιπρόσθετη ασφάλεια.
4. Στην περίπτωση του σεναρίου C του μοντέλου με τους μετρητές στάθμης, το 7ο (τελευταίο) διαμέρισμα είναι κενό, το οποίο εξηγείται καθώς το διαμέρισμα αυτό έχει τη μεγαλύτερη απόσταση από το γεωμετρικό κέντρο βάρους του βυτιοφόρου οχήματος.

6

Επίλογος

Στην ενότητα αυτή θα γίνει η σύνοψη της Διπλωματικής, η παρουσίαση των συμπερασμάτων και η διατύπωση μελλοντικών επεκτάσεων.

6.1 Σύνοψη και συμπεράσματα

Προκειμένου να ληφθούν υπόψιν οι περιορισμοί ευστάθειας για τη φόρτωση βυτιοφόρου οχήματος, πραγματοποιήθηκε η μοντελοποίηση των περιορισμών αυτών, οι οποίοι διακρίνονται στη μη υπέρβαση των μέγιστων επιτρεπόμενων ορίων βάρους κατ' άξονα και στην ομοιόμορφη κατανομή του βάρους του φορτίου για κάθε φάση της διαδρομής. Στη συνέχεια, έγινε η ενσωμάτωσή τους σε υπάρχοντα μοντέλα φόρτωσης μεικτού ακέραιου γραμμικού προγραμματισμού με ή χωρίς μετρητές στάθμης.

Μετά την εφαρμογή των μοντέλων φόρτωσης σε προβλήματα τα οποία αποτελούνται από πέντε πρατήρια καυσίμων, προέκυψαν τα εξής γενικά συμπεράσματα:

- ✓ Στην περίπτωση βυτιοφόρου οχήματος χωρίς μετρητές στάθμης, προκειμένου να εξασφαλιστεί η ευστάθεια αναφορικά με τη μη υπέρβαση του βάρους των αξόνων, μειώνεται η συνολική ποσότητα φόρτωσης και το ποσοστό αξιοποίησης της χωρητικότητας του οχήματος.
- ✓ Στην περίπτωση βυτιοφόρου οχήματος χωρίς μετρητές στάθμης, δεν μπορεί να εξασφαλιστεί ο περιορισμός που αφορά την ομοιόμορφη κατανομή του βάρους του φορτίου.
- ✓ Στην περίπτωση βυτιοφόρου οχήματος με μετρητές στάθμης, προκειμένου να εξασφαλιστεί η ευστάθεια αναφορικά με τη μη υπέρβαση του βάρους των αξόνων και την εξασφάλιση της ομοιόμορφης κατανομής του βάρους του φορτίου, μειώνεται η συνολική ποσότητα φόρτωσης και το ποσοστό αξιοποίησης της χωρητικότητας του οχήματος.
- ✓ Στην περίπτωση βυτιοφόρου οχήματος με μετρητές στάθμης, ο περιορισμός της ομοιόμορφης κατανομής του βάρους του φορτίου μειώνει περαιτέρω τη συνολική ποσότητα φόρτωσης, συγκριτικά με το εάν υπήρχε μόνο ο περιορισμός της μη υπέρβασης του βάρους των αξόνων, καθιστώντας τον αρκετά αυστηρό.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

- ✓ Η ύπαρξη μετρητών στάθμης βελτιώνει τόσο το επίπεδο εξυπηρέτησης των πελατών, καθώς η ποσότητα φόρτωσης είναι μεγαλύτερη, όσο και την οδική ασφάλεια του οχήματος, καθώς λαμβάνεται υπόψιν η ομοιόμορφη κατανομή του βάρους του φορτίου.

Συνεπώς, είναι προφανή τα «trade-offs» που προκύπτουν αναφορικά με την ποσότητα φόρτωσης, τις συνολικές αποκλίσεις επί της αρχικής ζήτησης, του ποσοστού αξιοποίησης της χωρητικότητας του βυτιοφόρου οχήματος και τους περιορισμούς ευστάθειας, καθώς όσο «βελτιώνεται» η ευστάθεια και κατ' επέκταση η οδική ασφάλεια, τόσο «επιβαρύνεται» η βέλτιστη φόρτωση του βυτιοφόρου οχήματος.

Εν γένει, τα μαθηματικά μοντέλα φόρτωσης με τους περιορισμούς ευστάθειας που αναπτύχθηκαν, μπορούν να ενσωματωθούν σε Συστήματα Στήριξης Αποφάσεων για τη φόρτωση οχημάτων, δημιουργώντας ένα χρήσιμο εργαλείο το οποίο θα εξασφαλίζει με αναλυτικό τρόπο την ευστάθεια και την οδική ασφάλεια των οχημάτων.

6.2 Μελλοντικές επεκτάσεις

Αναφέρονται οι εξής προτάσεις για επέκταση της Διπλωματικής Εργασίας:

- ✓ Εφαρμογή των μοντέλων του μεικτού ακέραιου γραμμικού προγραμματισμού τα οποία συμπεριλαμβάνουν τους περιορισμούς ευστάθειας, σε προβλήματα μεγαλύτερης κλίμακας, με περισσότερους πελάτες και παραγγελθέντα προϊόντα, με πραγματικά δεδομένα, καθώς και με διαφορετικούς τύπους οχημάτων, ώστε να ενισχυθούν τα συμπεράσματα και τα ευρήματα που προκύπτουν.
- ✓ Επίλυση του προβλήματος ανεφοδιασμού πρατηρίων καυσίμων, ως συνδυαστικό πρόβλημα δρομολόγησης και φόρτωσης στόλου βυτιοφόρων οχημάτων με περιορισμούς ευστάθειας.
- ✓ Ανάπτυξη αλγορίθμων (πχ ευρετικών) για την επίλυση προβλημάτων μεγαλύτερης κλίμακας.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

7

Βιβλιογραφία

[1] ADR, European Agreement Concerning the International Carriage of Dangerous Goods by Road, Economic Commission for Europe, Committee of Inland Transport, United Nations, New York and Geneva, 2016, ISBN: 978-92-1-139140-4

[2] Alonso, M. T., Alvarez-Valdes, R., Iori, M., Parreño, F., & Tamarit, J. M. (2017). Mathematical models for multicontainer loading problems. Omega (United Kingdom), 66, 106–117. https://doi.org/10.1016/j.omega.2016.02.002

[3] Alonso, M. T., Alvarez-Valdes, R., Parreño, F., & Tamarit, J. M. (2016). Algorithms for pallet building and truck loading in an interdepot transportation problem. Mathematical Problems in Engineering, 2016. https://doi.org/10.1155/2016/3264214

[4] Androutsopoulos, K.N. and E. Karouti (2021). A Load, Time and Population dependent risk metric for the fuel distribution problem in urban areas. Book of Abstracts of the 31st European Conference On Operational Research (Hybrid), 11-14 July, Athens, Greece.

[5] Avella, P., Boccia, M., Sforza, A., 2004. Solving a fuel delivery problem by heuristic and exact approaches. European Journal of Operational Research 152, 170 – 179.

[6] Balakirsky, Stephen & Christensen, Henrik & Kramer, Thomas & Kolhe, Pushkar & Proctor, Frederick. (2010). Mixed pallet stacking: an overview and summary of the 2010 PerMIS special session. 10.1145/2377576.2377588.

[7] Baldi, Mauro & Perboli, Guido & Tadei, Roberto. (2012). The three-dimensional knapsack problem with balancing constraints. Applied Mathematics and Computation. 218. 9802–9818. 10.1016/j.amc.2012.03.052.

[8] Bischoff, E.E., Janetz, F., Ratcliff, M.S.W., 1995. Loading pallets with non-identical items. European Journal of Operational Research 84, 681–692.

[9] Bischoff, E.E., Marriott, M.D., 1990. A comparative evaluation of heuristics for container loading. European Journal of Operational Research 44, 267–276

[10] Bortfeldt A., Gehring H., 1997. A genetic algorithm for solving the container loading problem. International Transactions in Operational Research 4, 401-418.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

[11] Bortfeldt, A., & Wäscher, G. (2013). Constraints in container loading-A state-of-the-art review. European Journal of Operational Research, 229(1), 1–20. https://doi.org/10.1016/j.ejor.2012.12.006

[12] Bortfeldt, A., Gehring, H. & Mack, D., 2003. A parallel tabu search algorithm for solving the container loading problem. Parallel Computing, Τόμος 29, p. 641–662

[13] Bortfeldt, A., Gehring, H., 1997. Applying tabu search to container loading problems. In: Operations Research Proceedings. Springer, Berlin, pp. 533–538

[14] Bortfeldt, A., Gehring, H., 2001. A hybrid genetic algorithm for the container loading problem. European Journal of Operational Research 131, 143–161

[15] Brown, G.G, Ellis, C.J., Graves, G.W., Ronen, D., 1987. Real-Time Wide Area Dispatch of Mobil Tank Trucks. Interfaces, 17(1), 107-120.

[16] Coelho, C., Laporte, G., 2015. Classification models and exact algorithms for multi-compartment delivery problems. European Journal of Operational Research 242,854-864.

[17] Cornillier, F., Boctor, F. F., Laporte, G., & Renaud, J. (2008b). An exact algorithm for the petrol station replenishment problem. Journal of the Operational Research Society, 59(5), 607–615. doi:10.1057/palgrave.jors.2602374

[18] Costa, M. da G., & Captivo, M. E. (2016). Weight distribution in container loading: A case study. International Transactions in Operational Research, 23(1–2), 239–263. https://doi.org/10.1111/itor.12145

[19] Davies, A. Paul and Bischoff, Eberhard E., (1999), Weight distribution considerations in container loading, European Journal of Operational Research, 114, issue 3, p. 509-527.

[20] E. Planas-Cuchi, N. Gasulla, A. Ventosa, J. Casal, "Explosion of a road tanker containing liquefied natural gas", Journal of Loss Prevention in the Process Industries 17: 315-321, 2004.

[21] Eley, M., 2002. Solving container loading problems by block arrangement. European Journal of Operational Research, 141(9), pp. 393-409

[22] Fanslau, T., Bortfeldt, A., 2010. A tree search algorithm for solving the container loading problem. INFORMS Journal on Computing 22, 222–235.

