UNIVERSITY OF THE AEGEAN

SCHOOL OF BUSINESS

DEPARTMENT OF FINANCIAL AND MANAGEMENT ENGINEERING

"Load Exchange Strategies for a Vehicle Routing Problem with Dynamic Pickups"

Michael Vitellas

Supervisor: I. Minis

Committee members: A. Platis

D. Drivaliaris

Chios, 2011

To my family and my friends

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude to Professor Ioannis Minis for the opportunity gave me to write the present diploma thesis. I thank him for his time, as well for the supervision and guidance both during the development of ideas and the writing of the thesis.

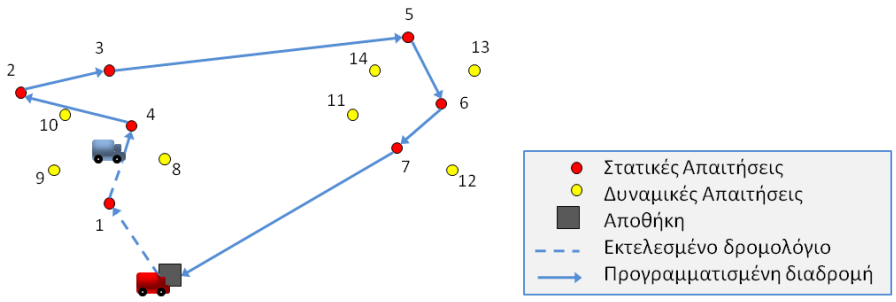
I would also like to thank the doctoral candidate George Ninikas for his invaluable help and patience throughout the course of writing. This thesis would not have been possible without him.

Another special mention goes out to all members of the Design, Operations & Production Systems Lab (DeOPSys) of the Financial and Management Engineering (FME) Department of the University of the Aegean, and mainly to George Dikas, Theodore Athanasopoulos, Kostas Mamasis and Lemonia Amygdalou. I thank them for their understanding and invaluable assistance during the conduct of my thesis.

I must also acknowledge my friends for their love and care. Finally, I am grateful to my family for their continued support and encouragement.

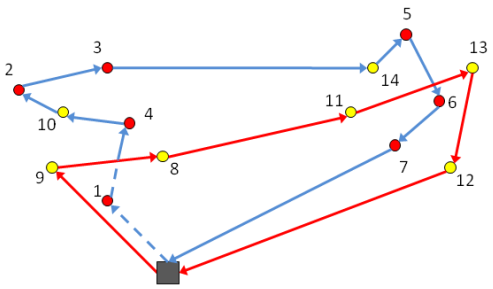
SUMMARY (IN GREEK)

Στην παρούσα διπλωματική εργασία διερευνάται το Πρόβλημα Δρομολόγησης Οχημάτων με Δυναμικές Παραλαβές (ΠΔΟΔΠ), το οποίο έχει παρακινηθεί από πρακτικές εφαρμογές courier (Ninikas *et al.*, 2011) και βασίζεται σε μελέτη που διεξάγεται από τους Ninikas and Minis (2011). Σε ένα τέτοιο πρόβλημα, ένα σύνολο οχημάτων μεταφοράς, τα οποία είναι εγκατεστημένα σε ένα τοπικό κόμβο διανομής (αποθήκη), προορίζονται να εξυπηρετήσουν αιτήματα παράδοσης (διανομής) τα οποία είναι γνωστά πρίν από την έναρξη των εργασιών (*στατικές απαιτήσεις*). Καθώς το πρόγραμμα εργασίας εξελίσσεται, ωστόσο, νεοαφιχθέντα αιτήματα πελατών λαμβάνονται μέσω ενός τηλεφωνικού κέντρου, ζητώντας επί τόπου παραλαβή φορτίου εντός της τρέχουσας περιόδου λειτουργίας (*δυναμικές απαιτήσεις*). Ένα χαρακτηριστικό παράδειγμα του ΠΔΟΔΠ παρουσιάζεται στο Σχήμα Π.1.



Σχήμα Π.1: Το Πρόβλημα Δρομολόγησης Οχημάτων με Δυναμικές Παραλαβές (ΠΔΟΔΠ)

Σε μια δεδομένη χρονική στιγμή (φάση ανασχεδιασμού), που επιλέγεται από τον υπεύθυνο δρομολόγησης, οι δυναμικές απαιτήσεις πρέπει να ενσωματωθούν στο εν μέρει εκτελεσμένο σχέδιο δρομολόγησης. Τα φορτία αυτών των αιτημάτων συλλέγονται από τα οχήματα και επιστρέφονται στην αποθήκη για περαιτέρω επεξεργασία. Στο Σχήμα Π.2 παρουσιάζεται η λύση που λαμβάνεται στο πρόβλημα του Σχήματος Π.1 κατά τη χρονική στιγμή του ανασχεδιασμού.



Σχήμα Π.2: Επίλυση του ΠΔΟΔΠ κατά την φάση ανασχεδιασμού

Οι μεταφορικές εταιρείες που λειτουργούν μέσα σε ένα τέτοιο πλαίσιο, αντιμετωπίζουν πολλές προκλήσεις κατά την προσπάθειά τους να ενσωματώσουν τις δυναμικές απαιτήσεις στα προγραμματισμένα δρομολόγια. Οι προκλήσεις αυτές έγκεινται κατά κύριο λόγο στις εξής:

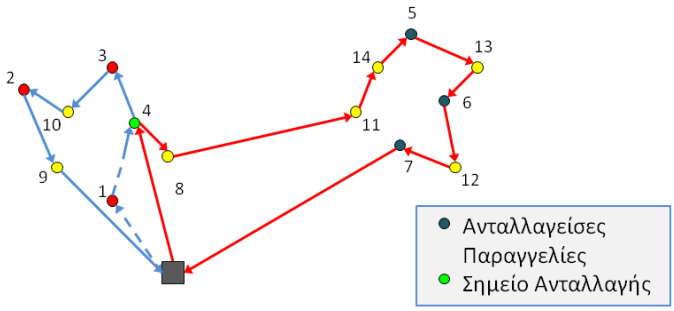
- Ο μεγάλος αριθμός δυναμικών αιτημάτων που προκύπτουν εντός της ημέρας περιπλέκει σημαντικά τη λήψη αποφάσεων, καθιστώντας απαραίτητη την εφαρμογή «καλών» στρατηγικών για τον ανασχεδιασμό του στόλου.
- Σε ένα υψηλής πυκνότητας αστικό περιβάλλον, υπάρχουν επικαλυπτόμενες περιοχές εξυπηρέτησης για τα οχήματα, με αποτέλεσμα να περιπλέκονται οι αποφάσεις όσον αφορά την ανάθεση του καταλληλότερου οχήματος για την εξυπηρέτηση μιας νεοαφιχθείσας απαίτησης.
- Η αρχική ανάθεση συγκεκριμένων αιτημάτων παράδοσης σε ένα όχημα, περιορίζει περαιτέρω τα περιθώρια βελτιστοποίησης, δεδομένου ότι αυτές οι παραγγελίες πρέπει να εξυπηρετηθούν μόνο από αυτό το όχημα (αφού μεταφέρει το φορτίο που πρέπει να παραδοθεί στους πελάτες), αναγκάζοντας έτσι τα οχήματα να ακολουθούν ένα σχέδιο δρομολόγησης που αλλάζει συνεχώς σε πραγματικό χρόνο.

Η παρούσα διπλωματική εργασία εξετάζει μία προσπάθεια εξομάλυνσης της τελευταίας πρόκλησης επιτρέποντας στα οχήματα να συναντηθούν σε πραγματικό χρόνο και να ανταλλάξουν αιτήματα διανομής (στατικές απαιτήσεις), εάν αυτό ευνοεί την αντικειμενική συνάρτηση. Επομένως, μια στατική απαίτηση μπορεί να εξυπηρετηθεί από δύο οχήματα· ένα όχημα που αρχικά διατηρεί το φορτίο του αιτήματος ενός πελάτη, ο οποίος εν τέλει εξυπηρετείται από ένα άλλο όχημα. Αναφερόμαστε σε αυτή τη καινοτόμο στρατηγική ως *Στρατηγική Ανταλλαγής Φορτίων (ΣΑΦ)*.

Η κύρια ιδέα της στρατηγικής αυτής έγκειται στον περιορισμό των οχημάτων στη περιοχή κάλυψής τους, προκειμένου να εξυπηρετήσουν δυναμικές απαιτήσεις που ενδέχεται να εμφανιστούν στη περιοχή αυτή. Ένα σημαντικό πλεονέκτημα της ΣΑΦ, είναι η δυνατότητα ανάθεσης κάθε παραγγελίας (είτε στατικής είτε δυναμικής) σε οποιοδήποτε όχημα, χωρίς να περιορίζονται οι επιλογές ανασχεδιασμού. Με αυτό τον τρόπο μπορεί να επιτευχθεί καλύτερη ανακατανομή του φόρτου εργασίας και καλύτερος ανασχεδιασμός του πλάνου δρομολόγησης. Ωστόσο, ένα μειονέκτημα

αυτής της στρατηγικής είναι οι πιθανές καθυστερήσεις που μπορεί να προκληθούν εξαιτίας της λειτουργίας της μεταφόρτωσης. Αυτές μπορεί να οφείλονται: α) στο γεγονός ότι, στις περισσότερες περιπτώσεις, τα οχήματα δεν φτάνουν ταυτόχρονα στο σημείο ανταλλαγής, με αποτέλεσμα να απαιτείται αναμονή ενός οχήματος μέχρι να φθάσει το άλλο προκειμένου να επιτευχθεί η ανταλλαγή των φορτίων, και β) στο χρονικό διάστημα που απαιτείται από τα οχήματα για την φορτοεκφόρτωση των παραγγελιών.

Στο Σχήμα Π.3 απεικονίζεται η λύση στο πρόβλημα του Σχήματος Π.1, κατά τη χρονική στιγμή του ανασχεδιασμού, εφαρμόζοντας την Στρατηγική Ανταλλαγής Φορτίου (ΣΑΦ). Παρατηρείται ότι η λύση που προκύπτει από την ΣΑΦ, βελτιώνει κατά πολύ τη λύση συγκριτικά με εκείνη που παίρνουμε στο Σχήμα Π.2 όπου δεν επιτρέπεται ανταλλαγή φορτίου μεταξύ των οχημάτων. Πιο συγκεκριμένα, στο παράδειγμα αυτό, η μείωση του κόστους διαδρομής είναι της τάξεως του 29.6%, το οποίο υποδεικνύει την ανάγκη περαιτέρω μελέτης της στρατηγικής αυτής.



Σχήμα Π.3: Επίλυση του ΠΔΟΔΠ με την Στρατηγική Ανταλλαγής Φορτίου (ΣΑΦ)

Η υφιστάμενη βιβλιογραφία στον τομέα της ανταλλαγής (μεταφόρτωσης) φορτίου μεταξύ των οχημάτων είναι περιορισμένη. Το μεγαλύτερο μέρος της έρευνας που έχει διεξαχθεί μέχρι σήμερα στον τομέα αυτό, επικεντρώνεται σε προβλήματα όπου τα αιτήματα των πελατών αφορούν παραλαβή ενός φορτίου από μια τοποθεσία και επίδοση σε μια άλλη μέσα στην ίδια ημέρα (Πρόβλημα Παραλαβής και Επίδοσης, ΠΠΕ). Η προσέγγισή μας διαφοροποιείται από την υφιστάμενη έρευνα ως προς τα ακόλουθα σημεία:

- Εφαρμόζουμε την ΣΑΦ σε ένα δυναμικό περιβάλλον όπου τα αιτήματα των πελατών καταφθάνουν δυναμικά με την πάροδο του χρόνου.
- Εφαρμόζουμε την στρατηγική σε many-to-one περιπτώσεις, στις οποίες κάθε απαίτηση συνδέεται με μια μόνο τοποθεσία (παραλαβή ή επίδοση ενός

αντικειμένου), όπου η λειτουργεία της μεταφόρτωσης φορτίου δεν είναι ευδιάκριτο ότι μπορεί να επιφέρει εξοικονόμηση κόστους.

- Εκτός από την περίπτωση της σταθερής τοποθεσίας για την ανταλλαγή των φορτίων, εξετάζουμε και εντοπίζουμε περιπτώσεις όπου δυναμικά σημεία μπορούν επίσης να ληφθούν υπόψην, επιφέροντας σε αρκετές περιπτώσεις ακόμη καλύτερα αποτελέσματα.

Η ανταλλαγή φορτίου είναι μια αρκετά περίπλοκη και δύσκολη στρατηγική όσον αφορά την υλοποίησή της, εξαιτίας του μεγάλου αριθμού παραμέτρων που πρέπει να καθοριστούν. Οι παράμετροι αυτές συνοψίζονται στον Πίνακα Π.1.

Πίνακας Π.1: Παράμετροι και εναλλακτικές πολιτικές για την ΣΑΦ

Παράμετρος	**Περιγραφή**	**Εναλλακτικές**
Χρόνος Υλοποίησης της ΣΑΦ	Η χρονική στιγμή ή οι συνθήκες κάτω από τις οποίες κάποιος θα μπορούσε να εφαρμόσει τη στρατηγική	• σε μια φάση ανασχεδιασμού • όταν ένα όχημα δεν έχει να εξυπηρετήσει άλλες απαιτήσεις • στον χρόνο αδράνειας ενός οχήματος • όταν ένα όχημα δεν είναι ικανό να εξυπηρετήσει τις υπόλοιπες απαιτήσεις που του είχαν ανατεθεί
Συνδυασμοί Συνάντησης	Ο αριθμός των καθ'οδών οχημάτων με τα οποία ένα όχημα μπορεί να ανταλλάξει στατικές απαιτήσεις	• ένα-προς-ένα • ένα-προς-πολλά
Τοποθεσίες Ανταλλαγής	Τα σημεία όπου τα οχήματα επιτρέπεται να ανταλλάξουν (μεταφορτώσουν) τα φορτία	• σημεία πελατών που δεν έχουν εξυπηρετηθεί ακόμη • καθ'οδόν • αποθήκη • προκαθορισμένες (στατικές) τοποθεσίες («θυρίδες», χώροι στάθμευσης κλπ.)

Στην παρούσα εργασία, η ΣΑΦ εφαρμόζεται στο ΠΔΟΔΠ, ενσωματώνοντας αρκετές από τις προαναφερόμενες παραμέτρους. Συγκεκριμένα, σε δεδομένη χρονική στιγμή ανασχεδιασμού, θεωρούμε έναν στόλο από δύο οχήματα όπου το ένα αντιστοιχεί πάντα σε ένα όχημα καθ'οδόν (εξυπηρετώντας το πλάνο που του έχει ανατεθεί), και το άλλο είτε καθ'οδόν είτε στην αποθήκη (σαν εφεδρικό όχημα) αναμένοντας να ενεργοποιηθεί κατά την εμφάνιση δυναμικών απαιτήσεων. Κατά τη φάση ανασχεδιασμού, η ΣΑΦ εφαρμόζεται για να εξετάσει κατά πόσον είναι κερδοφόρο για τα οχήματα να συναντηθούν και να ανταλλάξουν κάποιες παραγγελίες ή όχι· εάν όχι, ένας τυπικός αλγόριθμος ανασχεδιασμού εφαρμόζεται. Αυτός είναι και ο λόγος που αναφερόμαστε σε αυτή την καινοτομική μέθοδο ως «στρατηγική». Περιορισμοί χωρητικότητας για τα οχήματα δεν συμπεριλαμβάνονται στο πρόβλημα, επειδή το φορτίο της κάθε απαίτησης είναι σχετικά μικρό στις εξεταζόμενες εφαρμογές. Η προτεινόμενη προσέγγιση εξετάζει διάφορες επιλογές, όπως δυναμικά και σταθερά σημεία ανταλλαγής, ακριβής (exact) ή ευρετικές μεθόδους για την επίλυση των τμημάτων του συνολικού αλγορίθμου, κλπ.

Βασική Μεθοδολογία Επίλυσης

Η προτεινόμενη μεθοδολογία επίλυσης για την ΣΑΦ αποτελεί ένα αλγόριθμο τριών φάσεων, και υλοποιείται σε συνδυασμό με μια τυπική μέθοδο επίλυσης του ΠΔΟΔΠ, η οποία έχει δημιουργηθεί από τους Ninikas and Minis (2011), και την οποία ονομάζουμε *Μέθοδο Εισαγωγής (ΜΕ)*. Η ΜΕ είναι μια προσέγγιση ακριβούς επίλυσης (exact) που χρησιμοποιεί τη μέθοδο Δυναμικής Δημιουργίας Μεταβλητών (ΔΔΜ) ή Column Generation (CG), και δεν επιτρέπει την εκ νέου ανάθεση των στατικών απαιτήσεων μεταξύ των οχημάτων (δηλαδή κάθε όχημα πρέπει να εξυπηρετήσει τις απαιτήσεις διανομής που του είχαν ανατεθεί αρχικά). Σε μια φάση ανασχεδιασμού και αφού το πρόβλημα επιλυθεί με την ΜΕ, εφαρμόζεται η ΣΑΦ ώστε να ελέγξει αν η ανταλλαγή φορτίου μεταξύ των οχημάτων μπορεί να επιφέρει μια βελτιωμένη λύση. Οι φάσεις του αλγορίθμου της ΣΑΦ έχουν ως εξής:

Φάση Ι. Δρομολόγηση: Σε αυτή τη φάση, επιλύεται ένα πρόβλημα που μοιάζει με το τυπικό Πρόβλημα Δρομολόγησης Οχημάτων (ΠΔΟ) ή Vehicle Routing Problem (VRP). Στο πρόβλημα αυτό, θεωρούμε ένα δίκτυο το οποίο περιλαμβάνει όλες τις απαιτήσεις των πελατών που δεν έχουν εξυπηρετηθεί μέχρι τη χρονική στιγμή του ανασχεδιασμού. Όλες οι

απαιτήσεις θεωρούνται ομοιογενής, δηλαδή δεν έχουμε διάκριση των αιτημάτων σε απαιτήσεις παραλαβής και απαιτήσεις επίδοσης. Αυτή η χαλάρωση στο πρόβλημα επιτρέπει σε κάθε απαίτηση να μπορεί να εξυπηρετηθεί από οποιοδήποτε όχημα, ανεξάρτητα με την αρχική της ανάθεση. Για την επίλυση αυτής της φάσης, μπορεί να χρησιμοποιηθεί τόσο μια ακριβής, όσο και μια ευρετική μέθοδος, ανάλογα με το μέγεθος του προβλήματος.

Φάση ΙΙ. Συνάντηση: Κατά τη διάρκεια αυτής της φάσης, εξετάζεται εάν υπάρχουν παραγγελίες διανομής που αρχικά είχαν ανατεθεί σε ένα όχημα, και τώρα πρέπει να εξυπηρετηθούν από το άλλο (με βάση τη λύση του ΠΔΟ). Εάν όχι, η λύση του ΠΔΟ παραμένει αμετάβλητη και αποτελεί την τελική λύση. Διαφορετικά, χρησιμοποιείται μια διαδικασία η οποία αναζητά το καλύτερο εφικτό σημείο συνάντησης για τα οχήματα. Για την επίλυση αυτής της διαδικασίας, προτείνονται διάφοροι εναλλακτικοί αλγόριθμοι.

Φάση ΙΙΙ. Βελτιστοποίηση: Τέλος, στη λύση που λαμβάνεται από τις δύο προηγούμενες φάσεις εφαρμόζεται μια διαδικασία βελτιστοποίησης, η οποία εξετάζει εάν η εναλλαγή μιας απαίτησης μεταξύ των δύο δρομολογίων μπορεί να οδηγήσει σε βελτίωση της συνολικής λύσης.

Για την εύρεση του καλύτερου δυνατού σημείου ανταλλαγής κατά τη Φάση ΙΙ της μεθόδου, αναπτύχθηκαν τέσσερις (4) αλγόριθμοι. Από αυτούς, οι τρεις πρώτοι θεωρούν δυναμικά σημεία για την ανταλλαγή των φορτίων και συγκεκριμένα όλες τις τοποθεσίες των πελατών που δεν έχουν εξυπηρετηθεί ακόμη, ενώ ο τελευταίος εξετάζει την περίπτωση ενός σταθερού σημείου μεταφόρτωσης, το οποίο υποθέτουμε ότι είναι τοποθετημένο στο κέντρο βάρους των συντεταγμένων των πελατών. Ακολουθεί μια μικρή περιγραφή του κάθε αλγορίθμου:

1. ***Απλή Τοπική Αναζήτηση (ΑΤΑ)*:** Ο αλγόριθμος αυτός χρησιμοποιεί μια μέθοδο εισαγωγής, η οποία, με βάση τη λύση που προκύπτει από το ΠΔΟ της πρώτης φάσης, δοκιμάζει κάθε τοποθεσία πελάτη σαν πιθανό σημείο ανταλλαγής για τα οχήματα. Αυτό επιτυγχάνεται, εισάγοντας προσωρινά το υποψήφιο σημείο ανταλλαγής (τοποθεσία πελάτη) ενός δρομολογίου, μεταξύ δύο διαδοχικών πελατών του άλλου δρομολογίου. Λαμβάνοντας όλους τους εφικτούς συνδυασμούς δρομολογίων, επιλέγεται αυτός (εάν υπάρχει) με το

χαμηλότερο κόστος, ώστε να αποτελέσει τη λύση που θα προχωρήσει στην τρίτη φάση της μεθόδου.

2. ***Προηγμένη Τοπική Αναζήτηση (ΠΤΑ)*:** Η πρώτη λειτουργία αυτού του αλγορίθμου είναι ακριβώς όμοια με εκείνη στην ΑΤΑ. Για κάθε συνδυασμό δρομολογίων που προκύπτει από την ΑΤΑ, εφαρμόζεται μια διαδικασία βελτιστοποίησης στα δρομολόγια, η οποία αναζητά άλλες λύσεις (εναλλάσοντας μια απαίτηση μεταξύ των δύο δρομολογίων) που μπορούν να βελτιώσουν την τρέχουσα λύση. Από τους ανανεωμένους συνδυασμούς δρομολογίων, επιλέγεται εκείνος με το χαμηλότερο κόστος.

3. ***Σύνθετη Αναζήτηση (ΣΑ)*:** Ο αλγόριθμος αυτός χρησιμοποιεί επίσης την ΑΤΑ. Για κάθε συνδυασμό δρομολογίων που προκύπτει από την ΑΤΑ, επιλύεται ένα πρόβλημα όμοιο με το ΠΔΟ λαμβάνοντας υπόψη το δίκτυο όλων των πελατών που έχουν ανατεθεί μετά από την τοποθεσία ανταλλαγής. Ως εκ τούτου, τα τμήματα των διαδρομών μέχρι την λειτουργία της ανταλλαγής παραμένουν αμετάβλητα, και οι διαδρομές που προκύπτουν από το ΠΔΟ αποτελούν τα τμήματα δρομολογίων μετά από το σημείο ανταλλαγής. Πάλι, από τους ανανεωμένους συνδυασμούς, επιλέγεται εκείνος με το χαμηλότερο κόστος. Για την επίλυση του ΠΔΟ μπορεί να χρησιμοποιηθεί τόσο μια ακριβής, όσο και μια ευρετική μέθοδος, ανάλογα με το μέγεθος του προβλήματος.

4. ***Ενσωμάτωση Σταθερού Σημείου Μεταφόρτωσης (ΕΣΣΜ)*:** Στον παρών αλγόριθμο το σημείο ανταλλαγής θεωρείται ότι είναι γνωστό πριν από την έναρξη των εργασιών και αποτελεί ιδιοκτησία της εταιρίας. Σημαντικό πλεονέκτημα του σημείου αυτού είναι ότι επιτρέπει σε ένα όχημα να αποθέσει εκεί κάποιο φορτίο, το οποίο μπορεί να ληφθεί κάποια στιγμή αργότερα από το άλλο όχημα. Η βασική διεργασία του αλγορίθμου αφορά την εισαγωγή του σημείου μεταφόρτωσης μεταξύ δύο οποιονδήποτε διαδοχικών πελατών σε κάθε προγραμματισμένο δρομολογίο. Λαμβάνοντας όλους τους εφικτούς συνδυασμούς δρομολογίων, επιλέγεται εκείνος που ελαχιστοποιεί το συνολικό κόστος δρομολόγησης.

<u>Πειραματική Διερεύνηση</u>

Τα πειράματα δημιουργήθηκαν βάσει των κριτηρίων του Solomon (Solomon, 1987), τα οποία αποτελούνται από σύνολα R, C και R· όπου κάθε σύνολο υποδηλώνει μια διαφορετική γεωγραφική κατανομή πελατών (R: τυχαία, C: ομαδοποιημένη, RC: μικτή). Προκειμένου να παρέχουμε εκτενή πειραματική διερεύνηση και να προσδιορίσουμε τη συσχέτιση της απόδοσης της ΣΑΦ με τη γεωγραφική κατανομή των πελατών, δημιουργήσαμε σετ δοκιμών από όλες τις κατηγορίες. Για κάθε μια από αυτές τις κατηγορίες, θεωρήσαμε επίσης σύνολα δοκιμών που αποτελούνται από 15, 25 και 50 απαιτήσεις πελατών. Επιπλέον, για κάθε μια από τις παραπάνω περιπτώσεις εξετάσαμε περιπτώσεις με Χρονικά Παράθυρα (ΧΠ) και χωρίς, οδηγώντας σε ένα σύνολο 18 σετ πειραμάτων. Ο Πίνακας Π.2 συνοψίζει τα σετ πειραμάτων.

Πίνακας Π.2: Σετ Πειραμάτων

Σετ	Αριθμός απαιτήσεων/ πελατών	Γεωγραφική κατανομή πελατών	Χρονικά Παράθυρα (ΧΠ)	Αριθμός πειραμάτων που πραγματοποιήθηκαν
1	15	R	Ναι	8
2	15	C	Ναι	7
3	15	RC	Ναι	8
4	25	R	Ναι	8
5	25	C	Ναι	7
6	25	RC	Ναι	6
7	50	R	Ναι	6
8	50	C	Ναι	6
9	50	RC	Ναι	6
10	15	R	Όχι	6
11	15	C	Όχι	6
12	15	RC	Όχι	6
13	25	R	Όχι	7
14	25	C	Όχι	6
15	25	RC	Όχι	6
16	50	R	Όχι	6
17	50	C	Όχι	6
18	50	RC	Όχι	6

Οι ακόλουθες παραδοχές ισχύουν για όλα τα σύνολα πειραμάτων που δημιουργήθηκαν για την πειραματική διερεύνηση της ΣΑΦ:

- Θεωρούμε ένα μόνο κύκλο ανασχεδιασμού και η ΣΑΦ εφαρμόζεται μια φορά την χρονική στιγμή του ανασχεδιασμού.
- Η βέλτιστη λύση (που παράγεται πριν τη χρονική στιγμή 0) αντιστοιχεί στην ανάθεση όλων αιτημάτων παράδοσης σε ένα από τα δύο διαθέσιμα οχήματα,

ενώ το άλλο βρίσκεται στην αποθήκη (σαν εφεδρικό όχημα) για την εξυπηρέτηση δυναμικών απαιτήσεων.

- Η χρονική στιγμή ανασχεδιασμού αντιστοιχεί σε μια τυχαία στιγμή του χρόνου, και ένας αριθμός στατικών απαιτήσεων έχει εξυπηρετηθεί μέχρι εκείνη τη στιγμή.
- Για λόγους απλότητας, ο χρόνος για την εξυπηρέτηση μιας απαίτησης στην τοποθεσία του πελάτη θεωρείται ότι είναι μηδέν (0).

Για την πραγματοποίηση της πειραματικής διερεύνησης μέσω της ΣΑΦ χρησιμοποιήθηκαν τέσσερις εναλλακτικοί αλγόριθμοι. Οι αλγόριθμοι αυτοί βασίζονται στην μεθοδολογία επίλυσης και τις αλγοριθμικές διαδικασίες που περιγράφηκαν παραπάνω. Κύριος σκοπός της παρούσας εργασίας είναι η αξιολόγηση της αποτελεσματικότητάς τους όσον αφορά το κόστος και τον υπολογιστικό χρόνο που απαιτούν.

Ο Πίνακας Π.3 συνοψίζει τα κύρια χαρακτηριστικά των αλγορίθμων και τις μεθόδους που χρησιμοποιήθηκαν για τα σετ πειραμάτων. Εφαρμόσαμε διαφορετικές μεθόδους, έτσι ώστε να ταιριάζουν με την πολυπλοκότητα του κάθε προβλήματος (δηλαδή τον αριθμό των πελατών). Συγκεκριμένα, για την επίλυση της Φάσης Ι, χρησιμοποιήθηκε μια Branch-and-Price (B&P) μέθοδος που παρέχει ακριβής λύσεις για τα σετ πειραμάτων με 15 απαιτήσεις πελατών, ενώ για τα μεγαλύτερα προβλήματα (25 και 50 πελάτες), χρησιμοποιήθηκε ο αλγόριθμος των Clarke & Wright (C&W), ο οποίος αποτελεί ευρετική μέθοδο, σε συνδυασμό με μια διαδικασία βελτιστοποίησης (Route-Interchange).

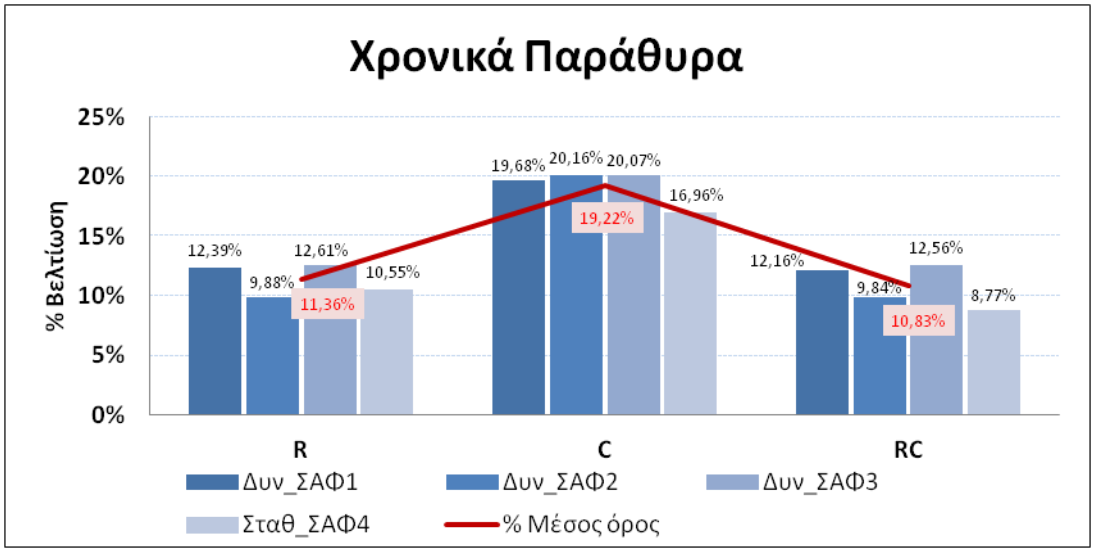
Πίνακας Π.3: Βασικά χαρακτηριστικά των αλγορίθμων επίλυσης

Αλγόριθμος	Όνομα	Σημείο Ανταλλαγής	15 πελάτες		25, 50 πελάτες	
			Φάση Ι *(Δρομολόγηση)*	**Φάση ΙΙ** *(Συνάντηση)*	**Φάση Ι** *(Δρομολόγηση)*	**Φάση ΙΙ** *(Συνάντηση)*
1	Δυν_ΣΑΦ1	Δυναμικό	B&P	ΑΤΑ	C&W + Route Interchange	ΑΤΑ
2	Δυν_ΣΑΦ2	Δυναμικό	B&P	ΣΑ (B&P για τα ΠΔΟ)	C&W + Route Interchange	ΣΑ (C&W για τα ΠΔΟ)
3	Δυν_ΣΑΦ3	Δυναμικό	B&P	ΠΤΑ	C&W + Route Interchange	ΠΤΑ
4	Σταθ_ΣΑΦ4	Σταθερό	B&P	ΕΣΣΜ	C&W + Route Interchange	ΕΣΣΜ

Παρακάτω παρουσιάζονται συνοπτικά τα αποτελέσματα που αποκτήθηκαν για όλα τα σετ πειραμάτων, αναλύοντάς τα με βάση: α) τα Χρονικά Παράθυρα (ΧΠ), β) την γεωγραφική κατανομή των πελατών και γ) την υπολογιστική προσπάθεια των αλγορίθμων. Κάθε πείραμα επιλύθηκε αρχικά με την Μέθοδο Εισαγωγής (ΜΕ) και στη συνέχεια με καθέναν από τους τέσσερις αλγόριθμους.

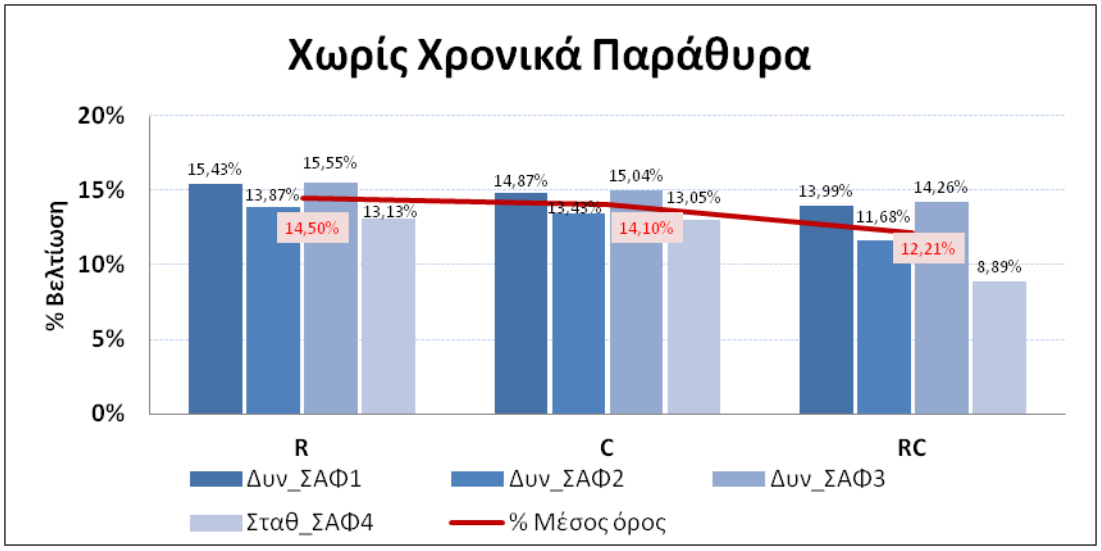
Στα επόμενα τρία Σχήματα απεικονίζεται το ποσοστό βελτίωσης καθενός από τους τέσσερις αλγόριθμους συγκριτικά με τα αποτελέσματα που προέκυψαν από την ΜΕ (Y-άξονας). Ο X-άξονας αναφέρεται στην γεωγραφική κατανομή των πελατών, δηλαδή R, C και RC. Κάθε γράφημα στα σχήματα αντιπροσωπεύει τα αποτελέσματα που αποκτήθηκαν από καθέναν από τους τέσσερις προτεινόμενους αλγόριθμους. Μια πρόσθετη κόκκινη γραμμή χρησιμοποιείται επίσης σε κάθε σχήμα για να δείξει τη μέση βελτίωση όλων των αλγορίθμων.

Στο Σχήμα Π.4 παρουσιάζεται το μέσο ποσοστό βελτίωσης κάθε αλγορίθμου για όλα τα σετ πειραμάτων με ΧΠ. Είναι αρκετά σαφές ότι οι αλγόριθμοι παρέχουν σημαντική βελτίωση σε σύγκριση με τη ΜΕ. Από το διάγραμμα φαίνεται ότι η ΣΑΦ οδηγεί σε σημαντικές βελτιώσεις όταν οι πελάτες απαιτούν εξυπηρέτηση μέσα σε ένα χρονικό παράθυρο, και ιδίως στις περιπτώσεις με ομαδοποιημένη γεωγραφική κατανομή (C-προβλήματα). Όσον αφορά τους εναλλακτικούς αλγόριθμους, μπορούμε να πούμε ότι ο 'Δυν_ΣΑΦ3' φαίνεται να υπερτερεί των υπολοίπων.



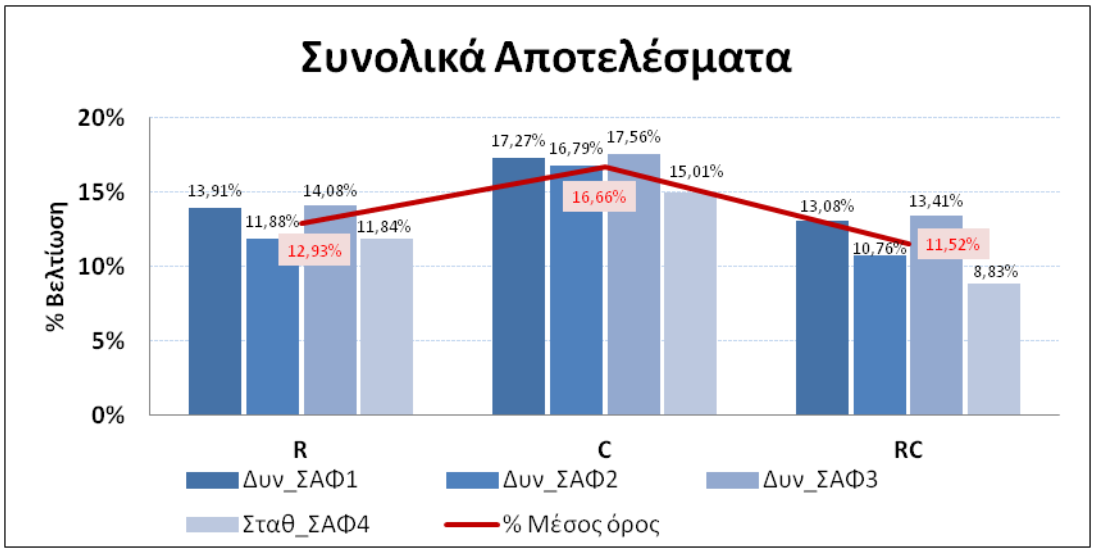
Σχήμα Π.4: Μέσο ποσοστό βελτίωσης κάθε αλγορίθμου για όλα τα σετ πειραμάτων με ΧΠ

Στο Σχήμα Π.5 παρουσιάζονται τα συγκεντρωτικά αποτελέσματα που προέκυψαν για τα σετ πειραμάτων χωρίς ΧΠ. Για τις περιπτώσεις αυτές, φαίνεται να υπάρχει μια ισορροπημένη βελτίωση ανεξάρτητα από τη γεωγραφική κατανομή των πελατών. Αυτό οφείλεται κυρίως στο γεγονός ότι οι πελάτες μπορούν τώρα να εξυπηρετηθούν ανά πάσα στιγμή μέσα στο διαθέσιμο χρονικό ορίζοντα, ανεξάρτητα από την θέση των πελατών.



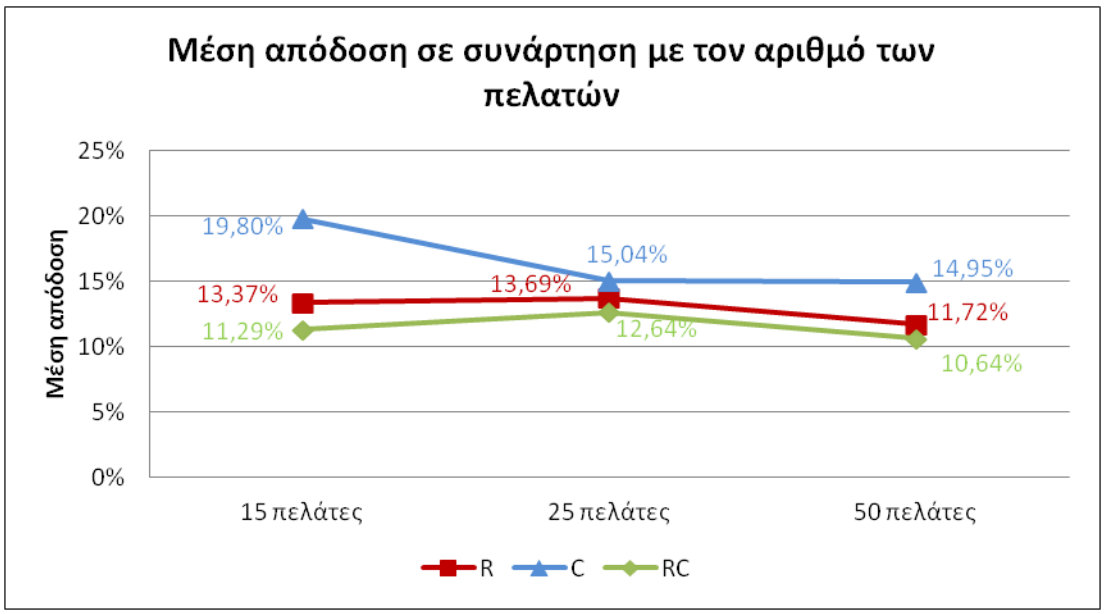
Σχήμα Π.5: Μέσο ποσοστό βελτίωσης κάθε αλγορίθμου για όλα τα σετ πειραμάτων χωρίς ΧΠ

Το Σχήμα Π.6 συνοψίζει τα αποτελέσματα από όλα τα σετ πειραμάτων. Είναι αρκετά σαφές, ότι η ΣΑΦ αποδίδει καλύτερα στα C-προβλήματα, με μια μέση βελτίωση 16.7%. Επιπλέον, ο αλγόριθμος 'Δυν_ΣΑΦ3' φαίνεται να είναι ο πιο αποτελεσματικός όσον αφορά το κόστος δρομολόγησης για όλα τα σετ πειραμάτων.



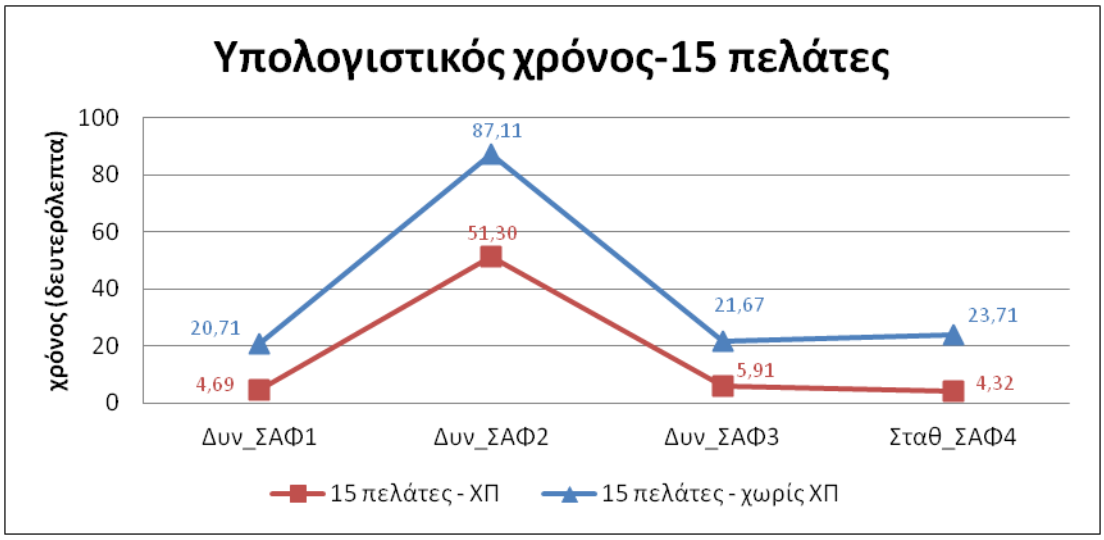
Σχήμα Π.6: Μέσο ποσοστό βελτίωσης κάθε αλγορίθμου για όλα τα σετ πειραμάτων

Στο Σχήμα Π.7 παρουσιάζεται μια επισκόπηση της απόδοσης της ΣΑΦ ως συνάρτηση του αριθμού των πελατών, με βάση την γεωγραφική τους κατανομή. Από το Σχήμα αυτό, μπορούμε να συμπεράνουμε ότι η Στρατηγική φαίνεται να έχει την καλύτερη απόδοση στα C-προβλήματα, όπως ήταν αναμενόμενο, και ιδίως στα μικρά παραδείγματα (σετ 15 πελατών). Επιπλέον, πρέπει να αναφερθεί ότι ο ευρετικός μηχανισμός που χρησιμοποιείται για την επίλυση της Φάσης I των αλγορίθμων είναι υπεύθυνος για αυτή τη μικρή επιδείνωση της βελτίωσης στα μεγαλύτερα παραδείγματα (25 και 50 πελάτες). Αυτό σημαίνει ότι η χρήση μιας ακριβούς μεθόδου (π.χ. Column Generation) μπορεί να οδηγήσει σε καλύτερα αποτελέσματα.



Σχήμα Π.7: Μέση απόδοση της ΣΑΦ ως συνάρτηση του αριθμού των πελατών

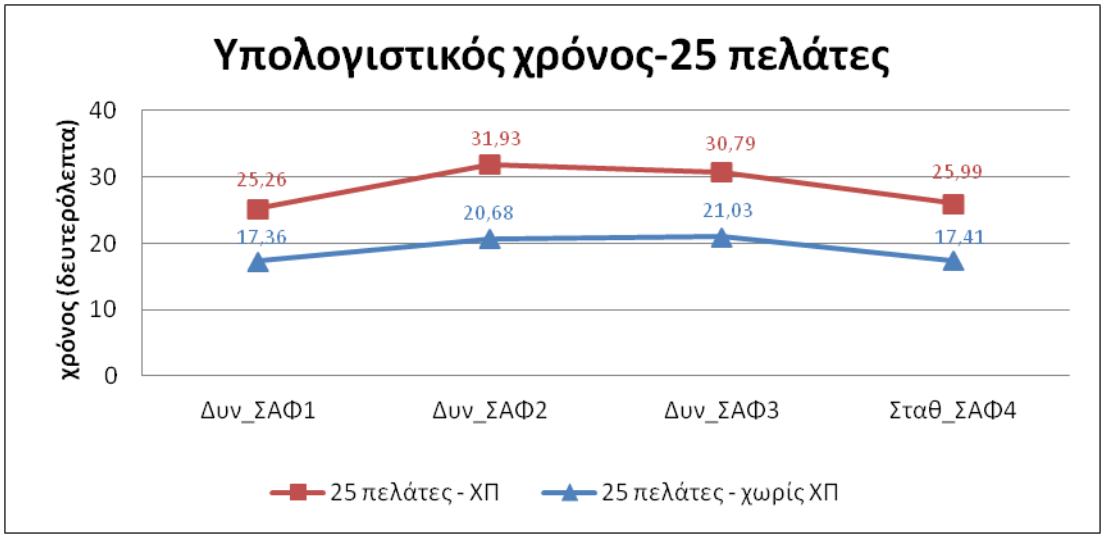
Στο Σχήμα Π.8 απεικονίζεται ο μέσος χρόνος που δαπανάται από κάθε αλγόριθμο για να λύσει ένα πρόβλημα 15 πελατών με ΧΠ (κόκκινη γραμμή) και χωρίς ΧΠ (μπλε γραμμή). Όπως αναμενόταν, τα προβλήματα με ΧΠ επιλύονται ταχύτερα, εξαιτίας του περιορισμένου χώρου λύσεων. Επίσης παρατηρείται ότι, στα προβλήματα με 15 πελάτες ο αλγόριθμος 'Δυν_ΣΑΦ2' απαιτεί περισσότερο υπολογιστικό χρόνο, όπως ήταν αναμενόμενο, λόγω της τεχνικής Branch-and-Price που χρησιμοποιείται στη Φάση ΙΙ.



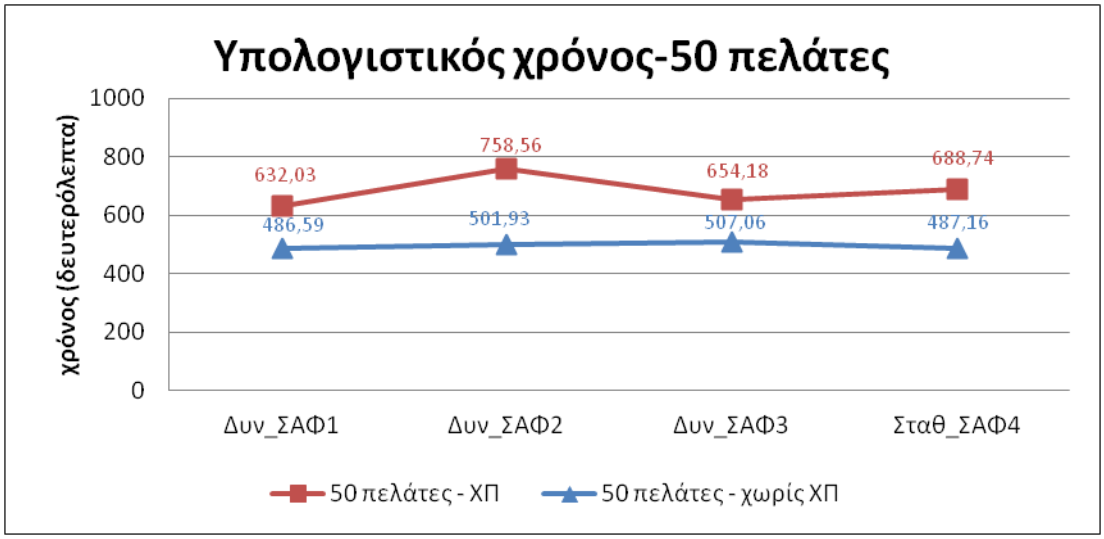
Σχήμα Π.8: Μέσος χρόνος κάθε αλγορίθμου για τα σετ πειραμάτων με 15 πελάτες

Για μεγαλύτερης κλίμακας προβλήματα, δεν υπάρχει σημαντική διαφορά μεταξύ των αλγορίθμων όσον αφορά τον υπολογιστικό χρόνο. Τα σχήματα Π.9 και Π.10 παρέχουν την υπολογιστική προσπάθεια όλων των αλγορίθμων για τα προβήματα με 25 και 50 πελάτες αντίστοιχα. Θα πρέπει να σημειωθεί εδώ, ότι το μεγαλύτερο μέρος του χρόνου που απαιτείται από τους αλγόριθμους, δαπανάται κατά τη διάρκεια της πρώτης φάσης της μεθόδου του κάθε αλγορίθμου (επίλυση του ΠΔΟ). Αν και ο αλγόριθμος των Clarke and Wright που χρησιμοποιείται είναι αρκετά γρήγορος, η διαδικασία βελτιστοποίησης Route Interchange απαιτεί σημαντική υπολογιστική προσπάθεια (ιδίως στα προβήματα με 50 πελάτες). Ωστόσο, κάποιος θα μπορούσε να χρησιμοποιήσει γρηγορότερες ευρετικές ή μετευρετικές μεθόδους (π.χ. Tabu Search), οι οποίες μπορούν να μειώσουν σημαντικά την υπολογιστική προσπάθεια που απαιτείται για τη διαδικασία της βελτιστοποίησης.

Αξίζει επίσης να σημειωθεί ότι οι αλγόριθμοι απαιτούν λιγότερο χρόνο στα προβλήματα που δεν έχουμε ΧΠ, διότι η C&W, που χρησιμοποιείται για την επίλυση της πρώτης φάσης, παρέχει ταχύτερα λύσεις όταν δεν υφίστανται ΧΠ.



Σχήμα Π.9: Μέσος χρόνος κάθε αλγορίθμου για τα σετ πειραμάτων με 25 πελάτες



Σχήμα Π.10: Μέσος χρόνος κάθε αλγορίθμου για τα σετ πειραμάτων με 50 πελάτες

Τα σημαντικότερα συμπεράσματα που προέκυψαν από τη συνολική διεξαγωγή των πειραμάτων μπορούν να συνοψιστούν ως εξής:

- Η ΣΑΦ μπορεί να παρέχει καλύτερες λύσεις από τη ΜΕ σε αρκετές περιπτώσεις.
- Οι μεγαλύτερες βελτιώσεις λαμβάνονται στις περιπτώσεις που οι πελάτες έχουν ΧΠ και είναι ομαδοποιημένοι γεωγραφικά.

- Στα προβλήματα χωρίς ΧΠ, φαίνεται ότι η ΣΑΦ παρέχει ισοδύναμες (κατά μέσο όρο) βελτιώσεις σε όλες τις περιπτώσεις ανεξάρτητα από την γεωγραφική κατανομή των πελατών.
- Γενικά, παρατηρείται ότι οι λύσεις που προκύπτουν από την ΣΑΦ οδηγούν σε έναν πιο ισορροπημένο φόρτο εργασίας μεταξύ των οχημάτων.
- Όσον αφορά τα σημεία όπου γίνεται η ανταλλαγή (μεταφόρτωση) των φορτίων, η χρήση δυναμικών τοποθεσιών (σημεία πελατών) οδηγεί σε καλύτερες λύσεις σε σχέση με το σταθερό σημείο μεταφόρτωσης.
- Ο αλγόριθμος 'Δυν_ΣΑΦ1' παρέχει τις ταχύτερες λύσεις. Παρ' όλα αυτά, οι 'ΔυνΣΑΦ3' και 'Σταθ_ΣΑΦ4' φαίνεται να έχουν παρόμοιες επιδόσεις.
- Στα μικρότερα προβλήματα (15-πελάτες), οι καλύτερες λύσεις λαμβάνονται από τον αλγόριθμο 'Δυν_ΣΑΦ2', ο οποίος όμως απαιτεί αρκετό υπολογιστικό χρόνο. Για τα μεγαλύτερα προβλήματα, η απόδοση του αλγορίθμου επιδεινώνεται εξαιτίας της ευρετικής μεθόδου (C&W) που χρησιμοποιείται για την εκτέλεση της Φάσης ΙΙ.
- Σε γενικές γραμμές, ο αλγόριθμος 'ΔυνΣΑΦ3' φαίνεται να είναι ο πιο αποτελεσματικός, τόσο σε ποιότητα λύσης, όσο και σε υπολογιστική προσπάθεια.

ABSTRACT

The problem addressed in the present thesis, concerns a Vehicle Routing Problem with Dynamic Pickups (VRPDP). In such a problem, vehicles are destined to serve delivery requests known prior to the start of operations, and as the working plan unfolds newly arrived pickup requests are assigned to the fleet of vehicles. Solution approaches proposed by the DeOPSys lab to address this problem, allocate the dynamic requests to the most appropriate vehicles, allowing, if it's profitable, the change in sequence of the delivery orders of a certain vehicle. Each vehicle, however, is restricted to serve the delivery orders assigned to it at the beginning of the time horizon.

The main purpose of the current thesis was to solve the VRPDP using a novel strategy that allows transshipment of delivery orders between vehicles. This strategy leads to a holistic view of routing operations at a reoptimization period allowing each request (static or dynamic) to be served by any vehicle; therefore the strategy has the potential to provide more effective solutions, in terms of travel costs and service quality. So, the main contribution of the thesis concerns the definement of this original strategy, first proposed in the DeOPSys lab, and the implementation of Load Exchange Strategy (LES) in a VRPDP environment of two vehicles. Computational results illustrate that the proposed strategy offers superior results in many cases, improving the solutions of the previous approaches over 15% on average. The strategy investigated in this thesis may form the basis for further research in the DeOPSys laboratory.

CONTENTS

Chapter 1: Introduction 1

Chapter 2: Theoretical Background 7

2.1 The Vehicle Routing Problem (VRP) 8

2.2 The Dynamic Vehicle Routing Problem (DVRP) 11

2.3 Advanced Strategies for DVRP 17

2.4 Load Exchange-related Strategies 21

2.5 Research Gaps and Contribution of the Thesis 28

Chapter 3: Problem Description 30

3.1 The Vehicle Routing Problem with Dynamic Pickups (VRPDP) 31

3.2 Load Exchange Strategy for the VRPDP 35

3.3 Decision Components and Alternative Policies for the LES 38

3.4 Decision Policies Addressed in the Current Thesis 46

Chapter 4: Solution Methodology 48

4.1 Assumptions and Overview of the Solution Methodology 49

4.2 Dynamic Exchange Location 50

4.3 Fixed Exchange Location 62

4.4 An Illustrative Example 64

Chapter 5: Experimental Analysis 70

5.1 Experimental Setup 71

5.2 Computational Results 75

5.3 Conclusions and Findings 85

Chapter 6: Conclusions 87

References 91

Appendices 97

LIST OF FIGURES

2.1 A solution example of the VRP 9

2.2 A dynamic vehicle routing scenario: (a) Initial routing solution, (b) Emergence of DRs, (c) Incorporation of DRs into the plan 13

2.3 Example of a Pickup and Delivery Problem allowing the exchange of loads between vehicles 22

3.1 Illustrative example of the VRPDP 33

3.2 VRPDP solution process during the available planning horizon 34

3.3 Example of Load Exchange Strategy 36

3.4 Components and alternative decision policies 38

3.5 Example of LES when there are no remaining customers 39

3.6 Example of LES in case vehicle is unable to serve remaining orders due to unforeseen events 40

3.7 Example of LES in case of a breakdown vehicle 41

3.8 Example of the one-to-one policy 42

3.9 Example of the one-to-many policy 43

3.10 Potential exchange location decision policies 46

3.11 Decision policies considered from the entire LES decision tree 47

4.1 Overview of the solution methodology 50

4.2 Route Interchange within a single route 55

4.3 Route Interchange between two routes 55

4.4 Pseudocode of the Simple Local Search 58

4.5 Example of Simple Local Search 59

4.6 Pseudocode of the post-optimization process 60

4.7 Example of Fixed Transshipment Point Incorporation 63

4.8 Pseudocode of the Fixed Transshipment Point Incorporation 64

4.9 Computational Example: (a) Initial planned schedule at time t=0, (b) Schedule status at replanning timestamp (t=40) 66

4.10 Optimal Solution when using a traditional approach (Total Cost: 912.3) 66

4.11 Phase I - VRP solution (Total Cost: 602.9) 67

4.12 Four (4) potential solutions of Phase II (Simple Local Search). Solution (c) is the qualified one due to feasibility reasons (Total Cost: 663.8) 68

4.13 Phase III – Post-Optimization (Final Cost: 641.98)..... 68

4.14 LES solution using a fixed transshipment point (Total Cost: 728.53)..... 69

5.1 Average improvement percentage of each algorithm for 15 customer requests with TWs..... 76

5.2 Average improvement percentage of each algorithm for 25 customer requests with TWs..... 77

5.3 Average improvement percentage of each algorithm for 50 customer requests with TWs..... 77

5.4 Average improvement percentage of each algorithm for 15 customer requests without TWs..... 78

5.5 Average improvement percentage of each algorithm for 25 customer requests without TWs..... 79

5.6 Average improvement percentage of each algorithm for 50 customer requests without TWs..... 79

5.7 Average improvement percentage of each algorithm for all TWed cases 80

5.8 Average improvement percentage of each algorithm for all non-TWed test cases..... 81

5.9 Average improvement percentage of each algorithm for all test cases..... 81

5.10 Average performance of LES as a function of the number of customers 82

5.11 Average time of each algorithm for test sets with 15 customer requests..... 83

5.12 Average time of each algorithm for test sets with 25 customer requests..... 84

5.13 Average time of each algorithm for test sets with 50 customer requests..... 84

LIST OF TABLES

1.1 Parameters and alternative policies for the LES 4

2.1 Some problems addressed in the literature considering the transfer operation between vehicles 27

4.1 Computational example: the data of the requests 65

5.1 Sets of test cases 71

5.2 Created patterns of TWs for the sets of 25 and 50 customers 73

5.3 Characteristics of the data sets 73

5.4 Basic features for the solution algorithms 75

5.5 Average time (in seconds) of each algorithm for each test case 82

A.1 Results for 15 customers with Time Windows 97

A.2 Results for 25 customers with Time Windows 97

A.3 Results for 50 customers with Time Windows 98

A.4 Results for 15 customers without Time Windows 98

A.5 Results for 25 customers without Time Windows 99

A.6 Results for 50 customers without Time Windows 99

CHAPTER 1 INTRODUCTION

Supply chains have become a competitive leverage in the global economy. Supply chain management refers to the design and management of all operations and activities related with procurement procedures, production, processing and all distribution activities. The remarkable advances in telecommunications and information technology have enabled companies to focus on velocity and timeliness throughout the supply chain (Larsen *et al*, 2008). Low-cost and lean transportation and distribution activities have received significant attention, since they affect the overall supply chain costs, quality of customer service and the total cost of the product. Additionally, in addition to cost-reduction aspects of the transportation activities, environmental issues appear to play a very important role in the strategic and operational perspectives of modern companies worldwide. Optimizing the routing of vehicles may lead to better vehicle utilization and, consequently, to the reduction of CO_2 emissions, thus leading to more environment-friendly operations in transportation activities.

In an attempt to address the above issues, significant research has been conducted in vehicle routing. The majority of this research has focused on deterministic and static models, mainly for helping companies during the planning phase of their routing procedures. Distribution companies often use a fleet of vehicles (own or rented) commencing from a single or multiple depots in order to serve a number of customers with known demands and service costs. The Vehicle Routing Problem (VRP), as addressed in the literature, is usually modeled as an integer programming problem, where the objective is the minimization of distribution costs.

However, the current way of conducting business has raised significantly customer service expectations; it is not unusual for customers to require service in real-time during the execution of the designed plan. Additional disruptions in the execution of the original distribution plan may stem from delays due to traffic congestion, or to unavailability of docking space, vehicle breakdowns, temporary alterations in the road network, etc. The disruption caused by these dynamic and unexpected events, led research community to focus on the dynamic counterpart of the generic VRP (*Dynamic VRP – DVRP*). The DVRP is a typical example of distribution, in which companies must quickly and smartly use real-time information, in order to reduce their total costs and provide superior customer service. For instance, in courier operations, dynamism is commonly implied by arrival of new requests. Those requests arise dynamically over time as the working plan unfolds. Several variations of the DVRP exist in order to adapt to various practical characteristics and constraints. DVRP corresponds to a more demanding and difficult problem than the VRP. As a consequence, it's not always feasible to obtain optimal solutions to problems of practical sizes within reasonable timeframes.

During the past decade more and more researchers deal with dynamic transportation models. The DVRP is only a subset of these models. A typical way of solving a DVRP is to employ a sequence of reoptimization steps, where at each step a suitable (static) problem is solved, incorporating an appropriate portion of the dynamic up-to-date information. Several methods for solving DVRPs exist in the literature, ranging from exact algorithms, simple policy based techniques, problem specific heuristics and metaheuristics. Typically, a static algorithm is initially applied to the information known *a priori* in order to design an original plan. Two basic approaches are usually applied to deal with the newly emerging information: i) local update procedures, that

try to incorporate the newly received information to the currently designed plan and ii) reoptimization procedures, that provide a solution of the past and new information from scratch. Lately, several advanced other strategies are being investigated in order to react to the occurrence of dynamic events, such as waiting strategies, diversion, anticipation of future requests and time-dependent solution methods.

This thesis is based on the work of Ninikas and Minis (2011) which was motivated by practical courier applications (Ninikas *et al.*, 2011). As stated in the aforementioned work: "in a typical courier distribution setting, a set of delivery vehicles, originating from a local distribution hub (depot), is tasked to serve delivery orders known prior to the start of operations, called *static orders*. As the working plan unfolds, however, customer orders are received through a call center, called *dynamic orders*, requesting on-site pickup within the current period of operations. At a given time instance, selected by the dispatcher, the newly arrived orders need to be incorporated in the partially executed plan. Those pickup orders have to be collected and returned to the depot for further processing". "Backup" vehicles may be initially located at the depot, in order to be dispatched when necessary to serve newly arrived orders. Exactly this case is considered in the current thesis and for simplicity is referred as the *Vehicle Routing Problem with Dynamic Pickups, VRPDP (Ninikas and Minis, 2011).*

Transportation companies that operate within such a context, face a lot of difficulties during their effort to incorporate the dynamic orders into the planned routes. Those difficulties correspond to the following:

- The large number of dynamic requests that arise within the day complicates significantly the decision-taking of the dispatchers in order to implement "good" strategies for replanning their fleet in a real-time fashion.
- In a high density urban environment, there are overlapping service regions for vehicles, resulting to complex decisions regarding which vehicle will be assigned to serve a newly received order.
- The original assignment of specific delivery orders to each vehicle, further limits the optimization margins, since those assigned orders must be served only by this vehicle (since it carries the load to be delivered to the customers), forcing the vehicles to follow a routing plan that is constantly changing in a real-time manner.

In the present thesis, we address the latter limitation by allowing vehicles to meet each other in real-time and exchange delivery orders, in order to share workload and better adopt in the new, updated picture of the routing plan. We refer to this novel approach, as the *Load Exchange Strategy (LES)*.

Load Exchange is a rather complex and hard to implement, due to the high number of parameters that have to be determined. Those parameters are summarized in Table 1.1.

Table 1.1: Parameters and alternative policies for the LES

Parameter	Description	Alternatives
Implementation time of the LES	The time instance or the conditions under which someone could apply the strategy	• at a replanning timestamp • when there are no remaining orders on the vehicle • on vehicle's idle time • when a vehicle is unable to serve its remaining orders
Meeting combinations	The allowance for a vehicle to serve a number of delivery orders originally assigned to one or more than one other vehicles *en route*	• one-to-one • one-to-many
Exchange locations	The locations where vehicles are allowed to exchange the loads	• on-site of not yet served customers • en route • depot • predefined (static) meeting locations (pigeonholes or parking lots)

In the current thesis, LES is applied in the VRPDP, incorporating several of the aforementioned parameters. Specifically, at any given replanning timestamp, a fleet of two vehicles is considered where one corresponds to a vehicle always *en route*, and the other one either *en route* or located at the depot; the latter ("backup" vehicle) may

be waiting to be dispatched for the service of newly received orders. During the replanning timestamp, LES is triggered to examine whether it is profitable for vehicles to meet and exchange orders or not; if not, a typical replanning algorithm is applied. This is why we refer to this novel method as "strategy". Capacity constraints aren't included in the problem, because the load of each request is relatively small in the applications considered (e.g. courier service industries). The proposed approach investigates several options, such as dynamic or fixed transshipment locations, exact or heuristic methods for the solution of components of the overall algorithm, etc.

Our proposed algorithm is comprised of three steps. During the first phase, a VRP is being solved with all remaining orders (static and dynamic). No pickup or delivery requirements are assumed, in order for all vehicles to be able to serve all requests. The second phase, examines whether there are delivery orders initially assigned to a vehicle that now have been assigned to the other vehicle (based on the VRP solution). If not, the VRP solution remains unchanged and comprises the final solution. Otherwise, a new procedure is being used searching for the optimal incorporation of the exchange location into the routes. Finally, in the third phase of the approach, a post-optimization process identifies if interchanging a customer between the two routes can further improve the current solution at-hand.

The main objective of the current diploma thesis is to investigate this strategy, which can be applied to VRPs in order to improve the solution results. The objective is to use original ideas for solving dynamic VRPs by incorporating the novel proposed strategy (LES). The main contribution of the thesis is the application of LES to a dynamic problem with static deliveries and dynamic pickups (VRPDP). To the best of our knowledge, this scheme has not been studied and has yet to be investigated. References related with load exchange-related strategies have been published in the literature, mostly for pickup and delivery problems with fixed transshipment points.

The algorithmic approach proposed will be used by the Design, Operations & Production Systems Lab (DeOPSys) of the Financial and Management Engineering (FME) Department of the University of the Aegean, where several algorithms for problems of the dynamic VRP class have been developed and studied.

The remainder of the thesis is structured as follows: Chapter 2 provides the basic theoretical background of the problem at-hand. A thorough discussion regarding

other related advanced strategies applied in dynamic problems as well as methodologies related to the LES is given, followed by the related research gaps and contribution of the thesis. Chapter 3 presents the dynamic problem considered (VRPDP) and the LES framework, and discusses several characteristics of the strategy and the investigated parameters. Chapter 4 provides a detailed description of the solution methodology. Results from extensive computational testing and evaluation are provided in Chapter 5. Finally, Chapter 6 summarizes the main findings and concluding remarks, along with directions for further research.

CHAPTER 2 THEORETICAL BACKGROUND

The present chapter overviews the basic theoretical background of the problem solved in the current thesis. Initially, the Vehicle Routing Problem (VRP) is introduced, followed by the Dynamic Vehicle Routing Problem (DVRP) which defines the family of problems, to which the one undertaken in this thesis belongs. The basic solution methods proposed in the literature for the DVRP are described, as well as recent advanced solution approaches. In Section 2.4 we introduce the idea of load transshipment between vehicles, while applications and references relevant to the Load Exchange Strategy (LES) are also presented. Finally, Section 2.5 provides the related research gaps and the contribution of the current thesis.

2.1 THE VEHICLE ROUTING PROBLEM

The vehicle routing problem (VRP) is one of the most studied problems of Operations Research, and many mathematical programming techniques have been developed for solving it. Practical supply chain and distribution systems are highly connected with the VRP and its extensions, since it forms the basic tool that models and provides solutions in the field of transportation, distribution and logistics. Mathematical programming and efficient algorithmic approaches for addressing this problem play a significant role for companies active in the supply chain are, since the related solutions may have a significant impact on related operating costs. The majority of real-world applications (both in North America and Europe), have shown that the use of computerized procedures for planning the distribution process result in substantial savings (generally from 5% to 20%) in transportation costs (Toth & Vigo, 2002).