[23] Hemminki, J., 1994. Container loading with variable strategies in each layer. Presented at ESI-X, EURO Summer Institute, Jouy-en-Josas France

[24] Hifi, M., 2004. Exact algorithms for unconstrained three-dimensional cutting problems: a comparative study. Computers & Operations Research, 31(5), p. 657–674

[25] Lim, A., Ma, H., Qiu, C., & Zhu, W. (2013). The single container loading problem with axle weight constraints. International Journal of Production Economics, 144(1), 358–369. https://doi.org/10.1016/j.ijpe.2013.03.001

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

[26] Liu, J., Yue, Y., Dong, Z., Maple, C., & Keech, M. (2011). A novel hybrid tabu search approach to container loading. Computers and Operations Research, 38(4), 797–807. https://doi.org/10.1016/j.cor.2010.09.002

[27] Liu, M., Man, X., Zheng, F., & Chu, C. (2017, July 28). An integer programming model for the single container loading problem with axle weight constraints. 14th International Conference on Services Systems and Services Management, ICSSSM 2017 - Proceedings. https://doi.org/10.1109/ICSSSM.2017.7996151

[28] Liu, N.C., Chen, L.C., 1981. A new algorithm for container loading. In: Compsac 81- 5th International Computer Software and Application Conference Proceedings. IEEE Press, pp. 292–299.

[29] Loh, H.T., Nee, A.Y.C., 1992. A packing algorithm for hexahedral boxes. In: Proceedings of the Industrial Automation'92 Conference. Singapore, pp. 115–126.

[30] Martello, S., Pisinger, D. & Vigo, D., 2000. The three dimensional bin packing problem. Operations Research, Τόμος 48, p. 256 – 267.

[31] Moon, I., & Nguyen, T. V. L. (2014). Container packing problem with balance constraints. OR spectrum, 36(4), 837-878.

[32] Moura, A., Oliveira, J., 2005. A GRASP approach to the container-loading problem. Intelligent Systems IEEE 20, 50–57.

[33] Novas, J. M., Ramello, J. I., & Rodríguez, M. A. (2020). Generalized disjunctive programming models for the truck loading problem: A case study from the non-alcoholic beverages industry. Transportation Research Part E: Logistics and Transportation Review, 140. https://doi.org/10.1016/j.tre.2020.101971

[34] P. Kumar, "Fire disaster following LPG tanker explosion at Chala in Kannur (Kerala, India): August 27, 2012", Burns 39: 1479-1487, 2013.

[35] Parreño, F., Alvarez-Valdes, R., Oliveira, J. & Tamarit, J., 2010. Neighborhood structures for the container loading problem: a VNS implementation. Journal of Heuristics, Τόμος 16, pp. 1-22

[36] Pollaris, H., Braekers, K., Caris, A., Janssens, G. K., & Limbourg, S. (2014). Capacitated vehicle routing problem with sequence-based pallet loading and axle weight constraints. EURO Journal on Transportation and Logistics, 5(2), 231–255. doi: 10.1007/s13676-014-0064-2

[37] Portmann, M.C., 1990. An efficient algorithm for container loading. Methods of Operations Research 64, 563–572.

[38] Ramos, A. G., Silva, E., & Oliveira, J. F. (2018). A new load balance methodology for container loading problem in road transportation. European Journal of Operational Research, 266(3), 1140–1152. https://doi.org/10.1016/j.ejor.2017.10.050

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

[39] Sciomachen, Anna & Tanfani, Elena. (2007). A 3D-BPP approach for optimising stowage plans and terminal productivity. European Journal of Operational Research. 183. 1433-1446. 10.1016/j.ejor.2005.11.067.

[40] Techanitisawad, Anulark & Tangwiwatwong, (2004). A GA-based Heuristic for the Interrelated Container Selection Loading Problems. IEMS. 3. 22-37.

[41] Terno, Johannes & Scheithauer, Guntram & Sommerwei[ss, Uta & Riehme, Jan. (2000). An Efficient Approach for the Multi-Pallet Loading Problem. European Journal of Operational Research. 123. 372-381. 10.1016/S0377-2217(99)00263-5.

[42] Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. European Journal of Operational Research, 183(3), 1109–1130. https://doi.org/10.1016/j.ejor.2005.12.047

[43] Y. Zengin, R. Dursun, M. Içer, E. Gündüz, H. M. Durgun, S. Erbatur, Ö. Damar, C. Güloğlu, "Fire disaster caused by LPG tanker explosion at Lice in Diyarbakir (Turkey): July 21, 2014", Burns 41: 1347-1352, 2015.

[44] El Yamani, O., & Saddoune, M. (2018, October 24). Gas-stations filling procurement problem with a multi-Size compartments vehicles and full tankers under limited number of trucks. ACM International Conference Proceeding Series. https://doi.org/10.1145/3289402.3289545

[45] Θεωρητική εκπαίδευση υποψήφιων οδηγών φορτηγών, Υπουργείο Μεταφορών και Επικοινωνιών, Αθήνα, 2008, ISBN: 960-337-032-0.

[46] Οδική Μεταφορά Επικίνδυνων Εμπορευμάτων, Υπουργείο Υποδομών, Μεταφορών και Δικτύων, Αθήνα, Ιούλιος 2013, ISBN: 978-960-87771-7-0 2.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Παράρτημα