The VRP is a generalization of the classic Traveling Salesman Problem (TSP) (Christofides, 1979; Cornuejols and Nemhauser, 1978; Gendreau *et al.*, 1997), and it concerns the distribution of goods between depots and customers (final users). It was firstly introduced by Dantzig and Ramser (1959), who proposed a mathematical programming formulation and an algorithmic approach to solve a real-life problem for the delivery of gasoline to service stations. The definition of VRP and its variants, as well as an extensive analysis of solution methods are presented by Toth & Vigo (2002). Nowadays numerous commercial software applications are available that embed advanced algorithmic approaches for the solution of different real-life VRPs.

In a typical VRP setting, every customer represents a node of a network. Every customer has a known demand and must be served once by only one vehicle. Every arc (i, j) of the network (where i and j correspond to all nodes of the network) is associated with a cost c_{ij} which represents the cost of traveling from i to j. Every vehicle has a specific capacity and its route must start and end at a specific depot. The total demand of the customers served by a vehicle cannot exceed the vehicle's capacity. The objective of the problem is to minimize the total cost traveled by all vehicles. Figure 2.1 presents a network of customers along with the feasible solution of the vehicle routing problem.

According to Steward and Golden (1983), a compact and convenient formulation for the VRP can be written as follows:

$$Minimize \sum_{k} \sum_{i,j} c_{ij} x_{ijk}$$

Subject to

$$\sum_{i,j} \mu_i x_{ijk} \leq Q \quad k = 1, 2, \ldots, m$$

$$x = \lfloor x_{ijk} \rfloor \in S_m$$

where:

c_{ij} = the cost of traveling from i to j

x_{ijk} = 1 if the vehicle k travels from i to j and 0 otherwise

m = the number of available vehicles

S_m = the set of all feasible solutions in m-traveling salesman problem (m-TSP)

μ_i = the demand at location i

Q = the vehicle capacity

According to the above formulation, VRP is modeled as an integer-programming problem. VRP falls into the category of NP-hard optimization problems, in which computational time increases exponentially with the problem's size. That is the main reason that exact methods fail to solve VRP optimally in reasonable computational time. As a result, exact solution methods are used for limited-size problem instances, while heuristics and metaheuristics are proposed in most of the other cases.

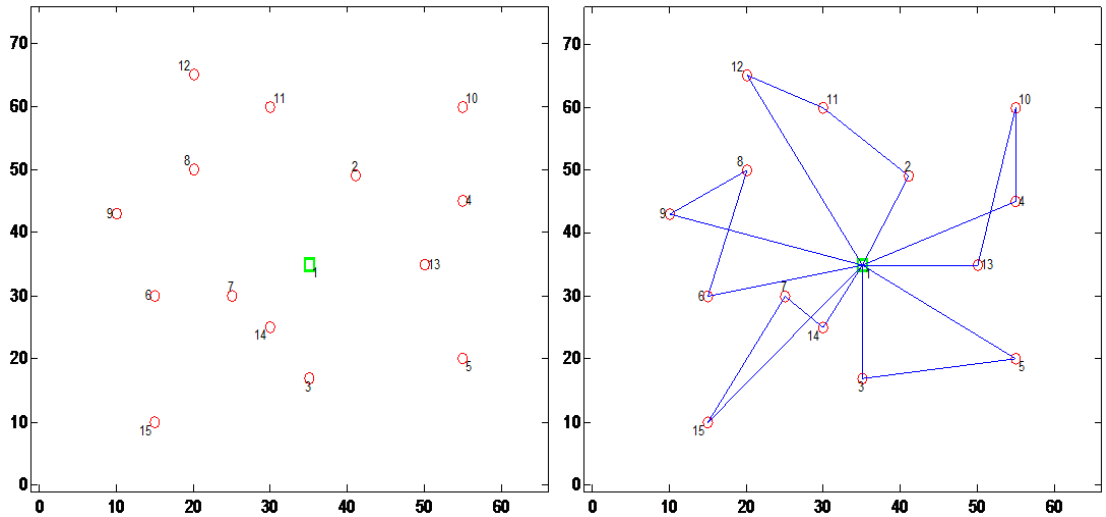


Figure 2.1: A solution example of the VRP

There are many variations of this classical problem, depending on the constraints of the problem at-hand. The main variations are discussed below:

- **Capacitated VRP (CVRP):** It is almost identical to the conventional VRP, since vehicles in most of the models introduced to the literature have restricted capacity. In this problem all demands are deterministic, known *a-priori* and cannot be separated. The objective is to minimize the total routing cost by servicing all customers exactly once without exceeding capacity constraints (Toth & Vigo, 2002).
- **VRP with Time Windows (VRPTW):** It's an extension of the CVRP, including time window constraints. The time window is a specific, predefined time interval associated with a customer in which the customer must start being served. This time interval is not the same for all customers.
- **VRP with Pickup and Delivery (VRPPD):** Restrictions on delivery and collection of goods are included in this problem. Here, every customer is associated with two service locations, one for the pickup and one for the delivery of goods. The goal is to find optimal routes for a fleet of vehicles to visit the pickup and drop-off locations.
- **VRP with Backhauls (VRPB):** This problem is also an extension of the CVRP, in which customers are divided into two sets. The first set of customers requires a quantity of goods to be picked up from their location and returned back to the depot, while the other requires a quantity of goods to be delivered to them.

Other VRP variations may include:

- the **Distance Constrained VRP** (Toth & Vigo, 2002)
- the **Multi-Depot VRP** (Bianco *et al*., 1994; Carpaneto *et al*., 1989)
- the **Heterogeneous Capacitated VRP** (Taillard, 1996)
- the **Multi-Period VRP** (Tan and Beasley, 1984; Christofides and Beasley, 1984).

There is an extensive literature regarding methods solving the VRP. The interested reader may refer to the work of Toth and Vigo (2002), Christofides, Mingozzi *et al*. (1979), Desrochers *et al*. (1990), Laporte (1992), Golden and Assad (1988), and Laporte and Osman (1995).

2.2 THE DYNAMIC VEHICLE ROUTING PROBLEM

The Dynamic Vehicle Routing Problem (DVRP) is the dynamic counterpart of the generic VRP mentioned above (Larsen *et al.*, 2008). Dynamic routing of a fleet of vehicles refers to distribution problems in which information is dynamically revealed to the decision maker. During the past decade, the research community focuses more and more on dynamic problems, developing various related models and algorithms. Rapid growth in telecommunications and information technology have led to this direction, since, based on these advancements, distribution companies are able to monitor the vehicles' location and status in a real-time fashion. Related applications and systems include dynamic fleet management systems, courier service systems, dial-a-ride systems, emergency systems, etc.

According to Larsen *et al.* (2007) the DVRP has two main differences compared with the VRP:

- Not all information relevant to the planning of the routes is known by the planner when the routing process begins
- Information may change after the initial routes have been designed.

DVRP is a more elaborate and complex problem than its static counterpart, since solution methods and algorithms are based on data received from the fleet of vehicles in real time. It also belongs to the class of NP-hard optimization problems, and as a result, it's not always feasible to obtain optimal solutions to problems of practical size within a reasonable timeframe.

Psaraftis (1988, 1995) presents the following 12 issues that distinguish the dynamic vehicle routing problem from the conventional static routing problem:

1. *Time dimension is essential*
2. *The problem may be open-ended*
3. *Future information may be imprecise or unknown*
4. *Near-term events are more important*
5. *Information update mechanisms are essential*
6. *Re-sequencing and reassigning decisions may be warranted*
7. *Faster computation times are necessary*
8. *Indefinite deferment mechanisms are essential*

9. *Objective function may be different*
10. *Time constraints may be different*
11. *Flexibility to vary vehicle fleet size is lower*
12. *Queuing considerations may become important.*

The full discussion of these issues can be found in Psaraftis (1988) and Psaraftis (1995).

Figure 2.2 presents a simple example of a dynamic vehicle routing situation. In this example, two un-capacitated vehicles must serve static orders that are known *a-priori* (represented by black nodes) as well as dynamic requests (depicted by white nodes) that are revealed during the execution of the designed routing plan. Figure 2.2a represents the initial routing solution prior to the vehicles leaving the depot. During the situation of Figure 2.2b, vehicles have already performed a part of the plan (dashed lines) and are on their way to their next destination (thick lines). However, at this moment, several dynamic requests *(DRs)* are received that need to be incorporated in the current plan. Ideally, the DRs should be inserted into the already planned routes without changing the order of the non-served customers and with the minimal increase in total cost traveled. This is depicted in figure 2.2c, where a DR can be successfully fit on the current plan of the trip (trip to the North of the depot). However, as illustrated by the South route of the same figure, the insertion of the new customer creates a large detour, illustrating that the insertion of DRs in the existing plan is much more complex, usually requiring a re-planning of the routes in order to incorporate the newly received requests.

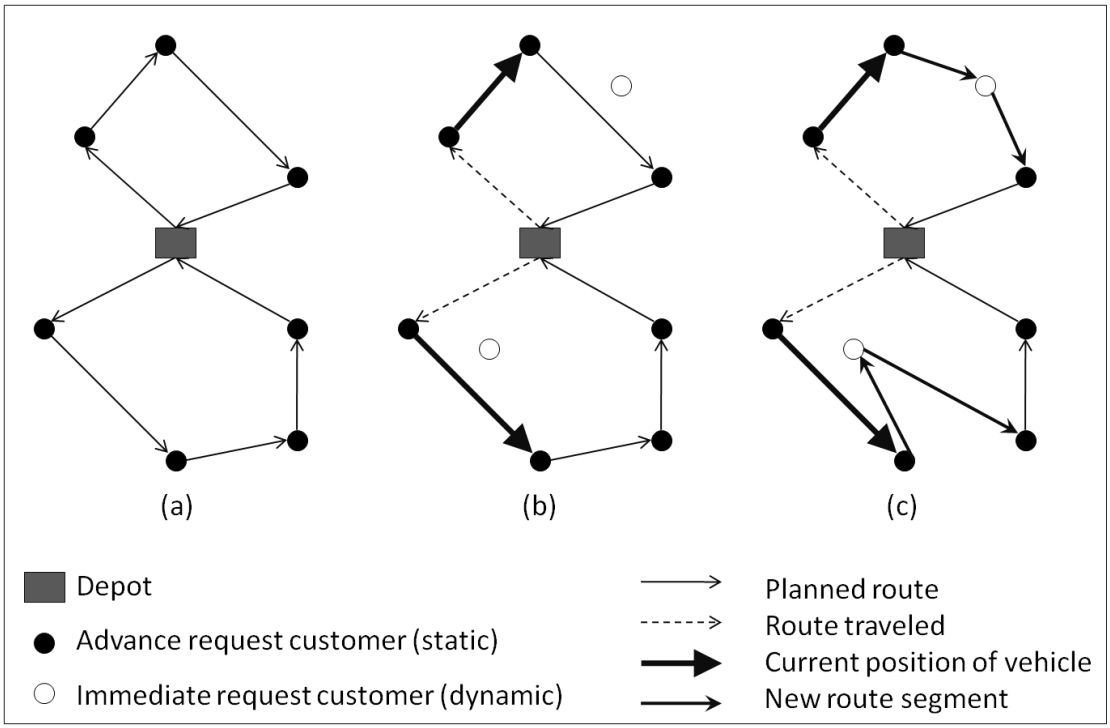


Figure 2.2: A dynamic vehicle routing scenario: (a) Initial routing solution, (b) Emergence of DRs, (c) Incorporation of DRs into the plan

Generally speaking, the more restricted and complex the routing problem is, the more complicated the insertion of new dynamic customers will be. For instance, the insertion of new customers in a time window constrained routing problem will usually be much more difficult than in a non-time constrained problem (Larsen, 2000).

2.2.1 Classification of DVRPs

A DVRP can be either deterministic or stochastic (Powell *et al.*, 1995). In deterministic and dynamic problems, some parameters or variables depend on time but there is no randomness. On the contrary, in stochastic problems, the actual demand or time to start service at a customer location may be a random variable over time. Uncertainty is an inherent characteristic of this kind of problems.

Powell *et al.* (1995), also distinguish between dynamism within a problem, a model and the application of a model. According to them:

- A **problem** is dynamic if one or more of its parameters can be defined as function of time. This includes models with dynamic data that change constantly as well as problems with time-dependent data which are known in advance.

- A **model** is dynamic if it explicitly incorporates the interaction of activities over time. Here once should distinguish between deterministic dynamic models and stochastic models.
- An **application** is dynamic if the underlying model is solved repeatedly as new information is received. Consequently, solving models within dynamic applications require significant computational resources.

Dynamic vehicle routing problems can be divided into two basic categories, depending on the major feature that causes the dynamism:

1. Problems for which dynamism depends on the travel time (**DVRPs-Travel Times**).
2. Problems for which dynamism depends on customer requests (either occurrence of new requests, or the variability of the demand required) (**DVRPs-Customer Requests**).

DVRPs – Travel times

In these problems, travel times among customers vary depending on the time the route is executed. This models the effects of different levels of road traffic during the day, road construction, accidents, weather conditions, etc. These deviations have varying degrees of predictability, and forecasts may be useful in order to estimate the travel times between customer requests. The problem dealt in this thesis is not related to this class of problems.

DVRPs – Customer Requests

The most significant part of the literature has focused on cases in which the dynamism relates to customer requests. This class of problems may be defined as follows: A fleet of vehicles is *en route* to serve customers revealed dynamically over time (i.e. during the shift). With the occurrence of a new request, the current plan must be re-designed in order to incorporate the up-to-date information (i.e. include the new request), by taking under consideration all past and new information.

This class of problems may be further divided into two major subclasses, based on the service type of the dynamic requests:

- **Many-to-one** (one-to-many) problems, in which each dynamic request is associated with a single location (e.g. pickup or delivery of an object, e.g. next-day courier services).
- **Many-to-many** problems, in which each dynamic request may be associated with more than one locations (e.g. pickup and delivery of an item, same-day courier services, dial-a-ride, etc.).

Additionally, regarding the degree of dynamism, DVR problems can be classified into three main levels:

- **Weak dynamic systems**: In those systems, only a small portion of customer requests is revealed dynamically, while the largest one is known in advance (i.e. prior to start of execution). The objective of those systems is mainly the minimization of the distribution costs.
- **Moderate dynamic systems**: Dynamic requests occupy a significant percentage of total service requests, but not at the level that one should take into account when designing the initial plan. The objective here comprises a combination of cost minimization and response time to dynamic service requests.
- **Highly dynamic systems**: They comprise the most extreme case of dynamic routing systems, met mainly in emergency services such as police, fire department and ambulances. On those cases, no requests are known in advance and the routing plan is constantly changing (in a real-time fashion) based on the newly received requests. Those applications are characterized by a strong focus on response time minimization.

The complexity of a dynamic vehicle routing system can be seen as a function of the number of customers and their spatial distribution, but most significantly, it depends on the number of dynamic events and their temporal distribution (Larsen *et al.*, 2008).

In the problem investigated in this thesis, a portion of service requests are revealed after the start of operations. Specifically, we investigate a problem with an *a-priori* designed routing plan, where dynamic requests (pickups) occur in real-time. Therefore, this problem is a many-to-one DVRP and it concerns a moderate dynamic system.

2.2.2 BASIC METHODS AND APPROACHES FOR DVRP

In this section, we overview the basic methodologies, algorithms and solution approaches for DVRPs, in which dynamism is due to the occurrence of new dynamic requests. The decision making related to the solution approach depends strongly on the following factors:

- Problem size, which concerns the number of both static (known in advance), and dynamic customer requests that occur during the execution of the initially designed plan (depending on the total number of requests to serve), and
- Computational effort needed, which concerns the time required for the problem to be solved. This factor is quite significant in DVR Problems due to the constantly changing information and the new status of the logistics resources after each time unit.

The DVRP is usually solved in a sequential manner, by repeatedly updating the existing route, either during the occurrence of some external factor (one or more dynamic requests), or at regular time intervals. Very often, the overall dynamic problem is decomposed into a sequence of static sub-problems which are solved by a static algorithm repeatedly (e.g. every hour), based on the current available information. Depending on the type of the basic solution procedure applied to solve the static sub-problems, solution methods for DVRP can be divided into four main categories:

- Exact algorithms
- Simple policy based techniques
- Problem specific heuristics
- Metaheuristics.

Typically, a static algorithm is initially applied to the requests known *a-priori* (e.g. early in the morning) in order to design an original route. Three basic approaches are usually applied to deal with the newly emerging requests:

- **Local update procedures**. Simple policy-based techniques and various heuristics are often used to incorporate the DRs in the current routing plan, by applying a fast local update procedure (e.g. insertion methods). The main characteristic of these procedures is their computational efficiency and their simplicity. However,

due to their myopic nature, they may lead to a local minimum/maximum. The interested reader can find more information regarding those approaches in Bertsimas and Van Ryzin (1991), Larsen *et al*. (2002) and Madsen *et al*. (1995).

- **Re-optimization procedures**. These procedures mainly use exact algorithms and metaheuristics in order to re-optimize the total VRP solution from scratch, by considering all the available information up to the related point in time and not any pre-designed routing solution. These procedures explore the overall feasible space of the problem and may yield solutions of improved quality in comparison to the first strategy. The drawback is the significant computational effort required. More information regarding this kind of solution approaches can be found in Bell *et al*. (1983), Fisher *et al*. (1982), Brown *et al*. (1987), Bausch *et al*. (1995), Gendreau *et al*. (1999), Montemanni *et al*. (2005) and Gambardella *et al*. (2003). Many researchers have combined local update procedures with re-optimization procedures creating hybrid algorithms in order to exploit the advantages of each approach.

- **Advanced strategies**. This category includes more advanced procedures, such as waiting strategies, diversion, anticipation of future requests, etc. which are described in the next section.

2.3 Advanced Strategies for DVRP

Recently, more sophisticated approaches than the aforementioned conventional solution approaches are proposed in the literature for solving the DVRPs. In this Section, we review some of these novel approaches.

2.3.1 Diversion

The purpose of this strategy is to divert a vehicle that is on its way to the next service destination in order to cover a request that just entered the system. Diversion is a very interesting area of research that has not been investigated in depth. Recent advances in telecommunications and information technologies (e.g. global positioning systems,

telematics, etc.) enable this strategy, since they allow dispatchers to be fully aware of the current state of the logistics system. However, it's difficult to integrate diversion into a solution methodology and a number of issues must be carefully addressed. For that reason, the next destination of a vehicle is considered fixed in most methods. Regan *et al*. (1995) were the first to apply diversion and they empirically evaluated the benefits of this approach in several ways.

The work of Ichoua *et al.* (2000) addresses a diversion strategy motivated from a courier service application. In this setting, parcels are collected from customers placed in a local area and are brought back to a central depot for further processing. The diversion strategy was integrated into a parallel tabu search heuristic originally proposed by Gendreau *et al.* (1999). Other optimization methodologies can be used as well. The suggested approach is applied whenever a new request occurs and seeks to exploit the new diversion opportunities that are offered in a highly dynamic environment. When such a strategy is enabled, the algorithm considers as starting point the current vehicle's location, instead of its next destination as usually considered by conventional solution approaches. For that reason, more options are available in order to incorporate the newly received requests.

2.3.2 Anticipation of Future Requests

Strategies of this type use historical information regarding the arrival of new requests. Specifically, various patterns for the arrival of the dynamic requests (e.g. time and location) may be used in order for appropriate strategies to be adopted based on the outcome of such information. In this case, human dispatchers are able to better manage their transportation resources by anticipating future needs, since they now have some valuable knowledge about spatial and temporal distribution of the DRs. Historical data may be used in order to determine probability distributions that can be used for the occurrence of new customer requests (both in terms of time and space). This has motivated researchers to develop solution procedures to exploit this knowledge. A large volume of historical data is required in order to extract good quality data regarding the customers' behavior. References related to these strategies can be found in the work of Ichoua *et al.* (2006) and Ghiani *et al.* (2009).

2.3.2.1 WAITING STRATEGIES

Waiting strategies examine the possibility of positioning vehicles at strategic locations in order to wait for the arrival of potential new (dynamic) requests. In case of problems with time-windows, there are cases in which a vehicle is forced to wait at the location of its next destination prior to the opening of the time window of this customer. The vehicle may wait at this location for sufficient time in order to reach its next destination, either exactly at the opening of the time window (earliest departure policy), or exactly before the closing of the time window (latest departure policy).

Waiting strategies may also be used as policies that allow vehicles to wait at new strategic locations, either during the execution of their original plan, or (more often) when they have completed their current service plan. For that case, more sophisticated strategies must be invented to determine the waiting times of the vehicles at strategic locations. A drawback of waiting strategies is that a vehicle may wait at a customer location longer than necessary.

Mitrovic-Minic and Laporte (2004) analyze this issue and present four waiting strategies for a pickup and delivery problem with time windows. These strategies concern: i) the drive-first strategy, ii) the wait-first strategy, iii) the dynamic waiting strategy and, iv) the advanced dynamic waiting strategy. The first two concern rather simple strategies, while the latter two were developed in order to improve distribution of the waiting times along routes, in order to facilitate future request insertions. The advanced dynamic waiting strategy, which combines earliest and latest departure times, seems to be the most efficient one with respect to the number of vehicles used and the total route length.

In Ichoua (2001), the distribution area is partitioned into geographic zones and the time horizon is divided into time periods. According to this work, a vehicle that has completed its service at a customer location is forced to wait for some time, if: a) its next destination is far enough, b) the probability of a request arrival in the vehicle's neighborhood in the near future is high enough and, c) there aren't many vehicles in the current service zone. This strategy seems to be very effective, especially for harder problems (i.e. small fleet of vehicles and high demand rates).

In the work of Branke *et al.* (2005), various waiting strategies have been used for a dynamic vehicle routing problem with no time windows. The authors consider a set of planned routes and the occurrence of a single new request, with either known or unknown arrival time, which is uniformly allocated within the service area. They examine if forcing a vehicle to wait at a customer location increases the probability of being able to serve the new request without violating time constraints. For the cases of one and two vehicles, they extract theoretical results about the best waiting strategy. Actually, the optimal strategy for the case of a single vehicle is not to wait, while for two vehicles the authors propose the optimal waiting strategy. Several waiting strategies and an evolutionary algorithm for the optimization of the waiting strategy are also proposed and tested. Furthermore, it is demonstrated that in comparison to the "no wait" strategy, the distribution of the slack time among the customers based on a proper waiting strategy can a) significantly increase the probability of serving the new request (up to 10%), while b) reduce the average detour incurred to serve this request (decrease to up to 35%).

2.3.2.2 DOUBLE HORIZON

This approach has been introduced in Mitrovic-Minic *et al.* (2004) for a dynamic pickup and delivery problem with time windows. The authors propose double horizon based heuristics for solving this problem, in which both a short-term and a long-term planning horizon are considered. In each planning horizon a different objective is sought. For example, the goal for the short-term horizon may be the minimization of the total distance traveled, while the objective for the long-term horizon may favor large slack times in the routes to better manage future requests. This idea is useful in contexts where near future actions depend on the solution proposed for the distant future. Extensive computational results have demonstrated the benefits of a double horizon approach compared with the classical single horizon approach.

2.3.2.3 FRUITFUL REGIONS

This approach is presented by Van Hemert and La Poutré (2004); the authors suggest a vehicle to move to a "fruitful" region, if this move doesn't violate any constraint for the known requests. A region is called fruitful, when there is a high possibility of occurrence of a new customer request in the specific region. This is achieved by

capitalizing on the related probability distributions. This strategy is incorporated into an evolutionary algorithm developed for a dynamic pickup and delivery problem. An interesting approach of this strategy has also been introduced in Van Hemert and La Poutré (2009).

2.3.3 Load Exchange-related Strategies

This is an innovative strategy recently addressed in the literature. It is applied in vehicle routing problems and allows transshipment of cargos between vehicles when advantageous. So, a request can be served by two vehicles. A vehicle can originally carry the load of a customer request, but finally another vehicle distributes it to the delivery location. Load exchange is an advanced strategy, used in practice by some courier companies for requests that require a load to be picked up from a location and be delivered to another in the same day. This is further described in the next section.

2.4 Load Exchange-related Strategies

In a typical VRP setting, delivery requests are usually loaded to the vehicle at the beginning of the shift. This gives a certain degree of limitation to the problem, since each delivery request can be served only by the vehicle that carries the load for this specific customer. The basic idea of the current Section (and of this thesis) is to relax this limitation by allowing loads (and, consequently, customers) to be transshipped (or exchanged) between vehicles.

Load exchange-related strategies have been inspired from many courier companies, who empirically apply this method in practice. Usually, these companies serve customers requiring transportation of an object from a pickup location to a delivery location (Pickup and Delivery Problem) within the same day. They usually partition the distribution area into a number of geographic zones and each vehicle is allowed to work in a certain zone. In case a delivery location of a request belongs to a service region of another vehicle, the drivers communicate and decide where and when they will meet to exchange the corresponding load needed. In those cases a customer request might be served by two vehicles; one collects the package from the pickup

location and the other one delivers it to the delivery location. Typically, there are predefined locations where this operation may be performed, usually referred to as transshipment points. This case has been basically addressed in the literature so far, while in the current thesis, additional options of allowing dynamic exchange locations are also considered. Figure 2.3 provides an illustrative example, where the solution obtained by a conventional approach (Fig. 2.3b) is compared to the one obtained when enabling a transshipment operation (Fig. 2.3c); it is quite clear that allowing such an operation may yield significant savings.

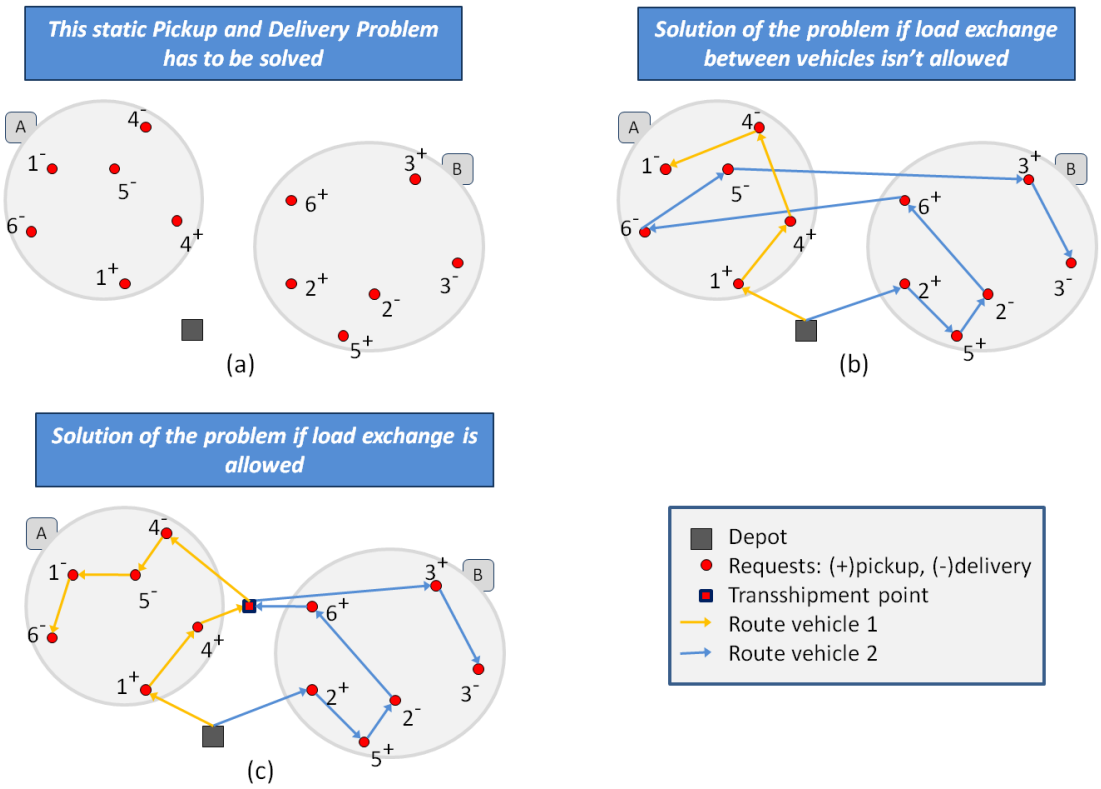


Figure 2.3: Example of a Pickup and Delivery Problem allowing the exchange of loads between vehicles

The majority of the research that has been conducted so far in this area focuses on the following three (3) major characteristics regarding transshipment operations:

- They address the static case of vehicle routing problems
- They deal only with many-to-many problems (i.e. pickup and delivery or dial-a-ride problems)
- They allow vehicles to meet at predefined transshipment locations.

In the following paragraphs, we review work related to the Load Exchange Strategy addressed in this thesis. Although our approach differs in all three aforementioned characteristics, there are several similarities with the work found in the literature. These are presented below.

Mitrovic-Minic and Laporte (2006), motivated by a large San Francisco courier company that allows transshipment of loads between vehicles, attempted to investigate the usefulness of having transshipment points in the service area. The authors propose a two-phase heuristic for the static case of a pickup and delivery problem with time windows and transshipment points (PDPTWT). Contrary to the standard PDPTW statement, in which an entire request must be served by the same vehicle, transshipment allows for a request to be served by two vehicles; one vehicle can collect the load at the pickup location and drop it at a transshipment point, and then another vehicle collects the load from the transshipment point and transfers it at the delivery location. Therefore, a request, that has to be transferred to another vehicle, is split in two requests; pickup location-transshipment point, and transshipment point-delivery location. Of course, the two requests have suitable time windows and precedence constraints.

The heuristic proposed includes a construction phase followed by an improvement phase. A random multi-start cheapest insertion procedure is considered for the construction phase. Several solutions are constructed using different random initial orderings of the requests and the best solution is used as the initial for the improvement phase. This solution is changed, based on request re-insertions, and the procedure stops if the solution can't be improved anymore. Transshipment decisions, whether a request will be split or not, are made during the both phases. Capacity constraints aren't included in the problem, since the load of each request is relatively small in the applications considered. The heuristic has solved randomly generated instances with up to 100 requests and 4 transshipment points. Computational results showed, both in clustered and uniformly distributed requests, that transshipment points can reduce the total distance traveled by vehicles. Especially for clustered instances, transshipment points seem to be very useful and their usefulness increases when the cluster size becomes smaller.

In the work of Cortes *et al.* (2010), the authors present a strict arc-based formulation of the static pickup and delivery problem with transfers (PDPT), i.e. allowing the option for passengers to transfer between vehicles. The transfer locations are considered to be fixed and known in advance. This mathematical approach includes special modeling of transfer locations as well as additional variables to identify certain customers and their interaction with vehicles at pickup, delivery and transfer points. They define a request, as a set of passengers (objects to be transported, such as people, freight etc.) traveling from the same origin to the same destination. It is also assumed that passengers of the same request can't be served by different vehicles. Time windows and capacity constraints are included in the problem. The mathematical formulation of PDPT is built by systematically adding variables and constraints. Furthermore, the authors provide an illustrative example, proving that there exist some configurations, in which the transfer option between vehicles can yield more efficient solutions than those obtained from the classical PDP without transfers. The proposed formulation is solved with an exact solution method based upon a branch-and-cut technique using Benders Decomposition (Benders, 1962); the latter is advantageous (especially in terms of running time) when compared against a straight branch-and-bound approach. The proposed method decomposes the set of constraints into pure integer and mixed constraints and then a branch-and-cut process is applied to the resulting pure integer problem, by using real variables and constraints related as cut generators. The authors solved very small instances with up to six customers and one transfer point located in the geographical center of the customer nodes. Since this exact method can handle only small instances, the authors provide some guidelines for further improvements. They postulate that transfer operations become more and more profitable in high demand instances and they expect metaheuristics, such as Tabu Search, to perform well under high demand conditions. Insights for more efficient set partitioning formulations (route-based) and algorithms to solve real-size problems are also provided.

Kerivin *et al.* (2008) consider a relaxation of the standard PDP where any load can be unloaded (fully or partially) at any intermediate stop (node), and picked up later by the same or another vehicle. They call this unloading/picking up process as a reload, and refer to the problem as the splittable pickup and delivery problem with reloads (SPDPR). A reload can be repeated several times for a customer request until it

reaches its destination. Also there is no constraint on vehicle routes, and each vehicle can visit a node of the network or link as many times as needed. The authors discuss the complexity of the problem and prove that it is a NP-hard problem. They also present two mixed-integer linear programming formulations (a multicommodity flow-based formulation and a metric constrained one) based on a space-time graph. Furthermore, they describe some valid inequalities for the problem, which may be added to strengthen the associated linear relaxations, along with separation routines. A branch-and-cut algorithm is developed for solving to optimality the two models, for small-size problems with up to 10 vertices and 15 demands. The main purpose of this paper is more to provide a basic frame for further research, and generate lower bounds for checking the efficiency of heuristics developed for the problem.

In his PhD dissertation, Nadarajah (2008) provides a collaborative framework for LTL (less-than-truckload) carriers in an urban region. Collaborative logistics (CL) is a recent business model designed to eliminate transportation inefficiencies. The author models this collaboration framework as a variant of the VRP which is referred to as the COLaborative VRPTW (COL-VRPTW). The problem arises in urban areas where the routes of different carriers overlap, and the exploitation of goods transshipment between collaborating carriers can be mutually beneficial. In the COL-VRPTW, customer requests don't concern pickup and delivery of goods. Goods are loaded at the depot and are destined to be delivered to customer locations, with the allowance of exchanging goods with other carriers at transshipment points. Transshipment is allowed just between two vehicles belonging to different carriers. Simple examples are presented in the dissertation explaining the benefits of carrier collaboration.

The author proposes a two stage collaborative framework, which can be used between LTL carriers. The first stage involves exchange of (partial) loads between carriers at logistics platforms located at the entry to the city (solution of simple VRP-like problems), while in the second stage trucks make such exchanges during local delivery (solution of COL-VRPTW). To solve the mathematical problem that results from the two stage collaborative framework, a novel integrated three-phase heuristic is presented. The first phase uses either a modified tabu search, or a guided local search, to solve the vehicle routing problems with time windows. The preceding methods use a constraint-programming engine for feasibility checks and reduction of the search space, and then the solution is used to create the COL-VRPTW instances.

Given a COL-VRPTW instance, the second phase of the algorithmic framework uses an adaptive quadtree search to create clusters of customers that can be considered for collaborative exchange of partial loads at transshipment points. The site of the transshipment point is also located in the cluster by this method. In the last phase, an integrated greedy local search method is used to construct collaborative routes, using three new transshipment-specific moves for neighborhood definition. An optimization module is utilized within local search to combine multiple moves at each iteration, thereby taking efficient advantage of information from neighborhood exploration.

Extensive computational tests were performed on random data sets for problems of realistic size, and sensitivity analysis was conducted on key parameters. Overall results showed that the collaboration leads to 12% and 15% improvements in route distance and time, respectively.

In addition to the work presented earlier, several other researchers introduce the idea of transshipment operations in terms of operational strategies that could potentially support efficient transfer operations. Interesting information on various issues related to the transfer operations in vehicle routing can be found in Aldaihani and Dessouky (2003), Liaw *et al.* (1996), Hickman and Blume (2000), Mues *et al.* (2005), Nakao *et al.* (2008), Shang *et al.* (1996) etc.

Table 2.1 summarizes the research related to the idea of transshipment operations in a vehicle routing setting. The solution methodologies used are also presented, along with several proposals for future research, as provided by the authors.

Table 2.1: Some problems addressed in the literature considering the transfer operation between vehicles

Authors	Problem	Method	Research Extensions
Mitrovic-Minic and Laporte (2006)	PDPTW and Transshipment points	Heuristic including two phases: construction & insertion	i) dynamic case of the problem ii) definition of conditions under which a PDPT solution can outperform a PDP solution iii) algorithms which can solve large instances iv) polyhedral study for faster solutions v) use of a set partitioning formulation
Cortes *et al.* (2010)	PDP with Transfers	Exact method based upon a branch-and-cut technique using Benders Decomposition	
Kerivin *et al.* (2008)	Splittable PDP with reloads	Branch-and-cut algorithm	i) further valid inequalities to strengthen the formulation ii) consider a column generation approach based on an arc-path formulation iii) model without time indexation
Nadarajah (2008)	Collaborative VRPTW	Integrated three-phase heuristic	i) synergies between shipper and carrier collaboration ii) more effective algorithmic approaches iii) incorporated inventory replenishment issues
Aldaihani and Dessouky (2003)	Dial-a-ride as a hybrid system consisting of both on-demand vehicles and fixed route lines	Three-phase heuristic (Identification of the candidate path set- Insertion-Improvement) and Tabu Search	i) Adaptation of the Improvement heuristic to a real-time environment
Hickman and Blume (2000)	Integrated transit service	Two stage scheduling heuristic that includes both the passenger and operator objectives	
Mues *et al.* (2005)	Intermodal transportation problem	Column Generation based approach	i) implementation of the sketched solution method

2.5 Research Gaps and Contribution of the Thesis

Several authors have introduced the idea of load transshipment between vehicles in order to better handle the customer requests. However, the related papers concern mostly the static case of many-to-many problems (pickup and delivery problem and dial-a-ride systems), where the transshipment operation can be performed on predefined locations.

In the present thesis, we apply a novel approach of the transshipment strategy, called hereafter *Load Exchange Strategy (LES)*. This was originally proposed by the on-going research of Ninikas and Minis (2011) to be implemented to a dynamic vehicle routing problem, referred to as the *Vehicle Routing Problem with Dynamic Pickups (VRPDP)*. In this dynamic setting, each vehicle is assigned to serve a predefined customer set, consisting of delivery orders known in advance. As the working plan unfolds, however, customer orders are received through a call center, requesting on-site pickup within the current period of operations. The load exchange strategy allows for vehicles to meet in real-time and exchange some delivery orders if this is profitable. The location of the exchange operations may be either dynamic (i.e. most favorable locations) or static (predefined). In our approach, we differentiate from the related work to-date, based on the following three points:

- We apply LES on a dynamic setting, where customer requests arrive dynamically over time
- We apply LES on many-to-one cases, where transshipment operations may not have that much discrete savings
- We examine and identify cases where dynamic locations should be also considered for the exchange operation, in addition to the fixed location case.

Extensive experimental analysis of LES under various dynamic settings showed that in the majority of cases tested, allowing transfers results in solutions of improved quality.

LES is a complex strategy to implement. Thus, the current thesis attempts to investigate several key aspects related to this strategy. Of course, further research is required to understand the extensions of LES, the problems in which it can be

implemented and the circumstances under which it provides solutions of superior quality. Finally, due to the fact that the method proposed here provides solutions for instances with a fleet of only two vehicles, research may also focus on how a load exchange scheme can be implemented for a large fleet.

CHAPTER 3 PROBLEM DESCRIPTION

The main scope of this Chapter is to review the *Vehicle Routing Problem with Dynamic Pickups (VRPDP)* and set the foundation for the solution methodology of *Load Exchange Strategy (LES)* that's further discussed in Chapter 4. This Chapter mostly overviews work developed in the context of the Ph.D. current research of G. Ninikas in the DeOPSys lab. Section 3.1 presents basic characteristics and assumptions of VRPDP, as well as the static problem to be considered at each replanning timestamp. Section 3.2 provides an overview of the LES framework, while Section 3.3 addresses the decision components that have to be defined for the application of the LES. Finally, Section 3.4 presents the cases/characteristics considered in the current thesis among these possible alternatives.

3.1 THE VEHICLE ROUTING PROBLEM WITH DYNAMIC PICKUPS

3.1.1 INTRODUCTION

The problem addressed is a variant of the *DVRP* described in Chapter 2 and is referred to as the *Vehicle Routing Problem with Dynamic Pickups (VRPDP)*. Due to its dynamic nature, not all information is known *a-priori* (during the original planning phase) and some information is revealed during the execution of the designed plan. Dynamic information refers to the occurrence of dynamic events, e.g. the arrival of new customers requiring service during plan execution. Each time a new dynamic request is received, the current fleet *en route* is located in different positions and several requests may have already been served. Thus, the routing plan has to be updated in order to incorporate the up-to-date information, which usually implies that the originally designed routes should be re-optimized.

The *VRPDP* has been motivated by the courier industry, where new customer orders are received continuously over time (Ninikas *et al*., 2011 and Ninikas and Minis, 2011). In this setting, distribution vehicles depart loaded early in the morning from a Local Service Point (LSP) to perform deliveries known *a-priori*; typically, each delivery vehicle serves a certain geographical area. However, the LSP dispatcher typically knows in advance only a subset of the tasks. The rest request on-site pickup within the current period of operations, and arrive in the system in a real-time manner through a call center. These pickup orders have to be collected and returned to the depot for further processing. As mentioned above, this problem has been formalized by Ninikas and Minis (2011) and is overviewed in Section 3.1.2. For simplicity and comparison purposes with the solution method of the current thesis, we refer hereafter to solution approach of the related work to as *Insertion Method*.

3.1.2 PROBLEM OVERVIEW

Consider a set of homogeneous vehicles, originating from a local distribution hub (depot). The vehicles are tasked to serve delivery orders known prior to the start of operations, referred to as *static orders*. At time prior to the beginning of the shift, a static VRP is solved which assigns customer orders and their service sequence to vehicles, referred to as *planned routes* (Figure 3.1a). As the working plan unfolds,

however, pickup requests are received and have to be served within the current service shift (Figure 3.1b). These arriving requests will be referred as *Dynamic Requests (DRs)*. Each order (static or dynamic) may be associated with a time window within which the order must be served. Dynamic requests can be also seen as flexible orders, since they can be served by each vehicle (either *en route* or located at the depot), while static orders are inflexible, since they have to be served by the vehicle originally assigned to it.

At a given time instance t_1 *(replanning timestamp)*, the newly received information (DRs) need to be incorporated in the partially executed plan. Therefore, a "static" problem *(replanning problem)* has to be solved at this replanning timestamp, considering all the information known up to this point in time (i.e. information concerning the static orders that haven't been served yet and the newly received DRs). A solution method may be applied, such as the *Insertion Method*, in order to allocate the dynamically revealed orders to the most appropriate vehicles (Figure 3.1c). Figure 3.1d represents the new solution obtained at replanning timestamp t1, when applying this method.

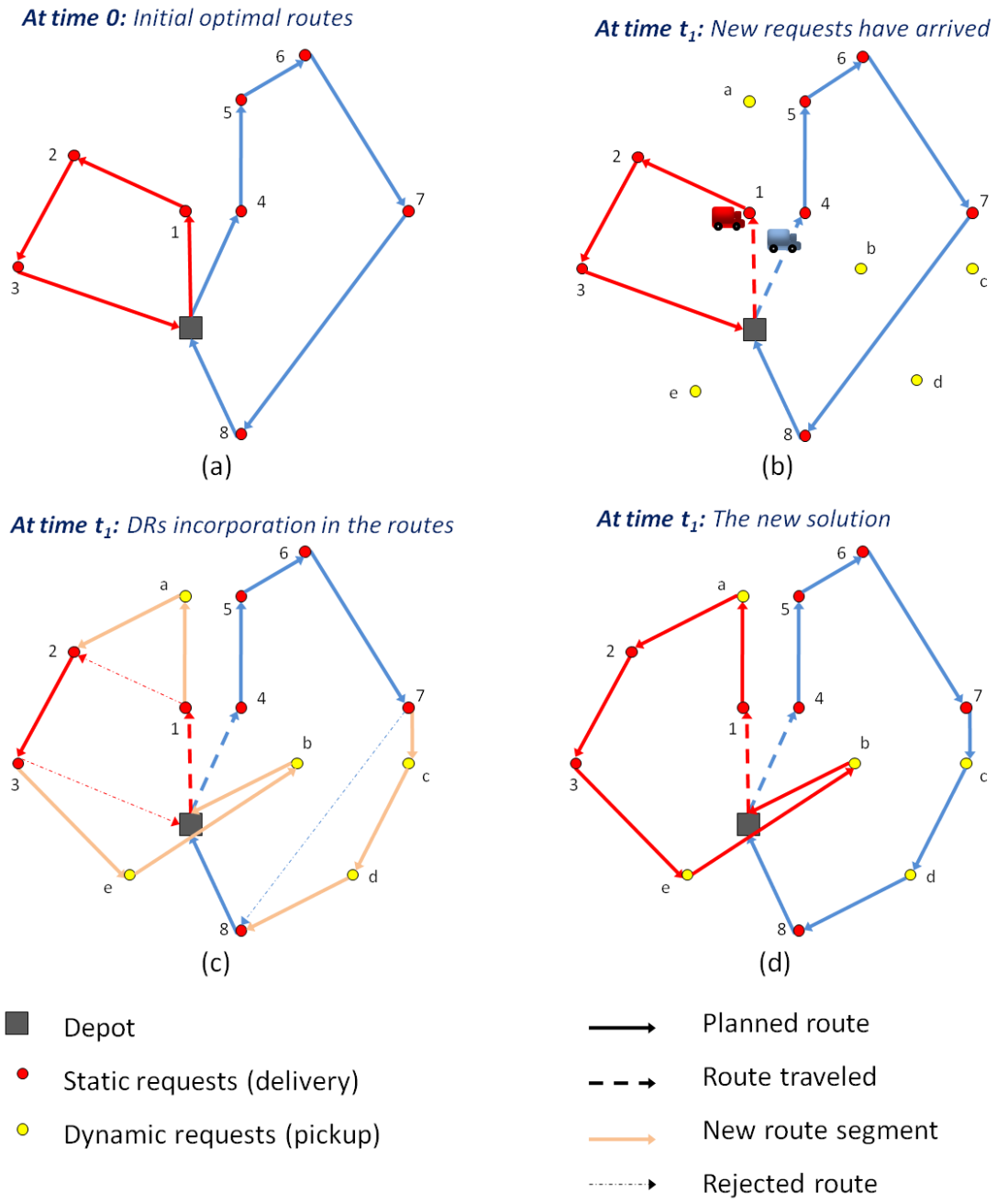


Figure 3.1: Illustrative example of the VRPDP

The objective of the *VRPDP* is to find a set of vehicle routes that minimizes the total routing cost and the number of unserved DRs throughout the available shift, while satisfying the following constraints:

- Each vehicle must start and end at the depot
- All static (delivery) orders should be served, while there is no such constraint for DRs
- Each order may be served at most once by a single vehicle

- Each order has to be served within a certain time window selected by the customer. In case a vehicle arrives at a request before its earliest service time (*early time window*), it has to wait, but can't serve the request later than the latest service time (*late time window*)
- Static (delivery) orders cannot be reassigned among vehicles, i.e. the orders originally assigned to a vehicle, must be served only by this vehicle. Of course, the sequence of servicing delivery orders by a certain vehicle may be changed, if this favors the objective function. (In section 3.2 we relax this constraint and introduce a new policy that allows transferring of delivery orders between vehicles if favorable)
- All vehicle routes have a duration constraint equal to the length of the planning horizon

Figure 3.2 summarizes the entire process followed within an available shift (t=0 to Tmax) for solving the VRPDP (see Ninikas and Minis, 2011). Initially, a static VRP is solved, assigning the static (delivery) orders to the fleet of vehicles and producing the initial routes to execute. At each replanning timestamp (specified by the dispatcher) the DRs occurred are assigned to the most appropriate vehicles through the Insertion Method. Of course, orders arriving after the last replanning timestamp (t_n) are either rejected or served during the next service period.

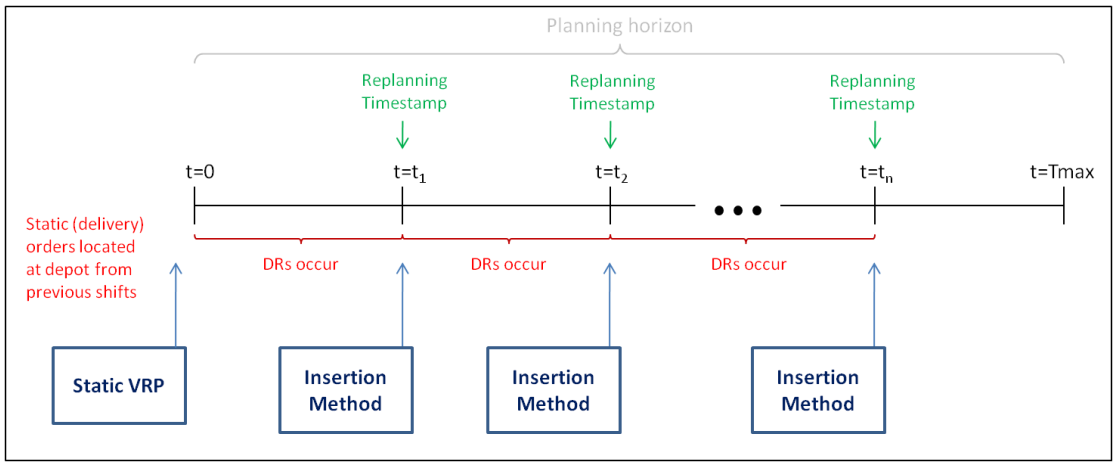


Figure 3.2: VRPDP solution process during the available planning horizon

3.1.3 Assumptions of the generic problem

The following characteristics/assumptions concern the operating scenarios considered for the VRPDP (as taken from the work of Ninikas and Minis, 2011):

i. The current status of the logistics operations (i.e. current location of each vehicle of the fleet and time for service, remaining unserved customers, etc.) is assumed to be known at any time instance. In practice, this is achieved by employing appropriate fleet monitoring systems.

ii. A vehicle commits to travel at the latest possible time. For example, if a vehicle is planned to arrive at a customer prior to the opening of its time window, the vehicle has to wait at the location of the previously served customer. This assumption facilitates replanning changes in case appropriate new orders arrive to the system.

iii. The route is updated only at customer locations, i.e. the problem considered does not allow diversion (Ichoua *et al.*, 2000). Once a vehicle has left its previous service location and is *en route* to its next destination, the vehicle cannot be diverted.

iv. A number of vehicles may be available at the depot (as "backup" vehicles) at the beginning of the planning horizon, ready to be dispatched when necessary for the service of DRs. The trigger of this action is when vehicles *en route* can't serve DRs, or when this action favors the objective function.

v. We assume that the load of each request (letter or small parcel) is relatively small, thus, we don't consider capacity constraints during the solution of the problem at hand.

3.2 LOAD EXCHANGE STRATEGY FOR THE VRPDP

In this Section, we introduce the framework of the Load Exchange Strategy. As already mentioned, each static (delivery) order is assigned to a specific vehicle at the beginning of the planning horizon, and is prevented to be reassigned to other vehicles at a next replanning cycle. Oftentimes, this intrinsic constraint, combined with the continuous arrival of new requests, may reduce the plan's quality or, even worse it

may lead to inability of servicing a number of requests. The main idea of LES, explored in the present thesis, is to relax this constraint by transferring delivery orders among vehicles. This operation can be performed by allowing vehicles to meet *en route* at appropriate locations and exchange loads (actually, orders), if this favors the objective function.

Figure 3.3 illustrates the LES strategy, using the example given in Figure 3.1.

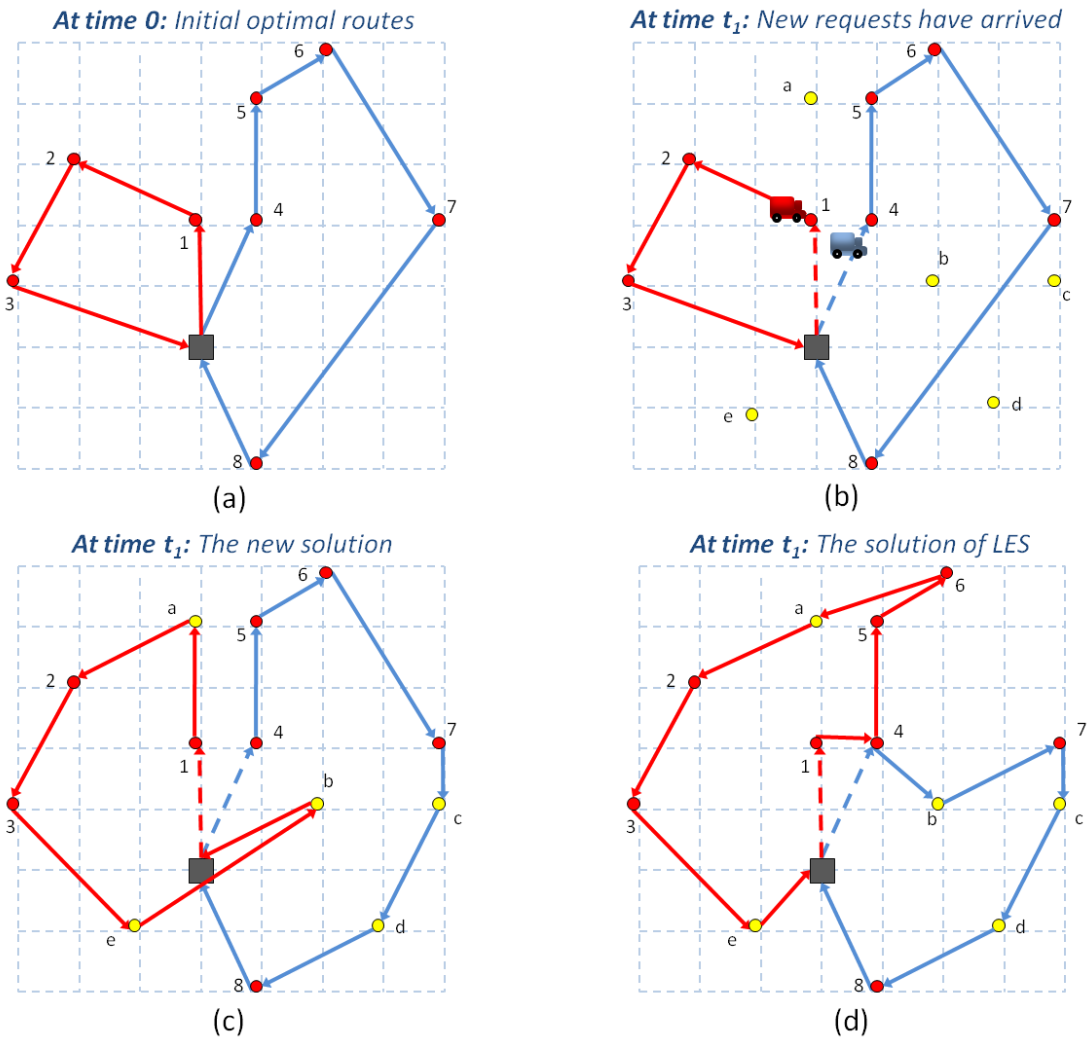


Figure 3.3: Example of Load Exchange Strategy

Consider the case of Figure 3.3a where two (2) optimal routes have been designed at time $t = 0$. Customer locations of the example have been placed on the nodes of a grid, where all arcs equal to one unit of distance. The length of the available planning horizon (Tmax) is considered to be 18 units of time and a vehicle travels 1 unit of distance in one unit of time. At replanning timestamp $t_1 = 2\,units$, five DRs have arrived (Figure 3.3b). The solution under the Insertion Method (Fig. 3.3c) mentioned

above returns a total cost of 33.8 units. Figure 3.3d provides a solution that may be obtained with LES, assuming that the vehicles are able to meet and transfer delivery orders between each other at any unserved customer location. As shown in the Figure, the (red) vehicle initially assigned to serve delivery orders $\{1, 2, 3\}$, after serving customer 1, travels to the location of customer 4 in order to meet the other vehicle (blue) and take the load of orders 5 and 6. The total distance traveled (cost) under this solution is 30.96, improved by 8.4% compared to the one obtained with the Insertion Method.

The main reasoning behind allowing transshipment lies in the idea of restraining vehicles to their service region, in order to serve potentially new DRs that may occur in their region. A significant advantage of this novel strategy is the ability of assigning each order (either static or dynamic) to any vehicle without limiting the replanning options. Therefore, this exchange operation allows the dispatcher to better manage the fleet of vehicles and better re-distribute the workload as needed in a real-time fashion. However, a drawback of such a strategy is possible delays due to the transfer operation. Such delays may be due to a) the fact that, in most cases, vehicles don't arrive at the exchange location simultaneously, and b) the time required for the vehicles to load/unload the appropriate load (orders).

Furthermore, it should be noted that, LES is not a panacea, since it doesn't always provide a "better" solution. Therefore, LES is a policy that can be applied as an extension to the conventional solution of the VRPDP (Insertion Method) at a replanning timestamp. In case LES doesn't return a feasible solution, we may adopt the one returned from the Insertion Method.

Additionally, the following characteristics/assumptions are assumed regarding the operating scenarios considered for the problems solved with LES:

- All constraints of the VRPDP, mentioned in 3.1.2, must be satisfied (except the one preventing reassignment of the static orders between vehicles).
- The vehicles should meet prior to the service of any exchanged static (delivery) order.

3.3 Decision Components and Alternative Policies for the LES

LES is a very complicated and multidimensional strategy, since many options are available to the dispatcher. When dealing with such a strategy, the following questions arise and should be defined:

1) Implementation time instances to apply LES: *When* should one apply the LES? This concerns the time instance or the conditions under which the implementation of the strategy could provide an efficient solution.
2) Meeting combinations: *Who* is going to meet and how many times? This parameter determines which vehicles will be examined for load exchange; i.e. if a vehicle is able to meet one or more than one other vehicles *en route*.
3) Exchange locations: *Where* is it allowed for vehicles to exchange loads? This parameter refers to the locations where vehicles are allowed to meet and exchange their loads.

Figure 3.4 summarizes the aforementioned parameters and potential alternative decision policies. In the following subsections, each parameter is thoroughly described.

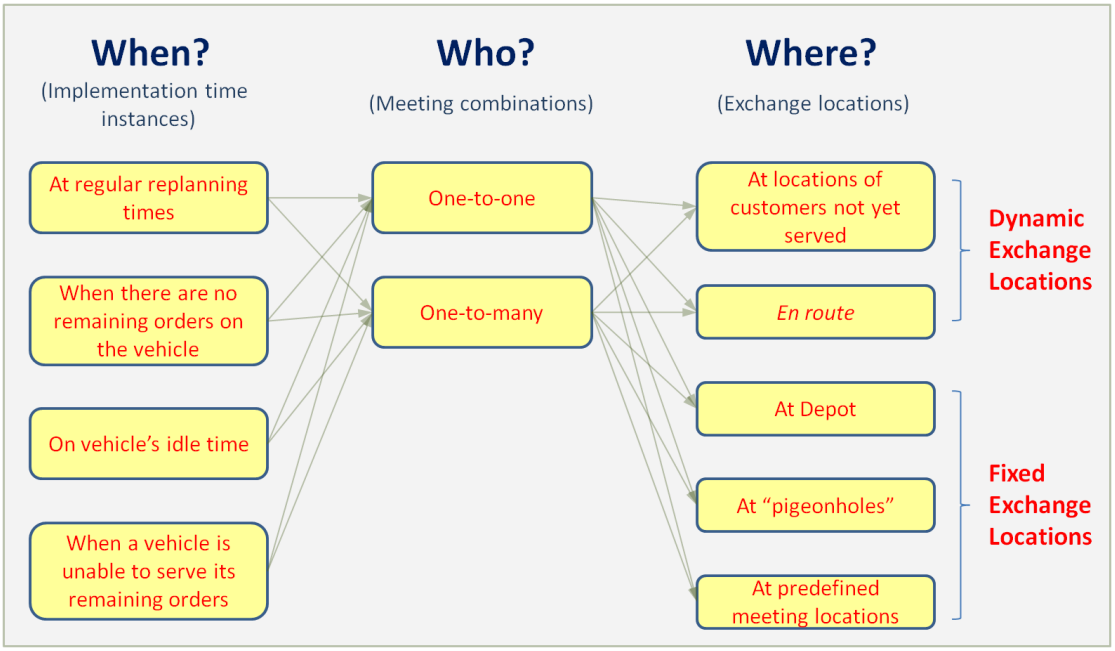


Figure 3.4: Components and alternative decision policies

3.3.1 Implementation Time Instances

As it has already been described, the most typical implementation time instance for LES is during each replanning cycle, i.e. at the replanning timestamp, where the current routing plan has to be updated in order to incorporate the newly received information. The application of LES at each replanning cycle may not yield significant results each time, but possibly during the first replanning cycles, where there are still many options available and only a limited portion of the routing plan has been executed.

When there are no remaining orders on the vehicle

An alternative approach regarding to the time instance to implement LES can be when a vehicle has no remaining orders in its planned route and it is ready to return to the depot. Consider the example of Figure 3.5a. At time $t = X$ the red vehicle has no remaining customers to serve, while seven (7) orders are still assigned to the blue vehicle. In this case, when applying LES (Figure 3.5b) the blue vehicle meets and exchanges several orders with the red vehicle at customer location A. One of the main advantages of LES in that case, is the balanced workload of the solution between the vehicles. This way, future DRs that may arrive can be better allocated among the vehicles, since more options are now available.

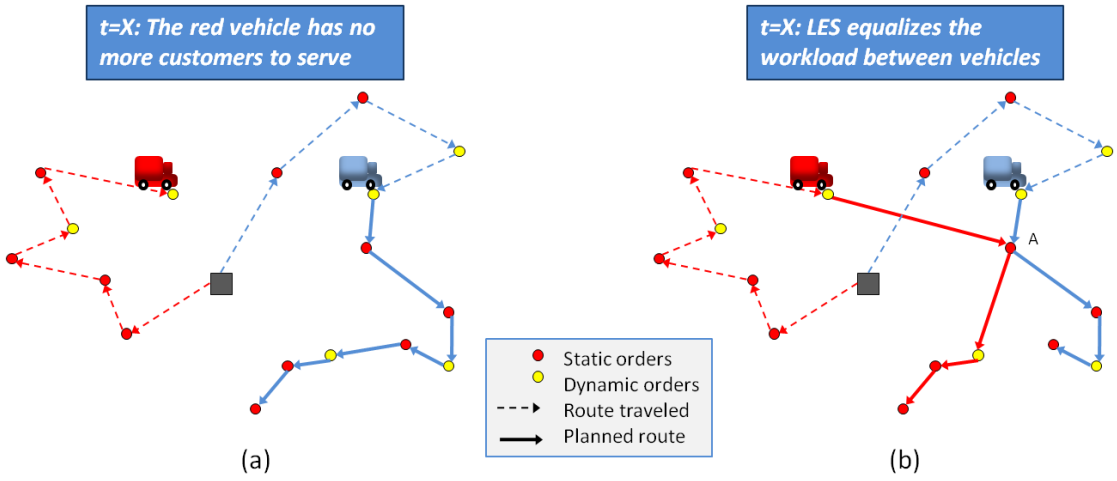


Figure 3.5: Example of LES when there are no remaining customers

The same concept can be also applied for cases where a vehicle is idle, e.g. in cases where a vehicle is forced to wait at a customer location for a significant amount of time, until the time window of the next customer to be served opens.

When a vehicle is unable to serve its remaining orders

Sometimes, during the execution of the designed plan, dynamic events such as traffic congestion, road blocks (due to street markets, demonstrations, etc), extreme weather conditions, vehicle breakdown, etc., can significantly deteriorate the distribution plan. LES can also be applied in those cases in order to improve the total service of the customers and balance the remaining workload of the available vehicles.

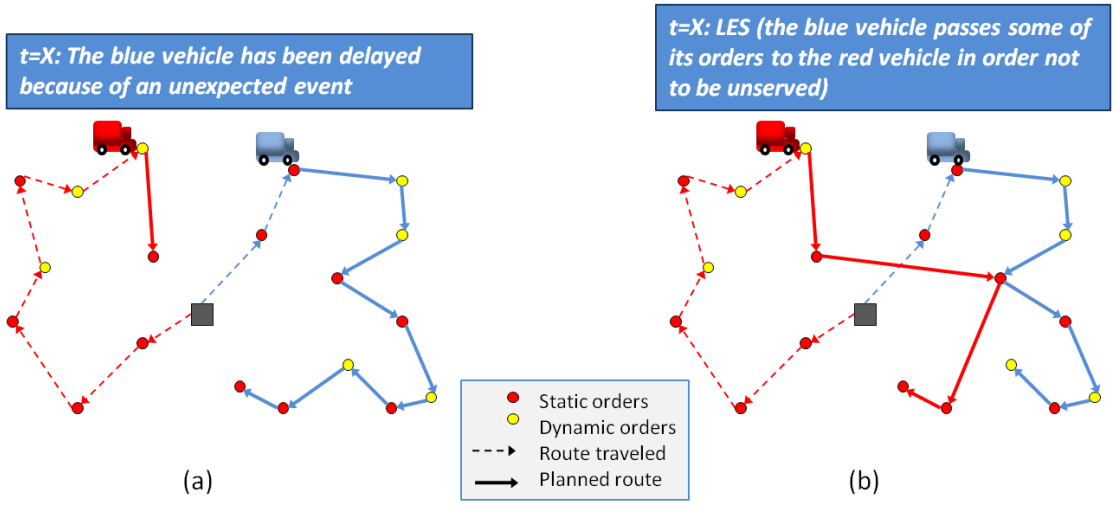


Figure 3.6: Example of LES in case vehicle is unable to serve remaining orders due to unforeseen events

Assume that during a distribution operation, a vehicle is facing significant delays due to one of the above reasons (e.g. traffic congestion). If the delay is such that the original distribution plan can't be completed within the scheduled timeframe, then the original plan must be redesigned. This scenario is presented in Figure 3.6a, where the blue vehicle has been delayed and doesn't have enough time to serve the orders assigned to it. Therefore, at time $t = X$, LES is used in order to re-distribute the remaining unserved orders among all available vehicles (Figure 3.6b). It should be noted here, that a vehicle located at the depot could have also been dispatched in order to serve several orders of the blue vehicle.

A relevant interesting case is when a vehicle *en route* breaks down. Typically, in this case, the delivery of goods of the immobilized vehicle is suspended for the next day. However, as an alternative, the dispatcher may redesign the routes of the remaining active vehicles through LES, to reduce the impact of the cancellation of the delivery of goods to customers of the breakdown vehicle. Figure 3.7 illustrates this scenario.

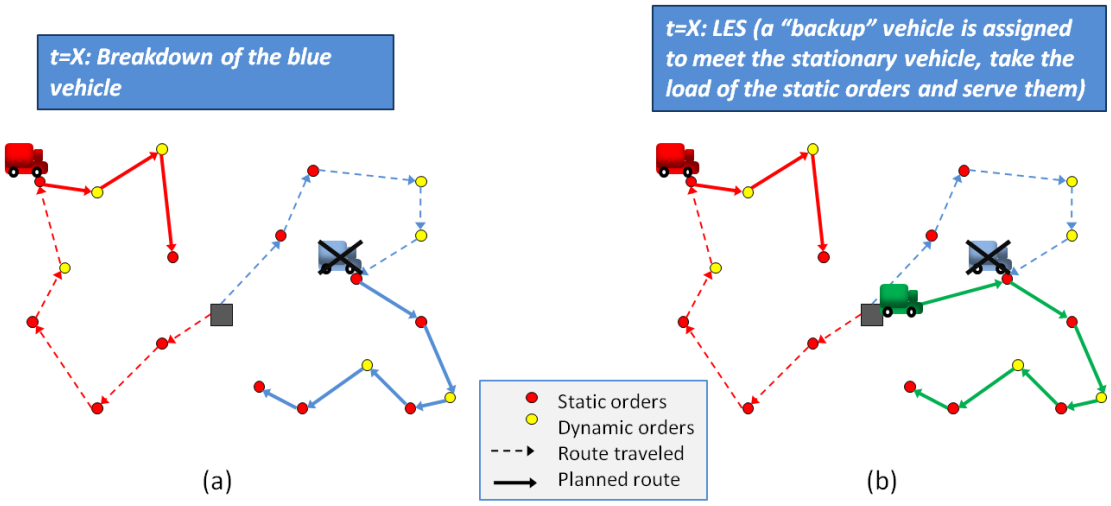


Figure 3.7: Example of LES in case of a breakdown vehicle

3.3.2 MEETING COMBINATIONS

Another important parameter of LES comprises the allowable combinations of vehicles that should participate in the exchange operation. For this component, we investigate two (2) alternative decision policies: i) one-to-one, and ii) one-to-many.

One-to-one

For this meeting combination policy, we consider that exchange operations can be performed only per couple of vehicles. Specifically, a vehicle, either *en route* or located at depot ("backup"), may serve only delivery requests originally assigned to a different vehicle *en route* and not from other vehicles.

An illustrative example for this policy is presented in Figure 3.8.