Παρακάτω, παρουσιάζεται αναλυτικά ο κώδικας και οι κλάσεις που αναπτύχθηκαν στην Java.

```
import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.Map;

import ilog.concert.IloException;

public class Main {

    public static void main(String[] args) throws IloException, IOException {

        // ShipTo Object Initialization
        ShipTo shipTo1  = new ShipTo("18015040", 23.7098, 37.93025);
        ShipTo shipTo2  = new ShipTo("507702", 23.71844, 37.98871);
        ShipTo shipTo3  = new ShipTo("18011583", 23.80933, 38.08931);
        ShipTo shipTo4  = new ShipTo("505686", 23.74405, 37.91968);
        ShipTo shipTo5  = new ShipTo("505842", 23.68993, 38.0177);

        //Create ArrayList for shipTo Objects
        ArrayList<ShipTo> shipToList = new ArrayList<ShipTo>();
        shipToList.add(shipTo1);
        shipToList.add(shipTo2);
        shipToList.add(shipTo3);
        shipToList.add(shipTo4);
        shipToList.add(shipTo5);

        //OrderItem Object Initialization
        OrderItem orderItem1 = new OrderItem("10", 2901, 0.7);
        OrderItem orderItem2 = new OrderItem("10", 5569, 0.7);
        OrderItem orderItem3 = new OrderItem("10", 3466, 0.7);
        OrderItem orderItem4 = new OrderItem("20", 3340, 0.7);
        OrderItem orderItem5 = new OrderItem("10", 3168, 0.7);
        OrderItem orderItem6 = new OrderItem("10", 2406, 0.7);
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
35
36          //Create ArrayList for OrderItem Objects
37          ArrayList<OrderItem> orderItemList = new ArrayList<OrderItem>();
38          orderItemList.add(orderItem1);
39          orderItemList.add(orderItem2);
40          orderItemList.add(orderItem3);
41          orderItemList.add(orderItem4);
42          orderItemList.add(orderItem5);
43          orderItemList.add(orderItem6);
44
45          //Order Object Initialization
46          Order order1 = new Order("18015040","18015040");
47          Order order2 = new Order("507702","507702");
48          Order order3 = new Order("18011583","18011583");
49          Order order4 = new Order("505686","505686");
50          Order order5 = new Order("505842","505842");
51
52          //Compartments Object Initialization
53          Compartment comp0 = new Compartment("0", 0 , 0.090 , 5.440 , 6.740 , 5600 , 0 , 2.31 );
54          Compartment comp1 = new Compartment("1", 3200 , 1.712 , 3.638 , 4.938 , 622 , 2.31 , 3.60 );
55          Compartment comp2 = new Compartment("2", 2450 , 2.854 , 2.496 , 3.796 , 476 , 3.60 , 4.59 );
56          Compartment comp3 = new Compartment("3", 3100 , 3.977 , 1.373 , 2.673 , 603 , 4.59 , 5.85 );
57          Compartment comp4 = new Compartment("4", 3450 , 5.301 , 0.049 , 1.349 , 671 , 5.85 , 7.24 );
58          Compartment comp5 = new Compartment("5", 2550 , 6.514 , 1.164 , 0.136 , 496 , 7.24 , 8.27 );
59          Compartment comp6 = new Compartment("6", 3350 , 7.707 , 2.357 , 1.057 , 651 , 8.27 , 9.63 );
60          Compartment comp7 = new Compartment("7", 3500 , 9.092 , 3.742 , 2.442 , 681 , 9.63 , 11.05 );
61
62          // Create ArrayList for Compartment Objects
63          ArrayList<Compartment> compList = new ArrayList<Compartment>();
64          compList.add(comp0);
65          compList.add(comp1);
66          compList.add(comp2);
67          compList.add(comp3);
68          compList.add(comp4);
69          compList.add(comp5);
70          compList.add(comp6);

71           compList.add(comp7);
72
73           // ArrayList for AXLES_POS in Vehicle Object
74           ArrayList<Double> axlesPos = new ArrayList<Double>();
75           axlesPos.add(1.245);
76           axlesPos.add(6.595);
77           axlesPos.add(7.895);
78
79           //Vehicle Object Initialization
80           Vehicle v1 = new Vehicle(compList, 16000, 5.35, 1.3, 9800, axlesPos, 7191,11009);
81
82           //OrderItem ArrayList Objects initialization
83           ArrayList<OrderItem> orderItemList1 = new ArrayList<OrderItem>(); // List for Order "18015040"
84           orderItemList1.add(orderItem1);
85           order1 = new Order(order1, orderItemList1);
86           order1.setOrderWeight(orderItemList1); // Setting order_weight for order1
87
88           ArrayList<OrderItem> orderItemList2 = new ArrayList<OrderItem>(); // List for Order "507702"
89           orderItemList2.add(orderItem2);
90           order2 = new Order(order2, orderItemList2);
91           order2.setOrderWeight(orderItemList2); // Setting order_weight for order2
92
93           ArrayList<OrderItem> orderItemList3 = new ArrayList<OrderItem>(); // List for Order "18011583"
94           orderItemList3.add(orderItem3);
95           orderItemList3.add(orderItem4);
96           order3 = new Order(order3, orderItemList3);
97           order3.setOrderWeight(orderItemList3); // Setting order_weight for order3
98
99           ArrayList<OrderItem> orderItemList4 = new ArrayList<OrderItem>(); // List for Order "505686"
100          orderItemList4.add(orderItem5);
101          order4 = new Order(order4, orderItemList4);
102          order4.setOrderWeight(orderItemList4); // Setting order_weight for order4
103
104          ArrayList<OrderItem> orderItemList5 = new ArrayList<OrderItem>(); // List for Order "505842"
105          orderItemList5.add(orderItem6);
106          order5 = new Order(order5, orderItemList5);
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
        order5.setOrderWeight(orderItemList5); // Setting order_weight for order5

        // Create ArrayList containing all Orders respecting the optimal route
        ArrayList<Order> allOrders = new ArrayList<Order>();
        allOrders.add(order5);
        allOrders.add(order2);
        allOrders.add(order1);
        allOrders.add(order4);
        allOrders.add(order3);

        // problemInput Object Initialization
        problemInput pprob = new problemInput(shipToList, v1, allOrders, orderItemList, compList);

        Model mdl1 = new Model();
        mdl1.MathLoadingOrders_with3AxleConstr_withOscillation_FlowMeter_TYPA(v1, pprob);

        Model_2 mdl2 = new Model_2();
        mdl2.MathLoadingOrders_with3AxleConstr_withOscillation_NoFlowMeter_TYPA(v1, pprob);

        System.out.println(knnMethod(shipToList));

    }


    // KNN Method (ShipTo)

    public static ArrayList<Integer> knnMethod(ArrayList<ShipTo> allShipTo) {
        double la = 38.03677;
        double lo = 23.61218;
        int counterDepot = 0;
        ArrayList<Integer> distanceSequence = new ArrayList<Integer>();

        for(int i =0; i < allShipTo.size(); i ++) {

            if(counterDepot == 0) {
                ArrayList<Double> distanceDepot = new ArrayList<Double>();
                Map<Double, Integer> mapDepot = new HashMap<Double, Integer>();
                for (int j = 0; j < allShipTo.size(); j ++) {
                    double dist = getDistance(la, lo, allShipTo.get(j));
                    distanceDepot.add(dist);
                    mapDepot.put(dist, j);
                }

                distanceSequence.add(mapDepot.get(Collections.min(distanceDepot)) +1);
                counterDepot +=1;
            }

            // get last item of distanceSequence
            int last = distanceSequence.get(distanceSequence.size()-1);
            //System.out.println("Last is " + last);
            allShipTo.get(last-1).setIsVisited(true); // set previously visited node as visited

            // create temp ArrayList
            ArrayList<Double> distanceNodes = new ArrayList<Double>();
            Map<Double, Integer> mapNodes = new HashMap<Double, Integer>();

            for(int k = 0; k < allShipTo.size(); k ++) {
                if(!allShipTo.get(k).getIsVisited()) {
                double distance = getDistance(allShipTo.get(last-1), allShipTo.get(k));
                    distanceNodes.add(distance);
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
                    mapNodes.put(distance, k);
                }

                // check if all nodes are visited
                int counter = allShipTo.size();
                for(int x = 0; x <allShipTo.size();x++) {
                    if (allShipTo.get(x).getIsVisited() == true) {
                        counter --;
                    }
                }
                if (counter ==0) {
                    return distanceSequence;
                }
            }

            distanceSequence.add(mapNodes.get(Collections.min(distanceNodes)) +1);

        }
        return distanceSequence;


    }

    public static double getDistance(ShipTo a, ShipTo b) {
        double R = 6371;
        double dLat = (b.getlatitude()-a.getlatitude())*(Math.PI/180);
        double dLon = (b.getlongtitude()-a.getlongtitude())*(Math.PI/180);
        double A =  Math.sin(dLat/2) * Math.sin(dLat/2) + Math.cos((a.getlatitude()*(Math.PI/180))) *
                Math.cos((b.getlatitude()*(Math.PI/180))) *
                Math.sin(dLon/2) * Math.sin(dLon/2);
        double c = 2 * Math.atan2(Math.sqrt(A), Math.sqrt(1-A));
        double d = R * c;
        return d;
    }

     public static double getDistance(double la, double lo, ShipTo b) {
         double R = 6371;
         double dLat = (b.getlatitude()-la)*(Math.PI/180);
         double dLon = (b.getlongtitude()-lo)*(Math.PI/180);
         double A =  Math.sin(dLat/2) * Math.sin(dLat/2) + Math.cos((la*(Math.PI/180))) *
                 Math.cos((b.getlatitude()*(Math.PI/180))) *
                 Math.sin(dLon/2) * Math.sin(dLon/2);
         double c = 2 * Math.atan2(Math.sqrt(A), Math.sqrt(1-A));
         double d = R * c;
         return d;
     }
}
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```java

public class ShipTo {

    private String ShipToId;
    private double longtitude;
    private double latitude;
    private boolean isVisited;

    public ShipTo(){}

    public ShipTo(String ShipToId,double longtitude,double latitude){

        this();
        this.setShipToId(ShipToId);
        this.setlongtitude(longtitude);
        this.setlatitude(latitude);
        this.isVisited = false;
    }

    public String getShipToId(){
        return ShipToId;
    }

    public void setShipToId(String ShipToId){
        this.ShipToId=ShipToId;
    }

    public double getlongtitude() {
        return longtitude;
    }

    public void setlongtitude(double longtitude) {
        this.longtitude = longtitude;
    }


    public double getlatitude() {
        return latitude;
    }

    public void setlatitude(double latitude) {
        this.latitude = latitude;
    }

    public void setIsVisited(boolean isVisited) {
        this.isVisited = isVisited;
    }

    public boolean getIsVisited() {
        return isVisited;
    }

}
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```java

import java.util.ArrayList;

public class Order {

    private String Order_Number;
    private String ShipToId;
    private double order_weight;
    private ArrayList<OrderItem> lineItemList=new ArrayList<OrderItem>();


    public Order() {}

    public Order(String Order_Number,String ShipToId)
    {
        this.setOrderNumber(Order_Number);
        this.setShipToId(ShipToId);
    }

    public Order(Order order,ArrayList<OrderItem> order_elements)
    {
        this.setOrderNumber(order.getOrder_Number());
        this.setShipToId(order.getShipToId());
        this.setOLIList(order_elements);
    }

    public double getOrderWeight()
    {
        return this.order_weight;
    }

   public void setOrderWeight(ArrayList<OrderItem> order_elements){

       double order_weight=0;

       for(int i=0; i<order_elements.size(); i++){
           order_weight=order_weight+order_elements.get(i).getQuantity()*order_elements.get(i).getSpecificWeight();
       }

       this.order_weight=order_weight;
   }

   public String getOrder_Number() {
       return Order_Number;
   }

   public void setOrderNumber(String Order_Number) {
       this.Order_Number = Order_Number;
   }

   public String getShipToId() {
       return ShipToId;
   }

   public void setShipToId(String ShipToId) {
       this.ShipToId = ShipToId;
   }

     public OrderItem getOrderLineItem(int i){
            return this.lineItemList.get(i);
     }

```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
63      public void setOLIList(ArrayList<OrderItem> oli_list)
64      {
65          this.lineItemList=oli_list;
66      }
67
68
69      public int getOrderLineItemSize() {
70             return this.lineItemList.size();
71        }
72
73      public ArrayList<OrderItem> getOrderLineItems(){
74            return lineItemList;
75      }
76
77
78
79  }
```

```
1
2   public class OrderItem {
3
4       private String DeliveryItem_SN;
5       private int quantity;
6       private double Specific_Weight;
7
8
9       public OrderItem(){}
10
11      public OrderItem(String DeliveryItem_SN,int quantity,double specific_weight) {
12          this.DeliveryItem_SN=DeliveryItem_SN;
13          this.quantity=quantity;
14          this.Specific_Weight=specific_weight;
15      }
16
17
18      public void setDeliveryItem_SN(String DeliveryItem_SN) {
19          this.DeliveryItem_SN= DeliveryItem_SN;
20      }
21
22      public String getDeliveryItem_SN() {
23          return DeliveryItem_SN;
24      }
25
26      public void setSpecificWeight(double specific_weight){
27          this.Specific_Weight=specific_weight;
28      }
29
30      public double getSpecificWeight(){
31          return Specific_Weight;
32      }
33
34      public int getQuantity() {
35          return quantity;
36      }
37
38       public void setQuantity(int quantity) {
39           this.quantity = quantity;
40       }
41
42
43  }
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
1
2  import java.util.ArrayList;
3
4  public class Vehicle {
5
6      private ArrayList<Compartment> compartments=new ArrayList<Compartment>();
7      private double Va_Gross_Weight;
8      private double Va_Tare;
9      private double Va_Max_Net;
10     private double Max1;
11     private double Max2;
12     private double L12;
13     private double L23;
14     private ArrayList<Double> axlesPos = new ArrayList<Double>();
15
16     public Vehicle()
17     {
18     }
19
20     public Vehicle(ArrayList<Compartment> compartments,
21                    double Va_MaxNet, double L12, double L23, double Va_Tare, ArrayList<Double> axlesPos, double Max1, double Max2) {
22         this();
23
24         this.compartments=compartments;
25         this.setVA_Max_Net(Va_MaxNet);
26         this.setL1_2(L12);
27         this.setL2_3(L23);
28         this.setVA_Tare(Va_Tare);
29         this.axlesPos = axlesPos;
30         this.setMax_1(Max1);
31         this.setMax_2(Max2);
32     }
```