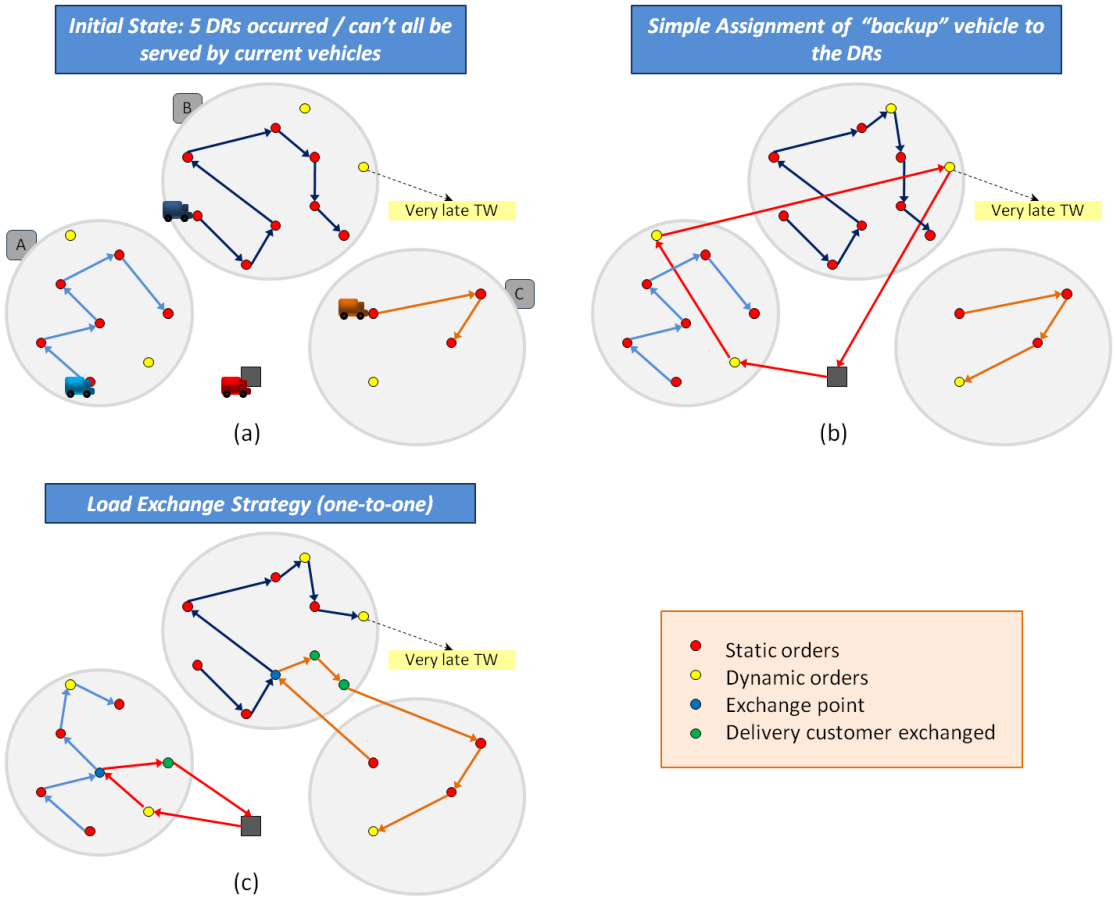


Figure 3.8: Example of the one-to-one policy

One-to-many

For this policy, we assume that a vehicle, either *en route* or located at depot, is allowed to serve a number of orders originally assigned to more than one vehicles *en route*. This, of course, means that the vehicle has to meet more than one vehicles at different locations. Obviously, this case is more complicated. Below we present an example of a "backup" vehicle receiving several static (delivery) orders from more than one vehicles *en route*. We believe that this is the most useful application of the one-to-many policy. Of course, such a policy could also be applied for a vehicle *en route*, especially when it has finished servicing the orders assigned to it.

Consider the initial state of the Figure 3.9a, where 3 vehicles are executing their planned routes while a "backup" vehicle is located at the depot in order to be dispatched when necessary for servicing DRs. In this initial state 5 DRs have arrived, which can't be served by current vehicles *en route*. Figure 3.9b presents the solution, resulting by the simple assignment of the "backup" vehicle to serve the DRs.

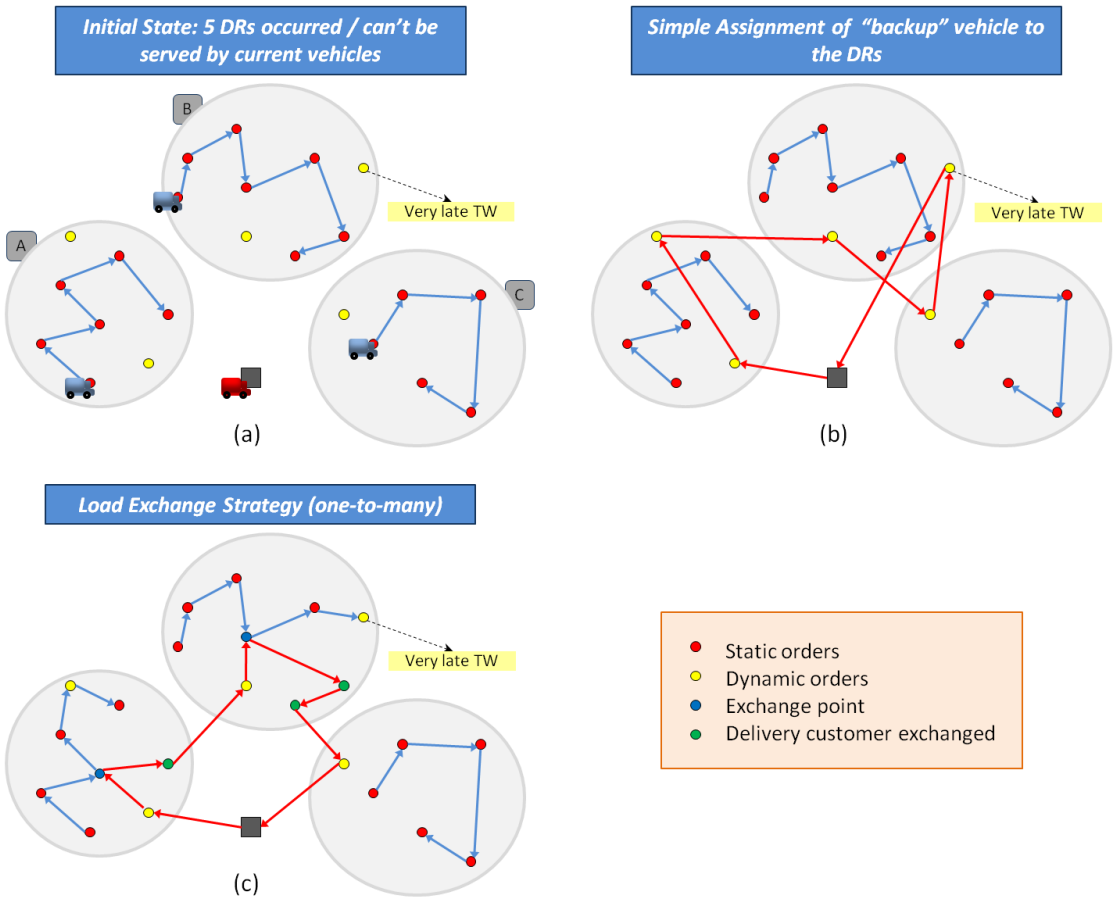


Figure 3.9: Example of the one-to-many policy

However, an one-to-many policy may provide a more efficient solution, as illustrated by Figure 3.9c. In the latter case a "backup" vehicle traverses all service regions, regardless the presence of any other vehicles. During its passing from a specific service region, we check whether the "backup" vehicle may meet the regional vehicle and serve some delivery customers on its behalf. This will discharge the latter vehicle and provide some extra capacity (in terms of time) and potential to serve dynamic orders that may appear in this region.

3.3.3 Exchange Locations

Depending on the characteristics of the business scenarios, the distribution area and the number of customers requiring service, a dispatcher may define the exchange locations that are relevant to the company's operations. An exchange location may be either known prior to the start of operations (*fixed exchange location*), or dynamically revealed as vehicles execute their planned routes (*dynamic exchange location*). An important advantage of the former case lies in the capability of vehicles not having to

be at a certain location at the same time. This means that a vehicle may travel at any moment to this fixed location and drop loads corresponding to customers that are now assigned to a different vehicle. Subsequently, the corresponding vehicle may travel and pickup those loads at any favorable point in time. In the following paragraphs, several alternative options are presented for both static and dynamic exchange locations.

Dynamic Exchange Locations

A) At locations of customers not yet served

An important and effective way for the achievement of load exchange is allowing vehicles to meet at any unserved customer location (i.e. locations of all static and dynamic orders not served until the replanning phase). Doing so, a great number of possible meeting locations is provided, thus, increasing the possibility of one of them to be beneficial for the exchange operation. Another major advantage of this policy is the slight alteration of the current routing plan of each vehicle, since only one of the vehicles is forced to deviate from its scheduled tasks in order to meet the other vehicle.

B) En-route

This case considers that all vehicles may meet at any moment in time, at any place. Of course, this is not so applicable in practice, since a number of events may prevent the exchange operation (e.g. lack of parking, adverse weather conditions). Furthermore, such a policy may be complex to implement.

Fixed Exchange Locations

A) Depot

A convenient implementation of LES is to consider as exchange location the depot. There are cases, in which exchange of loads through the depot may provide very efficient solutions. The main advantage of this policy can be seen in cases, in which load from only one vehicle has to be transshipped, since in this case vehicles don’t have to meet at the same time instant. However, in this case, both vehicles have to be diverted and return to depot, which in most practical cases is quite far from the responsible service region of the vehicle.

B) Transshipment points ("Pigeonholes")

This policy, concerns facilities (*"pigeonholes"*), established and owned by the company, where a vehicle is able to discharge load that may be later picked-up by a different vehicle. These transfer points offer the same features with the depot, but moreover, may be located everywhere in the distribution area. Additional investigation may be needed in order to optimally determine and allocate those pigeonhole(s).

C) Predefined meeting locations

Another effective exchange location is a place with enough space (e.g. parking lot, square etc.) to allow vehicles to park for a short time.

Figure 3.10 summarizes all of the aforementioned alternative policies that can be used as exchange locations. Each policy is followed by a comprehensive illustrative example.

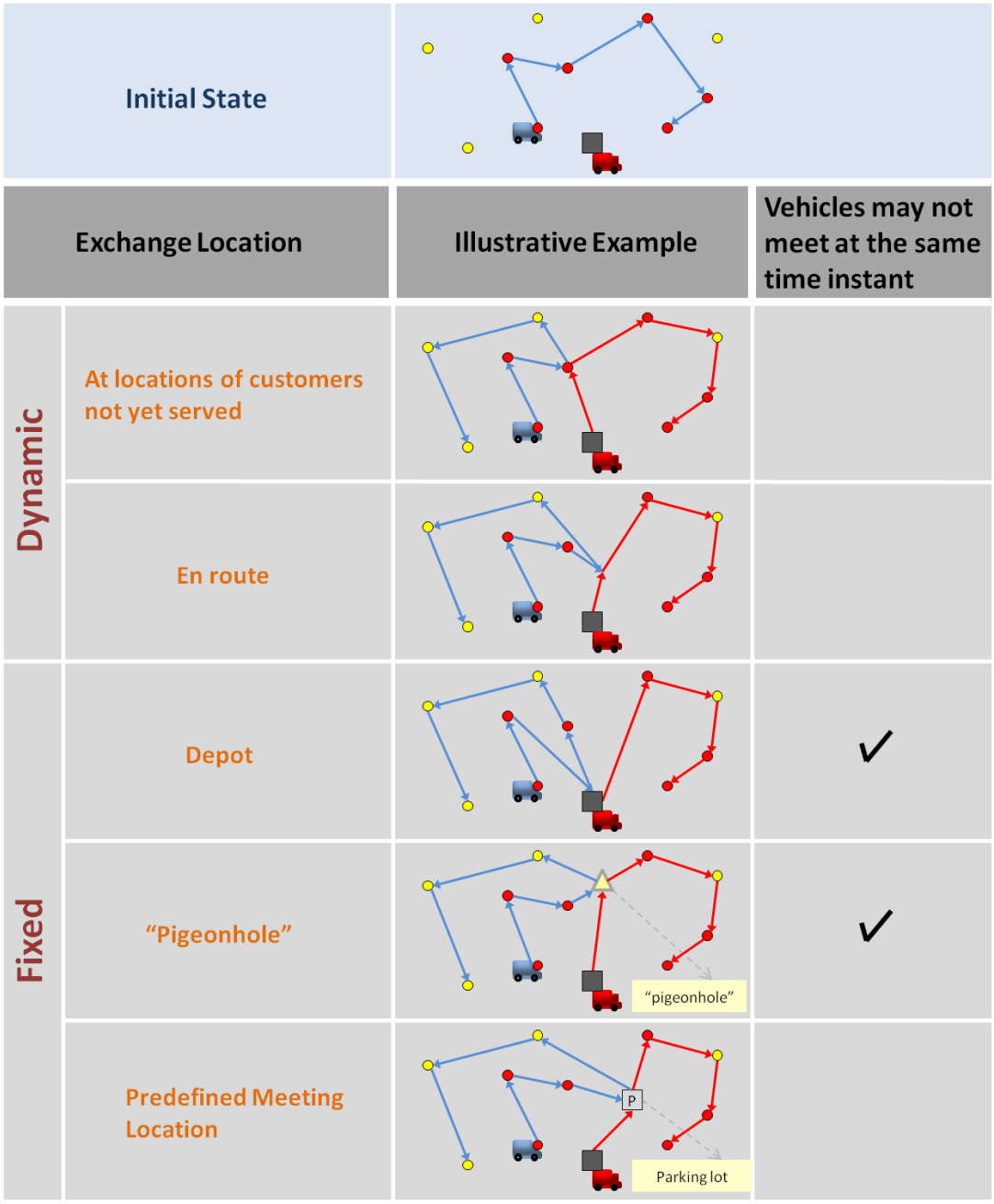


Figure 3.10: Potential exchange location decision policies

3.4 Decision Policies Addressed in the Current Thesis

In this thesis, a relatively simple problem is solved for a fleet of two vehicles in order to investigate the potential of LES. For more than two vehicles (not considered in this thesis), the problem's complexity increases even more. Furthermore, in reference to the parameters discussed above, the present thesis assumes the following:

- We used the replanning timestamp to be the implementation time instance where LES will be applied. LES will be implemented combined with a conventional solution approach, as described in Section 4.
- Since we deal with a fleet of two vehicles, a one-to-one policy is selected; the two vehicles are only allowed to meet once to exchange loads.
- For the exchange locations parameter, we investigate the option of allowing vehicles to meet on-site at customer locations. Furthermore, we consider a "pigeonhole", where a vehicle may drop loads from several customers in order to be picked up later by a different vehicle. This location is set-up on the center of gravity of all customer locations. Finally, we investigated the case in which the depot is used as exchange location for both cases (dynamic and fixed).

Figure 3.11 illustrates the decision policies selected for the case considered in this thesis (denoted with bold lines), in comparison to the decision tree that was presented in Section 3.3. Those policies will be used as the basis for the solution method described in the following Chapter.

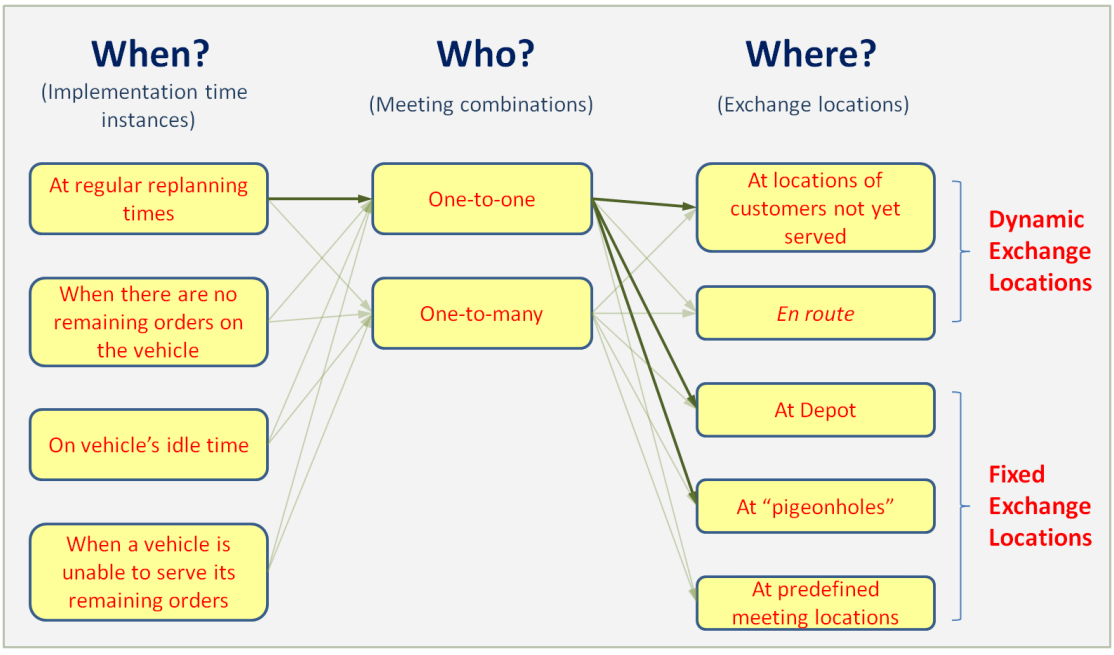


Figure 3.11: Decision policies considered from the entire LES decision tree

CHAPTER 4 SOLUTION METHODOLOGY

In this chapter we describe the solution methodology for the Load Exchange Strategy which is implemented in combination with a typical solution method for the VRPDP. This latter approach for the solution of the VRPDP, referred to as Insertion Method, is based on a Column Generation framework and doesn't allow re-assignment of delivery orders among vehicles. The solution approach of this algorithm can be found in Ninikas and Minis (2011). The solution methodology of this thesis is based on a heuristic approach, whereas exact algorithm features are employed for certain solution steps and for applicable cases, i.e. limited scale problems.

4.1 Assumptions and Overview of the Solution Methodology

Recall that the following assumptions are considered for the operating scenario examined in the current thesis:

- There are only two (2) identical vehicles available for execution. For that reason, by default, the one-to-one meeting policy is used.
- Replanning and, consequently, the application of LES is performed only at the replanning timestamp (where a number of pickup requests have been received).
- Vehicles are able to meet either on-site of any not yet served customer (dynamic exchange location), or at a predefined exchange location ("pigeonhole"). The depot can be also used as an exchange location point for both cases, providing an alternative location for the vehicles, if it's profitable.

The proposed solution approach is a three-phase algorithm, which is applied after the problem has been solved with the insertion approach and it is implemented only if the solution returned is better than the one obtained by the insertion approach. The phases of the algorithm are as follows:

Phase I. Routing: In this phase, a VRP-like problem is solved. This problem considers a network with all unserved customer requests up to the replanning timestamp. All requests are assumed to be homogeneous, i.e. no pickup or delivery order type discrimination is considered. This relaxation allows both vehicles to be able to serve each one of those orders. For the solution of this phase, both an exact and a heuristic method may be employed, based on the problem scale.

Phase II. Meeting: During this phase, a process that seeks the best available meeting location (if any) is employed. Several alternative algorithms may be used for this phase, as will be discussed later on.

Phase III. Post-Optimization: Finally, an optimization procedure is applied to the solution of the two previous phases that seeks to further improve the solution currently obtained.

Figure 4.1 overviews our 3-phase solution approach. Each phase is thoroughly discussed in the next sections.

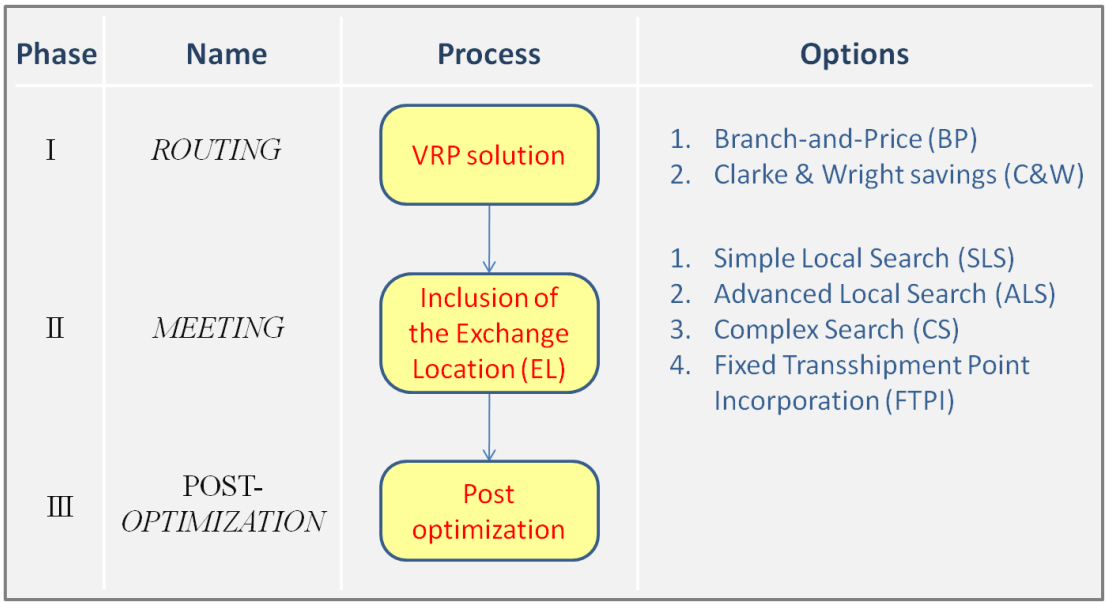


Figure 4.1: Overview of the solution methodology

As already mentioned, several alternative decision policies are investigated regarding the exchange location, i.e. for both dynamic and fixed exchange locations. For that reason, we distinguish the solution approach for each one of those categories. Initially, Section 4.2, discusses how the algorithm is applied for the general case of dynamic exchange location (meeting of the vehicles at customer locations), whereas the modifications and alterations of the solution method for the fixed exchange location policy are presented in Section 4.3.

4.2 Dynamic Exchange Location

We consider dynamic exchange locations for the problem, i.e. points that are not determined prior to start of operations, but become known after the application of the algorithm at each replanning cycle. The candidate exchange points comprise all unserved customer locations that correspond to either delivery or pickup customer orders.

4.2.1 Phase I: VRP Solution *(Routing)*

The first phase of the method (*Routing*) comprises the re-assignment of orders among the available vehicles, without considering the limitation that delivery orders

originally assigned to a vehicle must be served by the same vehicle. This can be performed by assuming all customer orders of the same order type (no pickup or delivery type discrimination). To do so, a typical VRP is solved, allowing for all unserved requests to be served by any vehicle. At the beginning of the replanning cycle, each vehicle may be located:

- On-site at a customer location, already serving him
- On its way to a customer location
- At depot (in case a "backup" vehicle)

For the latter case, the earliest time that this vehicle is able to start service is set equal to the replanning timestamp. For the two former cases, the current vehicle's location is set to a customer's location, since no diversion is allowed. For that reason, assuming customer h as the current vehicle's location, the time value that a vehicle is able to start the distribution is set equal to $\max(a_h, w_h) + s_h$ where a_h represents the opening of the time window of customer h, w_h the time the vehicle is set to reach customer h, and s_h the service time spent.

In order to provide the appropriate information to the VRP and guide the solution algorithm to obtain the desirable results, we apply several modifications in the order information, as well as in the distance matrix. Those modifications consist of the following:

- The TW of the customer h, where $h \neq depot$, which corresponds to a current vehicle's location, is set to $[0,\ \max(a_h, w_h) + s_h]$.
- The time needed to travel from depot to customer h is set to $\max(a_h, w_h) + s_h$. The above two modifications are needed in order to force the vehicle to travel from the depot directly to this customer (since this is its current location).
- The time needed to travel from any other customer to customer h is set to infinity (since this customer may be reached only from depot).
- The value of the replanning timestamp is added to the times needed to travel from depot to any customer, apart from current vehicles location.

The VRP problem in this phase is solved through an exact method (Branch-and-Price) or a heuristic approach (Clarke & Wright savings), depending on the problem's size. Each one of the aforementioned methods is described below.

Branch-and-Price

In order to describe the Branch-and-Price solution method used for the solution of Phase I, we adopt the description of the method as stated in the work of Ninikas and Minis (2011). Branch-and-price (Barnhart *et al*., 1998; Desaulniers *et al*., 1998; Desrosiers and Lübbecke, 2005) consists of a column generation algorithm embedded within a branch-and-bound scheme. Column generation is used to compute lower bounds at each node of the branch-and-bound search tree, while branch-and-bound is used to obtain the optimal integer solution.

Column generation is regarded as one of the most promising methods to solve vehicle routing problems by finding "good" lower bounds of the cost minimization version of the problem. In this setting, a VRP is modeled as a set-partitioning problem, in which each variable is a column representing a feasible route; the objective is to find the best set of routes that satisfy all problem constraints. Since the explicit generation of all feasible routes (columns) is clearly impractical, a column generation framework is used, in which a restricted problem is solved based on a limited portion of possible, "good" routes. The latter are generated by solving a series of subproblems.

When solving the linear relaxation of this restricted problem, the most appropriate routes from a restricted set of available ones are selected, aiming to determine the routing plan with the minimum cost. The solution to this linear program is then used to determine if there are any routes not included in the formulation that can further reduce the objective function value. This is the column generation step. The values of the optimal dual variables provided by the restricted problem are incorporated as modified cost in the objective function of the subproblems (usually simpler optimization problems), which in turn provide promising new routes (i.e. routes with negative reduced cost) that should be included in the formulation. Then the linear relaxation of this expanded problem is resolved. This is performed iteratively until no other columns that can reduce the objective function value are found.

In general, as defined in Bramel and Simchi-Levi (2002), the column generation (CG) approach for solving the linear relaxation of a problem $\mathcal{H}$ can be described by the following steps:

Step 1. Generate an initial set of columns $\mathcal{R}'$, which is a subset of all feasible columns $\mathcal{R}$ of problem $\mathcal{H}$ ($\mathcal{R}'$ could be a partial set of all possible feasible routes)

Step 2. Solve the restricted problem $\mathcal{H}'$ (containing a portion of all variables of problem $\mathcal{H}$, i.e. only columns $\mathcal{R}'$) and obtain optimal primal variables, $\bar{y}$, and optimal dual variables $\bar{\pi}$

Step 3. Solve the column generating subproblem, i.e. somehow identify columns $r \in \mathcal{R}$ the inclusion of which in the basis further reduces the objective function value (i.e. satisfying $\bar{c}_r < 0$, which is a modified cost that incorporates the dual variables $\bar{\pi}$).

Step 4. For every $r \in \mathcal{R}$ with $\bar{c}_r < 0$ add the column r to $\mathcal{R}'$ and go to Step 2.

Step 5. If no columns r with $\bar{c}_r < 0$ exist, i.e. $\bar{c}_{min} \geq 0$, then stop. The optimal solution has been obtained.

Note that since the column generation procedure described above operates on the relaxed restricted problem, integer optimality is not guaranteed. For that reason, and in order to obtain the optimal integer solution, the column generation procedure is embedded in a Branch & Bound framework. This entire process is then referred to as the *Branch & Price* algorithm.

Clarke and Wright Savings

The Clarke and Wright algorithm (Clarke and Wright, 1964) is a heuristic method and, therefore, doesn't guarantee an optimal solution to the problem. However, it produces a solution in very short computational time and is very effective, often yielding a solution that is very close to the optimal one.

The key idea of the C&W heuristic is the computation of savings. Savings is a measure of how much of the trip length or cost can be reduced by "merging" a pair of nodes and creating a tour which can then be assigned to a single vehicle. Consider a depot (denoted as 0) and two nodes i and j included in the network. Assuming that c_{ij} corresponds to the cost needed to travel from node i to node j, the total cost needed for serving both nodes in the network with separate vehicles is $[c_{0i} + c_{i0} + c_{0j} +$

c_{j0}]. On the other hand, if we add both nodes to vehicle single tour to be assigned to a single vehicle, the total cost would be $[c_{0i} + c_{ij} + c_{j0}]$, and the improvement, i.e. the "saving" resulted from this operation would be:

$$S_{ij} = c_{io} + c_{oj} - c_{ij}$$

Higher values of the S_{ij} indicate that the inclusion of arc (i, j) (i.e. to combine nodes i and j in a single route) is more desirable. Of course, all feasibility constraints should be satisfied by such an operation. Therefore, the algorithm combines routes based on savings to improve the total routing costs. The C&W heuristic can be defined by the following discrete steps:

Step 1. Generate an initial solution that comprises of a route for each node i, i.e. $(0, i, 0), \forall i \in N$ where N is the set of nodes in the network, without including the start and end location of the vehicles.

Step 2. Calculate the savings S_{ij} for each pair $(i, j), \forall i, j \in N$.

Step 3. Create the savings list and sort it in descending order (rank from largest to smallest saving)

Step 4. Starting from the top of the list and moving downwards, include link (i, j) in a route if no constraints are violated (i.e. feasibility is maintained). A node which is neither the first nor the last at a route cannot be involved in the merge operations (because no arc exists connecting the node to the depot), as well as if the nodes are extremes of the same route. Proceed until there are no other savings left to be done within the current solution.

The solution to the VRP consists of the routes resulting during the above process.

Furthermore, in the method proposed, after the solution obtained by the C&W algorithm, a *Route Interchange* procedure can be employed, in order to improve the current solution. A route interchanging move (2-opt) can be performed, either within a single route or between two different routes, by simultaneously interchanging route sections. Relevant references can be found in Croes (1958) and Lin (1965).

The implementation of a route interchange within a single route implies the deletion of two arcs of the route, and the generation of two new arcs, while the section

between the generated arcs is reversed. An illustrative example is presented in Figure 4.2.

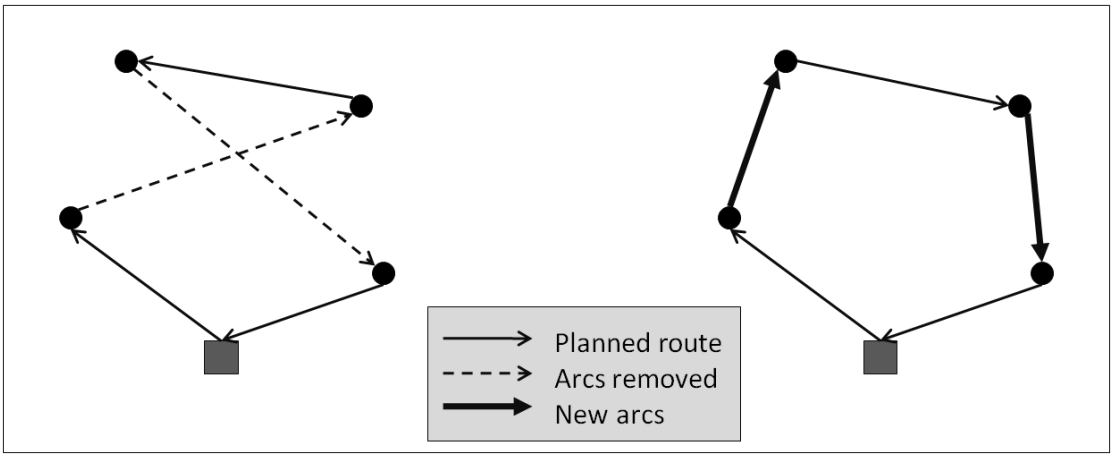


Figure 4.2: Route Interchange within a single route

For the case of a pair of routes, when a route interchange move is performed, each of the routes is essentially divided into two parts (the initial and the terminating), by deleting the intermediate arc. Then, the initial section of each route is connected with the terminating part of the other route. Therefore, two new arcs are generated for the connection of the routes. Figure 4.3 presents an example of this route interchange move.

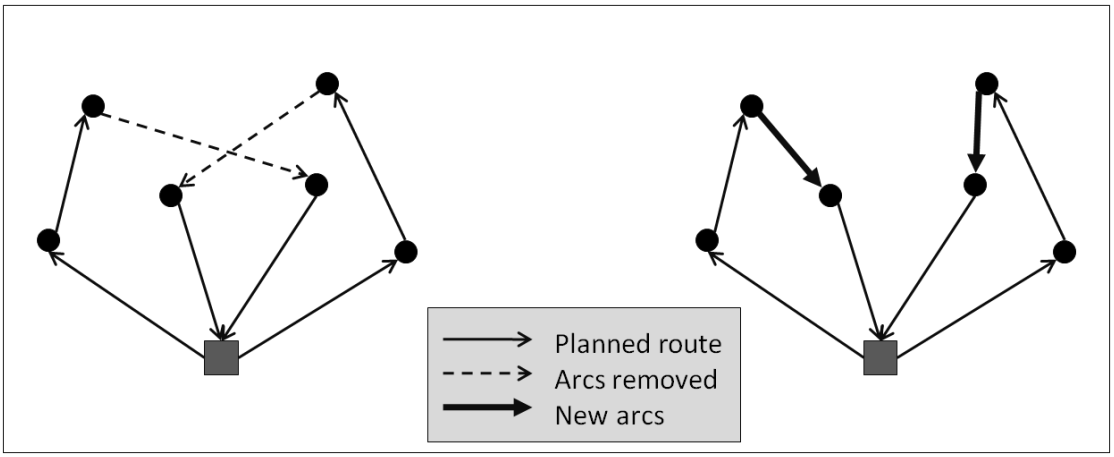


Figure 4.3: Route Interchange between two routes

4.2.2 PHASE II: INCLUSION OF THE EXCHANGE LOCATION *(MEETING)*

Obviously, the solution obtained from the previous phase does not necessarily constitute a feasible solution to the problem, because of the relaxation that allows each request to be served by all vehicles available. The first operation of this phase is

to identify from the VRP solution of the first phase whether there are any delivery orders that are assigned to a different vehicle(s) than the one originally assigned. If not, this implies that no exchange operation is required, and the solution resulting from the conventional approach constitutes the final solution to the problem. Otherwise, a vehicle should somehow receive its delivery orders that are loaded to the original vehicle. We refer to these orders as *exchanged orders*. Obviously these orders cannot be served before the vehicles meet and exchange the appropriate loads.

In order to determine the meeting location, we developed three algorithms that seek to identify the best exchange location according to the current solution, by respecting, of course, all feasibility constraints of the problem in hand. A brief description of each one follows:

5. ***Simple Local Search (SLS)*:** This algorithm employs an insertion method, where based on the VRP solution of the first phase, all possible customer locations are tested as potential meeting locations. This means that each candidate meeting location of one route is temporarily inserted between two consecutive customer orders of the other route. The combination of routes with the lowest cost will be the solution selected to proceed in the third phase of the method.

6. ***Advanced Local Search (ALS)*:** The first operation of this algorithm is exactly similar to the one in SLS. For each solution combination resulted from SLS, a post-optimization process is applied to the routes that seeks other solutions, by interchanging an order between the two routes, that can further improve the solution currently obtained. From the updated combinations, the one with the lowest cost is selected.

7. ***Complex Search (CS)*:** The final alternative algorithm also employs the SLS. For each solution combination resulted from SLS, a VRP-like problem is solved considering the network of all customer orders assigned after the exchange location. Therefore, the route segments before the exchange operation remain unchanged, and the routes resulting from the VRP, constitute the route portions after the exchange location. Again, from the updated combinations, the one with the lowest cost is selected.