```
33
34         public void setMax_1(double max1)
35         {
36             this.Max1=max1;
37         }
38
39         public double getMax_1()
40         {
41                 return this.Max1;
42         }
43
44         public void setMax_2(double max2)
45         {
46             this.Max2=max2;
47         }
48
49         public double getMax_2()
50         {
51                 return this.Max2;
52         }
53
54         public void setL1_2(double l12)
55         {
56             this.L12=l12;
57         }
58
59         public double getL1_2()
60         {
61                 return this.L12;
62         }
63
64         public void setL2_3(double l23)
65         {
66             this.L23=l23;
67         }
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
68
69          public double getL2_3()
70          {
71                  return this.L23;
72          }
73
74          public ArrayList<Double> getAxlePos(){
75              return this.axlesPos;
76          }
77
78      public void setVA_Max_Net(double va_max_net){
79          this.Va_Max_Net=va_max_net;
80      }
81
82      public double getVA_Max_Net(){
83          return this.Va_Max_Net;
84      }
85
86      public void setCompartments(ArrayList<Compartment> compartments){
87          this.compartments.addAll(compartments);
88      }
89
90      public ArrayList<Compartment> getCompartments(){
91          return compartments;
92      }
93
94      public double getVA_Tare(){
95          return this.Va_Tare;
96      }
97
98      public void setVA_Tare(double VA_Tare){
99          this.Va_Tare=VA_Tare;
100     }

101
102     public double getVA_Gross_Weight(){
103         return this.Va_Gross_Weight;
104     }
105
106     public void setVA_Gross_Weight(double Vehicle_Gross_Weight){
107         this.Va_Gross_Weight=Vehicle_Gross_Weight;
108     }
109
110
111
112 }
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
1
2  public class Compartment {
3
4  private String VehicleCompartment_Id;
5  private int VehicleCompartment_Capacity;
6  private double dist_axle1;
7  private double dist_axle2;
8  private double dist_axle3;
9  private double comp_weight;
10 private double startPos;
11 private double endPos;
12
13 public void setDist_axle1(double dist1){
14 this.dist_axle1=dist1;
15 }
16 public double getDist_axle1(){
17 return dist_axle1;
18 }
19
20 public void setDist_axle2(double dist2){
21 this.dist_axle2=dist2;
22 }
23 public double getDist_axle2(){
24 return dist_axle2;
25 }
26
27 public void setDist_axle3(double dist3){
28 this.dist_axle3=dist3;
29 }
30 public double getDist_axle3(){
31 return dist_axle3;
32 }
33
34 public void setComp_weight(double compweight){
35 this.comp_weight=compweight;
36 }
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
public double getComp_weight(){
return comp_weight;
}
public double getStartPos(){
    return this.startPos;
}
public double getEndPos(){
    return this.endPos;
}
public void setStartPos(double x){
    this.startPos = x;
}
public void setEndPos(double x){
    this.endPos = x;
}

    public Compartment(String VehicleCompartment_Id,int VehicleCompartment_Capacity, double dist_axle1,
            double dist_axle2,double dist_axle3, double comp_weight, double startPos, double endPos) {
        this.setVehicleCompartment_Id(VehicleCompartment_Id);
        this.setVehicleCompartment_Capacity(VehicleCompartment_Capacity);
        this.setDist_axle1(dist_axle1);
        this.setDist_axle2(dist_axle2);
        this.setDist_axle3(dist_axle3);
        this.setComp_weight(comp_weight);
        this.setStartPos(startPos);
        this.setEndPos(endPos);
    }

    public Compartment(Compartment comp){
        this.setVehicleCompartment_Capacity(comp.getVehicleCompartment_Capacity());
    }

    public int getVehicleCompartment_Capacity() {
        return VehicleCompartment_Capacity;
    }


     public void setVehicleCompartment_Capacity(int VehicleCompartment_Capacity) {
         this.VehicleCompartment_Capacity = VehicleCompartment_Capacity;
     }

     public void setVehicleCompartment_Id(String VehicleCompartment_Id) {
         this.VehicleCompartment_Id = VehicleCompartment_Id;
     }

}
```

```
import java.util.*;

public class problemInput {

  ArrayList<ShipTo> allLandmarks;
  Vehicle v;
  ArrayList<Order> all_orders;
  ArrayList<OrderItem> all_orderItems;
  ArrayList<Compartment> all_compartments;


  public problemInput(){

  }
  public problemInput(ArrayList<ShipTo> vallLandmarks,Vehicle v,ArrayList<Order> vall_orders,
          ArrayList<OrderItem> vall_orderItems,ArrayList<Compartment> vall_compartments){
      this.all_orderItems=vall_orderItems;
      this.v = v;
      this.all_orders = vall_orders;
      this.all_compartments = vall_compartments;
  }


public ArrayList<Order> get_all_orders(){
    return this.all_orders;
}

public ArrayList<OrderItem> get_all_order_items(){
    return this.all_orderItems;
}

```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
public ArrayList<Compartment> get_all_compartments(){
    return this.all_compartments;

}

// Supporting Methods

// Get total weight of all orders
public double getTotalWeight() {
    double total_weight = 0;
    for(int i =0; i <get_all_orders().size();i++) {
        total_weight += get_all_orders().get(i).getOrderWeight();
    }
    return total_weight;
}


}
```

```
import java.io.IOException;
import java.util.ArrayList;

import ilog.concert.*;
import ilog.cplex.*;

public class Model {


    public void MathLoadingOrders_with3AxleConstr_withOscillation_FlowMeter_TYPA(Vehicle veh,problemInput pprob) throws IloException,IOException{


            // Parameters initialization
            double B = veh.getVA_Max_Net();
            double BT = veh.getVA_Tare();
            double a_percent_upper=0.20;
            double a_percent_lower=0.20;
            double B1 = 2809;
            double B4 = 6991;
            double xt = 5.525;
            double L123 = veh.getL1_2() + (veh.getL2_3()/2);
            double p = 0.30;

            double totalWeight = pprob.getTotalWeight();

            if ((totalWeight)<veh.getVA_Max_Net()){
            ArrayList<Order> vlOrders = pprob.get_all_orders();
            int nComp=veh.getCompartments().size();

            ArrayList<ArrayList<OrderItem>> elementsByOrder = new ArrayList<ArrayList<OrderItem>>();
            ArrayList<OrderItem> elementsList = new ArrayList<OrderItem>();
            ArrayList<Integer> elementsSequence = new ArrayList<Integer>();
            ArrayList<ArrayList<Integer>> elementsRefByOrder= new ArrayList<ArrayList<Integer>>();
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
35
36          int iref=0;
37          for (int i=0;i<vlOrders.size();i++){
38              for (int l=0;l<vlOrders.get(i).getOrderLineItems().size();l++){
39                  elementsSequence.add(i);
40                  elementsList.add(vlOrders.get(i).getOrderLineItems().get(l));
41              }
42              elementsByOrder.add(vlOrders.get(i).getOrderLineItems());
43              ArrayList<Integer> tempIntList = new ArrayList<Integer>();
44              for (int ielem=1;ielem<=vlOrders.get(i).getOrderLineItems().size();ielem++){
45                  tempIntList.add(iref);
46                  iref=iref+1;
47              }
48              elementsRefByOrder.add(tempIntList);
49          }
50
51          int nElements = elementsList.size();
52
53          ArrayList<String> fueltypes =new ArrayList<String>();
54          ArrayList<Double> fuelDensity =new ArrayList<Double>();
55          int fuel_no=1;
56          int samefuel;
57          String fuel_type=elementsList.get(0).getDeliveryItem_SN();
58          double fuelDens= elementsList.get(0).getSpecificWeight();
59          fueltypes.add(fuel_type);
60          fuelDensity.add(fuelDens);
61          for (int i=1;i<nElements;i++){
62              samefuel=0;
63                fuel_type=elementsList.get(i).getDeliveryItem_SN();
64                fuelDens=  elementsList.get(i).getSpecificWeight();
65
66                for (int counter = 0; counter < fueltypes.size(); counter++) {
67                    if (fueltypes.get(counter).equals(elementsList.get(i).getDeliveryItem_SN())){
68                        samefuel=samefuel+1;
69                    }
70                }
71
72                if (samefuel==0){
73
74                      fueltypes.add(elementsList.get(i).getDeliveryItem_SN());
75                      fuelDensity.add(fuelDens);
76                      System.out.println("fuel_type: "+fuel_type+" density: "+fuelDens);
77                      fuel_no=fuel_no+1;
78                }
79      }
80
81
82  //cplex
83            try
84            {
85            IloCplex cplex = new IloCplex();
86            //Decision Variables
87            IloNumVar[][] xVar = new IloNumVar[nElements][nComp];
88            IloNumVar[][] yVar = new IloNumVar[nElements][nComp];
89            IloNumVar[] dVar = new IloNumVar[nElements];
90            int nRouteLinks = vlOrders.size()+1;
91            IloNumVar[][] wVar = new IloNumVar[nRouteLinks][nComp];
92            IloNumVar[][][] phiVar = new IloNumVar[nRouteLinks][nComp][2];
93            IloNumVar[][] fVar = new IloNumVar[fuel_no][nComp];
94
95            for(int k=0;k<nComp;k++){
96            for (int j=0; j<fuel_no; j++)   {
97                    fVar[j][k] = cplex.boolVar();
98                }
99            }
100
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
101             for (int iFromNode=0;iFromNode<nRouteLinks;iFromNode++){
102                 for (int j=0; j<nComp; j++) {
103                 for(int k=0;k<2;k++){
104                     phiVar[iFromNode][j][k] = cplex.boolVar();
105                 }
106                 }
107                 }
108 
109             for (int iFromNode=0;iFromNode<nRouteLinks;iFromNode++){
110                 for(int j=0; j<nComp; j++){
111                     wVar[iFromNode][j] = cplex.numVar(0, Double.MAX_VALUE,IloNumVarType.Float);
112                 }
113             }
114 
115 
116             for (int i=0; i<nElements; i++){
117                 for(int j=0; j<nComp; j++){
118                     yVar[i][j] = cplex.numVar(0, Double.MAX_VALUE,IloNumVarType.Float);
119                     xVar[i][j] = cplex.boolVar();
120                 }
121             }
122 
123             for (int i=0; i<nElements; i++){
124                 dVar[i] = cplex.numVar(0, Double.MAX_VALUE,IloNumVarType.Float);
125 
126             }
127 
128             System.out.println("CONSTRAINT 4");
129             IloLinearNumExpr[] constraint_4 = new IloLinearNumExpr[1];
130             constraint_4[0] = cplex.linearNumExpr();
131             for(int j=0;j<nComp;j++){
132                 for (int i=0;i<elementsList.size();i++){
133                     double specific_weight =elementsList.get(i).getSpecificWeight();
134                     constraint_4[0].addTerm(specific_weight,yVar[i][j]);

135                  }
136              }
137                  cplex.addLe(constraint_4[0],veh.getVA_Max_Net());
138 
139                  System.out.println("CONSTRAINT 8");
140                  IloLinearNumExpr[] constraint_8 = new IloLinearNumExpr[nElements];
141                   for (int i=0;i<elementsList.size();i++){
142                       double element_volume = elementsList.get(i).getQuantity();
143                       constraint_8[i] = cplex.linearNumExpr();
144                       for (int j=0;j<nComp;j++){
145                           constraint_8[i].addTerm(1, yVar[i][j]);
146                       }
147                       constraint_8[i].addTerm(-1, dVar[i]);
148 
149                      cplex.addLe(constraint_8[i],element_volume);
150                   }
151 
152                      System.out.println("CONSTRAINT 9");
153                      IloLinearNumExpr[] constraint_9 = new IloLinearNumExpr[nElements];
154                       for (int i=0;i<elementsList.size();i++){
155                           double element_volume = elementsList.get(i).getQuantity(); //πr
156                           constraint_9[i] = cplex.linearNumExpr();
157                           for (int j=0;j<nComp;j++){
158                               constraint_9[i].addTerm(1, yVar[i][j]);
159                           }
160                           constraint_9[i].addTerm(1, dVar[i]);
161 
162                          cplex.addGe(constraint_9[i],element_volume);
163                       }
164 
165                          System.out.println("CONSTRAINT 10a");
166                          IloLinearNumExpr[] constraint_10a = new IloLinearNumExpr[nElements];
167                          for (int i=0;i<nElements;i++){
168                              constraint_10a[i] = cplex.linearNumExpr();
169                              for(int j=0;j<nComp;j++){
170                                  constraint_10a[i].addTerm(1,yVar[i][j]);
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
171                         }
172                         double upper_limit = elementsList.get(i).getQuantity()*(1+a_percent_upper);
173                         cplex.addLe(constraint_10a[i],upper_limit);
174                     }
175 
176                     System.out.println("CONSTRAINT 10b");
177                     IloLinearNumExpr[] constraint_10b = new IloLinearNumExpr[nElements];
178                     for (int i=0;i<nElements;i++){
179                         constraint_10b[i] = cplex.linearNumExpr();
180                         for(int j=0;j<nComp;j++){
181                             constraint_10b[i].addTerm(1,yVar[i][j]);
182                         }
183                         double lower_limit = elementsList.get(i).getQuantity()*(1-a_percent_lower);
184                         cplex.addGe(constraint_10b[i],lower_limit);
185                     }
186 
187                     System.out.println("CONSTRAINT 11");
188                     IloLinearNumExpr[] constraint_11 = new IloLinearNumExpr[nComp];
189                     for(int i=0; i<nComp; i++){
190                         constraint_11[i]=cplex.linearNumExpr();
191                         constraint_11[i].addTerm(1.0, wVar[0][i]);
192                         for (int r=0;r<nElements;r++){
193                             double specific_weight = elementsList.get(r).getSpecificWeight();
194                             constraint_11[i].addTerm((-1*specific_weight), yVar[r][i]);
195                         }
196                         cplex.addEq(constraint_11[i],0);
197                     }
198 
199                     System.out.println("CONSTRAINT 12");
200                     IloLinearNumExpr[] constraint_12 = new IloLinearNumExpr[nComp];
201                     for(int i=0; i<nComp; i++){
202                         constraint_12[i]=cplex.linearNumExpr();
203                         constraint_12[i].addTerm(1.0, wVar[nRouteLinks-1][i]);//τελευταίος πελάτης
204                         cplex.addEq(constraint_12[i],0);
205                     }
206 
207                     System.out.println("CONSTRAINT 13");
208                     IloLinearNumExpr[][] constraint_13 = new IloLinearNumExpr[nComp][nRouteLinks];
209                     for (int iFromNode=1;iFromNode<nRouteLinks-1;iFromNode++){
210                         for(int c=0; c<nComp; c++){
211                             constraint_13[c][iFromNode]=cplex.linearNumExpr();
212                             constraint_13[c][iFromNode].addTerm(1.0, wVar[iFromNode-1][c]);
213                             constraint_13[c][iFromNode].addTerm(-1.0, wVar[iFromNode][c]);
214 
215                             for (int icount=0;icount<elementsRefByOrder.get(iFromNode-1).size();icount++){
216                                 int r=elementsRefByOrder.get(iFromNode-1).get(icount);
217                                 double specific_weight =elementsList.get(r).getSpecificWeight();
218                                 constraint_13[c][iFromNode].addTerm((-1.0*specific_weight), yVar[r][c]);
219                             }
220                             cplex.addEq(constraint_13[c][iFromNode],0);
221                         }
222                     }
223 
224                     System.out.println("CONSTRAINT 17");
225                     IloLinearNumExpr[] constraint_17 = new IloLinearNumExpr[nComp];
226                     for(int c=0; c<nComp; c++){
227                         constraint_17[c]=cplex.linearNumExpr();
228 
229                         for(int j=0;j<fuel_no;j++){
230 
231                             constraint_17[c].addTerm(1,fVar[j][c]);
232                         }
233                          cplex.addLe(constraint_17[c],1);
234                     }
235 
236                     System.out.println("CONSTRAINT 18");
237                     IloLinearNumExpr[][] constraint_18 = new IloLinearNumExpr[fuel_no][nComp];
238                     for(int c=0; c<nComp; c++){
239                         for(int i=0;i<fuel_no;i++){
240                             constraint_18[i][c]=cplex.linearNumExpr();
241 
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
                    for(int k=0; k<nElements; k++){
                    if(fueltypes.get(i).equals(elementsList.get(k).getDeliveryItem_SN())){
                        constraint_18[i][c].addTerm(-1,xVar[k][c]);

                    }
                    }
                    constraint_18[i][c].addTerm(nElements,fVar[i][c]);


                cplex.addGe(constraint_18[i][c],0);
                }
            }

            System.out.println("CONSTRAINT 20");
            IloLinearNumExpr[][] constraint_20 = new IloLinearNumExpr[fuel_no][nComp];
            for (int c=0;c<nComp;c++){
                for(int i=0;i<fuel_no;i++){
                    constraint_20[i][c] = cplex.linearNumExpr();
                    for(int k=0; k<nElements; k++){
                        if(fueltypes.get(i).equals(elementsList.get(k).getDeliveryItem_SN())){