In the following subsections, each algorithm is thoroughly described separately.

Algorithm 1: Simple Local Search (SLS)

Consider two routes denoted as R_1 and R_2 (resulting from the VRP solution of the first phase), each one assigned with a set of orders, $M = \{o_1, o_2, \dots, o_m\}$ and $N = \{o_{m+1}, o_{m+2}, \dots, o_n\}$ respectively. Each route is represented as a vector, i.e. $R_1 = [r_1(1), \dots, r_1(m), r_1(0)]$ and $R_2 = [r_2(1), \dots, r_2(n), r_2(0)]$, where node 0 corresponds to the depot. A node of a route k is denoted as $r_k(i),\ k \in \{1,2\}$, where i depicts the position of an order into the route. Nodes $r_1(1)$ and $r_2(1)$ correspond to the current vehicle locations at the replanning timestamp. We also denote as $r_k(e_k), k \in \{1,2\}$ the first exchanged node (customer request) of route k (i.e. the first node of the route sequence that was originally assigned to the other vehicle), where $e_k, k \in \{1,2\}$ corresponds to the position of the first exchanged node into the route. These nodes are crucial in order to ensure that the exchange operation will be strictly taken only before the first exchanged order of each route. The process of SLS is described in the pseudocode of Figure 4.4. The values $Cost_{IM}$ and $Routes_{IM}$ denote the total cost and the routes resulting from the solution of the conventional approach (Insertion Method).

$COST = Cost_{IM},\ ROUTES = Routes_{IM}$
For each $k_1 \in \{1, 2\}$
  For each $k_2 \in \{1, 2\}$
    If $k_1 \neq k_2$
      If $k_1 = 1$ **then** $a = m$ **else** $a = n$ **End**
      For each $j = 1$ **to** e_{k_2} **do**
        For each $i = 1$ **to** $e_{k_1} - 1$ **do**
          $R_{k_1} = [r_{k_1}(1), \dots, r_{k_1}(i), r_{k_2}(j), r_{k_1}(i+1), \dots, r_{k_1}(0)]$
          $R_{k_2} = [r_{k_2}(1), \dots, r_{k_2}(a), r_{k_2}(0)]$
          Compute cost $C^{ij} = C(R_{k_1}) + C(R_{k_2})$
          If $C^{ij} < COST$
            Check feasibility of R_{k_1} and R_{k_2}
            If R_{k_1} and R_{k_2} feasible
              $COST = C^{ij}$
              $ROUTES = \{R_{k_1}, R_{k_2}\}$
              $EL = r_{k_2}(j)$
            End
          End
        End
      End
    End
  End
End

Figure 4.4: Pseudocode of the Simple Local Search

At each iteration of the procedure, the algorithm inserts a node of route R_{k_2} between two consecutive nodes of route R_{k_1}. Of course this process concerns the nodes before the exchanged nodes of each route sequence. In case feasibility is maintained (in terms of time windows and time horizon constraints) and the total cost (distance) of the updated routes is the lowest one so far, the location $r_{k_2}(j)$ will be the exchange location (EL) where the vehicles are going to meet and transship the appropriate loads (orders). The final solution consists of the routes R_{k_1} and R_{k_2} saved in the value $ROUTES$.

Figure 4.5 presents an illustrative example of the Simple Local Search. Route sequences are represented on a line and all possible combinations of solutions generated by the SLS are displayed.

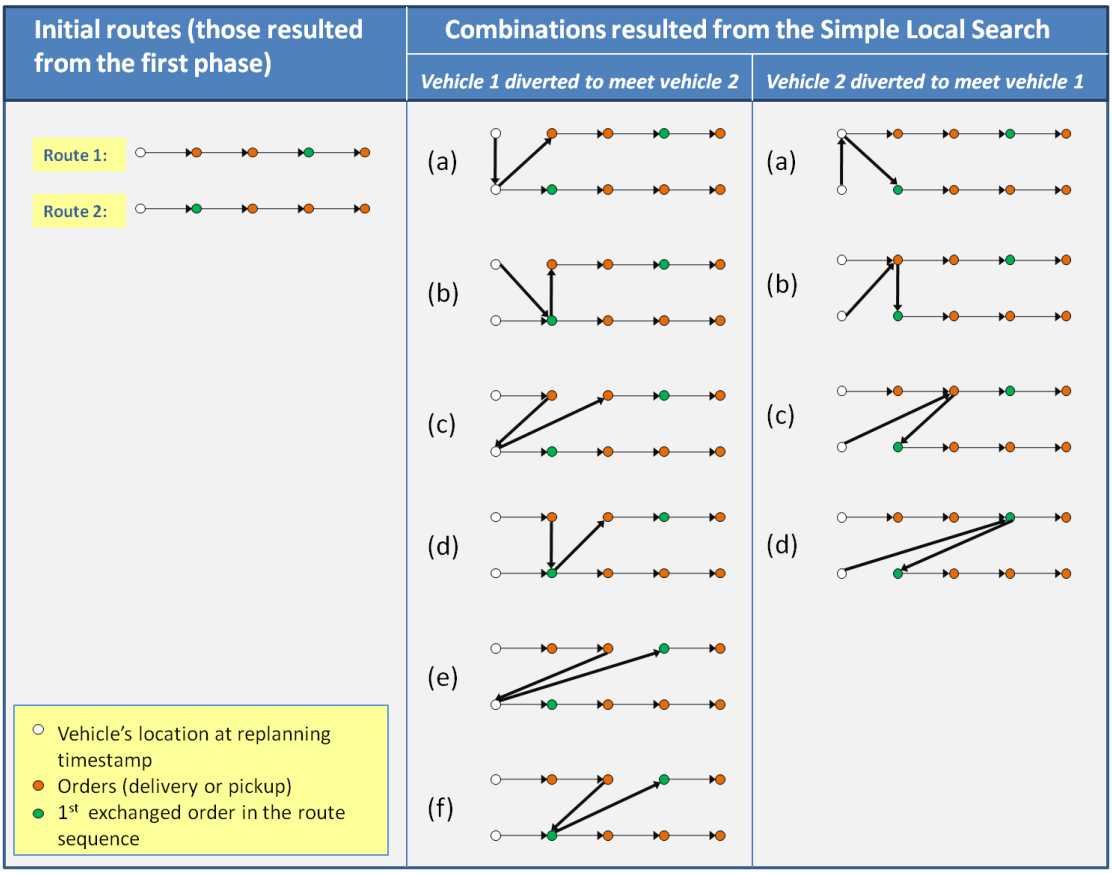


Figure 4.5: Example of Simple Local Search

Algorithm 2: Advanced Local Search (ALS)

ALS enhances the SLS algorithm with an additional process that can be seen as a post-optimization procedure. The basic idea of this process is to identify, if interchanging a customer between the two routes can provide a better solution, in terms of routing cost. Of course, feasibility checks are included in the process, in order to ensure that no constraints are violated. In addition to the time windows and time horizon constraints, the process also ensures that all exchanged orders are assigned to the vehicles after meeting at the exchange location, and that the exchange location isn't interchanged. The post-optimization procedure is applied at each iteration of the SLS, after the updated routes have been constructed, examining if there is any other solution that can further improve the solution currently obtained. Below, we describe this post-optimization process.

Consider the two routes denoted as R_1 and R_2 (resulted from an iteration of the SLS), and exactly the same notation as in SLS. Furthermore, both routes have a common node (apart from depot), denoted as $r_k(EL)$, $k \in \{1,2\}, EL = M \cap N$ that corresponds to the exchange location (EL). As $C[i,j]$, is denoted the cost of traveling from a node i to a node j.The pseudocode of the post-optimization process is shown in Figure 4.6.

For each $k_1 \in \{1,2\}$
  For each $k_2 \in \{1,2\}$
    If $k_1 \neq k_2$
      If $k_1 = 1$ **then** $a = m$ and $b = n$ **else** $a = n$ and $b = m$ **End**
      $w = 1$
      For each $i = 2$ **to** a **do**
        If $r_{k_1}(i) \neq r_{k_1}(EL)$
          For each $j = 1$ **to** b **do**
            $R_{k_1} = [r_{k_1}(1) \dots, r_{k_1}(i-1), r_{k_1}(i+1), \dots, r_{k_1}(0)]$
            $R_{k_2} = [r_{k_2}(1) \dots, r_{k_2}(j), r_{k_1}(i), r_{k_2}(j+1), \dots, r_{k_2}(0)]$
            Compute the value:
            $D_{ij} = \{C[r_{k_1}(i-1), r_{k_1}(i)] + C[r_{k_1}(i), r_{k_1}(i+1)] -$
              $C[r_{k_1}(i-1), r_{k_1}(i+1)] + C[r_{k_2}(j), r_{k_2}(j+1)] -$
              $C[r_{k_2}(j), r_{k_1}(i)] - C[r_{k_1}(i), r_{k_2}(j+1)]\}$
            $S(w) = [D_{ij}, R_{k_1}, R_{k_2}]$
            $w = w + 1$
          End
        End
      End
    End
  End
End
Sort the rows of S in descending order (only positive values of S)
For each D_{ij} of the S
  Check feasibility of R_{k_1} and R_{k_2}
  If R_{k_1} and R_{k_2} feasible
    $ROUTES = \{R_{k_1}, R_{k_2}\}$
    Break the for loop and return current solution
  End
End

Figure 4.6: Pseudocode of the post-optimization process

An important characteristic of applying this refinement to each route combination of the SLS is that an infeasible solution of the SLS may yield a feasible solution after the

post-optimization process. Furthermore, it should be mentioned that ALS can provide only better solutions than the ones of SLS. Moreover, on several cases, the solution is significantly improved without dramatically increasing the computational effort.

Algorithm 3: Complex Search (CS)

CS commences its operation also by employing SLS, in order to generate all possible solutions of routes (resulted from the first phase) that contain also a feasible exchange location. After this operation, a VRP is being solved for the requests planned to be served after the exchange location. Specifically, for each iteration of the SLS, the route segments until the exchange location remain unchanged for both routes, while for the remaining requests a typical VRP is solved. The VRP solution considers that vehicles start their routes from the exchange location and end at the depot. As a result of the process, we have to solve as many VRPs as the candidate solutions of SLS. Of course, if the unchangeable route segments of an iteration are not feasible (because of the time windows constraints), or in case there is not enough time for servicing the remaining orders, then the VRP doesn't need to be solved and the specific solution (combination) is discarded.

The VRPs can be solved through an exact method (Branch-and-Price) or a heuristic approach (Clarke & Wright savings), based on the problem scale, similarly to the VRP of the first phase. The present algorithm is able to provide better solutions than the SLS and ALS in some cases, especially in small scale instances where the exact method is used. However, the main drawback of CS is that requires more computational time to provide a solution, compared with the other two algorithms.

4.2.3 PHASE III: POST-OPTIMIZATION

The outcome of the procedure described so far will be two vehicle routes that have incorporated the load exchange operation. This last phase of the proposed methodology considers a post-optimization process, which is used as a refinement to improve the resulting solution. Its basic idea lies in allowing a request to be interchanged between the routes, if such a move favors the objective function. The process of post-optimization has been described in the previous subsection (Figure 4.6) for the ALS algorithm. Note that in ALS this post-optimization process is applied

during Phase II at each combination of resulting routes; however, in this phase, the said procedure is used only at the final solution of Phase II.

4.3 FIXED EXCHANGE LOCATION

A key advantage of taking into account the customer sites as exchange locations is that only one of the vehicles has to be diverted from its current route. For this reason, the vehicle with more available time is usually the one selected by the algorithm to be diverted from its path.

The proposed solution methodology may also be used for the case of a fixed transshipment point. That is, the exchange location is considered to be known prior to the start of operations and comprises a location owned by the operator. The exchange location operates as a "pigeonhole", where a vehicle can drop off the load to be exchanged in order for the other vehicle to pick up at a later point in time. The main advantage of such an option, compared to the dynamic one, is that vehicles don't have to meet at the same time instance and can perform the exchange activities independently. On the other hand, an obvious drawback is that both vehicles have to travel to this location. Furthermore, the fact that the exchange location is fixed within the service area, implies that it does not adapt to the service pattern as this changes in time (e.g. it may be effective today but not tomorrow). For that reason, historical information could be employed for determining the optimal position of such a location in the distribution area.

The solution methodology of the problem using such an exchange location is approximately the same with the one described above for dynamic exchange locations. Phases I and III remain the same. The main difference lies in the second phase, where the transshipment point has to be incorporated in the solution obtained from the VRP of the first phase. The algorithm used for this phase of the methodology, referred to as *Fixed Transshipment Point Incorporation (FTPI),* is described below. The basic idea of the process is the insertion of the fixed transshipment point in any consecutive nodes of the planned sequence of each route. Obviously, the transshipment point must be visited before the first exchanged order of each route. An illustrative example of the Fixed Transshipment Point Incorporation is presented in Figure 4.7. Route sequences are represented on a line, a "pigeonhole" is

located somewhere in the service area and all possible combinations of solutions generated by the FTPI are displayed.

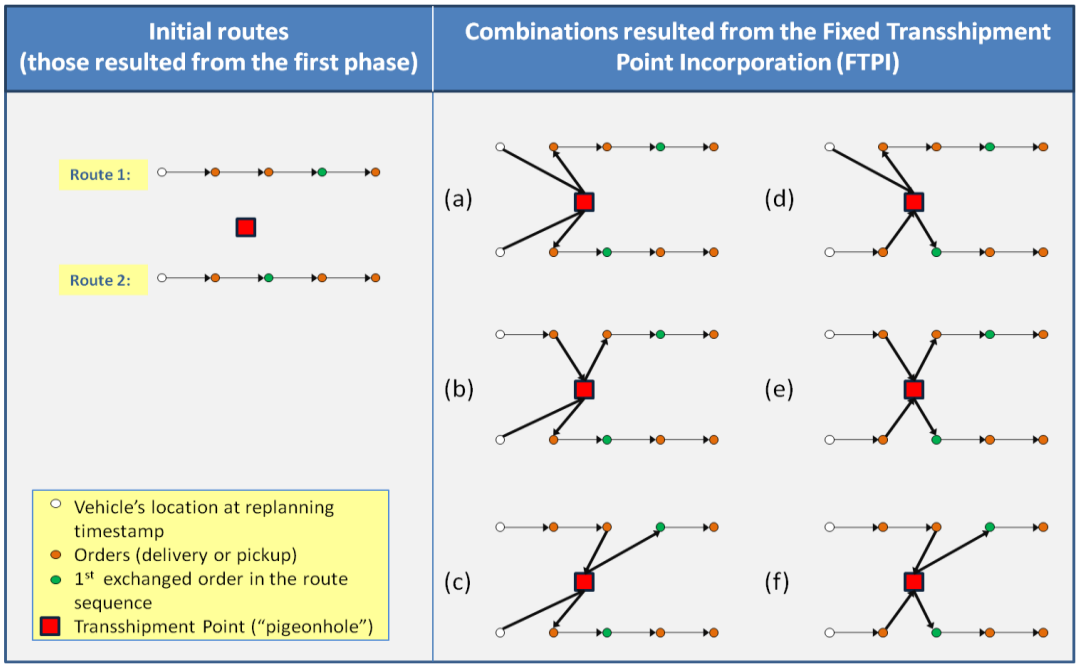


Figure 4.7: Example of Fixed Transshipment Point Incorporation

Consider two routes denoted as R_1 and R_2 (resulted from the VRP solution of the first phase), and exactly the same notation as in SLS. We also denote as tp the fixed transshipment point, and as $r_k(e_k), k \in \{1,2\}$ the first exchanged node of route k, where $e_k, k \in \{1,2\}$ corresponds to the position of the first exchanged node into the route. These nodes are crucial in order to ensure that the exchange operation will be strictly taken only before the first exchanged order of each route. The FTPI algorithm is described in the pseudocode of Figure 4.8. The values $Cost_{IM}$ and $Routes_{IM}$ denote the total cost and the routes resulted from the solution of the customary approach (Insertion Method).

$COST = Cost_{IM}, \quad ROUTES = Routes_{IM}$
For each $i = 1$ **to** $e_1 - 1$ **do**
  For each $j = 1$ **to** $e_2 - 1$ **do**
    $R_1 = [r_1(1), \dots, r_1(i),\ tp,\ r_1(i+1), \dots, r_1(0)]$
    $R_2 = [r_2(1), \dots, r_2(j),\ tp,\ r_2(j+1), \dots, r_2(0)]$
    Compute cost $C^{ij} = C(R_1) + C(R_2)$
    If $C^{ij} < COST$
      Check feasibility of R_1 and R_2
      If R_1 and R_2 feasible
        $COST = C^{ij}$
        $ROUTES = \{R_1, R_2\}$
      End
    End
  End
End

Figure 4.8: Pseudocode of the Fixed Transshipment Point Incorporation

At each iteration of the procedure, the algorithm inserts the transshipment point (tp) between two consecutive nodes of route R_1; the same operation also applies for R_2. Of course, this operation is performed only for the nodes before the exchanged ones of each route. A feasible solution (i.e. a valid incorporation of the tp) is provided when all time constraints are satisfied. An additional feasibility criteria is also incorporated here, where the sequence (in time) under which vehicles visit the exchange location is also examined; the vehicle that will drop-off the load of the exchanged orders, must be the first to visit the tp. In case both vehicles dispose exchanged orders, they have to meet at the transshipment point at the same time. If feasibility is maintained and the total cost (distance) of the updated routes is better than the current one, then the transshipment point ("pigeonhole") will be the location used for the exchange operation.

4.4 An Illustrative Example

In the following we provide an illustrative example to illustrate how the algorithm operates, and to prove that it can be beneficial compared to the traditional approach (Insertion Method).

Consider the case where a fleet of two identical vehicles are located at a single depot. The planning horizon is assumed to start at time 0 and end at time 480 (8 hours shift). The distances between all nodes are Euclidean, while the time needed to travel one unit of distance is 1 unit of time. Seven delivery requests are known in advance and have been assigned to a single vehicle at time 0 prior to start of operations. During execution, at replanning timestamp $t = 40$, seven pickup orders have arrived and need to be incorporated in the current plan. Both pickup and delivery requests require service within pre-specified time windows, while service time on-site to customer's location is considered to be zero.

Table 4.1 summarizes the main information associated to each request. The first column refers to the customer's ID, while the second one shows its type (P for pickup request, D for delivery request). The third column provides the coordinates of the customers' locations, the fourth one indicates the corresponding time windows and, finally, the last column (fifth) refers to the time when each request becomes known to the system. Obviously, all delivery requests have zero release time (i.e. are known prior to the starting of operations).

Table 4.1: Computational example: the data of the requests

Request	Pickup/ Delivery	Coordinates	Time Window	Call in time
0 (depot)		(70, 0)	[0, 480]	
1	D	(40, 25)	[0, 60]	0
2	D	(0, 60)	[80, 260]	0
3	D	(40, 65)	[130, 250]	0
4	D	(50, 50)	[60, 100]	0
5	D	(185, 75)	[190, 350]	0
6	D	(200, 55)	[240, 360]	0
7	D	(180, 40)	[280, 400]	0
8	P	(65, 40)	[100, 480]	7
9	P	(15, 35)	[50, 370]	13
10	P	(20, 50)	[100, 300]	15
11	P	(160, 50)	[150, 260]	20
12	P	(205, 35)	[250, 420]	27
13	P	(215, 65)	[200, 360]	32
14	P	(170, 65)	[150, 350]	38

Figure 4.9a provides the initial solution, where all requests known at the beginning of the planning horizon are planned to a single route. The second vehicle is located at the depot, and it can be used in case new requests arrive and the first one is unable to

serve them ("backup" vehicle). Figure 4.9b illustrates the status of the planned schedule at replanning timestamp t = 40, where 7 pickup requests have been received. During this time instance, the vehicle en route is located on-site at customer 4.

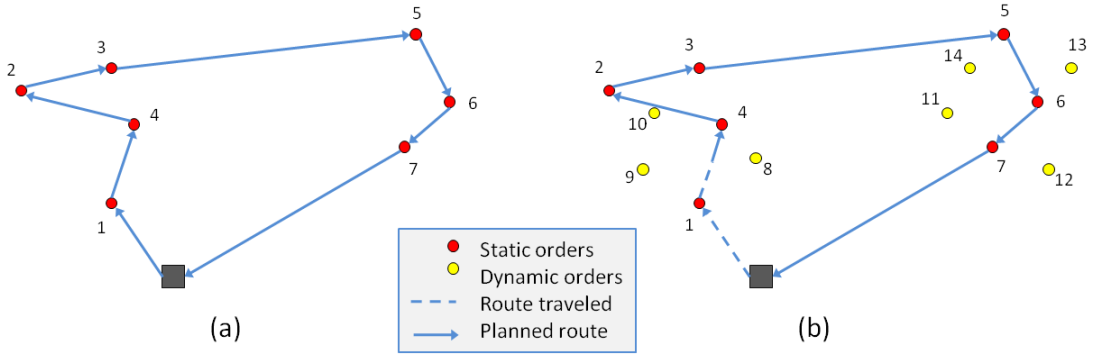


Figure 4.9: Computational Example: (a) Initial planned schedule at time t=0, (b) Schedule status at replanning timestamp (t=40)

The newly received requests have to be incorporated in the current plan. Figure 4.10 provides the optimal solution by applying a conventional solution approach (Branch-and-Price algorithm of Ninikas and Minis, 2011), in which no delivery orders may be re-arranged among vehicles. According to this solution, several pickup requests are assigned to the vehicle already en route, while the rest are assigned to the "backup" vehicle located at the depot. The total cost of this new solution is 912.3 distance units.

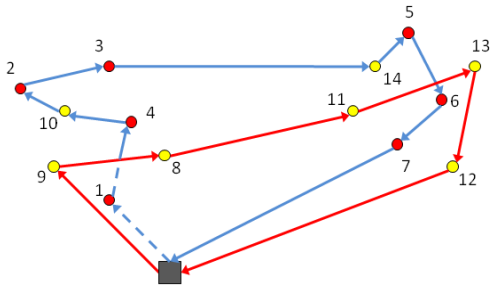


Figure 4.10: Optimal Solution when using a traditional approach (Total Cost: 912.3)

In the following, we compare the aforementioned solution with the one generated when using the LES. Each one of the algorithmic phases will be presented analytically.

Phase I. Initially, a modified VRP-like problem is solved. This problem considers both vehicles, spatially located at their current locations (i.e. the vehicle en route is located at customer 4, while the "backup" vehicle is located at the depot). Temporally,

the "backup" vehicle may start executing after time instance $t = 40$, while the vehicle *en route* after time instance $t = 60$, when the TW of customer 4 opens. Figure 4.11 provides the plan generated from the solution of this VRP-like problem. It is obvious from the figure, that delivery requests 5, 6 and 7 originally assigned to the vehicle en route (blue vehicle) have now been transferred to the vehicle that was located at the depot (red vehicle). As described earlier, in order for this solution to be valid, the blue vehicle has to meet the red vehicle in order to transfer the loads of customers 5, 6 and 7.

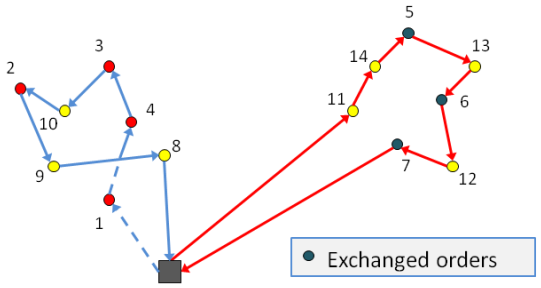


Figure 4.11: Phase I - VRP solution (Total Cost: 602.9)

Phase II. Determining the exchange location is performed with the Simple Local Search (SLS) process. Figure 4.12 provides four indicative solutions generated by SLS. Among those solutions, only the one presented in Figure 4.12c comprises a feasible solution in terms of time constraints, and therefore this solution is qualified for the next phase of the algorithm. The solution implies that the red vehicle should visit the location of customer 4 before serving any other request in order to acquire the load of the exchanged orders 5, 6 and 7. Based on this solution, the blue vehicle has to wait for 27.87 time units, after serving request 4, until the arrival of the red vehicle. The total cost of this solution is 663.8 distance units.

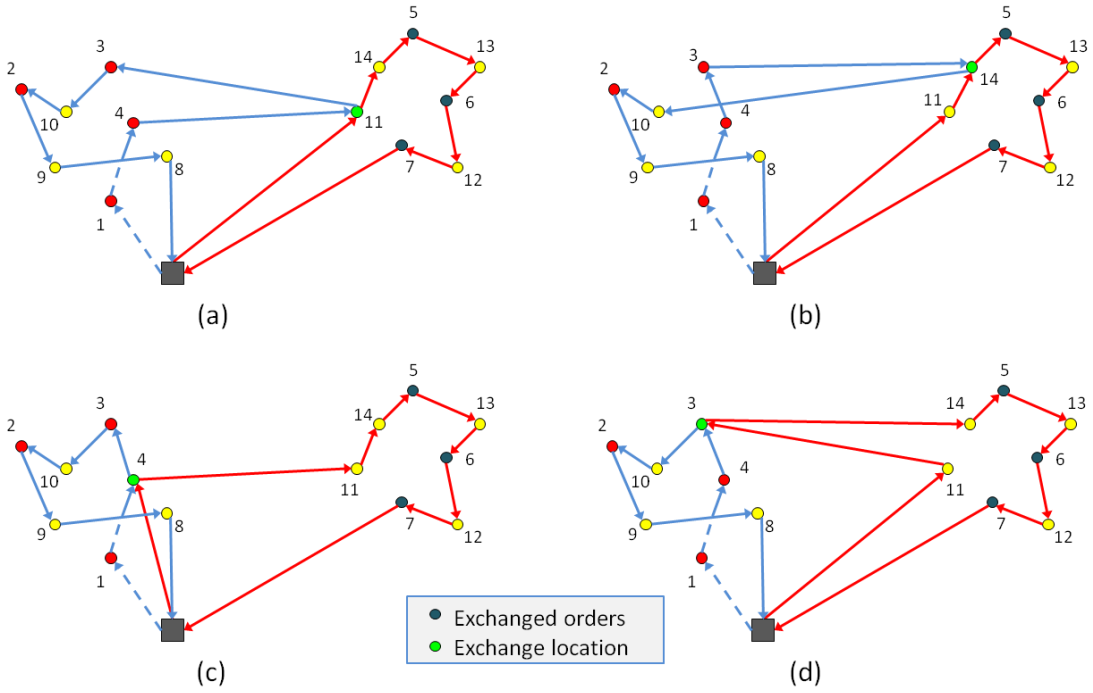


Figure 4.12: Four (4) potential solutions of Phase II (Simple Local Search). Solution (c) is the qualified one due to feasibility reasons (Total Cost: 663.8)

Phase III. Finally, a post-optimization process is applied to the qualified solution in order to test if it can further be improved. The result of this process is illustrated in Figure 4.13. This post-optimization procedure interchanges customer 8 from blue to the red vehicle, providing a total cost of 641.98 units.

It is obvious from this example, that the solution obtained when allowing vehicles to meet and exchange loads, improves the total travelled costs by 29.63% compared to the solution generated by the conventional approach, illustrating the utility of LES.

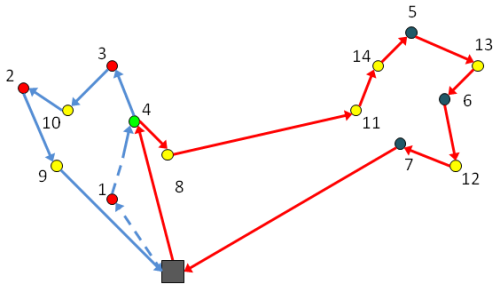


Figure 4.13: Phase III – Post-Optimization (Final Cost: 641.98)

Furthermore, we present a solution for the case in which a predefined exchange location was set. Figure 4.14 presents the solution of the current example, obtained from the LES considering a fixed transshipment point ("pigeonhole") as exchange

location for the vehicles (i.e. FTPI algorithm was employed for the second phase). The "pigeonhole" is assumed to be located in the geographical center of all customer coordinates. This method provides a solution with a total cost of 728.53 distance units, still better than the one obtained with a conventional approach.

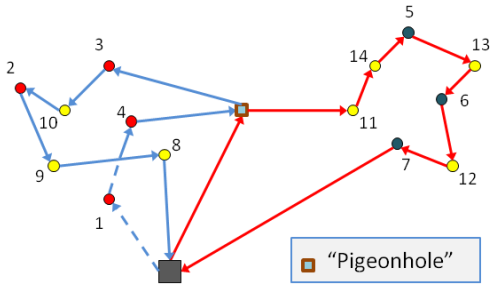


Figure 4.14: LES solution using a fixed transshipment point (Total Cost: 728.53)

CHAPTER 5 EXPERIMENTAL ANALYSIS

This chapter includes the computational experiments that have been conducted in order to test the efficiency of LES, when implemented on a VRPDP framework (as discussed in Chapter 3). As mentioned already, since LES comprises a strategy to be applied in combination to a conventional method, each test has been also solved with the Insertion Method of Ninikas and Minis (2011) which is implemented on a Branch-and-Price framework. After the generation of this solution, LES is performed to check if the application of the load exchange strategy may yield to an improved solution. Extensive computational tests demonstrate that such a strategy outperforms the conventional approach in many cases, with savings up to 24%. The main purpose of this Chapter is to evaluate the effectiveness of LES and identify the cases where the application of such a strategy outperforms the Insertion Method. Four algorithms (variants of the LES) are proposed, and implemented for each instance in order to draw useful conclusions about the most appropriate procedures to be applied to the LES.

5.1 EXPERIMENTAL SETUP

The tests were generated based on the Solomon benchmarks (Solomon, 1987), which comprise of sets R, C and RC; where each set denotes a different geographical distribution of customers (R: random, C: clustered, RC: mixed). In order to provide an extensive experimentation and identify the correlation of the LES performance over the geographical distribution of customers, we adopted tests from all three categories.

Additionally, we considered datasets consisting of 15, 25, and 50 customer requests per problem category, yielding 9 test cases. We limited the number of customer requests to 50, due to the assumption that a fleet of only two vehicles is available. Furthermore, for each one of the aforementioned nine test cases, we investigated both cases with and without time windows, yielding to a final set of 18 test cases. Each test case has been solved several times. This way we attempted to avoid case-sensitive results, by obtaining the average results of all runs.

Table 5.1 summarizes the previously described test bed.

Table 5.1: Sets of test cases

Set	Number of customer requests	Customers distribution	Time Windows (TWs)	Number of tests performed
1	15	R	Yes	8
2	15	C	Yes	7
3	15	RC	Yes	8
4	25	R	Yes	8
5	25	C	Yes	7
6	25	RC	Yes	6
7	50	R	Yes	6
8	50	C	Yes	6
9	50	RC	Yes	6
10	15	R	No	6
11	15	C	No	6
12	15	RC	No	6
13	25	R	No	7
14	25	C	No	6
15	25	RC	No	6
16	50	R	No	6
17	50	C	No	6
18	50	RC	No	6

Note that the Solomon benchmarks were reduced to m customer requests, $m \in \{15,25,50\}$. To do this we used the following process:

- For R and C problems, we produce a random number k, $k \in [0,100-m]$, and then from the Solomon problem we select the m consecutive customers starting from the $k-th$ customer in the data set until the $(k+m)-th$ customer (please note that for C problems, clusters of customers are given in sequential order in the dataset, that's why we select consecutive customers).
- For RC problems we produce two random numbers k_1 and $k_2, k_1 \in [0,50-a]$, $k_2 \in [50,100-b]$, where for integers a and b the following holds: $a+b=m$. a denotes the number of clustered orders and b the number of uniformly and randomly dispersed orders, respectively. This is because in RC problems of Solomon instances, customers from 0 to 50 are clustered while the others are uniformly dispersed. Then we select customers from (k_1+1) to (k_1+a) and from (k_2+1) to (k_2+b) to be the customers in our dataset.

For each customer-set generated from the above process, we randomly select orders to be static (based on the number of requests shown in Table 5.3) and the rest to be dynamic.

Regarding customer TWs, we use the ones proposed by Solomon. However, several modifications were applied in order for the data sets to correspond to fit the environment of our problem. These are described in the following:

- For the 15-customer data sets, we used the TWs as in the Solomon benchmark instances.
- For the other datasets (25 and 50-customers), we modified the TWs in order to be applied to our case. Our goal was to calibrate the test instances in order to apply to real-life conditions, where customer TWs may vary from very tight to very wide. For that reason, we used two (2) patterns of TWs as presented in Table 5.2. For each pattern, this table presents the percentage of customers, the time window duration of which is adjusted as shown in the following column; Tmax denotes the available time horizon, e.g. for the first pattern, 20% of the customers have TWs with duration equal to the remaining time horizon, 30%

of customers have TWs with duration equal to 80% of the available time horizon, etc.

- The duration of the planning horizon is not equal for all test cases.
- Finally, the opening TW of each customer request is set as a random number chosen from the interval $[0, T_{max} - R_i]$, where R_i denotes the length of the TW of each customer i.

Table 5.2: Created patterns of TWs for the sets of 25 and 50 customers

1st Pattern for TWs		2nd Pattern for TWs	
Percentage of all customers	Duration of their TW	Percentage of all customers	Duration of their TW
20%	Tmax	30%	0.8 * Tmax
30%	0.8 * Tmax	30%	0.6 * Tmax
40%	0.4 * Tmax	30%	0.4 * Tmax
10%	0.2 * Tmax	10%	0.2 * Tmax

Table 5.3 summarizes significant information associated with the test cases and the aforementioned assumptions. The first three (3) columns indicate the total number of requests considered in each case and the number of static and dynamic requests involved respectively. The fourth column depicts the number of served requests until the replanning timestamp. From now on, when referring to the customer characteristics, we imply their coordinates in the service area and their time windows (TWs).

Table 5.3: Characteristics of the data sets

Number of customer requests			Served requests at replanning timestamp	Customers distribution	Customer characteristics	
Total	**Static (Deliveries)**	**Dynamic (Pickup)**			**TWs**	**No TWs**
15	9	6	2	R, C, RC	Solomon	Solomon
25	13	12	2	R, C, RC	Solomon*	Solomon
50	32	18	5	R, C, RC	Solomon*	Solomon

(*) Solomon instances with several modifications

The following assumptions hold for all test instances generated for the experimentation of LES:

- Only one replanning cycle is considered and LES is applied once at the replanning timestamp.
- The optimal solution (generated prior to time 0) corresponds to the assignment of all delivery (static) requests to one of the two available vehicles, while the other is located at depot (as "backup" vehicle) for the service of dynamic orders.
- The replanning timestamp corresponds to a random time instant, based, however, on the number of served requests considered on each case. This means that by this time several static requests have already been served. In case vehicle is *en route* to its next destination at this instant, the location of this customer is considered as the current location of the vehicle.
- Capacity restrictions are not considered, since we assume that the operating scenario involves the distribution of letters or small parcels.
- For simplicity, the on-site service time at each request location is considered to be zero (0).