                            constraint_20[i][c].addTerm(1,yVar[k][c]);
                        }

                    }
                    Compartment vcomp = pprob.get_all_compartments().get(c);
                    constraint_20[i][c].addTerm(-vcomp.getVehicleCompartment_Capacity(),fVar[i][c]);


                    cplex.addLe(constraint_20[i][c],0);
                }

            }
```

```
        IloLinearNumExpr[][] constraint_19 = new IloLinearNumExpr[fuel_no][nComp];
        for(int c=0; c<nComp; c++){
            for(int i=0;i<fuel_no;i++){
                constraint_19[i][c]=cplex.linearNumExpr();

                for(int k=0; k<nElements; k++){
                    if(fueltypes.get(i).equals(elementsList.get(k).getDeliveryItem_SN())){
                        constraint_19[i][c].addTerm(-1,xVar[k][c]);

                    }
                        }
                constraint_19[i][c].addTerm(1,fVar[i][c]);

                cplex.addLe(constraint_19[i][c],0);

            }
        }

        System.out.println("CONSTRAINT 5");
        IloLinearNumExpr[][] constraint_5 = new IloLinearNumExpr[nElements][nComp];
        for(int c=0; c<nComp; c++){

            for(int k=0; k<nElements; k++){
                constraint_5[k][c]=cplex.linearNumExpr();

                for(int i=0;i<fuel_no;i++){

                if(fueltypes.get(i).equals(elementsList.get(k).getDeliveryItem_SN())){

                constraint_5[k][c].addTerm(-1,yVar[k][c]);
                constraint_5[k][c].addTerm(pprob.get_all_compartments().get(c).getVehicleCompartment_Capacity(),xVar[k][c]);
                }}
                 cplex.addGe(constraint_5[k][c],0);

            }


              }

          System.out.println("CONSTRAINT 6");
          IloLinearNumExpr[][] constraint_6 = new IloLinearNumExpr[nElements][nComp];
          for(int c=0; c<nComp; c++){

                for(int k=0; k<nElements; k++){
                    constraint_6[k][c]=cplex.linearNumExpr();

                    for(int i=0;i<fuel_no;i++){

                    if(fueltypes.get(i).equals(elementsList.get(k).getDeliveryItem_SN())){

                    constraint_6[k][c].addTerm(1,yVar[k][c]);
                    constraint_6[k][c].addTerm(-1,xVar[k][c]);
                    }}
                     cplex.addGe(constraint_6[k][c],0);
                }

                }

                        System.out.println("CONSTRAINT 22");
                        IloLinearNumExpr[][] constraint_22 = new IloLinearNumExpr[nRouteLinks][nComp];
                        for (int iFromNode=0;iFromNode<nRouteLinks-1;iFromNode++){
                            for(int i=0; i<nComp; i++){
                                Compartment vcomp = pprob.get_all_compartments().get(i);

                                    constraint_22[iFromNode][i]=cplex.linearNumExpr();
                                    double maxcap = 0.8 *vcomp.getVehicleCompartment_Capacity() ;
                                    for(int j=0;j<fuel_no;j++){
                                        constraint_22[iFromNode][i].addTerm(maxcap*fuelDensity.get(j), fVar[j][i]);
                                    }
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
                                        constraint_22[iFromNode][i].addTerm(-1, wVar[iFromNode][i]);
                                        constraint_22[iFromNode][i].addTerm(vcomp.getVehicleCompartment_Capacity() ,
                                                phiVar[iFromNode][i][0]);

                                cplex.addLe(constraint_22[iFromNode][i],vcomp.getVehicleCompartment_Capacity());

                                    }
                                }

                                System.out.println("CONSTRAINT 22");
                                IloLinearNumExpr[][] constraint_23 = new IloLinearNumExpr[nRouteLinks][nComp];
                                for (int iFromNode=0;iFromNode<nRouteLinks-1;iFromNode++){
                                for(int i=0; i<nComp; i++){

                                    Compartment vcomp = pprob.get_all_compartments().get(i);

                                        constraint_23[iFromNode][i]=cplex.linearNumExpr();
                                        for(int j=0;j<fuel_no;j++){
                                            constraint_23[iFromNode][i].addTerm(-0.2*vcomp.getVehicleCompartment_Capacity()*
                                                    fuelDensity.get(j), fVar[j][i]);
                                        }
                                        constraint_23[iFromNode][i].addTerm(1, wVar[iFromNode][i]);
                                        constraint_23[iFromNode][i].addTerm(vcomp.getVehicleCompartment_Capacity() ,
                                                phiVar[iFromNode][i][1]);

                                cplex.addLe(constraint_23[iFromNode][i],vcomp.getVehicleCompartment_Capacity());
                                    }
                                    }

                                System.out.println("CONSTRAINT 24");
                                IloLinearNumExpr[][] constraint_24 = new IloLinearNumExpr[nRouteLinks][nComp];
                                for (int iFromNode=0;iFromNode<nRouteLinks-1;iFromNode++){

                                for(int i=0; i<nComp; i++){
                                constraint_24[iFromNode][i]=cplex.linearNumExpr();
                                constraint_24[iFromNode][i].addTerm(1,phiVar[iFromNode][i][0]);
                                constraint_24[iFromNode][i].addTerm(1,phiVar[iFromNode][i][1]);

                                cplex.addEq(constraint_24[iFromNode][i],1);

                                    }
                                    }

                            IloLinearNumExpr[] constraint_25 = new IloLinearNumExpr[nRouteLinks];
                                for (int iFromNode=0;iFromNode<nRouteLinks-1;iFromNode++){

                                        constraint_25[iFromNode]=cplex.linearNumExpr();

                                        for(int c=0; c<nComp; c++){
                                            constraint_25[iFromNode].addTerm(pprob.get_all_compartments().get(c).getDist_axle1()/L123,
                                                    wVar[iFromNode][c]);
                                        }

                                        cplex.addLe(constraint_25[iFromNode],veh.getMax_2());
                                    }

                            IloLinearNumExpr[] constraint_26 = new IloLinearNumExpr[nRouteLinks];
                                for (int iFromNode=0;iFromNode<nRouteLinks-1;iFromNode++){

                                        constraint_26[iFromNode]=cplex.linearNumExpr();

                                        for(int c=0; c<nComp; c++){
                                            constraint_26[iFromNode].addTerm((7.245-(pprob.get_all_compartments().get(c).getStartPos()+
                                                    pprob.get_all_compartments().get(c).getEndPos())/2)/L123,wVar[iFromNode][c]);
                                        }

                                        cplex.addLe(constraint_26[iFromNode],veh.getMax_1());
                                    }


                             IloLinearNumExpr[] constraint_27 = new IloLinearNumExpr[nRouteLinks];
                                for (int iFromNode=0;iFromNode<nRouteLinks-1;iFromNode++){

                                        constraint_27[iFromNode]=cplex.linearNumExpr();

                                        for(int c=0; c<nComp; c++){
                                            constraint_27[iFromNode].addTerm(pprob.get_all_compartments().get(c).getDist_axle1()/L123,
                                                    wVar[iFromNode][c]);
                                        }

                                        cplex.addGe(constraint_27[iFromNode],-B4);
                                    }

                             IloLinearNumExpr[] constraint_28 = new IloLinearNumExpr[nRouteLinks];
                                for (int iFromNode=0;iFromNode<nRouteLinks-1;iFromNode++){

                                        constraint_28[iFromNode]=cplex.linearNumExpr();

                                        for(int
                                                c=0; c<nComp; c++){
                                            constraint_28[iFromNode].addTerm((7.245-(pprob.get_all_compartments().get(c).getStartPos()
                                                    +pprob.get_all_compartments().get(c).getEndPos())/2)/L123,wVar[iFromNode][c]);
                                        }

                                        cplex.addGe(constraint_28[iFromNode],-B1);
                                    }

                                IloLinearNumExpr[] constraint_29 = new IloLinearNumExpr[nRouteLinks];
                                for (int iFromNode=0;iFromNode<nRouteLinks-1;iFromNode++){

                                        constraint_29[iFromNode]=cplex.linearNumExpr();
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
                                        for(int c=0; c<nComp; c++){
                                            constraint_29[iFromNode].addTerm(pprob.get_all_compartments().get(c).getDist_axle1()/L123,
                                                    wVar[iFromNode][c]);
                                            constraint_29[iFromNode].addTerm(-0.25,wVar[iFromNode][c]);
                                        }

                                        cplex.addGe(constraint_29[iFromNode],0.25*veh.getVA_Tare()-B4);
                                    }

                                IloLinearNumExpr[] constraint_30 = new IloLinearNumExpr[nRouteLinks];
                                for (int iFromNode=0;iFromNode<nRouteLinks-1;iFromNode++){

                                        constraint_30[iFromNode]=cplex.linearNumExpr();

                                        for(int c=0; c<nComp; c++){
                                            constraint_30[iFromNode].addTerm((pprob.get_all_compartments().get(c).getDist_axle1()+
                                                    veh.getAxlePos().get(0)),wVar[iFromNode][c]);
                                            constraint_30[iFromNode].addTerm(-(xt+p),wVar[iFromNode][c]);
                                        }

                                        cplex.addLe(constraint_30[iFromNode],0);
                                        }

                                IloLinearNumExpr[] constraint_31 = new IloLinearNumExpr[nRouteLinks];
                                for (int iFromNode=0;iFromNode<nRouteLinks-1;iFromNode++){

                                        constraint_31[iFromNode]=cplex.linearNumExpr();

                                        for(int c=0; c<nComp; c++){
                                            constraint_31[iFromNode].addTerm(-((pprob.get_all_compartments().get(c).getDist_axle1()+
                                                    veh.getAxlePos().get(0))),wVar[iFromNode][c]);
                                            constraint_31[iFromNode].addTerm((xt-p),wVar[iFromNode][c]);
                                        }


                                             cplex.addLe(constraint_31[iFromNode],0);
                                             }


// OBJECTIVE FUNCTION


                                  IloLinearNumExpr F1 = cplex.linearNumExpr();
                                      for (int i=0;i<elementsList.size();i++){
                                          F1.addTerm(1000,dVar[i]);
                                      }


                                      for (int i=0;i<elementsList.size();i++){
                                          for (int j=0;j<nComp;j++){
                                              for(int k=0;k<fuel_no;k++){
                                                  if(fueltypes.get(k).equals(elementsList.get(i).getDeliveryItem_SN())){
                                                       Compartment vcomp = pprob.get_all_compartments().get(j);
                                                       F1.addTerm(0.01*vcomp.getVehicleCompartment_Capacity(), fVar[k][j]);
                                                       F1.addTerm(-0.01, yVar[i][j]);
                                                  }
                                              }
                                          }
                                      }


                                      cplex.addMinimize(F1);


            double totalDemand=0;
            for (int i=0;i<nElements;i++){
                OrderItem item = elementsList.get(i);
                totalDemand+=item.getQuantity();
            }
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
512                 System.out.println("Total Demand:"+ totalDemand);
513 
514             if (cplex.solve()){
515 
516                 System.out.println(" MODEL SOLVED. ABOUT TO SET UP XPAR,YPAR ");
517                 int[][] xpar = new int[nElements][nComp];
518                 double[][] ypar = new double[nElements][nComp];
519                 double[] totalVolpar = new double[nComp];
520                 double[] UrValues = new double[nElements];
521 
522                 int[][] fpar = new int[fuel_no][nComp];
523                 for (int i=0;i<nElements;i++){
524                     for (int j=0;j<nComp;j++){
525                         xpar[i][j] = (int) Math.round(cplex.getValue(xVar[i][j]));
526                         ypar[i][j] = (int) Math.round(cplex.getValue(yVar[i][j]));
527 
528                         {
529                              System.out.println("y["+i+"]["+j+"]="+ypar[i][j]+ "  x["+i+"]["+j+"]="+xpar[i][j]);
530                         }
531                         totalVolpar[j]=totalVolpar[j]+ypar[i][j];
532 
533                     }
534                     UrValues[i] = (double) Math.round(cplex.getValue(dVar[i]));
535 
536                     System.out.println("orderId: " + (i+1) + ", Ur:" + UrValues[i]);
537                 }
538                 double totalVolparSum = 0;
539                 for(int counterLast = 0; counterLast < totalVolpar.length; counterLast ++) {
540                     totalVolparSum += totalVolpar[counterLast];
541                 }
542                 System.out.println("Total order assigned to the vehicle: " + totalVolparSum);
543                 for (int k=0;k<fuel_no;k++){
544                     for (int j=0;j<nComp;j++){
545                         fpar[k][j] = (int) Math.round(cplex.getValue(fVar[k][j]));
546                         System.out.println("f["+k+"]["+j+"]="+fpar[k][j]);
547                     }
548 
549                 }
550                 System.out.println("Objective Function Value: " + cplex.getObjValue());
551 
552 
553             }
554             else
555             {
556 
557             }
558 
559             cplex.end();
560             }
561             catch(IloException e)
562             {
563                 System.err.println("Concert exception '" + e + "' caught");
564                 e.getStackTrace();
565             }
566         }
567 
568     }
569 
570 }
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```java
import java.io.IOException;
import java.util.ArrayList;

import ilog.concert.*;
import ilog.cplex.*;

public class Model_2{


    public void MathLoadingOrders_with3AxleConstr_withOscillation_NoFlowMeter_TYPA(Vehicle veh,problemInput pprob) throws IloException,IOException{


            // Parameters initialization
            double B = veh.getVA_Max_Net();
            double BT = veh.getVA_Tare();
            double a_percent_upper=0.35;
            double a_percent_lower=0.35;
            double B1 = 2809;
            double B4 = 6991;
            double xt = 5.525;
            double L123 = veh.getL1_2() + (veh.getL2_3()/2);
            double p = 0.35;

            double totalWeight = pprob.getTotalWeight();

            if ((totalWeight)<veh.getVA_Max_Net()){
            ArrayList<Order> vlOrders = pprob.get_all_orders();
            int nComp=veh.getCompartments().size();

            ArrayList<ArrayList<OrderItem>> elementsByOrder = new ArrayList<ArrayList<OrderItem>>();
            ArrayList<OrderItem> elementsList = new ArrayList<OrderItem>();
            ArrayList<Integer> elementsSequence = new ArrayList<Integer>();
            ArrayList<ArrayList<Integer>> elementsRefByOrder= new ArrayList<ArrayList<Integer>>();


                int iref=0;
                for (int i=0;i<vlOrders.size();i++){
                    for (int l=0;l<vlOrders.get(i).getOrderLineItems().size();l++){
                        elementsSequence.add(i);
                        elementsList.add(vlOrders.get(i).getOrderLineItems().get(l));
                    }
                    elementsByOrder.add(vlOrders.get(i).getOrderLineItems());
                    ArrayList<Integer> tempIntList = new ArrayList<Integer>();
                    for (int ielem=1;ielem<=vlOrders.get(i).getOrderLineItems().size();ielem++){
                        tempIntList.add(iref);
                        iref=iref+1;
                    }
                    elementsRefByOrder.add(tempIntList);
                }

                int nElements = elementsList.size();

                ArrayList<String> fueltypes =new ArrayList<String>();
                ArrayList<Double> fuelDensity =new ArrayList<Double>();
                int fuel_no=1;
                int samefuel;
                String fuel_type=elementsList.get(0).getDeliveryItem_SN();
                double fuelDens= elementsList.get(0).getSpecificWeight();
                fueltypes.add(fuel_type);
                fuelDensity.add(fuelDens);
                for (int i=1;i<nElements;i++){
                    samefuel=0;
                      fuel_type=elementsList.get(i).getDeliveryItem_SN();
                      fuelDens=  elementsList.get(i).getSpecificWeight();

                      for (int counter = 0; counter < fueltypes.size(); counter++) {
                          if (fueltypes.get(counter).equals(elementsList.get(i).getDeliveryItem_SN())){
                              samefuel=samefuel+1;
                          }
                      }
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
70
71                  if (samefuel==0){
72
73                        fueltypes.add(elementsList.get(i).getDeliveryItem_SN());
74                        fuelDensity.add(fuelDens);
75                        System.out.println("fuel_type: "+fuel_type+" density: "+fuelDens);
76                        fuel_no=fuel_no+1;
77                  }
78        }
79
80
81 //cplex
82             try
83             {
84             IloCplex cplex = new IloCplex();
85
86             //Decision Variables
87             IloNumVar[][] xVar = new IloNumVar[nElements][nComp];
88             IloNumVar[][] yVar = new IloNumVar[nElements][nComp];
89             IloNumVar[] dVar = new IloNumVar[nElements];
90             int nRouteLinks = vlOrders.size()+1;
91             IloNumVar[][] wVar = new IloNumVar[nRouteLinks][nComp];
92             IloNumVar[][] phiVar = new IloNumVar[nComp][2];
93
94
95             for (int iFromNode=0;iFromNode<nRouteLinks;iFromNode++){
96                 for (int j=0; j<nComp; j++) {
97                 for(int k=0;k<2;k++){
98                     phiVar[j][k] = cplex.boolVar();
99                 }
100                }
101                }
102
103            for (int iFromNode=0;iFromNode<nRouteLinks;iFromNode++){
104                for(int j=0; j<nComp; j++){

105                   wVar[iFromNode][j] = cplex.numVar(0, Double.MAX_VALUE,IloNumVarType.Float);
106               }
107           }
108
109           for (int i=0; i<nElements; i++){
110               for(int j=0; j<nComp; j++){
111                   yVar[i][j] = cplex.numVar(0, Double.MAX_VALUE,IloNumVarType.Float);
112                   xVar[i][j] = cplex.boolVar();
113               }
114           }
115
116           for (int i=0; i<nElements; i++){
117               dVar[i] = cplex.numVar(0, Double.MAX_VALUE,IloNumVarType.Float);
118
119           }
120
121           System.out.println("CONSTRAINT 2");
122           IloLinearNumExpr[] constraint_2 = new IloLinearNumExpr[nComp];
123           for(int i=0; i<nComp; i++){
124               constraint_2[i]=cplex.linearNumExpr();
125               for (int r=0;r<nElements;r++){
126                   constraint_2[i].addTerm(1, xVar[r][i]);
127               }
128                 cplex.addLe(constraint_2[i],1);
129           }
130
131           System.out.println("CONSTRAINT 3");
132           IloLinearNumExpr[] constraint_3 = new IloLinearNumExpr[nComp];
133           for(int i=0; i<nComp; i++){
134               constraint_3[i]=cplex.linearNumExpr();
135               for (int r=0;r<nElements;r++){
136                   constraint_3[i].addTerm(1, yVar[r][i]);
137                   Compartment vcomp = pprob.get_all_compartments().get(i);
138                   constraint_3[i].addTerm(-vcomp.getVehicleCompartment_Capacity(), xVar[r][i]);
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
            }
            cplex.addLe(constraint_3[i],0);
        }

        System.out.println("CONSTRAINT 4");
        IloLinearNumExpr[] constraint_4 = new IloLinearNumExpr[1];
        constraint_4[0] = cplex.linearNumExpr();
        for(int j=0;j<nComp;j++){
            for (int i=0;i<elementsList.size();i++){
                double specific_weight =elementsList.get(i).getSpecificWeight();
                constraint_4[0].addTerm(specific_weight,yVar[i][j]);
            }
        }
            cplex.addLe(constraint_4[0],veh.getVA_Max_Net());

            System.out.println("CONSTRAINT 8");
            IloLinearNumExpr[] constraint_8 = new IloLinearNumExpr[nElements];
             for (int i=0;i<elementsList.size();i++){
                 double element_volume = elementsList.get(i).getQuantity();
                 constraint_8[i] = cplex.linearNumExpr();
                 for (int j=0;j<nComp;j++){
                     constraint_8[i].addTerm(1, yVar[i][j]);
                 }
                 constraint_8[i].addTerm(-1, dVar[i]);

                cplex.addLe(constraint_8[i],element_volume);
             }

                System.out.println("CONSTRAINT 9");
                IloLinearNumExpr[] constraint_9 = new IloLinearNumExpr[nElements];
                 for (int i=0;i<elementsList.size();i++){
                     double element_volume = elementsList.get(i).getQuantity(); //πρ
                     constraint_9[i] = cplex.linearNumExpr();
                     for (int j=0;j<nComp;j++){
                         constraint_9[i].addTerm(1, yVar[i][j]);
                      }
                      constraint_9[i].addTerm(1, dVar[i]);

                     cplex.addGe(constraint_9[i],element_volume);
                  }

                  System.out.println("CONSTRAINT 10a");
                     IloLinearNumExpr[] constraint_10a = new IloLinearNumExpr[nElements];
                     for (int i=0;i<nElements;i++){
                         constraint_10a[i] = cplex.linearNumExpr();
                         for(int j=0;j<nComp;j++){
                             constraint_10a[i].addTerm(1,yVar[i][j]);
                         }
                         double upper_limit = elementsList.get(i).getQuantity()*(1+a_percent_upper);
                         cplex.addLe(constraint_10a[i],upper_limit);
                     }

                     System.out.println("CONSTRAINT 10b");
                     IloLinearNumExpr[] constraint_10b = new IloLinearNumExpr[nElements];
                     for (int i=0;i<nElements;i++){
                         constraint_10b[i] = cplex.linearNumExpr();
                         for(int j=0;j<nComp;j++){
                             constraint_10b[i].addTerm(1,yVar[i][j]);
                         }
                         double lower_limit = elementsList.get(i).getQuantity()*(1-a_percent_lower);
                         cplex.addGe(constraint_10b[i],lower_limit);
                     }

                     System.out.println("CONSTRAINT 11");
                     IloLinearNumExpr[] constraint_11 = new IloLinearNumExpr[nComp];
                     for(int i=0; i<nComp; i++){
                         constraint_11[i]=cplex.linearNumExpr();
                         constraint_11[i].addTerm(1.0, wVar[0][i]);
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
208                         for (int r=0;r<nElements;r++){
209                             double specific_weight = elementsList.get(r).getSpecificWeight();
210                             constraint_11[i].addTerm((-1*specific_weight), yVar[r][i]);
211                         }
212                         cplex.addEq(constraint_11[i],0);
213                     }
214
215                     System.out.println("CONSTRAINT 12");
216                     IloLinearNumExpr[] constraint_12 = new IloLinearNumExpr[nComp];
217                     for(int i=0; i<nComp; i++){
218                         constraint_12[i]=cplex.linearNumExpr();
219                         constraint_12[i].addTerm(1.0, wVar[nRouteLinks-1][i]);//τελευταίος πελάτης
220                         cplex.addEq(constraint_12[i],0);
221                     }
222
223                     System.out.println("CONSTRAINT 13");
224                     IloLinearNumExpr[][] constraint_13 = new IloLinearNumExpr[nComp][nRouteLinks];
225                     for (int iFromNode=1;iFromNode<nRouteLinks-1;iFromNode++){
226                         for(int c=0; c<nComp; c++){
227                             constraint_13[c][iFromNode]=cplex.linearNumExpr();
228                             constraint_13[c][iFromNode].addTerm(1.0, wVar[iFromNode-1][c]);
229                             constraint_13[c][iFromNode].addTerm(-1.0, wVar[iFromNode][c]);
230
231                             for (int icount=0;icount<elementsRefByOrder.get(iFromNode-1).size();icount++){
232                                 int r=elementsRefByOrder.get(iFromNode-1).get(icount);
233                                 double specific_weight =elementsList.get(r).getSpecificWeight();