The experimentation is performed under four alternative algorithms for solving the LES. These algorithms are based on the solution methodology and the algorithmic processes outlined at Chapter 4. The main purpose is to evaluate their effectiveness in terms of cost and their efficiency in terms of computational time. For the first three algorithms, the exchange location (EL) where vehicles are allowed to meet is considered to be a dynamic location (the location of one of the unserved customers), while the latter algorithm considers a fixed exchange location ("pigeonhole"), which is located in the geometric center of all customer coordinates.

Table 5.4 summarizes the main features and methods used for the experiments. We applied different methods that suit the complexity of each problem (i.e. number of customers). Specifically, a Branch-and-Price (B&P) framework that provides exact solutions is used for the solution of phase I for the datasets with 15 customer requests, while for larger scale problems (i.e. 25 and 50 customers), we employed the Clarke & Wright savings algorithm (C&W savings) followed by a post-optimization process (Route-Interchange).

However, the main difference among the algorithms concerns the solution of Phase II (inclusion of the exchange location). In this Phase, each algorithm follows a different approach, as described in Chapter 4. Specifically, we refer to as "Dyn_LES1" the algorithm that uses the SLS, "Dyn_LES2" refers to CS, while "Dyn_LES3" refers to ALS. Finally, we refer to as "Fix_LES4" the algorithm that allows vehicles to meet at a predefined fixed exchange location. In this case, Phase II of the method is solved using FTPI, as described in Section 4.2.

Table 5.4: Basic features for the solution algorithms

Algorithm	Name	Exchange location	15 customers		25, 50 customers	
			Phase I *(ROUTING)*	Phase II *(MEETING)*	Phase I *(ROUTING)*	Phase II *(MEETING)*
1	Dyn_LES1	Dynamic	B&P	SLS	C&W savings + Route Interchange	SLS
2	Dyn_LES2	Dynamic	B&P	CS (B&P for the VRPs)	C&W savings + Route Interchange	CS (C&W savings for the VRPs)
3	Dyn_LES3	Dynamic	B&P	ALS	C&W savings + Route Interchange	ALS
4	Fix_LES4	Fixed	B&P	FTPI	C&W savings + Route Interchange	FTPI

The algorithms have been implemented in Matlab® 7.9.0 (R2009b) software. All tests were performed on an Intel Core i7 PC System with processor speed 1.6 GHz and 4.00 GB of RAM running Windows 7®.

5.2 COMPUTATIONAL RESULTS

This Section provides the computational results obtained from the above experiments. Section 5.2.1 summarizes the results obtained for the problems where time windows are imposed, while Section 5.2.2 investigates problems without time windows. In each Section all three cases, i.e. for the 15, 25 and 50 customers, are presented. Finally, Section 5.2.3 summarizes the results and provides an overview of the LES performance.

Each Figure corresponds to a set of test cases comprising of the same number of customers. This Figure summarizes the improvement percentage of each of the 4 algorithms over the Insertion Method (Y-axis). The x-axis refers to the test case, i.e. R, C and RC. Each graph in a Figure represents the results obtained by each of the four (4) proposed algorithms. An additional line is used in the Figure to indicate the mean improvement of the four algorithms. Detailed information regarding the results obtained for each test case has been included in Appendix A.

5.2.1 CASES WITH TIME WINDOWS

Figure 5.1 presents the improvement percentage of each algorithm for the sets 1, 2 and 3 of Table 5.1. It is quite clear that the algorithms provide significant improvement in comparison to the Insertion Method. The first three algorithms that allow vehicles to meet at dynamic exchange locations offer better results than the "Fix_LES4" algorithm. From the former three algorithms, it appears that "Dyn_LES2" seems to provide slightly better solutions compared to the other algorithms. In this case, this is caused by the fact that Phase II is solved with a more elaborate algorithm (CS) that employs an exact algorithm (B&P) for the solution of the problem. Furthermore, it seems that LES outperforms significantly the Insertion Method for clustered instances (C-problems).

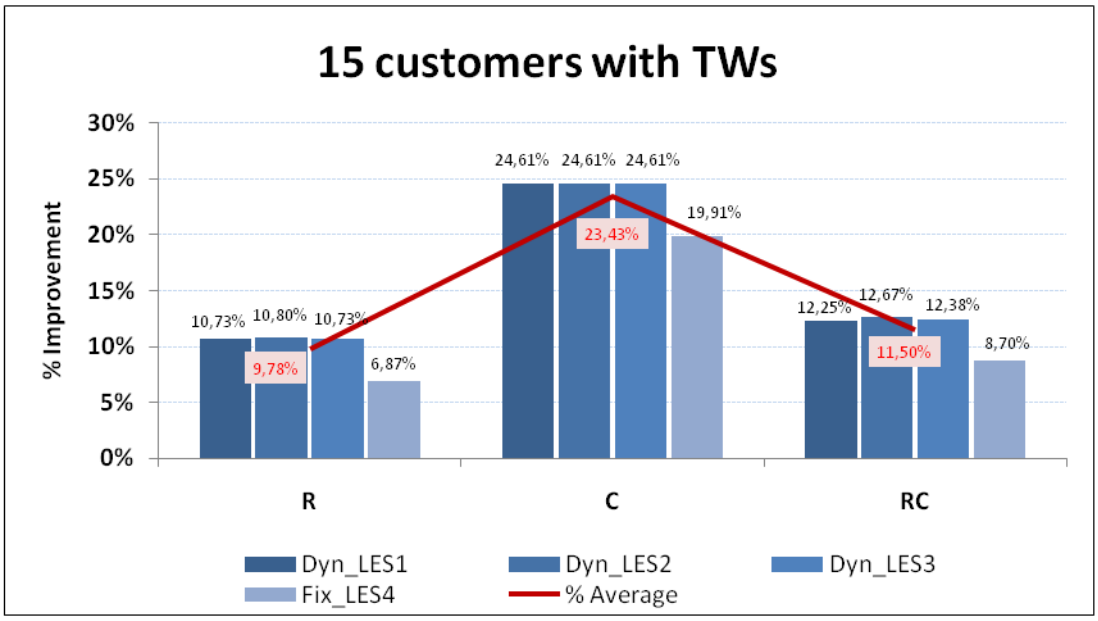


Figure 5.1: Average improvement percentage of each algorithm for 15 customer requests with TWs

Figure 5.2 presents the results of the algorithms for sets 4 to 6 of Table 5.1. For these test cases, the algorithms depict again better results for clustered cases. However, in this case, "Dyn_LES2" does not provide the best overall results over all algorithms. This is caused by the fact that in the 25-customer case, "Dyn_LES2" employs the C&W savings algorithm for the VRPs of Phase II (CS), which is a heuristic approach.

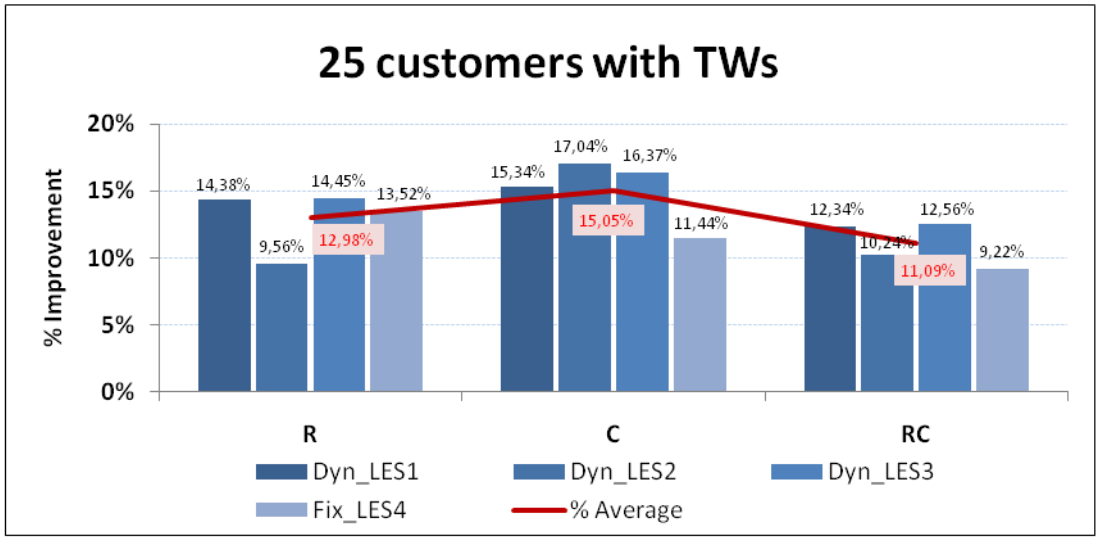


Figure 5.2: Average improvement percentage of each algorithm for 25 customer requests with TWs

Similar results are also obtained for the sets 7 to 9 of Table 5.1 for the 50-customer test cases. The results are shown in Figure 5.3. In this case "Dyn_LES3" seems to provide the best solutions over all test cases.

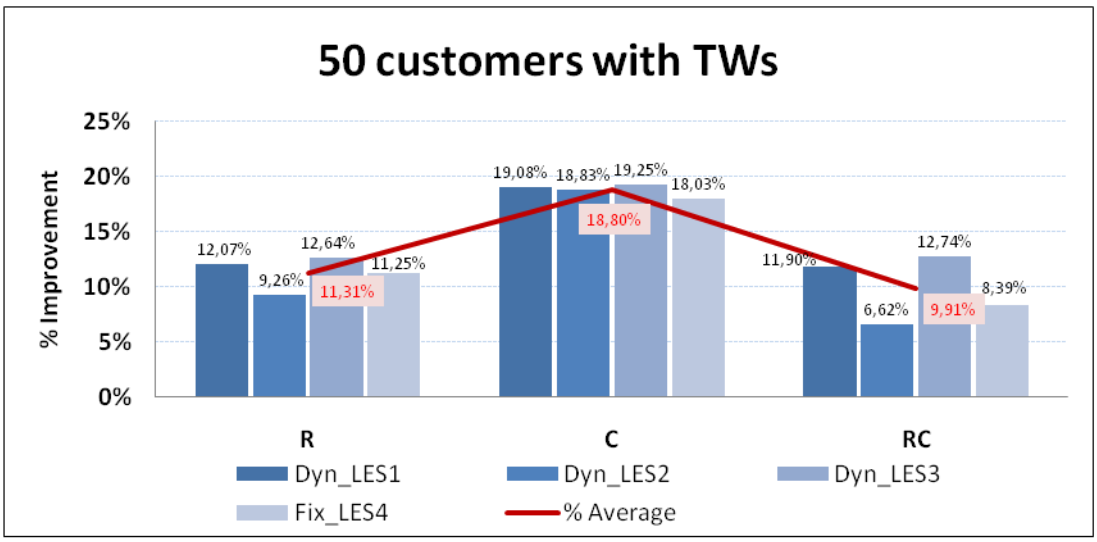


Figure 5.3: Average improvement percentage of each algorithm for 50 customer requests with TWs

5.2.2 NON TIME WINDOWED CASES

Figure 5.4 provides the results obtained for the 15-customer dataset (sets 10 to 12 of Table 5.1). "Dyn_LES2" leads to the best overall solutions (as in the 15-customer time-windowed case). This is due to the B&P technique used at Phase II of the algorithm, which provides optimal results (for each step of Phase II – not for the overall solution). Generally, there is a slight improvement in R problems compared to the C problems. The application of LES in RC non-time-windowed problems, however, seems to yield to lower improvements compared to the other test-cases.

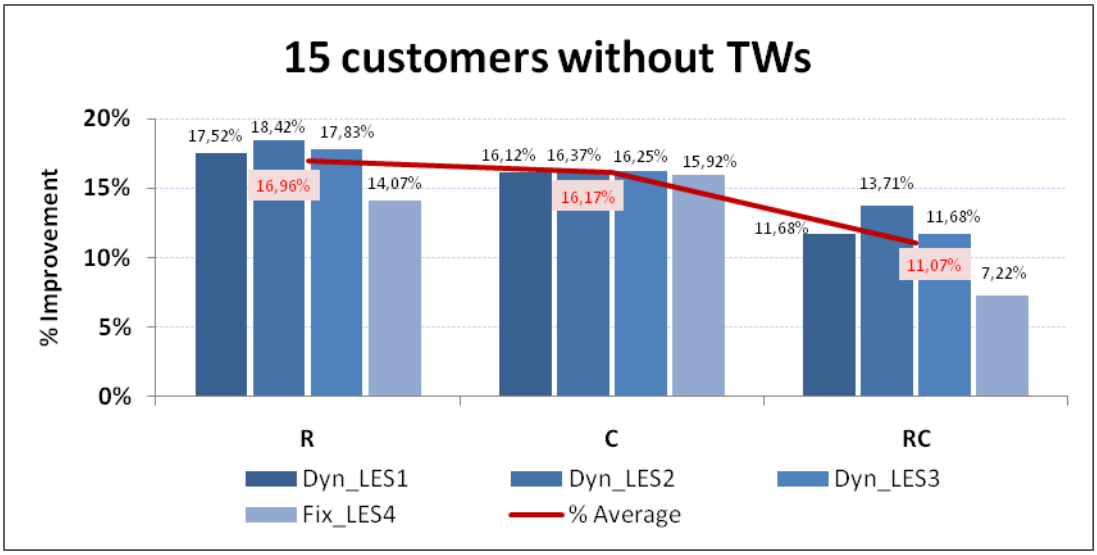


Figure 5.4: Average improvement percentage of each algorithm for 15 customer requests without TWs

Figures 5.5 and 5.6 present the improvement percentage of each algorithm for the sets 13 to 15 (25-customers) and 16 to 18 (50-customers) of Table 5.1 respectively. Algorithm "Dyn_LES2" seems to provide inferior solutions, probably for the same reason mentioned above for the equivalent problems. For those cases, it seems that there is no significant difference regarding the geographical distribution of the customers.

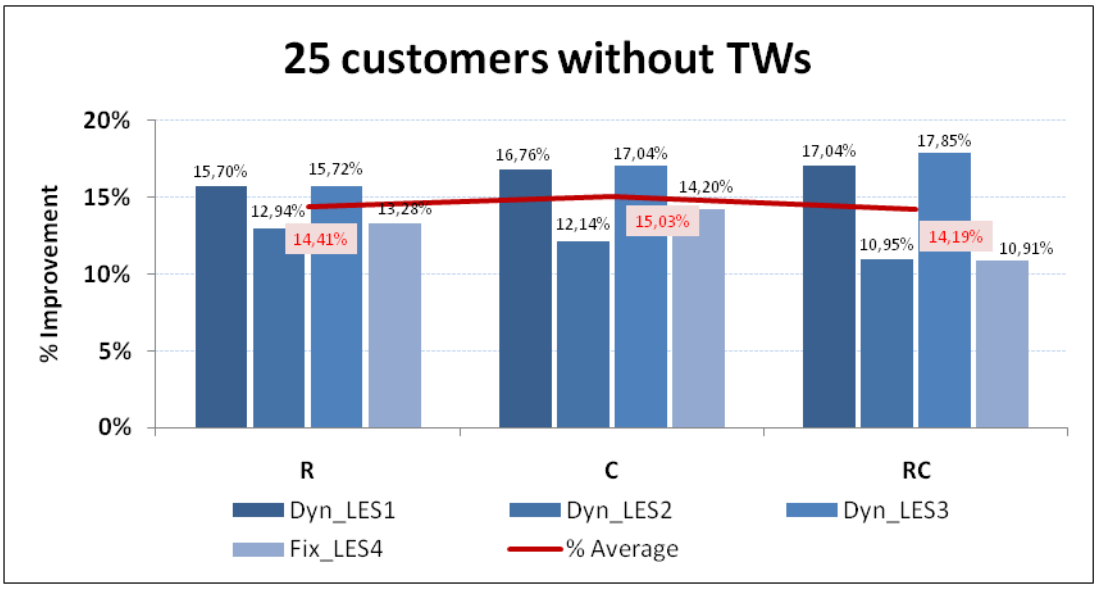


Figure 5.5: Average improvement percentage of each algorithm for 25 customer requests without TWs

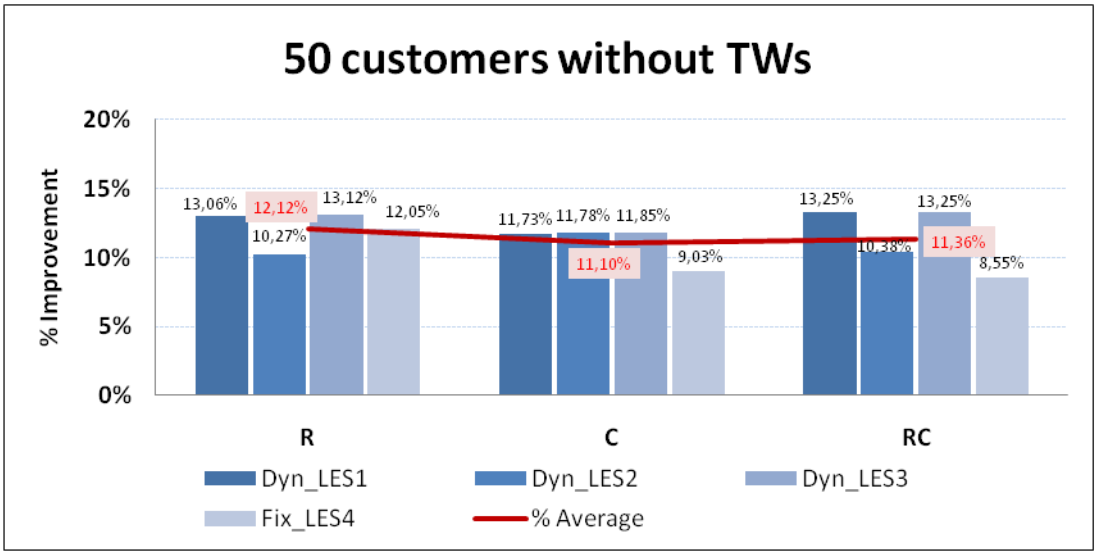


Figure 5.6: Average improvement percentage of each algorithm for 50 customer requests without TWs

5.2.3 Overview of LES Performance

In this Section we summarize the results obtained for all test cases by analyzing the results based: a) on the time-windows, b) on the geographical distribution of customer requests, and c) the computational effort of each algorithm. Figure 5.7 presents the average improvement percentage of each algorithm for all test-cases with time windows. From the Figure it appears that LES leads to significant improvements when customers require service within a time-window (up to 20% improvement in

terms of cost compared to the conventional solution approach), and especially in the clustered instances (C-problems). Regarding the alternative algorithms, it can be said that "Dyn_LES3" seems to outperform the other two algorithms that consider dynamic exchange locations. "Fix_LES4" seems to provide better results for the C-problems, which makes sense if one considers the fact that a vehicle is required to travel long distances in order to reach a customer of a cluster from the depot or from another cluster and, thus, the existence of a an appropriate exchange location may yield improved results.

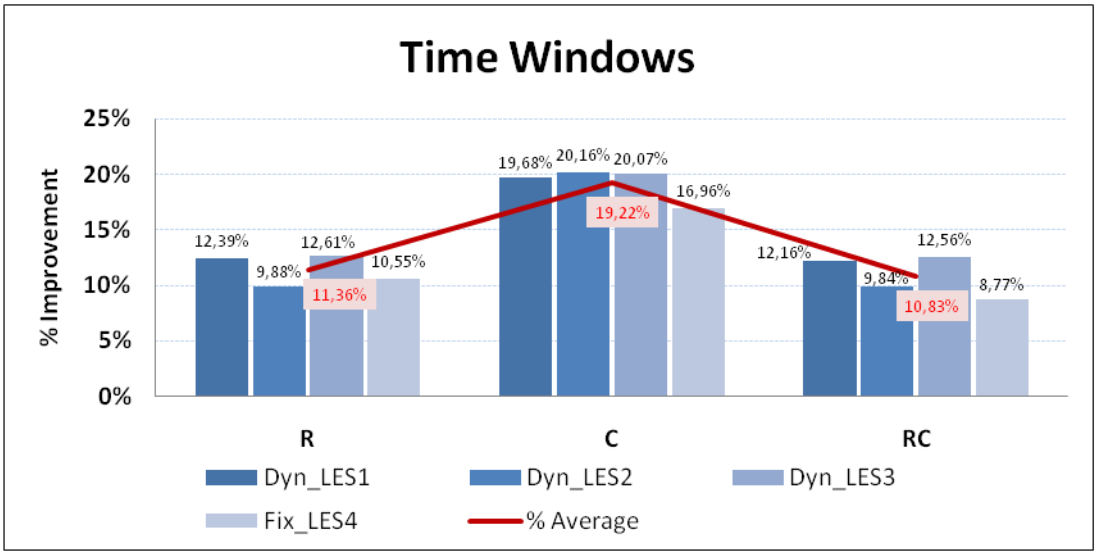


Figure 5.7: Average improvement percentage of each algorithm for all TWed cases

Figure 5.8 presents the aggregated results obtained for the cases without time windows. Significant improvements are provided as well, in comparison to the Insertion Method. For these cases, the improvement of LES seems to be balanced regardless the geographical distribution of customers. This is mainly caused by the fact that customers can be now served throughout the available planning horizon at any moment, which seems to provide a certain flexibility to LES regardless the location of the customer requests.

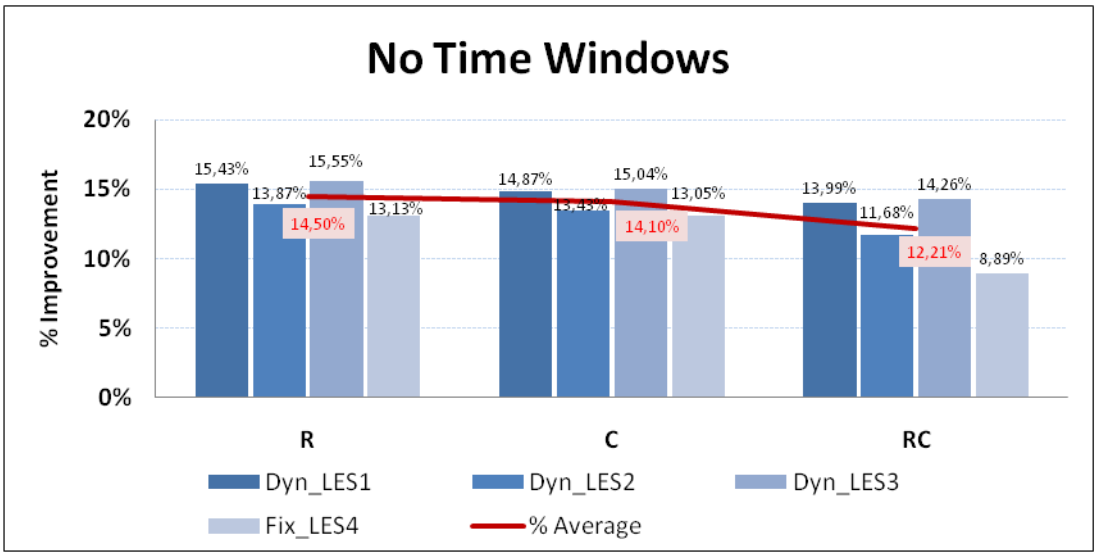


Figure 5.8: Average improvement percentage of each algorithm for all non-TWed test cases

Figure 5.9 summarizes the results from all test cases. It is rather clear, that LES performs better on C-problems, to an average improvement of 16.7%. Furthermore, "Dyn_LES3" seems to be the most efficient algorithm in terms of routing cost over all test cases.

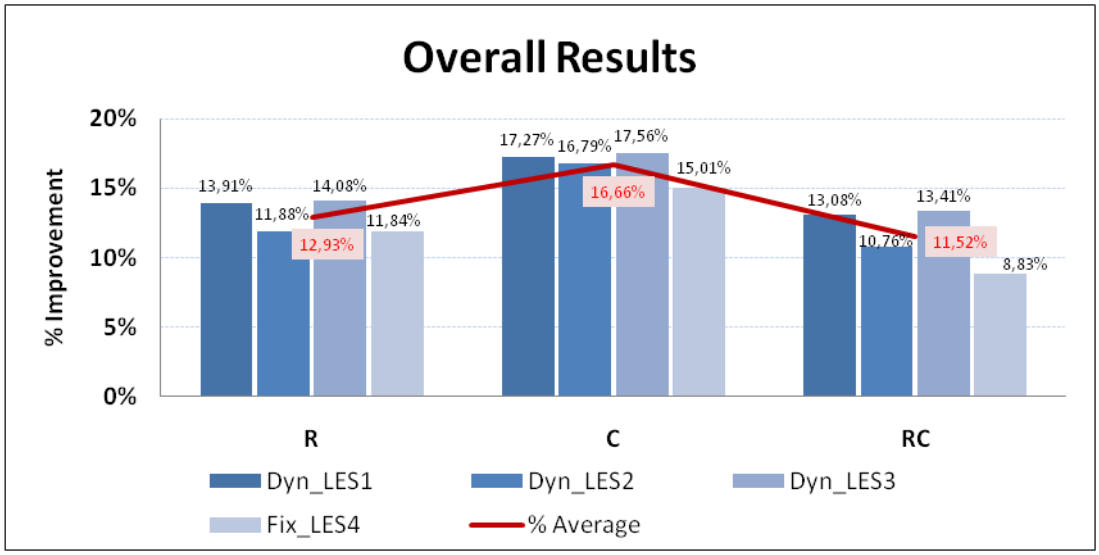


Figure 5.9: Average improvement percentage of each algorithm for all test cases

As a conclusion, Figure 5.10 presents an overview of LES performance as a function of the number of customers, based on their geographical distribution. From this figure, it can be concluded that LES seems to have the best performance on C-problems, as it was expected, and especially in small instances (15-customer sets). This lies in the geographical distinction of the customers in clusters, where a vehicle may have to travel in more than one cluster, if a load exchange operation is not allowed. It can also

be seen that there is a slight improvement in R-problems compared to the RC-problems. Furthermore, it should be mentioned that the heuristic mechanism employed for the solution of the first phase of the algorithm for the larger scale instances (25 and 50 customers) is responsible for this slight deterioration of the improvement in those instances. This means that the use of an exact-based method (e.g. Column generation) can lead to better results.

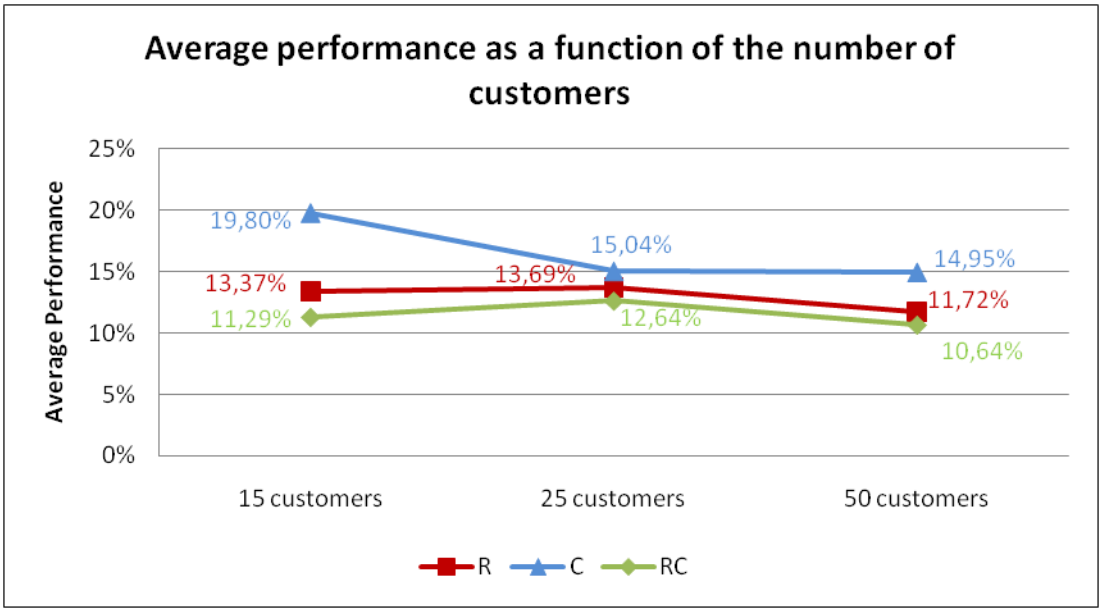


Figure 5.10: Average performance of LES as a function of the number of customers

Table 5.5 presents the average computational time of the algorithms for each case set. As expected, in the 15-customer problems, "Dyn_LES2" requires more computational time due to the Branch-and-Price technique used in Phase II. Clarke and Wright savings provides fast solutions, without significantly increasing the computational time compared to the other algorithms.

Table 5.5: Average time (in seconds) of each algorithm for each test case

	TWs			no TWs		
	15 customers	25 customers	50 customers	15 customers	25 customers	50 customers
Dyn_LES1	4,69	25,26	632,03	20,71	17,36	486,59
Dyn_LES2	51,30	31,93	758,56	87,11	20,68	501,93
Dyn_LES3	5,91	30,79	654,18	21,67	21,03	507,06
Fix_LES4	4,32	25,99	688,74	23,71	17,41	487,16

Figure 5.11 shows the average time spent by each algorithm to solve a problem of 15 customers with time windows (red line) and without TWs (blue line). As expected, problems with time windows are solved faster, due to the constrained solution space.

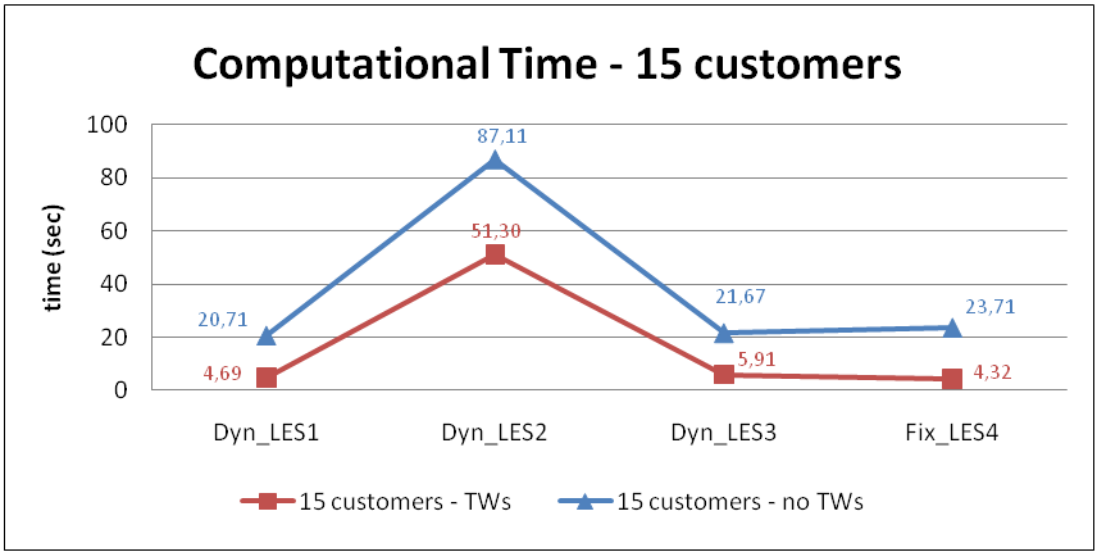


Figure 5.11: Average time of each algorithm for test sets with 15 customer requests

For larger scale problems, there is no significant difference in terms of computational time. Figures 5.12 and 5.13 provide the computational effort of all algorithms for problems with 25 and 50 customers, respectively. It should be noted here, that the greatest part of the time required by the algorithms, is spent during the first phase of the method of each algorithm (solution of the VRP-like problem). Although the Clarke and Wright savings used is a quite fast algorithm, the Route-Interchange post-optimization process requires significant computational effort (especially for the 50-customer test cases). However, one could use faster heuristics or metaheuristics (e.g. Tabu Search algorithm), which can significantly reduce the computational effort needed for this post-optimization process.

It is also worth mentioning that the algorithms require less time for problems without time windows, because the C&W savings provides a solution faster when TWs are not included.