234                                 constraint_13[c][iFromNode].addTerm((-1.0*specific_weight), yVar[r][c]);
235                             }
236                             cplex.addEq(constraint_13[c][iFromNode],0);
237                         }
238                     }
239
240                     System.out.println("CONSTRAINT 14");
241                     IloLinearNumExpr[] constraint_14 = new IloLinearNumExpr[nComp];
242                     for(int i=0; i<nComp; i++){

243                         constraint_14[i]=cplex.linearNumExpr();
244                         Compartment vcomp = pprob.get_all_compartments().get(i);
245                         constraint_14[i].addTerm(-0.8*vcomp.getVehicleCompartment_Capacity(), phiVar[i][0]);
246                         for (int r=0;r<nElements;r++){
247                             constraint_14[i].addTerm(1, yVar[r][i]);
248                         }
249                         cplex.addGe(constraint_14[i],0);
250                     }
251
252                     System.out.println("CONSTRAINT 15");
253                     IloLinearNumExpr[] constraint_15 = new IloLinearNumExpr[nComp];
254                     for(int i=0; i<nComp; i++){
255                         constraint_15[i]=cplex.linearNumExpr();
256                         Compartment vcomp = pprob.get_all_compartments().get(i);
257                         constraint_15[i].addTerm(vcomp.getVehicleCompartment_Capacity(), phiVar[i][1]);
258                         for (int r=0;r<nElements;r++){
259                             constraint_15[i].addTerm(1, yVar[r][i]);
260                         }
261                         cplex.addLe(constraint_15[i],1.2*vcomp.getVehicleCompartment_Capacity());
262                     }
263
264                     System.out.println("CONSTRAINT 16");
265                     IloLinearNumExpr[] constraint_16 = new IloLinearNumExpr[nComp];
266                     for(int i=0; i<nComp; i++){
267                         constraint_16[i]=cplex.linearNumExpr();
268                         constraint_16[i].addTerm(1,phiVar[i][0]);
269                         constraint_16[i].addTerm(1,phiVar[i][1]);
270                         cplex.addEq(constraint_16[i],1);
271
272                         }
273
274                     System.out.println("CONSTRAINT 5");
275                     IloLinearNumExpr[][] constraint_5 = new IloLinearNumExpr[nElements][nComp];
276                     for(int c=0; c<nComp; c++){
277
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
278                         for(int k=0; k<nElements; k++){
279                             constraint_5[k][c]=cplex.linearNumExpr();
280
281                             for(int i=0;i<fuel_no;i++){
282
283                             if(fueltypes.get(i).equals(elementsList.get(k).getDeliveryItem_SN())){
284
285                             constraint_5[k][c].addTerm(-1,yVar[k][c]);
286                             constraint_5[k][c].addTerm(pprob.get_all_compartments().get(c).getVehicleCompartment_Capacity(),xVar[k][c]);
287                             }}
288                              cplex.addGe(constraint_5[k][c],0);
289
290                         }
291
292                         }
293
294                     System.out.println("CONSTRAINT 6");
295                     IloLinearNumExpr[][] constraint_6 = new IloLinearNumExpr[nElements][nComp];
296                     for(int c=0; c<nComp; c++){
297
298                             for(int k=0; k<nElements; k++){
299                                 constraint_6[k][c]=cplex.linearNumExpr();
300
301                                 for(int i=0;i<fuel_no;i++){
302
303                                 if(fueltypes.get(i).equals(elementsList.get(k).getDeliveryItem_SN())){
304
305                                 constraint_6[k][c].addTerm(1,yVar[k][c]);
306                                 constraint_6[k][c].addTerm(-1,xVar[k][c]);
307                                 }}
308                                  cplex.addGe(constraint_6[k][c],0);
309                             }
310
311                         }
312
313                             IloLinearNumExpr[] constraint_25 = new IloLinearNumExpr[nRouteLinks];
314                                 for (int iFromNode=0;iFromNode<nRouteLinks-1;iFromNode++){
315
316                                         constraint_25[iFromNode]=cplex.linearNumExpr();
317
318                                         for(int c=0; c<nComp; c++){
319                                             constraint_25[iFromNode].addTerm(pprob.get_all_compartments().get(c).getDist_axle1()/L123,
320                                                     wVar[iFromNode][c]);
321                                         }
322
323                                         cplex.addLe(constraint_25[iFromNode],veh.getMax_2());
324                                     }
325
326                             IloLinearNumExpr[] constraint_26 = new IloLinearNumExpr[nRouteLinks];
327                                 for (int iFromNode=0;iFromNode<nRouteLinks-1;iFromNode++){
328
329                                         constraint_26[iFromNode]=cplex.linearNumExpr();
330
331                                         for(int c=0; c<nComp; c++){
332                                             constraint_26[iFromNode].addTerm((7.245-(pprob.get_all_compartments().get(c).getStartPos()+
333                                                     pprob.get_all_compartments().get(c).getEndPos())/2)/L123,wVar[iFromNode][c]);
334                                         }
335
336                                         cplex.addLe(constraint_26[iFromNode],veh.getMax_1());
337                                     }
338
339                             IloLinearNumExpr[] constraint_27 = new IloLinearNumExpr[nRouteLinks];
340                                 for (int iFromNode=0;iFromNode<nRouteLinks-1;iFromNode++){
341
342                                         constraint_27[iFromNode]=cplex.linearNumExpr();
343
344                                         for(int c=0; c<nComp; c++){
345                                             constraint_27[iFromNode].addTerm(pprob.get_all_compartments().get(c).getDist_axle1()/L123,
346                                                     wVar[iFromNode][c]);
347                                         }
348
349                                         cplex.addGe(constraint_27[iFromNode],-B4);
350                                     }
351
352                             IloLinearNumExpr[] constraint_28 = new IloLinearNumExpr[nRouteLinks];
353                                 for (int iFromNode=0;iFromNode<nRouteLinks-1;iFromNode++){
354
355                                         constraint_28[iFromNode]=cplex.linearNumExpr();
356
357                                         for(int c=0; c<nComp; c++){
358                                             constraint_28[iFromNode].addTerm((7.245-(pprob.get_all_compartments().get(c).getStartPos()+
359                                                     pprob.get_all_compartments().get(c).getEndPos())/2)/L123,wVar[iFromNode][c]);
360                                         }
361
362                                         cplex.addGe(constraint_28[iFromNode],-B1);
363                                     }
364
365                             IloLinearNumExpr[] constraint_29 = new IloLinearNumExpr[nRouteLinks];
366                                 for (int iFromNode=0;iFromNode<nRouteLinks-1;iFromNode++){
367
368                                         constraint_29[iFromNode]=cplex.linearNumExpr();
369
370                                         for(int c=0; c<nComp; c++){
371                                             constraint_29[iFromNode].addTerm(pprob.get_all_compartments().get(c).getDist_axle1()/L123,
372                                                     wVar[iFromNode][c]);
373                                             constraint_29[iFromNode].addTerm(-0.25,wVar[iFromNode][c]);
374                                         }
375
376                                         cplex.addGe(constraint_29[iFromNode],0.25*veh.getVA_Tare()-B4);
377                                     }
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
                                    IloLinearNumExpr[] constraint_30 = new IloLinearNumExpr[nRouteLinks];
                                    for (int iFromNode=0;iFromNode<nRouteLinks-1;iFromNode++){

                                            constraint_30[iFromNode]=cplex.linearNumExpr();

                                            for(int c=0; c<nComp; c++){
                                                constraint_30[iFromNode].addTerm((pprob.get_all_compartments().get(c).getDist_axle1()+
                                                        veh.getAxlePos().get(0)),wVar[iFromNode][c]);
                                                constraint_30[iFromNode].addTerm((-xt-p),wVar[iFromNode][c]);
                                            }

                                            cplex.addLe(constraint_30[iFromNode],0);
                                            }

                                    IloLinearNumExpr[] constraint_31 = new IloLinearNumExpr[nRouteLinks];
                                    for (int iFromNode=0;iFromNode<nRouteLinks-1;iFromNode++){

                                            constraint_31[iFromNode]=cplex.linearNumExpr();

                                            for(int c=0; c<nComp; c++){
                                                constraint_31[iFromNode].addTerm(-((pprob.get_all_compartments().get(c).getDist_axle1()+
                                                        veh.getAxlePos().get(0))),wVar[iFromNode][c]);
                                                constraint_31[iFromNode].addTerm((xt-p),wVar[iFromNode][c]);
                                            }

                                            cplex.addLe(constraint_31[iFromNode],0);
                                            }

// OBJECTIVE FUNCTION


                                IloLinearNumExpr F1 = cplex.linearNumExpr();
                                    for (int i=0;i<elementsList.size();i++){

                                        F1.addTerm(1000,dVar[i]);
                                     }

                                     for (int i=0;i<elementsList.size();i++){
                                         for (int j=0;j<nComp;j++){
                                            Compartment vcomp = pprob.get_all_compartments().get(j);
                                            F1.addTerm(0.01*vcomp.getVehicleCompartment_Capacity(), xVar[i][j]);
                                            F1.addTerm(-0.01, yVar[i][j]);
                                         }
                                     }

                                     cplex.addMinimize(F1);


                double totalDemand=0;
                for (int i=0;i<nElements;i++){
                    OrderItem item = elementsList.get(i);
                    totalDemand+=item.getQuantity();
                }
                System.out.println("Total Demand:"+ totalDemand);

            if (cplex.solve()){

                System.out.println(" MODEL SOLVED. ABOUT TO SET UP XPAR,YPAR ");
                int[][] xpar = new int[nElements][nComp];
                double[][] ypar = new double[nElements][nComp];
                double[] totalVolpar = new double[nComp];
                double[] UrValues = new double[nElements];

                for (int i=0;i<nElements;i++){
                    for (int j=0;j<nComp;j++){
                        xpar[i][j] = (int) Math.round(cplex.getValue(xVar[i][j]));
                        ypar[i][j] = (int) Math.round(cplex.getValue(yVar[i][j]));
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
446                         {
447                              System.out.println("y["+i+"]["+j+"]="+ypar[i][j]+ "  x["+i+"]["+j+"]="+xpar[i][j]);
448                         }
449                         totalVolpar[j]=totalVolpar[j]+ypar[i][j];
450 
451                     }
452                     UrValues[i] = (double) Math.round(cplex.getValue(dVar[i]));
453                     System.out.println("orderId: " + (i+1) + ", Ur:" + UrValues[i]);
454                 }
455                 double totalVolparSum = 0;
456                 for(int counterLast = 0; counterLast < totalVolpar.length; counterLast ++) {
457                     totalVolparSum += totalVolpar[counterLast];
458                 }
459                 System.out.println("Total order assigned to the vehicle: " + totalVolparSum);
460 
461                 System.out.println("Objective Function Value: " + cplex.getObjValue());
462             }
463             else
464             {
465 
466             }
467 
468             cplex.end();
469             }
470             catch(IloException e)
471             {
472                 System.err.println("Concert exception '" + e + "' caught");
473                 e.getStackTrace();
474             }
475         }
476 
477     }
478 
479 }
```


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