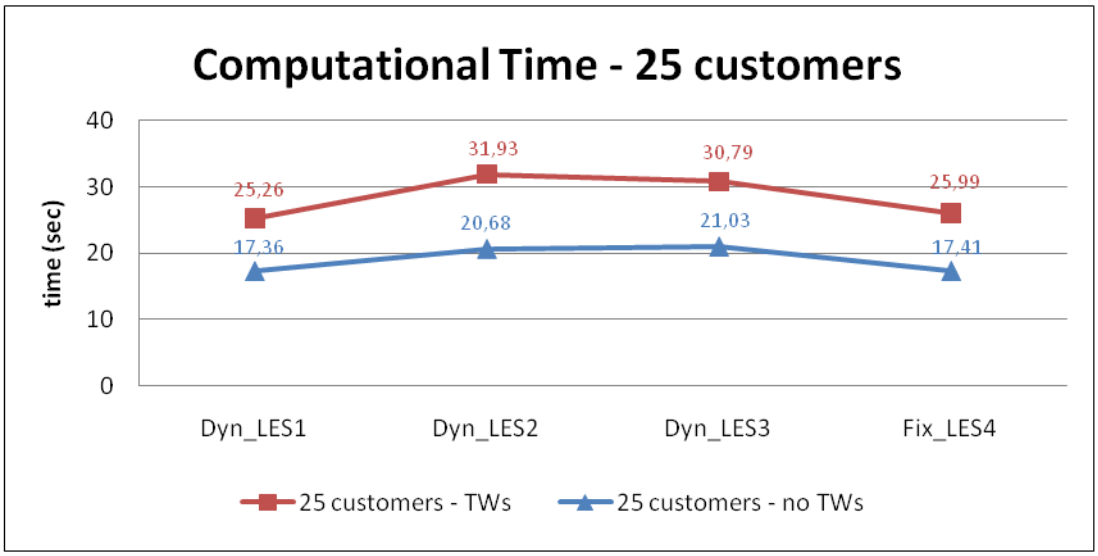


Figure 5.12: Average time of each algorithm for test sets with 25 customer requests

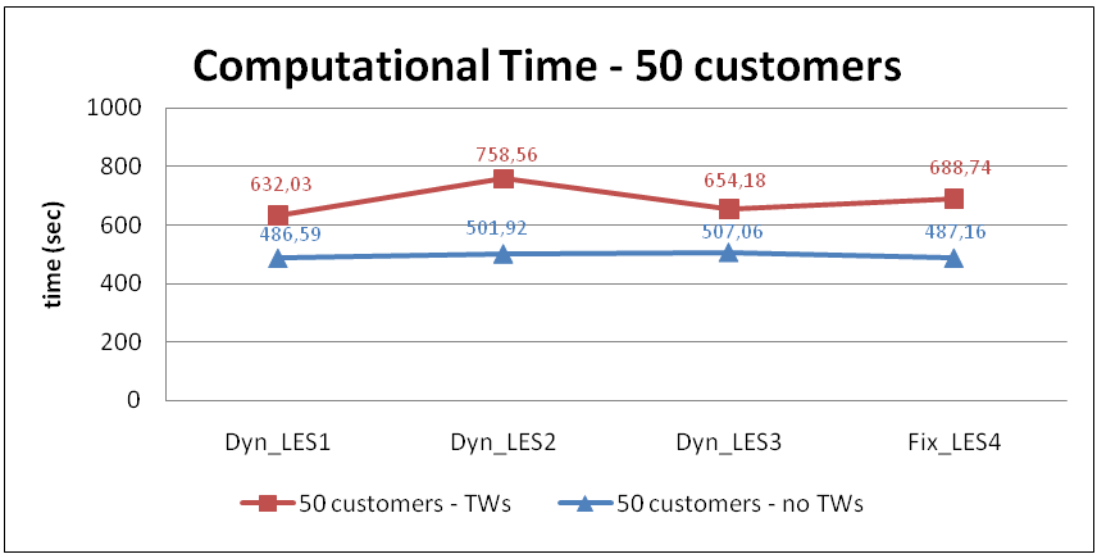


Figure 5.13: Average time of each algorithm for test sets with 50 customer requests

5.3 CONCLUSIONS AND FINDINGS

Four algorithms, variants of the Load Exchange Strategy, were evaluated in this Chapter. All algorithms use Branch-and-Price as the main solution framework for small size problems (15 customers). For larger instances (25, 50 customers), a heuristic approach based on Clarke and Wright savings is employed, followed by post-optimization process (Route Interchange) in order to improve the current solution obtained by C&W. The use of an exact method for such larger scale problems could probable lead to better improvements than those obtained, but it would also result to a significant increase in computational effort – possibly unreasonable for real-time solutions. The conclusions of the experimental investigation may be summarized as follows:

- The LES provides significant improvements vs. the Insertion Method, in which no load exchange is allowed. This holds for all cases examined in the current thesis.
- Especially for problems with time windows, where the spatial distribution of customers is distinctively clustered (C problems), LES provided the most pronounced results, leading to an average improvement of up to 20%.
- For the test-cases with no time windows, it seems that the LES provides equivalent improvements (on average) on all cases regardless the spatial distribution of customers.
- Allowing vehicles to meet at dynamic exchange locations produces better solutions compared to the strategy of allowing them to meet at a fixed transshipment point. It is worth noting that a potential analysis of historical information in order to identify the optimal location of such a fixed location (or more than one), may yield better results.
- Overall, tests showed that the use of dynamic exchange locations led to an average improvement of 15.02% while the use of a fixed transshipment point produces solutions with average improvement of 11.89%.
- Regarding the dynamic exchange location algorithms, we may conclude the following:

- In all test cases, “DynLES1” seems to be faster. This was expected, since “DynLES2” and “DynLES3” use a more elaborate process in the second phase”.
- For the 15-customer test cases, “DynLES2” algorithm produces the best results in terms of routing cost, but consumes more computational time. For larger instances, the algorithm is not so effective, because of the heuristic approach (C&W savings) used in the second phase of the method, which doesn’t always provide a “good” solution. For those instances, “DynLES3” seems to be the most appropriate.
- In general, we may conclude that “DynLES3” seems to be the most effective algorithm, both in solution quality and in computational effort.

CHAPTER 6 CONCLUSIONS

In the present thesis, we developed the Load Exchange Strategy for solving the Vehicle Routing Problem with Dynamic Pickups (VRPDP). VRPDP considers vehicles which are routed to serve delivery orders known prior to the start of operations. During the execution of the planned routes, newly arrived orders request on-site pickup and have to be served by the fleet of vehicles within the current period of operations. Therefore, at a certain replanning time, the dispatcher needs to incorporate these orders in the current routes, minimizing the total distance travelled by vehicles and maximizing the number of served customers. The main idea of LES is to allow for delivery orders to be transferred from one vehicle to another, if it's profitable. Of course, this implies an exchange location to be used for the transshipment of loads. This strategy provides improved solutions in many cases, compared to the conventional approach, in which each vehicle serves all orders pre-assigned to it.

Method Description

The solution methodology of LES concerns a fleet of two vehicles and allows exchange operations to be performed at either preferable locations within the service area, or at a fixed transshipment point ("pigeonhole"). The method is divided into three basic phases: (a) Routing, (b) Meeting, and (c) Post-Optimization. At the first phase, a typical Vehicle Routing Problem is solved with all requests allowed to be served by any vehicle. Secondly, Phase II of the method seeks to find the meeting location where, if needed, vehicles can exchange the appropriate loads. In this phase, we analyze several algorithms that are employed for seeking the exchange location, i.e. the Simple Local Search (SLS), the Advanced Local Search (ALS) and the Complex Search (CS) for the preferable location case. We also study the Fixed Transshipment Point Incorporation (FTPI) for the case of a predefined exchange location. At Phase III of the method, a post-optimization process is employed, that interchanges a request between the routes, if such an action can further improve the current solution. An example is also presented in order to illustrate the proposed solution approach.

Analysis and Results

The main scope of the experimental analysis was to identify whether or not LES is beneficial, compared to a conventional approach that does not allow exchange operations. An extensive set of test cases was conducted based on Solomon's test instances. Each test was initially solved using a conventional approach (Insertion Method) and then all four algorithms (variants of LES) proposed were also tested.

From the experiments conducted, it is clear that LES performs better (in terms of solution quality) than the conventional approach in all cases. Specifically, the following conclusions were obtained:

- LES provides better quality solutions in comparison to the Insertion Method by an average of up to 15% for the case of dynamic exchange locations, and up to 12% in case of a fixed transshipment location.
- Larger improvements for cases with time windows, in which customers are geographically clustered.

- The solutions obtained from LES lead to a more balanced workload between vehicles.
- For smaller-scale instances, the best solutions were obtained from the "DynLES2" algorithm; the latter however, requires significant computational time. For larger-scale instances, the performance of "DynLES2" seems to deteriorate, due to the heuristic method used for the execution of Phase II.
- "Dyn_LES1" provides the fastest solutions. Nevertheless, "DynLES3" and "Fix_LES4" seem to have comparable performance to "Dyn_LES1".
- Generally, the best trade-off between computational time and total routing cost is provided by "DynLES3".

Future Research

The main scope of the current thesis was to test and identify the benefits of an innovative strategy that was initially proposed by the on-going research of Ninikas and Minis (2011) currently conducted in the Design, Operations & Production Systems Lab (DeOPSys) of the Department of Financial and Management Engineering (FME) of the University of the Aegean.

The strategy will be further expanded and used as an extension of an algorithm solving the VRPDP, by examining if enabling load exchange operations will provide better quality solutions. It is noted that LES doesn't concern a new problem but it is a strategy, the solution of which is applied only when it is better than that the one provided by the approach, in which no load exchange is allowed. We can safely conclude that this strategy is very efficient, especially in cases in which a great number of dynamic requests occur in the service region of a vehicle and a "backup" vehicle is forced to serve the majority of those orders.

As indicated by an extensive experimental analysis, LES can yield solutions that are more efficient than the no exchange approach in many cases. This result is highly suggestive for further research leading to directions of defining the appropriate conditions under which an LES solution outperforms the no exchange approach.

Further work on the LES includes the incorporation of a Column Generation algorithm as the solution method in order to provide exact and optimal solutions. We also suggest that diversion strategies may be used in combination to LES resulting to

even better solutions. Another significant direction of future work regards the case of a larger fleet of vehicles, where the strategy becomes even more complicated. Finally, further investigation on the fixed transshipment point may be conducted to determine the optimal location of these points within the service area (e.g. through analysis of historical information).

REFERENCES

1. Aldaihani, M., Dessouky, M., (2003), "Hybrid scheduling methods for paratransit operations", *Computers and Industrial Engineering*, 45, 75–96.
2. Barnhart, C., Johnson, E., Nemhauser, G., Savelsbergh, M., Vance, P., (1998), "Branch-and-Price: Column generation for solving huge integer programs", Operations Research, 46(3), 316-329.
3. Bausch, D.O., Brown, G., Ronen, D., (1995), "Consolidating and dispatching truck shipments of heavy petroleum products", *Interfaces*, 25, 1-17.
4. Bell, W., Dalberto, L., Fisher, M., Greenfield, A., Jaikumar, R., Kedia, P., Mack, R., Prutzman, P., (1983), "Improving the distribution of industrial gases with an online computerized routing and scheduling optimizer", *Interfaces*, 13, 4-23.
5. Benders, J.F., (1962), "Partitioning procedures for solving mixed-variables programming problems", *Numerische Mathematik*, 4, 238-252.
6. Bertsimas, D., Van Ryzin, G., (1991), "A stochastic and dynamic vehicle routing problem in the Euclidean plane", *Operations Research*, 39, 601-615.
7. Bianco, L., Mingozzi, A., Ricciardelli, S., (1994), "A set partitioning approach to the multiple depot vehicle scheduling problem", *Optimization Methods and Software*, 3, 163-194.
8. Bramel, J., Simchi-Levi, D., (2002), "Set-covering-based algorithms for the capacitated VRP", In: Toth, P., Vigo, D., editors, "The Vehicle Routing Problem", SIAM, Philadelphia, 85-108.
9. Branke, J., Middendorf, M., Noeth, G., Dessouky, M., (2005), "Waiting strategies for dynamic vehicle routing", *Transportation Science*, 39, 298-312.
10. Brown, G., Ellis, C., Graves, G., Ronen, D., (1987), "Real time wide area dispatch of Mobil tank trucks", *Interfaces*, 17, 107-120.
11. Carpaneto, G., Ammico, M.D., Fischetti, M., Toth, P., (1989), "A branch and bound algorithm for the multiple depot vehicle problem", *Networks*, 19(5), 531-548.
12. Christofides, N., (1979), "Traveling salesman problem", In: Christofides, N., Mingozzi, A., Toth, P., Sandi, C., editors, *Combinatorial Optimization*, Wiley, New York, 131-149.

13. Christofides, N., Beasley, J.E., (1984), "The period routing problem", *Networks*, 14, 237-256.
14. Christofides, N., Mingozzi, A., Toth, P., (1979), "The vehicle routing problem", In: Christofides, N., Mingozzi, A., Toth, P., Sandi, C., editors, *Combinatorial Optimization*, Wiley, New York, 315-338.
15. Clarke, G., Wright, J.W., (1964), "Scheduling of vehicles from a central depot to a number of delivery points", *Operations Research*, 12(4), 568-581.
16. Cornuejols, G., Nemhauser, G.L., (1978), "Tight bounds for Christofides traveling salesman heuristic", *Mathematical Programming*, 14(1), 116-121.
17. Cortes, C., Matamala, M., Contardo, C., (2010), "The pickup and delivery problem with transfers: Formulation and a Branch-and-cut solution method", *European Journal of Operational Research 200*, 711-724.
18. Croes, G., (1958), "A method for solving traveling salesman problems", *Operations Research*, 6, 791-812.
19. Dantzig, G.B., Ramser, J.H., (1959), "The truck dispatching problem", *Management Science*, 6(1), 80-91.
20. Desaulniers, G., Desrosiers, J., Ioachim, I., Solomon, M.M., Soumis, F., Villeneuve. D., (1998), "A unified framework for deterministic time constrained vehicle routing and crew scheduling problems", In: Crainic, T.G. Laporte, G., editors, *Fleet Management and Logistics*, Kluwer, Norwell, MA, 57–93.
21. Desrochers, M., Lenstra, J.K., Savelsbergh, M.W.P, (1990), "A classification scheme for the vehicle routing and scheduling problems", *Journal of Operational Research Society*, 46, 322-332.
22. Desrosiers, J., Lübbecke, M.E., (2005), "Selected topics in column generation", *Operations Research*, 53(6), 1007-1023.
23. Dikas, G., (2008), "A toolkit for the optimal solution of the vehicle routing problem with time windows and capacity constraints", Diploma Thesis, Chios.
24. Fisher, M.L., Greenfield, A.J., Jaikumar, R., Kedia, P., (1982), "Real-time scheduling of a bulk-delivery fleet: Practical application of a Lagrangian relaxation", Report 82-10-11, Decision Sciences Department, University of Pennsylvania, Philadelphia, U.S.A.
25. Gambardella, L.M., Rizzoli, A.E., Oliverio, F., Casagrande, N., Donati, A.V., Montemanni, R., Lucibello, E., (2003), "Ant colony optimization for vehicle

routing in advanced logistic systems", In: International Workshop on Modelling and Applied Simulation, Bergeggi, Italy, 3-9.

26. Gendreau, M., Guertin, F., Potvin, J.-Y., Taillard, E.D., (1999), "Parallel tabu search for real-time vehicle routing and dispatching", *Transportation Science*, 33, 381-390.
27. Gendreau, M., Laporte, G., Hertz, A., (1997), "An approximation algorithm for the Traveling Salesman Problem with Backhauls", *Operations Research*, 45(4), 639-641.
28. Ghiani, G., Manni, E., Quaranta, A., Triki, C., (2009), "Anticipatory algorithms for same-day courier dispatching", *Transportation Research Part E*, 45, 96-106.
29. Golden, B.L., Assad, A.A., (1988), "Vehicle routing: Methods and studies", North-Holland, Amsterdam.
30. Hickman, M., Blume, K., (2000), "A method for scheduling integrated transit service", In: 8th International Conference on Computer-Aided Scheduling of public Transport (CASPT), Berlin, Germany.
31. Ichoua, S., Gendreau, M., Potvin, J-Y, (2007), "Planned route optimization for real-time vehicle routing", In: Zeimpekis, V., Tarantilis, C.D., Giaglis, G.M., Minis, I., editors, "Dynamic Fleet Management: Concepts, Systems, Algorithms & Case Studies", *Operations Research/Computer Science Interfaces Series*, Springer,1-18.
32. Ichoua, S., (2001), "Problémes de gestion de flottes de véhicules en temps réel", PhD, Département d'informatique et de recherché opérationnelle, Université de Montréal, Montreal, Canada.
33. Ichoua, S., Gendreau, M., Potvin, J.-Y., (2000), "Diversion issues in real-time vehicle dispatching", *Transportation Science*, 34, 426-438.
34. Ichoua, S., Gendreau, M., Potvin, J.-Y., (2006), "Exploiting knowledge about future demands for real-time vehicle dispatching", *Transportation Science*, 40(2), 211-225.
35. Kerivin, H.L.M., Lacroix, M., Mahjoub, A.R., Quilliot, A., (2008), "The splittable pickup and delivery problem with reloads", *European Journal of Industrial Engineering*, 2(2), 112–133.
36. Laporte, G., (1992), "The vehicle routing problem: An overview of exact and approximate algorithms", *European Journal of Operational Research*, 59, 345-358.

37. Laporte, G., Osman, I.H., (1995), "Routing problems: A bibliography", *Annals of Operations Research*, 61, 227-262.

38. Larsen, A., (2000), "The dynamic vehicle routing problem", PhD thesis, Department of Mathematical Modeling (IMM) at the Technical University of Denmark (DTU).

39. Larsen, A., Madsen, O., Solomon, M., (2002), "Partially dynamic vehicle routing-models and algorithms", *Journal of Operational Research Society*, 53, 637-646.

40. Larsen, A., Madsen, O.B.G., Solomon, M.M., (2007), "Classification of dynamic vehicle routing systems", In: Zeimpekis, V., Giaglis, G.M., Tarantilis, C.D., Minis, I., editors, "Dynamic Fleet Management: Concepts, Systems, Algorithms & Case Studies", *Operations Research/Computer Science Interfaces Series*, Springer, 19-40.

41. Larsen, A., Madsen, O.B.G., Solomon, M.M., (2008), "Recent developments in dynamic vehicle routing systems", In: Golden, B., Raghavan, S., Wasil, E., editors, "The Vehicle Routing Problem: Latest Advances and New Challenges", *Operations Research/Computer Science Interfaces Series*, Springer, New York, *199-218*.

42. Liaw, C., White, C., Bander, J., (1996), "A decision support system for the bimodal dial-a-ride problem", *IEEE Transactions on System, Man and Cybernetics*, 26, 552–565.

43. Lin, S., (1965), "Computer solutions of the traveling salesman problem", *Bell System Technical Journal*, 44, 2245-2269.

44. Madsen, O.B.G., Ravn, H.F., Rygaard, J.M., (1995), "A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives", *Annals of Operations* Research, 60, 193–208.

45. Mitrovic-Minic, S., Krishnamurti, R., Laporte, G., (2004), "Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows", *Transportation Research Part B*, 38, 669-685.

46. Mitrovic-Minic, S., Laporte, G., (2004), "Waiting strategies for the dynamic pickup and delivery problem with time windows", *Transportation Research B*, 38, 635-655.

47. Mitrovic-Minic, S., Laporte, G., (2006), "The pickup and delivery problem with time windows and transshipment", *INFOR*, 44, 217-227.

48. Montemanni, R., Gambardella, L.M., Rizzoli, A.E., Donati, A.V., (2005), "Ant colony system for a dynamic vehicle routing problem", *Journal of Combinatorial Optimization*, 10, 327-343.

49. Mues, C., Pickl, S., (2005), "Transshipment and time windows in vehicle routing", In: Proceedings of the 8th International Symposium on Parallel Architectures, *Algorithms and Networks*, 113–119.

50. Nadarajah, S., (2008), "Collaborative logistics in vehicle routing", PhD thesis, University of Waterloo, Canada.

51. Nakao, Y., Nagamochi, H., (2008), "Worst case analysis for pickup and delivery problems with transfer", *IEICE Tranactions. on Fund. of Electronics, Comm. and Computer Science*, E91-A(9), 2328-2334.

52. Ninikas, G., Minis, I., (2011), "Replanning Strategies for the Vehicle Routing Problem with Dynamic Pickups. A Branch-and-Price approach", *working paper*.

53. Ninikas, G., Athanassopoulos, Th., Zeimpekis, V., Minis, I. (2011), "Routing Problem in Hybrid Courier Operations", *working paper*.

54. Powell, W.B., Jaillet, P., Odoni, A., (1995), "Stochastic and dynamic networks and routing", In Ball, M.O. et al., editor, Network Routing, Handbooks in OR & MS, Elsevier Science, Amsterdam, 141-295.

55. Psaraftis, H.N., (1988), "Dynamic vehicle routing", In: Golden, B.L. and Assad, A.A., editors, "Vehicle routing: Methods and studies", North-Holland, Amsterdam, 223-248.

56. Psaraftis, H.N., (1995), "Dynamic vehicle routing: Status and prospects", *Annals of Operations Research*, 61, 143-164.

57. Regan, A.C., Mahmassani, H.S., Jaillet, P., (1995), "Improving efficiency of commercial vehicle operations using real-time information: Potential uses and assignment strategies", *Transportation Research Record*, 1493, 188-197.

58. Reimann, M., Doerner, K., Hartl, R.F., (2003), "D-Ants: Savings Based Ants divided and conquer the vehicle problem", *Computers & Operations Research*, 31(4) , 563-591.

59. Shang, J.S., Cuff, C.K., (1996), "Multicriteria pickup and delivery problem with transfer opportunity," *Comput. Ind. Eng.*, 30(4), 631–645.

60. Solomon, M.M., (1987), "Algorithms for the vehicle routing and scheduling problem with time window constraints", *Operations Research*, 35, 254-265.

61. Steward, W.R., Golden, B.L., (1983), "Stochastic vehicle routing: A comprehensive approach", *European Journal of Operations Research*, 14, 371-385.

62. Taillard, E.D., (1996), "A heuristic Column Generation method for the heterogeneous fleet VRP", *RAIRO*, 33, 1-14.

63. Tan, C.C.R., Beasley, J.E., (1984), "A heuristic algorithm for the period vehicle routing problem", *Omega*, 12(5), 497-504.

64. Toth, P., Vigo, D., (2002), "The vehicle routing problem", SIAM, Philadelphia.

65. Van Hemert, J.I., La Poutré, J.A., (2004), "Dynamic routing problems with fruitful regions: Models and evolutionary computation", In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J., Tino, P., Kabán, A., Schwefel, H.-P., editors, "Parallel Problem Solving from Nature VIII", Springer, Berlin, 690-699.

66. Van Hemert, J.I., La Poutré, J.A., (2009), "Exploiting fruitful regions in dynamic routing using evolutionary computation", In: Pereira, F.B., Tavares, G., editors, "Bio-inspired algorithms for the vehicle routing problem", Series: Studies in computational intelligence, Springer, 161, 131-149.

67. Zachariadis, E.E., Tarantilis, C.D., Kiranoudis, C.T. (2009), "A Guided Tabu Search for the Vehicle Routing Problem with Two-Dimensional Loading Constraints", *European Journal of Operational Research*, 195(3), 729-743.

APPENDIX A: ANALYTICAL RESULTS

Table A.1: Results for 15 customers with Time Windows

Problem Class	# of Run	Pro.	Insertion Method	Dyn_LES1		Dyn_LES2		Dyn_LES3		Fix_LES4	
			Cost	Cost	% Impr.	Cost	% Impr.	Cost	% Impr.	Cost	% Impr.
R	1	R104	285,1	259,29	9,05%	259,29	9,05%	259,29	9,05%	276,81	2,91%
	2	R110	295,2	264,74	10,32%	264,74	10,32%	264,74	10,32%	277,51	5,99%
	3	R111	333,8	297,59	10,85%	297,59	10,85%	297,59	10,85%	303,98	8,93%
	4	R201	341,1	303,55	11,01%	303,55	11,01%	303,55	11,01%	314,63	7,76%
	5	R203	247,4	220,9	10,71%	220,9	10,71%	220,9	10,71%	235,85	4,67%
	6	R205	269,7	243,58	9,68%	243,58	9,68%	243,58	9,68%	259,96	3,61%
	7	R206	286,8	268,41	6,41%	268,41	6,41%	268,41	6,41%	272,73	4,91%
	8	R210	301,9	248,08	17,83%	246,47	18,36%	248,08	17,83%	253,03	16,19%
C	1	C101	260,4	181,87	30,16%	181,87	30,16%	181,87	30,16%	183,47	29,54%
	2	C102	195,7	146,92	24,93%	146,92	24,93%	146,92	24,93%	164,17	16,11%
	3	C103	302,8	222,35	26,57%	222,35	26,57%	222,35	26,57%	222,35	26,57%
	4	C105	182,8	143,94	21,26%	143,94	21,26%	143,94	21,26%	160,64	12,12%
	5	C106	185,4	149,01	19,63%	149,01	19,63%	149,01	19,63%	165,71	10,62%
	6	C107	219,1	177,5	18,99%	177,5	18,99%	177,5	18,99%	177,89	18,81%
	7	C108	204,3	141,53	30,72%	141,53	30,72%	141,53	30,72%	151,94	25,63%
RC	1	RC103	302,6	233,98	22,68%	233,65	22,79%	233,98	22,68%	254,91	15,76%
	2	RC107	267,9	239,62	10,56%	235,73	12,01%	239,62	10,56%	240,69	10,16%
	3	RC108	302,4	280,61	7,21%	280,61	7,21%	280,61	7,21%	-	-
	4	RC201	313,2	268,84	14,16%	268,84	14,16%	268,84	14,16%	278,57	11,06%
	5	RC202	272,4	245,18	9,99%	245,18	9,99%	245,18	9,99%	262,76	3,54%
	6	RC203	394,2	336,01	14,76%	332,96	15,54%	336,01	14,76%	366,97	6,91%
	7	RC205	395,2	370,01	6,37%	365,88	7,42%	365,88	7,42%	371,54	5,99%
	8	RC206	249,9	219,31	12,24%	219,31	12,24%	219,31	12,24%	231,14	7,51%

Table A.2: Results for 25 customers with Time Windows

Problem Class	# of Run	Insertion Method	Dyn_LES1		Dyn_LES2		Dyn_LES3		Fix_LES4	
		Cost	Cost	% Impr.	Cost	% Impr.	Cost	% Impr.	Cost	% Impr.
R	1	426,2	374,94	12,03%	400,67	5,99%	374,94	12,03%	374,78	12,06%
	2	420,3	343,88	18,18%	364,74	13,22%	343,88	18,18%	344,19	18,11%
	3	507,6	398,82	21,43%	422,21	16,82%	396,89	21,81%	406,83	19,85%
	4	462,3	416,24	9,96%	425,71	7,91%	415,28	10,17%	427,18	7,60%
	5	480	394,81	17,75%	429,12	10,60%	394,81	17,75%	397,76	17,13%
	6	487,2	435,24	10,67%	421,59	13,47%	435,24	10,67%	444,8	8,70%
	7	476	402,19	15,51%	455,25	4,36%	402,19	15,51%	401,6	15,63%
	8	392,7	355,36	9,51%	376,44	4,14%	355,36	9,51%	357,01	9,09%
C	1	224,6	186,89	16,79%	199,83	11,03%	186,89	16,79%	-	-
	2	231,8	184,72	20,31%	191,43	17,42%	182,18	21,41%	193,99	16,31%
	3	240,3	210,72	12,31%	192,75	19,79%	210,72	12,31%	218,83	8,93%

	4	262,4	236,32	9,94%	237,84	9,36%	236,32	9,94%	243,03	7,38%
	5	308,8	269,83	12,62%	239,54	22,43%	254,29	17,65%	271,69	12,02%
	6	291,3	253,46	12,99%	254,66	12,58%	253,46	12,99%	272,29	6,53%
	7	410,8	318,61	22,44%	301,31	26,65%	314,32	23,49%	338,92	17,50%
RC	**1**	557,3	507,52	8,93%	500,9	10,12%	500,52	10,19%	501,2	10,07%
	2	499,8	412,55	17,46%	419,77	16,01%	412,55	17,46%	429,71	14,02%
	3	566,8	459,91	18,86%	518,11	8,59%	459,91	18,86%	482,51	14,87%
	4	499,3	453,84	9,10%	465,93	6,68%	453,84	9,10%	471,55	5,56%
	5	401,8	369,63	8,01%	373,63	7,01%	369,33	8,08%	382,77	4,74%
	6	453	400,18	11,66%	393,92	13,04%	400,18	11,66%	425,63	6,04%

Table A.3: Results for 50 customers with Time Windows

Problem Class	# of Run	Insertion Method	Dyn_LES1		Dyn_LES2		Dyn_LES3		Fix_LES4	
		Cost	Cost	% Impr.	Cost	% Impr.	Cost	% Impr.	Cost	% Impr.
R	**1**	673	601,27	10,66%	609,39	9,45%	601,27	10,66%	604,42	10,19%
	2	670,5	592,65	11,61%	-	-	589,82	12,03%	601,93	10,23%
	3	721,8	621,45	13,90%	-	-	621,45	13,90%	621,15	13,94%
	4	635,1	562,42	11,44%	605,73	4,62%	558,73	12,02%	586,58	7,64%
	5	715,7	612,97	14,35%	623,14	12,93%	612,97	14,35%	613,65	14,26%
	6	659,3	590,45	10,44%	593,04	10,05%	574,39	12,88%	-	-
C	**1**	529	433,66	18,02%	-	-	433,66	18,02%	457,35	13,54%
	2	427,8	324,98	24,03%	347,24	18,83%	324,98	24,03%	-	-
	3	619,4	462,43	25,34%	-	-	462,25	25,37%	467,01	24,60%
	4	523,4	467,65	10,65%	-	-	467,43	10,69%	-	-
	5	566,9	484,18	14,59%	-	-	478,87	15,53%	495,93	12,52%
	6	576,5	450,64	21,83%	-	-	450,64	21,83%	452,71	21,47%
RC	**1**	664,2	596,06	10,26%	-	-	596,06	10,26%	-	-
	2	702,8	627,21	10,76%	679,06	3,38%	616,31	12,31%	-	-
	3	715,6	615,16	14,04%	-	-	615,16	14,04%	634,76	11,30%
	4	790,6	666,46	15,70%	-	-	643,55	18,60%	-	-
	5	666,4	590,01	11,46%	611,96	8,17%	587,73	11,81%	-	-
	6	637,7	579,04	9,20%	584,77	8,30%	577,47	9,44%	602,69	5,49%

Table A.4: Results for 15 customers without Time Windows

Problem Class	# of Run	Insertion Method	Dyn_LES1		Dyn_LES2		Dyn_LES3		Fix_LES4	
		Cost	Cost	% Impr.	Cost	% Impr.	Cost	% Impr.	Cost	% Impr.
R	**1**	260,1	234,66	9,78%	229,84	11,63%	229,84	11,63%	237,56	8,67%
	2	325,5	239,61	26,39%	239,61	26,39%	239,61	26,39%	278,89	14,32%
	3	312,1	231,24	25,91%	228,38	26,82%	231,24	25,91%	236,77	24,14%
	4	332,8	281,11	15,53%	281,11	15,53%	281,11	15,53%	-	-
	5	280	235,47	15,90%	235,47	15,90%	235,47	15,90%	239,37	14,51%
	6	312,7	276,47	11,59%	268,26	14,21%	276,47	11,59%	285,41	8,73%
C	**1**	183,6	154,98	15,59%	154,98	15,59%	153,54	16,37%	158,21	13,83%
	2	210,9	181,89	13,76%	181,89	13,76%	181,89	13,76%	187,19	11,24%
	3	225,3	194,73	13,57%	193,77	13,99%	194,73	13,57%	202,41	10,16%

	4	265,4	186,94	29,56%	186,94	29,56%	186,94	29,56%	181,34	31,67%
	5	215,6	186,6	13,45%	186,6	13,45%	186,6	13,45%	177,7	17,58%
	6	195,9	174,76	10,79%	172,66	11,86%	174,76	10,79%	174,24	11,06%
RC	**1**	338,8	303,64	10,38%	271,29	19,93%	303,64	10,38%	-	-
	2	355,6	301,55	15,20%	298,78	15,98%	301,55	15,20%	341,96	3,84%
	3	366	333,22	8,96%	333,22	8,96%	333,22	8,96%	343,76	6,08%
	4	351,2	303,03	13,72%	303,03	13,72%	303,03	13,72%	317,81	9,51%
	5	311,8	277,93	10,86%	277,93	10,86%	277,93	10,86%	-	-
	6	346,2	308,23	10,97%	301,89	12,80%	308,23	10,97%	313,4	9,47%

Table A.5: Results for 25 customers without Time Windows

Problem Class	# of Run	Insertion Method	Dyn_LES1		Dyn_LES2		Dyn_LES3		Fix_LES4	
		Cost	Cost	% Impr.	Cost	% Impr.	Cost	% Impr.	Cost	% Impr.
R	**1**	350,6	308,92	11,89%	329,29	6,08%	308,92	11,89%	318,61	9,12%
	2	371,3	318,89	14,12%	341,46	8,04%	318,43	14,24%	336,48	9,38%
	3	364	324,13	10,95%	316	13,19%	324,13	10,95%	326,72	10,24%
	4	370,4	301,25	18,67%	301,25	18,67%	301,25	18,67%	312,2	15,71%
	5	380,3	323,7	14,88%	340,68	10,42%	323,66	14,89%	342,71	9,88%
	6	455,8	352,47	22,67%	358,44	21,36%	352,47	22,67%	354,18	22,29%
	7	387,3	322,45	16,74%	337,69	12,81%	322,45	16,74%	324,21	16,29%
C	**1**	288,5	237,17	17,79%	268,57	6,91%	237,17	17,79%	243,28	15,67%
	2	274,6	244,79	10,86%	251,73	8,33%	243,56	11,30%	251,2	8,52%
	3	294,6	249,61	15,27%	271,11	7,97%	249,61	15,27%	252,42	14,32%
	4	283,2	239,3	15,50%	241,82	14,61%	236,16	16,61%	245,39	13,35%
	5	307,2	255,5	16,83%	274,36	10,69%	255,18	16,93%	262,99	14,39%
	6	335,03	253,62	24,30%	253,62	24,30%	253,62	24,30%	271,52	18,96%
RC	**1**	403,7	306,93	23,97%	360,66	10,66%	306,93	23,97%	349,47	13,43%
	2	396,6	323,12	18,53%	360,2	9,18%	323,12	18,53%	331,38	16,44%
	3	364	335,05	7,95%	357,41	1,81%	317,45	12,79%	340,46	6,47%
	4	342,2	294,75	13,87%	318,39	6,96%	294,75	13,87%	331,95	3,00%
	5	418,1	362,01	13,42%	364,93	12,72%	362,01	13,42%	373,61	10,64%
	6	494,5	373,32	24,51%	373,88	24,39%	373,32	24,51%	418,02	15,47%

Table A.6: Results for 50 customers without Time Windows

Problem Class	# of Run	Insertion Method	Dyn_LES1		Dyn_LES2		Dyn_LES3		Fix_LES4	
		Cost	Cost	% Impr.	Cost	% Impr.	Cost	% Impr.	Cost	% Impr.
R	**1**	574,1	500,09	12,89%	502,01	12,56%	498,54	13,16%	500,22	12,87%
	2	535	463,34	13,39%	471,33	11,90%	463,34	13,39%	462,61	13,53%
	3	622,8	506,2	18,72%	536,3	13,89%	505,78	18,79%	527,62	15,28%
	4	555,3	490,09	11,74%	481,36	13,32%	490,09	11,74%	494,78	10,90%
	5	542,2	487,66	10,06%	511,04	5,75%	487,66	10,06%	494,82	8,74%
	6	570,9	504,98	11,55%	546,86	4,21%	504,98	11,55%	508,18	10,99%
C	**1**	489,9	445,15	9,13%	373,26	23,81%	445,15	9,13%	-	-
	2	340,5	286,28	15,92%	299,88	11,93%	286,28	15,92%	298,9	12,22%
	3	338,9	291,71	13,92%	291,75	13,91%	291,05	14,12%	304,3	10,21%

	4	323,6	283,09	12,52%	290,54	10,22%	282	12,86%	299,88	7,33%
	5	373,1	338,16	9,36%	368,32	1,28%	338,16	9,36%	350,24	6,13%
	6	434,8	393,39	9,52%	393,39	9,52%	392,65	9,69%	394,55	9,26%
RC	**1**	604,1	514,65	14,81%	536,94	11,12%	514,65	14,81%	558,65	7,52%
	2	572,9	522,72	8,76%	538,57	5,99%	522,72	8,76%	517,08	9,74%
	3	667,4	542,96	18,65%	597,32	10,50%	542,96	18,65%	-	-
	4	615	521,02	15,28%	520,27	15,40%	521,02	15,28%	527,96	14,15%
	5	553,4	483,42	12,65%	495,04	10,55%	483,42	12,65%	514,29	7,07%
	6	511,8	463,79	9,38%	467,21	8,71%	463,79	9,38%	490,03	4,25%